SOFTWARE CONTROL FOR AN AIRBORNE REAL-TIME

DATA ACQUISITION AND RECORDING SYSTEM

Submitted to University Of Cape Town
Department Of Electrical Engineering
towards partial completion of
Master Of Science in Electrical Engineering degree

NICHOLAS CONSTANTINIDES

April 1987

ABSTRACT

A real-time microprocessor based airborne data acquisition system is described with emphasis on the real-time control software. The system is being currently used by a wide number of airlines in their jetliners for monitoring aircraft performance during flight and recording such data on what is commonly known as the "black box" recorder.

The system was primarily developed for use in the new generation of airliners by Boeing Aircraft Company, namely the Boeing 767 and the Boeing 757. At the time of writing this report however, its use has extended to some European aircraft manufacturers, namely Airbus Industries and Fokker.

The entire project was a team effort, especially in the design of the hardware. The author's part in this project was the complete design, development and implementation of the real-time control software, including some involvement at a systems level in the hardware design.

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks and appreciation to Professor K. Reineck for his assistance and patience. Thanks also to Helen for her assistance in the word-processing and proof-reading of this report.

Dedicated to the memory of Peter John Sacks.

CONTENTS

ABBREVIATIONS

ADC	-	*Analog-to-Digital converter*
AMX	-	*Analog Multiplexer*
bin	-	*Binary*
CPU	-	*Central Processing Unit*
DITS	-	*Digital Information Transfer System*
DFDAU	-	*Digital Flight Data Acquisition Unit*
DFDR	-	*Digital Flight Data Recorder*
DMX	-	*Discrete Multiplexer*
DMEP	-	*Data Management Entry Panel*
FAA	-	*Federal Aviation Administration*
FDEP	-	*Flight Data Entry Panel*
GMT	-	*Greenwich Mean Time*
Hex	-	*Hexadecimal*
IOB	-	*Inverse Offset Binary*
I/O	-	*Input/Output*
I/R	-	*Interrupt*
ms	-	*Milliseconds*
MUX	-	*Multiplexer*
LSB	-	*Least Significant Bit*
LSH	-	*Least Significant Half*
MSB	-	*Most Significant Bit*
MSH	-	*Most Significant Half*
PGA	-	*Programmable Gain Amplifier*
PDL	-	*Program Design Language*
QAR	-	*Quick Access Recorder*
S/H	-	*Sample And Hold*
S/F	-	*Subframe*
W/A	-	*Wraparound*

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Airborne Flight Data Acquisition (FDAU's) Units are currently used by all large commercial (and some military) aircraft to monitor the performance of the aircraft during flight. This is a mandatory requirement for all commercial airlines and is enforced by the civil aviation authority of the country in which the airline is based.

The primary function of the FDAU's is to monitor and record the various aircraft parameters such as engine temperature, turbine pressure ratios (for jets), 3-axis accelerations, speed and angle of attack, on various data recording devices and other storage media for later ground analysis.

As a secondary function, the FDAU's act as a type of caution and warning system. By monitoring a number of critical engine parameters such as engine temperature, oil pressure, turbine rpm and vibration levels, the FDAU can automatically generate "exceedence reports" on cockpit displays and printers whenever any one of a number of important parameters goes beyond a pre-defined critical level. This, of course, is in addition to the primary Aircraft Caution And Warning System (ACAWS), which is an entirely separate aircraft system.

One of these data recorders is the Flight Data Recorder (FDR), usually referred to as the "black box" recorder. This recorder is housed in a specially designed casing, able to withstand extreme conditions such as an aircrash. The FDR can later be retrieved for data analysis resulting to the possible determination of the cause of the aircrash.

Besides recording aircraft parameters onto data recorders, the FDAU can also generate performance reports on various cockpit devices such as display panels and printers, upon request during flight. A typical performance report will contain information such as flight number, flight leg, GMT time, date, and engine

parameters such as temperature, pressure, thrust, vibration levels, fuel consumption rate and airspeed. In addition, engine comparison reports can be generated periodically, whereby the operating parameters of all the aircraft engines are compared and output in the form of a report, on the cockpit printer.

Up to this time, commercial FDAU's processed only analog and discrete type signals. Data from aircraft sensors such as strain-gauges, thermocouples, and flow-meters would be presented to the FDAU or directly to the FDR in an analog form, converted to a digital data value internally and then processed.

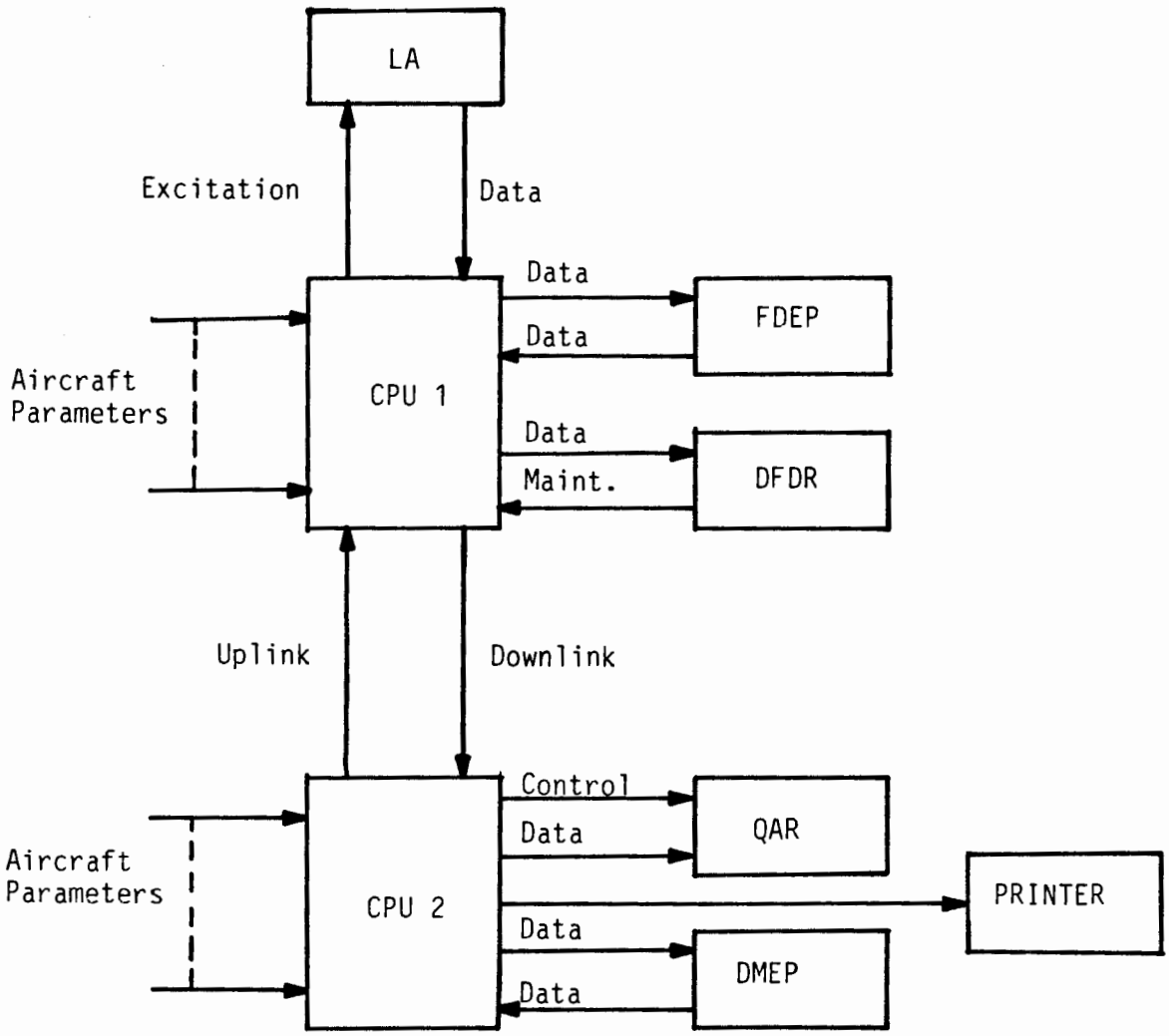
However, due to the electrically noisy environment of the aircraft, it became evident that the next generation of airliners would be have to be "digital", meaning that almost all the data paths within the aircraft would be digital data busses.

With the introduction of the new generation of "digital" airliners by Boeing, the 757 and the 767, a new generation of Flight Data Acquisition units became necessary to be able to process digital data streams as input data. Thus, the Digital Flight Data Acquisition Unit or the DFDAU with its associated Digital Flight Data Recorder or DFDR (Reference 18).

The DFDAU would have to be able to process any type of electrical input signal, whether analog, digital or discrete, and in addition, have the facility of expandability so that the specific requirements of a particular airline can be satisfied.

1.2 THE DFDAU - A BROAD DESCRIPTION

Figure 1.1 shows a basic block diagram of a Digital Flight Data Acquisition Unit.



LA = Linear Accelerometer
DFDR = Digital Data Flight Recorder
FDEP = Flight Data Entry Panel
QAR = Quick Access Recorder
DMEP = Data Management Entry Panel

Figure 1.1 Digital Flight Data Acquisition Unit

The DFDAU is divided into two major functional blocks. The mandatory section and the non-mandatory section.

The mandatory section consists simply of a CPU, a Digital Flight Data Recorder and a control panel which can be incorporated into a Flight Data Entry Panel (FDEP). These basic components are mandatory for every airline, based on regulation laid out by agreement of governing authorities of all commercial carriers.

The linear accelerometer is a device sensitive to the three-axis acceleration of the aircraft and is one of the primary sensors. All mandatory data acquisition and recording functions are controlled by the CPU 1 processor (Reference 1).

The non-mandatory section performs similar data gathering and recording functions but is not a mandatory requirement and therefore its exact configuration depends on the requirements of the particular airline. Its basic function however is to sample and record non-mandatory parameters (whichever the particular airline decides upon), and to generate the various "exceedence" and other types of performance reports. The acquisition and recording of these non-mandatory parameters is controlled by the CPU 2 processor. This processor works interactively with the crew via a DMEP (Data Management Entry Panel). The DMEP enables the crew to be able to call up any aircraft parameter and display its value on the DMEP screen or on a cockpit printer. The generation of performance reports can also be initiated via the DMEP.

The two functional blocks of the DFDAU are linked via a serial asynchronous RS422 communication link via the two processors. Thus, all data acquired by the mandatory CPU can be made available to the non-mandatory CPU.

1.3 THE PROBLEM DEFINITION

The fundamental objective of this project was to design and implement an interrupt driven real-time software system to drive and control all of the data acquisition and recording functions of the DFDAU.

The software design would be based on a set of requirements and specifications (described in detail in chapter 3) laid out by Boeing Aircraft Company, the Federal Aviation Administration (the civil aviation authority of the United States), Aeronautical Radio Incorporated (see appendix A) and the particular airline concerned.

The software design had to be structured around a priority interrupt system. This enables the processor to execute routine tasks and application programs, as well as service the various demand functions such as reading the value of a parameter whenever the acquisition hardware signals to the processor that

the parameter is available, by means of an interrupt.

A further design consideration of the software would to adhere to the Boeing Aircraft Company's software design standards (reference 12), particularly in the aspect modularity and partitioning, whereby it would be required that each program module (subroutine) performs only a single function, and that the whole software system can be broken down into a number of independent functional elements.

1.4 OBJECTIVES

This thesis sets out to describe the software system that was developed to control all the data acquisitions and recording functions of the DFDAU. However, in order to do this in a more clear fashion, a fair amount of attention has to be given to the hardware operation of the unit, including a description of the operation and electronic design of each of the hardware modules.

It should be noted that beyond the "minimum system" which is mandatory for all commercial airlines, the requirements of each airline will vary from one to another and no particular airline will utilize all the possible functions of the DFDAU.

More specifically, this thesis concentrates mainly on the mandatory functions of the DFDAU with only some reference to the non-mandatory functions.

Although this thesis is based on a design for the new British Airways 767 jetliners, the report will go a little beyond those particular requirements, so that most if not all of the mandatory functions of the system can be described.

CHAPTER 2

SYSTEM LEVEL DESCRIPTION

2.1 INTRODUCTION

This section looks at the DFDAU from a system level point of view and provides a description of each "block" with reference to the "minimum system" necessary to accommodate mandatory flight data recording and other data acquisition needs, as well as to the "expanded", twin-processor system.

The differences between the "expanded" and the "minimum" system configurations, are described with reference to their main system components, namely the DFDR, the FDEP and the Linear Accelerometer. These are in turn described in some detail.

The last section in this chapter deals with the three main types of input signals that can be processed by the DFDAU, namely analog, digital and discrete, with reference to their format, range of values and accuracy requirements.

2.2 THE MINIMUM SYSTEM

The minimum system required by all airlines is a single processor system with input channels connected to the aircraft sensors for the mandatory parameters, the linear accelerometer, and an output channel for the Digital Flight Data Recorder (DFDR) (Reference 1, 18).

A block diagram is given in figure 2.1

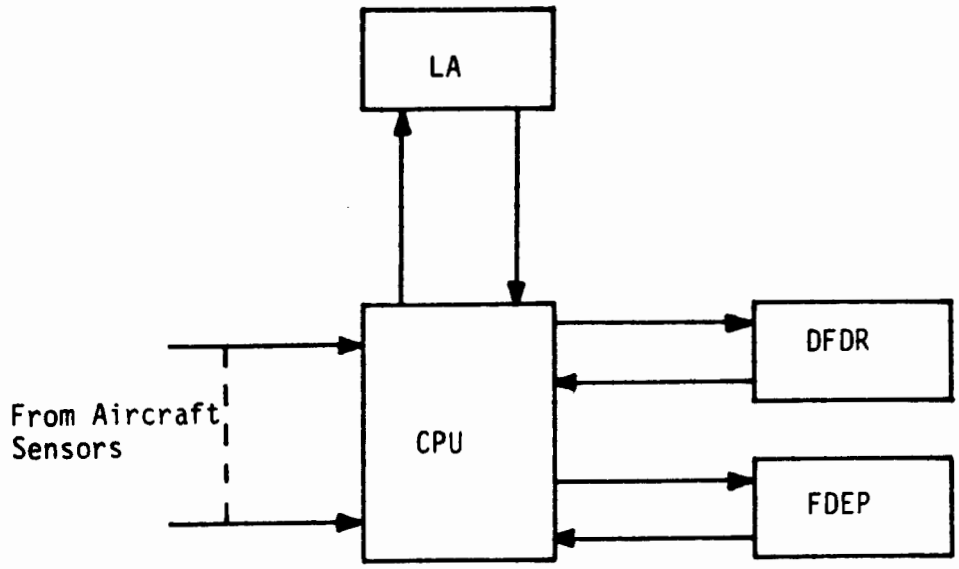


Figure 2.1 The Minimum System

An optional cockpit Flight Data Entry Panel (FDEP) may be included for entering documentary data such as crew and flight identification data. The FDEP can then also be used to incorporate the mandatory control panel, which must be mounted in the flight deck to provide the necessary switches and indicators for pre-flight checks and system tests.

2.3 EXPANDED SYSTEM

The "minimum system" may be combined with other modules and devices to form a highly specialized data acquisition and recording system.

The expanded system may for instance, include a second processor with additional input ports to acquire additional aircraft parameters, and additional output ports for Quick Access Recorders (QAR's), expanded data display/entry panels etc.

The expanded system is shown in figure 1.1

2.4 THE DIGITAL FLIGHT DATA RECORDER (DFDR)

The DFDR, or otherwise known as the "black box" recorder, is the main recording device of the DFDAU. It is simply a tape recording unit, housed in a casing, especially designed to withstand extreme temperatures and shocks. In other words, it is designed to be able to survive an aircrash so that it can later be retrieved, the data analyzed, and possibly assist in the determination of the cause of the aircrash. The casing of the DFDR is painted a bright orange colour (not black) so that it can be easily identified and located.

In addition, an underwater locator transmitter is mounted on the front panel of the recorder, and is activated on impact or upon contact with water. This enables the recorder to be located even if the aircraft crashes into the sea.

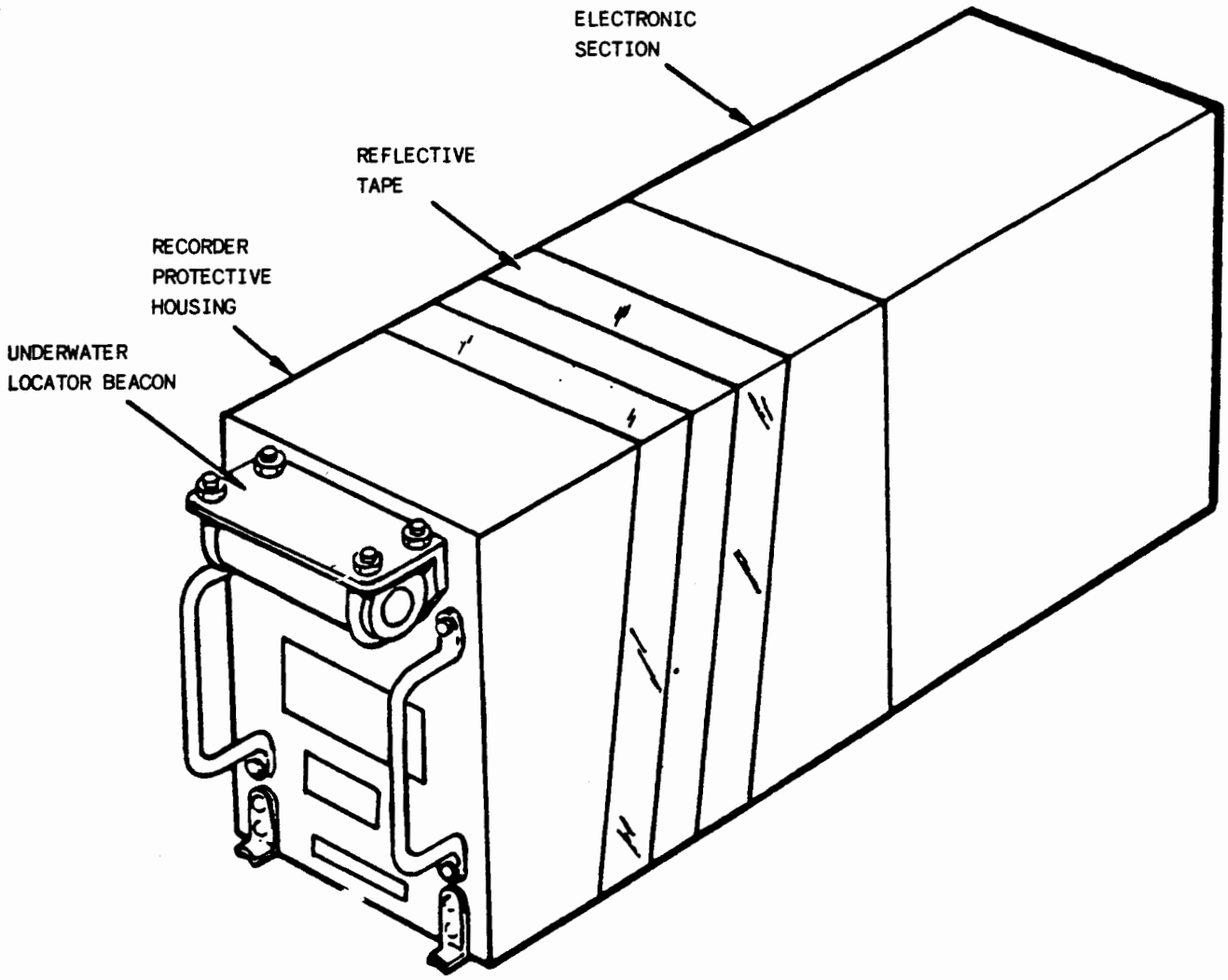


Figure 2.2 Digital Flight Data Recorder (DFDR)

The DFDR is designed to provide a minimum of 25 hours of continuous recording and retaining capability. The data is sent by the DFDAU to the DFDR at a rate of 64, 12-bit words per second. The data transmission takes place in Harvard Bi-phase format via a shielded twisted pair as shown in figure 2.3

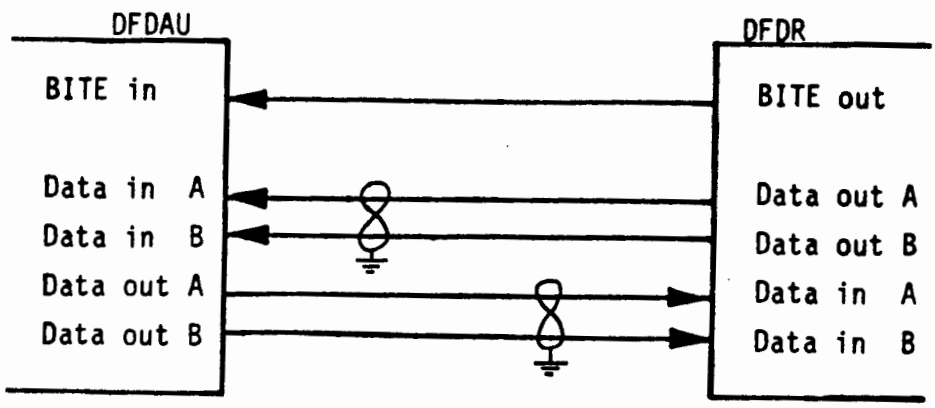


Figure 2.3 DFDAU-DFDR Signal Connections

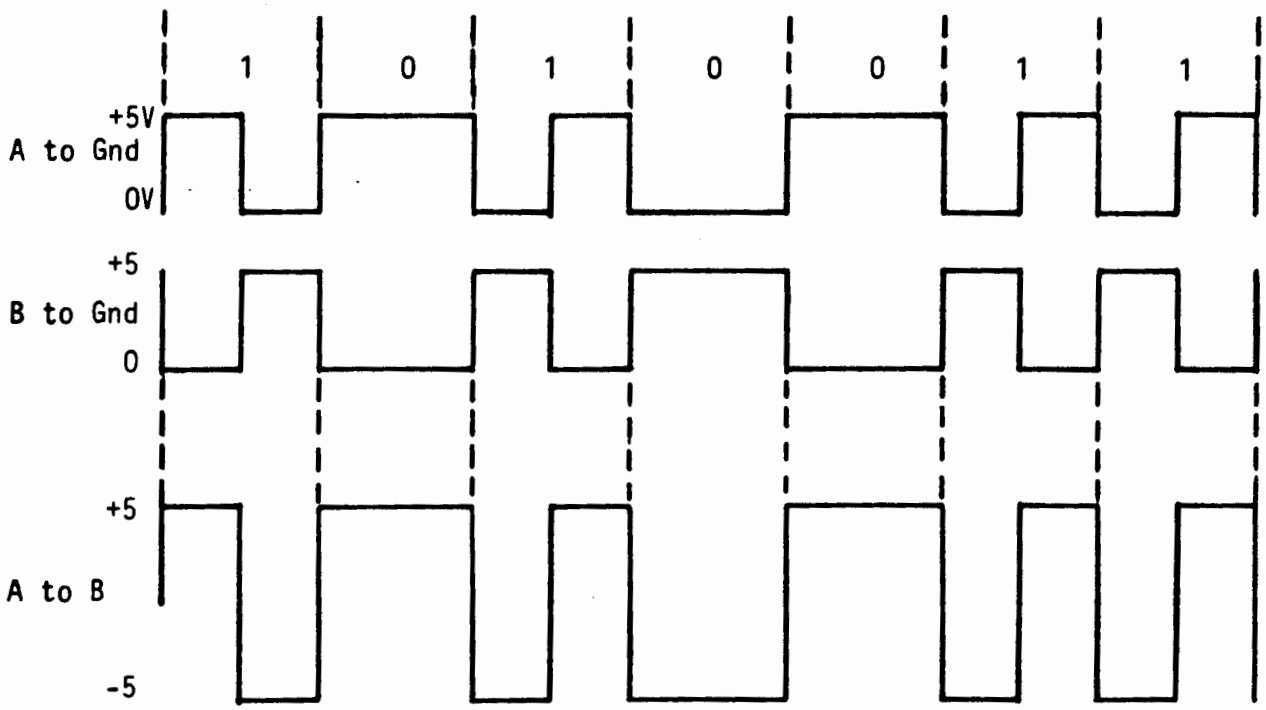


Figure 2.4 Harvard Bi-phase

The recorded data is organized into a "frame" which is repeated once every 4 seconds. Each frame in turn consists of 4 "subframes" which occupy 1 second each. Each subframe consists of 64, 12-bit words. The first word in each subframe consists of a frame synchronization pattern.

The pattern consists of various configurations of the Barker Code and includes identification of each subframe. Barker Codes are various type of optimum synchronization bit patterns used in telemetry for synchronizing multiple data frames (reference 15). The octal codes for the synchronization patterns for subframes 1 to 4 are 1107, 2670, 5107 and 6670.

This is shown in figure 2.5

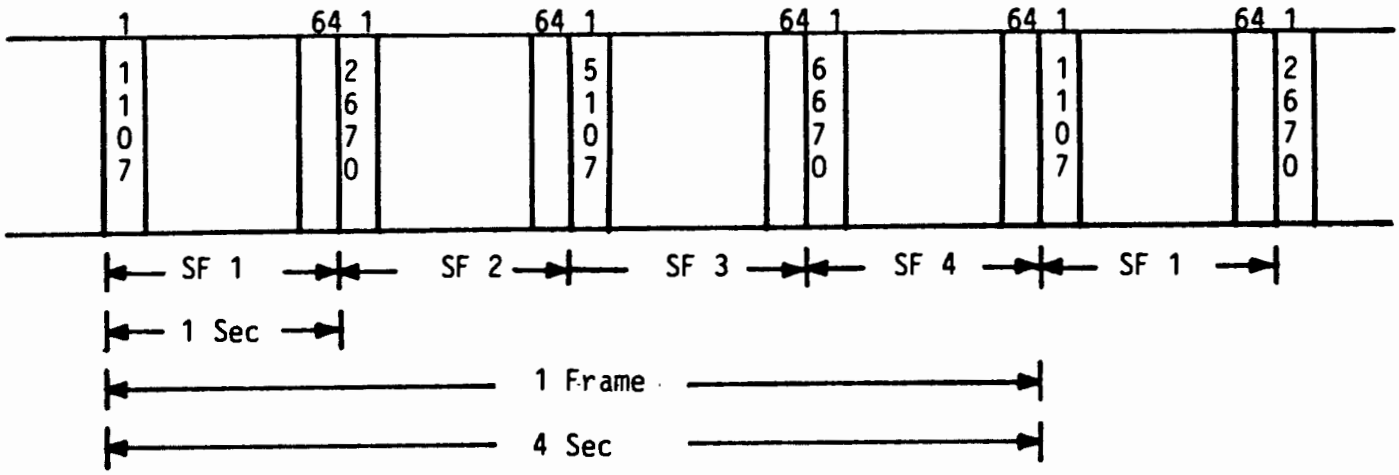


Figure 2.5 Data Organization On The DFDR

Some DFDR models are also capable of providing a playback function, so that the data received can be played back to the DFDAU, where it can be checked by the DFDAU software for accuracy.

Furthermore, the DFDR also provides two status output signals, namely the DFDR Maintenance Flags, one to the DFDAU in the form of a discrete type parameter and one to the FDEP for fault annunciation.

2.5 FLIGHT DATA ENTRY PANEL (FDEP)

The optional FDEP may be used for the manual entry of certain documentary data from the cockpit. Typical of such data are date, flight number, flight leg, and pilot identification.

The FDEP consists of an alphanumeric keyboard, as well as a set of special function "soft" keys, whose function can be programmed to satisfy the requirements of the particular airline, and a 48 character alphanumeric LED display. See figure 2.6

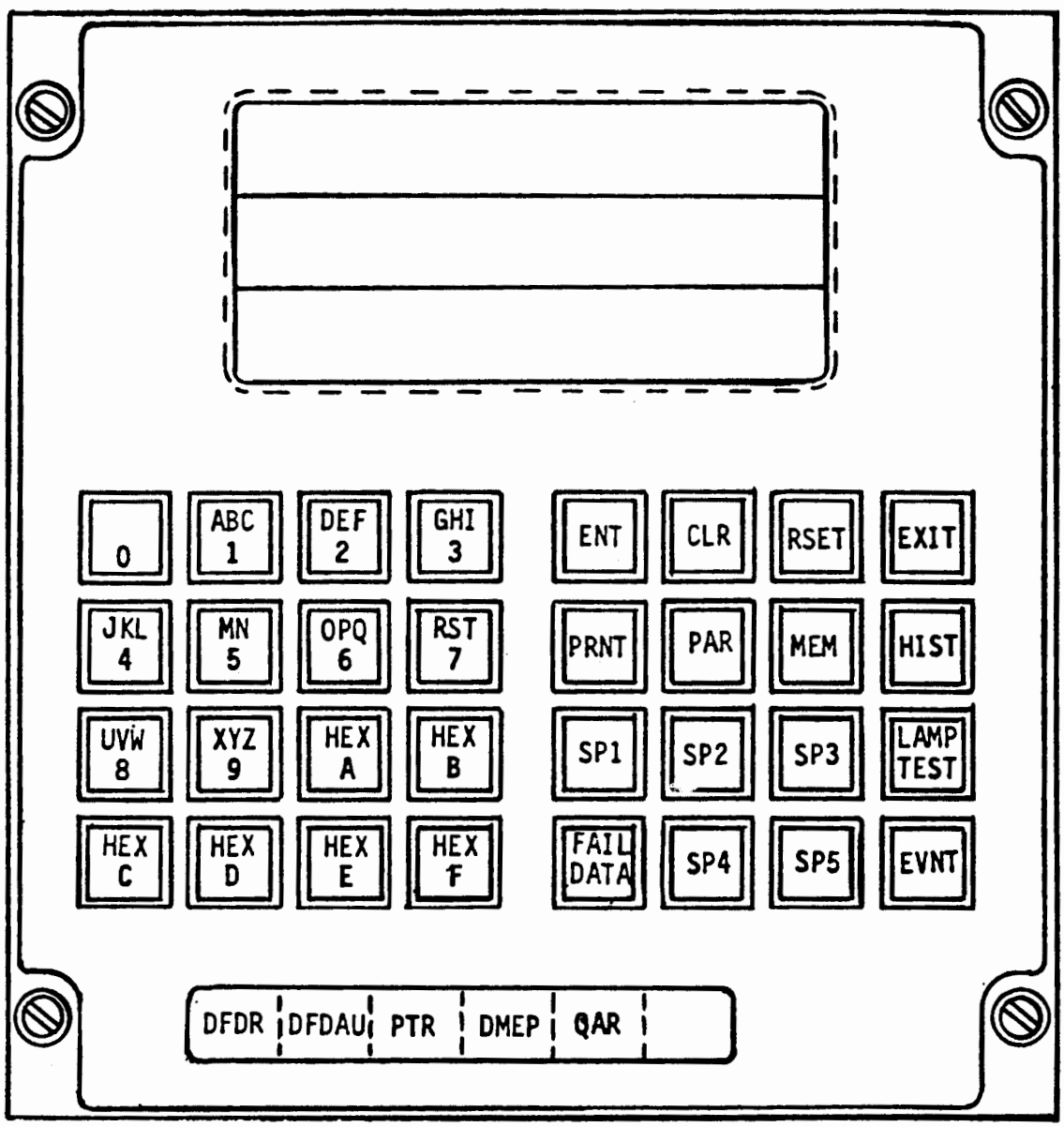


Figure 2.6 Flight Data Entry Panel

An FDEP may also be used to provide pre-flight system tests, fault annunciation, trouble shooting assistance and data display.

The FDEP is connected to the CPU via an asynchronous RS422 interface, with system status signals from the DFDR and the DFDAU.

2.6 THREE-AXIS LINEAR ACCELEROMETER

The accelerometer is an instrument sensitive to acceleration in all three axes of motion of the aircraft. It provides acceleration information required for flight recording and is a primary data source of the DFDAU.

The standard accelerometer is a pressure sealed instrument, which is designed to withstand excessive accelerations of up to 10 times full scale values.

It is basically a force-balanced, closed loop, servo-pendulum design. Three identical complete accelerometer units are included within a common case. Each of the three accelerometer units are identical in function, and differ only in their orientation within the case.

To explain the operation of the accelerometer, a single-axis system will be considered as shown in figure 2.7.

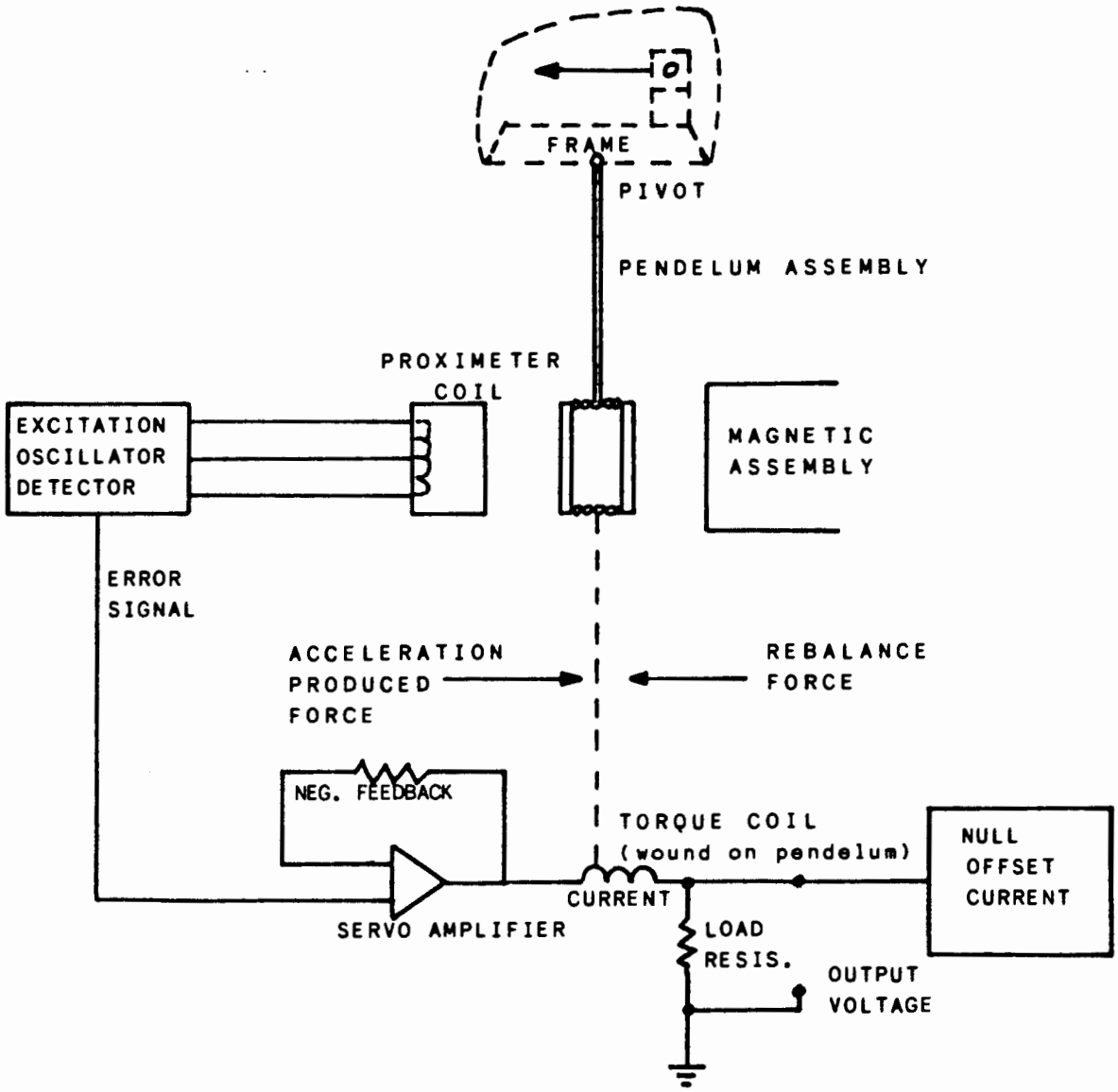


Figure 2.7 Single-Axis Linear Accelerometer

The basic elements of the accelerometer are the acceleration sensing mechanism (including the frame, pendulum and torque coil), the proximeter coil, servo amplifier, output load resistor, null offset generation circuits and filters.

The servo concept balances the input force of acceleration with an equal and opposite electrically generated restoring force. The measuring of this restoring force provides the desired output

signal in terms of output volts per applied unit of acceleration.

When the accelerometer mechanism is accelerated along the sensitive axis, a minute displacement of the pendulum occurs. This displacement is sensed by the proximeter (motion detector), which generates an output error signal. The amplitude of this error signal is directly proportional to the displacement magnitude. The error signal serves as the input to the servo amplifier, whose corresponding output current flows through the torque (rebalance) coil and load resistor.

The current flowing through the torque coil generates an equal and opposite restoring force applied to the pendulum. Any change in input acceleration produces a corresponding change in restoring force, and the pendulum is thereby maintained in a fixed position-captured mode.

A load resistor is placed in the output current loop, and the voltage drop across the resistor is an accurate electrical analog of the input acceleration.

2.7 INPUT DATA TYPES

The DFDAU can accept three basic types of input signals, namely analog, digital and discrete (on-off). Each of these types of signals adheres to certain standards established by the ARINC Characteristic 717 (appendix A, reference 1) to ensure interchangeability of wiring, and thus the interchangeability of equipment.

2.7.1 Analog Input Signals

All analog input signals to the DFDAU are fed via an analog-to-digital converter, whose resolution is 1/4096 of full range or better. The following types of analog data are accepted by the DFDAU:

Synchro Signals

Range	:	0 to 360 degrees continuous unlimited rotations.
Voltage	:	11.8 VAC line to line at 26 VAC ref.
Reference	:	26 VAC 400 Hz.
Scaling	:	0 degrees = zero count 360 degrees = max count + 1 (all zeros)
Accuracy	:	+/- 0.17% (0.6 degrees)

AC Voltage Ratio 1

Range	:	0 - 5 VAC in and out of phase referenced to 26 VAC.
Scaling	:	5 VAC out of phase = zero count 5 VAC in phase = full count
Accuracy	:	+/- 0.2% (20 mVAC)

AC Voltage Ratio 2

Range	:	As for AC Ratio 1 but 0 - 26 VAC
Accuracy	:	+/- 0.2% (104 mVAC)

DC Voltage Absolute

Range	:	0 to 5 VDC
Scaling	:	0 VDC = zero count 5 VDC = full count
Accuracy	:	+/- 0.2% (10 mVDC)

DC Voltage Ratio 1 and 2 (3-wire input)

Range	:	0 to 5 VDC
Reference	:	5 VDC or 32 VDC
Scaling	:	0 VDC = zero count 5 VDC = full count
Accuracy	:	+/- 0.2% (10 mVDC)

Potentiometer

Range	:	0 to Excitation
Excitation	:	5 VDC (Internal)
Scaling	:	0 VDC = zero count excitation = full count
Accuracy	:	+/- 0.2% (10 mVDC)

Thermocouple

Type	:	Chrome/Alumel
Range	:	0 to 35 mVDC

Internal cold junction compensation.

2.7.2 Digital Input Signals

The digital data input signals are all in serial form with data words transmitted at intervals, and their format adheres to the ARINC 429 Digital Input Transfer System (DITS) specifications. (appendix A, reference 2).

The basic information element is a digital word containing 32 bits, and within this 32-bit word, there are three possible types of data, ie. binary, BCD, and discrete data.

The type of information contained within this 32-bit word is identified by a three character label. These are octal characters coded in binary in the first 8 (bits 0 to 7) bits of the word.

Bits 8 and 9 form what is called the Source/Destination Identifier (SDI), and are generally considered as part of the label.

Bit 28 forms the sign bit for binary data, 0 meaning positive and 1 meaning negative.

Bits 29 and 30 form the Sign/Status Matrix (SSM) and are used to determine the sign of BCD data or the status of binary data, according to the following table:

30	29	BCD Data	Binary Data
0	0	+	Invalid Data
0	1	Invalid Data	Invalid Data
1	0	Invalid Data	Data OK
1	1	-	Data OK

Table 2.1 DITS SSM Matrix

Bit 31 forms the parity bit of the 32-bit word, and is generally set to odd parity.

2.7.3 Discrete Input Signals

These signals are individual binary on-off type signals and are generally used to "flag" the existence of certain states or conditions, e.g landing gear up/down, engine start, flaps extended or the position of the various aircraft switches.

Discrete signals are divided into four types, namely series discretes, shunt discretes, AC discretes and Marker Beacon discretes, all of which are described in chapter 4.

Another class of discretes, the Ident discretes, are hard-wired and are used for identification purposes for the aircraft type, airline, engine configuration and fleet number. The DFDAU can then determine the type of aircraft it is installed in, simply by reading these discretes.

CHAPTER 3

SYSTEM SPECIFICATIONS AND REQUIREMENTS

3.1 INTRODUCTION

This chapter outlines the system requirements and specifications for the DFDAU. It focusses mainly on the software requirements, as the detailed hardware requirements and specifications for the DFDAU are clearly outlined in ARINC Characteristic 717-4 (Appendix A, Reference 1). However, a general hardware outline of the DFDAU hardware specifications and requirements is given in this chapter to correlate with the hardware module descriptions in chapter 4.

The hardware configuration for the DFDAU is generally the same for all users of the system, whereas the actual usage thereof, with regards to the parameters to be acquired, the DFDR word slots, DFDR frame formats and I/O port configurations, depends on the specific user (airline) concerned, and hence the software.

3.2 HARDWARE SPECIFICATIONS AND REQUIREMENTS

The DFDAU hardware should contain the necessary circuitry and signal conditioning capable of accepting all of the signal types outlined in section 2.7, and convert the data to 12-bit form with the specified accuracies. As a minimum, it should contain the circuitry needed for the following functions:

- *Input signal isolation and signal conditioning for the signals described in section 2.7.*
- *Analog to digital conversion.*
- *DFDAU control logic and data frame formatting.*
- *Data output drivers.*
- *Excitation and reference signal sources for the sensors.*
- *BITE (Built-In Test Equipment) and self calibration circuitry.*
- *Power supply.*

The data output sections should be capable of driving a DFDR as described in section 2.4 at a rate of 64, 12-bit words per second, at a bit rate of 768 bits per second, in Harvard bi-phase format (figure 4.18). In addition, the data output section should be capable of driving an additional output device (Auxiliary Output) at rates of 64, 128 or 256 12-bit words per second, in bipolar RZ format (figure 4.22). This output is used to connect an optional cassette type quick access recorder (QAR) whose data frame is at a minimum, a duplicate of the DFDR data frame, and is used to ensure the integrity of flight recording. QAR's are available in 64, 128 or 256 words per subframe formats.

The BITE circuitry should be able to perform certain self-tests on the DFDAU, generate a number of status conditions and output these to the front panel displays. These self-tests are generally defined by the user and are dealt with in the next section and in chapter 5.

The power supply should, in addition to supplying all the voltage levels required by the DFDAU, be capable of providing power to the RAM for a period of 200 milliseconds after DFDAU power has been lost. It should also have sufficient capacity so that power interrupts of greater than 10 milliseconds and less than 200 milliseconds will not affect data output or processor operation.

3.3 SOFTWARE SPECIFICATIONS AND REQUIREMENTS

The mandatory data acquisition and recording functions of the DFDAU will be under the control of the CPU 1 microprocessor.

The system will acquire all the parameters listed in Appendix C (Parameter Input List). These parameters will be of various signal types, such as analog digital and discrete, as described in section 2.7.

All input parameters will be formatted to 10, 11 or 12 bit digital words, suitable for recording on the DFDR. In the cases where 10 or 11 bits are required to represent the parameters, the least significant 1 or 2 bits (bit 1 or bit 2) will be used to represent discrete input values. Discrete values from the DITS input channels will be "packed" into 12 bit data words.

All analog inputs are to be sequentially sampled, signal conditioned and converted to digital values. Table C-1 of appendix C lists all the analog parameters to be processed. It indicates the input port number, the analog channel used and the signal type. It also shows the output word length and word position in the DFDR data stream.

The discrete inputs will be sequentially sampled, and placed into bit 1 and bit 2 positions of selected output words of the DFDR. Table C-2 of appendix C lists all the discrete parameters to be processed, their DFDR output word, bit position and the signal type.

The DFDAU will acquire digital data (DITS) simultaneously from all digital input channels. Table C-3 of appendix C lists all the DITS parameters to be processed. It indicates the speed selection required, the digital channel used, the parameter label and SDI code, and the position of the "useful data" field in the 32-bit word.

The DFDAU will output to the DFDR a continuous data stream. This data is organized into a data frame which is repeated every 4 seconds. The data frame is made up of four subframes, each consisting of 64, 12-bit words, of which the first word will be a unique synchronization code (section 2.3). The synchronization codes (octal) for subframes 1 to 4 are 1107, 2670, 5107 and 6670 respectively. Appendix D contains the specific output word assignments for the DFDR data frame.

Data to the DFDR will be continuous under normal operation, and will begin at most, 500 milliseconds after power-on initialization.

Power interrupts of greater than 10 but less than 200 milliseconds will not affect the contents of the RAM but the processor operation and data output will halt. After the resumption of power, a 250 millisecond delay will allow the acquisition of new data. The data output will resume with the next subframe, from where the interruption occurred. If power is interrupted for more than 200 milliseconds, the contents of the RAM will be lost and a new power-on initialization has to be performed. Data output will resume with a new data frame.

All data values to the DFDR have to be acquired no later than 250 milliseconds prior to their output.

The Aircraft Type Identification discretes shown in the table below, will be monitored by the DFDAU during power-on initialization, to establish the type of aircraft the DFDAU is installed in, and the type of engine that is used on the aircraft.

Discrete Port			Aircraft/Engine Type
84	83	82	
		X	B767 with PW Engines
	X		B767 with GE Engines
X			B757 with GE Engines

"X" - denotes discrete state 1
PW - Pratt and Whittney Engines
GE - General Electric Engines

Table 3.1 Aircraft Type Identification Discretes

All parameters acquired by CPU 1 are to be downlinked to CPU 2 via an RS422 serial communication link, along with results of all self testing done by CPU 1. The downlinking will be performed once per second at a 9600 baud rate.

Data from CPU 2 to CPU 1 will consist of documentary data acquired by CPU 2 and results of self testing done by CPU 2. Transmissions from CPU 2 to CPU 1 will take place only when self test results in CPU 2 change, or when new documentary data has been input via the CPU 2 data entry panel. CPU 2 will then uplink the data at a 1200 baud rate.

3.3.1 Data Scaling And Formatting

All analog data received will be converted to a 14 bit digital value by the analog-to-digital converter. The full range of this converter is from -10VDC to +10VDC. As all analog inputs are conditioned to a 0 to +5VDC range, the processor will rescale the 14 bits to 12 bits by discarding the most significant 2 bits. In the cases where only 10 bits are allocated for the output word, the least significant bit (bit 1) will be discarded and bit 2 will be used to round off bits 3 through 12. Where only 11 bits are allocated for the output word, the least significant bit (bit 1) will be used to round off bits 2 through 12.

In the case of variable digital input data (DITS), the data stream will be examined for correct parity. If a parity error is detected, the corresponding data value will be set to full scale (all ones data field) and output to the recorders.

DITS data (except for BCD types) will also be examined for status using bits 29 and 30, and the table below:

Bit 30	Bit 29	Status
0	0	Failure
0	1	No computed data
1	0	Valid Data
1	1	Valid Data

Table 3.2 DITS Status Codes

If the status bits indicate that the data is not valid, the data value will be set to full scale minus 1 bit (all ones data field except for the LSB which is set to zero), and output to the recorder.

Where a sign bit is required, it will be placed in bit 12 of the output word.

3.3.2 Special Data Generation For The DFDR

3.3.2.1 Frame Counter -

To allow detection, during ground processing of recorded data, of short time data loss due to recorder or synchronization problems, the output data frame to the DFDR will contain a frame counter. This counter will be cleared during initial power-on and allowed to increment by one for each data output frame. The counter will overflow and restart after reaching a binary count of 4095 (12 bits). This counter will be output in word 64 of subframe 1 of the DFDR data frame.

3.3.2.2 Superframe Data -

To accommodate the recording of DFDAU generated data words, a superframe format is mechanized. The superframe is 16 frames long and is synchronized to the frame counter. Thus, frame count 0, 16, 32, 48 and 64 designates the beginning of a superframe. The superframe data is located in word 64 of subframes 2, 3 and 4 and these data words are as follows:

System Status Data	*Subframe 2*
Documentary Data	*Subframe 3*
Calibration Data	*Subframe 4*

Each of these data sets will be stored in a 16 word RAM buffer and one word from each of these buffers will be output during

each data frame. Therefore after 16 data frames (1 superframe), the whole 16 word buffer will have been output to the DFDR and the process will start again. In the meantime however, the software will be updating these buffers with new data, if any change has occurred.

The system status word consists of status codes generated by the DFDAU self-tests. These are shown in table 3.5.

The calibration word contains the digital value of the calibration voltages used to calibrate the analog input channels, and are as follows:

Signal	Value(Octal)	Tolerance(%)
Ground Reference	0000	5
2.5 VDC Cal	2000	10
5 VDC Cal	4000	10
25 VDC Cal	6000	10

Table 3.3 Calibration Values

The documentary data received from CPU 2 or from a parameter input will be recorded in word 64 of subframe 3. These are shown in table D-2 of appendix D.

3.3.3 System Self Tests

A number of self tests are to be carried out by the DFDAU and when necessary, if the tests fail, status codes will be generated, stored and displayed upon request. The status codes are to be stored in a RAM buffer and also in non-volatile memory (EAROM).

3.3.3.1 DFDR Self-Test -

These tests are carried out in the DFDR and are not under the control of the DFDAU. The results of these tests are made available to the DFDAU via the DFDR Maintenance Flag discrete input. A "0" state input indicates correct operation and a "1" state input indicates a failure. A fault condition will illuminate the "DFDR FAIL" indicator on the front panel of the DFDAU, and the resulting status code will be included in the system status superframe buffer, and the EAROM.

3.3.3.2 DFDAU Self Tests -

The self tests done on the DFDAU fall into two categories. Failure of tests performed on the power supply voltages, data output to the DFDR, or the contents of the CPU 1 program memory (EPROM) will result in the "DFDAU FAIL" indicator being turned on. Failure of CPU 2 program memory (EPROM), inter-cpu communication failure or any individual analog or digital input circuits, will result in the "DFDAU CAUTION" indicator being turned on. In addition, the corresponding status codes generated will be stored in the system status word and the EAROM.

3.3.3.2.1 DFDAU Failure -

All power supply voltage levels (BITE 1 thru 6) are tested for accuracy according to table 3.4 below. If any of these exceed the tolerances shown, the corresponding status code will be generated and the "DFDAU FAIL" indicator will be turned on at the front panel.

BITE NO.	Voltage Tested	Nominal	Minimum	Maximum
1	+12,+5V,-15	2.88	2.09	3.72
2	+28V Excit	4.96	4.464	5.456
3	+5V Pot Excit	5.0	4.0	6.0
4	+15V,-5V	2.88	2.04	3.77
5	-33,28V	2.5	0.0	5.45
6	Ground	0	-1.0	+1.0

BITE = Built In Test Equipment

Table 3.4 Power Supply BITE Limits

The DFDR output data will be continuously monitored via the DFDR wraparound circuitry. In this case, a word-by-word comparison of the data in the output buffer to the data from the output of the line driver. An error of more than 4 consecutive seconds (256 failures) generates a status code and turns on the "DFDAU FAIL" indicator. The wraparound arrangement is shown below in figure 3.19.

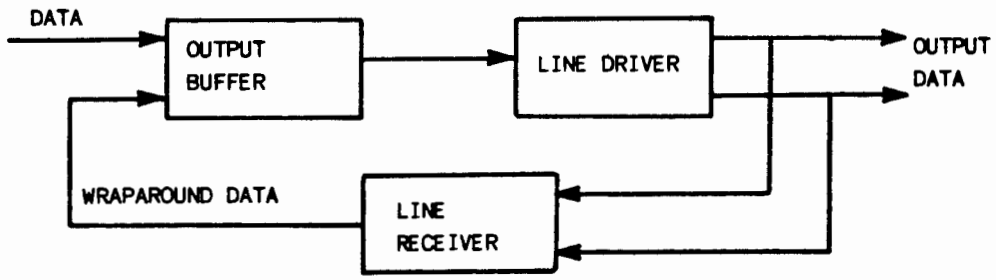


Figure 3.19 DFDR Wraparound Configuration

The DFDAU software will perform a periodic sumcheck on all EPROM locations of CPU 1 where the mandatory operational software resides. Any failure of this sumcheck will generate the corresponding status code and turn on the "DFDAU FAIL" indicator.

3.3.3.2.2 DFDAU Caution -

The analog calibration values for the ground and calibration voltages of each analog channel will be checked. If the ground voltage (ideally zero volts) changes by more than 5%, or if any of the calibration voltages changes by more than 10%, that channel will be declared as having failed. The corresponding status code will be set, and the "DFDAU CAUTION" indicator on the front panel will be turned on.

During the power-on initialization stage, all digital input channels (DITS) will be self-tested. This will be done by the processor sending pre-determined 32-bit data patterns to the input of each digital input port and testing the received data. The failure of any channel will cause the appropriate status code to be set and the "DFDAU CAUTION" indicator to be turned on.

3.3.3.2.3 Status Code Display And Recording -

Up to sixteen unique status codes have to be able to be displayed

on the front panel status display and recorded in RAM. A 16-word System Status buffer will accumulate these codes. When the "READ" switch on the front panel is depressed and held, the status display will sequentially indicate all the status codes, each for 4 seconds. A display of "000" will indicate a non-failure.

Similarly, the status codes will be sequentially recorded in each superframe. All unused words within the superframe will be set to zero.

Table 3.5 below show the system status error codes

DESCRIPTION	STATUS CODE (HEX)	FRONT PANEL LED DISPLAY
DFDR BITE	001	DFDR FAIL
CPU1 EPROM SUMCHECK	101	DFDAU FAIL
DFDR W/A	102	DFDAU FAIL
POWER SUPPLY BITE	103 - 108	DFDAU FAIL
ANALOG CALIBRATION	201 - 216	DFDAU CAUT.
DITS 1 W/A (Chan 0-7)	401 - 408	DFDAU CAUT.
DITS 2 W/A (Chan 0-7)	409 - 410	DFDAU CAUT.
CPU1/CPU2 Xmit Fail	501	DFDAU CAUT.
CPU2 EPROM SUMCHECK	901	DFDAU CAUT.
CPU2/CPU1 Xmit Fail	A01	None
CPU2 DMEP Fail	A02	None
CPU2 QAR Fail	A03	None
CPU2 PRINTER Fail	A04	None

Table 3.5 System Status Codes

3.3.3.3 DFDAU Lamp Test -

Any time the front panel "READ" switch is depressed, all the

front panel indicators will be turned on, and the status display will indicate "FFF" for 1 second and then "888". After 4 seconds, the indicators will be turned off and the display will indicate "000", unless a specific failure has occurred.

CHAPTER 4

HARDWARE DESCRIPTION

4.1 INTRODUCTION

Although the purpose of this thesis is to describe the software aspects of the DFDAU, it would be incomplete without some reference to the operation and design philosophy of the hardware.

This chapter therefore, sets out to describe each hardware module of the DFDAU, at a functional level, with reference to the block diagrams of the various hardware modules.

In addition, a description of the hardware will help to clarify the techniques used in the software functions directly related to the hardware, such as device drivers and interrupt service routines.

A section is also included to justify the choice of microprocessor used as the central processor in the DFDAU, the TMS9900, in relation to its internal architecture, operating speed and other important characteristics.

A number of sections in this chapter have been dedicated to describing some of the TMS9900 family components in detail, which could have been included in an appendix. However, to preserve a logical flow in the description of the hardware, without making multiple references to appendices, the approach of including it in this chapter was chosen.

4.2 GENERAL

The DFDAU, under microprocessor control, provides the necessary data acquisition, signal conditioning, conversions, error detection and correction, data formatting, calculations and self tests to generate and control the required input and output data streams.

The various functions of the DFDAU are performed by a number of modules, whose function and operation will be described in the following sections. These modules are:

i. *CPU/Memory Module 1 (CPU 1)*
ii. *Analog Multiplexer Module (AMX)*
iii. *Analog-to-Digital Converter Module (ADC)*
iv. *Discrete Multiplexer Module (DMX)*
v. *Digital I/O Control Module 1 (DM1)*
vi. *Digital I/O Control Module 2 (DM2)*
vii. *CPU/Memory Module 2 (CPU 2)*
viii. *Digital Input (DITS) Interface (DM3)*
ix. *Very Low Level/Tachometer Module (VLLDC/TACH)*

Each of these modules is resident on a single card and occupies one slot in the DFDAU chassis.

The CPU 1 and CPU 2 modules are identical in their electronic design and differ only in their software. CPU 1 controls the mandatory side of the DFDAU and CPU 2 controls the non-mandatory side.

Similarly, the DM1 module provides the I/O interface for CPU 1 and the DM2 module provides the I/O interface for CPU 2.

The DFDAU can accommodate a maximum of four Digital Input (DITS) modules and these are denoted by:

DM3 - 1
DM3 - 2
DM3 - 3
DM3 - 4

A system block of the DFDAU is shown in figure 4.1 giving the interrelationship of the various modules:

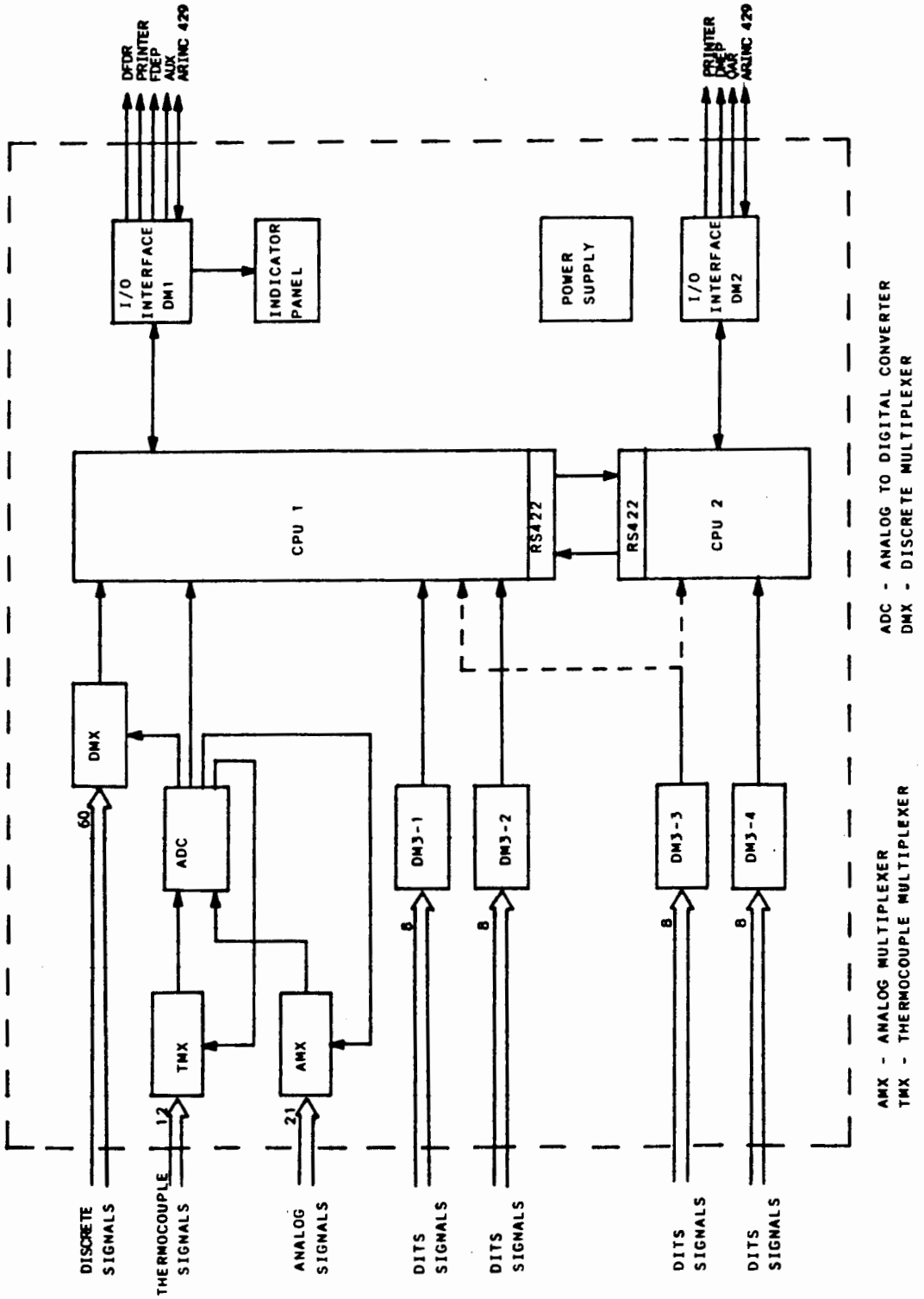


Figure 4.1 Simplified DFDAU Block Diagram

4.3 CENTRAL PROCESSING UNIT/MEMORY MODULE

4.3.1 General

The CPU/Memory module contains the microprocessor, a Texas Instruments TMS9900 16-bit processor, associated logic and memory, consisting of EPROM, RAM, and EAROM. The module controls the DFDAU, provides data processing, program and data storage, as well as interface control functions.

The memory capacity is 16K words of which 4K is RAM and 12K is EPROM used for program storage. Also available are 64 words of EAROM.

In addition, the following features are available:

- Separate 15-bit address and 16-bit data bus.
- 16 Prioritized, 15 maskable interrupts.
- 16 Memory Map I/O decodes.
- 16 Communication Register Unit (CRU) I/O decodes.
- TMS9901 Programmable Systems Interface.
- Reset integrator logic.
- Power down/hold logic.

A block diagram of the CPU module is shown in figure 4.2

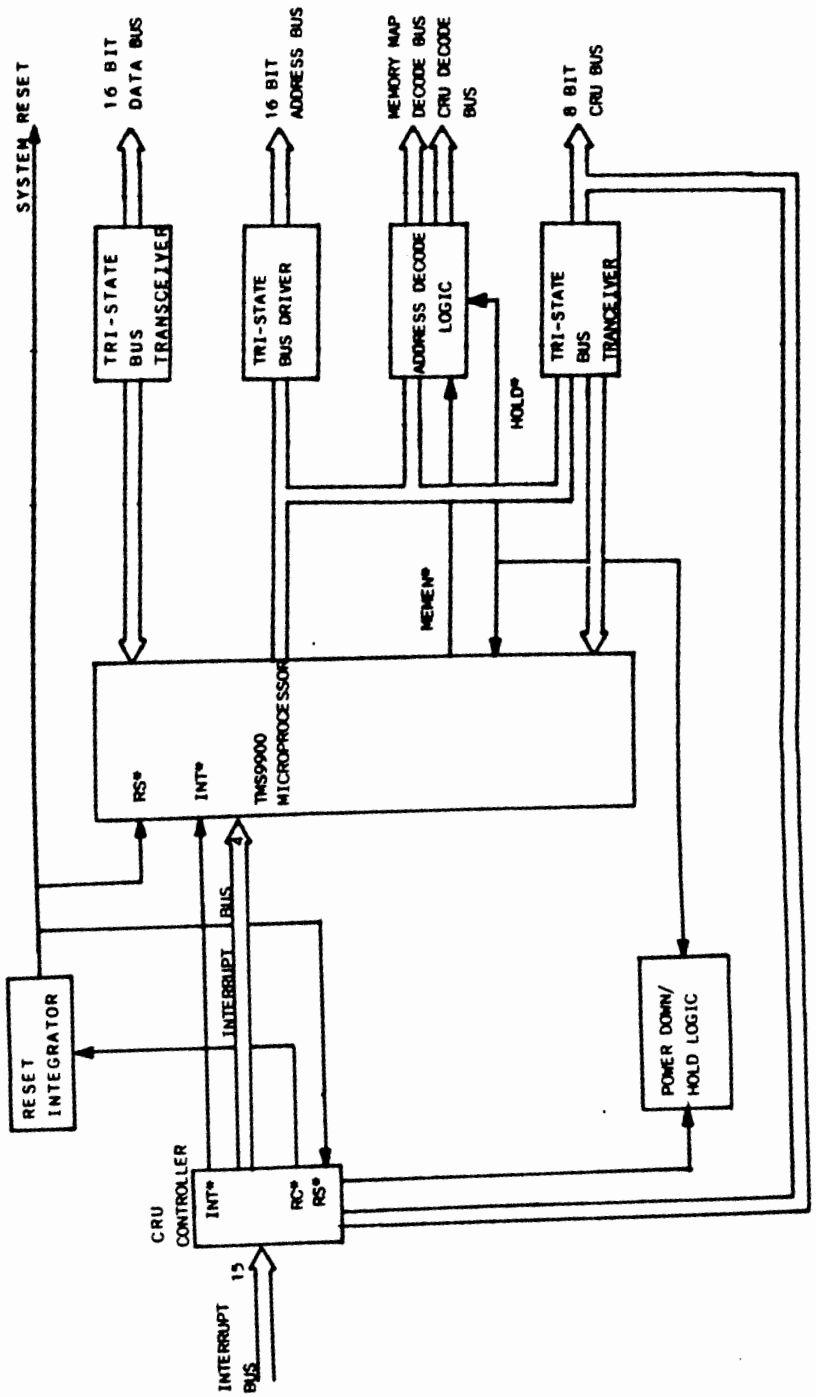


Figure 4.2 CPU Module Functional Block Diagram

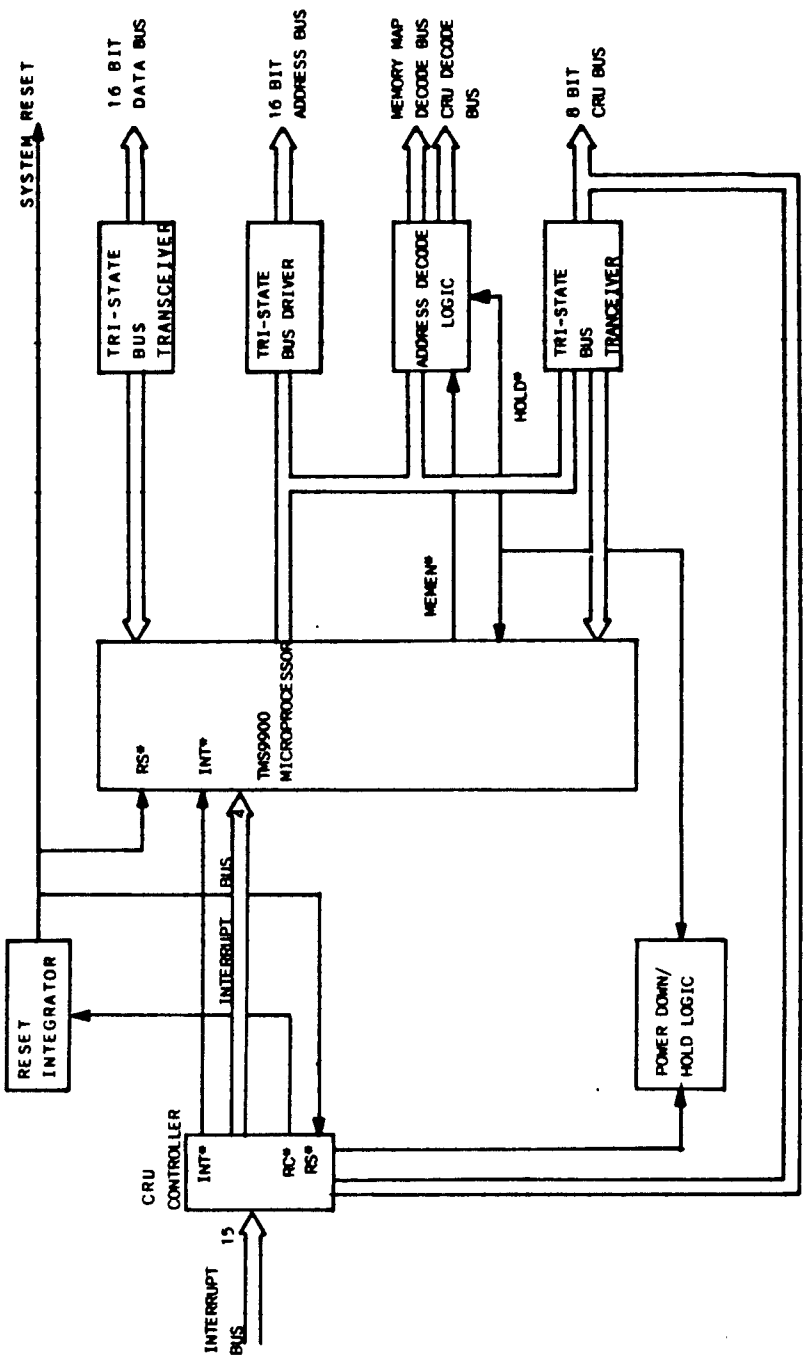


Figure 4.2 CPU Module Functional Block Diagram

4.3.2 Choice Of Microprocessor

The selection of the microprocessor must be directly related to the task involved. At the time of the conception of this project, (early 1981), microprocessors fell into two general industrially proven categories, namely 8-bit and 16-bit processors. 8-Bit microprocessors are generally used in byte-oriented data transfer applications with small amounts of code conversion, i.e intelligent terminals, keyboards, printers and process control functions.

Applications which require computation and general data processing, are typically the application area for 16-bit microprocessors, since they require the movement of large amounts of data from memory, to the processor's ALU and back to memory.

Arrays of data are often compared, scanned or combined via lengthy algorithms. These criteria are a greater measure of throughput capabilities than individual instruction times.

Various sample programs were written for the Texas Instruments TMS9900 and for the Intel 8085 and the Intel 8086 involving a number of data acquisition and manipulation tasks. The software timing was calculated and in all cases, the TMS9900 was significantly faster, even though the individual instruction times are quite close. This is largely due to the fact that it has separate address and data busses (ie. not a single multiplexed bus).

Another criterion for the selection of the TMS9900, was its memory-oriented architecture. This allows the register file to be memory resident, rather than a fixed set of hardware registers. This feature has a number of advantages in an interrupt driven system such as the DFDAU, further described in the following paragraphs (Reference 21).

In the hardware register configuration, all functions must share the same register file, therefore their contents must be saved in memory each time a new processing function is started. Upon completion of that function, (or interrupt service), the register file of the previous function must be retrieved to provide continuity in that routine. Depending on the number of registers available, execution times can be significant.

A memory-oriented architecture processor design allows the register file to be memory resident. The advantage of this is that it eliminates the requirement of moving data to a register (ie an accumulator) for processing, and then moving the resultant data back to memory for storage.

4.3 CENTRAL PROCESSING UNIT/MEMORY MODULE

4.3.1 General

The CPU/Memory module contains the microprocessor, a Texas Instruments TMS9900 16-bit processor, associated logic and memory, consisting of EPROM, RAM, and EAROM. The module controls the DFDAU, provides data processing, program and data storage, as well as interface control functions.

The memory capacity is 16K words of which 4K is RAM and 12K is EPROM used for program storage. Also available are 64 words of EAROM.

In addition, the following features are available:

- Separate 15-bit address and 16-bit data bus.
- 16 Prioritized, 15 maskable interrupts.
- 16 Memory Map I/O decodes.
- 16 Communication Register Unit (CRU) I/O decodes.
- TMS9901 Programmable Systems Interface.
- Reset integrator logic.
- Power down/hold logic.

A block diagram of the CPU module is shown in figure 4.2

Another advantage of the memory-oriented architecture, is that it minimizes the need for saving the register files in memory, since the registers themselves are memory resident. This results in additional memory saving, since the program routine and memory space to store and retrieve the register file is not required. A saving of execution time as well as memory space results. Thus, the TMS9900 memory-oriented architecture ranks above the register-oriented architecture in overall system performance for this particular application (Reference 3, 21).

Table 4.1 shows a comparison of the features of the three microprocessors that were considered for the project, namely the TMS9900, Intel 8085 and the Intel 8086.

	TMS9900	Intel 8085	Intel 8086
Word Length	16	8	16
Address Modes	8	4	5
Clock Freq.	3-4 MHz	3-5 MHz	5 MHz
No. of Instr.	69	80	78
H/W Multiply	YES	NO	YES
H/W Divide	YES	NO	YES
High Level Languages	PASCAL FORTRAN BASIC	PLM-80/85 FORTRAN BASIC	PASCAL COBOL FORTRAN PLM-86
Proc. Speed (microseconds)	@ 3 MHz	@ 3 MHz (8 bit)	@ 5 MHz
ADD	4.6	1.3	0.6
MULTIPLY	17.3	-	23.6
DIVIDE	41.3	-	28.8
I/R Response	7.3	16.0	12.8
I/R Return	4.6	16.5	17.0
Registers	16 General Purpose	8 General Purpose/Dedic	8 General Purpose/Dedic
Separate Data and Addr. Bus	YES	NO	NO

Table 4.1 Microprocessor Comparison

It should be noted that memory-to-memory architecture and instructions in the TMS9900 do not sacrifice the concept of registers. The general "register file" is conceptually retained as a block of 16 words of memory (figure 4.3a).

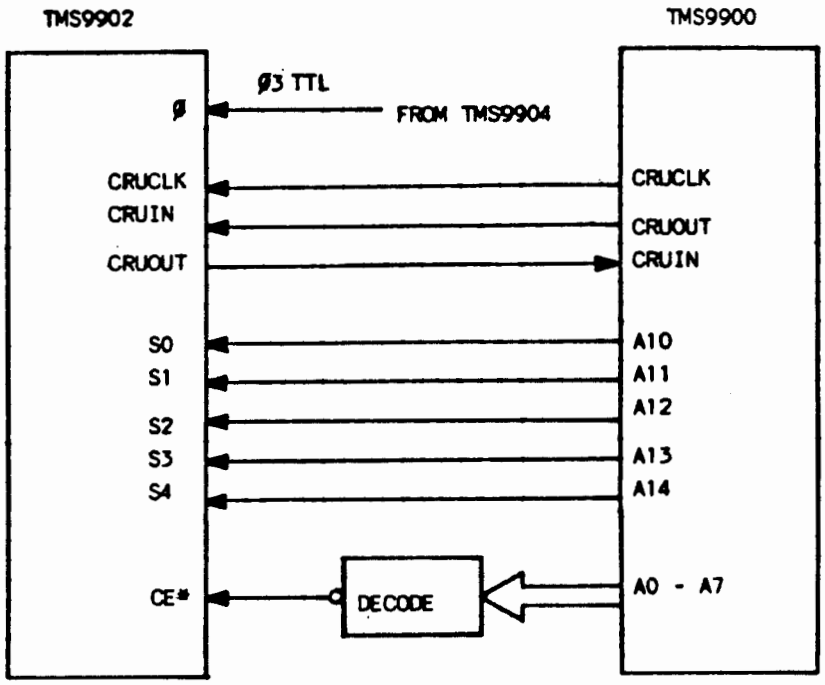


Figure 4.5 Interconnections between CPU and TMS9902

The TMS9902 interfaces to the asynchronous communications channel on 5 lines: Request-To-Send (RTS), Data-Set-Ready (DSR), Clear-To-Send (CTS), Serial Transmit Data (XOUT), and Serial Receive Data (RIN).

The RTS goes active low whenever the transmitter is activated. However, before data transmission begins, the CTS input must be active. The DSR input does not affect the receiver or the transmitter. When the DSR or the CTS changes level, an interrupt is generated.

Transmission characteristics of the TMS9902 are fully programmable with data rates up to 19.2 Kbaud, 5 to 8 bit character length, even, odd or no parity, and 1 or 2 stop bits.

These characteristics are set up by the software at initialization time by setting selected bits in the various control registers (reference 3). This will be described in chapter 5.

- 16-Bit CPU

- Minicomputer Instruction Set
  69 Instructions including multiply and divide.
  8 Address Modes.

- High Speed performance.
  3.0 MHz clock
  Addition: 4.6 microseconds
  Multiplication: 17.3 microseconds

- Memory-to-memory architecture
  Provides 16 general-purpose working registers
  per program context.

- I/O
  Separate I/O and Interrupt Bus structure.
  16 Priority Interrupts.
  Programmed DMA I/O capability.
  64K Bytes of direct memory addressing.

4.3.4 Communications Register Unit (CRU)

The TMS9900 provides non-multiplexed parallel I/O and memory control for maximum performance when needed, with 16-bit address and data bus. It has also a separate serial bus to allow the use of peripherals for relatively slow I/O processes which will tolerate this reduced speed. This is the Communications Register Unit (CRU).

The CRU is a bit addressable (4096 bits), serial input/output bus which is part of the 9900 microprocessor. The CRU address space is in addition to the 32K memory address space. For a bit to be output onto the CRU bus, an address is placed on the address bus by the microprocessor. This is decoded as a CRU address if the control line MEMEN* (figure 4.2) is high, and it is decoded as a memory address if MEMEN* is low.

Operations can be performed on 1 to 16 bits per instruction. If the I/O function is a multi-bit operation, the microprocessor places the appropriate address on the address bus, places the bit data on the CRU output bus, and raises the CPU output clock, or inputs the bit data from the CRU input bus, increments the address bus and repeats the process for the next bit.

4.3.5 Serial I/O Communications Controller

Serial I/O data communications for the TMS9900 microprocessor are handled by the Texas Instruments TMS9902 Asynchronous Communications Controller. Data communications via the TMS9902 is established around the RS232C protocol, and is controlled by the Communications Register Unit. Figure 4.4 shows a typical application of the TMS9902 controller.

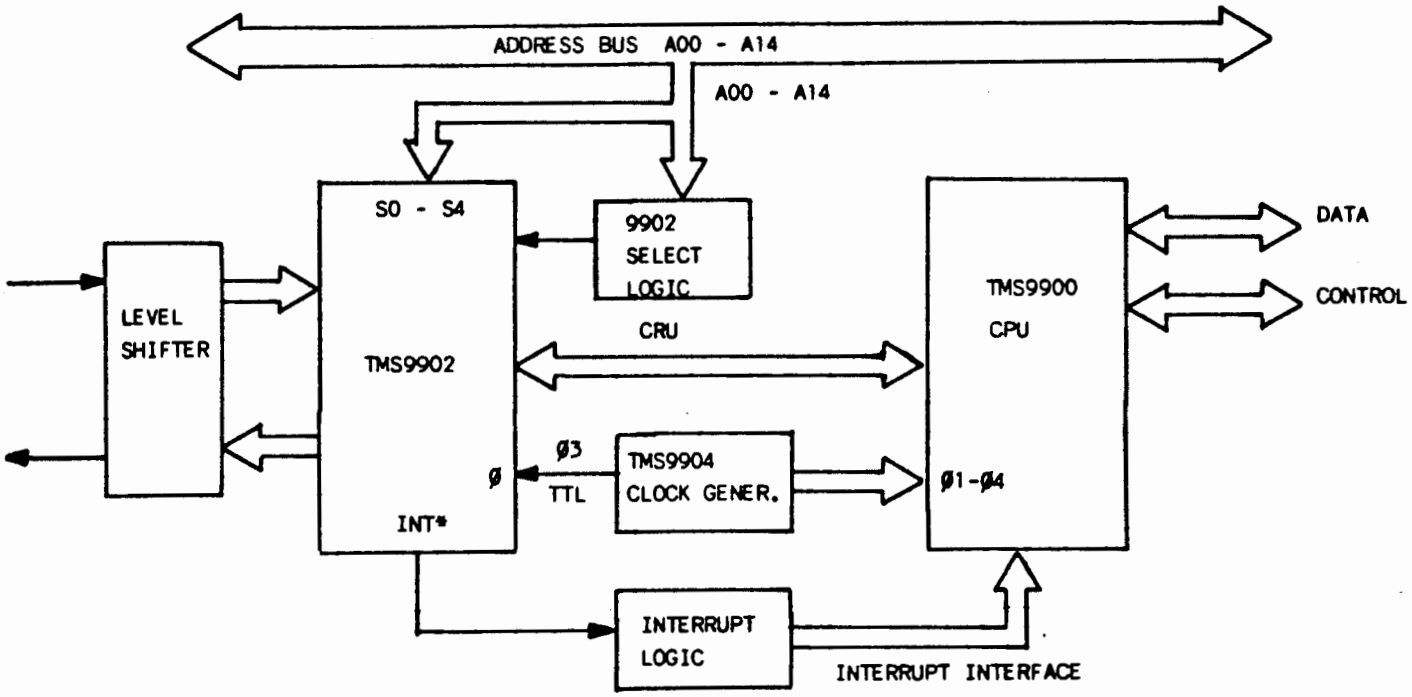


Figure 4.4 Typical Application of the TMS9902

The CRU interface consists of 5 address select lines (S0-S4), chip enable (CE*) and 3 CRU lines (CRUIN, CRUOUT, CRUCLK). An additional input to the CPU is the interrupt line (INT*). The TMS9902 occupies 32 bits of CRU space, and each of the 32 bits is selected individually by the processor lines A0 - A14, which are connected to the TMS9902 select lines S0 - S4. Chip enable (CE*) is generated by decoding address lines A0 - A9 for CRU cycles.

Figure 4.5 shows the interconnections between the CPU and the TMS9902.

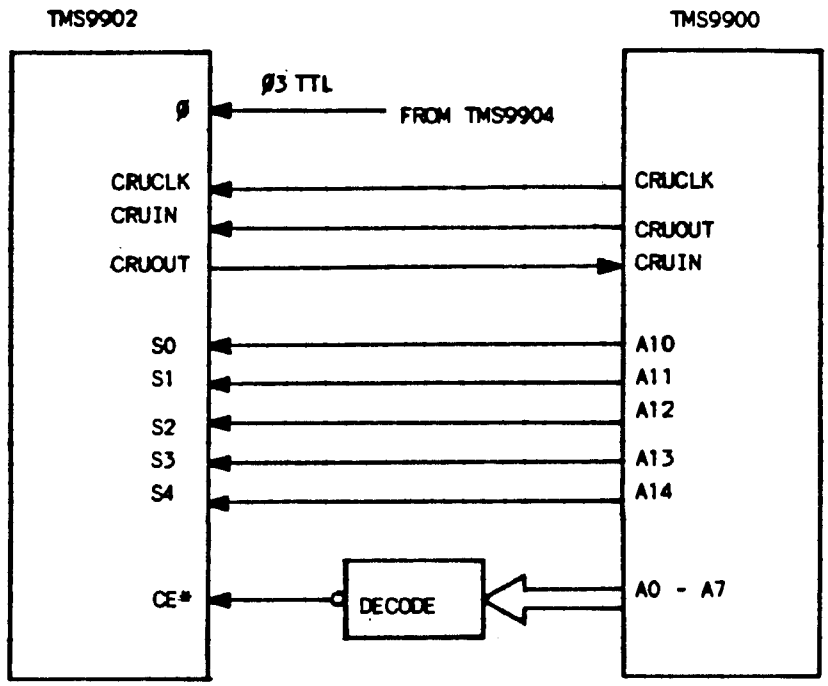


Figure 4.5 Interconnections between CPU and TMS9902

The TMS9902 interfaces to the asynchronous communications channel on 5 lines: Request-To-Send (RTS), Data-Set-Ready (DSR), Clear-To-Send (CTS), Serial Transmit Data (XOUT), and Serial Receive Data (RIN).

The RTS goes active low whenever the transmitter is activated. However, before data transmission begins, the CTS input must be active. The DSR input does not affect the receiver or the transmitter. When the DSR or the CTS changes level, an interrupt is generated.

Transmission characteristics of the TMS9902 are fully programmable with data rates up to 19.2 Kbaud, 5 to 8 bit character length, even, odd or no parity, and 1 or 2 stop bits.

These characteristics are set up by the software at initialization time by setting selected bits in the various control registers (reference 3). This will be described in chapter 5.

4.3.6 Programmable Systems Interface

General purpose parallel I/O and interrupt control is provided by the TMS9901 Programmable Systems Interface.

Figure 4.6 below shows a typical application of the TMS9901.

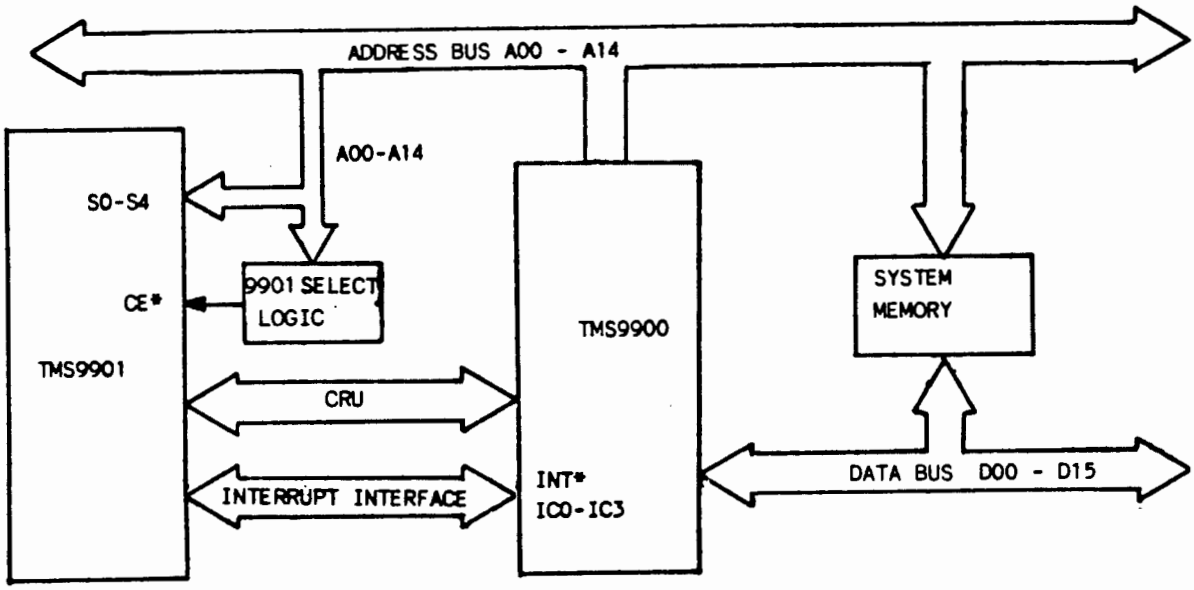


Figure 4.6 Typical TMS9901 Application

The TMS9901 is designed to interface with the TMS9900 and its CRU bus. It provides 15 lines of interrupt prioritization, a 15 bit interrupt mask setable under program control, a 14 bit timer/real-time clock which is setable and readable under program control, and a 15 bit I/O port. Figure 4.7 shows a block diagram of the TMS9901.

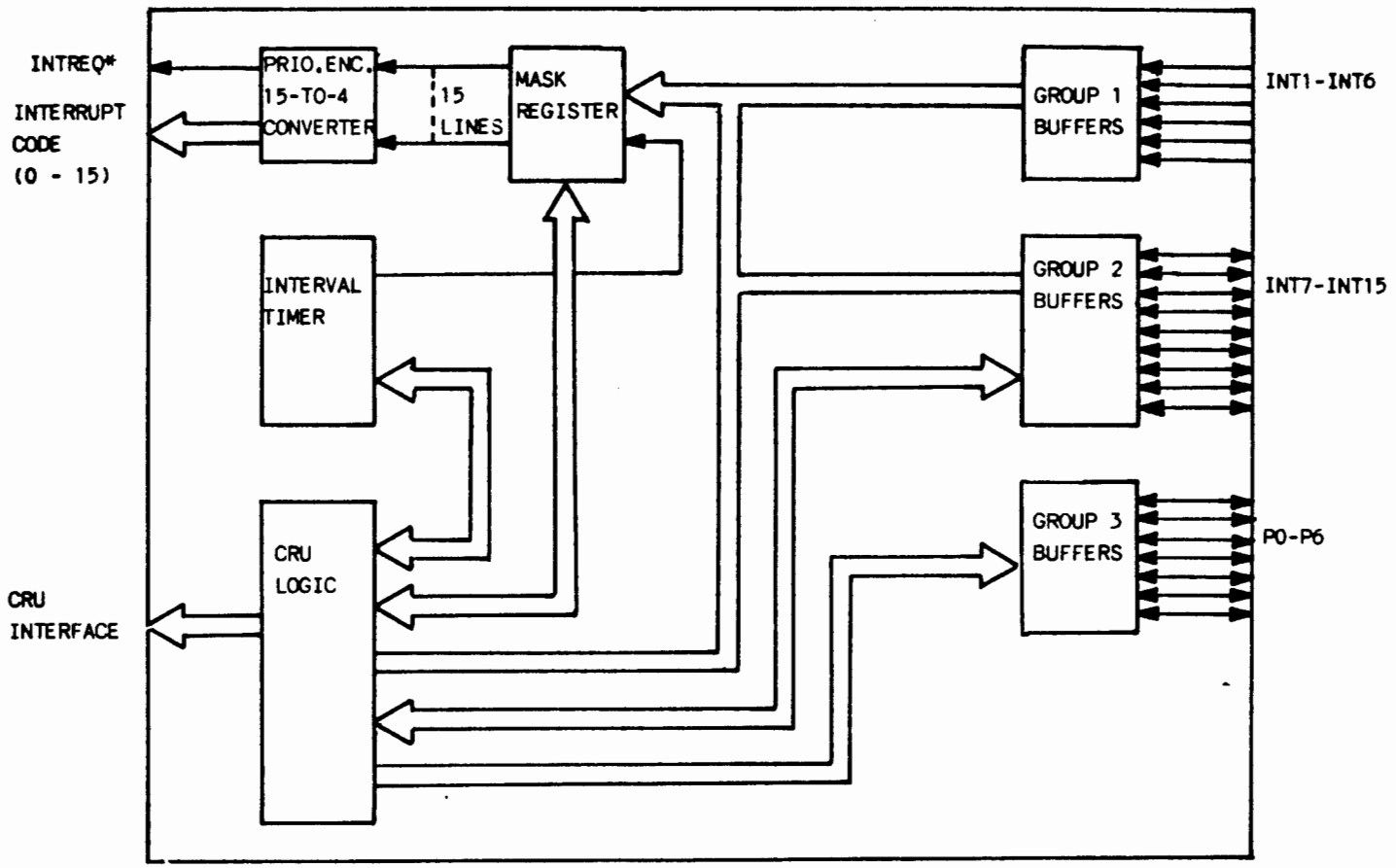


Figure 4.7 Block Diagram of the TMS9901

The TMS9901 interfaces with the CPU via the CRU and the interrupt lines as shown in figure 4.6. It occupies 32 bits of the CRU I/O space. The five most significant bits of the address bits are connected to the "S" lines of the TMS9901 to address one of the 32 bits.

At reset, all I/O ports are programmed as inputs. By writing to any I/O port, that port will be programmed as an output port until another reset occurs.

The interval timer on the TMS9901 is accessed by writing a logic 1 to select bit zero (control bit) which then puts the TMS9901 CRU interface in the clock mode. Once in the clock mode, the 14-bit clock contents can be written into or read. Writing to the clock register will re-initialize the clock and cause it to start decrementing. When it reaches zero, an interrupt occurs,

and the clock gets reset to its initial value. Figure 4.8 shows the interval timer section.

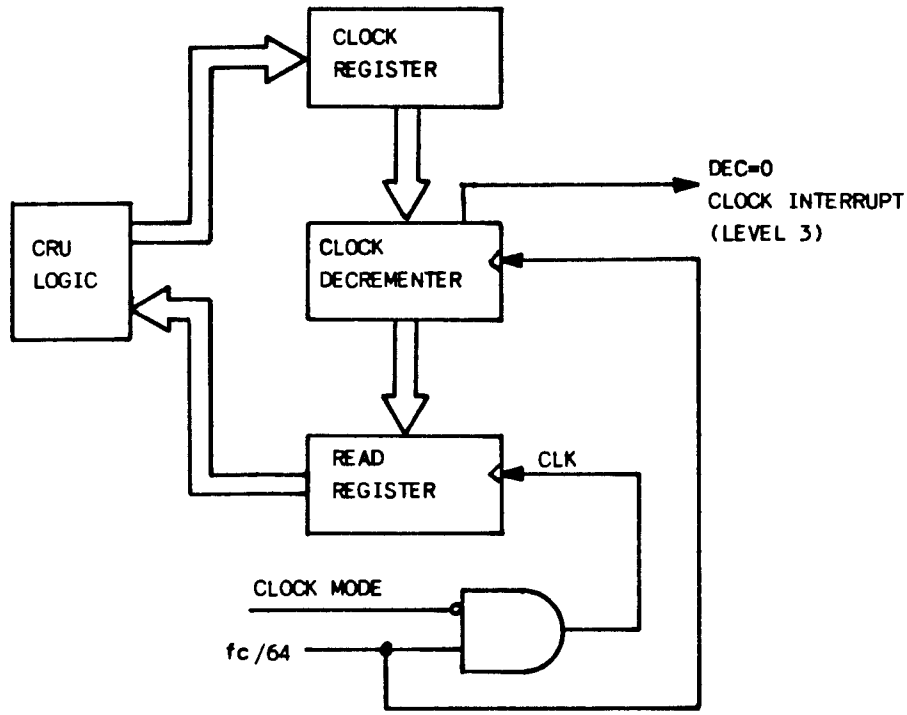


Figure 4.8 Interval Timer Section

The 14-bit timer decrements at a rate of fc/64 where fc is the clock frequency.

Hence $f(dec) = fc/64$
$T(dec) = 1/f(dec) = 64/3000000$
$= 21.33$ microseconds

Interrupt inputs to the TMS9901 are synchronized with the clock, inverted, and then ANDed with the appropriate mask bit. (There are 6 dedicated, and 9 programmable interrupts). The interrupt section of the TMS9901 is shown in figure 4.9 below.

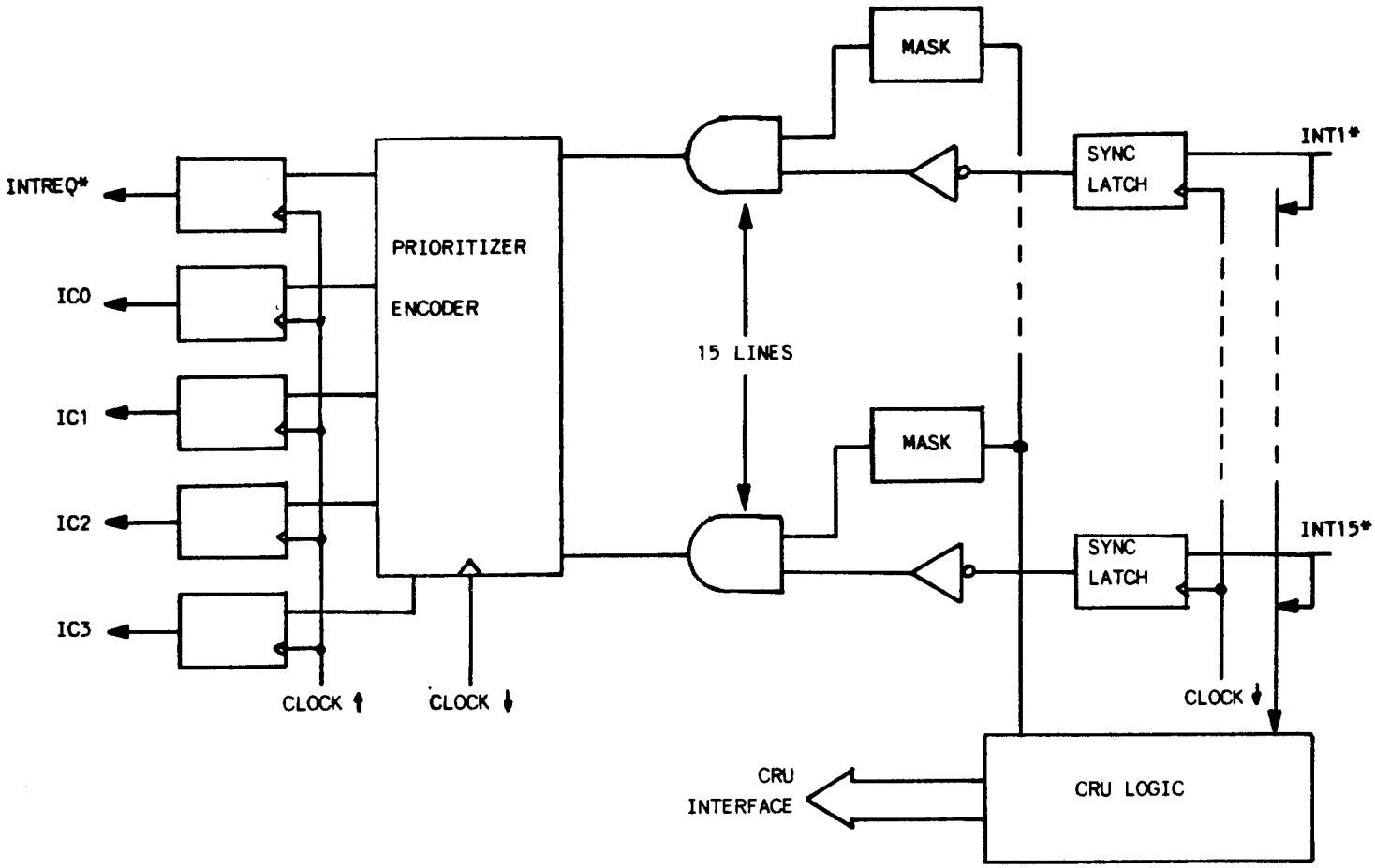


Figure 4.9 TMS9901 Interrupt Control Section

On the rising edge of the clock, the prioritizer and the encoder senses the masked interrupts and produces a four-bit encoding of the highest priority interrupt present. The four bit priority code, along with the INTREQ, constitute the interrupt request.*

To illustrate a typical use of the TMS9901 in a 9900 system, consider figure 4.10

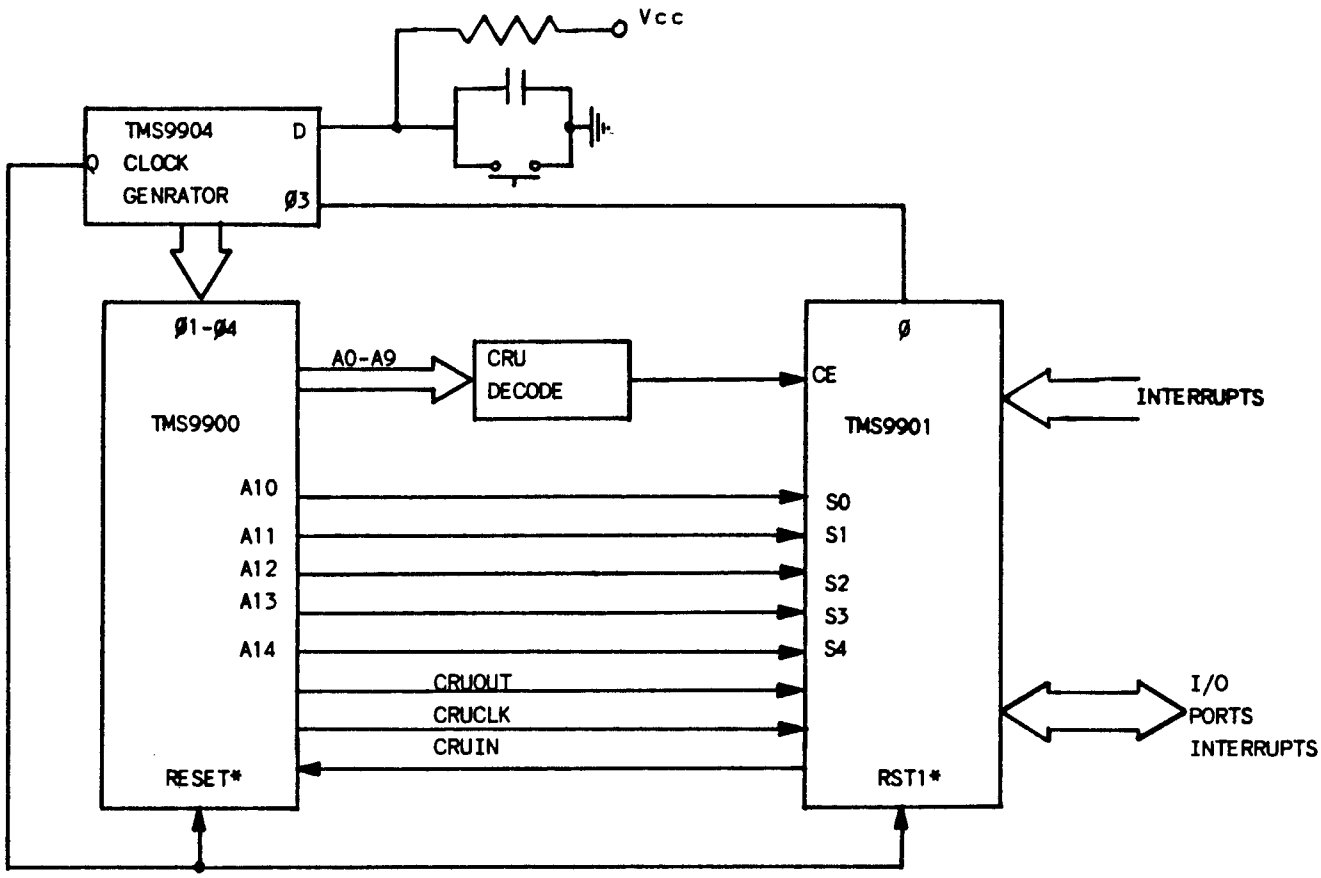


Figure 4.10 TMS9900/TMS9901 Interface

The clock generator syncs the RESET for both the 9901 and the CPU. The RC circuit provides the power-up and push-button reset signal. Address lines A0 to A9 are decoded on the CRU cycles to select the 9901. Address lines A10 to A14 are sent directly to the 9901 select lines S0 to S4 respectively to select which 9901 CRU bit is to be accessed.*

4.3.7 Built-In Test (BIT) Feature - Watchdog Timer

BIT is accomplished in the CPU module and in conjunction with other modules of the DFDAU. The computer module drives a reset integrator (watchdog timer) which, if allowed to timeout, causes a system reset. If the processor is unable to run its executive routines, and therefore unable to process data (ie. the clock

circuit were to fail), the reset integrator would time out and cause a system reset. The system reset integrator therefore allows the system to re-initialize and recover from a non-catastrophic failure.

The watchdog timer has to be reset by the software background tasks at regular intervals which are shorter than the time-out period (500ms). If a "hangup" somehow occurs in the microprocessor, the reset pulses will cease and the reset integrator will time out. This causes the system reset logic (see figure 4.2) to provide a system reset signal which resets the microprocessor and forces it to restart as though power had momentarily been turned off.

Wraparound circuitry is provided on the various I/O ports (such as the digital I/O and DITS modules) so that the health of the system can be continuously monitored.

4.3.8 Memory Map I/O

The processor will process an input signal either in response to an interrupt or because of its normal program routine. The processor selects the proper input signal by issuing address and clock signals in the same way it would read/write data from memory, but instead of memory locations, it is reading/writing from hardware registers. This type of I/O processing is called "Memory Map I/O".

The microprocessor outputs data much the same way, except in this case it issues a write command instead of a read command. The output circuitry also conditions the buffers and issues an interrupt on completion of the output process.

All input data from the DITS port and the analog section are received by the processor using the memory map I/O method.

4.3.9 Power Up/Down Logic

The processor module is provided with a power-on signal from the power supply which changes level from a "zero" to a "one" on power-on initialization , after all voltages are up and stabilized. The signal changes from a "one" to a "zero" after a power interruption of greater than 11 milliseconds. The change occurs at least 1 millisecond before voltages begin to change.

When the processor senses the signal changing from a "zero" to a "one", it causes a system reset signal to be generated and takes the microprocessor out of the "HOLD" state.

When the processor senses the signal changing from a "one" to a "zero", it generates a level 1 interrupt (the highest level being 0 - system reset). This causes the microprocessor software to save workspace pointers, DFDR frame and word numbers, and place itself in a "HOLD" state which effectively halts the computer.

4.3.10 Memory Module

The memory module is resident on the CPU module, and consists of RAM, EPROM and EAROM. The current design can accommodate up to 32K words of addressable memory, although at this stage only 16K words have been implemented. The TI25L32 EPROM chips and the HM6533-2 CMOS RAM are used. These chips have bit arrangements of 4K by 8 bits and 1K by 4 bits respectively.

The partitioning is such that 4K words of the 16K memory are used as RAM, while the remaining 12K words are employed as EPROM. Although significantly more expensive than standard ROM, EPROM is used, in order to provide ease of software modification after hardware completion. In a system such as the DFDAU, where changes to the software can be predicted with a high degree of accuracy, the ability to merely unplug and reinsert a new memory module containing the modified software, is extremely important. Furthermore, the old memory module can be easily erased (under ultra-violet light) and re-programmed.

An EAROM (Electrically Alterable ROM) section consisting of 64 16-bit words is included and is used primarily for the storage of a history file of the DFDAU system failure/status data. The EAROM memory is not part of the main memory address space.

4.3.10.1 Program Memory (EPROM) -

The software is stored in EPROM, which is composed of TI25L32 integrated circuits. These are ultra-violet erasable, electrically programmable read-only memory modules. In circuit, these are organized into 4K word (16 bit) blocks.

4.3.10.2 Random Access Memory (RAM) -

Random access memory is used for buffers and the temporary storage of data. The types used on the DFDAU are CMOS HM6533-2 integrated circuits. These were chosen primarily for their speed-power factor, as well as their low stand-by power requirement (1mW typical).

The RAM is organized in 1K word (16 bit) blocks. Each block has enough electrical capacity to keep data valid for a power interruption of up to 200 milliseconds.

The memory module develops a memory valid signal. This signal is set high by the processor after it has initialized the RAM, and will stay valid until power has been removed for longer than 200 milliseconds. This signal is used to direct the processor for a complete initialization (power off for more than 200 milliseconds) or a partial initialization (power off less than 200 milliseconds).

4.3.10.3 Electrically Alterable Read Only Memory (EAROM) -

EAROM employs NMOS technology, the writing mechanism being to tunnel a charge between the gate and the substrate of the device. Once this charge has been trapped in the substrate, a logic 1 has been created, and because of the very long decay rate of this charge, (10 years at 25 degrees Celsius), an effective permanent storage exists. This tunnel charge may be erased and re-written as well, and devices are arranged in arrays allowing random access techniques as in RAMS.

Some restrictions apply however. After approximately 100,000 writes or erasures, the storage times decay logarithmically, and hence its reliability as a storage medium declines. Furthermore, the average time to erase/write a single word of data is approximately 500ms, and the read time is approximately 64 microseconds, both extremely slow. These restrictions therefore eliminate the possibility of using the EAROM as RAM or as in-circuit alterable EPROM.

The EAROM circuit consists of an address decoder, an EAROM control latch, an input data latch, an output data latch and two ES2501 EAROM memory integrated circuits as shown in figure 4.11.

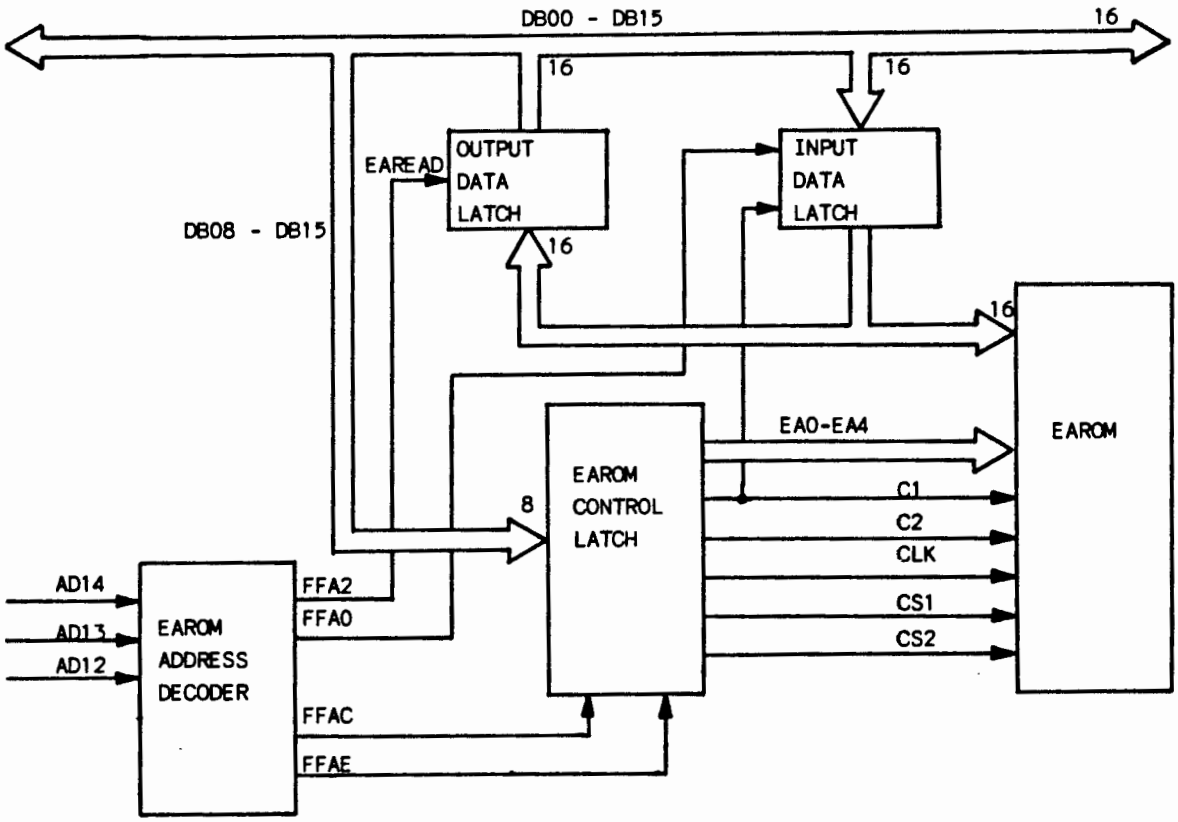


Figure 4.11 EAROM Circuitry Block Diagram
Two 32 X 16 bit EAROM chips organized as a 64 X 16 bit EAROM

The CPU controls the EAROM in a brute-force manner, meaning that the CPU (under software control) must sequence the two control lines C1 and C2, the clock line CLK, the two chip-select lines CS1, CS2 and the five address lines by means of hardware (memory map I/O) addresses.

To write to the EAROM, the desired address is placed on the data bus DB08 - DB15. Address FFAE is placed on the address bus AD00 - AD14. The address decoder produces a signal which latches the EAROM control latch, which presents the address to the EAROMS on EA0 through EA4 lines.

Then the EAROM input data latch is loaded with the data to be written to the EAROM via the data bus. Address FFA0 is placed on the address bus AD00 - AD14. The address decoder produces a

signal, EARSEL, which latches the data into the EAROM input latch, which presents the data to the EAROM.

Then, the chip select and control signals are set to the Erase Mode (to erase the contents of the desired destination location) by placing the proper code on the data bus and address FFAC on the address bus. The address decoder produces a signal which latches the control signals into the EAROM. The control signals are as follows:

C1	-	LOW
C2	-	HIGH
CLK	-	LOW
CS1,CS2	-	Select the upper or lower 32 addresses respectively

The control signals are then changed to the Write Mode as follows:

C1	-	LOW
C2	-	LOW
CLK	-	LOW
CS1,CS2	-	As required

The control, clock and chip select lines are controlled by memory map I/O address as follows:

Bit	Control Function
0	C1
1	C2
2	CS1
3	CS2
4	CLK

Hence, the sequence of steps for writing to the EAROM would be as follows:

1. Load EAROM address buffer (FFAE) with the EAROM address (0-63).
2. Load EAROM data buffer (FFA0) with the data to be written to the EAROM.
3. Set control lines to erase mode and select desired chip by toggling the chip select lines CS1 and CS2.
4. Wait 200ms.
5. Initiate write mode by turning C2 off.
7. Wait 200ms.
8. Turn chip select off and clear all control lines.

To read an address from the EAROM, the desired address is placed

on the data bus (DB08 - DB15). Address FFAE is then placed on the address bus AD00 - AD14. The address decoder produces a signal which latches the EAROM address into the EAROM control latch which presents the address to the EAROM on the EA0 to EA4 lines.

The chip-select and control signals are set to the Read Mode by placing the proper code on the data bus and address FFAE on the address bus. The address decoder produces a signal which latches the control signals into the EAROM control latch. The control signals are as follows.

C1	-	HIGH
C2	-	LOW
CLK	-	HIGH then LOW
CS1,CS2	-	As required

The data is read via the data bus by placing address FFA2 on the address bus. The address decoder produces a signal EAREAD which latches the data from the EAROM into the output data latch. Then the chip-select signal is forced low.

The logic for reading from the EAROM would therefore be as follows:

1. Load EAROM address buffer (FFAE) with the EAROM address (0-63).
2. Set to read mode.
3. Turn CLK on.
4. Wait 20 microseconds.
5. Turn CLK off.
6. Read EAROM data from memory map address FFA2
7. Clear all control signals.

4.4 DIGITAL INFORMATION TRANSFER SYSTEM (DITS) INPUT INTERFACE

4.4.1 General

This module provides the ARINC 429 (reference 2) digital data input interface to the DFDAU. The DITS inputs are not multiplexed. Instead each input port has its own dedicated controller and this allows all the DITS input ports to be looked at simultaneously. A block diagram of this module is shown in figure 4.12

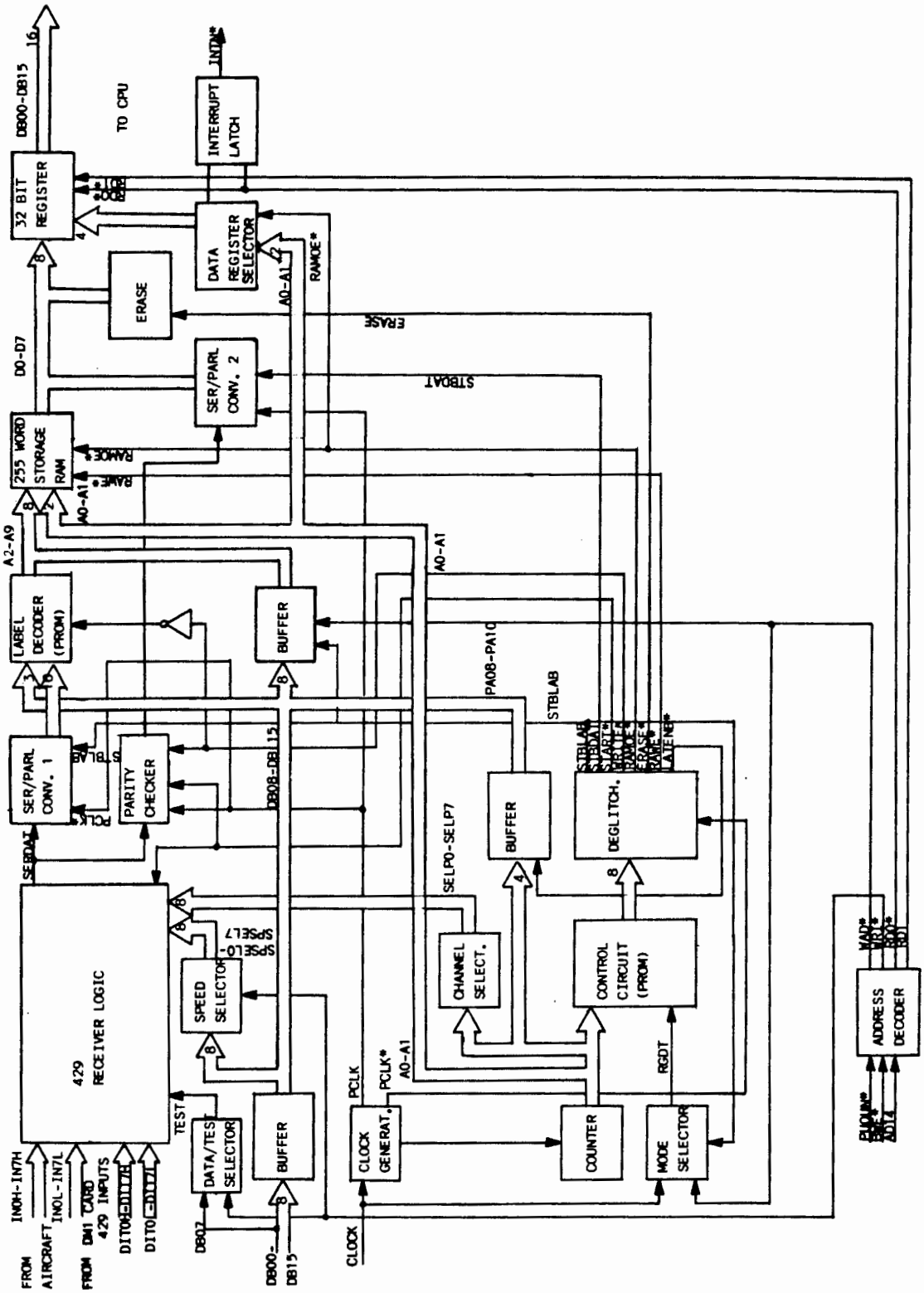


Figure 4.12 ARINC 429 DITS Interface Module Functional Block Diagram

4.4.2 Functional Description

The DITS input interface accepts up to 8 channels on the IN0-IN7 lines of 32-bit bipolar-RZ coded words with a 4-bit minimum gap between each word at data rates of either 12 to 14.5 or 100 kbps. Each word is comprised of an 8 or 10-bit label code and a 24 or 22-bit word respectively. Bipolar-RZ coding and word organizations are shown in figure 4.13

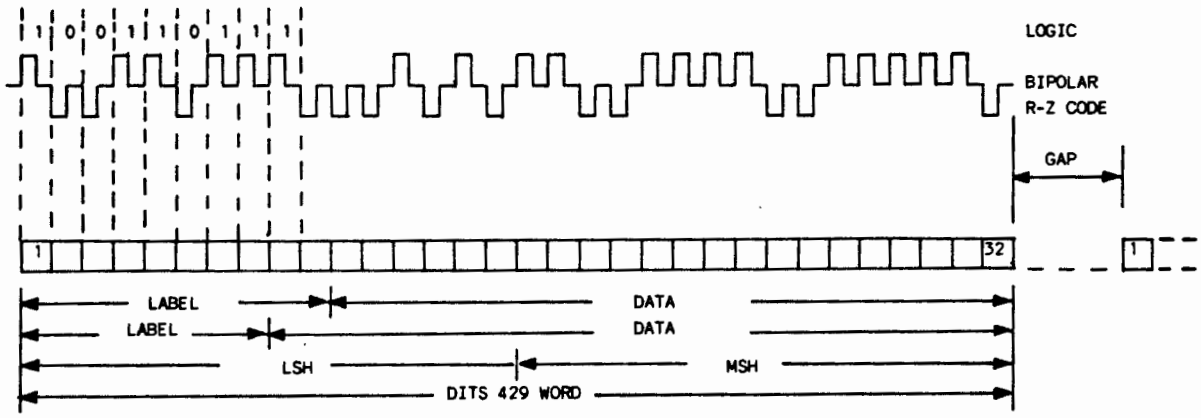


Figure 4.13 Bipolar-RZ Coding And Word Organization for the ARINC 429 DITS Input Interface

Since all input channels are identical, only one will be described.

4.4.2.1 Write Mode -

Data enters the input protection circuit which provides line isolation and input over-voltage protection. The bipolar-RZ data is then converted to NRZ data by the bipolar-RZ to NRZ converter (figure 4.14) and then presented to the receiver block.

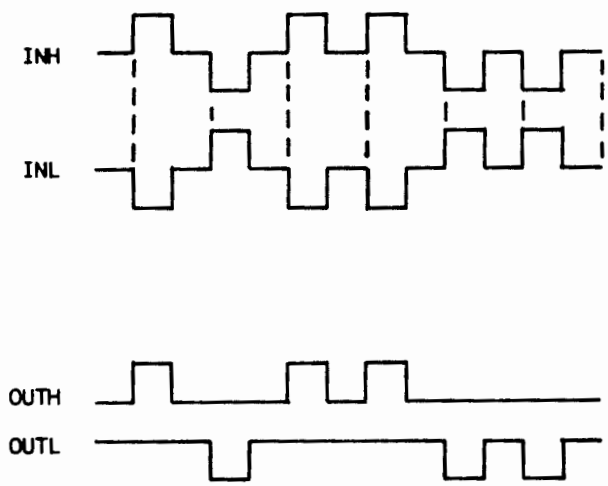


Figure 4.14 Bipolar RZ-to-NRZ timing diagram

Initially, the microprocessor controls the receiver logic by selecting either data or test mode and by selecting either low or high speed mode via the TEST and the SPSEL0 through SPSEL7 lines respectively. These signals are generated by placing the proper data on the data bus DB07 through DB15, and by placing the proper address on the address bus AD00 through AD14, and by causing the Write Enable BWE signal to go low. The proper address and data for this operation is shown in figure 4.15.*

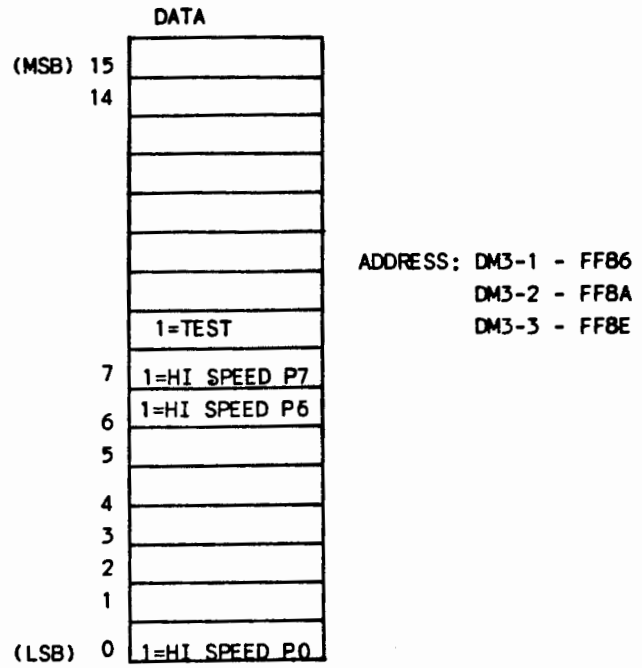


Figure 4.15 DITS Test and Speed Select

The address decoder produces a signal WRI which enables the data/test selector and speed selector. The data/test selector produces the TEST signal and the speed selector produces the SPSEL0 to SPSEL7 signals.*

The receiver interface logic then waits for a gap. When a gap is detected, it counts the bits until the next gap is detected. If 32 bits are received, it stores the 32-bit data in an intermediate register and sets a flip-flop. At the 32nd bit time, the control circuit produces a START signal which will cause the data to be loaded into an internal output register. As new data is received, it replaces the old data word with the new data word.*

Each receiver logic block is accessed in sequence by means of SELP0 through SELP7 lines. This is achieved by the channel selector which is driven by the output of the counter, which in turn is fed by the 1MHz clock generator.

The SELP0 through SELP7 signals select each of the receiver logic blocks in sequence and the PCLK signal clocks the 32-bit data word from the internal output register onto the SERDAT line.

The first 8 or 10 bits (depending on the actual data source on the aircraft) of each 32-bit word on the SERDAT line (the label) is clocked into serial-to-parallel converter 1 by means of the PCLK* signal and the 32-bit word in the internal output register is erased. Simultaneously the 32-bit word is processed by the parity checker and indicator, and the resulting data DATPAR is clocked into serial-to-parallel converter 2.

When the label has entered serial-to-parallel converter 1, the STBLAB signal goes high and causes the label to appear on the bus connected to the label decoder EPROM. Simultaneously, three address lines PA08 through PA10 which are outputs of the counter are applied to the label decoder EPROM. These three address lines, together with the 10 data bits from the label data, form the address for the decoder EPROM. The decoder EPROM is an 8K x 8-bit device, programmed to produce numbers 0 through 255 which form the address for the 256-word storage RAM.

The WRITE line is high at this time and is inverted when presented to the label decoder EPROM. If a correct label is received (one of the labels programmed into the EPROM decoder), the decoder produces an address representing the address of the parameter corresponding to that label, to appear on the A2 through A9 bus; otherwise it produces address 00(Hex). The address received by the RAM, and A0 (high) and A1 (low) from the counter become the total address for the 256-word storage RAM.

By this time, serial-to-parallel converter 2 has stored the label into its internal output register. The STBDAT line changes state and causes the label to appear on the data lines D0 through D7. The RAWE* signal goes high to write the data on the D0 to D7 data bus into the second RAM location (determined by the label decoder).

As the remainder of the 32-bit data word enters serial-to-parallel converter 2, the A0 and A1 lines from the counter change the address location in the RAM to the 3rd, 4th and 1st locations, thus storing the 32-bit data word in four, 8-bit locations. The 256 word RAM can store 32 x 32-bit words from each of the eight input channels.

The RAM approach is necessary, because data from all DITS data sources arrives simultaneously at the DFDAU input channels at a "data-item" rate independent of the operation of the DFDAU. Therefore, when the processor wants to read a DITS data item it reads it from the DITS on-board RAM.

The parity checker and indicator checks the parity of the 32-bit word (including parity-bit, bit 32). If the parity is correct (odd), bit 32 is changed, if necessary, to a high. This indicates to the microprocessor, when it reads a word from RAM, that the word is in error.

4.4.2.2 Read Mode -

When the microprocessor reads a word stored in the 256-word RAM, it places a number (parameter id) representing the desired parameter on the data bus DB08 to DB15. The WAD signal goes high causing the number representing the desired parameter to enter the next buffer. When WRITE goes low, the parameter number is placed on the A2 through A9 lines. At this time, A0 is high and A1 is low. Since RAWE* is high, the RAM is in the read mode.*

The RAMOE goes low and causes the RAM to place the first 8 bits (label) of the requested data onto data bus D0 through D7. The data register selector then produces a signal that causes the label to enter the 32-bit data data register. The ERASE* signal then goes low and causes the erase logic to place all lows on D0 through D8. At this time RAWE* is low and allows the eight zeros to enter the RAM and erase the label portion of the data word.*

By erasing the label portion of the 32-bit data word, the software then has a way of knowing if the data item has been updated next time it attempts to read it from the DITS RAM. If it has not been updated since it was last read, the label portion remains zero, otherwise it will have been replaced by the next sample of that parameter.

The counter causes A0 through A1 to count from 01 to 10, 11 and 00 which retrieves the last three bytes of the 32-bit data and places in the remaining 24-bit locations in the 32-bit data register.

When the last byte enters the 32-bit data register, an interrupt signal INTN is produced by the interrupt latch to inform the processor that the requested data is ready to be read.*

The processor reads the stored data word by causing the BWE line to go low. The data register selector develops the RD0* signal which causes the least significant 16-bits (LSH) of the data word to be placed on the data bus.*

4.5 DIGITAL I/O MODULE

4.5.1 General

This module provides the digital interface circuits between the DFDAU'S Central Processing Unit and the various display and recording devices that interface with the acquisition system. The functional elements of this module, summarized below are explained in the following paragraphs and illustrated in figure 4.16.

- *DFDR Interface*
- *Auxiliary (AUX) Interface*
- *429 Output Interface*
- *Control/Indicator Panel Interface*
- *Clock Generators*
- *Printer Interface*
- *Asynchronous Communications Controller Interface*

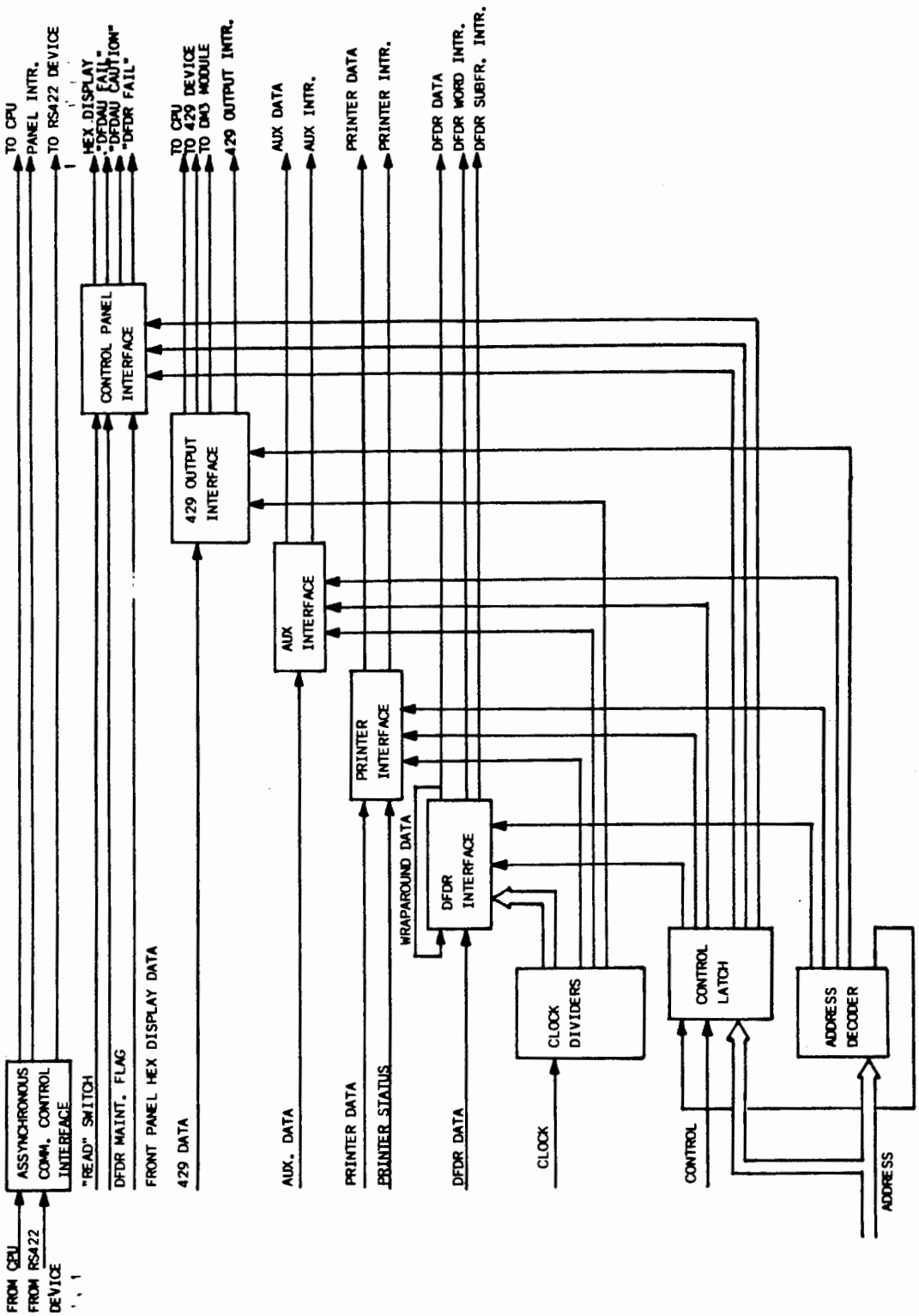


Figure 4.16 Digital I/O Module Functional Block Diagram

4.5.2 Clock Circuits

The clock circuits shown in figure 4.17 provide the timing and synchronization signals required by the various interface circuits. All the output clock signals are derived from the internal master clock of the CPU.

The clocks for the output circuits C512B through C64B provide the necessary timing to generate the output data formats. C256B for example 3.073 KHz is divided by 6 to provide C512 and in turn produce word rate interrupts for the CPU. The signal is further divided by 2048 to provide the frame SYNC pulses. The C64B (768 Hz) clock is the basic bit rate for the 64 wps output as is the 1.536 KHz clock (C128B) the bit rate signal for the 128wps output. The C45 signal(0.25Hz) provides the 4 second frame interrupt signal.

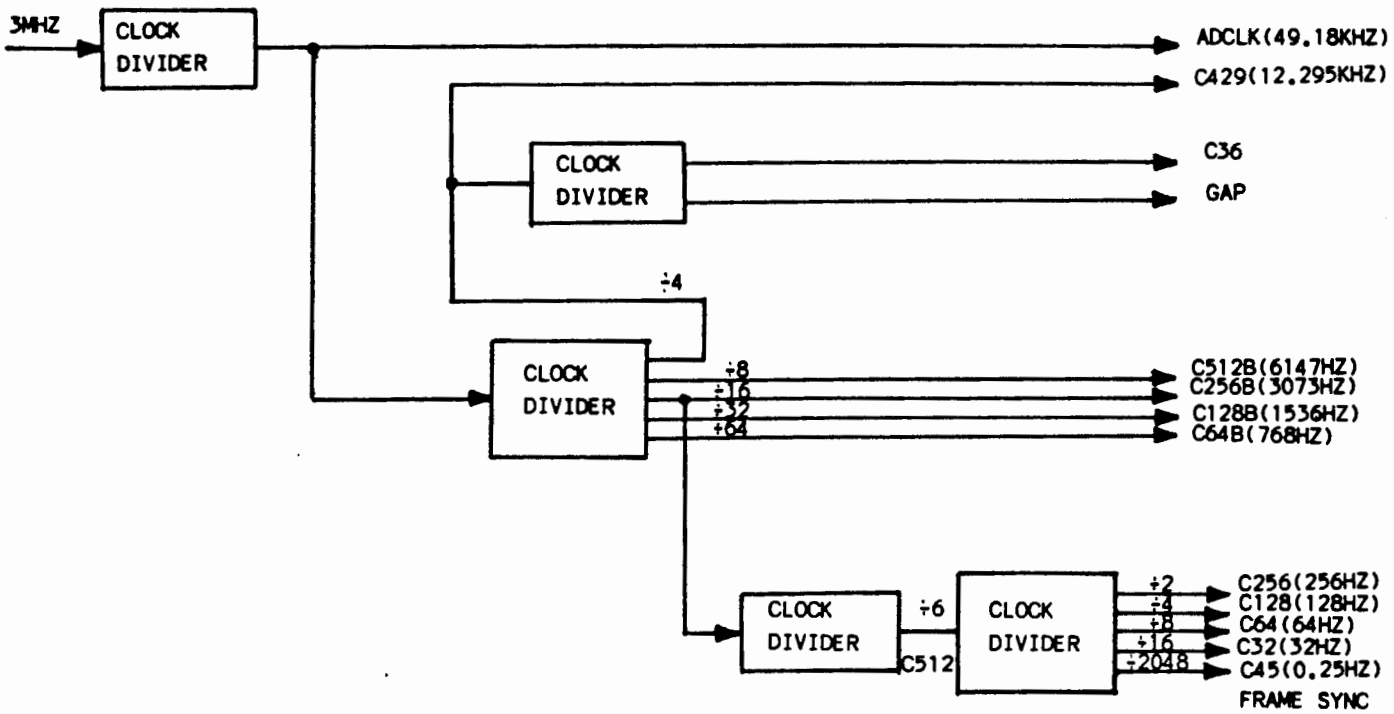


Figure 4.17 Clock Circuitry Block Diagram

The signal ADCLK is a clock signal used by the Analog-to-Digital Converter module, and the C429 is a clock signal used by the DITS output interface on the Digital I/O module.

4.5.3 Address Decoder And Control Latch

The address decoder develops the appropriate enable signals whenever the proper addresses are placed on the address bus and thus enabling the various interface circuits.

The control latch is the means by which the CPU develops control signals for the various interface circuits on the module. The control latch is enabled by the address decoder, and the address lines select at which interface the control signals will appear.

4.5.4 DFDR Interface

The output of this section provides the proper data encoding and timing signals to drive the DFDR.

The serial data to the DFDR is coded in Harvard Bi-phase as shown in figure 4.18.

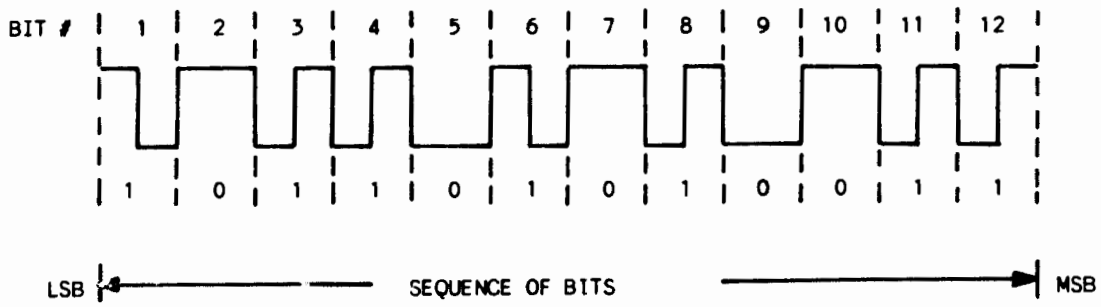


Figure 4.18 Harvard Bi-phase Code for the DFDR

The data stream is made up of 256 word frames which are subdivided into four 64-word subframes, separated by four 12-bit sync-code words as shown in figure 4.19.

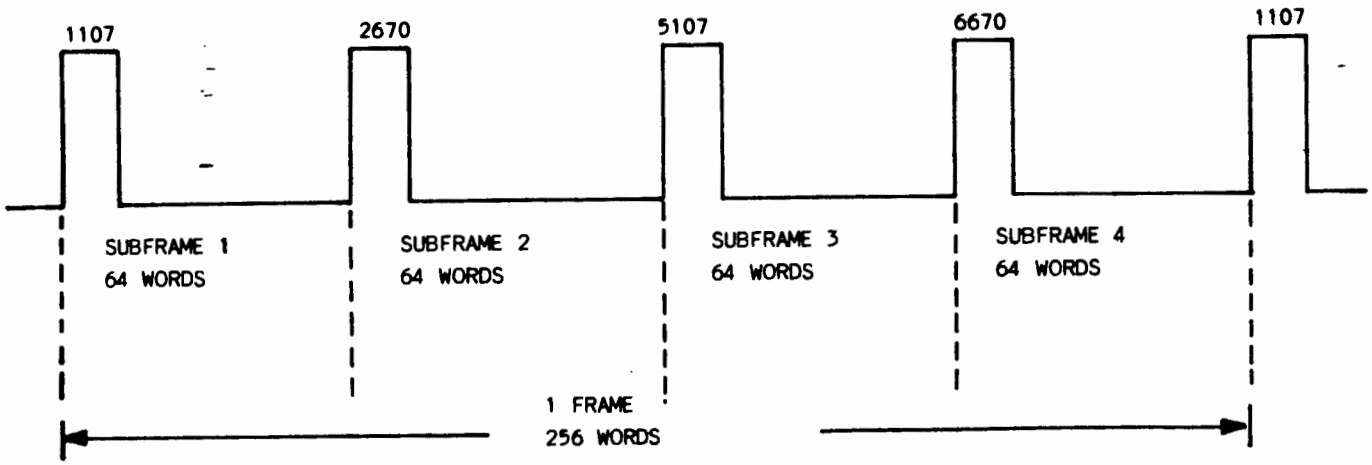


Figure 4.19 DFDR Data Stream

When the microprocessor sends data to the DFDR, it waits for a frame interrupt which occurs every 4 seconds. The microprocessor then sends 12-bit serial data words to the DFDR. The first data word is clocked into a register within the DFDR logic by means of the 3MHz clock.

The second data word is clocked into a second register within the DFDR logic while the first data word is being clocked out of the first register by signal C32. Whenever a buffer has been emptied, an interrupt signal, 64INT is provided to indicate to the processor that the DFDR interface is ready for another data word. The NRZ data is then converted to Harvard Bi-phase by the 768Hz signal C64B and the 1.536 KHz signal C128B and then appears on the DRH and DRL lines. The line drivers then provide the current to drive the transmission lines DFDRH and DFDRL to the DFDR.*

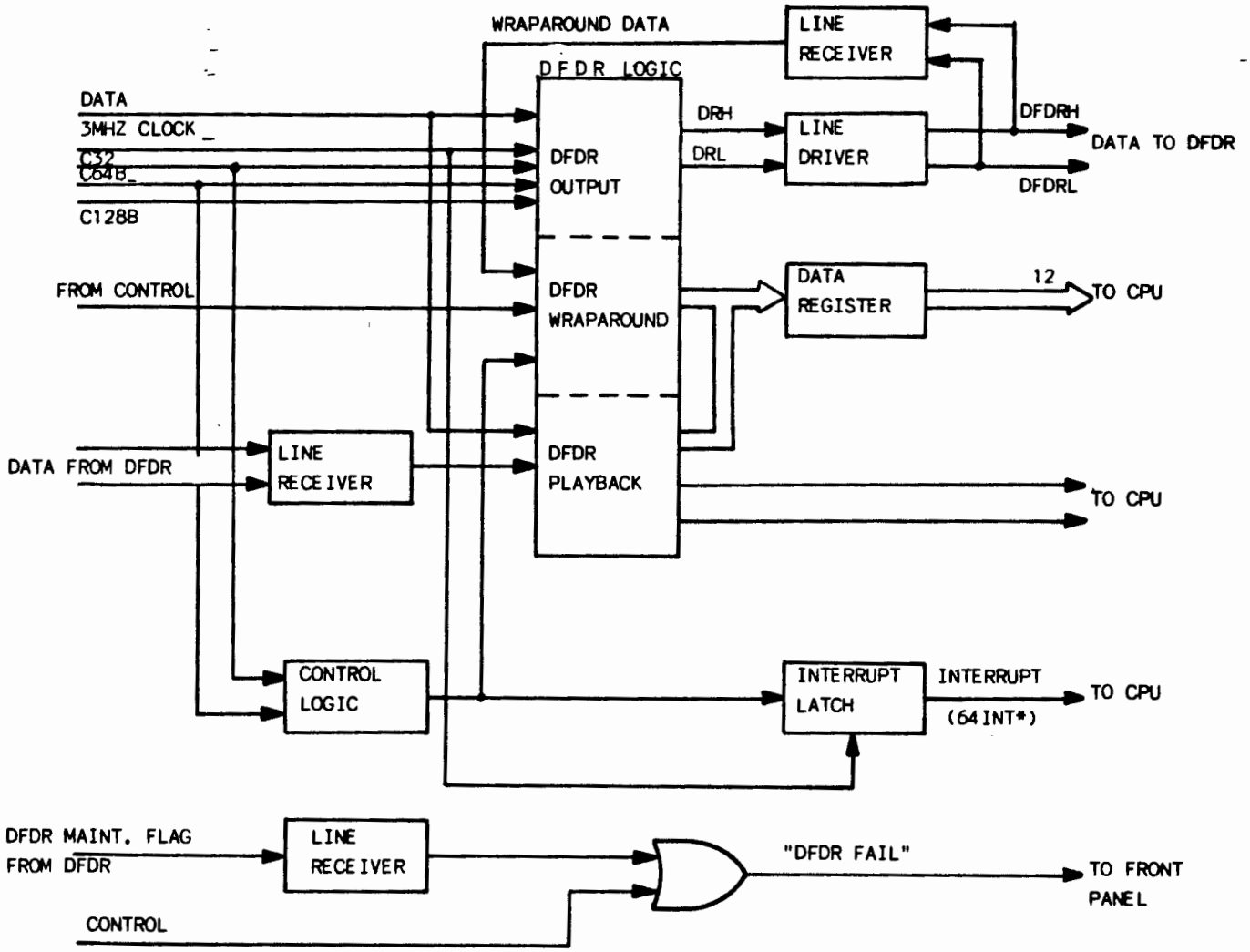


Figure 4.20 DFDR Interface Block Diagram

The DFDR output is double buffered to enable a wraparound check on the transmitted data which is fed back to the wraparound logic via the line receiver.

4.5.5 Bi-polar Auxiliary Output

The digital I/O module provides an auxiliary bi-polar R-Z interface which is shown in figure 4.21.

The interface can provide word rates of 64, 128 and 256 words per second. This is determined by the clock signals connected to the AUXWD lines as shown in figure 4.21. The word rate of this interface is decided by the type of auxiliary recording device that is being used in the system ie. 64, 128 or 256 words per subframe type.

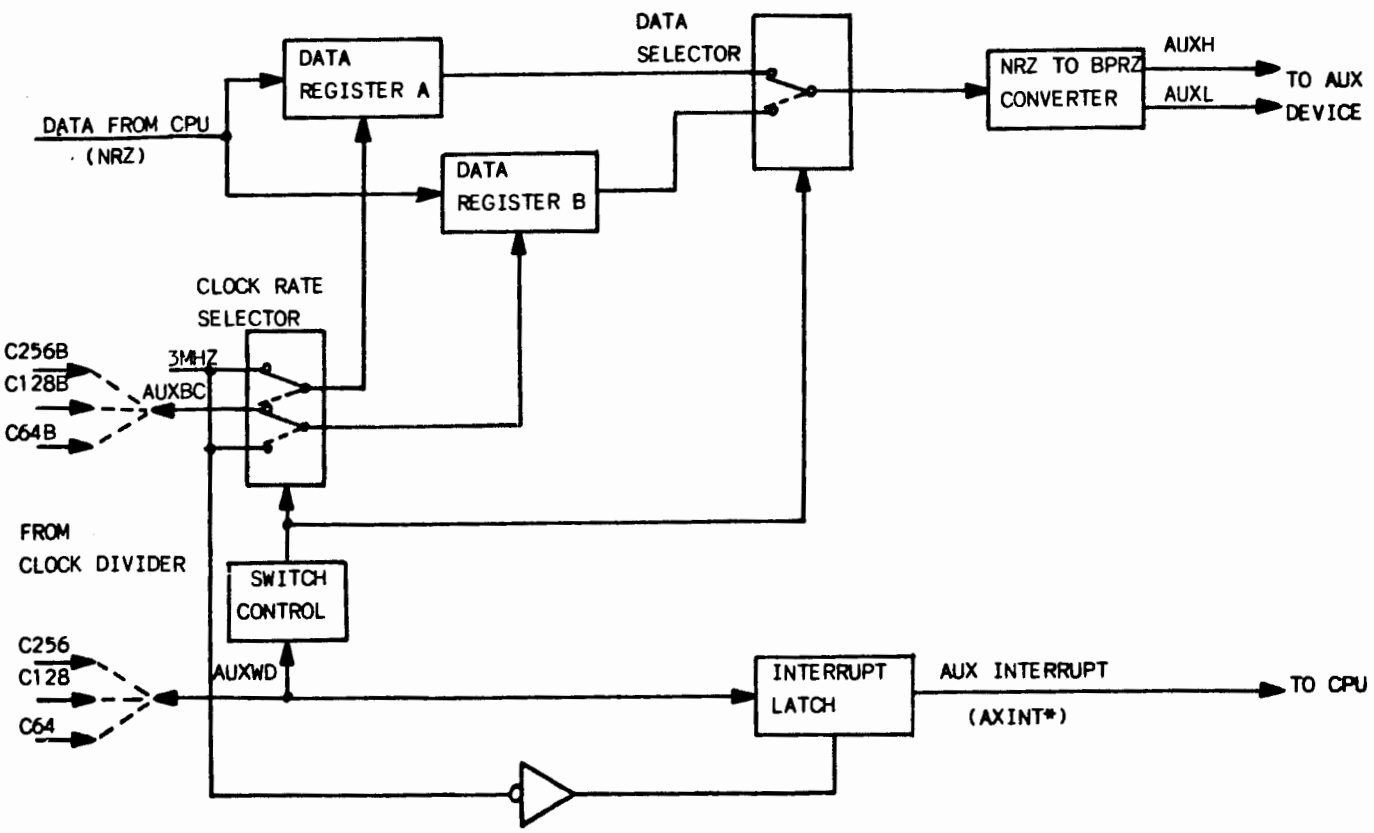


Figure 4.21 Auxiliary Interface Functional Block Diagram

The first 12-bit serial data word is clocked into register A by 3MHz clock. At this time AUXWD causes the interrupt latch to produce AXINT, an interrupt signal to indicate to the CPU that another data word can be accepted. Simultaneously, AUXWD causes the switch activator to force the clock rate selector and data selector to change switch positions.*

The second data word is clocked into register B by the 3MHz clock, and the first data word is clocked at a slower rate out of register A by the AUXBC clock.

When the clock clocked the first bit into the data register, the inverted clock resets the interrupt latch.

The NRZ data is converted into bipolar RZ data and sent to the device on the AUXH and AUXL lines. Bipolar RZ is illustrated in figure 4.22.

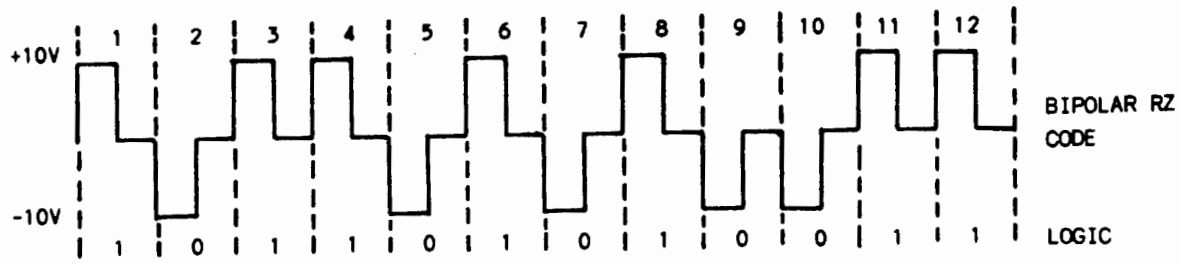


Figure 4.22 Bi-polar RZ code for AUX Interface

The AUXWD clock changes state again, causing a second interrupt signal and causing the clock rate selector to change switch positions. The third data word enters register A, while the second data word is clocked out of register B and the process continues.

The data rate on the AUXH and AUXL lines is determined by the clock signal connected to AUXBC and AUXWD lines, as described in Table 4.1.

Data Rate	Mnemonic	Frequency	Mnemonic	Frequency
256 wps	C256B	3.073 kHz	C256	256 Hz
128 wps	C128B	1.536 kHz	C128	128 Hz
64 wps	C64B	0.768 kHz	C64	64 Hz

Table 4.1 Data Rates vs Clock Rates for AUX Interface

4.5.6 Control/Indicator Panel And Interface

The indicator panel contains the status and failure indicators

for the DFDAU and the DFDR, and a switch to operate the status indicator display.

The indicator panel interface provides the means for the microprocessor to operate the indicators on the indicator panel. The panel consists of a 3-digit (hexadecimal) display, three indicator lamps and a READ switch, as shown in figure 4.23.

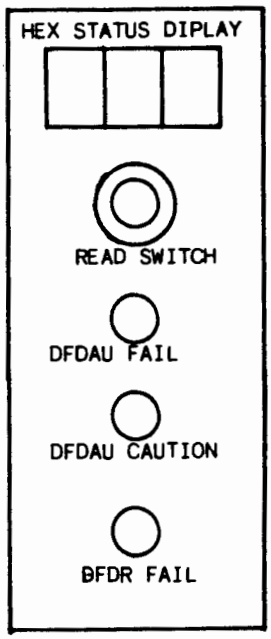


Figure 4.23 Control/Indicator Panel

The following are the conditions under which the software will cause the appropriate front panel indicator to be turned on:

The DFDAU FAIL indicator will light if the power supply voltages are out of tolerance, if the program memory has been corrupted or if the data being sent to the DFDR is incorrect (fails the wraparound check).

The DFDR FAIL indicator will light if the DFDR power supply voltages are out of tolerance, if tape motion is absent, if writehead current is absent, or if the DFDR is not installed.

The DFDAU CAUTION indicator will light up if any analog or ARINC 429 (DITS) input channel fails its wraparound test (see table 3.5).

By depressing and holding down the READ switch, the software initiates a front panel lamp test causing the 3-digit DFDAU status display to display 888, and the three indicators will light for 4 seconds. Then, while the read switch is still being depressed, the software will cause the DFDAU status display to sequentially indicate the codes of all system failures, up to a maximum of 16.

4.5.7 Asynchronous Communications Controller Interface

The Asynchronous Communications Controller interface contains differential-to-single-ended line receivers and single-ended-to-differential line drivers. These provide an RS422 interface to and from any standard RS422 device connected to the DFDAU, such as a FDEP display panel or a standard CRT terminal via an RS232-to-RS422 interface.

4.5.8 ARINC 429 Interface

The ARINC 429 Interface provides the necessary circuitry for the CPU to communicate with external serial ARINC 429 devices, as well as provide an output to the DITS Interface module for wraparound testing.

4.6 ANALOG SECTION

4.6.1 General

The analog section of the DFDAU is made up of the following modules:

- Analog Multiplexer Module (AMX)
- Analog to Digital Converter Module (ADC)
- Discrete Multiplexer Module (DMX)
- Very Low Level DC/Tachometer module (VLLDC/TACH)

The above modules can be further subdivided into the following building blocks:

- First Level Multiplexers
- Second Level Multiplexers
- Programmable Gain Amplifiers
- Precision Voltage References
- Sample and Hold Circuits
- Analog-to-Digital Converter

The analog section contains its own internal calibration and utilizes internally generated signals to perform system verification.

The input multiplexers, programmable gain amplifiers, reference select multiplexers and all the other functions that the DFDAU requires to be set up to process an input, are addressed by the microprocessor. The system is self-calibrated according to the type of input being received. The parameter to be measured is applied to the ADC via the AMX, processed and output as a 14-bit digital word. The software then converts it to either a 10, 11 or 12-bit data word.

4.6.2 Analog Multiplexer

The Analog Multiplexer module is capable of accepting 22 channels of any of the following types of signals:

- Synchro Angle
- AC Voltage Ratio 1
- AC Voltage Ratio 2
- Potentiometer
- DC Voltage Absolute
- DC Voltage Ratio 1 and 2
- Very Low Level DC Voltage
- Thermocouple
- Temperature Bulbs

Figure 4.24 shows a functional block diagram of the AMX module.

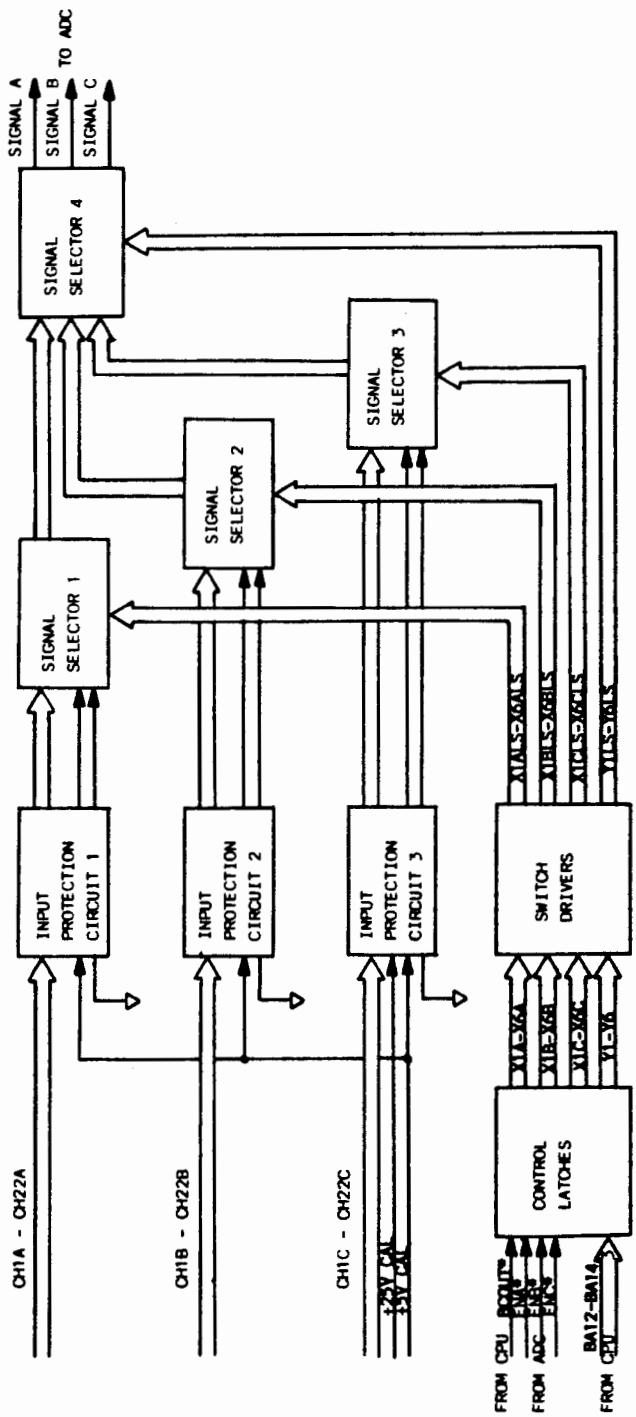


Figure 4.24 AMX Module Functional Block Diagram

An analog channel is made up of a 3-wire input, i.e signal, common and reference.

Each data input is designed to provide a high degree of isolation between the DFDAU and the monitored aircraft parameter. The input circuit shown in figure 4.25 protects the DFDAU from the application of excessive voltages. Also, the circuits protect the aircraft system sensors from damage due to any type of failure that could occur in the DFDAU.

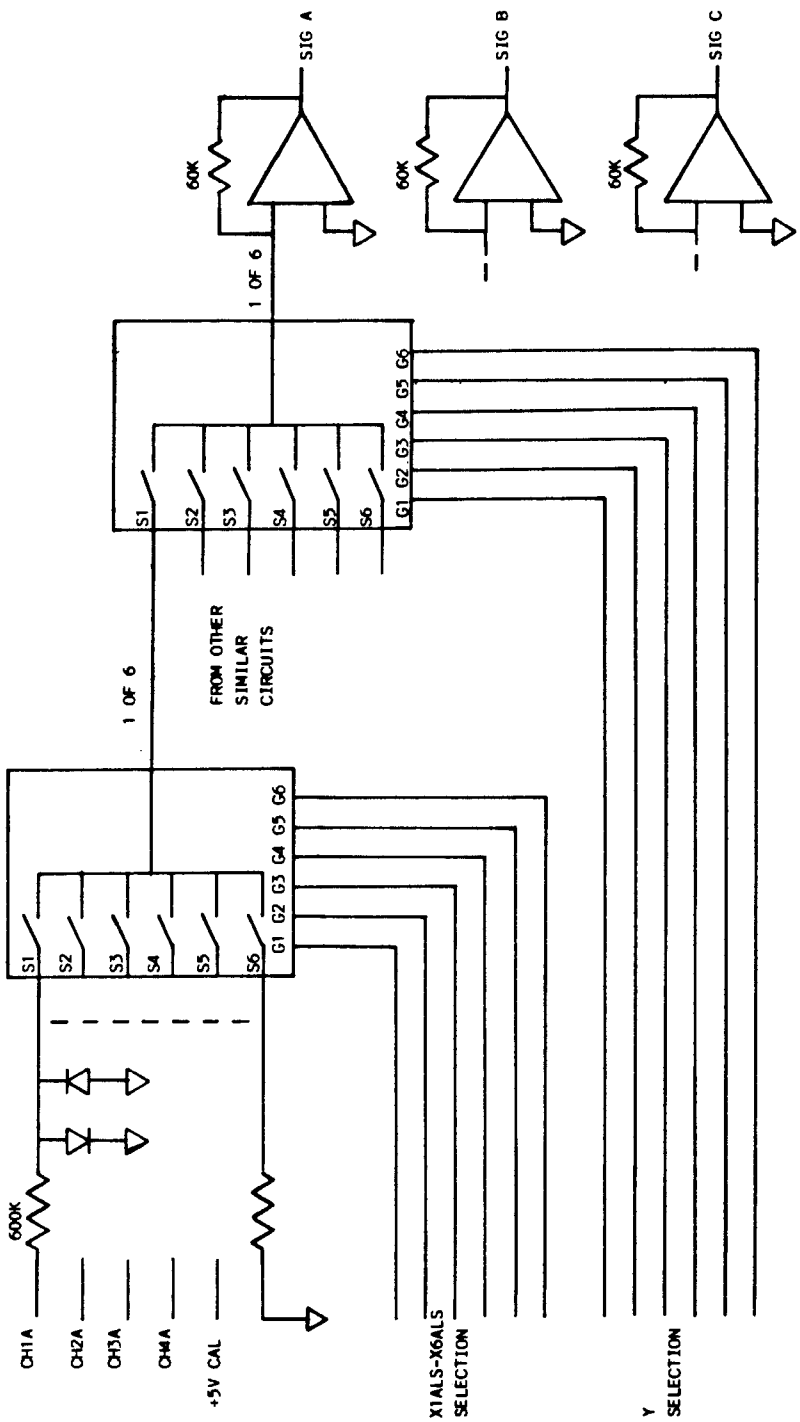


Figure 4.25 Analog Multiplexer Input Circuit

In figure 4.25, only the A-inputs for channels 1 to 4 are shown, the rest are identical. Since the input protection circuitry is passive, the bi-directional isolation capability is never altered due to power supply failure. Hence, inadvertent application of any excessive voltage, AC or DC, will not harm or degrade the input circuitry.

Signal selectors 1,2,3 and 4 are all identical. Selectors 1,2 and 3 provide the first stage selection and 4 provides the second (final) stage selection. Considering selector 1, when signals X1ALS through X6ALS go low, input S1 through S6 respectively are selected and appear at the output. Only one input can be selected at a time. The selectors are analog switches Intersil G117.

The 3 buffers are identical for input signals A, B and C. They are operational amplifiers with a gain of 60/600 = 0.1.

The switch drivers are open collector devices which convert the TTL levels from the control latches to +5V and -15V levels required to drive the analog switches.

The control latches receive data from the microprocessor on the BCOUT line to determine which analog input signal will be selected. The control latches are enabled when ENA* through ENC* are low; whereas the BA12 through BA14 lines select at which output (Q0 through Q7) the BCOUT* signal will appear.*

Figure 4.26 shows the control latch and switch driver arrangement in a little more detail.

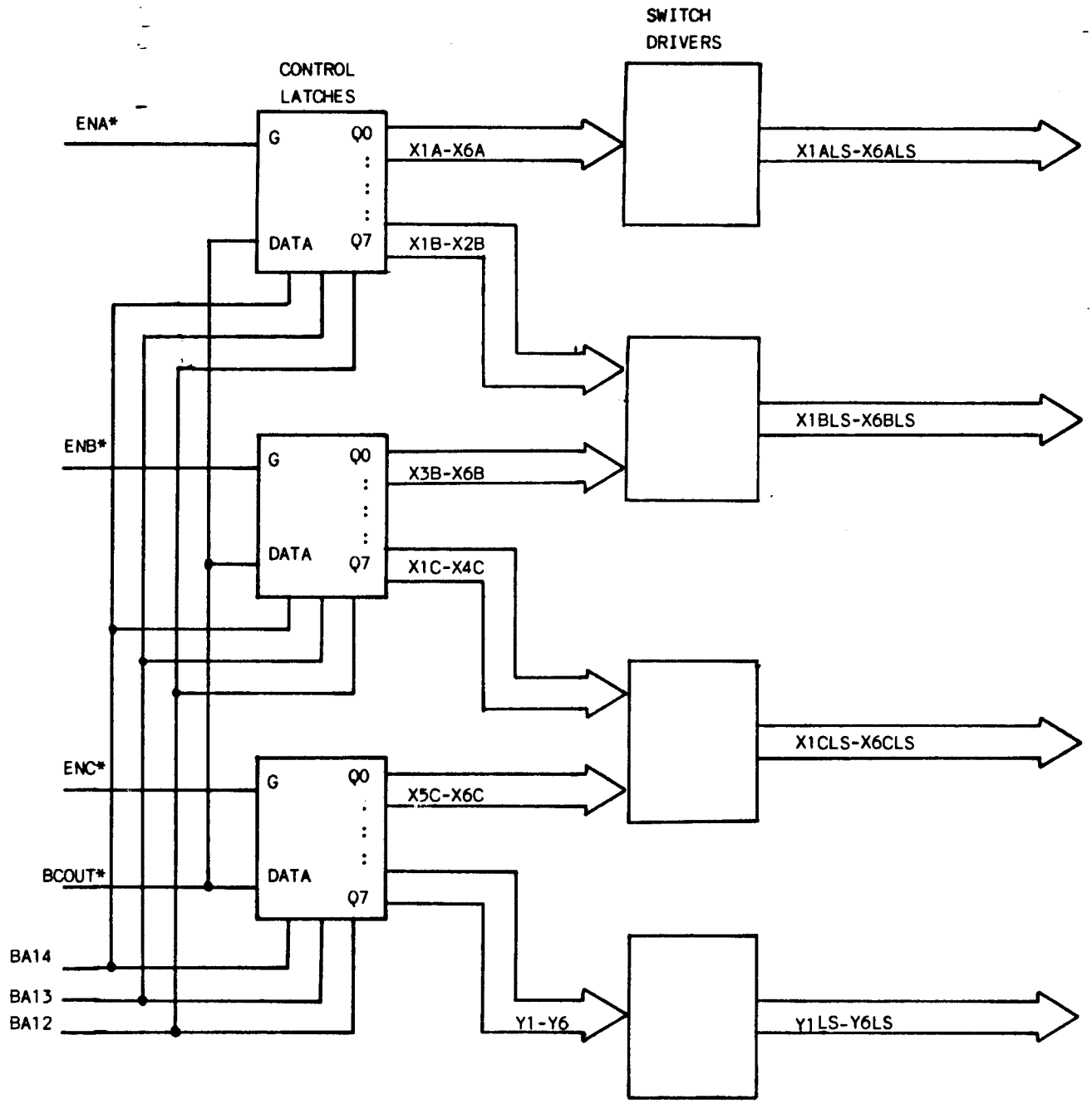


Figure 4.26 Control Latch/Switch Driver Arrangement

4.6.3 Analog-to-Digital Converter Module

The ADC module is capable of accepting data from up to three Analog Multiplexer modules and a low level DC signal module. A block diagram is shown in figure 4.27.

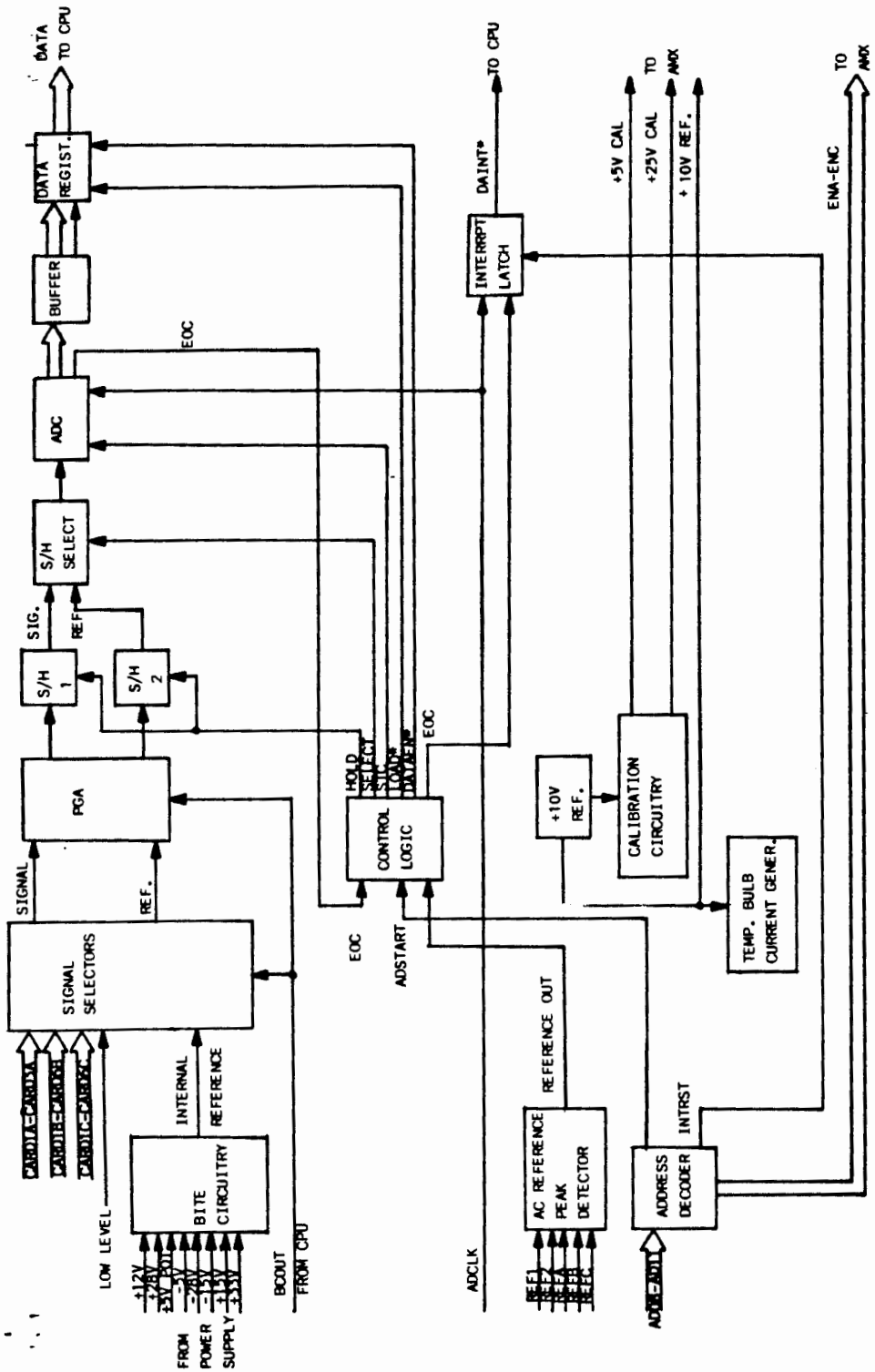


Figure 4.27 ADC Module Functional Block Diagram

The signal selectors accept data from AMX modules 1, 2 and 3, the low level signal module and the internal reference from the BITE circuit. The signal selector is controlled by a control latch which is addressed by the microprocessor.

The BITE circuitry conditions each of the power supply voltages and places the resultant voltages, as required, on the INTERNAL REFERENCE line. The selection is executed by a control latch which is addressed by the microprocessor.

The signal inputs to the programmable gain amplifiers (PGA) need to be conditioned so that the ADC never sees more than +/- 5V. Referring to figure 4.28, the gain is the ratio RF/RS and programmable gain is accomplished by switching in different values for RS. The programmable gain amplifier block is shown in in figure 4.29.

A programmable gain amplifier arrangement is shown in figure 4.28.

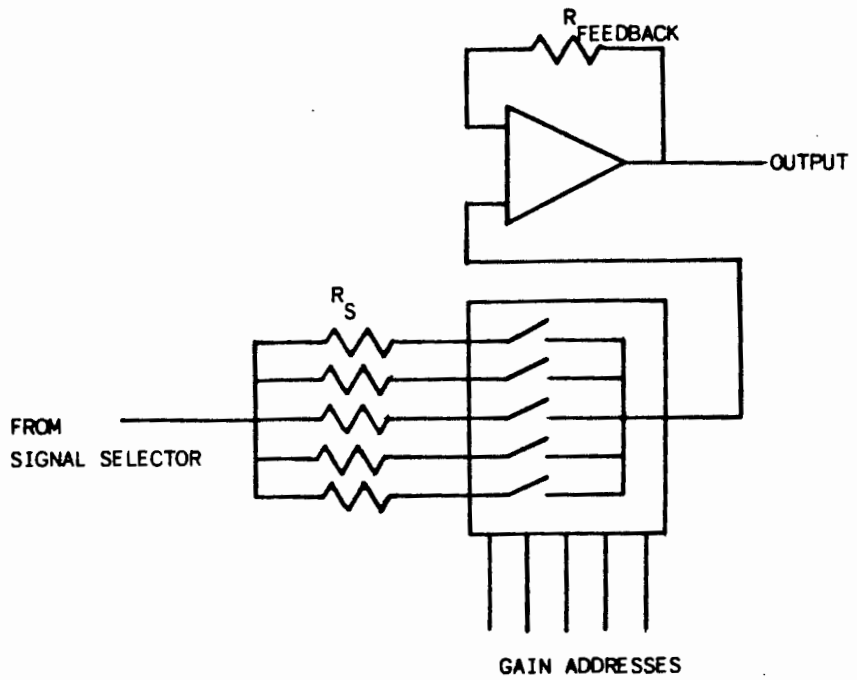


Figure 4.28 Programmable Gain Amplifier

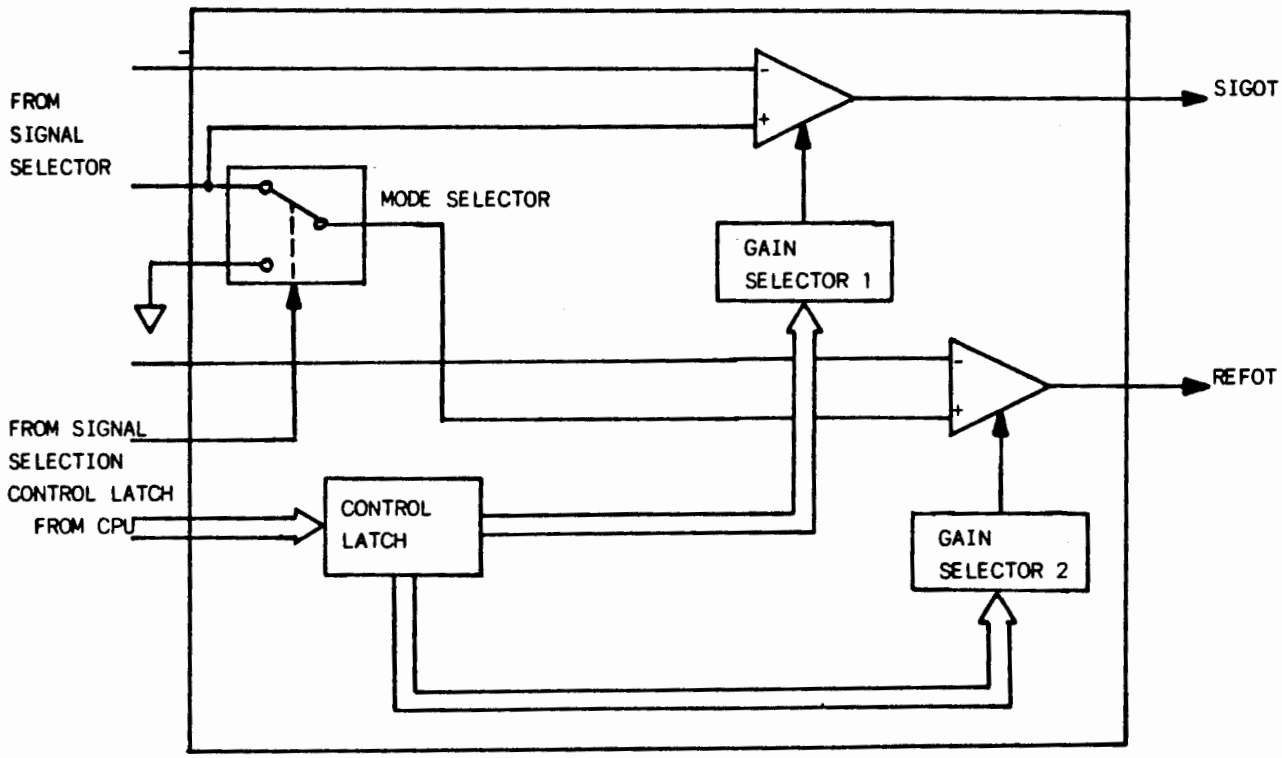


Figure 4.29 PGA block

The gains of the PGA'S are controlled by the gain selectors 1 and 2. The gain selectors are controlled by the control latch which is addressed by the microprocessor. The mode selector is controlled by a signal from the signal selector control latch and connects the amplifiers for a 3-wire (signal/reference/ground) measurement such as a ratio or a synchro measurement, or for a 2-wire differential measurement.

The Sample-and-Hold (S/H) amplifiers are controlled by the HOLD line from the control logic and continuously sample the signals on the SIGOT and REFOT lines until the HOLD signal goes high. Then they hold that level (when HOLD goes high) for a sufficient time to allow the actual analog-to-digital converter to perform a conversion. The S/H selector is controlled by the HLDSEL line from the control logic and selects the output of either one of the two S/H amplifiers to be presented to the ADC.

The ADC starts a conversion when the STC signal from the control logic goes high. The ADCLK signal clocks the converted data into the output registers within the ADC. When the conversion is complete, the ADC sends an end-of-conversion signal EOC to the control logic to indicate that conversion has been completed.

The_ output buffers provide the current drive to the data registers. Data enters the data registers when the LOAD* signal from the control logic goes low.

The interrupt latch is controlled by the control logic and produces an interrupt signal, DAINT*, to inform the microprocessor that the data is ready in the data register. The microprocessor responds by producing an INTRST* signal via the address decoder to reset the interrupt latch and then causes the DATAEND to go low. This in turn causes the data register to place its contents on the data bus DB00 through DB15.

The ac-reference peak detector converts the five aircraft-generated 400HZ reference signals into TTL squarewave signals and presents them as requested, to the control logic. These signals are used for synchro and ac ratio measurements.

The calibration circuitry produces two calibration voltages +5VCAL and +25VCAL, from a +10V precision reference signal generated by the +10V reference generator. These two calibration signals are required by the AMX module.

Similarly, the temperature bulb constant current generators produce sixteen precision currents for the aircraft temperature bulbs, from the +10V precision reference signal.

4.6.4 Discrete Multiplexer

The Discrete Multiplexer is designed to accept 60 discrete signals, (channel 1-60). The channels 1 through 30 are series discretes, channels 32 through 54 are shunt discretes, channels 55 and 56 are marker beacon discretes and channels 57 through 60 are ac discretes. The Ident discretes are part of the series discrete channels.

A block diagram of the Discrete Multiplexer module is shown in figure 4.30.

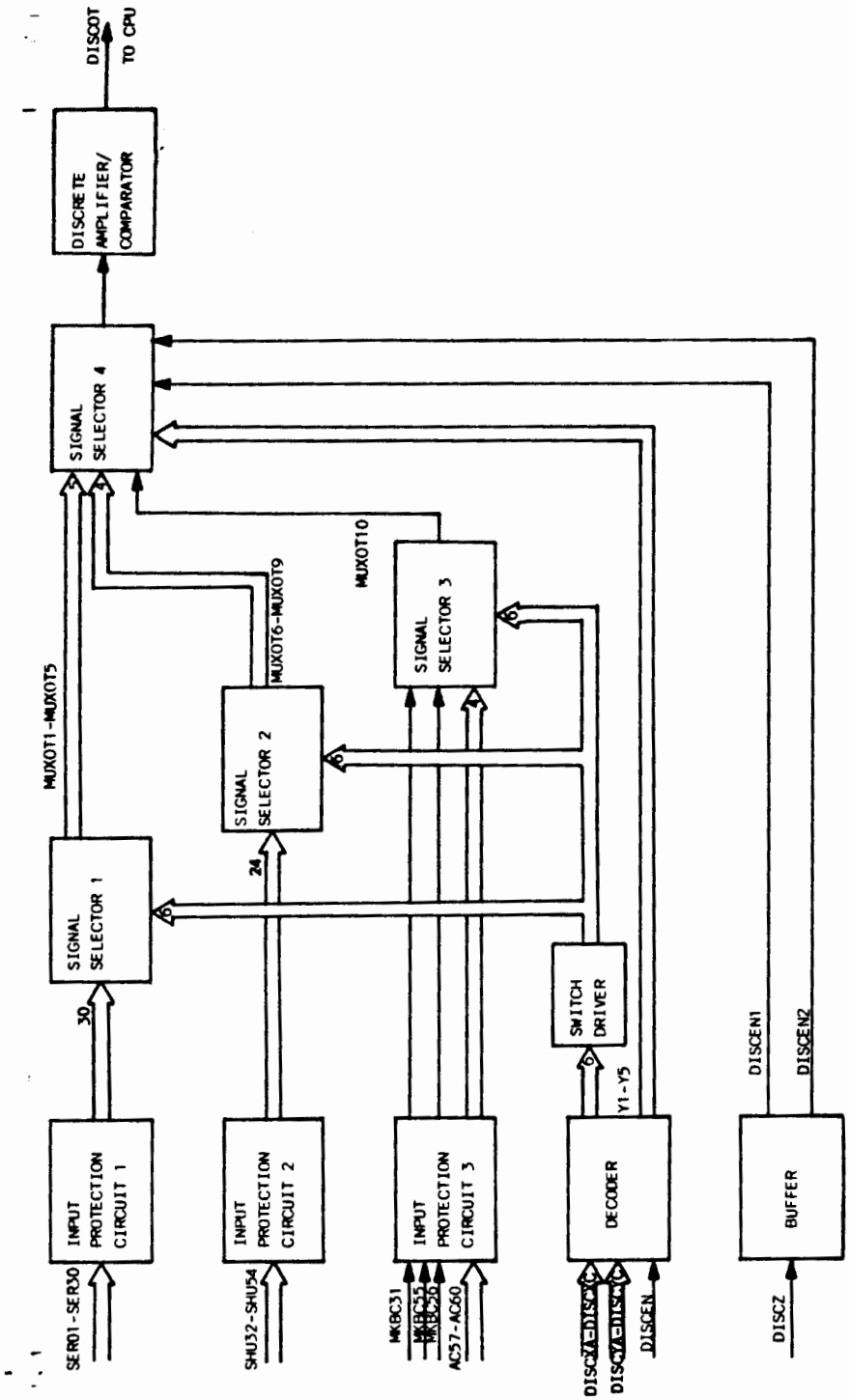


Figure 4.30 Discrete Multiplexer Module

The signal selectors are controlled by the decoder. It develops control signals from the DISCXA through DISCXC, DISCYA through DISCYC, and DISCEN signals from the ADC module, which select the desired input channel to appear at the input of the discrete amplifier. The buffer develops two signals, DISCEN1 and DISCEN2 from DISCZ signal, which enable the proper section of the signal selector 4.

The signal selector blocks are made up of a number of MOSFET switches, each having 6 inputs and one output, the selection being done by 6 select lines. Consider figure 4.31.

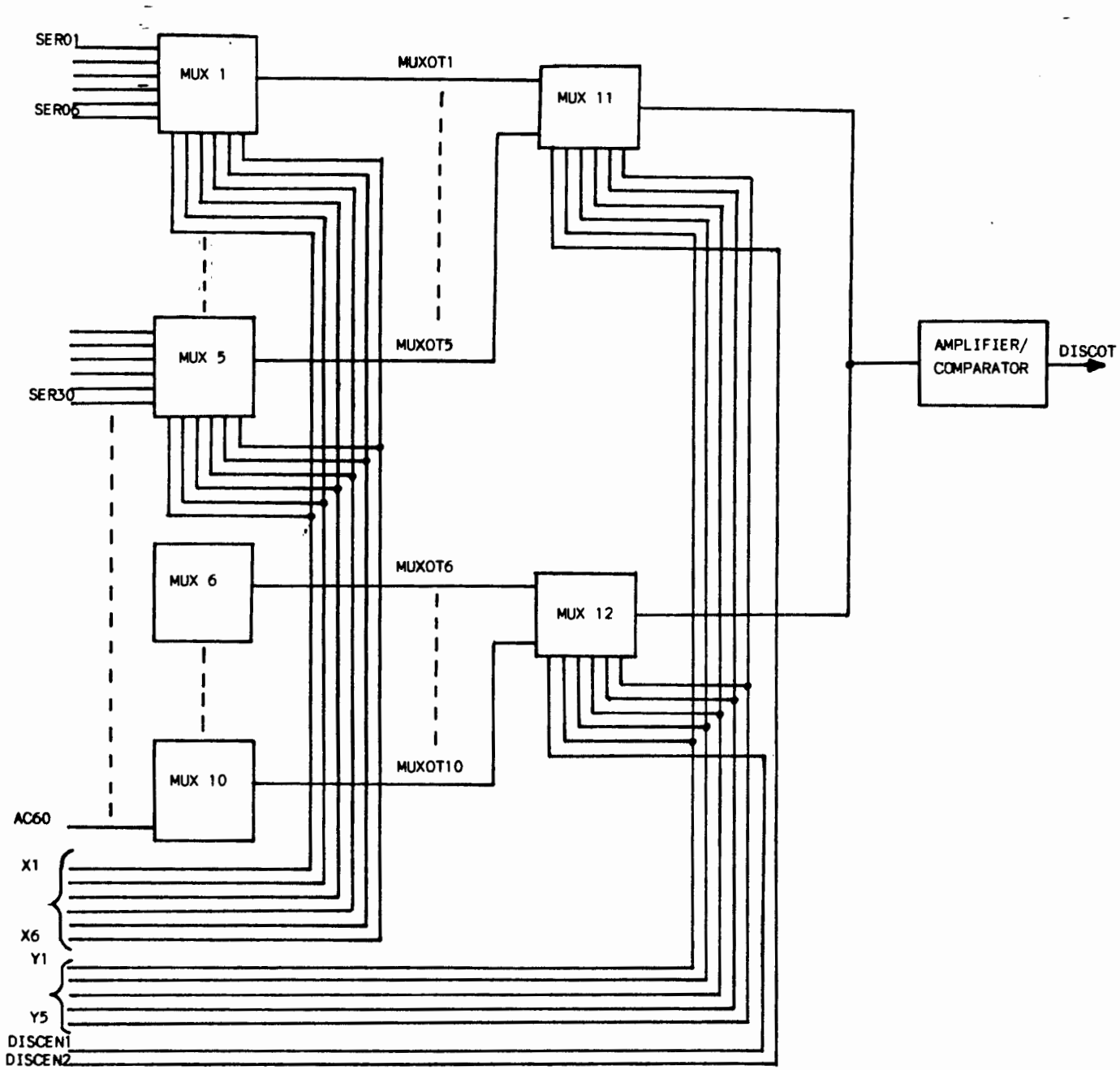


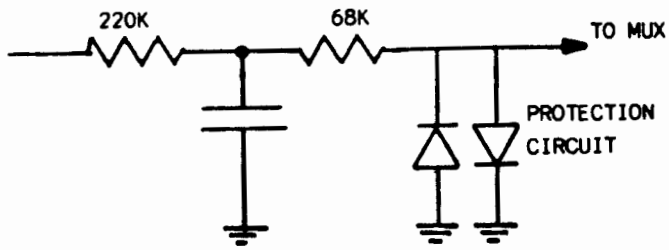
Figure 4.31 Discrete Multiplexer Signal Selectors

Since multiplexers 1 through 10 are identical, consider multiplexer 1. When X1 through X6 signals go low, channels 1 through 6 respectively are selected and appear on the MUXOT01 line. Only one channel can be selected at a time. Multiplexers 1 through 10 form the first level channel selection.

The second stage of selection is formed by multiplexers 11 and 12. Since they are identical, consider multiplexer 11. When signals Y1 through Y5 go low, signals MUXOT1 through MUXOT5, respectively are selected. However, they only appear at the output if the DISCEN signal is low. Similarly for multiplexer 12 and DISCEN2. Thus only one channel at a time is presented to the amplifier.

The different types of discrete signals will now be described.

4.6.4.1 Series Discretes -



LOGIC

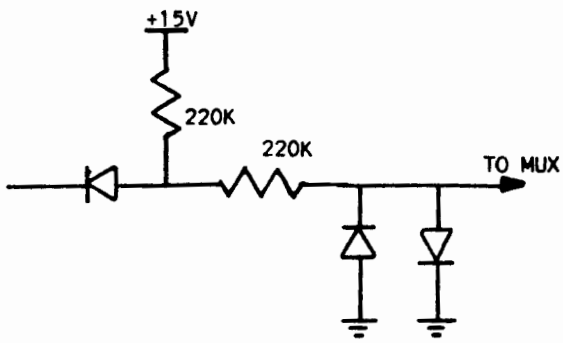
1	0
>7V	<3V
	OPEN

Figure 4.32 Series Discretes

The series discrete receives a voltage input from the device it is connected to, in the aircraft. When a voltage greater than +7V is present, a logic 1 will appear at the output of the comparator when that channel is addressed. When a voltage <3V or the line is open (>100Kohms to ground), a logic 0 will appear

at the comparator output.

4.6.4.2 Shunt Discretes -



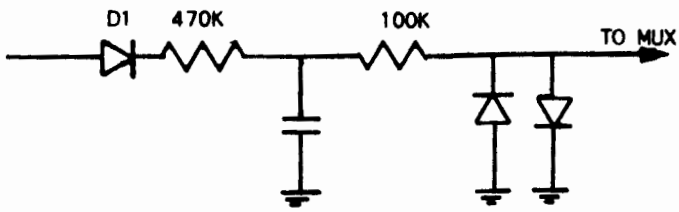
LOGIC

1	0
>7V OPEN	<3V

Figure 4.33 Shunt Discretes

The shunt discrete supplies its own voltage source. When the input is grounded, the voltage is reduced to about 0.7V (diode reverse voltage) above ground. When the input is open, the current goes into the amplifier and the output is a logic 1 when the channel is selected. When an input >7V is applied, the output is again a logic 1 when the channel is selected. When the input is grounded, or a signal <3V is applied, the output goes to a logic 0.

4.6.4.3 AC Discretes -



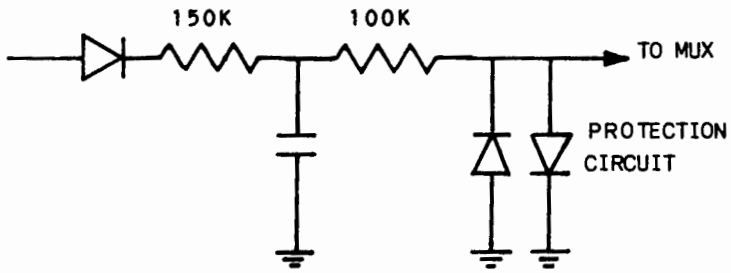
LOGIC

1	0
>18V	<7V

Figure 4.34 AC Discretes

The AC discretes will respond to an input voltage, AC (400Hz) or DC over the range 0-120V. Half-wave rectification is used so that only the positive excursions of the AC signal pass through. When a voltage <7V or ground is applied, the comparator output is logic 0.

4.6.4.4 Marker Beacon Discretes -



LOGIC

1	0
>2.5V	<1.5V

Figure 4.35 Marker Beacon Discretes

The three marker beacon discretes are designed to operate over the range 400Hz to 3.0 KHz. A logic 1 will be output when the input is >2.5V and a logic 0, when the input is <1.5V.

The marker beacon sensors respond to the signals generated by the inner, middle and outer markers along the aircraft's instrument approach path on the ground (reference 14).

4.6.5 Very Low Level DC/Tachometer Module

The VLLDC/TACH module performs two functions. The tachometer portion will accept four differential inputs from tachometer type transducers of frequency 0.7Hz to 70Hz.

The VLLDC portion accepts nine channels of very low level dc input in the range of -400 to +400 millivolts It performs the necessary channel selection and presents the result to the ADC module. Of the nine input channels, one is reserved for the calibration voltage, and another is reserved for ground calibration.

4.6.5.1 Tachometer Section -

Looking at figure 4.36, the input circuitry filters the signals and presents them to a 4-to-1 multiplexer. The signals are then used to start and stop a counter that counts a precise clock frequency. A 12 bit counter is used and its output, along with two status bits, and two select bits are output to the data bus from the data bus buffer.

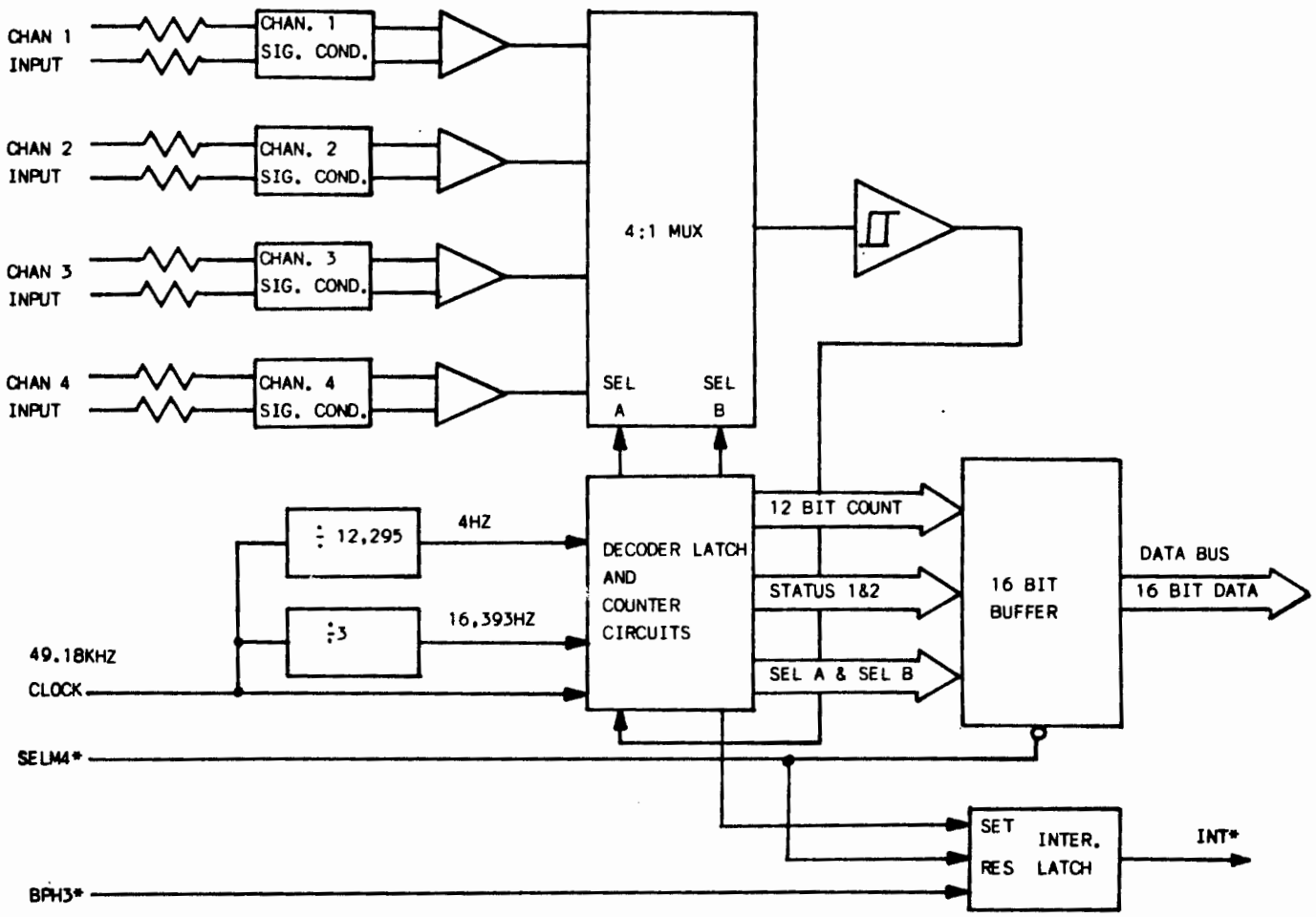


Figure 4.36 VLLDC/TACH Module
Tachometer Section

Each channel is read once per second. At each 1/4 second period, an interrupt is generated and sent to the CPU. The CPU responds

by sending a read signal, which places the contents of the data bus buffer onto the bus, and at the same time, resets the interrupt latch.

The status bits describe; if there was no tachometer signal, only a start pulse, overflow, or when there was a valid count (one start and one stop pulse in 1/4 second). The two select bits encode which of the four channels is being currently read.

The tachometer clock (49.18Khz) is the main counting frequency. The clock is divided by 3 (1/3 f1) and used as an "expanded scale" clock. Looking at figure 4.37, when the 12 bit counter reaches a count of 2048, (bit 12 active), the 1/3 f1 clock frequency is selected and the slope of the count frequency changes.

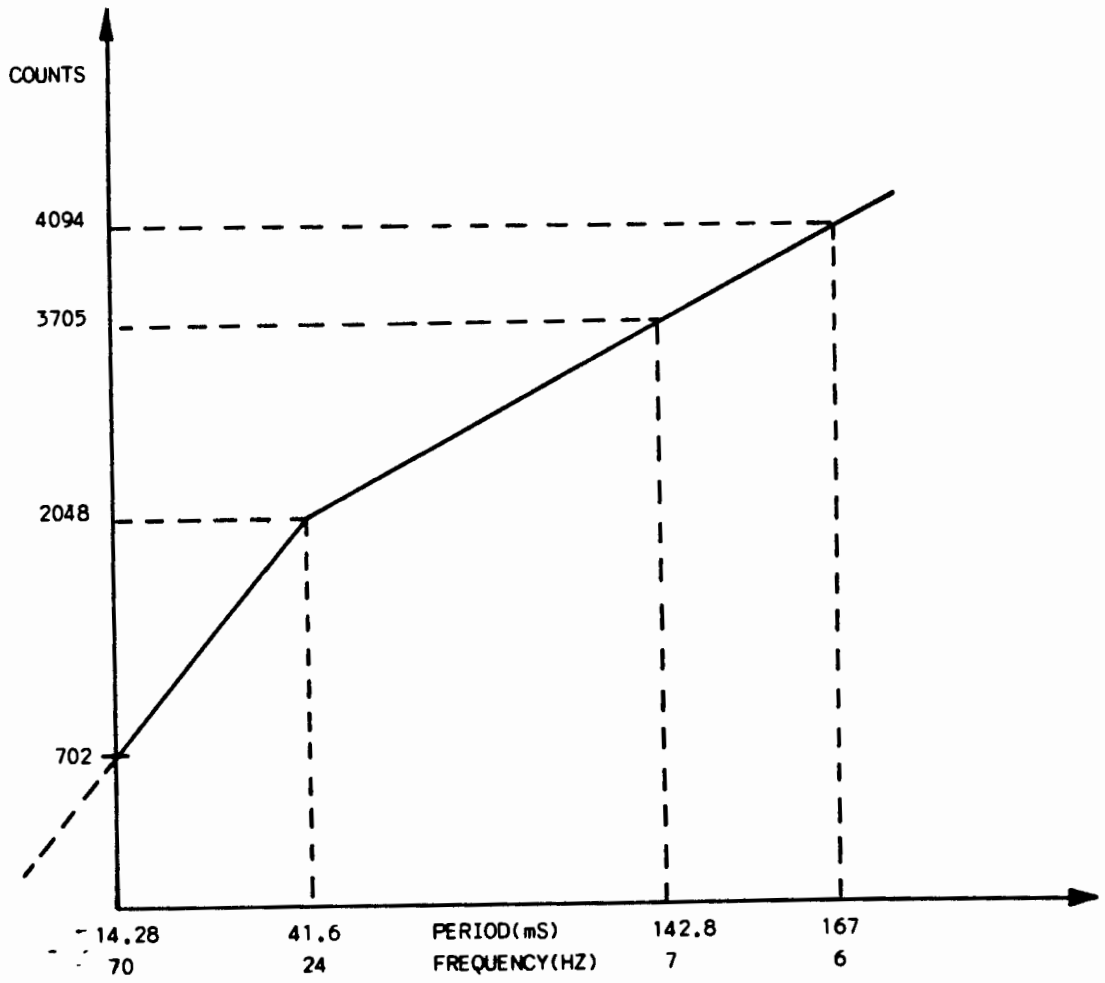


Figure 4.37 Output Counts vs Input Frequency

The 4Hz frame counter is also a division of (divide by 12,295) the main clock. Figure 4.38 shows a timing example of the Tachometer Module.

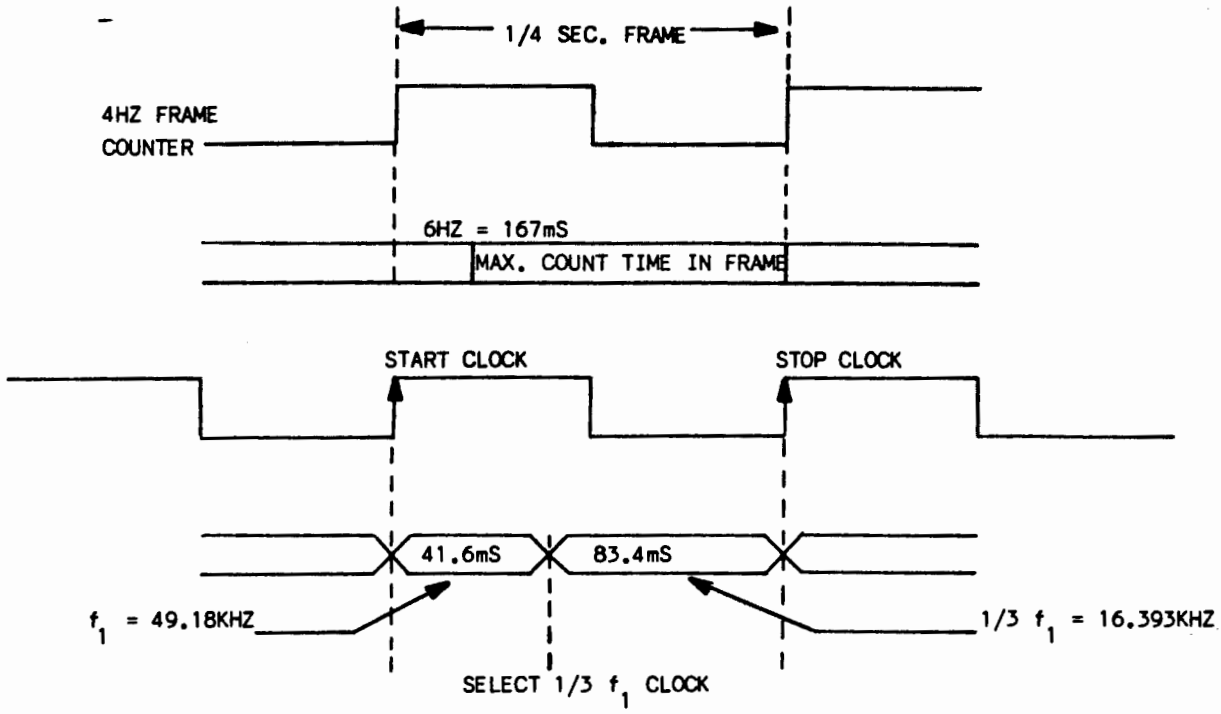


Figure 4.38 Tachometer Timing Example

The 16-bit output data word is available at memory map address FFC0. A read at this address places the data onto the data bus and resets the interrupt. The output data word is shown in figure 4.39.

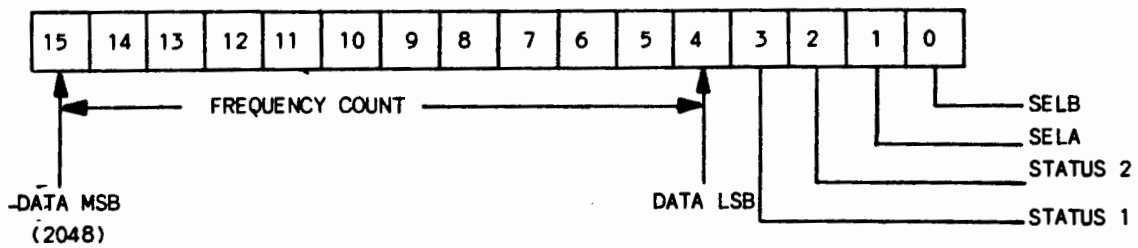


Figure 4.39 Tachometer Output Data Word

The two status bits describe the following states that can occur.

Condition	Status 1	Status 2
1. Only one rising edge	0	0
2. No rising edge	1	0
3. Two rising edges (normal)	0	1
4. Overflow	1	1

Condition 1 : Caused by tachometer speed of less than 4Hz. (period greater than 250 milliseconds).

Condition 2 : Caused by very slow tachometer speed or no input.

Condition 3 : All speeds within range.

Condition 4 : There were 2 rising edges. However the tachometer speed was less than 6Hz, but greater or equal to 4Hz.

The two select bits are decoded as follows:

Channel	SEL A	SEL B
1	0	0
2	1	0
3	0	1
4	1	1

4.6.5.2 Very Low Level DC Section (VLLDC) -

A functional block diagram of the VLLDC section is shown in figure 4.40.

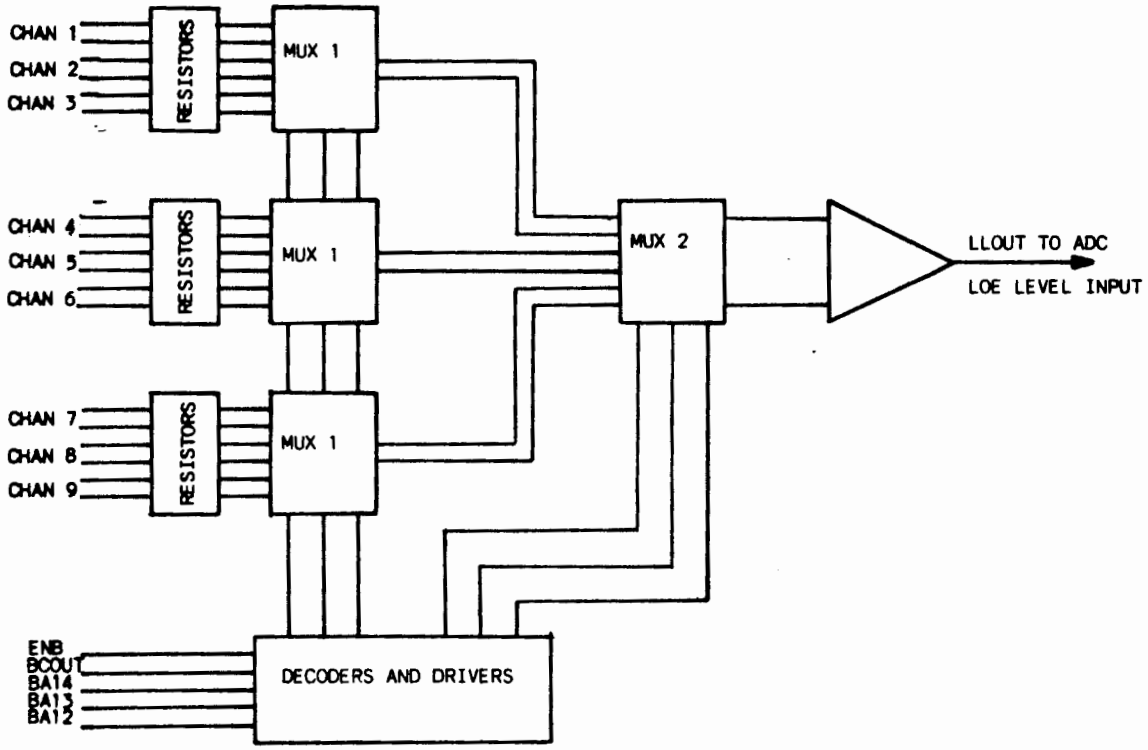


Figure 4.40 VLLDC/TACH Module
VLLDC Section

The VLLDC section has nine available input channels, seven of which are used for differential low level dc input, one for reference voltage calibration and one for ground reference calibration.

It consists of three first level multiplexers, one second level multiplexer, an output amplifier and a decoder/driver section.

The calibration channel voltage is 320mV generated by dividing down from the +10V precision reference on the ADC module. When corrected with the value of the ground channel, it will determine the system gain.

The input range is -400mV, with an output count of 0000 to +400mV, with an output count of 4095 (full scale). This is shown graphically in figure 4.41.

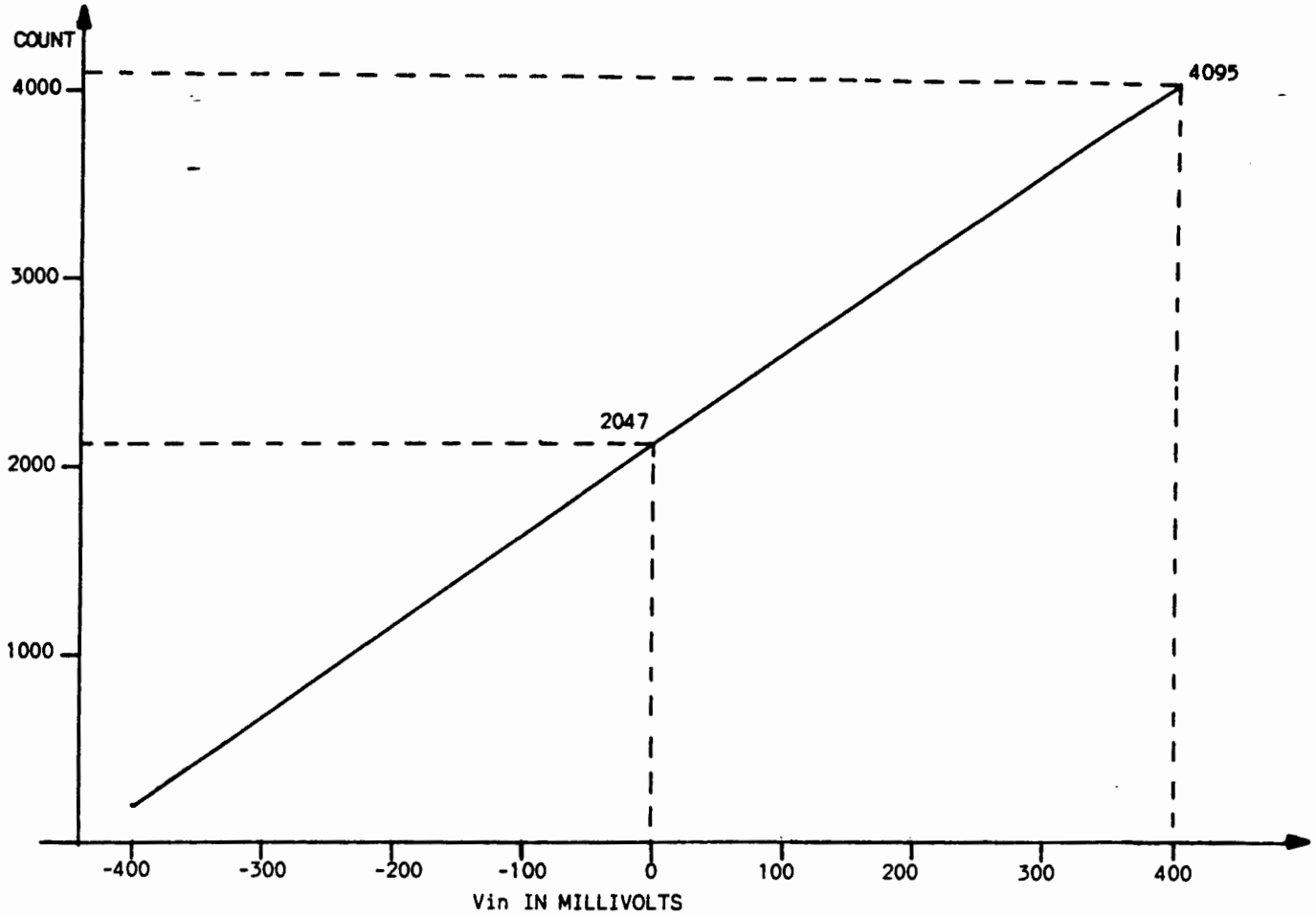


Figure 4.41 VLLDC Input vs Output

The binary count equivalent of the input voltage is calculated as follows:

$$Vin = \frac{Vtransducer - Vground}{Vcalibration - Vground} * VCAL$$

where VCAL = 320mV

*Counts = (Vin + 400) * 5.119*
(Vin in millivolts)

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 DFDAU SOFTWARE OVERVIEW

5.1.1 Introduction

The overall design philosophy of the DFDAU software is based on the fact that it is a real-time interrupt-driven system. Therefore, the central "kernel" of the software system is a background executive controlled by a real-time clock, and a number of interrupt service routines to process the various I/O requests. A favourable aspect of a system like the DFDAU is that each parameter to be acquired has a set of well-defined specifications with regards to its source, required accuracy, output destination and data field length.

Thus, a data base, for each of the three types of parameters consisting of a number of look-up tables can be set up, where the characteristics of each individual parameter are described. Each parameter to be acquired can then be assigned a "parameter id" that can be used as the entry key into one of the three data bases (or look-up tables) from where the processing, formatting and output information for that parameter can be retrieved. The process then becomes one of table-driven operations instead of one where the processor has to make multiple decisions with regards to the parameter operations.

The trade-offs to be considered in this situation are speed and memory usage. The primary constraint in the DFDAU (as with any real-time system), is the speed of operation, where a certain number of tasks have to be executed within a finite time "slice". In a table-driven situation, memory usage is relatively high due to the fact that long and intricate tables have to be set up to describe the operations that have to be performed, but execution time is low because the processor does not have to execute slow decision-making operations. In a non-table-driven approach however, memory usage is low but the system processing speed tends to be rather high (Reference 5).

For this reason therefore, the table-driven approach was chosen for the DFDAU software, where processing speed was the primary consideration and memory usage was of secondary importance.

Another advantage of this approach is that of ease of software modification. If the specifications of a parameter that is being acquired change, or if parameters have to be added or deleted from the system, the only change that has to be made to the software, in most cases, is a change in the parameter data base.

This chapter deals with functions and operations of the DFDAU software. The first section gives an overview of the software design in terms of the overall activity, data flow and data processing. The second section breaks the software down into its functional elements. Each functional element is then described in terms of its functions and main component modules, with reference to the system requirements discussed in chapter 3. The individual modules are then described with the aid of module trees, Program Design Language (PDL) descriptions, control and data flow diagrams.

5.1.2 General

The DFDAU software controls the overall operation of the DFDAU. It acquires data from the analog, discrete and digital (DITS) input signals, applies the necessary procedures such as calibration, scaling, correction and formatting on the received data, formulates the specified data frames and outputs these data frames to the DFDR and other designated output devices.

It continuously checks the integrity of the DFDR output data, indicates this and other malfunctions to the front panel, and upon receiving a command from the user, displays the DFDAU status.

The software also provides the communications interface between the two processors in a dual processor system.

Figure 5.1 shows a partial I/O block diagram of the DFDAU software.

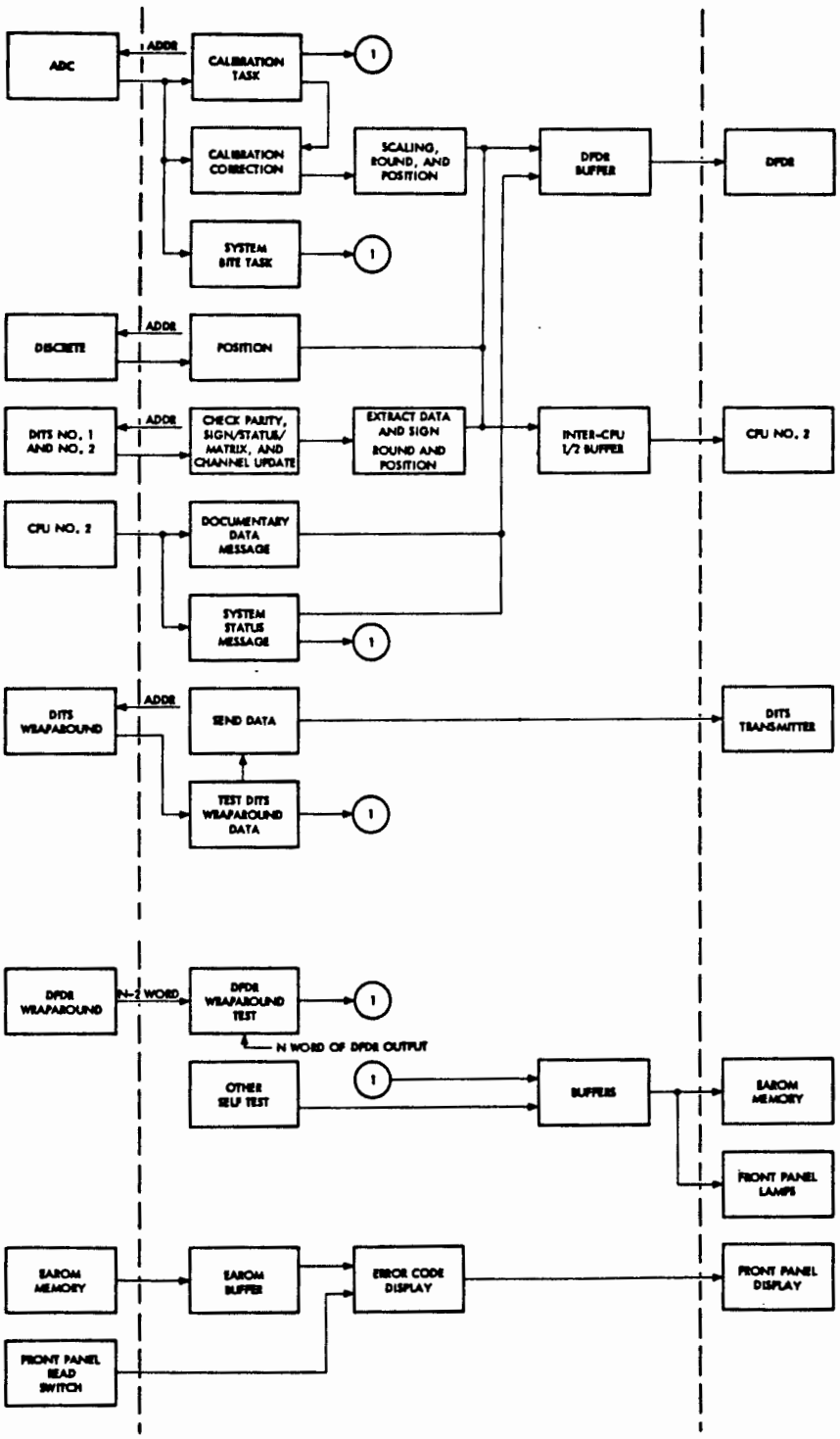


Figure 5.1
DFDAU Partial I/O Block Diagram

5.1.3 Overall Activity

The DFDAU activity is comprised of four data handling blocks and one control block as shown in figure 5.2. The raw data, analog, DITS and discretes are passed to the "PROCESS DATA" block where gain control, analog-to-digital conversion, trigonometric functions, truncation and rounding are applied as needed. This data is then sent to the "FORMAT AND BUFFER DATA" block where it is accumulated into blocks ready for output.

The data is also made available for use by the "PROGRAM MONITOR" which uses the results to determine the state of health of data acquisition hardware. Formatted data is placed in buffers ready for output to the recorder/transmitter devices by the "OUTPUT DATA AND COMMANDS " block. Also made available for transmission is the health of the DFDAU both to the front panel and to the DFDR.

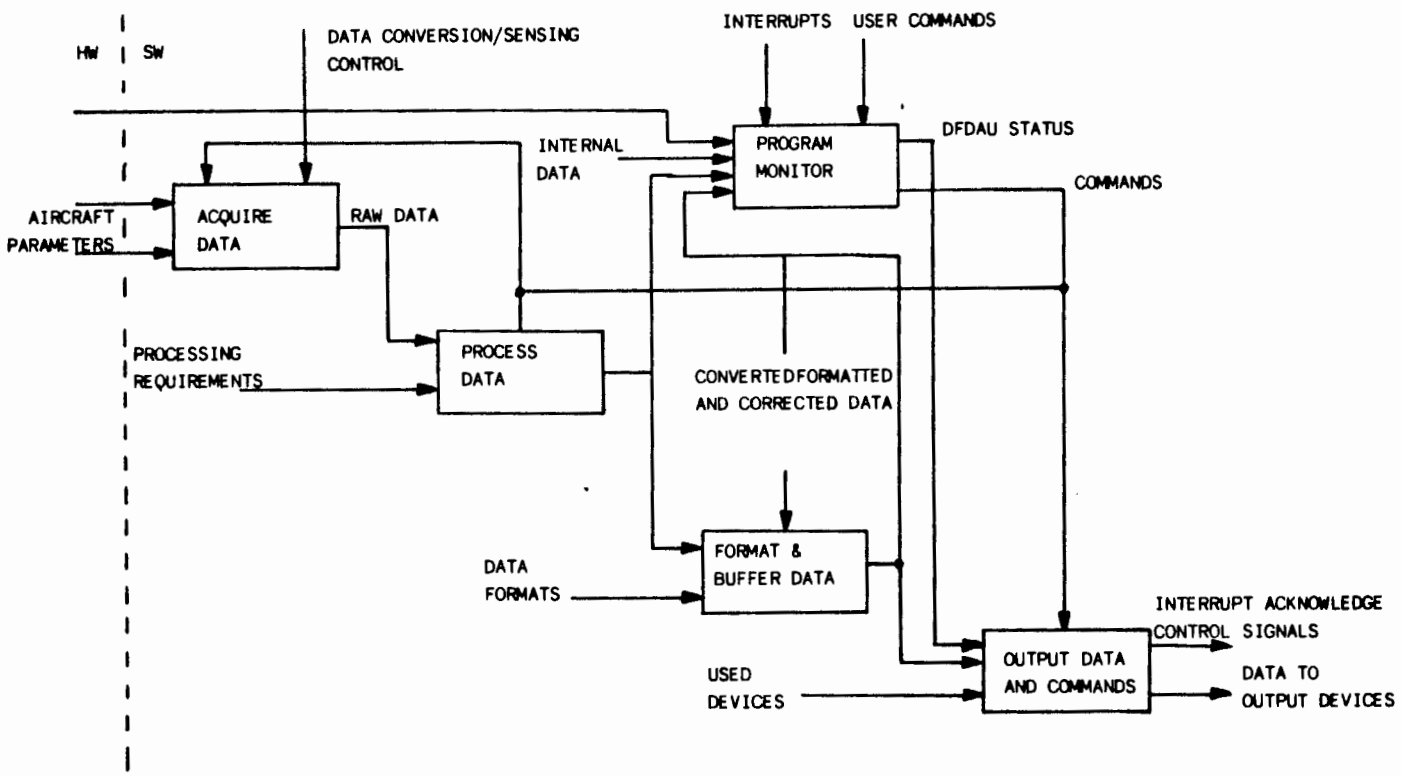


Figure 5.2 DFDAU General Functions

The commands for these blocks come from the "PROGRAM MONITOR" block which itself is driven by interrupts.

5.1.4 Data Flow

A general overview of the DFDAU data flow is shown in figure 5.3.

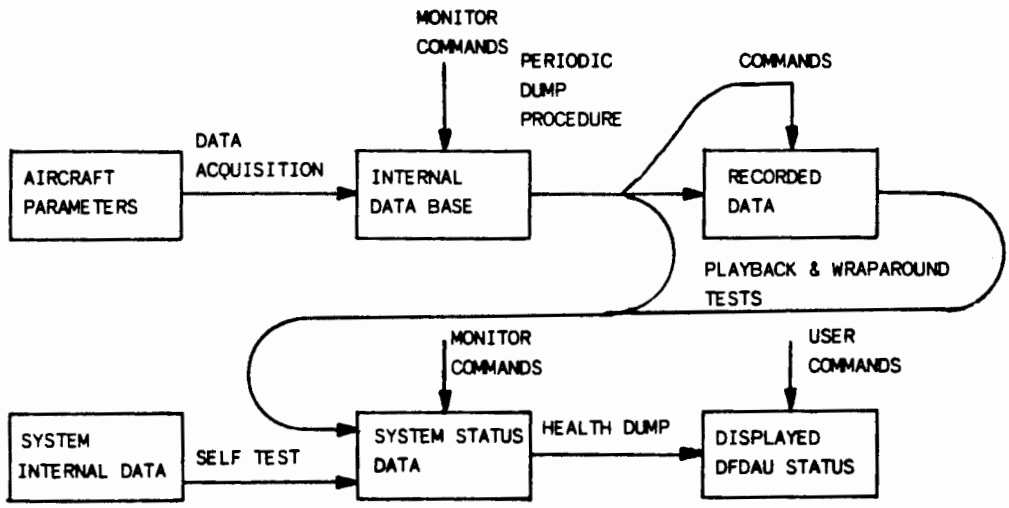


Figure 5.3 DFDAU Data Flow

The raw parameters received from the aircraft sensors are processed in accordance with the information in the parameter data base through the monitor command and then dumped periodically to the recorders which are controlled internally by interrupts.

The wraparound data returning from the output module is compared with the transmitted data, and error flags and status conditions are generated where necessary.

Through these and other microprocessor internal checks, a system status is developed, which in response to the user command will be displayed on the front panel as well as being recorded in non-volatile memory (EAROM).

5.1.5 Data Process

The data process cycle is shown in figure 5.4.

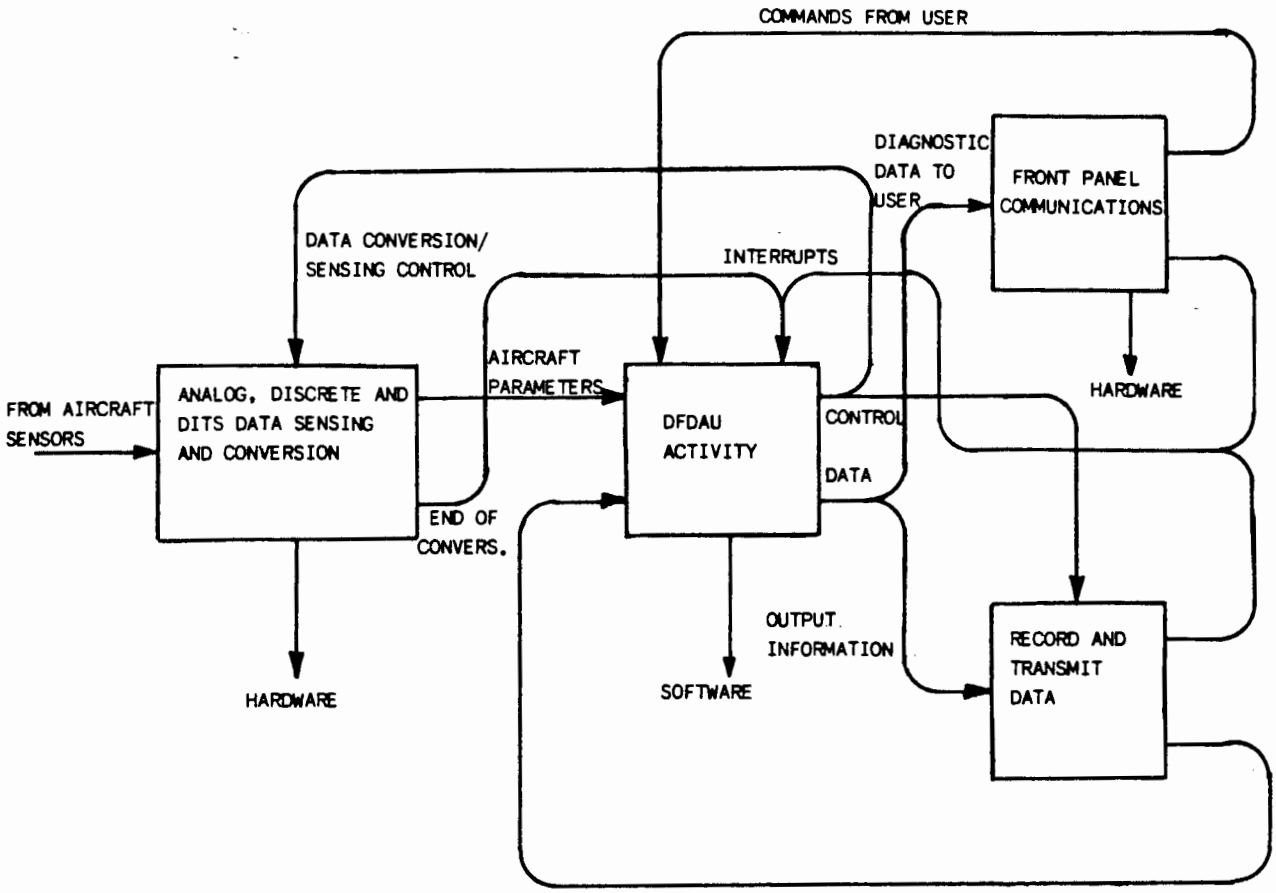


Figure 5.4 DFDAU Data Process Cycle

To receive an analog parameter from the aircraft sensors, the DFDAU sends the proper address to the multiplexer, waits for the ADC data-ready (end of conversion) interrupt which is asserted when data is ready, and then reads the data from the analog-to-digital converter output latch.

In the case of digital (DITS) parameters, where it is required that data be acquired simultaneously from all channels, the software sends the parameter id (section 4.4.2.2) to the DITS module, and reads the data from the output latch when the DITS data-ready interrupt is asserted.

The discrete parameters are read directly (no interrupt) from the multiplexer output latch after sending the appropriate channel address to the multiplexer.

The DFDAU outputs the information data word to the output ports on the Digital I/O Interface module. These ports pass the information to the recorders and other devices, and at the end of this transmission, prompt an interrupt back to the DFDAU, indicating the readiness of the port to receive a new data item. They also return the information as wraparound data back to the DFDAU for internal data checks.

The DFDAU continuously monitors the front-panel "READ" switch, and when depressed, displays the diagnostic data on the front panel display.

5.2 OPERATIONAL SOFTWARE

5.2.1 Target System Configuration

The target system for which the software system was designed is shown in its functional block diagram form below, in figure 5.5

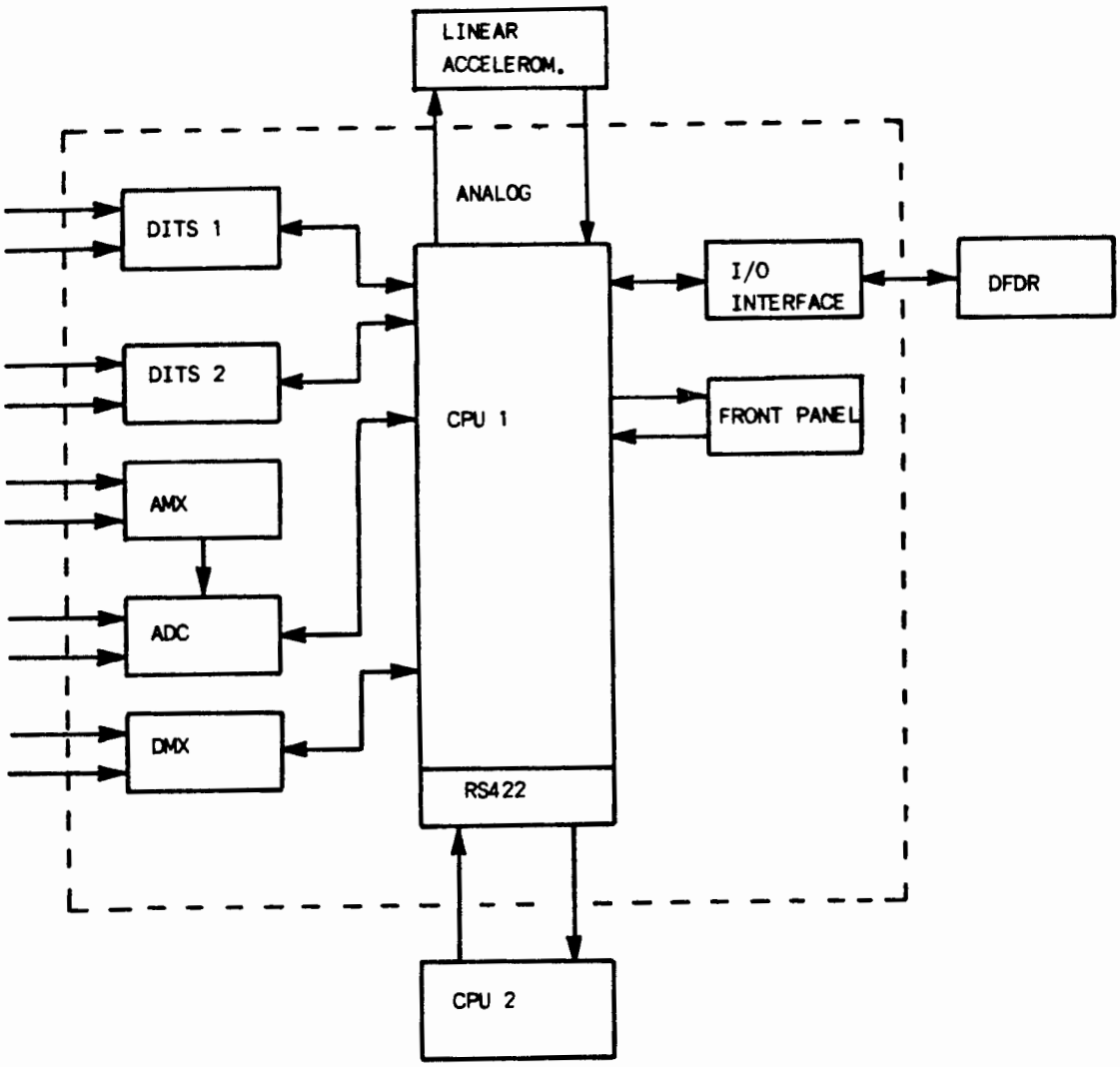


Figure 5.5 Target System Functional Block Diagram

The system consists of two CPU's that communicate via an RS422 link. CPU 1 performs the mandatory functions of the DFDAU as well as send all the parameters it acquires to CPU 2 for additional processing.

As discussed previously, this thesis deals primarily with the mandatory functions of the DFDAU, with some reference to the functions of CPU 2.

5.2.2 DFDAU Software Module Structure

To fulfill the requirements of modularity, the software was designed in such a way that it can be broken up into a number of functional elements (sections) as follows: The functional elements to be described are listed below:

- POWER-ON SECTION
- REAL-TIME SECTION
- EXECUTIVE
- ANALOG DATA ACQUISITION
- ANALOG CHANNEL CALIBRATION
- POWER SUPPLY BITE TEST
- DITS DATA ACQUISITION
- DISCRETE DATA ACQUISITION
- DFDR OUTPUT
- SYSTEM STATUS
- INTER-CPU COMMUNICATIONS

The breaking up of the software into independent functional elements makes the testing phase considerably easier as each functional element can be tested independently of the rest of the software prior to final system integration.

These sections, along with their component modules are shown in a top-down module chart in figure 5.6. These charts also show in a broad sense the control hierarchy of the system as the top (root) module in each section is called by an external interrupt and in turn, calls the various lower level modules. The executive section is initially entered via the power-on section, and subsequently by the real-time interrupt.

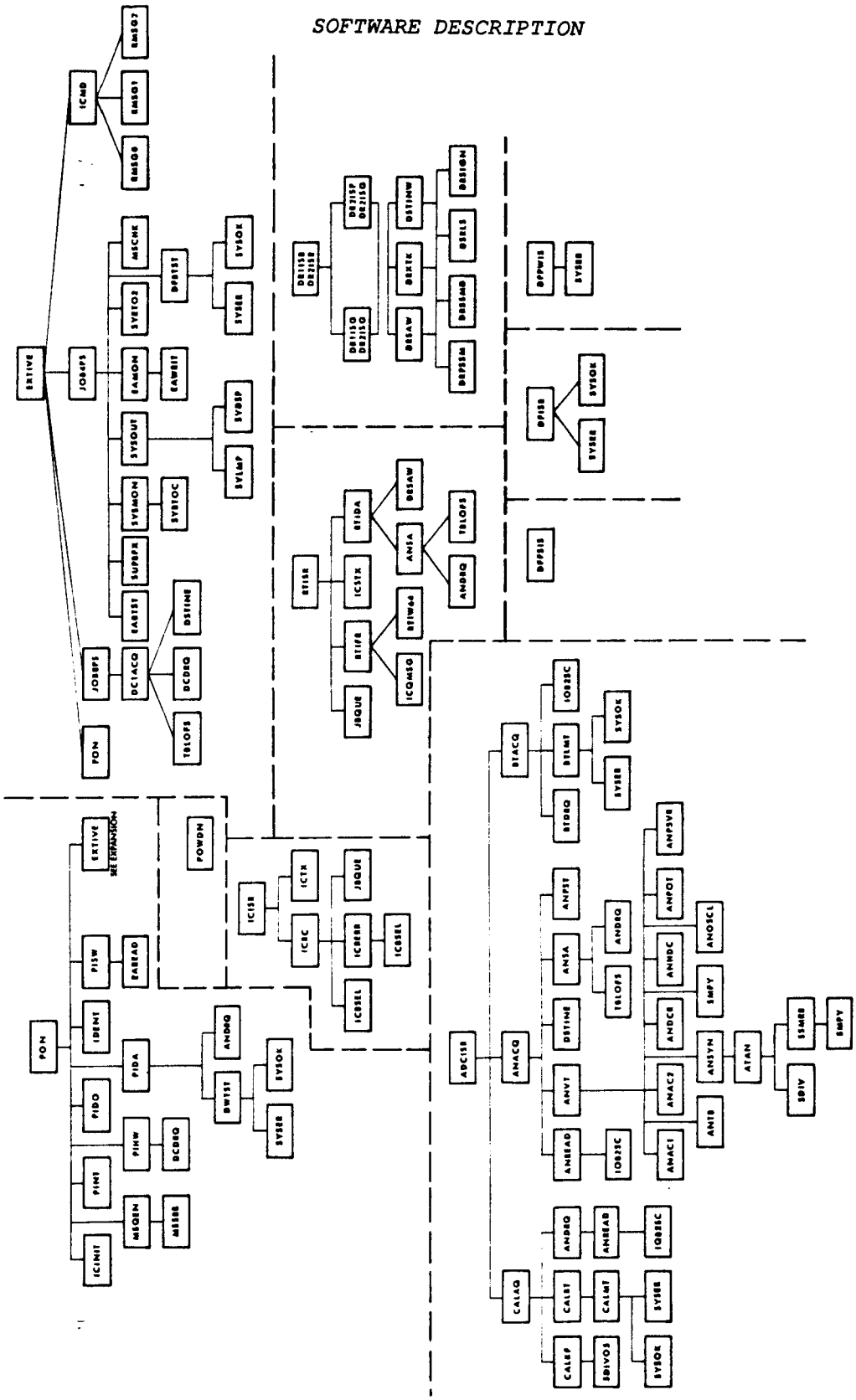


Figure 5.6 DFDAU Software Module Tree

The power-down section consists of only one module, POWDN, which is entered via the CPU power-down interrupt, which is generated by the CPU when it senses that its various supply voltages are dropping below their operating level. This module simply puts the microprocessor in a "hold" state, and execution is resumed via the PON routine once the supply voltages are at an operating level again. The memory hardware is designed in such a way that enough power is stored to hold the memory contents valid for a period of 200 milliseconds after a power-down interrupt (section 3.3). Thus, if power is resumed during this period, execution can resume as if un-interrupted.

5.2.3 Interrupt Levels And System Control Flow

The various tasks of the DFDAU Software are classified into two groups, namely the foreground tasks and the background tasks.

The foreground tasks are invoked by external hardware interrupts and perform the timekeeping and input-output functions. As previously mentioned the DFDAU hardware can accommodate 16 prioritized interrupts (0 - 15, 0 having highest priority) and 15 of them (1-15) are maskable under software control. Interrupt 0 is the power-on interrupt and is non-maskable for obvious reasons. The priority structure of the DFDAU interrupts is as follows:

Interrupt Level	Function
0	Power-On
1	Power-Down
2	Not Used
3	Real-Time
4	*Auxiliary Output
5	DFDR Output
6	ADC data ready
7	Inter-CPU
8	*Tachometer
9	Not Used
10	DITS 1 data ready
11	DITS 2 data ready
12	Not Used
13	Not Used
14	Not Used
15	*Printer

* - Not used in this configuration

The assignment of interrupt 0 to the power-on interrupt is fixed by the hardware and cannot be assigned to any other operation.

Interrupt 2 is reserved for special types of DFDR that issue a frame interrupt and a service routine is necessary to provide the interrupt reset. Interrupt 3 was assigned to the real-time interrupt. It has to have a higher priority than any I/O interrupt because it provides the basic timing functions. The DFDR output interrupt is a free-running interrupt that occurs every 15.625ms and therefore has to be serviced at regular intervals. It is therefore assigned a priority higher than the data acquisition interrupts.

In configurations where an auxiliary recording device used and has a subframe length of greater than 64 words, the free-running auxiliary output interrupt will occur at a higher rate than that of the DFDR interrupt. If it were assigned a priority lower than that of the DFDR interrupt, and if an auxiliary output interrupt occurred while the software was servicing the slower DFDR interrupt, that particular auxiliary output interrupt would not be serviced and therefore would result in a gap in the auxiliary output data. To prevent this therefore, the auxiliary output interrupt is assigned a higher priority than the DFDR interrupt.

For the data acquisition interrupts, the priority assignments were based on the reasoning that the longer it takes for an interrupt to be asserted after a data request is made, the higher the assigned priority. For example, it takes approximately 3ms for the analog data-ready interrupt to be asserted after an analog data request is made. Therefore, the software can make an analog data request and while waiting for the analog data-ready interrupt, a number of DITS data requests can be made and serviced. The DITS interrupt is asserted approximately 64 microseconds after a data request is made. In this way, an analog data-ready interrupt occurring while the software is servicing a higher priority data-ready interrupt is avoided.

The foreground jobs are therefore:

- Power-on Interrupt Service
- Power-down Interrupt Service
- Real-Time Interrupt Service
- Analog Data Acquisition
- DITS Data Acquisition
- DFDR Output
- Inter-CPU Communications

The background tasks are invoked by the executive, and perform the general "housekeeping", number crunching, data formatting and system status generation. These background jobs are queued and executed in a sequential fashion. The background tasks are the following:

- *System Status Code Generation*
- *DFDR BITE Test*
- *Front Panel Display*
- *Memory Sumcheck*
- *Write to EAROM*
- *Discrete Data Acquisition*

Thus, when the system is not servicing any interrupts, control is passed to the executive which then in turn executes the next background job in the job queue. Figure 5.7 below shows the general control flow of the DFDAU Software.

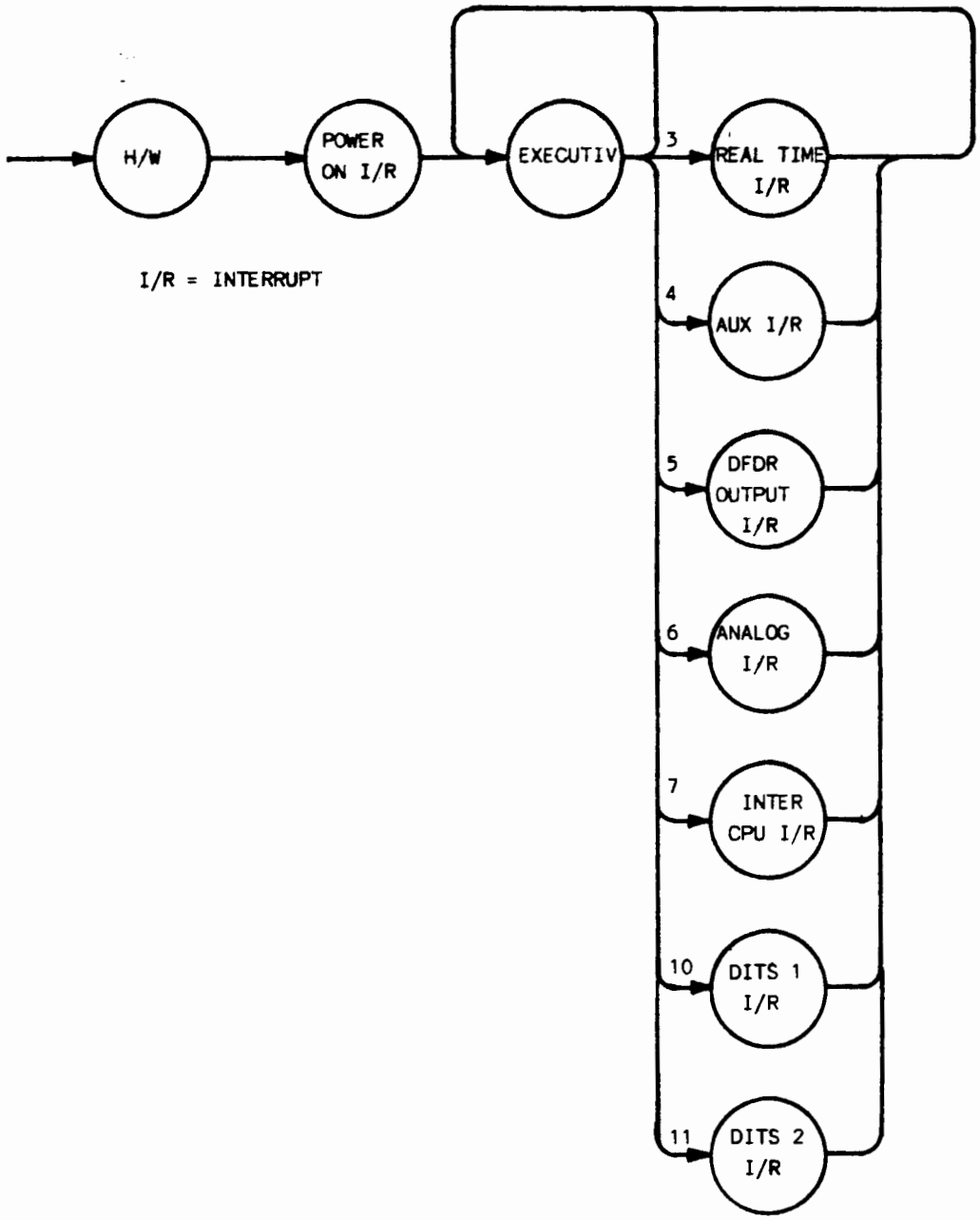


Figure 5.7 DFDAU Control Flow

5.2.4 Data Acquisition And Timing

The whole data acquisition and data output process runs on a

real-time basis provided by the real-time interrupt. The frequency of the real-time interrupt is determined by the required rate of data output. In this case, it is determined by data output rate to the DFDR which is 64 words per second (1 subframe), ie. a total of 256 words in a 4 second data frame. The software can then maintain a count of the occurrences of the real-time interrupt to determine the start of a new data subframe and to update the frame counter (section 3.3.2.1) for the DFDR superframe data. For a rate of 64 words per second, the real-time interrupt period turns out to be 15.625 milliseconds.

From the input parameter list in appendix C, it can be determined that the highest required output rate for a parameter is 8 times per subframe (ie. 8 times per second), as for Vertical Acceleration on page C-2. This means that the highest acquisition rate necessary for a parameter is once every 125 milliseconds. Using this result, the 1 second subframe period is broken up into 8 cycles of 125 milliseconds (a total of 32 cycles per data frame) as shown below:

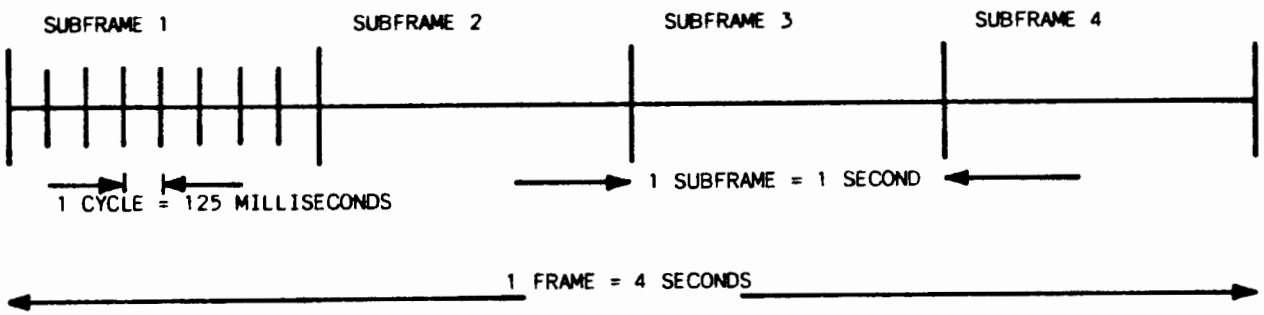


Figure 5.8 DFDAU Data Acquisition Timing

A maximum of 8 words can be output to the DFDR during this 125ms interval, and they all have to be acquired a maximum of 250ms prior to their output (see requirements section 3.3). This then implies that a parameter acquired at the beginning of a 125ms acquisition cycle, has to be output to the DFDR, at the latest, at the end of the next 125ms cycle, ie. a maximum of 250ms delay between data acquisition and data output. The constraint in this case is therefore that a minimum of 8 parameters have to be acquired during a 125ms period, and that the highest acquisition rate for a parameter is 8 times per second; both well within the

operating limits and requirements of the DFDAU.

A software counter, the cycles-per-frame counter (0-31) is maintained so that the software can keep track of the data acquisition and output process with respect to the 4-second time frame. This counter is also used as the initial index into the parameter data base to determine the number of parameters of a particular type that have to be acquired during that cycle (see later in the parameter acquisition sections).

To be able to transmit the most recent data present at the input ports, the parameters that are acquired during a particular 125ms data acquisition cycle, are transmitted to the output ports during the next 125ms cycle. Simultaneously with this transmission, the next data acquisition cycle starts and data for the next output cycle is acquired. Hence data is output at a maximum of 250ms after data acquisition. The real-time constraint here is of course, that the 125ms period has to be long enough so that all the parameters that have to be output in the corresponding DFDR word slots, can be acquired. See figure 5.9.

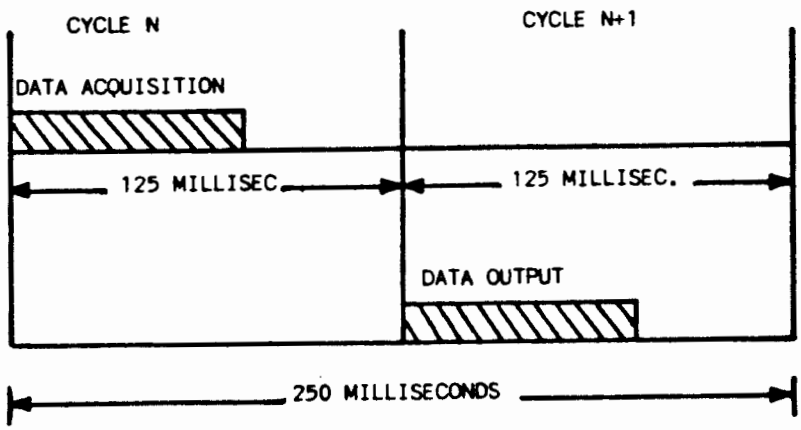
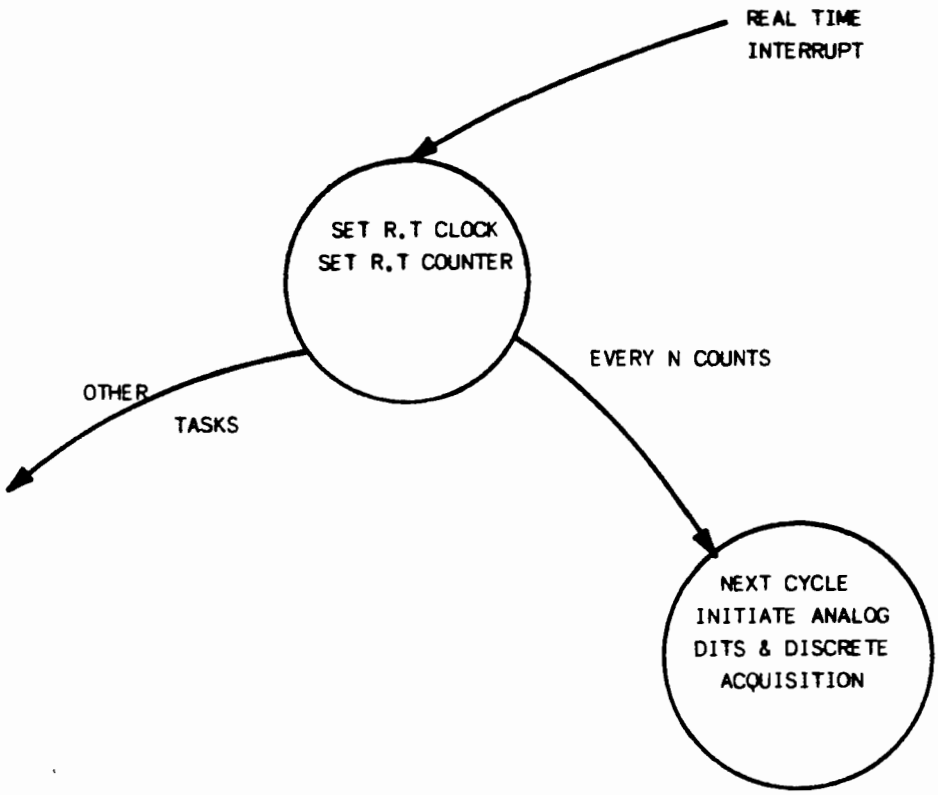


Figure 5.9 Data Acquisition and Output Cycle.

After power-on initialization, the data acquisitions and transmissions are initiated and the very first data acquisition cycle begins. A continuous (free-running) output to the DFDR and auxiliary output would follow from this moment, each time being invoked by the DFDR or the auxiliary output buffer-empty interrupt.

Subsequent real-time interrupts will initiate the analog, discrete and DITS data acquisition (see figure 5.10). From this moment on data acquisition is initiated at the beginning of each cycle and controlled by the real-time interrupt.



Note:

1. *Real-Time Interrupt is free running*
2. *Real-Time I/R rate is 15.625ms (64/sec)*
3. *Real-Time counter resets at each N(8) counts*
4. *Each cycle is 125 milliseconds*

Figure 5.10
Real-Time Interrupt Initiates Data Acquisition Cycle.

All interrupts for CPU 1 use the same clock signal from the CPU module and are therefore synchronized to each other.

Figure 5.11 shows the DFDAU data acquisition and data output phases on a time scale.

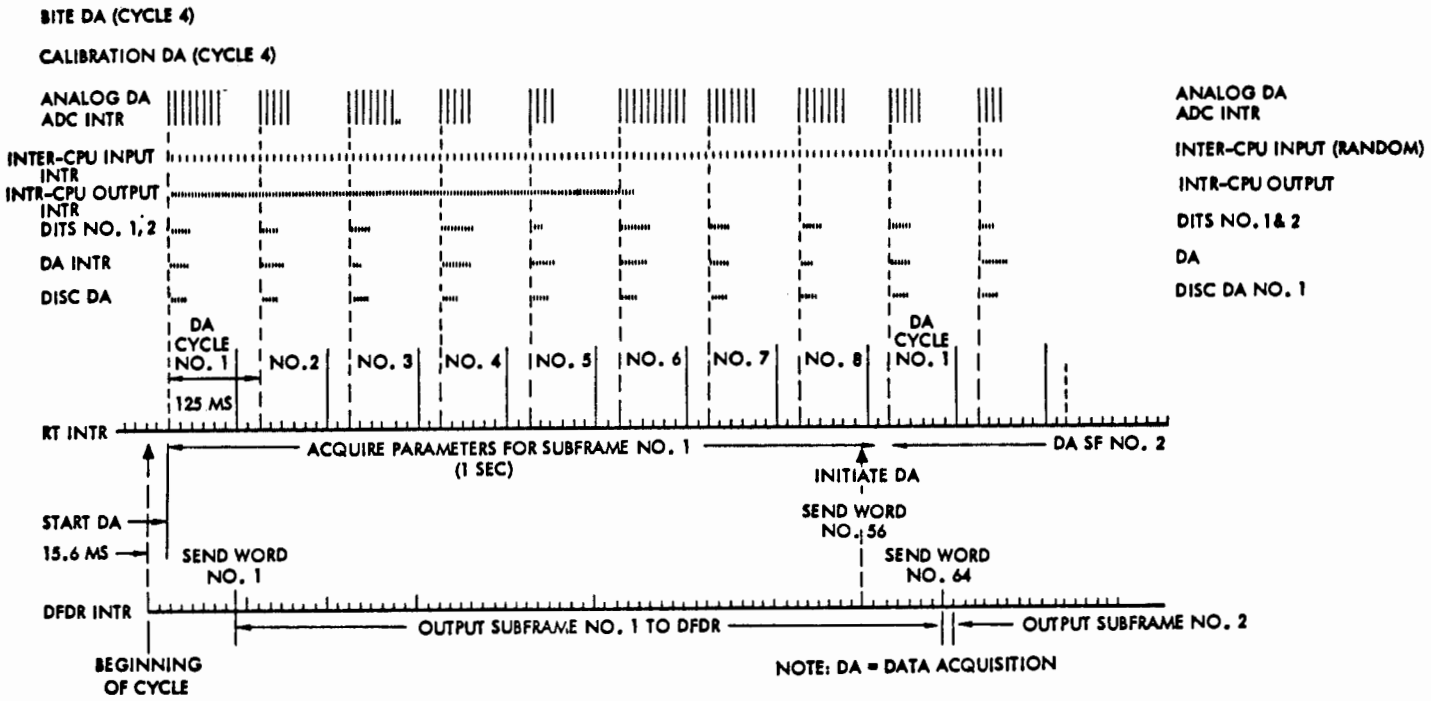


Figure 5.11 DFDAU Software Timing Diagram

5.2.5 Functional Element Description

As mentioned in section 5.2.2, the software is broken down into a number of sections, or functional elements. Each functional element is made up of a number of modules, and when linked together, can be tested as one complete self-contained unit. Some modules however are common to more than one functional element.

The following sections describe the operation of each of these functional elements with the aid of module trees, control and data flow diagrams and Program Design Language (PDL) descriptions.

The functional elements to be described are listed below:

- *POWER-ON SECTION*
- *REAL-TIME SECTION*
- *EXECUTIVE*
- *ANALOG DATA ACQUISITION*

- ANALOG CHANNEL CALIBRATION
- POWER SUPPLY BITE TEST
- DITS DATA ACQUISITION
- DISCRETE DATA ACQUISITION
- DFDR OUTPUT
- SYSTEM STATUS GENERATION
- INTER-CPU COMMUNICATIONS

5.2.5.1 Power-On Section -

Upon application of power to the DFDAU, and after all voltages have stabilized to their operational levels, the power-on interrupt (level 0) is asserted by the hardware and is serviced by the module PON. This module then performs the following functions, either within itself or by calling other lower level modules.

- Initialize hardware
- Determine aircraft and engine type
- Clear RAM
- Initialize software counters
- Initialize interrupts
- Enable Real-Time Interrupt
- Initiate data acquisition process
- Calibrate analog channels
- Initialize data output process
- Initialize Inter-CPU communications
- Perform DITS wraparound test
- Read contents of EAROM and copy into RAM image
- Generate program memory sumcheck

The first tasks that the module PON has to perform, is to decide whether the power interrupt received was a "cold start" power-on or a momentary loss of power. This is done by checking a fixed pattern (AAAA 1234 5678), which is written into three consecutive pre-defined RAM locations, when the system is initially turned on. The RAM has the capability of of retaining its contents for 200 milliseconds after power is cut-off. If the patterns read are not what was originally written there, it is assumed that the power interrupt was from a "cold start", and complete system and memory initialization (clearing) is initiated.

If the power interrupt was less than 200 milliseconds (RAM pattern good), the cycle-per-frame counter is restored so that the software can continue from where it was interrupted.

Once the power-on service process is completed, control is passed to the executive, and from there on the real-time interrupt takes control.

Figure 5.12 shows the hierarchical structure of the power-on section in the form of a module tree with the module PON as the root module.

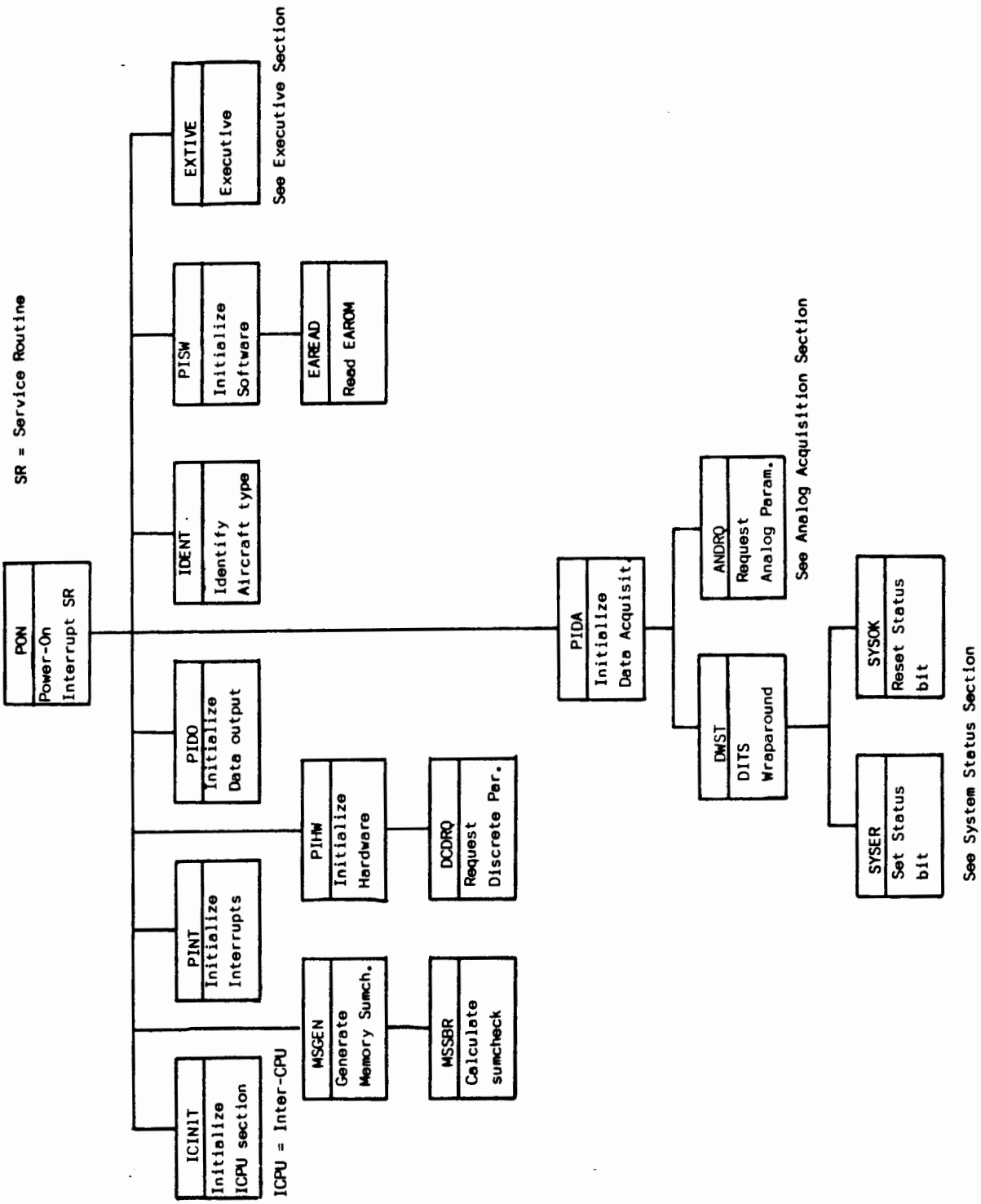


Figure 5.12 Power-On Module Tree

In PDL form the module PON is as follows:

```
PON PDL Description
-------------------
DISABLE ALL INTERRUPTS
SET LOOP COUNTER = 3
GET 3 POWER-ON TRANSIENT FIXED PATTERNS
GET 3 POWER-ON TRANSIENT RAM PATTERNS
DOUNTIL LOOP COUNTER = 0
  COMPARE FIXED PATTERN WITH RAM PATTERN
  IF NOT EQUAL - RAM IS BAD
        PLACE FIXED PATTERNS INTO RAM
        CLEAR ALL RAM WORDS
        RESET CYCLE PER FRAME COUNTER
        RESET FRAME-SUBFRAME COUNTER
  ELSE - RAM IS GOOD
        RESTORE CYCLES-PER-FRAME COUNTER
        RESTORE FRAME-SUBFRAME COUNTER TO START AT NEXT
             SUBFRAME
  ENDIF
ENDDO
CALL PIHW TO INITIALIZE HARDWARE
CALL PISW TO INITIALIZE SOFTWARE
CALL MSGEN TO GENERATE PROGRAM MEMORY SUMCHECK VALUES
CALL PINT TO INITIALIZE INTERRUPTS
CALL PIDA TO INITIALIZE DATA ACQUISITION
CALL PIDO TO INITIALIZE DATA OUTPUT
CALL IDENT TO TO GET ENGINE/AIRCRAFT IDENTIFICATION
CALL ICINIT TO INITIALIZE INTER-CPU HARDWARE AND SOFTWARE
CALL EXTIVE TO ENTER BACKGROUND EXECUTIVE
END
```

The module PIHW turns off any front panel indicator lamps that may be on by writing a single-bit zero to CRU address 5E0(Hex).

PISW clears all software counters, initializes the executive job queue and all buffer pointers, and then calls the module EAREAD to read the contents of the EAROM and copy them into the RAM image buffer (section 5.2.5.9).

The module MSGEN, along with MSSBR calculate the program memory (EPROM) sumchecks in 1K blocks (ie. 1 sumcheck word for every 1K of EPROM), store it, so that it can be used later by the executive for periodic comparisons with the hard-coded expected values to check software validity (section 3.3.3.2.1).

PINT resets the data acquisition interrupts (analog and DITS), initializes the interval counter for the real-time interrupt to 15.625 milliseconds, and then unmasks all the interrupts.

The data acquisition phase is started by the module PIDA. Prior to any data acquisition, it initiates a DITS wraparound test (section 3.3.3.2.2) by calling the module DWTST which in turn calls SYSER or SYSOK to either set or reset an error bit in the system status buffer (see later in System Status Generation section), depending on which DITS channel passed or failed the DITS wraparound test. After that, the module ANDRQ is called to initiate the analog interrupt for the calibration phase of the analog data acquisition. In this case, all analog channels are calibrated prior to any data acquisition so that calibration data is available for the initial analog data acquisition phase (channel calibration is also periodically performed during actual operation). The results of the calibration are compared to pre-defined upper and lower limits listed in the analog look-up tables, and the appropriate status bits are set or reset in the system status buffer.

When the calibration phase is completed, the data acquisition table pointers for DITS, discrete and analog data and the inter-CPU and DFDR output buffers are initialized.

The analog interrupt is then again initialized along with the DITS interrupt, and at this stage the system is ready to begin data acquisition.

The module IDENT reads the engine and aircraft type Ident discretes to determine the aircraft and engine types, and setting a variable, SSEG accordingly. This variable is used as an offset into the data acquisition and destination tables in the parameter data bases in the cases where different parameters have to be acquired for for various configurations of engines and aircraft (table 3.1).

PIDO enables all output interrupts, ie. DFDR and auxiliary (if needed), and then sets the DFDR output word counter to 55, which is the beginning of the last data acquisition cycle of subframe 4. This is done to synchronize the start of data acquisition with the DFDR output. As explained in section 5.2.4, data acquisition leads data output by 125ms. So during the time when the last 8 words of subframe 4 are being output, the first 8 words of subframe 1 are being acquired. This however means that the first 8 words that are output to the DFDR after power-on are meaningless, but after that, valid data output starts with word 1 of subframe 1.

The inter-CPU communications functions are initialized by the module ICINIT. This task consists of initializing the receive and transmit data pointers, and then setting the communications uart (TMS9902) to the required conditions ie. 9600 baud for transmit (CPU 1 to CPU 2), 1200 baud for receive (CPU 2 to CPU 1) 1 stop bit 8 data bits and odd parity.

Control is then passed (forever) to the executive section via module EXTIVE so that data acquisition and background job execution can begin. This is depicted in the DFDAU control flow diagram in figure 5.7.

Figure 5.13 below shows the data flow for the power-on section.

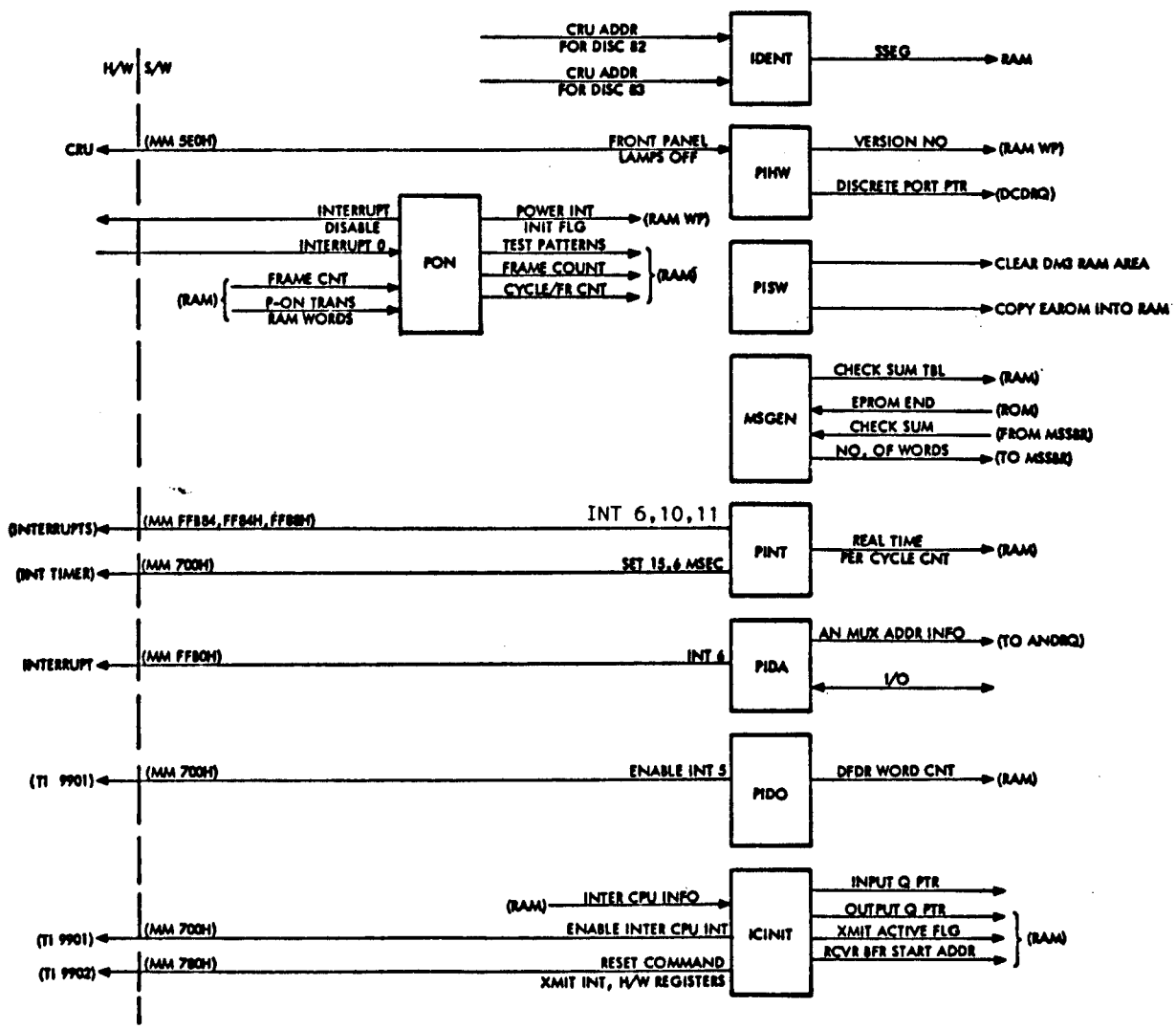


Figure 5.13 Power-On Section Data Flow

5.2.5.2 Real-Time Section -

The real-time interrupt, as explained in section 5.2.4, is set to occur every 15.625 milliseconds. This is used as the basis of all time keeping functions within the DFDAU software. The priority for this interrupt is level 3 and is serviced by the module RTISR, which in turn calls a number of other routines to perform the various tasks indicated below:

- *Service real-time interrupt and queue background jobs and inter-CPU messages.*
- *Initialize frame counters and update timers.*
- *Initialize frame counters and update timers.*
- *Initialize frame counters and update timers.*
- *Checks to see if inter-CPU is operational and sets an error bit if not.*
- *Initiates a data transmission to CPU 2.*
- *Outputs data to superframe buffers.*
- *Initiates the data acquisition phase for the next data acquisition cycle.*
- *Outputs superframe data to the DFDR superframe word (word 64).*
- *Calculates the acquisition and destination tables offsets for the next parameter to be acquired.*

The hierarchical structure of the real-time section is shown in the module tree in figure 5.14.

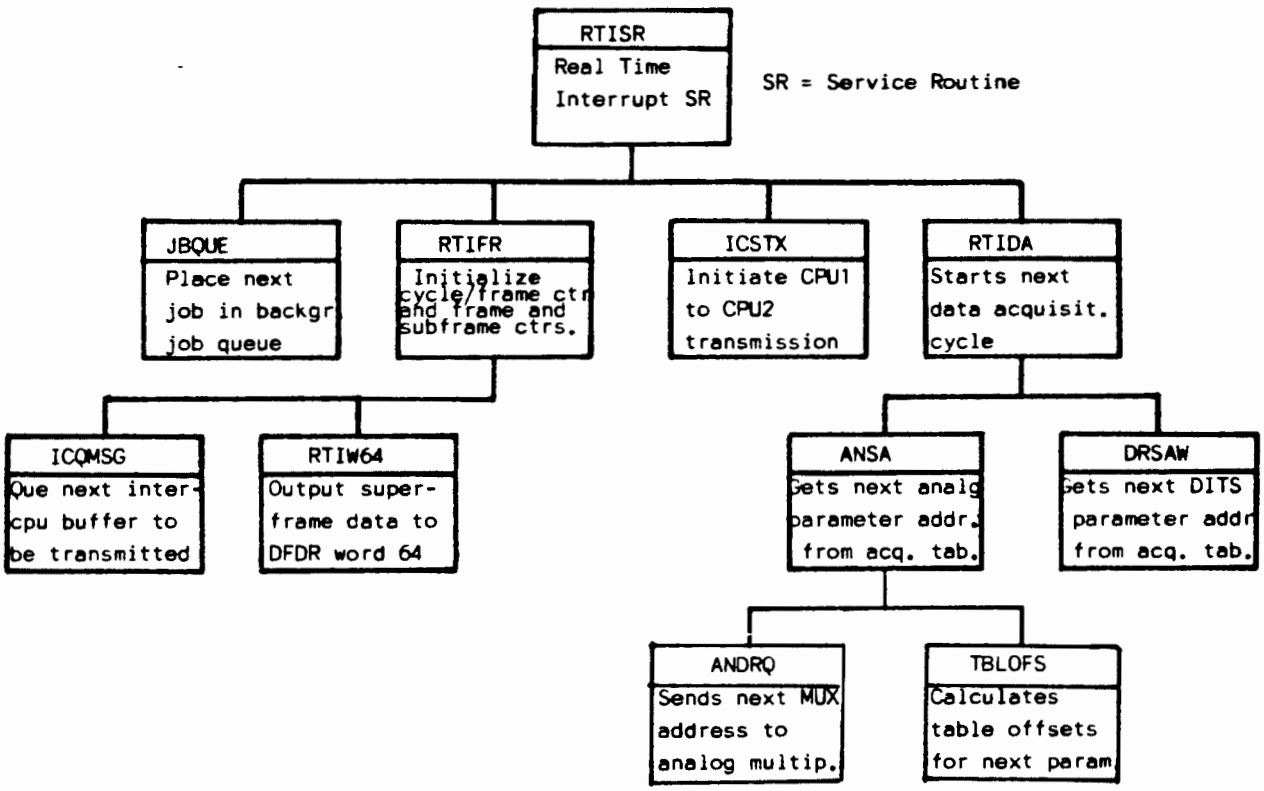


Figure 5.14 Real-Time Section Module Tree.

*Upon entry into RTISR, the interrupt is reset, and the clock counter is incremented. Then, the counts-per-cycle counter (RTPCYC) is decremented by one. This counter has a maximum count of 8, that is 125msec (15.625*8) for each cycle. There are 8 of these cycles per subframe, that is 1000msec (8*125) per subframe. Therefore, whenever the counter RTPCYC reaches 0, the following will happen:*

- *Module RTIFR is called to check for system synchronization, superframe data output, and preparation and queuing of the data buffer to be transmitted to CPU 2.*

- *The system is synchronized when the DFDR output word pointer = 55, which is the last data item acquired during the previous acquisition cycle. As mentioned previously, the output lags the acquisition (input) by 125msec, which in turn equals 8 counts of the sub-cycle counter RTPCYC.*

This is shown in figure 5.15 below:

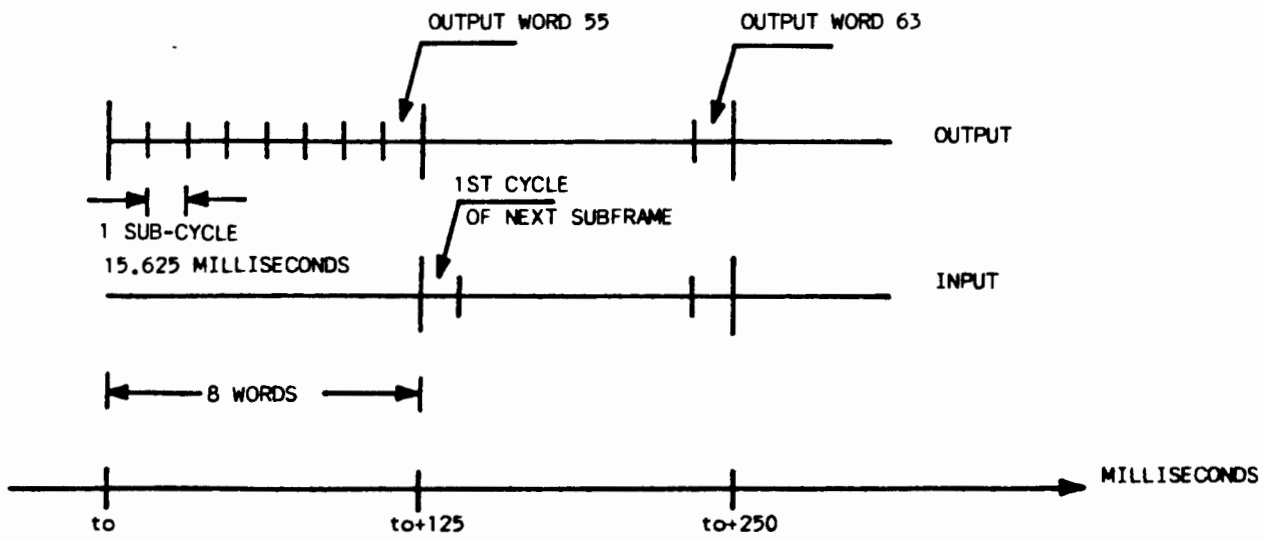


Figure 5.15 Time Relationship Between Input and Output

- *Module RTIDA is called to initiate the next data acquisition cycle.*
- *Module JBQUE is called to queue the next background job to be executed by the executive.*
- *Finally, module ICSTX is called to initiate a CPU 1 to CPU 2 data transmission.*

In PDL form, RTISR is as follows:

```
RTISR PDL Description
---------------------
RESET REAL-TIME INTERRUPT
INCREMENT LSW OF CLOCK COUNTER BY 1 (1 COUNT = 15.625 MSEC)
IF LSW OF CLOCK COUNTER = 0 (OVERFLOW)
        INCREMENT MSW OF CLOCK COUNTER BY 1
ENDIF
DECREMENT REAL-TIME SUB-CYCLE-PER-CYCLE COUNTER (RTPCYC)
IF RTPCYC = 0
        SET RTPCYC = 8
        CALL RTIFR TO CHECK INITIALIZATION
        IF SYNCHRONIZED WITH DFDR WORD 55
           CALL RTIDA TO INITIALIZE NEXT DATA ACQUISITION CYCLE
           CALL JBQUE TO QUEUE BACKGROUND JOBS
        ENDIF
        IF 2ND, 4TH, 6TH AND 8TH CYCLE
           CALL JBQUE TO QUEU JB4PS (4 TIMES PER SECOND TASKS)
        ENDIF
ENDIF
CALL ICSTX TO START INTER-CPU TRANSMISSION
END
```

*The module RTIFR is called when a new acquisition cycle begins. In this module, the cycles-per-frame counter (CYPFRC) is incremented. This counter has a starting value of 0 and a maximum value of 31 for the 32 acquisition cycles per frame of data (8 per subframe). Thus, the total time for this counter is 4 seconds (32*125 milliseconds) and is used, amongst other things for determining acquisition and destination table offsets.*

By checking the 3 least significant bits of this counter, RTIFR determines whether a new subframe is to be started. If this is not the case, the routine sets a flag, exits and data acquisition continues.

In the case of a new subframe, ie. the 3 least significant bits of CYPFRC are all zero, RTIFR checks for input-output synchronization by comparing the DFDR output word counter, DFWRC with 55. If DFWRC is less than 55, it means that synchronization has been lost. In this case, a flag is set and control returns to the calling routine, which in turn terminates without initiating a new data acquisition cycle. Synchronization is re-established in the DFDR output routine (see section 5.2.5.8).

However, if the system operation is synchronized, RTIFR continues on to perform the following functions:

- *Output superframe data by calling RTIW64.*

- *Prepare the inter-CPU transmission buffers for output to CPU 2, and then queue the buffer by calling ICQMSG.*

If the inter-CPU message queue is full, an error counter, ICOER is incremented and when this counter reaches a value of 3 (three consecutive failures in attempting to queue an inter-CPU message), an error bit is set in the system status buffer.

- Lastly, in the case of the start of a new data frame, the cycles-per-frame counter, CYPFRC is zeroed.

The frame-subframe counter FRCNT is a 14-bit counter where the last 2 bits hold the subframe count (0 to 3) in the current frame, and the upper 12 bits form the actual frame counter which counts from 0 to 4095, and then restarts.

This frame counter is displayed as superframe data in word 64 of subframe 1, and is used as a time reference when the DFDR tape is played back.

A PDL description of RTIFR is given below:

```
RTIFR PDL Description
---------------------
INCREMENT CYCLES-PER-FRAME COUNTER (CYPFRC)
IF TIME FOR NEW SUBFRAME
        IF DFDR WORD COUNT > = 55
          INCREMENT FRAME-SUBFRAME COUNTER (FRCNT)
          CALL RTIW64 TO OUTPUT DATA TO SUPERFRAME
          TRANSFER 64 WORDS OF DFDR DATA TO INTER-CPU BUFFER
          TRANSFER SYSTEM STATUS DATA TO INTER-CPU BUFFER
          CALL ICQMSG TO QUEUE INTER-CPU MESSAGE
          IF QUEU IS FULL
                INCREMENT ERROR COUNTER (ICOER)
                IF ICOER > 3
                        SET ERROR BIT IN SYSTEM STATUS BUFFER
                ENDIF
          ELSE
                RESET ERROR BIT IN SYSTEM STATUS BUFFER
                SET ICOER = 0
          ENDIF
          GET START ADDRESS OF NEW INTER-CPU BUFFER
          PLACE NEXT DFDR SYNC CODE IN DFDR OUTPUT BUFFER
          IF NEW FRAME
                RESET CYCLES-PER-FRAME COUNTER
          ENDIF
                SET SYNCHRONIZATION FLAG TO SYNCHRONIZED
        ELSE
                SET SYNCHRONIZATION FLAG TO NOT-SYNCHRONIZED
        ENDIF
ENDIF
RETURN
END
```

The module RTIDA is called by RTISR if, upon return of RTIFR to RTISR, the flag indicating that the system output is synchronized is true, so that the next data acquisition cycle can be initiated.

The first task of RTIDA is to clear the next 8 words of the DFDR buffer where the next 8 words that will be acquired, will be stored.

After this, acquisition and destination table pointers for DITS and analog are set up using the cycles-per-frame counter, CYPFRC. This counter is used to extract the "parameter quantity" for that cycle, for the DITS, analog and discrete parameters, from the data acquisition look-up tables. This variable specifies the number of parameters of each type that have to be acquired during that cycle, and is used to set up loop counter limits in the acquisition routines. CYPFRC is also used to extract the offsets into the multiplexer address tables for the analog parameters, and into the parameter number (parameter id) tables for the DITS parameters.

Acquisition is then started by calling routines ANSA, to send the analog parameter address to the analog multiplexer, and DRSAW to send the parameter id to the DITS on-board RAM address decoder (see section 4.4.2.2).

A PDL description of RTIDA is given below:

```
RTIDA PDL Description
---------------------
SAVE LINK REGISTER
GET NO. OF WORDS TO BE CLEARED
GET CYCLE PER FRAME COUNTER
EXTRACT CYCLES PER SUBFRAME COUNT ONLY
DETERMINE INDEX OF 1ST OUTPUT WORD OF NEW CYCLE
IF NOT 1ST CYCLE OF NEW S/F
        BUMP INDEX TO SKIP OVER SYNC CODE WORD
ENDIF
GET START ADDRESS OF DFDR OUTPUT
DOUNTIL ALL 8 WORDS CLEARED
        CLEAR DESTINATION WORD
ENDDO
GET ANALOG PARAMETER QUANTITY FROM ANALOG LOOK-UP TABLES
GET INDEX FOR ANALOG OFFSET TABLES
GET DITS 1 PARAMETER QUANTITY FROM ANALOG LOOK-UP TABLES
GET INDEX FOR DITS 1 OFFSET TABLES
GET DITS 2 PARAMETER QUANTITY FROM ANALOG LOOK-UP TABLES
GET INDEX FOR DITS 2 OFFSET TABLES
RESET ANALOG INTERRUPT
CALL ANSA TO SEND NEXT ANALOG ADDRESS AND START NEXT
                                          ADC INTERRUPT
CALL DRSAW TO START DITS 1 DATA ACQUISITION
CALL DRSAW TO START DITS 2 DATA ACQUISITION
RESTORE LINK REGISTER
RETURN
END
```

The module RTIW64 is called by RTIFR to output the superframe data to word 64 of the DFDR. This data is as follows:

subframe 1 word 64	-	frame counter
subframe 2 word 64	-	system status
subframe 3 word 64	-	documentary data
subframe 4 word 64	-	calibration data

The frame counter increments once every 4 seconds at the end of each data frame (see requirements section 3.3.2.1).

The system status, documentary and calibration data are stored in 3 buffers each 16 words deep, SYSTWB, SYDOCB, and SYCALB respectively.

Every time this module is entered, the two least significant bits of FRCNT (frame-subframe counter), are used to determine the current subframe and hence the appropriate buffer for that subframe.

The pointer into the selected buffer is derived from bits 2 to 5 of FRCNT which determine which frame of the current superframe is presently being output. These 4 bits count repetitively from 0000(bin) to 1111(bin) ie. 0 to 15. Therefore the contents of the superframe buffers are output sequentially from word 0 to word 15 and then repeated for the next superframe (see requirements section 3.3.2.2). The frame counter is determined by disregarding the 2 LSB's of FRCNT, and is also written into a location in the inter-CPU buffer for transmission to CPU 2. The fact that there are 4 subframes to a frame, and 16 frames to a superframe, makes it convenient to use a single 16-bit word (FRCNT) to determine the current state of the DFDAU as follows:

FRCNT Bits	DFDAU State
0 - 1	Current Subframe
2 - 5	Frame Number for Current Superframe
6 - 13	Current Superframe Number
2 - 13	Frame Counter Value (0 - 4095)

The module ICQMSG is called to queue the next buffer to be transmitted to CPU 2. This will be explained in the inter-CPU section later.

Figure 5.16 shows the data flow for the real-time section.

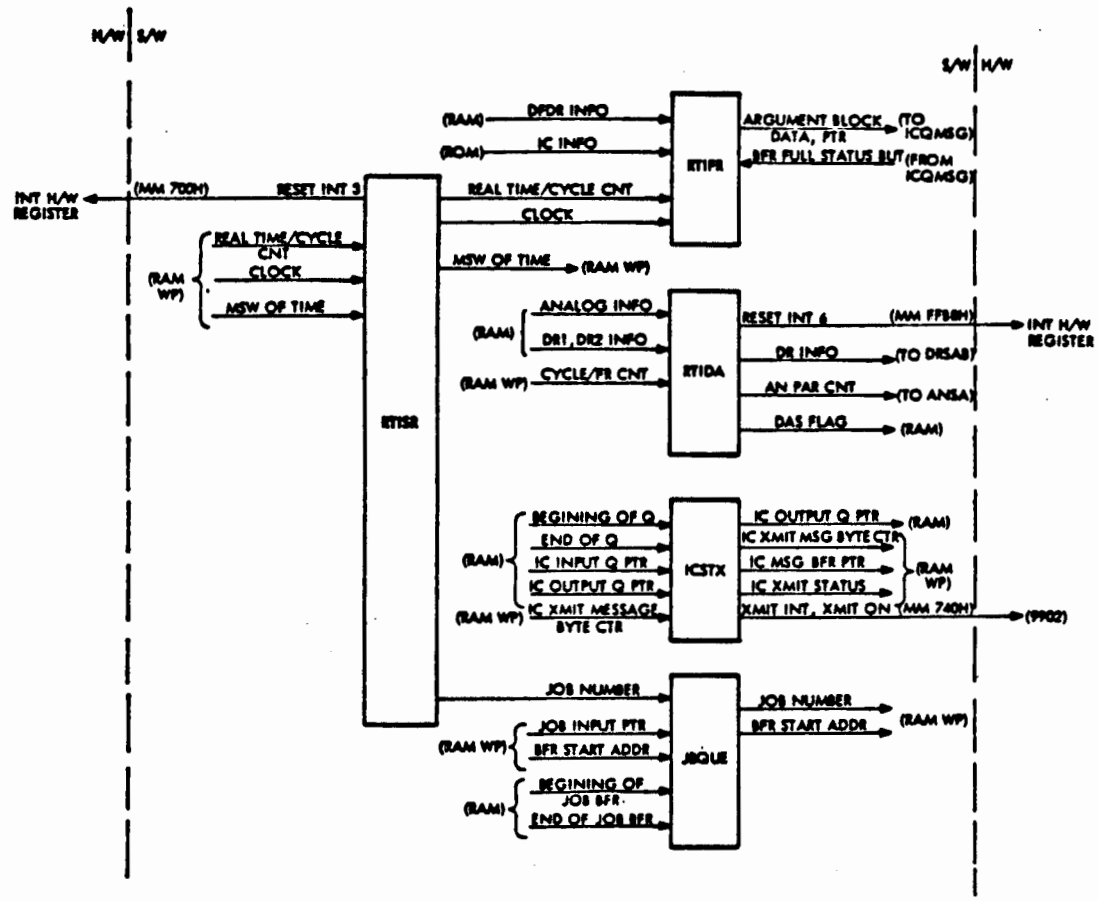


Figure 5.16 Real-Time Section Data Flow

5.2.5.3 Executive Section -

The executive section is initially entered via the module PON after all power-on initialization tasks have been completed. Thereafter, control remains in the executive (background) except when an interrupt occurs, in which case control is temporarily transferred to the interrupt service routine.

The module tree of the executive section is shown in figure 5.17 with the module EXTIVE as the root module.

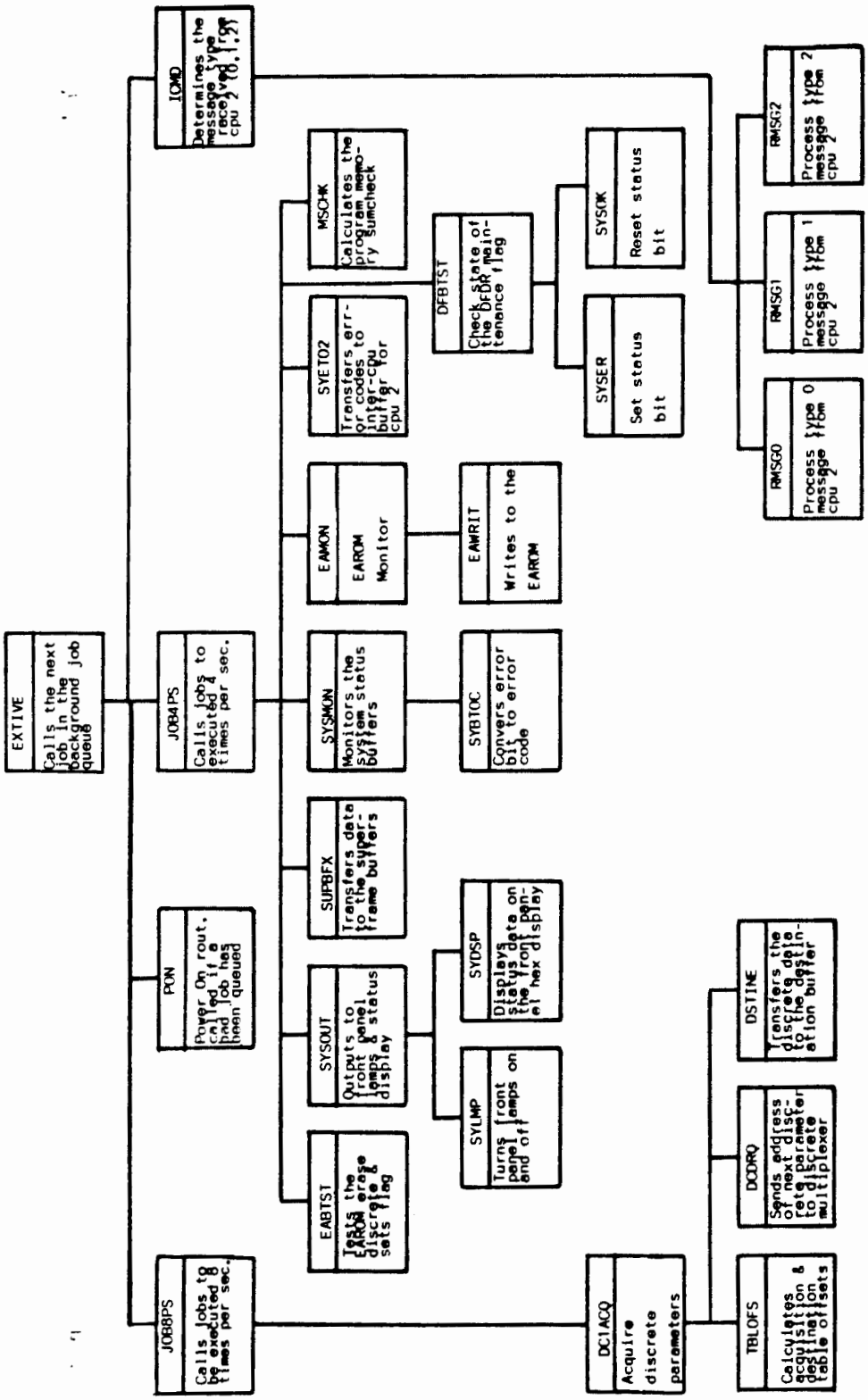


Figure 5.17 Executive Section Module Tree

The EXTIVE module is initially entered into from the power-on routine after all power-on initialization has taken place. Its task is simply to check the background job-queue for any tasks that may have been placed there, and call the appropriate routines to execute these jobs. If the job-queue is empty, it loops and waits until an entry has been made or an interrupt occurs.

The jobs to be placed in the job queue are as follows:

JOB8PS	- Calls routines to be executed 8 times per second.
JOB4PS	- Calls routines to be executed once, two, three or four times per second.
ICMD	- Called whenever a message from CPU 2 is received.

JOB8PS and JOB4PS are placed in the job-queue by RTISR at the required rate (8 or 4 times per second).

A PDL description of EXTIVE is as follows:

```
EXTIVE PDL Description
----------------------
DO FOREVER
        DO WHILE JBOPTR = JBIPTR (IS JOB WAITING?)
        ENDDO
        IF JBOPTR = JBEBUF (END OF JOB BUFFER)
                SET JBOPTR = START ADDRESS OF JOB BUFFER
        ENDIF
        GET JOB INDEX (I.E., JOB TYPE)
        GET BUFFER POINTER FOR THE JOB TO BE PROCESSED
        IF JOB INDEX < OR = MAX JOB NUMBER
                GET JOB FROM JOB TABLE
                EXECUTE CHOSEN JOB
        ELSE
                GO TO POWER-ON VIA VECTOR
        ENDIF
ENDDO
```

JOB8PS is called 8 times per second and in turn it calls any task that has to be executed at that rate.

In this particular case, the only such task is the discrete data acquisition module DC1ACQ.

JOB4PS is called four times per second (every 250ms) and its first task is to hit the reset integrator (watchdog timer) by toggling bit 17 of CRU address 0700(Hex). It then increments the call cycle counter (JB4CYL) which has a range of 0 to 3 and counts the number of times JOB4PS has been called during the

current second. Then, depending on the value of JB4CYL, it calls the following modules:

SUPBFX - Transfer data to superframe buffers (4ps).
EAMON - EAROM Monitor (4ps).
SYSMON - System Status Monitor (1ps).
SYSOUT - Front panel display (1ps).
DFBTST - Test DFDR Maintenance flag and sets or resets status bit.
EABTST - Tests EAROM erase discrete and sets clear flag if necessary.
MSCHK - Calculate program memory sumcheck an compare with the hard-coded values.

A PDL description of JOB4PS is given below:

```
JOB4PS PDL Description
----------------------
HIT RESET INTEGRATOR (EVERY 250 MSEC)
INCREMENT CALL CYCLE COUNTER (0 TO 3)
CALL SUPBFX - SET UP SUPERFRAME BUFFERS
CALL EAMON - EAROM MONITOR
IF CALL CYCLE COUNTER = 1 (2ND CYCLE)
        CALL SYSMON - SYSTEM MONITOR TO CHECK IF A NEW
        ERROR HAS BEEN GENERATED OR DELETED
        CALL SYSOUT - SYSTEM CAUTION/ERROR OUTPUT
        CALL DFBTST - DFDR BITE TEST
ENDIF
IF CALL CYCLE COUNTER = 2 (3RD CYCLE)
        CALL EABTST - EAROM ERASE DISCRETE TEST
ENDIF
IF CALL CYCLE COUNTER = 3 (4TH CYCLE)
        CALL MSCHK - MEMORY SUM CHECK
ENDIF
RETURN
```

The module SUPBFX collects all the data that is to be output as the superframe data and puts it in the appropriate buffers in preparation for output to the superframe (see requirements section 3.3.2.2). It also inserts a frame counter (0 to 15) in the upper 4 bits of system documentary data buffer as follows:

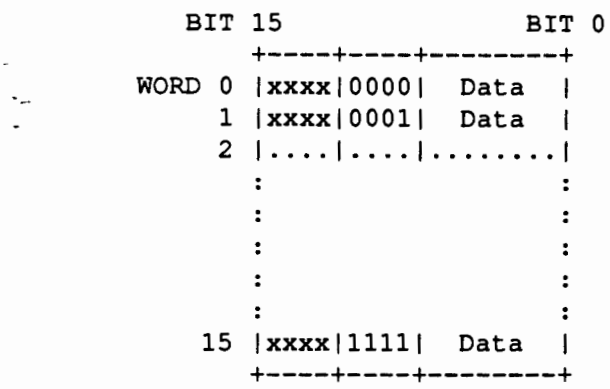


Figure 5.18 Cycle Counter in System Documentary Data Buffer.

This counter is used to determine the frame number of the current superframe when the DFDR data is played back.

The modules SYSMON and SYSOUT are part of the system status section and will be explained later.

The module DFBTST tests the DFDR maintenance discrete to check if the DFDR is operating correctly and sets or resets an error bit in the system status buffer (see requirements section 3.3.3.1)

EABTST tests the EAROM erase discrete and sets a flag, CLRFLG, if the discrete has been set. This flag is checked by the module EAMON (see later) to see whether an EAROM clear request has been made.

The routine MSCHK calculates a sumcheck on the program memory and compares it to the pre-programmed values to verify that the EPROM data has not been corrupted. If the calculated and pre-programmed values do not match, a bit is set in the system status buffer to flag the error condition.

5.2.5.4 Analog Data Acquisition Section -

The analog section is entered via the analog-data-ready interrupt and performs the data acquisition, calibration, power supply BITE tests and data formatting of all analog parameters.

It consists of 3 sections, namely acquisition, calibration and power supply BITE test.

A module tree of the analog section is shown below in figure 5.19.

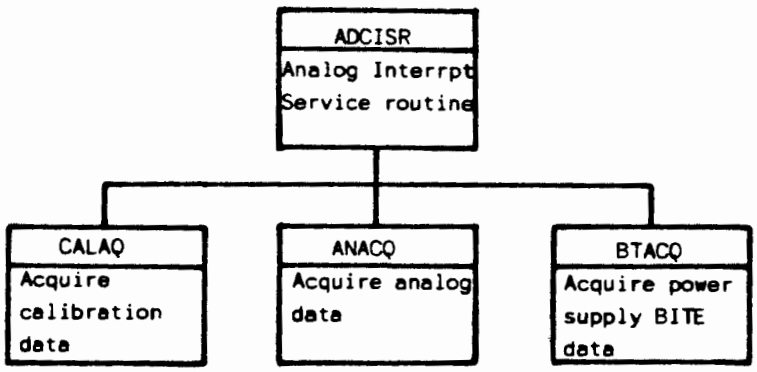


Figure 5.19 Analog Section Module Tree.

When an analog-data-ready interrupt occurs, it is serviced by the analog interrupt service routine ADCISR. The same interrupt is used whether the analog data item is an analog parameter, calibration data or power supply data. A flag (DASFLG = 0, 1 or 2) is therefore used to determine which one of these three analog signals is being acquired. Then, depending on the value of DASFLG, the appropriate section, either acquisition, calibration of power supply BITE section is entered. In all three cases, analog data has to be acquired via the analog-to-digital converter (ADC), the only difference being the source of that data. In the case of data acquisition, the data source is the external sensors to the various parts of the aircraft (see Input Parameter List, appendix C). In the case of calibration, the data source is the various voltage references within the DFDAU unit on the ADC module and in the case of power supply BITE, the data source is the various power supply voltages.

All these 3 source go to one ADC module via a multiplexer as shown below in figure 5.20.

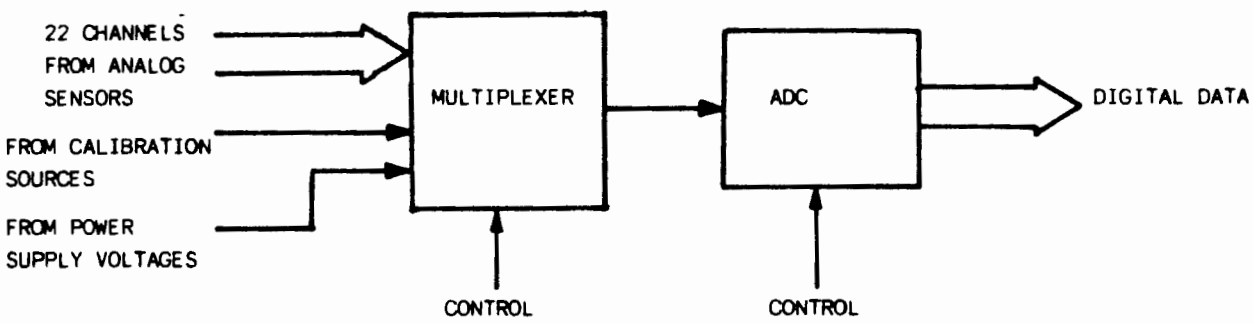


Figure 5.20 Analog Section Hardware Arrangement.

A PDL description of ADCISR is given below:

```
ADCISR PDL Description
----------------------
RESET ADC INTERRUPT (INT6)
IF DAS-FLAG > OR = 0, AND < 3.
        DO CASE DAS-FLAG (-1 <DAS-FLAG <3)
                =0: CALL ANACQ TO ACQUIRE ANALOG DATA
                =1: CALL CALAQ TO ACQUIRE CALIBRATION DATA
                =2: CALL BTACQ TO ACQUIRE BITE DATA
        ENDDO
ENDIF
RETURN
```

The data flow diagram for the analog section is given in figure 5.21 below.

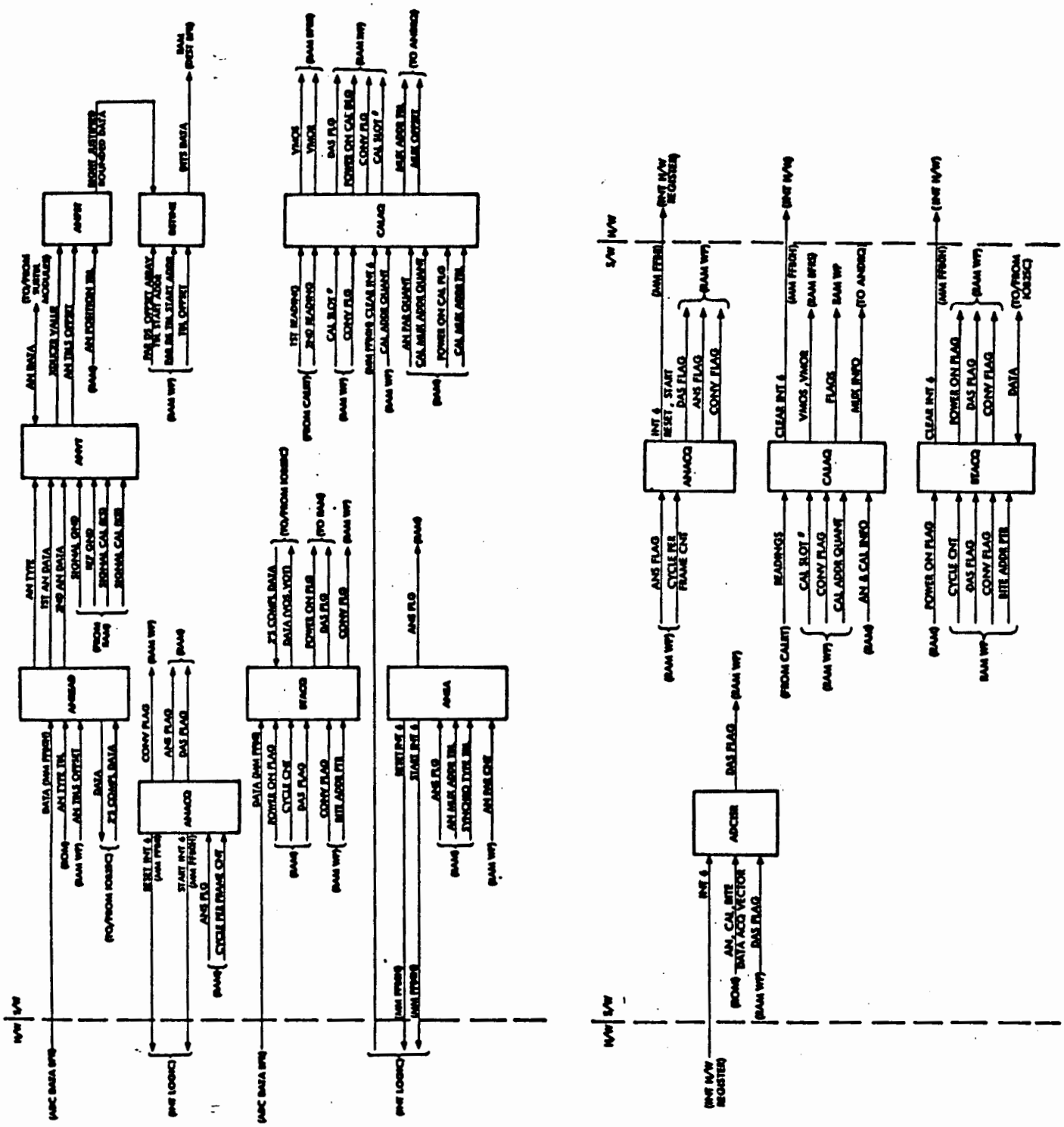


Figure 5.21 Analog Section Data Flow

As mentioned previously, the whole of the data acquisition, formatting and destination routing is table driven, so that for different configurations of the system where different parameters have to be acquired, only the data base tables need to be changed.

The analog acquisition and calibration data base configuration is shown in figure 5.22 below.

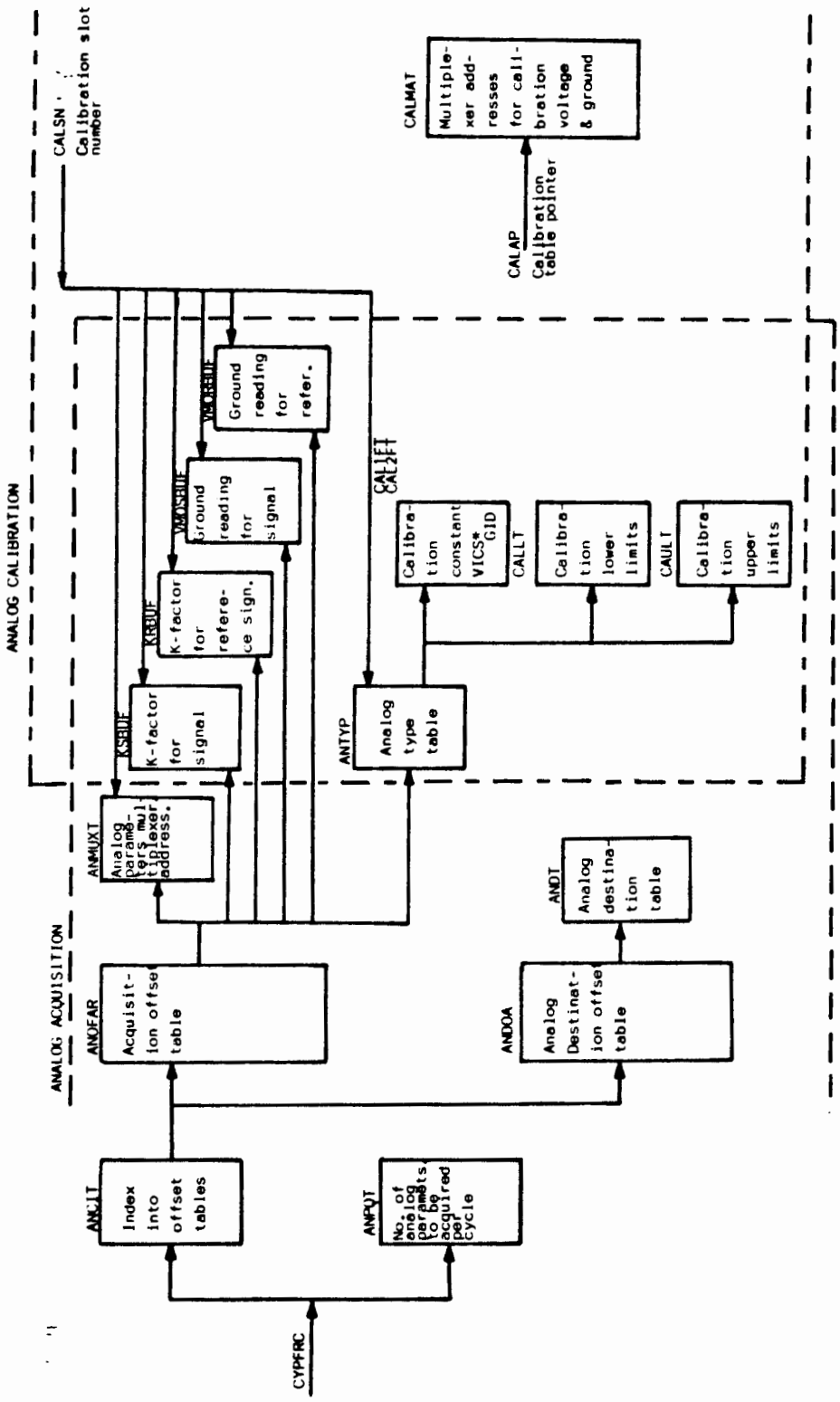


Figure 5.22 Analog Section Data Base Configuration.

5.2.5.4.1 Analog Data Acquisition -

The analog data acquisition section is entered via the analog data ready interrupt through the analog interrupt service routine ADCISR. A module tree of this section is shown in figure 5.23 below.

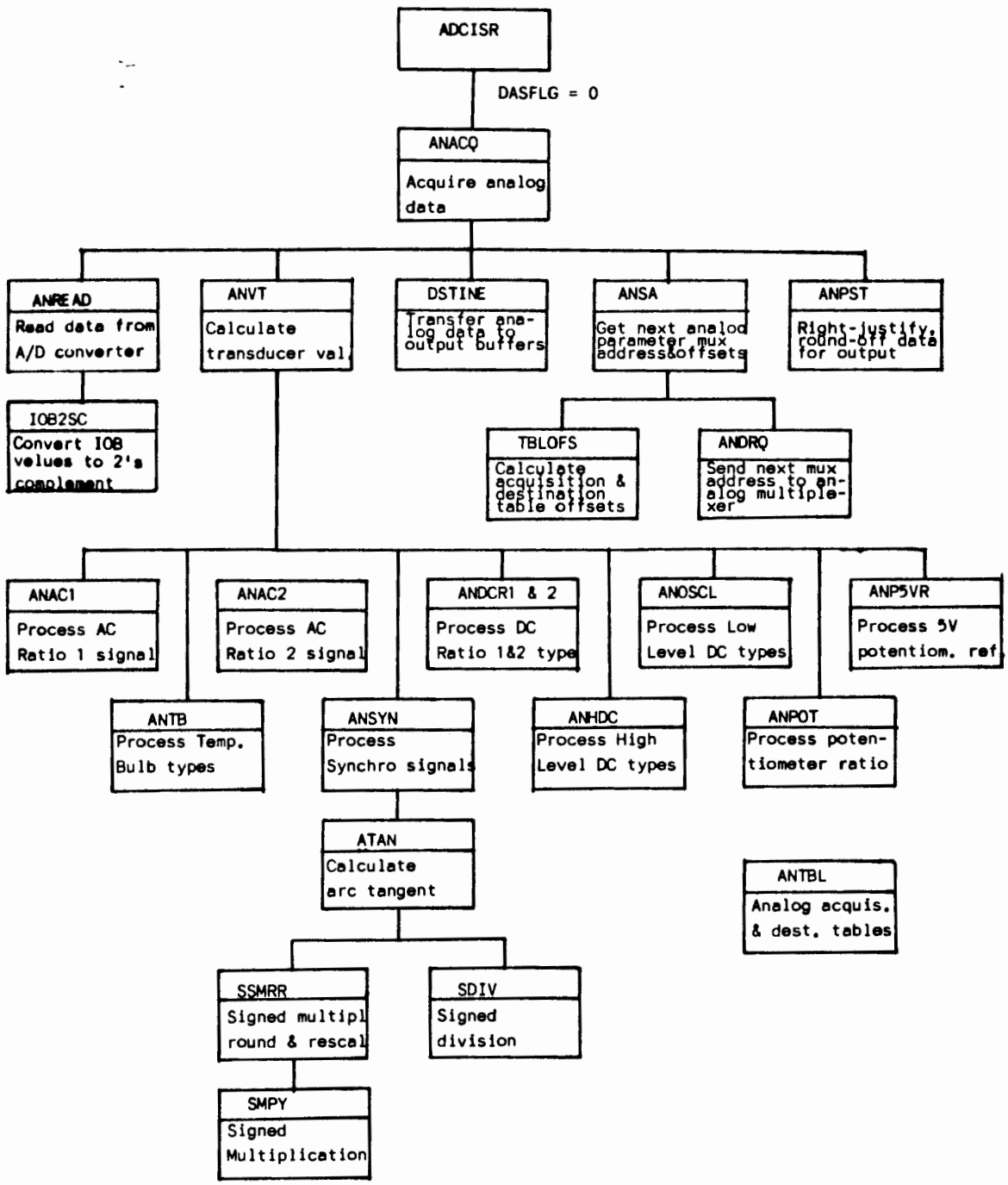


Figure 5.23 Analog Data Acquisition Module Tree.

Analog data is divided up into a number of different signal types with an assigned type-id as follows:

Signal	Type-Id
Low Level DC absolute (No Reference)	0
High Level DC absolute(No Reference)	1
Potentiometer	2
Internal 5V Potentiometer Reference	3
Temperature Bulbs (No Reference)	4
AC Ratio Type 1	5
DC Ratio Type 1	6
DC Ratio Type 2	7
Synchro	8
AC Ratio Type 2	9
Thermocouples (Not used in this system)	10

The electrical characteristics of these signal types are described in more detail in section 2.7.1.

The general type of calculations that an analog signal goes through to correct for any type of system offset is as follows :

VICS/R - Calibration input voltage for signal or reference (Constant).
GIS/R - Ideal gain for signal or reference (Constant).
VMCS/R - Measured calibration voltage for signal or reference.
VMOS/R - Measured ground voltage for signal or reference.
VMTS/R - Measured transducer voltage for signal or reference.
VCTS/R - Calibrated transducer voltage for signal or reference.

$$VCTS = (VMTS - VMOS) * KS$$
$$VCTR = (VMTR - VMOS) * KR$$

where
$$KS = \frac{GIS * VICS}{VMCS - VMOS}$$
$$KR = \frac{GIR * VICR}{VMCR - VMOR}$$

The values GIS * VICS and GIR * VICR are constant and are stored in tables CAL1FT and CAL2FT in the analog data base.

For an absolute signal

$$VOUT = VCTS$$

For a ratio type signal

$$VOUT = \frac{VCTS}{VCTR} = R$$

The value VOUT is then converted into counts (0 to 4095) and output. The signal types that special attention should be given to, are synchros for the reason that they somewhat differ in their calculations of the output voltage.

For each analog parameter, the first interrupt is ignored to enable the hardware to stabilize, and the data is only read after the second interrupt.

Therefore upon entry into ANACQ, it checks the flag ANSFLG to see if the current interrupt is the first or the second one. If it is only the first one, the interrupt is reset, re-enabled and the second analog-to-digital conversion process is started.

On the second interrupt, the module ANREAD is called to read VMTS and VMTR. Then ANVT is called to calculate VCTS and VCTR, and then in turn ANVT calls the appropriate routine for the signal type to calculate VOUT and then convert the result to a 12-bit binary count.

After this, ANPST is called to position and format the result for output. The module DSTINE places the formatted result in the pre-determined location in the output buffers for output.

At the end of all this, ANSA is called to check if there are any more parameters that have to be acquired during the current acquisition cycle. If so, then the table offsets and pointers for the next parameter are determined by TBLOFS, and ANDRQ sends the next set of MUX addresses to the analog multiplexer.

If no more parameters are to be acquired during the current cycle, the data acquisition status flag, DASFLG is set to either 1 or 2 for calibration or power supply BITE readings respectively, depending on the value of the cycles-per-frame counter, CYPFRC. This value of CYPFRC is determined by checking the various parameter quantity tables (number of parameters to be acquired in each cycle), to determine the least busy cycles in each subframe.

A PDL description of ANACQ is given below:

```
ANACQ PDL Description
---------------------
IF ANSFLG = SET FOR DUMMY READ
   SET ANSFLG = -1
   RESET INT6 (START ADC)
   START INT6
ELSE
   CALL ANREAD TO READ VMTS AND VMTR DATA
   CALL ANVT TO GET XDUCER VALUE
   CALL ANPST TO RIGHT JUSTIFY ANALOG DATA
   CALL DSTINE TO PLACE DATA INTO DESTINATION BUFFERS
   CALL ANSA TO SEND ANALOG ADDRESS
   IF DATA ACQUISITION FINISHED
        COMPARE CYCLE/FRAME COUNT WITH 3
        IF CYCLE/FRAME COUNT = 4TH CYCLE
                SET DASFLG = 2 (SELECT NEXT BYTE IN INT6)
                CLEAR CNVFLG TO JUMP OFFSET IN BTACQ
                CALL BTACQ TO ACQUIRE BITE DATA
        ELSE
                SAVE SUBFRAME CYCLE COUNT
                IF CYCLE COUNT = 3 (4TH CYCLE)
                   SET DASFLG = 1 (TO SELECT
                   CALIBRATION ACQUISITION (CALAQ)
                         IN NEXT INTERRUPT)
                   CLEAR CNVFLG TO JUMP OFFSET IN
                         CALAQ
                   CALL CALAQ TO ACQUIRE CAL. DATA
                ENDIF
        ENDIF
   ENDIF
ENDIF
RETURN
```

The module ANREAD is called whenever an analog-data-ready interrupt is received to read the converted data from the analog-to-digital converter. The data is initially in 14-bit Inverse Offset Binary form (IOB) and is converted to 12-bit 2's complement via the module IOB2SC, prior to any calculations being made.

If the signal type is a ratio (ie. AC ratio or DC ratio or potentiometer types) the output of the ADC is read twice the first time for the transducer (VMTS) output and the second time for the reference (VMTR) value. The hardware automatically latches in the VMTR value after VMTS is read. Absolute signals like low level DC (LLDC) need only VMTS. The signal type is determined by the corresponding entry in the table ANTYP, which contains the list of signal types.

ANVT calculates VCTS = (VMTS - VMOS) * KS and if necessary VCTR = (VMTR - VMOS) * KR.

Then it calls the appropriate routine for the signal type to calculate VOUT and the counts equivalent. (0 - 4095 for 0 - 5 volts). All signals are scaled down to 5 volts. The signal type routines are as follows.

ANOSCL - Low Level DC Absolute (LLDC) No Scalar
ANHDC - High Level DC (HLDC)
ANPOT - Potentiometer reading
ANPSVR - Potentiometer Reference
ANTB - Temperature Bulb
ANAC1 - AC Ratio 1
ANAC2 - AC Ratio 2
ANDCR1 - DC Ratio 1
ANSYN - Synchro
ANDCR2 - DC Ratio 2

The general type of calculation for most signal types is to first calculate VOUT and then scale the value down to 0 - 5 Volt scale to get the counts equivalent.

The calculations for synchro type signals are somewhat different. These calculations are done by ANSYN using the following synchro equivalent circuit:

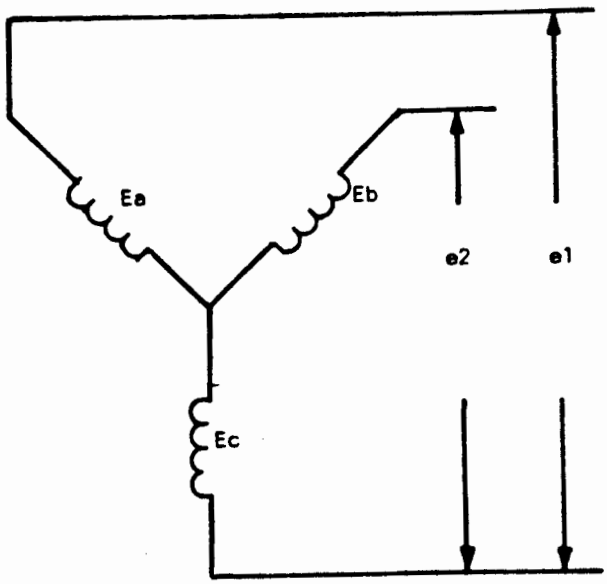


Figure 5.24 Synchro Equivalent Circuit.

Ea = Es Sin PHI
Eb = Es Sin (PHI+240)

$$Ec = Es \sin (PHI+120)$$

Where $Es = Em \sin wt$ and w is the angular frequency of the reference voltage.

The synchro angle is determined by the ratio of e1 and e2 and the phase of e1 and e2 with respect to the reference voltage.

e1 and e2 are defined as follows:

$$e1 = Es \sin PHI + Es \sin (PHI + 120)$$
$$e2 = Es \sin (PHI + 120) + Es \sin (PHI + 240)$$

the equations can be rewritten as follows:

$$e1 = Es \sin (PHI + 60)$$
$$e2 = Es \sin PHI$$

If $E1 = |e1|$ and $E2 = |e2|$ then R, the ratio of E2 to E1 or E1 to E2 ($R <= 1$) for say $E2 < E1$

$$R = \frac{E2}{E1} = \frac{\sin PHI}{\sin(PHI + 60)}$$

Substitute

$$\sin(PHI + 60) = (\cos PHI)*(\sin 60) + (\sin PHI)*(\cos 60)$$

$$\frac{\sin PHI}{\cos PHI} = \frac{0.866R}{1-0.5R} = \tan PHI$$

therefore $PHI = \text{Arc Tan} \frac{0.866}{1-0.5R} = PHI1$

for synchro angles of 150 to 180

$$R = \frac{\sin PHI}{\sin (PHI + 60)}$$ which results in

$$PHI = \text{Arc Tan} \frac{0.866R}{1 + 0.5R} = PHI2$$

The resultant synchro angle is then calculated using the following table.

CASE	Signal versus Reference	SIGN VCTS	SIGN VCTR	RATIO	ANGLE EQUATION
5	|VCTS|>|VCTR|	+	+	|VCTR|	VT=240-PHI1
6		+	-	R=------	VT=240+PHI2
2		-	+	|VCTS|	VT= 60+PHI2
1		-	-		VT= 60-PHI1
4	|VCTR|>|VCTS|	+	+	|VCTS|	VT=120+PHI1
7		+	-	R=------	VT=300-PHI2
3		-	+	|VCTR|	VT=120-PHI2
8		-	-		VT=300+PHI1

Table 5.1 *Synchro Conversion Relationship.*

The resultant angle VT is then represented in counts as :

$$1 \text{ count} = \frac{360 \text{ degrees}}{4096}$$

The arctan calculations are done in the module ATAN using a numerical approximation method derived from a Taylor expansion, (Reference 23, 25) where:

$$\arctan(Y/X = Z) = ((((C9*Z^{**}2+C7)*Z^{**}2+C5)*Z^{**}2+C3)*Z^{**}2+C1)*Z$$

where $C1 = 0.3183026$
$C3 = -0.1058774$
$C5 = 0.0616068$
$C7 = -0.370617$
$C9 = 0.0167602$

The modules SDIV, SMPY and SMRR are arithmetic routines for signed divide, multiply and scaling.

In the case of AC ratios, the phase difference between the signal and the reference has to be considered when the ratio of the two signals is calculated.

$$VOUT = \frac{VCTS}{VCTR} = R$$

$R < 0$ if VCTS is out of phase with respect to VCTR ($VCTS < 0$) and $R > 0$. If VCTS is in phase with VCTR. Therefore, to calculate the counts equivalent, the following formula is used.

$R >= 0$: Counts = 2047 (1 + R)
$R < 0$: Counts = 2047 (1 - R)

ie. for VCTS = -5V Counts = 0000
VCTS = +5V Counts = 4095

The potentiometer and DC ratio output is simply the ratio of VCTS with VCTR and then counts equivalent is calculated as follows:

$$Counts = \frac{VCTS}{VCTR} * 4095$$

The temperature bulb is a transducer whose resistance increases proportionally to temperature. The type of transducer used in this case has a resistance range of 68.27 ohms to 242.7 ohms. A constant current source of 12 mA provides a voltage at the transducer terminals proportional to the temperature. The calculations are as follows:

R = 68.27; Vo = 68.27 * .012 * GID = 1.40647 V
R = 242.7; Vo = 242.7 * .012 * GID = 5.00001 V
Full Scale - Offset = 5.00001 V - 1.40647 V = 3.59354 V

To Scale to 5 Volts Multiply by $\frac{5.00001}{3.59354} = 1.39139$

Hence VT (Scaled) = (VCTS - 1.40647) * 1.39139

$$Therefore\ Counts = \frac{4095}{5} * VT$$

The module ANDRQ supplies the MUX addresses for the parameter to be acquired. These MUX addresses select the channel on each of the three AMX modules which will be routed to the ADC module, and the type of conversion (dual or single). They also select the system gains to be applied to the input signal, and which reference will be used. The CRU (Communications Register Unit) address space for the analog multiplexer, design, is from 300(Hex) to 34F(Hex) and is divided up as follows:

300 - 317	Select the Channel
318 - 31C	Select which of the 3 AMX's
31D - 31F	Select signal type (AC or DC)
320 - 329	Determines gains of the programmable gain amplifiers
32A - 32E	Select the AC reference voltage
34D - 34E	Select calibration voltages
34F	Selects sample/hold (1 of 2) amplifier

Therefore, having set up the conditions for the acquisition of the parameter, the "START ADC" signal is given and A/D conversion begins.

The module ANTBL contains all the analog acquisition tables:

ANPQT	*- analog parameter quantity table contains the number of analog parameters to be acquired in each cycle.*
ANCIT	*- Offset into ANOFAR*
ANOFAR	*- Offset into ANMUXT*
ANMUXT	*- Transducer MUX Tables*
ANTYPT	*- Analog parameter type*
ANDOA	*- Destination Table offset*
ANDT	*- Destination Table*
CALMAT	*- Calibration MUX Table*
CAL1FT/CAL2FT	*- Calibration constants (VICS * GID)*
CALLT/CALULT	*- Calibration upper and lower limits*
ANPT	*- Analog position table*

5.2.5.4.2 Analog Calibration -

The module tree of the analog calibration section is shown below:

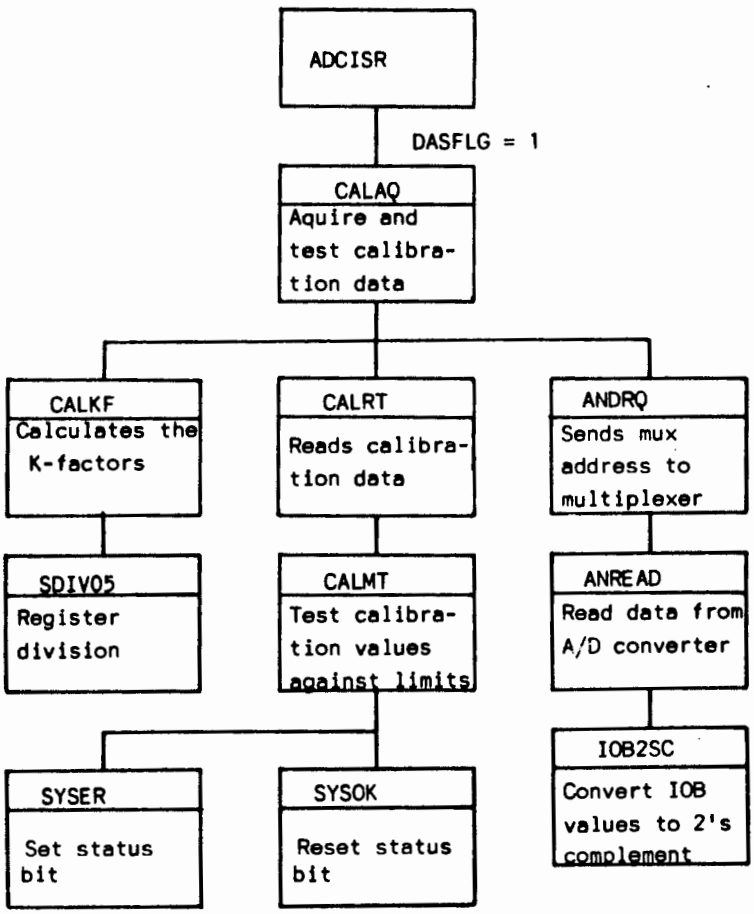


Figure 5.25 Calibration Section Module Tree

The module CALAQ is entered via the analog data-ready interrupt service routine whenever DASFLG = 1. The calibration procedure for each channel is performed in four phases, determined by the flag CNVFLG. These phases are:

1. *Send MUX address for ground channel to multiplexer and start conversion.*
2. *Read and test ground channel data and store results.*
3. *Send MUX address for calibration voltage to multiplexer and start conversion.*
4. *Read and test calibration voltage data, calculate and store K-factors (section 5.2.5.4.1).*

Each time a calibration data item is read, it is checked against pre-defined upper and lower limits in the analog tables

(CALULT,CALLT), and if not within these limits, a status bit is set in the system status buffer.

```
CALAQ PDL Description
---------------------
IF CONV FLAG < 5
  DO CASE CONV FLAG
  =0:   CONV FLAG = CONV FLAG + 2
        IF CAL SLOT NUMBER = OR < 0
                CAL SLOT NUMBER = AN PARAM COUNT
                CAL ADDR QUANT = CAL MUX ADDR QUANT
                SET POWER-ON CAL FLAG = -1 (END OF POWER
                                            ON CAL TEST)
        ENDIF
        CAL SLOT NUMBER = CAL SLOT NUMBER - 2
        CAL ADDR QUANT = CAL ADDR QUANT - 1
        MUX ADDR = CAL MUX ADDR TABLE
        OFFSET = CAL ADDR QUANT
        CALL ANDRQ TO REQUEST GND DATA
   =2:  CONV FLAG = CONV FLAG + 2
        OFFS = 0
        CALL CALRT TO READ AND TEST GND CAL DATA
        VMOS-BUFF (INDEXED BY CAL-ADDR QUANT) = 1ST READ
        VMOR BUFF (INDEXED BY CAL-ADDR QUANT) = 2ND READ
        MUX ADDR = CAL MUX ADDR TABLE
        OFFSET =  CAL ADDR QUANT
        CALL ANDRQ TO REQUEST VOLTAGE DATA
   =4:  CONV FLAG = 0
        DAS FLAG = POWER-ON CAL FLAG
        OFFS = 1
        CALL CALRT TO READ AND TEST VCAL DATA
        CALL CALKF TO CALCULATE KS/KR FACTOR
        IF DAS FLAG < 0 (NOT IN POWER-ON)
                UNDO CONV FLAG
        ENDIF
   ENDDO
   CLEAR INTERRUPT 6 (ADC INTERRUPT)
ENDIF
RETURN
```

CALRT reads the converted data by calling ANREAD which in turn calls IOB2SC to convert the 14 bit Inverse Offset Binary (IOB) data to 2's complement.

It then calls CALMT, to check the value of the calibration and ground voltage against the upper and lower limits from tables CALLT and CALULT. If the values exceed the limits, it sets or resets a status bit in the system status buffer (SYEBF) by calling SYSER or SYSOK respectively. The calibration data is also output to the system calibration buffer SYCALB for output to

the superframe, in word 64 subframe 3 of the DFDR.

CALKF is called to calculate the K-factors KS and KR where:

$$KS = \frac{GIS * VICS}{VMCS - VMOS} \quad \text{and}$$

$$KR = \frac{GIR * VICR}{VMCS - VMOS}$$

These values are then stored to be used later for the calculation of the calibrated value of the transducer channel output. The PDL description is given below.

```
CALKF PDL Description
---------------------
VS = VMOS-BUFF (INDEXED BY CAL-SLOT-NUMBER) - VMCS
IF VS NOT = 0
        CALL SDIV05 (KS = (VICS*GIS / VS)
ENDIF
KS-BUFF (INDEXED BY CAL-ADDR-QUANT) = KS
VR = VMOR-BUFF (INDEXED BY CAL-SLOT-NUMBER) - VMCR
IF VR NOT = 0
        CALL SDIV05 (KR = (VICR*GIR) / VR)
ENDIF
KR-BUFF (INDEXED BY CAL-SLOT-NUMBER) = KR
RETURN
```

5.2.5.4.3 Power Supply BITE Acquisition -

The Power Supply BITE (Built In Test Equipment) Acquisition section works in a similar way to the analog data and calibration acquisition. The main acquisition module is BTACQ and is entered via the analog data-ready interrupt service routine, ADCISR, when DASFLG equals 2. The purpose of this section is to read the converted values of the various power supply voltages, compare them to a set of pre-defined upper and lower limits, and if not within these tolerance limits (section 3.3.3.2.1), set a status bit in the system status buffer.

A module tree of the power supply BITE section is shown in figure 5.26.

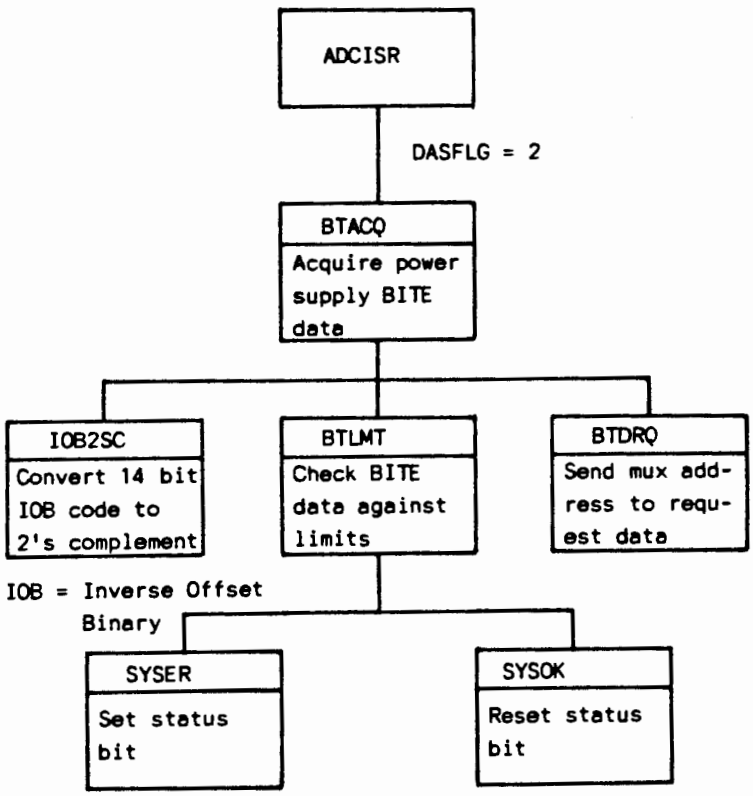


Figure 5.26 Power Supply BITE Section

5.2.5.5 DITS Data Acquisition -

DITS data acquisition is achieved via two hardware modules, each capable of addressing 8 channels and a total of 256 different parameters per module. Each of the DITS modules has its own independent interrupt vector, with its own interrupt service routine, namely DR1ISR and DR2ISR. Beyond the interrupt service stage, the DITS acquisition section is identical for both DITS 1 and DITS 2. Figure 5.27 shows a module tree of the DITS data acquisition section.

A different set of DITS parameters will be acquired depending on what type of engine configuration has been installed in the aircraft (Pratt and Whittney or GE). The variable SSEG, determined during power-on initialization by reading the Ident discretes, is used as a pointer to the appropriate sections in

the tables.

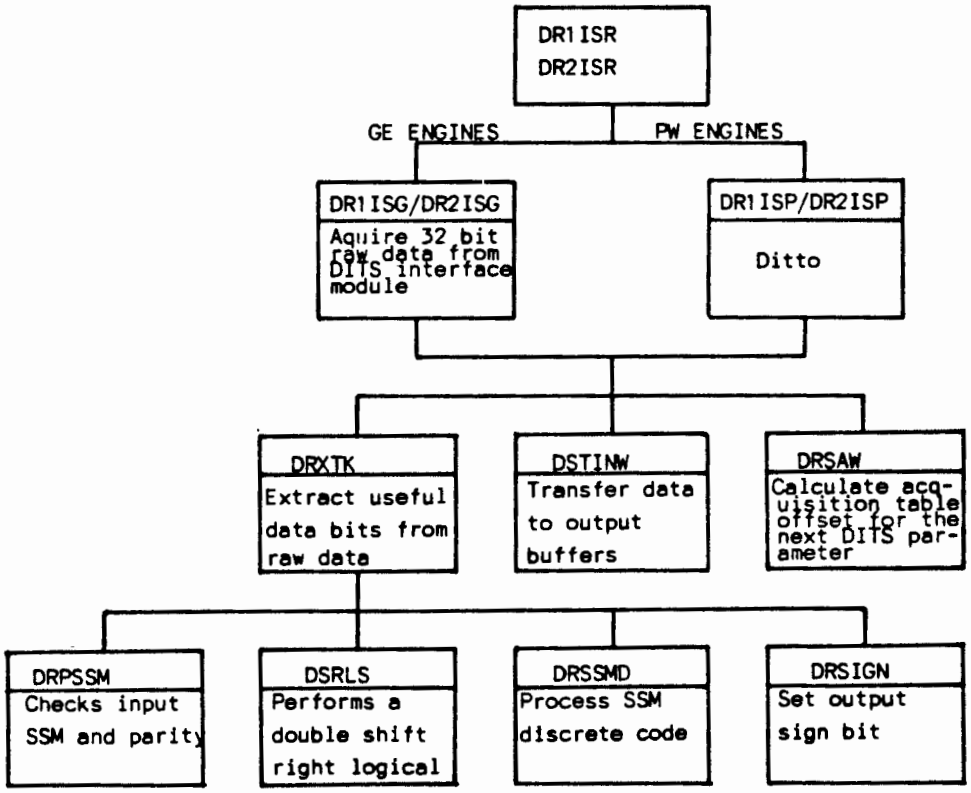


Figure 5.27 DITS Data Acquisition Module Tree.

The modules DR1ISR and DR2ISR are entered via the interrupt vectors and direct control to the appropriate acquisition routine via SSEG. The acquisition routines are DR1ISG and DR2ISG for GE engines, and DR1ISP and DR2ISP for Pratt and Whittney.

On entry, the acquisition routine reads the DITS hardware addresses, FF84 (FF88 for DITS 2) for the least significant 16 bits, and address FF86 (FF8A for DITS 2) for the most significant 16 bits, to acquire the 32 bit value of the DITS parameter being processed.

The raw DITS parameter consists of 32 bits of which the 10 least significant bits form an identifier consisting of an 8 bit label (0-255) and a 2 bit source-destination identifier code (SDI). The data can be all or part of bits 10 to 27. Bit 28 is used as a sign bit, bits 29 and 30 form the SSM code and bit 31 is used

as a parity bit.

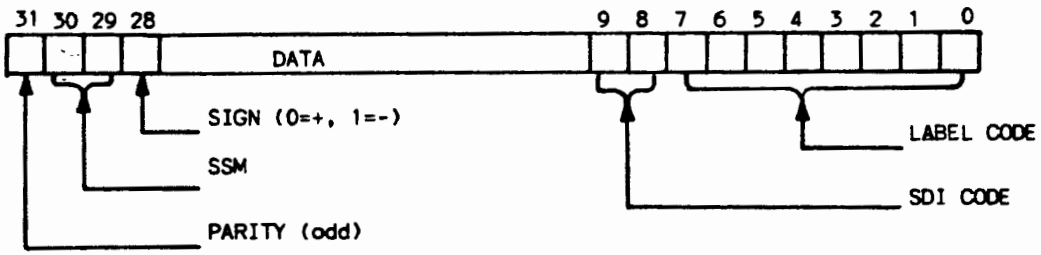


Figure 5.28 DITS Data Word

The SSM (Source Status Modifier or System Status Matrix) code is used to determine the state of the data source with the following possible conditions:

Bit 30	Bit 29	Status
0	0	Failure
0	1	No computed data
1	0	Valid Data
1	1	Valid Data

Table 5.2 DITS System Status Matrix

The routine DRXTK is then called by the acquisition routine to extract the valid data bits and format the data for output using the information from the parameter position table, DRxPST. This module then in turn calls DRSIGN to determine the sign (+ or -) of the data, DRPSSM and DRSSMD to determine the condition of the SSM code and parity. The output is set to FFFE if an SSM error it detected and FFFF if a parity error is detected. (See requirements section 3.3.1). The parameter is now formatted and ready for output.

The routine DSTINW is now called to place the formatted parameter onto the output buffers according to the destination table, DRxDT.

Finally, the routine DRSAW is called to check if any more parameters are to be acquired, and if so, to determine the address and parameter number (parameter id) of the next parameter to be acquired. It then sends it to the DITS hardware module, and resets the interrupt. The operation is explained in section

4.4.2.2.

The DITS data flow is shown in figure 5.29 below.

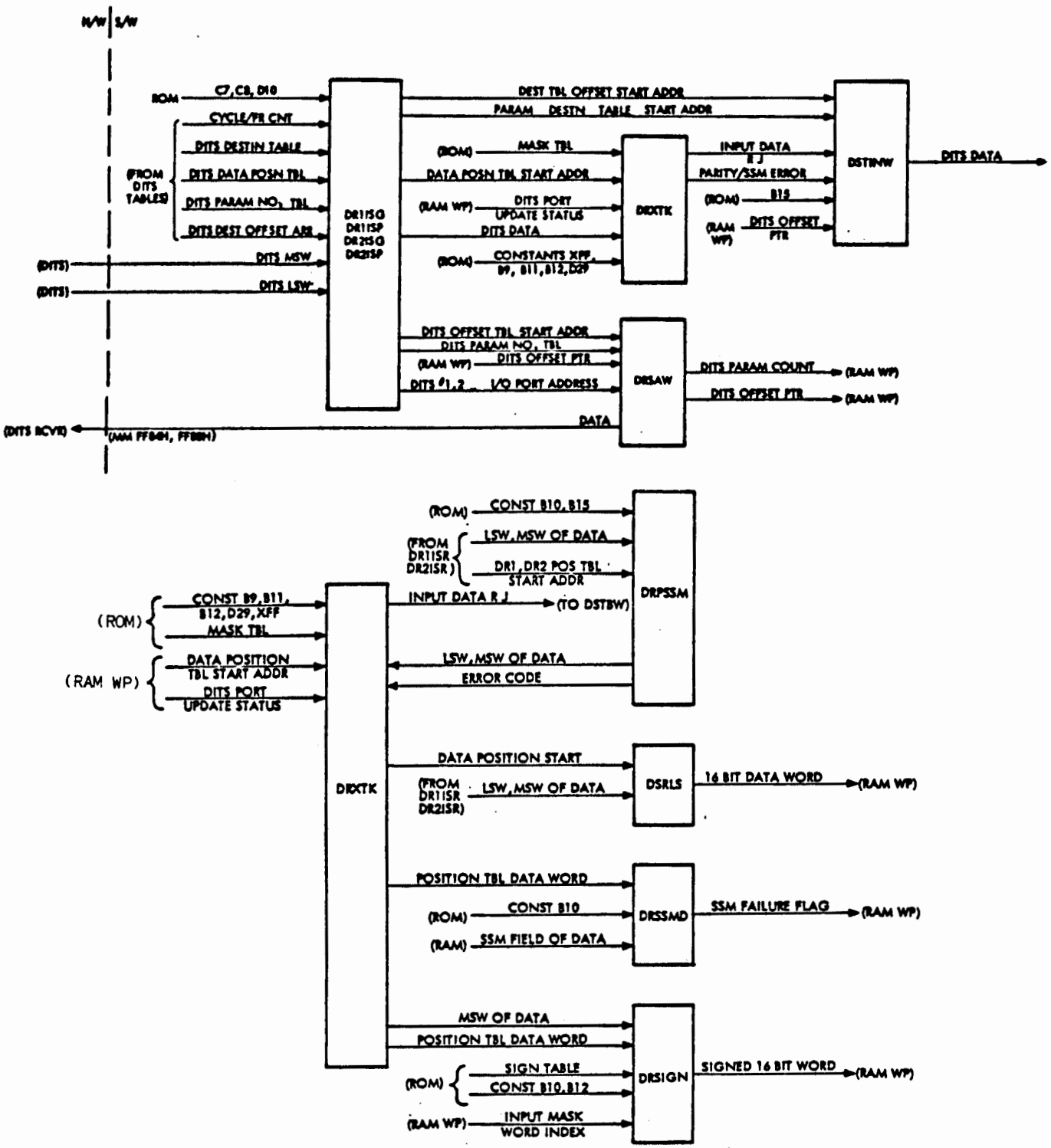


Figure 5.29 DITS Data Acquisition Data Flow

Figure 5.30 shows the structure of the DITS data base.

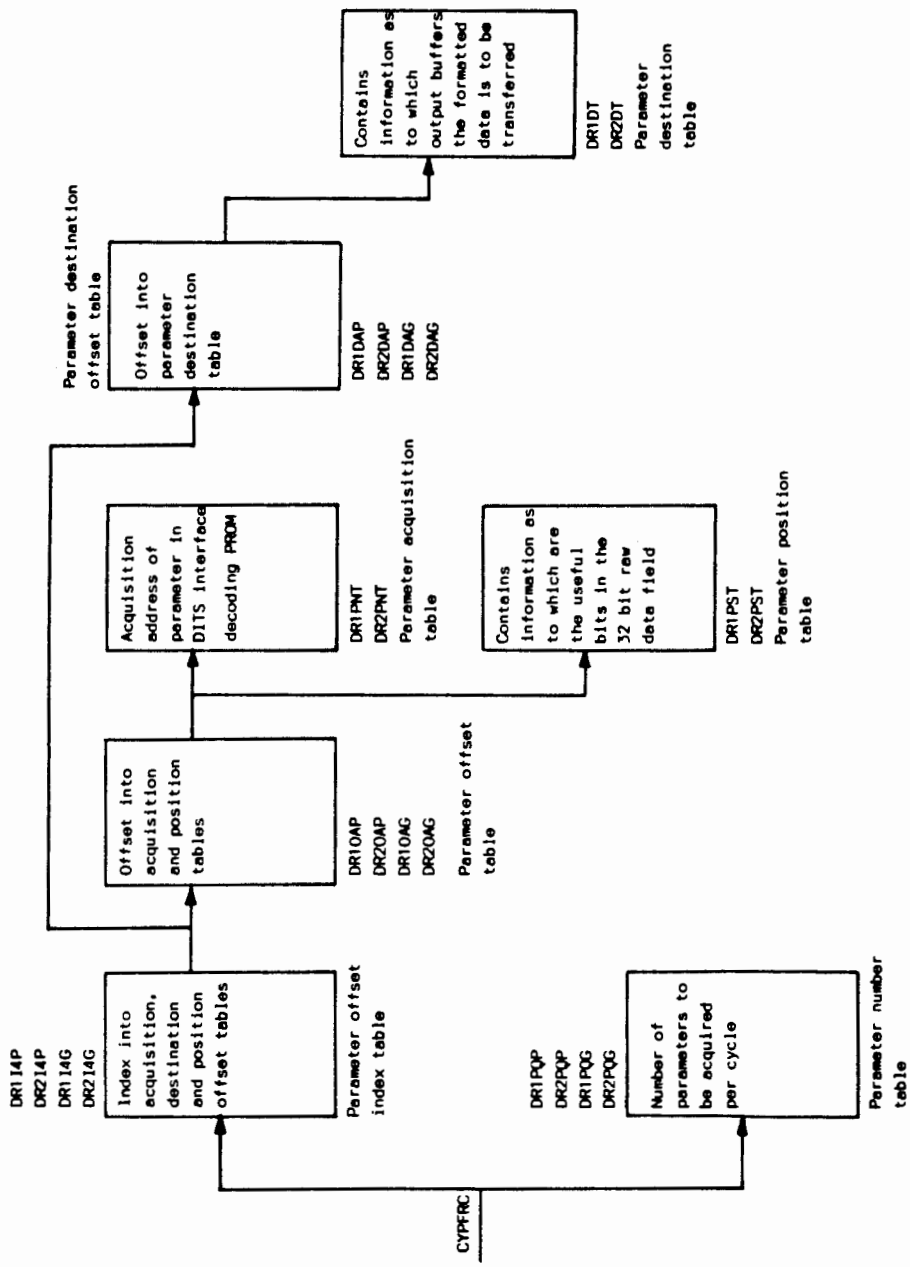


Figure 5.30 DITS Data Base Configuration

The cycle-per-frame counter CYPFRC is used to access an entry in the parameter quantity table to determine the number of parameters to be acquired during the current acquisition cycle.

It is also used to get an entry from the offset index table to determine the entry point in the acquisition position and destination offset tables.

The acquisition table contains a unique id for each DITS parameter to be acquired. The parameter id's are arbitrarily assigned when the DITS data base is initially set up, and the decoding prom is programmed accordingly, using this id, the parameter label, SDI and channel number to generate unique addresses for the DITS 256-word RAM.

This parameter id is then sent to the decoding prom on the DITS hardware module where it is decoded into the corresponding address for the DITS RAM, from where the current value of that specific parameter is obtained (section 4.4.2.2). See figure 5.31 below:

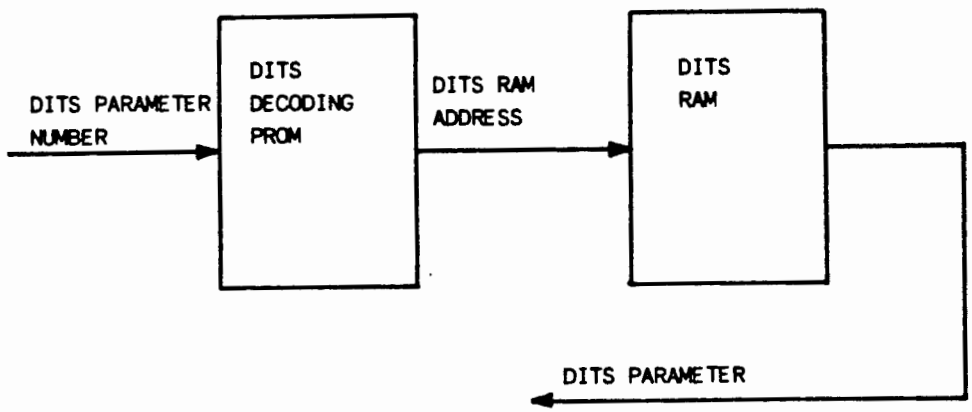


Figure 5.31 DITS Parameter Acquisition

The position table contains the information regarding which are the significant input data bits and how that parameter is to be formatted for output. Each entry in this table is a 16-bit word, and this was found to be sufficient to store the necessary information. The 16-bit table entry is divided into the following bit groups:

Bit Numbers	Information
0 - 4	Input Start Bit
5 - 8	Data Field Length - 1
9	1 = Sign 0 = No Sign
10	1 = Binary Data 0 = BCD Data
11	Not Used
12	1 = SSM Check 0 = No SSM Check
13	1 = Parity Check 0 = No Parity Check
14	Not Used
15	1 = Apply Rounding 0 = No Rounding

The destination offset table contains the offset into the destination table for each parameter. The destination table, in turn, contains the information on which output buffers (queues), the parameter is to be directed, along with the word and bit positions in these queues. Each entry in the parameter destination table is a 16-bit word, and the bits are grouped as follows:

Bit Numbers	Information
0 - 7	Output Buffer Offset
8 - 9	Buffer Type 0 = DFDR Output Buffer 1 = Inter-CPU Output Buffer 2 = Auxiliary Output Buffer 3 = Spare
10 - 13	Shift Count
14	1 = Clear destination word 0 = No destination clear necessary
15	1 = No more destinations to follow 0 = More destinations to follow

Thus, a single 16-bit entry into each of these two tables, provides all the information as to how the parameter arrives at the input, and how it should be formatted prior to output.

A PDL description of DRxISG/DRxISP and DRXTK is given below:

```
DRxISG/DRxISP PDL Description
-----------------------------
GET LSW OF DITS DATA
PUT LABEL IN LOWER BYTE OF LSW
GET MSW OF DITS DATA
CALL DRXTK - EXTRACT USEFUL DATA FIELD
CALL DSTINW - TRANSFER DATA TO OUTPUT BUFFERS
CALL DRSAW - GET NEXT PARAMETER ADDRESS
RETURN


DRXTK PDL Description
---------------------
GET PARAMETER POSITION TABLE ENTRY
RESET SSM FLAG
CHECK BIT 12 OF PARAMETER POSITION TABLE ENTRY
IF PARITY CHECK REQUESTED
        CALL DRPSSM - CHECK FOR PARITY AND SSM ERRORS
ENDIF
SAVE MSH OF INPUT DATA
CALL DSRLS - RIGHT JUSTIFY INPUT DATA
EXTRACT USEFUL DATA FIELDS
CLEAR UNWANTED DATA BITS
CHECK IF DISCRETE SSM CHECK IS REQUIRED
IF SSM CHECK REQUESTED
        CALL DRSSMD - SET APPROPRIATE CONDITION
ENDIF
CHECK BIT 9 OF PARAMETER POSITION TABLE ENTRY
IF SIGN REQUESTED
        CALL DRSIGN - SET SIGN BIT
ENDIF
IF ROUNDING REQUESTED (BIT 15)
        CHECK SSM/PARITY ERROR FLAG
        IF NO SSM/PARITY ERROR
           INCREMENT LSB
        ELSE
           DROP LSB BY SHIFTING 1 BIT RIGHT LOGICAL
        ENDIF
ENDIF
IF SSM ERROR ONLY (NOT PARITY)
        SET OUTPUT DATA FFFE
ENDIF
RETURN
```

5.2.5.6 Discrete Data Acquisition -

The discrete parameter acquisition process begins by calling the module DC1ACQ and DC2ACQ (if two discrete multiplexers are

installed), via the executive module JOB8PS at a rate of 8 times per second.

Each time, the parameter quantity remaining to be acquired for the current cycle is checked. If the amount is non-zero, then TBLOFS is called to determine the acquisition and destination tables offsets and the multiplexer channel address for the next parameter to be acquired.

Next, the module DCDRQ is called. This module sends the 8-bit MUX address of the parameter to be acquired to the appropriate discrete multiplexer (1 or 2) by writing this address to the CRU register at the address 680(hex). After a delay of 10 microseconds the discrete can be read from the CRU address 726(hex).

The discrete parameter is then right-justified and placed in the output queue in the designated bit positions (table C-2, appendix C), by calling the module DSTINE.

A module tree of the discrete data acquisition section is shown in figure 5.32 followed by PDL descriptions of DC1ACQ and DCDRQ.

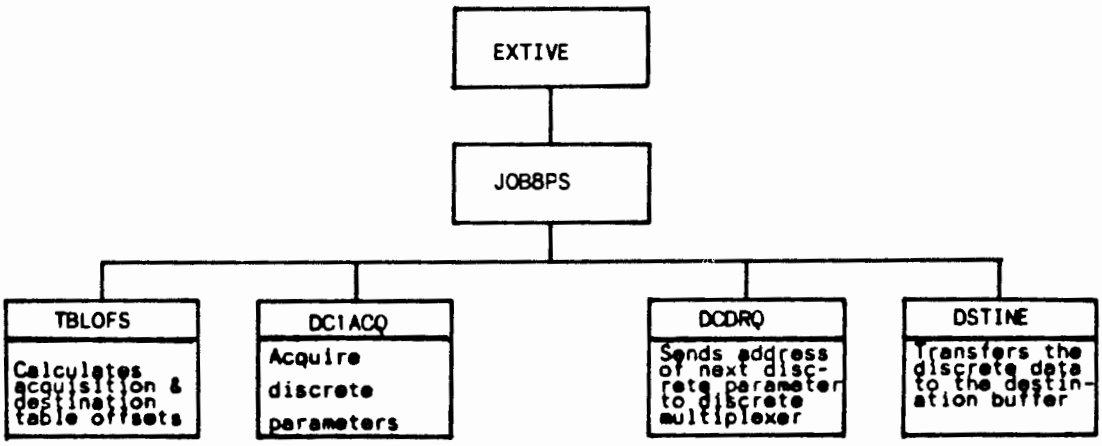


Figure 5.32 Discrete Data Acquisition Module Tree.

```
DC1ACQ PDL Description
----------------------
SET UP START ADDRESS OF 4 S/F OFFSET TABLE
DETERMINE PRESENT CYCLE FROM CYCLE COUNTER
OBTAIN PARAMETER QUANTITY FOR THIS CYCLE FROM DC1PQT
GET INDEX FOR OFFSET TABLES
DOWHILE PARAMETER QUANTITY > 0
   CALL TBLOFS - GET NEXT PARAMETER TABLES OFFSETS
   CALL DCDRQ - GET DISCRETE BIT
   CALL DSTINE - TRANSFER BIT TO OUTPUT BUFFER SPECIFIED
                 BY DESTINATION TABLE
ENDO
SET UP START ADDRESS OF EVERY S/F OFFSET TABLE
DETERMINE PRESENT CYCLE FROM CYCLE COUNTER
OBTAIN PARAMETER QUANTITY FOR THIS CYCLE FROM DC1PQT
GET INDEX FOR OFFSET TABLES
DOWHILE PARAMETER QUANTITY > 0
   CALL TBLOFS - GET NEXT PARAMETER TABLES OFFSETS
   CALL DCDRQ - GET DISCRETE BIT
   CALL DSTINE - TRANSFER BIT TO OUTPUT BUFFER SPECIFIED
                 BY DESTINATION TABLE
ENDO
RETURN
```

```
DCDRQ PDL Description
---------------------
CLEAR DISCRETE DATA REGISTER
GET DISCRETE CHANNEL ADDRESS
SEND TO DISCRETE MULTIPLEXER
WAIT 10 MICROSECONDS
GET DISCRETE CRU READ-ADDRESS
READ DISCRETE BIT
RIGHT JUSTIFY
RETURN
```

The discrete acquisition data base follows the same basic structure as that for analog and DITS, that is, it uses offset tables, parameter quantity tables, and destination tables. It does not need a parameter position table because they are single-bit parameters. This is shown in figure 5.33 below.

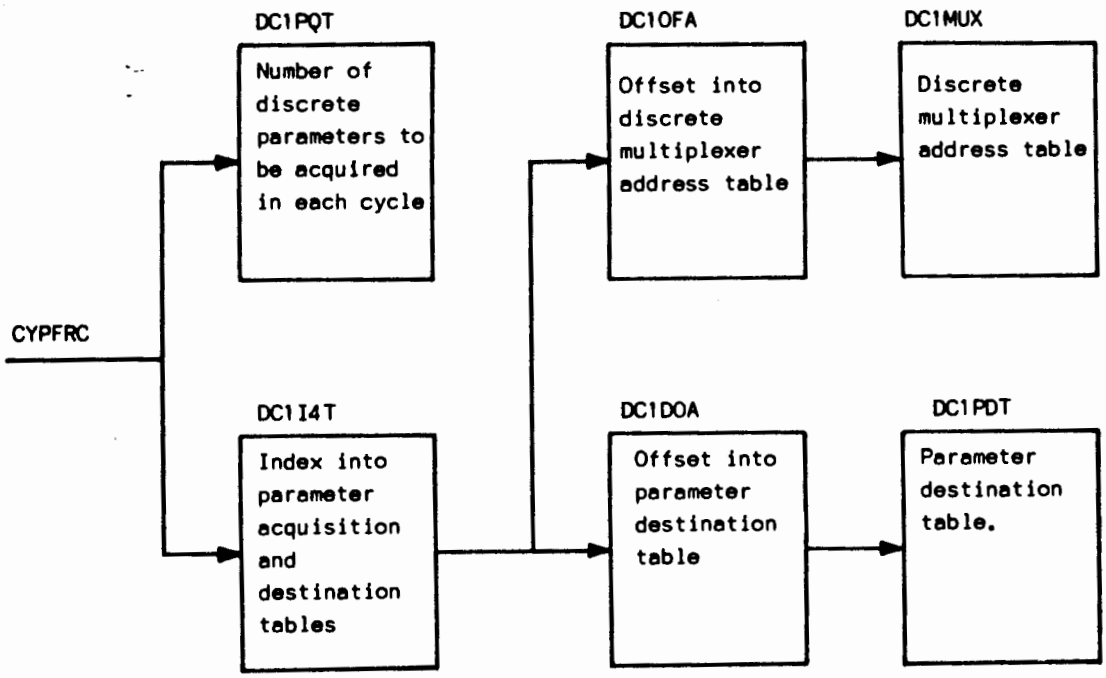


Figure 5.33 Discrete Data Base Configuration

5.2.5.7 Inter-CPU Communications -

The Inter-CPU Communications section provides the communications services between the two CPU's of the DFDAU over an RS422 serial link. Message transfers in both direction are accommodated. The messages are placed in an output queue by an application routine using the communications handler services, and the messages are transmitted in the order that they are generated.

The queue system was used so that in the case of long transmission buffers, when a transmission request might be made prior to the end of transmission of the current buffer, that request is not lost.

Two message types are received from CPU 2, namely documentary data and status data. The received messages are placed in the received message queue, and then are presented by the executive to the message processing routines for processing, in the order that they are received.

A module tree for the Inter-CPU Communications section is given in figure 5.34.

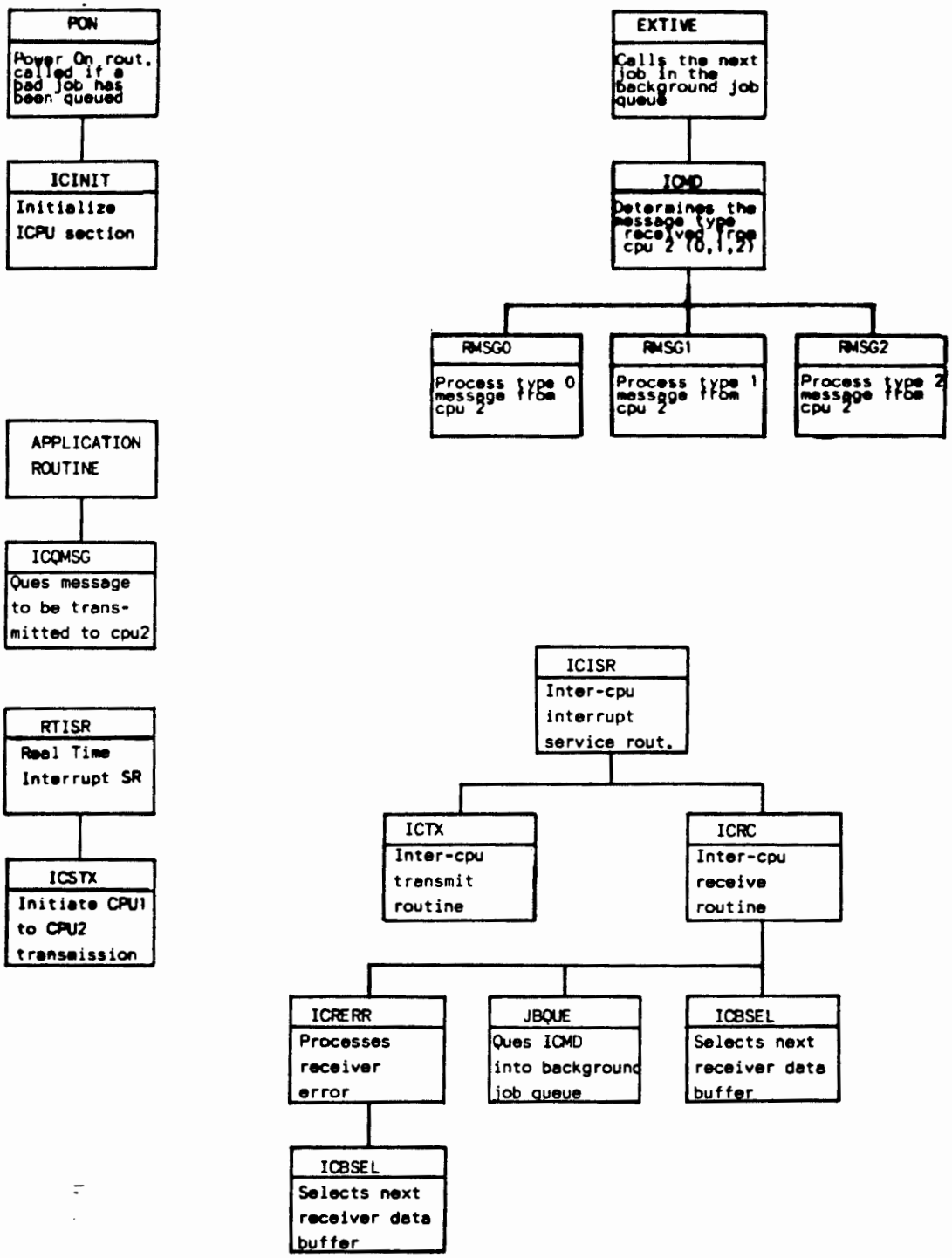


Figure 5.34 Inter-CPU Communications Module Tree.

The communications UART (Universal Asynchronous Receiver Transmitter, TMS9902) is initialized at power-on via the module ICINIT, to the following:

CPU 1 to CPU 2 transmit speed: 9600 baud
CPU 2 to CPU 1 transmit speed: 1200 baud
8 data bits
odd parity
one stop bit

Transmission to CPU 2 from CPU 1 takes place once per second and therefore the transmit speed has to be high enough (9600 baud) so that the whole of the inter-CPU transmit buffer (maximum size set to 512 words) can be transmitted during this interval.

The CPU 2 to CPU 1 transmission is set to a low speed (1200 baud) because the buffer length is only 16 words, and these transmissions take place only if there is a change of status data or documentary data on CPU 2 (ie. not at regular intervals).

A control flow diagram of the communication process is shown in figure 5.35.

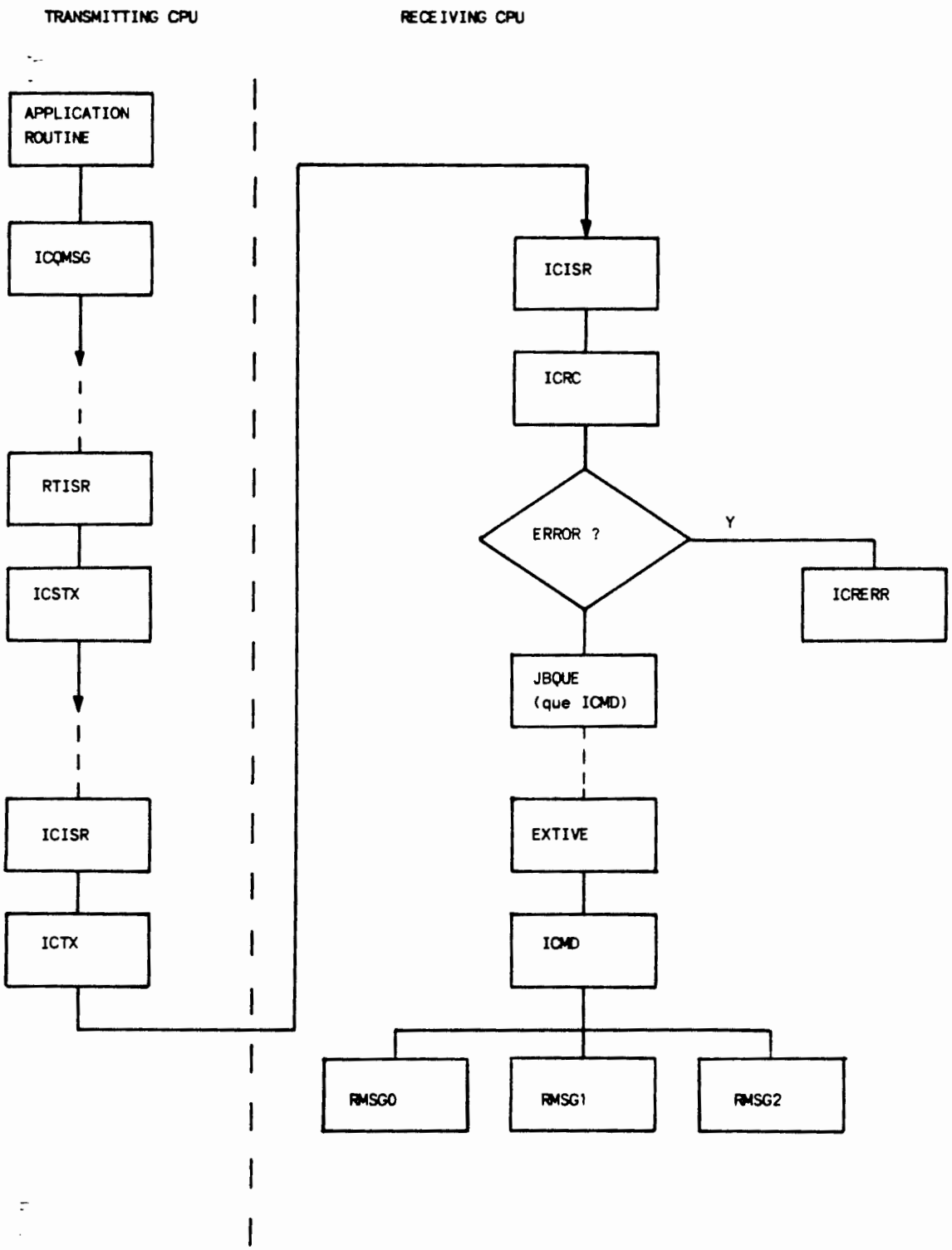


Figure 5.35 Inter-CPU Communications Control Flow

The process begins when an application routine issues a call to ICQMSG to place a message buffer in the transmit message queue. The real-time interrupt service routine in the meantime, periodically checks this queue, and if a message is waiting to be transmitted, issues a call to ICSTX to start the inter-CPU transmission. This is done by turning on the inter-CPU transmitter (setting a bit on the TMS9902 uart - see reference 3) and enabling the inter-CPU transmitter interrupt. This in turn invokes the transmitter-buffer-empty interrupt and the transmit routine, ICTX, responds by loading a character (8 bits) into the inter-CPU uart.

On the receiving side, reception of a character causes a receive-buffer-full interrupt, to which the routine ICRC responds by unloading (reading) a character from the uart.

The above process continues until the complete message has been transmitted, at which time the transmitter is turned off by resetting a bit in the CRU address space of the inter-CPU uart and at the same time, disabling the transmitter interrupt.

The received and transmitted data for the inter-CPU communications is double-buffered, that is, it uses a set of "flip-flop " buffers for transmitting and receiving, to ensure that no data is lost or overwritten. With this method, the one buffer can be loaded with new data while the other is being transmitted, and on the receiving side, the incoming data can be loaded into the one receiving buffer while the previous set is being processed. This prevents any data corruption that might occur in a single-buffer system. Figure 5.36 explains this:

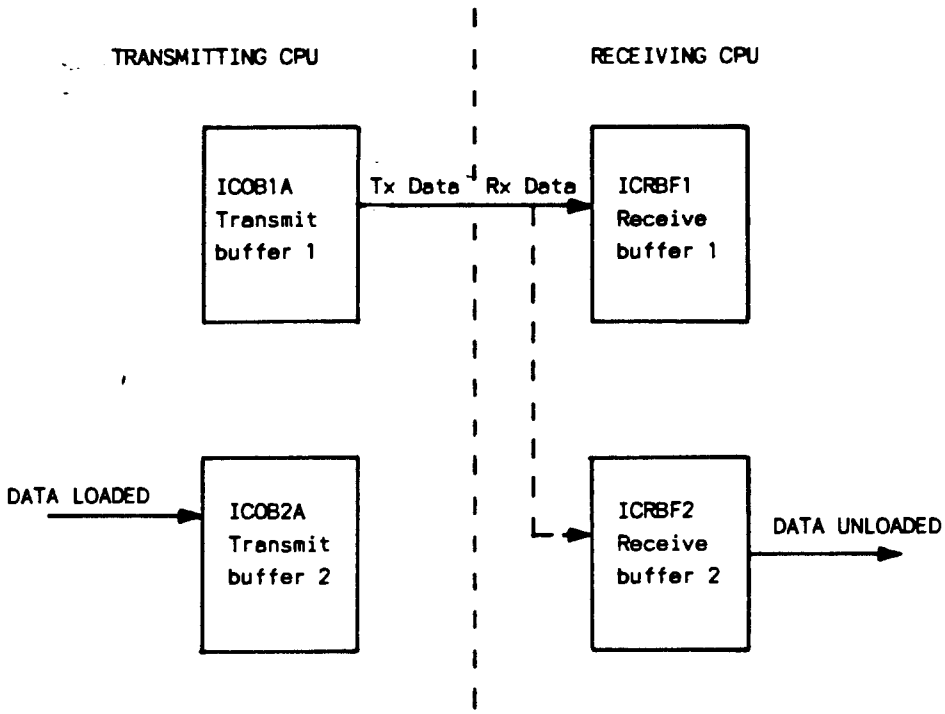


Figure 5.36 Inter-CPU Communications Data Flow

The message buffer that is transmitted from CPU 1 to CPU 2 has the following format:

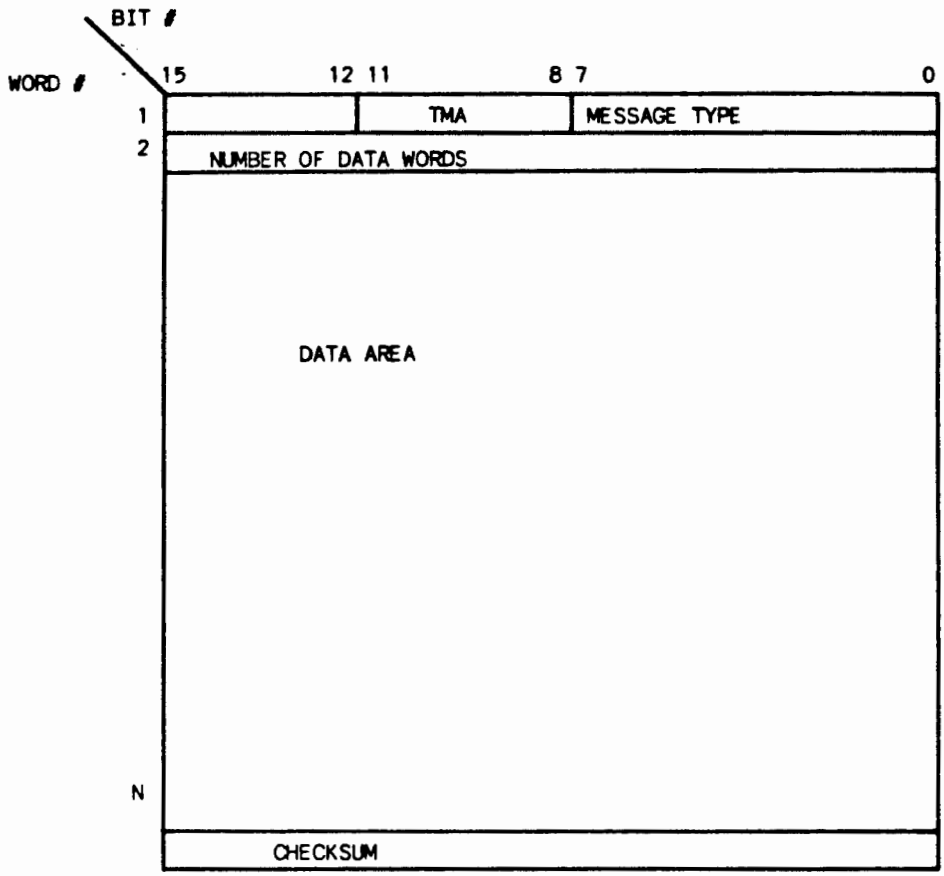


Figure 5.37 Transmit Buffer Format

MESSAGE TYPE:	*Type of message 0 - 127*
NUMBER OF DATA WORDS:	*Number of data words = n - 2*
CHECKSUM:	*2's complement of summation of word 1 to n, ignoring overflows*
TRANSMIT MESSAGE STATUS:	*(Used by transmitter) Bit 15 = 1 - Transmission not complete*

With this checksum, the sum of all message words is zero.

After the complete message has been received correctly (sum of all words in message buffer is zero), the module JBQUE is called to place the module ICMD in the executive job queue. When ICMD is called by the executive, it looks at the next message waiting to be processed, and calls the appropriate processing routine (RMSG0, RMSG1, RMSG2) according to message type.

Figure 5.38 shows the buffer sizes and message types for the inter-CPU communications:

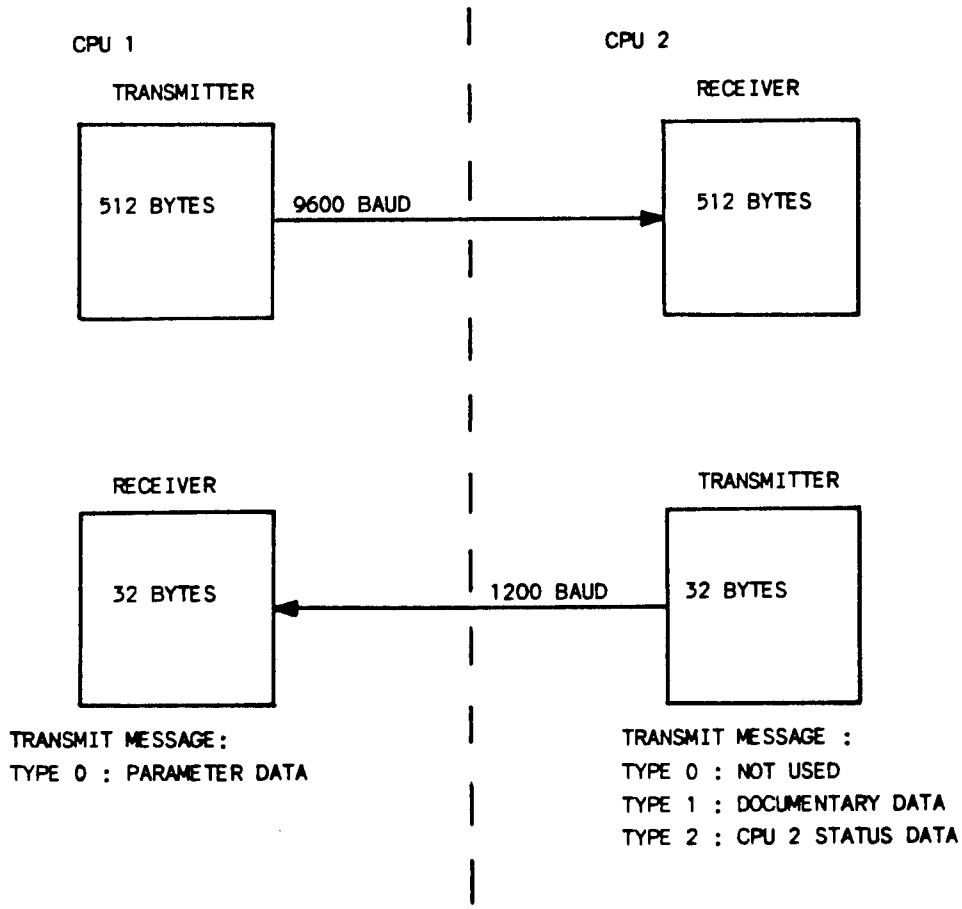


Figure 5.38 Inter-CPU Message Types

The receive message buffer format is the same as that of the transmit buffer. In this case however, bit 15 is used to show whether this buffer is being used or not. When a routine no longer needs the buffer, it resets bit 15 to logic 0, thus making it available for other incoming messages.

Referring to the diagram in figure 5.30 the following are the PDL descriptions of the inter-CPU communication modules.

```
ICINIT PDL Description
----------------------
CLEAR INTER-CPU RAM AREA
GET BEGINNING OF INTER-CPU QUEUE
SET UP INPUT AND OUTPUT POINTERS
GET 1ST INTER-CPU RECEIVER BUFFER
GET INTER-CPU 9902 START ADDRESS
ISSUE RESET COMMAND
LOAD CONTROL REGISTER WITH CONTROL BYTE
LOAD RECEIVE REGISTER WITH 1200 BAUD
LOAD TRANSMIT REGISTER WITH 9600 BAUD
ENABLE RECEIVE INTERRUPT
ENABLE INTER-CPU INTERRUPT
SET INTER-CPU TRANSMITTER NOT ACTIVE
RETURN


ICSTX - Start Message Transmission by turning on
        inter-CPU transmitter and enabling transmitter
        interrupt.

ICSTX PDL Description
---------------------
GET INTER-CPU OUTPUT QUEUE POINTER
IF OUTPUT POINTER NOT EQUAL INPUT POINTER
   IF TRANSMIT MESSAGE BYTE COUNTER = 0
               (NOT TRANSMITTING MESSAGE)
        INCREMENT OUTPUT QUEUE POINTER
        IF AT END OF QUEUE
                GET BEGINNING QUEUE POINTER
        ENDIF
        SAVE ENTER CPU OUTPUT QUEUE POINTER
        GET MESSAGE BUFFER ADDRESS
        SAVE MESSAGE BUFFER ADDRESS
        CLEAR INTER-CPU TRANSMIT MESSAGE BYTE COUNTER
        ENABLE TRANSMIT INTERRUPT
        TURN ON TRANSMITTER
   ENDIF
ENDIF
RETURN
```

ICQMSG - Queues the next message buffer to be transmitted.

ICQMSG PDL Description

```
GET MSG BUFFER START ADDRESS
STORE MSG TYPE IN MSG BUFFER
STORE NUMBER OF DATA WORDS IN MSG BUFFER
STORE MSG ACKNOWLEDGE FLAG BIT IN MSG BUFFER
STORE TRANSMIT NOT COMPLETE FLAG IN MSG BUFFER
DO UNTIL END OF BUFFER
        ADD WORDS IN MSG BUFFER
ENDDO
STORE 2'S COMPLEMENT CHECKSUM WORD IN LAST WORD OF MSG BUFFER
DISABLE ALL INTERRUPTS
INCREMENT INPUT QUEUE
INCREMENT INPUT QUEUE POINTER
IF END OF QUEUE
        SET PTR TO BEGINNING OF QUEUE
ENDIF
IF QUEUE NOT FULL
        PLACE MSG BUFFER IN QUEUE
ENDIF
RETURN
```

ICISR - Inter-CPU Interrupt Service Module. Called by hardware whenever an inter-CPU uart interrupt (receive or transmit) occurs. By checking the CRU status bits, ICISR decides whether it was a transmit or a receive interrupt and calls the appropriate routines ICTX or ICRC.

ICISR PDL Description

```
TEST RECEIVER INTERRUPT BIT
IF SET
        CALL ICRC - INTER-CPU RCVR INTERRUPT
        TEST TRANSMITTER INTERRUPT
        IF SET
                CALL ICTX - INTER-CPU TRANSMITTER INTERRUPT
        ENDIF
ELSE
        TEST TRANSMITTER INTERRUPT
        IF SET
                CALL ICTX - INTER-CPU TRANSMITTER INTERRUPT
        ELSE
                INCREMENT INTER-CPU ERROR COUNTER
        ENDIF
ENDIF
RETURN
```

```
ICRC -  Inter-CPU receive module. Called by ICISR whenever
        it determines the Inter-CPU interrupt to be a
        receive-buffer-full interrupt.

ICRC PDL Description
--------------------
IF RECEIVER ERROR BIT SET
        SAVE STATUS REGISTER
ENDIF
IF RCVR INT OCCURRED < 30 MSEC AGO
        IF RCVR MSG BYTE CTR >= 38 BYTES
                INCREMENT LONG MESS ERR. CTR
                IF CONTINUOUS RCVR INT = 3 SEC
                        DISABLE INTER-CPU RCVR INT
                ENDIF
                CALL ICRERR TO PROCESS RECEIVER ERRORS
        ENDIF
ELSE
        SET CONT LONG MSG ERROR CTR = 0
        IF RC MSG BYTECTR NOT EQUAL 0
                CALL ICRERR TO PROCESS RECEIVER ERRORS
        ENDIF
ENDIF
STORE BYTE IN RCVR BUFFER
INCREMENT RCVR MESSAGE BYTE COUNTER
IF RC MSG BYTE CTR = 38
        IF SUMCHECK OK
                IF MESSAGE TYPE LEGAL
                        SET BUFFER BUSY BIT
                        SET BUFFER START ADDRESS
                        CALL JBQUE TO PLACE JOB IN JOB BUFFER
                        CLEAR RC MSG BYTE CTR
                        CALL RC MSG BYTE CTR
                        CALL ICBSEL TO SELECT MESSAGE BUFFER
                ELSE
                        CALL ICRERR TO PROCESS ERRORS
                ENDIF
        ELSE
                CALL ICRERR TO PROCESS ERRORS
        ENDIF
ENDIF
SET RCVR TIMER = LSW OF REAL-TIME COUNTER
RETURN


38 BYTES = 32 MESSAGE + 4 OVERHEAD + 2 SUMCHECK
```

```
ICTX -  Inter-CPU Transmitter Module. Called by ICISR
        whenever the transmitter is turned on and the
        transmitter-buffer-empty signal in the uart is
        generated.

ICTX PDL Description
--------------------
IF INTER-CPU TRANSMIT ACTIVE
        IF TRANSMISSION NOT IN PROGRESS
           GET BUFFER ADDRESS
           TURN ON TRANSMITTER
           PLACE BYTE ON CRU
           CLEAR BYTE FROM BUFFER
           INCREMENT MESSAGE COUNTER
           IF LAST CHARACTER WAS TRANSMITTED
                SET MS FIELD (1ST BYTE) = TRANSMIT COMPLETE
                RESET INTER-CPU TRANSMIT COUNTER
                DISABLE TRANSMIT INTERRUPT
                TURN OFF TRANSMITTER
           ENDIF
        ELSE
           DISABLE TRANSMIT INTERRUPT
        ENDIF
ELSE
        DISABLE TRANSMIT INTERRUPT
ENDIF
RETURN


ICBSEL - Inter-CPU receiver message buffer selection.
         Called to find which message buffer will be
         available for the next received message.

ICBSEL PDL Description
----------------------
GET INTER-CPU RCVR BUFFER ADDRESS TABLE
GET NUMBER OF BUFFER ADDRESSES
DO UNTIL RECEIVE MESSAGE BUFFER AVAILABLE
        GET BUFFER START ADDRESS
        IF WORD AT START ADDRESS > OR = 0 (BFR AVAILABLE)
                SET MSB TO BUSY
        ENDIF
ENDDO
IF NO BUFFER AVAILABLE
        INCREMENT BFR NOT AVAILABLE ERROR
ENDIF
RETURN
```

ICRERR - Receiver Error Module. Called by ICRC whenever
an error in the received message is detected.

ICRERR PDL Description

```
INCREMENT ERROR COUNTER
GET BUFFER START ADDRESS
SET ERROR BIT DEFINED IN INTER-CPU ERROR TABLE
CLEAR RCVR MESSAGE BYTE COUNTER
CALL ICBSEL TO SELECT THE NEXT BUFFER
RETURN
```

RMSG1 - Receive Message Type 1 Module. Called whenever the
received message type from CPU 2 is a message type 1
i.e. documentary data. It moves the data from the
receiver buffer to the appropriate section of the
system documentary data buffer, SYDOCB.

RMSG1 PDL Description

```
SAVE RECEIVER BUFFER START ADDRESS
GET NO. OF DATA WORDS INCLUDING THE TWO OVERHEAD WORDS (4+2)
GET DESTINATION BUFFER START ADDRESS (SYDOCB)
DO UNTIL WORD NUMBER > 0
        MOVE WORD TO SYDOCB
ENDDO
MAKE RECEIVER BUFFER AVAILABLE
SET BIT 8 IN TMA WORD TO INDICATE RECEIVED OK
RETURN
```

RMSG2 - Receive Message Type 2 Module.
Called whenever the message received from CPU 2 is
a type 2 - status data. The module moves the data from
the receiver buffer to the appropriate system section
of the status buffer, SYEBF2.

RMSG2 PDL Description

```
SAVE RECEIVER BUFFER START ADDRESS
GET NO. OF DATA WORDS INCLUDING THE 2 OVERHEAD WORDS (8+2=10)
GET DESTINATION BUFFER START ADDRESS (SYEBF2)
DO UNTIL WORD NUMBER > 0
        MOVE WORD TO SYEBF2 BUFFER
ENDDO
MAKE RECEIVER BUFFER AVAILABLE
SET BIT 9 IN TMA WORD TO INDICATE RECEIVED OK
RETURN
```

ICMD - Inter-CPU Message dispatcher.
Calls the appropriate received message processing routine, RMSG0, RMSG1, RMSG2.

ICMD PDL Description

```
GET MSG BUFFER POINTER FROM CALLER'S R0
GET 1ST WORD OF MESSAGE
GET ACTUAL MESSAGE TYPE FROM LOWER BYTE OF 1ST WORD OF MESSAGE
IF LEGAL MESSAGE TYPE ( =0,1,2)
        GET MSG TYPE SUBROUTINE - RMSGx
        CALL RMSGx
ELSE
        INCREMENT R0
ENDIF
RETURN
```

5.2.5.8 DFDR Output -

The DFDR output section is entered via the DFDR interrupt service routine, DFISR. This routine services the DFDR output buffer-empty interrupt. This is a free-running interrupt occurring at a rate of 15.625 milliseconds.

Upon entry, DFISR checks if the next word to be output is the last word of the second-last 8-word cycle of that subframe (word 56), by checking the value of the output word counter DFWRC. If this is the case, the output is "force-resynchronized" with the data input (in case it has lost synchronization), by forcing the 3 least-significant bits of CYPFRC, the cycles-per-frame counter to be all 1's and setting the value of RTPCYC, the real-time clock "ticks" per cycle counter to 1, the begining of the last cycle.

This effectively sets the three least significant bits of CYPFRC to 7, the last cycle of the current subframe so that the next data acquisition cycle will acquire the data for the first 8 words of the next subframe to be output to the DFDR.

RTPCYC maintains a count of the number of real-time interrupt occurrences (sub-cycles) during a cycle. It is initialized to 8 at the begining of every cycle and is decremented by 1 at each occurrence of the real-time interrupt. Whenever it reaches zero, a new data acquisition cycle is started by RTISR calling RTIDA (see section 5.2.5.2).

The 12-bit data is output to the DFDR via CRU address 5A0(Hex).

Figure 5.39 shows the DFDR output data flow.

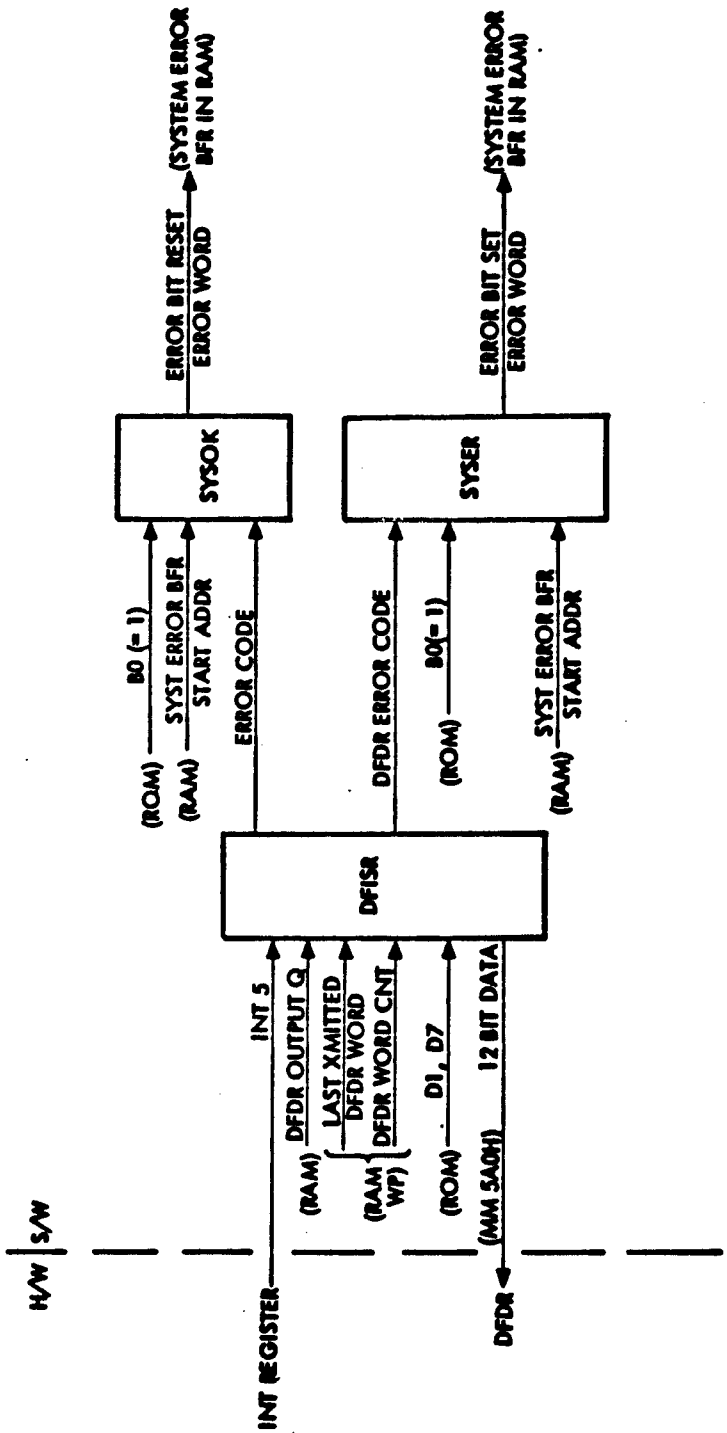


Figure 5.39 DFDR Output Data Flow Diagram

The wraparound register is read from address FF94(Hex) and its contents compared with the previously transmitted word. The DFDR output is double buffered, so that whenever the output register is loaded with new data, the previous data that was there is transferred to the wraparound register from where it can be read. This provides a means of checking the integrity of the DFDR output. The wraparound arrangement is shown in figure 5.40 below.

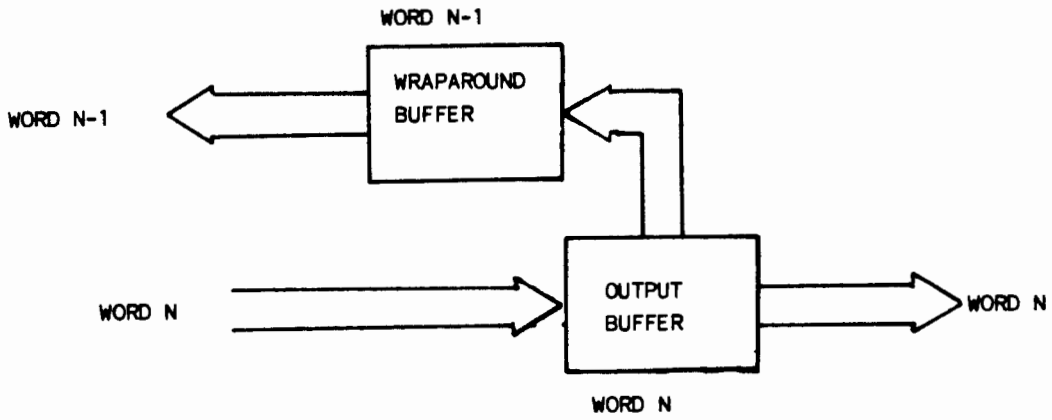


Figure 5.40 DFDR Output Buffer Arrangement

If the data in the wraparound buffer does not match the previously transmitted word, a failure counter, DFAILC is incremented. If more than 256 consecutive failures occur (see section 3.3.3.2.1), the module SYSER is called to set a status bit in the system status buffer, SYEBF. If the data matches, then DFAILC is cleared and the module SYSOK is called to reset the DFDR wraparound status bit in SYEBF.

A further check on the operational status of the DFDR is done by monitoring the DFDR Maintenance Flag discrete. This is done by periodically calling the module DFBTST, which reads the state of that discrete from the discrete multiplexer. If this discrete is set to logic 1, it means that the DFDR is in a non-operative state and hence SYSER is called to set a status bit in SYEBF.

A PDL description of DFISR and DFBTST is given below:

```
DFISR PDL Description
---------------------
IF DF-WORD-COUNT = 55
        RT-PER-FRAME-COUNT = 1
        CYCLE PER FRAME COUNT = 7
ENDIF
SEND THE NEXT DATA ITEM FROM THE DFDR OUTPUT QUEUE
        TO DFDR (INDEXED BY DFWRC)
IF WRAPAROUND DATA WORD NOT EQUAL TO LAST TRANSMITTED WORD
        INCREMENT FAIL COUNTER
        IF FAIL COUNTER >= 255
                CALL SYSER - SET STATUS BIT IN SYEBF
        ENDIF
ELSE
        SET FAIL COUNTER = 0
        CALL SYSOK - RESET STATUS BIT IN SYEBF
ENDIF
SET LAST TRANSMITTED DATA WORD = CURRENT DATA WORD
INCREMENT WORD COUNTER (DFWRC)
IF DFWRC = 64
        SET DFWRC = 0
ENDIF
```

```
DFBTST PDL Description
----------------------
GET DFDR BITE MAINTENANCE FLAG
IF NO DFDR FAILURE
        CALL SYSOK - RESET ERROR BIT IN SYSTEM STATUS BUFFER
ELSE
        CALL SYSER - SET ERROR BIT IN SYSTEM STATUS BUFFER
ENDIF
RETURN
```

5.2.5.9 System Status Generation -

This section describes how the health of the system is continuously monitored, stored and displayed, and how the EAROM memory is used to store a limited history of the system status.

Each subsystem of the DFDAU, (analog, DITS, power supply, etc) periodically performs self tests on its hardware and sets or resets an assigned bit in an assigned word in the system status buffer, SYEBF, by calling the routines SYSER or SYSOK respectively. SYEBF is a 16 word RAM buffer where each bit of the 16 words is used to record the results of a system test or condition. The following table shows the word/bit assignments in SYEBF. If a bit is set to logic 1, it means that the corresponding test has failed, and if it is a logic 0, it means

that the test has passed.

These status bits are converted to status codes which are displayed on the front panel display whenever the "READ" switch is depressed.

This method of having individual bits representing status codes internally can accommodate a total of 256 status conditions in 16 words of memory, a more economical approach than storing each status condition code form in a single memory word.

Table 5.3 below shows the various status code and status bit relationships.

DESCRIPTION	STATUS CODE (HEX)	STATUS BUFFER WORD - BIT	FRONT PANEL LED DISPLAY
DFDR BITE	001	0-0	DFDR FAIL
CPU 1 EPROM SUMCHECK	101	1-0	DFDAU FAIL
DFDR W/A	102	1-1	DFDAU FAIL
POWER SUPPLY BITE	103 - 108	1-2 to 1-7	DFDAU FAIL
ANALOG CALIBRATION	201 - 216	2-0 to 3-6	DFDAU CAUT.
DITS 1 W/A (Chan 0-7)	401 - 408	4-0 to 4-7	DFDAU CAUT.
DITS 2 W/A (Chan 0-7)	409 - 410	4-8 to 4-15	DFDAU CAUT.
CPU 1/CPU 2 Xmit Fail	501	5-0	DFDAU CAUT.
CPU 2 EPROM SUMCHECK	901	9-2	DFDAU CAUT.
CPU 2/CPU 1 Xmit Fail	A01	10-0	None
CPU2 DMEP Fail	A02	10-1	None
CPU2 QAR Fail	A03	10-2	None
CPU2 PRINTER Fail	A04	10-3	None

Table 5.3 System Status Word/Bit Assignments

The first 8 words of SYEBF are reserved for status codes generated in CPU 1 and the second set of 8 words of SYEBF are reserved for the status data received from CPU 2. As can be seen from the table, the status codes are derived from a combination a status buffer word and bit number.

Figure 5.41 shows a structure chart of the System Status G

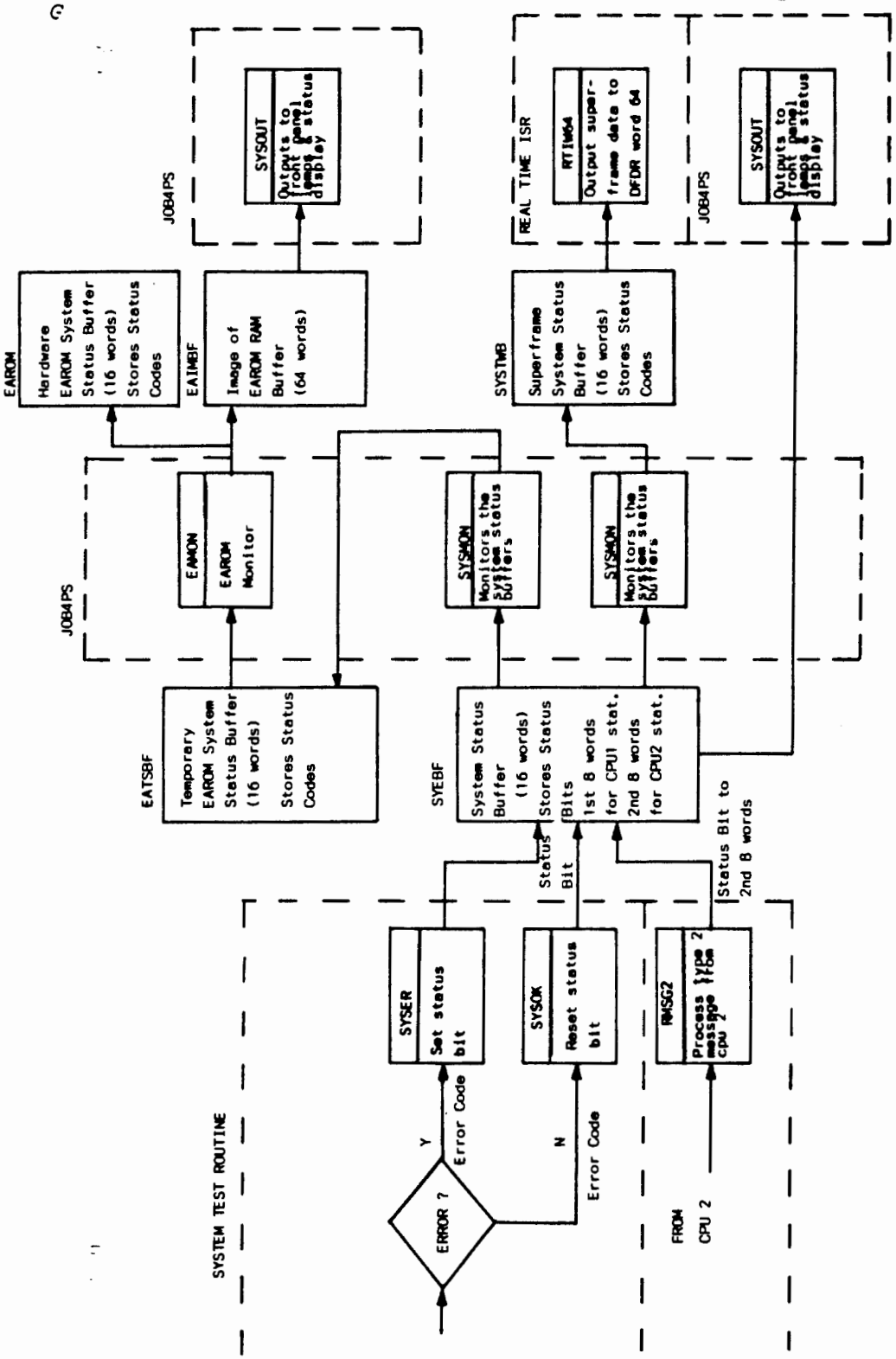


Figure 5.41
System Status Generation Section Structure Chart

When a system test is performed, a status bit is set or reset in SYEBF by SYSER or SYSOK respectively. The module SYSMON is periodically called by the executive to check if any new status bits have been set in SYEBF. If so, the module SYBTOC is called to convert the status bit to the corresponding status code. This status code is then written into the temporary EAROM system status buffer, EATSBF, as well as the superframe system status buffer, SYSTWB, which in turn is sent to word 64, subframe 2 of the DFDR (see section 3.3.3.2).

A PDL description of SYSMON, and SYBTOC is given below:

```
SYSMON PDL Description
----------------------
SAVE LINK ADDRESS
SELECT NOT NEW SUPERFRAME FUNCTION (SET JUMP REGISTER = 2)
IF NEW SUPERFRAME
        CLEAR SUPERFRAME SYSTEM STATUS BUFFER
        SET JUMP REGISTER = 0
ENDIF
SET WORD COUNTER = 0 FOR SYBTOC ROUTINE
CLEAR BIT COUNTER FOR SYBTOC ROUTINE
INITIALIZE ERROR COUNTER (SET = 0)
DO UNTIL ERROR ERROR COUNTER = 16 (16 STATUS CODES GENERATED)
        CALL SYBTOC TO CONVERT ERROR BIT TO CODE
        IF STATUS CODE IN SYSTEM STATUS BUFFER
                IF JUMP REGISTER (R3) = 0
                  STORE STATUS CODE IN SUPERFRAME
                  SYST. STAT. BUFF. INDEX BY ERROR COUNTER
                ELSE
                  IF STATUS CODE NOT IN SUPERFRAME BUFF
                     STORE STATUS CODE IN AVAILABLE
                     WORD OF SUPERFRAME SYSTEM STAT.
                     BUFFER WHICH MAY BE OUTPUT IN
                     THE NEXT SUPERFRAME STAT. DATA CYCLE
                  ENDIF
                ENDIF
        ENDIF
ENDDO
```

```
SYBTOC PDL Description
----------------------
IF NOT FIRST CALL
        GOTO IF ERROR BIT
ENDIF
DO UNTIL 16 ERROR WORDS HAVE BEEN CHECKED
        IF ERROR TYPE = ERROR FOR THIS WORD
                DO UNTIL 16 BITS CHECKED
                   IF ERROR BIT SET IN THIS STATUS WORD
                        COMBINE BIT COUNTER AND ERROR
                        TYPE CODE (USING WORD COUNTER
                        AS INDEX) FROM ERROR TYPE TABLE
                        GOT TO RETURN
                   ENDIF
                ENDDO
        ENDIF
        CLEAR BIT COUNTER
ENDDO
CLEAR ERROR FLAG
RETURN
```

At the same time, EAMON, the EAROM monitor routine is periodically called by the executive. This routine controls any operations to be performed on the EAROM. Upon entry, EAMON checks whether the EAROM clear request flag, CLRFLG, has been set. If so, it simply writes zeroes into the 64 words of the EAROM and exits.

If CLRFLG has not been set, it checks whether any new status codes have been written to EATSBF by SYSMON. If so, it copies this status code into the EAROM and into the EAROM Image Buffer EAIMBF, and exits. A maximum of 16 status codes are written into the EAROM.

The EAROM clearing is a maintenance function, and is never used when the DFDAU is in the operational mode.

Although the EAROM has a capacity of 64 16-bit words, only the first 16 words are currently used. Any status codes that are generated after this are ignored. This situation is however very unlikely under normal operating conditions. When this status code has been written into the EAROM, the corresponding error code in EATSBF is cleared, so that at any time EATSBF hold status codes that have not yet been transferred to the EAROM.

The operation of actually writing data to the EAROM is performed by EAWRIT. EAIMBF is a 64 word RAM buffer containing an exact image of the data currently in the EAROM.

A PDL description of EAMON and EAWRIT is shown below:

```
EAMON PDL Description
---------------------
IF EAMODE = 0 (EAROM NOT BUSY)
  IF EAROM CLEAR FLAG = -1
        DO UNTIL 16 WORD OF EAROM HAVE BEEN WRITTEN INTO
              GET EAROM ADDRESS
              ZERO EAROM DATA ITEM
        ENDO
        CLEAR EAROM CLEAR FLAG
        CLEAR EAROM ADDRESS REGISTER
  ENDIF
  DO UNTIL 16 WORDS OF TEMP SYSTEM STATUS BUFFER HAVE BEEN
                        SEARCHED FOR ZERO CODE
          IF 16 STATUS CODES HAVE NOT BEEN WRITTEN TO EAROM
              GET SYST STATUS EAROM ADDRESS
              CALCULATE EAROM ADDRESS
              CALL EAWRIT - WRITE TO EAROM
          ENDIF
        ENDIF
ELSE
  IF 250 MSEC HAVE ELAPSED
        IF EAMODE = 0 (EAROM NOT BUSY)
          TURN BIT C2 OFF IN IMAGE AND H/W CONTROL REGISTER
          SET TIMER TO PRESENT + 250 MSEC
          SET TO WRITE MODE
        ELSE
          CLEAR IMAGE AND H/W CONTROL REGISTERS
          STORE DATA IN IMAGE OF EAROM BUFFER
          CLEAR DATA
          SET EAROM MODE TO INACTIVE
          IF DOC DATA FLAG = 0
              INCREMENT SYST STATUS EAROM ADDRESS
          ELSE
              CLEAR DOC DATA FLAG
          ENDIF
        ENDIF
  ENDIF
ENDIF
RETURN
```

```
EAWRIT PDL Description
----------------------
IF EAROM MODE = 0 (NOT ACTIVE)
   GET EAROM ADDRESS
   MOVE EAROM ADDRESS TO EAROM H/W ADDRESS REGISTER (FFAE)
   GET EAROM DATA POINTER
   MOVE EAROM DATA TO EAROM H/W DATA REGISTER (FFAO)
   IF ADDRESS IN LOWER HALF (WORDS 0 - 31)
        SET CHIP SELECT 1 TO ON
   ELSE
        SET CHIP SELECT 2 TO ON
   ENDIF
   PLACE CONTROL BITS INTO H/W CONTROL REGISTER
   SET EAROM MODE TO ERASE
   WAIT 250 MSEC
ENDIF
RETURN
```

Status output to the front panel is controlled by the module SYSOUT. This module is periodically called by the executive to perform the DFDAU front panel display tasks. The front panel consists of a "READ" switch, a 3-digit display and 3 status LED'S namely "DFDAU FAIL", "DFDAU CAUTION" and "DFDR FAIL". The status history of the DFDAU can be viewed at any time by depressing and holding down the "READ" switch. While the "READ" switch is being depressed, the 16 status codes currently stored in EAIMBF (and hence the EAROM) are displayed on the 3 digit display for 4 seconds each. It should be noted that because of the high access times of the EAROM memory (64msec for a read operation) the status codes are displayed from the image buffer, EAIMBF, instead of the actual EAROM memory.

The "DFDAU FAIL" and "DFDAU CAUTION" LED'S are used to show current status conditions. That is, they are turned only on when a corresponding fault condition is current.

Two modules are called by SYSOUT. One them, SYLMP, is called unconditionally to check through the buffer SYEBF and turn the "DFDAU CAUTION" and "DFDAU FAIL" lamps accordingly. The other, SYDSP, to display the status history from the EAROM image buffer upon request (when the READ switch is depressed).

CHAPTER 6

SOFTWARE DEVELOPMENT INTEGRATION AND TESTING

6.1 INTRODUCTION

This section describes the configuration of the software development, integration and test environment for the DFDAU. It describes the steps and procedures used to develop the software from the coding stages, through the testing phase, up to the final version that is programmed into the EPROM memory. It does not however cover the phase of the project that deals with the hardware module testing, as these are assumed to be fully operational at the time of the start of the software testing phase.

However, with a project such as the DFDAU, it is inevitable that certain problems directly related to the hardware will surface during the software testing phase. These are generally resolved in a parallel fashion.

6.2 GENERAL DESCRIPTION

The following block diagrams show the hardware and software used in the development integration and test phases of the DFDAU software.

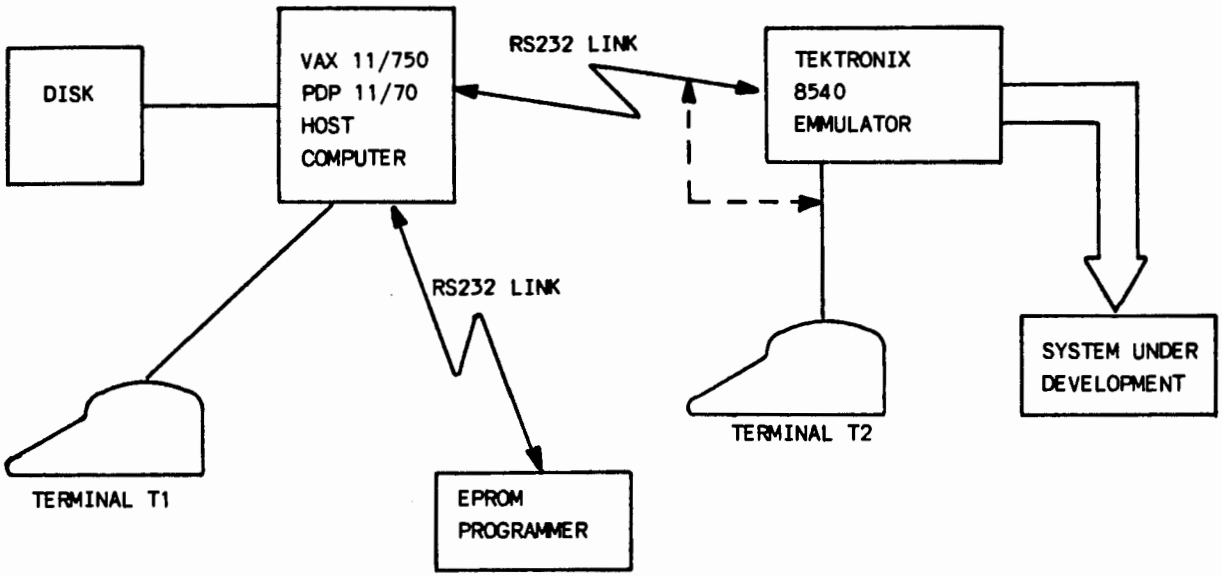


Figure 6.37(a) Development Environment Hardware

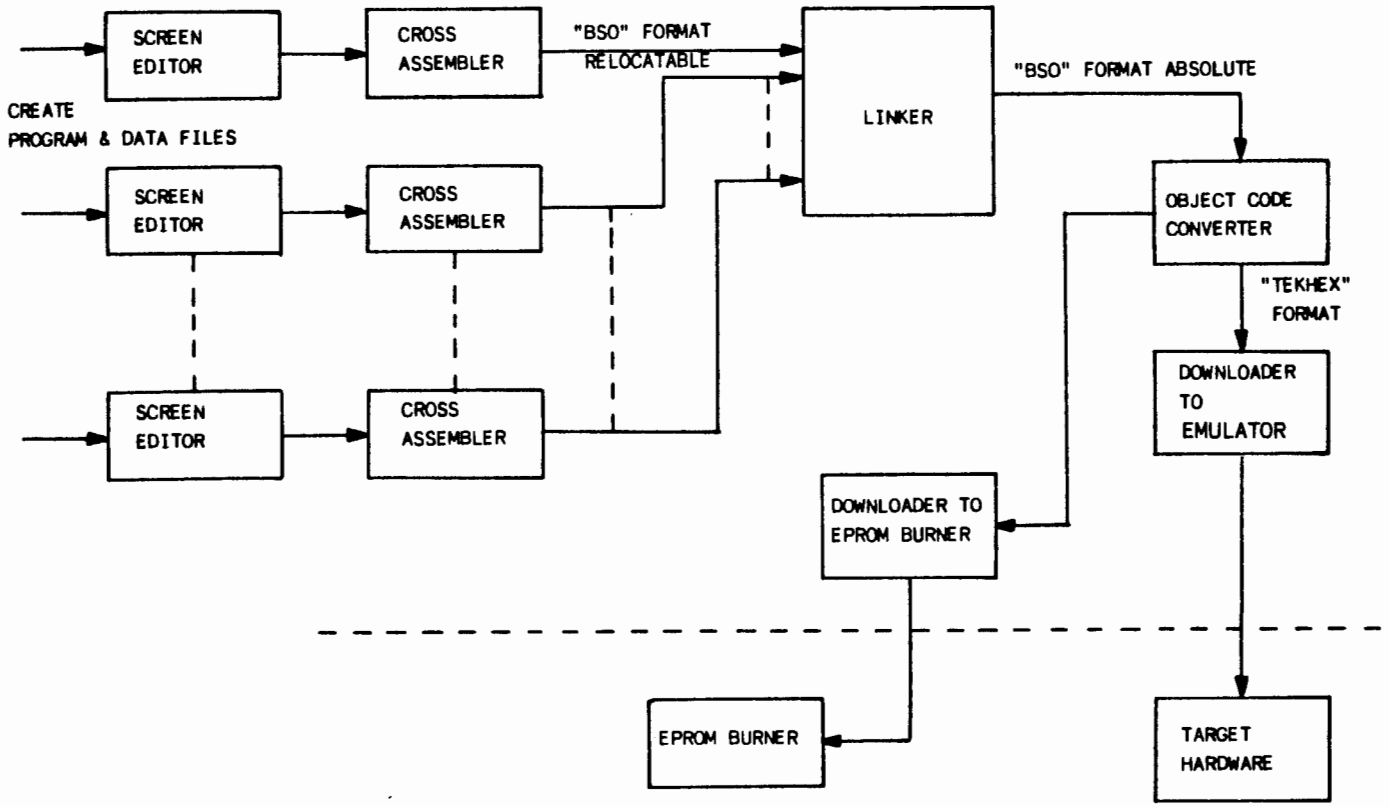


Figure 6.37(b) Development Environment Software

6.3 SOFTWARE DEVELOPMENT ENVIRONMENT

In general, program and data files are entered into the host computer (DEC PDP 11/70 or VAX 11/750) via a text editor and stored on disk. The programs are written in TMS9900 assembly language.

These program and data files are then assembled via a cross-assembler, and a relocatable version in "BSO" (Boston Systems Office) internal format is created. Boston Systems Office is the supplier of the development software used for the

DFDAU software development (cross-assembler, linker and code-converter).

When all files have been assembled and all assembly time errors have been resolved, the relocatable modules are linked via a cross-linkage editor to form the absolute module,still in "BSO" format. During this process all the global symbols are allocated actual (absolute) addresses, and as a result a map file is created, listing the absolute addresses of all global symbols.

This absolute module is still not executable by the TMS9900. It is now passed through a code-converter which converts it to one of the industry standard absolute code formats (eg.Tekhex, Extended Tekhex, Motorola S-Records, etc. See Reference 11). The format used for this project is Tekhex, chosen simply for the reason that all the in-house development software (downloaders) were originally developed for the Tekhex format.

The converted object code is passed through a host computer resident process to calculate the the sumcheck words for the object code file. These sumcheck words are then converted into 2's complement (negative) values, formed into a Tekhex record with a target address immediately following the highest address in the object code, and included in the object code file.

This new object code file is downloaded to the emulator (Tektronix 8540), from where it can be tested on the target hardware. Here, all hardware bugs are also resolved. When all software and hardware bugs have been cleared out and the software is fully operational, the same object module is downloaded from the host computer to an EPROM programmer, where it is burned onto the EPROMS. These EPROMS are then installed in the target hardware, and the system is complete. Figures 6.38(a) and 6.38(b) show the lifecycle of software development.

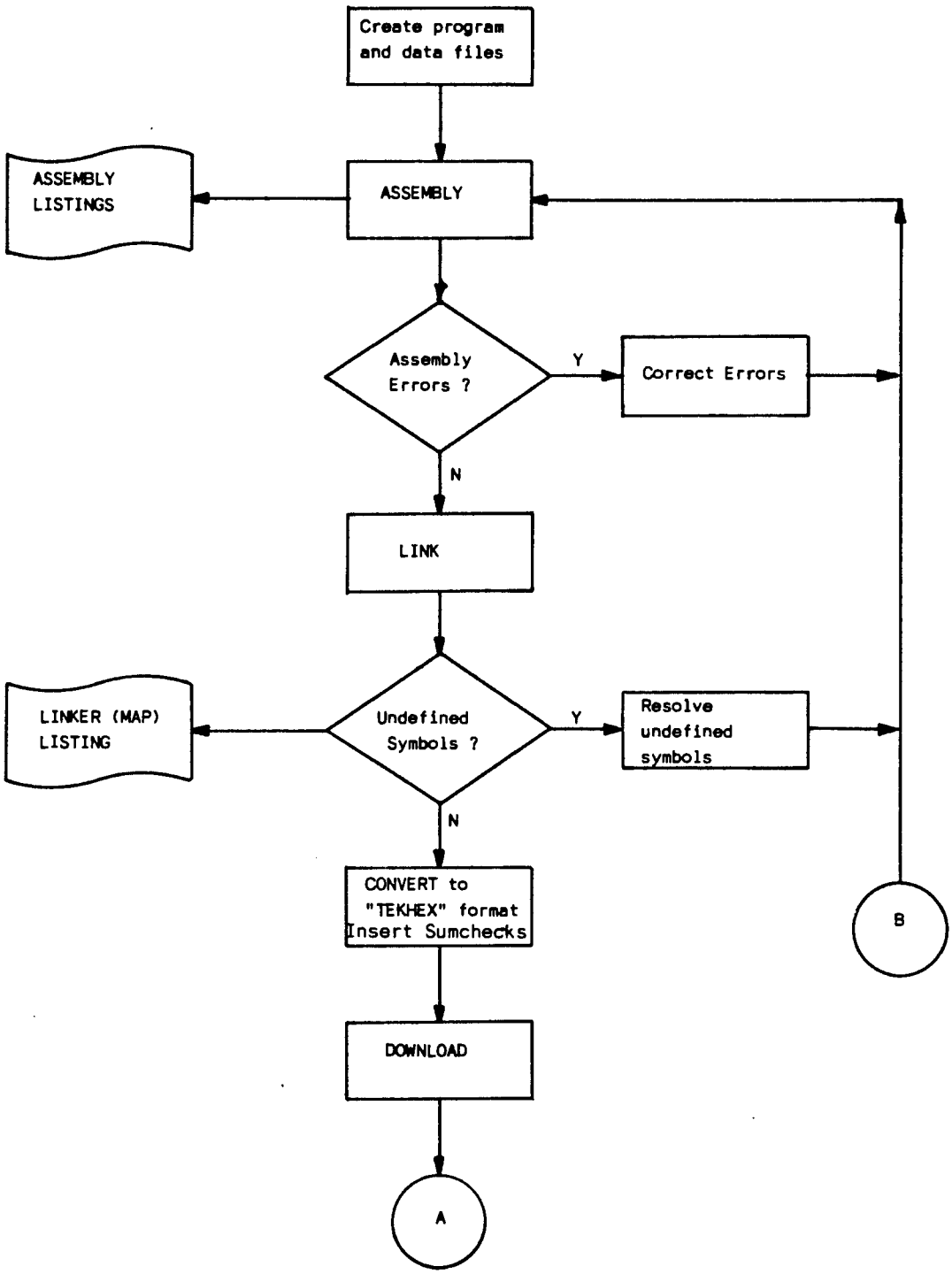


Figure 6.38(a) Lifecycle of Software Development

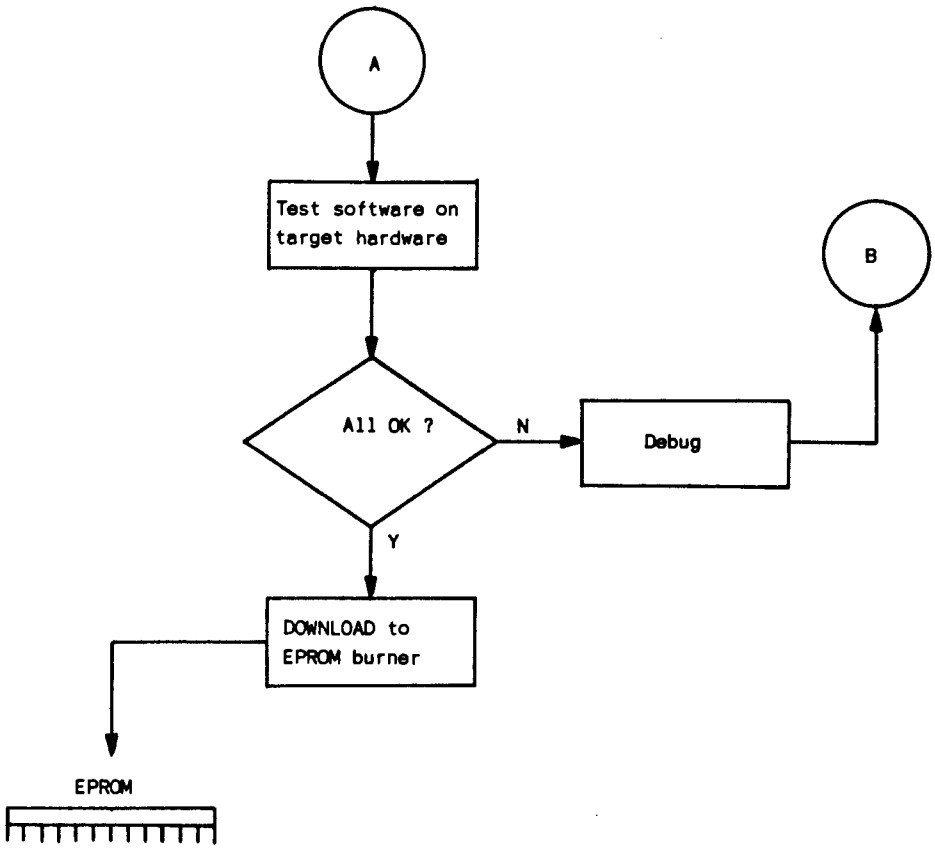


Figure 6.38(b) Lifecycle of Software Development

A summary of software tools:

Host operating systems: RSX 11/M+,	*VAX/VMS 4.2*
Cross-Assembler	*Commercially available*
Linker	*Marketed by*
Code Converter	*Boston Systems Office*
Emulator and	*In-house application*
EPROM burner downloaders	*programs.*

6.4 TARGET HARDWARE AND SOFTWARE TESTING ENVIRONMENT

To test the hardware with the operational software, various types of equipment are needed to provide output monitoring, signal

input and power sypplies. Figure 6.39 shows the actual hardware test configuration arrangement

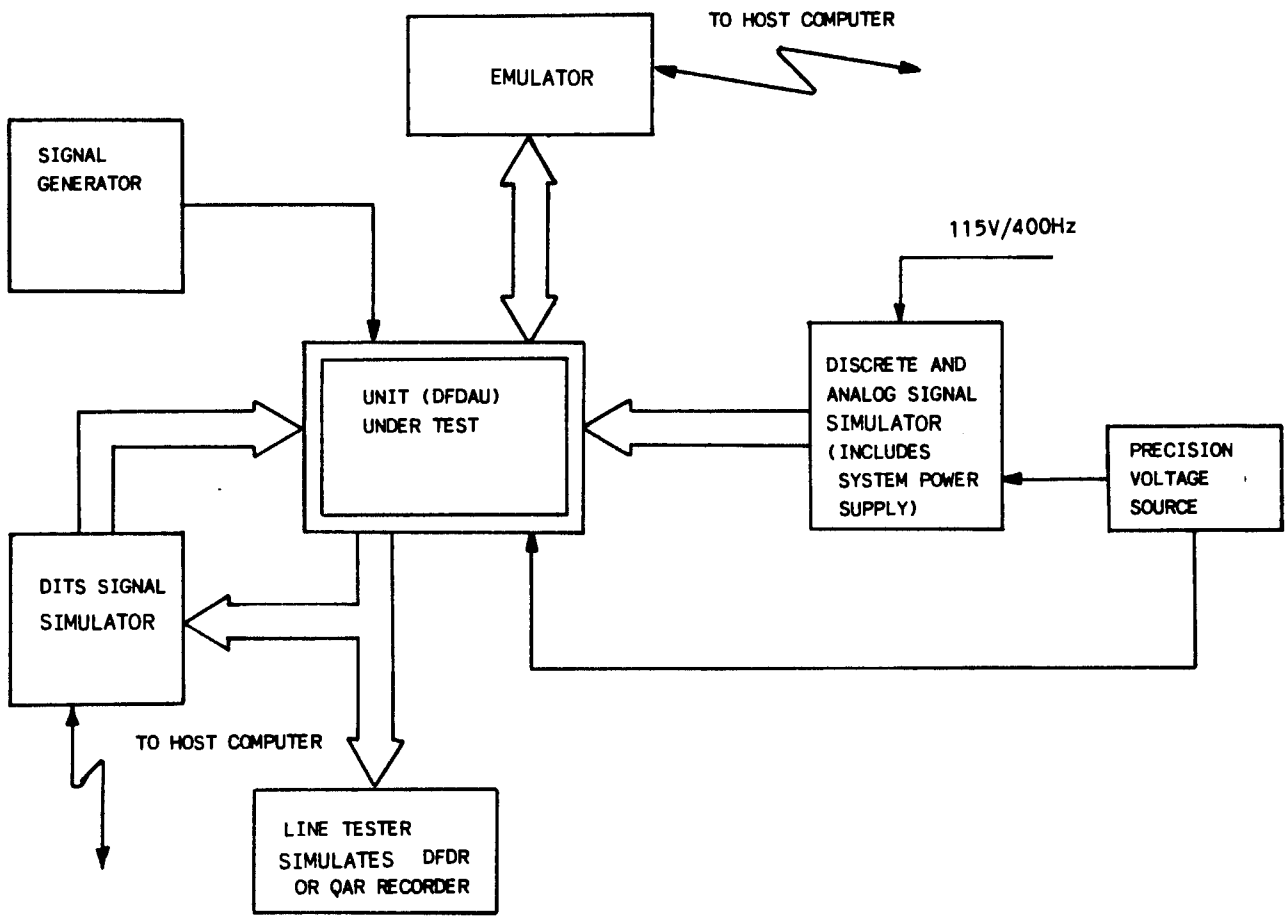


Figure 6.39 Hardware Test Environment

The emulator provides all the functions of the 9900 processor while the system is under test. When the emulator is plugged into the unit, the actual processor chip on the CPU module in the DFDAU goes into a high impedence state, and hence does not interfere with the operation of the emulator.

The Line Tester is a display instrument which simulates a DFDR. It has a keyboard on the front panel that enables the selection of subframe and word number to be displayed. The display is a 4 character display switchable between octal and hexadecimal. It

also has an indicator light on the front panel to show if the DFDR output is synchronized or not. That is, it receives a synchronizing code one per second, every 64th word, in the sequence of:

1107
2670
5107 Octal
6670

This is explained in section 2.3.

6.4.1 Input Parameter Simulation And DFDR Data Frame Generation

By using the various input signal simulators, a DFDR data frame can be generated, output to the DFDR output of the DFDAU and monitored on the Line Tester. The signal simulators are fully controllable and by simultaneously observing the selected DFDR output word on the line tester and varying the corresponding input signal simulator, the DFDAU output can be checked for accuracy and correct formatting.

The DITS simulator provides the 32-bit digital input test signals based on the ARINC 429 specifications (section 3.4, reference 2). A test file is set up in the host computer containing the labels, SDI codes, input channel number and the formatting information for all the DITS parameters to be acquired by that particular system configuration of the DFDAU (appendix C). This file is then downloaded to the DITS simulator which uses it to generate the necessary signals in the correct 32-bit ARINC 429 format.

The simulator sets up a 32-bit data word containing the necessary label, data field, sign and parity bits for each DITS parameter, and transmits it to the DFDAU in bipolar RZ format at the required rate (12 or 14.5kbps). The simulator is also connected to the DFDR output of the DFDAU, and after transmitting the DITS data word, it checks the DFDAU output to the DFDR for correct formatting and correct DFDR data stream slot for that parameter.

Each DITS parameter is transmitted to the DFDAU a number of times, each time with a randomly different data field. During the process, an error log file is generated by the simulator indicating which parameters failed the test and where the failure occurred (eg. incorrect parity, missing sign bit, incorrect rounding etc). This file can then be downloaded to the host computer and used to modify the DITS data base or the software accordingly.

The analog and discrete signal simulator provides all the discrete and analog signals using a precision voltage source as its voltage reference. It also provides power to the unit under test (115V, 400Hz).

The discrete signals are a set of on/off switches (one for each discrete channel), each connected to the appropriate signal source, ie. shunt, series, ac marker beacon or ident discretes (section 4.6.4).

The analog test input signals are fed to the DFDAU via a patch panel that connects the appropriate signal type to the corresponding DFDAU analog channel. For example, DFDAU analog channel 19 is connected to the low level DC (LLDC) output of the analog simulator to provide the stimulus for the Flap Handle Position parameter (table C-1, appendix C). The signal is then varied and the output on the Line Tester is observed to change accordingly.

The signal generator provides the tachometer test signal (square wave variable between 0.7 and 70 HZ) for DFDAU configurations incorporating a tachometer interface.

In this way, test stimuli for all the required parameters can be generated, and the DFDR data frames created and checked as per appendix D.

6.4.2 System Status Simulation

In a similar way, the system status generation and display is checked by generating various error conditions described in the following paragraphs.

A sumcheck failure can be simulated by modifying the program memory while the system is running under emulator control, using the emulator memory to store the operational program instead of emulating directly from the target system program memory (EPROM). When the background tasks now calculate the program sumcheck words, it will not match the pre-calculated values embedded in the downloaded object file and thus generating a sumcheck status condition.

Wraparound errors can be generated by temporarily breaking the wraparound connections on the appropriate hardware module (DITS and DFDR output). When the wraparound registers are now read by the software, their contents will not match the expected values, and therefore resulting in a wraparound error.

In a similar way, analog calibration and power supply BITE failures can be generated by disconnecting the various supply and calibration voltages from the corresponding analog multiplexer channels. When a limit check is performed on these signals, the test will fail, thus causing a status condition to be set.

Inter-CPU transmission errors are generated by grounding the inter-CPU interrupt line. This prevents an interrupt from being generated when a transmit request is made, resulting in a status condition.

The power-down recovery feature of the system is tested by connecting an impulse switch to the power connection of the DFDAU. When this switch is depressed, it momentarily disconnects the supply voltage from the DFDAU for a pre-setable period. The switch is set to 200ms and is depressed while the frame counter is observed on the Flight Line Tester. When the switch is depressed, a power loss of 200ms is generated, and for correct operation on the DFDAU, the frame counter should continue to count as if no power interruption had occurred.

CHAPTER 7

CONCLUSION

The basic version of the system described here was completed towards the end of 1982. Since then, a number of versions of the system have evolved, mainly due to improvement ideas and specific airline requirements, to include such features as touch panel interface, air-to ground communications monitoring and automatic weather update.

One of the major hazards of flying has been the unpredictability of wind-shear conditions which can be extremely dangerous to aircraft during their departure and approach phases. A number of warning systems are currently being developed to be able to predict the existence or occurrence of such conditions. The high speed data acquisition and recording capabilities of the DFDAU have been invaluable in recording the fast changes in acceleration and wind speed and direction that occur under wind-shear conditions, and as a result, bringing about some sort of "predictability factor" to the situation.

The limiting factors of the system were found to be its limited addressable memory (32K, 15 address lines), and its operating speed (3MHz), both of which only became evident as the system applications began to expand beyond those originally discussed.

An updated design of the system is currently being proposed based on the Motorola M6C8020 microprocessor. The MC68020 is a full 32-bit processor with a non-multiplexed 32-bit data and address bus, capable of addressing 4 Gigabytes of memory, and a processor speed of 12 or 16MHz. Another important feature of this processor is that it has a co-processor interface to the MC68881, which is a full IEEE floating point processor.

APPENDIX A

ARINC SYSTEM SPECIFICATIONS

Aeronautical Radio Incorporated (ARINC) is a corporation in the United States in which scheduled airlines are the principal stockholders. Other stockholders include a variety of other air transport companies, aircraft manufacturers and foreign (non-U.S.) airlines.

Activities of ARINC include the operation of an extensive system of U.S. and overseas aeronautical land radio stations, the fulfillment of system requirements to accomplish ground and airborne compatibility, the allocation and assignment of frequencies to meet those needs, the coordination incident to standard airborne communications and electronic systems and the exchange of technical information.

ARINC sponsors the Airlines Electronic Engineering Committee (AEEC), composed of airline technical personnel. The AEEC formulates standards for electronic equipment and systems for the airlines. The establishment of Equipment Characteristics is a principal function of the Committee.

An ARINC Report (Specifications or Characteristic) has a twofold purpose which is:

1. *To indicate to the prospective manufacturer of airline electronic equipment the considered opinion of the airline technical people coordinated on an industry basis concerning requisites of new equipment.*

2. *To channel new equipment designs in a direction which can result in the maximum possible standardization of those physical and electrical characteristics which influence the interchangeability of equipment without seriously hampering engineering initiative.*

APPENDIX B

TMS9900 ARCHITECTURE AND INSTRUCTION SET

B.1 TMS9900 ARCHITECTURE

The 9900 system is illustrated in figure B-1.

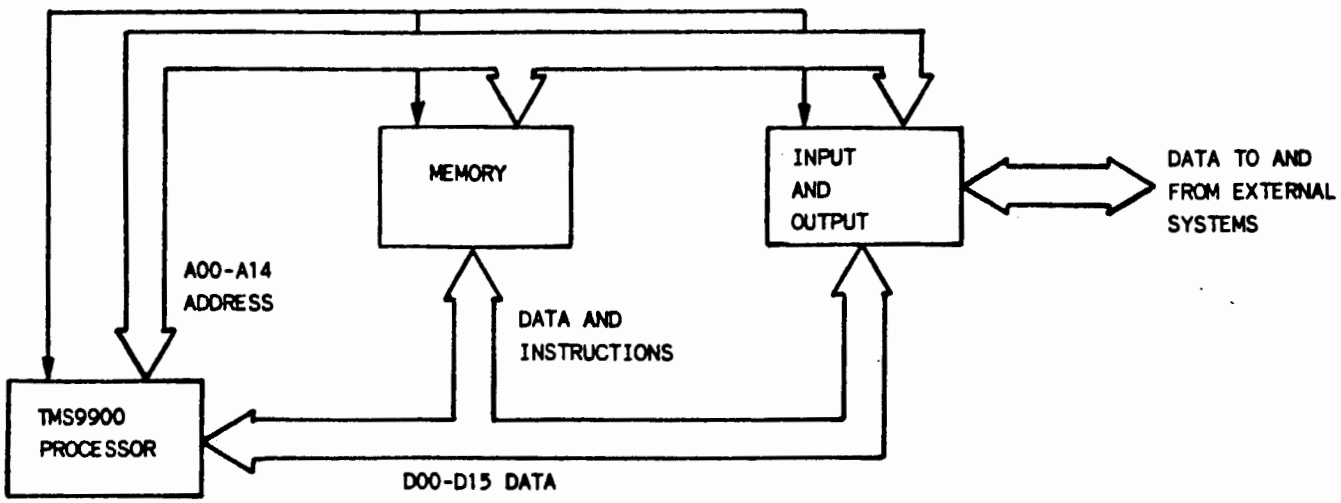


Figure B-1 General 9900 System Architecture

The major subsystems are the 9900 processor, memory for the program and data storage, and input and output devices for external communication and control.

Input data, as previously described, is in the form of analog, discrete and digital signals, while the output is digital. All the data I/O is handled through the 9900 system's three I/O modes, namely DMA, memory mapped and communications register unit (CRU).

B.1.1 The Processor

The processor controls the fetching of data and instructions from memory or input devices and the transferring of data. The instructions are transferred in 16-bit words, and these are addressed by the 15 address lines A1 - A15. This means that a total of 32,768 (32K) words of memory can be addressed with the 15 address lines.

Internally, the processor generates a 16-bit address, but the least significant bit, A0, is not sent to the memory. Each word is further broken down into two bytes (8 bits each) as shown in figure B-2

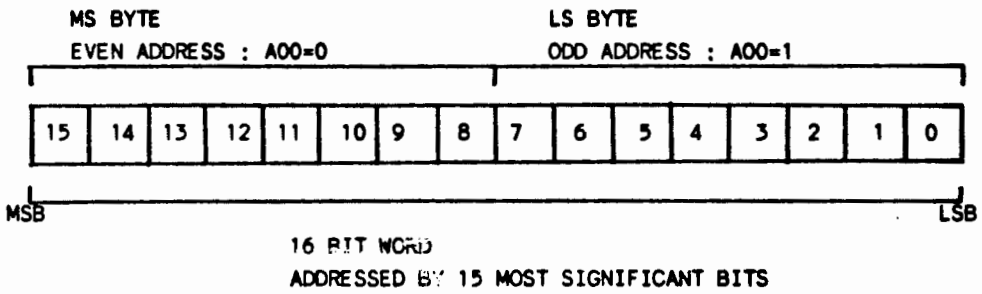


Figure B-2 Words and Bytes

The first byte is located at an even address (A0=0). The second byte is located at an odd address (A0=1). Byte selection is done internally in the processor once the full 16-bit data word is obtained from one of the 32K word locations. Byte addressing is used only on instructions that use byte operations.

The processor contains certain basic elements as shown in figure B-3

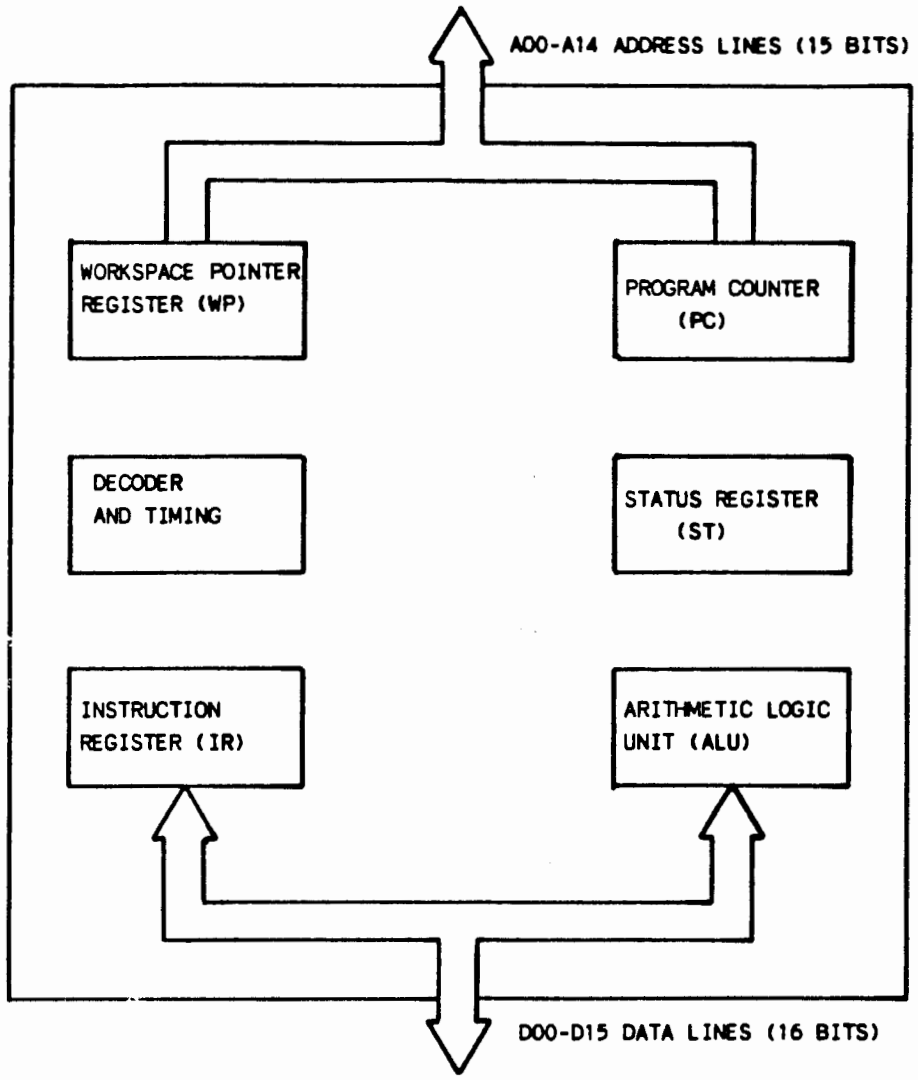


Figure B-3 Basic 9900 Elements

The blocks that are of interest from a software point of view, are the Arithmetic Logic Unit (ALU), Status Register (ST), Workspace Pointer (WP) and the Program Counter (PC).

B.1.1.1 Program Counter (PC) -

The PC contains the address of the next instruction to be executed. After execution of an instruction, the contents of the PC are incremented so that they point to the next instruction to

be executed in the sequence. However, this can be controlled by the software, by means of a branch or a jump instruction.

B.1.1.2 Status Register (ST) -

This register stores the general logic and arithmetic conditions that result from the execution of each instruction. It contains information on whether the last operation caused a result greater than or equal to some reference value, information regarding the sign of the result (positive or negative), and carry or overflow information. The ST also contains a 4-bit code known as the interrupt mask, which defines which of the 16 interrupts will be recognized and responded to by the processor.

The ST is shown in figure B-4 below.

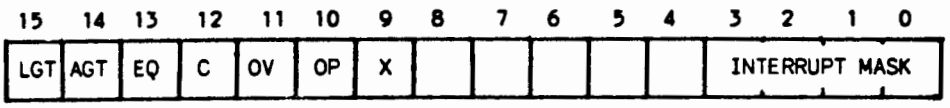


Figure B-4 TMS9900 Status Register

LGT	-	Logical Greater Than
AGT	-	Arithmetic Greater Than
EQ	-	Equal
C	-	Carry
OV	-	Overflow
OP	-	Odd Parity
X	-	Extended Operation

B.1.1.3 Workspace Pointer (WP) -

This register addresses the first word in a group of 16 consecutive memory words called a workspace as shown in figure B-5 below.

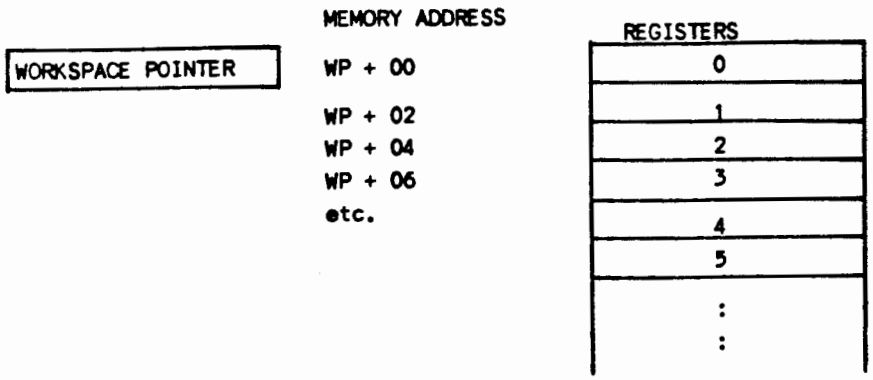


Figure B-5 TMS9900 Workspace Structure

Some of the workspace registers are reserved for specific tasks as shown above. When a certain type of subroutine branch, called the Branch And Link (BL), register 11 is used to save the contents of the Program Counter at the time of the branch. In another type of subroutine branch, called the Branch And Link Workspace (BLWP), registers 13, 14 and 15 are used to save the values of the WP, PC, and ST registers respectively that were in the processor at the time the branch instruction occurred. This then allows a return to the situation that existed prior to the branch.

Register 12 is used for the base address of the communications register unit (CRU) subsystem.

B.1.1.4 Program Environment (Context) -

The contents of the ST, PC and WP completely define the status of the system program at any given time.

As illustrated in figure B-6, the PC keeps track of the part of the program currently being executed, the ST keeps track of the logical and arithmetic conditions that result from the execution of each instruction, and the WP keeps track of the location in memory of the 16 general purpose workspace registers currently being used by the program.

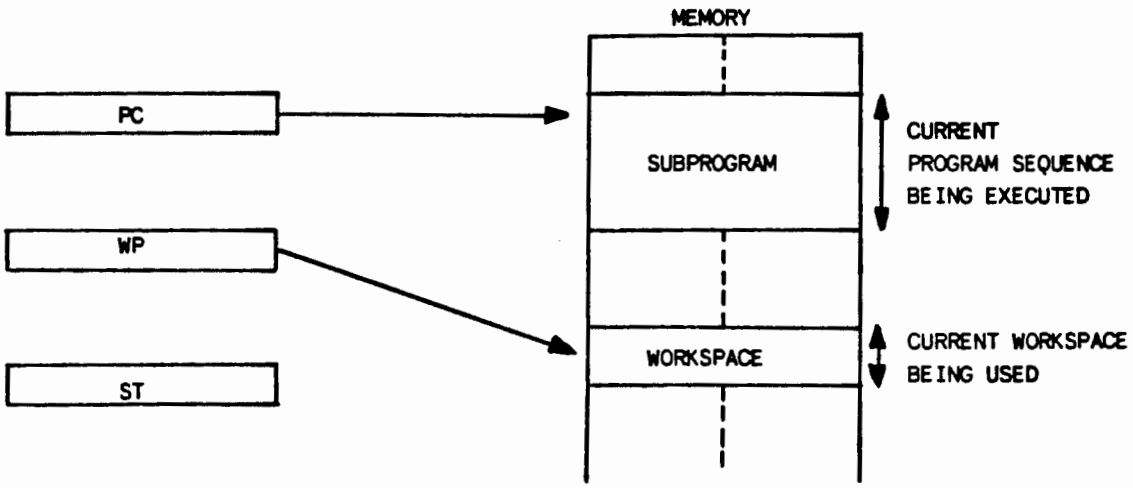


Figure B-6 Program Context

A change in the contents of these registers will change the environment or context of the program and similarly by storing the ST, WP and PC, the program environment will be switched to the original context.

B.1.2 Memory Organization

The TMS9900 memory word is 16 bits long, and with the 15-bit address bus, 32K words can be addressed. The memory is divided into three sections as shown in figure B-7.

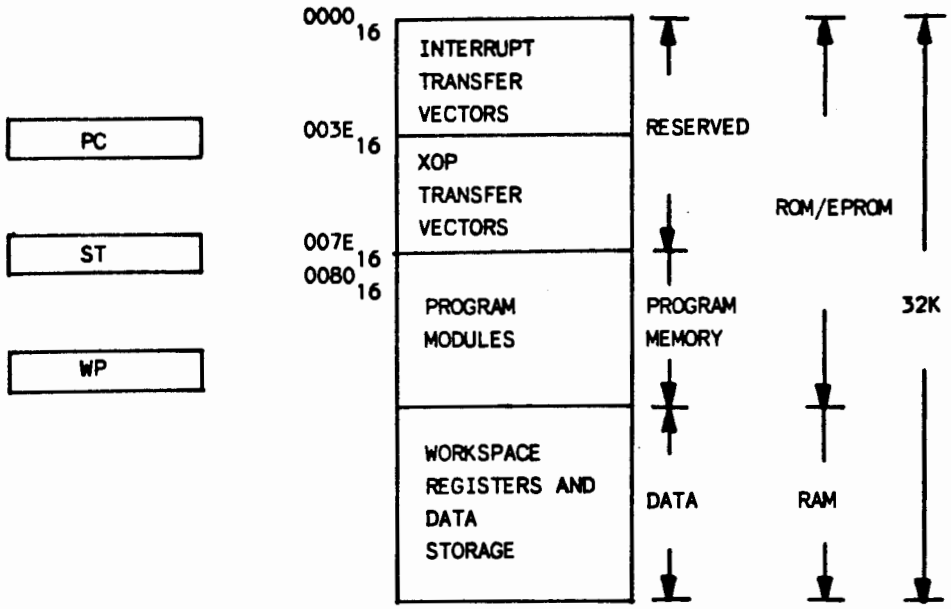


Figure B-7 Memory Organization

B.1.2.1 Reset Vectors -

The first two words are reserved for the RESET vector. The RESET vector is used to load the new WP and PC whenever the cpu RESET signal occurs. The first word contains the WP which is the starting address of the RESET workspace. The second word contains the new PC which is the starting address of the RESET service routine.

B.1.2.2 Interrupt Vectors -

The next 30 words of memory 0004(Hex) to 003E(Hex) are reserved for the storage of interrupt transfer vectors for levels 1 through 15. Each interrupt level uses a word for the WP and a word for the starting address of the interrupt service routine.

B.1.2.3 Extended Operations (XOP) Vectors -

The next thirty two words, 0040(Hex) to 007E(Hex), are used for the WP and PC of 16 Extended Operation (XOP) instructions.

B.1.3 Interrupts

In general, the 9900 performs its functions under the control of a 4-phase clock. The instruction cycle consists of the following basic operations:

a. Instruction fetch
b. Instruction execute
c. Increment contents of the PC to the address of the next instruction.

After all the steps have been completed, the processor checks to see if interrupts are pending. If not, it continues with the next instruction. However, if an interrupt were present, the processor would fetch the appropriate interrupt vector and execute the corresponding interrupt service routine.

When this occurs, the old PC, WP and ST are saved in registers 13, 14 and 15 of the new workspace and bits 12 to 15 of the ST are set to a value 1 less than the received interrupt level, thus disabling interrupts of lower priority. At the end of the interrupt service routine, the old PC, ST and WP are restored.

B.1.4 Address Modes

There are 8 addressing modes supported by the 9900 processor.

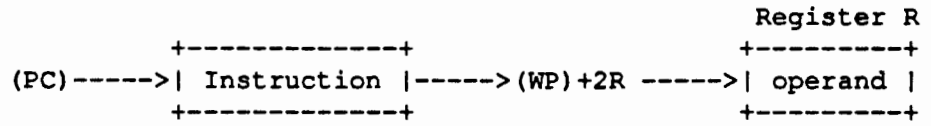
B.1.4.1 Workspace Register Addressing -

The data or address to be used by the instruction is contained in the specified workspace register.

```
e.g     DEC     R2
```

decrements the contents of workspace register 2 (R2) by one.

The address is calculated as follows:



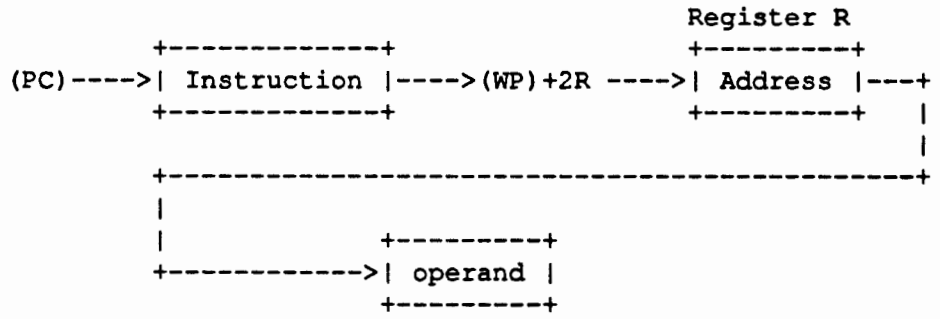
B.1.4.2 Workspace Register Indirect Addressing -

The operand address is contained in the workspace register specified in the instruction and preceded by an asterisk (*).

*e.g CLR *R3*

clears the contents of the data word addressed by the contents of workspace register 3 (R3). The contents of R3 remain unaltered.

The operand address is computed as follows:



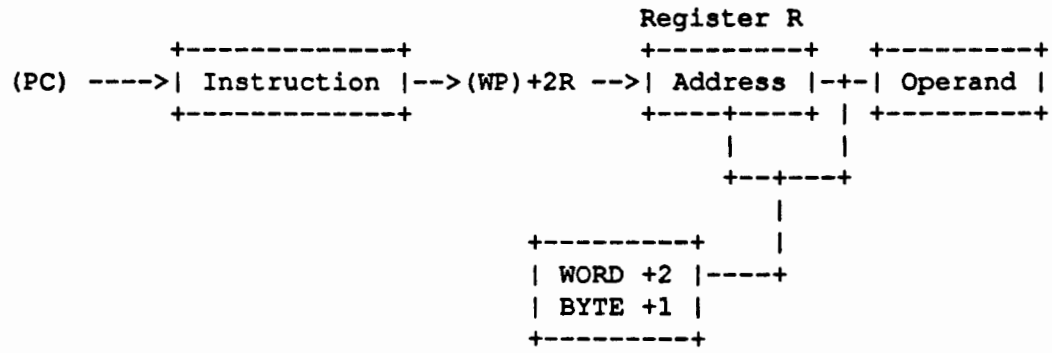
B.1.4.3 Workspace Register Indirect With Auto-increment -

This mode is the same as the previous one except that upon completion of the instruction, the contents of the address register are incremented by 2 if a word instruction had been executed or by 1 if a byte instruction had been performed.

*e.g CLR *R3+*

clears the contents of the data word addressed by the contents of R3, and the contents of R3 are then incremented by 2. The plus (+) sign indicates auto-increment.

The following computations occur with this kind of instruction:



B.1.4.4 Symbolic Or Direct Addressing -

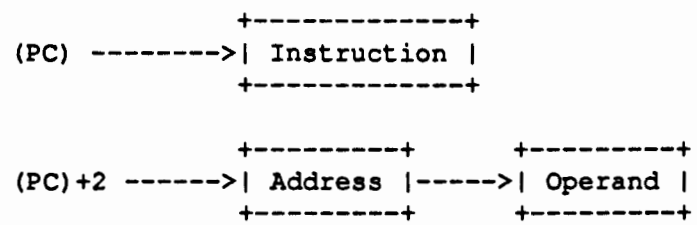
The address of the memory word is contained in the operand field, and is contained in the program memory in the word immediately following the instruction. As an example, to clear the memory word at address 1000(HEX) the following could be used:

```
CLR     @>1000
```

The "@" indicates direct addressing and the ">" indicates a hexadecimal constant. Alternately, a symbolic name (label) can be used:

```
CLR     @LABEL
```

where the LABEL is later equated to 1000HEX. The address computations are as follows:



B.1.4.5 Indexed Addressing -

This mode of addressing is a combination of symbolic and register

indirect addressing. Registers 1 to 15 are used as index registers. The memory word address is obtained by adding the contents of the index register specified to the constant contained in the instruction eg.

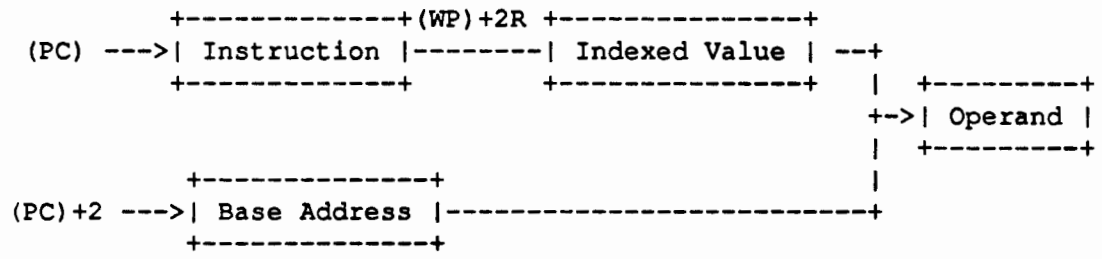
```
CLR     @4(R5)
```

clears the memory word at the address obtained by adding 4 to the contents of register 5.

Alternately, a symbolic name could be used eg.

```
CLR     @LABEL(R5)
```

The address, in general, is calculated as follows:

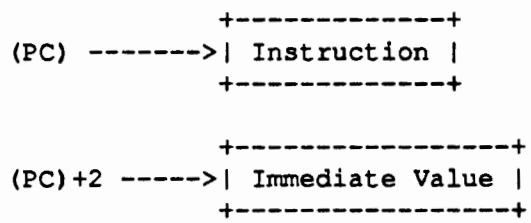


B.1.4.6 Immediate Addressing -

In this case, the instruction contains the actual operand to be used eg.

```
LI      R1,5
```

loads register R1 with the value 5. This can be illustrated as follows:



B.1.4.7 Program Counter Relative Addressing -

These are conditional or unconditional jump instructions. The instruction includes a signed displacement value between -128 and +127, eg.

```
JUMP    LOOP
```

Where loop is the label of a program location, 10(Hex) and the above instruction is at address 18(Hex), then the branch address (BA) is calculated as follows:

```
BA = (18   + 2) = 1A(Hex)
```

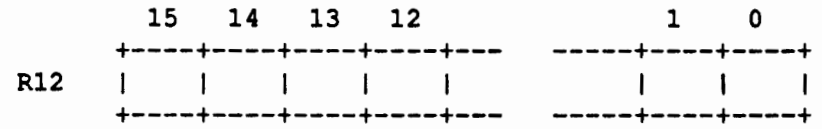
Subtracting from this the location loop

```
1A(Hex)   - 10(Hex)    = A(Hex)      = 10 decimal
```

The displacement is 5 (one half of the value) and is negative sine LOOP is 5 words prior to location 18(HEX) + 2.

B.1.4.8 CRU Addressing -

The CRU uses the number contained in bits 1 thru 11 of register R12, to form a hardware base address.



CRU Hardware Base Address = Contents of R12 divided by 2.

Thus if R12 contains 0400(Hex) (the software base address), bits 1 thru 11 will be 0200(Hex). This is used to indicate the starting CRU bit address for multiple CRU transfer instructions. It is added to the displacement contained in single bit CRU instructions to form the CRU bit address. For example, to set CRU bit 208 to one, with register R12 containing 400(Hex), the following instruction would be used:

```
SBO     8
```

so that the CRU address is 200(Hex) + 8 = 208(Hex).

B.1.5 Instruction Description

The instructions of the 9900 are divided into the following groups:

Data Transfer

Load
Move

Arithmetic Instruction

Addition
Subtraction
Negative and Absolute Value
Increment and Decrement
Multiply
Divide

Compare Instructions

Compare Immediate
Compare Words
Compare Bytes
Compare 1's corresponding
Compare 0's corresponding

Logical Instructions

OR and AND
Exclusive OR
Complement/Clear
Set Bits corresponding (Inclusive OR)

Shift Instructions

Shift Left Arithmetic
Shift Right Arithmetic
Shift Right Logical

Branch Instructions

Control/CRU Instructions

B.1.6 Subroutine Calls

Three types of subroutine calls are used with the 9900. These are as follows.

B.1.6.1 Branch And Link(BL) -

This method shares the workspace with the calling program. Execution of a BL causes the contents of the program counter to be stored in register R11. An example is shown below:

```
LAB     BL      @SUB1
```

*SUB1 is the label of the first instruction of the subroutine being called. LAB is the symbolic address of the BL instruction. After the BL instruction is executed, the value (LAB+4) is stored in R11. Return from the subroutine is achieved by an RT instruction which is effectively a B *R11 (Branch). The process is illustrated in figure B-8.*

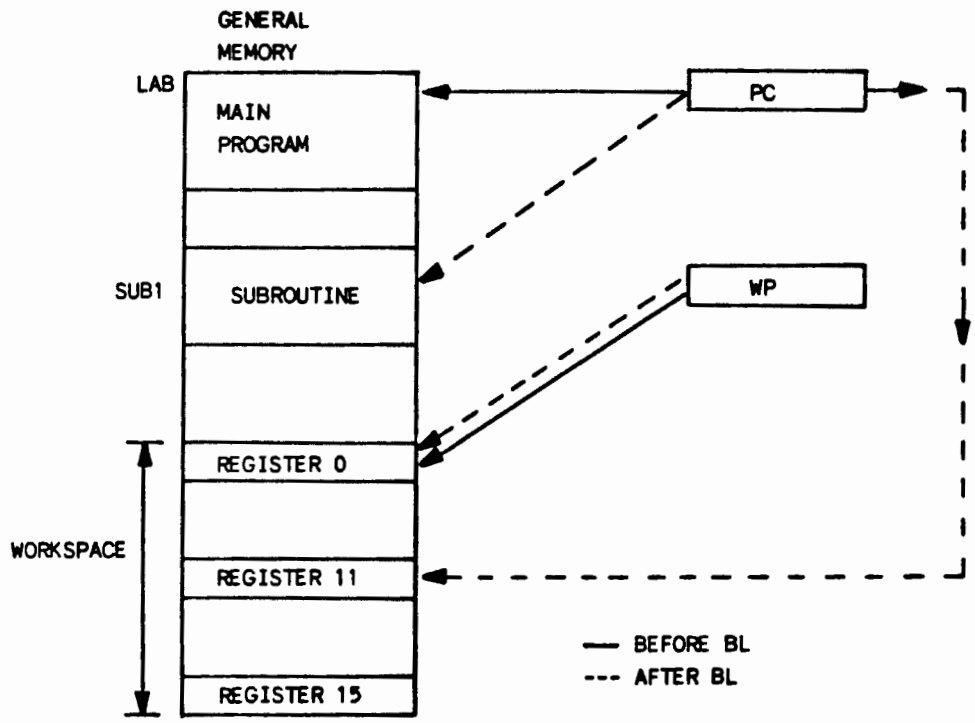


Figure B-8 Effects of the BL Instruction

If multiple levels of BL are used then a stack must be developed to save intermediate linkages.

B.1.6.2 Branch An Load Workspace Pointer(BLWP) -

This call initiates a context switch, meaning that it uses a new workspace and returns via an RTWP instruction. The old WP, PC and ST are saved in registers R13, R14 and R15. As an example, the instruction:

```
LAB1    BLWP    @TUAL
```

is illustrated in figure B-9 below.

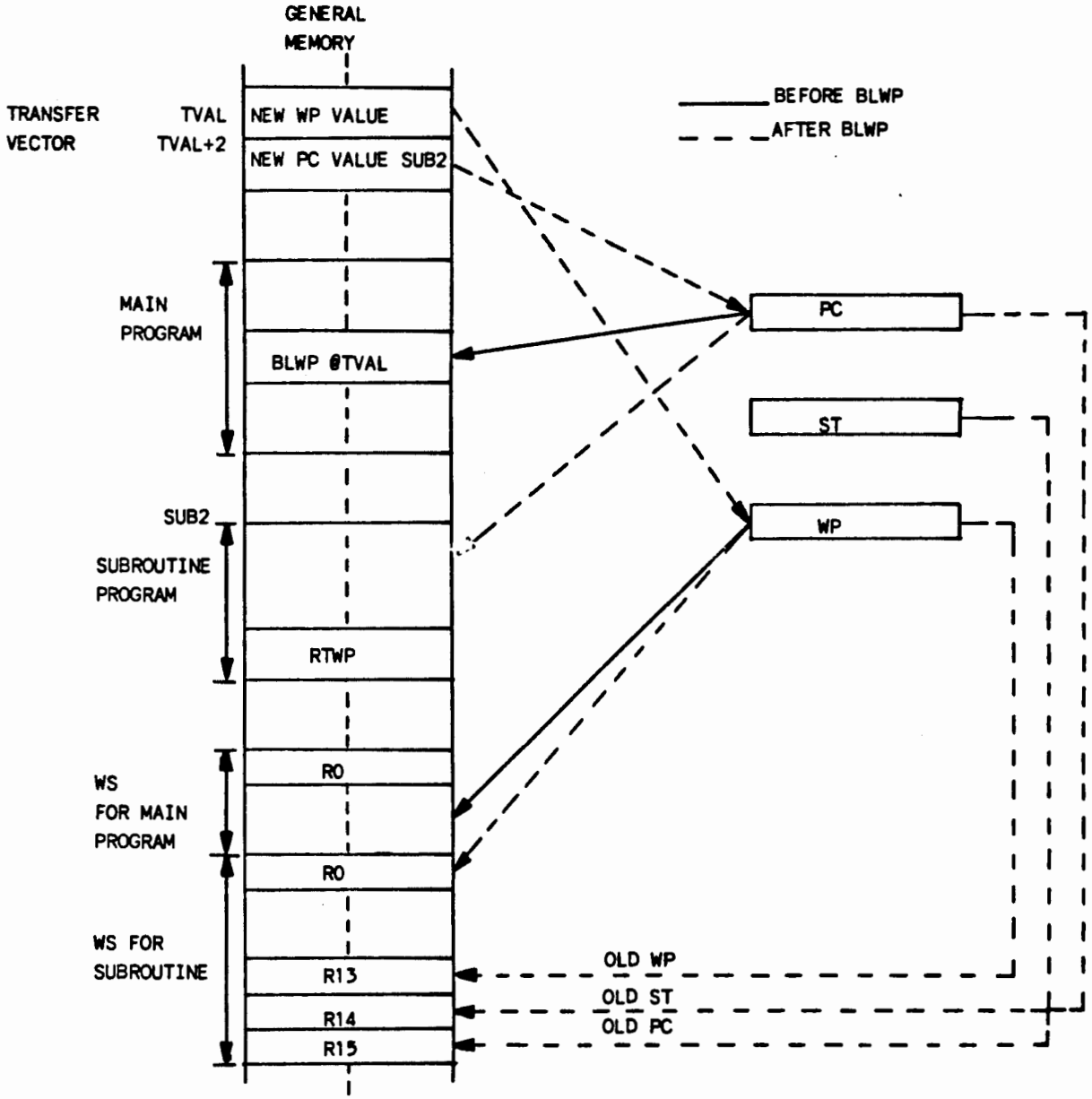


Figure B-9 Effects of the BLWP Instruction

TUAL is the symbolic address of the transfer vector which provides new values for the WP and PC.

Instruction	Format	Effect	Opcode
Load Communications Register	LDCR G,T	Note 1	3000
Set Bit to One	SBO E	Note 2	1D00
Set Bit to Zero	SBZ E	Note 3	1E00
Store Communications Register	STCR G,T	Note 4	3400
Test Bit	TB E	Note 5	1F00

Table B-4. CRU Instructions

NOTES:

1. Transfers consecutive data bits from the byte address specified by G to the CRU. The number of bits to be transferred is specified by T. The CRU address is the contents of R12 of the current workspace. The least significant bit of the byte address specified by G is placed in the CRU bit addressed by R12.

2. Sets CRU bit at address R12+E to 1.

3. Sets CRU bit at address R12+E to 0.

4. Transfers consecutive data bits to the byte address specified by G from the CRU. The number of bits to be transferred is specified by T. The CRU address is the contents of R12 of the current workspace. The CRU bit addressed by R12 is placed in the least significant bit of the byte addressed by G.

5. Tests CRU bit at address in R12+E. Sets status bit 13 to the value of the CRU bit.

Instruction	Format	Effect	Opcode
Load Immediate	LI R,I	I -> (R)	0200
Load Interrupt Mask	LIMI I	Note 1	0300
			0320
Load Workspace Pointer	LWPI I	I -> (WP)	02E0
Move Words	MOV G1,G2	(G1) -> (G2)	C000
Move Bytes	MOVB G1,G2	(G1) -> (G2)	D000
Store Status	STST R	(ST) -> (R)	02C0
Store WP	STWP R	(WP) -> (R)	02A0
Swap Bytes	SWPB G	Note 2	06C0

Table B-5. Load and Move Instructions

NOTES:

1. Places the least significant 4 bits of the immediate value I in the interrupt mask.

2. Interchanges bits 0-7 with bits 8-15 of word at address specified by G.

B.1.6.3 XOP Instruction -

The XOP instruction is similar to BLWP with the added feature of direct passing of a parameter address to the R11 of the new workspace. By using the assembler directive DXOP, a mnemonic string can be defined to represent one of the 16 XOP vectors. As an example, to define XOP is as the mnemonic SAMPL, the following can be used:

```
DXOP    SAMPL,15
```

then, instead if the standard XOP entry in the program:

```
XOP     @PARAM,15
```

the newly defined mnemonic:

```
SAMPL   @PARAM
```

can be used. This is effectively a user defined instruction. The effect of the XOP instruction is described in figure B-10 below.

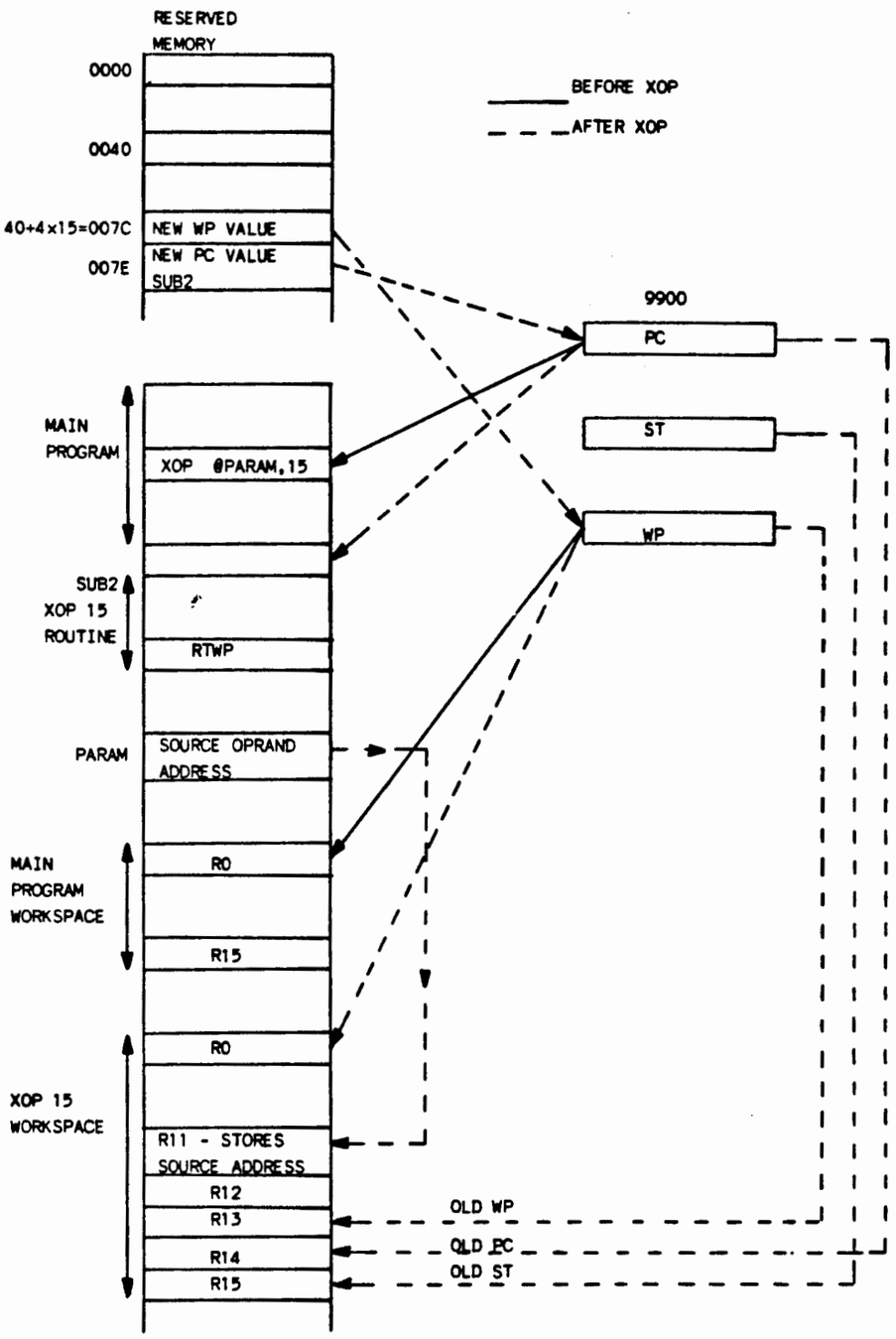


Figure B-10 Execution of XOP Instruction XOP @PARAM, 15

B.2 TMS900 INSTRUCTION SET

The source formats of the TMS9900 machine instructions are presented in 8 tables. Arithmetic instructions are listed in table B-1 and branch instructions are listed in table B-2. Table B-3 lists the compare instructions and table B-4 lists the CRU instructions. Load and move instructions are listed in table B-5 and logical instructions are listed in table B-6. Workspace register shift instructions are listed in table B-7 and the pseudo-instructions are listed in table B-8.

The following symbols are used in tables B-1 through B-8.

G, G1, G2 - A general address

R - Workspace register

S - Symbolic address

I - Immediate value

T - A term

() - Contents of the address within the parentheses

-> - "replaces"

:- - "is compared to"

Instruction	Format		Effect	Opcode
Add Words	A	G1,G2	(G1)+(G2) -> (G2)	A000
Add Bytes	AB	G1,G2	(G1)+(G2) -> (G2)	B000
Absolute Value	ABS	G	Absolute (G) -> (G)	0740
Add Immediate	AI	R,I	(R)+I -> (R)	0220
Decrement	DEC	G	(G)-1 -> (G)	0600
Decrement by 2	DECT	G	(G)-2 -> (G)	0640
Divide	DIV	G,R	Note 1	3C00
Increment	INC	G	(G)+1 -> (G)	0580
Increment by 2	INCT	G	(G)+2 -> (G)	05C0
Multiply	MPY	G,R	Note 2	3800
Negate	NEG	G	-(G) -> (G)	0500
Subtract	S	G1,G2	(G2)-(G1) -> (G2)	6000
Subtract Bytes	SB	G1,G2	(G2)-(G1) -> (G2)	7000

Table B-1. Arithmetic Instructions

NOTES:

1. Contents of register R and the next consecutive register (32-bits) are divided by G (16-bits). The quotient is placed in R and the remainder is placed in R+1.

2. (G) is multiplied by (R). The result (32-bits) is placed in R and R+1. R contains the most significant half of the result.

Instruction	Format		Effect
Branch	B	G	G -> (PC)
Branch and Link	BL	G	G -> (PC) (PC) -> (R11)
Branch and Link WP	BLWP	G	Note 1
Jump If Equal	JEQ	S	S -> (PC)
Jump If High or Equal	JHE	S	S -> (PC)
Jump If Greater Than	JGT	S	S -> (PC)
Jump If Logical High	JH	S	S -> (PC)
Jump If Logical Low	JL	S	S -> (PC)
Jump If Less or Equal	JLE	S	S -> (PC)
Jump If Less Than	JLT	S	S -> (PC)
Unconditional Jump	JMP	S	S -> (PC)
Jump If No Carry	JNC	S	S -> (PC)
Jump If Not Equal	JNE	S	S -> (PC)
Jump If No Overflow	JNO	S	S -> (PC)
Jump If Odd Parity	JOP	S	S -> (PC)
Jump On Carry	JOC	S	S -> (PC)
Return WP	RTWP		Note 2

Table B-2. Jump Instructions

NOTES:

1. BLWP
   (G) -> (WP)
   (G+2) -> (PC)
   (original WP) -> (R13)
   (old PC) -> (R14)
   (ST) -> (R15)

2. RTWP
   (R13) -> (WP)
   (R14) -> (PC)
   (R15) -> (ST)

Instruction	Format	Opcode
Compare Words	C G1,G2	8000
Compare Bytes	CB G1,G2	9000
Compare Immediate	CI R,I	0280
Compare Ones Corresponding	COC G,R Note 1	2000
Compare Zeros Corresponding	CZC G,R Note 2	2400

Table B-3. Compare Instructions

NOTES:

General: Compare instructions only set the status bits and have no other effect.

1. The bits in the destination operand that correspond to bits equal to 1 in the source operand are compared to 1. If the corresponding bits are equal to 1, status bit 13 is set. Otherwise the status bit is set to 0.

2. The bits in the destination operand that correspond to bits equal to 0 in the source operand are compared to 0. If the corresponding bits are equal to 0, status bit 13 is set. Otherwise the status bit is set to 0.

Instruction	Format	Effect	Opcode
AND Immediate	ANDI R,I	(R) AND I -> (R)	0240
Clear	CLR G	0 -> (G)	04C0
Invert Bits	INV G	Note 1	0540
Set to Ones	SETO G	>FFFF -> (G)	0700
Set Ones Corresponding	SOC G1,G2	Note 2	E000
Set Ones Corresponding Bytes	SOCB G1,G2	Note 2	F000
Set Zeros Corresponding	SZC G1,G2	Note 3	4000
Set Zeros Corresponding Bytes	SZCB G1,G2	Note 3	5000
Exclusive OR	XOR G,R	(G) XOR (R) -> (R)	2800

> - Hexadecimal Value

Table B-6. Logical Instructions

NOTES:

1. Places 1's complement of contents of location G in location G.

2. Sets bits to 1 in G2 that correspond to bits equal to 1 in G1. (Inclusive OR).

```
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0  G1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0  G2
--------------------------------
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  G2 (result)
```

2. Sets bits to 0 in G2 that correspond to bits equal to 1 in G1.

```
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0  G1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0  G2
--------------------------------
0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0  G2 (result)
```

Instruction	Format	Value In Vacated Bit on Each Shift	Opcode
Shift Right Arithmetic	SRA R,C	Original Value	0800
Shift Right Logical	SRL R,C	Logical Zero	0900
Shift Left Arithmetic	SLA R,C	Logical Zero (note 1)	0A00
Shift Right Circular	SRC R,C	Rightmost bit moves to leftmost bit	0B00

Table B-7. Shift Instructions

NOTES:

General: If C is 0, the 4 least significant bits of R0 contain the shift value. If the 4 least significant bits of R0 equal 0, shift 16 positions. Otherwise shift C positions. The value of the last bit shifted out of the register is placed in status bit 12. The shifted value is compared to zero, setting status bits 15-13.

1. If the sign of the value in R changes during shift, sets status bit 11.

Instruction	Equivalent Instruction	Opcode
NOP	JMP $+2	1000
RT	B *R11	045B

$ - Current Location

Table B-8. Pseudo-Instructions

APPENDIX C

DFDAU INPUT PARAMETER LIST

The following is the input parameter list for the DFDAU, listed along with the source and destination of each parameter. In the cases where the destination is listed as a DFDR word greater than 64, the parameter is downloaded to cpu2 and then output to the QAR recorder in the assigned word. The QAR recorder used in this system configuration has a 128 word data subframe. Words 1 to 64 of the QAR are identical to words 1 to 64 of the DFDR.

The assigned port numbers for the parameters (column 1) are the actual hardware ports that link the DFDAU to the external sensors on the aircraft. These ports are then internally wired to the assigned channels of the various DFDAU hardware modules. For example, on page C-2, analog port 14 is connected to the sensor measuring the "Flap Handle Position" parameter and is internally wired to the AMX channel 19. These connections however are transparent to the software as it concerned only with channel numbers.

For the DITS parameters, the source is given as one of the external aircraft data busses. For example on page C-8, the source TMC means that the data comes from the external data bus connected to the "Throttle Management Computer". This again is of no significance to the software as it is internally connected to an assigned DITS channel, in this case, channel 1 of the DITS 2 module. An SDI notation of "XX" indicates that any SDI code is acceptable.

TABLE C-1

ANALOG INPUT PARAMETERS

Analog Port No.	AMX Card/Ch No.	Parameter Name	Signal Type	Output Bits	Aux Output Seq No.	DFDR Output	
						SF	Word
14	XA03-19	Flap Handle Position	LLDC	12-3		ALL	8,40
1	XA03-1	Vertical Acceleration	LLDC	12-1		ALL	2,10,18 26,34 42,50 58
3	XA03-2	Lateral Acceleration	LLDC	12-2		ALL	15,31 47,63
13	XA03-9	Longitudinal Acceleration	LLDC	12-2		ALL	55
15	XA03-20	Spoiler Handle Pos	Synchro	12-3		ALL	16,48
17	XA03-10	Flap Pos L	Synchro	12-3		ALL	39
16	XA03-11	Flap Pos R	Synchro	12-3		ALL	7
21	XA03-13	Horizontal Stab. Pos	Synchro	12-3		ALL	24,56
-	XA03-22	Potentiometer Excitation	+5V	-	-	-	-

DFDAU INPUT PARAMETER LIST

TABLE C-2

DISCRETE INPUT PARAMETERS

Discrete Port No.	DMX Card No.	Input Channel Number	Parameter Name	Signal Type	Output				
					Aux Output		DFDR		
					Seq No.	Bit	SF No.	Word No.	Bit
37	XA01	17	Air Gnd SW	Series			ALL	9,25,41 57	1
33		13	EFIS Sym Gen Capt				2	43	1
39		19	EICAS SW Pos				4	51	1
25		5	L.E.Slats All Full Ex				2	51	2
26		6	L.E.Slats All Part Ex	Series			2	51	1
15		46	Capt's ADC SW Pos	Shunt			1	43	1
13		44	Capt's FMC SW Pos				4	43	1
7		38	HF Keying L				ALL	14	1
8		39	HF Keying R				ALL	13	1
6		37	L.E. Slats Agree/Disagree				ALL	15	1
19		50	L.E. Slats InBd (L) EX				2&4	7	1
17		48	L.E. Slats OBd (L) EX				1&3	7	1
20		51	L.E. Slats InBd (R) EX				2&4	7	2
18		49	L.E. Slats OBd (R) EX				1&3	7	2
5		36	VHF Keying C				ALL	5	1
3		34	VHF Keying L				ALL	3	1
4		35	VHF Keying R				ALL	4	1
11		42	Capt's IRS SW Pos	Shunt			3	43	1
75		57	Engine Start L	AC			2+4	62	1
76		58	Engine Start R	AC			2+4	62	2
40		20	YAW Damper Eng L	Series			2	31	1
41		21	YAW Damper Eng R	Series			4	31	1
82		1	A/C Type Ident. LSB				-	-	-
83		2	A/C Type Ident. 2nd				-	-	-
84		3	A/C Type Ident. MSB				-	-	-
88		4	A/C Fleet Ident. LSB				-	-	-
89		7	A/C Fleet Ident. MSB				-	-	-
92		8	A/C Number Ident.						
93		9							
94		10							
95		11							
96		12							
97		14							
98		15							
99	XA01	16	A/C Number Ident.	Series					

DFDAU INPUT PARAMETER LIST

TABLE C-3

DITS INPUT PARAMETERS

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
1	EEC L	Lo	346-XX	XA11-1	N1 Actual Left (GE only)	27-17		11	12-2		ALL	5
1	EEC L		133-XX	XA11-1	TLA	28-18	SR	11	12-2		ALL	88
1	EEC L		130-XX	XA11-1	TT2	28-17	SR	12	12-1		ALL	81
1	EEC L		342-XX	XA11-1	N1 Max (GE only)	28-17		12	12-1		1+3	119
1	EEC L		342-XX	XA11-1	EPR Limit (P&W only)	28-17		12	12-1		1+3	119
2	EEC R		346-XX	XA11-2	N1 Actual Right (GE only)	27-17		11	12-2		ALL	37
2	EEC R		133-XX	XA11-2	TLA	28-18	SR	11	12-2		ALL	89
2	EEC R		130-XX	XA11-2	TT2	28-17	SR	12	12-1		ALL	82
2	EEC R		342-XX	XA11-2	N1 Max (GE only)	28-17		12	12-1		2+4	119
2	EEC R		342-XX	XA11-2	EPR Limit (P&W only)	28-17		12	12-1		2+4	119
5	ADC		221-XX	XA11-3	Angle of Attack (Ind.)	27-18	SR	10	12-3		ALL	83
			211		Total Air Temp	26-18	S	10	12-3		ALL	12
			206		Computed Airspeed	27-15	R	12	12-1		ALL	21
			203		Press Alt 29.92	27-16	SR	12	12-1		ALL	6
			205		Mach	26-16	R	10	12-3		ALL	65
			204		Altitude Baro NR 1	27-16	SR	12	12-1		1	84
			220		Altitude Baro NR 2	27-16	SR	12	12-1		2	84
			251		Altitude Baro NR 3	27-16	SR	12	12-1		3	84
			252		Altitude Baro NR 4	27-16	SR	12	12-1		4	84
			212		Altitude Rate	28-18	SR	11	12-2		2+4	90
			222		Angle of Attack L	28-18	SR	11	12-2		1+3	38
			223		Angle of Attack R	28-17	SR	12	12-1		1+3	79
			235		Baro.Cor. No. 1 (IN)*	29-11	S	20	See Note A		1+3	85
			234		Baro.Cor. No. 1 (MB)*	29-11	S	20			2+4	85
			237		Baro.Cor. No. 2 (IN)*	29-11	S	20			1+3	86
			236		Baro.Cor. No. 2 (MB)*	29-11	S	20			2+4	86
			035		Baro.Cor. No. 3 (IN)*	29-11	S	20			1+3	87
			034		Baro.Cor. No. 3 (MB)*	29-11	S	20			2+4	87
			241		Corrected A.O.A.	28-17	SR	12	12-1		2+4	79
			215		Impact Pressure	28-17	SR	12	12-1		2+4	77
			207		Max. OPR TG Schedule	28-17	SR	12	12-1		1+3	80
			213		Static Air Temp	28-18	S	12	12-1		1+3	76
			233		Static Air Temp D*	29-19	S	12	12-1		2+4	76
			231		Total Air Temp D*	29-19	S	12	12-1		1+3	77
			242		Total Pressure	28-17	SR	12	12-1		ALL	122
			210		True Airspeed	27-16	-	12	12-1		ALL	78
5	ADC	Lo	230-XX	XA11-3	True Airspeed D*	29-19	-	11	11-1		2+4	80

S: Sign Required
R: Rounding Required

Note A: See Appendix C for bit format

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
7	Clock	Lo	150-XX	XA11-6	GMT Seconds	17-12	-	6	6-1		1	23
7	Clock	Lo	150-XX	XA11-6	GMT Minutes	23-18	-	6	12-7		1	23
7	Clock	Lo	150-XX	XA11-6	GMT Hours	28-24	-	5	12-8		2	23

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name		Data Scaling: Input Bits	Data Scaling: Output Bits	Aux Output Seq No.	DFDR Output: SF	DFDR Output: Word
1	EEC-L	Lo	270-00	XA11-1	Eng Model Code	#1	17	1		3	112
						#2	18	2			
						#3	19	3			
						#4	20	4			
						#5	21	5			
					Eng Model Code	#6	22	6		3	112
					ECS Pack	G.E.	23	2		ALL	13
					ECS Mode	only	24	2		4	51
					Cowl Anti-ice		25	1		ALL	16
					Wing Anti-ice		26	2		ALL	24
					ADP On		27	1		ALL	32
			270-00		Isol Valve Open (Note)		28	2		ALL	39
			271-00		EEC/PMC On		20	1		1	40
			271-00		Latched Fault		22	1		2	40
			271-01		ECS Pack		21	2		ALL	13
					ECS Mode		22	2		4	51
					Cowl Anti-ice	P&W	25	1		ALL	16
					Wing Anti-ice	Only	24	2		ALL	24
					ADP ON		23	1		ALL	32
1	EEC-L			XA11-1	Isol Valve Open		26	2		ALL	39
2	EEC-R			XA11-2	ECS Pack		21	2		ALL	16
					ECS Mode		22	1		3	51
					Cowl Anti-ice	P&W	25	1		ALL	20
					Wing Anti-ice	Only	24	1		ALL	24
					ADP ON		23	1		ALL	39
			271-01		Isol Valve Open		26	2		ALL	40
			271-00		EEC/PMC On		20	1		3	40
			271-00		Latched Fault		22	1		4	40
			270-00		Eng Model Code	#1	17	1			112
						#2	18	2			
						#3	19	3			
						#4	20	4			
						#5	21	5			
					Eng Model Code	#6	22	6		4	112
					ECS Pack	G.E.	23	2		ALL	16
					ECS Mode	Only	24	1		3	51
					Cowl Anti-ice		25	1		ALL	20
					Wing Anti-ice		26	1		ALL	24
					ADP ON		27	1		ALL	39
2	EEC-R	Lo	270-00	XA11-2	Isol Valve Open		28	2		ALL	40

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
8	TMC	Lo	142-00	XA12-1	A/T Fast/Slow CMD	28-21	S	9	12-4		ALL	93
↕	↕	↕	341-10	↕	EPR Bug Drive L/N1 Bug Dr L	28-17	SR	12	12-1		1+3	95
			341-01		EPR Bug Drive R/N1 Bug Dr R	28-17	SR	12	12-1		2+4	95
			342-00		EPR Reference /N1 Ref.	28-17	SR	12	12-1		3	94
			303-00		Max EPR Limit /N1 Max	28-17	SR	12	12-1		2	94
			134-10		Power Lever Angle L	28-17	SR	12	12-1		ALL	91
			134-01		Power Lever Angle R	28-17	SR	12	12-1		ALL	92
8	TMC	Lo	213-00	XA12-1	Temp Selected	28-18	S	12	12-1		1	94
11	EFIS	Lo	324-XX	XA11-5	Pitch Attitude	28-19	S	11	12-2		ALL	4,20, 36,52
↕	↕	↕	325-XX	↕	Roll Attitude	28-19	S	11	12-2		↕	17,49
			173-XX		Localizer Dev.	28-19	S	11	12-2			53
			174-XX		Glide Slope Dev.	28-19	S	11	12-2			14
			164-XX		Radio Attitude	27-17	S	12	12-1		ALL	60
			320-XX		Magnetic Heading	28-19	S	11	12-2		ALL	3
			317-XX		Track Angle Mag	28-19	S	11	12-2		1+3	96
			313-XX		Track Angle True	28-19	S	11	12-2		2+4	96
11	EFIS	Lo	314-XX	XA11-5	True Heading	28-19	S	11	12-2		ALL	97
12	FCC-L	Lo	137-XX	XA12-2	Flap Position	28-19	S	11	12-2		ALL	98
↕	↕	↕	141-XX	↕	Flt Dir-Pitch	28-17	SR	12	12-1		1+3	100
			140-XX		Flt Dir-Roll	28-17	SR	12	12-1		2+4	100
			206-XX		FMC Airspeed Ref	28-17	SR	12	12-1		1+3	102
			101-XX		Heading Selected	28-20	S	10	12-3		2+4	102
			004-XX		Runway Dist to Go*	29-19		11	11-1		1+3	101
			017-XX		Select Runway Hdg*	29-19		11	11-1		2+4	101
			315-XX		Stab Position	28-17	SR	12	12-1		ALL	99
12	FCC-L	Lo	104-XX	XA12-2	V/S (Vertical Speed)	28-19	S	10	12-2		1+3	90

Note : The EPR associated parameters are for P & W only.
The N1 associated parameters are for G E only.

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
5	ADC	Lo	352-XX	XA11-3	A/C Type Mem Check	17	7		2	118
					PT Mem Check	16	6			
					PS Mem Check	15	5			
					PROM #4 Check	14	4			
					PROM #3 Check	13	3			
					PROM #2 Check	12	2			
5	ADC	Lo	352-XX	XA11-3	PROM #1 Check	11	1		2	118
6	GPWS	Lo	270-XX	XA11-4	Terrain Pull-up	20	12		ALL	32
					Minimums	19	11			
					Glide Slope	18	10			
					Too Low Terrain	17	9			
					Too Low Flaps	16	8			
					Too Low Gear	15	7			
					Don't Sink	14	6			
					Terrain	13	5			
					Pull Up	12	4			
6	GPWS	Lo	270-XX	XA11-4	Sink Rate	11	3		ALL	32

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
5	ADC	Lo	270-XX	XA11-3	VMO Alt #1	22	10		3	51
					R. A.O.A. Heat On	18	7			
					L. A.O.A. Heat On	17	6			
					TAT Probe Heat On	16	5			
					Pitot/Stat Heat On L	14	3			
					ADC Invalid	13	2			
					Onside A.O.A. Fail	20	9			
					Over Speed	19	8			
					VMO Alt #2	23	11			
					VMO Alt #3	24	12			
					Pitot /Stat Heat On R	15	4		3	51
					Zero SSEC (Mach)	29	8		1	112
					Baro Port "A"	28	7			
					A.O.A. C Alternate	27	6			
					SSEG Alternate	26	5			
					VMO Alternate #4	25	4			
					A.O.A. Unique	21	3			
					Spare	12	2			
			270-XX		Spare	11	1		1	112
			206-XX		CAS Flag	31-30	1		2	112
			205-XX		Mach Flag	31-30	2			
			221-XX		A.O.A. Flag	31-30	3			
			203-XX		Press Alt Flag	31-30	4			
			211-XX		T.A.T. Flag	31-30	5			
			271-XX		#4 Baro Alt = #3	18	10			
			271-XX		#4 Baro Alt = #2	17	9			
			271-XX		#4 Baro Alt = #1	16	8			
			271-XX		Ext A.O.A. Mon (Fail)	12	7			
			271-XX		Zero SSEC (A.O.A.)	11	6		2	112
			350-XX		F/D Conversion Test	29	12		2	117
					PS Calib Test	28	11			
					PS Sens. Temp Test	27	10			
					PS Sens. Per Test	26	9			
					Prog. Mem Test	25	8			
					RAM Test	24	7			
5	ADC	Lo	350-XX	XA11-3	Processor Test	23	6		2	117

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
5	ADC	Lo	350-XX	XA11-3	A to D Test	22	5		2	117
					OSPD Hdwr Test	21	4		2	117
					A/C Type Const Test	20	3		2	117
					A/C Type Prog Test	19	2		2	117
					BARO #3 Test	18	1		2	117
					BARO #2 Test	17	12		4	117
					BARO #1 Test	16	11		4	117
					T.A.T. Input Test	15	10		4	117
					R A.O.A. Vane Test	12	9		4	117
			350-XX		L A.O.A. Vane Test	11	8		4	117
			351-XX		VMO Test	27	10		3	117
					BARO #4 Test	26	9			
					EAROM Test	25	8			
					PT PLL	24	7			
					PS PLL	23	6			
					Prog Seq Test	22	5			
					Temp PS= Temp Pt	21	4			
					Average A.O.A. Test	20	3			
					PS=PT	19	2			
					A.O.A. Compare Test	18	1		3	117
					Power Supply Test	17	7		4	
					ARINC XMTR Test	16	6			
					ARINC XMTR Test	15	5			
					ARINC XMTR Test	14	4			
					PT Calib Test	13	3			
					PT Sens Temp Test	12	2			
			351-XX		PT Sens Per Test	11	1		4	117
			352-XX		A/C Type Parity	29	10		1	118
					A/C Type MSB	28	9			
					A/C Type LSB +3	27	8			
					A/C Type LSB +2	26	7			
					A/C Type LSB +1	25	6			
					A/C Type LSB	24	5			
					Spare	21	4			
					Spare	20	3			
					Spare	19	2			
5	ADC	Lo	352-XX	XA11-3	Spare	18	1		1	118

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
8	TMC	Lo	145-XX	XA12-1	Air Driven Pump	24	12		ALL	103
					Wing Anti-Ice	23	11			
					Cowl Anti-Ice-R	21	10			
					Cowl Anti-Ice-L	20	9			
					Isol Valve Right	18	8			
					Isol Valve Left	17	7			
					Spare	16	6			
					Spare	15	5			
					ECS Pack R H/L	14	4			
					ECS Pack R	13	3			
					ECS Pack L H/L	12	2			
					ECS Pack L	11	1		ALL	103
					TMC V NAV Oper	29	12		ALL	30
					FLCH Mode Oper } GE only	22	8		ALL	30
8	TMC	Lo	145-XX	XA12-1	Throttle Hold Oper }	25	7		ALL	30

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling: Input Bits	Data Scaling: Output Bits	Aux Output Seq No.	DFDR Output: SF	DFDR Output: Word
8	TMC	Lo	270-XX	XA12-1	Pre-Select Climb	24	1		ALL	90
					Engine Indent 12	23	12		1	116
					11	22	11			
					10	21	10			
					9	20	9			
					8	19	8			
					7	18	7			
					6	17	6			
					5	16	5			
					4	15	4			
					3	14	3			
					2	13	2			
					Engine Indent 1	12	1		1	116
			270-XX		Temp Derate Status	11	1		ALL	29
			146-XX		Rating 2 Oper	24	12		2+4	51
					Rating 1 Oper	23	11			
					G/A Mode Oper	22	10			
					CRZ Mode Oper	21	9			
					Con Mode Oper	20	8			
					Climb Mode Oper	19	7			
					T/O Mode Oper	18	6			
					Thrust Mode Oper	13	5			
					Mach Mode Oper	12	4			
					IAS Mode Oper	11	3		2+4	51
					Flap Limit	15	1		ALL	30
					Min Speed	16	2			30
					Ground Test	17	1			83
					Idle Thrust Oper	25	3			30
					A/T Disconnect	28	5			30
			146-XX		TMC Valid	29	2			83
			145-XX		EEC/PMC Valid L	26	9			30
			145-XX		EEC/PMC Valid R	27	10			30
			146-XX		A/T Engaged	27	4			30
			146-XX		Speed Limit (G E only)	14	2			29
			145-XX		Flair Retard Mode (G E only)	28	11			30
8	TMC	Lo	145-XX	XA12-1	A/T Go Around Mode Oper (G E only)	19	6		ALL	30

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	270-XX	XA11-5	Mag/TRU Data	19	1		ALL	73
					DH Alert	14	2		ALL	45
					H Alert	16	1		ALL	48
					Groundspeed Source	17	2		2+4	48
					Track Angle Source	18	2		1+3	48
			270-XX		DH + ΔH Alert	15	1		ALL	45
			274-XX		G/S Mode Oper	29	12		ALL	113
					Flare Oper	28	11			
					G/A Mode Oper-P	27	10			
					T/O Mode Oper-P	26	9			
					Alt Hold Mode Oper	25	8			
					V/S Mode Oper	24	7			
					V/Nav Mode Oper	23	6			
					FL CH Mode Oper	22	5			
					Throttle Retard	21	4			
					IAS Mode Set	20	3			
					Mach Mode Set	19	2			
					Alt Mode Oper	11	1		ALL	113
11	EFIS	Lo	274-XX	XA11-5	Pitch Speed Cntrl	10	1		ALL	104

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	352-XX	XA11-5	WXR Data Fault	29	12		2	116
					MLS Data Fault	28	11			
					ILS Data Fault	27	10			
					RA Data Fault	26	9			
					R-DME Data Fault	25	8			
					L-DME Data Fault	24	7			
					R-VOR Data Fault	23	6			
					L-VOR Data Fault	22	5			
					R-ADC Data Fault	21	4			
					L-ADC Data Fault	20	3			
					R-IRS Data Fault	19	2			
					C-IRS Data Fault	18	1		2	116
					L-IRS Data Fault	17	12		3	116
					TMC Data Fault	16	11			
					R-FMC Data Fault	15	10			
					L-FMC Data Fault	14	9			
					R-FCC Data Fault	13	8			
					C-FCC Data Fault	12	7			
			352-XX 353-XX		L-FCC Data Fault	11	6			
					SG Fault	18	5			
					R EHSI Fault	17	4			
					L EHSI Fault	16	3			
					R CP Fault	15	2			
					L CP Fault	14	1		3	116
					R EADI Fault	12	12		4	116
11	EFIS	Lo	353-XX	XA11-5	L EADI Fault	11	11		4	116

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	350-XX	XA11-5	SG I/O Proc No. 3	20	10		4	116
					SG I/O Proc No. 2	19	9			
					SG I/O Proc No. 1	18	8			
					SG Digital Output	17	7			
					SG Controller	16	6			
					SG Dsply Sequencer	15	5			
					SG Dsply Drive	14	4			
					SG Main Memory	13	3			
					SG Main Processor	12	2			
			350-XX		SC Overtemp	11	1		4	116
			351-XX		R-CP Fault	25	1		1+3	115
					L-CP Fault	24	1		2+4	115
					R-EHSI Overtemp	23	12		1	117
					R-EHSI Beam Fail	22	11			
					R-EHSI Anomalies	21	10			
					L-EHSI Overtemp	20	9			
					L-EHSI Beam Fail	19	8			
					L-EHSI Anomalies	18	7			
					R-EADI Overtemp	17	6			
					R-EADI Beam Fail	16	5			
					R-EADI Anomalies	15	4			
					L-EADI Overtemp	14	3			
					L-EADI Beam Fail	13	2			
11	EFIS	Lo	351-XX	XA11-5	L-EADI Anomalies	12	1		1	117

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
12	FCC-L	LO	272-XX	XA12-2	Test Bit 1	14	12		ALL	114
			272-XX		A/P Eng A Detent	11	11			
			273-XX		Autotrim Down Arm	29	10			
					Autotrim Down Cont	28	9			
					Autotrim Eng L Sam	27	8			
					Autotrim Eng R Sam	26	7			
					Autotrim Up Arm	25	6			
					Autotrim Up Cont	24	5			
					Autotrim Growth 1	23	4			
					Autotrim Growth 2	22	3			
					Appr Mode Arm	16	2			
					Low Mode (A/O)	15	1		ALL	114
					Appr Mode Oper	13	12		ALL	115
			273-XX		V-Nav Mode (A/O)	12	11			
			274-XX		Throttle Retard	21	10			
					IAS Mode Set	20	9			
					Mach Mode Set	19	8		ALL	115
					Mach Limit Oper	15	3		4	23
					IAS Limit Oper	16	4			
					Mach	17	5			
					IAS	18	6			
					FlCh Mode Oper	22	7			
					V-NAV Mode Oper	23	8			
					V/S Mode Oper	24	9			
					Alt Hold Mode Oper	25	10			
			274-XX		T/O Mode Oper (Pitch)	26	11		4	23
			275-XX		G/A Mode Oper (Pitch)	27	11		ALL	19
					T/O Mode Oper (Roll)	19	1		2	23
					G/A Mode Oper (Roll)	20	2			
					L-NAV Mode Oper	24	3			
					HDG Set Mode Oper	25	4			
					HDG Hold Mode Oper	26	5			
					B/CRS Mode Oper	29	6		2	23
					Trk Hold Mode Oper	23	7		ALL	115
12	FCC-L	LO	275-XX	XA12-2	Att Hold Mode Oper	22	6		ALL	115

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
12	FCC-L	Lo	272-XX	XA12-2	A/P CMD Eng C	24	4		ALL	33
					A/P CMD Eng R	25	5		ALL	33
					A/P CMD Eng L	26	6		ALL	33
					A/P CWS Eng C	27	1		ALL	37
					A/P CWS Eng R	28	1		ALL	36
					A/P CWS Eng L	29	1		ALL	11
					FMA Fault 1	13	1		ALL	33
					Land 2 (Green)	18	3			
			272-XX		Land 3 (Green)	17	2			
			274-XX		FMA Fault 2	12	7			
			274-XX		Flare Mode Oper	28	8			
			274-XX		G/S Mode Oper	29	9			
			275-XX		FMA Fault 3	21	10			
			275-XX		Loc Mode Oper	27	11			
			275-XX		Rollout Mode Oper	28	12		ALL	33
			272-XX		Flt Dir On F/O	22	1		3	23
			272-XX		Flt Dir On Capt	23	2			
			273-XX		L NAV Mode Arm	11	3			
					Flare Mode Arm	17	4			
					G/S Mode Arm	18	5			
					Rollout Mode Arm	19	6			
					B/CRS Mode Arm	20	7			
			273-XX		Loc Mode Arm	21	8			
			274-XX		Alt Mode Arm	11	9		3	
			274-XX		Flap Limit	13	1		4	
12	FCC-L	Lo	274-XX	XA12-2	Min Speed	14	2		4	23

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name (• Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
14	EICAS	Hi	017-01	XA12-3	EGT L	27-17		11	12-2		ALL	35
			017-10		EGT R	27-17		11	12-2		ALL	19
			017-11		Fuel Flow Left	28-17		12	12-1		ALL	70
			020-00		Fuel Flow Right	28-17		12	12-1		ALL	121
			024-01		Fuel Press L	28-20		9	11-3		2+4	71
			024-10		Fuel Press R	28-20		9	11-3		1+3	71
			023-11		EPR Left	28-17		12	12-1		ALL	28
			024-00		EPR Right	28-17		12	12-1		ALL	44
			023-01		Vib. (IND) L	28-18		11	11-1		ALL	66
			023-10		Vib. (IND) R	28-18		11	11-1		ALL	67
			020-11		N2 Left	27-17	R	10	12-3		ALL	68
			021-00		N2 Right	27-17	R	10	12-3		ALL	69
			021-11		Oil Press L	23-17		7	9-3		ALL	72
			022-00		Oil Press R	23-17		7	9-3		ALL	73
			022-01		Oil Qty L	28-20		9	11-3		1+3	75
			022-10		Oil Qty R	28-20		9	11-3		2+4	75
			022-11		Oil Temp L	26-17	SR	10	12-3		1+3	74
			023-00		Oil Temp R	26-17	SR	10	12-3		2+4	74
			001-00		Aileron Pos. Inbd L	28-19	S	11	12-2		ALL	59
			001-01		Aileron Pos. Inbd R	28-19	S	11	12-2			27
			033-00		Rudder Position	28-20	S	10	12-3			9,25,41,57
			016-11		Elevator Pos. L	28-20	S	10	12-3			29,61
			017-00		Elevator Pos. R	28-20	S	10	12-3			13,45
			001-10		Aileron Pos L Otbd	28-19	S	11	12-2			11
			001-11		Aileron Pos R Otbd	28-19	S	11	12-2			43
			035-11		Eng. Vib. Filter 1 Left Fan	28-18		11	11-1			105
			035-01		Eng. Vib. Filter 2 Left LPT							106
			034-11		Eng. Vib. Filter 3 Left N2							107
			036-00		Eng. Vib. Filter 1 Right Fan							108
			035-10		Eng. Vib. Filter 2 Right LPT							109
			035-00		Eng. Vib. Filter 3 Right N2	28-18		11	11-1		ALL	110
			003-11		APU RPM	28-19		10	10-1		4	64
			003-00		APU EGT	28-17		12	12-1		4	64
			052-00		Gross Weight	28-14		15			3	64
			020-01		N1 Left (P&W only)	27-17		11	12-2		ALL	5
14	EICAS	Hi	020-10	XA12-3	N1 Right (P&W only)	27-17		11	12-2		ALL	37

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	037-10	XA12-3	EICAS Fault	26	1		ALL	27
			010-00		Standby Bus Off	17	1			46
			015-01		C IRS DC Fail	13	2			46
			010-10		L IRS DC Fail	27	1			65
			012-00		R IRS DC Fail	15	1			49
			014-00		L IRS Fault	27	1			52
			015-10		R IRS Fault	15	1		ALL	55
			014-01		L Gen Drive Oil	25	2		2+4	38
			013-00		R Gen Drive Oil	18	1		ALL	53
			010-10		Capt Inst Bus Xfer	20	2		ALL	56
			014-00		FO Inst Bus Xfer	20	1		ALL	56
			010-01		L AC Bus Tie ISOL	19	2		2+4	54
			013-11		R AC Bus Tie ISOL	19	1		2+4	54
			010-10		L Gen GCB Open	18	2		ALL	57
			014-00		R Gen GCB Open	18	1		ALL	59
			011-11		C IRS DC ON	13	2		1+3	61
			013-00		L IRS DC ON	27	2		2+4	61
			016-00		R IRS DC ON	15	2		1+3	62
			010-10		DC Bus Tie	21	1		1+3	62
			014-01		L Util Bus	19	1		ALL	61
			012-11		R Util Bus	19	1		ALL	63
			037-00		Eng Low Oil Press Left	26	1		1+3	47
			015-01		Eng Low Oil Press Right	26	1		2+4	47
			015-00		Left Fwd Entry Door	23	-		-	-
			015-00		Left Eng Fuel Cut-Off	21	-		-	-
14	EICAS	HI	016-01	XA12-3	Right Eng Fuel Cut-Off	20	-		-	-

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	014-00	XA12-3	C Bleed Isol Valve	14	3		4	22
			014-00		Cabin Altitude >10K Feet	26	4			
			014-10		L Fwd Fuel Pump Lo Press	25	5			
			015-11		R Eng Hi Stage Bleed Fault	18	6			
			015-11		L Eng Bleed Ovht	19	7			
			015-11		L Hyd Sys Lo Press	26	8			
			016-01		R Eng Bleed Ovht	18	9			
			016-01		Stabilizer	22	10			
			016-01		Rat Unlocked	28	11			
			037-00		L.E. Slat Asymmetry	17	12		4	22
			014-01		Unsched Stab Movement	21	1		1	51
			015-10		R Hyd Lo Qty	17	2			
			015-10		Fwd Cargo Door	23	3			
			016-00		R Eng Oil Filter Bypass	14	4			
			016-00		L Hyd Lo Qty	17	5			
			016-10		R Bleed Duct Leak	26	6			
			016-10		R Aft Fuel Pump Lo Press	27	7			
			036-11		Aft Cargo Fire	15	8			
			013-00		L Bleed Isol Valve	14	9			
			036-11		R Eng Ovht	25	10			
			037-01		Aft Cargo Door Open	24	11			
			010-01		R Yaw Damper	27	12		1	51
			036-11		L.E. Slats Disagree	19	1		ALL	89
			015-01		R Eng Lo Oil Press	26	1		3	31
			037-00		L Eng Lo Oil Press	26	1		1	31
			010-00		L Eng T/R Unlocked	15	1		ALL	12
			010-10		R Eng T/R Unlocked	15	2			12
			011-00		L Eng T/R Deployed	28	1			8
			013-00		R Eng T/R Deployed	15	2			8
			010-01		L Eng Fire	16	2			9
			011-00		R Eng Fire	26	1			17
			015-00		APU Fire	14	2			25
14	EICAS	Hi	010-00	XA12-3	GPWC Fault	24	2		ALL	32

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	010-00	XA12-3	STBY Bus Off	17	1		1	22
			010-01		L Fuel Filter Bypass	25	2			
			010-01		Gears Disagree	18	3			
			010-11		R AC Bus Off	19	4			
			011-00		C Left Fuel Pump Lo Press	27	5			
			011-01		R Hyd Sys Lo Press	15	6			
			011-01		Flaps (A)	19	7			
			011-01		Over Speed R ADC	22	8			
			011-01		Cabin Auto Press 1	23	9			
			012-11		L Bleed Duct Leak	16	10			
			012-11		Wheel Well Fire	18	11			
			012-11		L Yaw Damper	27	12		1	
			010-10		R Bleed Isol Valve	14	1		2	
			010-10		C Hyd Lo Press	19	2			
			011-11		C Right Fuel Pump Lo Press	16	3			
			012-00		C Hyd Lo Qty	17	4			
			012-01		Len-Hi Stage Bleed Fault	18	5			
			013-01		L Eng Ovht	14	6			
			013-01		Parking Brake (A)	18	7			
			013-01		L AC Buss Off	19	8			
			015-01		Aft Equip Smoke	15	9			
			014-11		Spoilers	19	10			
			014-11		Gear Door System #2	22	11			
			014-11		Cabin Auto Press 2	23	12		2	
			011-10		Fwd Equip Smoke	14	1		3	
			011-10		L Eng Oil Filter Bypass	26	2			
			010-01		Gears Disagree	18	3			
			012-10		Emergency Lights	24	4			
			013-00		Parking Brake (C)	25	5			
			015-00		L Aft Fuel Pump Lo Press	16	6			
			015-00		Gear Not Down	20	7			
			015-00		Fwd Equip. Over Heat	22	8			
			015-00		Fuel Crossfeed	28	9			
			015-01		R Fwd Fuel Pump Lo Press	16	10			
			015-01		APU Bleed Valve	20	11			
			015-01		Fwd Cargo Fire	25	12		3	
			013-11		Body Duct Leak	18	1		4	
14	EICAS	Hi	013-11	XA12-3	R Eng Fuel Filter Bypass	25	2		4	22

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	011-10	XA12-3	Gear Door System #1	20	12		ALL	104
			010-11		Spare	25	11			
			011-00		Spoiler Sys Fault	17	10			
			013-01		Spare	25	9			
			037-01		Main Batt Chcr Inop	21	8			
			010-10		Anti-Skid Inop	24	7			
			010-10		Auto Speed/Brake Inop	25	6			
			014-10		A/P Warn - 2 Batt	26	5			
			037-00		Altitude Alert	20	4			
			013-11		No Equip Cooling	17	3			
14	EICAS	Hi	013-11	XA12-3	Rudder PCU	16	2		ALL	104

DITS Part No.	Data Source	Word Jst	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
14	EICAS	Hi	036-10	XA12-3	R Bleed Duct PSI	28-22		7	9-3		2+4	46
↕	↕	↕	036-01	↕	L Bleed Duct PSI	28-22		7	9-3		2+4	38
↕	↕	↕	040-11	↕	L Pack Air Flow	28-21		8	10-3		1+3	46
↕	↕	↕	042-00	↕	R Pack Air Flow	28-21		8	10-3		2+4	54
↕	↕	↕	031-00	↕	L IDG Oil Temp	28-19	SR	10	12-3		1+3	54
14	EICAS	↕	031-01	XA12-3	R IDG Oil Temp	28-19	SR	10	12-3		2+4	62
15	IRU	↕	010-XX	XA12-4	Present Pos Lat MSH*	29-21	S	10	12-3		1	111
↕	↕	↕	010-XX	↕	Present Pos Lat LSH*	20-9		12	12-1		3	111
↕	↕	↕	011-XX	↕	Present Pos Long MSH*	29-21	S	10	12-3		2	111
↕	↕	↕	011-XX	↕	Present Pos Long LSH*	20-9		12	12-1		4	111
↕	↕	↕	010-XX	↕	Present Pos Lat MSH*	29-21	S	10	12-3		3	64
↕	↕	↕	010-XX	↕	Present Pos Lat LSH*	20-13		8	10-3		3	64
↕	↕	↕	011-XX	↕	Present Pos Long MSH*	29-21	S	10	12-3		3	64
15	IRU	Hi	011-XX	XA12-4	Present Pos Long LSH*	20-13		8	10-3		3	64
15	IRU	Hi	312-XX	XA10-3	Ground Speed	28-17		12	12-1		ALL	120
↕	↕	↕	315-XX	↕	Wind Speed	28-21		8	8-1		1	123
↕	↕	↕	316-XX	↕	Wind Direction	28-21	S	9	12-4		2	123
↕	↕	↕	322-XX	↕	Flt. Path Angle	28-18	S	12	12-1		3	123
15	IRU	Hi	321-XX	XA10-3	Drift Angle	28-18	S	12	12-1		4	123
1	EEC-L	Lo	341-XX	XA10-1	EPR Command Left	28-17		12	12-1		1+3	124
2	EEC-R	Lo	341-XX	XA10-2	EPR Command Right	28-17		12	12-1		2+4	124
10	FQ	Lo	247-XX	XA10-0	Total Fuel Quantity	28-16	R	12	12-1		4	118

APPENDIX D

DFDR OUTPUT DATA FRAME

Table D-1 of this appendix describes the DFDR output data frame. A non-discrete data item output to the DFDR is usually 12 bits long. In the cases where the data item is less than 12 bits, it is left justified and the unused bits on the right part of the DFDR word are used for outputting discrete parameters.

The discrete parameters can be discretes from the discrete multiplexer or discrete parameters coming from the DITS input stream (ie. single-bit DITS parameters). These discretes are identified by an asterisk (*) in the following pages.

Table D-2 describes the layout of the documentary data superframe buffer located in word 64 of subframe 3.

TABLE D-1

DFDR DATA FRAME LAYOUT

TABLE D-1

DFDR DATA FRAME LAYOUT

APPENDIX D

DFDR OUTPUT DATA FRAME

Table D-1 of this appendix describes the DFDR output data frame. A non-discrete data item output to the DFDR is usually 12 bits long. In the cases where the data item is less than 12 bits, it is left justified and the unused bits on the right part of the DFDR word are used for outputting discrete parameters.

The discrete parameters can be discretes from the discrete multiplexer or discrete parameters coming from the DITS input stream (ie. single-bit DITS parameters). These discretes are identified by an asterisk (*) in the following pages.

Table D-2 describes the layout of the documentary data superframe buffer located in word 64 of subframe 3.

DITS Part No.	Data Source	Rec Spd	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
14	EICAS	Hi	036-10	XA12-3	R Bleed Duct PSI	28-22		7	9-3		2+4	46
			036-01		L Bleed Duct PSI	28-22		7	9-3		2+4	38
			040-11		L Pack Air Flow	28-21		8	10-3		1+3	46
			042-00		R Pack Air Flow	28-21		8	10-3		2+4	54
			031-00		L IDG Oil Temp	28-19	SR	10	12-3		1+3	54
14	EICAS		031-01	XA12-3	R IDG Oil Temp	28-19	SR	10	12-3		2+4	62
15	IRU		010-XX	XA12-4	Present Pos Lat MSH*	29-21	S	10	12-3		1	111
			010-XX		Present Pos Lat LSH*	20-9		12	12-1		3	111
			011-XX		Present Pos Long MSH*	29-21	S	10	12-3		2	111
			011-XX		Present Pos Long LSH*	20-9		12	12-1		4	111
			010-XX		Present Pos Lat MSH*	29-21	S	10	12-3		3	64
			010-XX		Present Pos Lat LSH*	20-13		8	10-3		3	64
			011-XX		Present Pos Long MSH*	29-21	S	10	12-3		3	64
15	IRU	Hi	011-XX	XA12-4	Present Pos Long LSH*	20-13		8	10-3		3	64
15	IRU	Hi	312-XX	XA10-3	Ground Speed	28-17		12	12-1		ALL	120
			315-XX		Wind Speed	28-21		8	8-1		1	123
			316-XX		Wind Direction	28-21	S	9	12-4		2	123
			322-XX		Flt. Path Angle	28-18	S	12	12-1		3	123
15	IRU	Hi	321-XX	XA10-3	Drift Angle	28-18	S	12	12-1		4	123
1	EEC-L	Lo	341-XX	XA10-1	EPR Command Left	28-17		12	12-1		1+3	124
2	EEC-R	Lo	341-XX	XA10-2	EPR Command Right	28-17		12	12-1		2+4	124
10	FQ	Lo	247-XX	XA10-0	Total Fuel Quantity	28-16	R	12	12-1		4	118

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	011-10	XA12-3	Gear Door System #1	20	12		ALL	104
			010-11		Spare	25	11			
			011-00		Spoiler Sys Fault	17	10			
			013-01		Spare	25	9			
			037-01		Main Batt Cher Inop	21	8			
			010-10		Anti-Skid Inop	24	7			
			010-10		Auto Speed/Brake Inop	25	6			
			014-10		A/P Warn - 2 Batt	26	5			
			037-00		Altitude Alert	20	4			
			013-11		No Equip Cooling	17	3			
14	EICAS	Hi	013-11	XA12-3	Rudder PCU	16	2		ALL	104

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	010-00	XA12-3	STBY Bus Off	17	1		1	22
			010-01		L Fuel Filter Bypass	25	2			
			010-01		Gears Disagree	18	3			
			010-11		R AC Bus Off	19	4			
			011-00		C Left Fuel Pump Lo Press	27	5			
			011-01		R Hyd Sys Lo Press	15	6			
			011-01		Flaps (A)	19	7			
			011-01		Over Speed R ADC	22	8			
			011-01		Cabin Auto Press 1	23	9			
			012-11		L Bleed Duct Leak	16	10			
			012-11		Wheel Well Fire	18	11			
			012-11		L Yaw Damper	27	12		1	
			010-10		R Bleed Isol Valve	14	1		2	
			010-10		C Hyd Lo Press	19	2			
			011-11		C Right Fuel Pump Lo Press	16	3			
			012-00		C Hyd Lo Qty	17	4			
			012-01		Len-Hi Stage Bleed Fault	18	5			
			013-01		L Eng Ovht	14	6			
			013-01		Parking Brake (A)	18	7			
			013-01		L AC Buss Off	19	8			
			015-01		Aft Equip Smoke	15	9			
			014-11		Spoilers	19	10			
			014-11		Gear Door System #2	22	11			
			014-11		Cabin Auto Press 2	23	12		2	
			011-10		Fwd Equip Smoke	14	1		3	
			011-10		L Eng Oil Filter Bypass	26	2			
			010-01		Gears Disagree	18	3			
			012-10		Emergency Lights	24	4			
			013-00		Parking Brake (C)	25	5			
			015-00		L Aft Fuel Pump Lo Press	16	6			
			015-00		Gear Not Down	20	7			
			015-00		Fwd Equip. Over Heat	22	8			
			015-00		Fuel Crossfeed	28	9			
			015-01		R Fwd Fuel Pump Lo Press	16	10			
			015-01		APU Bleed Valve	20	11			
			015-01		Fwd Cargo Fire	25	12		3	
			013-11		Body Duct Leak	18	1		4	
14	EICAS	Hi	013-11	XA12-3	R Eng Fuel Filter Bypass	25	2		4	22

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	014-00	XA12-3	C Bleed Isol Valve	14	3		4	22
			014-00		Cabin Altitude >10K Feet	26	4			
			014-10		L Fwd Fuel Pump Lo Press	25	5			
			015-11		R Eng Hi Stage Bleed Fault	18	6			
			015-11		L Eng Bleed Ovht	19	7			
			015-11		L Hyd Sys Lo Press	26	8			
			016-01		R Eng Bleed Ovht	18	9			
			016-01		Stabilizer	22	10			
			016-01		Rat Unlocked	28	11			
			037-00		L.E. Slat Asymmetry	17	12		4	22
			014-01		Unsched Stab Movement	21	1		1	51
			015-10		R Hyd Lo Qty	17	2			
			015-10		Fwd Cargo Door	23	3			
			016-00		R Eng Oil Filter Bypass	14	4			
			016-00		L Hyd Lo Qty	17	5			
			016-10		R Bleed Duct Leak	26	6			
			016-10		R Aft Fuel Pump Lo Press	27	7			
			036-11		Aft Cargo Fire	15	8			
			013-00		L Bleed Isol Valve	14	9			
			036-11		R Eng Ovht	25	10			
			037-01		Aft Cargo Door Open	24	11			
			010-01		R Yaw Damper	27	12		1	51
			036-11		L.E. Slats Disagree	19	1		ALL	89
			015-01		R Eng Lo Oil Press	26	1		3	31
			037-00		L Eng Lo Oil Press	26	1		1	31
			010-00		L Eng T/R Unlocked	15	1		ALL	12
			010-10		R Eng T/R Unlocked	15	2			12
			011-00		L Eng T/R Deployed	28	1			8
			013-00		R Eng T/R Deployed	15	2			8
			010-01		L Eng Fire	16	2			9
			011-00		R Eng Fire	26	1			17
			015-00		APU Fire	14	2			25
14	EICAS	Hi	010-00	XA12-3	GPWC Fault	24	2		ALL	32

DITS Port No.	Data Source	Bus	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
14	EICAS	Hi	037-10	XA12-3	EICAS Fault	26	1		ALL	27
			010-00		Standby Bus Off	17	1			46
			015-01		C IRS DC Fail	13	2			46
			010-10		L IRS DC Fail	27	1			65
			012-00		R IRS DC Fail	15	1			49
			014-00		L IRS Fault	27	1			52
			015-10		R IRS Fault	15	1		ALL	55
			014-01		L Gen Drive Oil	25	2		2+4	38
			013-00		R Gen Drive Oil	18	1		ALL	53
			010-10		Capt Inst Bus Xfer	20	2		ALL	56
			014-00		FO Inst Bus Xfer	20	1		ALL	56
			010-01		L AC Bus Tie ISOL	19	2		2+4	54
			013-11		R AC Bus Tie ISOL	19	1		2+4	54
			010-10		L Gen GCB Open	18	2		ALL	57
			014-00		R Gen GCB Open	18	1		ALL	59
			011-11		C IRS DC ON	13	2		1+3	61
			013-00		L IRS DC ON	27	2		2+4	61
			016-00		R IRS DC ON	15	2		1+3	62
			010-10		DC Bus Tie	21	1		1+3	62
			014-01		L Util Bus	19	1		ALL	61
			012-11		R Util Bus	19	1		ALL	63
			037-00		Eng Low Oil Press Left	26	1		1+3	47
			015-01		Eng Low Oil Press Right	26	1		2+4	47
			015-00		Left Fwd Entry Door	23	-		-	-
			015-00		Left Eng Fuel Cut-Off	21	-		-	-
14	EICAS	HI	016-01	XA12-3	Right Eng Fuel Cut-Off	20	-		-	-

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name (* Parameter is BCD)	Data Scaling			Bit Format	Seq No.	DFDR Output	
						Input Bits	SR	No. Bits			SF	Word
14	EICAS	Hi	017-01	XA12-3	EGT L	27-17		11	12-2		ALL	35
			017-10		EGT R	27-17		11	12-2		ALL	19
			017-11		Fuel Flow Left	28-17		12	12-1		ALL	70
			020-00		Fuel Flow Right	28-17		12	12-1		ALL	121
			024-01		Fuel Press L	28-20		9	11-3		2+4	71
			024-10		Fuel Press R	28-20		9	11-3		1+3	71
			023-11		EPR Left	28-17		12	12-1		ALL	28
			024-00		EPR Right	28-17		12	12-1		ALL	44
			023-01		Vib. (IND) L	28-18		11	11-1		ALL	66
			023-10		Vib. (IND) R	28-18		11	11-1		ALL	67
			020-11		N2 Left	27-17	R	10	12-3		ALL	68
			021-00		N2 Right	27-17	R	10	12-3		ALL	69
			021-11		Oil Press L	23-17		7	9-3		ALL	72
			022-00		Oil Press R	23-17		7	9-3		ALL	73
			022-01		Oil Qty L	28-20		9	11-3		1+3	75
			022-10		Oil Qty R	28-20		9	11-3		2+4	75
			022-11		Oil Temp L	26-17	SR	10	12-3		1+3	74
			023-00		Oil Temp R	26-17	SR	10	12-3		2+4	74
			001-00		Aileron Pos. Inbd L	28-19	S	11	12-2		ALL	59
			001-01		Aileron Pos. Inbd R	28-19	S	11	12-2			27
			033-00		Rudder Position	28-20	S	10	12-3			9,25,41,57
			016-11		Elevator Pos. L	28-20	S	10	12-3			29,61
			017-00		Elevator Pos. R	28-20	S	10	12-3			13,45
			001-10		Aileron Pos L Otbd	28-19	S	11	12-2			11
			001-11		Aileron Pos R Otbd	28-19	S	11	12-2			43
			035-11		Eng. Vib. Filter 1 Left Fan	28-18		11	11-1			105
			035-01		Eng. Vib. Filter 2 Left LPT							106
			034-11		Eng. Vib. Filter 3 Left N2							107
			036-00		Eng. Vib. Filter 1 Right Fan							108
			035-10		Eng. Vib. Filter 2 Right LPT							109
			035-00		Eng. Vib. Filter 3 Right N2	28-18		11	11-1		ALL	110
			003-11		APU RPM	28-19		10	10-1		4	64
			003-00		APU EGT	28-17		12	12-1		4	64
			052-00		Gross Weight	28-14		15			3	64
			020-01		N1 Left (P&W only)	27-17		11	12-2		ALL	5
14	EICAS	Hi	020-10	XA12-3	N1 Right (P&W only)	27-17		11	12-2		ALL	37

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
12	FCC-L	Lo	272-XX	XA12-2	A/P CMD Eng C	24	4		ALL	33
					A/P CMD Eng R	25	5		ALL	33
					A/P CMD Eng L	26	6		ALL	33
					A/P CWS Eng C	27	1		ALL	37
					A/P CWS Eng R	28	1		ALL	36
					A/P CWS Eng L	29	1		ALL	11
					FMA Fault 1	13	1		ALL	33
					Land 2 (Green)	18	3			
			272-XX		Land 3 (Green)	17	2			
			274-XX		FMA Fault 2	12	7			
			274-XX		Flare Mode Oper	28	8			
			274-XX		G/S Mode Oper	29	9			
			275-XX		FMA Fault 3	21	10			
			275-XX		Loc Mode Oper	27	11			
			275-XX		Rollout Mode Oper	28	12		ALL	33
			272-XX		Flt Dir On F/O	22	1		3	23
			272-XX		Flt Dir On Capt	23	2			
			273-XX		L NAV Mode Arm	11	3			
					Flare Mode Arm	17	4			
					G/S Mode Arm	18	5			
					Rollout Mode Arm	19	6			
					B/CRS Mode Arm	20	7			
			273-XX		Loc Mode Arm	21	8			
			274-XX		Alt Mode Arm	11	9		3	
			274-XX		Flap Limit	13	1		4	
12	FCC-L	Lo	274-XX	XA12-2	Min Speed	14	2		4	23

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
12	FCC-L	Lo	272-XX	XA12-2	Test Bit 1	14	12		ALL	114
			272-XX		A/P Eng A Detent	11	11			
			273-XX		Autotrim Down Arm	29	10			
					Autotrim Down Cont	28	9			
					Autotrim Eng L Sam	27	8			
					Autotrim Eng R Sam	26	7			
					Autotrim Up Arm	25	6			
					Autotrim Up Cont	24	5			
					Autotrim Growth 1	23	4			
					Autotrim Growth 2	22	3			
					Appr Mode Arm	16	2			
					Low Mode (A/O)	15	1		ALL	114
					Appr Mode Oper	13	12		ALL	115
			273-XX		V-Nav Mode (A/O)	12	11			
			274-XX		Throttle Retard	21	10			
					IAS Mode Set	20	9			
					Mach Mode Set	19	8		ALL	115
					Mach Limit Oper	15	3		4	23
					IAS Limit Oper	16	4			
					Mach	17	5			
					IAS	18	6			
					FlCh Mode Oper	22	7			
					V-NAV Mode Oper	23	8			
					V/S Mode Oper	24	9			
					Alt Hold Mode Oper	25	10			
					T/O Mode Oper (Pitch)	26	11		4	23
			274-XX		G/A Mode Oper (Pitch)	27	12		ALL	19
			275-XX		T/O Mode Oper (Roll)	19	1		2	23
					G/A Mode Oper (Roll)	20	2			
					L-NAV Mode Oper	24	3			
					HDG Set Mode Oper	25	4			
					HDG Hold Mode Oper	26	5			
					B/CRS Mode Oper	29	6		2	23
					Trk Hold Mode Oper	23	7		ALL	115
12	FCC-L	Lo	275-XX	XA12-2	Att Hold Mode Oper	22	6		ALL	115

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	350-XX	XA11-5	SG I/O Proc No. 3	20	10		4	116
					SG I/O Proc No. 2	19	9			
					SG I/O Proc No. 1	18	8			
					SG Digital Output	17	7			
					SG Controller	16	6			
					SG Dsply Sequencer	15	5			
					SG Dsply Drive	14	4			
					SG Main Memory	13	3			
					SG Main Processor	12	2			
			350-XX		SG Overtemp	11	1		4	116
			351-XX		R-CP Fault	25	1		1+3	115
					L-CP Fault	24	1		2+4	115
					R-EHSI Overtemp	23	12		1	117
					R-EHSI Beam Fail	22	11			
					R-EHSI Anomalies	21	10			
					L-EHSI Overtemp	20	9			
					L-EHSI Beam Fail	19	8			
					L-EHSI Anomalies	18	7			
					R-EADI Overtemp	17	6			
					R-EADI Beam Fail	16	5			
					R-EADI Anomalies	15	4			
					L-EADI Overtemp	14	3			
					L-EADI Beam Fail	13	2			
11	EFIS	Lo	351-XX	XA11-5	L-EADI Anomalies	12	1		1	117

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	352-XX	XA11-5	WXR Data Fault	29	12		2	116
					MLS Data Fault	28	11			
					ILS Data Fault	27	10			
					RA Data Fault	26	9			
					R-DME Data Fault	25	8			
					L-DME Data Fault	24	7			
					R-VOR Data Fault	23	6			
					L-VOR Data Fault	22	5			
					R-ADC Data Fault	21	4			
					L-ADC Data Fault	20	3			
					R-IRS Data Fault	19	2			
					C-IRS Data Fault	18	1		2	116
					L-IRS Data Fault	17	12		3	116
					TMC Data Fault	16	11			
					R-FMC Data Fault	15	10			
					L-FMC Data Fault	14	9			
					R-FCC Data Fault	13	8			
					C-FCC Data Fault	12	7			
			352-XX		L-FCC Data Fault	11	6			
			353-XX		SG Fault	18	5			
					R EHSI Fault	17	4			
					L EHSI Fault	16	3			
					R CP Fault	15	2			
					L CP Fault	14	1		3	116
					R EADI Fault	12	12		4	116
11	EFIS	Lo	353-XX	XA11-5	L EADI Fault	11	11		4	116

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
11	EFIS	Lo	270-XX	XA11-5	Mag/TRU Data	19	1		ALL	73
					DH Alert	14	2		ALL	45
					H Alert	16	1		ALL	48
					Groundspeed Source	17	2		2+4	48
					Track Angle Source	18	2		1+3	48
			270-XX		DH + ΔH Alert	15	1		ALL	45
			274-XX		G/S Mode Oper	29	12		ALL	113
					Flare Oper	28	11			
					G/A Mode Oper-P	27	10			
					T/O Mode Oper-P	26	9			
					Alt Hold Mode Oper	25	8			
					V/S Mode Oper	24	7			
					V/Nav Mode Oper	23	6			
					FL CH Mode Oper	22	5			
					Throttle Retard	21	4			
					IAS Mode Set	20	3			
					Mach Mode Set	19	2			
					Alt Mode Oper	11	1		ALL	113
11	EFIS	Lo	274-XX	XA11-5	Pitch Speed Cntrl	10	1		ALL	104

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
8	TMC	Lo	270-XX	XA12-1	Pre-Select Climb	24	1		ALL	90
					Engine Indent 12	23	12		1	116
					11	22	11			
					10	21	10			
					9	20	9			
					8	19	8			
					7	18	7			
					6	17	6			
					5	16	5			
					4	15	4			
					3	14	3			
					2	13	2			
					Engine Indent 1	12	1		1	116
			270-XX		Temp Derate Status	11	1		ALL	29
			146-XX		Rating 2 Oper	24	12		2+4	51
					Rating 1 Oper	23	11			
					G/A Mode Oper	22	10			
					CRZ Mode Oper	21	9			
					Con Mode Oper	20	8			
					Climb Mode Oper	19	7			
					T/O Mode Oper	18	6			
					Thrust Mode Oper	13	5			
					Mach Mode Oper	12	4			
					IAS Mode Oper	11	3		2+4	51
					Flap Limit	15	1		ALL	30
					Min Speed	16	2			30
					Ground Test	17	1			83
					Idle Thrust Oper	25	3			30
					A/T Disconnect	28	5			30
			146-XX		TMC Valid	29	2			83
			145-XX		EEC/PMC Valid L	26	9			30
			145-XX		EEC/PMC Valid R	27	10			30
			146-XX		A/T Engaged	27	4			30
			146-XX		Speed Limit (G E only)	14	2			29
			145-XX		Flair Retard Mode (G E only)	28	11			30
8	TMC	Lo	145-XX	XA12-1	A/T Go Around Mode Oper (G E only)	19	6		ALL	30

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
8	TMC	Lo	145-XX	XA12-1	Air Driven Pump	24	12		ALL	103
					Wing Anti-Ice	23	11			
					Cowl Anti-Ice-R	21	10			
					Cowl Anti-Ice-L	20	9			
					Isol Valve Right	18	8			
					Isol Valve Left	17	7			
					Spare	16	6			
					Spare	15	5			
					ECS Pack R H/L	14	4			
					ECS Pack R	13	3			
					ECS Pack L H/L	12	2			
					ECS Pack L	11	1		ALL	103
					TMC V NAV Oper	29	12		ALL	30
					FLCH Mode Oper (GE only)	22	8		ALL	30
8	TMC	Lo	145-XX	XA12-1	Throttle Hold Oper (GE only)	25	7		ALL	30

DITS Port No.	Data Source	Speed	Label Code Octal	DM3 Card/Ch No.	Parameter Name	Data Scaling		Aux Output Seq No.	DFDR Output	
						Input Bits	Output Bits		SF	Word
5	ADC	Lo	350-XX	XA11-3	A to D Test	22	5		2	117
					OSPD Hdwr Test	21	4		2	117
					A/C Type Const Test	20	3		2	117
					A/C Type Prog Test	19	2		2	117
					BARO #3 Test	18	1		2	117
					BARO #2 Test	17	12		4	117
					BARO #1 Test	16	11		4	117
					T.A.T. Input Test	15	10		4	117
					R A.O.A. Vane Test	12	9		4	117
			350-XX		L A.O.A. Vane Test	11	8		4	117
			351-XX		VMO Test	27	10		3	117
					BARO #4 Test	26	9			
					EAROM Test	25	8			
					PT PLL	24	7			
					PS PLL	23	6			
					Prog Seq Test	22	5			
					Temp PS= Temp Pt	21	4			
					Average A.O.A. Test	20	3			
					PS=PT	19	2			
					A.O.A. Compare Test	18	1		3	117
					Power Supply Test	17	7		4	
					ARINC XMTR Test	16	6			
					ARINC XMTR Test	15	5			
					ARINC XMTR Test	14	4			
					PT Calib Test	13	3			
					PT Sens Temp Test	12	2			
			351-XX		PT Sens Per Test	11	1		4	117
			352-XX		A/C Type Parity	29	10		1	118
					A/C Type MSB	28	9			
					A/C Type LSB +3	27	8			
					A/C Type LSB +2	26	7			
					A/C Type LSB +1	25	6			
					A/C Type LSB	24	5			
					Spare	21	4			
					Spare	20	3			
					Spare	19	2			
5	ADC	Lo	352-XX	XA11-3	Spare	18	1		1	118

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	1	Sync Code 1107 Sync Code 2670 Sync Code 5107 Sync Code 6670	12-1	-	-	-	-	-	-
1 2 3 4	2	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	3	Magnetic Heading	12-2	-	11	-	-	VHF Key Left	3
1 2 3 4	4	Pitch Attitude	12-2	-	11	-	-	VHF Key Right	4
1 2 3 4	5	N1 Actual Left	12-2	-	1 or 14	-	-	VHF Key Ctr	5
1 2 3 4	6	Pressure Altitude (29.92)	12-1	-	5	-	-	-	-
1 2 3 4	7	T.E. Flap Position Right	12-3	16	-	L.E. Slats EX RO L.E. Slats EX R1 L.E. Slats EX RO L.E. Slats EX R1	18 20 18 20	L.E. Slats EX LO L.E. Slats EX L1 L.E. Slats EX LO L.E. Slats EX L1	17 19 17 19
1 2 3 4	8	Flap Handle Position	12-3	14	-	R T/R Deployed	14	L T/R Deployed	14

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	9	Rudder Position	12-3	-	14	Eng Fire L	14	Air/Gnd Relay	37
1 2 3 4	10	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	11	Aileron Position Left OutBd	12-2	-	14	-	-	A/P CWS Enga L	*12
1 2 3 4	12	Total Air Temp	12-3	-	5	T/R Unlock Right	14	T/R Unlock Left	14
1 2 3 4	13	Elevator Position Right	12-3	-	14	ECS Pack Left	1	H F Key Right	8
1 2 3 4	14	ILS Glideslope Deviation	12-2	-	11	-	-	H F Key Left	7
1 2 3 4	15	Lateral Acceleration	12-2	3	-	-	-	L.E.Slat Disagree	6
1 2 3 4	16	Spoiler Handle Position	12-3	15		ECSPack Right	2	Cowl Anti-ice Left	*1

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	17	Roll Attitude	12-2	-	11	-	-	Eng Fire R	#14
1 2 3 4	18	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	19	EGT Right	12-2	-	14	-	-	G/A Mode Oper. (Pitch)	#12
1 2 3 4	20	Pitch Attitude	12-2	-	11	-	-	Cowl Anti-ice Right	#2
1 2 3 4	21	Computer Airspeed (CAS)	12-1	-	5	-	-	-	-
1 2 3 4	22	EICAS Disc 1A EICAS Disc 2A EICAS Disc 3A EICAS Disc 4A	12-1	-	14 14 14 14	-	-	-	-
1 2 3 4	23	GMT Min/Sec GMT Hours FCC Disc 4A FCC Disc 2A FCC Disc 3A	12-1	-	7 7 12 12	-	-	-	-
1 2 3 4	24	Horizontal Stab. Pos.	12-3	21	-	Wing Anti-ice Left	1	Wing Anti-ice Right	#2

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	25	Rudder Position	12-3	-	14	APU Fire	14	Air/Gnd Relay	37
1 2 3 4	26	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	27	Aileron Position Right In Bd	12-2	-	14	-	-	EICAS Fault	*14
1 2 3 4	28	EPR Left (Actual)	12-1	-	14	-	-	-	-
1 2 3 4	29	Elevator Position Left	12-3	-	14	Speed Limit	*8	Temp Derate Status	*8
1 2 3 4	30	TMC Discrete	12-1	-	8	-	-	-	-
1 2 3 4	31	Lateral Acceleration	12-2	3	-	-	-	Low Oil Press L Yaw Dampler L Low Oil Press R Yaw Damper R	*14 40 *14 41
1 2 3 4	32	GPWS Warning Discretes	12-3	-	6	GPWS Fault	*14	Air Driven Pump Left	*1

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	33	FCC Disc 1A	12-1	-	12	-	-	-	-
1 2 3 4	34	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	35	EGT L	12-2	-	14	-	-	Spare	-
1 2 3 4	36	Pitch Attitude	12-2	-	11	-	-	A/P CWS Enga R	*12
1 2 3 4	37	N1 Actual Right	12-2	-	2 or 14	-	-	A/P CWS Enga C	*12
1 2 3 4	38	L Angle of Attack L Bleed Duct Psi L Angle of Attack L Bleed Duct Psi	12-2 9-3 12-2 9-3		5 14 5 14	L Gen Drive Oil L Gen Drive Oil	*14 *14	Event Marker	1
1 2 3 4	39	T.E. Flap Position Left	12-3	17	-	Isolation Valve Left	*1	AirDriven Pump Right	*2
1 2 3 4	40	Flap Handle Position	12-3	14	-	Isolation Valve Right	*2	PMC Enabled L Latched Fault L PMC Enabled R Latched Fault R	*1 *1 *2 *2

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	41	Rudder Position	12-3	-	14	-		Air/Gnd Relay	37
1 2 3 4	42	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	43	Aileron Position Right Out Bd	12-2	-	14	=	=	Capt ADC SwPos Capt EFIS SwPos Capt IRU SwPos Capt FMC SwPos	15 33 11 13
1 2 3 4	44	EPR Right (Actual)	12-1	-	14	-	-	-	-
1 2 3 4	45	Elevator Position Right	12-3	-	14	D H Alert	*11	DH+ ΔH Alert	*11
1 2 3 4	46	L Pack Air Flow R Bleed Duct Psi L Pack Air Flow R Bleed Duct Psi	10-3 9-3 10-3 9-3	-	14	Center IRS DC Fail	*14	Standby Bus Off	*14
1 2 3 4	47	Lateral Acceleration	12-2	3	-	-	-	Eng Lo Oil Pres L Eng Lo Oil Pres R Eng Lo Oil Pres L Eng Lo Oil Pres R	*14
1 2 3 4	48	Spoiler Handle Position	12-3	15	-	Trk AngleSource GrndSpd Source Trk AngleSource GrndSpd Source	*11 *11 *11 *11	H Alert	*11

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	49	Roll Attitude	12-2	-	11	-	-	Right IRS DC Fail	*14
1 2 3 4	50	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	51	EICAS Disc 5A TMC Disc ADC Discretes TMC Disc	12-1 12-3 12-2 12-3	-	14 8 5 8	- L.E.Slat Full Ex ECS Mode (L)	- 25 *1	- L.E. Slat Part Ex ECS Mode (R) EICAS Comp Sw	- 26 *2 39
1 2 3 4	52	Pitch Attitude	12-2	-	11	-	-	Left IRS Fault	*14
1 2 3 4	53	ILS Localizer Deviation	12-2	-	11			Right Gen Drive Oil	*14
1 2 3 4	54	L IDG Oil Temp R Pack Air Flow L IDG Oil Temp R Pack Air Flow	12-3 10-3 12-3 10-3	-	14	- L AC BusTie Isol - L AC BusTie Isol	- *14 - *14	- R AC Bus Tie Isol - R AC Bus Tie Isol	- *14 - *14
1 2 3 4	55	Longitudinal Acceleration	12-2	13	-	-	-	Right IRS Fault	*14
1 2 3 4	56	Horizontal Stab. Pos.	12-3	21	-	Capt Inst Bus Xfer	*14	F.O. Inst Bus Xfer	*14

SF	Word	Parameter	Sgnft Bits	Analog	Digital	Bit #2	Port #	Bit #1	Port #
1 2 3 4	57	Rudder Position	12-3	-	14	L Gen GCB Open	*14	Air/Gnd Relay	37
1 2 3 4	58	Vertical Acceleration (Normal)	12-1	1	-	-	-	-	-
1 2 3 4	59	Aileron Position Left InBd	12-2	-	14	-	-	R Gen GCB Open	*14
1 2 3 4	60	Radio Altitude	12-1	-	11	-	-	-	-
1 2 3 4	61	Elevator Position Left	12-3	-	14	C IRS DC On L IRS DC On C IRS DC On L IRS DC On	*14 *14 *14 *14	Left Util Bus	*14
1 2 3 4	62	R IDG Oil Temp R IDG Oil Temp	12-3 12-3		14 14	R IRS DC On Engine Start R R IRS DC On Engine Start R	*14 76 *14 76	DC Bus Tie Engine Start L DC Bus Tie Engine Start L	*14 75 *14 75
1 2 3 4	63	Lateral Acceleration	12-2	1	-	-	-	R Util Bus	*14
1 2 3 4	64	Frame Counter Status Doc Data * Calibration**	12-1	-	-	-	-	-	-

TABLE D-2

SUPERFRAME DOCUMENTARY DATA BUFFER LAYOUT

The table below shows the documentary data format located in Subframe 3 of Word 64

Cycle No	MSB 12 11 10 9	8 7 6 5	LSB 4 3 2 1	Contents
0	0 0 0 0	M S D	L S D	Day (2 characters)
1	0 0 0 1	M S D	L S D	Month (2 characters)
2	0 0 1 0	M S D	2nd MSD	Flight No. (4 characters)
3	0 0 1 1	3rd MSD	L S D	
4	0 1 0 0	0 0 0 0	X X X X	Leg No.
5	0 1 0 1	0 X X X	X X X X	Gross Weight - Input bits 28-22
6	0 1 1 0	X X X X	X X X X	Gross Weight - Input bits 21-14
7	0 1 1 1	0 0 S MSD	2nd MSD	Latitude (BCD)
8	1 0 0 0	0 0 0 0	3rd MSD	
9	1 0 0 1	4th MSD	L S D	
10	1 0 1 0	0 0 S MSD	2nd MSD	Longitude (BCD)
11	1 0 1 1	0 0 0 0	3rd MSD	
12	1 1 0 0	4th MSD	L S D	
13	1 1 0 1	X X X X	X X X X	Aircraft Ident. (See note below)
14	1 1 0 0	0 0 0 0	0 0 0 0	Unused (zeros)
15	1 1 1 1	0 0 0 0	0 0 0 0	

Superframe Word Counter

Note: Aircraft Ident Bit Positions are:

Bits	8	7	6	5	4	3	2	1
Input ports	95	94	93	92	99	98	97	96

APPENDIX E

PROGRAM LISTINGS

This appendix contains a partial listing of the DFDAU software. All the important routines such as interrupt services, device drivers and handlers and the executive routines are included. A list of all the global variables and constants is also included. A number of utility routines such as those performing general arithmetic and logic functions have been omitted.

The listings are arranged in an alphabetical order rather than in their functional element groups, as described in chapter 5.

```
                        1          IDT     ADCISR
                        2
                        3          SUBTTL  ADC DATA READY INTERRUPT (INT 6)
                        4
                        5  *       CALLING SEQ:    INTERRUP 6
                        6
                        7  *-------------------------------------------------------------+
                        8  *                                                             !
                        9  *       ADCISR ACCORDING TO A GIVEN FLAG (DAS-FLAG) BRANCHES TO
                       10  *       ANALOG, CALIBRATION OR BITE DATA ACQUISITION.
                       11  *                                                             !
                       12  *-------------------------------------------------------------+
                       13  *       VERSION : 2
                       14  *       PROGRAMMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY : N.CONSTANTINIDES
                       16
                       17
                       18          INTERN  ADCISR
                       19  * REFERD MODULES:
                       20  *               ANACQ
                       21  *               CALAQ
                       22  *               [illegible]
                       23  * GLOBAL (ROMV)
                       24          EXTERN  ADCBDV          AN, CAL, BT DATA ACQ VECTORS ADDR
                       25          EXTERN  RTSIN1
                       26
       =0009           27  DASFLG  EQU     R9      =       DAS-FLAG (INPUT & OUTPUT)
                       28          INCLUDE ENCLOS
                       30  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       32          INCLUDE REGDEF          REGISTER DEFENITIONS
                       51          INCLUDE CNSTNT          CONSTANTS
                      179          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      480          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      710          INCLUDE JMPMAC          JUMP MACROS
                      744          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      759          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
       =0000         1135          RSECT   ADCISR
                     1136  ********************************************************************************
0000'                1137  ADCISR
0000'  06A0 0002*    1138          BL      @RTSIN1
                     1139+         RESET   INT6,                   RESET ADC-INT
0004'  04E0 FFB8     1142A           CLR   @CADCLR
                     1150+         DOIF    DASFLG,GE,,,            IF DAS-FLAG >= 0 AND < 3,
0008'  C249          1284B             MOV  DASFLG,DASFLG
000A'  1108          1326E              JLT  91$
                     1456+         ANDIF   DASFLG,LEQ,C2,I,
000C'  0289 0002     1467B             CI   DASFLG,C2
0010'  1505          1527E              JGT  91$
0012'  C049          1653              MOV   DASFLG,R1               GET DAS-FLAG AS SUBR-TABLE OFFSET
0014'  0A21          1654              SLA   R1,C2
                     1655  *                                        DOCASE
                     1656  *                                          FOR DASFLG = 0, 1 AND 2  CALLS
```

```
                    1657  *                                        ANACQ, CALAQ AND BTACQ.
                    1658+           CALLWP  ADCBDV(R1)            ANACQV,CALAQV,& BTAQV
    0016' 0421 0001*    1660A       BLWP  @ADCBDV(R1)
                    1662  *                                          DOEND
                    1663+         ELSEDO                          ELSE
    001A' 1000          1924E           JMP  92$
    001C'               2054E 91$
                        2139+         ENDBLK                      IFEND
    001C'               2275E 92$
    001C' 0380          2359          RTWP
                        2360  ********************************************************************************
                        2361
                        2362
                        2363          END
```

```
                     1          IDT     ANACQ
                     2          SUBTTL  ACQ AND STORE ANALOG DATA
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: ANACQ.SRC                        AUTH: N.CONSTANTINIDES  *
                     6  * VERSION: 2                             DATE: 24-OCT-1981       *
                     7  *                                                                  *
                     8  * FUNCTION: READS THE REQUESTED ANALOG DATA, CALCULATES THE       *
                     9  *           TRANSDUCER VALUE, PUTS IT INTO DESTINATION BUFFER     *
                    10  *           AND THEN SENDS ANALOG MUX ADDRESS FOR NEXT ADC        *
                    11  *           CONVERSION.                                           *
                    12  *                                                                  *
                    13  * CALLING MODULES: ADCISR                                         *
                    14  *                                                                  *
                    15  * CALLING SEQ: CALLWP @ANACQ                                      *
                    16  *                                                                  *
                    17  * INPUTS: R8    =       ANALOG PARAM COUNT                        *
                    18  *         R9    =       ANALOG TABLES OFFSET                      *
                    19  *        R10    =       ANALOG ARRAY OFFSET                       *
                    20  *                                                                  *
                    21  * OUTPUTS: R9   =       ANALOG TABLES OFFSET                      *
                    22  *         R10   =       ANALOG [illegible]Y OFFSET                *
                    23  *                                                                  *
                    24  * MODULES REFERENCED: ANREAD,ANVT,ANSA,DSTINE                     *
                    25  *                                                                  *
                    26  * WORKSPACE AREA:                                                 *
                    27  *                                                                  *
                    28  * REGISTERS MODIFIED: R2,R3,R9,R10                                *
                    29  *                                                                  *
                    30  * VERSION HISTORY:                                                *
                    31  *                                                                  *
                    32  ********************************************************************
=0000               33          RSECT   ANACQ
                    34  *** CALL NAME
                    35          INTERN  ANACQ
                    36  *** VARIABLES REFERENCED
                    37          EXTERN  ANSFLG
                    38          EXTERN  CYFFRC,CNVFLG
                    39          EXTERN  DASFLG
                    40  *** CONSTANTS REFERENCED
                    41          EXTERN  D1
                    42          EXTERN  D2
                    43  *** TABLES REFERENCED
                    44          EXTERN  ANDT,ANDOA,ANPSTB
                    45  *** MODULES REFERENCED
                    46          EXTERN  ANREAD
                    47          EXTERN  ANPST
                    48          EXTERN  ANVT
                    49          EXTERN  ANSA
                    50          EXTERN  DSTINE
                    51          EXTERN  BTACQV
                    52          EXTERN  CALAQV
```

```
                     53  *** LIBRARY
                     54          INCLUDE ENCLOS
                     56  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     58          INCLUDE REGDEF          REGISTER DEFENITIONS
                     77          INCLUDE CNSTNT          CONSTANTS
                    205          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    506          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    736          INCLUDE JMPMAC          JUMP MACROS
                    770          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    785          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1161  *** REGISTERS DEFINITION
                   1162  *               R2      =       SCRATCH
                   1163  *               R3      =       SCRATCH
                   1164  *      THE FOLLOWING REG ARE SAME PARAM FOR THE WHOLE AN-DATA-ACQ
=0007              1165  ANOA    EQU     R7      =       ANALOG OFFSET ARRAY
=0008              1166  ANPCNT  EQU     R8      =       ANALOG PARAM COUNT (INPUT)
=0009              1167  ANTOF   EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
=000A              1168  ANARO   EQU     R10     =       ANALOG ARRAY OFFSET (INPUT; OUTPUT)
                   1169  *
```

```
                    1171  ********************************************************************
0000' C020 0001*    1172  ANACQ   MOV     @ANSFLG,R0              AN SYNCRO OR AC DUMMY READING FLAG
0004' 1607          1173          JNE     40$                     JIF NOT SYNCHRO OR AC TYPE
0006' 0720 0001*    1174          SETO    @ANSFLG                 OTHERWISE IGNORE 1ST INTERRUPT
                    1175  *                                       (INVALID DATA)
                    1176+         RESET   INT6,                   RESET ADC INTERRUPT
000A' 04E0 FFB8     1179A           CLR   @CADCLR
                    1187+         STARTI  INT6,                   START ADC INTERRUPT
000E' 04E0 FFB0     1189A           CLR   @CADCDB
0012' 102F          1191          JMP     60$                     EXIT
                    1192+ 40$     CALL    ANREAD,                 READ (VMTS AND VMTR) AN-DATA
0014' 06A0 000A*    1199A           BL    @ANREAD
                    1201
                    1202+         CALL    ANVT                    GET XDUCER VALUE
0018' 06A0 000C*    1209A           BL    @ANVT
                    1211  *** POSITION AND ROUND IF REQUESTED
001C' 0202 0009*    1212          LI      R2,ANPSTB               AN-POS-TABLE
0020' 06A0 000B*    1213          BL      @ANPST                  POSITION & ROUND
                    1214  *                                       GET AN-DS-TABLE, AN-DS-OFFSET-ARRAY
                    1215  *                                       ADDR AND PUT DATA INTO DEST
                    1216+         CALL    DSTINE,<R2,=,ANDOA,I>,<R3,=,ANDT,I>
0024' 0202 0008*    1226C           LI    R2,ANDOA
0028' 0203 0007*    1242D           LI    R3,ANDT
002C' 06A0 000E*    1257A           BL    @DSTINE
                    1259
                    1260+         CALL    ANSA                    SEND AN ADDR
0030' 06A0 000D*    1267A           BL    @ANSA
0034' 151E          1269          JGT     60$             JIF STILL ACQUIRING ANALOG DATA
0036' 131D          1270          JEQ     60$
                    1271  ***
0038' C020 0002*    1272          MOV     @CYFFRC,R0              CYCLE PER FRAME COUNTER (0-31)
003C' 0240 0007     1273          ANDI    R0,C7                   EXTRACT LS 4 BITS
0040' 0280 0004     1274          CI      R0,C4                   CYCLE 4?
0044' 160A          1275          JNE     50$                     JIF NOT
0046' C020 0002*    1276          MOV     @CYFFRC,R0              RESTORE R0
                    1277
                    1278  *** ANALOG BITE VOLTAG READINGS ONCE PER SEC
                    1279
004A' C820 0006*    1280          MOV     @D2,@DASFLG             SELECT NEXT BITE INTERRUPT (ISR) VIA ADCISR
004E' 0004*
0050' 04E0 0003*    1281          CLR     @CNVFLG                 JUMP OFFSET IN BTACQ
0054' 0420 000F*    1282          BLWP    @BTACQV                 INITIATE FIRST BITE
0058' 100C          1283          JMP     60$
                    1284  ***
005A' 0240 0007     1285  50$     ANDI    R0,C7                   SAVE SUBFRAME CYCLE COUNT ONLY
005E' 0280 0003     1286          CI      R0,C3                   5TH THRU 8TH CYCLE OF EVERY S/F?
0062' 1607          1287          JNE     60$                     JIF NOT
                    1288  *** ANALOG CAL DATA READINGS, 4 TIMES PER SEC
                    1289          MOV     @D1,@DASFLG             SELECT NEXT CAL INTERRUPT (ISR) VIA ADCISR
0064' C820 0006*
0068' 0004*
006A' 04E0 0003*    1290          CLR     @CNVFLG                 JUMP OFFSET IN CALAQ
006E' 0420 0010*    1291          BLWP    @CALAQV                 INITIATE FIRST CAL DATA
```

```
0072'  0380        1292   60$     RTWP
                   1293   $
                   1294           END
```

```
                    26
                    27          INCLUDE ENCLOS
                    29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    31          INCLUDE REGDEF          REGISTER DEFENITIONS
                    50          INCLUDE CNSTNT          CONSTANTS
                   178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   709          INCLUDE JMPMAC          JUMP MACROS
                   743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
      =0000       1134          RSECT   ANDRQ
                  1135  ********************************************************************************
0000'             1136  ANDRQ
                  1137  *** CLEAR MUX ADDRESS FIELD ONLY
0000'  04C0       1138          CLR     MXPT
                  1139+         CRUWRT  ANMCA1,MXPT,ZERO        CLEAR 300-30F
0002'  020C 0600  1141A           LI    CRU,ANMCA1
0006'  3000       1142A           LDCR  MXPT,ZERO
                  1144+         CRUWRT  ANMCA2,MXPT,C8
0008'  020C 0620  1146A           LI    CRU,ANMCA2
000C'  3200       1147A           LDCR  MXPT,C8
000E'  C009       1149          MOV     APT,MXPT        SAVE OFFSET
0010'  0A30       1150          SLA     MXPT,C3         OFFSET = OFFSET * 8 TO GET CORRECT MUX OFFSET (8 BYTES PER MUX VALUE

0012'  A001       1151          A       MTBL,MXPT       GET ADDR OF MUX
                  1152
                  1153+         CRUWRT  ANMCA1,*MXPT+,ZERO      SET 300-30F
0014'  020C 0600  1155A           LI    CRU,ANMCA1
```

```
                         1          IDT     ANDRQ
                         2
                         3          SUBTTL  REQUEST ANALOG DATA (SEND-MUX-ADDR)
                         4
                         5  *       CALLING SEQ:    CALL    @ANDRQ
                         6
                         7  *---------------------------------------------------------------+
                         8  *                                                               !
                         9  *       ANDRQ GETS THE POINTER TO THE MUX-ADDR-TABLE AND THEN
                        10  *       SENDS (EIGHT BYTES) MUX ADDRESSES   TO  RECEIVE  NEXT
                        11  *       ANALOG DATA.
                        12  *
                        13  *       CALLING MODULE : ANSA                                   !
                        14  *---------------------------------------------------------------+
                        15  *       VERSION : 1
                        16  *       PROGRAMMED BY : N.CONSTANTINIDES
                        17  *       CHECKED BY : N.CONSTANTINIDES
                        18
                        19
                        20          INTERN  ANDRQ
                        21
                        22
        =0000           23  MXPT    EQU     R0      =       SCRATCH
        =0001           24  MTBL    EQU     R1      =       MUX-ADDR-TBL (INPUT)
        =0009           25  APT     EQU     R9      =       MUX-OFFSET (INPUT)
                        26
                        27          INCLUDE ENCLOS
                        29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                        31          INCLUDE REGDEF          REGISTER DEFENITIONS
                        50          INCLUDE CNSTNT          CONSTANTS
                       178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                       479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                       709          INCLUDE JMPMAC          JUMP MACROS
                       743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                       758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000         1134          RSECT   ANDRQ
                      1135  ********************************************************************************
    0000'             1136  ANDRQ
                      1137  *** CLEAR MUX ADDRESS FIELD ONLY
    0000'  04C0       1138          CLR     MXPT
                      1139+         CRUWRT  ANMCA1,MXPT,ZERO        CLEAR 300-30F
    0002'  020C 0600  1141A           LI    CRU,ANMCA1
    0006'  3000       1142A           LDCR  MXPT,ZERO
                      1144+         CRUWRT  ANMCA2,MXPT,C8
    0008'  020C 0620  1146A           LI    CRU,ANMCA2
    000C'  3200       1147A           LDCR  MXPT,C8
    000E'  C009       1149          MOV     APT,MXPT        SAVE OFFSET
    0010'  0A30       1150          SLA     MXPT,C3         OFFSET = OFFSET * 8 TO GET CORRECT MUX OFFSET (8 BYTES PER MUX VALUE
)
    0012'  A001       1151          A       MTBL,MXPT       GET ADDR OF MUX
                      1152
                      1153+         CRUWRT  ANMCA1,*MXPT+,ZERO      SET 300-30F
    0014'  020C 0600  1155A           LI    CRU,ANMCA1
```

REQUEST ANALOG DATA (SEND-MUX-ADDR) ANDRQ.SRC

```
001B' 3030        1156A          LDCR  *MXPT+,ZERO
                  1158+        CRUWRT  ANMCA2,*MXPT+,ZERO      SET 310-31F
001A' 020C 0620   1160A          LI    CRU,ANMCA2
001E' 3030        1161A          LDCR  *MXPT+,ZERO
                  1163+        CRUWRT  ANMCA3,*MXPT+,ZERO      SET 320-32F
0020' 020C 0640   1165A          LI    CRU,ANMCA3
0024' 3030        1166A          LDCR  *MXPT+,ZERO
                  1168+        CRUWRT  ANMCA4,*MXPT+,C8        SET 330-337
0026' 020C 0660   1170A          LI    CRU,ANMCA4
002A' 3230        1171A          LDCR  *MXPT+,C8
                  1173+        CRUWRT  ANMCA5,*MXPT+,C8        SET 348-34F
002C' 020C 0690   1175A          LI    CRU,ANMCA5
0030' 3230        1176A          LDCR  *MXPT+,C8
0032' 045B        1178           RT
                  1179  ********************************************************************************
                  1180  *PRIVATE CONSTANTS
      =0600       1181  ANMCA1  EQU    >300*2  AN MUX CRU ADDR FOR 1ST+2ND BYTES (300-30F H)
      =0620       1182  ANMCA2  EQU    >310*2  AN MUX CRU ADDR FOR 3RD+4TH BYTES (310-31F H)
      =0640       1183  ANMCA3  EQU    >320*2  AN MUX CRU ADDR FOR 5TH+6TH BYTES (320-32F H)
      =0660       1184  ANMCA4  EQU    >330*2  AN MUX CRU ADDR FOR 7TH BYTE (330-337 H)
      =0690       1185  ANMCA5  EQU    >348*2  AN MUX CRU ADDR FOR 8TH BYTE (348-34F H)
                  1186
                  1187          END
```

```
                         1          IDT     ANPST
                         2          SUBTTL  ANALOG POSITION MODULE
                         3  ********************************************************************
                         4  *                                                                  *
                         5  * NAME: ANPST.SRC                      AUTH: N.COSTANTINIDES      *
                         6  * VERSION: 1                           DATE: 4-JUN-1982           *
                         7  *                                                                  *
                         8  * FUNCTION: RIGHT JUSTIFIES ANALOG DATA AND ROUNDS                 *
                         9  *          OFF LSB IF SELECTED.                                    *
                        10  *                                                                  *
                        11  * CALLING MODULES: ANACQ                                          *
                        12  *                                                                  *
                        13  * CALLING SEQ: BL @ANPST                                          *
                        14  *                                                                  *
                        15  * INPUTS: R2    =       ANL-POS-TBL START ADDRESS                 *
                        16  *         R4    =       ANALOG INPUT DATA                         *
                        17  *         R9    =       DATA ITEM IN ANL-TBL-OFFSET-ARR-TBL       *
                        18  *                                                                  *
                        19  * OUTPUTS:      R4=RIGHT JUSTIFIED DATA ROUNDED                   *
                        20  *                                                                  *
                        21  * MODULES REFERENCED: NONE                                        *
                        22  *                                                                  *
                        23  * WORKSPACE AREA: CALLER'S                                        *
                        24  *                                                                  *
                        25  * REGISTERS MODIFIED: R0,R2,R4                                    *
                        26  *                                                                  *
                        27  * VERSION HISTORY:                                                *
                        28  *                                                                  *
                        29  ********************************************************************
       =0000            30          RSECT   ANPST
                        31  *** CALL NAME
                        32          INTERN  ANPST
                        33  *** VARIABLES REFERENCED
                        34  *** CONSTANTS REFERENCED
                        35  *** TABLES REFERENCED
                        36  *** MODULES REFERENCED
                        37  *** LIBRARY
                        38  *** REGISTERS DEFINITION
                        39  ********************************************************************
   0000'                40  ANPST
   0000'  A089         41          A       R9,R2           =OFFSET+POS-TBL START ADDRESS
                        42  ***  RIGHT JUSTIFY 12 BIT INPUT DATA TO?
   0002'  D012         43          MOVB    *R2,R0          DATA SET IN ANL-POS-TBL
   0004'  06C0         44          SWPB    R0              R.J. DATA SET
   0006'  0240 0003    45          ANDI    R0,3            SAVE 2 BITS OF INPUT POSITION
   000A'  1301         46          JEQ     10$             JIF NO ADJUSTMENT NEEDED
   000C'  0904         47          SRL     R4,0            ELSE R.J. INPUT PER INPUT POSITION
                        48  ***  ROUNDING (OPTIONAL)
   000E'  D012         49  10$     MOVB    *R2,R0          DATA SET IN AN-POS-TBL
   0010'  1503         50          JGT     20$             JIF NO ROUNDING
   0012'  1302         51          JEQ     20$
   0014'  0584         52          INC     R4              ELSE BUMP LSB
```

```
0016'  0914          53          SRL     R4,1          GET RID OF LSB
                     54  ###
0018'  045B          55  20$     RT
                     56  ##########################################################
                     57          END
```

No errors detected

```
                      1          IDT     ANREAD
                      2
                      3          SUBTTL  READ ANALOG DATA
                      4
                      5  *       CALLING SEQ:    CALL    @ANREAD
                      6
                      7  *-------------------------------------------------------------------+
                      8  *                                                                   !
                      9  *       ANREAD READS THE FIRST RECEIVED ANALOG 14-BIT IOB DATA,
                     10  *       GETS ANALOG DATA TYPE, IF DOUBLE CONVERSION (ACCORDING
                     11  *       TO THE TYPE), READS THE SECOND ANALOG DATA AND RETURNS.
                     12  *                                                                   !
                     13  *-------------------------------------------------------------------+
                     14  *       VERSION : 1
                     15  *       PROGRAMMED BY : M.CONSTANTINIDES
                     16  *       CHECKED BY : M.CONSTANTINIDES
                     17
                     18          INTERN  ANREAD
                     19  * REFERD MODULES:
                     20          EXTERN  IOB2SC
                     21  * GLOBAL AREA:
                     22  *       (ROMD)
                     23          EXTERN  ANTYPT          AN-TYPE-TABLE
                     24
        =0002        25  ATYP    EQU     R2      =       ANALOG DATA TYPE (OUTPUT)
        =0003        26  FAND    EQU     R3      =       1ST ANALOG DATA (OUTPUT)
        =0004        27  SAND    EQU     R4      =       2ND ANALOG DATA (IF NOT DUAL, = 0) (OUTPUT)
        =0009        28  ATO     EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
                     34          INCLUDE REGDEF          REGISTER DEFENITIONS
                     53          INCLUDE CNSTNT          CONSTANTS
                    161          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    482          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    712          INCLUDE JMPMAC          JUMP MACROS
                    746          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    761          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000       1138          RSECT   ANREAD
                    1139  ********************************************************************
   0000'            1140  ANREAD
   0000'  C80B 0000'1141          MOV     LINK,@LINKZ             SAVE LINKER
                    1142  *                                       GET RECEIVED 14-BIT AN-DATA
                    1143+         CALL    IOB2SC,<R4,=,@CADCDB>    AND CONVET TO 12-BIT 2S COMP
   0004'  C120 FFB0 1159C           MOV     @CADCDB,R4
   0008'  06A0 0001* 1167A          BL      @IOB2SC
   000C'  C0C4      1169          MOV     SAND,FAND               1ST-READING =  ANALOG DATA
   000E'  04C4      1170          CLR     SAND                    2ND-READING = 0.
                    1171
                    1172+         MOVBRJ  @ANTYPT(ATO),ATYP,      GET THE TYPE RIGHT JUSTIF
   0010'  D0A9 0002* 1174A          MOVB    @ANTYPT(ATO),ATYP
   0014'  0982      1175A          SRL     ATYP,8
                    1177
                    1178+         DOIF    ATYP,GE,5,X,,           IF TYPE >= 5, DUAL CONVERSION
   0016'  0282 0005 1297B           CI      ATYP,5
```

```
001A' 1104          1354E              JLT   91$
                    1484  *                                          READ 14-BIT AN-DATA AS 12-BIT
                    1485+               CALL  IOB2SC,<R4,=,@CADCDB>  2S-COMP (2ND READING)
001C' C120 FFB0     1501C               MOV   @CADCDB,R4
0020' 06A0 0001$    1509A               BL    @IOB2SC
                    1511+             ENDBLK                         IFEND
0024'               1643E  91$
0024' C2E0 0000'    1731              MOV     @LINKZ,LINK            RESTORE LINKER
0028' 045B          1732              RT
                    1733  ********************************************************************************
                    1734  * PRICVATE AREA
                    1735+             LOCR    PRIV,LINKZ             LINK-SAVE AREA
0000' =0002         1738A LINKZ       BSS     2
                    1739
                    1740              END
```

```
                     1          IDT    ANSA
                     2
                     3          SUBTTL  SEND NEXT ANALOG ADDRESS
                     4
                     5  *       CALLING SEQ:    CALL    @ANSA
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       IF THERE ARE MORE ANLOG DATA TO ACQUIRE, ANSA GETS THE NEXT
                    10  *       AN-TABLES-OFFSET AND SEND MUX ADDRESSES TO  RECEIVE NEXT
                    11  *       DATA, ELSE IT SETS FLAG FOR NO MORE AND RETURNS.
                    12  *                                                                   !
                    13  *       CALLED BY RTIDA AND ANACQ
                    14  *-------------------------------------------------------------------+
                    15  *       VERSION : 2
                    16  *       PROGRAMMED BY : M.CONSTANTINIDES
                    17  *       CHECKED BY : M.CONSTANTINIDES
                    18
                    19          INTERN  ANSA
                    20  * REFERD MODULES:
                    21          EXTERN  ANDRQ
                    22          EXTERN  TBLOFS
                    23  * GLOBAL AREA:
                    24          EXTERN  ANMUXT          AN-MUX-ADDR-TABLE (ROM)
                    25          EXTERN  DASFLG          DAS-FLAG (RAM)
                    26          EXTERN  ANSFLG          AN SYNCHRO FLAG
                    27
                    28  *               R0      =       SCRATCH
                    29  *               R1      =       SCRATCH
=0007               30  ANDA    EQU     R7      =       AN-OFFSET-ARRAY (INPUT; OUTPUT)
=0008               31  ANPCNT  EQU     R8      =       ANALOG PARAM COUNT (INPUT)
=0009               32  ANTOF   EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
=000A               33  ANARO   EQU     R10     =       ANALOG OFFSET POINTER (INPUT; OUTPUT)
                    34          INCLUDE ENCLOS
                    36  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    38          INCLUDE REGDEF          REGISTER DEFENITIONS
                    57          INCLUDE CNSTNT          CONSTANTS
                   185          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   486          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   716          INCLUDE JMPMAC          JUMP MACROS
                   750          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   765          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1141          RSECT   ANSA
                  1142  ********************************************************************************
```

```
0000'                1144  ANSA
0000'  C0CB          1145        MOV     LINK,R3         SAVE LINKER
                     1146  *                                 R2 = AN-VERS TABLE AND
                     1147  *                                 FIND NEXT AN-TABLES-OFFSET
                     1148+       CALL    TBLOFS
0002'  06A0 0002*    1155A         BL    @TBLOFS                                  .
                     1157
                     1158+       DOIF    ANPCNT,GE,,,            IF AN-PARAM-COUNT >= 0 (MORE TO GET),
0006'  C208          1292B           MOV   ANPCNT,ANPCNT
0008'  110A          1334E             JLT   91$
                     1464+           CALL  ANDRQ,<R1,=,ANMUXT,I>     R1 = MUX-TBL ADDR AND SEND AN ADDR
000A'  0201 0003*    1474C           LI    R1,ANMUXT
000E'  06A0 0001*    1488A           BL    @ANDRQ
0012'  04E0 0005*    1490        CLR     @ANSFLG         SET FLAG TO READ ANALOG DATA TWICE IN ANACQ
                     1491+       RESET   INT6,                   RESET INTERRUPT
0016'  04E0 FFB8     1494A         CLR   @CADCLR
                     1502+       STARTI  INT6,                   START ADC INTERRUPT
001A'  04E0 FFB0     1504A          CLR  @CADCDB
                     1506+          ENDBLK
001E'                1638E  91$
001E'  C2C3          1726        MOV     R3,LINK                 RESTORE LINKER
0020'  C208          1727        MOV     ANPCNT,ANPCNT           FOR CALLER TO CHECK END OF ACQ FOR THIS CYCLE
0022'  045B          1728        RT
                     1729  ********************************************************************************
                     1730        END
```

```
                        1          IDT     ANP5VR
                        2
                        3          SUBTTL  5V. REF. POTENTIOMETER
                        4
                        5  *       CALLING SEQ:    CALL    @ANP5VR
                        6
                        7  *-------------------------------------------------------------------------------
                        8  *                                                                              !
                        9  *       ANP5VR SAVES THE RECEIVED 5V. REF. POT. VALUE FOR FUTURE USE
                       10  *       (BY "ANPOT").
                       11  *                                                                              !
                       12  *-------------------------------------------------------------------------------
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  ANP5VR
                       18  * GLOBALS:
                       19          EXTERN  VCPR            5V. POT. REF.
                       20
        =0003          21  VCTS    EQU     R3      =       AN 5V REF POT DATA (INPUT; OUTPUT)
                       22
                       28          INCLUDE REGDEF          REGISTER DEFENITIONS
                       47          INCLUDE CNSTNT          CONSTANTS
                      180          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      481          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      711          INCLUDE JMPMAC          JUMP MACROS
                      745          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      760          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1137          RSECT   ANP5VR
                     1138  ********************************************************************************
0000'                1139  ANP5VR
0002'  C803 0001*    1140          MOV     VCTS,@VCPR      SAVE 5V REF POT VALUE
0004'  0458         1141          RT
                     1142  ********************************************************************************
                     1143
                     1144          END
```

```
                        1          IDT     ANOSCL
                        2
                        3          SUBTTL  ANALOG WITH NO SCALING
                        4
                        5  *       CALLING SEQ:    CALL    @ANOSCL
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                     !
                        9  *       ANOSCL GETS THE ANALOG VALUE FOR THE ANALOG SIGNAL TYPES
                       10  *       WITH NO SCALING (LDC, ...).
                       11  *                                                                     !
                       12  *---------------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  ANOSCL
                       18
          =0003        19  VTI     EQU     R3      =       AN VALUE (INPUT)
          =0004        20  VTO     EQU     R4      =       AN VALUE (OUTPUT)
                       21
                       27          INCLUDE REGDEF          REGISTER DEFENITIONS
                       46          INCLUDE CNSTNT          CONSTANTS
                      179          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      480          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      710          INCLUDE JMPMAC          JUMP MACROS
                      744          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      759          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000      1136          RSECT   ANOSCL
                     1137  ********************************************************************************
     0000'           1138  ANOSCL
     0000' C103      1139          MOV     VTI,VTO         NO SCALING, R4 = ANSW
     0002' 045B      1140          RT
                     1141  ********************************************************************************
                     1142
                     1143          END
```

```
                        1          IDT     ANPOT
                        2
                        3          SUBTTL  POTENTIOMETER
                        4
                        5  *       CALLING SEQ:    CALL    @ANPOT
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                     !
                        9  *       ANPOT CALCULATES VT = (VCTS / VCPR) * 4096.
                       10  *       TO AVOID LOSS OF DATA IT PREFORMS FIRST (VCTS * 4096).
                       11  *       VCPR : EXCITATION-VOLTAGE, VCTS : TRANCDUCER READING
                       12  *       AND VT : THE RATIO.
                       13  *                                                                     !
                       14  *---------------------------------------------------------------------+
                       15  *       VERSION : 1
                       16  *       PROGRAMED BY : N.CONSTANTINIDES
                       17  *       CHECKED BY : N.CONSTANTINIDES
                       18
                       19          INTERN  ANPOT
                       20  * GLOBALS:
                       21          EXTERN  VCPR            5V. POT. REF. (RAM) (INPUT)
                       22
       =0003           23  VCTP    EQU     R3      =       AN-DATA (INPUT; OUTPUT(=VT))
       =0004           24  TMP     EQU     R4      =       SCRATCH
                       25
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      183          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      484          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      714          INCLUDE JMPMAC          JUMP MACROS
                      748          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      763          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
       =0000         1140          RSECT   ANPOT
                     1141  ********************************************************************************
0000'                1142  ANPOT
                     1143+       DOIF      @VCPR,LEQ,VCTP          IF VCPR =< VCTS
0000' 8CE0 0001*     1281B            C     @VCPR,VCTP
0004' 1503           1322E              JGT   91$
0006' 0203 0FFF      1449            LI      VCTP,C4095              VT = 4095
                     1450+       ELSEDO                            ELSE
000A' 1006           1711E              JMP   92$
000C'                1841E  91$
000C' C103           1926            MOV     VCTP,TMP
                     1927+           DOIF    ,NE,,                   IF VCTS NOT= 0,
000E' 1304           2102E              JEQ   93$
0010' 0843           2233             SRA   VCTP,C4                  VT = VCTS * 4096
0012' 0AC4           2234             SLA   TMP,C12
0014' 3CE0 0001*     2235             DIV   @VCPR,VCTP               VT = VCTS / VCPR
                     2236+           ENDBLK                          IFEND
0018'                2376E  93$
                     2456+       ENDBLK                            IFEND
0018'                2592E  92$
```

```
0018' 045B          2676          RT
                    2677  ********************************************************************************
                    2678
                    2679          END
```

No errors detected

```
                    1          IDT     ANTBL
                    2          SUBTTL  ANALOG TABLES
          =0000     3          RSECT   ANTBL
                    4  ********************************************************************************
                    5  *                                                                              *
                    6  *  NAME: ANTBL.SRC (BA)                AUTH: N. CONSTANTINIDES                *
                    7  *  VERSION:                            DATE: 21-JUN-1983                      *
                    8  *                                                                              *
                    9  *  FUNCTION: ANALOG TABLES SECTION CONTAINS THE PARAMETER OFFSET ARRAY,        *
                   10  *            PARAMETER QUANTITY PER CYCLE TABLE, XDUCER MUX VALUES,            *
                   11  *            PARAMETER MASK TABLE, PARAMETER TYPE, PARAMETER DEST. TABLE       *
                   12  *            OFFSET, PARAMETER DESTINATION AND ANALOG DATA POSITION.           *
                   13  *                                                                              *
                   14  *  REFERENCED BY:                                                              *
                   15  *                                                                              *
                   16  *  DATA POSITION FORMAT:                                                       *
                   17  *  BIT         7  6  5  4  3  2  1  0                                         *
                   18  *              N3 -- N2 N2 N2 N2 N1 N1                                         *
                   19  *                                                                              *
                   20  *  WHERE:        N1 = INPUT POSITION                                           *
                   21  *                N2 = INPUT MASK WORD INDEX                                    *
                   22  *                N3 = ROUNDING (OPTIONAL)                                      *
                   23  *                                                                              *
                   24  *  PARAMETER DESTINATION FORMAT:                                               *
                   25  *  BIT         15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0           *
                   26  *              N5  N4  N3  N3  N3  N3  N2 N2 N1 N1 N1 N1 N1 N1 N1 N1          *
                   27  *                                                                              *
                   28  *  WHERE:        N1 = BUFFER OFFSET WORD                                       *
                   29  *                N2 = BUFFER TYPE                                              *
                   30  *                N3 = BUFFER POSITION LEFT SHIFT COUNT                         *
                   31  *                N4 = CLEAR OUTPUT WORD (OPTIONAL)                             *
                   32  *                N5 = END OF DATA SET (OPTIONAL)                               *
                   33  *                                                                              *
                   34  *  VERSION HISTORY:                                                            *
                   35  *                                                                              *
                   36  ********************************************************************************
                 1228  *
                 1229          INTERN  ANDFAR,ANFQT,ANMUXT,ANPRQ,ANMSKT,ANTYPT,ANDOA,ANDT
                 1230          INTERN  CALMAT,CALMAQ,CAL1FT,CAL2FT,CALSOT,CALLT
                 1231          INTERN  ANPSTB,CALULT,ANCIT,ANSYNT
                 1232  *
```

```
                    1234  ********************************************************************************
                    1235  *
                    1236  *              ANALOG TABLES
                    1237  *              *************
                    1238  *
                    1239  *       ANALOG PARAMETER OFFSET ARRAY
                    1240  *
0000'               1241  ANOFAR
                    1242  *
                    1243  *  SUBFRAME 1 CYCLES 1 TO 8
                    1244  *
0000'  00 06 04     1245  AA11    BYTE    0,6,4
0003'  00 01 07     1246  AA12    BYTE    0,1,7
0006'  00 05        1247  AA13    BYTE    0,5
0008'  00 01        1248  AA14    BYTE    0,1
000A'  00 06 03     1249  AA15    BYTE    0,6,3
000D'  00 01 07     1250  AA16    BYTE    0,1,7
0010'  00 02 05     1251  AA17    BYTE    0,2,5
0013'  00 01        1252  AA18    BYTE    0,1
                    1253  *
                    1254  *  SUBFRAME 2 CYCLES 1 TO 8
                    1255  *
0015'  00 06 04     1256  AA21    BYTE    0,6,4
0018'  00 01 07     1257  AA22    BYTE    0,1,7
001B'  00 05        1258  AA23    BYTE    0,5
001D'  00 01        1259  AA24    BYTE    0,1
001F'  00 06 03     1260  AA25    BYTE    0,6,3
0022'  00 01 07     1261  AA26    BYTE    0,1,7
0025'  00 02 05     1262  AA27    BYTE    0,2,5
0028'  00 01        1263  AA28    BYTE    0,1
                    1264  *
                    1265  *  SUBFRAME 3 CYCLES 1 TO 8
                    1266  *
002A'  00 06 04     1267  AA31    BYTE    0,6,4
002D'  00 01 07     1268  AA3     BYTE    0,1,7
0030'  00 05        1269  AA33    BYTE    0,5
0032'  00 01        1270  AA34    BYTE    0,1
0034'  00 06 03     1271  AA35    BYTE    0,6,3
0037'  00 01 07     1272  AA36    BYTE    0,1,7
003A'  00 02 05     1273  AA37    BYTE    0,2,5
003D'  00 01        1274  AA38    BYTE    0,1
                    1275  *
                    1276  *  SUBFRAME 3 CYCLES 1 TO 8
                    1277  *
003F'  00 06 04     1278  AA41    BYTE    0,6,4
0042'  00 01 07     1279  AA42    BYTE    0,1,7
0045'  00 05        1280  AA43    BYTE    0,5
0047'  00 01        1281  AA44    BYTE    0,1
0049'  00 06 03     1282  AA45    BYTE    0,6,3
004C'  00 01 07     1283  AA46    BYTE    0,1,7
004F'  00 02 05     1284  AA47    BYTE    0,2,5
0052'  00 01        1285  AA48    BYTE    0,1
```

```
                    1286  *
                    1287  ****************************************
                    1288  *
                    1289  *       ANALOG PARAMETER QUANTITY PER CYCLE
                    1290  *
0054'               1291  ANPQT
                    1292  *
                    1293  *  SUBFRAME 1 CYCLES 1 TO 8
                    1294  *
0054' 03            1295  AQ11    BYTE    3
0055' 03            1296  AQ12    BYTE    3
0056' 02            1297  AQ13    BYTE    2
0057' 02            1298  AQ14    BYTE    2
0058' 03            1299  AQ15    BYTE    3
0059' 03            1300  AQ16    BYTE    3
005A' 03            1301  AQ17    BYTE    3
005B' 02            1302  AQ18    BYTE    2
                    1303  *
                    1304  *  SUBFRAME 2 CYCLES 1 TO 8
                    1305  *
005C' 03            1306  AQ21    BYTE    3
005D' 03            1307  AQ22    BYTE    3
005E' 02            1308  AQ23    BYTE    2
005F' 02            1309  AQ24    BYTE    2
0060' 03            1310  AQ25    BYTE    3
0061' 03            1311  AQ26    BYTE    3
0062' 03            1312  AQ27    BYTE    3
0063' 02            1313  AQ28    BYTE    2
                    1314  *
                    1315  *  SUBFRAME 3 CYCLES 1 TO 8
                    1316  *
0064' 03            1317  AQ31    BYTE    3
0065' 03            1318  AQ32    BYTE    3
0066' 02            1319  AQ33    BYTE    2
0067' 02            1320  AQ34    BYTE    2
0068' 03            1321  AQ35    BYTE    3
0069' 03            1322  AQ36    BYTE    3
006A' 03            1323  AQ37    BYTE    3
006B' 02            1324  AQ38    BYTE    2
                    1325  *
                    1326  *  SUBFRAME 4 CYCLES 1 TO 8
                    1327  *
006C' 03            1328  AQ41    BYTE    3
006D' 03            1329  AQ42    BYTE    3
006E' 02            1330  AQ43    BYTE    2
006F' 02            1331  AQ44    BYTE    2
0070' 03            1332  AQ45    BYTE    3
0071' 03            1333  AQ46    BYTE    3
0072' 03            1334  AQ47    BYTE    3
0073' 02            1335  AQ48    BYTE    2
                    1336  *
                    1337          EVEN
```

```
                    1338  *
                    1339  ********************************************
                    1340  *
                    1341  *  LIST OF SYNCHRO TYPE PARAMETERS TABLE
                    1342  *  CONTAINS OFFSET OF PARAMETER TABLE
                    1343  *
0074'  0004        1344  ANSYNT  DATA    4               NUMB OF SYNCHRO PARAMS
0076'  0009        1345          DATA    9
0078'  000A        1346          DATA    10
007A'  000C        1347          DATA    12
007C'  0013        1348          DATA    19
                    1349  *
                    1350  ********************************************
                    1351  *
                    1352  *  ANALOG CYCLE INDEX TABLE FOR ANARD(R10)
                    1353  *       USED TO GET OFFSET VALUE FROM ANOFAR TABLE
                    1354  *
007E'               1355  ANCIT
                    1356  *
                    1357  *       S/F 1 CYCLES 1 TO 8
                    1358  *
007E'  FF           1359          BYTE    -1
007F'  02           1360          BYTE    -1+3
0080'  05           1361          BYTE    -1+3+3
0081'  07           1362          BYTE    -1+3+3+2
0082'  09           1363          BYTE    -1+3+3+2+2
0083'  0C           1364          BYTE    -1+3+3+2+2+3
0084'  0F           1365          BYTE    -1+3+3+2+2+3+3
0085'  12           1366          BYTE    -1+3+3+2+2+3+3+3
                    1367  *
                    1368  *       S/F 2 CYCLES 1 TO 8
                    1369  *
0086'  14           1370          BYTE    20
0087'  17           1371          BYTE    20+3
0088'  1A           1372          BYTE    20+3+3
0089'  1C           1373          BYTE    20+3+3+2
008A'  1E           1374          BYTE    20+3+3+2+2
008B'  21           1375          BYTE    20+3+3+2+2+3
008C'  24           1376          BYTE    20+3+3+2+2+3+3
008D'  27           1377          BYTE    20+3+3+2+2+3+3+3
                    1378  *
                    1379  *       S/F 3 CYCLES 1 TO 8
                    1380  *
008E'  29           1381          BYTE    41
008F'  2C           1382          BYTE    41+3
0090'  2F           1383          BYTE    41+3+3
0091'  31           1384          BYTE    41+3+3+2
0092'  33           1385          BYTE    41+3+3+2+2
0093'  36           1386          BYTE    41+3+3+2+2+3
0094'  39           1387          BYTE    41+3+3+2+2+3+3
0095'  3C           1388          BYTE    41+3+3+2+2+3+3+3
                    1389  *
```

```
                    1390  *       S/F 4 CYCLES 1 TO 8
                    1391  *
 0096'  3E          1392          BYTE    62
 0097'  41          1393          BYTE    62+3
 0098'  44          1394          BYTE    62+3+3
 0099'  46          1395          BYTE    62+3+3+2
 009A'  48          1396          BYTE    62+3+3+2+2
 009B'  4B          1397          BYTE    62+3+3+2+2+3
 009C'  4E          1398          BYTE    62+3+3+2+2+3+3
 009D'  51          1399          BYTE    62+3+3+2+2+3+3+3
                    1400  *
                    1401          EVEN
                    1402  *
                    1403  ***+3*****************************************
                    1404  *
                    1405  *  +3XDUCER MUX VALUES
                    1406  *
 009E'              1407  ANMUXT
                    1408+         LLDC    XDUCER,3,1,           VERTICAL ACCELERATION
 009E'  1041        1429B                         DATA    )1041
 00A0'  8404        1530A                         DATA    )84*256+ZZVL
 00A2'  0004        1533A                         DATA    )0004
 00A4'  00          1534A                         BYTE    )00
 00A5'  91          1535A                         BYTE    )91
                    1538+         LLDC    XDUCER,3,2,           LATERAL ACCELERATION
 00A6'  2082        1562B                         DATA    )2082
 00A8'  8404        1660A                         DATA    )84*256+ZZVL
 00AA'  0004        1663A                         DATA    )0004
 00AC'  00          1664A                         BYTE    )00
 00AD'  91          1665A                         BYTE    )91
                    1668+         LLDC    XDUCER,3,9,           LONGITUDINAL ACCELERATION
 00AE'  1041        1689B                         DATA    )1041
 00B0'  8410        1790A                         DATA    )84*256+ZZVL
 00B2'  0004        1793A                         DATA    )0004
 00B4'  00          1794A                         BYTE    )00
 00B5'  91          1795A                         BYTE    )91
                    1798+         SYNCRO  XDUCER,3,10,PHASEB,   FLAP POSITION LEFT
 00B6'  2082        1834B                         DATA    )2082
 00B8'  6410        1933A                         DATA    )64*256+ZZVL
 00BA'  2042        1955A                         DATA    )2042
 00BC'  00          1963A                         BYTE    )00
 00BD'  91          1964A                         BYTE    )91
                    1965+         SYNCRO  XDUCER,3,11,PHASEB,   FLAP POSITION RIGHT
 00BE'  4104        2004B                         DATA    )4104
 00C0'  6410        2100A                         DATA    )64*256+ZZVL
 00C2'  2042        2122A                         DATA    )2042
 00C4'  00          2130A                         BYTE    )00
 00C5'  91          2131A                         BYTE    )91
                    2132+         SYNCRO  XDUCER,3,13,PHASEB,   HORIZONTAL STABILIZER POSITION
 00C6'  1041        2165B                         DATA    )1041
 00C8'  6420        2267A                         DATA    )64*256+ZZVL
 00CA'  2042        2289A                         DATA    )2042
```

```
00CC' 00          2297A                       BYTE    )00
00CD' 91          2298A                       BYTE    )91
                  2299+       LLDC    XDUCER,3,19,          FLAP HANDLE POSITION
00CE' 4104        2326B                       DATA    )4104
00D0' 8440        2421A                       DATA    )84*256+ZZVL
00D2' 0004        2424A                       DATA    )0004
00D4' 00          2425A                       BYTE    )00
00D5' 91          2426A                       BYTE    )91
                  2429+       SYNCRO  XDUCER,3,20,PHASEB,   SPOILER HANDLE POSITION
00D6' 8208        2471B                       DATA    )8208
00D8' 6440        2564A                       DATA    )64*256+ZZVL
00DA' 2042        2586A                       DATA    )2042
00DC' 00          2594A                       BYTE    )00
00DD' 91          2595A                       BYTE    )91
                  2596  *
00DE' 0008        2597  ANPRQ   DATA    $-ANMUXT/8    AN-PARAM-QUANT
                  2598  *
                  2599
                  2600  ****************************
                  2601
                  2602  *
                  2603  *       ANALOG PARAMETER'S MASK TABLE
                  2604  *
00E0'             2605  ANMSKT
                  2606  *
                  2607  *       AIRCRAFT TYPE   =       BOEING 757
                  2608  *       ENGINE TYPE     =       ROLLS ROYCE
                  2609  *       2 MAN CREW
                  2610  *
                  2611  ****************************
                  2612  *
                  2613  *       ANALOG PARAMETER TYPE
                  2614  *
00E0' 00 00 00    2615  ANTYPT  BYTE    0,0,0         0 - TYPE LLDC
00E3' 08 08 08    2616          BYTE    8,8,8   3 - TYPE POT REF
00E6' 00          2617          BYTE    0             8 - SYNCRO
00E7' 08          2618          BYTE    8             2 - POT
                  2619  *
                  2620  ****************************
                  2621  *
                  2622  *       ANALOG PARAMETER DESTINATION TABLE OFFSET
                  2623  *
                  2624  *  SUBFRAME 1 CYCLES 1 TO 8
                  2625  *
00E8'             2626  ANDOA
00E8' 00 10 16    2627          BYTE    0,16,22
00EB' 01 09 13    2628          BYTE    1,9,19
00EE' 02 1B       2629          BYTE    2,27
00F0' 03 0A       2630          BYTE    3,10
00F2' 04 11 18    2631          BYTE    4,17,24
00F5' 05 0B 14    2632          BYTE    5,11,20
00F8' 06 0E 1A    2633          BYTE    6,14,26
```

```
00FB' 07 0C       2634          BYTE   7,12
                  2635  *
                  2636  *  SUBFRAME 2 CYCLES 1 TO 8
                  2637  *
00FD' 00 10 16    2638          BYTE   0,16,22
0100' 01 09 13    2639          BYTE   1,9,19
0103' 02 1B       2640          BYTE   2,27
0105' 03 0A       2641          BYTE   3,10
0107' 04 11 18    2642          BYTE   4,17,24
010A' 05 0B 14    2643          BYTE   5,11,20
010D' 06 0E 1A    2644          BYTE   6,14,26
0110' 07 0C       2645          BYTE   7,12
                  2646  *
                  2647  *  SUBFRAME 3 CYCLES 1 TO 8
                  2648  *
0112' 00 10 16    2649          BYTE   0,16,22
0115' 01 09 13    2650          BYTE   1,9,19
0118' 02 1B       2651          BYTE   2,27
011A' 03 0A       2652          BYTE   3,10
011C' 04 11 18    2653          BYTE   4,17,24
011F' 05 0B 14    2654          BYTE   5,11,20
0122' 06 0E 1A    2655          BYTE   6,14,26
0125' 07 0C       2656          BYTE   7,12
                  2657  *
                  2658  *  SUBFRAME 4 CYCLES 1 TO 8
                  2659  *
0127' 00 10 16    2660          BYTE   0,16,22
012A' 01 09 13    2661          BYTE   1,9,19
012D' 02 1B       2662          BYTE   2,27
012F' 03 0A       2663          BYTE   3,10
0131' 04 11 18    2664          BYTE   4,17,24
0134' 05 0B 14    2665          BYTE   5,11,20
0137' 06 0E 1A    2666          BYTE   6,14,26
013A' 07 0C       2667          BYTE   7,12
                  2668  *
                  2669          EVEN
                  2670  *
                  2671  **********************************
                  2672  *
                  2673  *       ANALOG PARAMETER DESTINATION
                  2674  *
                  2675  *   FORM: GENDT N1,N2,N3,N4,N5
                  2676  *
                  2677  *       N1      =       BUFFER OFFSET WORD (1 TO 256)
                  2678  *       N2      =       BUFFER TYPE
                  2679  *                       0 FOR DFDR
                  2680  *                       1 FOR INTER-CPU
                  2681  *                       2 FOR SPARE
                  2682  *                       3 FOR SPARE
                  2683  *       N3      =       OUTPUT POSITION LEFT SHIFT COUNT
                  2684  *       N4      =       SPARE
                  2685  *       N5      =       END OF DATA SET
```

```
                    2686  *                    0 FOR CONTINUE
                    2687  *                    1 FOR END
                    2688  *
013C'               2689  ANDT
                    2690+       GENDT   2,0,0,0,END             VERTICAL ACCELERATION
013C' 8001          2710A               DATA    (XYZ*256+2-1)
                    2711+       GENDT   10,0,0,0,END                    *
013E' 8009          2731A               DATA    (XYZ*256+10-1)
                    2732+       GENDT   18,0,0,0,END                    *
0140' 8011          2752A               DATA    (XYZ*256+18-1)
                    2753+       GENDT   26,0,0,0,END                    *
0142' 8019          2773A               DATA    (XYZ*256+26-1)
                    2774+       GENDT   34,0,0,0,END                    *
0144' 8021          2794A               DATA    (XYZ*256+34-1)
                    2795+       GENDT   42,0,0,0,END                    *
0146' 8029          2815A               DATA    (XYZ*256+42-1)
                    2816+       GENDT   50,0,0,0,END                    *
0148' 8031          2836A               DATA    (XYZ*256+50-1)
                    2837+       GENDT   58,0,0,0,                       *
014A' 0039          2857A               DATA    (XYZ*256+58-1)
                    2858+       GENDT   1,1,0,0,END                     *       FOR CPU 2
014C' 8100          2878A               DATA    (XYZ*256+1-1)
                    2879  *
                    2880+       GENDT   15,0,1,0,END            LATERAL ACCELLERATION
014E' 840E          2900A               DATA    (XYZ*256+15-1)
                    2901+       GENDT   31,0,1,0,END                    *
0150' 841E          2921A               DATA    (XYZ*256+31-1)
                    2922+       GENDT   47,0,1,0,END                    *
0152' 842E          2942A               DATA    (XYZ*256+47-1)
                    2943+       GENDT   63,0,1,0,                       *
0154' 043E          2963A               DATA    (XYZ*256+63-1)
                    2964+       GENDT   2,1,1,0,END                     *       FOR CPU 2
0156' 8501          2984A               DATA    (XYZ*256+2-1)
                    2985  *
                    2986+       GENDT   55,0,1,0,               LONGITUDINAL ACCELLERATION
0158' 0436          3006A               DATA    (XYZ*256+55-1)
                    3007+       GENDT   3,1,1,0,END                     *       FOR CPU 2
015A' 8502          3027A               DATA    (XYZ*256+3-1)
                    3028  *
                    3029+       GENDT   8,0,2,0,END             FLAP HANDLE POSITION
015C' 8807          3049A               DATA    (XYZ*256+8-1)
                    3050+       GENDT   40,0,2,0,                       *
015E' 0827          3070A               DATA    (XYZ*256+40-1)
                    3071+       GENDT   7,1,2,0,END                     *       FOR CPU 2
0160' 8906          3091A               DATA    (XYZ*256+7-1)
                    3092  *
                    3093+       GENDT   16,0,2,0,END            SPOILER HANDLE POSITION
0162' 880F          3113A               DATA    (XYZ*256+16-1)
                    3114+       GENDT   48,0,2,0,                       *
0164' 082F          3134A               DATA    (XYZ*256+48-1)
                    3135+       GENDT   8,1,2,0,END                     *       FOR CPU 2
0166' 8907          3155A               DATA    (XYZ*256+8-1)
```

```
                    3156  *
                    3157+       GENDT   7,0,2,0,                 FLAP POSITION RIGHT
0168' 0806          3177A               DATA    (XYZ*256+7-1)
                    3178+       GENDT   5,1,2,0,END                     *       FOR CPU 2
016A' 8904          3198A               DATA    (XYZ*256+5-1)
                    3199  *
                    3200+       GENDT   39,0,2,0,                FLAP POSITION LEFT
016C' 0826          3220A               DATA    (XYZ*256+39-1)
                    3221+       GENDT   4,1,2,0,END                     *       FOR CPU 2
016E' 8903          3241A               DATA    (XYZ*256+4-1)
                    3242  *
                    3243+       GENDT   56,0,2,0,END             HORIZ. STABILIZER POSITION
0170' 8837          3263A               DATA    (XYZ*256+56-1)
                    3264+       GENDT   24,0,2,0,
0172' 0817          3284A               DATA    (XYZ*256+24-1)
                    3285+       GENDT   6,1,2,0,END                     *       FOR CPU 2
0174' 8905          3305A               DATA    (XYZ*256+6-1)
                    3306  *
                    3307  *
                    3308  ********************************************************************
                    3309  *
                    3310                      CALIBRATION
                    3311  *                   ==========
                    3312  *
                    3313  *  GNDREF AND VCAL MUX VALUES
                    3314  *
0176'               3315  CALMAT
                    3316+       LLDC    VCAL,3,1,
0176' 0810          3353B               DATA    )0810
0178' 8405          3438A               DATA    )84*256+ZZVL
017A' 0004          3441A               DATA    )0004
017C' 00            3442A               BYTE    )00
017D' 91            3443A               BYTE    )91
                    3446+       LLDC    GNDREF,3,1,
017E' 0820          3480B               DATA    )0820
0180' 8406          3568A               DATA    )84*256+ZZVL
0182' 0004          3571A               DATA    )0004
0184' 00            3572A               BYTE    )00
0185' 91            3573A               BYTE    )91
                    3576+       LLDC    VCAL,3,2,
0186' 0810          3613B               DATA    )0810
0188' 8405          3698A               DATA    )84*256+ZZVL
018A' 0004          3701A               DATA    )0004
018C' 00            3702A               BYTE    )00
018D' 91            3703A               BYTE    )91
                    3706+       LLDC    GNDREF,3,2,
018E' 0820          3740B               DATA    )0820
0190' 8406          3828A               DATA    )84*256+ZZVL
0192' 0004          3831A               DATA    )0004
0194' 00            3832A               BYTE    )00
0195' 91            3833A               BYTE    )91
                    3836+       LLDC    VCAL,3,9,
```

```
01B6' 0810      3873B                   DATA    )0810
01B8' 8411      3958A                   DATA    )84*256+ZZVL
01BA' 0004      3961A                   DATA    )0004
01BC' 00        3962A                   BYTE    )00
01BD' 91        3963A                   BYTE    )91
                3966+   LLDC    GNDREF,3,9,
01BE' 0820      4000B                   DATA    )0820
01C0' 8412      4088A                   DATA    )84*256+ZZVL
01C2' 0004      4091A                   DATA    )0004
01C4' 00        4092A                   BYTE    )00
01C5' 91        4093A                   BYTE    )91
                4096+   SYNCRO  VCAL,3,10,PHASEB,
01C6' 8808      4114A                   DATA    )8808
01C8' E480      4122A                   DATA    )E480
01CA' 2042      4157A                   DATA    )2042
01CC' 00        4165A                   BYTE    )00
01CD' 91        4166A                   BYTE    )91
                4167+   SYNCRO  GNDREF,3,10,PHASEB,
01CE' 0820      4213B                   DATA    )0820
01D0' E412      4312A                   DATA    )E4*256+ZZVL
01D2' 2042      4324A                   DATA    )2042
01D4' 00        4332A                   BYTE    )00
01D5' 91        4333A                   BYTE    )91
                4334+   SYNCRO  VCAL,3,11,PHASEB,
01D6' 8808      4352A                   DATA    )8808
01D8' E480      4360A                   DATA    )E480
01DA' 2042      4395A                   DATA    )2042
01DC' 00        4403A                   BYTE    )00
01DD' 91        4404A                   BYTE    )91
                4405+   SYNCRO  GNDREF,3,11,PHASEB,
01DE' 0820      4451B                   DATA    )0820
01E0' E412      4550A                   DATA    )E4*256+ZZVL
01E2' 2042      4562A                   DATA    )2042
01E4' 00        4570A                   BYTE    )00
01E5' 91        4571A                   BYTE    )91
                4572+   SYNCRO  VCAL,3,13,PHASEB,
01E6' 8808      4590A                   DATA    )8808
01E8' E480      4598A                   DATA    )E480
01EA' 2042      4633A                   DATA    )2042
01EC' 00        4641A                   BYTE    )00
01ED' 91        4642A                   BYTE    )91
                4643+   SYNCRO  GNDREF,3,13,PHASEB,
01EE' 0820      4689B                   DATA    )0820
01F0' E422      4788A                   DATA    )E4*256+ZZVL
01F2' 2042      4800A                   DATA    )2042
01F4' 00        4808A                   BYTE    )00
01F5' 91        4809A                   BYTE    )91
                4810+   LLDC    VCAL,3,19,
01F6' 0810      4847B                   DATA    )0810
01F8' 8441      4932A                   DATA    )84*256+ZZVL
01FA' 0004      4935A                   DATA    )0004
01FC' 00        4936A                   BYTE    )00
```

```
01DD' 91      4937A                       BYTE   )91
              4940+          LLDC   GNDREF,3,19,
01DE' 0820    4974B                       DATA   )0820
01E0' 8442    5062A                       DATA   )84*256+ZZVL
01E2' 0004    5065A                       DATA   )0004
01E4' 00      5066A                       BYTE   )00
01E5' 91      5067A                       BYTE   )91
              5070+          SYNCRO VCAL,3,20,PHASEB,
01E6' 8808    5088A                       DATA   )8808
01E8' E480    5096A                       DATA   )E480
01EA' 2042    5131A                       DATA   )2042
01EC' 00      5139A                       BYTE   )00
01ED' 91      5140A                       BYTE   )91
              5141+          SYNCRO GNDREF,3,20,PHASEB,
01EE' 0820    5187B                       DATA   )0820
01F0' E442    5286A                       DATA   )E4*256+ZZVL
01F2' 2042    5298A                       DATA   )2042
01F4' 00      5306A                       BYTE   )00
01F5' 91      5307A                       BYTE   )91
              5308   *
01F6' 0010    5309   CALMAQ  DATA   $-CALMAT/8      CAL-MUX-ADDR-QUANT
              5310   *
              5311
              5312   ***************************
              5313
              5314   *
              5315   *       CALIBRATION CONSTANTS
              5316   *
              5317   *               VICS * GIS  (FOR 1ST READING)
01F8' 1000    5318   CAL1FT  DATA   4096           DC VOLTS ABS (LLDC),   0, SINGLE
01FA' 0C80    5319           DATA   3200           HI LVL DC ABS,         1, SINGLE
01FC' 1000    5320           DATA   4096           POTENTIOMETER,         2, SINGLE
01FE' 1000    5321           DATA   4096           5V POT Ref,            3, SINGLE
0200' 0DBC    5322           DATA   3516           TEMP. BOLBS,           4, SINGLE
0202' 0B50    5323           DATA   2896           AC RATIO # 1,          5, DOUBLE
0204' 1000    5324           DATA   4096           DC RATIO # 1,          6, DOUBLE
0206' 1000    5325           DATA   4096           DC RATIO # 2,          7, DOUBLE
0208' 17F9    5326           DATA   6137           SYNCHRO,               8, DOUBLE
020A' 0AE1    5327           DATA   2785           AC RATIO # 2,          9, DOUBLE
              5328   *
              5329   *               VICR * GIR  (FOR 2ND READING)
020C' 1000    5330   CAL2FT  DATA   4096           DC VOLTS ABS (LLDC),   0, SINGLE
020E' 0C80    5331           DATA   3200           HI LVL DC ABS,         1, SINGLE
0210' 1000    5332           DATA   4096           POTENTIOMETER,         2, SINGLE
0212' 1000    5333           DATA   4096           5V POT REF,            3, SINGLE
0214' 0DBC    5334           DATA   3516           TEMP. BOLBS,           4, SINGLE
0216' 0AE1    5335           DATA   2785           AC RATIO # 1,          5, DOUBLE
0218' 1000    5336           DATA   4096           DC RATIO # 1,          6, DOUBLE
021A' 1400    5337           DATA   5120           DC RATIO # 2,          7, DOUBLE
021C' 17F9    5338           DATA   6137           SYNCHRO,               8, DOUBLE
021E' 0AE1    5339           DATA   2785           AC RATIO # 2,          9, DOUBLE
              5340   *
```

```
                    5341  *
                    5342
                    5343  ****************************
                    5344
                    5345  *
                    5346  *       SYSTEM-CALIBRATION-BUFFER-OFFSET  (USED BY "CALRT")
                    5347  *       FORMAT PER ANALOG TYPE TABLE (ANTYPT)
                    5348  *       (0: GND. 2 : 2.5 VOLT FOR TEMP BULB(TYPE 4)
                    5349  *        4: 5 VOLT FOR LLDC(TYPE 0), POT(TYPE 2 & 3)
                    5350  *        6: 25 VOLT FOR SYNCHRO(TYPE 8)
0220' 1A 1C 1C      5351  CALSOT  BYTE    26,28,28,28,28,28,28,28,28,30
0223' 1C 1C 1C
0226' 1C 1C 1C
0229' 1E
                    5352          EVEN
                    5353  *
                    5354
                    5355  ****************************
                    5356
                    5357  *
                    5358  *               USED BY CALMT
                    5359  *       CAL LIMITS      (A TOLERANCE OF %10 IS GIVEN.
                    5360  *                        THE LIMITS GIVEN HERE ARE THE LOWER OR UPPER
                    5361  *                        LIMITES.)
                    5362  *       LOWER LIMIT
022A' FEB8          5363  CALLT   DATA    ()FEB8)                 GND. (- 4096 * %5)
                    5364  *                       SIG-LIMIT
022C' 0E66          5365          DATA    ()2000-)119A)           DC-VOLTS-ABS (5V)
022E' 0B44          5366          DATA    ()2000-)14BC)           HILVL-DC-ABS (5V)
0230' 0E66          5367          DATA    ()2000-)119A)           POTENTIOMETER (5V)
0232' 0E66          5368          DATA    ()2000-)119A)           5V. POT. REF. (5V)
0234' 0C93          5369          DATA    ()2000-)136D)           TEMP. BOLBS (5V)
0236' 0E66          5370          DATA    ()2000-)119A)           AC-VOLTS-RATIO-#1 (5V)
0238' 0E66          5371          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#1 (5V)
023A' 0E66          5372          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#2 (5V ?)
023C' 159A          5373          DATA    ()2000-)0A66)           SYNCHRO (25V)
023E' 0B44          5374          DATA    ()2000-)14BC)           AC VOLTS RATIO #2 (25V)
                    5375  *                       REF-LIMIT
0240' FEB8          5376          DATA    )FEB8                   GND (- 4096 * 5%)
0242' 0000          5377          DATA    0                       DC-VOLTS-ABS (NOT USED)
0244' 0000          5378          DATA    0                       HILVL-DC-ABS (NOT USED)
0246' 0000          5379          DATA    0                       POTENTIOMETER (NOT USED)
0248' 0000          5380          DATA    0                       5V. POT. REF. (NOT USED)
024A' 0000          5381          DATA    0                       TEMP. BOLBS (NOT USED)
024C' 0B44          5382          DATA    ()2000-)14BC)           AC-VOLTS-RATIO-#1 (25V)
024E' 0E66          5383          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#1 (5V)
0250' 0E66          5384          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#2 (5V ?)
0252' 159A          5385          DATA    ()2000-)0A66)           SYNCHRO (25V)
0254' 0B44          5386          DATA    ()2000-)14BC)           AC VOLTS RATIO #2 (25V)
                    5387  *
                    5388  *               UPPER LIMIT
0256' 0148          5389  CALULT  DATA    ()2000-)1EB8)           GND. (- 4096 * %5)
```

```
                    5390  *                    SIG-LIMIT
025B'  119A         5391        DATA    ()2000-)0E66)       DC-VOLTS-ABS (5V)
025A'  0DA2         5392        DATA    ()2000-)125E)       HILVL-DC-ABS (5V)
025C'  119A         5393        DATA    ()2000-)0E66)       POTENTIOMETER (5V)
025E'  119A         5394        DATA    ()2000-)0E66)       5V. POT. REF. (5V)
0260'  0F3B         5395        DATA    ()2000-)10C5)       TEMP. BOLBS (5V)
0262'  119A         5396        DATA    ()2000-)0E66)       AC-VOLTS-RATIO-#1 (5V)
0264'  119A         5397        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#1 (5V)
0266'  119A         5398        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#2 (5V ?)
0268'  1A66         5399        DATA    ()2000-)059A)       SYNCHRO (25V)
026A'  0DA2         5400        DATA    ()2000-)125E)       AC VOLTS RATIO #2 (25V)
                    5401  *                    REF-LIMIT
026C'  0148         5402        DATA    )2000-)1EB8         GND (- 4096 * 5%)
026E'  0000         5403        DATA    0                   DC-VOLTS-ABS (NOT USED)
0270'  0000         5404        DATA    0                   HILVL-DC-ABS (NOT USED)
0272'  0000         5405        DATA    0                   POTENTIOMETER (NOT USED)
0274'  0000         5406        DATA    0                   5V. POT. REF. (NOT USED)
0276'  0000         5407        DATA    0                   TEMP. BOLBS (NOT USED)
0278'  0DA2         5408        DATA    ()2000-)125E)       AC-VOLTS-RATIO-#1 (25V)
027A'  119A         5409        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#1 (5V)
027C'  119A         5410        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#2 (5V ?)
027E'  1A66         5411        DATA    ()2000-)059A)       SYNCHRO (25V)
0280'  0DA2         5412        DATA    ()2000-)125E)       AC VOLTS RATIO #2 (25V)
                    5413
```

```
                    5415  ************************************************************************
                    5416  *
                    5417  *                       BITE
                    5418  *                       ====
                    5419  *
                    5420  *       INTERNAL REF. MUX ADDRESS TABLE
                    5421  *       (SEE BTDRQ.SRC AND BTLMT.SRC)
                    5422  *
                    5423  ************************************************************************
                    5424  *
                    5425  *       ANALOG POSITION TABLE
                    5426  *
0282'               5427  ANPSTB
                    5428  *       FORM: ANPT      N1,N2,N3
                    5429  *                       N1      =       INPUT START BIT
                    5430  *                       N2      =       INPUT MASK WORD INDEX
                    5431  *                       N3      =       ROUNDING
                    5432  *                                       BLANK FOR NO ROUNDING
                    5433  *                                       NONBLANK FOR ROUNDING
                    5434
                    5435+         ANPT    1,11,           VERTICAL ACCELLERATION
0282'  2C           5446A                 BYTE    1-1+XXX
                    5448+         ANPT    2,10,           LATERAL ACCELLERATION
0283'  29           5459A                 BYTE    2-1+XXX
                    5461+         ANPT    2,10,           LONGITUDINAL ACCELLERATION
0284'  29           5472A                 BYTE    2-1+XXX
                    5474+         ANPT    2,10,ROUND      T/E FLAP LEFT
0285'  A9           5483A                 BYTE    2-1+XXX+)B0
                    5487+         ANPT    2,10,ROUND      T/E FLAP RIGHT
0286'  A9           5496A                 BYTE    2-1+XXX+)B0
                    5500+         ANPT    2,10,ROUND      HORIZONTAL STABILIZER POSITION
0287'  A9           5509A                 BYTE    2-1+XXX+)B0
                    5513+         ANPT    3,9,            FLAP HANDLE POSITION
0288'  26           5524A                 BYTE    3-1+XXX
                    5526+         ANPT    2,10,ROUND      SPOILER HANDLE POSITION
0289'  A9           5535A                 BYTE    2-1+XXX+)B0
                    5539+         ANPT    1,0,0           5V POT REF
028A'  80           5548A                 BYTE    1-1+XXX+)B0
                    5552  *
                    5553          EVEN
                    5554  *
                    5555          END
```

```
                    1          IDT    ANSYN
                    2
                    3          SUBTTL  SYNCHRO SCALING/CONVERSION
                    4
                    5  *       CALLING SEQ:    CALL    @ANSYN
                    6
                    7  *-------------------------------------------------------------------+
                    8  *                                                                   !
                    9  *       ANSYN CONVERTS THE SYNCHRO DUAL INPUT DATA TO AN ANGLE VALUE
                   10  *       REPRESENTED IN 12 BITS (1 COUNT = 360 DEG / 4096).
                   11  *       IT TESTS THE SIGNS AND ABSOLUTE VALUE OF THE TWO INPUTS AND
                   12  *       ACCORDINGLY USES ANEQUATION (ONE OF EIGHT) TO COMPUTE THIS
                   13  *       ANGLE.
                   14  *                                                                   !
                   15  *-------------------------------------------------------------------+
                   16  *       VERSION : 1
                   17  *       PROGRAMMED BY : M.CONSTANTINIDES
                   18  *       CHECKED BY : M.CONSTANTINIDES
                   19
                   20
                   21          INTERN  ANSYN
                   22  * REFERD MODULES:
                   23          EXTERN  ATANV           (ATAN)
                   24
=0000              25  ATS     EQU     R0      =       SCRATCH
=0001              26  ATR     EQU     R1      =       SCRATCH
=0002              27  PNT     EQU     R2      =       SCRATCH
=0003              28  VCTS    EQU     R3      =       1ST AN-DATA (INPUT)
=0004              29  VCTR    EQU     R4      =       2ND AN-DATA (INPUT), XDUCER VALUE (OUTPUT)
=0005              30  SBP     EQU     R5      =       SCRATCH
                   31
                   37          INCLUDE REGDEF          REGISTER DEFENITIONS
                   56          INCLUDE CNSTNT          CONSTANTS
                  184          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  485          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  715          INCLUDE JMPMAC          JUMP MACROS
                  749          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  764          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000            1141          RSECT   ANSYN
                 1142  ********************************************************************************
```

```
0000'                1144   ANSYN
0000'  04C2          1145          CLR     PNT                     POINTER = 0, FOR VCTS, VCTR > 0
                     1146+         DOIF    VCTR,MI,,,,             IF VCTR < 0,
0002'  C104          1280B           MOV   VCTR,VCTR
0004'  1101          1328E           JLT   $+4
0006'  1001          1329E           JMP   91$
0008'  0582          1452          INC     PNT                       POINTER = 1
                     1453+         ENDBLK                          ENDIF
000A'                1585E  91$
                     1673+         DOIF    VCTS,MI,,,,             IF VCTS < 0,
000A'  C0C3          1807B           MOV   VCTS,VCTS
000C'  1101          1859E           JLT   $+4
000E'  1001          1860E           JMP   92$
0010'  05C2          1979          INCT    PNT                       POINTER = 2
                     1980+         ENDBLK                          ENDBLK
0012'                2116E  92$
0012'  0743          2200          ABS     VCTS
0014'  0744          2201          ABS     VCTR
0016'  C143          2202          MOV     VCTS,R5                 R5 = VCTS
0018'  C004          2203          MOV     VCTR,R0                 MSW R0 = VCTR
001A'  04C1          2204          CLR     R1                      LSW
                     2205+         DOIF    VCTS,LEQ,VCTR,          IF VCTS >= VCTR,
001C'  8103          2343B           C     VCTS,VCTR
001E'  1504          2392E           JGT   93$
0020'  0222 0004     2511          AI      PNT,C4                    POINTER ; (FOR VCTR > VCTS)
0024'  C003          2512          MOV     VCTS,R0                   MSW VCTS SCB16
0026'  C144          2513          MOV     VCTR,R5                   R5 = VCTR
                     2514+         ENDBLK                          ENDIF
0028'                2654E  93$
0028'  C040          2734          MOV     R0,R1                   R0,R1 = DIVIDEND (2 ** 14)
002A'  0AE1          2735          SLA     R1,C14
002C'  0820          2736          SRA     R0,C2
002E'  3C05          2737          DIV     R5,R0                   R = VMTR/VMTS OR VMTS/VMTR (SCB14)
                     2738   *                                COMPUTE ARC-TAN (.866*R/(1-.5R))=(Y/X)
                     2739   *                                        GEN. SYN-EQ-TABLE POINTER (EQ 0-7)
0030'  0204 0006     2740          LI      R4,C6                     6-BYTES PER DATA-ITEM IN TABLE
0034'  3902          2741          MPY     R2,R4                     EQ.NT.*6 = OFFSET FOR EQ. N
0036'  0225 0006'    2742          AI      R5,ANSETB                 R5 = SYN-EQ-TABLE POINTER FOR ER #N
                     2743   *                                COMPUTE Y = .866 * R
003A'  C040          2744          MOV     R0,R1                     R(SCB14)
003C'  3860 0000'    2745          MPY     @KF866,R1                 Y(SCB29 OR B13) = .866(SCB15) * R(SCB14)
                     2746   *                                COMPUTE X = 1 - (.5 * R)
0040'  C0C0          2747          MOV     R0,R3                     R (VCTR/VCTS) SCBB14
0042'  38E0 0002'    2748          MPY     @KF5,R3                   X1(SCB29 OR B13) = .5(SCB15) * R(SCB14)
0046'  C0A0 0004'    2749          MOV     @K1,R2                    1. SCBB13
004A'  C135          2750          MOV     *R5+,R4
                     2751+         DOIF    ,NE,,                   IF 1 - (.5 * R) TYPE
004C'  1301          2930E             JEQ   94$
004E'  0503          3057            NEG   R3                        X1 = -X1
                     3058+         ENDBLK                          ENDIF
0050'                3202E  94$
0050'  A083          3278          A       R3,R2                   X(SCB13) = X1
```

```
                    3279  *                                 COMPUTE ARC-TAN (Y / X)
                    3280+         CALLWP  ATANV                     R1 = PHI = ARC-TAN (X/Y) 180-DEG (@B15)
0052'  0420 0001*   3282A           BLWP  @ATANV
                    3284  *                                 COMPUTE VCT = ANGLE - PHI
0056'  C0F5         3285          MOV     *R5+,R3                   ANGLE (240, 60, 120 OR 300) FOR EQ $N
0058'  C115         3286          MOV     *R5,R4
                    3287+         DOIF    ,NE,,                     IF ANGLE + PHI TYPE
005A'  1301         3470E             JEQ   95$
005C'  0501         3593            NEG   R1                          PHI NEGATIVE
                    3594+         ENDBLK                            ENDIF
005E'               3742E 95$
005E'  A0C1         3814          A       R1,R3                     ANGLE +/- PHI = VCT (180-DEG SC@B15)
                    3815  *                                 RESCALE FOR OUTPUT TO 180-DEG @B11
0060'  0943         3816          SRL     R3,C4                     RESCALE 180-DEG @B11
0062'  045B         3817          RT
                    3818  ********************************************************************************
                    3819+         PRVDAT
0000'  6ED9         3821  KF866   DATA    28377             = .866 SC@B15
0002'  4000         3822  KF5     DATA    16384             = .5 SC@B15
0004'  2000         3823  K1      DATA    8192              = 1. SC@B15
                    3824  *
                    3825
                    3826  * ANSETB CONTAINS EIGHT SYNCHRO EQUATION INFORMATIONS FOR SYNCHRO
                    3827  * PARAMETER COMPUTATION (VCT). THE TABLE CONSISTS OF EIGHT DATA
                    3828  * ITEMS, ONE PERSYNCHRO EQUATION. EACH DATA ITEM CONTAINS THREE
                    3829  * WORDS; 1ST-WORD = -1(FOR PHI1), 0(FOR PHI2)
                    3830  *        2ND-WORD = ANGLE(180-DEG SC@B15)
                    3831  *        3RD-WORD = -1(FOR ANGLE -PHI), 0(FOR ANGLE +PHI).
                    3832  *
0006'               3834  ANSETB
                    3835+         DATBL   -1,43691,-1       PHI1, 240-DEG, -PHI1 (240-PHI1)
                    3856+         DATBL   0,43691,0,        PHI2, 240-DEG, +PHI2 (240+PHI2)
                    3877+         DATBL   0,10923,0,        PHI2, 60-DEG, +PHI2 (60 + PHI2)
                    3898+         DATBL   -1,10923,-1,      PHI1, 60-DEG, -PHI1 (60 - PHI1)
                    3919+         DATBL   -1,21845,0,       PHI1, 120-DEG, +PHI1 (120 + PHI1)
                    3940+         DATBL   0,54613,-1        PHI2, 300-DEG, -PHI2 (300 - PHI2)
                    3961+         DATBL   0,21845,-1,       PHI2, 120-DEG -PHI2 (120 - PHI2)
                    3982+         DATBL   -1,54613,0,       PHI1, 300-DEG, +PHI1 (300 + PHI1)
                    4003
                    4004          END
```

```
                    1          IDT     ATAN
                    2
                    3          SUBTTL  ATAN - SINGLE PRECISION ARC-TANGENT
                    4
                    5  *       CALLING SEQ:    CALLWP  @ATANV
                    6
                    7  *-------------------------------------------------------------------+
                    8  *                                                                   !
                    9  *       COMPUTES ARCTAN Y/X USING FOLLOWING APPROXIMAT:
                   10  *       ARCTAN Y/X(=Z) =((((C9*Z**2+C7)*Z**2+C5)*Z**2+C3)*Z**2+C1)*Z
                   11  *                                                                   !
                   12  *-------------------------------------------------------------------+
                   13  *       VERSION : 1
                   14  *       PROGRAMED BY ; N.CONSTANTINIDES
                   15  *       CHECKED BY   ; N.CONSTANTINIDES
                   16
                   17          INTERN  ATAN
                   18  * REFERD MODULES:
                   19          EXTERN  SDIV            = SIGNED DIVIDE
                   20          EXTERN  SSMRR           = SIGNED MULTIPLY/ROUND/RESCALE
                   21
                   22  * INPUTS:       CALLER'S R1 = Y
                   23  *               CALLER'S R2 = X
                   24  *               BOTH IN THE SAME UNITS AND WITH THE SAME SCALE
                   25  *               FACTOR.
                   26  * OUTPUTS:      CALLER'S R1 = ARCTAN Y/X 180 DEG SCALED AT B15
                   27  *       R0, R1, R3-THRU-R9  = MODIFIED
                   28
                   34          INCLUDE REGDEF          REGISTER DEFENITIONS
                   53          INCLUDE CNSTNT          CONSTANTS
                  186          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  487          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  717          INCLUDE JMPMAC          JUMP MACROS
                  751          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  766          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1143          RSECT   ATAN
                  1144  ********************************************************************************
```

```
0000'                  1146  ATAN
                       1147  *     GET ARGUMENT
0000' C24D             1148          MOV     R13,R9          GET CALLER'S WP
0002' 05C9             1149          INCT    R9              BUMP TO CALLER'S R1
0004' C159             1150          MOV     *R9,R5          GET Y
0006' C1A9 0002        1151          MOV     @2(R9),R6       GET X
                       1152  *
000A' C1C5             1153          MOV     R5,R7           SAVE Y IN R7
000C' 0745             1154          ABS     R5              ABSOLUTE (Y)
                       1155  *
000E' C206             1156          MOV     R6,R8           SAVE X IN R8
0010' 0746             1157          ABS     R6              ABSOLUTE (X)
0012' 1603             1158          JNE     2$              BRANCH IF ABS(X) NOT ZERO
                       1159  *     X = 0
0014' 0203 2000        1160          LI      R3,>2000        SET ANGLE TO 45 DEGREES
0018' 1025             1161          JMP     9$              EXIT
                       1162  *
001A' C045             1163  2$      MOV     R5,R1           ABS (Y)
001C' A046             1164          A       R6,R1           =ABS(Y) + ABS(X)
001E' 1503             1165          JGT     4$              JUMP IF NO OVERFLOW
0020' 1302             1166          JEQ     4$
                       1167  *       SUM OF ABS(Y) AND ABS(X) OVERFLOW
0022' 0815             1168          SRA     R5,1            DIVIDE ABS(Y) BY 2
0024' 0816             1169          SRA     R6,1            DIVIDE ABS(X) BY 2
                       1170  *
0026' C045             1171  4$      MOV     R5,R1
0028' 1603             1172          JNE     5$              JIF ABS(Y) NOT ZERO
                       1173  *     Y = 0
002A' 0203 E000        1174          LI      R3,>E000        SET ANGLE TO -45 DEGREES
002E' 101A             1175          JMP     9$
0030' A046             1176  5$      A       R6,R1           =ABS(Y) + ABS(X)
0032' C0C5             1177          MOV     R5,R3
0034' 60C6             1178          S       R6,R3           =ABS(Y) - ABS(X)
0036' 04C4             1179          CLR     R4              CLEAR R4 FOR DIVIDE INSTRUCTION
0038' 0813             1180          SRA     R3,1            RESCALE TO B14
                       1181+         CALL    SDIV            Z=ABS(Y) - ABS(X)/ABS(Y) + ABS(X)
003A' 06A0 0001*       1189A           BL    @SDIV
003E' C143             1190          MOV     R3,R5           SAVE Z IN R5
0040' C043             1191          MOV     R3,R1           Z IN R3 AND R1 FOR MPY
                       1192  *                               =Z*Z  ROUND AND RESCALE IN R3
                       1193+         CALL    SSMRR           MULTIPLY R1 & R3. ROUND & RESCALE TO B15(DRBO)
0042' 06A0 0002*       1200A           BL    @SSMRR
0046' C003             1202          MOV     R3,R0           Z**2
                       1203  *     HASTINGS'S APPROXIMATION COMPUTATION
0048' 0206 0006        1204          LI      R6,6            FOR 4 LOOPS
004C' C0E0 0008'       1205          MOV     @CAT9,R3        C9 CONSTANT
                       1206  *                               CONSTANT*Z**2(=R1*R3) ROUND & RESCALE
0050' C040             1207  8$      MOV     R0,R1           Z**2
                       1208+         CALL    SSMRR           MULTIPLY R1&R3. R3 ROUND & RESCALE TO B0
0052' 06A0 0002*       1215A           BL    @SSMRR
0056' A0E6 0000'       1217          A       @CAT1(R6),R3    ADD CONSTANT
005A' 0646             1218          DECT    R6              DECREMENT
```

```
00?C' 18F9          1219          JOC     8$              LOOP 4 TIMES
                    1220    *
00?E' C045          1221          MOV     R5,R1           =Z
                    1222    *                             R3=Z*(SUM)  ROUND & RESCALE
                    1223+         CALL    SSMRR           MULTIPLY R1&R3. R3 ROUND & RESCALE TO B0
0052' 06A0 0002*    1230A           BL    @SSMRR
                    1232    *     QUADRANT TEST
0056' C208          1233    9$      MOV     R8,R8           X=- ?
0058' 1103          1234            JLT     10$             BRANCH IF QUADRANT 2 OR 3
                    1235    *       DATA IN QUADRANT 1 OR 4
005C' 0223 2000     1236            AI      R3,>2000        ADD 45 DEGREES
0060' 1004          1237            JMP     12$
                    1238    *       DATA IN QUADRANT 2 OR 3
0062' 0201 6000     1239    10$     LI      R1,>6000        =135 DEGREES
0066' 6043          1240            S       R3,R1           =135 DEG - RESULT
0068' C0C1          1241            MOV     R1,R3
                    1242    *
006A' C1C7          1243    12$     MOV     R7,R7           Y=- ?
006C' 1502          1244            JGT     14$             BRANCH IF QUADRANT 1 OR 2
006E' 1301          1245            JEQ     14$
                    1246    *       DATA IN QUADRANT 3 OR 4
0070' 0503          1247            NEG     R3              NEGATE RESULT
                    1248    *
0072' C643          1249    14$     MOV     R3,*R9          PUT RESULT IN CALLER'S R1
0074' 0380          1250            RTWP
                    1251    ********************************************************************************
                    1252    * PRIVATE DATA
                    1253+           PRVDAT
                    1255    *     HASTING'S CONSTANT
                    1256    *
0076' 28BE          1257    CAT1    DATA    >28BE           C1  0.3183026  SC 1 AT B15
0078' F273          1258            DATA    >F273           C3  -.1058774  SC 1 AT B15
007A' 07E3          1259            DATA    >07E3           C5  0.0616068  SC 1 AT B15
007C' FB42          1260            DATA    >FB42           C7  -.0370617  SC 1 AT B15
007E' 0225          1261    CAT9    DATA    >0225           C9  0.0167601  SC 1 AT B15
0080' FF85          1262    CAT11   DATA    >FF85           C11 -.0037537  SC 1 AT B15
                    1263            END
```

```
                    1          IDT     ANVT
                    2          SUBTTL  AN-V-TRANSDUCER
                    3  *       CALLING SEQ:    CALL    @ANVT
                    4  ********************************************************************************
                    5  *                                                                              *
                    6  *       ANVT CALCULATES VCTS (VCTS=(VMTS-VMOS)*KS), IF DUAL CONVERSION         *
                    7  *       (TYPE > 4) CALCULATES VCTR (VCTR = (VMTR-VMOR)*KR) AND THEN            *
                    8  *       ACCORDING O THE PARAMETER TYPE CALLS THE SCALING/CONVERSION            *
                    9  *       SUBROUTINE. IN THE CASE OF 5V.-POT.-REF. ((TYPE=3) THE VALUE           *
                   10  *       IS ALREADY STORED INTO 'VCPR') AND DC-VOLTAGE THERE IS NO NEED          *
                   11  *       FOR SCALING (THE SUBR ONLY RETURNS).                                   *
                   12  *                                                                              *
                   13  ********************************************************************************
                   14  *       VERSION: 1
                   15  *       PROGRAMMED BY: N.CONSTANTINIDES
                   16  *       CHECKED BY: N.CONSTANTINIDES
                   17          INTERN  ANVT
                   18  * REFERRED MODULES:
                   19          EXTERN  SMPY
                   20          EXTERN  AMHDC
                   21          EXTERN  AMPOT
                   22          EXTERN  AMTB
                   23          EXTERN  AMAC1
                   24          EXTERN  AMDCR
                   25          EXTERN  AMSYN
                   26          EXTERN  AMAC2
                   27          EXTERN  AMOSCL
                   28          EXTERN  AMP5VR
                   29  * GLOBALS
                   30  *       (RAM)
                   31          EXTERN  VMOSBF          CAL-FACT VMOS BUFFER
                   32          EXTERN  VMORBF          CAL-FACT VMOR BUFF
                   33          EXTERN  KSBUF           CAL-FACT KS-BUFF
                   34          EXTERN  KRBUF           CAL-FACT KR BUFF
                   35
                   36  *               R0      =       SCRATCH
                   37  *               R1      =       SCRATCH
=0002              38  ANTYP   EQU     R2      =       AN-TYPE (INPUT)
=0003              39  VTS     EQU     R3      =       1ST AN-DATA (INPUT), XDUCER VALUE (OUTPUT)
=0004              40  VTR     EQU     R4      =       2ND AN-DATA (INPUT)
                   41  *               R5      =       SCRATCH
=0009              42  ATO     EQU     R9      =       AN-TABLES-OFFET (INPUT/OUTPUT)
                   43
                   44          INCLUDE ENCLOS
                   46  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   48          INCLUDE REGDEF          REGISTER DEFENITIONS
                   67          INCLUDE CNSTNT          CONSTANTS
                  195          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  496          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  726          INCLUDE JMPMAC          JUMP MACROS
                  760          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  775          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
```

```
          =0000       1151          RSECT   AMVT
                      1152  ##########################################################################################
```

```
0000'                1154  ANVT
0000'  C80B 0000'    1155          MOV     LINK,@LINKZ     SAVE LINKER
                     1156+         INDEX   ATO,R5,                 OFFSET = TABLES-OFFSET * 2
0004'  C149          1159A           MOV   ATO,R5
0006'  0A15          1160A           SLA   R5,1
0008'  C004          1165          MOV     VTR,R0                  SAVE VMTR
                     1166+         GETVCT  S,                      GET VCTS = (VMTS - VMOS) * KS
000A'  60E5 000B*    1167A           S     @VMOSBF(R5),R3          XS = VMTS - VMOS
000E'  C065 000D*    1185D           MOV   @KSBUF(R5),R1
0012'  06A0 0001*    1193B           BL    @SMPY
0016'  0A23          1195A           SLA   R3,C2                   SC@BO
0018'  09E4          1196A           SRL   R4,C14
001A'  E103          1197A           SOC   R3,R4
001C'  C0C0          1198          MOV     R0,VTS                  R3 = VMTR
001E'  C004          1199          MOV     VTR,R0                  SAVE VCTS
                     1200+         DOIF    ANTYP,GE,C5,X,,         IF TYPE > 4, DUAL-CONV
0020'  0282 0005     1319B           CI    ANTYP,C5
0024'  1109          1376E           JLT   91$
                     1506+           GETVCT R,                             GET VCTR = (VMTR - VMOR) * KR
0026'  60E5 000C*    1507A           S     @VMORBF(R5),R3          XR = VMTR - VMOR
002A'  C065 000E*    1525D           MOV   @KRBUF(R5),R1
002E'  06A0 0001*    1533B           BL    @SMPY
0032'  0A23          1535A           SLA   R3,C2                   SC@BO
0034'  09E4          1536A           SRL   R4,C14
0036'  E103          1537A           SOC   R3,R4
                     1538+         ENDBLK                          IFEND
0038'                1670E  91$
0038'  C0C0          1758          MOV     R0,VTS                  R3 = VCTS (VCTR IN R4)
                     1759+         INDEX   ANTYP,                  TYPE = TYPE * 2
003A'  0A12          1765A           SLA   ANTYP,1
                     1768   *                                      CALL THE SUB-ROUTINE POINTED BY TYPE
                     1769+         CALLPT  SUBTBL,ANTYP            ACCORDING TO THE TYPE SCALE .. VALUE
003C'  C022 0000'    1771A           MOV   @SUBTBL(ANTYP),R0
0040'  0690          1772A           BL    *R0
0042'  C103          1774          MOV     VTS,VTR                 R4 = ANSW
                     1775+         DOIF    VTR,GE,C4096,X,,        IF VT > 4095, VT = 4095 (FFF)
0044'  0284 1000     1894B           CI    VTR,C4096
0048'  1102          1955E           JLT   92$
004A'  0204 0FFF     2081            LI    VTR,C4095
                     2082+         ENDBLK                          IFEND
004E'                2218E  92$
                     2302+         DOIF    VTR,NI,,                IF VT < 0, VT = 0.
004E'  C104          2436B           MOV   VTR,VTR
0050'  1101          2492E           JLT   $+4
0052'  1001          2493E           JMP   93$
0054'  04C4          2608            CLR   VTR
                     2609+         ENDBLK                          IFEND
0056'                2749E  93$
0056'  C2E0 0000'    2829          MOV     @LINKZ,LINK             RESTORE LINKER
005A'  045B          2830          RT
                     2831  ********************************************************************************
                     2832  * PRIVATE AREA
```

```
                    2833+          LOCR    PRIV,LINKZ              LINK-SAVE AREA
    0000'  =0002    2836A  LINKZ   BSS     2
                    2837+          PRVDAT
                    2839   *               SUBR CALLED ACCORDING TO THE TYPE
                    2840+  SUBTBL  DATBL   ANOSCL,ANHDC,ANPOT,ANP5VR,ANTB,ANAC1,ANDCR,ANDCR,ANSYN,ANAC2
    0000'  0009*    2842A            DATA  ANOSCL
    0002'  0002*    2846B            DATA  ANHDC
    0004'  0003*    2850C            DATA  ANPOT
    0006'  000A*    2854D            DATA  ANP5VR
    0008'  0004*    2858E            DATA  ANTB
    000A'  0005*    2862F            DATA  ANAC1
    000C'  0006*    2866G            DATA  ANDCR
    000E'  0006*    2870H            DATA  ANDCR
    0010'  0007*    2874I            DATA  ANSYN
    0012'  0008*    2878J            DATA  ANAC2
                    2896           END
```

```
                     1          IDT     BTACQ
                     2
                     3          SUBTTL  BITE TEST
                     4
                     5  *       CALLING SEQ:    CALLWP  @BTACQW
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   |
                     9  *       BTACQ ACQUIRES BITE DATA (GND AND VOLTAGE) IN CYCLE 1,
                    10  *       IT SENDS BITE MUX ADDR AND RECEIVES GND- AND V-CAL VALUES,
                    11  *       TESTS THE LIMITS AND FOR ERROR SETS ERROR STATUS WORD.
                    12  *                                                                   |
                    13  *-------------------------------------------------------------------+
                    14  *       VERSION : 3
                    15  *       PROGRAMMED BY : N.CONSTANTINIDES
                    16  *       CHECKED BY : N.CONSTANTINIDES
                    17  *
                    18  *
                    19  *
                    20          INTERN  BTACQ
                    21  * REFERD MODULES:
                    22          EXTERN  BTDRQ
                    23          EXTERN  BTLMT
                    24          EXTERN  BTVLMT
                    25          EXTERN  IOB2SC
                    26  * GLOBAL AREA:
                    27  *       (ROM)
                    28          EXTERN  D4              = 4
                    29  *       (RAM)
                    30          EXTERN  CYPFRC          CYCLE-COUNT
                    31          EXTERN  DASFLG          DAS-FLAG
                    32          EXTERN  POCALF          POWER ON CAL/BITE FLAG
=0001               33  TPT     EQU     R1      =       SCRATCH
=0002               34  BTYP    EQU     R2      =       AN-TYPE
=0003               35  VOS     EQU     R3      =       1ST READING
=0004               36  VOT     EQU     R4      =       2ND READING
=0008               37  CNVFLG  EQU     R8      =       CONVERSION FLAG (INPUT, OUTPUT)
=000A               38  BTADP   EQU     R10     =       BT-ADDR-PTR (DOES NOT CHANG FOR BT-PROCESS)
                    39          INCLUDE ENCLOS
                    41  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    43          INCLUDE REGDEF          REGISTER DEFENITIONS
                    62          INCLUDE CNSTNT          CONSTANTS
                   190          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   491          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   721          INCLUDE JMPMAC          JUMP MACROS
                   755          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   770          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000              1146          RSECT   BTACQ
```

```
                    1148  ********************************************************************************
0000'               1149  BTACQ
0000'  0288 0004    1150        CI      CNVFLG,C4               0 TO 4 VALID
0004'  1B32         1151        JH      100$                    EXIT IF JUMP OFFSET INVALID
                    1152  *                                     DO ACCORDING TO CONV-FLAG (0, 2 OR 4)
                    1153+       DOCASE  CNVFLG,50$,51$,52$
0006'  0468 000A'   1154A         B     @234$(CNVFLG)
000A'               1156A 234$
000A'  1002         1159B           JMP   50$
000C'  100C         1162C           JMP   51$
000E'  1019         1165D           JMP   52$
                    1174  ***
0010'               1175  50$
0010'  05C8         1176        INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 1(*2)
0012'  060A         1177        DEC     BTADP                   BT-ADDR-PTR = BT ADDR-PTR - 1
                    1178+       DOIF    ,LEQ,,,,                IF BT-ADDR-PTR = 0,
0014'  1504         1357E           JGT   91$
0016'  020A 0005    1484          LI    BTADP,C5                        BT-ADDR-PTR = 5,
001A'  0720 0008*   1485        SETO    @POCALF                 END OF POWER ON BITE TEST
                    1486+       ENDBLK
001E'               1618E 91$
001E'  04C1         1706        CLR     TPT                     SET POINTER FOR GND-REF
                    1707+       CALL    BTDRQ                   SEND MUX ADDRESSES
0020'  06A0 0001*   1714A         BL    @BTDRQ
0024'  1020         1716        JMP     60$
                    1717  ***
0026'               1718  51$
0026'  05C8         1719        INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 1(*2)
                    1720+       CALL    IOB2SC,<VOT,=,@CADCDB>  READ GND-REF AND CONVERT TO 2S-COMP
0028'  C120 FFB0   1736C         MOV     @CADCDB,VOT
002C'  06A0 0004*   1744A         BL    @IOB2SC
                    1746+       CALL    BTLMT,<R0,=,C6,I>       CHECK LIMITS ON GND-REF
0030'  0200 0006    1756C         LI    R0,C6
0034'  06A0 0002*   1770A         BL    @BTLMT
0038'  C0C4         1772        MOV     VOT,VOS                 SAVE GND-REF VALUE
                    1773+       CALL    BTDRQ,<TPT,=,BTADP>     GET PTR, SEND MUX-ADR TO GET BT-VALUE
003A'  C04A         1789C         MOV   BTADP,TPT
003C'  06A0 0001*   1797A         BL    @BTDRQ
0040'  1012         1799        JMP     60$
                    1800  ***
0042'               1801  52$
0042'  04C8         1802        CLR     CNVFLG                  CONV-FLAG = 0
0044'  C820 0008*   1803        MOV     @POCALF,@DASFLG         DURING POWER ON BITE, POCALF = 2
0048'  0007*
                    1804  *                                     THEN SET TO -1TO DO ANALOG BITE DURING PWER ON
                    1805+       CALL    IOB2SC,<VOT,=,@CADCDB>  READ BITE AND CONVERT TO 2S-COMP
004A'  C120 FFB0   1821C         MOV     @CADCDB,VOT
004E'  06A0 0004*   1829A         BL    @IOB2SC
0052'  6103         1831        S       VOS,VOT                 VOT = VOT - VOS
                    1832+       CALL    BTVLMT,<R0,=,BTADP>     GET PTR, CHECK BITE LIMIT
0054'  C00A         1848C         MOV   BTADP,R0
0056'  06A0 0003*   1856A         BL    @BTVLMT
```

```
                    1858  ***
005A'  C020 0007*   1859          MOV     @DASFLG,R0
005E'  1105         1860          JLT     100$                    JIF NOT IN POWER ON ISR
                    1861+         STARTI  INT6,                   START ADC TO GET ADC INTERRUPT FOR NEXT BITE
0060'  04E0 FFB0    1863A          CLR    @CADCDB
0064'  1002         1865          JMP     100$
                    1866+ 60$     STARTI  INT6,                   START ADC INTERRUPT
0066'  04E0 FFB0    1868A          CLR    @CADCDB
                    1870+        ENDDO                          DOEND
006A'               1872A 90$
006A'  0380         1874  100$    RTWP
                    1875  ********************************************************************************
                    1876          END
```

```
                      1          IDT     BTDRQ
                      2
                      3          SUBTTL  BITE DATA REQUEST
                      4
                      5  *       CALLING SEQ:    CALL    @BTDRQ
                      6
                      7  *-------------------------------------------------------------+
                      8  *                                                             |
                      9  *       BTDRQ SENDS NEXT BITE (POWER SUPPLIE) MUX ADDR TO RECEIVE
                     10  *       BITE DATA.
                     11  *       THE VALUES TO BE SENT TO THE DIFFERENT CRU ADDRESS ARE THE
                     12  *       SAME AND USE A TABLE (SEE BELOW) BUT THE VALUE FOR THE CRU
                     13  *       ADDRESS 330 (HEX) WHICH ARE STORED IN ANOTHER TABLE.
                     14  *                                                             |
                     15  *-------------------------------------------------------------+
                     16  *       VERSION : 2
                     17  *       PROGRAMMED BY : M.CONSTANTINIDES
                     18  *       CHECKED BY : M.CONSTANTINIDES
                     19
                     20
                     21          INTERN  BTDRQ
                     22
                     23
        =0000        24  PTR     EQU     R0      =       SCRATCH
        =0001        25  BUP     EQU     R1      =       OFFSET (INPUT)
                     26
                     27          INCLUDE ENCLOS
                     29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     31          INCLUDE REGDEF          REGISTER DEFENITIONS
                     50          INCLUDE CNSTNT          CONSTANTS
                    178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    709          INCLUDE JMPMAC          JUMP MACROS
                    743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000      1134          RSECT   BTDRQ
                   1135  ********************************************************************
0000'              1136  BTDRQ
0000'  0200 0001'  1137          LI      PTR,BTVL1               SET POINTER TO MUX VALUES
                   1138  *                                       (VALUES IN HEX)
                   1139+         CRUWRT  BTCA1,*PTR+,C8          SEND MUX ADDR 90 TO CRU 318-31F
0004'  020C 0630   1141A           LI    CRU,BTCA1
0008'  3230        1142A           LDCR  *PTR+,C8
                   1144+         CRUWRT  BTCA2,*PTR+,ZERO        SEND MUX ADDR 0 TO CRU 320-32F
000A'  020C 0640   1146A           LI    CRU,BTCA2
000E'  3030        1147A           LDCR  *PTR+,ZERO
                   1149+         CRUWRT  BTCA3,@BTVL3(BUP),C8,   TO CRU 330-337 ACCORDING TO OFFSET
0010'  020C 0660   1151A           LI    CRU,BTCA3
0014'  3221 0006'  1152A           LDCR  @BTVL3(BUP),C8
                   1154+         CRUWRT  BTCA4,*PTR,C8           SEND MUX ADDR 110 TO CRU 340-34F
0018'  020C 0690   1156A           LI    CRU,BTCA4
001C'  3210        1157A           LDCR  *PTR,C8
```

```
                    1830  *                                        TO DO CAL OF ALL CHANNELS DURING POWER ON
                    1831+         CALL    CALRT,<CMT,=,C1,I>       READ AND TEST V-CAL DATA
0058' 0201 0001     1841C           LI    CMT,C1
005C' 06A0 0002*    1855A           BL    @CALRT
                    1857+         CALL    CALKF                    CALCULAT AND STOR KS/KR
0060' 06A0 0003*    1864A           BL    @CALKF
                    1866  ***
0064' C020 0009*    1867          MOV     @DASFLG,R0               JIF NOT ON POWER ON
0068' 1105          1868          JLT     100$
                    1869+         STARTI  INT6,                    START ADC TO GET ADC INTERRUPT FOR NEXT CAL
006A' 04E0 FFB0     1871A           CLR   @CADCDB
006E' 1002          1873          JMP     100$
                    1874+ 60$     STARTI  INT6,            START ADC INTERRUPT
0070' 04E0 FFB0     1876A           CLR   @CADCDB
                    1878+       ENDO                               ENDO
0074'               1880A 90$
0074' 0380          1882  100$    RTWP
                    1883  ********************************************************************************
                    1884          END
```

```
                     1          IDT     CALKF
                     2
                     3          SUBTTL  CALCULATE KS/KR FACTOR FOR CALIBRATION
                     4
                     5  *       CALLING SEQ:    CALL    @CALKF
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       CALKF CALCULATES THE CORRECTION AND SCALING VALUES (KS, KR)
                    10  *       FOR THE CALIBRATION OF THE TRANSDUCER MEASUREMENT.
                    11  *       EQUATION USED:  KS = (VICS * GIS) / (VMCS - VMOS),
                    12  *       IF DUAL-CONVERSION (TYPE > 4), KR AND VMOR REPLACE KS AND
                    13  *       VMOS IN ABOVE EQUATION.
                    14  *                                                                   !
                    15  *-------------------------------------------------------------------+
                    16  *       VERSION : 1
                    17  *       PROGRAMMED BY : N.CONSTANTINIDES
                    18  *       CHECKED BY : N.CONSTANTINIDES
                    19
                    20
                    21          INTERN  CALKF
                    22  * REFERD MODULES:
                    23          EXTERN  SDIV05
                    24
                    25  * GLOBAL AREA:
                    26  *       (RAM)
                    27          EXTERN  KSBUF           KS-BUFF
                    28          EXTERN  KRBUF           KR-BUFF
                    29          EXTERN  VMOSBF          VMOS-BUFF
                    30          EXTERN  VMORBF          VMOR-BUFF
                    31  *       (ROM)
                    32          EXTERN  CAL1FT,CAL2FT           VIC*GI TABLES
                    33
                    34
=0000               35  KFCT    EQU     R0      =       SCRATCH
=0001               36  MPL     EQU     R1      =       SCRATCH
=0002               37  VGP     EQU     R2      =       POINTER TO VIC*GI TABLE (INPUT)
=0003               38  VMCS    EQU     R3      =       1ST CAL READING, VMOS (2'S-C) (INPUT)
=0004               39  VMCR    EQU     R4      =       2ND CAL READING, VMOR (2'S-C) (INPUT)
=0006               40  CAQ     EQU     R6      =       CAL-SLOT-# (INPUT; OUTPUT)
                    41
                    42          INCLUDE ENCLOS
                    44  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    46          INCLUDE REGDEF          REGISTER DEFENITIONS
                    65          INCLUDE CNSTNT          CONSTANTS
                   193          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   494          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   724          INCLUDE JMPMAC          JUMP MACROS
                   758          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   773          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1149          RSECT   CALKF
                  1150  ********************************************************************************
```

```
001E' 045B          1159                    RT
                    1160  ********************************************************************************
                    1161  * PRIVATE DATA AND CONSTANT
                    1162  *
                    1163  *       CRU ADDR
        =0630       1164  BTCA1   EQU     >318*2
        =0640       1165  BTCA2   EQU     >320*2
        =0660       1166  BTCA3   EQU     >330*2
        =0690       1167  BTCA4   EQU     >348*2
                    1168  *
                    1169          PRVDAT
                    1171  *       MUX VALUES
0000' 00            1172  EXT     BYTE    0               START WITH EVEN
0001' 90            1173  BTVL1   BYTE    >90             FOR CRU 318-31F HEX
0002' 0000          1174  BTVL2   DATA    >0000           FOR CRU 320-32F HEX
0004' 11            1175  BTVL4   BYTE    >11             FOR CRU 348-34F HEX
                    1176          EVEN
                    1177  *
                    1178  *       MUX VALUES FOR CRU 330-337 HEX
0006' 20            1179  BTVL3   BYTE    >20             GND, REF
0007' 01            1180          BYTE    >01             +12V, +5V, -15V
0008' 02            1181          BYTE    >02             +28V, ACCEL.
0009' 04            1182          BYTE    >04             +5V, POT
000A' 08            1183          BYTE    >08             +15V, -5V
000B' 10            1184          BYTE    >10             +33V, -28V
                    1185
                    1186          END
```

```
                      1          IDT     BTLMT
                      2
                      3          SUBTTL  CHECK BITE LIMITS
                      4
                      5  *       CALLING SEQ:    CALL    @BTLMT
                      6
                      7  *-------------------------------------------------------------+
                      8  *                                                             |
                      9  *       BTLMT TEST THE LIMITS ON A RECEIVED BITE DATA AND IF IT IS
                     10  *       NOT IN THE GIVEN LIMIT, SETS ERROR STATUS WORD.
                     11  *                                                             |
                     12  *-------------------------------------------------------------+
                     13  *       VERSION : 1
                     14  *       PROGRAMMED BY : M.CONSTANTINIDES
                     15  *       CHECKED BY : M.CONSTANTINIDES
                     16  *       MODIFIED: 03-JAN-84 (M.CONSTANTINIDES)
                     17
                     18
                     19          INTERN  BTLMT
                     20  *** MODULES REFERENCED
                     21          EXTERN  SYSER,SYSOK
                     22
                     23  *       EQU     R0      =       BT ADDR PTR (INPUT) (1 TO 6)
                     24  *       EQU     R1      =       SCRATCH
        =0004        25  BTD     EQU     R4      =       BT-DATA (INPUT)
                     26
                     27          INCLUDE ENCLOS
                     29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     31          INCLUDE REGDEF          REGISTER DEFENITIONS
                     50          INCLUDE CNSTNT          CONSTANTS
                    178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    709          INCLUDE JMPMAC          JUMP MACROS
                    743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000      1134          RSECT   BTLMT
                   1135  ********************************************************************************
0000'  C80B 0000'  1136  BTLMT   MOV     LINK,@LINKZ
0004'  C040        1137          MOV     R0,R1                   R0 FOR ERROR CODE
0006'  0220 0102   1138          AI      R0,>102                 ADD WORD NO.AND BITE ERROR CODE BASE
                   1139+         INDEX   R1,                     PTR1 = BT-ADDR-PTR * 2
000A'  0A11        1145A           SLA   R1,1
000C'  0641        1148          DECT    R1                      LESS 2 (0 TO 10 RANGE)
                   1149+         DOIF    BTD,LEQ,@BTLLMT(R1),,,  IF DATA < LOWER LIMIT,
000E'  8844 0000'  1287B             C    BTD,@BTLLMT(R1)
0012'  1501        1328E              JGT  91$
                   1455+         ORIF    BTD,GE,@BTULMT(R1),,,,  OR DATA > UPPER LIMIT,
0014'  1003        1716E              JMP  82$
0016'              1846E  91$
0016'  8844 000C'  1960B             C    BTD,@BTULMT(R1)
001A'  1104        2002E              JLT  92$
001C'              2152E  82$
```

```
                    2233  *** ANALOG BITE FAILED
001C' C040          2234          MOV     R0,R1                   ERROR CODE >103 TO >108
001E' 06A0 0001*    2235          BL      @SYSER
0022' 1003          2236          JMP     100$
                    2237+         ENDBLK                          ENDIF
0024'               2373E 92$
                    2457  *** ANALOG BITE PASSED
0024' C040          2458          MOV     R0,R1                   ERROR CODE >103 TO >108
0026' 06A0 0002*    2459          BL      @SYSOK
002A' C2E0 0000'    2460  100$    MOV     @LINKZ,LINK
002E' 045B          2461          RT
                    2462  ********************************************************************************
                    2463  * PRICVATE AREA
                    2464
                    2465+         PRVDAT
                    2467  *       BITE LIMITS     (THE LIMITS GIVEN HERE ARE THE LOWER OR UPPER
                    2468  *                        LIMITES - 1 OR + 1, TO RESUALT IN A OUT OF
                    2469  *                        LIMIT VALUE)
                    2470  *       LOWER LIMIT
0000'               2471  BTLLMT
0000' F418          2472          DATA    >F418           +12V, +5V, -15V         BITE 1
0002' EE8F          2473          DATA    >EE8F           +28V, ACCEL             BITE 2
0004' ECCD          2474          DATA    >ECCD           +5V, POT,               BITE 3
0006' F3F0          2475          DATA    >F3F0           +15V, -5V               BITE 4
0008' EE8F          2476          DATA    >EE8F           +33V, -28V              BITE 5
000A' FCCC          2477          DATA    >FCCC           GND, REF (-1.0V)        BITE 6
                    2478  *               UPPER LIMIT
000C'               2479  BTULMT
000C' F950          2480          DATA    >F950           +12V, +5V, -15V
000E' F191          2481          DATA    >F191           +28V, ACCEL
0010' F333          2482          DATA    >F333           +5V, POT,
0012' F979          2483          DATA    >F979           +15V, -5V
0014' 0000          2484          DATA    >0000           +33V, -28V
0016' 0334          2485          DATA    >0334           GND, REF (+1.0V)
                    2486  ************************************************
                    2487  * PRIVATE RAM
                    2488+         LOCR    PRIV,LINKZ
0000' =0002         2491A LINKZ   BSS     2
                    2492  *
                    2493          END
```

```
                    1          IDT     CALAQ
                    2
                    3          SUBTTL  CALIBRATION DATA ACQUISITION
                    4
                    5  *       CALLING SEQ:    CALLWP  @CALAQV
                    6
                    7  *-------------------------------------------------------------+
                    8  *                                                             !
                    9  *       CALAQ ACQUIRES CALIBRARTIN DATA (GND AND VOLTAGE) IN THE
                   10  *       CYCLE 5, 6, 7 AND 8.
                   11  *       IT SENDS CAL MUX ADDR AND RECEIVES GND- AND V-CAL VALUES,
                   12  *       CONVERTS THEM INTO 2S-COMP, SAVES THE GND-CAL VALUES INTO
                   13  *       THE VMOS- VMOR-BUFF, CALCULATES THE CAL-FACTORS KS AND KR
                   14  *       AND SAVES THEM INTO KS- AND KR-BUFF.
                   15  *                                                             !
                   16  *-------------------------------------------------------------+
                   17  *       VERSION : 2
                   18  *       PROGRAMMED BY : N.CONSTANTINIDES
                   19  *       CHECKED BY : N.CONSTANTINIDES
                   20
                   21          INTERN  CALAQ
                   22  * REFERD MODULES:
                   23          EXTERN  ANDRQ
                   24          EXTERN  CALRT
                   25          EXTERN  CALKF
                   26  * GLOBAL AREA:
                   27  *       (RAM)
                   28          EXTERN  VMOSBF          VMOS-BUFF
                   29          EXTERN  VMORBF          VMOR-BUFF
                   30          EXTERN  CYPFRC          CYCLE PER FRAME COUNTER (0 - 31)
                   31          EXTERN  ANPRQ           AN-PARAM-QUANT
                   32          EXTERN  CALMAQ          CAL-MUX-ADDR-QUANT
                   33          EXTERN  DASFLG          DAS-FLAG
                   34          EXTERN  POCALF          POWER-ON CALIB FLAG
                   35  *       (ROM)
                   36          EXTERN  CALMAT          CAL-MUX-ADDR TABLE
                   37          EXTERN  D5              = 5
                   38
                   39  *               R0      =       SCRATCH
=0001              40  CNT     EQU     R1      =       SCRATCH
=0002              41  TYP     EQU     R2      =       AN-TYPE
=0003              42  VG1     EQU     R3      =       1ST READING
=0004              43  VG2     EQU     R4      =       2ND READING
                   44  *               FOLLOWING REG REMAIN THE SAME VARIALBLE
=0006              45  CALAP   EQU     R6      =       CAL-ADDR-QUANTITY
=0007              46  CALSW   EQU     R7      =       CAL-SLOT-#
=0008              47  CNVFLG  EQU     R8      =       CONV-FLAG
                   48
                   49          INCLUDE ENCLOS
                   51  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   53          INCLUDE REGDEF          REGISTER DEFENITIONS
                   72          INCLUDE CNSTNT          CONSTANTS
```

```
                    200         INCLUDE SUBMAC        FUNCTIONAL MACROS
                    501         INCLUDE MSCMAC        MISCELLANEOUS MACROS
                    731         INCLUDE JMPMAC        JUMP MACROS
                    765         INCLUDE BLKMAC        OTHER MACROS (BY D. SCOTT)
                    780         INCLUDE LBLMAC        HANDLES MACROS AUTOMATICALLY
          =0000    1156         RSECT   CALAQ
```

```
                    1158  ********************************************************************************
0000'               1159  CALAQ
0000'  0288 0004    1160          CI      CNVFLG,C4               0 TO 4 VALID
0004'  1B37         1161          JH      100$                    EXIT IF JUMP OFFSET INVALID
                    1162  $                                       DO ACCORDING TO CONV-FLAG
                    1163+       DOCASE    CNVFLG,50$,51$,52$
0006'  0468 000A'   1164A           B     @234$(CNVFLG)
000A'               1166A 234$
000A'  1002         1169B             JMP   50$
000C'  1013         1172C             JMP   51$
000E'  1020         1175D             JMP   52$
                    1184  *** START GND-CAL
0010'               1185  50$
0010'  05C8         1186          INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 2
                    1187+         DOIF    CALSN,LEQ,,,,           IF CAL-SLOT-# = 0,
0012'  C1C7         1321B             MOV   CALSN,CALSM
0014'  1506         1366E               JGT   91$
0016'  C1E0 0007$   1493              MOV   @ANPRQ,CALSM              CAL-SLOT-# = AN-PARAM-QUANT
001A'  C1A0 0008$   1494              MOV   @CALMAQ,CALAP             CAL-ADR-QUN = CAL-MUX-ADR-QUANT
001E'  0720 000A$   1495              SETO  @POCALF                   END OF POWER ON CAL TEST
                    1496+         ENDBLK                          ENDIF
0022'               1628E 91$
0022'  0607         1716          DEC     CALSN                   CAL-SLOT-# = CAL-SLOT-# - 2
0024'  0606         1717          DEC     CALAP                   CAL-ADR-QUNT = CAL-ADR-QUNT - 1
                    1718  $                                       GET CAL-MUX-ADDR TBL ADR & OFFSET
                    1719+         CALL    ANDRQ,<CNT,=,CALMAT,I>,<R9,=,CALAP>   AND SEND MUX ADDR FOR GND
0026'  0201 000B$   1729C             LI    CNT,CALMAT
002A'  C246         1751D             MOV   CALAP,R9
002C'  06A0 0001$   1760A             BL    @ANDRQ
0030'  0606         1762          DEC     CALAP
0032'  101E         1763          JMP     60$
                    1764  *** READ GND-CAL AND START V-CAL
0034'               1765  51$
0034'  05C8         1766          INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 2
0036'  04C1         1767          CLR     CNT                     OFFSET = 0 (GND CAL)
                    1768+         CALL    CALRT                   READ AND TEST RECEIVER DATA
0038'  06A0 0002$   1775A             BL    @CALRT
003C'  C983 0004$   1777          MOV     VG1,@VMOSBF(CALAP)      SAVE 1ST READING INTO VMOS-BUFF
0040'  C984 0005$   1778          MOV     VG2,@VMORBF(CALAP)      SAVE 2ND READING INTO VMRO-BUFF
                    1779  $                                       GET CAL-MUX-ADDR TABLE AND
                    1780+         CALL    ANDRQ,<CNT,=,CALMAT,I>,<R9,=,CALAP>   SEND MUX ADDR FOR VOLTAGE
0044'  0201 000B$   1790C             LI    CNT,CALMAT
0048'  C246         1812D             MOV   CALAP,R9
004A'  06A0 0001$   1821A             BL    @ANDRQ
004E'  1010         1823          JMP     60$
                    1824  *** READ V-CAL AND COMPUTE K FACTORS
0050'               1825  52$
                    1826  $                               START V-CAL
0050'  04C8         1827          CLR     CNVFLG                  CONV-FLAG = 0,
0052'  C820 000A$   1828          MOV     @POCALF,@DASFLG         DURING POWER ON CAL, POCALF=1
0056'  0009$
                    1829  $                                       THEN SET TO -1.
```

```
                    1          IDT     DC1ACQ
                    2
                    3          SUBTTL  DISCRETE #1                DATA ACQUISITION
                    4
                    5  *       CALLING SEQ:    CALLXO  DC1ACR
                    6  *                       XOP 3
                    7
                    8  *-------------------------------------------------------------------+
                    9  *                                                                   |
                   10  *       DC1ACQ GETS THE TABLES-OFFSET, SEND NEXT DC-ADDR,
                   11  *       REQUESTS NEXT  DSCRETE DATA BIT AND PUTS IT INTO
                   12  *       THE DEST OUTPUT BUFF. IT CONTINUES THIS PROCESS
                   13  *       FOR ALL DS-DATA ACCORDING TO THE DS-PARAM-COUNT.
                   14  *                                                                   |
                   15  *-------------------------------------------------------------------+
                   16  *       VERSION : 1
                   17  *       PROGRAMMED BY : M.CONSTANTINIDES
                   18  *       CHECKED BY : M.CONSTANTINIDES
                   19  *       CALLING MODULE: RTISR
                   20  *
                   21          INTERN  DC1ACQ
                   22  * REFERD MODULES:
                   23          EXTERN  DSTINE
                   24          EXTERN  DCDRQ
                   25          EXTERN  TBLOFS
                   26  * GLOBAL AREA
                   27  *       (ROM)
                   28          EXTERN  EAFLG           ERASE BIT FLAG
                   29          EXTERN  DC1GDT,DC1PDT   DC1-DS-TABLE
                   30          EXTERN  DC1DOA          DC1-DS-OFFSET-ARRAY
                   31          EXTERN  DC1MT           DC1-VERS-MASK-TABLE
                   32          EXTERN  DC1MUXT         DC1-MUX-VALUE-TABLE
                   33          EXTERN  SSEG
                   34  *
                   35          EXTERN  DC1DFA,CYPFRC,DC1PQT,DC1OFB,DC1BQT,DC1DOB
                   36  *** TABLE
                   37          EXTERN  DC1I4T,DC1IET
                   38  *       R1, R2, R3, R5  =       SCRATCH
                   39  *       THE FOLLLOWING REG REMAINE THE SAME PARAM FOR DISCRETE DA
      =0006        40  DCF     EQU     R6      =       DC-DATA ACQ FLAG
      =0007        41  DCO     EQU     R7      =       DC-OFFSET-ARRAY
      =0008        42  DCNT    EQU     R8      =       DC-PARAM-COUNT
      =0009        43  DCTO    EQU     R9      =       DC-TABLES-OFFSET
      =000A        44  DCOP    EQU     R10     =       DC-OFFSET-PTR
      =0726        45  H726    EQU     >726    =       DMX #1 CRU ADDRESS
                   46  *
                   47          INCLUDE ENCLOS
                   49  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   51          INCLUDE REGDEF          REGISTER DEFENITIONS
                   70          INCLUDE CNSTNT          CONSTANTS
                  198          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  499          INCLUDE MSCMAC          MISCELLANEOUS MACROS
```

```
                    729          INCLUDE JMPMAC          JUMP MACROS
                    763          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    778          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000      1154          RSECT   DC1ACQ
                   1155  ********************************************************************************
    0000'          1156  DC1ACQ
                   1157  *  DO DISCRETES IN 4 S/F TABLE FOR THIS CYCLE FIRST
                   1158  *** SET UP FOR FOUR S/F TABLE
    0000'  0720 0004*  1159          SETO    @EAFLG
    0004'  0207 000B*  1160          LI      DC0,DC1OFA              START ADDR OF 4 S/F OFFSET TABLE
    0008'  C060 000C*  1161          MOV     @CYFFRC,R1
                       1162+         MOVERJ  @DC1PQT(R1),DCNT        PARA QTY FOR THIS CYCLE
    000C'  D221 000D*  1164A           MOVB  @DC1PQT(R1),DCNT
    0010'  0988        1165A           SRL   DCNT,C8
    0012'  D2A1 0011*  1167          MOVB    @DC1I4T(R1),DCOP        INDEX FOR 4 S/F INDEX TBL
    0016'  088A        1168          SRA     DCOP,C8                 R.J
                       1169+       WHILE     DCNT,GE,,,,             DOWHILE DC-PARAM-COUNT > 0
    0018'              1298E  81$
    0018'  C208        1408B            MOV  DCNT,DCNT
    001A'  111D        1450E             JLT  91$
    001C'  131C        1579           JEQ     91$
                       1580+          CALL    TBLOFS                    NEXT DC-TABLES-OFFSET
    001E'  06A0 0003*  1587A            BL    @TBLOFS
                       1589+          CALL    DCDRQ,<R1,=,H726,I>,<R5,=,@DC1MUX(DCTO),B>   R5 = NEXT ADDR & [illegible]EST NEXT BIT
    0022'  0201 0726   1599C            LI    R1,H726
    0026'  D169 0009*  1618D            MOVB  @DC1MUX(DCTO),R5
    002A'  06A0 0002*  1630A            BL    @DCDRQ
                       1632  *                                          R3, R2 = DC-DS-, TABLE & OFFSET ADDR
    002E'  C0A0 000A*  1633          MOV     @SSEG,R2                      GET ENG CONFIGURATION
                       1634+         DOCASE  R2,50$,60$
    0032'  0462 0036'  1635A            B     @234$(R2)
    0036'              1637A  234$
    0036'  1001        1640B            JMP   50$
    0038'  1007        1643C            JMP   60$
                       1651+  50$    CALL    DSTINE,<R2,=,DC1DOA,I>,<R3,=,DC1PDT,I>  AND DATA INTO DEST
    003A'  0202 0007*  1661C            LI    R2,DC1DOA
    003E'  0203 0006*  1677D            LI    R3,DC1PDT
    0042'  06A0 0001*  1692A            BL    @DSTINE
    0046'  1006        1694           JMP     65$
                       1695+  60$    CALL    DSTINE,<R2,=,DC1DOA,I>,<R3,=,DC1GDT,I>  AND DATA INTO DEST
    0048'  0202 0007*  1705C            LI    R2,DC1DOA
    004C'  0203 0005*  1721D            LI    R3,DC1GDT
    0050'  06A0 0001*  1736A            BL    @DSTINE
                       1738+  65$    ENDO
    0054'              1740A  90$
                       1742+        ENDBLK                            ENDO
    0054'  10E1        1898E             JMP  81$
    0056'              2033E  91$
                       2121  ***
                       2122  *** DO DISCRETE IN EVERY S/F TABLE FOR THIS CYCLE
                       2123  ***
    0056'  0207 000E*  2124          LI      DC0,DC1OFB              START ADDR OF EVERY S/F OFFSET TBL
```

```
                    1147  ********************************************************************************
0000'               1148  CALRT
0000' C80B 0000'    1149          MOV     LINK,@LINKZ                     SAVE LINKER
0004' C801 0002'    1150          MOV     CLF,@BUF1
                    1151+         CALL    AMREAD,<R9,=,CSN>               READ AM-DATA FOR V-CAL
0008' C247          1167C           MOV   CSN,R9
000A' 06A0 0001*    1175A           BL    @AMREAD
                    1177
000E' C142          1178          MOV     TIP,OFST                        PARAM-TYPE AS OFFSET
0010' 0585          1179          INC     OFST
                    1180+         DOIF    CLF,EQ,,,                       IF GND-CAL,
0012' C041          1314B            MOV  CLF,CLF
0014' 1601          1350E             JNE  91$
0016' 04C5          1486            CLR   OFST                              OFFSET = 0
                    1487+         ENDBLK                                  ENDIF
0018'               1619E  91$
                    1707  *                                       SAVE THE VALUE FOR DFDR OUTPUT
                    1708+         MOVBRJ  @CALSOT(OFST),CLF               R1 = OFFSET TO SYS-CAL-BUFF
0018' D065 0005*    1710A           MOVB  @CALSOT(OFST),CLF
001C' 0981          1711A           SRL   CLF,C8
001E' C003          1713          MOV     VG1,RSV
0020' 0810          1714          SRA     RSV,C1
0022' C840 0004*    1715          MOV     RSV,@SYCALB(CLF)                CAL VALUE INTO SYS-CAL-BUFF
                    1716
                    1717  *               TEST THE LIMITS
                    1718
0026' C820 0002'    1719          MOV     @BUF1,@BUF1                     IS IT GND-REF
002A' 0002'
002C' 1611          1720          JNE     85$                             NO JUMP
                    1721
                    1722  *** GND-REF LIMIT TEST
                    1723
                    1724+         CALL    CALMT,<R1,=,OFST>,<R0,=,VG1>    TEST THE LIMIT ON 1ST READING
002E' C045          1740C           MOV   OFST,R1
0030' C003          1756D           MOV   VG1,R0
0032' 06A0 0002*    1765A           BL    @CALMT
                    1767+         DOIF    TIP,GE,C5,I,,                   IF DUAL CONVER,
0036' 0282 0005     1886B             CI  TIP,C5
003A' 1109          1947E              JLT  92$
003C' C020 0006*    2073          MOV     @SYNFLG,R0                      GET SYNCRO CAL FLAG
0040' 1617          2074          JNE     100$                            JIF SET (ERROR IN 1ST READING)
0042' 0225 000B     2075            AI    OFST,C11                          OFFSET FOR 'REF' START AT 11
                    2076+           CALL  CALMT,<R1,=,OFST>,<R0,=,VG2>      TEST LIMIT ON 2ND-READING
0046' C045          2092C           MOV   OFST,R1
0048' C004          2108D           MOV   VG2,R0
004A' 06A0 0002*    2117A           BL    @CALMT
                    2119+         ENDBLK
004E'               2255E  92$
004E' 1010          2339          JMP     100$
                    2340
                    2341  *** V-CAL LIMIT TEST
                    2342
```

```
                    2343+ 85$    CALL   CALMT;<R1,=,OFST>,<R0,=,VG1>   TEST THE LIMIT ON 1ST READING
0050' C045          2359C          MOV   OFST,R1
0052' C003          2375D          MOV   VG1,R0
0054' 06A0 0002*    2384A          BL    @CALMT
                    2386+          DOIF  TIP,GE,C5,I,,                 IF DUAL CONVER,
0058' 0282 0005     2505B            CI  TIP,C5
005C' 1109          2570E             JLT  93$
005E' C020 0006*    2692           MOV   @SYNFLG,R0                    GET SYNCRO CAL FLAG
0062' 1606          2693           JNE   100$                          JIF SET (ERROR IN 1ST READING)
0064' 0225 000B     2694             AI  OFST,C11                        OFFSET FOR 'REF' START AT 11
                    2695+            CALL CALMT,<R1,=,OFST>,<R0,=,VG2>   TEST LIMIT ON 2ND-READING
0068' C045          2711C            MOV OFST,R1
006A' C004          2727D            MOV VG2,R0
006C' 06A0 0002*    2736A            BL  @CALMT
                    2738+          ENDBLK                              ENDIF
0070'               2878E 93$
0070' C2E0 0000'    2958  100$     MOV   @LINKZ,LINK                   RESTORE LINKER
0074' 045B          2959           RT
                    2960  ********************************************************************************
                    2961  * PRIVATE AREA
                    2962+          LOCR  PRIV,LINKZ            LINK-SAVE AREA
0000' =0002         2965A LINKZ    BSS   2
                    2966+          LOCR  PRIV,BUF1
0002' =0002         2969A BUF1     BSS   2
                    2970
                    2971           END
```

```
                        1          IDT     DCDRQ
                        2
                        3          SUBTTL  ACQUIRE DISCRETE DATA BIT
                        4
                        5  *       CALLING SEQ:    CALL    @DCDRQ
                        6
                        7  *---------------------------------------------------------------+
                        8  *                                                               !
                        9  *       DCDRQ SENDS DISCRETE CHANNEL ADDRESS TO DISCRETE PORT,
                       10  *       WAITS FOR 10 USEC AND RECEIVES A DISCRETE BIT.
                       11  *                                                               !
                       12  *---------------------------------------------------------------+
                       13  *       VERSION : 2
                       14  *       PROGRAMMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY : N.CONSTANTINIDES
                       16  *       CALLING MODULE: DC1ACQ,DC2ACQ
                       17
                       18          INTERN  DCDRQ
                       19  *               R1      =       DISCRETE READ CRU ADDRESS (INPUT)
        =0004          20  DCD     EQU     R4      =       RECEIVED DISCRETE BIT (LS BIT) (OUTPUT)
        =0005          21  DCAD    EQU     R5      =       DC-ADDR (LEFT JUSTIF) TO SEND (INPUT)
                       22          INCLUDE ENCLOS
                       24  **      ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       26          INCLUDE REGDEF          REGISTER DEFENITIONS
                       45          INCLUDE CNSTNT          CONSTANTS
                      173          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      474          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      704          INCLUDE JMPMAC          JUMP MACROS
                      738          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      753          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1129          RSECT   DCDRQ
                     1130  ********************************************************************
0000'   04C4         1131  DCDRQ   CLR     DCD             CLEAR DISCRETE DATA REG
0002'   0720 0000'   1132          SETO    @FLG
                     1133+ 20$     CRUWRT  DCRUW,DCAD,C8,          SEND NEXT ADDR
0006'   020C 0680    1135A           LI    CRU,DCRUW
000A'   3205         1136A           LDCR  DCAD,C8
000C'   0A84         1138          SLA     DCD,C8                  WAIT 10 USEC
000E'   C301         1139          MOV     R1,CRU                  DISCRETE READ CRU ADDRESS
0010'   3444         1140          STCR    DCD,C1                  READ ONE DISCRETE BIT
0012'   3444         1141          STCR    DCD,C1
0014'   C820 0000'   1142          MOV     @FLG,@FLG
0018'   0000'
001A'   1303         1143          JEQ     10$
001C'   04E0 0000'   1144          CLR     @FLG
0020'   10F2         1145          JMP     20$
0022'   06C4         1146  10$     SWPB    DCD             BIT RIGHT JUSTIFY(CARD#1 IN BIT0, CARD#2 IN BIT2)
0024'   045B         1147          RT
                     1148  ********************************************************************
                     1149
                     1150  * PRIVATE CONSTANTS
        =06B0        1151  DCRUW   EQU     >340*2          DISCRETE ADDR OUTPUT CRU ADDR (340 H)
```

```
005A' C060 000C*  2125          MOV     @CYPFRC,R1
005E' 0241 0007   2126          ANDI    R1,C7
0062' D221 000F*  2127          MOVB    @DC1BQT(R1),DCNT        PARA QTY FOR THIS CYCLE OF EVERY S/F TBL
0066' 0988        2128          SRL     DCNT,C8                 R.J QTY
0068' D2A1 0012*  2129          MOVB    @DC1IET(R1),DCOF                INDEX FOR EVERY S/F OFFSET TBL
006C' 088A        2130          SRA     DCOF,C8                 R.J
                  2131+       WHILE     DCNT,GE,,,
006E'             2264E  82$
006E' C208        2370B             MOV  DCNT,DCNT
0070' 111D        2416E              JLT  92$
0072' 131C        2541          JEQ     92$
                  2542  ***
                  2543+         CALL    TBLOFS
0074' 06A0 0003*  2550A           BL      @TBLOFS
                  2552  ***
                  2553+         CALL    DCDRQ,<R1,=,H726,I>,<R5,=,@DC1MUX(DCTO),B>
0078' 0201 0726   2563C           LI      R1,H726
007C' D169 0009*  2582D           MOVB    @DC1MUX(DCTO),R5
0080' 06A0 0002*  2594A           BL      @DCDRQ
                  2596  ***
0084' C0A0 000A*  2597          MOV     @SSEG,R2                        GET ENG CONFIGURATION
0088' 0462 008C'  2598          B       @123$(R2)
008C'             2599  123$
008C' 1001        2600          JMP     30$
008E' 1007        2601          JMP     40$
                  2602+ 30$     CALL DSTINE,<R2,=,DC1DOB,I>,<R3,=,DC1PDT,I>
0090' 0202 0010*  2612C           LI      R2,DC1DOB
0094' 0203 0006*  2628D           LI      R3,DC1PDT
0098' 06A0 0001*  2643A           BL      @DSTINE
009C' 1006        2645          JMP     100$
                  2646+ 40$     CALL DSTINE,<R2,=,DC1DOB,I>,<R3,=,DC1GDT,I>
009E' 0202 0010*  2656C           LI      R2,DC1DOB
00A2' 0203 0005*  2672D           LI      R3,DC1GDT
00A6' 06A0 0001*  2687A           BL      @DSTINE
                  2689
00AA'             2690  100$
                  2691+       ENDBLK
00AA' 10E1        2851E             JMP  82$
00AC'             2986E  92$
00AC' 0380        3070        RTWP
                  3071  ********************************************************************************
                  3072          END
```

```
                    1547   *
00E3'  23 25 27     1548   DC1DA1  BYTE    35,37,39
00E6'  00 11 13     1549   DC1DA2  BYTE    0,17,19,21
00E9'  15
00EA'  06 08 0A     1550   DC1DA3  BYTE    6,8,10,12,14,16
00ED'  0C 0E 10
00F0'  01 19 1C     1551   DC1DA4  BYTE    1,25,28,31,34
00F3'  1F 22
00F5'  4B 2A 2D     1552   DC1DA5  BYTE    75,42,45,48,70
00F8'  30 46
00FA'  02 32 34     1553   DC1DA6  BYTE    2,50,52,53,71
00FD'  35 47
00FF'  36 37 38     1554   DC1DA7  BYTE    54,55,56,57,72
0102'  39 48
0104'  03 3A 3C     1555   DC1DA8  BYTE    3,58,60,62,64,66,68,73,77,79,80,82
0107'  3E 40 42
010C'  44 49 4D
010G'  4F 50 52
                    1556   *
                    1557           EVEN
                    1558   *
                    1559   ***************************************************************
                    1560   *
                    1561   *       DISCRETE #1 DESTINATION TABLE
                    1562   *
                    1563   *       FORM: GENDT     N1,N2,N3,N4,N5
                    1564   *
                    1565   *                       N1      =       BUFFER OFFSET WORD
                    1566   *                       N2      =       BUFFER TYPE
                    1567   *                                       0 = DFDR
                    1568   *                                       1 = INTER-CPU
                    1569   *                       N3      =       LEFT SHIFT COUNT
                    1570   *                       N4      =       CLEAR OUTPUT WORD
                    1571   *
                    1572   *                                       1 = CLEAR
                    1573   *                                       0 = NOT CLEAR
                    1574   *                       N5      =       DATA SET END
                    1575   *                                       BLANK = CONTINUE
                    1576   *                                       END   = END
                    1577   *
0110'               1578   DC1PDT
                    1579   *
                    1580   ***************************************************************
                    1581   *
                    1582+          GENDT   9,0,0,0,E                       AIR GND SW          0
0110'  8008         1602A                  DATA    (XYZ*256+9-1)
                    1603+          GENDT   25,0,0,0,E                      AIR GND SW
0112'  8018         1623A                  DATA    (XYZ*256+25-1)
                    1624+          GENDT   41,0,0,0,E                      AIR GND SW
0114'  8028         1644A                  DATA    (XYZ*256+41-1)
                    1645+          GENDT   57,0,0,0,                       AIR GND SW
0116'  0039         1665A                  DATA    (XYZ*256+57-1)
```

```
                    1666+       GENDT   9,1,0,0,E                   TO CPU 2 BUFFER
 0118' 8108         1686A               DATA    (XYZ*256+9-1)
                    1687
                    1688+       GENDT   43,0,0,0,E                  EFIS SYM GEN CAPT      5
 011A' 802A         1708A               DATA    (XYZ*256+43-1)
                    1709+       GENDT   10,1,0,0,E                  TO CPU 2 BUFFER
 011C' 8109         1729A               DATA    (XYZ*256+10-1)
                    1730
                    1731+       GENDT   51,0,0,0,E                  EICAS SW POS
 011E' 8032         1751A               DATA    (XYZ*256+51-1)
                    1752+       GENDT   10,1,1,0,E                  TO CPU 2 BUFFER
 0120' 8509         1772A               DATA    (XYZ*256+10-1)
                    1773
                    1774+       GENDT   51,0,1,0,E                  LE SLATS ALL FULL EX
 0122' 8432         1794A               DATA    (XYZ*256+51-1)
                    1795+       GENDT   9,1,3,0,E                   TO CPU 2 BUFFER        10
 0124' 8008         1815A               DATA    (XYZ*256+9-1)
                    1816
                    1817+       GENDT   51,0,0,0,E                  LE SLATS ALL PART EX
 0126' 8032         1837A               DATA    (XYZ*256+51-1)
                    1838+       GENDT   9,1,4,0,E                   TO CPU 2 BUFFER
 0128' 9103         1858A               DATA    (XYZ*256+9-1)
                    1859
                    1860+       GENDT   43,0,0,0,E                  CAPT'C ADC SW POS
 012A' 802A         1880A               DATA    (XYZ*256+43-1)
                    1881+       GENDT   10,1,11,0,E                 TO CPU 2 BUFFER
 012C' AD09         1901A               DATA    (XYZ*256+10-1)
                    1902
                    1903+       GENDT   43,0,0,0,E                  CAPT'S FMC SW POS      15
 012E' 802A         1923A               DATA    (XYZ*256+43-1)
                    1924+       GENDT   10,1,10,0,E                 TO CPU 2 BUFFER
 0130' A909         1944A               DATA    (XYZ*256+10-1)
                    1945
                    1946+       GENDT   14,0,0,0,                   HF KEYING L
 0132' 000D         1966A               DATA    (XYZ*256+14-1)
                    1967+       GENDT   10,1,7,0,E                  TO CPU 2 BUFFER
 0134' 9D09         1987A               DATA    (XYZ*256+10-1)
                    1988
                    1989+       GENDT   13,0,0,0,                   HF KEYING R
 0136' 000C         2009A               DATA    (XYZ*256+13-1)
                    2010+       GENDT   10,1,8,0,E                  TO CPU 2 BUFFER        20
 0138' A107         2030A               DATA    (XYZ*256+10-1)
                    2031
                    2032+       GENDT   15,0,0,0,                   LE SLATS AGR/DIASS
 013A' 000E         2052A               DATA    (XYZ*256+15-1)
                    2053+       GENDT   9,1,5,0,E                   TO CPU 2 BUFFER
 013C' 9503         2073A               DATA    (XYZ*256+9-1)
                    2074
                    2075+       GENDT   7,0,0,0,E                   LE SLATS INB L EX
 013E' 8006         2095A               DATA    (XYZ*256+7-1)
                    2096+       GENDT   7,0,0,0,E                   LE SLATS INB L EX
 0140' 8006         2116A               DATA    (XYZ*256+7-1)
```

```
                    2117+       GENDT   9,1,6,0,E                   TO CPU 2 BUFFER          25
0140' 9903          2137A               DATA    (XYZ*256+9-1)
                    2138
                    2139+       GENDT   7,0,0,0,E                   LE SLATS OBD L EX
0144' 8005          2159A               DATA    (XYZ*256+7-1)
                    2160+       GENDT   7,0,0,0,E                   LE SLATS OBD L EX
0146' 8005          2180A               DATA    (XYZ*256+7-1)
                    2181+       GENDT   9,1,7,0,E                   TO CPU 2 BUFFER
0148' 9D03          2201A               DATA    (XYZ*256+9-1)
                    2202
                    2203+       GENDT   7,0,1,0,E                   LE SLATS INB R EX
014A' B405          2223A               DATA    (XYZ*256+7-1)
                    2224+       GENDT   7,0,1,0,E                   LE SLATS INB R EX        30
014C' B405          2244A               DATA    (XYZ*256+7-1)
                    2245+       GENDT   9,1,8,0,E                   TO CPU 2 BUFFER
014E' A103          2265A               DATA    (XYZ*256+9-1)
                    2266
                    2267+       GENDT   7,0,1,0,E                   LE SLATS OBD R EX
0150' B405          2287A               DATA    (XYZ*256+7-1)
                    2288+       GENDT   7,0,1,0,E                   LE SLATS OBD R EX
0152' B405          2308A               DATA    (XYZ*256+7-1)
                    2309+       GENDT   9,1,9,0,E                   TO CPU 2 BUFFER
0154' A509          2329A               DATA    (XYZ*256+9-1)
                    2330
                    2331+       GENDT   5,0,0,0,                    VHF KEYING C             35
0156' 0004          2351A               DATA    (XYZ*256+5-1)
                    2352+       GENDT   10,1,6,0,E                  TO CPU 2 BUFFER
0158' 9909          2372A               DATA    (XYZ*256+10-1)
                    2373
                    2374+       GENDT   3,0,0,0,                    VHF KEYING L
015A' 0002          2394A               DATA    (XYZ*256+3-1)
                    2395+       GENDT   10,1,4,0,E                  TO CPU 2 BUFFER
015C' 9109          2415A               DATA    (XYZ*256+10-1)
                    2416
                    2417+       GENDT   4,0,0,0,                    VHF KEYING R
015E' 0003          2437A               DATA    (XYZ*256+4-1)
                    2438+       GENDT   10,1,5,0,E                  TO CPU 2 BUFFER          40
0160' 9509          2458A               DATA    (XYZ*256+10-1)
                    2459
                    2460+       GENDT   43,0,0,0,E                  CAPT'S IRS SW POS
0162' B02A          2480A               DATA    (XYZ*256+43-1)
                    2481+       GENDT   10,1,9,0,E                  TO CPU 2 BUFFER
0164' A509          2501A               DATA    (XYZ*256+10-1)
                    2502
                    2503+       GENDT   62,0,0,0,E                  ENGINE START L
0166' B03D          2523A               DATA    (XYZ*256+62-1)
                    2524+       GENDT   62,0,0,0,E                  ENGINE START L
0168' B03D          2544A               DATA    (XYZ*256+62-1)
                    2545+       GENDT   9,1,1,0,E                   TO CPU 2 BUFFER          45
016A' B503          2565A               DATA    (XYZ*256+9-1)
                    2566
                    2567+       GENDT   62,0,1,0,E                  ENGINE START R
```

```
016C' 843D      2587A           DATA    (XYZ*256+62-1)
                2588+   GENDT   62,0,1,0,E                  ENGINE START R
016E' 843D      2608A           DATA    (XYZ*256+62-1)
                2609+   GENDT   9,1,2,0,E                   TO CPU 2 BUFFER
0170' 8908      2629A           DATA    (XYZ*256+9-1)
                2630
                2631+   GENDT   31,0,0,0,E                  YAW DAMPER ENG L
0172' 801E      2651A           DATA    (XYZ*256+31-1)
                2652+   GENDT   10,1,2,0,E                  TO CPU 2 BUFFER         50
0174' 8909      2672A           DATA    (XYZ*256+10-1)
                2673
                2674+   GENDT   31,0,0,0,E                  YAW DAMPER ENG R
0176' 801E      2694A           DATA    (XYZ*256+31-1)
                2695+   GENDT   10,1,3,0,E                  TO CPU 2 BUFFER
0178' 8D09      2715A           DATA    (XYZ*256+10-1)
                2716
                2717+   GENDT   12,1,0,0,E          A/C TYPE IDENT LSB   - CPU 2 ONLY
017A' 810B      2737A           DATA    (XYZ*256+12-1)
                2738+   GENDT   12,1,1,0,E          A/C TYPE IDENT 2ND   - CPU 2 ONLY
017C' 850B      2758A           DATA    (XYZ*256+12-1)
                2759+   GENDT   12,1,2,0,E          A/C TYPE IDENT MSB   - CPU 2 ONLY       55
017E' 890B      2779A           DATA    (XYZ*256+12-1)
                2780
                2781+   GENDT   13,1,0,0,E          FLEET IDENT LSB      - CPU 2 ONLY
0180' 810C      2801A           DATA    (XYZ*256+13-1)
                2802+   GENDT   13,1,1,0,E          FLEET IDENT MSB      - CPU 2 ONLY
0182' 850C      2822A           DATA    (XYZ*256+13-1)
                2823
                2824+   GENDT   97,0,0,1,                   A/C NUMB IDENT LSB
0184' 4060      2844A           DATA    (XYZ*256+97-1)
                2845+   GENDT   14,1,0,0,E                  TO CPU 2 BUFFER
0186' 810D      2865A           DATA    (XYZ*256+14-1)
                2866
                2867+   GENDT   97,0,1,0,                   A/C NUMB IDENT 2ND      60
0188' 0460      2887A           DATA    (XYZ*256+97-1)
                2888+   GENDT   14,1,1,0,E                  TO CPU 2 BUFFER
018A' 850D      2908A           DATA    (XYZ*256+14-1)
                2909
                2910+   GENDT   97,0,2,0,                   A/C NUMB IDENT 3RD
018C' 0860      2930A           DATA    (XYZ*256+97-1)
                2931+   GENDT   14,1,2,0,E                  TO CPU 2 BUFFER
018E' 890D      2951A           DATA    (XYZ*256+14-1)
                2952
                2953+   GENDT   97,0,3,0,                   A/C NUMB IDENT 4TH
0190' 0C60      2973A           DATA    (XYZ*256+97-1)
                2974+   GENDT   14,1,3,0,E                  TO CPU 2 BUFFER         65
0192' 8D0D      2994A           DATA    (XYZ*256+14-1)
                2995
                2996+   GENDT   97,0,4,0,                   A/C NUMB IDENT 5TH
0194' 1060      3016A           DATA    (XYZ*256+97-1)
                3017+   GENDT   14,1,4,0,E                  TO CPU 2 BUFFER
0196' 910D      3037A           DATA    (XYZ*256+14-1)
```

```
                    3038
                    3039+          GENDT   97,0,5,0,                    A/C NUMB IDENT MSB
01B8' 1460          3059A                  DATA    (XYZ*256+97-1)
                    3060+          GENDT   14,1,5,0,E                   TO CPU 2 BUFFER
01BA' 950D          3080A                  DATA    (XYZ*256+14-1)
                    3081
                    3082+          GENDT   9,1,10,0,E          VIB SEL SWITCH       - CPU 2 ONLY        70
01BC' A903          3102A                  DATA    (XYZ*256+9-1)
                    3103
                    3104+          GENDT   11,1,0,0,E          QAR FAULT            - CPU 2 ONLY
01BE' 810A          3124A                  DATA    (XYZ*256+11-1)
                    3125
                    3126+          GENDT   11,1,1,0,E          TAPE LOW             - CPU 2 ONLY
01C0' 850A          3146A                  DATA    (XYZ*256+11-1)
                    3147
                    3148+          GENDT   67,0,0,0,                    DFDR MAINT FLAG
01C2' 0042          3168A                  DATA    (XYZ*256+67-1)
                    3169+          GENDT   11,1,2,0,E                   TO CPU 2 BUFFER
01C4' 890A          3189A                  DATA    (XYZ*256+11-1)
                    3190
                    3191+          GENDT   38,0,0,0,                    EVENT MARKER              75
01C6' 0025          3211A                  DATA    (XYZ*256+38-1)
                    3212+          GENDT   11,1,3,0,E                   TO CPU 2 BUFFER
01C8' 8D0A          3232A                  DATA    (XYZ*256+11-1)
                    3233
                    3234+          GENDT   101,0,0,0,                   EAROM ERASE DISCRETE
01CA' 0064          3254A                  DATA    (XYZ*256+101-1)
                    3255+          GENDT   11,1,5,0,E                   TO CPU 2 BUFFER
01CC' 950A          3275A                  DATA    (XYZ*256+11-1)
                    3276
                    3277+          GENDT   11,1,4,0,E                   TEST DISCRETE
01CE' 910A          3297A                  DATA    (XYZ*256+11-1)
                    3298
                    3299+          GENDT   97,0,6,0,                    A/C NO
01D0' 1860          3319A                  DATA    (XYZ*256+97-1)
                    3320+          GENDT   14,1,6,0,E                   TO CPU 2 BUFFER
01D2' 990D          3340A                  DATA    (XYZ*256+14-1)
                    3341
                    3342+          GENDT   97,0,7,0,                    A/C NO
01D4' 1C60          3362A                  DATA    (XYZ*256+97-1)
                    3363+          GENDT   14,1,7,0,E                   CPU 2 BUFFER
01D6' 9D0D          3383A                  DATA    (XYZ*256+14-1)
                    3384   ********************************************************************************
                    3385
01D8'               3386   DC1GDT
                    3387+          GENDT   9,0,0,0,E                    AIR GND SW                0
01DA' 8003          3407A                  DATA    (XYZ*256+9-1)
                    3408+          GENDT   25,0,0,0,E                   AIR GND SW
01DC' 8018          3428A                  DATA    (XYZ*256+25-1)
                    3429+          GENDT   41,0,0,0,E                   AIR GND SW
01DE' 8028          3449A                  DATA    (XYZ*256+41-1)
                    3450+          GENDT   57,0,0,0,                    AIR GND SW
```

```
01BE' 0038        3470A                DATA    (XYZ*256+57-1)
                  3471+          GENDT   9,1,0,0,E                    TO CPU 2 BUFFER
01C0' 8108        3491A                DATA    (XYZ*256+9-1)
                  3492
                  3493+          GENDT   43,0,0,0,E                   EFIS SYM GEN CAPT       5
01C2' 802A        3513A                DATA    (XYZ*256+43-1)
                  3514+          GENDT   10,1,0,0,E                   TO CPU 2 BUFFER
01C4' 8109        3534A                DATA    (XYZ*256+10-1)
                  3535
                  3536+          GENDT   51,0,0,0,E                   EICAS SW POS
01C6' 8032        3556A                DATA    (XYZ*256+51-1)
                  3557+          GENDT   10,1,1,0,E                   TO CPU 2 BUFFER
01C8' 8509        3577A                DATA    (XYZ*256+10-1)
                  3578
                  3579+          GENDT   51,0,1,0,E                   LE SLATS ALL FULL EX
01CA' 8432        3599A                DATA    (XYZ*256+51-1)
                  3600+          GENDT   9,1,3,0,E                    TO CPU 2 BUFFER        10
01CC' 8D08        3620A                DATA    (XYZ*256+9-1)
                  3621
                  3622+          GENDT   51,0,0,0,E                   LE SLATS ALL PART EX
01CE' 8032        3642A                DATA    (XYZ*256+51-1)
                  3643+          GENDT   9,1,4,0,E                    TO CPU 2 BUFFER
01D0' 9108        3663A                DATA    (XYZ*256+9-1)
                  3664
                  3665+          GENDT   43,0,0,0,E                   CAPT'C ADC SW POS
01D2' 802A        3685A                DATA    (XYZ*256+43-1)
                  3686+          GENDT   10,1,11,0,E                  TO CPU 2 BUFFER
01D4' AD09        3706A                DATA    (XYZ*256+10-1)
                  3707
                  3708+          GENDT   43,0,0,0,E                   CAPT'S FMC SW POS       15
01D6' 802A        3728A                DATA    (XYZ*256+43-1)
                  3729+          GENDT   10,1,10,0,E                  TO CPU 2 BUFFER
01D8' A909        3749A                DATA    (XYZ*256+10-1)
                  3750
                  3751+          GENDT   14,0,0,0,                    HF KEYING L
01DA' 000D        3771A                DATA    (XYZ*256+14-1)
                  3772+          GENDT   10,1,7,0,E                   TO CPU 2 BUFFER
01DC' 9D09        3792A                DATA    (XYZ*256+10-1)
                  3793
                  3794+          GENDT   13,0,0,0,                    HF KEYING R
01DE' 000C        3814A                DATA    (XYZ*256+13-1)
                  3815+          GENDT   10,1,8,0,E                   TO CPU 2 BUFFER        20
01E0' A109        3835A                DATA    (XYZ*256+10-1)
                  3836
                  3837+          GENDT   15,0,0,0,                    LE SLATS AGR/DIASG
01E2' 000E        3857A                DATA    (XYZ*256+15-1)
                  3858+          GENDT   9,1,5,0,E                    TO CPU 2 BUFFER
01E4' 9508        3878A                DATA    (XYZ*256+9-1)
                  3879
                  3880+          GENDT   7,0,0,0,E                    LE SLATS INB L EX
01E6' 8006        3900A                DATA    (XYZ*256+7-1)
                  3901+          GENDT   7,0,0,0,E                    LE SLATS INB L EX
```

```
01E8' 8006        3921A                DATA    (XYZ*256+7-1)
                  3922+          GENDT   9,1,6,0,E                    TO CPU 2 BUFFER          25
01EA' 9908        3942A                DATA    (XYZ*256+9-1)
                  3943
                  3944+          GENDT   7,0,0,0,E                    LE SLATS OBD L EX
01EC' 8006        3964A                DATA    (XYZ*256+7-1)
                  3965+          GENDT   7,0,0,0,E                    LE SLATS OBD L EX
01EE' 8006        3985A                DATA    (XYZ*256+7-1)
                  3986+          GENDT   9,1,7,0,E                    TO CPU 2 BUFFER
01F0' 9008        4006A                DATA    (XYZ*256+9-1)
                  4007
                  4008+          GENDT   7,0,1,0,E                    LE SLATS INB R EX
01F2' 8406        4028A                DATA    (XYZ*256+7-1)
                  4029+          GENDT   7,0,1,0,E                    LE SLATS INB R EX        30
01F4' 8406        4049A                DATA    (XYZ*256+7-1)
                  4050+          GENDT   9,1,8,0,E                    TO CPU 2 BUFFER
01F6' A108        4070A                DATA    (XYZ*256+9-1)
                  4071
                  4072+          GENDT   7,0,1,0,E                    LE SLATS OBD R EX
01F8' 8406        4092A                DATA    (XYZ*256+7-1)
                  4093+          GENDT   7,0,1,0,E                    LE SLATS OBD R EX
01FA' 8406        4113A                DATA    (XYZ*256+7-1)
                  4114+          GENDT   9,1,9,0,E                    TO CPU 2 BUFFER
01FC' A508        4134A                DATA    (XYZ*256+9-1)
                  4135
                  4136
                  4137+          GENDT   5,0,0,0,                     VHF KEYING C             35
01FE' 0004        4157A                DATA    (XYZ*256+5-1)
                  4158+          GENDT   10,1,6,0,E                   TO CPU 2 BUFFER
0200' 9909        4178A                DATA    (XYZ*256+10-1)
                  4179
                  4180+          GENDT   3,0,0,0,                     VHF KEYING L
0202' 0002        4200A                DATA    (XYZ*256+3-1)
                  4201+          GENDT   10,1,4,0,E                   TO CPU 2 BUFFER
0204' 9109        4221A                DATA    (XYZ*256+10-1)
                  4222
                  4223+          GENDT   4,0,0,0,                     VHF KEYING R
0206' 0003        4243A                DATA    (XYZ*256+4-1)
                  4244+          GENDT   10,1,5,0,E                   TO CPU 2 BUFFER          40
0208' 9509        4264A                DATA    (XYZ*256+10-1)
                  4265
                  4266+          GENDT   43,0,0,0,E                   CAPT'S IRS SW POS
020A' 802A        4286A                DATA    (XYZ*256+43-1)
                  4287+          GENDT   10,1,9,0,E                   TO CPU 2 BUFFER
020C' A509        4307A                DATA    (XYZ*256+10-1)
                  4308
                  4309+          GENDT   62,0,0,0,E                   ENGINE START L
020E' 803D        4329A                DATA    (XYZ*256+62-1)
                  4330+          GENDT   62,0,0,0,E                   ENGINE START L
0210' 803D        4350A                DATA    (XYZ*256+62-1)
                  4351+          GENDT   9,1,1,0,E                    TO CPU 2 BUFFER          45
0212' 8508        4371A                DATA    (XYZ*256+9-1)
```

```
                    4372
                    4373+          GENDT   62,0,1,0,E                ENGINE START R
    0214'  843D     4393A                  DATA    (XYZ*256+62-1)
                    4394+          GENDT   62,0,1,0,E                ENGINE START R
    0216'  843D     4414A                  DATA    (XYZ*256+62-1)
                    4415+          GENDT   9,1,2,0,E                 TO CPU 2 BUFFER
    0218'  890B     4435A                  DATA    (XYZ*256+9-1)
                    4436
                    4437+          GENDT   31,0,0,0,E                YAW DAMPER ENG L
    021A'  801E     4457A                  DATA    (XYZ*256+31-1)
                    4458+          GENDT   10,1,2,0,E                TO CPU 2 BUFFER          50
    021C'  8909     4478A                  DATA    (XYZ*256+10-1)
                    4479
                    4480+          GENDT   31,0,0,0,E                YAW DAMPER ENG R
    021E'  801E     4500A                  DATA    (XYZ*256+31-1)
                    4501+          GENDT   10,1,3,0,E                TO CPU 2 BUFFER
    0220'  8D09     4521A                  DATA    (XYZ*256+10-1)
                    4522
                    4523+          GENDT   12,1,0,0,E          A/C TYPE IDENT LSB   - CPU 2 ONLY
    0222'  810B     4543A                  DATA    (XYZ*256+12-1)
                    4544+          GENDT   12,1,1,0,E          A/C TYPE IDENT 2ND   - CPU 2 ONLY
    0224'  850B     4564A                  DATA    (XYZ*256+12-1)
                    4565+          GENDT   12,1,2,0,E          A/C TYPE IDENT MSB   - CPU 2 ONLY        55
    0226'  890B     4585A                  DATA    (XYZ*256+12-1)
                    4586
                    4587+          GENDT   13,1,0,0,E          FLEET IDENT LSB      - CPU 2 ONLY
    0228'  810C     4607A                  DATA    (XYZ*256+13-1)
                    4608+          GENDT   13,1,1,0,E          FLEET IDENT MSB      - CPU 2 ONLY
    022A'  850C     4628A                  DATA    (XYZ*256+13-1)
                    4629
                    4630+          GENDT   97,0,4,1,                 A/C NUMB IDENT LSB
    022C'  5060     4650A                  DATA    (XYZ*256+97-1)
                    4651+          GENDT   14,1,4,0,E                TO CPU 2 BUFFER
    022E'  910D     4671A                  DATA    (XYZ*256+14-1)
                    4672
                    4673+          GENDT   97,0,5,0,                 A/C NUMB IDENT 2ND       60
    0230'  1460     4693A                  DATA    (XYZ*256+97-1)
                    4694+          GENDT   14,1,5,0,E                TO CPU 2 BUFFER
    0232'  950D     4714A                  DATA    (XYZ*256+14-1)
                    4715
                    4716+          GENDT   97,0,6,0,                 A/C NUMB IDENT 3RD
    0234'  1860     4736A                  DATA    (XYZ*256+97-1)
                    4737+          GENDT   14,1,6,0,E                TO CPU 2 BUFFER
    0236'  990D     4757A                  DATA    (XYZ*256+14-1)
                    4758
                    4759+          GENDT   97,0,7,0,                 A/C NUMB IDENT 4TH
    0238'  1C60     4779A                  DATA    (XYZ*256+97-1)
                    4780+          GENDT   14,1,7,0,E                TO CPU 2 BUFFER          65
    023A'  9D0D     4800A                  DATA    (XYZ*256+14-1)
                    4801
                    4802+          GENDT   97,0,0,0,                 A/C NUMB IDENT 5TH
    023C'  0060     4822A                  DATA    (XYZ*256+97-1)
```

```
                    4823+          GENDT   14,1,0,0,E                    TO CPU 2 BUFFER
023B' 810D          4843A                  DATA    (XYZ*256+14-1)
                    4844
                    4845+          GENDT   97,0,1,0,                     A/C NUMB IDENT MSB
0240' 0460          4865A                  DATA    (XYZ*256+97-1)
                    4866+          GENDT   14,1,1,0,E                    TO CPU 2 BUFFER
0242' 850D          4886A                  DATA    (XYZ*256+14-1)
                    4887
                    4888+          GENDT   9,1,10,0,E          VIB SEL SWITCH      - CPU 2 ONLY       70
0244' A908          4908A                  DATA    (XYZ*256+9-1)
                    4909
                    4910+          GENDT   11,1,0,0,E          QAR FAULT           - CPU 2 ONLY
0246' 810A          4930A                  DATA    (XYZ*256+11-1)
                    4931
                    4932+          GENDT   11,1,1,0,E          TAPE LOW            - CPU 2 ONLY
0248' 850A          4952A                  DATA    (XYZ*256+11-1)
                    4953
                    4954+          GENDT   67,0,0,0,                     DFDR MAINT FLAG
024A' 0042          4974A                  DATA    (XYZ*256+67-1)
                    4975+          GENDT   11,1,2,0,E                    TO CPU 2 BUFFER
024C' 890A          4995A                  DATA    (XYZ*256+11-1)
                    4996
                    4997+          GENDT   38,0,0,0,                     EVENT MARKER              75
024E' 0025          5017A                  DATA    (XYZ*256+38-1)
                    5018+          GENDT   11,1,3,0,E                    TO CPU 2 BUFFER
0250' BD0A          5038A                  DATA    (XYZ*256+11-1)
                    5039
                    5040+          GENDT   101,0,0,0,                    EAROM ERASE DISCRETE
0252' 0064          5060A                  DATA    (XYZ*256+101-1)
                    5061+          GENDT   11,1,5,0,E                    TO CPU 2 BUFFER
0254' 950A          5081A                  DATA    (XYZ*256+11-1)
                    5082
                    5083+          GENDT   11,1,4,0,E                    TEST DISCRETE
0256' 910A          5103A                  DATA    (XYZ*256+11-1)
                    5104
                    5105+          GENDT   97,0,2,0,                     A/C NO
0258' 0B60          5125A                  DATA    (XYZ*256+97-1)
                    5126+          GENDT   14,1,2,0,E                    TO CPU 2 BUFFER
025A' 890D          5146A                  DATA    (XYZ*256+14-1)
                    5147
                    5148+          GENDT   97,0,3,0,                     A/C NO
025C' 0C60          5168A                  DATA    (XYZ*256+97-1)
                    5169+          GENDT   14,1,3,0,E                    CPU 2 BUFFER
025E' BD0D          5189A                  DATA    (XYZ*256+14-1)
                    5190
                    5191           END
```

No errors detected

```
                     1          IDT     DFISR
                     2          SUBTTL  DFDR BUFFER EMPTY ISR
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: DFISR.SRC                        AUTH: N.COSTANTINIDES    *
                     6  * VERSION: 2                             DATE: 26-MAY-1983        *
                     7  *                                                                  *
                     8  * FUNCTION: SENDS THE NEXT DFDR WORD FROM THE DFDR-OUTPUT-QUEU     *
                     9  *           TO DFDR, READS AND TESTS DFDR WRAPAROUND DATA AND      *
                    10  *           IF IT FAILS FOR MORE THAN 255 TIMES CONTINUOUSLY,      *
                    11  *           SETS THE ERROR STATUS WORD.                            *
                    12  *           DFISR HAS A FREQUENCY OF 15.626 MSEC                   *
                    13  *                                                                  *
                    14  * CALLING MODULES: H/W INTERRUPT 5                                 *
                    15  *                                                                  *
                    16  * CALLING SEQ: INTERRUPT 5                                         *
                    17  *                                                                  *
                    18  * INPUTS:                                                          *
                    19  *                                                                  *
                    20  * OUTPUTS:                                                         *
                    21  *                                                                  *
                    22  * MODULES REFERENCED:                                              *
                    23  *              SYSER   =       SET ERROR BIT IN STS ERR WORD       *
                    24  *              SYSOK   =       RESET "   "  "  "  "  "  "         *
                    25  *                                                                  *
                    26  * WORKSPACE AREA: DFW                                              *
                    27  *                                                                  *
                    28  * REGISTERS MODIFIED:R7,R8,R9,R10                                  *
                    29  *                                                                  *
                    30  * VERSION HISTORY:                                                 *
                    31  *                                                                  *
                    32  ********************************************************************
=0000               33         RSECT   DFISR
                    34  *** CALL NAME
                    35         INTERN  DFISR
                    36  *** VARIABLES REFERENCED
                    37         EXTERN  DFDQ                    DFDR OUTPUT QUEU
                    38         EXTERN  RTPCYC                  RT-PER-CYCLE-COUNT
                    39         EXTERN  CYPFRC                  COUNTS 32 CYCLES IN A FRAME
                    40  *** CONSTANTS REFERENCED
                    41         EXTERN  D7                      = 7
                    42         EXTERN  D1                      = 1
                    43  *** TABLES REFERENCED
                    44  *** MODULES REFERENCED
                    45         EXTERN  SYSER,SYSOK,RTSIM1
                    46  *** LIBRARY
                    47         INCLUDE ENCLOS
                    49  ***    ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    51         INCLUDE REGDEF          REGISTER DEFENITIONS
                    70         INCLUDE CNSTNT          CONSTANTS
                   198         INCLUDE SUBMAC          FUNCTIONAL MACROS
                   499         INCLUDE MSCMAC          MISCELLANEOUS MACROS
```

```
                    729          INCLUDE JMPMAC          JUMP MACROS
                    763          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    778          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1154  *** REGISTERS DEFINITION
          =0006    1155  PRW     EQU     R6      =       PRESENT TRANSMIT DFDR DATA WORD
          =0007    1156  WAD     EQU     R7      =       SCRATCH
                   1157  *
                   1158  *       THE FOLLOWING REMAIN THE SAME VARIABLE
                   1159  *
          =0008    1160  DFLWR   EQU     R8      =       LAST TRANSMITTED DFDR DATA WORD
          =0009    1161  DFWRC   EQU     R9      =       DFDR WORD COUNT
                   1162  *                               DF-WORD-COUNT IS ALSO USED AS DFDR-Q POINTER
                   1163  *                               IT RUNS 0-126 (INC BY 2) INSTEAD OF 64
          =000A    1164  DFAILC  EQU     R10     =       DFDR FAIL COUNT
                   1165  ********************************************************************************
```

```
0000'                1167  DFISR
0000'  06A0 0008$    1168          BL      @RTSIM1
                     1169+         DOIF    DFWRC,EQ,C55$2,X,,      IF DF-WR-CNT = 56 (LAST-CYCLE BEGIN),
0004'  0289 006E     1288B            CI    DFWRC,C55$2
0008'  1606          1339E             JNE   91$
                     1475  ***   SYNCHRONIZE DFDR OUTPUT WITH INPUT DADTA REGISTER
000A'  C820 0005$    1476          MOV     @D1,@RTPCYC             SET RT-PER-CYCLE-CNT TO LAST SUB CYCLE
000E'  0002$
0010'  E820 0004$    1477          SOC     @D7,@CYPFRC             SET TO LAST CYCLE OF SUBFRAME   TO
0014'  0003$
                     1478  *                                       SYNCHRONIZE WITH INPUT D/A.
                     1479  *                                       LOWER 3 BITS COUNTS CYCLE 0 TO 7
                     1480+         ENDBLK                          ENDIF
0016'                1612E 91$
                     1700
                     1701+         DOIF    DFWRC,EQ,C63$2,X,,,     IF DF-WR-CNT = 64,
0016'  0289 007E     1820B            CI    DFWRC,C63$2
001A'  1600          1875E             JNE   92$
                     2007+         ENDBLK                          ENDIF
001C'                2143E 92$
                     2227
                     2228+         CRUWRT  CADFO,@DFDQ(DFWRC),C12, SEND NEXT DFDR 12-BIT DATA WORD
001C'  020C 05A0     2230A            LI    CRU,CADFO
0020'  3329 0001$    2231A            LDCR  @DFDQ(DFWRC),C12
                     2233
0024'  C1E0 FF94     2234          MOV     @CKDFWA,WAD             READ THE WARP-AROUND DATA
0028'  0247 0FFF     2235          ANDI    WAD,CFFFH               (ONLY LS 12 BITS)
                     2236+         DOIF    WAD,NE,DFLWR,,,,                IF DF-WR-WORD NOT= LAST-TRANSM DF-DATA,
002C'  8207          2374B            C     WAD,DFLWR
002E'  1309          2411E             JEQ   93$
0030'  058A          2542             INC   DFAILC                    DF-FAIL-CNT = DF-FAIL-CNT + 1.
                     2543+            DOIF  DFAILC,GE,C100H,X,,,      IF FAIL-COUNT > 255
0032'  028A 0100     2662B            CI    DFAILC,C100H
0036'  1104          2731E             JLT   94$
                     2849  ***  DFDR WRAPAROUND FAILURE
0038'  C060 0000'    2850          MOV     @ERCOD,R1               DFDR WRAPAROUND ERROR CODE
003C'  06A0 0006$    2851          BL      @SYSER                  SET ERROR BIT IN SYEBF
                     2852+            ENDBLK                          ENDIF
0040'                2996E 94$
                     3072+         ELSEDO                          ELSE
0040'  1005          3345E             JMP   95$
0042'                3471E 93$
                     3548  ***  DFDR WRAPAROUND PASSED
0042'  C060 0000'    3549          MOV     @ERCOD,R1               DFDR WRAPAROUND ERROR CODE
0046'  06A0 0007$    3550          BL      @SYSOK                  RESET ERROR BIT IN SYEBF
004A'  04CA          3551             CLR   DFAILC                    DF-FAIL-CNT = 0.
                     3552+         ENDBLK                          ENDIF
004C'                3700E 95$
004C'  C206          3772          MOV     PRW,DFLWR               LAST-DFDR-DATA = PREV-DFDR-DATA
004E'  C1A9 0001$    3773          MOV     @DFDQ(DFWRC),PRW                PREV-DFDR-DATA = DF-DATA
0052'  0248 0FFF     3774          ANDI    DFLWR,CFFFH             (12 BIT WORD)
0056'  05C9          3775          INCT    DFWRC                   DF-WR-CNT = DF-WR-CNT + 2
```

```
0058'  0249 007F    3776          ANDI    DFWRC,C7FH              IF DF-WR-CNT = 64 (*2), DF-WR-CNT = 0.
005C'  0380         3777          RTWP
                    3778  ********************************************************************************
                    3779  * PRIVATE CONSTANTS / DATA
                    3780  *
                    3781+         PRVDAT
0000'  0102         3783  ERCOD   DATA    >102            DFDR-WA ERROR CODE
                    3784          END
```

```
                     1          IDT     DR1ISG
                     2
                     3          SUBTTL  DITS #1 DATA READY ISR FOR GE
                     4
                     5  *       CALLING SEQ:    INTERRUPT 8
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       DR1ISR READS RECEIVED DITS DATA, EXTRACT 12-BIT REQUESTED
                    10  *       DATA DR1D PUTS THE DATA INTO DESTINATION BUFFER.   THEN IF
                    11  *       THERE ARE MORE DITS #1 DATA TO ACQUIRE, IT SENDS THE NEXT
                    12  *       ADDR TO DITS # 1 PORT.
                    13  *                                                                   !
                    14  *-------------------------------------------------------------------+
                    15  *       VERSION : 1
                    16  *       PROGRAMMED BY : N.CONSTANTINIDES
                    17  *       MODIFIED : 4-MAY-1983 N.CONSTANTINIDES
                    19          INTERN  DR1ISG
                    20  * REFERD MODULES:
                    21          EXTERN  DRXTK
                    22          EXTERN  DSTINW
                    23          EXTERN  DRSAW
                    24  * GLOBAL AREA:
                    25  *       (ROM)
                    26          EXTERN  DR1DT           DR1-DS-TABLE
                    27          EXTERN  DR1PST          DR1-DATA-POS TABLE
                    28          EXTERN  DR1PNT          DR1-PARAM-NUMBER-TABLE
                    29          EXTERN  DR1DAG          DR1-DS-OFFSET-ARRAY
                    30          EXTERN  DR1UPT          DR1-CHANNEL UPDATE TABLE
                    31  *
                    32          EXTERN  DR1A1G,DR1IEG,DR1BRG
                    33          EXTERN  DR1A2G,CYFFRC,DR1DBG
                    34  *
                    35  *       R1, R2, R3      =       SCRATCH
                    36  *       THE FOLLWOING REG REMAINE SAME VARIABLE FOR THE WHOLE DITS # 1 PROCESS
=0007               37  D1OA    EQU     R7      =       DR1-OFFSET-ARRAY
=0008               38  D1PC    EQU     R8      =       DITS #1 PARAM COUNT (INPUT; OUTPUT)
=0009               39  D1TO    EQU     R9      =       DITS #1 TABLES OFFSET (INPUT; OUTPUT)
=000A               40  D1O     EQU     R10     =       DITS #1 ARRAY OFFSET  (INPUT; OUTPUT)
=FF86               41  CD1IM   EQU     >FF86
=FF84               42  CD1IL   EQU     >FF84
                    43  *
                    44          INCLUDE ENCLOS
                    46  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    48          INCLUDE REGDEF          REGISTER DEFENITIONS
                    67          INCLUDE CNSTNT          CONSTANTS
                   195          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   496          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   726          INCLUDE JMPMAC          JUMP MACROS
                   760          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   775          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
```

```
          =0000        1151           RSECT   DR1ISG
```

```
                    1153  ********************************************************************************
0000' C80B 0000'    1154  DR1ISG  MOV     R11,@LINKZ
0004' C120 FF84     1155         MOV     @CD1IL,R4       LSW OF DITS DATA
0008' 06C4          1156         SWPB    R4                      PUT LABEL IN LOWER BYTE
000A' C0E0 FF86     1157         MOV     @CD1IM,R3               MSW OF DITS DATA
                    1158  *    R2 = DR1-POS-TBL ADDR AND EXTRACT 12-BIT
                    1159+        CALL    DRXTK,<R2,=,DR1PST,I>,<R0,=,DR1UPT,I>
000E' 0202 0005*    1169C          LI    R2,DR1PST
0012' 0200 0008*    1185D          LI    R0,DR1UPT
0016' 06A0 0001*    1200A          BL    @DRXTK
                    1202  ***
001A' 0202 000E*    1203         LI      R2,DR1DBG
                    1204+        DOIF    D1OA,EQ,@DR1A1G,,,
001E' 8807 0009*    1342B          C     D1OA,@DR1A1G
0022' 1602          1374E           JNE   91$
0024' 0202 0007*    1510         LI      R2,DR1DAG               4 S/F DEST OFFSET TBL
                    1511+        ENDBLK
0028'               1643E  91$
                    1731  *                                      R3 = PAR-DS-TBL ADDR, R2= DS-OFS-ARY
                    1732  *                                      AND DATA INTO DESTINATION
                    1733+        CALL    DSTINW,<R3,=,DR1DT,I>
0028' 0203 0004*    1743C          LI    R3,DR1DT
002C' 06A0 0002*    1757A          BL    @DSTINW
                    1759  *                                      R4 = DR1 OUTPUT-RESET PORT, R5= PARAM-
                    1760  *                                      NUMBER ADDR, R2= MASK-TBL  AND SEND ADDR
                    1761+        CALL    DRSAW,<R4,=,CD1IOP,I>,<R5,=,DR1PNT,I>
0030' 0204 FF84     1771C          LI    R4,CD1IOP
0034' 0205 0006*    1787D          LI    R5,DR1PNT
0038' 06A0 0003*    1802A          BL    @DRSAW
                    1804+        DOIF    D1PC,MI,,,,             DOIF ACQUISITION DONE FOR THIS CYCLE
003C' C208          1938B          MOV   D1PC,D1PC
003E' 1101          1990E           JLT  $+4
0040' 1017          1991E           JMP  92$
                    2110+               DOIF    D1OA,EQ,@DR1A1G,,,      DO IF IT WAS FOR 4 S/F TBL
0042' 8807 0009*    2248B          C     D1OA,@DR1A1G
0046' 1613          2288E           JNE  93$
0048' C1E0 000C*    2416                MOV     @DR1A2G,D1OA    SET TO EVERY S/F TABLE START ADDRESS
004C' C060 000D*    2417                MOV     @CYPFRC,R1      32 CYCLE COUNTER (FRAME)
0050' 0241 0007     2418                ANDI    R1,C7           GET CYCLE COUNTER/SUBF
0054' D221 000B*    2419                MOVB    @DR1BOG(R1),D1PC        NO OF PARS FOR THIS CYCLE
0058' 0988          2420                SRL     D1PC,CB
005A' 0A11          2421                SLA     R1,1            CHANGE TO EVEN WORD ADDR
005C' C2A1 000A*    2422                MOV     @DR1IEG(R1),D1O         INDEX FOR EVERY S/F OFFSET TABLES
                    2423+               CALL    DRSAW,<R4,=,CD1IOP,I>,<R5,=,DR1PNT,I>
0060' 0204 FFB4     2433C          LI    R4,CD1IOP
0064' 0205 0006*    2449D          LI    R5,DR1PNT
0068' 06A0 0003*    2464A          BL    @DRSAW
006C' 1000          2466                NOP
                    2467+               ENDBLK
006E'               2607E  93$
006E' 1000          2687         JMP     92$             EXIT
                    2688+        ENDBLK
```

```
0070'                2824E  92$
0070'  C2E0 0000'    2908           MOV     @LINKZ,R11
0074'  045B          2909           RT
                     2910   ********************************************************************************
                     2911+          LOCR    PRIV,LINKZ
0000'  =0002         2914A  LINKZ   BSS     2
                     2915
                     2916           END
```

No errors detected

```
                         1          IDT     DR1ISR
                         2
                         3          SUBTTL  DITS #1 DATA READY ISR
                         4
                         5  *       CALLING SEQ:    INTERRUPT 8
                         6
                         7  *-------------------------------------------------------------+
                         8  *                                                             !
                         9  *       DR1ISR READS RECEIVED DITS DATA, EXTRACT 12-BIT REQUESTED
                        10  *       DATA DR1D PUTS THE DATA INTO DESTINATION BUFFER.   THEN IF
                        11  *       THERE ARE MORE DITS #1 DATA TO ACQUIRE, IT SENDS THE NEXT
                        12  *       ADDR TO DITS # 1 PORT.
                        13  *                                                             !
                        14  *-------------------------------------------------------------+
                        15  *       VERSION : 1
                        16  *       PROGRAMMED BY : N.CONSTANTINIDES
                        17  *       MODIFIED : N.CONSTANTINIDES 26-MAY-1983
                        19          INTERN  DR1ISR
                        20  * REFERD MODULES:
                        21          EXTERN  DR1ISP
                        22          EXTERN  DR1ISG,RTSIN1
                        23          EXTERN  SSEG
        =0000           24          RSECT   DR1ISR
                        25  *
                        26  ********************************************************************
    0000'               27  DR1ISR
    0000'  06A0 0003*   28          BL      @RTSIN1
    0004'  C820 0004*   29          MOV     @SSEG,@SSEG             CHECK CONFIGURATION
    0008'  0004*
    000A'  1603         30          JNE     10$
    000C'  06A0 0001*   31          BL      @DR1ISP                 PW
    0010'  1002         32          JMP     100$
    0012'  06A0 0002*   33  10$     BL      @DR1ISG                 GE
                        34
    0016'  0380         35  100$    RTWP
                        36  ********************************************************************
                        37          END
```

```
                    1          IDT     DRXTK
                    2          SUBTTL  EXTRACT 12 BITS FROM RECEIVED DITS DATA
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: DRXTK.SRC(TWA,MAN)              AUTH: M.COSTANTINIDES     *
                    6  * VERSION: 2                            DATE:   15-NOV-1982       *
                    7  *                                                                  *
                    8  * FUNCTION: GETS THE DATA-POS VALUE FROM DR-POS-TABLE, SHIFTS      *
                    9  *           DATA TO THE RIGHT ACCORDING TO THE SHIFT COUNT AND     *
                   10  *           IF SIGN IS REQUESTED, SIGN BIT/BITS FROM BIT          *
                   11  *           29/30-31 OF THE RECEIVED DATA SHALL BE ATTACHED       *
                   12  *           NEXT TO THE MSB OF DATA.                              *
                   13  *           IF ROUND IS SELECTED, ONE SHALL BE ADDED AND          *
                   14  *           RIGHT SHIFTED BY ONE.                                 *
                   15  *           IF PARITY/SSM CHECK IS SELECTED, THE PARITY FAILURE   *
                   16  *           SHALL SET THE DATA TO ALL ONES INCLUDING SIGN BIT.     *
                   17  *           FOR SSM FAILURE THE DATA SHALL BE SET TO ALL ONES      *
                   18  *           INCLUDING SIGN LESS ONE.                              *
                   19  *           IF SSM DISCRETE IS SELECTED, THE FAILURE CODE OF      *
                   20  *           2-BIT SSM FIELD SHALL BE CONVERTED TO ONE BIT         *
                   21  *           LOGIC '0' OR '1'.                                     *
                       *                                                                 *
                       * CALLING MODULES: DR1ISR, DR2ISR                                *
                   24  *                                                                 *
                   25  * CALLING SEQ: CALL @DRXTK                                       *
                   26  *                                                                 *
                   27  * INPUTS: R0=DR CHANNEL UPDATE TABLE START ADDRESS               *
                   28  *         R2=DR-POS TABLE START ADDRESS                          *
                   29  *         R3,R4=DITS RECEIVED DATA                               *
                   30  *         R5=PRESENT PARAMETER NUMBER                            *
                   31  *         R9=DR-TABLE-OFFSET                                     *
                   32  *                                                                 *
                   33  * OUTPUTS: R4=SIGNED ROUNDED RIGHT JUSTIFIED DITS DATA           *
                   34  *          R6=DITS PORT CHANNEL 0 TO 7 UPDATE STATUS(BIT0-7)     *
                   35  *         R12=PARITY/SSM ERROR STATUS                            *
                   36  *              0 FOR NO ERROR,-1 FOR PARITY,+1 FOR SSM ERROR.    *
                   37  *                                                                 *
                   38  * MODULES REFERENCED: DSRLS     RIGHT JUSTIFY DATA               *
                   39  *                     DRSIGN    SET SIGN BIT                     *
                   40  *                     DRPSSM    PARITY/SSM CHECK                 *
                   41  *                     DRSSMD    SSM DISCRETE                     *
                   42  *                                                                 *
                   43  * WORKSPACE AREA: CALLER'S                                       *
                   44  *                                                                 *
                   45  * REGISTERS MODIFIED: R0,R1,R2,R4,R5,R6,R12                      *
                   46  *                                                                 *
                   47  * VERSION HISTORY:                                               *
                   48  *                                                                 *
                   49  ********************************************************************
=0000              50         RSECT   DRXTK
                   51  *** CALL NAME
                   52         INTERN  DRXTK
```

```
                        53  *** VARIABLES REFERENCED
                        54  *** CONSTANTS REFERENCED
                        55          EXTERN  B0,B9,B11,B12,D29,XFF
                       .56  *** TABLES REFERENCED
                        57          EXTERN  DAINTB
                        58  *** MODULES REFERENCED
                        59          EXTERN  DSRLS,DRSIGN,DRPSSM,DRSSMD
                        60  *** LIBRARY
                        61          INCLUDE ENCLOS
                        63  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                        65          INCLUDE REGDEF          REGISTER DEFENITIONS
                        84          INCLUDE CNSTNT          CONSTANTS
                       212          INCLUDE SUBMAC          FUNCTIONAL MACROS
                       513          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                       743          INCLUDE JMPMAC          JUMP MACROS
                       777          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                       792          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                      1168  *** REGISTERS DEFINITION
                      1169  *               R0      =       SCRATCH
         =0001        1170  DDT     EQU     R1      =       SCRATCH
         =0002        1171  DRP     EQU     R2      =       DR-POS TABLE START ADDRESS (INPUT)
         =0003        1172  DMS     EQU     R3      =       MS-16-BIT OF DATA (INPUT)
         =0004        1173  DLS     EQU     R4      =       LS-16-BIT OF DATA (INPUT); DATA (OUTPUT)
                      1174  *       EQU     R5      =       PRESENT PARA NUMBER AND LINK SAVE WORD
         =0006        1175  DRXUP   EQU     R6      =       DITS PORT UPDATE STATUS
         =0009        1176  DRTO    EQU     R9      =       DR-TABLES-OFFSET (INPUT)
                      1177  *               R12     =        PARITY/ SSM ERROR FLAG, -1 = ERROR, 0 FOR OK
                      1178  *                                                               +1 = SSM ERROR
                      1179  ********************************************************************************
    0000'             1180  DRXTK
    0000'  A089       1181          A       DRTO,DRP        POINTER=OFFSET+POS-TBL
    0002'  C092       1182          MOV     *DRP,DRP        DR-DATA-POS-TABLE DATA WORD
                      1183  *** DITS PORT UPDATE CHECK, R6 CONTAINS THE STATUS
    0004'  04C1       1184          CLR     R1              CHANNEL OR LOOP COUNTER
    0006'  8C15       1185  LOOP    C       *R5,*R0+        PARA NO. VERSUS UPPER PARA NO.OF THIS CHANNEL
    0008'  1205       1186          JLE     FOUND           JIF FOUND IN THE TABLE
    000A'  05C1       1187          INCT    R1              BUMP
    000C'  0281 0012  1188          CI      R1,18
    0010'  12FA       1189          JLE     LOOP            JIF NOT THE LAST CHANNEL
    0012'  1005       1190          JMP     NOFIND          PARA NO. NOT IN THE TABLE
    0014'  2520 0006* 1191  FOUND   CZC     @XFF,DLS        LSW OF DITS DATA
    0018'  1302       1192          JEQ     NOFIND          JIF UPDATE BAD(LABEL NOT PRESENT)
                      1193  *   CHANNEL  UPDATE OK FOR THIS CHANNEL
    001A'  41A1 0001* 1194          SZC     @B0(R1),DRXUP   RESET NO-UPDATE FLAG TO OK FOR THIS CHANNEL
                      1195  *** PARITY AND SSM ERROR CHECK IF REQUESTED
    001E'  C14B       1196  NOFIND  MOV     LINK,R5         SAVE LINK REGISTER
    0020'  04CC       1197          CLR     R12                     RESET SSM ERROR FLAG
    0022'  20A0 0004* 1198          COC     @B12,DRP                DATA WORD OF POSITION TABLE
    0026'  1602       1199          JNE     10$
    0028'  06A0 000A* 1200          BL      @DRPSSM                 CHECK PARITY AND SSM ERRORS
    002C'  C043       1201  10$     MOV     DMS,DDT                 SAVE MSH OF INPUT DATA FOR SIGN
                      1202  ***     R.J. INPUT DATA USING I/P POSITION OF POS TBL
```

```
002E' C002        1203          MOV   DRP,R0              INPUT POS IN R0 FOR DSRLS
0030' 06A0 0008$  1204          BL    @DSRLS              R.J. 16 BIT INPUT DATA IN R4
                  1205  ***     MASK INPUT DATA USING I/P MASK WORD INDEX OF POS TBL
0034' C0C2        1206          MOV   DRP,R3
0036' 0943        1207          SRL   R3,C4               R.J. INPUT MASK WORD INDEX * 2
0038' 0243 001E   1208          ANDI  R3,>1E              R3=INPUT MASK WORD INDEX(FOR DRSIGN ALSO)
003C' 4123 0007$  1209          SZC   @DAIMTB(R3),DLS     CLEAR UNWANTED UPPER BITS
                  1210  *** CHECK IF SSM ERROR DISCRETE TYPE REQUESTED
0040' 20A0 0005$  1211          COC   @D29,DRP
0044' 1602        1212          JNE   20$                 JIF NOT SSM ERROR DISCRETE TYPE
0046' 06A0 000B$  1213          BL    @DRSSMD             INPUT: R2,R4
                  1214  ***     SET SIGN IN INPUT DATA USING SIGN AND BINARY/BCD ARGS OF POS TBL
004A' 20A0 0002$  1215  20$     COC   @B9,DRP
004E' 1602        1216          JNE   30$                 JIF NO SIGN
0050' 06A0 0009$  1217          BL    @DRSIGN             SET SIGN BIT. INPUT: R0,R1,R3,R4
                  1218  ***     ROUNDING
0054' C002        1219  30$     MOV   DRP,R0
0056' 1505        1220          JGT   40$                 JIF NO ROUNDING
0058' 1304        1221          JEQ   40$                 JIF NO ROUNDING
005A' C30C        1222          MOV   R12,R12
005C' 1601        1223          JNE   35$                 JIF SSM/PARITY ERROR
005E' 0584        1224          INC   DLS                 ELSE, BUMP LSB
0060' 0914        1225  35$     SRL   DLS,C1              GET RID OF LSB
                  1226  *** CHECK IF SSM ERROR ONLY, NOT PARITY.
0062' C30C        1227  40$     MOV   R12,R12
0064' 1303        1228          JEQ   60$                 EXIT IF NOT SSM ERROR
0066' 1102        1229          JLT   60$                 EXIT IF PARITY ERROR (DATA =-1)
0068' 0244 FFFE   1230          ANDI  DLS,>FFFE           CLEAR LSB OF STRIPPED I/P DATA TO INDICATE SSM ERROR
                  1231  ***     EXIT
006C' C2C5        1232  60$     MOV   R5,LINK             RESTORE LINK REGISTER
006E' 045B        1233          RT
                  1234          END
```

```
                        1          IDT     DRSAW
                        2
                        3          SUBTTL  SEND NEXT DITS ADDR
                        4
                        5  *       CALLING SEQ:    CALL    @DRSAW
                        6
                        7  *-------------------------------------------------------------+
                        8  *                                                             |
                        9  *       DRSAW SETS THE DITS PARAM COUNT AND IF THERE ARE MORE DITS
                       10  *       DATA TO TRANSFER, SENDS THE NEXT DITS ADDR AND RESETS THE
                       11  *       INTERRUPT.
                       12  *       2 BYTE DATA SET IN DITS OFFSET TABLE                  |
                       13  *                                                             |
                       14  *       CALLING MODULE: DR1ISR                                |
                       15  *                                                             |
                       16  *-------------------------------------------------------------+
                       17  *       VERSION : 1
                       18  *       PROGRAMMED BY : N.CONSTANTINIDES
                       19  *       CHECKED BY : N.CONSTANTINIDES
                       20
                       21
                       22          XNTERN  DRSAW
                       23
          =0000        24  IND     EQU     R0      =       SCRATCH
          =0002        25  VMK     EQU     R2      =       VERSION MASK (INPUT)
          =0003        26  LINKS   EQU     R3      =       LINKER SAVE AREA
          =0004        27  DPA     EQU     R4      =       DITS OUTPUT AND RESET ADDR (INPUT)
          =0005        28  PNT     EQU     R5      =       DR-PARAMETER-NUMBER-TABLE (INPUT)
          =0007        29  DOA     EQU     R7      =       DR-OFFSET-ARRAY (INPUT; OUTPUT)
          =0008        30  DPC     EQU     R8      =       DITS  PARAM COUNT (INPUT; OUTPUT)
          =0009        31  DTO     EQU     R9      =       DITS  TABLES OFFSET (INPUT; OUTPUT)
          =000A        32  DOP     EQU     R10     =       DITS  OFFSET POINTER (INPUT; OUTPUT)
                       33          INCLUDE ENCLOS
                       35  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       37          INCLUDE REGDEF          REGISTER DEFENITIONS
                       56          INCLUDE CNSTNT          CONSTANTS
                      184          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      485          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      715          INCLUDE JMPMAC          JUMP MACROS
                      749          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      764          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000      1140          RSECT   DRSAW
                     1141  ********************************************************************
0000'     0608       1142  DRSAW   DEC     DPC                     PARAM-CNT=PARAM-CNT-1
                     1143+         DOIF    ,GE,,,,                 IF PARAM-COUNT >=0;
0002'     1107       1319E             JLT   91$
0004'     05CA       1449            INCT  DOP                     PARM-ARRAY-OFFS=PARAM-SRY-OFS+1
0006'     C007       1450            MOV   DOA,IND         GET NEXT OFFSET FROM ARRAY TABLE
0008'     A00A       1451            A     DOP,IND
000A'     C250       1452            MOV   *IND,DTO
000C'     0A19       1453            SLA   DTO,1                   TABLES-OFFSET = TABLES-OFFSET *2
000E'     A149       1454            A     DTO,PNT
```

```
0010'  C515        1455            MOV   $FNT,$DFA                SEND WORD ADDRESS TO DR
                   1456+         ENDBLK                         ENDIF
0012'              1588E  91$
0012'  045B        1676          RT
                   1677  ##########################################################################
                   1678
                   1679          END
```

No errors detected

```
                        1          IDT     DRSSMD
                        2          SUBTTL
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: DRSSMD.SRC                        AUTH: N.COSTANTINIDES   *
                        6  * VERSION: 1                              DATE: 29 MAR 1982       *
                        7  *                                                                  *
                        8  * FUNCTION: IF REQUESTED IN THE POSITION TABLE, IT DECODES THE    *
                        9  *          FAILURE CODES OF SSM FIELD AND SET TO DISCRETE          *
                       10  *          LOGIC 0 (NO FAILURE) OR LOGIC 1 (FAILURE).              *
                       11  *                                                                  *
                       12  * CALLING MODULES: DRXTK.SRC                                      *
                       13  *                                                                  *
                       14  * CALLING SEQ: BL @DRSSMD                                         *
                       15  *                                                                  *
                       16  * INPUTS:       R4      =       STRIPPED DITS INPUT DATA          *
                       17  *               R2      =       DATA WORD OF POSITION TABLE       *
                       18  *                                                                  *
                       19  * OUTPUTS:      R4      =       0 OR 1                            *
                       20  *                                                                  *
                       21  * MODULES REFERENCED: NONE                                        *
                       22  *                                                                  *
                       23  * WORKSPACE AREA: CALLER'S                                        *
                       24  *                                                                  *
                       25  * REGISTERS MODIFIED: R4                                          *
                       26  *                                                                  *
                       27  * VERSION HISTORY:                                                *
                       28  *                                                                  *
                       29  ********************************************************************
          =0000        30          RSECT   DRSSMD
                       31  *** CALL NAME
                       32          INTERN  DRSSMD
                       33  *** VARIABLES REFERENCED
                       34  *** CONSTANTS REFERENCED
                       35          EXTERN  B10
                       36  *** TABLES REFERENCED
                       37  *** MODULES REFERENCED
                       38  *** LIBRARY
                       39  *** REGISTERS DEFINITION
          =0002        40  DRP     EQU     R2      DATA WORD OF POSITION TABLE (INPUT)
          =0004        41  DLS     EQU     R4      LS 16 BIT       OF DATA (INPUT); DATA (OUTPUT)
                       42  ********************************************************************
                       43  *
                       44  *** SSM ERROR DISCRETE
0000'                  45  DRSSMD
0000'  0284 0001       46          CI      DLS,1           01 CODE
0004'  1307            47          JEQ     50$             EXIT IF FAILURE (01 CODE)
0006'  1B05            48          JH      20$             JIF NOT FAILURE COODE (10 OR 11)
                       49  *       COME HERE FOR 00 CODE, CHECK BINARY OR BCD TYPE
0008'  20A0 0001*      50          COC     @B10,DRP
000C'  1303            51          JEQ     50$             EXIT IF BCD (00 IS NOT FAILURE FOR BCD)
                       52  *       COME HERE FOR 00 FAILURE CODE OF BINARY TYPE
```

```
                1          IDT     DSTINW
                2          SUBTTL  DATA INTO DESTINATION
                3  ********************************************************************
                4  *                                                                  *
                5  * NAME: DSTINW.SRC                      AUTH: N.COSTANTINIDES      *
                6  * VERSION: 2                            DATE: 27-MAY-1982          *
                7  *                                                                  *
                8  * FUNCTION: ACCORDING TO THE PARAM-OFFSET-PTR GETS THE ADDRESS     *
                9  *           THE DESTINATION INFORMATION, CONTAINING DATA SETUP,     *
               10  *           DEST BUFF OFFSET AND DEST-TYPE, IN PARAM-DS-TABLE,      *
               11  *           DOES THE NECESSARY SET-UP AND PUTS THE DATA INTO        *
               12  *           DESTINATION.                                           *
               13  *           2 BYTE DATA SET IN DESTINATION OFFSET TABLE             *
               14  *                                                                  *
               15  * CALLING MODULES: DR1ISR,DR2ISR,ANACQ,DC1ACQ,DC2ACQ               *
               16  *                                                                  *
               17  * CALLING SEQ: CALL @DSTINW                                        *
               18  *                                                                  *
               19  * INPUTS: R2    =       PARAM DEST OFFSET ARRAY TABLE START ADDR*
               20  *         R3    =       PARAM DEST TABLE                         *
               21  *         R4    =       INPUT DATA R,J.                          *
               22  *         R10   =       OFFSET OF TABLE                          *
               23  *                                                                  *
               24  * OUTPUTS: BITS DATA IN DESTINATION BUFFER VIA R3 POINTER          *
               25  *                                                                  *
               26  * MODULES REFERENCED: NONE                                         *
               27  *                                                                  *
               28  * WORKSPACE AREA: CALLER'S                                         *
               29  *                                                                  *
               30  * REGISTERS MODIFIED: R0,R1,R3,R4,R5                               *
               31  *                                                                  *
               32  * VERSION HISTORY:                                                 *
               33  *                                                                  *
               34  ********************************************************************
=0000          35         RSECT   DSTINW
               36  *** CALL NAME
               37         INTERN  DSTINW
               38  *** VARIABLES REFERENCED
               39         EXTERN  OUTBB
               40  *** CONSTANTS REFERENCED
               41         EXTERN  R15,B6
               42  *** TABLES REFERENCED
               43  *** MODULES REFERENCED
               44  *** LIBRARY
               45         INCLUDE ENCLOS
               47  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
               49         INCLUDE REGDEF          REGISTER DEFENITIONS
               68         INCLUDE CNSTNT          CONSTANTS
              196         INCLUDE SUBMAC          FUNCTIONAL MACROS
              497         INCLUDE MSCMAC          MISCELLANEOUS MACROS
              727         INCLUDE JMPMAC          JUMP MACROS
              761         INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
```

```
                    776          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1152  *** REGISTERS DEFINITION
          =0000    1153  TMP     EQU     R0      =       SCRATCH
          =0001    1154  DTY     EQU     R1      =       SCRATCH
          =0002    1155  PTD     EQU     R2      =       PARAM-DS-OFFSET-ARRAY (INPUT)
          =0003    1156  DST     EQU     R3      =       PARAM-DS-TABLE (INPUT)
          =0004    1157  PAR     EQU     R4      =       PARAM (INPUT)
                   1158  *       EQU     R5      =       SCRATCH
          =000A    1159  OFP     EQU     R10     =       PARAM-OFFSET-PTR (INPUT)
```

```
                    1161  ********************************************************************************
0000'               1162  DSTINW
0000'  A08A         1163          A       OFF,PTD         DS-OFFSET=DS-OFFSET-ARRAY (INDEXED BY
                    1164  *                               OFFSET-PTR)
0002'  C092         1165          MOV     *PTD,PTD
                    1166+         INDEX   PTD,            PTD=OFFSET*2
0004'  0A12         1172A           SLA   PTD,1
0006'  A083         1175          A       DST,PTD         DS-INFO-PTR=OFFSET + DS-TABLE-ADDR
                    1176  ***     GET OUTPUT WORD ADDRESS FROM PARAM-DEST-TABLE
0008'  C144         1177  10$     MOV     R4,R5           WORK WITH BITS DATA IN R5
000A'  C0D2         1178          MOV     *PTD,R3         DATA SET IN PAR-DEST-TABLE
000C'  0973         1179          SRL     R3,7            R.J. BUFFER TYPE * 2
000E'  0243 0006    1180          ANDI    R3,>6           SAVE 2 BITS
0012'  C063 0001*   1181          MOV     @OUTBR(R3),R1   DEST OUTPUT BUFFER START ADDRESS
0016'  C0D2         1182          MOV     *PTD,R3         DATA SET IN PAR-DEST-TABLE
0018'  0243 00FF    1183          ANDI    R3,>FF          SAVE BUFFER OFFSET WORD
001C'  0A13         1184          SLA     R3,1            CHANGE TO BYTE ADDRESS OFFSET
001E'  A0C1         1185          A       R1,R3           ADD OFFSET TO START ADDRESS=OUTPUT POINTER
                    1186  ***     POSITION INPUT DATA TO OUTPUT BIT LOCATIONS
0020'  C012         1187          MOV     *PTD,R0         DATA SET IN PARAM-DEST-TABLE
0022'  09A0         1188          SRL     R0,10           R.J OUT-POS-LEFT-SHIFT COUNT
0024'  0240 000F    1189          ANDI    R0,>F           SAVE 4 BITS, [illegible] USED IN CLEARING OUTPUT
0028'  1301         1190          JEQ     40$             JIF NO ADJUSTMENTS NEEDED
002A'  0A05         1191          SLA     R5,0            POSITION INPUT DATA VIA R0
                    1192  ***     SET INPUT TO DESTINATION
002C'  C012         1193  40$     MOV     *PTD,R0         DATA SET IN PAR DEST TABLE
002E'  0240 4000    1194          ANDI    R0,>4000        EXTRACT BIT 7
0032'  1301         1195          JEQ     41$             JIF NO CLEAR
0034'  04D3         1196          CLR     *R3
0036'  E4C5         1197  41$     SOC     R5,*R3          MOV TO DESTINATION
0038'  C30C         1198          MOV     R12,R12
003A'  1302         1199          JEQ     42$
                    1200  ***     SET SIGN BIT FOR PARITY/SSM ERROR
003C'  E4E0 0002*   1201          SOC     @B15,*R3
                    1202  ***     END OF DATA SET IN PAR-DEST-TABLE?
0040'  C032         1203  42$     MOV     *PTD+,R0        BUMP PAR-DEST-TABLE POINTER
0042'  15E2         1204          JGT     10$             JIF MORE DESTINATION DATA SET
0044'  13E1         1205          JEQ     10$             JUST IN CASE
                    1206  *** EXIT
0046'  045B         1207          RT
                    1208  ********************************************************************************
                    1209          END
```

No errors detected

```
                        1          IDT     DR1TBL
                        2          SUBTTL  DITS RECEIVER TABLES
          =0000         3          RSECT   DR1TBL
                        4  ;*****************************************************************
                        5  ;                                                                *
                        6  ;  NAME: DR1TBL.SRC                    AUTH: N. CONSTANTINIDES  *
                        7  ;  VERSION: 1                          DATE: 25-MAY-1983        *
                        8  ;                                                                *
                        9  ;  FUNCTION: CONTAINS PARAMETER ARRAY OFFSET, DITS PARAMETER QTY IN  *
                       10  ;            EACH CYCLE ARRAY OFFSET, PARAMETER MASK TABLE, PARAMETER  *
                       11  ;            NUMBER, RECEIVER DATA POSITION TABLE, DESTINATION TABLE   *
                       12  ;            OFFSET, PARAMETER DESTINATION INFORMATION.                *
                       13  ;                                                                *
                       14  ;  REFERENCED BY:                                               *
                       15  ;                                                                *
                       16  ;  DATA POSITION TABLE (DR1PST) FORMAT:                         *
                       17  ;  MACRO FORMAT: DRPT N1,N2,N3,N4                               *
                       18  ;                                                                *
                       19  ;  BIT      0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  *
                       20  ;           N1 N1 N1 N1 N1 N2 N2 N2 N2 N3 N3  -   -   -   -   N4  *
                       21  ;                                                                *
                       22  ;  WHERE:       N1 = INPUT POSITION                              *
                       23  ;               N2 = INPUT MASK WORD INDEX                       *
                       24  ;               N3 = SIGN (OPTIONAL)                             *
                       25  ;               N4 = ROUNDING (OPTIONAL)                         *
                       26  ;                                                                *
                       27  ;  PARAMETER DESTINATION TABLE (DR1DT) FORMAT:                   *
                       28  ;  MACRO FORMAT: GENDT N1,N2,N3,N4,N5                           *
                       29  ;                                                                *
                       30  ;  BIT      0  1  3  4  5  6  7  8  9  10  11  12  13  14  15    *
                       31  ;           N1 N1 N1 N1 N1 N1 N1 N2 N2 N3  N3  N3  N3  N4  N5   *
                       32  ;                                                                *
                       33  ;  WHERE:       N1 = BUFFER OFFSET WORD                          *
                       34  ;               N2 = BUFFER TYPE                                 *
                       35  ;               N3 = OUTPUT POSITION LEFT SHIFT COUNT            *
                       36  ;               N4 = CLEAR OUTPUT WORD (OPTIONAL)                *
                       37  ;               N5 = END OF DATA SET (OPTIONAL)                  *
                       38  ;                                                                *
                       39  ;  VERSION HISTORY:                                             *
                       40  ;                                                                *
                       41  ;                                                                *
                       42  ;*****************************************************************
                       43  ;
                     1179  ;
                     1180  ;** TABLES FOR DITS #1
                     1181  ;
                     1182          INTERN  DR1POP,DR1BOP,DR1PQG,DR1BQG
                     1183          INTERN  DR1I4P,DR1IEP,DR1I4G,DR1IEG
                     1184          INTERN  DR1DAP,DR1DBP,DR1DAG,DR1DBG
                     1185          INTERN  DR1DAP,DR1DBP,DR1DAG,DR1DBG
                     1186          INTERN  DR1MT,DR1DT,DR1PNT,DR1PST
                     1187          INTERN  DR1A1P,DR1A1G,DR1A2P,DR1A2G
```

```
                1188          INTERN  DR1UPT
                1189
```

```
                    1191  ********************************************************************************
                    1192  *
                    1193  *                    DITS #1
                    1194  *
                    1195  *                    *******
0000' 0080'         1196  DR1A1P  DATA    DR10AP          4 S/F DR1 OFFSET TBL (P & W)
0002' 0316'         1197  DR1A1G  DATA    DR10AG          "  "   "    "     "  (GE)
                    1198  *
0004' 00DA'         1199  DR1A2P  DATA    DR10BP          EVERY "    "      "  (P & W)
0006' 0372'         1200  DR1A2G  DATA    DR10BG            "   "    "      "  (GE)
                    1201  *
                    1202  *
                    1203  *              **************************
                    1204  ***************  P & W ENGINE SECTION ***********
                    1205  *              **************************
                    1206  *
                    1207  *       DITS #1 PARAMETER QTY IN EACH CYCLE
                    1208  *
                    1209  *
0008'               1210  DR1P0P
                    1211  *
                    1212
                    1213  *       SUBFRAME  1  CYCLES 1 TO 8
                    1214
0008' 01            1215          BYTE    1       DUMMY
0009' 01            1216          BYTE    1       DUMMY
000A' 02            1217          BYTE    2
000B' 01            1218          BYTE    1       DUMMY
000C' 02            1219          BYTE    2
000D' 01            1220          BYTE    1
000E' 01            1221          BYTE    1       DUMMY
000F' 01            1222          BYTE    1       DUMMY
                    1223
                    1224  *       SUBFRAME  2  CYCLES 1 TO 8
                    1225
0010' 01            1226          BYTE    1       DUMMY
0011' 01            1227          BYTE    1       DUMMY
0012' 01            1228          BYTE    1
0013' 01            1229          BYTE    1       DUMMY
0014' 01            1230          BYTE    1
0015' 01            1231          BYTE    1
0016' 01            1232          BYTE    1       DUMMY
0017' 01            1233          BYTE    1       DUMMY
                    1234
                    1235  *       SUBFRAME  3  CYCLES 1 TO 8
                    1236
0018' 01            1237          BYTE    1       DUMMY
0019' 01            1238          BYTE    1       DUMMY
001A' 01            1239          BYTE    1       DUMMY
001B' 01            1240          BYTE    1       DUMMY
001C' 01            1241          BYTE    1
001D' 01            1242          BYTE    1
```

```
001E'  0C          1243          BYTE    12
001F'  01          1244          BYTE    1       DUMMY
                   1245
                   1246  *       SUBFRAME  4  CYCLES 1 TO 8
                   1247
0020'  01          1248          BYTE    1       DUMMY
0021'  01          1249          BYTE    1       DUMMY
0022'  01          1250          BYTE    1       DUMMY
0023'  01          1251          BYTE    1       DUMMY
0024'  01          1252          BYTE    1       DUMMY
0025'  01          1253          BYTE    1
0026'  01          1254          BYTE    1
0027'  01          1255          BYTE    1       DUMMY
                   1256
                   1257  *       ALL SUBFRAMES  CYCLES 1 TO 8
                   1258
0028'              1259  DR1B0P
                   1260
0028'  1B          1261          BYTE    27
0029'  1A          1262          BYTE    26
002A'  1B          1263          BYTE    27
002B'  1F          1264          BYTE    31
002C'  18          1265          BYTE    24
002D'  1D          1266          BYTE    29
002E'  1D          1267          BYTE    29
002F'  21          1268          BYTE    33
                   1269
                   1270          EVEN
                   1271  ********************************************************
                   1272  *
                   1273  *  BITS #1 CYCLE INDEX TABLE FOR 4 S/F OFFSET TBL
                   1274  *  (USED TO GET OFFSET VALUE FROM DR1OFA TBL)
                   1275  *
                   1276  *
0030'              1277  DR1I4P
                   1278  *
                   1279  *       S/F 1 CYCLES 1 TO 8
                   1280  *
0030'  FFFE        1281          DATA    (-1)*2
0032'  0000        1282          DATA    (-1+1)*2
0034'  0002        1283          DATA    (-1+1+1)*2
0036'  0006        1284          DATA    (-1+1+1+2)*2
0038'  0008        1285          DATA    (-1+1+1+2+1)*2
003A'  000C        1286          DATA    (-1+1+1+2+1+2)*2
003C'  000E        1287          DATA    (-1+1+1+2+1+2+1)*2
003E'  0010        1288          DATA    (-1+1+1+2+1+2+1+1)*2
                   1289  *
                   1290  *       S/F 2 CYCLES 1 TO 8
                   1291  *
0040'  0012        1292          DATA    (9)*2
0042'  0014        1293          DATA    (9+1)*2
0044'  0016        1294          DATA    (9+1+1)*2
```

```
0048'  001B          1295          DATA    (9+1+1+1)*2
0049'  001A          1296          DATA    (9+1+1+1+1)*2
004A'  001C          1297          DATA    (9+1+1+1+1+1)*2
004C'  001E          1298          DATA    (9+1+1+1+1+1+1)*2
004E'  0020          1299          DATA    (9+1+1+1+1+1+1+1)*2
                     1300
                     1301  *       S/F 3 CYCLES 1 TO 8
                     1302  *
                     1303  *
0050'  0022          1304          DATA    (17)*2
0052'  0024          1305          DATA    (17+1)*2
0054'  0026          1306          DATA    (17+1+1)*2
0056'  0028          1307          DATA    (17+1+1+1)*2
0058'  002A          1308          DATA    (17+1+1+1+1)*2
005A'  002C          1309          DATA    (17+1+1+1+1+1)*2
005C'  002E          1310          DATA    (17+1+1+1+1+1+1)*2
005E'  0046          1311          DATA    (17+1+1+1+1+1+1+12)*2
                     1312  *
                     1313  *       S/F 4 CYCLES 1 TO 8
                     1314  *
0060'  0048          1315          DATA    (36)*2
0062'  004A          1316          DATA    (36+1)*2
0064'  004C          1317          DATA    (36+1+1)*2
0066'  004E          1318          DATA    (36+1+1+1)*2
0068'  0050          1319          DATA    (36+1+1+1+1)*2
006A'  0052          1320          DATA    (36+1+1+1+1+1)*2
006C'  0054          1321          DATA    (36+1+1+1+1+1+1)*2
006E'  0056          1322          DATA    (36+1+1+1+1+1+1+1)*2
                     1323  *
                     1324  *  DITS #1 CYCLE INDEX TBL FOR EVERY S/F OFFSET TBL
                     1325  *  (USED TO GET OFFSET VALUE FROM DR1OFB TBL)
                     1326  *
0070'                1327  DR1IEP
                     1328  *
0070'  FFFE          1329          DATA    (-1)*2
0072'  0034          1330          DATA    (-1+27)*2
0074'  0068          1331          DATA    (-1+27+26)*2
0076'  009E          1332          DATA    (-1+27+26+27)*2
0078'  00DC          1333          DATA    (-1+27+26+27+31)*2
007A'  010C          1334          DATA    (-1+27+26+27+31+24)*2
007C'  0146          1335          DATA    (-1+27+26+27+31+24+29)*2
007E'  0180          1336          DATA    (-1+27+26+27+31+24+29+29)*2
                     1337  *
                     1338  *
                     1339  ************************************************************************
                     1340  *
                     1341  *       DITS #1 PARAMETER-ARRAY-OFFSET
                     1342  *
                     1343  *
0080'                1344  DR1DAP
                     1345  *
                     1346  *       SUBFRAME 1 CYCLES 1 TO 8
```

```
                        1347  *
     0050'  00FA        1348        DATA    250             DUMMY
     0052'  00FA        1349        DATA    250             DUMMY
     0054'  00F7 00F8   1350        DATA    247,248
     0058'  00FA        1351        DATA    250             DUMMY
     005A'  000F 0038   1352        DATA    15,56
     005E'  00B6        1353        DATA    182
     0060'  00FA        1354        DATA    250             DUMMY
     0062'  00FA        1355        DATA    250             DUMMY
                        1356  *
                        1357  *     SUBFRAME 2 CYCLES 1 TO 8
                        1358  *
     0064'  00FA        1359        DATA    250             DUMMY
     0066'  00FA        1360        DATA    250             DUMMY
     0068'  00F9        1361        DATA    249
     006A'  00FA        1362        DATA    250             DUMMY
     006C'  0010        1363        DATA    16
     006E'  00B5        1364        DATA    181
     0070'  00FA        1365        DATA    250             DUMMY
     0072'  00FA        1366        DATA    250             DUMMY
                        1367  *
                        1368  *     SUBFRAME 3 CYCLES 1 TO 8
                        1369  *
     0074'  00FA        1370        DATA    250             DUMMY
     0076'  00FA        1371        DATA    250             DUMMY
     0078'  00FA        1372        DATA    250             DUMMY
     007A'  00FA        1373        DATA    250             DUMMY
     007C'  0038        1374        DATA    56
     007E'  00B6        1375        DATA    182
     0080'  0056 0057   1376        DATA    86,87,88,89,90,91,92,93,94,95,96,27
     0084'  0058 0059
     0088'  005A 005B
     008C'  005C 005D
     0090'  005E 005F
     0094'  0060 001B
     0098'  00FA        1377        DATA    250             DUMMY
                        1378  *
                        1379  *     SUBFRAME 4 CYCLES 1 TO 8
                        1380  *
     009A'  00FA        1381        DATA    250             DUMMY
     009C'  00FA        1382        DATA    250             DUMMY
     009E'  00FA        1383        DATA    250             DUMMY
     00A0'  00FA        1384        DATA    250             DUMMY
     00A2'  00FA        1385        DATA    250             DUMMY
     00A4'  00B5        1386        DATA    181
     00A6'  0012        1387        DATA    18
     00A8'  00FA        1388        DATA    250             DUMMY
                        1389  *
     00AA'              1390  DR1OBP
                        1391  *
                        1392  *     ALL SUBFRAMES CYCLES 1 TO 8
                        1393  *
```

```
00[illegible]  0031 00A9   1394  OPCYC1  DATA   49,169,174,1,2,73,74,75,76,77,78,79,80
00[illegible]  00AE 0001
00[illegible]  0002 0049
00[illegible]  004A 004B
00[illegible]  004C 004D
00[illegible]  004E 004F
00[illegible]  0050
00[illegible]  0051 0052   1395          DATA   81,82,83,84,85,86,87,88,89,90,91,244,251,252
00[illegible]  0053 0054
00[illegible]  0055 0056
01[illegible]  0057 0058
01[illegible]  0059 005A
01[illegible]  005B 00F4
01[illegible]  00FB 00FC
01[illegible]  0011 0013   1396  OPCYC2  DATA   17,19,26,47,172,24,92,93,94,95,96,97
01[illegible]  001A 002F
01[illegible]  00AC 0018
01[illegible]  005C 005D
01[illegible]  005E 005F
01[illegible]  0060 0061
01[illegible]  0062 0063   1397          DATA   98,99,100,101,102,103,104,105,106,107,108,109,110,245
01[illegible]  0064 0065
01[illegible]  0066 0067
01[illegible]  0068 0069
01[illegible]  006A 006B
01[illegible]  006C 006D
01[illegible]  006E 00F5
01[illegible]  0014 001C   1398  OPCYC3  DATA   20,28,29,48,169,170,25,111,112,113,114
0148'  001D 0030
01[illegible]  00A9 00AA
01[illegible]  0019 006F
0154'  0070 0071
0158'  0072
01[illegible]  0073 0074   1399          DATA   115,116,117,118,119,120,121,122,123,124,125,126,127
01[illegible]  0075 0076
01[illegible]  0077 0078
01[illegible]  0079 007A
01[illegible]  007B 007C
01[illegible]  007D 007E
0172'  007F
01[illegible]  0080 0081   1400          DATA   128,129,246
0178'  00F6
01[illegible]  0015 009F   1401  OPCYC4  DATA   21,159,160,161,162,163,164,165,166,167,168,130,131
01[illegible]  00A0 00A1
01[illegible]  00A2 00A3
01[illegible]  00A4 00A5
01[illegible]  00A6 00A7
01[illegible]  00A9 0082
01[illegible]  0083
01[illegible]  0084 0085   1402          DATA   132,133,134,135,136,137,138,139,140,141,142,143,144,145
01[illegible]  0086 0087
01[illegible]  0088 0089
```

```
01A0'  008A 008B
01A4'  008C 008D
01A8'  008E 008F
01AC'  0090 0091
01B0'  0092 0093    1403            DATA    146,147,148,247
01B4'  0094 00F7
01B8'  0016 001E    1404    OPCYC5  DATA    22,30,31,169,149,150,151,152,153
01BC'  001F 00A9
01C0'  0095 0096
01C4'  0097 0098
01C8'  0099
01CA'  009A 009B    1405            DATA    154,155,156,157,158,175,176,177,178,181,182,184,185,186,248
01CE'  009C 009D
01D2'  009E 00AF
01D6'  00B0 00B1
01DA'  00B2 00B5
01DE'  00B6 00B8
01E2'  00B9 00BA
01E6'  00F8
01E8'  00B3 00B4    1406    OPCYC6  DATA    179,180,183,46,50,51,52,53,54,187,188,189,190,191
01EC'  00B7 002E
01F0'  0032 0033
01F4'  0034 0035
01F8'  0036 00BB
01FC'  00BC 00BD
0200'  00BE 00BF
0204'  00C0 00C1    1407            DATA    192,193,194,195,196,197,198,199,200,201,202,203,204,205,249
0208'  00C2 00C3
020C'  00C4 00C5
0210'  00C6 00C7
0214'  00CB 00C9
0218'  00CA 00CB
021C'  00CC 00CD
0220'  00F9
0222'  00A9 00AA    1408    OPCYC7  DATA    169,170,171,55,56,57,58,59,60,61,206,207,208,209,210
0226'  00AB 0037
0228'  0038 0039
022C'  003A 003B
0230'  003C 003D
0234'  00CE 00CF
0238'  00D0 00D1
023C'  00D2
0240'  00D3 00D4    1409            DATA    211,212,213,214,215,216,217,218,219,220,221,222,223,224
0244'  00D5 00D6
0248'  00D7 00D8
024C'  00D9 00DA
0250'  00DB 00DC
0254'  00DD 00DE
0258'  00DF 00E0
025C'  00AD 003E    1410    OPCYC8  DATA    173,62,63,64,65,66,67,68,69,70,71,72,225,226,227,228,229
0260'  003F 0040
0264'  0041 0042
```

```
02E8'  0043 0044
02EC'  0045 0046
02F0'  0047 0048
02F4'  00E1 00E2
02F8'  00E3 00E4
02FC'  00E5
0300'  00E6 00E7    1411          DATA    230,231,232,233,234,235,236,237,238,239,240,241,242,243
0302'  00E8 00E9
0304'  00EA 00EB
0308'  00EC 00ED
030C'  00EE 00EF
0310'  00F0 00F1
0314'  00F2 00F3
0318'  0012 001B    1412          DATA    18,27
                    1413  *
                    1414  ********************************************************************************
                    1415  *
                    1416  *                ******************************
                    1417  ****************      GE ENGINE SECTION      ****************
                    1418  *                ******************************
                    1419  *
                    1420  *       PARAMETER QUANTITY TABLES
                    1421  *
031E'               1422  DR1PQG
                    1423
                    1424  *       SUBFRAME  1  CYCLES 1 TO 8
                    1425
0322'  01          1426          BYTE    1       DUMMY
0323'  01          1427          BYTE    1       DUMMY
0324'  02          1428          BYTE    2
0325'  01          1429          BYTE    1       DUMMY
0326'  02          1430          BYTE    2
0327'  01          1431          BYTE    1
0328'  01          1432          BYTE    1       DUMMY
0329'  01          1433          BYTE    1       DUMMY
                    1434
                    1435  *       SUBFRAME  2  CYCLES 1 TO 8
                    1436
032A'  01          1437          BYTE    1       DUMMY
032B'  01          1438          BYTE    1       DUMMY
032C'  01          1439          BYTE    1
032D'  01          1440          BYTE    1       DUMMY
032E'  01          1441          BYTE    1
032F'  01          1442          BYTE    1
0330'  01          1443          BYTE    1       DUMMY
0331'  01          1444          BYTE    1       DUMMY
                    1445
                    1446  *       SUBFRAME  3  CYCLES 1 TO 8
                    1447
0332'  01          1448          BYTE    1       DUMMY
0333'  01          1449          BYTE    1       DUMMY
0334'  01          1450          BYTE    1       DUMMY
```

```
02B1' 01          1451          BYTE    1       DUMMY
02B2' 02          1452          BYTE    2
02B3' 01          1453          BYTE    1
02B4' 0C          1454          BYTE    12
02B5' 01          1455          BYTE    1       DUMMY
                  1456
                  1457  *       SUBFRAME  4  CYCLES 1 TO 8
                  1458
02B6' 01          1459          BYTE    1       DUMMY
02B7' 01          1460          BYTE    1       DUMMY
02B8' 01          1461          BYTE    1       DUMMY
02B9' 01          1462          BYTE    1       DUMMY
02BA' 01          1463          BYTE    1
02BB' 01          1464          BYTE    1
02BC' 01          1465          BYTE    1
02BD' 01          1466          BYTE    1       DUMMY
                  1467
                  1468  *       ALL SUBFRAMES CYCLES 1 TO 8
                  1469  *
02BE'             1470  DR1BQG
                  1471
02BE' 20          1472          BYTE    32
02BF' 1D          1473          BYTE    29
02C0' 1D          1474          BYTE    29
02C1' 1F          1475          BYTE    31
02C2' 1D          1476          BYTE    29
02C3' 20          1477          BYTE    32
02C4' 1E          1478          BYTE    30
02C5' 20          1479          BYTE    32
                  1480
                  1481          EVEN
                  1482  ********************************************************
                  1483  *
                  1484  *  DITS #1 CYCLE INDEX TABLE FOR 4 S/F OFFSET TBL
                  1485  *   (USED TO GET OFFSET VALUE FROM DR1OFA TBL)
                  1486  *
                  1487  *
02DA'             1488  DR1I4G
                  1489  *
                  1490  *       S/F 1 CYCLES 1 TO 8
                  1491  *
02DA' FFFE        1492          DATA    (-1)*2
02DB' 0000        1493          DATA    (-1+1)*2
02DC' 0002        1494          DATA    (-1+1+1)*2
02DD' 0006        1495          DATA    (-1+1+1+2)*2
02DE' 000B        1496          DATA    (-1+1+1+2+1)*2
02E0' 000C        1497          DATA    (-1+1+1+2+1+2)*2
02E2' 000E        1498          DATA    (-1+1+1+2+1+2+1)*2
02E4' 0010        1499          DATA    (-1+1+1+2+1+2+1+1)*2
                  1500  *
                  1501  *       S/F 2 CYCLES 1 TO 8
                  1502  *
```

```
02B6'  0012          1503          DATA    (9)*2
02B8'  0014          1504          DATA    (9+1)*2
02BA'  0016          1505          DATA    (9+1+1)*2
02BC'  0018          1506          DATA    (9+1+1+1)*2
02BE'  001A          1507          DATA    (9+1+1+1+1)*2
02C0'  001C          1508          DATA    (9+1+1+1+1+1)*2
02C2'  001E          1509          DATA    (9+1+1+1+1+1+1)*2
02C4'  0020          1510          DATA    (9+1+1+1+1+1+1+1)*2
                     1511
                     1512  *       S/F 3 CYCLES 1 TO 8
                     1513  *
                     1514  *
02C6'  0022          1515          DATA    (17)*2
02C8'  0024          1516          DATA    (17+1)*2
02CA'  0026          1517          DATA    (17+1+1)*2
02CC'  0028          1518          DATA    (17+1+1+1)*2
02CE'  002A          1519          DATA    (17+1+1+1+1)*2
02D0'  002E          1520          DATA    (17+1+1+1+1+2)*2
02D2'  0030          1521          DATA    (17+1+1+1+1+2+1)*2
02D4'  0048          1522          DATA    (17+1+1+1+1+2+1+12)*2
                     1523  *
                     1524  *       S/F 4 CYCLES 1 TO 8
                     1525  *
02D6'  004A          1526          DATA    (37)*2
02D8'  004C          1527          DATA    (37+1)*2
02DA'  004E          1528          DATA    (37+1+1)*2
02DC'  0050          1529          DATA    (37+1+1+1)*2
02DE'  0052          1530          DATA    (37+1+1+1+1)*2
02E0'  0054          1531          DATA    (37+1+1+1+1+1)*2
02E2'  0056          1532          DATA    (37+1+1+1+1+1+1)*2
02E4'  0058          1533          DATA    (37+1+1+1+1+1+1+1)*2
                     1534  *
                     1535  *  DITS #1 CYCLE INDEX TBL FOR EVERY S/F OFFSET TBL
                     1536  *  (USED TO GET OFFSET VALUE FROM DR1OFB TBL)
                     1537  *
02E6'                1538  DR11E6
                     1539  *
02E6'  FFFE          1540          DATA    (-1)*2
02E8'  003E          1541          DATA    (-1+32)*2
02EA'  0078          1542          DATA    (-1+32+29)*2
02EC'  00B2          1543          DATA    (-1+32+29+29)*2
02EE'  00F0          1544          DATA    (-1+32+29+29+31)*2
02F0'  012A          1545          DATA    (-1+32+29+29+31+29)*2
02F2'  016A          1546          DATA    (-1+32+29+29+31+29+32)*2
02F4'  01A6          1547          DATA    (-1+32+29+29+31+29+32+30)*2
                     1548  *
                     1549  *
                     1550  *       PARAMETER OFFSET TABLES
                     1551  *
02F6'                1552  DR10A6
                     1553  *
                     1554  *       SUBFRAME 1 CYCLES 1 TO 8
```

DRSSMD.SRC

```
000E'  0584          53          INC     DLS             SET TO LOGIC 1
0010'  1001          54          JMP     50$             EXIT
                     55  *       COME HERE FOR 10 OR 11 CODE, NO FAILURES
0012'  04C4          56  20$     CLR     DLS             SET TO LOGIC 0
                     57  *** EXIT
0014'  045B          58  50$     RT
                     59  ********************************************************************************
                     60          END
```

```
                     1          IDT     DRSIGN
                     2          SUBTTL  SETS SIGN BIT FOR DITS RECEIVED DATA
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: DRSIGN.SRC                    AUTH: N.COSTANTINIDES       *
                     6  * VERSION: 1                          DATE: 18-FEB-1982           *
                     7  *                                                                  *
                     8  * FUNCTION: ATTACHES A SIGN BIT TO THE DITS DATA                   *
                     9  *          RIGHT NEXT TO THE MSB POSITION.                         *
                    10  *                                                                  *
                    11  * CALLING MODULES: DRXTK                                          *
                    12  *                                                                  *
                    13  * CALLING SEQ: BL @DRSIGN                                         *
                    14  *                                                                  *
                    15  * INPUTS:       R2      =       DATA OF POS TBL                   *
                    16  *               R1      =       MSW OF DITS I/P DATA              *
                    17  *               R4      =       R.J STRIPPED I/P DATA             *
                    18  *               R3      =       I/P MASK WORD INDEX               *
                    19  *                                                                  *
                    20  * OUTPUTS: R4 = DITS DATA WTIH SIGN BIT.                          *
                    21  *                                                                  *
                    22  * MODULES REFERENCED: NONE                                        *
                    23  *                                                                  *
                    24  * WORKSPACE AREA: CALLER'S                                        *
                    25  *                                                                  *
                    26  * REGISTERS MODIFIED: R1,R4                                       *
                    27  *                                                                  *
                    28  * VERSION HISTORY:                                                *
                    29  *                                                                  *
                    30  ********************************************************************
       =0000        31         RSECT   DRSIGN
                    32  *** CALL NAME
                    33         INTERN  DRSIGN
                    34  *** VARIABLES REFERENCED
                    35  *** CONSTANTS REFERENCED
                    36         EXTERN  B10,B12
                    37  *** TABLES REFERENCED
                    38         EXTERN  DRSGTB
                    39  *** MODULES REFERENECED
                    40  *** LIBRARY
                    41  *** REGISTERS DEFINITION
       =0001        42  DDT    EQU     R1
       =0004        43  DLS    EQU     R4
                    44  ********************************************************************
    0000'           45  DRSIGN
    0000' 20A0 0001* 46         COC     @B10,R2
    0004' 1306       47         JEQ     20$                     JIF BCD
                     48  *
                     49  *      BINARY SIGN. MOVE SIGN FROM BIT 29
                     50  *
    0006' 2060 0002* 51         COC     @B12,DDT
    000A' 160B       52         JNE     40$                     JIF PLUS SIGN (B29 = 0)
```

```
000C'  E123 0003*     53          SOC     @DRSGTB(R3),DLS         SET MINUS SIGN BIT NEXT TO MSB
0010'  1008           54          JMP     40$
                      55  *       BCD SIGN. MOVE SIGN FROM BIT 30 & 31.
                      56  *       BIT 30 & 31 = 00 FOR PLUS
                      57  *       BIT 30 & 31 = 11 FOR MINUS
0012'  0241 6000      58  20$     ANDI    DDT,>6000
0016'  1305           59          JEQ     40$                     JIF PLUS SIGN
0018'  0281 6000      60          CI      DDT,>6000
001C'  1602           61          JNE     40$                     JIF NOT MINUS SIGN
001E'  E123 0003*     62          SOC     @DRSGTB(R3),DLS         SET MINUS SIGN BIT NEXT TO MSB
                      63  *
0022'  045B           64  40$     RT
                      65  ********************************************************************************
                      66          END
```

```
                    1          IDT     DRPSSM
                    2          SUBTTL  DITS PARITY AND SSM ERROR CHECKS
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: DRPSSM.SRC                    AUTH: N.COSTANTINIDES       *
                    6  * VERSION: 1                          DATE: 17-MAY-1983           *
                    7  *                                                                  *
                    8  * FUNCTION: CHECKS DITS INPUT DATA FOR PARITY AND SSM ERROR       *
                    9  *          IF REQUESTED IN THE POSITION TABLE.                     *
                   10  *                                                                  *
                   11  * CALLING MODULES: DRXTK.SRC                                      *
                   12  *                                                                  *
                   13  * CALLING SEQ: BL @DRPSSM                                         *
                   14  *                                                                  *
                   15  * INPUTS:       R2      =       DATA WORD OF POSITION TABLE       *
                   16  *               R3      =       MSW OF DITS INPUT DATA            *
                   17  *               R4      =       LSW
                   18  *               R12     =       0       (ERROR FLAG OK)           *
                   19  *                                                                  *
                   20  * OUTPUTS:      IF NO ERROR: R12=0                                *
                   21  *               IF PARITY ERROR: R3,R4,R12 = -1                   *
                   22  *               IF SSM ERROR   : R3,R4     = -1 AND R12 = +1      *
                   23  *                                                                  *
                   24  * MODULES REFERENCED: NONE                                        *
                   25  *                                                                  *
                   26  * WORKSPACE AREA: CALLER'S                                        *
                   27  *                                                                  *
                   28  * REGISTERS MODIFIED:R0,R3,R4,R12                                 *
                   29  *                                                                  *
                   30  * VERSION HISTORY:                                                *
                   31  *                                                                  *
                   32  ********************************************************************
=0000              33         RSECT   DRPSSM
                   34  *** CALL NAME
                   35         INTERN  DRPSSM
                   36  *** VARIABLES REFERENCED
                   37  *** CONSTANTS REFERENCED
                   38         EXTERN  B10,B15
                   39  *** TABLES REFERENCED
                   40  *** MODULES REFERENCED
                   41  *** LIBRARY
                   42  *** REGISTERS DEFINITION
=0002              43  DRP     EQU     R2      =       DATA WORD OF POSITION TABLE (INPUT)
=0003              44  DMS     EQU     R3      =       MS 16 BIT OF DATA (INPUT)
=0004              45  DLS     EQU     R4      =       LS 16 BIT OF DATA (INPUT);DATA (OUTPUT)
```

```
                    47  ********************************************************************
                    48  *
                    49  *** CHECK PARITY FIRST
                    50  *
0000'               51  DRPSSM
                    52  *
0000' 20E0 0002*    53          COC     @B15,DMS                MSW OF INPUT DATA
0004' 1604          54          JNE     10$                     JIF NO PARITY ERROR
                    55  *
                    56  *       PARITY ERROR, SET BOTH WORDS TO 1'S, R12 = -1
                    57  *
0006' 0703          58          SETO    DMS                     MSW OF INPUT DATA = -1
0008' 0704          59          SETO    DLS                     LSW OF  "     "   = -1
000A' 070C          60          SETO    R12                     SET TO PARITY ERROR
000C' 100E          61          JMP     40$                     EXIT
                    62  *** CHECK SSM ERROR
000E' C003          63  10$     MOV     DMS,R0          MSW OF I/P DATA
0010' 0240 6000     64          ANDI    R0,>6000        SAVE SSM FIELD ONLY
0014' 1604          65          JNE     20$             JIF NOT 00, CHECK FOR 01 CODE
0016' 20A0 0001*    66          COC     @B10,DRP
001A' 1307          67          JEQ     40$             EXIT IF BCD (00-NOT ERROR FOR BCD)
001C' 1003          68          JMP     30$             JMP TO ERROR IF BINARY TYPE
                    69  *
001E' 0280 2000     70  20$     CI      R0,>2000        01 CODE
0022' 1603          71          JNE     40$             JIF     NO ERROR (NOT 00 OR 01)
                    72  *
                    73  *       SSM ERRPR FOR BOTH BINARY OR BCD TYPE (SSM= 01)
                    74  *
0024' 0703          75  30$     SETO    DMS             MSW OF INPUT DATA = -1
0026' 0704          76          SETO    DLS             LSW
0028' 058C          77          INC     R12             SET SSM ERROR FLAG
                    78  *** EXIT
002A' 045B          79  40$     RT
                    80          END
```

```
                     1          IDT    DSRLS
                     2          SUBTTL  DOUBLE SHIFT RIGHT LOGICAL (SPECIAL CASE)
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: DSRLS                          AUTH: N.COSTANTINIDES     *
                     6  * VERSION: 2                           DATE: 12-NOV-1981         *
                     7  *                                                                  *
                     8  * FUNCTION: SHIFTS DOUBLE RIGHT ACCORDING TO THE GIVEN SHIFT      *
                     9  *          COUNT. IT RETURNS WITH 16-BIT OF DATA AS END           *
                    10  *          RESULT (INDEPENDENT OF SHIFT COUNT)                    *
                    11  *                                                                  *
                    12  * CALLING MODULES:                                                *
                    13  *                                                                  *
                    14  * CALLING SEQ: BL @DSRLS                                          *
                    15  *                                                                  *
                    16  * INPUTS: R0    =      INPUT DATA POSITION (0 THRU 31)          *
                    17  *         R3,R4 =      MSH,LSH OF INPUT DATA                    *
                    18  *                                                                  *
                    19  * OUTPUTS: R4   =      16 BIT DATA R.J.                         *
                    20  *                                                                  *
                    21  * MODULES REFERENCED: NONE                                        *
                    22  *                                                                  *
                    23  * WORKSPACE AREA: CALLER'S                                        *
                    24  *                                                                  *
                    25  * REGISTERS MODIFIED: R0,R3,R4                                    *
                    26  *                                                                  *
                    27  * VERSION HISTORY:                                                *
                    28  *                                                                  *
                    29  ********************************************************************
=0000               30         RSECT   DSRLS
                    31  *** CALL NAME
                    32         INTERN  DSRLS
                    33  *** VARIABLES REFERENCED
                    34  *** CONSTANTS REFERENCED
                    35  *** TABLES REFERENCED
                    36  *** MODULES REFERENCED
                    37  *** LIBRARY
                    38  *** REGISTERS DEFINITION
=0000               39  SHC     EQU     R0      =       DATA-POS START (0-THRU-31)
=0003               40  MSH     EQU     R3      =       MSH OF DATA
=0004               41  LSH     EQU     R4      =       LSH OF DATA
                    42  ********************************************************************
                    43         INCLUDE ENCLOS
                    45  ***    ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    47         INCLUDE REGDEF          REGISTER DEFENITIONS
                    66         INCLUDE CNSTNT          CONSTANTS
                   194         INCLUDE SUBMAC          FUNCTIONAL MACROS
                   495         INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   725         INCLUDE JMPMAC          JUMP MACROS
                   759         INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   774         INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
```

```
                    1151  *
0000'               1152  DSRLS
0000'  0240 001F    1153          ANDI    SHC,C31               SHIFT COUNT < 32.
0004'  1310         1154          JEQ     100$
                    1155+         DOIF    SHC,GE,C16,X,,        IF SHIFT-COUNT > 16,
0006'  0280 0010    1274B            CI   SHC,C16
000A'  1106         1331E             JLT  91$
000C'  C103         1461            MOV   MSH,LSH               LS-H = MS-H
000E'  0220 FFF0    1462            AI    SHC,CFFF0H            SHIFT-CNT = SHIFT-CNT - 16
0012'  1309         1463            JEQ   100$                  JIF SHIFT COUNT OF ZERO
0014'  0904         1464            SRL   LSH,ZERO              SHIFT DATA
                    1465+         ELSEDO
0016'  1007         1726E             JMP  92$
0018'               1856E  91$
0018'  0904         1941            SRL   LSH,ZERO              SHIFT LS-H
001A'  0220 FFF0    1942            AI    SHC,CFFF0H            SHIFT-CNT = 16 - SHIFT-CNT
001E'  0740         1943            ABS   SHC
                    1944+           DOIF  ,NE,,                 IF R0 NOT= 0,
0020'  1301         2119E             JEQ   93$
0022'  0A03         2250              SLA MSH,ZERO                SHIFT (LEFT) MS-H
                    2251+           ENDBLK
0024'               2391E  93$
0024'  E103         2471            SOC   MSH,LSH               PUT DATA TOGETHER
                    2472+         ENDBLK
0026'               2608E  92$
0026'  045B         2692  100$    RT
                    2693  ********************************************************************************
                    2694  * PRIVATE CONSTANT
       =FFF0        2695  CFFF0H  EQU     >FFF0
                    2696
                    2697          END
```

No errors detected

```
                     1          IDT     CALRT
                     2
                     3          SUBTTL  READ AND TEST RECEIVED CALIBRATION DATA
                     4
                     5  *       CALLING SEQ:    CALL    @CALRT
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       CALRT READS THE RECEIVED ADC CALIBRATION DATA (GND OR VOLTAGE)
                    10  *       AND TEST THEM FOR THEIR LIMITS.
                    11  *                                                                   !
                    12  *-------------------------------------------------------------------+
                    13  *       VERSION : 1
                    14  *       PROGRAMMED BY : N.CONSTANTINIDES
                    15  *       CHECKED BY : N.CONSTANTINIDES
                    16  *       MODIFIED : 03-JAN-84 (N.CONSTANTINIDES)
                    17
                    18  *
                    19  *
                    20          INTERN  CALRT
                    21  * REFERD MODULES:
                    22          EXTERN  ANREAD
                    23          EXTERN  CALMT
                    24          EXTERN  CAVLMT
                    25  * GLOBALS:
                    26          EXTERN  SYCALB          SYSTEM-CALIB-BUFFER (RAM)
                    27          EXTERN  CALSOT          SYSTEM-CALIB-BUFF-OFFSET TABLE (ROM)
                    28          EXTERN  SYNFLG          SYNCRO FLAG
                    29
=0000               30  RSV     EQU     R0      =       SCRATCH
=0001               31  CLF     EQU     R1      =       CAL-VALUE LIMIT FLAG (0: GND, 1: V-CAL) (INPUT)
=0002               32  TIP     EQU     R2      =       AN-TYPE ( OUTPUT)
=0003               33  VG1     EQU     R3      =       1ST V-CAL READING (OUTPUT)
=0004               34  VG2     EQU     R4      =       2ND V-CAL READING (OUTPUT)
=0005               35  OFST    EQU     R5      =       SCRATCH
=0007               36  CSN     EQU     R7      =       CAL-SLOT-# (INPUT; OUTPUT)
                    37
                    38          INCLUDE ENCLOS
                    40  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    42          INCLUDE REGDEF          REGISTER DEFENITIONS
                    61          INCLUDE CNSTNT          CONSTANTS
                   189          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   490          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   720          INCLUDE JMPMAC          JUMP MACROS
                   754          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   769          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000              1145          RSECT   CALRT
```

```
0014'               2153E  82$
                    2234   *** ANALOG CALIBRATION FAILED
0014'  D067 0040'   2235           MOVB    @ANCHNT(CSM),TPO
0018'  0981         2236           SRL     TPO,8
001A'  0221 0200    2237           AI      TPO,>200              ADD WORD NO AND INC ERROR N (201 TO 216)
001E'  06A0 0003$   2238           BL      @SYSER                SET ERROR BIT IN 2ND WORD OF SYST ERROR BUFFER
0022'  0720 0005$   2239           SETO    @SYNFLG
0026'  1009         2240           JMP     100$
                    2241+          ENDBLK                        IFEND
0028'               2377E  92$
                    2461   *** ANALOG CALIBRATION PASSED
0028'  D067 0040'   2462           MOVB    @ANCHNT(CSM),TPO
002C'  0981         2463           SRL     TPO,8
002E'  0221 0200    2464           AI      TPO,>200              ADD WORD NO & INC ERROR NO (201 TO 216)
0032'  06A0 0004$   2465           BL      @SYSOK                RESET ERROR BIT IN 2ND WORD OF SYSTEM ERROR BUFFER
0036'  04E0 0005$   2466           CLR     @SYNFLG               CLEAR FLAG
003A'  C2E0 0000'   2467   100$    MOV     @LINKZ,LINK
003E'  045B         2468           RT
                    2469   *** ANALOG CHANNEL NUMBERS FOR SYSTEM ERROR, IN ORDER OF DATA MUX TABLES
0040'  01 02 09     2470   ANCHNT  BYTE    1,2,9,10,11,13,19,20,22
0043'  0A 0B 0D
0046'  13 14 16
                    2471           EVEN
                    2472   ********************************************************************************
                    2473   *  PRIVATE RAM
                    2474+          LOCR    PRIV,LINKZ
0000'  =0002       2477A  LINKZ   BSS     2
                    2478   *
                    2479           END
```

```
                                1          IDT     CALMT
                                2
                                3          SUBTTL  CHECK CALIBRATION LIMITS
                                4
                                5  *       CALLING SEQ:    CALL    @CALMT
                                6
                                7  *-------------------------------------------------------------------+
                                8  *                                                                   !
                                9  *       CALMT TEST THE LIMITS ON A RECEIVED CALIBRATION (GND
                               10  *       OR VOLTAGE) DATA AND IF IT IS NOT IN THE GIVEN LIMIT,
                               11  *       SETS ERROR STATUS WORD.
                               12  *                                                                   !
                               13  *-------------------------------------------------------------------+
                               14  *       VERSION : 1
                               15  *       PROGRAMMED BY : N.CONSTANTINIDES
                               16  *       CHECKED BY : N.CONSTANTINIDES
                               17  *       MODIFIED : 03-JAN-84 (N.CONSTANTINIDES)
                               18
                               19          INTERN  CALMT
                               20  * GLOBALS:
                               21  *       (ROM)
                               22          EXTERN  CALLT           CAL-LOWER-LIMIT-TABLE
                               23          EXTERN  CALULT          CAL-UPPER-LIMIT-TABLE
                               24          EXTERN  SYSER,SYSOK,SYNFLG
                               25
        =0000                  26  CALV    EQU     R0      =       CAL VALUE (GND OR V)  (INPUT)
        =0001                  27  TPO     EQU     R1      =       LIMIT-TABLE-OFFSET (INPUT)
        =0007                  28  CSN     EQU     R7      =       CAL-ADDR-QUANT (INPUT)
                               29  *                       NOTE:   CAL-SLOT-# IS ALSO CHANNEL-# (+1)
                               30
                               31          INCLUDE ENCLOS
                               33  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                               35          INCLUDE REGDEF          REGISTER DEFENITIONS
                               54          INCLUDE CNSTNT          CONSTANTS
                              182          INCLUDE SUBMAC          FUNCTIONAL MACROS
                              483          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                              713          INCLUDE JMPMAC          JUMP MACROS
                              747          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                              762          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000                1138          RSECT   CALMT
                             1139  ********************************************************************
0000' C80B 0000'             1140  CALMT   MOV     LINK,@LINKZ
                             1141+          INDEX   TPO,                    TBL-OFFS = CAL-ADDR-OFFSET * 2
0004' 0A11                   1147A            SLA   TPO,1
                             1150+          DOIF    CALV,LEQ,@CALLT(TPO),,  IF DATA < LOWER LIMIT,
0006' 8840 0001*             1288B             C    CALV,@CALLT(TPO)
000A' 1501                   1329E              JGT  91$
                             1456+          ORIF    CALV,GE,@CALULT(TPO),,  OR DATA > UPPER LIMIT,
000C' 1003                   1717E              JMP  82$
000E'                        1847E  91$
000E' 8840 0002*             1961B             C    CALV,@CALULT(TPO)
0012' 110A                   2003E              JLT  92$
```

```
0000'                1152  CALKF
0000'  C80B 0000'    1153        MOV   LINK,@LINKZ     SAVE LINKER
                     1154+       INDEX VGP,                  OFFSET = TYPE * 2
0004'  0A12          1160A         SLA VGP,1
0006'  04C0          1163        CLR   KFCT            FACTOR = 0
0008'  60E6 0004*    1164        S     @VMOSBF(CAQ),VMCS     GET VMCS - VMOS
                     1165+       DOIF  ,NE,,,                IF (VMCS - VMOS) NOT= 0,
000C'  1308          1332E           JEQ  91$
000E'  C022 0006*    1471          MOV  @CAL1FT(VGP),KFCT        R0-R1 = (VICS*GIS) * (2 ** 14)
0012'  C040          1472          MOV  KFCT,MPL
0014'  0820          1473          SRA  KFCT,C2
0016'  0AE1          1474          SLA  MPL,C14
                     1475+         CALL SDIV05,<R5,=,VMCS>       KS = (VICS*GIS) / (VMCS - VMOS)
0018'  C143          1491C         MOV  VMCS,R5
001A'  06A0 0001*    1499A         BL   @SDIV05
                     1501+       ENDBLK                      ENDIF
001E'                1633E  91$
001E'  C980 0002*    1721        MOV   KFCT,@KSBUF(CAQ)      KS-BUFF (INEDXED BY CAQ) = KS
0022'  04C0          1722        CLR   KFCT                  FACTOR = 0
0024'  6126 0005*    1723        S     @VMORBF(CAQ),VMCR     GET VMCR - VMOR
                     1724+       DOIF  ,NE,,,                IF (VMCR - VMOR) NOT= 0,
0028'  1308          1895E           JEQ  92$
002A'  C022 0007*    2030          MOV  @CAL2FT(VGP),KFCT        R0-R1 = (VICR*GIR) * (2 ** 14)
002E'  C040          2031          MOV  KFCT,MPL
0030'  0920          2032          SRL  KFCT,C2
0032'  0AE1          2033          SLA  MPL,C14
                     2034+         CALL SDIV05,<R5,=,VMCR>       KR = (VICR*GIR) / (VMCR - VMOR)
0034'  C144          2050C         MOV  VMCR,R5
0036'  06A0 0001*    2058A         BL   @SDIV05
                     2060+       ENDBLK                      ENDIF
003A'                2196E  92$
003A'  C980 0003*    2280        MOV   KFCT,@KRBUF(CAQ)      KR-BUFF (INEDXED BY CAQ) = KR
003E'  C2E0 0000'    2281        MOV   @LINKZ,LINK     RESTORE LINKER
0042'  045B          2282        RT
                     2283  ********************************************************************************
                     2284  * PRIVATE AREA:
                     2285+       LOCR  PRIV,LINKZ,
0000'  =0002        2288A LINKZ  BSS   2
                     2289        END
```

```
01DD' 91    4937A                       BYTE   )91
            4940+       LLDC   GNDREF,3,19,
01DE' 0820  4974B                       DATA   )0820
01E0' 8442  5062A                       DATA   )84*256+ZZVL
01E2' 0004  5065A                       DATA   )0004
01E4' 00    5066A                       BYTE   )00
01E5' 91    5067A                       BYTE   )91
            5070+       SYNCRO VCAL,3,20,PHASEB,
01E6' 8808  5088A                       DATA   )8808
01E8' E480  5096A                       DATA   )E480
01EA' 2042  5131A                       DATA   )2042
01EC' 00    5139A                       BYTE   )00
01ED' 91    5140A                       BYTE   )91
            5141+       SYNCRO GNDREF,3,20,PHASEB,
01EE' 0820  5187B                       DATA   )0820
01F0' E442  5286A                       DATA   )E4*256+ZZVL
01F2' 2042  5298A                       DATA   )2042
01F4' 00    5306A                       BYTE   )00
01F5' 91    5307A                       BYTE   )91
            5308  *
01F6' 0010  5309  CALMAQ  DATA   $-CALMAT/B     CAL-MUX-ADDR-QUANT
            5310  *
            5311
            5312  ****************************
            5313
            5314  *
            5315  *       CALIBRATION CONSTANTS
            5316  *
            5317  *             VICS * GIS  (FOR 1ST READING)
01F8' 1000  5318  CAL1FT  DATA   4096          DC VOLTS ABS (LLDC),   0, SINGLE
01FA' 0C80  5319         DATA   3200          HI LVL DC ABS,         1, SINGLE
01FC' 1000  5320         DATA   4096          POTENTIOMETER,         2, SINGLE
01FE' 1000  5321         DATA   4096          5V POT Ref,            3, SINGLE
0200' 0DBC  5322         DATA   3516          TEMP. BOLBS,           4, SINGLE
0202' 0B50  5323         DATA   2896          AC RATIO # 1,          5, DOUBLE
0204' 1000  5324         DATA   4096          DC RATIO # 1,          6, DOUBLE
0206' 1000  5325         DATA   4096          DC RATIO # 2,          7, DOUBLE
0208' 17F9  5326         DATA   6137          SYNCHRO,               8, DOUBLE
020A' 0AE1  5327         DATA   2785          AC RATIO # 2,          9, DOUBLE
            5328  *
            5329  *             VICR * GIR  (FOR 2ND READING)
020C' 1000  5330  CAL2FT  DATA   4096          DC VOLTS ABS (LLDC),   0, SINGLE
020E' 0C80  5331         DATA   3200          HI LVL DC ABS,         1, SINGLE
0210' 1000  5332         DATA   4096          POTENTIOMETER,         2, SINGLE
0212' 1000  5333         DATA   4096          5V POT REF,            3, SINGLE
0214' 0DBC  5334         DATA   3516          TEMP. BOLBS,           4, SINGLE
0216' 0AE1  5335         DATA   2785          AC RATIO # 1,          5, DOUBLE
0218' 1000  5336         DATA   4096          DC RATIO # 1,          6, DOUBLE
021A' 1400  5337         DATA   5120          DC RATIO # 2,          7, DOUBLE
021C' 17F9  5338         DATA   6137          SYNCHRO,               8, DOUBLE
021E' 0AE1  5339         DATA   2785          AC RATIO # 2,          9, DOUBLE
            5340  *
```

```
                    5341  *
                    5342
                    5343  ****************************
                    5344
                    5345  *
                    5346  *       SYSTEM-CALIBRATION-BUFFER-OFFSET  (USED BY "CALRT")
                    5347  *       FORMAT PER ANALOG TYPE TABLE (ANTYPT)
                    5348  *       (0: GND. 2 : 2.5 VOLT FOR TEMP BULB(TYPE 4)
                    5349  *        4: 5 VOLT FOR LLDC(TYPE 0), POT(TYPE 2 & 3)
                    5350  *        6: 25 VOLT FOR SYNCHRO(TYPE 8)
0220' 1A 1C 1C      5351  CALSOT  BYTE    26,28,28,28,28,28,28,28,28,30
0223' 1C 1C 1C
0226' 1C 1C 1C
0229' 1E
                    5352          EVEN
                    5353  *
                    5354
                    5355  ****************************
                    5356
                    5357  *
                    5358  *               USED BY CALMT
                    5359  *       CAL LIMITS      (A TOLERANCE OF %10 IS GIVEN.
                    5360  *                        THE LIMITS GIVEN HERE ARE THE LOWER OR UPPER
                    5361  *                        LIMITES.)
                    5362  *       LOWER LIMIT
022A' FEB8          5363  CALLT   DATA    ()FEB8)                 GND. (- 4096 * %5)
                    5364  *                       SIG-LIMIT
022C' 0E66          5365          DATA    ()2000-)119A)           DC-VOLTS-ABS (5V)
022E' 0B44          5366          DATA    ()2000-)14BC)           HILVL-DC-ABS (5V)
0230' 0E66          5367          DATA    ()2000-)119A)           POTENTIOMETER (5V)
0232' 0E66          5368          DATA    ()2000-)119A)           5V. POT. REF. (5V)
0234' 0C93          5369          DATA    ()2000-)136D)           TEMP. BOLBS (5V)
0236' 0E66          5370          DATA    ()2000-)119A)           AC-VOLTS-RATIO-#1 (5V)
0238' 0E66          5371          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#1 (5V)
023A' 0E66          5372          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#2 (5V ?)
023C' 159A          5373          DATA    ()2000-)0A66)           SYNCHRO (25V)
023E' 0B44          5374          DATA    ()2000-)14BC)           AC VOLTS RATIO #2 (25V)
                    5375  *                       REF-LIMIT
0240' FEB8          5376          DATA    )FEB8                   GND (- 4096 * 5%)
0242' 0000          5377          DATA    0                       DC-VOLTS-ABS (NOT USED)
0244' 0000          5378          DATA    0                       HILVL-DC-ABS (NOT USED)
0246' 0000          5379          DATA    0                       POTENTIOMETER (NOT USED)
0248' 0000          5380          DATA    0                       5V. POT. REF. (NOT USED)
024A' 0000          5381          DATA    0                       TEMP. BOLBS (NOT USED)
024C' 0B44          5382          DATA    ()2000-)14BC)           AC-VOLTS-RATIO-#1 (25V)
024E' 0E66          5383          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#1 (5V)
0250' 0E66          5384          DATA    ()2000-)119A)           DC-VOLTS-RATIO-#2 (5V ?)
0252' 159A          5385          DATA    ()2000-)0A66)           SYNCHRO (25V)
0254' 0B44          5386          DATA    ()2000-)14BC)           AC VOLTS RATIO #2 (25V)
                    5387  *
                    5388  *               UPPER LIMIT
0256' 0148          5389  CALULT  DATA.   ()2000-)1EB8)           GND. (- 4096 * %5)
```

```
                    5390  *                    SIG-LIMIT
0258' 119A          5391        DATA    ()2000-)0E66)       DC-VOLTS-ABS (5V)
025A' 0DA2          5392        DATA    ()2000-)125E)       HILVL-DC-ABS (5V)
025C' 119A          5393        DATA    ()2000-)0E66)       POTENTIOMETER (5V)
025E' 119A          5394        DATA    ()2000-)0E66)       5V. POT. REF. (5V)
0260' 0F3B          5395        DATA    ()2000-)10C5)       TEMP. BOLBS (5V)
0262' 119A          5396        DATA    ()2000-)0E66)       AC-VOLTS-RATIO-#1 (5V)
0264' 119A          5397        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#1 (5V)
0266' 119A          5398        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#2 (5V ?)
0268' 1A66          5399        DATA    ()2000-)059A)       SYNCHRO (25V)
026A' 0DA2          5400        DATA    ()2000-)125E)       AC VOLTS RATIO #2 (25V)
                    5401  *                    REF-LIMIT
026C' 0148          5402        DATA    )2000-)1EB8         GND (- 4096 * 5%)
026E' 0000          5403        DATA    0                   DC-VOLTS-ABS (NOT USED)
0270' 0000          5404        DATA    0                   HILVL-DC-ABS (NOT USED)
0272' 0000          5405        DATA    0                   POTENTIOMETER (NOT USED)
0274' 0000          5406        DATA    0                   5V. POT. REF. (NOT USED)
0276' 0000          5407        DATA    0                   TEMP. BOLBS (NOT USED)
0278' 0DA2          5408        DATA    ()2000-)125E)       AC-VOLTS-RATIO-#1 (25V)
027A' 119A          5409        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#1 (5V)
027C' 119A          5410        DATA    ()2000-)0E66)       DC-VOLTS-RATIO-#2 (5V ?)
027E' 1A66          5411        DATA    ()2000-)059A)       SYNCHRO (25V)
0280' 0DA2          5412        DATA    ()2000-)125E)       AC VOLTS RATIO #2 (25V)
                    5413
```

```
                    5415  ************************************************************************
                    5416  *
                    5417  *                    BITE
                    5418  *                    ====
                    5419  *
                    5420  *       INTERNAL REF. MUX ADDRESS TABLE
                    5421  *       (SEE BTDRQ.SRC AND BTLMT.SRC)
                    5422  *
                    5423  ************************************************************************
                    5424  *
                    5425  *       ANALOG POSITION TABLE
                    5426  *
0282'               5427  ANPSTB
                    5428  *       FORM: ANPT      N1,N2,N3
                    5429  *                       N1      =       INPUT START BIT
                    5430  *                       N2      =       INPUT MASK WORD INDEX
                    5431  *                       N3      =       ROUNDING
                    5432  *                                       BLANK FOR NO ROUNDING
                    5433  *                                       NONBLANK FOR ROUNDING
                    5434
                    5435+         ANPT    1,11,           VERTICAL ACCELLERATION
0282'  2C           5446A                 BYTE    1-1+XXX
                    5448+         ANPT    2,10,           LATERAL ACCELLERATION
0283'  29           5459A                 BYTE    2-1+XXX
                    5461+         ANPT    2,10,           LONGITUDINAL ACCELLERATION
0284'  29           5472A                 BYTE    2-1+XXX
                    5474+         ANPT    2,10,ROUND      T/E FLAP LEFT
0285'  A9           5483A                 BYTE    2-1+XXX+)80
                    5487+         ANPT    2,10,ROUND      T/E FLAP RIGHT
0286'  A9           5496A                 BYTE    2-1+XXX+)80
                    5500+         ANPT    2,10,ROUND      HORIZONTAL STABILIZER POSITION
0287'  A9           5509A                 BYTE    2-1+XXX+)80
                    5513+         ANPT    3,9,            FLAP HANDLE POSITION
0288'  26           5524A                 BYTE    3-1+XXX
                    5526+         ANPT    2,10,ROUND      SPOILER HANDLE POSITION
0289'  A9           5535A                 BYTE    2-1+XXX+)80
                    5539+         ANPT    1,0,0           5V POT REF
028A'  80           5548A                 BYTE    1-1+XXX+)80
                    5552  *
                    5553          EVEN
                    5554  *
                    5555          END
```

```
                    1          IDT    ANSYN
                    2
                    3          SUBTTL  SYNCHRO SCALING/CONVERSION
                    4
                    5  *       CALLING SEQ:    CALL    @ANSYN
                    6
                    7  *-------------------------------------------------------------------+
                    8  *                                                                   !
                    9  *       ANSYN CONVERTS THE SYNCHRO DUAL INPUT DATA TO AN ANGLE VALUE
                   10  *       REPRESENTED IN 12 BITS (1 COUNT = 360 DEG / 4096).
                   11  *       IT TESTS THE SIGNS AND ABSOLUTE VALUE OF THE TWO INPUTS AND
                   12  *       ACCORDINGLY USES ANEQUATION (ONE OF EIGHT) TO COMPUTE THIS
                   13  *       ANGLE.
                   14  *                                                                   !
                   15  *-------------------------------------------------------------------+
                   16  *       VERSION : 1
                   17  *       PROGRAMMED BY : M.CONSTANTINIDES
                   18  *       CHECKED BY : M.CONSTANTINIDES
                   19
                   20
                   21          INTERN  ANSYN
                   22  * REFERD MODULES:
                   23          EXTERN  ATANV           (ATAN)
                   24
=0000              25  ATS     EQU     R0      =       SCRATCH
=0001              26  ATR     EQU     R1      =       SCRATCH
=0002              27  PNT     EQU     R2      =       SCRATCH
=0003              28  VCTS    EQU     R3      =       1ST AN-DATA (INPUT)
=0004              29  VCTR    EQU     R4      =       2ND AN-DATA (INPUT), XDUCER VALUE (OUTPUT)
=0005              30  SBP     EQU     R5      =       SCRATCH
                   31
                   37          INCLUDE REGDEF          REGISTER DEFENITIONS
                   56          INCLUDE CNSTNT          CONSTANTS
                  184          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  485          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  715          INCLUDE JMPMAC          JUMP MACROS
                  749          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  764          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000            1141          RSECT   ANSYN
                 1142  ********************************************************************************
```

```
0000'              1144   ANSYN
0000'  04C2        1145          CLR     PNT                 POINTER = 0, FOR VCTS, VCTR > 0
                   1146+         DOIF    VCTR,MI,,,,         IF VCTR < 0,
0002'  C104        1280B           MOV   VCTR,VCTR
0004'  1101        1328E           JLT   $+4
0006'  1001        1329E           JMP   91$
0008'  0582        1452          INC     PNT                   POINTER = 1
                   1453+         ENDBLK                      ENDIF
000A'              1585E  91$
                   1673+         DOIF    VCTS,MI,,,,         IF VCTS < 0,
000A'  C0C3        1807B           MOV   VCTS,VCTS
000C'  1101        1859E           JLT   $+4
000E'  1001        1860E           JMP   92$
0010'  05C2        1979          INCT    PNT                   POINTER = 2
                   1980+         ENDBLK                      ENDBLK
0012'              2116E  92$
0012'  0743        2200          ABS     VCTS
0014'  0744        2201          ABS     VCTR
0016'  C143        2202          MOV     VCTS,R5             R5 = VCTS
0018'  C004        2203          MOV     VCTR,R0             MSW R0 = VCTR
001A'  04C1        2204          CLR     R1                  LSW
                   2205+         DOIF    VCTS,LEQ,VCTR,      IF VCTS >= VCTR,
001C'  8103        2343B           C     VCTS,VCTR
001E'  1504        2392E           JGT   93$
0020'  0222 0004   2511          AI      PNT,C4                POINTER ; (FOR VCTR > VCTS)
0024'  C003        2512          MOV     VCTS,R0               MSW VCTS SCB16
0026'  C144        2513          MOV     VCTR,R5               R5 = VCTR
                   2514+         ENDBLK                      ENDIF
0028'              2654E  93$
0028'  C040        2734          MOV     R0,R1               R0,R1 = DIVIDEND (2 ** 14)
002A'  0AE1        2735          SLA     R1,C14
002C'  0820        2736          SRA     R0,C2
002E'  3C05        2737          DIV     R5,R0               R = VMTR/VMTS OR VMTS/VMTR (SCB14)
                   2738   *                                COMPUTE ARC-TAN (.866*R/(1-.5R))=(Y/X)
                   2739   *                                GEN. SYN-EQ-TABLE POINTER (EQ 0-7)
0030'  0204 0006   2740          LI      R4,C6               6-BYTES PER DATA-ITEM IN TABLE
0034'  3902        2741          MPY     R2,R4               EQ.NT.*6 = OFFSET FOR EQ. N
0036'  0225 0006'  2742          AI      R5,ANSETB           R5 = SYN-EQ-TABLE POINTER FOR ER #N
                   2743   *                                COMPUTE Y = .866 * R
003A'  C040        2744          MOV     R0,R1               R(SCB14)
003C'  3860 0000'  2745          MPY     @KF866,R1           Y(SCB29 OR B13) = .866(SCB15) * R(SCB14)
                   2746   *                                COMPUTE X = 1 - (.5 * R)
0040'  C0C0        2747          MOV     R0,R3               R (VCTR/VCTS) SCBB14
0042'  38E0 0002'  2748          MPY     @KF5,R3             X1(SCB29 OR B13) = .5(SCB15) * R(SCB14)
0046'  C0A0 0004'  2749          MOV     @K1,R2              1, SCBB13
004A'  C135        2750          MOV     *R5+,R4
                   2751+         DOIF    ,NE,,               IF 1 - (.5 * R) TYPE
004C'  1301        2930E             JEQ   94$
004E'  0503        3057            NEG   R3                    X1 = -X1
                   3058+         ENDBLK                      ENDIF
0050'              3202E  94$
0050'  A083        3278          A       R3,R2               X(SCB13) = X1
```

```
                    3279  *                              COMPUTE ARC-TAN (Y / X)
                    3280+         CALLWP  ATANV                  R1 = PHI = ARC-TAN (X/Y) 180-DEG (@B15)
0052'  0420 0001*   3282A           BLWP  @ATANV
                    3284  *                              COMPUTE VCT = ANGLE - PHI
0056'  C0F5         3285          MOV     *R5+,R3                ANGLE (240, 60, 120 OR 300) FOR EQ #N
0058'  C115         3286          MOV     *R5,R4
                    3287+         DOIF    ,NE,,                  IF ANGLE + PHI TYPE
005A'  1301         3470E             JEQ   95$
005C'  0501         3593            NEG   R1                       PHI NEGATIVE
                    3594+         ENDBLK                         ENDIF
005E'               3742E 95$
005E'  A0C1         3814          A       R1,R3                  ANGLE +/- PHI = VCT (180-DEG SC@B15)
                    3815  *                              RESCALE FOR OUTPUT TO 180-DEG @B11
0060'  0943         3816          SRL     R3,C4                  RESCALE 180-DEG @B11
0062'  045B         3817          RT
                    3818  ********************************************************************************
                    3819+         PRVDAT
0000'  6ED9         3821  KF866   DATA    28377          = .866 SC@B15
0002'  4000         3822  KF5     DATA    16384          = .5 SC@B15
0004'  2000         3823  K1      DATA    8192           = 1. SC@B15
                    3824  *
                    3825
                    3826  * ANSETB CONTAINS EIGHT SYNCHRO EQUATION INFORMATIONS FOR SYNCHRO
                    3827  * PARAMETER COMPUTATION (VCT). THE TABLE CONSISTS OF EIGHT DATA
                    3828  * ITEMS, ONE PERSYNCHRO EQUATION. EACH DATA ITEM CONTAINS THREE
                    3829  * WORDS: 1ST-WORD = -1(FOR PHI1), 0(FOR PHI2)
                    3830  *        2ND-WORD = ANGLE(180-DEG SC@B15)
                    3831  *        3RD-WORD = -1(FOR ANGLE -PHI), 0(FOR ANGLE +PHI).
                    3832  *
0006'               3834  ANSETB
                    3835+         DATBL   -1,43691,-1    PHI1, 240-DEG, -PHI1 (240-PHI1)
                    3856+         DATBL   0,43691,0,     PHI2, 240-DEG, +PHI2 (240+PHI2)
                    3877+         DATBL   0,10923,0,     PHI2, 60-DEG, +PHI2 (60 + PHI2)
                    3898+         DATBL   -1,10923,-1,   PHI1, 60-DEG, -PHI1 (60 - PHI1)
                    3919+         DATBL   -1,21845,0,    PHI1, 120-DEG, +PHI1 (120 + PHI1)
                    3940+         DATBL   0,54613,-1     PHI2, 300-DEG, -PHI2 (300 - PHI2)
                    3961+         DATBL   0,21845,-1,    PHI2, 120-DEG -PHI2 (120 - PHI2)
                    3982+         DATBL   -1,54613,0,    PHI1, 300-DEG, +PHI1 (300 + PHI1)
                    4003
                    4004          END
```

```
                    1          IDT     ATAN
                    2
                    3          SUBTTL  ATAN - SINGLE PRECISION ARC-TANGENT
                    4
                    5  *       CALLING SEQ:    CALLWP  @ATANV
                    6
                    7  *--------------------------------------------------------------------------+
                    8  *                                                                          !
                    9  *       COMPUTES ARCTAN Y/X USING FOLLOWING APPROXIMAT:
                   10  *       ARCTAN Y/X(=Z) =((((C9*Z**2+C7)*Z**2+C5)*Z**2+C3)*Z**2+C1)*Z
                   11  *                                                                          !
                   12  *--------------------------------------------------------------------------+
                   13  *       VERSION : 1
                   14  *       PROGRAMED BY : N.CONSTANTINIDES
                   15  *       CHECKED BY   : N.CONSTANTINIDES
                   16
                   17          INTERN  ATAN
                   18  * REFERD MODULES:
                   19          EXTERN  SDIV            = SIGNED DIVIDE
                   20          EXTERN  SSMRR           = SIGNED MULTIPLY/ROUND/RESCALE
                   21
                   22  * INPUTS:       CALLER'S R1 = Y
                   23  *               CALLER'S R2 = X
                   24  *               BOTH IN THE SAME UNITS AND WITH THE SAME SCALE
                   25  *               FACTOR.
                   26  * OUTPUTS:      CALLER'S R1 = ARCTAN Y/X 180 DEG SCALED AT B15
                   27  *       R0, R1, R3-THRU-R9  = MODIFIED
                   28
                   34          INCLUDE REGDEF          REGISTER DEFENITIONS
                   53          INCLUDE CNSTNT          CONSTANTS
                  186          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  487          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  717          INCLUDE JMPMAC          JUMP MACROS
                  751          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  766          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000            1143          RSECT   ATAN
                 1144  ********************************************************************************
```

```
0000'                1146  ATAN
                     1147  *     GET ARGUMENT
0000' C24D           1148        MOV   R13,R9        GET CALLER'S WP
0002' 05C9           1149        INCT  R9            BUMP TO CALLER'S R1
0004' C159           1150        MOV   *R9,R5        GET Y
0006' C1A9 0002      1151        MOV   @2(R9),R6     GET X
                     1152  *
000A' C1C5           1153        MOV   R5,R7         SAVE Y IN R7
000C' 0745           1154        ABS   R5            ABSOLUTE (Y)
                     1155  *
000E' C206           1156        MOV   R6,R8         SAVE X IN R8
0010' 0746           1157        ABS   R6            ABSOLUTE (X)
0012' 1603           1158        JNE   2$            BRANCH IF ABS(X) NOT ZERO
                     1159  *     X = 0
0014' 0203 2000      1160        LI    R3,>2000      SET ANGLE TO 45 DEGREES
0018' 1025           1161        JMP   9$            EXIT
                     1162  *
001A' C045           1163  2$    MOV   R5,R1         ABS (Y)
001C' A046           1164        A     R6,R1         =ABS(Y) + ABS(X)
001E' 1503           1165        JGT   4$            JUMP IF NO OVERFLOW
0020' 1302           1166        JEQ   4$
                     1167  *     SUM OF ABS(Y) AND ABS(X) OVERFLOW
0022' 0815           1168        SRA   R5,1          DIVIDE ABS(Y) BY 2
0024' 0816           1169        SRA   R6,1          DIVIDE ABS(X) BY 2
                     1170  *
0026' C045           1171  4$    MOV   R5,R1
0028' 1603           1172        JNE   5$            JIF ABS(Y) NOT ZERO
                     1173  *     Y = 0
002A' 0203 E000      1174        LI    R3,>E000      SET ANGLE TO -45 DEGREES
002E' 101A           1175        JMP   9$
0030' A046           1176  5$    A     R6,R1         =ABS(Y) + ABS(X)
0032' C0C5           1177        MOV   R5,R3
0034' 60C6           1178        S     R6,R3         =ABS(Y) - ABS(X)
0036' 04C4           1179        CLR   R4            CLEAR R4 FOR DIVIDE INSTRUCTION
0038' 0813           1180        SRA   R3,1          RESCALE TO B14
                     1181+       CALL  SDIV          Z=ABS(Y) - ABS(X)/ABS(Y) + ABS(X)
003A' 06A0 0001*     1188A         BL  @SDIV
003E' C143           1190        MOV   R3,R5         SAVE Z IN R5
0040' C043           1191        MOV   R3,R1         Z IN R3 AND R1 FOR MPY
                     1192  *                        =Z*Z  ROUND AND RESCALE IN R3
                     1193+       CALL  SSMRR         MULTIPLY R1 & R3, ROUND & RESCALE TO B15(ORB0)
0042' 0640 0002*     1200A         BL  @SSMRR
0046' C003           1202        MOV   R3,R0         Z**2
                     1203  *     HASTINGS APPROXIMATION COMPUTATION
0048' 0206 0006      1204        LI    R6,6          FOR 4 LOOPS
004C' C0E0 0008'     1205        MOV   @CAT9,R3      C9 CONSTANT
                     1206  *                        CONSTANT*Z**2(=R1*R3) ROUND & RESCALE
0050' C040           1207  8$    MOV   R0,R1         Z**2
                     1208+       CALL  SSMRR         MULTIPLY R1&R3, R3 ROUND & RESCALE TO B0
0052' 06A0 0002*     1215A         BL  @SSMRR
0056' A0E6 0000'     1217        A     @CAT1(R6),R3  ADD CONSTANT
005A' 0646           1218        DECT  R6            DECREMENT
```

```
00?C' 18F9          1219          JOC     8$              LOOP 4 TIMES
                    1220  *
00??' C045          1221          MOV     R5,R1           =Z
                    1222  *                               R3=Z*(SUM)  ROUND & RESCALE
                    1223+         CALL    SSMRR           MULTIPLY R1&R3. R3 ROUND & RESCALE TO B0
00?0' 06A0 0002*    1230A           BL    @SSMRR
                    1232  *     QUADRANT TEST
00??' C208          1233  9$      MOV     R8,R8           X=- ?
00??' 1103          1234          JLT     10$             BRANCH IF QUADRANT 2 OR 3
                    1235  *       DATA IN QUADRANT 1 OR 4
00??' 0223 2000     1236          AI      R3,>2000        ADD 45 DEGREES
00??' 1004          1237          JMP     12$
                    1238  *       DATA IN QUADRANT 2 OR 3
00??' 0201 6000     1239  10$     LI      R1,>6000        =135 DEGREES
00??' 6043          1240          S       R3,R1           =135 DEG - RESULT
00??' C0C1          1241          MOV     R1,R3
                    1242  *
00??' C1C7          1243  12$     MOV     R7,R7           Y=- ?
00??' 1502          1244          JGT     14$             BRANCH IF QUADRANT 1 OR 2
00??' 1301          1245          JEQ     14$
                    1246  *       DATA IN QUADRANT 3 OR 4
00??' 0503          1247          NEG     R3              NEGATE RESULT
                    1248  *
00??' C643          1249  14$     MOV     R3,*R9          PUT RESULT IN CALLER'S R1
00??' 0380          1250          RTWP
                    1251  ********************************************************************************
                    1252  * PRIVATE DATA
                    1253+         PRVDAT
                    1255  *     HASTINGS CONSTANT
                    1256  *
00??' 28BE          1257  CAT1    DATA    >28BE           C1  0.3183026  SC 1 AT B15
00??' F273          1258          DATA    >F273           C3  -.1058774  SC 1 AT B15
00??' 07E3          1259          DATA    >07E3           C5  0.0616068  SC 1 AT B15
00??' FB42          1260          DATA    >FB42           C7  -.0370617  SC 1 AT B15
00??' 0225          1261  CAT9    DATA    >0225           C9  0.0167601  SC 1 AT B15
00??' FF85          1262  CAT11   DATA    >FF85           C11 -.0037537  SC 1 AT B15
                    1263          END
```

```
                    1          IDT     ANVT
                    2          SUBTTL  AN-V-TRANSDUCER
                    3  *       CALLING SEQ:    CALL    @ANVT
                    4  ********************************************************************************
                    5  *                                                                              *
                    6  *       ANVT CALCULATES VCTS (VCTS=(VMTS-VMOS)*KS), IF DUAL CONVERSION         *
                    7  *       (TYPE > 4) CALCULATES VCTR (VCTR = (VMTR-VMOR)*KR) AND THEN            *
                    8  *       ACCORDING O THE PARAMETER TYPE CALLS THE SCALING/CONVERSION            *
                    9  *       SUBROUTINE, IN THE CASE OF 5V.-POT.-REF, ((TYPE=3) THE VALUE           *
                   10  *       IS ALREADY STORED INTO 'VCPR') AND DC-VOLTAGE THERE IS NO NEED         *
                   11  *       FOR SCALING (THE SUBR ONLY RETURNS).                                   *
                   12  *                                                                              *
                   13  ********************************************************************************
                   14  *       VERSION: 1
                   15  *       PROGRAMMED BY: N.CONSTANTINIDES
                   16  *       CHECKED BY: N.CONSTANTINIDES
                   17          INTERN  ANVT
                   18  * REFERRED MODULES:
                   19          EXTERN  SMPY
                   20          EXTERN  ANNDC
                   21          EXTERN  ANPOT
                   22          EXTERN  ANTB
                   23          EXTERN  ANAC1
                   24          EXTERN  ANDCR
                   25          EXTERN  ANSYN
                   26          EXTERN  ANAC2
                   27          EXTERN  ANOSCL
                   28          EXTERN  ANP5VR
                   29  * GLOBALS
                   30  *       (RAM)
                   31          EXTERN  VMOSBF          CAL-FACT VMOS BUFFER
                   32          EXTERN  VMORBF          CAL-FACT VMOR BUFF
                   33          EXTERN  KSBUF           CAL-FACT KS-BUFF
                   34          EXTERN  KRBUF           CAL-FACT KR BUFF
                   35
                   36  *               R0      =       SCRATCH
                   37  *               R1      =       SCRATCH
     =0002         38  ANTYP   EQU     R2      =       AN-TYPE (INPUT)
     =0003         39  VTS     EQU     R3      =       1ST AN-DATA (INPUT), XDUCER VALUE (OUTPUT)
     =0004         40  VTR     EQU     R4      =       2ND AN-DATA (INPUT)
                   41  *               R5      =       SCRATCH
     =0009         42  ATO     EQU     R9      =       AN-TABLES-OFFET (INPUT/OUTPUT)
                   43
                   44          INCLUDE ENCLOS
                   46  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   48          INCLUDE REGDEF          REGISTER DEFENITIONS
                   67          INCLUDE CNSTNT          CONSTANTS
                  195          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  496          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  726          INCLUDE JMPMAC          JUMP MACROS
                  760          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  775          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
```

```
          =0000       1151          RSECT   AMVT
                      1152    ##############################################################################
```

```
0000'              1154  ANVT
0000'  C80B 0000'  1155          MOV     LINK,@LINKZ     SAVE LINKER
                   1156+         INDEX   ATO,R5,                 OFFSET = TABLES-OFFSET * 2
0004'  C149        1159A           MOV   ATO,R5
0006'  0A15        1160A           SLA   R5,1
0008'  C004        1165          MOV     VTR,R0                  SAVE VMTR
                   1166+         GETVCT  S,                      GET VCTS = (VMTS - VMOS) * KS
000A'  60E5 000B*  1167A           S     @VMOSBF(R5),R3          XS = VMTS - VMOS
000E'  C065 000D*  1185D           MOV   @KSBUF(R5),R1
0012'  06A0 0001*  1193B           BL    @SMPY
0016'  0A23        1195A           SLA   R3,C2                   SC@BO
0018'  09E4        1196A           SRL   R4,C14
001A'  E103        1197A           SOC   R3,R4
001C'  C0C0        1198          MOV     R0,VTS                  R3 = VMTR
001E'  C004        1199          MOV     VTR,R0                  SAVE VCTS
                   1200+         DOIF    ANTYP,GE,C5,X,,         IF TYPE > 4, DUAL-CONV
0020'  0282 0005   1319B             CI    ANTYP,C5
0024'  1109        1376E              JLT   91$
                   1506+           GETVCT R,                             GET VCTR = (VMTR - VMOR) * KR
0026'  60E5 000C*  1507A           S     @VMORBF(R5),R3          XR = VMTR - VMOR
002A'  C065 000E*  1525D           MOV   @KRBUF(R5),R1
002E'  06A0 0001*  1533B           BL    @SMPY
0032'  0A23        1535A           SLA   R3,C2                   SC@BO
0034'  09E4        1536A           SRL   R4,C14
0036'  E103        1537A           SOC   R3,R4
                   1538+         ENDBLK                          IFEND
0038'              1670E  91$
0038'  C0C0        1758          MOV     R0,VTS                  R3 = VCTS (VCTR IN R4)
                   1759+         INDEX   ANTYP,                  TYPE = TYPE * 2
003A'  0A12        1765A           SLA   ANTYP,1
                   1768  *                                       CALL THE SUB-ROUTINE POINTED BY TYPE
                   1769+         CALLPT  SUBTBL,ANTYP            ACCORDING TO THE TYPE SCALE .. VALUE
003C'  C022 0000'  1771A           MOV   @SUBTBL(ANTYP),R0
0040'  0690        1772A           BL    *R0
0042'  C103        1774          MOV     VTS,VTR                 R4 = ANSW
                   1775+         DOIF    VTR,GE,C4096,X,,        IF VT > 4095, VT = 4095 (FFF)
0044'  0284 1000   1894B             CI    VTR,C4096
0048'  1102        1955E              JLT   92$
004A'  0204 0FFF   2081              LI    VTR,C4095
                   2082+         ENDBLK                          IFEND
004E'              2218E  92$
                   2302+         DOIF    VTR,NI,,                IF VT < 0, VT = 0.
004E'  C104        2436B             MOV   VTR,VTR
0050'  1101        2492E              JLT   $+4
0052'  1001        2493E              JMP   93$
0054'  04C4        2608              CLR   VTR
                   2609+         ENDBLK                          IFEND
0056'              2749E  93$
0056'  C2E0 0000'  2829          MOV     @LINKZ,LINK             RESTORE LINKER
005A'  045B        2830          RT
                   2831  ********************************************************************************
                   2832  * PRICVATE AREA
```

```
                    2833+          LOCR    PRIV,LINKZ              LINK-SAVE AREA
    0000'  =0002    2836A  LINKZ   BSS     2
                    2837+          PRVDAT
                    2839   *               SUBR CALLED ACCORDING TO THE TYPE
                    2840+  SUBTBL  DATBL   ANOSCL,ANHDC,ANPOT,ANP5VR,ANTB,ANAC1,ANDCR,ANDCR,ANSYN,ANAC2
    0000'  0009*    2842A            DATA  ANOSCL
    0002'  0002*    2846B            DATA  ANHDC
    0004'  0003*    2850C            DATA  ANPOT
    0006'  000A*    2854D            DATA  ANP5VR
    0008'  0004*    2858E            DATA  ANTB
    000A'  0005*    2862F            DATA  ANAC1
    000C'  0006*    2866G            DATA  ANDCR
    000E'  0006*    2870H            DATA  ANDCR
    0010'  0007*    2874I            DATA  ANSYN
    0012'  0008*    2878J            DATA  ANAC2
                    2896           END
```

```
                    1          IDT    BTACQ
                    2
                    3          SUBTTL BITE TEST
                    4
                    5  *       CALLING SEQ:   CALLWP  @BTACQW
                    6
                    7  *-------------------------------------------------------------+
                    8  *                                                             |
                    9  *       BTACQ ACQUIRES BITE DATA (GND AND VOLTAGE) IN CYCLE 1.
                   10  *       IT SENDS BITE MUX ADDR AND RECEIVES GND- AND V-CAL VALUES,
                   11  *       TESTS THE LIMITS AND FOR ERROR SETS ERROR STATUS WORD.
                   12  *                                                             |
                   13  *-------------------------------------------------------------+
                   14  *       VERSION : 3
                   15  *       PROGRAMMED BY : N.CONSTANTINIDES
                   16  *       CHECKED BY : N.CONSTANTINIDES
                   17  *
                   18  *
                   19  *
                   20          INTERN BTACQ
                   21  * REFERD MODULES:
                   22          EXTERN BTDRQ
                   23          EXTERN BTLMT
                   24          EXTERN BTVLMT
                   25          EXTERN IOB2SC
                   26  * GLOBAL AREA:
                   27  *       (ROM)
                   28          EXTERN D4              = 4
                   29  *       (RAM)
                   30          EXTERN CYFFRC          CYCLE-COUNT
                   31          EXTERN DASFLG          DAS-FLAG
                   32          EXTERN POCALF          POWER ON CAL/BITE FLAG
=0001              33  TPT     EQU    R1      =       SCRATCH
=0002              34  BTYP    EQU    R2      =       AN-TYPE
=0003              35  VOS     EQU    R3      =       1ST READING
=0004              36  VOT     EQU    R4      =       2ND READING
=0008              37  CNVFLG  EQU    R8      =       CONVERSION FLAG (INPUT, OUTPUT)
=000A              38  BTADP   EQU    R10     =       BT-ADDR-PTR (DOES NOT CHANG FOR BT-PROCESS)
                   39          INCLUDE ENCLOS
                   41  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   43          INCLUDE REGDEF         REGISTER DEFENITIONS
                   62          INCLUDE CNSTNT         CONSTANTS
                  190          INCLUDE SUBMAC         FUNCTIONAL MACROS
                  491          INCLUDE MSCMAC         MISCELLANEOUS MACROS
                  721          INCLUDE JMPMAC         JUMP MACROS
                  755          INCLUDE BLKMAC         OTHER MACROS (BY D. SCOTT)
                  770          INCLUDE LBLMAC         HANDLES MACROS AUTOMATICALLY
=0000            1146          RSECT  BTACQ
```

```
                    1148  ********************************************************************************
0000'               1149  BTACQ
0000'  0288 0004    1150        CI    CNVFLG,C4              0 TO 4 VALID
0004'  1B32         1151        JH    100$                   EXIT IF JUMP OFFSET INVALID
                    1152  *                                  DO ACCORDING TO CONV-FLAG (0, 2 OR 4)
                    1153+     DOCASE  CNVFLG,50$,51$,52$
0006'  0468 000A'   1154A       B     @234$(CNVFLG)
000A'               1156A 234$
000A'  1002         1159B       JMP   50$
000C'  100C         1162C       JMP   51$
000E'  1019         1165D       JMP   52$
                    1174  ***
0010'               1175  50$
0010'  05C8         1176        INCT  CNVFLG                 CONV-FLAG = CONV-FLAG + 1(*2)
0012'  060A         1177        DEC   BTADP                  BT-ADDR-PTR = BT ADDR-PTR - 1
                    1178+       DOIF  ,LEQ,,,,               IF BT-ADDR-PTR = 0,
0014'  1504         1357E           JGT   91$
0016'  020A 0005    1484          LI    BTADP,C5                     BT-ADDR-PTR = 5,
001A'  0720 0008*   1485          SETO  @POCALF              END OF POWER ON BITE TEST
                    1486+       ENDBLK
001E'               1618E 91$
001E'  04C1         1706        CLR   TPT                    SET POINTER FOR GND-REF
                    1707+       CALL  BTDRQ                  SEND MUX ADDRESSES
0020'  06A0 0001*   1714A         BL    @BTDRQ
0024'  1020         1716        JMP   60$
                    1717  ***
0026'               1718  51$
0026'  05C8         1719        INCT  CNVFLG                 CONV-FLAG = CONV-FLAG + 1(*2)
                    1720+       CALL  IOB2SC,<VOT,=,@CADCDB> READ GND-REF AND CONVERT TO 2S-COMP
0028'  C120 FFB0    1736C         MOV   @CADCDB,VOT
002C'  06A0 0004*   1744A         BL    @IOB2SC
                    1746+       CALL  BTLMT,<R0,=,C6,I>      CHECK LIMITS ON GND-REF
0030'  0200 0006    1756C         LI    R0,C6
0034'  06A0 0002*   1770A         BL    @BTLMT
0038'  C0C4         1772        MOV   VOT,VOS                SAVE GND-REF VALUE
                    1773+       CALL  BTDRQ,<TPT,=,BTADP>    GET PTR, SEND MUX-ADR TO GET BT-VALUE
003A'  C04A         1789C         MOV   BTADP,TPT
003C'  06A0 0001*   1797A         BL    @BTDRQ
0040'  1012         1799        JMP   60$
                    1800  ***
0042'               1801  52$
0042'  04C8         1802        CLR   CNVFLG                 CONV-FLAG = 0
0044'  C820 0008*   1803        MOV   @POCALF,@DASFLG        DURING POWER ON BITE, POCALF = 2
0048'  0007*
                    1804  *                                  THEN SET TO -1TO DO ANALOG BITE DURING PWER ON
                    1805+       CALL  IOB2SC,<VOT,=,@CADCDB> READ BITE AND CONVERT TO 2S-COMP
004A'  C120 FFB0    1821C         MOV   @CADCDB,VOT
004E'  06A0 0004*   1829A         BL    @IOB2SC
0052'  6103         1831        S     VOS,VOT                VOT = VOT - VOS
                    1832+       CALL  BTVLMT,<R0,=,BTADP>    GET PTR, CHECK BITE LIMIT
0054'  C00A         1848C         MOV   BTADP,R0
0056'  06A0 0003*   1856A         BL    @BTVLMT
```

```
                    1858  ***
005A'  C020 0007*   1859          MOV     @DASFLG,R0
005E'  1105         1860          JLT     100$                    JIF NOT IN POWER ON ISR
                    1861+         STARTI  INT6,                   START ADC TO GET ADC INTERRUPT FOR NEXT BITE
0060'  04E0 FFB0    1863A          CLR    @CADCDB
0064'  1002         1865          JMP     100$
                    1866+ 60$     STARTI  INT6,                   START ADC INTERRUPT
0066'  04E0 FFB0    1868A          CLR    @CADCDB
                    1870+        ENDO                            DOEND
006A'               1872A 90$
006A'  0380         1874  100$    RTWP
                    1875  ********************************************************************************
                    1876          END
```

```
                      1          IDT     BTDRQ
                      2
                      3          SUBTTL  BITE DATA REQUEST
                      4
                      5  *       CALLING SEQ:    CALL    @BTDRQ
                      6
                      7  *-------------------------------------------------------------+
                      8  *                                                             !
                      9  *       BTDRQ SENDS NEXT BITE (POWER SUPPLIE) MUX ADDR TO RECEIVE
                     10  *       BITE DATA.
                     11  *       THE VALUES TO BE SENT TO THE DIFFERENT CRU ADDRESS ARE THE
                     12  *       SAME AND USE A TABLE (SEE BELOW) BUT THE VALUE FOR THE CRU
                     13  *       ADDRESS 330 (HEX) WHICH ARE STORED IN ANOTHER TABLE.
                     14  *                                                             !
                     15  *-------------------------------------------------------------+
                     16  *       VERSION : 2
                     17  *       PROGRAMMED BY : M.CONSTANTINIDES
                     18  *       CHECKED BY : M.CONSTANTINIDES
                     19
                     20
                     21          INTERN  BTDRQ
                     22
                     23
        =0000        24  PTR     EQU     R0      =       SCRATCH
        =0001        25  BUP     EQU     R1      =       OFFSET (INPUT)
                     26
                     27          INCLUDE ENCLOS
                     29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     31          INCLUDE REGDEF          REGISTER DEFENITIONS
                     50          INCLUDE CNSTNT          CONSTANTS
                    178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    709          INCLUDE JMPMAC          JUMP MACROS
                    743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000      1134          RSECT   BTDRQ
                   1135  ********************************************************************************
0000'              1136  BTDRQ
0000'  0200 0001'  1137          LI      PTR,BTVL1               SET POINTER TO MUX VALUES
                   1138  *                                       (VALUES IN HEX)
                   1139+         CRUWRT  BTCA1,*PTR+,C8          SEND MUX ADDR 90 TO CRU 318-31F
0004'  020C 0630   1141A           LI    CRU,BTCA1
0008'  3230        1142A           LDCR  *PTR+,C8
                   1144+         CRUWRT  BTCA2,*PTR+,ZERO        SEND MUX ADDR 0 TO CRU 320-32F
000A'  020C 0640   1146A           LI    CRU,BTCA2
000E'  3030        1147A           LDCR  *PTR+,ZERO
                   1149+         CRUWRT  BTCA3,@BTVL3(BUP),C8,   TO CRU 330-337 ACCORDING TO OFFSET
0010'  020C 0660   1151A           LI    CRU,BTCA3
0014'  3221 0006'  1152A           LDCR  @BTVL3(BUP),C8
                   1154+         CRUWRT  BTCA4,*PTR,C8           SEND MUX ADDR 110 TO CRU 340-34F
0018'  020C 0690   1156A           LI    CRU,BTCA4
001C'  3210        1157A           LDCR  *PTR,C8
```

```
                    1830   *                                          TO DO CAL OF ALL CHANNELS DURING POWER ON
                    1831+          CALL    CALRT,<CNT,=,C1,I>         READ AND TEST V-CAL DATA
0058'  0201 0001    1841C            LI    CNT,C1
005C'  06A0 0002*   1855A            BL    @CALRT
                    1857+          CALL    CALKF                      CALCULAT AND STOR KS/KR
0060'  06A0 0003*   1864A            BL    @CALKF
                    1866   ***
0064'  C020 0009*   1867           MOV     @DASFLG,R0              JIF NOT ON POWER ON
0068'  1105         1868           JLT     100$
                    1869+          STARTI  INT6,                   START ADC TO GET ADC INTERRUPT FOR NEXT CAL
006A'  04E0 FFB0    1871A           CLR    @CADCDB
006E'  1002         1873           JMP     100$
                    1874+  60$     STARTI  INT6,           START ADC INTERRUPT
0070'  04E0 FFB0    1876A           CLR    @CADCDB
                    1878+         ENDO                              ENDO
0074'               1880A  90$
0074'  0380         1882   100$     RTWP
                    1883   ********************************************************************************
                    1884           END
```

```
                     1          IDT     CALKF
                     2
                     3          SUBTTL  CALCULATE KS/KR FACTOR FOR CALIBRATION
                     4
                     5  *       CALLING SEQ:    CALL    @CALKF
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       CALKF CALCULATES THE CORRECTION AND SCALING VALUES (KS, KR)
                    10  *       FOR THE CALIBRATION OF THE TRANSDUCER MEASUREMENT.
                    11  *       EQUATION USED:  KS = (VICS * GIS) / (VMCS - VMOS),
                    12  *       IF DUAL-CONVERSION (TYPE > 4), KR AND VMOR REPLACE KS AND
                    13  *       VMOS IN ABOVE EQUATION.
                    14  *                                                                   !
                    15  *-------------------------------------------------------------------+
                    16  *       VERSION : 1
                    17  *       PROGRAMMED BY : M.CONSTANTINIDES
                    18  *       CHECKED BY : M.CONSTANTINIDES
                    19
                    20
                    21          INTERN  CALKF
                    22  * REFERD MODULES:
                    23          EXTERN  SDIV05
                    24
                    25  * GLOBAL AREA:
                    26  *       (RAM)
                    27          EXTERN  KSBUF           KS-BUFF
                    28          EXTERN  KRBUF           KR-BUFF
                    29          EXTERN  VMOSBF          VMOS-BUFF
                    30          EXTERN  VMORBF          VMOR-BUFF
                    31  *       (ROM)
                    32          EXTERN  CAL1FT,CAL2FT           VIC*GI TABLES
                    33
                    34
=0000               35  KFCT    EQU     R0      =       SCRATCH
=0001               36  MPL     EQU     R1      =       SCRATCH
=0002               37  VGP     EQU     R2      =       POINTER TO VIC*GI TABLE (INPUT)
=0003               38  VMCS    EQU     R3      =       1ST CAL READING, VMOS (2'S-C) (INPUT)
=0004               39  VMCR    EQU     R4      =       2ND CAL READING, VMOR (2'S-C) (INPUT)
=0006               40  CAQ     EQU     R6      =       CAL-SLOT-# (INPUT; OUTPUT)
                    41
                    42          INCLUDE ENCLOS
                    44  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    46          INCLUDE REGDEF          REGISTER DEFENITIONS
                    65          INCLUDE CNSTNT          CONSTANTS
                   193          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   494          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   724          INCLUDE JMPMAC          JUMP MACROS
                   758          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   773          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000              1149          RSECT   CALKF
                   1150  ********************************************************************************
```

```
001E' 045B        1159                  RT
                  1160  ********************************************************************************
                  1161  * PRIVATE DATA AND CONSTANT
                  1162  *
                  1163  *       CRU ADDR
      =0630       1164  BTCA1   EQU     >318*2
      =0640       1165  BTCA2   EQU     >320*2
      =0660       1166  BTCA3   EQU     >330*2
      =0690       1167  BTCA4   EQU     >348*2
                  1168  *
                  1169+         PRVDAT
                  1171  *       MUX VALUES
0000' 00          1172  EXT     BYTE    0               START WITH EVEN
0001' 90          1173  BTVL1   BYTE    >90             FOR CRU 318-31F HEX
0002' 0000        1174  BTVL2   DATA    >0000           FOR CRU 320-32F HEX
0004' 11          1175  BTVL4   BYTE    >11             FOR CRU 348-34F HEX
                  1176          EVEN
                  1177  *
                  1178  *       MUX VALUES FOR CRU 330-337 HEX
0006' 20          1179  BTVL3   BYTE    >20             GND, REF
0007' 01          1180          BYTE    >01             +12V, +5V, -15V
0008' 02          1181          BYTE    >02             +28V, ACCEL,
0009' 04          1182          BYTE    >04             +5V, POT
000A' 08          1183          BYTE    >08             +15V, -5V
000B' 10          1184          BYTE    >10             +33V, -28V
                  1185
                  1186          END
```

```
                        1          IDT     BTLMT
                        2
                        3          SUBTTL  CHECK BITE LIMITS
                        4
                        5  *       CALLING SEQ:    CALL    @BTLMT
                        6
                        7  *-------------------------------------------------------------+
                        8  *                                                             !
                        9  *       BTLMT TEST THE LIMITS ON A RECEIVED BITE DATA AND IF IT IS
                       10  *       NOT IN THE GIVEN LIMIT, SETS ERROR STATUS WORD.
                       11  *                                                             !
                       12  *-------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY : N.CONSTANTINIDES
                       16  *       MODIFIED: 03-JAN-84 (N.CONSTANTINIDES)
                       17
                       18
                       19          INTERN  BTLMT
                       20  *** MODULES REFERENCED
                       21          EXTERN  SYSER,SYSOK
                       22
                       23  *       EQU     R0      =       BT ADDR PTR (INPUT) (1 TO 6)
                       24  *       EQU     R1      =       SCRATCH
        =0004          25  BTD     EQU     R4      =       BT-DATA (INPUT)
                       26
                       27          INCLUDE ENCLOS
                       29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      709          INCLUDE JMPMAC          JUMP MACROS
                      743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1134          RSECT   BTLMT
                     1135  ********************************************************************************
0000' C80B 0000'     1136  BTLMT   MOV     LINK,@LINKZ
0004' C040           1137          MOV     R0,R1                   R0 FOR ERROR CODE
0006' 0220 0102      1138          AI      R0,>102                 ADD WORD NO.AND BITE ERROR CODE BASE
                     1139+         INDEX   R1,                     PTR1 = BT-ADDR-PTR * 2
000A' 0A11           1145A           SLA   R1,1
000C' 0641           1148          DECT    R1                      LESS 2 (0 TO 10 RANGE)
                     1149+         DOIF    BTD,LEQ,@BTLLMT(R1),,,  IF DATA < LOWER LIMIT,
000E' 8844 0000'     1287B           C     BTD,@BTLLMT(R1)
0012' 1501           1328E            JGT  91$
                     1455+         ORIF    BTD,GE,@BTULMT(R1),,,,  OR DATA > UPPER LIMIT,
0014' 1003           1716E            JMP  82$
0016'                1846E  91$
0016' 8844 000C'     1960B           C     BTD,@BTULMT(R1)
001A' 1104           2002E            JLT  92$
001C'                2152E  82$
```

```
                    2233  *** ANALOG BITE FAILED
001C' C040          2234        MOV   R0,R1                 ERROR CODE >103 TO >108
001E' 06A0 0001*    2235        BL    @SYSER
0022' 1003          2236        JMP   100$
                    2237+       ENDBLK                      ENDIF
0024'               2373E 92$
                    2457  *** ANALOG BITE PASSED
0024' C040          2458        MOV   R0,R1                 ERROR CODE >103 TO >108
0026' 06A0 0002*    2459        BL    @SYSOK
002A' C2E0 0000'    2460  100$  MOV   @LINKZ,LINK
002E' 045B          2461        RT
                    2462  ********************************************************************************
                    2463  * PRICVATE AREA
                    2464
                    2465+       PRVDAT
                    2467  *     BITE LIMITS     (THE LIMITS GIVEN HERE ARE THE LOWER OR UPPER
                    2468  *                      LIMITES - 1 OR + 1, TO RESUALT IN A OUT OF
                    2469  *                      LIMIT VALUE)
                    2470  *     LOWER LIMIT
0000'               2471  BTLLMT
0000' F41B          2472        DATA  >F41B         +12V, +5V, -15V       BITE 1
0002' E             2473        DATA  >EE8F         +28V, ACCEL           BITE 2
0004' ECCD          2474        DATA  >ECCD         +5V, POT,             BITE 3
0006' F3F0          2475        DATA  >F3F0         +15V, -5V             BITE 4
0008' EE8F          2476        DATA  >EE8F         +33V, -28V            BITE 5
000A' FCCC          2477        DATA  >FCCC         GND, REF (-1.0V)      BITE 6
                    2478  *           UPPER LIMIT
000C'               2479  BTULMT
000C' F950          2480        DATA  >F950         +12V, +5V, -15V
000E' F191          2481        DATA  >F191         +28V, ACCEL
0010' F333          2482        DATA  >F333         +5V, POT,
0012' F979          2483        DATA  >F979         +15V, -5V
0014' 0000          2484        DATA  >0000         +33V, -28V
0016' 0334          2485        DATA  >0334         GND, REF (+1.0V)
                    2486  ************************************************
                    2487  * PRIVATE RAM
                    2488+       LOCR  PRIV,LINKZ
0000' =0002         2491A LINKZ BSS   2
                    2492  *
                    2493        END
```

```
                    1           IDT     CALAQ
                    2
                    3           SUBTTL  CALIBRATION DATA ACQUISITION
                    4
                    5   *       CALLING SEQ:    CALLWP  @CALAQV
                    6
                    7   *-------------------------------------------------------------+
                    8   *                                                             !
                    9   *       CALAQ ACQUIRES CALIBRARTIN DATA (GND AND VOLTAGE) IN THE
                   10   *       CYCLE 5, 6, 7 AND 8.
                   11   *       IT SENDS CAL MUX ADDR AND RECEIVES GND- AND V-CAL VALUES,
                   12   *       CONVERTS THEM INTO 2S-COMP, SAVES THE GND-CAL VALUES INTO
                   13   *       THE VMOS- VMOR-BUFF, CALCULATES THE CAL-FACTORS KS AND KR
                   14   *       AND SAVES THEM INTO KS- AND KR-BUFF.
                   15   *                                                             !
                   16   *-------------------------------------------------------------+
                   17   *       VERSION : 2
                   18   *       PROGRAMMED BY : N.CONSTANTINIDES
                   19   *       CHECKED BY : N.CONSTANTINIDES
                   20
                   21           INTERN  CALAQ
                   22   * REFERD MODULES:
                   23           EXTERN  ANDRQ
                   24           EXTERN  CALRT
                   25           EXTERN  CALKF
                   26   * GLOBAL AREA:
                   27   *       (RAM)
                   28           EXTERN  VMOSBF          VMOS-BUFF
                   29           EXTERN  VMORBF          VMOR-BUFF
                   30           EXTERN  CYPFRC          CYCLE PER FRAME COUNTER (0 - 31)
                   31           EXTERN  ANPRQ           AN-PARAM-QUANT
                   32           EXTERN  CALMAQ          CAL-MUX-ADDR-QUANT
                   33           EXTERN  DASFLG          DAS-FLAG
                   34           EXTERN  POCALF          POWER-ON CALIB FLAG
                   35   *       (ROM)
                   36           EXTERN  CALMAT          CAL-MUX-ADDR TABLE
                   37           EXTERN  D5              = 5
                   38
                   39   *               R0      =       SCRATCH
     =0001         40   CNT     EQU     R1      =       SCRATCH
     =0002         41   TYP     EQU     R2      =       AN-TYPE
     =0003         42   VG1     EQU     R3      =       1ST READING
     =0004         43   VG2     EQU     R4      =       2ND READING
                   44   *               FOLLOWING REG REMAIN THE SAME VARIALBLE
     =0006         45   CALAP   EQU     R6      =       CAL-ADDR-QUANTITY
     =0007         46   CALSN   EQU     R7      =       CAL-SLOT-#
     =0008         47   CNVFLG  EQU     R8      =       CONV-FLAG
                   48
                   49           INCLUDE ENCLOS
                   51   ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   53           INCLUDE REGDEF          REGISTER DEFENITIONS
                   72           INCLUDE CNSTNT          CONSTANTS
```

```
                    200         INCLUDE SUBMAC        FUNCTIONAL MACROS
                    501         INCLUDE MSCMAC        MISCELLANEOUS MACROS
                    731         INCLUDE JMPMAC        JUMP MACROS
                    765         INCLUDE BLKMAC        OTHER MACROS (BY D. SCOTT)
                    780         INCLUDE LBLMAC        HANDLES MACROS AUTOMATICALLY
          =0000    1156         RSECT  CALAQ
```

```
                    1158  ********************************************************************************
0000'               1159  CALAQ
0000'  0288 0004    1160          CI      CNVFLG,C4               0 TO 4 VALID
0004'  1B37         1161          JH      100$                    EXIT IF JUMP OFFSET INVALID
                    1162  *                                       DO ACCORDING TO CONV-FLAG
                    1163+       DOCASE    CNVFLG,50$,51$,52$
0006'  0468 000A'   1164A           B     @234$(CNVFLG)
000A'               1166A 234$
000A'  1002         1169B             JMP   50$
000C'  1013         1172C             JMP   51$
000E'  1020         1175D             JMP   52$
                    1184  *** START GND-CAL
0010'               1185  50$
0010'  05C8         1186          INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 2
                    1187+         DOIF    CALSN,LEQ,,,            IF CAL-SLOT-# = 0,
0012'  C1C7         1321B           MOV   CALSN,CALSN
0014'  1506         1366E             JGT   91$
0016'  C1E0 0007*   1493            MOV   @ANPRQ,CALSN              CAL-SLOT-# = AN-PARAM-QUANT
001A'  C1A0 0008*   1494            MOV   @CALMAQ,CALAP             CAL-ADR-QUN = CAL-MUX-ADR-QUANT
001E'  0720 000A*   1495            SETO  @POCALF                   END OF POWER ON CAL TEST
                    1496+         ENDBLK                          ENDIF
0022'               1628E 91$
0022'  0607         1716          DEC     CALSN                   CAL-SLOT-# = CAL-SLOT-# - 2
0024'  0606         1717          DEC     CALAP                   CAL-ADR-QUNT = CAL-ADR-QUNT - 1
                    1718  *                                       GET CAL-MUX-ADDR TBL ADR & OFFSET
                    1719+         CALL    ANDRQ,<CNT,=,CALMAT,I>,<R9,=,CALAP>   AND SEND MUX ADDR FOR GND
0026'  0201 000B*   1729C           LI    CNT,CALMAT
002A'  C246         1751D           MOV   CALAP,R9
002C'  06A0 0001*   1760A           BL    @ANDRQ
0030'  0606         1762          DEC     CALAP
0032'  101E         1763          JMP     60$
                    1764  *** READ GND-CAL AND START V-CAL
0034'               1765  51$
0034'  05C8         1766          INCT    CNVFLG                  CONV-FLAG = CONV-FLAG + 2
0036'  04C1         1767          CLR     CNT                     OFFSET = 0 (GND CAL)
                    1768+         CALL    CALRT                   READ AND TEST RECEIVED DATA
0038'  06A0 0002*   1775A           BL    @CALRT
003C'  C983 0004*   1777          MOV     VG1,@VMOSBF(CALAP)      SAVE 1ST READING INTO VMOS-BUFF
0040'  C984 0005*   1778          MOV     VG2,@VMORBF(CALAP)      SAVE 2ND READING INTO VMRO-BUFF
                    1779  *                                       GET CAL-MUX-ADDR TABLE AND
                    1780+         CALL    ANDRQ,<CNT,=,CALMAT,I>,<R9,=,CALAP>   SEND MUX ADDR FOR VOLTAGE
0044'  0201 000B*   1790C           LI    CNT,CALMAT
0048'  C246         1812D           MOV   CALAP,R9
004A'  06A0 0001*   1821A           BL    @ANDRQ
004E'  1010         1823          JMP     60$
                    1824  *** READ V-CAL AND COMPUTE K FACTORS
0050'               1825  52$
                    1826  *                                 START V-CAL
0050'  04C8         1827          CLR     CNVFLG                  CONV-FLAG = 0,
0052'  C820 000A*   1828          MOV     @POCALF,@DASFLG         DURING POWER ON CAL, POCALF=1
0056'  0009*
                    1829  *                                       THEN SET TO -1.
```

```
                        1          IDT     ADCISR
                        2
                        3          SUBTTL  ADC DATA READY INTERRUPT (INT 6)
                        4
                        5  *       CALLING SEQ:    INTERRUP 6
                        6
                        7  *-------------------------------------------------------------------+
                        8  *                                                                   |
                        9  *       ADCISR ACCORDING TO A GIVEN FLAG (DAS-FLAG) BRANCHES TO
                       10  *       ANALOG, CALIBRATION OR BITE DATA ACQUISITION.
                       11  *                                                                   |
                       12  *-------------------------------------------------------------------+
                       13  *       VERSION : 2
                       14  *       PROGRAMMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY : N.CONSTANTINIDES
                       16
                       17
                       18          INTERN  ADCISR
                       19  * REFERD MODULES:
                       20  *               ANACQ
                       21  *               CALAQ
                       22  *               .....
                       23  * GLOBAL (ROMV)
                       24          EXTERN  ADCBDV          AN, CAL, BT DATA ACQ VECTORS ADDR
                       25          EXTERN  RTSIN1
                       26
       =0009           27  DASFLG  EQU     R9      =       DAS-FLAG (INPUT & OUTPUT)
                       28          INCLUDE ENCLOS
                       30  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       32          INCLUDE REGDEF          REGISTER DEFENITIONS
                       51          INCLUDE CNSTNT          CONSTANTS
                      179          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      480          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      710          INCLUDE JMPMAC          JUMP MACROS
                      744          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      759          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
       =0000         1135          RSECT   ADCISR
                     1136  ********************************************************************************
0000'                1137  ADCISR
0000'  06A0 0002*    1138          BL      @RTSIN1
                     1139+         RESET   INT6,                   RESET ADC-INT
0004'  04E0 FFB8     1142A           CLR   @CADCLR
                     1150+         DOIF    DASFLG,GE,,,            IF DAS-FLAG >= 0 AND < 3,
0008'  C249          1284B             MOV  DASFLG,DASFLG
000A'  1108          1326E              JLT  91$
                     1456+         ANDIF   DASFLG,LEQ,C2,I,
000C'  0289 0002     1467B             CI    DASFLG,C2
0010'  1505          1527E              JGT  91$
0012'  C049          1653              MOV   DASFLG,R1               GET DAS-FLAG AS SUBR-TABLE OFFSET
0014'  0A21          1654              SLA   R1,C2
                     1655  *                                        DOCASE
                     1656  *                                           FOR DASFLG = 0, 1 AND 2  CALLS
```

```
                    1657  *                                        ANACQ, CALAQ AND BTACQ.
                    1658+           CALLWP  ADCBDV(R1)            ANACQV,CALAQV,& BTAQV
0016'  0421 0001*   1660A           BLWP  @ADCBDV(R1)
                    1662  *                                          DOEND
                    1663+         ELSEDO                          ELSE
001A'  1000         1924E             JMP  92$
001C'               2054E  91$
                    2139+         ENDBLK                          IFEND
001C'               2275E  92$
001C'  0380         2359          RTWP
                    2360  ********************************************************************************
                    2361
                    2362
                    2363          END
```

```
                     1          IDT     ANACQ
                     2          SUBTTL  ACQ AND STORE ANALOG DATA
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: ANACQ.SRC                        AUTH: N.CONSTANTINIDES  *
                     6  * VERSION: 2                             DATE: 24-OCT-1981       *
                     7  *                                                                  *
                     8  * FUNCTION: READS THE REQUESTED ANALOG DATA, CALCULATES THE       *
                     9  *           TRANSDUCER VALUE, PUTS IT INTO DESTINATION BUFFER     *
                    10  *           AND THEN SENDS ANALOG MUX ADDRESS FOR NEXT ADC        *
                    11  *           CONVERSION.                                           *
                    12  *                                                                  *
                    13  * CALLING MODULES: ADCISR                                         *
                    14  *                                                                  *
                    15  * CALLING SEQ: CALLWP @ANACQ                                      *
                    16  *                                                                  *
                    17  * INPUTS: R8    =       ANALOG PARAM COUNT                        *
                    18  *         R9    =       ANALOG TABLES OFFSET                      *
                    19  *        R10    =       ANALOG ARRAY OFFSET                       *
                    20  *                                                                  *
                    21  * OUTPUTS: R9   =       ANALOG TABLES OFFSET                      *
                    22  *         R10   =       ANALOG [illegible] OFFSET                 *
                    23  *                                                                  *
                    24  * MODULES REFERENCED: ANREAD,ANVT,ANSA,DSTINE                     *
                    25  *                                                                  *
                    26  * WORKSPACE AREA:                                                 *
                    27  *                                                                  *
                    28  * REGISTERS MODIFIED: R2,R3,R9,R10                                *
                    29  *                                                                  *
                    30  * VERSION HISTORY:                                                *
                    31  *                                                                  *
                    32  ********************************************************************
=0000               33          RSECT   ANACQ
                    34  *** CALL NAME
                    35          INTERN  ANACQ
                    36  *** VARIABLES REFERENCED
                    37          EXTERN  ANSFLG
                    38          EXTERN  CYFFRC,CNVFLG
                    39          EXTERN  DASFLG
                    40  *** CONSTANTS REFERENCED
                    41          EXTERN  D1
                    42          EXTERN  D2
                    43  *** TABLES REFERENCED
                    44          EXTERN  ANDT,ANDOA,ANFSTB
                    45  *** MODULES REFERENCED
                    46          EXTERN  ANREAD
                    47          EXTERN  ANFST
                    48          EXTERN  ANVT
                    49          EXTERN  ANSA
                    50          EXTERN  DSTINE
                    51          EXTERN  BTACQV
                    52          EXTERN  CALAQV
```

```
                     53  *** LIBRARY
                     54          INCLUDE ENCLOS
                     56  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     58          INCLUDE REGDEF          REGISTER DEFENITIONS
                     77          INCLUDE CNSTNT          CONSTANTS
                    205          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    506          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    736          INCLUDE JMPMAC          JUMP MACROS
                    770          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    785          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1161  *** REGISTERS DEFINITION
                   1162  *               R2      =       SCRATCH
                   1163  *               R3      =       SCRATCH
                   1164  *      THE FOLLOWING REG ARE SAME PARAM FOR THE WHOLE AN-DATA-ACQ
=0007              1165  ANOA    EQU     R7      =       ANALOG OFFSET ARRAY
=0008              1166  ANPCNT  EQU     R8      =       ANALOG PARAM COUNT (INPUT)
=0009              1167  ANTOF   EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
=000A              1168  ANARO   EQU    R10      =       ANALOG ARRAY OFFSET (INPUT; OUTPUT)
                   1169  *
```

```
                    1171  ********************************************************************************
0000' C020 0001*    1172  ANACQ   MOV     @ANSFLG,R0              AN SYNCRO OR AC DUMMY READING FLAG
0004' 1607          1173          JNE     40$                     JIF NOT SYNCHRO OR AC TYPE
0006' 0720 0001*    1174          SETO    @ANSFLG                 OTHERWISE IGNORE 1ST INTERRUPT
                    1175  *                                       (INVALID DATA)
                    1176+         RESET   INT6,                   RESET ADC INTERRUPT
000A' 04E0 FFB8     1179A           CLR   @CADCLR
                    1187+         STARTI  INT6,                   START ADC INTERRUPT
000E' 04E0 FFB0     1189A           CLR   @CADCDB
0012' 102F          1191          JMP     60$                     EXIT
                    1192+ 40$     CALL    ANREAD,                 READ (VMTS AND VMTR) AN-DATA
0014' 06A0 000A*    1199A           BL    @ANREAD
                    1201
                    1202+         CALL    ANVT                    GET XDUCER VALUE
0018' 06A0 000C*    1209A           BL    @ANVT
                    1211  *** POSITION AND ROUND IF REQUESTED
001C' 0202 0009*    1212          LI      R2,ANPSTB               AN-POS-TABLE
0020' 06A0 000B*    1213          BL      @ANPST                  POSITION & ROUND
                    1214  *                                       GET AN-DS-TABLE, AN-DS-OFFSET-ARRAY
                    1215  *                                       ADDR AND PUT DATA INTO DEST
                    1216+         CALL    DSTINE,<R2,=,ANDOA,I>,<R3,=,ANDT,I>
0024' 0202 0008*    1226C           LI    R2,ANDOA
0028' 0203 0007*    1242D           LI    R3,ANDT
002C' 06A0 000E*    1257A           BL    @DSTINE
                    1259
                    1260+         CALL    ANSA                    SEND AN ADDR
0030' 06A0 000D*    1267A           BL    @ANSA
0034' 151E          1269          JGT     60$             JIF STILL ACQUIRING ANALOG DATA
0036' 131D          1270          JEQ     60$
                    1271  ***
0038' C020 0002*    1272          MOV     @CYPFRC,R0              CYCLE PER FRAME COUNTER (0-31)
003C' 0240 0007     1273          ANDI    R0,C7                   EXTRACT LS 4 BITS
0040' 0280 0004     1274          CI      R0,C4                   CYCLE 4?
0044' 160A          1275          JNE     50$                     JIF NOT
0046' C020 0002*    1276          MOV     @CYPFRC,R0              RESTORE R0
                    1277
                    1278  *** ANALOG BITE VOLTAG READINGS ONCE PER SEC
                    1279
004A' C820 0006*    1280          MOV     @D2,@DASFLG             SELECT NEXT BITE INTERRUPT (ISR) VIA ADCISR
004E' 0004*
0050' 04E0 0003*    1281          CLR     @CNVFLG                 JUMP OFFSET IN BTACQ
0054' 0420 000F*    1282          BLWP    @BTACQV                 INITIATE FIRST BITE
0058' 100C          1283          JMP     60$
                    1284  ***
005A' 0240 0007     1285  50$     ANDI    R0,C7                   SAVE SUBFRAME CYCLE COUNT ONLY
005E' 0280 0003     1286          CI      R0,C3                   5TH THRU 8TH CYCLE OF EVERY S/F?
0062' 1607          1287          JNE     60$                     JIF NOT
                    1288  *** ANALOG CAL DATA READINGS, 4 TIMES PER SEC
0064' C820 0005*    1289          MOV     @D1,@DASFLG             SELECT NEXT CAL INTERRUPT (ISR) VIA ADCISR
0068' 0004*
006A' 04E0 0003*    1290          CLR     @CNVFLG                 JUMP OFFSET IN CALAQ
006E' 0420 0010*    1291          BLWP    @CALAQV                 INITIATE FIRST CAL DATA
```

```
0072'  0380          1292   60$     RTWP
                     1293   $
                     1294           END
```

```
                        1          IDT     ANDRQ
                        2
                        3          SUBTTL  REQUEST ANALOG DATA (SEND-MUX-ADDR)
                        4
                        5  *       CALLING SEQ:    CALL    @ANDRQ
                        6
                        7  *-------------------------------------------------------------------+
                        8  *                                                                   |
                        9  *       ANDRQ GETS THE POINTER TO THE MUX-ADDR-TABLE AND THEN
                       10  *       SENDS (EIGHT BYTES) MUX ADDRESSES   TO  RECEIVE  NEXT
                       11  *       ANALOG DATA.
                       12  *
                       13  *       CALLING MODULE : ANSA                                       |
                       14  *-------------------------------------------------------------------+
                       15  *       VERSION : 1
                       16  *       PROGRAMMED BY : N.CONSTANTINIDES
                       17  *       CHECKED BY : N.CONSTANTINIDES
                       18
                       19
                       20          INTERN  ANDRQ
                       21
                       22
        =0000          23  MXPT    EQU     R0      =       SCRATCH
        =0001          24  MTBL    EQU     R1      =       MUX-ADDR-TBL (INPUT)
        =0009          25  APT     EQU     R9      =       MUX-OFFSET (INPUT)
                       26
                       27          INCLUDE ENCLOS
                       29  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      709          INCLUDE JMPMAC          JUMP MACROS
                      743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1134          RSECT   ANDRQ
                     1135  ********************************************************************************
    0000'            1136  ANDRQ
                     1137  *** CLEAR MUX ADDRESS FIELD ONLY
    0000'  04C0      1138          CLR     MXPT
                     1139+         CRUWRT  ANMCA1,MXPT,ZERO        CLEAR 300-30F
    0002'  020C 0600 1141A           LI     CRU,ANMCA1
    0006'  3000      1142A           LDCR   MXPT,ZERO
                     1144+         CRUWRT  ANMCA2,MXPT,C8
    0008'  020C 0620 1146A           LI     CRU,ANMCA2
    000C'  3200      1147A           LDCR   MXPT,C8
    000E'  C009      1149          MOV     APT,MXPT        SAVE OFFSET
    0010'  0A30      1150          SLA     MXPT,C3         OFFSET = OFFSET * 8 TO GET CORRECT MUX OFFSET (8 BYTES PER MUX VALUE
)
    0012'  A001      1151          A       MTBL,MXPT       GET ADDR OF MUX
                     1152
                     1153+         CRUWRT  ANMCA1,*MXPT+,ZERO      SET 300-30F
    0014'  020C 0600 1155A           LI     CRU,ANMCA1
```

REQUEST ANALOG DATA (SEND-MUX-ADDR) ANDRQ.SRC

```
0018' 3030          1156A           LDCR  *MXPT+,ZERO
                    1158+         CRUWRT  ANMCA2,*MXPT+,ZERO      SET 310-31F
001A' 020C 0620     1160A           LI    CRU,ANMCA2
001E' 3030          1161A           LDCR  *MXPT+,ZERO
                    1163+         CRUWRT  ANMCA3,*MXPT+,ZERO      SET 320-32F
0020' 020C 0640     1165A           LI    CRU,ANMCA3
0024' 3030          1166A           LDCR  *MXPT+,ZERO
                    1168+         CRUWRT  ANMCA4,*MXPT+,C8        SET 330-337
0026' 020C 0660     1170A           LI    CRU,ANMCA4
002A' 3230          1171A           LDCR  *MXPT+,C8
                    1173+         CRUWRT  ANMCA5,*MXPT+,C8        SET 348-34F
002C' 020C 0690     1175A           LI    CRU,ANMCA5
0030' 3230          1176A           LDCR  *MXPT+,C8
0032' 045B          1178            RT
                    1179  ********************************************************************************
                    1180  *PRIVATE CONSTANTS
          =0600     1181  ANMCA1  EQU     >300*2  AN MUX CRU ADDR FOR 1ST+2ND BYTES (300-30F H)
          =0620     1182  ANMCA2  EQU     >310*2  AN MUX CRU ADDR FOR 3RD+4TH BYTES (310-31F H)
          =0640     1183  ANMCA3  EQU     >320*2  AN MUX CRU ADDR FOR 5TH+6TH BYTES (320-32F H)
          =0660     1184  ANMCA4  EQU     >330*2  AN MUX CRU ADDR FOR 7TH BYTE (330-337 H)
          =0690     1185  ANMCA5  EQU     >348*2  AN MUX CRU ADDR FOR 8TH BYTE (348-34F H)
                    1186
                    1187          END
```

```
                        1          IDT     ANPST
                        2          SUBTTL  ANALOG POSITION MODULE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: ANPST.SRC                      AUTH: N.COSTANTINIDES      *
                        6  * VERSION: 1                           DATE: 4-JUN-1982           *
                        7  *                                                                  *
                        8  * FUNCTION: RIGHT JUSTIFIES ANALOG DATA AND ROUNDS                 *
                        9  *          OFF LSB IF SELECTED.                                    *
                       10  *                                                                  *
                       11  * CALLING MODULES: ANACQ                                          *
                       12  *                                                                  *
                       13  * CALLING SEQ: BL @ANPST                                          *
                       14  *                                                                  *
                       15  * INPUTS: R2    =       ANL-POS-TBL START ADDRESS                 *
                       16  *         R4    =       ANALOG INPUT DATA                         *
                       17  *         R9    =       DATA ITEM IN ANL-TBL-OFFSET-ARR-TBL       *
                       18  *                                                                  *
                       19  * OUTPUTS:      R4=RIGHT JUSTIFIED DATA ROUNDED                    *
                       20  *                                                                  *
                       21  * MODULES REFERENCED: NONE                                        *
                       22  *                                                                  *
                       23  * WORKSPACE AREA: CALLER'S                                        *
                       24  *                                                                  *
                       25  * REGISTERS MODIFIED: R0,R2,R4                                    *
                       26  *                                                                  *
                       27  * VERSION HISTORY:                                                *
                       28  *                                                                  *
                       29  ********************************************************************
          =0000        30          RSECT   ANPST
                       31  *** CALL NAME
                       32          INTERN  ANPST
                       33  *** VARIABLES REFERENCED
                       34  *** CONSTANTS REFERENCED
                       35  *** TABLES REFERENCED
                       36  *** MODULES REFERENCED
                       37  *** LIBRARY
                       38  *** REGISTERS DEFINITION
                       39  ********************************************************************
0000'                  40  ANPST
0000'  A089           41          A       R9,R2           =OFFSET+POS-TBL START ADDRESS
                       42  ***  RIGHT JUSTIFY 12 BIT INPUT DATA TO?
0002'  D012           43          MOVB    *R2,R0          DATA SET IN ANL-POS-TBL
0004'  06C0           44          SWPB    R0              R.J. DATA SET
0006'  0240 0003      45          ANDI    R0,3            SAVE 2 BITS OF INPUT POSITION
000A'  1301            46          JEQ     10$             JIF NO ADJUSTMENT NEEDED
000C'  0904            47          SRL     R4,0            ELSE R.J. INPUT PER INPUT POSITION
                       48  ***  ROUNDING (OPTIONAL)
000E'  D012            49  10$     MOVB    *R2,R0          DATA SET IN AN-POS-TBL
0010'  1503            50          JGT     20$             JIF NO ROUNDING
0012'  1302            51          JEQ     20$
0014'  0584            52          INC     R4              ELSE BUMP LSB
```

```
0016'  0914          53          SRL     R4,1            GET RID OF LSB
                     54  ***
0018'  045B          55  20$     RT
                     56  ************************************************************
                     57          END
```

No errors detected

```
                      1          IDT     ANREAD
                      2
                      3          SUBTTL  READ ANALOG DATA
                      4
                      5  *       CALLING SEQ:    CALL    @ANREAD
                      6
                      7  *-------------------------------------------------------------------+
                      8  *                                                                   !
                      9  *       ANREAD READS THE FIRST RECEIVED ANALOG 14-BIT IOB DATA,
                     10  *       GETS ANALOG DATA TYPE, IF DOUBLE CONVERSION (ACCORDING
                     11  *       TO THE TYPE), READS THE SECOND ANALOG DATA AND RETURNS.
                     12  *                                                                   !
                     13  *-------------------------------------------------------------------+
                     14  *       VERSION : 1
                     15  *       PROGRAMMED BY : M.CONSTANTINIDES
                     16  *       CHECKED BY : M.CONSTANTINIDES
                     17
                     18          INTERN  ANREAD
                     19  * REFERD MODULES:
                     20          EXTERN  IOB2SC
                     21  * GLOBAL AREA:
                     22  *       (ROMD)
                     23          EXTERN  ANTYPT          AN-TYPE-TABLE
                     24
        =0002        25  ATYP    EQU     R2      =       ANALOG DATA TYPE (OUTPUT)
        =0003        26  FAND    EQU     R3      =       1ST ANALOG DATA (OUTPUT)
        =0004        27  SAND    EQU     R4      =       2ND ANALOG DATA (IF NOT DUAL, = 0) (OUTPUT)
        =0009        28  ATO     EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
                     34          INCLUDE REGDEF          REGISTER DEFENITIONS
                     53          INCLUDE CNSTNT          CONSTANTS
                    161          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    482          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    712          INCLUDE JMPMAC          JUMP MACROS
                    746          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    761          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000       1138          RSECT   ANREAD
                    1139  ********************************************************************
   0000'            1140  ANREAD
   0000'  C80B 0000'1141          MOV     LINK,@LINKZ             SAVE LINKER
                    1142  *                                       GET RECEIVED 14-BIT AN-DATA
                    1143+         CALL    IOB2SC,<R4,=,@CADCDB>    AND CONVET TO 12-BIT 2S COMP
   0004'  C120 FFB0 1159C           MOV     @CADCDB,R4
   0008'  06A0 0001* 1167A          BL      @IOB2SC
   000C'  C0C4      1169          MOV     SAND,FAND               1ST-READING =  ANALOG DATA
   000E'  04C4      1170          CLR     SAND                    2ND-READING = 0.
                    1171
                    1172+         MOVBRJ  @ANTYPT(ATO),ATYP,      GET THE TYPE RIGHT JUSTIF
   0010'  D0A9 0002* 1174A          MOVB    @ANTYPT(ATO),ATYP
   0014'  0982      1175A          SRL     ATYP,8
                    1177
                    1178+         DOIF    ATYP,GE,5,X,,           IF TYPE >= 5, DUAL CONVERSION
   0016'  0282 0005 1297B           CI      ATYP,5
```

```
001A' 1104          1354E              JLT   91$
                    1484  *                                          READ 14-BIT AN-DATA AS 12-BIT
                    1485+               CALL  IOB2SC,<R4,=,@CADCDB>  2S-COMP (2ND READING)
001C' C120 FFB0     1501C               MOV   @CADCDB,R4
0020' 06A0 0001$    1509A               BL    @IOB2SC
                    1511+              ENDBLK                        IFEND
0024'               1643E  91$
0024' C2E0 0000'    1731               MOV    @LINKZ,LINK           RESTORE LINKER
0028' 045B          1732               RT
                    1733  ********************************************************************************
                    1734  * PRICVATE AREA
                    1735+              LOCR   PRIV,LINKZ            LINK-SAVE AREA
0000' =0002         1738A LINKZ        BSS    2
                    1739
                    1740               END
```

```
                     1          IDT     ANSA
                     2
                     3          SUBTTL  SEND NEXT ANALOG ADDRESS
                     4
                     5  *       CALLING SEQ:    CALL    @ANSA
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       IF THERE ARE MORE ANLOG DATA TO ACQUIRE, ANSA GETS THE NEXT
                    10  *       AN-TABLES-OFFSET AND SEND MUX ADDRESSES TO  RECEIVE NEXT
                    11  *       DATA, ELSE IT SETS FLAG FOR NO MORE AND RETURNS.
                    12  *                                                                   !
                    13  *       CALLED BY RTIDA AND ANACQ
                    14  *-------------------------------------------------------------------+
                    15  *       VERSION : 2
                    16  *       PROGRAMMED BY : N.CONSTANTINIDES
                    17  *       CHECKED BY : N.CONSTANTINIDES
                    18
                    19          INTERN  ANSA
                    20  * REFERD MODULES:
                    21          EXTERN  ANDRQ
                    22          EXTERN  TBLOFS
                    23  * GLOBAL AREA:
                    24          EXTERN  ANMUXT          AN-MUX-ADDR-TABLE (ROM)
                    25          EXTERN  DASFLG          DAS-FLAG (RAM)
                    26          EXTERN  ANSFLG          AN SYNCHRO FLAG
                    27
                    28  *               R0      =       SCRATCH
                    29  *               R1      =       SCRATCH
=0007               30  ANOA    EQU     R7      =       AN-OFFSET-ARRAY (INPUT; OUTPUT)
=0008               31  ANPCNT  EQU     R8      =       ANALOG PARAM COUNT (INPUT)
=0009               32  ANTOF   EQU     R9      =       ANALOG TABLES OFFSET (INPUT; OUTPUT)
=000A               33  ANARQ   EQU     R10     =       ANALOG OFFSET POINTER (INPUT; OUTPUT)
                    34          INCLUDE ENCLOS
                    36  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    38          INCLUDE REGDEF          REGISTER DEFENITIONS
                    57          INCLUDE CNSTNT          CONSTANTS
                   185          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   486          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   716          INCLUDE JMPMAC          JUMP MACROS
                   750          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   765          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1141          RSECT   ANSA
                  1142  ********************************************************************************
```

```
0000'                1144  ANSA
0000'  C0CB          1145        MOV   LINK,R3        SAVE LINKER
                     1146  *                               R2 = AN-VERS TABLE AND
                     1147  *                               FIND NEXT AN-TABLES-OFFSET
                     1148+       CALL  TBLOFS
0002'  06A0 0002*    1155A         BL  @TBLOFS
                     1157
                     1158+       DOIF  ANPCNT,GE,,,        IF AN-PARAM-COUNT >= 0 (MORE TO GET),
0006'  C208          1292B          MOV ANPCNT,ANPCNT
0008'  110A          1334E           JLT  91$
                     1464+         CALL ANDRQ,<R1,=,ANMUXT,I>    R1 = MUX-TBL ADDR AND SEND AN ADDR
000A'  0201 0003*    1474C         LI   R1,ANMUXT
000E'  06A0 0001*    1488A         BL   @ANDRQ
0012'  04E0 0005*    1490        CLR   @ANSFLG       SET FLAG TO READ ANALOG DATA TWICE IN ANACQ
                     1491+       RESET INT6,               RESET INTERRUPT
0016'  04E0 FFB8     1494A         CLR  @CADCLR
                     1502+       STARTI INT6,              START ADC INTERRUPT
001A'  04E0 FFB0     1504A         CLR  @CADCDB
                     1506+         ENDBLK
001E'                1638E  91$
001E'  C2C3          1726        MOV   R3,LINK             RESTORE LINKER
0020'  C208          1727        MOV   ANPCNT,ANPCNT       FOR CALLER TO CHECK END OF ACQ FOR THIS CYCLE
0022'  045B          1728        RT
                     1729  ********************************************************************************
                     1730        END
```

```
                        1          IDT     ANP5VR
                        2
                        3          SUBTTL  5V. REF. POTENTIOMETER
                        4
                        5  *       CALLING SEQ:    CALL    @ANP5VR
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                      !
                        9  *       ANP5VR SAVES THE RECEIVED 5V. REF. POT. VALUE FOR FUTURE USE
                       10  *       (BY "ANPOT").
                       11  *                                                                      !
                       12  *---------------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  ANP5VR
                       18  * GLOBALS:
                       19          EXTERN  VCPR            5V. POT. REF.
                       20
        =0003          21  VCTS    EQU     R3      =       AN 5V REF POT DATA (INPUT; OUTPUT)
                       22
                       28          INCLUDE REGDEF          REGISTER DEFENITIONS
                       47          INCLUDE CNSTNT          CONSTANTS
                      180          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      481          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      711          INCLUDE JMPMAC          JUMP MACROS
                      745          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      760          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1137          RSECT   ANP5VR
                     1138  ********************************************************************************
0000'               1139  ANP5VR
0002'  C803 0001*   1140          MOV     VCTS,@VCPR      SAVE 5V REF POT VALUE
0004'  045B         1141          RT
                     1142  ********************************************************************************
                     1143
                     1144          END
```

```
                        1          IDT     ANDSCL
                        2
                        3          SUBTTL  ANALOG WITH NO SCALING
                        4
                        5  *       CALLING SEQ:    CALL    @ANDSCL
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                     !
                        9  *       ANDSCL GETS THE ANALOG VALUE FOR THE ANALOG SIGNAL TYPES
                       10  *       WITH NO SCALING (LDC, ...).
                       11  *                                                                     !
                       12  *---------------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  ANDSCL
                       18
          =0003        19  VTI     EQU     R3      =       AN VALUE (INPUT)
          =0004        20  VTO     EQU     R4      =       AN VALUE (OUTPUT)
                       21
                       27          INCLUDE REGDEF          REGISTER DEFENITIONS
                       46          INCLUDE CNSTNT          CONSTANTS
                      179          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      480          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      710          INCLUDE JMPMAC          JUMP MACROS
                      744          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      759          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000      1136          RSECT   ANDSCL
                     1137  ********************************************************************
    0000'            1138  ANDSCL
    0000' C103       1139          MOV     VTI,VTO         NO SCALING, R4 = ANSW
    0002' 045B       1140          RT
                     1141  ********************************************************************
                     1142
                     1143          END
```

```
                        1          IDT     ANPOT
                        2
                        3          SUBTTL  POTENTIOMETER
                        4
                        5  *       CALLING SEQ:    CALL    @ANPOT
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                     !
                        9  *       ANPOT CALCULATES VT = (VCTS / VCPR) * 4096.
                       10  *       TO AVOID LOSS OF DATA IT PREFORMS FIRST (VCTS * 4096).
                       11  *       VCPR : EXCITATION-VOLTAGE, VCTS : TRANCDUCER READING
                       12  *       AND VT : THE RATIO.
                       13  *                                                                     !
                       14  *---------------------------------------------------------------------+
                       15  *       VERSION : 1
                       16  *       PROGRAMED BY : N.CONSTANTINIDES
                       17  *       CHECKED BY : N.CONSTANTINIDES
                       18
                       19          INTERN  ANPOT
                       20  * GLOBALS:
                       21          EXTERN  VCPR            5V. POT. REF. (RAM) (INPUT)
                       22
        =0003          23  VCTP    EQU     R3      =       AN-DATA (INPUT; OUTPUT(=VT))
        =0004          24  TMP     EQU     R4      =       SCRATCH
                       25
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      183          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      484          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      714          INCLUDE JMPMAC          JUMP MACROS
                      748          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      763          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1140          RSECT   ANPOT
                     1141  ********************************************************************************
0000'                1142  ANPOT
                     1143+       DOIF    @VCPR,LEQ,VCTP          IF VCPR =< VCTS
0000' 80E0 0001*     1281B           C   @VCPR,VCTP
0004' 1503           1322E             JGT   91$
0006' 0203 0FFF      1449            LI      VCTP,C4095              VT = 4095
                     1450+       ELSEDO                          ELSE
0008' 1006           1711E             JMP   92$
000A'                1841E  91$
000A' C103           1926            MOV     VCTP,TMP
                     1927+           DOIF    ,NE,,                   IF VCTS NOT= 0,
000E' 1304           2102E             JEQ   93$
0010' 0843           2233              SRA   VCTP,C4
0012' 0AC4           2234              SLA   TMP,C12
0014' 3CE0 0001*     2235              DIV   @VCPR,VCTP                VT = VCTS / VCPR
                     2236+           ENDBLK                          IFEND
0018'                2376E  93$
                     2456+       ENDBLK                          IFEND
0018'                2592E  92$
```

```
0018' 045B          2676          RT
                    2677  ********************************************************************************
                    2678
                    2679          END
```

No errors detected

```
                    1          IDT     ANTBL
                    2          SUBTTL  ANALOG TABLES
          =0000     3          RSECT   ANTBL
                    4  ********************************************************************
                    5  *                                                                  *
                    6  *  NAME: ANTBL.SRC (BA)                AUTH: N. CONSTANTINIDES    *
                    7  *  VERSION:                            DATE: 21-JUN-1983          *
                    8  *                                                                  *
                    9  *  FUNCTION: ANALOG TABLES SECTION CONTAINS THE PARAMETER OFFSET ARRAY, *
                   10  *            PARAMETER QUANTITY PER CYCLE TABLE, XDUCER MUX VALUES,  *
                   11  *            PARAMETER MASK TABLE, PARAMETER TYPE, PARAMETER DEST. TABLE*
                   12  *            OFFSET, PARAMETER DESTINATION AND ANALOG DATA POSITION. *
                   13  *                                                                  *
                   14  *  REFERENCED BY:                                                  *
                   15  *                                                                  *
                   16  *  DATA POSITION FORMAT:                                           *
                   17  *  BIT         7  6  5  4  3  2  1  0                             *
                   18  *              N3 -- N2 N2 N2 N2 N1 N1                             *
                   19  *                                                                  *
                   20  *  WHERE:        N1 = INPUT POSITION                              *
                   21  *                N2 = INPUT MASK WORD INDEX                       *
                   22  *                N3 = ROUNDING (OPTIONAL)                         *
                   23  *                                                                  *
                   24  *  PARAMETER DESTINATION FORMAT:                                   *
                   25  *  BIT         15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0 *
                   26  *              N5  N4  N3  N3  N3  N3  N2 N2 N1 N1 N1 N1 N1 N1 N1 N1 *
                   27  *                                                                  *
                   28  *  WHERE:        N1 = BUFFER OFFSET WORD                          *
                   29  *                N2 = BUFFER TYPE                                 *
                   30  *                N3 = BUFFER POSITION LEFT SHIFT COUNT            *
                   31  *                N4 = CLEAR OUTPUT WORD (OPTIONAL)                *
                   32  *                N5 = END OF DATA SET (OPTIONAL)                  *
                   33  *                                                                  *
                   34  *  VERSION HISTORY:                                               *
                   35  *                                                                  *
                   36  ********************************************************************
                 1228  *
                 1229          INTERN  ANDFAR,ANPQT,ANMUXT,ANPRQ,ANMSKT,ANTYPT,ANDOA,ANDT
                 1230          INTERN  CALMAT,CALMAQ,CAL1FT,CAL2FT,CALSDT,CALLT
                 1231          INTERN  ANPSTB,CALULT,ANCIT,ANSYNT
                 1232  *
```

```
                    1234  ********************************************************************************
                    1235  *
                    1236  *               ANALOG TABLES
                    1237  *               *************
                    1238  *
                    1239  *       ANALOG PARAMETER OFFSET ARRAY
                    1240  *
0000'               1241  ANOFAR
                    1242  *
                    1243  *  SUBFRAME 1 CYCLES 1 TO 8
                    1244  *
0000'  00 06 04     1245  AA11    BYTE    0,6,4
0003'  00 01 07     1246  AA12    BYTE    0,1,7
0006'  00 05        1247  AA13    BYTE    0,5
0008'  00 01        1248  AA14    BYTE    0,1
000A'  00 06 03     1249  AA15    BYTE    0,6,3
000D'  00 01 07     1250  AA16    BYTE    0,1,7
0010'  00 02 05     1251  AA17    BYTE    0,2,5
0013'  00 01        1252  AA18    BYTE    0,1
                    1253  *
                    1254  *  SUBFRAME 2 CYCLES 1 TO 8
                    1255  *
0015'  00 06 04     1256  AA21    BYTE    0,6,4
0018'  00 01 07     1257  AA22    BYTE    0,1,7
001B'  00 05        1258  AA23    BYTE    0,5
001D'  00 01        1259  AA24    BYTE    0,1
001F'  00 06 03     1260  AA25    BYTE    0,6,3
0022'  00 01 07     1261  AA26    BYTE    0,1,7
0025'  00 02 05     1262  AA27    BYTE    0,2,5
0028'  00 01        1263  AA28    BYTE    0,1
                    1264  *
                    1265  *  SUBFRAME 3 CYCLES 1 TO 8
                    1266  *
002A'  00 06 04     1267  AA31    BYTE    0,6,4
002D'  00 01 07     1268  AA3     BYTE    0,1,7
0030'  00 05        1269  AA33    BYTE    0,5
0032'  00 01        1270  AA34    BYTE    0,1
0034'  00 06 03     1271  AA35    BYTE    0,6,3
0037'  00 01 07     1272  AA36    BYTE    0,1,7
003A'  00 02 05     1273  AA37    BYTE    0,2,5
003D'  00 01        1274  AA38    BYTE    0,1
                    1275  *
                    1276  *  SUBFRAME 3 CYCLES 1 TO 8
                    1277  *
003F'  00 06 04     1278  AA41    BYTE    0,6,4
0042'  00 01 07     1279  AA42    BYTE    0,1,7
0045'  00 05        1280  AA43    BYTE    0,5
0047'  00 01        1281  AA44    BYTE    0,1
0049'  00 06 03     1282  AA45    BYTE    0,6,3
004C'  00 01 07     1283  AA46    BYTE    0,1,7
004F'  00 02 05     1284  AA47    BYTE    0,2,5
0052'  00 01        1285  AA48    BYTE    0,1
```

```
                    1286  *
                    1287  ****************************************
                    1288  *
                    1289  *       ANALOG PARAMETER QUANTITY PER CYCLE
                    1290  *
0054'               1291  ANPQT
                    1292  *
                    1293  *  SUBFRAME 1 CYCLES 1 TO 8
                    1294  *
0054'  03           1295  AQ11    BYTE    3
0055'  03           1296  AQ12    BYTE    3
0056'  02           1297  AQ13    BYTE    2
0057'  02           1298  AQ14    BYTE    2
0058'  03           1299  AQ15    BYTE    3
0059'  03           1300  AQ16    BYTE    3
005A'  03           1301  AQ17    BYTE    3
005B'  02           1302  AQ18    BYTE    2
                    1303  *
                    1304  *  SUBFRAME 2 CYCLES 1 TO 8
                    1305  *
005C'  03           1306  AQ21    BYTE    3
005D'  03           1307  AQ22    BYTE    3
005E'  02           1308  AQ2[illegible]  BYTE    2
005F'  02           1309  AQ24    BYTE    2
0060'  03           1310  AQ25    BYTE    3
0061'  03           1311  AQ26    BYTE    3
0062'  03           1312  AQ27    BYTE    3
0063'  02           1313  AQ28    BYTE    2
                    1314  *
                    1315  *  SUBFRAME 3 CYCLES 1 TO 8
                    1316  *
0064'  03           1317  AQ31    BYTE    3
0065'  03           1318  AQ32    BYTE    3
0066'  02           1319  AQ33    BYTE    2
0067'  02           1320  AQ34    BYTE    2
0068'  03           1321  AQ35    BYTE    3
0069'  03           1322  AQ36    BYTE    3
006A'  03           1323  AQ37    BYTE    3
006B'  02           1324  AQ38    BYTE    2
                    1325  *
                    1326  *  SUBFRAME 4 CYCLES 1 TO 8
                    1327  *
006C'  03           1328  AQ41    BYTE    3
006D'  03           1329  AQ42    BYTE    3
006E'  02           1330  AQ43    BYTE    2
006F'  02           1331  AQ44    BYTE    2
0070'  03           1332  AQ45    BYTE    3
0071'  03           1333  AQ46    BYTE    3
0072'  03           1334  AQ47    BYTE    3
0073'  02           1335  AQ48    BYTE    2
                    1336  *
                    1337          EVEN
```

```
                    1338  *
                    1339  ********************************************
                    1340  *
                    1341  *  LIST OF SYNCHRO TYPE PARAMETERS TABLE
                    1342  *  CONTAINS OFFSET OF PARAMETER TABLE
                    1343  *
0074'  0004        1344  ANSYNT  DATA    4               NUMB OF SYNCHRO PARAMS
0076'  0009        1345          DATA    9
0078'  000A        1346          DATA    10
007A'  000C        1347          DATA    12
007C'  0013        1348          DATA    19
                    1349  *
                    1350  ********************************************
                    1351  *
                    1352  *  ANALOG CYCLE INDEX TABLE FOR ANARO(R10)
                    1353  *       USED TO GET OFFSET VALUE FROM ANOFAR TABLE
                    1354  *
007E'               1355  ANCIT
                    1356  *
                    1357  *       S/F 1 CYCLES 1 TO 8
                    1358  *
007E'  FF           1359          BYTE    -1
007F'  02           1360          BYTE    -1+3
0080'  05           1361          BYTE    -1+3+3
0081'  07           1362          BYTE    -1+3+3+2
0082'  09           1363          BYTE    -1+3+3+2+2
0083'  0C           1364          BYTE    -1+3+3+2+2+3
0084'  0F           1365          BYTE    -1+3+3+2+2+3+3
0085'  12           1366          BYTE    -1+3+3+2+2+3+3+3
                    1367  *
                    1368  *       S/F 2 CYCLES 1 TO 8
                    1369  *
0086'  14           1370          BYTE    20
0087'  17           1371          BYTE    20+3
0088'  1A           1372          BYTE    20+3+3
0089'  1C           1373          BYTE    20+3+3+2
008A'  1E           1374          BYTE    20+3+3+2+2
008B'  21           1375          BYTE    20+3+3+2+2+3
008C'  24           1376          BYTE    20+3+3+2+2+3+3
008D'  27           1377          BYTE    20+3+3+2+2+3+3+3
                    1378  *
                    1379  *       S/F 3 CYCLES 1 TO 8
                    1380  *
008E'  29           1381          BYTE    41
008F'  2C           1382          BYTE    41+3
0090'  2F           1383          BYTE    41+3+3
0091'  31           1384          BYTE    41+3+3+2
0092'  33           1385          BYTE    41+3+3+2+2
0093'  36           1386          BYTE    41+3+3+2+2+3
0094'  39           1387          BYTE    41+3+3+2+2+3+3
0095'  3C           1388          BYTE    41+3+3+2+2+3+3+3
                    1389  *
```

```
                    1390  *       S/F 4 CYCLES 1 TO 8
                    1391  *
 0096'  3E          1392          BYTE    62
 0097'  41          1393          BYTE    62+3
 0098'  44          1394          BYTE    62+3+3
 0099'  46          1395          BYTE    62+3+3+2
 009A'  48          1396          BYTE    62+3+3+2+2
 009B'  4B          1397          BYTE    62+3+3+2+2+3
 009C'  4E          1398          BYTE    62+3+3+2+2+3+3
 009D'  51          1399          BYTE    62+3+3+2+2+3+3+3
                    1400  *
                    1401          EVEN
                    1402  *
                    1403  ***+3*****************************************
                    1404  *
                    1405  *  +3XDUCER MUX VALUES
                    1406  *
 009E'              1407  ANMUXT
                    1408+         LLDC    XDUCER,3,1,           VERTICAL ACCELERATION
 009E'  1041        1429B                         DATA    )1041
 00A0'  8404        1530A                         DATA    )84*256+ZZVL
 00A2'  0004        1533A                         DATA    )0004
 00A4'  00          1534A                         BYTE    )00
 00A5'  91          1535A                         BYTE    )91
                    1538+         LLDC    XDUCER,3,2,           LATERAL ACCELERATION
 00A6'  2082        1562B                         DATA    )2082
 00A8'  8404        1660A                         DATA    )84*256+ZZVL
 00AA'  0004        1663A                         DATA    )0004
 00AC'  00          1664A                         BYTE    )00
 00AD'  91          1665A                         BYTE    )91
                    1668+         LLDC    XDUCER,3,9,           LONGITUDINAL ACCELERATION
 00AE'  1041        1689B                         DATA    )1041
 00E0'  8410        1790A                         DATA    )84*256+ZZVL
 00B2'  0004        1793A                         DATA    )0004
 00B4'  00          1794A                         BYTE    )00
 00B5'  91          1795A                         BYTE    )91
                    1798+         SYNCRO  XDUCER,3,10,PHASEB,   FLAP POSITION LEFT
 00B6'  2082        1834B                         DATA    )2082
 00B8'  6410        1933A                         DATA    )64*256+ZZVL
 00BA'  2042        1955A                         DATA    )2042
 00BC'  00          1963A                         BYTE    )00
 00BD'  91          1964A                         BYTE    )91
                    1965+         SYNCRO  XDUCER,3,11,PHASEB,   FLAP POSITION RIGHT
 00BE'  4104        2004B                         DATA    )4104
 00C0'  6410        2100A                         DATA    )64*256+ZZVL
 00C2'  2042        2122A                         DATA    )2042
 00C4'  00          2130A                         BYTE    )00
 00C5'  91          2131A                         BYTE    )91
                    2132+         SYNCRO  XDUCER,3,13,PHASEB,   HORIZONTAL STABILIZER POSITION
 00C6'  1041        2165B                         DATA    )1041
 00C8'  6420        2267A                         DATA    )64*256+ZZVL
 00CA'  2042        2289A                         DATA    )2042
```

```
00CC' 00          2297A                       BYTE    )00
00CD' 91          2298A                       BYTE    )91
                  2299+       LLDC    XDUCER,3,19,          FLAP HANDLE POSITION
00CE' 4104        2326B                       DATA    )4104
00D0' 8440        2421A                       DATA    )84*256+ZZVL
00D2' 0004        2424A                       DATA    )0004
00D4' 00          2425A                       BYTE    )00
00D5' 91          2426A                       BYTE    )91
                  2429+       SYNCRO  XDUCER,3,20,PHASEB,     SPOILER HANDLE POSITION
00D6' 8208        2471B                       DATA    )8208
00D8' 6440        2564A                       DATA    )64*256+ZZVL
00DA' 2042        2586A                       DATA    )2042
00DC' 00          2594A                       BYTE    )00
00DD' 91          2595A                       BYTE    )91
                  2596  *
00DE' 0008        2597  ANPRQ   DATA    $-ANMUXT/8      AN-PARAM-QUANT
                  2598  *
                  2599
                  2600  ****************************
                  2601
                  2602  *
                  2603  *       ANALOG PARAMETER'S MASK TABLE
                  2604  *
00E0'             2605  ANMSKT
                  2606  *
                  2607  *       AIRCRAFT TYPE   =       BOEING 757
                  2608  *       ENGINE TYPE     =       ROLLS ROYCE
                  2609  *       2 MAN CREW
                  2610  *
                  2611  ****************************
                  2612  *
                  2613  *       ANALOG PARAMETER TYPE
                  2614  *
00E0' 00 00 00    2615  ANTYPT  BYTE    0,0,0           0 - TYPE LLDC
00E3' 08 08 08    2616          BYTE    8,8,8   3 - TYPE POT REF
00E6' 00          2617          BYTE    0               8 - SYNCRO
00E7' 08          2618          BYTE    8               2 - POT
                  2619  *
                  2620  ****************************
                  2621  *
                  2622  *       ANALOG PARAMETER DESTINATION TABLE OFFSET
                  2623  *
                  2624  *  SUBFRAME 1 CYCLES 1 TO 8
                  2625  *
00E8'             2626  ANDOA
00E8' 00 10 16    2627          BYTE    0,16,22
00EB' 01 09 13    2628          BYTE    1,9,19
00EE' 02 1B       2629          BYTE    2,27
00F0' 03 0A       2630          BYTE    3,10
00F2' 04 11 18    2631          BYTE    4,17,24
00F5' 05 0B 14    2632          BYTE    5,11,20
00F8' 06 0E 1A    2633          BYTE    6,14,26
```

```
00FB' 07 0C       2634          BYTE   7,12
                  2635  *
                  2636  *  SUBFRAME 2 CYCLES 1 TO 8
                  2637  *
00FD' 00 10 16    2638          BYTE   0,16,22
0100' 01 09 13    2639          BYTE   1,9,19
0103' 02 1B       2640          BYTE   2,27
0105' 03 0A       2641          BYTE   3,10
0107' 04 11 18    2642          BYTE   4,17,24
010A' 05 0B 14    2643          BYTE   5,11,20
010D' 06 0E 1A    2644          BYTE   6,14,26
0110' 07 0C       2645          BYTE   7,12
                  2646  *
                  2647  *  SUBFRAME 3 CYCLES 1 TO 8
                  2648  *
0112' 00 10 16    2649          BYTE   0,16,22
0115' 01 09 13    2650          BYTE   1,9,19
0118' 02 1B       2651          BYTE   2,27
011A' 03 0A       2652          BYTE   3,10
011C' 04 11 18    2653          BYTE   4,17,24
011F' 05 0B 14    2654          BYTE   5,11,20
0122' 06 0E 1A    2655          BYTE   6,14,26
0125' 07 0C       2656          BYTE   7,12
                  2657  *
                  2658  *  SUBFRAME 4 CYCLES 1 TO 8
                  2659  *
0127' 00 10 16    2660          BYTE   0,16,22
012A' 01 09 13    2661          BYTE   1,9,19
012D' 02 1B       2662          BYTE   2,27
012F' 03 0A       2663          BYTE   3,10
0131' 04 11 18    2664          BYTE   4,17,24
0134' 05 0B 14    2665          BYTE   5,11,20
0137' 06 0E 1A    2666          BYTE   6,14,26
013A' 07 0C       2667          BYTE   7,12
                  2668  *
                  2669          EVEN
                  2670  *
                  2671  ***********************************
                  2672  *
                  2673  *       ANALOG PARAMETER DESTINATION
                  2674  *
                  2675  *   FORM: GENDT N1,N2,N3,N4,N5
                  2676  *
                  2677  *       N1      =       BUFFER OFFSET WORD (1 TO 256)
                  2678  *       N2      =       BUFFER TYPE
                  2679  *                       0 FOR DFDR
                  2680  *                       1 FOR INTER-CPU
                  2681  *                       2 FOR SPARE
                  2682  *                       3 FOR SPARE
                  2683  *       N3      =       OUTPUT POSITION LEFT SHIFT COUNT
                  2684  *       N4      =       SPARE
                  2685  *       N5      =       END OF DATA SET
```

```
                    2686  *                    0 FOR CONTINUE
                    2687  *                    1 FOR END
                    2688  *
013C'               2689  ANDT
                    2690+       GENDT   2,0,0,0,END             VERTICAL ACCELERATION
013C'  8001         2710A               DATA    (XYZ*256+2-1)
                    2711+       GENDT   10,0,0,0,END                    *
013E'  8009         2731A               DATA    (XYZ*256+10-1)
                    2732+       GENDT   18,0,0,0,END                    *
0140'  8011         2752A               DATA    (XYZ*256+18-1)
                    2753+       GENDT   26,0,0,0,END                    *
0142'  8019         2773A               DATA    (XYZ*256+26-1)
                    2774+       GENDT   34,0,0,0,END                    *
0144'  8021         2794A               DATA    (XYZ*256+34-1)
                    2795+       GENDT   42,0,0,0,END                    *
0146'  8029         2815A               DATA    (XYZ*256+42-1)
                    2816+       GENDT   50,0,0,0,END                    *
0148'  8031         2836A               DATA    (XYZ*256+50-1)
                    2837+       GENDT   58,0,0,0,                       *
014A'  0039         2857A               DATA    (XYZ*256+58-1)
                    2858+       GENDT   1,1,0,0,END                     *       FOR CPU 2
014C'  8100         2878A               DATA    (XYZ*256+1-1)
                    2879  *
                    2880+       GENDT   15,0,1,0,END            LATERAL ACCELLERATION
014E'  840E         2900A               DATA    (XYZ*256+15-1)
                    2901+       GENDT   31,0,1,0,END                    *
0150'  841E         2921A               DATA    (XYZ*256+31-1)
                    2922+       GENDT   47,0,1,0,END                    *
0152'  842E         2942A               DATA    (XYZ*256+47-1)
                    2943+       GENDT   63,0,1,0,                       *
0154'  043E         2963A               DATA    (XYZ*256+63-1)
                    2964+       GENDT   2,1,1,0,END                     *       FOR CPU 2
0156'  8501         2984A               DATA    (XYZ*256+2-1)
                    2985  *
                    2986+       GENDT   55,0,1,0,               LONGITUDINAL ACCELLERATION
0158'  0436         3006A               DATA    (XYZ*256+55-1)
                    3007+       GENDT   3,1,1,0,END                     *       FOR CPU 2
015A'  8502         3027A               DATA    (XYZ*256+3-1)
                    3028  *
                    3029+       GENDT   8,0,2,0,END             FLAP HANDLE POSITION
015C'  8807         3049A               DATA    (XYZ*256+8-1)
                    3050+       GENDT   40,0,2,0,                       *
015E'  0827         3070A               DATA    (XYZ*256+40-1)
                    3071+       GENDT   7,1,2,0,END                     *       FOR CPU 2
0160'  8906         3091A               DATA    (XYZ*256+7-1)
                    3092  *
                    3093+       GENDT   16,0,2,0,END            SPOILER HANDLE POSITION
0162'  880F         3113A               DATA    (XYZ*256+16-1)
                    3114+       GENDT   48,0,2,0,                       *
0164'  082F         3134A               DATA    (XYZ*256+48-1)
                    3135+       GENDT   8,1,2,0,END                     *       FOR CPU 2
0166'  8907         3155A               DATA    (XYZ*256+8-1)
```

```
                    3156  *
                    3157+       GENDT  7,0,2,0,              FLAP POSITION RIGHT
0168' 0806          3177A              DATA   (XYZ*256+7-1)
                    3178+       GENDT  5,1,2,0,END                 *      FOR CPU 2
016A' 8904          3198A              DATA   (XYZ*256+5-1)
                    3199  *
                    3200+       GENDT  39,0,2,0,             FLAP POSITION LEFT
016C' 0826          3220A              DATA   (XYZ*256+39-1)
                    3221+       GENDT  4,1,2,0,END                 *      FOR CPU 2
016E' 8903          3241A              DATA   (XYZ*256+4-1)
                    3242  *
                    3243+       GENDT  56,0,2,0,END          HORIZ. STABILIZER POSITION
0170' 8837          3263A              DATA   (XYZ*256+56-1)
                    3264+       GENDT  24,0,2,0,
0172' 0817          3284A              DATA   (XYZ*256+24-1)
                    3285+       GENDT  6,1,2,0,END                 *      FOR CPU 2
0174' 8905          3305A              DATA   (XYZ*256+6-1)
                    3306  *
                    3307  *
                    3308  ********************************************************************
                    3309  *
                    3310                     CALIBRATION
                    3311  *                  ==========
                    3312  *
                    3313  *  GNDREF AND VCAL MUX VALUES
                    3314  *
0176'               3315  CALMAT
                    3316+       LLDC   VCAL,3,1,
0176' 0810          3353B              DATA   )0810
0178' 8405          3438A              DATA   )84*256+ZZVL
017A' 0004          3441A              DATA   )0004
017C' 00            3442A              BYTE   )00
017D' 91            3443A              BYTE   )91
                    3446+       LLDC   GNDREF,3,1,
017E' 0820          3480B              DATA   )0820
0180' 8406          3568A              DATA   )84*256+ZZVL
0182' 0004          3571A              DATA   )0004
0184' 00            3572A              BYTE   )00
0185' 91            3573A              BYTE   )91
                    3576+       LLDC   VCAL,3,2,
0186' 0810          3613B              DATA   )0810
0188' 8405          3698A              DATA   )84*256+ZZVL
018A' 0004          3701A              DATA   )0004
018C' 00            3702A              BYTE   )00
018D' 91            3703A              BYTE   )91
                    3706+       LLDC   GNDREF,3,2,
018E' 0820          3740B              DATA   )0820
0190' 8406          3828A              DATA   )84*256+ZZVL
0192' 0004          3831A              DATA   )0004
0194' 00            3832A              BYTE   )00
0195' 91            3833A              BYTE   )91
                    3836+       LLDC   VCAL,3,9,
```

```
01B6' 0810      3873B                   DATA    )0810
01B8' 8411      3958A                   DATA    )84*256+ZZVL
01BA' 0004      3961A                   DATA    )0004
01BC' 00        3962A                   BYTE    )00
01BD' 91        3963A                   BYTE    )91
                3966+   LLDC    GNDREF,3,9,
01BE' 0820      4000B                   DATA    )0820
01C0' 8412      4088A                   DATA    )84*256+ZZVL
01C2' 0004      4091A                   DATA    )0004
01C4' 00        4092A                   BYTE    )00
01C5' 91        4093A                   BYTE    )91
                4096+   SYNCRO  VCAL,3,10,PHASEB,
01C6' 8808      4114A                   DATA    )8808
01C8' E480      4122A                   DATA    )E480
01CA' 2042      4157A                   DATA    )2042
01CC' 00        4165A                   BYTE    )00
01CD' 91        4166A                   BYTE    )91
                4167+   SYNCRO  GNDREF,3,10,PHASEB,
01CE' 0820      4213B                   DATA    )0820
01D0' E412      4312A                   DATA    )E4*256+ZZVL
01D2' 2042      4324A                   DATA    )2042
01D4' 00        4332A                   BYTE    )00
01D5' 91        4333A                   BYTE    )91
                4334+   SYNCRO  VCAL,3,11,PHASEB,
01D6' 8808      4352A                   DATA    )8808
01D8' E480      4360A                   DATA    )E480
01DA' 2042      4395A                   DATA    )2042
01DC' 00        4403A                   BYTE    )00
01DD' 91        4404A                   BYTE    )91
                4405+   SYNCRO  GNDREF,3,11,PHASEB,
01DE' 0820      4451B                   DATA    )0820
01E0' E412      4550A                   DATA    )E4*256+ZZVL
01E2' 2042      4562A                   DATA    )2042
01E4' 00        4570A                   BYTE    )00
01E5' 91        4571A                   BYTE    )91
                4572+   SYNCRO  VCAL,3,13,PHASEB,
01E6' 8808      4590A                   DATA    )8808
01E8' E480      4598A                   DATA    )E480
01EA' 2042      4633A                   DATA    )2042
01EC' 00        4641A                   BYTE    )00
01ED' 91        4642A                   BYTE    )91
                4643+   SYNCRO  GNDREF,3,13,PHASEB,
01EE' 0820      4689B                   DATA    )0820
01F0' E422      4788A                   DATA    )E4*256+ZZVL
01F2' 2042      4800A                   DATA    )2042
01F4' 00        4808A                   BYTE    )00
01F5' 91        4809A                   BYTE    )91
                4810+   LLDC    VCAL,3,19,
01F6' 0810      4847B                   DATA    )0810
01F8' 8441      4932A                   DATA    )84*256+ZZVL
01FA' 0004      4935A                   DATA    )0004
01FC' 00        4936A                   BYTE    )00
```

```
                        1555   *
     03:6'  00FA        1556           DATA    250             DUMMY
     03:8'  00FA        1557           DATA    250             DUMMY
     03:A'  00F7 00F8   1558           DATA    247,248
     03:E'  00FA        1559           DATA    250             DUMMY
     0302'  000F 0038   1560           DATA    15,56
     0304'  00B6        1561           DATA    182
     0306'  00FA        1562           DATA    250             DUMMY
     0308'  00FA        1563           DATA    250             DUMMY
                        1564   *
                        1565   *       SUBFRAME 2 CYCLES 1 TO 8
                        1566   *
     030A'  00FA        1567           DATA    250             DUMMY
     030C'  00FA        1568           DATA    250             DUMMY
     030E'  00F9        1569           DATA    249
     0310'  00FA        1570           DATA    250             DUMMY
     0312'  0010        1571           DATA    16
     0314'  00B5        1572           DATA    181
     0316'  00FA        1573           DATA    250             DUMMY
     0318'  00FA        1574           DATA    250             DUMMY
                        1575   *
                        1576   *       SUBFRAME 3 CYCLES 1 TO 8
                        1577   *
     031A'  00FA        1578           DATA    250             DUMMY
     031C'  00FA        1579           DATA    250             DUMMY
     031E'  00FA        1580           DATA    250             DUMMY
     0320'  00FA        1581           DATA    250             DUMMY
     0322'  0020 0038   1582           DATA    32,56
     0326'  00B6        1583           DATA    182
     0328'  0056 0057   1584           DATA    86,87,88,89,90,91,92,93,94,95,96,41
     032C'  0058 0059
     0330'  005A 005B
     0334'  005C 005D
     0338'  005E 005F
     033C'  0060 0029
     0340'  00FA        1585           DATA    250             DUMMY
                        1586   *
                        1587   *       SUBFRAME 4 CYCLES 1 TO 8
                        1588   *
     0342'  00FA        1589           DATA    250             DUMMY
     0344'  00FA        1590           DATA    250             DUMMY
     0346'  00FA        1591           DATA    250             DUMMY
     0348'  00FA        1592           DATA    250             DUMMY
     034A'  0021        1593           DATA    33
     034C'  00B5        1594           DATA    181
     034E'  000A        1595           DATA    10
     0350'  00FA        1596           DATA    250             DUMMY
                        1597   *
     0352'              1598   DR1UB6
                        1599   *
                        1600   *       ALL SUBFRAMES CYCLES 1 TO 8
                        1601   *
```

```
03F2'  0000 0031    1602   DCYC1   DATA    0,49,169,174,1,2,3,4,5,6,7,8,73,74,75,76,77,78,79,80
03F6'  00A9 00AE
03FA'  0001 0002
03FE'  0003 0004
0402'  0005 0006
0406'  0007 0008
040A'  0049 004A
040E'  004B 004C
0412'  004D 004E
0416'  004F 0050
041A'  0051 0052    1603           DATA    81,82,83,84,85,86,87,88,89,90,91,244
041E'  0053 0054
0422'  0055 0056
0426'  0057 0058
042A'  0059 005A
042E'  005B 00F4
0432'  0009 000B    1604   DCYC2   DATA    9,11,40,47,172,10,15,16,24,92,93,94,95,96,97
0436'  0028 002F
043A'  00AC 000A
043E'  000F 0010
0442'  0018 005C
0446'  005D 005E
044A'  005F 0060
044E'  0061
0450'  0062 0063    1605           DATA    98,99,100,101,102,103,104,105,106,107,108,109,110,245
0454'  0064 0065
0458'  0066 0067
045C'  0068 0069
0460'  006A 006B
0464'  006C 006D
0468'  006E 00F5
046C'  000C 002A    1606   DCYC3   DATA    12,42,43,48,169,170,25,32,33,111,112,113,114
0470'  002B 0030
0474'  00A9 00AA
0478'  0019 0020
047C'  0021 006F
0480'  0070 0071
0484'  0072
0486'  0073 0074    1607           DATA    115,116,117,118,119,120,121,122,123,124,125,126,127
048A'  0075 0076
048E'  0077 007B
0492'  0079 007A
0496'  007B 007C
049A'  007D 007E
049E'  007F
04A0'  0080 00B1    1608           DATA    128,129,246
04A4'  00F6
04A6'  000D 009F    1609   DCYC4   DATA    13,159,160,161,162,163,164,165,166,167,168,130,131
04AA'  00A0 00A1
04AE'  00A2 00A3
04B2'  00A4 00A5
04B6'  00A6 00A7
```

```
0434'  00A8 0082
043E'  0083
0448'  0084 0085    1610            DATA    132,133,134,135,136,137,138,139,140,141,142,143,144,145
0454'  0086 0087
045E'  0088 0089
0468'  008A 008B
0472'  008C 008D
047C'  008E 008F
0486'  0090 0091
0490'  0092 0093    1611            DATA    146,147,148,247
049A'  0094 00F7
04A4'  000E 0017    1612    DCYC5   DATA    14,23,44,45,169,34,35,36,37,149,150,151,152,153
04B2'  002C 002D
04BC'  00A9 0022
04C6'  0023 0024
04D0'  0025 0095
04DA'  0096 0097
04E4'  0098 0099
04EE'  009A 009B    1613            DATA    154,155,156,157,158,175,176,177,178,181,182,184,185,186,248
04F8'  009C 009D
0502'  009E 00AF
050C'  00B0 00B1
0516'  00B2 00B5
0520'  00B6 00B8
052A'  00B9 00BA
0534'  00F8
053E'  00B3 00D4    1614    DCYC6   DATA    179,180,183,38,39,41,46,50,51,52,53,54,187,188,189,190,191
0548'  00B7 0026
0552'  0027 0029
055C'  002E 0032
0566'  0033 0034
0570'  0035 0036
057A'  00BB 00BC
0584'  00BD 00BE
058E'  00BF
0598'  00C0 00C1    1615            DATA    192,193,194,195,196,197,198,199,200,201,202,203,204,205,249
05A2'  00C2 00C3
05AC'  00C4 00C5
05B6'  00C6 00C7
05C0'  00C8 00C9
05CA'  00CA 00CB
05D4'  00CC 00CD
05DE'  00F9
05E8'  00A9 00AA    1616    DCYC7   DATA    169,170,171,55,56,57,58,59,60,61,206,207,208,209,210
05F2'  00AB 0037
05FC'  0038 0039
0606'  003A 003B
0610'  003C 003D
061A'  00CE 00CF
0624'  00D0 00D1
062E'  00D2
0638'  00D3 00D4    1617            DATA    211,212,213,214,215,216,217,218,219,220,221,222,223,224
```

```
05300'  00D5 00D6
05304'  00D7 00D8
05308'  00D9 00DA
0530C'  00DB 00DC
05310'  00DD 00DE
05314'  00DF 00E0
05318'  00FB          1618            DATA    251
0531A'  00AD 00BE     1619    OCYC8   DATA    173,62,63,64,65,66,67,68,69,70,71,72,225,226,227,228,229
0531E'  00BF 00C0
05322'  00C1 00C2
05326'  00C3 00C4
0532A'  00C5 00C6
0532E'  00C7 00C8
05332'  00E1 00E2
05336'  00E3 00E4
0533A'  00E5
0533C'  00E6 00E7     1620            DATA    230,231,232,233,234,235,236,237,238,239,240,241,242,243
05340'  00E8 00E9
05344'  00EA 00EB
05348'  00EC 00ED
0534C'  00EE 00EF
05350'  00F0 00F1
05354'  00F2 00F3
05358'  00FC          1621            DATA    252
                      1622    *
                      1623    ********************************************************
                      1624    *       DITS #1 PARAMETER MASK TABLE
                      1625    *
0535A'                1626    DR1MT
                      1627    *
                      1628    *       AIRCRAFT TYPE: B767 WITH G.E ENGINES
                      1629    *
                      1630    ********************************************************
                      1631    *
                      1632    *       DITS #1 PARAMETER NUMBER (ADDRESS)
                      1633    *
                      1634    *                               NAME, LABEL(OCT)
                      1635    *
                      1636    *       CHANNEL #1, PORT #10 - LO SPEED
                      1637    *
0535A'                1638    DR1PNT
                      1639
                      1640    *       CHANNEL NUMBER  1
                      1641
                      1642
0535A'  0001          1643            DATA    1                       N1 ACTUAL L (GE)    346-04  0
0535C'  0002          1644            DATA    2                       TLA                 133-04
0535E'  0003          1645            DATA    3                       TT2                 130-04
05360'  0004          1646            DATA    4                       ENG MDL CODE 1 (GE) 270-00
05362'  0004          1647            DATA    4                       ENG MDL CODE 2 (GE) 270-00
05364'  0004          1648            DATA    4                       ENG MDL CODE 3 (GE) 270-00  5
05366'  0004          1649            DATA    4                       ENG MDL CODE 4      270-00
```

```
05E3' 0004      1650          DATA    4                 ENG MDL CODE 5 (GE)  270-00
05E4' 0004      1651          DATA    4                 ENG MDL CODE 6 (GE)  270-00
05E5' 0004      1652          DATA    4                 ECS BLEED (GE)       270-00
05E6' 0004      1653          DATA    4                 ECS MODE (GE)        270-00  10
05E7' 0004      1654          DATA    4                 CA 1 BLEED (GE)      270-00
05E8' 0004      1655          DATA    4                 WA 1 BLEED           270-00
05E9' 0004      1656          DATA    4                 ADP ON               270-00
05EA' 0004      1657          DATA    4                 ISOL VALVE OPEN (GE) 270-00
05EB' 0005      1658          DATA    5                 EEC/PMC ON           271-00  15
05EC' 0005      1659          DATA    5                 LATCHED FAULT (GE)   271-00
05ED' 0006      1660          DATA    6                 ECS BLEED (PW)       271-01
05EE' 0006      1661          DATA    6                 ECS MODE (PW)        271-01
05EF' 0006      1662          DATA    6                 CA 1 BLEED (PW)      271-01
05F0' 0006      1663          DATA    6                 WA 1 BLEED (PW)      271-01  20
05F1' 0006      1664          DATA    6                 ADP ON (PW)          271-01
05F2' 0006      1665          DATA    6                 ISOL VALVE OPEN (PW) 271-01
                1666
                1667
                1668  *       CHANNEL NUMBER  2
                1669
                1670
05F3' 0007      1671          DATA    7                 N1 ACT R (GE)        346-04
05F4' 0008      1672          DATA    8                 TLA                  133-04
05F5' 0009      1673          DATA    9                 TT2                  130-04  25
05F6' 000A      1674          DATA    10                ECS BLEED (PW)       271-01
05F7' 000A      1675          DATA    10                ECS MODE (PW)        271-01
05F8' 000A      1676          DATA    10                CA 1 BLEED           271-01
05F9' 000A      1677          DATA    10                WA 1 BLEED           271-01
05FA' 000A      1678          DATA    10                ADP ON               271-01  30
05FB' 000A      1679          DATA    10                ISOL VALVE OPEN (PW) 271-01
05FC' 000B      1680          DATA    11                EEC/PMC ON           271-00
05FD' 000B      1681          DATA    11                LATCHED FAULT        271-00
05FE' 000C      1682          DATA    12                ENG MDL CODE 1       270-00
05FF' 000C      1683          DATA    12                ENG MDL CODE 2       270-00  35
0600' 000C      1684          DATA    12                ENG MDL CODE 3       270-00
0601' 000C      1685          DATA    12                ENG MDL CODE 4       270-00
0602' 000C      1686          DATA    12                ENG MDL CODE 5       270-00
0603' 000C      1687          DATA    12                ENG MDL CODE 6       270-00
0604' 000C      1688          DATA    12                ECS BLEED            270-00  40
0605' 000C      1689          DATA    12                ECS MODE             270-00
0606' 000C      1690          DATA    12                CA 1 BLEED           270-00
0607' 000C      1691          DATA    12                WA 1 BLEED           270-00
0608' 000C      1692          DATA    12                ADP ON               270-00
0609' 000C      1693          DATA    12                ISOL VALVE OPEN (GE) 270-00  45
                1694
                1695
                1696  *       CHANNEL NUMBER  3
                1697
                1698
060A' 000D      1699          DATA    13                ANGLE OF ATTACK      221-04
060B' 000E      1700          DATA    14                TOTAL AIR TEMP       211-04
060C' 000F      1701          DATA    15                COMPUTED AIRSPEED    206-04
```

05F0'	0010	1702	DATA	16	PRESS ALT 29.92	203-04	
05F2'	0011	1703	DATA	17	MACH	205-04	50
05C0'	0012	1704	DATA	18	ALT BARO 1	204-04	
05C2'	0013	1705	DATA	19	ALT BARO 2	220-04	
05C4'	0014	1706	DATA	20	ALT BARO 3	251-04	
05C6'	0015	1707	DATA	21	ALT BARO 4	252-04	
05C8'	0016	1708	DATA	22	ALTTITUDE RATE	212-04	55
05CA'	0017	1709	DATA	23	ANGLE OF ATTACK L	222-04	
05CC'	0018	1710	DATA	24	ANGLE OF ATTACK R	223-04	
05CE'	0019	1711	DATA	25	BARO CORR 1 (IN)	235-04	
05D0'	0019	1712	DATA	25	BARO CORR 1 (IN)	235-04	
05D2'	001A	1713	DATA	26	BARO CORR 1 (MB)	234-04	60
05D4'	001A	1714	DATA	26	BARO CORR 1 (MB)	234-04	
05D6'	001B	1715	DATA	27	BARO CORR 2 (IN)	237-04	
05D8'	001B	1716	DATA	27	BARO CORR 2 (IN)	237-04	
05DA'	001C	1717	DATA	28	BARO CORR 2 (MB)	236-04	
05DC'	001C	1718	DATA	28	BARO CORR 2 (MB)	236-04	65
05DE'	001D	1719	DATA	29	BARO CORR 3 (IN)	35-04	
05E0'	001D	1720	DATA	29	BARO CORR 3 (IN)	35-04	
05E2'	001E	1721	DATA	30	BARO CORR 3 (MB)	34-04	
05E4'	001E	1722	DATA	30	BARO CORR 3 (MB)	34-04	
05E6'	001F	1723	DATA	31	CORRECTED AOA	241-04	70
05E8'	0020	1724	DATA	32	IMPACT PRESS	215-04	
05EA'	0021	1725	DATA	33	MAX OPR TG SCHED	207-04	
05EC'	0022	1726	DATA	34	STATIC AIR TEMP	213-04	
05EE'	0023	1727	DATA	35	STATIC AIR TEMP D	233-04	
05F0'	0024	1728	DATA	36	TOTAL AIR TEMP D	231-04	75
05F2'	0025	1729	DATA	37	TOTAL PRESS	242-04	
05F4'	0026	1730	DATA	38	TRUE AIRSPEED	210-04	
05F6'	0027	1731	DATA	39	TRUE AIRSPEED D	230-04	
05F8'	0028	1732	DATA	40	A/C TYPE MEM CHECK	352-04	
05FA'	0028	1733	DATA	40	PT CHECK	352-04	80
05FC'	0028	1734	DATA	40	PS CHECK	352-04	
05FE'	0028	1735	DATA	40	PROM 4 CHECK	352-04	
0600'	0028	1736	DATA	40	PROM 3 CHECK	352-04	
0602'	0028	1737	DATA	40	PROM 2 CHECK	352-04	
0604'	0028	1738	DATA	40	PROM 1 CHECK	352-04	85
0606'	0029	1739	DATA	41	VMO ALT 1	270-04	
0608'	0029	1740	DATA	41	RAOA HEAT ON	270-04	
060A'	0029	1741	DATA	41	LAOA HEAT ON	270-04	
060C'	0029	1742	DATA	41	TAT PROBE HEAT ON	270-04	
060E'	0029	1743	DATA	41	PILOT/STAT HEAT ON	270-04	90
0610'	0029	1744	DATA	41	ADC INVALID	270-04	
0612'	0029	1745	DATA	41	ONSIDE AOA FAIL	270-04	
0614'	0029	1746	DATA	41	OVER SPEED	270-04	
0616'	0029	1747	DATA	41	VMO ALT 2	270-04	
0618'	0029	1748	DATA	41	VMO ALT 3	270-04	95
061A'	0029	1749	DATA	41	PILOT/STAT HEAT ON R	270-04	
061C'	0029	1750	DATA	41	ZERO SSEC (MACH)	270-04	
061E'	0029	1751	DATA	41	BARO PORT A	270-04	
0620'	0029	1752	DATA	41	AOA C ALTERNATE	270-04	
0622'	0029	1753	DATA	41	ESEC ALTERNATE	270-04	100

062A'	0029	1754	DATA	41	VMD ALT 4	270-04	
062C'	0029	1755	DATA	41	ADA UNIQUE	270-04	
062E'	0029	1756	DATA	41	PILOT HEAT ON	270-04	
0630'	0029	1757	DATA	41	ICING DETEC ON	270-04	
0632'	000F	1758	DATA	15	CAS FLAG	206-04	105
0633'	0011	1759	DATA	17	MACH FLAG	205-04	
0636'	000D	1760	DATA	13	ADA FLAG	221-04	
0638'	0010	1761	DATA	16	PRESS ALT FLAG	203-04	
063A'	000E	1762	DATA	14	TAT FLAG	211-04	
063C'	002F	1763	DATA	47	4 BARO ALT =3	271-04	110
063E'	002F	1764	DATA	47	4 BARO ALT =2	271-04	
0640'	002F	1765	DATA	47	4 BARO ALT =1	271-04	
0642'	002F	1766	DATA	47	EXT ADA MON (FAIL)	271-04	
0644'	002F	1767	DATA	47	ZERO SSEC (ADA)	271-04	
0646'	0030	1768	DATA	48	F/D CONVERSION TEST	350-04	115
0648'	0030	1769	DATA	48	PS CALIB TEST	350-04	
064A'	0030	1770	DATA	48	PS SENS TEST	350-04	
064C'	0030	1771	DATA	48	PS SENS PER TEST	350-04	
064E'	0030	1772	DATA	48	PROG MEM TEST'	350-04	
0650'	0030	1773	DATA	48	RAM TEST	350-04	120
0652'	0030	1774	DATA	48	PROCESSOR TEST	350-04	
0654'	0030	1775	DATA	48	A TO D TEST	350-04	
0656'	0030	1776	DATA	48	DSPD HW TEST	350-04	
0658'	0030	1777	DATA	48	A/C TYPE CONST TEST	350-04	
065A'	0030	1778	DATA	48	A/C TYPE PROG TEST	350-04	125
065C'	0030	1779	DATA	48	BARO 3 TEST	350-04	
065E'	0030	1780	DATA	48	BARO 2 TEST	350-04	
0660'	0030	1781	DATA	48	BARO 1 TEST	350-04	
0662'	0030	1782	DATA	48	TAT INPUT TEST	350-04	
0664'	0030	1783	DATA	48	RADA VANE TEST	350-04	130
0666'	0030	1784	DATA	48	LADA VANE TEST	350-04	
0668'	0031	1785	DATA	49	VMD TEST	351-04	
066A'	0031	1786	DATA	49	BARO 4 TEST	351-04	
066C'	0031	1787	DATA	49	EAROM TEST	351-04	
066E'	0031	1788	DATA	49	PT PLL	351-04	135
0670'	0031	1789	DATA	49	PS PLL	351-04	
0672'	0031	1790	DATA	49	PROG SEQ TEST	351-04	
0674'	0031	1791	DATA	49	TEMP PS=TEMP PT	351-04	
0676'	0031	1792	DATA	49	AVGE ADA TEST	351-04	
0678'	0031	1793	DATA	49	PS = PT	351-04	140
067A'	0031	1794	DATA	49	ADA COMPARE TEST	351-04	
067C'	0031	1795	DATA	49	POWER SUPPLY TEST	351-04	
067E'	0031	1796	DATA	49	ARINC XMTR TEST	351-04	
0680'	0031	1797	DATA	49	ARINC XMTR TEST	351-04	
0682'	0031	1798	DATA	49	ARINC XMTR TEST	351-04	145
0684'	0031	1799	DATA	49	PT CALIB TEST	351-04	
0686'	0031	1800	DATA	49	PT SENS TEMP TEST	351-04	
0688'	0031	1801	DATA	49	PT SENS PER TEST	351-04	
068A'	0028	1802	DATA	40	A/C TYPE PARITY	352-04	
068C'	0028	1803	DATA	40	A/C TYPE MSB	352-04	150
068E'	0028	1804	DATA	40	A/C TYPE LSB+3	352-04	
0690'	0028	1805	DATA	40	A/C TYPE LSB+2	352-04	

0692'	0028	1806		DATA	40	A/C TYPE LSB+1	352-04	
0696'	0028	1807		DATA	40	A/C TYPE LSB	352-04	
069A'	0028	1808		DATA	40	A/C TYPE MEM CHECK	352-04	155
069E'	0028	1809		DATA	40	PT MEM CHECK	352-04	
06A2'	0028	1810		DATA	40	PS MEM CHECK	352-04	
06A6'	0028	1811		DATA	40	PROM 8 CHECK	352-04	
		1812						
		1813						
		1814	*	CHANNEL NUMBER 4				
		1815						
		1816						
06A9'	0033	1817		DATA	51	TERRAIN PULL UP	270-04	
06AD'	0033	1818		DATA	51	MINIMUMS	270-04	160
06B1'	0033	1819		DATA	51	GLIDE SLOPE	270-04	
06B5'	0033	1820		DATA	51	TOO LOW TERRAIN	270-04	
06B9'	0033	1821		DATA	51	TOO LOW FLAPS	270-04	
06BD'	0033	1822		DATA	51	TOO LOW GEAR	270-04	
06C1'	0033	1823		DATA	51	DON'T SINK	270-04	165
06C5'	0033	1824		DATA	51	TERRAIN	270-04	
06C9'	0033	1825		DATA	51	PULL UP	270-04	
06CD'	0033	1826		DATA	51	SINK RATE	270-04	
		1827						
		1828						
		1829	*	CHANNEL NUMBER 5				
		1830						
		1831						
06D1'	0034	1832		DATA	52	PITCH ATTITUDE	324-04	
06D5'	0035	1833		DATA	53	ROLL ATTITUDE	325-04	170
06D9'	0036	1834		DATA	54	LOCALIZER DEV	173-04	
06DD'	0037	1835		DATA	55	GLIDE SLOPE DEV	174-04	
06E1'	0038	1836		DATA	56	RADIO ALT	164-04	
06E5'	0039	1837		DATA	57	MAGN HEADING	320-04	
06E9'	003A	1838		DATA	58	TRACK ANGLE MAG	317-04	175
06ED'	003B	1839		DATA	59	TRACK ANGLE TRUE	313-04	
06F1'	003C	1840		DATA	60	TRUE HEADING	314-04	
06F5'	003D	1841		DATA	61	MAG TRUE DATA	270-04	
06F9'	003D	1842		DATA	61	DH ALERT	270-04	
06FD'	003D	1843		DATA	61	H ALERT	270-04	180
0701'	003D	1844		DATA	61	GROUNDSPEED SOURCE	270-04	
0705'	003D	1845		DATA	61	TRACK ANGLE SOURCE	270-04	
0709'	003D	1846		DATA	61	DH + H ALERT	270-04	
070D'	003E	1847		DATA	62	G/S MODE OPER	274-04	
0711'	003E	1848		DATA	62	FLARE OPER	274-04	185
0715'	003E	1849		DATA	62	G/A MODE OPER-P	274-04	
0719'	003E	1850		DATA	62	T/O MODE OPER-P	274-04	
071D'	003E	1851		DATA	62	ALT HOLD MODE OPER	274-04	
0721'	003E	1852		DATA	62	V/S MODE OPER	274-04	
0725'	003E	1853		DATA	62	V/NAV MODE OPER	274-04	190
0729'	003E	1854		DATA	62	FL CH MODE OPER	274-04	
072D'	003E	1855		DATA	62	THROTTLE RETARD	274-04	
0731'	003E	1856		DATA	62	IAS MODE OPER	274-04	
0735'	003E	1857		DATA	62	MACH MODE SET	274-04	

06E0'	003E	1858	DATA	62	ALT MODE OPER	274-04	195
06E7'	003E	1859	DATA	62	PITCH SPEED CNTRL	274-04	
06E4'	003F	1860	DATA	63	WXR DATA FAULT	352-04	
06E6'	003F	1861	DATA	63	MLS DATA FAULT	352-04	
06E8'	003F	1862	DATA	63	ILS DATA FAULT	352-04	
06EA'	003F	1863	DATA	63	RA DATA FAULT	352-04	200
06EC'	003F	1864	DATA	63	D-DME DATA FAULT	352-04	
06EE'	003F	1865	DATA	63	L-DME DATA FAULT	352-04	
06F0'	003F	1866	DATA	63	R-VOR DATA FAULT	352-04	
06F2'	003F	1867	DATA	63	L-VOR DATA FAULT	352-04	
06F4'	003F	1868	DATA	63	R-ADC DATA FAULT	352-04	205
06F6'	003F	1869	DATA	63	L-ADC DATA FAULT	352-04	
06F8'	003F	1870	DATA	63	R-IRS DATA FAULT	352-04	
06FA'	003F	1871	DATA	63	C-IRS DATA FAULT	352-04	
06FC'	003F	1872	DATA	63	L-IRS DATA FAULT	352-04	
06FE'	003F	1873	DATA	63	TMC DATA FAULT	352-04	210
0700'	003F	1874	DATA	63	R-FMC DATA FAULT	352-04	
0702'	003F	1875	DATA	63	L-FMC DATA FAULT	352-04	
0704'	003F	1876	DATA	63	R-FCC DATAS FAULT	352-04	
0706'	003F	1877	DATA	63	C-FCC DATA FAULT	352-04	
0708'	003F	1878	DATA	63	L-FCC DATA FAULT	352-04	215
070A'	0040	1879	DATA	64	SG FAULT	353-04	
070C'	0040	1880	DATA	64	R EHSI FAULT	353-04	
070E'	0040	1881	DATA	64	L EHSI FAULT	353-04	
0710'	0040	1882	DATA	64	R CP FAULT	353-04	
0712'	0040	1883	DATA	64	L CP FAULT	353-04	220
0714'	0040	1884	DATA	64	R EADI FAULT	353-04	
0716'	0040	1885	DATA	64	L EADI FAULT	353-04	
0718'	0041	1886	DATA	65	SG I/O PROC 3	350-04	
071A'	0041	1887	DATA	65	SG I/O PROC 2	350-04	
071C'	0041	1888	DATA	65	SG I/O PROC 1	350-04	225
071E'	0041	1889	DATA	65	SG DIGITAL O/P	350-04	
0720'	0041	1890	DATA	65	SG CONTROLLER	350-04	
0722'	0041	1891	DATA	65	SG DSPLY SEQCER	350-04	
0724'	0041	1892	DATA	65	SG DSPLY DRIVE	350-04	
0726'	0041	1893	DATA	65	SG MAIN MEMORY	350-04	230
0728'	0041	1894	DATA	65	SG MAIN PROCESSOR	350-04	
072A'	0041	1895	DATA	65	SG OVERTEMP	350-04	
072C'	0042	1896	DATA	66	R-CP FAULT	351-04	
072E'	0042	1897	DATA	66	L-CP FAULT	351-04	
0730'	0042	1898	DATA	66	R-EHSI OVERTEMP	351-04	235
0732'	0042	1899	DATA	66	R-EHSI BEAM FAIL	351-04	
0734'	0042	1900	DATA	66	R-EHSI ANOMALIES	351-04	
0736'	0042	1901	DATA	66	L-EHSI OVERTEMP	351-04	
0738'	0042	1902	DATA	66	L-EHSI BEAM FAIL	351-04	
073A'	0042	1903	DATA	66	L-EHSI ANOMALIES	351-04	240
073C'	0042	1904	DATA	66	R-EADI OVERTEMP	351-04	
073E'	0042	1905	DATA	66	R-EADI BEAM FAIL	351-04	
0740'	0042	1906	DATA	66	R-EADI ANOMALIES	351-04	
0742'	0042	1907	DATA	66	L-EADI OVERTEMP	351-04	
0744'	0042	1908	DATA	66	L-EADI BEAM FAIL	351-04	245
0746'	0042	1909	DATA	66	L-EADI ANOMALIES	351-04	

```
                    1910
                    1911
                    1912  *       CHANNEL NUMBER  6
                    1913
                    1914
    07[illegible]  0043     1915          DATA    67                      GMT SECONDS          150-04
    0746'  0043     1916          DATA    67                      GMT MINUTES          150-04
    074C'  0043     1917          DATA    67                      GMT HOURS            150-04  249
                    1918  *
    074E'  0044     1919          DATA    68                ****** DUMMY *******
                    1920  *
                    1921  ***  ADDITIONS  ****
                    1922
                    1923  *       CHANNEL NUMBER  1
                    1924
    0750'  0045     1925          DATA    69                      N1 MAX/EPR LIMIT     342-04
                    1926
                    1927  *       CHANNEL NUMBER  2
                    1928
    0752'  0046     1929          DATA    70                      N1 MAX/EPR LIMIT     342-04
                    1930  *
                    1931  ***********************************************
                    1932  *
                    1933  *       DITS UPDATE TABLE
                    1934  *
    0754'           1935  DR1UPT
    0754'  0000     1936          DATA    0
    0756'  0006     1937          DATA    6
    0758'  000C     1938          DATA    12
    075A'  0031     1939          DATA    49
    075C'  0033     1940          DATA    51
    075E'  0042     1941          DATA    66
    0760'  0043     1942          DATA    67
    0762'  0000     1943          DATA    0
                    1944
                    1945  ***********************************************
                    1946  *
                    1947  *       DITS #1 POSITION TABLE
                    1948  *
                    1949  *
                    1950  *       N1      =       LSB POSITION OF DATA
                    1951  *       N2      =       INPUT MASK WORD INDEX
                    1952  *       N3      =       SIGN  1 - SIGN
                    1953  *                             0 - NO SIGN
                    1954  *       N4      =       BINARY/BCD FORMAT
                    1955  *                       0 = BINARY
                    1956  *                       1 = BCD
                    1957  *       N5      =       UPDATE CHECK
                    1958  *                       1 = UPDATE
                    1959  *                       0 = NO UPDATE
                    1960  *       N6      =       SSM CHECK
                    1961  *                       1 = SSM
```

```
                    1962  *                    0 = NO SSM
                    1963  *       N7     =     PARITY
                    1964  *                    1 = PARITY
                    1965  *                    0 = NO PARITY
                    1966  *       N8     =     ROUNDING
                    1967  *                    BLANK = NO ROUNDING
                    1968  *
                    1969  *
0764'               1970  DR1FST
                    1971  *
                    1972  *       CHANNEL NUMBER  1
                    1973
                    1974
                    1975+         DRPT   17,10,0,0,0,1,1,          N1 ACTUAL L (GE)     346-04
0764' 3150          2006A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2008+         DRPT   18,10,1,0,0,1,1,R         TLA                  133-04
0766' B351          2037A                DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    2041+         DRPT   17,11,1,0,0,1,1,R         TT2                  130-04
0768' B370          2070A                DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    2074+         DRPT   17,0,0,0,0,0,0,           ENG MDL CODE 1 (GE)  270-04
076A' 0010          2105A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2107+         DRPT   18,0,0,0,0,0,0,           ENG MDL CODE 2 (GE)  270-04
076C' 0011          2138A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2140+         DRPT   19,0,0,0,0,0,0,           ENG MDL CODE 3 (GE)  270-04
076E' 0012          2171A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2173+         DRPT   20,0,0,0,0,0,0,           ENG MDL CODE 4       270-04
0770' 0013          2204A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2206+         DRPT   21,0,0,0,0,0,0,           ENG MDL CODE 5 (GE)  270-04
0772' 0014          2237A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2239+         DRPT   22,0,0,0,0,0,0,           ENG MDL CODE 6 (GE)  270-04
0774' 0015          2270A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2272+         DRPT   23,0,0,0,0,0,0,           ECS BLEED (GE)       270-04
0776' 0016          2303A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2305+         DRPT   24,0,0,0,0,0,0,           ECS MODE (GE)        270-04
0778' 0017          2336A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2338+         DRPT   25,0,0,0,0,0,0,           CA 1 BLEED (GE)      270-04
077A' 0018          2369A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2371+         DRPT   26,0,0,0,0,0,0,           WA 1 BLEED           270-04
077C' 0019          2402A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2404+         DRPT   27,0,0,0,0,0,0,           ADP ON               270-04
077E' 001A          2435A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2437+         DRPT   28,0,0,0,0,0,0,           ISOL VALVE OPEN (GE) 270-04
0780' 001B          2468A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2470+         DRPT   20,0,0,0,0,0,0,           EEC/PMC ON           271-04
0782' 0013          2501A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2503+         DRPT   22,0,0,0,0,0,0,           LATCHED FAULT (GE)   271-04
0784' 0015          2534A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2536+         DRPT   21,0,0,0,0,0,0,           ECS BLEED (PW)       271-01
0786' 0014          2567A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2569+         DRPT   22,0,0,0,0,0,0,           ECS MODE (PW)        271-01
0788' 0015          2600A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    2602+         DRPT   25,0,0,0,0,0,0,           CA 1 BLEED (PW)      271-01
```

```
07D4' 0018  2633A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2635+      DRPT      24,0,0,0,0,0,0,                WA 1 BLEED (PW)      271-01
07D5' 0017  2666A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2668+      DRPT      23,0,0,0,0,0,0,                ADP ON (PW)          271-01
07D6' 0016  2699A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2701+      DRPT      26,0,0,0,0,0,0,                ISOL VALVE OPEN (PW) 271-01
07D7' 0019  2732A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2734
            2735
            2736   *   CHANNEL NUMBER  2
            2737
            2738
            2739+      DRPT      17,10,0,0,0,1,1,               N1 ACT R (GE)        346-04
07D8' 3150  2770A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2772+      DRPT      18,10,1,0,0,1,1,R              TLA                  133-04
07D9' B351  2801A                DATA   XX1+XX2+XX3+XX4+XX5+XX6+)B000
            2805+      DRPT      17,11,1,0,0,1,1,R              TT2                  130-04
07DA' B370  2834A                DATA   XX1+XX2+XX3+XX4+XX5+XX6+)B000
            2838+      DRPT      21,0,0,0,0,0,0,                ECS BLEED (PW)       271-01
07DB' 0014  2869A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2871+      DRPT      22,0,0,0,0,0,0,                ECS MODE (PW)        271-01
07DC' 0015  2902A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2904+      DRPT      25,0,0,0,0,0,0,                CA 1 BLEED           271-01
07DD' 0018  2935A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2937+      DRPT      24,0,0,0,0,0,0,                WA 1 BLEED           271-01
07DE' 0017  2968A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            2970+      DRPT      23,0,0,0,0,0,0,                ADP ON               271-01
07DF' 0016  3001A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3003+      DRPT      26,0,0,0,0,0,0,                ISOL VALVE OPEN (PW) 271-01
07E0' 0019  3034A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3036+      DRPT      20,0,0,0,0,0,0,                EEC/PMC ON           271-04
07E1' 0013  3067A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3069+      DRPT      22,0,0,0,0,0,0,                LATCHED FAULT        271-04
07E2' 0015  3100A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3102+      DRPT      17,0,0,0,0,0,0,                ENG MDL CODE 1       270-04
07E3' 0010  3133A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3135+      DRPT      18,0,0,0,0,0,0,                ENG MDL CODE 2       270-04
07E4' 0011  3166A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3168+      DRPT      19,0,0,0,0,0,0,                ENG MDL CODE 3       270-04
07E5' 0012  3199A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3201+      DRPT      20,0,0,0,0,0,0,                ENG MDL CODE 4       270-04
07E6' 0013  3232A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3234+      DRPT      21,0,0,0,0,0,0,                ENG MDL CODE 5       270-04
07E7' 0014  3265A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3267+      DRPT      22,0,0,0,0,0,0,                ENG MDL CODE 6       270-04
07E8' 0015  3298A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3300+      DRPT      23,0,0,0,0,0,0,                ECS BLEED            270-04
07E9' 0016  3331A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3333+      DRPT      24,0,0,0,0,0,0,                ECS MODE             270-04
07EA' 0017  3364A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
            3366+      DRPT      25,0,0,0,0,0,0,                CA 1 BLEED           270-04
07EB' 0018  3397A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
```

```
                    3399+       DRPT   26,0,0,0,0,0,0,                 WA 1 BLEED           270-04
07E8' 0019          3430A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3432+       DRPT   27,0,0,0,0,0,0,                 ADP ON               270-04
07E9' 001A          3463A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3465+       DRPT   28,0,0,0,0,0,0,                 ISOL VALVE OPEN (GE) 270-04
07EA' 001B          3496A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3498
                    3499
                    3500   *    CHANNEL NUMBER  3
                    3501
                    3502
                    3503+       DRPT   18,9,1,0,0,1,1,R                ANGLE OF ATTACK      221-04
07EC' B331          3532A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3536+       DRPT   18,8,1,0,0,1,1,                 TOTAL AIR TEMP       211-04
07ED' 3311          3567A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3569+       DRPT   15,12,0,0,0,1,1,R               COMPUTED AIRSPEED    206-04
07EE' B18E          3598A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3602+       DRPT   16,11,1,0,0,1,1,R               PRESS ALT 29.92      203-04
07EF' B36F          3631A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3635+       DRPT   16,10,0,0,0,1,1,R               MACH                 205-04
07F0' B14F          3664A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3668+       DRPT   16,11,1,0,0,1,1,R               ALT BARO 1           204-04
07F2' B36F          3697A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3701+       DRPT   16,11,1,0,0,1,1,R               ALT BARO 2           220-04
07F3' B36F          3730A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3734+       DRPT   16,11,1,0,0,1,1,R               ALT BARO 3           251-04
07F5' B36F          3763A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3767+       DRPT   16,11,1,0,0,1,1,R               ALT BARO 4           252-04
07F6' B36F          3796A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3800+       DRPT   18,10,1,0,0,1,1,R               ALTTITUDE RATE       212-04
07F8' B351          3829A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3833+       DRPT   18,10,1,0,0,1,1,R               ANGLE OF ATTACK L    222-04
07F9' B351          3862A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3866+       DRPT   17,11,1,0,0,1,1,R               ANGLE OF ATTACK R    223-04
07FB' B370          3895A              DATA   XX1+XX2+XX3+XX4+XX5+XX6+)8000
                    3899+       DRPT   11,11,0,1,0,1,1,                BARO CORR 1 (IN)*    235-04
07FC' 356A          3930A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3932+       DRPT   23,6,1,1,0,1,1,                 BARO CORR 1 (IN)*    235-04
07FD' 36D6          3963A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3965+       DRPT   11,11,0,1,0,1,1,                BARO CORR 1 (MB)*    234-04
07FF' 356A          3996A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    3998+       DRPT   23,6,1,1,0,1,1,                 BARO CORR 1 (MB)*    234-04
0800' 36D6          4029A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    4031+       DRPT   11,11,0,1,0,1,1,                BARO CORR 2 (IN)*    237-04
0802' 356A          4062A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    4064+       DRPT   23,6,1,1,0,1,1,                 BARO CORR 2 (IN)*    237-04
0803' 36D6          4095A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    4097+       DRPT   11,11,0,1,0,1,1,                BARO CORR 2 (MB)*    236-04
0805' 356A          4128A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    4130+       DRPT   23,6,1,1,0,1,1,                 BARO CORR 2 (MB)*    236-04
0806' 36D6          4161A              DATA   XX1+XX2+XX3+XX4+XX5+XX6
                    4163+       DRPT   11,11,0,1,0,1,1,                BARO CORR 3 (IN)*     35-04
```

```
07F8' 356A      4194A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4196+   DRPT    23,6,1,1,0,1,1,          BARO CORR 3 (IN)*     35-04
07FA' 36D6      4227A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4229+   DRPT    11,11,0,1,0,1,1,         BARO CORR 3 (MB)*     34-04
07FC' 356A      4260A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4262+   DRPT    23,6,1,1,0,1,1,          BARO CORR 3 (MB)*     34-04
07FE' 36D6      4293A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4295+   DRPT    17,11,1,0,0,1,1,R        CORRECTED AOA         241-04
0800' B370      4324A                DATA    XX1+XX2+XX3+XX4+XX5+XX6+)8000
                4328+   DRPT    17,11,1,0,0,1,1,R        IMPACT PRESS          215-04
0802' B370      4357A                DATA    XX1+XX2+XX3+XX4+XX5+XX6+)8000
                4361+   DRPT    17,11,1,0,0,1,1,R        MAX OPR TG SCHED      207-04
0804' B370      4390A                DATA    XX1+XX2+XX3+XX4+XX5+XX6+)8000
                4394+   DRPT    18,10,1,0,0,1,1,         STATIC AIR TEMP       213-04
0806' 3351      4425A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4427+   DRPT    19,10,1,1,0,1,1,         STATIC AIR TEMP D*    233-04
0808' 3752      4458A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4460+   DRPT    19,10,1,1,0,1,1,         TOTAL AIR TEMP D*     231-04
080A' 3752      4491A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4493+   DRPT    17,11,1,0,0,1,1,R        TOTAL PRESS           242-04
080C' B370      4522A                DATA    XX1+XX2+XX3+XX4+XX5+XX6+)8000
                4526+   DRPT    16,11,0,0,0,1,1,         TRUE AIRSPEED         210-04
080E' 316F      4557A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4559+   DRPT    19,10,0,1,0,0,1,         TRUE AIRSPEED D*      230-04
0810' 2552      4590A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4592+   DRPT    17,0,0,0,0,0,0,          A/C TYPE MEM CHECK    352-04
0812' 0010      4623A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4625+   DRPT    16,0,0,0,0,0,0,          PT CHECK              352-04
0814' 000F      4656A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4658+   DRPT    15,0,0,0,0,0,0,          PS CHECK              352-04
0816' 000E      4689A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4691+   DRPT    14,0,0,0,0,0,0,          PROM 4 CHECK          352-04
0818' 000D      4722A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4724+   DRPT    13,0,0,0,0,0,0,          PROM 3 CHECK          352-04
081A' 000C      4755A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4757+   DRPT    12,0,0,0,0,0,0,          PROM 2 CHECK          352-04
081C' 000B      4788A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4790+   DRPT    11,0,0,0,0,0,0,          PROM 1 CHECK          352-04
081E' 000A      4821A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4823+   DRPT    22,0,0,0,0,0,0,          VMO ALT 1             270-04
0820' 0015      4854A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4856+   DRPT    18,0,0,0,0,0,0,          RADA HEAT ON          270-04
0822' 0011      4887A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4889+   DRPT    17,0,0,0,0,0,0,          LADA HEAT ON          270-04
0824' 0010      4920A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4922+   DRPT    16,0,0,0,0,0,0,          TAT PROBE HEAT ON     270-04
0826' 000F      4953A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4955+   DRPT    14,0,0,0,0,0,0,          PILOT/STAT HEAT ON    270-04
0828' 000D      4986A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                4988+   DRPT    13,0,0,0,0,0,0,          ADC INVALID           270-04
082A' 000C      5019A                DATA    XX1+XX2+XX3+XX4+XX5+XX6
                5021+   DRPT    20,0,0,0,0,0,0,          ONSIDE AOA FAIL       270-04
```

```
0810' 0013  5052A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5054+  DRPT  19,0,0,0,0,0,0,   OVER SPEED          270-04
0818' 0012  5085A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5087+  DRPT  23,0,0,0,0,0,0,   VMO ALT 2           270-04
0820' 0016  5118A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5120+  DRPT  24,0,0,0,0,0,0,   VMO ALT 3           270-04
0822' 0017  5151A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5153+  DRPT  15,0,0,0,0,0,0,   PILOT/STAT HEAT ON R 270-04
0824' 000E  5184A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5186+  DRPT  29,0,0,0,0,0,0,   ZERO SSEC (MACH)    270-04
0826' 001C  5217A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5219+  DRPT  28,0,0,0,0,0,0,   BARO PORT A         270-04
0828' 001B  5250A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5252+  DRPT  27,0,0,0,0,0,0,   AOA C ALTERNATE     270-04
082A' 001A  5283A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5285+  DRPT  26,0,0,0,0,0,0,   SSEG ALTERNATE      270-04
082C' 0019  5316A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5318+  DRPT  25,0,0,0,0,0,0,   VMO ALT 4           270-04
082E' 001B  5349A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5351+  DRPT  21,0,0,0,0,0,0,   AOA UNIQUE          270-04
0830' 0014  5382A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5384+  DRPT  12,0,0,0,0,0,0,   SPARE               270-04
0832' 000B  5415A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5417+  DRPT  11,0,0,0,0,0,0,   SPARE               270-04
0834' 000A  5448A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5450+  DRPT  30,1,0,0,0,0,0,   CAS FLAG            206-04
0836' 003D  5481A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5483+  DRPT  30,1,0,0,0,0,0,   MACH FLAG           205-04
0838' 003D  5514A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5516+  DRPT  30,1,0,0,0,0,0,   AOA FLAG            221-04
083A' 003D  5547A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5549+  DRPT  30,1,0,0,0,0,0,   PRESS ALT FLAG      203-04
083C' 003D  5580A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5582+  DRPT  30,1,0,0,0,0,0,   TAT FLAG            211-04
083E' 003D  5613A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5615+  DRPT  18,0,0,0,0,0,0,   4 BARO ALT =3       271-04
0840' 0011  5646A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5648+  DRPT  17,0,0,0,0,0,0,   4 BARO ALT =2       271-04
0842' 0010  5679A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5681+  DRPT  16,0,0,0,0,0,0,   4 BARO ALT =1       271-04
0844' 000F  5712A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5714+  DRPT  12,0,0,0,0,0,0,   EXT AOA MON (FAIL)  271-04
0846' 000B  5745A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5747+  DRPT  11,0,0,0,0,0,0,   ZERO SSEC (AOA)     271-04
0848' 000A  5778A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5780+  DRPT  29,0,0,0,0,0,0,   F/D CONVERSION TEST 350-04
084A' 001C  5811A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5813+  DRPT  28,0,0,0,0,0,0,   PS CALIB TEST       350-04
084C' 001B  5844A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5846+  DRPT  27,0,0,0,0,0,0,   PS SENS TEST        350-04
084E' 001A  5877A            DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5879+  DRPT  26,0,0,0,0,0,0,   PS SENS PER TEST    350-04
```

```
0850' 0019  5910A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5912+  DRPT    25,0,0,0,0,0,0,                PROG MEN TEST'     350-04
0892' 0018  5943A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5945+  DRPT    24,0,0,0,0,0,0,                RAM TEST           350-04
08D4' 0017  5976A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            5978+  DRPT    23,0,0,0,0,0,0,                PROCESSOR TEST     350-04
0916' 0016  6009A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6011+  DRPT    22,0,0,0,0,0,0,                A TO D TEST        350-04
0958' 0015  6042A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6044+  DRPT    21,0,0,0,0,0,0,                OSPD HW TEST       350-04
099A' 0014  6075A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6077+  DRPT    20,0,0,0,0,0,0,                A/C TYPE CONST TEST 350-04
09DC' 0013  6108A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6110+  DRPT    19,0,0,0,0,0,0,                A/C TYPE PROG TEST 350-04
0A1E' 0012  6141A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6143+  DRPT    18,0,0,0,0,0,0,                BARO 3 TEST        350-04
0A60' 0011  6174A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6176+  DRPT    17,0,0,0,0,0,0,                BARO 2 TEST        350-04
0AA2' 0010  6207A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6209+  DRPT    16,0,0,0,0,0,0,                BARO 1 TEST        350-04
0AE4' 000F  6240A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6242+  DRPT    15,0,0,0,0,0,0,                TAT INPUT TEST     350-04
0B26' 000E  6273A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6275+  DRPT    12,0,0,0,0,0,0,                RADA VANE TEST     350-04
0B68' 000B  6306A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6308+  DRPT    11,0,0,0,0,0,0,                LADA VANE TEST     350-04
0BAA' 000A  6339A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6341+  DRPT    27,0,0,0,0,0,0,                VMO TEST           351-04
0BEC' 001A  6372A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6374+  DRPT    26,0,0,0,0,0,0,                BARO 4 TEST        351-04
0C2E' 0019  6405A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6407+  DRPT    25,0,0,0,0,0,0,                EAROM TEST         351-04
0C70' 0018  6438A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6440+  DRPT    24,0,0,0,0,0,0,                PT PLL             351-04
0CB2' 0017  6471A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6473+  DRPT    23,0,0,0,0,0,0,                PS PLL             351-04
0CF4' 0016  6504A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6506+  DRPT    22,0,0,0,0,0,0,                PROG SEQ TEST      351-04
0D36' 0015  6537A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6539+  DRPT    21,0,0,0,0,0,0,                TEMP PS=TEMP PT    351-04
0D78' 0014  6570A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6572+  DRPT    20,0,0,0,0,0,0,                AVGE ADA TEST      351-04
0DBA' 0013  6603A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6605+  DRPT    19,0,0,0,0,0,0,                PS = PT            351-04
0DFC' 0012  6636A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6638+  DRPT    18,0,0,0,0,0,0,                ADA COMPARE TEST   351-04
0E3E' 0011  6669A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6671+  DRPT    17,0,0,0,0,0,0,                POWER SUPPLY TEST  351-04
0E80' 0010  6702A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6704+  DRPT    16,0,0,0,0,0,0,                ARINC XMTR TEST    351-04
0EC2' 000F  6735A          DATA   XX1+XX2+XX3+XX4+XX5+XX6
            6737+  DRPT    15,0,0,0,0,0,0,                ARINC XMTR TEST    351-04
```

```
0B74'  000E    6768A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6770+        DRPT    14,0,0,0,0,0,0,                ARINC XMTR TEST     351-04
0B86'  000D    6801A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6803+        DRPT    13,0,0,0,0,0,0,                PT CALIB TEST       351-04
0B98'  000C    6834A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6836+        DRPT    12,0,0,0,0,0,0,                PT SENS TEMP TEST   351-04
0BAA'  000B    6867A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6869+        DRPT    11,0,0,0,0,0,0,                PT SENS PER TEST    351-04
0BBC'  000A    6900A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6902+        DRPT    29,0,0,0,0,0,0,                A/C TYPE PARITY     352-04
0BCE'  001C    6933A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6935+        DRPT    28,0,0,0,0,0,0,                A/C TYPE MSB        352-04
0BE0'  001B    6966A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               6968+        DRPT    27,0,0,0,0,0,0,                A/C TYPE LSB+3      352-04
0BF2'  001A    6999A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7001+        DRPT    26,0,0,0,0,0,0,                A/C TYPE LSB+2      352-04
0C04'  0019    7032A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7034+        DRPT    25,0,0,0,0,0,0,                A/C TYPE LSB+1      352-04
0C16'  0018    7065A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7067+        DRPT    24,0,0,0,0,0,0,                A/C TYPE LSB        352-04
0C28'  0017    7098A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7100+        DRPT    21,0,0,0,0,0,0,                SPARE               352-04
0C3A'  0014    7131A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7133+        DRPT    20,0,0,0,0,0,0,                SPARE               352-04
0C4C'  0013    7164A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7166+        DRPT    19,0,0,0,0,0,0,                SPARE               352-04
0C5E'  0012    7197A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7199+        DRPT    18,0,0,0,0,0,0,                SPARE               352-04
0C70'  0011    7230A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7232
               7233
               7234  *      CHANNEL NUMBER  4
               7235
               7236
               7237+        DRPT    20,0,0,0,0,0,0,                TERRAIN PULL UP     270-04
0C82'  0013    7268A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7270+        DRPT    19,0,0,0,0,0,0,                MINIMUMS            270-04
0C94'  0012    7301A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7303+        DRPT    18,0,0,0,0,0,0,                GLIDE SLOPE         270-04
0CA6'  0011    7334A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7336+        DRPT    17,0,0,0,0,0,0,                TOO LOW TERRAIN     270-04
0CB8'  0010    7367A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7369+        DRPT    16,0,0,0,0,0,0,                TOO LOW FLAPS       270-04
0CCA'  000F    7400A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7402+        DRPT    15,0,0,0,0,0,0,                TOO LOW GEAR        270-04
0CDC'  000E    7433A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7435+        DRPT    14,0,0,0,0,0,0,                DON'T SINK          270-04
0CEE'  000D    7466A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7468+        DRPT    13,0,0,0,0,0,0,                TERRAIN             270-04
0D00'  000C    7499A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
               7501+        DRPT    12,0,0,0,0,0,0,                PULL UP             270-04
0D12'  000B    7532A                DATA   XX1+XX2+XX3+XX4+XX5+XX6
```

```
                    7534+       DRPT    11,0,0,0,0,0,0,                 SINK RATE           270-04
     0ED4'  000A    7585A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7587
                    7588
                    7569  *     CHANNEL NUMBER  5
                    7570
                    7571+       DRPT    19,9,1,0,0,1,1,                  PITCH ATTITUDE      324-04
     0ED9'  3332    7602A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7604+       DRPT    19,9,1,0,0,1,1,                  ROLL ATTITUDE       325-04
     0EDB'  3332    7635A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7637+       DRPT    19,9,1,0,0,1,1,                  LOCALIZER DEV       173-04
     0EDE'  3332    7668A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7670+       DRPT    19,9,1,0,0,1,1,                  GLIDE SLOPE DEV     174-04
     0EE0'  3332    7701A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7703+       DRPT    17,10,1,0,0,1,1,                 RADIO ALT           164-04
     0EE3'  3350    7734A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7736+       DRPT    19,9,1,0,0,1,1,                  MAGN HEADING        320-04
     0EE6'  3332    7767A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7769+       DRPT    19,9,1,0,0,1,1,                  TRACK ANGLE MAG     317-04
     0EE8'  3332    7800A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7802+       DRPT    19,9,1,0,0,1,1,                  TRACK ANGLE TRUE    313-04
     0EEA'  3332    7833A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7835+       DRPT    19,9,1,0,0,1,1,                  TRUE HEADING        314-04
     0EED'  3332    7866A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7868+       DRPT    19,0,0,0,0,0,0,                  MAG TRUE DATA       270-04
     0EEF'  0012    7899A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7901+       DRPT    14,0,0,0,0,0,0,                  DH ALERT            270-04
     0EF2'  000D    7932A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7934+       DRPT    16,0,0,0,0,0,0,                  H ALERT             270-04
     0EF4'  000F    7965A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    7967+       DRPT    17,0,0,0,0,0,0,                  GROUNDSPEED SOURCE  270-04
     0EF7'  0010    7998A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8000+       DRPT    18,0,0,0,0,0,0,                  TRACK ANGLE SOURCE  270-04
     0EF9'  0011    8031A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8033+       DRPT    15,0,0,0,0,0,0,                  DH + H ALERT        270-04
     0EFC'  000E    8064A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8066+       DRPT    29,0,0,0,0,0,0,                  G/S MODE OPER       274-04
     0EFE'  001C    8097A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8099+       DRPT    28,0,0,0,0,0,0,                  FLARE OPER          274-04
     0F01'  001B    8130A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8132+       DRPT    27,0,0,0,0,0,0,                  G/A MODE OPER-P     274-04
     0F03'  001A    8163A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8165+       DRPT    26,0,0,0,0,0,0,                  T/O MODE OPER-P     274-04
     0F06'  0019    8196A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8198+       DRPT    25,0,0,0,0,0,0,                  ALT HOLD MODE OPER  274-04
     0F08'  0018    8229A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8231+       DRPT    24,0,0,0,0,0,0,                  V/S MODE OPER       274-04
     0F0B'  0017    8262A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8264+       DRPT    23,0,0,0,0,0,0,                  V/NAV MODE OPER     274-04
     0F0D'  0016    8295A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8297+       DRPT    22,0,0,0,0,0,0,                  FL CH MODE OPER     274-04
     0F10'  0015    8328A               DATA    XX1+XX2+XX3+XX4+XX5+XX6
```

```
                    8330+   DRPT    21,0,0,0,0,0,0,                 THROTTLE RETARD     274-04
06E3' 0014          8361A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8363+   DRPT    20,0,0,0,0,0,0,                 IAS MODE OPER       274-04
06E6' 0013          8394A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8396+   DRPT    19,0,0,0,0,0,0,                 MACH MODE SET       274-04
06E9' 0012          8427A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8429+   DRPT    11,0,0,0,0,0,0,                 ALT MODE OPER       274-04
06EA' 000A          8460A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8462+   DRPT    10,0,0,0,0,0,0,                 PITCH SPEED CNTRL   274-04
06EC' 0009          8493A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8495+   DRPT    29,0,0,0,0,0,0,                 WXR DATA FAULT      352-04
06EE' 001C          8526A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8528+   DRPT    28,0,0,0,0,0,0,                 MLS DATA FAULT      352-04
06F0' 001B          8559A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8561+   DRPT    27,0,0,0,0,0,0,                 ILS DATA FAULT      352-04
06F2' 001A          8592A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8594+   DRPT    26,0,0,0,0,0,0,                 RA DATA FAULT       352-04
06F4' 0019          8625A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8627+   DRPT    25,0,0,0,0,0,0,                 D-DME DATA FAULT    352-04
06F6' 0018          8658A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8660+   DRPT    24,0,0,0,0,0,0,                 L-DME DATA FAULT    352-04
06F8' 0017          8691A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8693+   DRPT    23,0,0,0,0,0,0,                 R-VOR DATA FAULT    352-04
06FA' 0016          8724A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8726+   DRPT    22,0,0,0,0,0,0,                 L-VOR DATA FAULT    352-04
06FC' 0015          8757A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8759+   DRPT    21,0,0,0,0,0,0,                 R-ADC DATA FAULT    352-04
06FE' 0014          8790A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8792+   DRPT    20,0,0,0,0,0,0,                 L-ADC DATA FAULT    352-04
0700' 0013          8823A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8825+   DRPT    19,0,0,0,0,0,0,                 R-IRS DATA FAULT    352-04
0702' 0012          8856A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8858+   DRPT    18,0,0,0,0,0,0,                 C-IRS DATA FAULT    352-04
0704' 0011          8889A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8891+   DRPT    17,0,0,0,0,0,0,                 L-IRS DATA FAULT    352-04
0706' 0010          8922A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8924+   DRPT    16,0,0,0,0,0,0,                 TMC DATA FAULT      352-04
0708' 000F          8955A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8957+   DRPT    15,0,0,0,0,0,0,                 R-FMC DATA FAULT    352-04
070A' 000E          8988A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    8990+   DRPT    14,0,0,0,0,0,0,                 L-FMC DATA FAULT    352-04
070C' 000D          9021A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9023+   DRPT    13,0,0,0,0,0,0,                 R-FCC DATAS FAULT   352-04
070E' 000C          9054A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9056+   DRPT    12,0,0,0,0,0,0,                 C-FCC DATA FAULT    352-04
0710' 000B          9087A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9089+   DRPT    11,0,0,0,0,0,0,                 L-FCC DATA FAULT    352-04
0712' 000A          9120A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9122+   DRPT    18,0,0,0,0,0,0,                 SG FAULT            353-04
0714' 0011          9153A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9155+   DRPT    17,0,0,0,0,0,0,                 R EHSI FAULT        353-04
0716' 0010          9186A           DATA    XX1+XX2+XX3+XX4+XX5+XX6
```

```
                    9188+   DRPT   16,0,0,0,0,0,0,                L EHSI FAULT        353-04
0919' 000F          9219A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9221+   DRPT   15,0,0,0,0,0,0,                R CP FAULT          353-04
091A' 000E          9252A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9254+   DRPT   14,0,0,0,0,0,0,                L CP FAULT          353-04
091C' 000D          9285A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9287+   DRPT   12,0,0,0,0,0,0,                R EADI FAULT        353-04
091E' 000B          9318A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9320+   DRPT   11,0,0,0,0,0,0,                L EADI FAULT        353-04
0920' 000A          9351A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9353+   DRPT   20,0,0,0,0,0,0,                SG I/O PROC 3       350-04
0922' 0013          9384A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9386+   DRPT   19,0,0,0,0,0,0,                SG I/O PROC 2       350-04
0924' 0012          9417A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9419+   DRPT   18,0,0,0,0,0,0,                SG I/O PROC 1       350-04
0926' 0011          9450A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9452+   DRPT   17,0,0,0,0,0,0,                SG DIGITAL O/P      350-04
0928' 0010          9483A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9485+   DRPT   16,0,0,0,0,0,0,                SG CONTROLLER       350-04
092A' 000F          9516A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9518+   DRPT   15,0,0,0,0,0,0,                SG DSPLY SEQCER     350-04
092C' 000E          9549A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9551+   DRPT   14,0,0,0,0,0,0,                SG DSPLY DRIVE      350-04
092E' 000D          9582A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9584+   DRPT   13,0,0,0,0,0,0,                SG MAIN MEMORY      350-04
0930' 000C          9615A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9617+   DRPT   12,0,0,0,0,0,0,                SG MAIN PROCESSOR   350-04
0932' 000B          9648A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9650+   DRPT   11,0,0,0,0,0,0,                SG OVERTEMP         350-04
0934' 000A          9681A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9683+   DRPT   25,0,0,0,0,0,0,                R-CP FAULT          351-04
0936' 0018          9714A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9716+   DRPT   24,0,0,0,0,0,0,                L-CP FAULT          351-04
0938' 0017          9747A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9749+   DRPT   23,0,0,0,0,0,0,                R-EHSI OVERTEMP     351-04
093A' 0016          9780A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9782+   DRPT   22,0,0,0,0,0,0,                R-EHSI BEAM FAIL    351-04
093C' 0015          9813A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9815+   DRPT   21,0,0,0,0,0,0,                R-EHSI ANOMALIES    351-04
093E' 0014          9846A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9848+   DRPT   20,0,0,0,0,0,0,                L-EHSI OVERTEMP     351-04
0940' 0013          9879A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9881+   DRPT   19,0,0,0,0,0,0,                L-EHSI BEAM FAIL    351-04
0942' 0012          9912A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9914+   DRPT   18,0,0,0,0,0,0,                L-EHSI ANOMALIES    351-04
0944' 0011          9945A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9947+   DRPT   17,0,0,0,0,0,0,                R-EADI OVERTEMP     351-04
0946' 0010          9978A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    9980+   DRPT   16,0,0,0,0,0,0,                R-EADI BEAM FAIL    351-04
0948' 000F         10011A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
                   10013+   DRPT   15,0,0,0,0,0,0,                R-EADI ANOMALIES    351-04
094A' 000E         10044A          DATA    XX1+XX2+XX3+XX4+XX5+XX6
```

```
                    10046+          DRPT    14,0,0,0,0,0,0,                 L-EADI OVERTEMP      351-04
 0940'  000D        10077A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10079+          DRPT    13,0,0,0,0,0,0,                 L-EADI BEAM FAIL     351-04
 0948'  000C        10110A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10112+          DRPT    12,0,0,0,0,0,0,                 L-EADI ANOMALIES     351-04
 0950'  000B        10143A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10145
                    10146
                    10147   *       CHANNEL NUMBER  6
                    10148
                    10149
                    10150+          DRPT    12,5,0,0,0,1,1,                 GMT SECONDS          150-04
 0958'  30AB        10181A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10183+          DRPT    18,5,0,0,0,1,1,                 GMT MINUTES          150-04
 0960'  30B1        10214A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10216+          DRPT    24,4,0,0,0,1,1,                 GMT HOURS            150-04
 0968'  3097        10247A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10249   *
                    10250+          DRPT    18,0,0,0,0,0,0,          ***** DUMMY *****
 0970'  0011        10281A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10283   *
                    10284   ***  ADDITIONS  ****
                    10285
                    10286   *       CHANNEL NUMBER  1
                    10287
                    10288+          DRPT    17,11,0,0,0,1,1,                N1 MAX/EPR LIMIT     342-04
 0978'  3170        10319A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10321
                    10322   *       CHANNEL NUMBER  2
                    10323
                    10324+          DRPT    17,11,0,0,0,1,1,                N1 MAX/EPR LIMIT     342-04
 0980'  3170        10355A                  DATA    XX1+XX2+XX3+XX4+XX5+XX6
                    10357   *
                    10358   ********************************************************************************
                    10359   *
                    10360   *       DITS : DESTINATION OFSET TABLE
                    10361   *
                    10362   *               ************************
                    10363   ***************  P & W ENGINES SECTION ******************
                    10364   *               ************************
                    10365   *
 0988'              10366   DR1DAP
                    10367   *       SUBFRAME 1 CYCLES 1 TO 8
                    10368   *
 0988'  0118        10369           DATA    280             DUMMY
 098A'  0118        10370           DATA    280             DUMMY
 098C'  0141 0143   10371           DATA    321,323
 098E'  0118        10372           DATA    280             DUMMY
 0990'  0016 0059   10373           DATA    22,89
 0994'  00FC        10374           DATA    252
 0996'  0118        10375           DATA    280             DUMMY
 0998'  0118        10376           DATA    280             DUMMY
```

```
                        10377   *
                        10378   *       SUBFRAME 2 CYCLES 1 TO 8
                        10379   *
     0972'  0118        10380           DATA    280             DUMMY
     0974'  0118        10381           DATA    280             DUMMY
     0976'  0145        10382           DATA    325
     0978'  0118        10383           DATA    280             DUMMY
     097A'  0018        10384           DATA    24
     097C'  00F9        10385           DATA    249
     097E'  0118        10386           DATA    280             DUMMY
     0980'  0118        10387           DATA    280             DUMMY
                        10388   *
                        10389   *       SUBFRAME 3 CYCLES 1 TO 8
                        10390   *
     0982'  0118        10391           DATA    280             DUMMY
     0984'  0118        10392           DATA    280             DUMMY
     0986'  0118        10393           DATA    280             DUMMY
     0988'  0118        10394           DATA    280             DUMMY
     098A'  005A        10395           DATA    90
     098C'  00FD        10396           DATA    253
     098E'  0079 007B   10397           DATA    121,123,125,127,129,131,133,135,137,139,141,44
     0992'  007D 007F
     0996'  0081 0083
     099A'  0085 0087
     099E'  0089 008B
     09A2'  008D 002C
     09A6'  0118        10398           DATA    280             DUMMY
                        10399   *
                        10400   *       SUBFRAME 4 CYCLES 1 TO 8
                        10401   *
     09AA'  0118        10402           DATA    280             DUMMY
     09AC'  0118        10403           DATA    280             DUMMY
     09AE'  0118        10404           DATA    280             DUMMY
     09B0'  0118        10405           DATA    280             DUMMY
     09B2'  0118        10406           DATA    280             DUMMY
     09B4'  00FA        10407           DATA    250
     09B6'  001C        10408           DATA    28
     09B8'  0118        10409           DATA    280             DUMMY
                        10410   *
     09BA'              10411   DR1DBP
                        10412   *
                        10413   *       ALL SUBFRAMES CYCLES 1 TO 8
                        10414   *
     09BC'  0051 00E1   10415   DPCYC1  DATA    81,225,239,2,3,108,109,110,111,112,113,114,115
     09D0'  00EF 0002
     09D4'  0003 006C
     09D8'  006D 006E
     09DC'  006F 0070
     09E0'  0071 0072
     09E4'  0073
     09E6'  0074 0075   10416           DATA    116,117,118,119,120,122,124,126,128,130,132,318,328,329
     09EA'  0076 0077
```

```
                        53          EXTERN  TIMEL
                        54          EXTERN  WPJB1
                        55  *** CONSTANTS REFERENCED
                        56          EXTERN  B1
                        57          EXTERN  D15
                        58  *** TABLES REFERENCED
                        59  *** MODULES REFERENCED
                        60          EXTERN  EAWRIT
                        61  *** LIBRARY
                        62          INCLUDE CNSTNT
                       190  *** REGISTERS DEFINITION
        =0006          191  EATMR   EQU     R6      =       EAROM TIMER
        =0007          192  EAWDP   EQU     R7      =       EAROM DATA WORD POINTER
        =0008          193  EAADR   EQU     R8      =       EAROM ADDRESS
        =0009          194  EACNT   EQU     R9      =       EAROM CONTROL REG IMAGE
        =000A          195  EAMODE  EQU     R10     =       EAROM MODE
                       196  ********************************************************************************
                       197  *
0000'  0008*           198  EAMON   DATA    WPJB1
0002'  0004'           199          DATA    1$
0004'  C28A            200  1$      MOV     EAMODE,EAMODE           EAROM MODE?
0006'  1634            201          JNE     50$                     JIF EAROM BUSY
                       202  *** EAROM NOT BUSY (AVAILABLE)
                       203  *** CHECK IF ANY NON-ZERO DOC DATA ITEMS IN FDETBF
0008'  C020 0001*      204          MOV     @CLRFLG,R0              CHECK CLEAR FLAG
000C'  130F            205          JEQ     19$                     JIF NOT SET
000E'  8820 0006*      206          C       @EASADR,@D15            CHECK IF ALL 16 ADDRESSES WRITTEN INTO
0012'  000A*
0014'  1B07            207          JH      18$                     JIF
0016'  0200 0006*      208          LI      R0,EASADR               EAROM ADDRESS
001A'  0201 0002*      209          LI      R1,D0                   ZERO EAROM DATA
001E'  06A0 000B*      210          BL      @EAWRIT                 WRITE ZEROS INTO EAROM
0022'  1044            211          JMP     100$                    EXIT
0024'  04E0 0001*      212  18$     CLR     @CLRFLG                 CLEAR FLAG
0028'  04E0 0006*      213          CLR     @EASADR                 CLEAR EAROM ADDRESS
                       214  *** WRITE SYSTEM ERROR CODE IN EAROM IF
                       215  *** ERROR CODE FOUND IN TEMPORARY EAROM SYSTEM STATUS BUFFER (EATSBF)
002C'  0201 001E       216  19$     LI      R1,15*2                 16 WORDS IN EATSBF
0030'  C021 0005*      217  10$     MOV     @EATSBF(R1),R0  CHECK ERROR CODE IN TEMP EAROM S.S. BUFFER
0034'  1603            218          JNE     12$                     JIF FOUND ERROR CODE
0036'  0641            219          DECT    R1
0038'  18FB            220          JOC     10$                     LOOP 16 TIMES
003A'  1038            221          JMP     100$                    EXIT, NO ERROR CODE
                       222  *  FOUND ERROR CODE, CHECK IF 16 ERROR CODES HAVE
                       223  *  BEEN WRITTEN INTO EAROM.
003C'  8820 0006*      224  12$     C       @EASADR,@D15
0040'  000A*
0042'  1B34            225          JH      100$                    EXIT IF ALREADY 16 ERRORS IN EAROM
0044'  0280 0001       226          CI      R0,>0001                CHECK IF ERROR CODE IS 0001
0048'  1304            227          JEQ     25$                     JIF YES
004A'  0280 0102       228          CI      R0,>0102                CHECK IF ERROR CODE IS 0102
004E'  1301            229          JEQ     25$                     JIF YES
```

```
                    1          IDT    EAMON
                    2          SUBTTL EAROM MONITIOR
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: EAMON.SRC                        AUTH: N.COSTANTINIDES    *
                    6  * VERSION: 2                             DATE: 5-MAY-1983         *
                    7  *                                                                  *
                    8  * FUNCTION:     EAROM MONITOR.                                     *
                    9  *               THIS ROUTINE MONITORS THE ACTIVITIES OF            *
                   10  *               THE EAROM TASKS. IT MAKES SURE THAT A MINIMUM      *
                   11  *               OF 250 MSEC HAS ELAPSED SINCE THE ISSUING OF       *
                   12  *               THE 'ERASE' COMMAND, ENABLED BY THE EAWRIT         *
                   13  *               ROUTINE. WHEN THE 250 MSEC HAS ELAPSED FOR THE     *
                   14  *               ERASE MODE, A 'WRITE' MODE WILL BE ENTERED         *
                   15  *               ALSO FOR A PERIOD OF 250 MSEC.                     *
                   16  *               UPON COMPLETION OF THE 250 WRITEMSEC TIME PERIOD   *
                   17  *               THE H/W LOGIC HAS WRITTEN EAROM DATA INTO          *
                   18  *               EAROM MEMORY. THE SAME EAROM DATA WILL BE STORED   *
                   19  *               IN THE EAROM IMAGE BUFFER AND THE EAROM MODE       *
                   20  *               WILL BESET TO INACTIVE STATE                       *
                   21  *               ITERATION RATE = 4 SPS                             *
                   22  *                                                                  *
                   23  * CALLING MODULES: JOB4PS                                         *
                   24  *                                                                  *
                   25  * CALLING SEQ: BLWP @EAMON                                        *
                   26  *                                                                  *
                   27  * INPUTS:       R6      =       EAROM TIMER                       *
                   28  *               R7      =       EAROM WRITE DATA WORD POINTER     *
                   29  *               R9      =       EAROM CONTROL REGISTER IMAGE      *
                   30  *               R10     =       EAROM MODE                        *
                   31  *                                                                  *
                   32  * OUTPUTS:      R6      =       EAROM TIMER = LSW TIMER + 250MS   *
                   33  *               R9      =       EAROM CONTROL REGISTER IMAGE      *
                   34  *               R10     =       EAROM MODE                        *
                   35  *          EASADR     =       SYSTEM STATUS EAROM ADDR WORD     *
                   36  *                                                                  *
                   37  * MODULES REFERENCED: EAWRIT                                      *
                   38  *                                                                  *
                   39  * WORKSPACE AREA: WPJB1                                           *
                   40  *                                                                  *
                   41  * REGISTERS MODIFIED: R0,R1,R3,R6,R7,R9,R10                       *
                   42  *                                                                  *
                   43  * VERSION HISTORY:                                                *
                   44  *                                                                  *
                   45  ********************************************************************
=0000              46         RSECT  EAMON
                   47  *** CALL NAME
                   48         INTERN EAMON
                   49  *** VARIABLES REFERENCED
                   50         EXTERN CLRFLG
                   51         EXTERN B0
                   52         EXTERN EATC,EAIMBF,EATSBF,EASADR
```

```
0006' 1311          53          JEQ     100$                JIF NOT SET
0008' C020 0002*    54          MOV     @EABTF,R0           GET DISCRETE FROM DFDR BUFFER
000C' 04E0 0002*    55          CLR     @EABTF              CLEAR DISCRETE BIT
0010' C060 0003*    56          MOV     @EATBUF,R1          GET PREVIOUS VALUE OF DISCRETE
0014' 8040          57          C       R0,R1               COMPARE
0016' 1505          58          JGT     10$                 JIF NO 1 TO 0 TRANSITION
0018' 1304          59          JEQ     10$
001A' 0720 0001*    60          SETO    @CLRFLG             SET THE EAROM CLEAR FLAG
001E' 04E0 0004*    61          CLR     @EASADR             CLEAR THE EAROMADDRESS COUNTER
                    62  *
0022' C800 0003*    63  10$     MOV     R0,@EATBUF          SAVE LATEST VALUE
                    64  *
0026' 04E0 0005*    65          CLR     @EAFLG              CLEAR FLAG
002A'               66  100$
002A' C2C3          67          MOV     R3,LINK             UNSAVE LINK
                    68  *
002C' 045B          69          RT
                    70  ********************************************************************************
                    71          END
```

No errors detected

```
                        1          IDT     EABTST
                        2          SUBTTL  TEST EAROM CLEAR DISCRETE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: EABTST.SRC                      AUTH: N. CONSTANTINIDES *
                        6  * VERSION: 1                            DATE: 3-MAR-1983        *
                        7  *                                                                  *
                        8  * FUNCTION:     THIS MODULE WILL TEST THE EAROM ERASE DISCRETE  *
                        9  *               (53). IF THE DISCRETE CHANGES FROM LOGIC 1 TO 0 *
                       10  *               IT WILL SET A FLAG AND CLEAR THE EAROM ADDRESS  *
                       11  *               COUNTER.                                        *
                       12  *               ACQUIRED AT A RATE OF 4SPS AND STORED IN WORD   *
                       13  *               68 OF THE DFDR BUFFER.                          *
                       14  *                                                               *
                       15  * CALLING MODULES: JOB4PS                                      *
                       16  *                                                               *
                       17  * CALLING SEQ: BL @EABTST                                      *
                       18  *                                                               *
                       19  * INPUTS: SHUNT DISCRETE PORT #53                              *
                       20  *               EATBUF  =       OLD VALUE OF DIC #53           *
                       21  *               EABTF   =       DISCRETE #53                   *
                       22  *                                                               *
                       23  * OUTPUTS:      CLRFLG  =       EAROM CLEAR FLAG               *
                       24  *               EASADR  =       EAROM ADDRESS COUNTER          *
                       25  *               EATBUF  =       NEW VALUE OF DISC #53          *
                       26  *                                                               *
                       27  * MODULES REFERENCED: NONE                                     *
                       28  *                                                               *
                       29  * WORKSPACE AREA: CALLER'S                                     *
                       30  *                                                               *
                       31  * REGISTERS MODIFIED: R0,R1                                    *
                       32  *                                                               *
                       33  * VERSION HISTORY:                                             *
                       34  *                                                               *
                       35  ********************************************************************
          =0000        36         RSECT   EABTST
                       37  *** CALL NAME
                       38         INTERN  EABTST
                       39  *** VARIABLES REFERENCED
                       40         EXTERN  CLRFLG
                       41         EXTERN  EABTF,EATBUF,EASADR,EAFLG
                       42  *** CONSTANTS REFERENCED
                       43  *** TABLES REFERENCED
                       44  *** MODULES REFERENCED
                       45  *** LIBRARY
                       46  *** REGISTERS DEFINITION
          =000B        47  LINK    EQU     R11
                       48  ********************************************************************
                       49  *
    0000'              50  EABTST
    0000'  C0CB        51         MOV     LINK,R3              SAVE LINK
    0002'  C020 0005*  52         MOV     @EAFLG,R0            CHECK FLAG
```

```
000A' 1305          71          JEQ     20$                 JIF NOT FAILURE
                    72  *** DFDR BITE ON - FAILURE
000C' 06A0 0003*    73          BL      @SYSER              SET ERROR BIT IN SYSTEM ERROR BUFFER
0010' 04E0 0001*    74          CLR     @DFBTF              CLEAR DFDR BITE DISCRETE WORD
0014' 1002          75          JMP     40$                 EXIT
                    76  *** DFDR BITE OFF - PASSED
0016'               77  20$
0016' 06A0 0002*    78          BL      @SYSOK              RESET ERROR BIT IN SYSTEM ERROR BUFFER
                    79  ***
001A' C2C3          80  40$     MOV     R3,LINK             UNSAVE LINK
001C' 045B          81          RT
                    82  ***
001E' 0001          83  ERCOD   DATA    001                 DFDR ERROR BITE CODE
                    84  ********************************************************************************
                    85          END
```

```
                     1          IDT     DFBTST
                     2          SUBTTL  TEST DFDR BITE
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: DFBTST                          AUTH: N.COSTANTINIDES     *
                     6  * VERSION:                              DATE: 25-MAY-1982         *
                     7  *                                                                  *
                     8  * FUNCTION:     THIS SUBROUTINE TESTS THE DFDR BITE DISCRETE      *
                     9  *               WHICH IS ACQUIRED BY THE DMX #1 ACQUISITION       *
                    10  *               MODULE AT A RATE OF 4 SPS AND STORED IN 67TH      *
                    11  *               WORD OF THE DFDR OUTPUT BUFFER.FOR INT USE ONLY.  *
                    12  *               IF THE BIT IS ON, (FAIL), DFDR ERROR BIT          *
                    13  *               ( 1ST BIT ) WILL BE SET IN THE FIRST WORD OF      *
                    14  *               THE SYSTEM ERROR BUFFER (SYEBF).                  *
                    15  *               IF THE BIT IS OFF (OK), THE DFDR ERROR BIT        *
                    16  *               WILL BE RESET.                                    *
                    17  *                                                                  *
                    18  * CALLING MODULES: JOB4PS                                         *
                    19  *                                                                  *
                    20  * CALLING SEQ: BL @DFBTST                                         *
                    21  *                                                                  *
                    22  * INPUTS: DFBTF = DFDR BITE DISCRETE BIT                          *
                    23  *                                                                  *
                    24  * OUTPUTS: 1ST WORD 1ST BIT OF SYEBF = 1 IF FAILED                *
                    25  *          "   "   "   "   "   "     = 0 IF PASSED                *
                    26  *                                                                  *
                    27  * MODULES REFERENCED: SYSER,SYSOK                                 *
                    28  *                                                                  *
                    29  * WORKSPACE AREA: CALLERS                                         *
                    30  *                                                                  *
                    31  * REGISTERS MODIFIED: R0,R1                                       *
                    32  *                                                                  *
                    33  * VERSION HISTORY:                                                *
                    34  *                                                                  *
                    35  ********************************************************************
=0000               36         RSECT   DFBTST
                    37  *** CALL NAME
                    38         INTERN  DFBTST
                    39  *** VARIABLES REFERENCED
                    40         EXTERN  DFBTF
                    41  *** CONSTANTS REFERENCED
                    42  *** TABLES REFERENCED
                    43  *** MODULES REFERENCED
                    44         EXTERN  SYSOK,SYSER
                    45  *** LIBRARY
                    46         INCLUDE REGDEF
                    65  *** REGISTERS DEFINITION
                    66  ********************************************************************
                    67  *
0000' C0CB          68  DFBTST  MOV     LINK,R3             SAVE LINK
0002' C060 001E'    69         MOV     @ERCOD,R1           DFDR BITE ERROR CODE
0006' C020 0001*    70         MOV     @DFBTF,R0           DFDR BITE DISCRETE WORD
```

```
                    1393
                    1394   * Dits wraparound error codes
                    1395
0084' 0402 0403     1396   DWECT  DATA   )402,)403,)404,)405,)406,)407              DITS 1
0088' 0404 0405
008C' 0406 0407
0090' 040A 040B     1397          DATA   )40A,)40B,)40C,)40D                        DITS 2
0094' 040C 040D
0098' 0411 0412     1398          DATA   )411,)412,)413,)414,)415,)416,)417,)418   DITS 3
009C' 0413 0414
00A0' 0415 0416
00A4' 0417 0418
                    1399
00A8' 0000          1400   DR1SP  DATA   0              CHAN 5 = HI
00AA' 0018          1401   DR2SP  DATA   )18            CHAN 3 = HI
                    1402
                    1403+         LOCR   PRIV,LINKZ
0000' =0002         1406A  LINKZ   BSS    2
                    1407          END
```

```
001E' 0500 607F   1358          DATA    )0500,)607F,)7AAA       CHAN 6 - PARAM 254
0022' 7AAA
                  1359
                  1360  * Dits 2 test data table
                  1361
0024'             1362  D2WAT
0024' 0540 107F   1363          DATA    )0540,)107F,)7AAA       CHAN 1 - PARAM 249
0028' 7AAA
002A' 0500 207F   1364          DATA    )0500,)207F,)7AAA       CHAN 2 - PARAM 250
002E' 7AAA
0030' 0540 307F   1365          DATA    )0540,)307F,)7AAA       CHAN 3 - PARAM 251
0034' 7AAA
0036' 0500 407F   1366          DATA    )0500,)407F,)7AAA       CHAN 4 - PARAM 252
003A' 7AAA
                  1367
                  1368  * Dits 3 test data table
                  1369
003C' 0500 007F   1370  D3WAT   DATA    )0500,)007F,)7AAA       CHAN 0 - PARAM 248
0040' 7AAA
0042' 0540 107F   1371          DATA    )0540,)107F,)7AAA       CHAN 1 - PARAM 249
0046' 7AAA
0048' 0500 207F   1372          DATA    )0500,)207F,)7AAA       CHAN 2 - PARAM 250
004C' 7AAA
004E' 0540 307F   1373          DATA    )0540,)307F,)7AAA       CHAN 3 - PARAM 251
0052' 7AAA
0054' 0500 407F   1374          DATA    )0500,)407F,)7AAA       CHAN 4 - PARAM 252
0058' 7AAA
005A' 0540 507F   1375          DATA    )0540,)507F,)7AAA       CHAN 5 - PARAM 253
005E' 7AAA
0060' 0500 607F   1376          DATA    )0500,)607F,)7AAA       CHAN 6 - PARAM 254
0064' 7AAA
0066' 0540 707F   1377          DATA    )0540,)707F,)7AAA       CHAN 7 - PARAM 255
006A' 7AAA
                  1378
006C'             1379  DWATE
                  1380
006C' 0006        1381  D1WAC   DATA    D2WAT-DWXAT/6           NO. OF TESTS FOR DITS 1
006E' 0004        1382  D2WAC   DATA    D3WAT-D2WAT/6           NO. OF TESTS FOR DITS 2
0070' 0000        1383  D3WAC   DATA    0                       TAKE OUT IF DITS 3 IS USED
                  1384  *       DWATE-D3WAT/6           NO. OF TESTS FOR DITS 3
                  1385
                  1386  * Dits 1, 2 & 3 parameter number tables
                  1387
0072' F9 FA FB    1388  DWPNT   BYTE    249,250,251,252,253,254                 DITS 1
0075' FC FD FE
0078' F9 FA FB    1389          BYTE    249,250,251,252                         DITS 2
007B' FC
007C' F8 F9 FA    1390          BYTE    248,249,250,251,252,253,254,255         DITS 3
007F' FB FC FD
0082' FE FF
                  1391
                  1392          EVEN
```

```
00DE' 0200 000A  1312          LI    R0,10              64 MICROSEC DELAY BETWEEN EACH CLEAR
00E2' 0600       1313  100$    DEC   R0
00E4' 15FE       1314          JGT   100$
00E6' 0607       1315          DEC   R7
00E8' C1C7       1316          MOV   R7,R7
00EA' 15F5       1317          JGT   96$
                 1318
                 1319  *** SET DITS SPEED
                 1320
00EC' C820 00AB' 1321          MOV   @DR1SP,@)FF86      DITS 1
00F0' FF86
00F2' C820 00AA' 1322          MOV   @DR2SP,@)FF8A      DITS 2
00F6' FF8A
                 1323
                 1324  *** RESET WATCHDOG TIMER
                 1325
00F8' 020C 0700  1326          LI    R12,)0700          9901 CRU ADDRESS
00FC' 1D 11      1327          SBO   17
00FE' 1E 11      1328          SBZ   17
                 1329
                 1330  *** RESET AND ENABLE INTERRUPTS
                 1331
0100' C020 FF84  1332          MOV   @)FF84,R0          RESET INT 10
0104' C020 FF88  1333          MOV   @)FF88,R0          RESET INT 11
0108' 020C 0700  1334          LI    R12,)0700
010C' 1D 0A      1335          SBO   10                 ENABLE 10
010E' 1D 0B      1336          SBO   11                 ENABLE 11
                 1337
0110' 0300 000D  1338          LIMI  INTMSK             RE-ENABLE ALL INTERRUPTS
                 1339
0114'            1340  120$
0114' C2E0 0000' 1341          MOV   @LINKI,R11         RESTORE LINK
0118' 045B       1342          RT
                 1343  ********************************************************************
                 1344  * The following tables and constants are used for the DITS wraparound test
                 1345
                 1346+         PRVDAT
                 1348
0000'            1349  DWXAT
                 1350
                 1351  * Dits 1 test data table
                 1352
0000' 0540 107F  1353          DATA  )0540,)107F,)7AAA  CHAN 1 - PARAM 249
0004' 7AAA
0006' 0500 207F  1354          DATA  )0500,)207F,)7AAA  CHAN 2 - PARAM 250
000A' 7AAA
000C' 0540 307F  1355          DATA  )0540,)307F,)7AAA  CHAN 3 - PARAM 251
0010' 7AAA
0012' 0500 407F  1356          DATA  )0500,)407F,)7AAA  CHAN 4 - PARAM 252
0016' 7AAA
0018' 0540 507F  1357          DATA  )0540,)507F,)7AAA  CHAN 5 - PARAM 253
001C' 7AAA
```

```
                    1260  *** READ W/A DATA FROM RECEIVER RAM
                    1261
007E' C174          1262        MOV   *R4+,R5          LSH
0080' C194          1263        MOV   *R4,R6           MSH
0082' 0644          1264        DECT  R4               RESET TO H/W ADDRESS FOR LSH
0084' 06C5          1265        SWPB  R5
0086' 0245 FF00     1266        ANDI  R5,>FF00         IGNORE LABEL
                    1267
                    1268  *** CHECK W/A DATA
                    1269
008A' 0246 7FFF     1270        ANDI  R6,>7FFF         IGNORE PARITY BIT
008E' C032          1271        MOV   *R2+,R0          LSH OF DATA
0090' 0240 FF00     1272        ANDI  R0,>FF00         IGNORE LABEL
0094' 2940          1273        XOR   R0,R5
0096' 29B2          1274        XOR   *R2+,R6
0098' E146          1275        SOC   R6,R5            RESULT IN R5
009A' 1303          1276        JEQ   94$              JIF NO ERROR
                    1277
                    1278  *** FAILED TEST
                    1279
009C' 06A0 0001*    1280        BL    @SYSER           SET ERROR BIT
00A0' 1002          1281        JMP   95$
                    1282
                    1283  *** PASSED TEST
                    1284
00A2' 06A0 0002*    1285  94$   BL    @SYSOK           RESET ERROR BT
                    1286
00A6' 0607          1287  95$   DEC   R7               DECREMENT COUNTER
00A8' 15D5          1288        JGT   93$              JIF NOT FINISHED
00AA' C249          1289        MOV   R9,R9            CHECK IF DITS 2 W/A COMPLETED
00AC' 110A          1290        JLT   97$
                    1291
00AE' C1E0 006E'    1292        MOV   @D2WAC,R7        COUNT=TESTS FOR DITS 2
00B2' 1307          1293        JEQ   97$              JIF NOT USED
00B4' 0204 FF88     1294        LI    R4,>FF88         H/W ADDRESS FOR DITS 2
00B8' 0282 006C'    1295        CI    R2,DWATE
00BC' 130A          1296        JEQ   98$
00BE' 0709          1297        SETO  R9               SET DITS 2 FLAG
00C0' 10C9          1298        JMP   92$
                    1299
00C2' C1E0 0070'    1300  97$   MOV   @D3WAC,R7        COUNT=TESTS FOR DITS 3
00C6' 1305          1301        JEQ   98$              JIF NOT USED
00C8' 0204 FF8C     1302        LI    R4,>FF8C         H/W ADDRESS FOR DITS 3
00CC' 0282 006C'    1303        CI    R2,DWATE
00D0' 16C1          1304        JNE   92$
                    1305
                    1306  *** CLEAR RAM
                    1307
00D2' 0207 0100     1308  98$   LI    R7,>0100
00D6' C807 FF84     1309  96$   MOV   R7,@>FF84         CLEAR DITS 1 RAM
00DA' C807 FF88     1310        MOV   R7,@>FF88                2
                    1311
```

```
0016' C807 FF8A  1208          MOV   R7,@)FF8A              2
001A' C807 FF8E  1209          MOV   R7,@)FF8E              3
                 1210
001E' C807 FF84  1211  91$     MOV   R7,@)FF84        CLEAR DITS 1 RAM
0022' C807 FF88  1212          MOV   R7,@)FF88              2
0026' C807 FF8C  1213          MOV   R7,@)FF8C              3
                 1214
002A' 0200 000A  1215          LI    R0,10            64 MICROSEC DELAY BETWEEN EACH CLEAR
002E' 0600       1216  10$     DEC   R0
0030' 15FE       1217          JGT   10$
0032' 0607       1218          DEC   R7
0034' C1C7       1219          MOV   R7,R7
0036' 15F3       1220          JGT   91$
                 1221
0038' 1000       1222          NOP                    FOR TEST
003A' 1000       1223          NOP
003C' 0300 0001  1224          LIMI  1                DIASABLE ALL I/R EXCEPT PDOWN
0040' C1E0 006C' 1225          MOV   @D1WAC,R7        NO. OF TESTS
0044' 0208 0084' 1226          LI    R8,DWECT         ERROR CODE POINTER
0048' 0202 0000' 1227          LI    R2,DWXAT         TEST ADDR TABLE
004C' 0203 0072' 1228          LI    R3,DWPNT         PAR NO. TABLE
0050' 0204 FF84  1229          LI    R4,)FF84
                 1230
0054'            1231  92$
0054' 020C 05E8  1232  93$     LI    R12,)05E8        RESET TRANSMIT PORT
0058' 1D 00      1233          SBO   0
005A' 1E 00      1234          SBZ   0
005C' C332       1235          MOV   *R2+,R12         CRU ADDRESS FOR CHANNEL
005E' 3032       1236          LDCR  *R2+,0           SEND DATA
0060' 3012       1237          LDCR  *R2,0
0062' 0642       1238          DECT  R2
                 1239
                 1240  *** DELAY 10 MSEC FOR TRANSMISSION
                 1241
0064' 0200 000A  1242          LI    R0,10
0068' 0A70       1243          SLA   R0,7
006A' 0600       1244  198$    DEC   R0
006C' 15FE       1245          JGT   198$
                 1246
                 1247  *** WRITE DITS PAR. NO. OF TX DATA TO RECEIVER
                 1248
006E' C078       1249          MOV   *R8+,R1          ERROR CODE
0070' D033       1250          MOVB  *R3+,R0
0072' 0980       1251          SRL   R0,8             RJ
0074' C500       1252          MOV   R0,*R4           SEND PARAMETER NO.
                 1253
                 1254  *** DELAY APPROX 128 MICROSEC TO READ FROM H/W
                 1255
0076' 0200 0014  1256          LI    R0,20
007A' 0600       1257  30$     DEC   R0
007C' 15FE       1258          JGT   30$
                 1259
```

```
                     53  *           ENDO                                                *
                     54  *           REPEAT FOR DITS 2                                   *
                     55  *           REPEAT FOR DITS 3                                   *
                     56  *                                                               *
                     57  **PD-                                                          *
                     58  * CALLING MODULES: Power On                                    *
                     59  *                                                               *
                     60  * INPUTS:                                                      *
                     61  *                                                               *
                     62  * OUTPUTS:                                                     *
                     63  *                                                               *
                     64  * MODULES CALLED: SYSER, SYSOK                                 *
                     65  *                                                               *
                     66  **DD-                                                          *
                     67  *                                                               *
                     68  * CALLING SEQUENCE: BL DWTST                                   *
                     69  *                                                               *
                     70  * VERSION HISTORY:                                             *
                     71  *                                                               *
                     72  *  VERSION 1: INITIAL RELEASE   DATE: 10-AUG-84 AUTH: N.CONSTANTINIDES  *
                     73  *                                                               *
                     74  *****************************************************************
        =0000        75          RSECT   DWTST
                     76  *** CALL NAME
                     77          INTERN  DWTST
                     78  *** VARIABLES REFERENCED
                     79  *** CONSTANTS REFERENCED
                     80  *** TABLES REFERENCED
                     81  *** MODULES REFERENCED
                     82          EXTERN  SYSER,SYSOK
                     83  *** LIBRARY
                     84          INCLUDE ENCLOS
                     86  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     88          INCLUDE REGDEF          REGISTER DEFENITIONS
                    107          INCLUDE CNSTNT          CONSTANTS
                    240          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    541          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    771          INCLUDE JMPMAC          JUMP MACROS
                    805          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    820          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1196  *** REGISTER DEFINITIONS
                   1197  *****************************************************************
0000'              1198  DWTST
0000'  C80B 0000'  1199          MOV     R11,@LINKZ
0004'  04C9        1200          CLR     R9                      CLEAR DITS2 W/A COMPLETED FLAG
                   1201
                   1202  *** CLEAR DITS 1 & 2 H/W MWMORY
                   1203
0006'  C020 FF84   1204  START   MOV     @)FF84,R0               RESET DITS 1 I/R
000A'  C020 FF88   1205          MOV     @)FF88,R0                       2
000E'  0207 0100   1206          LI      R7,)0100                COUNTER=100HEX
0012'  C807 FF86   1207          MOV     R7,@)FFB6               SET DITS 1 = TEST
```

```
                IDT     DWTST
                SUBTTL  DITS WRAPAROUND TEST
*****************************************************************************
**DD+                                                                       *
* MODULE NAME: DWTST                                                        *
*                                                                           *
* VERSION: 01                             AUTHOR: N.CONSTANTINIDES          *
*                                                                           *
* FILENAME: DWTST                         DATE: 25-JUN-84                   *
*                                                                           *
* FUNCTIONAL DESCRIPTION:                                                   *
*       This module performs the dits wraparound test and sets the          *
*       necessary error bits.                                               *
*                                                                           *
* DESIGN DESCRIPTION:                                                       *
**PD+                                                                       *
*%S DITS WRAPAROUND MODULE                                                  *
*           CLEAR DITS 3 FLAG                                               *
*           RESET INTERRUPT 10 AND 11                                       *
*           SET COUNTER = 100H                                              *
*           SET DITS  1 AT LOW SPEED FOR TEST                               *
*           DO UNTIL COUNTER = 0                                            *
*              CLEAR DITS  1                                                *
*              DELAY 64  SEC BETWEEN CLEARS                                 *
*           ENDDO                                                           *
*           SET CNT = NUMBER OF TESTS FOR DITS  1                           *
*           SET ERROR CODE POINTER = START OF DITS ERROR CODE TBL.          *
*           SET 429 ADDR PTR = START OF DITS WA ADDRESS TBL                 *
*           SET PARAM NUMBER PTR - START OF DITS WA TEST PARAMETER          *
*                                  NUMBER TABLE                             *
*           SET DITS RCVR PORT MEMORY MAP ADDR XMITTER MEMORY               *
*           MAP ADDRESS                                                     *
*           DO UNTIL CNT = 0                                                *
*              RESET 429 TRANSMIT PORT                                      *
*              SEND 2 16 BIT 429 ADDRESSES TO THE CRU                       *
*              DECREMENT 429 ADDRESS TBL PTR TO RESET TO FIRST ADDRESS      *
*              DELAY 3 MSEC FOR DATA TO BE XMITTED AND RECEIVED             *
*              SEND PARAMETER NUMBER TO DITS RCVR PORT                      *
*              DELAY 66  SEC TO READ FROM HARDWARE                          *
*              READ LSH OF 32 BIT DATA                                      *
*              READ MSH                                                     *
*              RESET HARDWARE ADDRESS                                       *
*              SWAP BYTES (H/W BYTES ARE REVERSED)                          *
*              IGNORE LABEL                                                 *
*              LSH = SENT ADDRESS (XOR) LSH                                 *
*              MSH = SEND ADDRESS (XOR) MSH                                 *
*              LSH = LSH (OR) MSH                                           *
*              IF LSH () 0 (NO MATCH)                                       *
*                     CALL SYSER TO SET ERROR BIT IN ERROR BFR              *
*              ELSE                                                         *
*                     CALL SYSOK TO RESET ERROR BIT IN ERROR BFR            *
*              ENDIF                                                        *
```

```
                17428
1038' 8023      17429+
                17449A          GENDT   36,1,3,0,E
                17450                   DATA    (XYZ*256+36-1)          R-EADI ANOMALIES        - CPU 2 ONLY
103C' 8923      17451+
                17471A          GENDT   36,1,2,0,E
                17472                   DATA    (XYZ*256+36-1)          L-EADI OVERTEMP         - CPU 2 ONLY
103E' 8523      17473+
                17493A          GENDT   36,1,1,0,E
                17494                   DATA    (XYZ*256+36-1)          L-EADI BEAM FAIL        - CPU 2 ONLY
1040' 8123      17495+
                17515A          GENDT   36,1,0,0,E
                17516                   DATA    (XYZ*256+36-1)          L-EADI ANOMALIES        - CPU 2 ONLY            320
1042' 8016      17517+
                17537A          GENDT   23,0,0,0,E
                17538+                  DATA    (XYZ*256+23-1)          GMT SECONDS
1044' 8153      17558A          GENDT   84,1,0,0,E
                17559                   DATA    (XYZ*256+84-1)          TO CPU 2 BUFFER
                17560+          GENDT   23,0,6,0,E
1046' 9816      17580A                  DATA    (XYZ*256+23-1)          GMT MINUTES
                17581+          GENDT   84,1,6,0,E
1048' 9953      17601A                  DATA    (XYZ*256+84-1)          TO CPU 2 BUFFER
                17602
                17603+          GENDT   23,0,7,0,E                      GMT HOURS               325
104A' 9C16      17623A                  DATA    (XYZ*256+23-1)
                17624+          GENDT   83,1,7,0,E                      TO CPU 2 BUFFER
104C' 9952      17644A                  DATA    (XYZ*256+83-1)
                17645
                17646+          GENDT   66,0,0,0,E              ***** DUMMY ****
104E' 8041      17666A                  DATA    (XYZ*256+66-1)
                17667
                17668   *** ADDITIONS ****
                17669
                17670   *       CHANNEL NUMBER 1
                17671
                17672+          GENDT   169,1,0,0,E     N1 MAX/EPR LIMIT        - CPU 2 ONLY
1050' 81A8      17692A                  DATA    (XYZ*256+169-1)
                17693
                17694   *       CHANNEL NUMBER 2
                17695
                17696+          GENDT   170,1,0,0,E     N1  MAX/EPR LIMIT       - CPU 2 ONLY
1052' 81A9      17716A                  DATA    (XYZ*256+170-1)
                17717
                17718           END
```

No errors detected

```
1016'  9D21     17053A                 DATA    (XYZ*256+34-1)
                17054
                17055+         GENDT   34,1,6,0,E          SG DIGITAL O/P       - CPU 2 ONLY
1018'  9921     17075A                 DATA    (XYZ*256+34-1)
                17076
                17077+         GENDT   34,1,5,0,E          SG CONTROLLER        - CPU 2 ONLY
101A'  9521     17097A                 DATA    (XYZ*256+34-1)
                17098
                17099+         GENDT   34,1,4,0,E          SG DSPLY SEQCER      - CPU 2 ONLY
101C'  9121     17119A                 DATA    (XYZ*256+34-1)
                17120
                17121+         GENDT   34,1,3,0,E          SG DSPLY DRIVE       - CPU 2 ONLY
101E'  8D21     17141A                 DATA    (XYZ*256+34-1)
                17142
                17143+         GENDT   34,1,2,0,E          SG MAIN MEMORY       - CPU 2 ONLY
1020'  8921     17163A                 DATA    (XYZ*256+34-1)
                17164
                17165+         GENDT   34,1,1,0,E          SG MAIN PROCESSOR    - CPU 2 ONLY
1022'  8521     17185A                 DATA    (XYZ*256+34-1)
                17186
                17187+         GENDT   34,1,0,0,E          SG OVERTEMP          - CPU 2 ONLY
1024'  8121     17207A                 DATA    (XYZ*256+34-1)
                17208
                17209+         GENDT   35,1,8,0,E          R-CP FAULT           - CPU 2 ONLY
1026'  A122     17229A                 DATA    (XYZ*256+35-1)
                17230
                17231+         GENDT   35,1,7,0,E          L-CP FAULT           - CPU 2 ONLY
1028'  9D22     17251A                 DATA    (XYZ*256+35-1)
                17252
                17253+         GENDT   36,1,11,0,E         R-EHSI OVERTEMP      - CPU 2 ONLY
102A'  AD23     17273A                 DATA    (XYZ*256+36-1)
                17274
                17275+         GENDT   36,1,10,0,E         R-EHSI BEAM FAIL     - CPU 2 ONLY
102C'  A923     17295A                 DATA    (XYZ*256+36-1)
                17296
                17297+         GENDT   36,1,9,0,E          R-EHSI ANOMALIES     - CPU 2 ONLY
102E'  A523     17317A                 DATA    (XYZ*256+36-1)
                17318
                17319+         GENDT   36,1,8,0,E          L-EHSI OVERTEMP      - CPU 2 ONLY
1030'  A123     17339A                 DATA    (XYZ*256+36-1)
                17340
                17341+         GENDT   36,1,7,0,E          L-EHSI BEAM FAIL     - CPU 2 ONLY
1032'  9D23     17361A                 DATA    (XYZ*256+36-1)
                17362
                17363+         GENDT   36,1,6,0,E          L-EHSI ANOMALIES     - CPU 2 ONLY
1034'  9923     17383A                 DATA    (XYZ*256+36-1)
                17384
                17385+         GENDT   36,1,5,0,E          R-EADI OVERTEMP      - CPU 2 ONLY
1036'  9523     17405A                 DATA    (XYZ*256+36-1)
                17406
                17407+         GENDT   36,1,4,0,E          R-EADI BEAM FAIL     - CPU 2 ONLY
1038'  9123     17427A                 DATA    (XYZ*256+36-1)
```

```
                    16659+      GENDT   32,1,0,0,E          C-IRS DATA FAULT     - CPU 2 ONLY
OFF4' 811F          16679A              DATA    (XYZ*256+32-1)
                    16680
                    16681+      GENDT   33,1,11,0,E         L-IRS DATA FAULT     - CPU 2 ONLY
OFF6' A020          16701A              DATA    (XYZ*256+33-1)
                    16702
                    16703+      GENDT   33,1,10,0,E         TMC DATA FAULT       - CPU 2 ONLY
OFF8' A920          16723A              DATA    (XYZ*256+33-1)
                    16724
                    16725+      GENDT   33,1,9,0,E          R-FMC DATA FAULT     - CPU 2 ONLY      285
OFFA' A520          16745A              DATA    (XYZ*256+33-1)
                    16746
                    16747+      GENDT   33,1,8,0,E          L-FMC DATA FAULT     - CPU 2 ONLY
OFFC' A120          16767A              DATA    (XYZ*256+33-1)
                    16768
                    16769+      GENDT   33,1,7,0,E          R-FCC DATAS FAULT    - CPU 2 ONLY
OFFE' 9020          16789A              DATA    (XYZ*256+33-1)
                    16790
                    16791+      GENDT   33,1,6,0,E          C-FCC DATA FAULT     - CPU 2 ONLY
1000' 9920          16811A              DATA    (XYZ*256+33-1)
                    16812
                    16813+      GENDT   33,1,5,0,E          L-FCC DATA FAULT     - CPU 2 ONLY
1002' 9520          16833A              DATA    (XYZ*256+33-1)
                    16834
                    16835+      GENDT   33,1,4,0,E          SG FAULT             - CPU 2 ONLY      290
1004' 9120          16855A              DATA    (XYZ*256+33-1)
                    16856
                    16857+      GENDT   33,1,3,0,E          R EHSI FAULT         - CPU 2 ONLY
1006' 8D20          16877A              DATA    (XYZ*256+33-1)
                    16878
                    16879+      GENDT   33,1,2,0,E          L EHSI FAULT         - CPU 2 ONLY
1008' 8920          16899A              DATA    (XYZ*256+33-1)
                    16900
                    16901+      GENDT   33,1,1,0,E          R CP FAULT           - CPU 2 ONLY
100A' 8520          16921A              DATA    (XYZ*256+33-1)
                    16922
                    16923+      GENDT   33,1,0,0,E          L CP FAULT           - CPU 2 ONLY
100C' 8120          16943A              DATA    (XYZ*256+33-1)
                    16944
                    16945+      GENDT   34,1,11,0,E         R EADI FAULT         - CPU 2 ONLY      295
100E' AD21          16965A              DATA    (XYZ*256+34-1)
                    16966
                    16967+      GENDT   34,1,10,0,E         L EADI FAULT         - CPU 2 ONLY
1010' A921          16987A              DATA    (XYZ*256+34-1)
                    16988
                    16989+      GENDT   34,1,9,0,E          SG I/O PROC 3        - CPU 2 ONLY
1012' A521          17009A              DATA    (XYZ*256+34-1)
                    17010
                    17011+      GENDT   34,1,8,0,E          SG I/O PROC 2        - CPU 2 ONLY
1014' A121          17031A              DATA    (XYZ*256+34-1)
                    17032
                    17033+      GENDT   34,1,7,0,E          SG I/O PROC 1        - CPU 2 ONLY
```

```
                    16284
                    16285+          GENDT   31,1,3,0,E              THROTTLE RETARD     - CPU 2 ONLY        265
    0FD2' 8D1E      16305A                  DATA    (XYZ*256+31-1)
                    16306
                    16307+          GENDT   31,1,2,0,E              IAS MODE OPER       - CPU 2 ONLY
    0FD4' 891E      16327A                  DATA    (XYZ*256+31-1)
                    16328
                    16329+          GENDT   31,1,1,0,E              MACH MODE SET       - CPU 2 ONLY
    0FD6' 851E      16349A                  DATA    (XYZ*256+31-1)
                    16350
                    16351+          GENDT   31,1,0,0,E              ALT MODE OPER       - CPU 2 ONLY
    0FD8' 811E      16371A                  DATA    (XYZ*256+31-1)
                    16372
                    16373+          GENDT   30,1,6,0,E              PITCH SPEED CNTRL   - CPU 2 ONLY
    0FDA' 991D      16393A                  DATA    (XYZ*256+30-1)
                    16394
                    16395+          GENDT   32,1,11,0,E             WXR DATA FAULT      - CPU 2 ONLY        270
    0FDC' AD1F      16415A                  DATA    (XYZ*256+32-1)
                    16416
                    16417+          GENDT   32,1,10,0,E             MLS DATA FAULT      - CPU 2 ONLY
    0FDE' A91F      16437A                  DATA    (XYZ*256+32-1)
                    16438
                    16439+          GENDT   32,1,9,0,E              ILS DATA FAULT      - CPU 2 ONLY
    0FE0' A51F      16459A                  DATA    (XYZ*256+32-1)
                    16460
                    16461+          GENDT   32,1,8,0,E              RA DATA FAULT       - CPU 2 ONLY
    0FE2' A11F      16481A                  DATA    (XYZ*256+32-1)
                    16482
                    16483+          GENDT   32,1,7,0,E              D-DME DATA FAULT    - CPU 2 ONLY
    0FE4' 9D1F      16503A                  DATA    (XYZ*256+32-1)
                    16504
                    16505+          GENDT   32,1,6,0,E              L-DME DATA FAULT    - CPU 2 ONLY        275
    0FE6' 991F      16525A                  DATA    (XYZ*256+32-1)
                    16526
                    16527+          GENDT   32,1,5,0,E              R-VOR DATA FAULT    - CPU 2 ONLY
    0FE8' 951F      16547A                  DATA    (XYZ*256+32-1)
                    16548
                    16549+          GENDT   32,1,4,0,E              L-VOR DATA FAULT    - CPU 2 ONLY
    0FEA' 911F      16569A                  DATA    (XYZ*256+32-1)
                    16570
                    16571+          GENDT   32,1,3,0,E              R-ADC DATA FAULT    - CPU 2 ONLY
    0FEC' 8D1F      16591A                  DATA    (XYZ*256+32-1)
                    16592
                    16593+          GENDT   32,1,2,0,E              L-ADC DATA FAULT    - CPU 2 ONLY
    0FEE' 891F      16613A                  DATA    (XYZ*256+32-1)
                    16614
                    16615+          GENDT   66,0,1,0,E              ****** DUMMY ******             280
    0FF0' 8441      16635A                  DATA    (XYZ*256+66-1)
                    16636
                    16637+          GENDT   32,1,1,0,E              R-IRS DATA FAULT    - CPU 2 ONLY
    0FF2' 851F      16657A                  DATA    (XYZ*256+32-1)
                    16658
```

```
                15871A          DATA    (XYZ*256+45-1)
                15872+  GENDT   30,1,0,0,E                      TO CPU 2 BUFFER
OFAC' 811D      15892A          DATA    (XYZ*256+30-1)
                15893
                15894+  GENDT   48,0,0,0,                       H ALERT
OFAE' 002F      15914A          DATA    (XYZ*256+48-1)
                15915+  GENDT   30,1,2,0,E                      TO CPU 2 BUFFER
OFB0' 891D      15935A          DATA    (XYZ*256+30-1)
                15936
                15937+  GENDT   48,0,1,0,E                      GROUNDSPEED SOURCE
OFB2' 842F      15957A          DATA    (XYZ*256+48-1)
                15958+  GENDT   48,0,1,0,E                      GROUNDSPEED SOURCE      250
OFB4' 842F      15978A          DATA    (XYZ*256+48-1)
                15979
                15980+  GENDT   30,1,3,0,E                      TO CPU 2 BUFFER
OFB6' 801D      16000A          DATA    (XYZ*256+30-1)
                16001+  GENDT   48,0,1,0,E                      TRACK ANGLE SOURCE
OFB8' 842F      16021A          DATA    (XYZ*256+48-1)
                16022
                16023+  GENDT   48,0,1,0,E                      TRACK ANGLE SOURCE
OFBA' 842F      16043A          DATA    (XYZ*256+48-1)
                16044+  GENDT   30,1,4,0,E                      TO CPU 2 BUFFER
OFBC' 911D      16064A          DATA    (XYZ*256+30-1)
                16065
                16066+  GENDT   45,0,0,0,                       DH + H ALERT            255
OFBE' 002C      16086A          DATA    (XYZ*256+45-1)
                16087+  GENDT   30,1,1,0,E                      TO CPU 2 BUFFER
OFC0' 851D      16107A          DATA    (XYZ*256+30-1)
                16108
                16109+  GENDT   31,1,11,0,E     G/S MODE OPER       - CPU 2 ONLY
OFC2' AD1E      16129A          DATA    (XYZ*256+31-1)
                16130
                16131+  GENDT   31,1,10,0,E     FLARE OPER          - CPU 2 ONLY
OFC4' A91E      16151A          DATA    (XYZ*256+31-1)
                16152
                16153+  GENDT   31,1,9,0,E      G/A MODE OPER-P     - CPU 2 ONLY
OFC6' A51E      16173A          DATA    (XYZ*256+31-1)
                16174
                16175+  GENDT   31,1,8,0,E      T/O MODE OPER-P     - CPU 2 ONLY        260
OFC8' A11E      16195A          DATA    (XYZ*256+31-1)
                16196
                16197+  GENDT   31,1,7,0,E      ALT HOLD MODE OPER  - CPU 2 ONLY
OFCA' 9D1E      16217A          DATA    (XYZ*256+31-1)
                16218
                16219+  GENDT   31,1,6,0,E      V/S MODE OPER       - CPU 2 ONLY
OFCC' 991E      16239A          DATA    (XYZ*256+31-1)
                16240
                16241+  GENDT   31,1,5,0,E      V/NAV MODE OPER     - CPU 2 ONLY
OFCE' 951E      16261A          DATA    (XYZ*256+31-1)
                16262
                16263+  GENDT   31,1,4,0,E      FL CH MODE OPER     - CPU 2 ONLY
OFD0' 911E      16283A          DATA    (XYZ*256+31-1)
```

```
                    15420
                    15421+      GENDT   4,0,1,0,E                   PITCH ATTITUDE          225
    0FB2'  8403     15441A              DATA    (XYZ*256+4-1)
                    15442+      GENDT   20,0,1,0,E                  PITCH ATTITUDE
    0FC4'  8413     15462A              DATA    (XYZ*256+20-1)
                    15463+      GENDT   36,0,1,0,E                  PITCH ATTITUDE
    0FB6'  8423     15483A              DATA    (XYZ*256+36-1)
                    15484+      GENDT   52,0,1,0,                   PITCH ATTITUDE
    0FC8'  0433     15504A              DATA    (XYZ*256+52-1)
                    15505+      GENDT   94,1,1,0,E                  TO CPU 2 BUFFER
    0FBA'  855D     15525A              DATA    (XYZ*256+94-1)
                    15526
                    15527+      GENDT   17,0,1,0,E                  ROLL ATTITUDE           230
    0FBC'  8410     15547A              DATA    (XYZ*256+17-1)
                    15548+      GENDT   49,0,1,0,                   ROLL ATTITUDE
    0FBE'  0430     15568A              DATA    (XYZ*256+49-1)
                    15569+      GENDT   95,1,1,0,E                  TO CPU 2 BUFFER
    0FC0'  855E     15589A              DATA    (XYZ*256+95-1)
                    15590
                    15591+      GENDT   53,0,1,0,                   LOCALIZER DEV
    0FC2'  0434     15611A              DATA    (XYZ*256+53-1)
                    15612+      GENDT   96,1,1,0,E                  TO CPU 2 BUFFER
    0FC4'  855F     15632A              DATA    (XYZ*256+96-1)
                    15633
                    15634+      GENDT   14,0,1,0,                   GLIDE SLOPE DEV         235
    0FC6'  040D     15654A              DATA    (XYZ*256+14-1)
                    15655+      GENDT   97,1,1,0,E                  TO CPU 2 BUFFER
    0FC8'  8560     15675A              DATA    (XYZ*256+97-1)
                    15676
                    15677+      GENDT   60,0,0,0,                   RADIO ALT
    0FCA'  003B     15697A              DATA    (XYZ*256+60-1)
                    15698+      GENDT   98,1,0,0,E                  TO CPU 2 BUFFER
    0FCC'  8161     15718A              DATA    (XYZ*256+98-1)
                    15719
                    15720+      GENDT   3,0,1,0,                    MAGN HEADING
    0FCE'  0402     15740A              DATA    (XYZ*256+3-1)
                    15741+      GENDT   99,1,1,0,E                  TO CPU 2 BUFFER         240
    0FA0'  8562     15761A              DATA    (XYZ*256+99-1)
                    15762
                    15763+      GENDT   100,1,1,0,E         TRACK ANGLE MAG     - CPU 2 ONLY
    0FA2'  8563     15783A              DATA    (XYZ*256+100-1)
                    15784
                    15785+      GENDT   101,1,1,0,E         TRACK ANGLE TRUE    - CPU 2 ONLY
    0FA4'  8564     15805A              DATA    (XYZ*256+101-1)
                    15806
                    15807+      GENDT   102,1,1,0,E         TRUE HEADING        - CPU 2 ONLY
    0FA6'  8565     15827A              DATA    (XYZ*256+102-1)
                    15828
                    15829+      GENDT   30,1,5,0,E          MAG TRUE DATA       - CPU 2 ONLY
    0FA8'  951D     15849A              DATA    (XYZ*256+30-1)
                    15850
                    15851+      GENDT   45,0,1,0,                   DH ALERT                245
```

```
                    14969+      GENDT   25,1,0,0,E          PROM B CHECK        - CPU 2 ONLY
     0F58' 8118     14989A              DATA    (XYZ*256+25-1)
                    14990
                    14991+      GENDT   32,0,11,0,                  TERRAIN PULL UP          205
     0F5A' 2C1F     15011A              DATA    (XYZ*256+32-1)
                    15012+      GENDT   18,1,11,0,E                 TO CPU 2 BUFFER
     0F5C' AD11     15032A              DATA    (XYZ*256+18-1)
                    15033
                    15034+      GENDT   32,0,10,0,                  MINIMUMS
     0F5E' 281F     15054A              DATA    (XYZ*256+32-1)
                    15055+      GENDT   18,1,10,0,E                 TO CPU 2 BUFFER
     0F60' A911     15075A              DATA    (XYZ*256+18-1)
                    15076
                    15077+      GENDT   32,0,9,0,                   GLIDE SLOPE
     0F62' 241F     15097A              DATA    (XYZ*256+32-1)
                    15098+      GENDT   18,1,9,0,E                  TO CPU 2 BUFFER          210
     0F64' A511     15118A              DATA    (XYZ*256+18-1)
                    15119
                    15120+      GENDT   32,0,8,0,                   TOO LOW TERRAIN
     0F66' 201F     15140A              DATA    (XYZ*256+32-1)
                    15141+      GENDT   18,1,8,0,E                  TO CPU 2 BUFFER
     0F68' A111     15161A              DATA    (XYZ*256+18-1)
                    15162
                    15163+      GENDT   32,0,7,0,                   TOO LOW FLAPS
     0F6A' 1C1F     15183A              DATA    (XYZ*256+32-1)
                    15184+      GENDT   18,1,7,0,E                  TO CPU 2 BUFFER
     0F6C' 9D11     15204A              DATA    (XYZ*256+18-1)
                    15205
                    15206+      GENDT   32,0,6,0,                   TOO LOW GEAR             215
     0F6E' 181F     15226A              DATA    (XYZ*256+32-1)
                    15227+      GENDT   18,1,6,0,E                  TO CPU 2 BUFFER
     0F70' 9911     15247A              DATA    (XYZ*256+18-1)
                    15248
                    15249+      GENDT   32,0,5,0,                   DON'T SINK
     0F72' 141F     15269A              DATA    (XYZ*256+32-1)
                    15270+      GENDT   18,1,5,0,E                  TO CPU 2 BUFFER
     0F74' 9511     15290A              DATA    (XYZ*256+18-1)
                    15291
                    15292+      GENDT   32,0,4,0,                   TERRAIN
     0F76' 101F     15312A              DATA    (XYZ*256+32-1)
                    15313+      GENDT   18,1,4,0,E                  TO CPU 2 BUFFER          220
     0F78' 9111     15333A              DATA    (XYZ*256+18-1)
                    15334
                    15335+      GENDT   32,0,3,0,                   PULL UP
     0F7A' 0C1F     15355A              DATA    (XYZ*256+32-1)
                    15356+      GENDT   18,1,3,0,E                  TO CPU 2 BUFFER
     0F7C' 8D11     15376A              DATA    (XYZ*256+18-1)
                    15377
                    15378+      GENDT   32,0,2,0,                   SINK RATE
     0F7E' 081F     15398A              DATA    (XYZ*256+32-1)
                    15399+      GENDT   18,1,2,0,E                  TO CPU 2 BUFFER
     0F80' 8911     15419A              DATA    (XYZ*256+18-1)
```

```
                    14594
                    14595+          GENDT   24,1,0,0,E              AOA COMPARE TEST     - CPU 2 ONLY
    0F36'  8117     14615A                  DATA    (XYZ*256+24-1)
                    14616
                    14617+          GENDT   23,1,6,0,E              POWER SUPPLY TEST    - CPU 2 ONLY
    0F38'  9916     14637A                  DATA    (XYZ*256+23-1)
                    14638
                    14639+          GENDT   23,1,5,0,E              ARINC XMTR TEST      - CPU 2 ONLY
    0F3A'  9516     14659A                  DATA    (XYZ*256+23-1)
                    14660
                    14661+          GENDT   23,1,4,0,E              ARINC XMTR TEST      - CPU 2 ONLY        190
    0F3C'  9116     14681A                  DATA    (XYZ*256+23-1)
                    14682
                    14683+          GENDT   23,1,3,0,E              ARINC XMTR TEST      - CPU 2 ONLY
    0F3E'  8D16     14703A                  DATA    (XYZ*256+23-1)
                    14704
                    14705+          GENDT   23,1,2,0,E              PT CALIB TEST        - CPU 2 ONLY
    0F40'  8916     14725A                  DATA    (XYZ*256+23-1)
                    14726
                    14727+          GENDT   23,1,1,0,E              PT SENS TEMP TEST    - CPU 2 ONLY
    0F42'  8516     14747A                  DATA    (XYZ*256+23-1)
                    14748
                    14749+          GENDT   23,1,0,0,E              PT SENS PER TEST     - CPU 2 ONLY
    0F44'  8116     14769A                  DATA    (XYZ*256+23-1)
                    14770
                    14771+          GENDT   25,1,9,0,E              A/C TYPE PARITY      - CPU 2 ONLY        195
    0F46'  A518     14791A                  DATA    (XYZ*256+25-1)
                    14792
                    14793+          GENDT   25,1,8,0,E              A/C TYPE MSB         - CPU 2 ONLY
    0F48'  A118     14813A                  DATA    (XYZ*256+25-1)
                    14814
                    14815+          GENDT   25,1,7,0,E              A/C TYPE LSB+3       - CPU 2 ONLY
    0F4A'  9D18     14835A                  DATA    (XYZ*256+25-1)
                    14836
                    14837+          GENDT   25,1,6,0,E              A/C TYPE LSB+2       - CPU 2 ONLY
    0F4C'  9918     14857A                  DATA    (XYZ*256+25-1)
                    14858
                    14859+          GENDT   25,1,5,0,E              A/C TYPE LSB+1       - CPU 2 ONLY
    0F4E'  9518     14879A                  DATA    (XYZ*256+25-1)
                    14880
                    14881+          GENDT   25,1,4,0,E              A/C TYPE LSB         - CPU 2 ONLY        200
    0F50'  9118     14901A                  DATA    (XYZ*256+25-1)
                    14902
                    14903+          GENDT   25,1,3,0,E              A/C TYPE MEM CHECK   - CPU 2 ONLY
    0F52'  8D18     14923A                  DATA    (XYZ*256+25-1)
                    14924
                    14925+          GENDT   25,1,2,0,E              PT MEM CHECK         - CPU 2 ONLY
    0F54'  8918     14945A                  DATA    (XYZ*256+25-1)
                    14946
                    14947+          GENDT   25,1,1,0,E              PS MEM CHECK         - CPU 2 ONLY
    0F56'  8518     14967A                  DATA    (XYZ*256+25-1)
                    14968
```

```
     0F12'  8D15      14219A                DATA    (XYZ*256+22-1)
                      14220
                      14221+          GENDT   22,1,2,0,E          A/C TYPE CONST TEST  - CPU 2 ONLY          170
     0F14'  8915      14241A                DATA    (XYZ*256+22-1)
                      14242
                      14243+          GENDT   22,1,1,0,E          A/C TYPE PROG TEST   - CPU 2 ONLY
     0F16'  8515      14263A                DATA    (XYZ*256+22-1)
                      14264
                      14265+          GENDT   22,1,0,0,E          BARO 3 TEST          - CPU 2 ONLY
     0F18'  8115      14285A                DATA    (XYZ*256+22-1)
                      14286
                      14287+          GENDT   23,1,11,0,E         BARO 2 TEST          - CPU 2 ONLY
     0F1A'  AD16      14307A                DATA    (XYZ*256+23-1)
                      14308
                      14309+          GENDT   23,1,10,0,E         BARO 1 TEST          - CPU 2 ONLY
     0F1C'  A916      14329A                DATA    (XYZ*256+23-1)
                      14330
                      14331+          GENDT   23,1,9,0,E          TAT INPUT TEST       - CPU 2 ONLY          175
     0F1E'  A516      14351A                DATA    (XYZ*256+23-1)
                      14352
                      14353+          GENDT   23,1,8,0,E          RADA VANE TEST       - CPU 2 ONLY
     0F20'  A116      14373A                DATA    (XYZ*256+23-1)
                      14374
                      14375+          GENDT   23,1,7,0,E          LADA VANE TEST       - CPU 2 ONLY
     0F22'  9D16      14395A                DATA    (XYZ*256+23-1)
                      14396
                      14397+          GENDT   24,1,9,0,E          VMO TEST             - CPU 2 ONLY
     0F24'  A517      14417A                DATA    (XYZ*256+24-1)
                      14418
                      14419+          GENDT   24,1,8,0,E          BARO 4 TEST          - CPU 2 ONLY
     0F26'  A117      14439A                DATA    (XYZ*256+24-1)
                      14440
                      14441+          GENDT   24,1,7,0,E          EAROM TEST           - CPU 2 ONLY          180
     0F28'  9D17      14461A                DATA    (XYZ*256+24-1)
                      14462
                      14463+          GENDT   24,1,6,0,E          PT PLL               - CPU 2 ONLY
     0F2A'  9917      14483A                DATA    (XYZ*256+24-1)
                      14484
                      14485+          GENDT   24,1,5,0,E          PS PLL               - CPU 2 ONLY
     0F2C'  9517      14505A                DATA    (XYZ*256+24-1)
                      14506
                      14507+          GENDT   24,1,4,0,E          PROG SEQ TEST        - CPU 2 ONLY
     0F2E'  9117      14527A                DATA    (XYZ*256+24-1)
                      14528
                      14529+          GENDT   24,1,3,0,E          TEMP PS=TEMP PT      - CPU 2 ONLY
     0F30'  8D17      14549A                DATA    (XYZ*256+24-1)
                      14550
                      14551+          GENDT   24,1,2,0,E          AVGE ADA TEST        - CPU 2 ONLY          185
     0F32'  8917      14571A                DATA    (XYZ*256+24-1)
                      14572
                      14573+          GENDT   24,1,1,0,E          PS = PT              - CPU 2 ONLY
     0F34'  8517      14593A                DATA    (XYZ*256+24-1)
```

```
                    13825+          GENDT   21,1,1,0,E              MACH FLAG           - CPU 2 ONLY
 0EF0' 8514         13845A                  DATA    (XYZ*256+21-1)
                    13846
                    13847+          GENDT   21,1,2,0,E              AOA FLAG            - CPU 2 ONLY
 0EF2' 8914         13867A                  DATA    (XYZ*256+21-1)
                    13868
                    13869+          GENDT   21,1,3,0,E              PRESS ALT FLAG      - CPU 2 ONLY
 0EF4' 8D14         13889A                  DATA    (XYZ*256+21-1)
                    13890
                    13891+          GENDT   21,1,4,0,E              TAT FLAG            - CPU 2 ONLY        155
 0EF6' 9114         13911A                  DATA    (XYZ*256+21-1)
                    13912
                    13913+          GENDT   21,1,9,0,E              4 BARO ALT =3       - CPU 2 ONLY
 0EF8' A514         13933A                  DATA    (XYZ*256+21-1)
                    13934
                    13935+          GENDT   21,1,8,0,E              4 BARO ALT =2       - CPU 2 ONLY
 0EFA' A114         13955A                  DATA    (XYZ*256+21-1)
                    13956
                    13957+          GENDT   21,1,7,0,E              4 BARO ALT =1       - CPU 2 ONLY
 0EFC' 9D14         13977A                  DATA    (XYZ*256+21-1)
                    13978
                    13979+          GENDT   21,1,6,0,E              EXT AOA MON (FAIL)  - CPU 2 ONLY
 0EFE' 9914         13999A                  DATA    (XYZ*256+21-1)
                    14000
                    14001+          GENDT   21,1,5,0,E              ZERO SSEC (AOA)     - CPU 2 ONLY        160
 0F00' 9514         14021A                  DATA    (XYZ*256+21-1)
                    14022
                    14023+          GENDT   22,1,11,0,E             F/D CONVERSION TEST - CPU 2 ONLY
 0F02' AD15         14043A                  DATA    (XYZ*256+22-1)
                    14044
                    14045+          GENDT   22,1,10,0,E             PS CALIB TEST       - CPU 2 ONLY
 0F04' A915         14065A                  DATA    (XYZ*256+22-1)
                    14066
                    14067+          GENDT   22,1,9,0,E              PS SENS TEST        - CPU 2 ONLY
 0F06' A515         14087A                  DATA    (XYZ*256+22-1)
                    14088
                    14089+          GENDT   22,1,8,0,E              PS SENS PER TEST    - CPU 2 ONLY
 0F08' A115         14109A                  DATA    (XYZ*256+22-1)
                    14110
                    14111+          GENDT   22,1,7,0,E              PROG MEM TEST'      - CPU 2 ONLY        165
 0F0A' 9D15         14131A                  DATA    (XYZ*256+22-1)
                    14132
                    14133+          GENDT   22,1,6,0,E              RAM TEST            - CPU 2 ONLY
 0F0C' 9915         14153A                  DATA    (XYZ*256+22-1)
                    14154
                    14155+          GENDT   22,1,5,0,E              PROCESSOR TEST      - CPU 2 ONLY
 0F0E' 9515         14175A                  DATA    (XYZ*256+22-1)
                    14176
                    14177+          GENDT   22,1,4,0,E              A TO D TEST         - CPU 2 ONLY
 0F10' 9115         14197A                  DATA    (XYZ*256+22-1)
                    14198
                    14199+          GENDT   22,1,3,0,E              OSPD HW TEST        - CPU 2 ONLY
```

```
                    13412+      GENDT   51,0,8,0,E                  ONSIDE AOA FAIL
0ED8' A032          13432A              DATA    (XYZ*256+51-1)
                    13433+      GENDT    19,1,8,0,E                 TO CPU 2 BUFFER
0EDC' A112          13453A              DATA    (XYZ*256+19-1)
                    13454
                    13455+      GENDT    51,0,7,0,E                 OVER SPEED                135
0EE0' 9C32          13475A              DATA    (XYZ*256+51-1)
                    13476+      GENDT    19,1,7,0,E                 TO CPU 2 BUFFER
0EE4' 9D12          13496A              DATA    (XYZ*256+19-1)
                    13497
                    13498+      GENDT    51,0,10,0,E                VMD ALT 2
0EE8' A832          13518A              DATA    (XYZ*256+51-1)
                    13519+      GENDT    19,1,10,0,E                TO CPU 2 BUFFER
0EEC' A912          13539A              DATA    (XYZ*256+19-1)
                    13540
                    13541+      GENDT    51,0,11,0,E                VMD ALT 3
0EF0' AC32          13561A              DATA    (XYZ*256+51-1)
                    13562+      GENDT    19,1,11,0,E                TO CPU 2 BUFFER           140
0EF4' AD12          13582A              DATA    (XYZ*256+19-1)
                    13583
                    13584+      GENDT    51,0,3,0,E                 PILOT/STAT HEAT ON R
0EF8' 8C32          13604A              DATA    (XYZ*256+51-1)
                    13605+      GENDT    19,1,3,0,E                 TO CPU 2 BUFFER
0EFC' 8D12          13625A              DATA    (XYZ*256+19-1)
                    13626
                    13627+      GENDT    20,1,7,0,E      ZERO SSEC (MACH)     - CPU 2 ONLY
0F00' 9D13          13647A              DATA    (XYZ*256+20-1)
                    13648
                    13649+      GENDT    20,1,6,0,E      BARO PORT A          - CPU 2 ONLY
0F04' 9913          13669A              DATA    (XYZ*256+20-1)
                    13670
                    13671+      GENDT    20,1,5,0,E      AOA C ALTERNATE      - CPU 2 ONLY          145
0F08' 9513          13691A              DATA    (XYZ*256+20-1)
                    13692
                    13693+      GENDT    20,1,4,0,E      ESEC ALTERNATE       - CPU 2 ONLY
0F0C' 9113          13713A              DATA    (XYZ*256+20-1)
                    13714
                    13715+      GENDT    20,1,3,0,E      VMO ALT 4            - CPU 2 ONLY
0F10' 8D13          13735A              DATA    (XYZ*256+20-1)
                    13736
                    13737+      GENDT    20,1,2,0,E      AOA UNIQUE           - CPU 2 ONLY
0F14' 8913          13757A              DATA    (XYZ*256+20-1)
                    13758
                    13759+      GENDT    20,1,1,0,E      PILOT HEAT ON        - CPU 2 ONLY
0F18' 8513          13779A              DATA    (XYZ*256+20-1)
                    13780
                    13781+      GENDT    20,1,0,0,E      ICING DETEC ON       - CPU 2 ONLY          150
0F1C' 8113          13801A              DATA    (XYZ*256+20-1)
                    13802
                    13803+      GENDT    21,1,0,0,E      CAS FLAG             - CPU 2 ONLY
0F20' 8114          13823A              DATA    (XYZ*256+21-1)
                    13824
```

```
                    12999
                    13000+      GENDT   17,1,6,0,E          PROM 7 CHECK        - CPU 2 ONLY
    0EA4' 9910      13020A              DATA    (XYZ*256+17-1)
                    13021
                    13022+      GENDT   17,1,5,0,E          PROM 6 CHECK        - CPU 2 ONLY   115
    0EA6' 9510      13042A              DATA    (XYZ*256+17-1)
                    13043
                    13044+      GENDT   17,1,4,0,E          PROM 5 CHECK        - CPU 2 ONLY
    0EA8' 9110      13064A              DATA    (XYZ*256+17-1)
                    13065
                    13066+      GENDT   17,1,3,0,E          PROM 4 CHECK        - CPU 2 ONLY
    0EAA' 8D10      13086A              DATA    (XYZ*256+17-1)
                    13087
                    13088+      GENDT   17,1,2,0,E          PROM 3 CHECK        - CPU 2 ONLY
    0EAC' 8910      13108A              DATA    (XYZ*256+17-1)
                    13109
                    13110+      GENDT   17,1,1,0,E          PROM 2 CHECK        - CPU 2 ONLY
    0EAE' 8510      13130A              DATA    (XYZ*256+17-1)
                    13131
                    13132+      GENDT   17,1,0,0,E          PROM 1 CHECK        - CPU 2 ONLY   120
    0EB0' 8110      13152A              DATA    (XYZ*256+17-1)
                    13153
                    13154+      GENDT   51,0,9,0,E                  VMO ALT 1
    0EB2' A432      13174A              DATA    (XYZ*256+51-1)
                    13175+      GENDT   19,1,9,0,E                  TO CPU 2 BUFFER
    0EB4' A512      13195A              DATA    (XYZ*256+19-1)
                    13196
                    13197+      GENDT   51,0,6,0,E                  SAOA HEAT ON
    0EB6' 9832      13217A              DATA    (XYZ*256+51-1)
                    13218+      GENDT   19,1,6,0,E                  TO CPU 2 BUFFER
    0EB8' 9912      13238A              DATA    (XYZ*256+19-1)
                    13239
                    13240+      GENDT   51,0,5,0,E                  LAOA HEAT ON            125
    0EBA' 9432      13260A              DATA    (XYZ*256+51-1)
                    13261+      GENDT   19,1,5,0,E                  TO CPU 2 BUFFER
    0EBC' 9512      13281A              DATA    (XYZ*256+19-1)
                    13282
                    13283+      GENDT   51,0,4,0,E                  TAT PROBE HEAT ON
    0EBE' 9032      13303A              DATA    (XYZ*256+51-1)
                    13304+      GENDT   19,1,4,0,E                  TO CPU 2 BUFFER
    0EC0' 9112      13324A              DATA    (XYZ*256+19-1)
                    13325
                    13326+      GENDT   51,0,2,0,E                  PILOT/STAT HEAT ON
    0EC2' 8832      13346A              DATA    (XYZ*256+51-1)
                    13347+      GENDT   19,1,2,0,E                  TO CPU 2 BUFFER         130
    0EC4' 8912      13367A              DATA    (XYZ*256+19-1)
                    13368
                    13369+      GENDT   51,0,1,0,E                  ADC INVALID
    0EC6' 8432      13389A              DATA    (XYZ*256+51-1)
                    13390+      GENDT   19,1,1,0,E                  TO CPU 2 BUFFER
    0EC8' 8512      13410A              DATA    (XYZ*256+19-1)
                    13411
```

```
0E80' 913F      12624A              DATA    (XYZ*256+64-1)
                12625
                12626+      GENDT    67,1,0,0,E          BARO CORR 2 (IN)     - CPU 2 ONLY
0E82' 8142      12646A              DATA    (XYZ*256+67-1)
                12647
                12648+      GENDT    66,1,4,0,E          BARO CORR 2 (IN)     - CPU 2 ONLY
0E84' 9141      12668A              DATA    (XYZ*256+66-1)
                12669
                12670+      GENDT    69,1,0,0,E          BARO CORR 2 (MB)     - CPU 2 ONLY
0E86' 8144      12690A              DATA    (XYZ*256+69-1)
                12691
                12692+      GENDT    68,1,4,0,E          BARO CORR 2 (MB)     - CPU 2 ONLY   100
0E88' 9143      12712A              DATA    (XYZ*256+68-1)
                12713
                12714+      GENDT    71,1,0,0,E          BARO CORR 3 (IN)     - CPU 2 ONLY
0E8A' 8146      12734A              DATA    (XYZ*256+71-1)
                12735
                12736+      GENDT    70,1,4,0,E          BARO CORR 3 (IN)     - CPU 2 ONLY
0E8C' 9145      12756A              DATA    (XYZ*256+70-1)
                12757
                12758+      GENDT    73,1,0,0,E          BARO CORR 3 (MB)     - CPU 2 ONLY
0E8E' 8148      12778A              DATA    (XYZ*256+73-1)
                12779
                12780+      GENDT    72,1,4,0,E          BARO CORR 3 (MB)     - CPU 2 ONLY
0E90' 9147      12800A              DATA    (XYZ*256+72-1)
                12801
                12802+      GENDT    74,1,0,0,E          CORRECTED AOA        - CPU 2 ONLY   105
0E92' 8149      12822A              DATA    (XYZ*256+74-1)
                12823
                12824+      GENDT    75,1,0,0,E          IMPACT PRESS         - CPU 2 ONLY
0E94' 814A      12844A              DATA    (XYZ*256+75-1)
                12845
                12846+      GENDT    76,1,0,0,E          MAX OPR TG SCHED     - CPU 2 ONLY
0E96' 814B      12866A              DATA    (XYZ*256+76-1)
                12867
                12868+      GENDT    77,1,0,0,E          STATIC AIR TEMP      - CPU 2 ONLY
0E98' 814C      12888A              DATA    (XYZ*256+77-1)
                12889
                12890+      GENDT    78,1,0,0,E          STATIC AIR TEMP D    - CPU 2 ONLY
0E9A' 814D      12910A              DATA    (XYZ*256+78-1)
                12911
                12912+      GENDT    79,1,0,0,E          TOTAL AIR TEMP D     - CPU 2 ONLY   110
0E9C' 814E      12932A              DATA    (XYZ*256+79-1)
                12933
                12934+      GENDT    80,1,0,0,E          TOTAL PRESS          - CPU 2 ONLY
0E9E' 814F      12954A              DATA    (XYZ*256+80-1)
                12955
                12956+      GENDT    81,1,0,0,E          TRUE AIRSPEED        - CPU 2 ONLY
0EA0' 8150      12976A              DATA    (XYZ*256+81-1)
                12977
                12978+      GENDT    82,1,0,0,E          TRUE AIRSPEED D      - CPU 2 ONLY
0EA2' 8151      12998A              DATA    (XYZ*256+82-1)
```

```
                12211A          DATA    (XYZ*256+12-1)
                12212+  GENDT    51,1,2,0,E                    TO CPU 2 BUFFER
                12232A          DATA    (XYZ*256+51-1)
                12233
                12234+  GENDT    21,0,0,0,                     COMPUTED AIRSPEED
                12254A          DATA    (XYZ*256+21-1)
                12255+  GENDT    52,1,0,0,E                    TO CPU 2 BUFFER
                12275A          DATA    (XYZ*256+52-1)
                12276
                12277+  GENDT    6,0,0,0,                       PRESS ALT 29.92
                12297A          DATA    (XYZ*256+6-1)
                12298+  GENDT    53,1,0,0,E                    TO CPU 2 BUFFER
                12318A          DATA    (XYZ*256+53-1)
                12319
                12320+  GENDT    54,1,2,0,E          MACH                - CPU 2 ONLY
                12340A          DATA    (XYZ*256+54-1)
                12341
                12342+  GENDT    55,1,0,0,E          ALT BARO 1          - CPU 2 ONLY
                12362A          DATA    (XYZ*256+55-1)
                12363
                12364+  GENDT    56,1,0,0,E          ALT BARO 2          - CPU 2 ONLY
                12384A          DATA    (XYZ*256+56-1)
                12385
                12386+  GENDT    57,1,0,0,E          ALT BARO 3          - CPU 2 ONLY
                12406A          DATA    (XYZ*256+57-1)
                12407
                12408+  GENDT    58,1,0,0,E          ALT BARO 4          - CPU 2 ONLY
                12428A          DATA    (XYZ*256+58-1)
                12429
                12430+  GENDT    59,1,1,0,E          ALTTITUDE RATE      - CPU 2 ONLY
                12450A          DATA    (XYZ*256+59-1)
                12451
                12452+  GENDT    38,0,1,0,E                    ANGLE OF ATTACK L
                12472A          DATA    (XYZ*256+38-1)
                12473+  GENDT    38,0,1,0,E                    ANGLE OF ATTACK L
                12493A          DATA    (XYZ*256+38-1)
                12494+  GENDT    60,1,1,0,E                    TO CPU 2 BUFFER
                12514A          DATA    (XYZ*256+60-1)
                12515
                12516+  GENDT    61,1,0,0,E          ANGLE OF ATTACK R   - CPU 2 ONLY
                12536A          DATA    (XYZ*256+61-1)
                12537
                12538+  GENDT    63,1,0,0,E          BARO CORR 1 (IN)    - CPU 2 ONLY
                12558A          DATA    (XYZ*256+63-1)
                12559
                12560+  GENDT    62,1,4,0,E          BARO CORR 1 (IN)    - CPU 2 ONLY
                12580A          DATA    (XYZ*256+62-1)
                12581
                12582+  GENDT    65,1,0,0,E          BARO CORR 1 (MB)    - CPU 2 ONLY
                12602A          DATA    (XYZ*256+65-1)
                12603
                12604+  GENDT    64,1,4,0,E          BARO CORR 1 (MB)    - CPU 2 ONLY
```

```
                    11779+      GENDT   16,1,0,0,E          ENG MDL CODE 1      - CPU 2 ONLY
 0E34'  810F        11799A              DATA    (XYZ*256+16-1)
                    11800
                    11801+      GENDT   16,1,1,0,E          ENG MDL CODE 2      - CPU 2 ONLY
 0E36'  850F        11821A              DATA    (XYZ*256+16-1)
                    11822
                    11823+      GENDT   16,1,2,0,E          ENG MDL CODE 3      - CPU 2 ONLY        60
 0E38'  890F        11843A              DATA    (XYZ*256+16-1)
                    11844
                    11845+      GENDT   16,1,3,0,E          ENG MDL CODE 4      - CPU 2 ONLY
 0E3A'  8D0F        11865A              DATA    (XYZ*256+16-1)
                    11866
                    11867+      GENDT   16,1,4,0,E          ENG MDL CODE 5      - CPU 2 ONLY
 0E3C'  910F        11887A              DATA    (XYZ*256+16-1)
                    11888
                    11889+      GENDT   16,1,5,0,E          ENG MDL CODE 6      - CPU 2 ONLY
 0E3E'  950F        11909A              DATA    (XYZ*256+16-1)
                    11910
                    11911+      GENDT   16,0,1,0,                   ECS BLEED
 0E40'  040F        11931A              DATA    (XYZ*256+16-1)
                    11932+      GENDT   16,1,6,0,E                  TO CPU 2 BUFFER     65
 0E42'  990F        11952A              DATA    (XYZ*256+16-1)
                    11953
                    11954+      GENDT   51,0,0,0,E                  ECS MODE
 0E44'  8032        11974A              DATA    (XYZ*256+51-1)
                    11975+      GENDT   16,1,7,0,E                  TO CPU 2 BUFFER
 0E46'  9D0F        11995A              DATA    (XYZ*256+16-1)
                    11996
                    11997+      GENDT   20,0,0,0,                   CA 1 BLEED
 0E48'  0013        12017A              DATA    (XYZ*256+20-1)
                    12018+      GENDT   16,1,8,0,E                  TO CPU 2 BUFFER
 0E4A'  A10F        12039A              DATA    (XYZ*256+16-1)
                    12039
                    12040+      GENDT   24,0,0,0,                   WA 1 BLEED          70
 0E4C'  0017        12060A              DATA    (XYZ*256+24-1)
                    12061+      GENDT   16,1,9,0,E                  TO CPU 2 BUFFER
 0E4E'  A50F        12081A              DATA    (XYZ*256+16-1)
                    12082
                    12083+      GENDT   39,0,0,0,                   ADP ON
 0E50'  0026        12103A              DATA    (XYZ*256+39-1)
                    12104+      GENDT   16,1,10,0,E                 TO CPU 2 BUFFER
 0E52'  A90F        12124A              DATA    (XYZ*256+16-1)
                    12125
                    12126+      GENDT   40,0,1,0,                   ISOL VALVE OPEN (GE)
 0E54'  0427        12146A              DATA    (XYZ*256+40-1)
                    12147+      GENDT   16,1,11,0,E                 TO CPU 2 BUFFER     75
 0E56'  AD0F        12167A              DATA    (XYZ*256+16-1)
                    12168
                    12169+      GENDT   50,1,2,0,E          ANGLE OF ATTACK     - CPU 2 ONLY
 0E58'  8931        12189A              DATA    (XYZ*256+50-1)
                    12190
                    12191+      GENDT   12,0,2,0,                   TOTAL AIR TEMP
```

```
                    11347
                    11348+      GENDT   37,0,1,0,                      N1 ACT R (GE)
0E0C' 0424          11368A              DATA    (XYZ*256+37-1)
                    11369+      GENDT   147,1,1,0,E                     TO CPU 2 BUFFER
0E0E' 8592          11389A              DATA    (XYZ*256+147-1)
                    11390
                    11391+      GENDT   48,1,1,0,E          TLA                 - CPU 2 ONLY        40
0E10' 852F          11411A              DATA    (XYZ*256+48-1)
                    11412
                    11413+      GENDT   49,1,0,0,E          TT2                 - CPU 2 ONLY
0E12' 8130          11433A              DATA    (XYZ*256+49-1)
                    11434
                    11435+      GENDT   16,0,1,0,                       ECS BLEED (PW)
0E14' 040F          11455A              DATA    (XYZ*256+16-1)
                    11456+      GENDT   16,1,6,0,E                      TO CPU 2 BUFFER
0E16' 990F          11476A              DATA    (XYZ*256+16-1)
                    11477
                    11478+      GENDT   51,0,0,0,E                      ECS MODE (PW)
0E18' 8032          11498A              DATA    (XYZ*256+51-1)
                    11499+      GENDT   16,1,7,0,E                      TO CPU 2 BUFFER     45
0E1A' 9D0F          11519A              DATA    (XYZ*256+16-1)
                    11520
                    11521+      GENDT   20,0,0,0,                       CA 1 BLEED
0E1C' 0013          11541A              DATA    (XYZ*256+20-1)
                    11542+      GENDT   16,1,8,0,E                      TO CPU 2 BUFFER
0E1E' A10F          11562A              DATA    (XYZ*256+16-1)
                    11563
                    11564+      GENDT   24,0,0,0,                       WA 1 BLEED
0E20' 0017          11584A              DATA    (XYZ*256+24-1)
                    11585+      GENDT   16,1,9,0,E                      TO CPU 2 BUFFER
0E22' A50F          11605A              DATA    (XYZ*256+16-1)
                    11606
                    11607+      GENDT   39,0,0,0,                       ADP ON              50
0E24' 0026          11627A              DATA    (XYZ*256+39-1)
                    11628+      GENDT   16,1,10,0,E                     TO CPU 2 BUFFER
0E26' A90F          11648A              DATA    (XYZ*256+16-1)
                    11649
                    11650+      GENDT   40,0,1,0,                       ISOL VALVE OPEN (PW)
0E28' 0427          11670A              DATA    (XYZ*256+40-1)
                    11671+      GENDT   16,1,11,0,E                     TO CPU 2 BUFFER
0E2A' AD0F          11691A              DATA    (XYZ*256+16-1)
                    11692
                    11693+      GENDT   40,0,0,0,E                      EEC/PMC ON
0E2C' 8027          11713A              DATA    (XYZ*256+40-1)
                    11714+      GENDT   16,1,12,0,E                     TO CPU 2 BUFFER     55
0E2E' B10F          11734A              DATA    (XYZ*256+16-1)
                    11735
                    11736+      GENDT   40,0,0,0,E                      LATCHED FAULT
0E30' 8027          11756A              DATA    (XYZ*256+40-1)
                    11757+      GENDT   16,1,13,0,E                     TO CPU 2 BUFFER
0E32' B50F          11777A              DATA    (XYZ*256+16-1)
                    11778
```

```
                    10896+      GENDT   15,1,9,0,E                  TO CPU 2 BUFFER
 0DE2' A50E         10916A              DATA    (XYZ*256+15-1)
                    10917
                    10918+      GENDT   32,0,0,0,                   ADP ON
 0DE4' 001F         10938A              DATA    (XYZ*256+32-1)
                    10939+      GENDT   15,1,10,0,E                 TO CPU 2 BUFFER
 0DE6' A90E         10959A              DATA    (XYZ*256+15-1)
                    10960
                    10961+      GENDT   39,0,1,0,                   ISOL VALVE OPEN (GE)      20
 0DE8' 0426         10981A              DATA    (XYZ*256+39-1)
                    10982+      GENDT   15,1,11,0,E                 TO CPU 2 BUFFER
 0DEA' AD0E         11002A              DATA    (XYZ*256+15-1)
                    11003
                    11004+      GENDT   40,0,0,0,E                  EEC/PMC ON
 0DEC' 8027         11024A              DATA    (XYZ*256+40-1)
                    11025+      GENDT   15,1,12,0,E                 TO CPU 2 BUFFER
 0DEE' B10E         11045A              DATA    (XYZ*256+15-1)
                    11046
                    11047+      GENDT   40,0,0,0,E                  LATCHED FAULT (GE)
 0DF0' 8027         11067A              DATA    (XYZ*256+40-1)
                    11068+      GENDT   15,1,13,0,E                 TO CPU 2 BUFFER           25
 0DF2' B50E         11088A              DATA    (XYZ*256+15-1)
                    11089
                    11090+      GENDT   13,0,1,0,                   ECS BLEED (PW)
 0DF4' 040C         11110A              DATA    (XYZ*256+13-1)
                    11111+      GENDT   15,1,6,0,E                  TO CPU 2 BUFFER
 0DF6' 990E         11131A              DATA    (XYZ*256+15-1)
                    11132
                    11133+      GENDT   51,0,1,0,E                  ECS MODE (PW)
 0DF8' 8432         11153A              DATA    (XYZ*256+51-1)
                    11154+      GENDT   15,1,7,0,E                  TO CPU 2 BUFFER
 0DFA' 9D0E         11174A              DATA    (XYZ*256+15-1)
                    11175
                    11176+      GENDT   16,0,0,0,                   CA 1 BLEED (PW)           30
 0DFC' 000F         11196A              DATA    (XYZ*256+16-1)
                    11197+      GENDT   15,1,8,0,E                  TO CPU 2 BUFFER
 0DFE' A10E         11217A              DATA    (XYZ*256+15-1)
                    11218
                    11219+      GENDT   24,0,1,0,                   WA 1 BLEED (PW)
 0E00' 0417         11239A              DATA    (XYZ*256+24-1)
                    11240+      GENDT   15,1,9,0,E                  TO CPU 2 BUFFER
 0E02' A50E         11260A              DATA    (XYZ*256+15-1)
                    11261
                    11262+      GENDT   32,0,0,0,                   ADP ON (PW)
 0E04' 001F         11282A              DATA    (XYZ*256+32-1)
                    11283+      GENDT   15,1,10,0,E                 TO CPU 2 BUFFER           35
 0E06' A90E         11303A              DATA    (XYZ*256+15-1)
                    11304
                    11305+      GENDT   39,0,1,0,                   ISOL VALVE OPEN (PW)
 0E08' 0426         11325A              DATA    (XYZ*256+39-1)
                    11326+      GENDT   15,1,11,0,E                 TO CPU 2 BUFFER
 0E0A' AD0E         11346A              DATA    (XYZ*256+15-1)
```

```
                    10521  *                    N5      =       DATA SET END
                    10522  *                                    BLANK = CONTINUE
                    10523  *                                    END   = END
                    10524  *
 0DC0'              10525  DR1DT
                    10526  *
                    10527+         GENDT    5,0,1,0,                        N1 ACTUAL L (GE)          0
 0DC0'  0404        10547A                  DATA    (XYZ*256+5-1)
                    10548+         GENDT    146,1,1,0,E                      TO CPU 2 BUFFER
 0DC2'  8591        10568A                  DATA    (XYZ*256+146-1)
                    10569
                    10570+         GENDT    45,1,1,0,E          TLA                  - CPU 2 ONLY
 0DC4'  852D        10590A                  DATA    (XYZ*256+45-1)
                    10591
                    10592+         GENDT    46,1,0,0,E          TT2                  - CPU 2 ONLY
 0DC6'  812D        10612A                  DATA    (XYZ*256+46-1)
                    10613
                    10614+         GENDT    15,1,0,0,E            J MDL CODE 1 (GE)  - CPU 2 ONLY
 0DC8'  810E        10634A                  DATA    (XYZ*256+15-1)
                    10635
                    10636+         GENDT    15,1,1,0,E          ENG MDL CODE 2 (GE)  - CPU 2 ONLY   5
 0DCA'  850E        10656A                  DATA    (XYZ*256+15-1)
                    10657
                    10658+         GENDT    15,1,2,0,E          ENG MDL CODE 3 (GE)  - CPU 2 ONLY
 0DCC'  890E        10678A                  DATA    (XYZ*256+15-1)
                    10679
                    10680+         GENDT    15,1,3,0,E          ENG MDL CODE 4       - CPU 2 ONLY
 0DCE'  8D0E        10700A                  DATA    (XYZ*256+15-1)
                    10701
                    10702+         GENDT    15,1,4,0,E          ENG MDL CODE 5 (GE)  - CPU 2 ONLY
 0DD0'  910E        10722A                  DATA    (XYZ*256+15-1)
                    10723
                    10724+         GENDT    15,1,5,0,E          ENG MDL CODE 6 (GE)  - CPU 2 ONLY
 0DD2'  950E        10744A                  DATA    (XYZ*256+15-1)
                    10745
                    10746+         GENDT    13,0,1,0,                  ECS BLEED (GE)            10
 0DD4'  040C        10766A                  DATA    (XYZ*256+13-1)
                    10767+         GENDT    15,1,6,0,E                 TO CPU 2 BUFFER
 0DD6'  990E        10787A                  DATA    (XYZ*256+15-1)
                    10788
                    10789+         GENDT    51,0,1,0,E                 ECS MODE (GE)
 0DD8'  8432        10809A                  DATA    (XYZ*256+51-1)
                    10810+         GENDT    15,1,7,0,E                 TO CPU 2 BUFFER
 0DDA'  9D0E        10830A                  DATA    (XYZ*256+15-1)
                    10831
                    10832+         GENDT    16,0,0,0,                  CA 1 BLEED (GE)
 0DDC'  000F        10852A                  DATA    (XYZ*256+16-1)
                    10853+         GENDT    15,1,8,0,E                 TO CPU 2 BUFFER           15
 0DDE'  A10E        10873A                  DATA    (XYZ*256+15-1)
                    10874
                    10875+         GENDT    24,0,1,0,                  WA 1 BLEED
 0DE0'  0417        10895A                  DATA    (XYZ*256+24-1)
```

```
     0E3A' 0111 0112
     0E3C' 0113 0114
     0E3E' 0115 0116
     0E42' 0145
     0E44' 00E4 00E7   10501   DCYC7   DATA    228,231,233,88,91,92,93,94,95,96,279,281,282,283,284
     0E48' 00E9 0058
     0E4A' 005B 005C
     0E4C' 005D 005E
     0E4E' 005F 0060
     0E50' 0117 0119
     0E52' 011A 011B
     0E54' 011C
     0E56' 011D 011E   10502           DATA    285,286,287,288,289,290,291,292,293,294,295,296,297,298,328
     0E58' 011F 0120
     0E5A' 0121 0122
     0E5C' 0123 0124
     0E5E' 0125 0126
     0E60' 0127 0128
     0E62' 0129 012A
     0E64' 0148
     0E66' 00ED 0061   10503   DCYC8   DATA    237,97,98,99,100,101,102,103,104,105,106,107,299,300,301
     0E6A' 0062 0063
     0E6C' 0064 0065
     0E6E' 0066 0067
     0E70' 0068 0069
     0E72' 006A 006B
     0E74' 012B 012C
     0E76' 012D
     0E78' 012E 012F   10504           DATA    302,303,304,305,306,307,308,309,310,311,312,313,314,315
     0E7C' 0130 0131
     0E7E' 0132 0133
     0E80' 0134 0135
     0E82' 0136 0137
     0E84' 0138 0139
     0E86' 013A 013B
     0E88' 013C 013D   10505           DATA    316,317,329
     0E8A' 0149
                       10506   *
                       10507   ************************************************
                       10508   *
                       10509   *       DITS #1 PARAMETER DESTINATION INFORMATION
                       10510   *
                       10511   *       FORM: GENDT     N1,N2,N3,N4,N5
                       10512   *
                       10513   *                       N1      =       BUFFER OFFSET WORD
                       10514   *                       N2      =       BUFFER TYPE
                       10515   *                                       0 = DFDR
                       10516   *                                       1 = INTER-CPU
                       10517   *                       N3      =       LEFT SHIFT COUNT
                       10518   *                       N4      =       CLEAR OUTPUT WORD
                       10519   *                                       1 = CLEAR
                       10520   *                                       0 = NOT CLEAR
```

```
     0C72'  00A4 00A5
     0C76'  00A6 00A7
     0C7A'  00A8 00A9
     0C7E'  00AA 00AB
     0C82'  00AC 00AD
     0C86'  00AE 00AF
     0C8A'  0140
     0C8C'  0012 00CD   10494   DCYC4   DATA    18,205,207,209,211,213,215,217,219,221,223,176,177,178,179
     0C90'  00CF 00D1
     0C94'  00D3 00D5
     0C98'  00D7 00D9
     0C9C'  00DB 00DD
     0CA0'  00DF 00B0
     0CA4'  00B1 00B2
     0CA8'  00B3
     0CAA'  00B4 00B5   10495           DATA    180,181,182,183,184,185,186,187,188,189,190,191,192
     0CAE'  00B6 00B7
     0CB2'  00B8 00B9
     0CB6'  00BA 00BB
     0CBA'  00BC 00BD
     0CBE'  00BE 00BF
     0CC2'  00C0
     0CC4'  00C1 00C2   10496           DATA    193,194,322
     0CC8'  0142
     0CCA'  0014 0026   10497   DCYC5   DATA    20,38,72,74,227,58,59,60,61,195,196,197,198,199,200
     0CCE'  0048 004A
     0CD2'  00E3 003A
     0CD6'  003B 003C
     0CDA'  003D 00C3
     0CDE'  00C4 00C5
     0CE2'  00C6 00C7
     0CE6'  00C8
     0CE8'  00C9 00CA   10498           DATA    201,202,203,204,241,242,243,244,251,254,257,258,259,324
     0CEC'  00CB 00CC
     0CF0'  00F1 00F2
     0CF4'  00F3 00F4
     0CF8'  00FB 00FE
     0CFC'  0101 0102
     0D00'  0103 0144
     0D04'  00F5 00F7   10499   DCYC6   DATA    245,247,255,62,63,67,76,83,84,85,86,87,260,261,262,263,264
     0D08'  00FF 003E
     0D0C'  003F 0043
     0D10'  004C 0053
     0D14'  0054 0055
     0D18'  0056 0057
     0D1C'  0104 0105
     0D20'  0106 0107
     0D24'  0108
     0D26'  0109 010A   10500           DATA    265,266,267,268,269,270,271,272,273,274,275,276,277,278,326
     0D2A'  010B 010C
     0D2E'  010D 010E
     0D32'  010F 0110
```

```
                    10475  *
     0EC8' 0118     10476         DATA    280             DUMMY
     0ECA' 0118     10477         DATA    280             DUMMY
     0ECC' 0118     10478         DATA    280             DUMMY
     0ECE' 0118     10479         DATA    280             DUMMY
     0ED0' 0038     10480         DATA    56
     0ED2' 00FA     10481         DATA    250
     0ED4' 000C     10482         DATA    12
     0ED6' 0118     10483         DATA    280             DUMMY
     0ED8'          10484  DR1DBG
                    10485  *
                    10486  *      ALL SUBFRAMES CYCLES 1 TO 8
                    10487  *
     0E[illegible]' 0000 0051  10488  DCYC1  DATA    0,81,225,239,2,3,4,5,6,7,8,9,108,109,110,111,112,113,114,115
     0E[illegible]' 00E1 00EF
     0E[illegible]' 0002 0003
     0E[illegible]' 0004 0005
     0E[illegible]' 0006 0007
     0E[illegible]' 0008 0009
     0E[illegible]' 006C 006D
     0E[illegible]' 006E 006F
     0E[illegible]' 0070 0071
     0E[illegible]' 0072 0073
     0C[illegible]' 0074 0075  10489         DATA    116,117,118,119,120,122,124,126,128,130,132,318
     0C[illegible]' 0076 0077
     0C[illegible]' 0078 007A
     0C[illegible]' 007C 007E
     0C[illegible]' 0080 0082
     0C[illegible]' 0084 013E
     0C[illegible]' 000A 000E  10490  DCYC2  DATA    10,14,64,77,235,13,23,25,40,134,136,138,140,142
     0C[illegible]' 0040 004D
     0C[illegible]' 00EB 000D
     0C[illegible]' 0017 0019
     0C[illegible]' 0028 0086
     0C[illegible]' 0083 008A
     0C[illegible]' 008C 008E
     0C[illegible]' 008F 0090  10491         DATA    143,144,145,146,147,148,149,150,151,152,153,154,155,156,319
     0C[illegible]' 0091 0092
     0C[illegible]' 0093 0094
     0C[illegible]' 0095 0096
     0C[illegible]' 0097 0098
     0C[illegible]' 0099 009A
     0C[illegible]' 009B 009C
     0C[illegible]' 013F
     0C[illegible]' 0010 0044  10492  DCYC3  DATA    16,68,70,79,226,230,41,55,57,157,158,159,160,161
     0C[illegible]' 0046 004F
     0C[illegible]' 00E2 00E6
     0C[illegible]' 0029 0037
     0C[illegible]' 0039 009D
     0C[illegible]' 009E 009F
     0C[illegible]' 00A0 00A1
     0C[illegible]' 00A2 00A3  10493         DATA    162,163,164,165,166,167,168,169,170,171,172,173,174,175,320
```

```
0E[illegible]  0136 0137
0E[illegible]  0138 0139
0E[illegible]  013A 013B
0E[illegible]  013C 013D   10432          DATA    316,317,29,45
0E[illegible]  001D 002D
                           10433   *
                           10434   *
                           10435   *              ******************************
                           10436   ***************    GE ENGINES SECTION      ********************
                           10437   *              ******************************
                           10438   *
0F[illegible]              10439   DR1DA6
                           10440   *
                           10441   *       SUBFRAME 1 CYCLES 1 TO 8
                           10442   *
0E[illegible]  0118        10443           DATA    280             DUMMY
0E[illegible]  0118        10444           DATA    280             DUMMY
0E[illegible]  0141 0143   10445           DATA    321,323
0E[illegible]  0118        10446           DATA    280             DUMMY
0E[illegible]  0016 0059   10447           DATA    22,89
0E[illegible]  00FC        10448           DATA    252
0E[illegible]  0118        10449           DATA    280             DUMMY
0F[illegible]  0118        10450           DATA    280             DUMMY
                           10451   *
                           10452   *       SUBFRAME 2 CYCLES 1 TO 8
                           10453   *
0E[illegible]  0118        10454           DATA    280             DUMMY
0F[illegible]  0118        10455           DATA    280             DUMMY
0E[illegible]  0145        10456           DATA    325
0E[illegible]  0118        10457           DATA    280             DUMMY
0E[illegible]  0018        10458           DATA    24
0E[illegible]  00F9        10459           DATA    249
0E[illegible]  0118        10460           DATA    280             DUMMY
0E[illegible]  0118        10461           DATA    280             DUMMY
                           10462   *
                           10463   *       SUBFRAME 3 CYCLES 1 TO 8
                           10464   *
0E[illegible]  0118        10465           DATA    280             DUMMY
0F[illegible]  0118        10466           DATA    280             DUMMY
0E[illegible]  0118        10467           DATA    280             DUMMY
0E[illegible]  0118        10468           DATA    280             DUMMY
0E[illegible]  0036 005A   10469           DATA    54,90
0E[illegible]  00FD        10470           DATA    253
0E[illegible]  0079 007B   10471           DATA    121,123,125,127,129,131,133,135,137,139,141,66
0E[illegible]  007D 007F
0E[illegible]  0081 0083
0E[illegible]  0085 0087
0E[illegible]  0089 008B
0E[illegible]  008D 0042
0E[illegible]  0118        10472           DATA    280             DUMMY
                           10473   *
                           10474   *       SUBFRAME 4 CYCLES 1 TO 8
```

```
     0A92'  00C3 00C4
     0AA2'  00C5 00C6
     0AA8'  00C7 00C8
     0AAA'  00C9 00CA   10425          DATA    201,202,203,204,241,242,243,244,251,254,257,258,259,324
     0AB8'  00CB 00CC
     0AB2'  00F1 00F2
     0AB4'  00F3 00F4
     0AB6'  00FB 00FE
     0AC0'  0101 0102
     0AC2'  0103 0144
     0AC6'  00F5 00F7   10426  DPCYC6  DATA    245,247,255,76,83,84,85,86,87,260,261,262,263,264
     0ACC'  00FF 004C
     0ACE'  0053 0054
     0AD2'  0055 0056
     0AD4'  0057 0104
     0AD8'  0105 0106
     0ADA'  0107 010B
     0AE0'  0109 010A   10427          DATA    265,266,267,268,269,270,271,272,273,274,275,276,277,278,326
     0AE2'  010B 010C
     0AE8'  010D 010E
     0AEE'  010F 0110
     0AF2'  0111 0112
     0AF6'  0113 0114
     0AFA'  0115 0116
     0AFE'  0146
     0B00'  00E4 00E7   10428  DPCYC7  DATA    228,231,233,88,91,92,93,94,95,96,279,281,282,283,284
     0B08'  00E9 005B
     0B0C'  005B 005C
     0B10'  005D 005E
     0B14'  005F 0060
     0B18'  0117 0119
     0B1C'  011A 011B
     0B20'  011C
     0B22'  011D 011E   10429          DATA    285,286,287,288,289,290,291,292,293,294,295,296,297,298
     0B2A'  011F 0120
     0B2E'  0121 0122
     0B30'  0123 0124
     0B34'  0125 0126
     0B38'  0127 0128
     0B3C'  0129 012A
     0B3E'  00ED 0061   10430  DPCYC8  DATA    237,97,98,99,100,101,102,103,104,105,106,107,299,300,301
     0B46'  0062 0063
     0B4A'  0064 0065
     0B4E'  0066 0067
     0B52'  0068 0069
     0B56'  006A 006B
     0B5A'  012B 012C
     0B60'  012D
     0B62'  012E 012F   10431          DATA    302,303,304,305,306,307,308,309,310,311,312,313,314,315
     0B70'  0130 0131
     0B74'  0132 0133
     0B78'  0134 0135
```

```
     09DA'  007B 007A
     09DE'  007C 007E
     09E2'  0080 0082
     09E6'  0084 013E
     09EA'  0148 0149
     09EE'  001A 001E   10417   DPCYC2  DATA    26,30,42,77,235,40,134,136,138,140,142
     09F2'  002A 004D
     09F6'  00EB 0028
     09FA'  0086 0088
     09FE'  008A 008C
     0A02'  008E
     0A03'  008F 0090   10418           DATA    143,144,145,146,147,148,149,150,151,152,153,154,155,156,319
     0A07'  0091 0092
     0A0C'  0093 0094
     0A10'  0095 0096
     0A14'  0097 0098
     0A18'  0099 009A
     0A1C'  009B 009C
     0A20'  013F
     0A22'  0020 002E   10419   DPCYC3  DATA    32,46,48,79,226,230,41,157,158,159,160,161
     0A26'  0030 004F
     0A2A'  00E2 00E6
     0A2E'  0029 009D
     0A32'  009E 009F
     0A36'  00A0 00A1
     0A3A'  00A2 00A3   10420           DATA    162,163,164,165,166,167,168,169,170,171,172,173,174,175,320
     0A3E'  00A4 00A5
     0A42'  00A6 00A7
     0A46'  00A8 00A9
     0A4A'  00AA 00AB
     0A4E'  00AC 00AD
     0A52'  00AE 00AF
     0A56'  0140
     0A58'  0022 00CD   10421   DPCYC4  DATA    34,205,207,209,211,213,215,217,219,221,223,176,177,178,179
     0A5C'  00CF 00D1
     0A60'  00D3 00D5
     0A64'  00D7 00D9
     0A68'  00DB 00DD
     0A6C'  00DF 00B0
     0A70'  00B1 00B2
     0A74'  00B3
     0A75'  00B4 00B5   10422           DATA    180,181,182,183,184,185,186,187,188,189,190,191,192
     0A79'  00B6 00B7
     0A7E'  00B8 00B9
     0A82'  00BA 00BB
     0A86'  00BC 00BD
     0A8A'  00BE 00BF
     0A8E'  00C0
     0A8F'  00C1 00C2   10423           DATA    193,194,322
     0A93'  0142
     0A94'  0024 0032   10424   DPCYC5  DATA    36,50,52,227,195,196,197,198,199,200
     0A98'  0034 00E3
```

```
                        1          IDT     ICISR
                        2          SUBTTL  INTER CPU INTERRUPT SERVICE ROUTINE
                        3  ********************************************************************************
                        4  *                                                                              *
                        5  *  NAME: ICISR                  AUTH: N.COSTANTINIDES   *
                        6  *  VERSION: 1                   DATE: 17 DEC 81                            *
                        7  *                                                                              *
                        8  *  FUNCTION:    INTER-CPU INTERRUPT SERVICE ROUTINE.                         *
                        9  *               IT PROCESS INTER-CPU INTERRUPTS THAT SPECIFIES                *
                       10  *               TRANSMITTER AND/OR RECEIVER.  THE ICISR CALLS THE             *
                       11  *               TRANSMITTER INTERRUPT ROUTINE AND RECEIVER INTERRUPT          *
                       12  *               ROUTINE WHEN ITS INTERRUPT IS DETECTED.                       *
                       13  *                                                                             *
                       14  *  CALLING MODULE: INTER-CPU INTERRUPT (HARDWARE)                             *
                       15  *                                                                             *
                       16  *  CALLING SEQ: HARDWARE                                                      *
                       17  *                                                                             *
                       18  *  INPUTS: TMS9902 INTERFACE REG                                              *
                       19  *                                                                             *
                       20  *  OUTPUTS:     CALLS ICRC                                                    *
                       21  *               CALLS ICTX                                                    *
                       22  *                                                                             *
                       23  *  MODULE REFERENCED:   ICRC = INTER CPU RECEIVER INTERRUPT                   *
                       24  *                       ICTX = INTER CPU TRANSMITTER INTERRUPT                *
                       25  *                                                                             *
                       26  *  WORKSPACE AREA:  WPIC                                                      *
                       27  *                                                                             *
                       28  *  REGISTERS MODIFIED:  R11                                                   *
                       29  *                                                                             *
                       30  *  VERSION HISTORY:                                                           *
                       31  *                                                                             *
                       32  ********************************************************************************
       =0000           34          RSECT   ICISR
                       35  *** CALL NAME
                       36          INTERN ICISR
                       37  *** VARIABLES REFERENCED
                       38          EXTERN ICIER
                       39  *** MODULES REFERENCED
                       40          EXTERN ICRC,ICTX,RTSIN1
                       41  *** LIBRARY
                       42          INCLUDE ICEQUATE.LIB
                       64  *** REGISTER DEFINITION
       =000C           65  CRU     EQU     R12     CRU
                       66  ********************************************************************************
       =0000'          67  ICISR   EQU     $
  0000' 06A0 0004*     68          BL      @RTSIN1
  0004' 020C 0780      69          LI      CRU,C9902I
  0008' 1F 10          70          TB      16                      CHECK RECEIVER INTERRUPT
  000A' 1609           71          JNE     ICISR2                  JIF NOT
                       72  *
                       73  *** RECEIVER INTERRUPT
                       74  *
```

```
00[illegible] 0920 0003*    174  ICRC60  MOV   @TIMEL,@ICRTMR        RC-TIMER = LSW REAL TIMER
00[illegible] 0006*
              175  * EXIT
00[illegible] 02C3          176          MOV   R3,R11               UNSAVE LINK
00[illegible] 045B          177          RT                         RETURN
              178  ;*****************************************************************
00[illegible] 0003          179  ICRK    DATA  6                    NO. OF MSG'S IN 3 SEC(3*110CHAR/36CHAR PER MSG)
              180          END
```

```
                    122  *
00[illegible] 05A0 0004*  123          INC     @ICRCTR                 BUMP RC-MSG-BYTE-COUNTER
                    124  *
                    125  * CHECK MSG BYTE COUNTER WITH LIMIT AGAIN
                    126  *
0050' C160 0004*    127          MOV     @ICRCTR,R5
0074' 0285 0026     128          CI      R5,MBBRSZ               MSG BYTE SIZE
0078' 1629          129          JNE     ICRC60                  JIF MSG RECEPTION NOT COMPLETE
                    130  *
                    131  * MSG RECEPTION COMPLETE
                    132  *       CHECK SUMCHECK. ADD FROM BEG TO CHECKSUM WORD.
007A' C0A0 0005*    133          MOV     @ICRBSA,R2              MSG BUFFER START ADDR
007E' 0201 0012     134          LI      R1,MBWRSZ               MSG BUFFER WORD SIZE.
0082' 04C0          135          CLR     R0                      CLEAR SUM
0084' A032          136  20$     A       *R2+,R0                 ADD EVERY WORDS IN MSG BUFFER
0086' 0601          137          DEC     R1
0088' 15FD          138          JGT     20$                     LOOP
008A' A032          139          A       *R2+,R0                 FINALLY ADD TO SUMCHECK WORD
008E' 1305          140          JEQ     ICRC30                  JIF SUMCHECK OK
                    141  *** SUM CHECK ERROR. MS = HEX 9
                    142  ***     INCREMENT ERROR COUNTER. SET ERROR CODE. QUEUE ERROR JOB.
                    143  ***     CLEAR MSG BYTE COUNTER. SELECT NEXT AVAILABLE BUFFER.
0090' 0201 0000     144          LI      R1,0                    SUM CHECK ERROR INDICATOR FOR ICRERR
0094' 06A0 0009*    145          BL      @ICRERR
0098' 101A          146          JMP     ICRC60                  TO EXIT
                    147  *
                    148  * SUM CHECK OK.
                    149  *
                    150  *
009A' C1A0 0005*    151  ICRC30  MOV     @ICRBSA,R6
009E' C056          152          MOV     *R6,R1                  GET 1ST WORD OF MSG
00A0' 0241 00FF     153          ANDI    R1,>FF                  SAVE MSG TYPE ONLY
00A4' 8501 0008*    154          C       R1,@ICMMAX              ACTUAL .VS. MAX. NO.
00A8' 1205          155          JLE     ICRC40                  JIF LEGAL MSG TYPE
                    156  *** ILLEGAL MESSAGE TYPE ERROR. MS = HEX A
                    157  ***     INCREMENT ERROR COUNTER. SET ERROR CODE. QUEUE ERROR JOB.
                    158  ***     CLEAR MSG BYTE COUNTER. SELECT NEXT AVAILABLE BUFFER.
00AA' 0201 0002     159          LI      R1,2                    ILLEGAL MSG TYPE SPECIFIER FOR ICRERR
00AE' 06A0 0009*    160          BL      @ICRERR
00B2' 100D          161          JMP     ICRC60                  TO EXIT
                    162  *
                    163  * LEGAL & GOOD SUMCHECK MESSAGE.
                    164  *       QUEUE MSG DISPATCHER (JOB #1)
00B4' C1A0 0005*    165  ICRC40  MOV     @ICRBSA,R6
00B8' F5A0 0007*    166          SOCB    @BIT0,*R6               SET BUFFER BUSY BIT
00BC' C020 0005*    167          MOV     @ICRBSA,R0              BUFFER START ADDRESS
00C0' 0420 000B*    168          BLWP    @JBQUE                  QUEUE JOB. MSG DISPATCHER.
00C4' 0000          169          DATA    0                       JOB NUMBER (#1)
                    170  *
00C6' 04E0 0004*    171          CLR     @ICRCTR                 CLEAR RC-MSG-BYTE-COUNTER
00CA' 06A0 000A*    172          BL      @ICBSEL                 SELECT NEXT AVAILABLE BUFFER
                    173  *** SET RC-TIMER
```

```
                    71  ********************************************************************************
0040' C0CB          72  ICRC    MOV     R11,R3                  SAVE LINK
                    73  * CHECK RECEIVER ERROR
0042' 020C 0780     74          LI      CRU,C9902I
0046' 1F 09         75          TB      9                       RCVERR  BIT?
0048' 1602          76          JNE     10$                     JIF NO RECEIVE ERROR
                    77  ** RECEIVE ERROR, SAVE STATUS REG.
004A' 3420 0001*    78          STCR    @ICSREG,0               SAVE ALL 16 BIT OF STATUS REG.
                    79  * CHECK (REAL TIMER-RC TIMER) ) 3?
004E' C020 0003*    80  10$     MOV     @TIMEL,R0               GET LSW TIMER (1 COUNT = 10 MSEC)
0052' C1E0 0006*    81          MOV     @ICRTMR,R7
0056' 6007          82          S       R7,R0                   LSW TIMER - RC TIMER = R0
0058' 0740          83          ABS     R0
005A' 0280 0004     84          CI      R0,4
005E' 1A0A          85          JL      ICRC5                   JIF RC INT. OCCUR .LT. 30 MSEC
                    86  ** WAIT TIME FOR NEXT CHAR HAS ELAPSED
0060' 04E0 0002*    87          CLR     @ICRLME                 CONTINUOUS LONG MSG ERROR CTR
0064' C020 0004*    88          MOV     @ICRCTR,R0
0068' 1315          89          JEQ     ICRC10                  JIF NOT ERROR. GO READ 1ST CHAR.
                    90  *** SHORT BYTE COUNT ERROR. MS = HEX B
                    91  ***     INCREMENT ERROR COUNTER. SET ERROR CODE. QUEUE ERROR JOB.
                    92  ***     CLEAR MSG BYTE COUNTER. SELECT NEXT AVAILABLE BUFFER.
006A' 0201 0004     93          LI      R1,4                    SHORT BYTE COUNT ERROR SPECIFIER
006E' 06A0 0009*    94          BL      @ICRERR
0072' 1010          95          JMP     ICRC10
                    96  * CHECK MSG BYTE COUNTER WITH LIMIT
0074' C160 0004*    97  ICRC5   MOV     @ICRCTR,R5
0078' 0225 0076     98          CI      R5,MBBASZ               MSG BYTE SIZE
007C' 1A0B          99          JL      ICRC10                  JIF MSG RECEPTION NOT COMPLETE
                   100  *** LONG BYTE COUNT ERROR. MS = HEX C
                   101  ***     INCREMENT ERROR COUNTER. SET ERROR CODE. QUEUE ERROR JOB.
                   102  ***     CLEAR MSG BYTE COUNTER. SELECT NEXT AVAILABLE BUFFER.
007E' 05A0 0002*   103          INC     @ICRLME                 CONTINOUS LONG MSG ERROR COUNTER
0082' 8B20 0002*   104          C       @ICRLME,@ICRK
0086' 00D2'
0088' 1501         105          JGT     16$                     JIF NOT CONTINOUS RECEIVER INTERRUPT FOR 3 SEC
                   106  ***     DISABLE INTER-CPU RECEIVER INTERRUPT
008A' 1E 12        107          SBZ     18                      DISABLE RECEIVER
008C' 0201 0006    108  16$     LI      R1,6                    LONG BYTE COUNT ERROR SPECIFIER FOR ICRERR
0090' 06A0 0009*   109          BL      @ICRERR
                   110  *
                   111  * READ CHAR AND STORE IT IN BUFFER POINTED BY ICRBSA
                   112  * AND ICRCTR. RESET INTERRUPT
                   113  *
0094' 04C0         114  ICRC10  CLR     R0
0096' 3600         115          STCR    R0,8                    READ CHAR
0098' 1D 12        116          SBO     18                      RESET RECEIVE INTERRUPT
009A' C1A0 0005*   117          MOV     @ICRBSA,R6
009E' C046         118          MOV     R6,R1                   GET MSG BUFFER START ADDR
00A0' C160 0004*   119          MOV     @ICRCTR,R5
00A4' A045         120          A       R5,R1                   ADD BYTE COUNTER
00A6' D440         121          MOVB    R0,*R1                  STORE CHAR
```

```
                        54          EXTERN  ICRCTR,ICRBSA,ICRTMR
                        55  *** CONSTANTS REFERENCED
                        56          EXTERN  BIT0
                        57  *** TABLES REFERENCED
                        58          EXTERN ICMMAX
                        59  *** MODULES REFERENCED
                        60          EXTERN  ICRERR,ICRSEL
                        61          EXTERN  JBQUE
                        62  *** LIBRARY
                        63          INCLUDE ICEQUATE.LIB
=0780                   64  U9902I          EQU     >780
=0026                   65  MBERSZ          EQU     38
=0012                   66  MBWRSZ          EQU     18
                        67
                        68  *** REGISTER DEFINITION
=000C                   69  CRU     EQU     R12             CRU
```

```
                1          IDT     ICRC
                2          SUBTTL  INTER CPU RECEIVER INTERRUPT
                3  ********************************************************************
                4  *                                                                  *
                5  * NAME: ICRC                    AUTHUR: N.CONSTANTINIDES           *
                6  * VERSION: 1                    DATE: 19-JUL-82                    *
                7  *                                                                  *
                8  * FUNCTION:     INTER-CPU RECEIVER INTERRUPT.                      *
                9  *               IT PROCESSES INTER-CPU RECEIVE INTERRUPTS THAT SIGNAL  *
               10  *               THE CHARACTER RECEPTIONS, CHECKS FOR RECEIVER ERRORS  *
               11  *               AND PLACES MESSAGE ERROR JOB ON THE JOB QUEUE, AND    *
               12  *               ALSO CHECKS FOR LAST CHARACTER AND PLACES MESSAGE     *
               13  *               DISPATCHER JOB IN THE JOB QUEUE.                   *
               14  *                                                                  *
               15  *               THE RECEIVER ERRORS ARE AS FOLLOWS:                *
               16  *               1. SUM CHECK ERRORS                                *
               17  *               2. ILLEGAL MESSAGE TYPE                            *
               18  *               3. SHORT BYTE COUNT                                *
               19  *               4. LONG BYTE COUNT                                 *
               20  *                                                                  *
               21  *               IF RECEIVER INTERRUPT OCCURES CONTINUOUSLY FOR 3 SEC  *
               22  *               (327 BYTES AT 1200 BAR), RECEIVER INTERRUPT WILL BE   *
               23  *               DISABLED.                                          *
               24  *                                                                  *
               25  * CALLING MODULE: ICISR = INTER-CPU ISR                            *
               26  *                                                                  *
               27  * CALLING SEQ: BL @ICRC                                           *
               28  *                                                                  *
               29  * INPUTS: NONE                                                     *
               30  *                                                                  *
               31  * OUTPUTS:      RECEIVE CHARACTER IN RECEIVE BUFFER                *
               32  *               JOB PLACED IN JOB QUEUE                            *
               33  *                                                                  *
               34  * MODULE REFERENCED:    JBQUE   = QUE JOB                          *
               35  *                       ICRERR  = RECEIVER ERROR                   *
               36  *                       ICBSEL  = SELECT NEXT RECEIVE BUFFER       *
               37  *                                                                  *
               38  * WORKSPACE AREA: WPIC                                             *
               39  *                                                                  *
               40  * REGISTERS MODIFIED:   R0,R1,R2,R3                                *
               41  *                       R5,R6,R7,R8,R9                             *
               42  *                       R11,R12                                    *
               43  *                                                                  *
               44  * VERSION HISTORY:                                                 *
               45  *                                                                  *
               46  ********************************************************************
=0000          48         RSECT   ICRC
               49  *** CALL NAME
               50         INTERN  ICRC
               51  *** VARIABLES REFERENCED
               52         EXTERN  ICSREG,ICRLME
               53         EXTERN  TIMEL
```

INTERCPU TRANSMITTER INTERRUPT ICTX.SRC

```
0000' 0720 0001*    75          SETO   @ICIFL6              SET IC INTERRUPT FLAG
0004' C28A          76          MOV    ICTACT,ICTACT   PROTECT FROM ERRONEOUS INTERRUPT
0006' 1315          77          JEQ    10$           JIF INTER-CPU TX NOT ACTIVE
0008' 0288 0200     78          CI     R8,MBBSIZ            CTR .VS. BUFFER BYTE SIZE
000C' 1412          79          JNE    10$                  JIF TRANSMISSION IN PROGRESS
                    80  **
000E' C049          81          MOV    ICTBSA,R1            GET MSG-BUFFER-START-ADDR
0010' A048          82          A      R8,R1                ADD BYTE COUNT
0012' 020C 0780     83          LI     CRU,C9902I
0016' 1D 10         84          SBO    16                   TURN ON THE TRANSMITTER
0018' 3211          85          LDCR   *R1,8                OUTPUT TO CPU
001A' 5460 0002*    86          SZCB   @XFFFF,*R1           CLEAR OUTPUT BYTE FOR DATA ACQUISITON
001E' 0588          87          INC    R8                   BUMP MSG COUNTER
0020' 0288 0200     88          CI     R8,MBBSIZ            CTR. VS. BUFFER BYTE SIZE
0024' 1A08          89          JL     100$                 JIF NOT THE LAST CHARACTER
                    90
                    91  * TRANSMITTED LAST CHARACTER
                    92
0026' 5660 0002*    93          SZCB   @XFFFF,*R9           RESET MS FIELD = TX COMPLETE
002A' 04C8          94          CLR    R8                   RESET IC TX COUNTER
002C' 1E 13         95          SBZ    19                   DISABLE TX INTERRUPT
002E' 04CA          96          CLR    ICTACT               SET TX NOT ACTIVE
0030' 1002          97          JMP    100$
                    98  ***
0032' 1E 13         99  10$     SBZ    19                   DISABLE TX INTERRUPT
0034' 1000         100          JMP    ICTEXT
                   101  *
0036'              102  100$
                   103  ***
0036' 045B         104  ICTEXT  RT                          RETURN
                   105          END
```

No errors detected

```
                    1          IDT     ICTX
                    2          SUBTTL  INTERCPU TRANSMITTER INTERRUPT
                    3  ********************************************************************************
                    4  *                                                                              *
                    5  * NAME: ICTX                              AUTH: M.COSTANTINIDES   *
                    6  * VERSION: 2                              DATE: 03-JAN-84                 *
                    7  *                                                                              *
                    8  * FUNCTION:     INTER-CPU TRANSMITTER INTERRUPT.                              *
                    9  *               IT PROCESS INTER-CPU TRANSMITTER INTERRUPTS THAT SIGNALS*
                   10  *               THE COMPLETION OF CHARACTER TRANSMISSION.  IT WRITES    *
                   11  *               THE NEXT CHARACTER IF ALL THE CHARACTERS ARE NOT        *
                   12  *               TRANSMITTED IN A MESSAGE.  WHEN THE LAST CHARACTER      *
                   13  *               IS TRANSMITTED, THE ICTX SETS THE 'TRANSMISSION         *
                   14  *               COMPLETE FLAG' BIT IN THE MS FIELD (1ST BYTE) OF        *
                   15  *               MESSAGE BUFFER AND TURNS OFF THE TRANSMITTER (RTSON).   *
                   16  *               AFTER EVERY BYTE TRANSMISSION, THE BYTE SHALL BE CLEARED*
                   17  *               IN THE MESSAGE BUFFER FOR THE DATA ACQUISITON TASK.     *
                   18  *                                                                              *
                   19  *                                                                              *
                   20  * CALLING MODULE: ICISR = INTER-CPU ISR                                        *
                   21  *                                                                              *
                   22  * CALLING SEQ: BL @ICTX                                                       *
                   23  *                                                                              *
                   24  * INPUTS: CRU = C9902I                                                         *
                   25  *                                                                              *
                   26  * OUTPUTS: WRITE NEXT CHARACTER FROM MSG BUFFER                                *
                   27  *                                                                              *
                   28  * MODULE REFERENCED: NONE                                                      *
                   29  *                                                                              *
                   30  * WORKSPACE AREA: WPIC                                                         *
                   31  *                                                                              *
                   32  * REGISTERS MODIFIED: R1,R8                                                    *
                   33  *                                                                              *
                   34  * VERSION HISTORY:                                                             *
                   35  *                                                                              *
                   36  ********************************************************************************
=0000              38         RSECT   ICTX
                   39  *** CALL NAME
                   40         INTERN  ICTX
                   41  *** VARIABLES REFERENCED
                   42         EXTERN  ICIFLG
                   43  *** CONSTANTS REFERENCED
                   44         EXTERN  XFFFF
                   45  *** LIBRARY
                   46         INCLUDE ICEQUATE.LIB
                   68  *** REGISTER DEFINITION
=000A              69  ICTACT  EQU     R10             INTER-CPU TX ACTIVE FLAG (-1)
=000C              70  CRU     EQU     R12             CRU
=0008              71  ICTCTR  EQU     R8              BYTE COUNTER
=0009              72  ICTBSA  EQU     R9              BUFFER START ADDRESS
                   73  ********************************************************************************
=0000'             74  ICTX    EQU     $
```

```
                    50  ********************************************************************************
0000' C08B          51  ICRERR  MOV     R11,R2                  SAVE LINK
                    52  * INCREMENT ERROR COUNTER, SET ERROR CODE.
0002' 05A1 0001*    53          INC     @ICREBF(R1)             BUMP ERROR COUNTER
0006' C1A0 0003*    54          MOV     @ICRBSA,R6              BUFFER START ADDRESS
000A' 04D6          55          CLR     *R6                     MAKE RX MSG BUFFER AVAILABLE
000C' E821 0005*    56          SOC     @ICRETB(R1),@ICRER8             SET ERROR BIT DEFINED IN ICRETB
0010' 0004*
                    57  * RESET MSG BYTE COUNTER, SELECT AVAILABLE BUFFER.
0012' 04E0 0002*    58          CLR     @ICRCTR                 CLEAR RC-MSG-BYTE-COUNTER
0016' 06A0 0006*    59          BL      @ICRSEL                 SELECT NEXT AVAILABLE BUFFER
001A' C2C2          60          MOV     R2,R11                  UNSAVE LINK
001C' 045B          61          RT                              RETURN
                    62  *
                    63          END
```

No errors detected

```
                    1          IDT     ICRERR
                    2          SUBTTL  INTER-CPU       RECEIVER ERROR
                    3  ********************************************************************************
                    4  *                                                                              *
                    5  * NAME: ICRERR                  AUTH: N.COSTANTINIDES   *
                    6  * VERSION: 1                    DATE: 17 DEC 81                              *
                    7  *                                                                              *
                    8  * FUNCTION:     INTER-CPU RECEIVER ERROR.                                      *
                    9  *               WHEN RECEIVER INTERRUPT (ICRC) DETECTS AN ERROR THIS           *
                   10  *               ROUTINE IS CALLED WITH ERROR TYPE.  THE ICRERR                 *
                   11  *               INCREMENTS THE ERROR COUNTER, SETS ERROR BIT IN THE            *
                   12  *               RECEIVED MSG ERROR WORD.                                       *
                   13  *               IT ALSO CLEARS THE RC-MSG-BYTE COUNTER                         *
                   14  *               AND CALLS 'ICBSEL' TO SELECT AVAILABLE MSG BUFFER.             *
                   15  *                                                                              *
                   16  *                                                                              *
                   17  * CALLING MODULE: ICRC = INTER-CPU RECEIVE INTERRUPT                           *
                   18  *                                                                              *
                   19  * CALLING SEQ: BL @ICRERR                                                     *
                   20  *                                                                              *
                   21  * INPUTS: R1 = ERROR TYPE, 0 FOR SUM CHECK ERROR (B0,LSB)                      *
                   22  *                          2 FOR ILLEGAL MSG TYPE (B1)                         *
                   23  *                          4 FOR SHORT BYTE COUNT (B2)                         *
                   24  *                          6 FOR LONG BYTE COUNT (B3)                          *
                   25  *                                                                              *
                   26  * OUTPUTS:      ICREBF +0,+2,+4,OR +6 INCREMENTED.                             *
                   27  *               ICRER8 = ERROR BIT                                             *
                   28  *               ICRCTR=0                                                       *
                   29  *                                                                              *
                   30  * MODULE REFERENCED: ICBSEL = SELECT AVILABLE BUFFER                           *
                   31  *                                                                              *
                   32  * WORKSPACE AREA: WPIC                                                         *
                   33  *                                                                              *
                   34  * REGISTERS MODIFIED: R0,R2,R5,R6,R11                                          *
                   35  *                                                                              *
                   36  * VERSION HISTORY:                                                             *
                   37  *                                                                              *
                   38  ********************************************************************************
      =0000        39          RSECT   ICRERR
                   40  ### CALL NAME
                   41          INTERN  ICRERR
                   42  ### VARIABLES REFERENCED
                   43          EXTERN  ICREBF,ICRCTR,ICRBSA,ICRER8
                   44  ### TABLES REFERENCED
                   45          EXTERN ICRETB
                   46  ### MODULES REFERENCED
                   47          EXTERN ICBSEL
                   48  ### REGISTERS DEFINITION
```

```
                    1          IDT    ICROM
                    2          SUBTTL  INTER-CPU ROM DEFENITIONS
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: ICROM.SRC                      AUTH: N.COSTANTINIDES      *
                    6  * VERSION: 1                           DATE: 31-MAR-1982          *
                    7  *                                                                  *
                    8  * FUNCTION: ALL INTER-CPU ROM DEFENITIONS                         *
                    9  *                                                                  *
                   10  * CALLING MODULES: MOST INTER-CPU COMMUNICATION MODULES            *
                   11  *                                                                  *
                   12  * VERSION HISTORY:                                                *
                   13  *                                                                  *
                   14  ********************************************************************
        =0000      15          RSECT   ICROM
                   16  *** CALL NAME
                   17          INTERN  ICRETB,ICRBTB
                   18          INTERN  ICMMAX,ICMDTB,XFFFF,BIT0
                   19  *** VARIABLES REFERENCED
                   20  *** CONSTANTS REFERENCED
                   21  *** TABLES REFERENCED
                   22  *** MODULES REFERENCED
                   23          EXTERN  ICRBF1,ICRBF2,RMSG0,RMSG1,RMSG2
                   24  *** LIBRARY
                   25  *** REGISTERS DEFINITION
                   26  ********************************************************************
                   27  * INTER-CPU RECEIVER ERROR CODE TABLE (MS FIELD)
                   28  * SET IN ICRERB WORD
0000' 0001         29  ICRETB  DATA    1               RC SUM CHECK ERROR
0002' 0002         30          DATA    2               RC ILLEGAL MSG TYPE ERROR
0004' 0004         31          DATA    4               RC SHORT BYTE COUNT ERROR
0006' 0008         32          DATA    8               RC LONG BYTE COUNT ERROR
                   33  ********************************************************************
                   34  * INTER-CPU RECEIVER BUFFER START ADDRESS TABLE
0008' 0002         35  ICRBTB  DATA    2               NUMBER OF BUFFERS
000A' 0001*        36          DATA    ICRBF1          BUFFER#1 START ADDRESS
000C' 0002*        37          DATA    ICRBF2          BUFFER#2 START ADDRESS
                   38  ********************************************************************
                   39  * INTER-CPU RECEIVER MSG TYPE SUBROUTINE TABLE
000E' 0002         40  ICMMAX  DATA    2               MAXIMUM NUMBER OF MSG'S LESS 1
0010' 0003*        41  ICMDTB  DATA    RMSG0           RECEIVE MSG TYPE#0 SUBR ADDR (PARA DATA)
0012' 0004*        42          DATA    RMSG1           RECEIVE MSG TYPE#1 SUBR ADDR (DOC DATA)
0014' 0005*        43          DATA    RMSG2           RECEIVE MSG TYPE #2 (SYSTEM STATUS)
                   44  *
0016' FFFF         45  XFFFF   DATA    >FFFF
0018' 8000         46  BIT0    DATA    >8000
                   47  ********************************************************************
                   48          END
```

No errors detected

```
                    53  **              BIT10=MSG TYPE#4 RECEIVED OK
0024'  =0002        54  TMA     BSS     2
                    55  **      INTER-CPU RECEIVER BUFFERS
                    56  **              LISTED IN RECEIVER BUFFER START ADDRESS TABLE(ICRBTB)
0026'  =0080        57  ICRBF1  BSS     64*2            RECEIVER BUFFER #1
00A6'  =0080        58  ICRBF2  BSS     64*2            RECEIVER BUFFER #2
                    59  **
                    60  ** SAVE LINK REGISTER (R11)WORD
0126'  =0002        61  LNKSV0  BSS     2
                    62  ********************************************************************
                    63  * INTER-CPU INTERRUPT SERVICE ROUTINE WORKSPACE AREA
                    64  *
0128'  =0002        65  WPIC    BSS     2               R0
       =0006        66          BSS     6
       =0002        67          BSS     2               R4
0132'  =0002        68  ICRCTR  BSS     2               R5
0134'  =0002        69  ICRBSA  BSS     2               R6
0136'  =0002        70  ICRTMR  BSS     2               R7
0138'  =0002        71  ICTCTR  BSS     2               R8
013A'  =0002        72  ICTBSA  BSS     2               R9
013C'  =0002        73  ICTACT  BSS     2               R10 TX ACTIVE FLAGE (-1 ACTIVE, 0 INACTIVE)
       =000A        74          BSS     10
0148'  =0002        75  ICIFLG  BSS     2               IC INTERRUPT FLAG
014A'               76  ICERAM
                    77  *
                    78          END
```

No errors detected

```
                        1          IDT    ICRAM
                        2          SUBTTL  INTER-CPU RAM DEFENITIONS
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: ICRAM.SRC                      AUTH: N.COSTANTINIDES      *
                        6  * VERSION: 1                           DATE: 3-JAN-84             *
                        7  *                                                                  *
                        8  * FUNCTION: INTER-CPU RAM DEFENITIONS                             *
                        9  *                                                                  *
                       10  * CALLING MODULES: MOST INTE-CPU COMMUNICATION MODULES            *
                       11  *                                                                  *
                       12  * VERSION HISTORY:                                               *
                       13  *                                                                  *
                       14  ********************************************************************
          =0000        15          RSECT   ICRAM
                       16  *** CALL NAME
                       17          INTERN  ICBRAM,ICIER,ICSREG,ICREBF,ICRER8,ICRER9,ICRLME
                       18          INTERN  ICIQP,ICOQP,ICBQUE,ICEQUE,TMA,ICRBF1,ICRBF2
                       19          INTERN  LNKSVO,ICERAM,WPIC,ICRCTR,ICRBSA,ICRTMR,ICTCTR,ICTBSA
                       20          INTERN  ICTACT,ICIFLG
                       21  *** VARIABLES REFERENCED
                       22  *** CONSTANTS REFERENCED
                       23  *** TABLES REFERENCED
                       24  *** MODULES REFERENCED
                       25  *** LIBRARY
                       26  *** REGISTERS DEFINITION
                       27  ********************************************************************
                       28  *  INTER-CPU RAM LOCATIONS BEGINS AT ICBRAM AND ENDS AT ICERAM
0000'                  29  ICBRAM
                       30  **      INTER-CPU ISR
0000'  =0002          31  ICIER   BSS     2              COUNTS INTERRUPTS OTHER THAN TX & RX (ICISR)
0002'  =0002          32  ICSREG  BSS     2              RECEIVER INPUT REGISTER (HARDWARE)
                       33  **      INTER-CPU RECEIVER ERROR COUNTER BUFFER
0004'  =0002          34  ICREBF  BSS     2              RECEIVER SUM CHECK ERROR
       =0002          35          BSS     2              ILLEGAL MSG TYPE ERROR
       =0002          36          BSS     2              SHORT BYTE COUNT ERROR
       =0002          37          BSS     2              LONG BYTE COUNT ERROR
                       38  **
000C'  =0002          39  ICRER8  BSS     2              RECEIVED MESSAGE ERROR WORD(BIT PATTERN DEFINED IN ICRETB)
000E'  =0002          40  ICRER9  BSS     2              BUFFER NOT AVAILABLE ERROR
0010'  =0002          41  ICRLME  BSS     2              CONTINUOUS LONG MSG ERROR CTR(ICRC)
                       42  **      INTER-CPU INPUT QUEUE
0012'  =0002          43  ICIQP   BSS     2              INPUT QUEUE POINTER
0014'  =0002          44  ICOQP   BSS     2              OUTPUT QUEUE POINTER
0016'  =000C          45  ICBQUE  BSS     12             QUEUE BUFFER (BEG)
0022'  =0002          46  ICEQUE  BSS     2              END OF QUEUE BUFFER
                       47  **
                       48  **      TMA=INTER-CPU TRANSMIT MSG ACKNOWLEDGE FLAG BIT
                       49  **      REFERENCED BY ICQMSG
                       50  **              BIT7=MSG TYPE#1 RECEIVED OK
                       51  **              BIT8=MSG TYPE#2 RECEIVED OK
                       52  **              BIT9=MSG TYPE#3 RECEIVED OK
```

```
          =000C          201   CRU     EQU     R12
                         202   ********************************************************************************
          =0000'         203   ICINIT  EQU     $
                         204   *** CLEAR ICPU RAM AREA
    0000' 0201 0005*     205           LI      R1,ICBRAM               START ADDRESS
    0004' 04F1           206   10$     CLR     *R1+
    0006' 0281 0006*     207           CI      R1,ICERAM               END?
    000A' 1AFC           208           JL      10$                     NO
                         209   *     INIT INTER-CPU QUEUE POINTER
    000C' 0201 0001*     210           LI      R1,ICBQUE               BEG ADDRESS OF INTER-CPU QUEUE
    0010' C801 0002*     211           MOV     R1,@ICIQP                 TO INPUT POINTER
    0014' C801 0003*     212           MOV     R1,@ICOQP                 AND TO OUTPUT POINTER
                         213   *
                         214   *  INIT INTER-CPU RECEIVER BUFFER POINTER
    0018' 0201 0008*     215           LI      R1,ICRBF1               FIRST RECEIVER BUFFER
    001C' C801 0007*     216           MOV     R1,@ICRBSA              RECEIVER BUFFER START ADDRESS
    0020' 020C 0780      217           LI      CRU,C9902I              INTER-CPU 9902 CRU BASE ADDR
    0024' 1D 1F          218           SBO     31                      ISSUE RESET COMMAND (BIT 31=1)
    0026' 3220 0044'     219           LDCR    @CNTRL,8                LOAD CONTROL AND RESET LDCTRL
    002A' 3220 0045'     220           LDCR    @INTVL,8                LOAD INTERVAL AND RESET LDIR
    002E' 32E0 0046'     221           LDCR    @RDR,11                 LOAD RDR AND RESET LRDR
    0032' 3320 0048'     222           LDCR    @XDR,12                 LOAD XDR AND RESET LXDR
    0036' 1D 12          223           SBO     18                      ENABLE RECEIVE INTERRUPT
                         224   *** ENABLE INTER-CPU INTERRUPT IN TMS 9901
    0038' 020C 0700      225           LI      CRU,C9901
    003C' 1D 07          226           SBO     7
                         227   ***
    003E' 04E0 0004*     228           CLR     @ICTACT         SET INTER-CPU TX NOT ACTIVE
    0042' 045B           229           RT                              RETURN
                         230   ***
    0044' B3             231   CNTRL   BYTE    >B3                     ONE STOP BIT, ODD PARITY, 8 BIT CHAR
    0045' 9C             232   INTVL   BYTE    10000/64                10 MSEC INTERVAL
    0046' 01A1           233   RDR     DATA    417             1200 BAUD FOR CPU #1 RECEIVER(1E+6/(2*1200))
    0048' 0034           234   XDR     DATA    52              9600 BAUD FOR CPU #1 TRANSMITTER
                         235   ***
                         236           END

No errors detected
```

```
                    1          IDT     ICINIT
                    2          SUBTTL  INTER-9902 INITIALIZATION
                    3  ********************************************************************************
                    4  *                                                                              *
                    5  * NAME: ICINIT                          AUTH: N.COSTANTINIDES   *
                    6  * VERSION: 1                            DATE: 8 DEC 81                        *
                    7  *                                                                              *
                    8  * FUNCTION:     INTER-CPU TMS9902 INITIALIZATION.                              *
                    9  *               IT INITIALIZES THE HARDWARE REGISTERS FOR THE INTER-CPU *
                   10  *               COMMUNICATION OPERATION.                                       *
                   11  *               THE CHARACTER LENGTH OF 8 BITS IN SELECTED PLUS ONE     *
                   12  *               STOP BIT AND ODD PARITY BIT.                                   *
                   13  *               THE TRANSMIT DATA RATE IS SET TO 9600 BAUD FOR CPU #1   *
                   14  *               AND 1200 BAUD FOR CPU #2.                                      *
                   15  *               THE RECEIVE DATA RATE IS SET TO 1200 BAUD FOR CPU #1    *
                   16  *               AND 9600 BAUD FOR CPU #2.                                      *
                   17  *               BOTH TRANSMIT AND RECEIVE INTERRUPTS ARE ENABLED.      *
                   18  *                                                                              *
                   19  * CALLING MODULES: POWON (POWER ON ISR)                                        *
                   20  *                                                                              *
                   21  * CALLING SEQ: BL @ICINIT                                                     *
                   22  *                                                                              *
                   23  * INPUTS: NONE                                                                *
                   24  *                                                                              *
                   25  * OUTPUTS: CONTROL REG = ONE STOP BIT, ODD PARITY, AND 8 BIT CHAR.     *
                   26  *          INTERVAL REG= 10 MSECOND                                            *
                   27  *          RECEIVE DATA RATE REG = 1200 BAUD FOR CPU #1                *
                   28  *                               = 9600 BAUD FOR CPU #2                 *
                   29  *          TRANSMIT DATA RATE REG= 9600 BAUD FOR CPU #1                *
                   30  *                               = 1200 BAUD FOR CPU #2                 *
                   31  *          TRANSMIT BUFFER AND RECEIVE INTERRUPTS ENABLED.             *
                   32  *                                                                              *
                   33  * MODULES REFERENCED: NONE                                                    *
                   34  *                                                                              *
                   35  * WORKSPACE AREA: CALLER'S                                                    *
                   36  *                                                                              *
                   37  * REGISTERS MODIFIED: R1,R12                                                  *
                   38  *                                                                              *
                   39  * VERSION HISTORY:                                                            *
                   40  *                                                                              *
                   41  ********************************************************************************
=0000              42          RSECT   ICINIT
                   43  *** CALL NAME
                   44          INTERN  ICINIT
                   45  *** VARIABLES
                   46          EXTERN  ICBQUE,ICIQP,ICOQP,ICTACT
                   47          EXTERN  ICBRAM,ICERAM
                   48          EXTERN  ICRBSA,ICRBF1
                   49  *** LIBRARY
                   50          INCLUDE CNSTNT
                  178          INCLUDE ICEQUATE.LIB
                  200  *** REGISTER DEFINITION
```

```
                    53  ********************************************************************
0000' 0003*         54  EXTIVE  DATA    WPEX            WORKSPACE POINTER SAME AS NEWJOB
0002' 0004'         55          DATA    EXT1
                    56  *
0004'               57  EXT1
0004' 8209          58          C       JBOPTR,JBIPTR           JOB WAITING?
0006' 1601          59          JNE     EXT2            JIF THERE IS
0008' 10FD          60          JMP     EXT1            LOOP
                    61  ********************************
                    62  * CHECK QUED UP JOB
000A'               63  EXT2
000A' 0289 0002*    64          CI      JBOPTR,JBEBUF   IS POINTER AT END OF BUFFER
000E' 1A02          65          JL      EXT3            JIF NOT
0010' 0209 0001*    66          LI      JBOPTR,JBBBUF   ELSE SET POINTER TO TOP OF LIST
0014'               67  EXT3
0014' C079          68          MOV     *JBOPTR+,R1             GET INDEX
0016' C039          69          MOV     *JBOPTR+,R0     BUFFER POINTER FOR JOB
0018' 8801 0006*    70          C       R1,@JBMAX               CHECK FOR MAX JOB NUMBER
001C' 1202          71          JLE     EXT4            JIF OK
                    72  ********************************
                    73  * BAD JOB, GO TO POWER ON ISR
001E' 0420 0004*    74          BLWP    @INOVCT         ELSE GO TO POWER-ON VIA VECTOR
                    75  ********************************
                    76  * EXECUTE QUEUED JOB
0022'               77  EXT4
0022' 0A21          78          SLA     R1,2            *4
0024' 0221 0005*    79          AI      R1,JBTBL        POINT TO JOB VECTOR
0028' 0411          80          BLWP    *R1             EXECUTE WAITING JOB WITH R0=BUFFER POINTER
002A' 10EC          81          JMP     EXT1            LOOP
                    82  *
                    83          END
```

```
                    1          IDT    EXTIVE
                    2          SUBTTL  BACKGROUND EXECUTIVE
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: EXTIVE (DFBAU)                AUTH: N.COSTANTINIDES      *
                    6  * VERSION: 1                          DATE: 19-FEB-82            *
                    7  *                                                                  *
                    8  * FUNCTION:     BACKGROUND EXECUTIVE.                             *
                    9  *              IT RUNS IN THE BACKGROUND AND EXECUTS               *
                   10  *              THE JOB THAT'S BEEN QUEUED IN THE JOB BUFFER        *
                   11  *              (JBBUF).                                            *
                   12  *              FOR OTHER THAN THE ALLOWABLE JOB NUMBER             *
                   13  *              IN THE JOB BUFFER, IT WILL TRANSFER CONTROLS TO     *
                   14  *              THE POWER ON ROUTINE.                               *
                   15  *              A JOB GETS QUEUED BY THE JBQUE.                     *
                   16  *                                                                  *
                   17  * CALLING MODULES: POWER ON ISR                                    *
                   18  *                                                                  *
                   19  * CALLING SEQ:  BLWP @EXTIVE                                      *
                   20  *                                                                  *
                   21  * INPUT: R8=JOB BUFFER INPUT POINTER                               *
                   22  *                                                                  *
                   23  * OUTPUT: CALLS BACKGROUND JOB                                    *
                   24  *         INCREMENTS JOB BUFFER OUTPUT POINTER BY 4                *
                   25  *                                                                  *
                   26  * MODULES REFERENCED:                                             *
                   27  *                       PON=POWER ON ISR                          *
                   28  *                       JOBS LISTED IN JBTBL TABLE                *
                   29  *                                                                  *
                   30  * WORKSPACE AREA: WPEX                                            *
                   31  *                                                                  *
                   32  * REGISTERS MODIFIED: R0,R1,R9                                    *
                   33  *                                                                  *
                   34  * VERSION HISTORY                                                 *
                   35  *                                                                  *
                   36  ********************************************************************
=0000              37         RSECT   EXTIVE
                   38  *** CALL NAME
                   39         INTERN  EXTIVE
                   40  *** VARIABLES REFERENCED
                   41         EXTERN  JBBBUF,JBEBUF
                   42         EXTERN  WPEX
                   43  *** MODULES REFERENCED
                   44         EXTERN  INOVCT
                   45  *** TABLES REFERENCED
                   46         EXTERN  JBTBL,JBMAX
                   47  *** LIBRARY
                   48  *** REGISTER DEFINITION
=0006              49  MSERR   EQU     R6              MEMORY SUM CHECK ERROR
=0007              50  MSIX    EQU     R7                "     "    "   INDEX
=0008              51  JBIPTR  EQU     R8              JOBBUF INPUT POINTER
=0009              52  JBOPTR  EQU     R9              JOBBUF OUTPUT POINTER
```

```
                     53  *** TABLES REFERENCED
                     54  *** MODULES REFERENCED
                     55  *** LIBRARY
                     56          INCLUDE CNSTNT
                    184  *** REGISTERS DEFINITION
        =0006       185  EATMR   EQU     R6      =       EAROM TIMER
        =0007       186  EAWDP   EQU     R7      =         *   WRITE DATA WORD POINTER
        =0008       187  EAADR   EQU     R8      =         *   ADDRESS
        =0009       188  EACNT   EQU     R9      =         *   CONTROL REGISTER IMMAGE
        =000A       189  EAMODE  EQU     R10     =         *   MODE
                    190  ********************************************************************************
                    191  *
    0000'           192  EAWRIT
    0000'  C28A     193          MOV     EAMODE,EAMODE           EAROM MODE
    0002'  1616     194          JNE     100$                    EXIT IF EAROM BUSY
                    195  ***
    0004'  C210     196          MOV     *R0,EAADR               GET EAROM ADDRESS
    0006'  C808 FFAE 197         MOV     EAADR,@CEAADR           MOVE TO H/W REGISTER
    000A'  C1C1     198          MOV     R1,EAWDP                GET EAROM DATA POINTER
    000C'  C811 FFA0 199         MOV     *R1,@CEAWD              DATA TO H/W REGISTER
    0010'  C260 0003* 200        MOV     @B2,EACNT               ASSUME LOWER 32 WORDS (0 TO 31)
    0014'  0288 0020 201         CI      EAADR,32
    0018'  1102     202          JLT     10$                     JIF LOWER 32 WORDS
    001A'  C260 0004* 203        MOV     @B3,EACNT               SET TO UPPER 32 WORDS (32-63)
    001E'  E260 0002* 204  10$   SOC     @B1,EACNT               SET "C2" TO ON
    0022'  C809 FFAC 205         MOV     EACNT,@CEACNT           CONTROL BITS TO H/W REGISTER
    0026'  070A     206          SETO    EAMODE                  EAROM MODE = ERASE
    0028'  C1A0 0001* 207        MOV     @TIMEL,EATMR            LSW OF TIMER
    002C'  A1A0 0005* 208        A       @EATC,EATMR             + 250 MSEC = EA TIME
                    209  *
    0030'  045B     210  100$    RT                              RETURN
                    211  ********************************************************************************
                    212  *
                    213          END
```

No errors detected

```
                    1          IDT    EAWRIT
                    2          SUBTTL EAROM WRITE
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: EAWRIT.SRC                      AUTH: N.COSTANTINIDES      *
                    6  * VERSION: 1                            DATE: 29-JUN-1982          *
                    7  *                                                                  *
                    8  * FUNCTION: THIS SUBROUTINE WILL BE CALLED WHEN IT IS DESIRED      *
                    9  *           TO WRITE A WORD IN THE EAROM MEMORY.                   *
                   10  *           IT TAKES APPROX 500 MSEC TO WRITE ONE WORD INTO        *
                   11  *           EAROM. EAROM CONTAINS 64 WORDS OF MEMORY.              *
                   12  *           THE FIRST 250 MSEC OF THIS TIME PERIOD IS DESIGNED     *
                   13  *           FOR THE ERASE CYCLE WHICH THE SELECTED LOCATION IS     *
                   14  *           BEING ERASED.                                          *
                   15  *           IT IS FOLLOWED BY ANOTHER 250 MSEC WRITE CYCLE         *
                   16  *           DURING WHICH THE DATA IS BEING WRITTEN IN THE EAROM    *
                   17  *           LOCATION.                                              *
                   18  *           THIS SUBROUTINE INITIATES THE ERASE CYCLE BY SETTING   *
                   19  *           EAROM ADDRESS,DATA & CONTROL REGISTERS WITH THE        *
                   20  *           GIVEN DATA  AND ALSO SET THE 250 MSEC TIMER.           *
                   21  *                                                                  *
                   22  * CALLING MODULES: EAMON                                          *
                   23  *                                                                  *
                   24  * CALLING SEQ: BL @EAWRIT                                         *
                   25  *                                                                  *
                   26  * INPUTS: R0    =       EAROM ADDRESS WORD POINTER                *
                   27  *         R1    =       EAROM DATA WORD POINTER                   *
                   28  *                                                                  *
                   29  * OUTPUTS: R6   =       TIME + 250 MSEC                           *
                   30  *          R7   =       EAROM WRITE DATA WORD POINTER             *
                   31  *          R8   =       EAROM ADDRESS + 1                         *
                   32  *          R9   =       B1 ON AND B2 OR B3 ON                     *
                   33  *          R10  =       -1 (ERASE MODE)                           *
                   34  *          EAROM ADDR WORD = EAROM ADDR WORD + 1                  *
                   35  *                                                                  *
                   36  * MODULES REFERENCED: NONE                                        *
                   37  *                                                                  *
                   38  * WORKSPACE AREA: WPJB1                                           *
                   39  *                                                                  *
                   40  * REGISTERS MODIFIED:R0,E6,R7,R8,R9,R10                           *
                   41  *                                                                  *
                   42  * VERSION HISTORY:                                                *
                   43  *                                                                  *
                   44  ********************************************************************
=0000              45          RSECT  EAWRIT
                   46  *** CALL NAME
                   47          INTERN EAWRIT
                   48  *** VARIABLES REFERENCED
                   49          EXTERN TIMEL
                   50  *** CONSTANTS REFERENCED
                   51          EXTERN B1,B2,B3
                   52          EXTERN EATC
```

```
0012'  0200 0019     180          LI      R0,>0019              UPPER 32 WORDS (32 - 63)
0016'  C800 FFAC     181  12$     MOV     R0,@CEACNT      SET CHIP SELECT, READ TO ON, AND CLOCK TO ON
                     182  *** WAIT 20 MICRO SEC
001A'  0AF3          183          SLA     R3,15           DELAY
001C'  0AF3          184          SLA     R3,15
001E'  0AF3          185          SLA     R3,15
                     186  *** SET CONTROL REGISTER
0020'  0240 000F     187          ANDI    R0,>000F        CLOCK TO OFF
0024'  C800 FFAC     188          MOV     R0,@CEACNT
                     189  *** WAIT 4 MICRO SEC
0028'  0581          190          INC     R1              BUMP EAROM ADDRESS
                     191  *** READ EAROM DATA AND STORE IN IMAGE OF EAROM BUFFER
002A'  C8A0 FFA2     192          MOV     @CEARD,@EAIMBF(R2)
002E'  0001*
                     193  *** SET CONTROL REGISTER
0030'  04E0 FFAC     194          CLR     @CEACNT         CHIP SELECT TO OFF
                     195  ***
0034'  05C2          196          INCT    R2              BUMP BUFFER INDEX
0036'  0281 0040     197          CI      R1,64
003A'  11E4          198          JLT     10$             LOOP 64 TIMES
003C'  045B          199          RT
                     200  ********************************************************************************
                     201          END
```

No errors detected

```
                        1          IDT     EAREAD
                        2          SUBTTL  EAROM READ
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: EAREAD(TWA,MAN)                AUTH: N.COSTANTINIDES     *
                        6  * VERSION: 2                           DATE:   16-NOV-1982       *
                        7  *                                                                  *
                        8  * FUNCTION:     THIS ROUTIME IS CALLED WHEN POWER ON INTERRUPT    *
                        9  *              OCCURS. IT WILL COPY THE CONTENTS OF 64 WORDS      *
                       10  *              OF EAROM MEMORY INTO IMAGE OF EAROM BUFFER (RAM)  *
                       11  *              WHENEVER A WORD IS WRITTEN INTO EAROM LOC THE      *
                       12  *              SAME WORD IS ALSO WRITTEN INTO THIS RAM BUFFER     *
                       13  *                                                                  *
                       14  * CALLING MODULES: PON                                           *
                       15  *                                                                  *
                       16  * CALLING SEQ: BL @EAREAD                                        *
                       17  *                                                                  *
                       18  * INPUTS: NONE                                                   *
                       19  *                                                                  *
                       20  * OUTPUTS: IMAGE OF EAROM IN IMAGE OF EAROM IMAGE BUFF (EAIMBF)  *
                       21  *                                                                  *
                       22  * MODULES REFERENCED:                                            *
                       23  *                                                                  *
                       24  * WORKSPACE AREA:                                                *
                       25  *                                                                  *
                       26  * REGISTERS MODIFIED: R0,R1,R2,R3                                *
                       27  *                                                                  *
                       28  * VERSION HISTORY:                                               *
                       29  *       \                                                          *
                       30  *                                                                  *
                       31  ********************************************************************
          =0000        32          RSECT   EAREAD
                       33  *** CALL NAME
                       34          INTERN  EAREAD
                       35  *** VARIABLES REFERENCED
                       36          EXTERN  EAIMBF
                       37  *** CONSTANTS REFERENCED
                       38  *** TABLES REFERENCED
                       39  *** MODULES REFERENCED
                       40  *** LIBRARY
                       41          INCLUDE CNSTNT
                      169  *** REGISTERS DEFINITION
                      170  ********************************************************************
0000'                 171  EAREAD
0000' 04C1            172          CLR     R1              EAROM ADDRESS
0002' 04C2            173          CLR     R2              INDEX OF EAROM IMAGE BUFFER
                      174  *** WRITE EAROM ADDRESS
0004' C801 FFAE       175  10$     MOV     R1,@CEAADR      WRITE ADDR TO H/W ADDR
                      176  *** SET CONTROL REGISTER
0008' 0200 0015       177          LI      R0,>0015        ASSUME LOWER 32 WORDS (0 - 31)
000C' 0281 0020       178          CI      R1,32
0010' 1102            179          JLT     12$             JIF NOT LOWER 32 WORDS
```

```
0050' 1008          230          JMP     30$                 CONTINUE
0052' 0A18          231  25$     SLA     EAADR,1             EAROM ADDRESS * 2
0054' CA00 0004*    232          MOV     R0,@EAIMBF(EAADR)   STORE IN EAROM IMAGE ONLY
0058' 05A0 0006*    233          INC     @EASADR             BUMP SS EAROM ADDRESS WORD
005C' 04E1 0005*    234          CLR     @EATSBF(R1)         CLEAR DATA
0060' 1025          235          JMP     100$                EXIT
0062' 0200 0006*    236  30$     LI      R0,EASADR           SYSTEM STATUS EAROM ADDR
0066' 0221 0005*    237          AI      R1,EATSBF           TEMPORARY EAROM SYSTEM STATUS BUFFER
006A' 06A0 000B*    238          BL      @EAWRIT             INITIATE EAROM WRITE COMMAND
006E' 101E          239          JMP     100$                EXIT
                    240  *** EAROM WRITING
0070' C060 0007*    241  50$     MOV     @TIMEL,R1           LSW OF TIMER
0074' 6046          242          S       EATMR,R1            - EAROM TIMER + 250 MSEC = ?
0076' 111A          243          JLT     100$                EXIT IF 250 MSEC NOT UP YET
0078' C28A          244          MOV     EAMODE,EAMODE       EAROM MODE
007A' 150B          245          JGT     55$                 JIF WRITE MODE
                    246  *** ERASE MODE
007C' 4260 0009*    247          SZC     @B1,EACNT           C2 CONTROL BIT OFF IN IMAGE AND
0080' C809 FFAC     248          MOV     EACNT,@CEACNT       IN HW REG, TO
0084' C1A0 0007*    249          MOV     @TIMEL,EATMR        LSW OF TIMER
0088' A1A0 0003*    250          A       @EATC,EATMR         + 250 MSEC = EA TIMER
008C' 020A 0001     251          LI      EAMODE,1            SET TO WRITE MODE
0090' 1003          252          JMP     100$                EXIT
                    253  *** WRITE MODE
0092' 04C9          254  55$     CLR     EACNT               CHIP SELECT CONTROL OFF IN IMAGE AND
0094' 04E0 FFAC     255          CLR     @CEACNT             IN H/W REG.
                    256  *  STORE
0098' 0A18          257          SLA     EAADR,1             *2
009A' CA17 0004*    258          MOV     *EAWDP,@EAIMBF(EAADR)  STORE DATA IN IMAGE OF EAROM BUFFER
009E' C020 0001*    259          MOV     @CLRFLG,R0          CLEAR FLAG
00A2' 1601          260          JNE     57$
00A4' 04D7          261          CLR     *EAWDP              CLEAR DATA
00A6' 04CA          262  57$     CLR     EAMODE              EAROM MODE = INACTIVE
00A8' 05A0 0006*    263          INC     @EASADR             BUMP S.S EAROM ADDR WORD
                    264  ***
00AC'               265  100$
00AC' 0380          266          RTWP
                    267  ********************************************************************
                    268          END
```

No errors detected

```
000C' 06A0 0002*    75          BL      @ICRC                   CALL RECEIVER
                    76  *
0010' 020C 0780     77          LI      CRU,C9902I
0014' 1F 11         78          TB      17                      CHECK TRANSMITTER INTERRUP
0016' 160A          79          JNE     ICIEXT                  EXIT IF NOT
                    80  *
                    81  *** TRANSMITTER INTERRUPT
                    82  *
0018' 06A0 0003*    83          BL      @ICTX                   CALL TRANSMITTER
001C' 1007          84          JMP     ICIEXT                  EXIT
                    85  *
                    86  *
                    87  *
001E' 1F 11         88  ICISR2  TB      17                      CHECK TRANSMITTER INTERRUPT
0020' 1603          89          JNE     ICISR4                  JIF NOT
                    90  *
                    91  *** NOT RECEIVER BUT TRANSMITTER INTERRUPT
                    92  *
0022' 06A0 0003*    93          BL      @ICTX                   CALL TRANSMITTER
0026' 1002          94          JMP     ICIEXT                  EXIT
                    95  *
                    96  *** NOT RECEIVER OR TRANSMITTER INTERRUPT
                    97  *
0028' 05A0 0001*    98  ICISR4  INC     @ICIER                  BUMP IC-ISR-ERROR-CTR
                    99  *
002C' 0380         100  ICIEXT  RTWP                            RETURN
                   101  *
                   102          END

No errors detected
```

```
                     1          IDT    ICSTX
                     2          SUBTTL INTER CPU START MSG TRANSMISSION
                     3  ********************************************************************************
                     4  *                                                                              *
                     5  * NAME: ICSTX                          AUTH: M.COSTANTINIDES   *
                     6  * VERSION: 1                           DATE: 03-JAN-84                 *
                     7  *                                                                              *
                     8  * FUNCTION:     INTER CPU START MESSAGE TRANSMISSION.                         *
                     9  *               IF MESSAGE IS QUEUED IT SETS UP TX REGISTERS AND ENABLES*
                    10  *               TMS 9902 INTERFACE TO INITIATE MESSAGE TRANSMISSION (TO *
                    11  *               INITIATE THE INTER-CPU TRANSMIT INTERRUPT). ALL THE     *
                    12  *               CHARACTERS ARE TRANSMITTED BY THE INTER-CPU TRANSMITTER *
                    13  *               INTERRUPT (ICTX).                                        *
                    14  *                                                                              *
                    15  *                                                                              *
                    16  * CALLING MODULE: RTISR = REAL TIME INTERRUPT                                  *
                    17  *                                                                              *
                    18  * CALLING SEQ: BL @ICSTX                                                      *
                    19  *                                                                              *
                    20  * INPUTS: MESSAGE IS QUEUED FOR TRANSMISSION                                  *
                    21  *                                                                              *
                    22  * OUTPUTS: IF MESSAGE IS QUEUED, ENABLES TRANSMITTER IN THE TMS9902           *
                    23  *                                                                              *
                    24  * MODULE REFERENCED: NONE                                                     *
                    25  *                                                                              *
                    26  * WORKSPACE AREA: WPRT                                                        *
                    27  *                                                                              *
                    28  * REGISTERS MODIFIED: R0,R1,R12                                               *
                    29  *                                                                              *
                    30  * VERSION HISTORY:                                                            *
                    31  *                                                                              *
                    32  ********************************************************************************
        =0000       34         RSECT  ICSTX
                    35  *** CALL NAME
                    36         INTERN ICSTX
                    37  *** VARIABLES REFERENCED
                    38         EXTERN ICBQUE,ICEQUE
                    39         EXTERN ICIQP,ICOQP
                    40         EXTERN ICTCTR,ICTBSA
                    41         EXTERN ICTACT,ICIFLG
                    42  *** LIBRARY
                    43         INCLUDE ICEQUATE.LIB
                    65  *** REGISTERS DEFINITION
        =000C       66  CRU    EQU    R12
                    67  ********************************************************************************
        =0000'      68  ICSTX  EQU    $
0000'  C060 0004*   69         MOV    @ICOQP,R1          INTER-CPU OUTPUT QUEUE POINTER
0004'  8801 0003*   70         C      R1,@ICIQP          OUTPUT PTR .VS. INPUT PTR
0008'  1316         71         JEQ    ICSEXT             EXIT IF MSG NOT QUEUED
                    72  * MESSAGE IS QUEUED
000A'  C020 0005*   73         MOV    @ICTCTR,R0         TX-MSG-BYTE-COUNTER
000E'  1613         74         JNE    ICSEXT             EXIT IF TRANSMITTING MSG
```

```
                        75  **  TRANSMITTER IS AVAILABLE
                        76  **      GET MSG BUFFER POINTER FROM QUEUE
0010'  05C1             77          INCT    R1                      BUMP OUTPUT POINTER
0012'  0281 0002*       78          CI      R1,ICEQUE
0016'  1202             79          JLE     20$                     JIF NOT END-OF-QUEUE
                        80  **
0018'  0201 0001*       81          LI      R1,ICBQUE               ELSE, SET TO BEG-OF-QUEUE
                        82  **
001C'  C801 0004*       83  20$     MOV     R1,@ICOQP               INTER-CPU OUTPUT QUEUE POINTER
0020'  C051             84          MOV     *R1,R1                  GET MSG BUFFER ADDR
0022'  C801 0006*       85          MOV     R1,@ICTBSA              INTER-CPU MSG BUFFER POINTER
0026'  04E0 0008*       86          CLR     @ICIFLG                 CLEAR IC INTERRUPT FLAG
                        87  **
002A'  0720 0007*       88          SETO    @ICTACT                 SET INTER-CPU TX TO ACTIVE
002E'  020C 0780        89          LI      CRU,C9902I              INTER-CPU 9902 CRU BASE ADDR
0032'  1D 13            90          SBO     19                      ENABLE TX INTERRUPT
0034'  1D 10            91          SBO     16                      TURN ON TRANSMITTER (RTSON)
                        92  **
0036'  045B             93  ICSEXT  RT                              RETURN
                        94  *
                        95          END

    No errors detected
```

```
                1          IDT     ICQMSG
                2          SUBTTL  INTER-CPU QUEUE MESSAGE
                3  ********************************************************************************
                4  *                                                                              *
                5  * NAME: ICQMSG                          AUTH: N.COSTANTINIDES   *
                6  * VERSION: 1                            DATE: 22-JUL-82                 *
                7  *                                                                              *
                8  * FUNCTION:     INTER-CPU QUEUE MESSAGE.                                       *
                9  *               THIS SUBROUTINE SHALL BE CALLED WHEN IT IS DESIRED TO   *
               10  *               TRANSMIT A MESSAGE TO THE INTER CPU.                           *
               11  *               THE ICQMSG PLACES THE MESSAGE IN THE OUTPUT MESSAGE     *
               12  *               QUEUE.  IF THE OUTPUT MESSAGE QUEUE IS FULL WHEN THE   *
               13  *               CALL IS MADE THE CALL IS REJECTED AND THE NEW MESSAGE   *
               14  *               IS NOT QUEUED.  THE PROCESSOR EQ (EQUAL) BIT IS USED   *
               15  *               AS AN INDICATOR FOR THIS CONDITION.                            *
               16  *               IF ON RETURN FROM ICQMSG THE EQ BIT IS SET, THE        *
               17  *               OUTPUT MESSAGE QUEUE WAS FULL AND THE MESSAGE WAS      *
               18  *               NOT PLACED IN THE QUEUE.  THE CALLING ROUTINE SHOULD   *
               19  *               TEST FOR THIS CONDITION AND TAKE APPROPRIATE ACTION.   *
               20  *               THE DATA BUFFER WILL USED AS A MESSAGE BUFFER.         *
               21  *               THEREFORE ALLOCATE 2 ADDITIONAL WORDS ABOVE THE DATA   *
               22  *               BUFFER FOR MESSAGE TYPE & NUMBER OF DATA WORDS; AND    *
               23  *               ONE ADDITIONAL WORD BELOW THE DATA BUFFER FOR THE       *
               24  *               CHECKSUM.                                                      *
               25  *                                                                              *
               26  *                                                                              *
               27  * CALLING MODULES:RTIFR                                                        *
               28  *                                                                              *
               29  * CALLING SEQ:  LI R1,ARGUMENT-BLOCK                                           *
               30  *               BL @ICQMSG                                                     *
               31  *               JEQ QUEUE-FULL                                                 *
               32  *                                                                              *
               33  * INPUTS: R1 = ADDRESS OF ARGUMENT BLOCK                                       *
               34  *              ARGUMENT-BLOCK   DATA A  ;DATA BUFFER START ADDR          *
               35  *                               DATA B  ;MSG TYPE                        *
               36  *                               DATA C  ;NO. OF DATA WORDS               *
               37  *         TMA BIT7 TO BIT4 = TX MSG ACKNOWLEDGE FLAG BITS                      *
               38  *                                                                              *
               39  * OUTPUTS: EQ BIT = 1   QUEUE WAS FULL.                                        *
               40  *          EQ BIT = 0   MSG PLACED IN QUEUE (ICBQUE).                          *
               41  *                       ICIQP INCREMENTED BY 2.                                *
               42  *                                                                              *
               43  * MODULES REFERENCED: NONE                                                     *
               44  *                                                                              *
               45  * WORKSPACE AREA: CALLER'S                                                     *
               46  *                                                                              *
               47  * REGISTERS MODIFIED: R0,R1,R2,R3                                              *
               48  *                                                                              *
               49  * VERSION HISTORY:                                                             *
               50  *                                                                              *
               51  ********************************************************************************
=0000          53          RSECT   ICQMSG
```

```
                        54  *** CALL NAME
                        55          INTERN  ICQMSG
                        56  *** VARIABLES REFERENCED
                        57          EXTERN  ICIQP,ICOQP,ICBQUE,ICEQUE
                        58          EXTERN  TMA
                        59  *** CONSTANTS REFERENCED
                        60          EXTERN  BIT0
                        61  *** LIBRARY
                        62          INCLUDE CNSTNT
                       190          INCLUDE ICEQUATE.LIB
                       212  *** REGISTER DEFINITION
        =0001          213  ARG1    EQU     R1                  ARGUMENT POINTER (SCRATCH)
        =0002          214  PTR2    EQU     R2                  MSG BUFFER POINTER (SCRATCH)
                       215  ********************************************************************************
        =0000'         216  ICQMSG  EQU     $
  0000' C0B1           217          MOV     *ARG1+,PTR2         DATA BUFFER START ADDR
  0002' 0642           218          DECT    PTR2
  0004' 0642           219          DECT    PTR2                TO GET TO MSG BUFFER
  0006' C002           220          MOV     PTR2,R0             SAVE FOR QUEUE
  0008' C4B1           221          MOV     *ARG1+,*PTR2        MOVE MSG TYPE IN 1ST WORD
  000A' E8A0 0005*     222          SOC     @TMA,*PTR2          SET MSG ACKNOWLEDE FLAG BIT
  000E' 04E0 0005*     223          CLR     @TMA                 AND CLEAR IT
  0012' E8A0 0006*     224          SOC     @BIT0,*PTR2+        SET TRANSMISSION NOT COMPLETE FLAG
  0016' C491           225          MOV     *ARG1,*PTR2         MOVE NO. OF DATA WORDS IN 2ND WORD
  0018' 0642           226          DECT    PTR2                MSG BUFFER START ADDR
                       227  * COMPUTE CHEKSUM WORD
  001A' 0201 00FF      228          LI      ARG1,MBWSIZ         MSG BUFFER SIZE (WORDS)
  001E' 04C3           229          CLR     R3                  RESULT
  0020' A0F2           230  10$     A       *PTR2+,R3           ADD START TO END OF BUFFER
  0022' 0601           231          DEC     ARG1                COUNT DOWN
  0024' 15FD           232          JGT     10$                 LOOP
                       233  * STORE CHECKSUM WORD
  0026' 0503           234          NEG     R3                  2'S COMPLEMENT CHECKSUM WORD
  0028' C483           235          MOV     R3,*PTR2            MOV IT INMSG BUFFER
                       236  *
                       237  * PLACE MESSAGE IN QUEUE
  002A' 0300 0000      238          LIMI    0                   DISABLE ALL INTERRUPTS
  002E' C0E0 0001*     239          MOV     @ICIQP,R3           GET INPUT QUEUE POINTER
  0032' 05C3           240          INCT    R3                  BUMP IT
  0034' 0283 0004*     241          CI      R3,ICEQUE
  0038' 1202           242          JLE     20$                 JIF NOT END-OF-QUEUE
                       243  **
  003A' 0203 0003*     244          LI      R3,ICBQUE           ELSE, SET TO BEG-OF-QUEUE
                       245  ** CHECK IF QUEUE FULL
  003E' 8803 0002*     246  20$     C       R3,@ICOQP           IN VERSUS OUT
  0042' 1303           247          JEQ     ICQEXT              EXIT IF QUEUE FULL (EQ=1)
                       248  ** PLACE MSG IN QUEUE
  0044' C4C0           249          MOV     R0,*R3              PLACE MSG BUFFER IN QUEUE
  0046' C803 0001*     250          MOV     R3,@ICIQP           RESTORE INPUT QUEUE POINTER (EQ=0)
                       251  *
  004A' 045B           252  ICQEXT  RT                          RETURN WITH EQ STATUS
                       253  ***
```

254   END

No errors detected

```
                    1          IDT     ICMD
                    2          SUBTTL  INTER CPU MSG DISPATCHER JOB
                    3  ********************************************************************************
                    4  *                                                                              *
                    5  * NAME: ICMD                              AUTH: N.COSTANTINIDES   *
                    6  * VERSION: 1                              DATE: 13-JUL-82                      *
                    7  *                                                                              *
                    8  * FUNCTION:     INTER-CPU MESSAGE DISPATCHER JOB,                              *
                    9  *               THE ICMD IS CALLED BY THE EXTIVE WHEN THIS JOB IS              *
                   10  *               PLACED IN THE JOB QUEUE BY THE INTER-CPU RECEIVE               *
                   11  *               INTERRUPT.  THE ICMD CHECKS THE MESSAGE TYPE, IF               *
                   12  *               IT IS LEGAL, TRANSFERS CONTROL TO THE MESSAGE                  *
                   13  *               SUBROUTINE LISTED IN THE RECEIVE MESSAGE TYPE                  *
                   14  *               SUBROUTINE ADDRESS TABLE (ICMDTB).                             *
                   15  *                                                                              *
                   16  *                                                                              *
                   17  * CALLING MODULES: EXTIVE                                                     *
                   18  *                                                                              *
                   19  * CALLING SEQ: BLWP @R1                                                       *
                   20  *                                                                              *
                   21  * INPUTS: R0 = RECEIVE MESSAGE BUFFER START ADDRESS                           *
                       *                                                                              *
                       * OUTPUTS: CALLS MESSAGE TYPE SUBROUTINE (RMSG0 TO RMSG#)                     *
                   24  *                                                                              *
                   25  * MODULE REFERENCED: RMSG0 TO RMSG#                                           *
                   26  *                                                                              *
                   27  * WORKSPACE AREA: WPJB                                                        *
                   28  *                                                                              *
                   29  * REGISTERS MODIFIED: R0,R1,R2                                                *
                   30  *                                                                              *
                   31  * VERSION HISTORY:                                                            *
                   32  *                                                                              *
                   33  ********************************************************************************
=0000              34          RSECT   ICMD
                   35  *** CALL NAME
                   36          INTERN  ICMD
                   37  *** TABLES REFERENCED
                   38          EXTERN  ICMDTB,ICMMAX
                   39  *** MODULES REFERENCED
```

```
                    41  ********************************************************************************
      =0000'        42  ICMD    EQU     $
0000' C05D          43          MOV     *R13,R1                 GET MSG BUF PTR FROM CALLER'S R0
0002' C091          44          MOV     *R1,R2                  GET 1ST WORD OF MSG
0004' 0242 00FF     45          ANDI    R2,>FF
0008' 8802 0002*    46          C       R2,@ICMMAX              ACTUAL MSG TYPE .VS. MAX TYPE
000C' 1203          47          JLE     10$                     JIF LEGAL MSG TYPE
                    48  *** ILLEGAL MESSAGE TYPE, JUST IN CASE IT IS ALREADY CAPTURED IN ICRC (ISR)
000E' 04D1          49          CLR     *R1             MAKE IC RC MSG BUFFER AVAILABLE
0010' 0580          50          INC     R0                      KEEP TRACK OF ERROR
0012' 1004          51          JMP     ICMEXT                  EXIT
                    52  *** LEGAL MESSAGE TYPE, CALL MSG TYPE SUBROUTINE.
0014' 0A12          53  10$     SLA     R2,1                    MSG TYPE * 2
0016' C022 0001*    54          MOV     @ICMDTB(R2),R0          GET MSG TYPE SUBR
001A' 0690          55          BL      *R0                     CALL MSG TYPE SUBR WITH R1=BUF PTR
001C' 0380          56  ICMEXT  RTWP                            RETURN TO EXECUTIVE
                    57  ********************************************************************************
                    58          END
```

No errors detected

```
                     1          IDT     ICBSEL
                     2          SUBTTL  INTER-CPU RECEIVE MSG BUFFER SELECTION
                     3  ********************************************************************************
                     4  *                                                                              *
                     5  * NAME: ICBSEL                  AUTH: N.COSTANTINIDES   *
                     6  * VERSION: 1                    DATE: 17 DEC 81                             *
                     7  *                                                                              *
                     8  * FUNCTION:     IT SELECTS THE AVAILABLE RECEIVE MESSAGE BUFFER FOR     *
                     9  *               THE NEXT MESSAGE. IF BUFFER IS NOT AVAILABLE IT         *
                    10  *               INCREMENTS THE 'BUFFER NOT AVAIALBLE' ERROR AND         *
                    11  *               SELECTS THE LAST BUFFER LISTED IN THE INTER-CPU RECEIVER*
                    12  *               ADDRESS TABLE (ICRBTB).                                 *
                    13  *               THE MSB OF FIRST WORD IN THE BUFFER INDICATES WHETHER   *
                    14  *               THE BUFFER IS AVAILABLE (LOGIC 0) OR NOT AVAILABLE      *
                    15  *               (LOGIC 1).                                              *
                    16  *                                                                       *
                    17  *                                                                       *
                    18  * CALLING MODULES:      ICRC    = INTER-CPU RECEIVE INTERRUPT          *
                    19  *                       ICRERR  = INTER-CPU RECEIVE ERROR              *
                    20  *                                                                       *
                    21  * CALLING SEQ: BL @ICBSEL                                              *
                    22  *                                                                       *
                    23  * INPUTS: NONE                                                         *
                    24  *                                                                       *
                    25  * OUTPUTS: ICRBSA = RECEIVE MSG BUFFER START ADDRESS                    *
                    26  *                                                                       *
                    27  * MODULE REFERENCED: NONE                                              *
                    28  *                                                                       *
                    29  * WORKSPACE AREA: WPIC                                                 *
                    30  *                                                                       *
                    31  * REGISTERS MODIFIED: R0,R1,R6                                         *
                    32  *                                                                       *
                    33  * VERSION HISTORY:                                                     *
                    34  *                                                                       *
                    35  ********************************************************************************
       =0000        36          RSECT   ICBSEL
                    37  *** CALL NAME
                    38          INTERN  ICBSEL
                    39  *** VARIABLES REFERENCED
                    40          EXTERN  ICRER9,ICRBSA
                    41  *** CONSTANTS REFERENCED
                    42          EXTERN  BIT0
                    43          EXTERN  D0
                    44  *** TABLES REFERENCED
                    45          EXTERN  ICRBTB
                    46  *** REGISTER DEFINITION
                    47  ********************************************************************************
       =0000'       48  ICBSEL  EQU     $
0000'  0201 0005*   49          LI      R1,ICRBTB               INTER-CPU RECEIVE BUFFER ADDRESS TABLE
0004'  C031         50          MOV     *R1+,R0                 NUMBER OF BUFFER ADDRESSES
                     51  * FIND AVAILABLE RECEIVE MSG BUFFER
0006'  C831 0002*   52  10$     MOV     *R1+,@ICRBSA            GET BUFFER START ADDRESS
```

```
000A' C1A0 0002%    53          MOV     @ICRBSA,R6
000E' 8816 0004%    54          C       *R6,@D0             MSB OF 1ST WORD = 1 FOR BUSY
0012' 1505          55          JGT     ICB10               JIF BUFFER AVILABLE (+ VALUE)
0014' 1304          56          JEQ     ICB10               JIF BUFFER AVAILABLE (+0)
0016' 0600          57          DEC     R0
0018' 15F6          58          JGT     10$                 LOOP
                    59  * BUFFER NOT AVAIALBLE.
001A' 05A0 0001%    60          INC     @ICRER9             MSG BUFFER NOT AVAILABLE ERROR
                    61  * BUFFER AVAILABLE
001E' C1A0 0002%    62  ICB10   MOV     @ICRBSA,R6
0022' E5A0 0003%    63          SOC     @BIT0,*R6           SET TO BUSY
                    64  * EXIT
0026' 045B          65          RT                          RETURN
                    66  *
                    67          END
```

No errors detected

```
                     1          IDT     IDENT
                     2          SUBTTL  IDENTIFY AIRCRAFT TYPE
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: IDENT.SRC                        AUTH: N. CONSTANTINIDES *
                     6  * VERSION: 1                             DATE: 3-MAY-1983         *
                     7  *                                                                  *
                     8  * FUNCTION:     READS THE IDENT DISCRETES FROM THE DISCRETE       *
                     9  *               CHANNELS 82 AND 83. FOR P&W VERSION (CH 82)       *
                    10  *               IT SETS A WORD, SSEG TO 0. FOR GE VERSION         *
                    11  *               (CH 83) SETS SSEG TO 2.                           *
                    12  *                                                                  *
                    13  * CALLING MODULES: PON                                            *
                    14  *                                                                  *
                    15  * CALLING SEQ: BL @IDENT                                          *
                    16  *                                                                  *
                    17  * INPUTS:       NONE                                              *
                    18  *                                                                  *
                    19  * OUTPUTS:      SSEG WORD                                         *
                    20  *                                                                  *
                    21  * MODULES REFERENCED: DCDRQ                                       *
                    22  *                                                                  *
                    23  * WORKSPACE AREA: CALLER'S                                        *
                    24  *                                                                  *
                    25  * REGISTERS MODIFIED: R0,R1,R2,R3,R4,R5                           *
                    26  *                                                                  *
                    27  * VERSION HISTORY:
                    28  *                                                                  *
                    29  ********************************************************************
       =0000        30          RSECT   IDENT
                    31  *** CALL NAME
                    32          INTERN  IDENT
                    33  *** VARIABLES REFERENCED
                    34          EXTERN  SSEG
                    35  *** CONSTANTS REFERENCED
                    36          EXTERN  D2
                    37  *** TABLES REFERENCED
                    38  *** MODULES REFERENCED
                    39          EXTERN  DCDRQ
                    40  *** LIBRARY
                    41          INCLUDE ENCLOS
                    43  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    45          INCLUDE REGDEF          REGISTER DEFENITIONS
                    64          INCLUDE CNSTNT          CONSTANTS
                   192          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   493          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   723          INCLUDE JMPMAC          JUMP MACROS
                   757          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   772          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                  1148  *** REGISTERS DEFINITION
                  1149  ********************************************************************
    0000'         1150  IDENT
```

```
                    1151
0000' C80B 0000'    1152          MOV     R11,@LINKZ          SAVE LINK
0004' 04E0 0001*    1153          CLR     @SSEG               CLEAR WORD
0008' 04C2          1154          CLR     R2                  CLEAR TABLE POINTER
000A' 04C3          1155          CLR     R3                  CLEAR DATA REGISTER
                    1156  *
000C' C162 003C'    1157          MOV     @TAB(R2),R5         CHANNEL ADDRESS
0010' 0201 0726     1158          LI      R1,>0726
0014' 06A0 0003*    1159          BL      @DCDR@              READ CHANNEL
0018' E0C4          1160          SOC     R4,R3               SET BIT IN R3
001A' 05C2          1161          INCT    R2                  INCREMENT POINTER
                    1162  *
001C' C162 003C'    1163          MOV     @TAB(R2),R5         ADDRESS OF CHANNEL 83
0020' 0201 0726     1164          LI      R1,>0726
0024' 06A0 0003*    1165          BL      @DCDR@              READ CHANNEL
0028' 0A14          1166          SLA     R4,1                MOVE BIT TO 2ND POS
002A' E0C4          1167          SOC     R4,R3               SET BIT IN R3
                    1168  *
002C' 0283 0002     1169          CI      R3,2                CHECK RESULT
0030' 1102          1170          JLT     10$                 JIF <2 (PW)
0032' 05E0 0001*    1171          INCT    @SSEG
                    1172
0036' C2E0 0000'    1173  10$     MOV     @LINKZ,R11          RESTORE LINK
003A' 045B          1174          RT
                    1175
                    1176  ********************************************************************
003C' 4900          1177  TAB     DATA    >4900
003E' 4A00          1178          DATA    >4A00
0040' 4B00          1179          DATA    >4B00
                    1180
                    1181+         LOCR    PRIV,LINKZ
0000' =0002         1184A LINKZ   BSS     2
                    1185
                    1186          END

No errors detected
```

```
                      1          IDT     IOB2SC
                      2
                      3          SUBTTL  CONVERT IOB TO 2'S COMPLEMENT
                      4
                      5  *       CALLING SEQ:    CALL    @IOB2SC
                      6
                      7  *-------------------------------------------------------------+
                      8  *                                                             !
                      9  *       IOB2SC CONVERTS 14-BIT IOB DATA INTO 12-BIT 2'S COMPLEMENT.
                     10  *       VOLT            -10     -5      0       5       10
                     11  *       IOB (HEX)       3FFF    3000    2000    1000    0
                     12  *       2'SC (HEX)      2001    3000    0       1000    2000
                     13  *                                                             !
                     14  *-------------------------------------------------------------+
                     15  *       VERSION : 1
                     16  *       PROGRAMMED BY : N.CONSTANTINIDES
                     17  *       CHECKED BY : N.CONSTANTINIDES
                     18
                     19
                     20          INTERN  IOB2SC
                     21
                     22
       =0000         23  SCR     EQU     R0      =       SCRATCH
       =0004         24  VAL     EQU     R4      =       IOB DATA(INPUT); 12-BIT 2'S COMPL (OUTPUT)
                     25
                     26
       =0000        157          RSECT   IOB2SC
                    158  ********************************************************************************
0000'                159  IOB2SC
0000'  0A24          160          SLA     VAL,C2          14-BIT LEFT-JUSTIF (WORK WITH 14 BIT DATA)
0002'  0504          161          NEG     VAL             VAL = - VAL
0004'  0224 8000     162          AI      VAL,CB000H      VAL = VAL + 2000(#4) (CARRY AWAY)
0008'  0824          163          SRA     VAL,C2          REGAIN 16-BIT/2 DATA
000A'  045B          164          RT
                    165  ********************************************************************************
                    166
                    167          END
```

```
                        1          IDT     JBQUE
                        2          SUBTTL  JOB QUEUE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: JBQUE                          AUTH: N.COSTANTINIDES     *
                        6  * VERSION: 1                           DATE: 23-OCT-81           *
                        7  *                                                                  *
                        8  * FUNCTION:     IT QUEUES THE BACKGROUND JOB IN THE JOB BUFFER   *
                        9  *               (JBBUF).  IT PLACES THE JOB NUMBER AND THE       *
                       10  *               BUFFER START ADDRESS IN THE QUEUE.  INPUT QUEUE  *
                       11  *               POINTER(JBIPTR) IS INCREMENTED BY 4 BYTES.       *
                       12  *               THE CALLER'S PC IS INCREMENTED BY 2 BYTES TO     *
                       13  *               SKIP OVER THE 'DATA' STATEMENT. JOB ARE          *
                       14  *               LISTED IN THE JOB LIST TABLE(JBTBL).             *
                       15  *               THE QUEUED JOB WILL BE EXECUTED BY THE EXTIVE.   *
                       16  *                                                                  *
                       17  * CALLING MODULES: REAL TIME ISR, INTER-CPU RECEIVER ISR,         *
                       18  *                                                                  *
                       19  * CALLING SEQ:  BLWP @JBQUE     JOB QUEUE                      *
                       20  *               DATA    N       JOB NUMBER (0 TO N)            *
                       21  *                                                                  *
                       22  * INPUT:        R0=BUFFER START ADDRESS                          *
                       23  *                                                                  *
                       24  * OUTPUT:       JOB NUMBER AND BUFFER POINTER IN JOBBUF BUFFER   *
                       25  *                                                                  *
                       26  * MODULES REFERENCED: NONE                                       *
                       27  *                                                                  *
                       28  * WORKSPACE AREA: WPEX                                           *
                       29  *                                                                  *
                       30  * REGISTERS MODIFIED: R8,R14                                     *
                       31  *                                                                  *
                       32  ********************************************************************
       =0000           33         RSECT   JBQUE
                       34  *** CALL NAME
                       35         INTERN  JBQUE
                       36  *** VARIABLES REFERENCED
                       37         EXTERN  JBBBUF,JBERUF
                       38         EXTERN  WPEX
                       39  *** REGISTER DEFINITION
       =0008           40  JBIPTR  EQU     R8              INPUT POINTER FOR JBBUF
       =0009           41  JBOPTR  EQU     R9              OUTPUT POINTER FOR JBBUF
       =000D           42  WP      EQU     R13             WORKSPACE POINTER
                       43  ********************************************************************
0000'  0003*           44  JBQUE   DATA    WPEX            WORKSPACE AREA
0002'  0004'           45         DATA    JB
                       46  *
0004'  0300 0001       47  JB      LIMI    1               DISABLE ALL INTERUPTS
0008'  0288 0002*      48         CI      JBIPTR,JBEBUF   INPUT POINTER ,VS. END OF JOB BUFFER
000C'  1A02            49         JL      20$             JIF NOT AT THE END
000E'  0208 0001*      50         LI      JBIPTR,JBBBUF   SET TO BEG OF JOB BUFFER
                       51  *
0012'  CE3E            52  20$     MOV     *R14+,*JBIPTR+  PLACE JOB NUMBER IN QUEUE,BUMP CALLER'S PC.
```

JOB QUEUE JBQUE.SRC

```
0014' CE1D      53          MOV   *R13,@JBIFTR+  PLACE BUFFER START ADDRESS
                54  *                            CALLER'S R0 IN QUEUE.
0016' 0380      55          RTWP                  RETURN
                56  *
                57          END
```

No errors detected

```
                    1          IDT     JOB8PS
                    2          SUBTTL  BACKGROUND JOB
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: JOB8PS.SRC                      AUTH: N.COSTANTINIDES     *
                    6  * VERSION: 1                            DATE: 5-APRIL-1982        *
                    7  *                                                                  *
                    8  * FUNCTION:THIS JOB IS PERFORMED 8 TIMES PER SECOND, INITIATED     *
                    9  *          BY THE REAL TIME ISR.                                   *
                   10  *          THE FOLLOWING JOBS ARE PERFORMED:                       *
                   11  *               1. DMX #1 & #2 DISCRETE ACQUISITION                *
                   12  *                  AND OUTPUT VIA DESTINATION TABLE                *
                   13  *                                                                  *
                   14  * CALLING MODULES: BACKGROUND                                      *
                   15  *                                                                  *
                   16  * CALLING SEQ: BL @JOB8PS                                         *
                   17  *                                                                  *
                   18  * INPUTS:                                                         *
                   19  *                                                                  *
                   20  * OUTPUTS:                                                        *
                   21  *                                                                  *
                   22  * MODULES REFERENCED:                                             *
                   23  *                                                                  *
                   24  * WORKSPACE AREA:                                                 *
                   25  *                                                                  *
                   26  * REGISTERS MODIFIED:                                             *
                   27  *                                                                  *
                   28  * VERSION HISTORY:                                                *
                   29  *                                                                  *
                   30  ********************************************************************
=0000              31         RSECT   JOB8PS
                   32  *** CALL NAME
                   33         INTERN  JOB8PS
                   34  *** VARIABLES REFERENCED
                   35  *** CONSTANTS REFERENCED
                   36  *** TABLES REFERENCED
                   37  *** MODULES REFERENCED
                   38         EXTERN  DC1ACQ
                   39  *** LIBRARY
                   40         INCLUDE ENCLOS
                   42  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   44         INCLUDE REGDEF          REGISTER DEFENITIONS
                   63         INCLUDE CNSTNT          CONSTANTS
                  191         INCLUDE SUBMAC          FUNCTIONAL MACROS
                  492         INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  722         INCLUDE JMPMAC          JUMP MACROS
                  756         INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  771         INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                 1147  *** REGISTERS DEFINITION
```

```
                    1149  ********************************************************************************
                    1150  *
0000'               1151  JOB8PS
                    1152  *
                    1153  *** DMX #1 DISCRETE ACQUISITION AND OUTPUT
                    1154  *
                    1155+         CALLXO  BC1ACQ
0000' 2CC0          1161A         XOP     R0,3
                    1167  *
0002' 0380          1168          RTWP
                    1169          END
```

No errors detected

```
                    1          IDT     JOB4PS
                    2          SUBTTL  BACKGROUND JOB
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: JOB4PS.SRC                      AUTH: N.COSTANTINIDES     *
                    6  * VERSION: 2                            DATE: 27-MAY-1983         *
                    7  *                                                                  *
                    8  * FUNCTION:BACKGROUND JOB CALLED BY EXECUTIVE                      *
                    9  *          THIS JOB IS PERFORMED 4 TIMES PER SECOND, INITIATED     *
                   10  *          BY THE REAL TIME ISR.                                   *
                   11  *          THE FOLLOWING JOBS ARE PERFORMED:                       *
                   12  *               1. MEMORY SUM CHECK ONCE EVERY SEC                 *
                   13  *                  ( 1 KWORD EVERY SEC )                           *
                   14  *               2. MONITORS EAROM TASK FOUR PER SEC (EAMON)        *
                   15  *                  MONITORS ERASE/WRITE PROCESS                    *
                   16  *                  INITIATE WRITE                                  *
                   17  *               3. MONITORS SYSTEM STATUS 1 PER SEC (SYSMON)       *
                   18  *                  CONVERTS ERROR BIT IN SYEBF OF BUFFER TO        *
                   19  *                       ERROR CODE WHICH WILL BE STORED IN         *
                   20  *                       TEMPORARY EAROM STATUSBUFFER AND SUPER-     *
                   21  *                       FRAME STATUS BUFFER                        *
                   22  *               4. PANEL LED DISPLAY & LAMP CONTROL 1 PER SEC      *
                   23  *                       (SYSOUT)                                   *
                   24  *                  WHEN READ SWITCH IS DEPRESSED IT CONTROLS       *
                   25  *                       LED DISPLAY & LAMPS                        *
                   26  *               5. EAROM ERASE DISCRETE TEST                       *
                   27  *                                                                  *
                   28  * CALLING MODULES: EXTIVE                                         *
                   29  *                                                                  *
                   30  * CALLING SEQ: BLWP @JOB4PS                                       *
                   31  *                                                                  *
                   32  * INPUTS: JB4CYL = JOB 4 PER SEC COUNTER (0,1,2,&3)               *
                   33  *                                                                  *
                   34  * OUTPUTS:                                                        *
                   35  *       JB4CYL = JOB 4 PER SEC COUNTER                            *
                   36  *                                                                  *
                   37  * MODULES REFERENCED:                                             *
                   38  *               EAMON   =       EAROM MONITOR                     *
                   39  *               SYSMON  =       SYSTEM STATUS MONITOR             *
                   40  *               SYSOUT  =       LAMPS & DISPLAY                   *
                   41  *               MSCHK   =       MEMORY SUM CHECK                  *
                   42  *               EABTST  =       EAROM ERASE BIT TEST              *
                   43  *                                                                  *
                   44  * WORKSPACE AREA: WPJB                                            *
                   45  *                                                                  *
                   46  * REGISTERS MODIFIED: R9                                          *
                   47  *                                                                  *
                   48  * VERSION HISTORY:                                                *
                   49  *       VERSION 2 ; SCR 000050                                    *
                   50  *                                                                  *
                   51  ********************************************************************
=0000              52         RSECT   JOB4PS
```

```
                    53  ;;; CALL NAME
                    54          INTERN  JOB4PS
                    55  ;;; VARIABLES REFERENCED
                    56  ;;; CONSTANTS REFERENCED
                    57  ;;; TABLES REFERENCED
                    58  ;;; MODULES REFERENCED
                    59          EXTERN  DFBTST
                    60          EXTERN  EAMON,EABTST
                    61          EXTERN  MSCHK,SUPRFX
                    62          EXTERN  SYSMON,SYSOUT
                    63  ;;; LIBRARY
                    64          INCLUDE ENCLOS
                    66  ;;;     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    68          INCLUDE REGDEF          REGISTER DEFENITIONS
                    87          INCLUDE CNSTNT          CONSTANTS
                   215          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   516          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   746          INCLUDE JMPMAC          JUMP MACROS
                   780          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   795          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                  1171  ;;; REGISTERS DEFINITION
=0009             1172  JB4CYL  EQU     R9              JOB4PS COUNTER, INCREMENTED EVERY JOB4PS CALL  (0 TO 3)
=000A             1173  RSCTR   EQU     R10             READ SWITCH COUNTER (SYSOUT)
```

```
                    1175  ********************************************************************
                    1176  *
                    1177* JOB4PS  ENABLE  C17             RESET INTEGRATOR EVERY 250 MSEC
0000' 020C 0700     1178A         LI      CRU,C9901
0004' 1D 11         1179A         SBO     C17
0006' 1E 11         1180          SBZ     C17
                    1181  *** INCREMENT FROM CYCLE COUNTER EVERY CALL
0008' 0589          1182          INC     JB4CYL
000A' 0249 0003     1183          ANDI    JB4CYL,3        COUNTER 0 TO 3 ONLY
                    1184  ********************************************************************
                    1185  *** EAROM MONITOR, 4 PER SEC
000E' 0420 0002*    1186          BLWP    @EAMON
0012' 06A0 0005*    1187          BL      @SUPBFX
                    1188  ***
0016' 0289 0001     1189          CI      JB4CYL,1
001A' 1606          1190          JNE     20$             JIF NOT SECOND CYCLE
                    1191  ********************************************************************
                    1192  *   2ND CYCLE OF 4 CYCLE
                    1193  *** SYSTEM MONITOR, 1 PER SEC
001C' 06A0 0006*    1194          BL      @SYSMON
                    1195  *** PANEL LED DISPLAY AND LAMPS, 1 PER SEC,
0020' 06A0 0007*    1196          BL      @SYSOUT
0024' 06A0 0001*    1197          BL      @DFBTST         DFDR BITE
                    1198  ***
0028' 0289 0002     1199  20$     CI      JB4CYL,2
002C' 1602          1200          JNE     30$             JIF NOT 3RD CYCLE
                    1201  ********************************************************************
                    1202  *   3RD CYCLE OF 4 CYCLE
                    1203  *** DFDR PLAYBACK SYNC CODE SEARCH COMMAND
002E' 06A0 0003*    1204          BL      @EABTST         TEST EAROM ERASE DISCRETE
0032' 0289 0003     1205  30$     CI      JB4CYL,3
0036' 1602          1206          JNE     JOB4EX          JIF NOT 4TH CYCLE
                    1207  ********************************************************************
                    1208  *   4TH CYCLE OF 4 CYCLE
                    1209  *** MEMORY SUM CHECK, 1 PER SEC
                    1210  *
0038' 06A0 0004*    1211          BL      @MSCHK
003C' 0380          1212  JOB4EX  RTWP
                    1213  ********************************************************************
                    1214          END
```

No errors detected

```
                        1          IDT     MSCHK
                        2
                        3          SUBTTL  MEMORY CHECK-SUM
                        4
                        5  *       CALLING SEQ:    CALL    @MSCHK
                        6
                        7  *--------------------------------------------------------------------+
                        8  *                                                                     !
                        9  *       MSCHK TEST EPROM (CHECK-SUM) IN THE BACKGROUND.
                       10  *       CALLING MODULE: JOB4PS
                       11  *                                                                     !
                       12  *--------------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  MSCHK
                       18  *** VARIABLES REFERENCED:
                       19          EXTERN  MSIX,MSERR
                       20  *** CONSTANTS REFERENCED
                       21          EXTERN  B0
                       22  * REFERD MODULES:
                       23          EXTERN  MSSBR
                       24          EXTERN  SYSER,SYSOK
                       25  *** TABLES REFERENCE:
                       26          EXTERN  MEND,MSBUF
          =0000        27  SUM     EQU     R0      =       SCRATCH
          =0001        28  CNT     EQU     R1      =       SCRATCH
          =0005        29  MSP     EQU     R5      =       SCRATCH
                       30  *               R2,R3,R4        =       SCRATCH
                       31
                       32          INCLUDE ENCLOS
                       34  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       36          INCLUDE REGDEF          REGISTER DEFENITIONS
                       55          INCLUDE CNSTNT          CONSTANTS
                      188          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      489          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      719          INCLUDE JMPMAC          JUMP MACROS
                      753          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      768          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000      1144          RSECT   MSCHK
                     1145  ***:*****************************************************************
0000 C0CB            1146  MSCHK   MOV     LINK,R3         SAVE LINK REG
0002 C160 0001*      1147          MOV     @MSIX,MSP       MEMORY SUM INDEX(0,2,4,...)
0006 0201 0800       1148          LI      CNT,>800        1 KWORD CONSTANT
000A 8805 0007*      1149          C       MSP,@MEND       MEMORY SUM LIMIT
000E 1A02            1150          JL      2$
0010 1304            1151          JEQ     3$              JUMP IF LAST K WORD SECTION
0012 04C5            1152          CLR     MSP             RESET  TO START FROM 1ST KWORD
                     1153  *     1ST THRU ONE BEFORE LAST K WORD BLOCK
0014                 1154  2$
                     1155+         CALL    MSSBR           SUM 1 KWORD BLOCK. RESULT IN R0 (SUM)
```

```
00[illegible]  06A0 0004*  1162A          BL    @MSSBR
0018'  1007        1164          JMP   5$            CHEK RESULT
                   1165  *     LAST BLOCK
0018'              1166  3$
0018'  C085        1167          MOV   MSP,R2       POINTER
001C'  0AA2        1168          SLA   R2,10        * 1K
001E'  0201 0007*  1169          LI    CNT,MEND     LAST WORD TO BE SUM
0022'  6042        1170          S     R2,CNT       NO. OF WORDS TO BE SUMMED
                   1171+         CALL  MSSBR        SUM THE REST OF WORDS
0024'  06A0 0004*  1178A          BL    @MSSBR
0028'              1180  5$
0028'  A025 0008*  1181          A     @MSBUF(MSP),SUM
002C'  1303        1182          JEQ   7$           JUMP IF SUM OK
                   1183  *  FAIL CHECK SUM TEST
002E'  E825 0003*  1184          SOC   @B0(MSP),@MSERR      SET ERROR BIT FOR THIS BLOCK
0032'  0002*
0034'  C060 0000'  1185          MOV   @ERCOD,R1            ERROR CODE
0038'  06A0 0005*  1186          BL    @SYSER               SET ERROR BIT IN SYEBF
003C'  100A        1187          JMP   10$
                   1188  *  PASS CHECK SUM FOR THIS BLOCK
003E'  4825 0003*  1189  7$      SZC   @B0(MSP),@MSERR RESET ERROR BIT FOR THIS BLOCK
0042'  0002*
0044'  C020 0002*  1190          MOV   @MSERR,R0
0048'  1604        1191          JNE   10$          JIF SOME BLOCK IS FAILING
                   1192  *  PASS MEMORY CHECK SUM FOR ALL BLOCKS
004A'  C060 0000'  1193          MOV   @ERCOD,R1            ERROR CODE
004E'  06A0 0006*  1194          BL    @SYSOK       RESET ERROR BIT IN SYEBF
0052'  05C5        1195  10$     INCT  MSP          BUMP POINTER FOR NEXT ITERATION
0054'  C805 0001*  1196          MOV   MSP,@MSIX    RESTORE MEMORY SUM INDEX
0058'  C2C3        1197          MOV   R3,LINK      UNSAVE LINK REG
005A'  045B        1198          RT
                   1199  ********************************************************************************
                   1200+         PRVDAT
0[illegible]'  0101        1202  ERCOD   DATA  >101         CHECK-SUM ERROR CODE
                   1203          END
```

```
                    1          IDT     MSGEN
                    2
                    3          SUBTTL MEMORY SUM WORD GENERATION
                    4
                    5  *       CALLING SEQ:    CALL    @MSGEN
                    6
                    7  *---------------------------------------------------------------------+
                    8  *                                                                      !
                    9  *       GENMS GENERATES MEMORY SUM WORDS FOR EVERY 1K WORD BLOCK
                   10  *       AND STORE THEM IN MEMORY SUM BUFFER (MSBUF) [IN EPROM].
                   11  *       IT IS USED ONLY AT SYSTEM INTEGRATION LEVEL !!
                   12  *                                                                      !
                   13  * CALLING MODULE: PON (POWER ON ISR)                                   !
                   14  *---------------------------------------------------------------------+
                   15  *       VERSION : 2
                   16  *       PROGRAMED BY : N.CONSTANTINIDES
                   17  *       CHECKED BY : N.CONSTANTINIDES
                   18
                   19          INTERN  MSGEN
                   20  * REFERD MODULES:
                   21          EXTERN  MSSBR
                   22  * GLOBALS:
                   23  *       (ROM)
                   24          EXTERN  MEND            .2 EPROM END (START OF CHECK-SUM TABLE)
                   25  *                               ALSO CHECK-SUM LIMIT
                   26          EXTERN  MSBUF           CHEK-SUM-TABLE
                   27  *       (RAM)
                   28
=0000              29  VAL     EQU     R0      =       SCRATCH
=0001              30  CNT     EQU     R1      =       SCRATCH
=0002              31  TMP     EQU     R2      =       SCRATCH
=0004              32  CLM     EQU     R4      =       SCRATCH
=0005              33  CSP     EQU     R5      =       SCRATCH
                   34  *
                   35          INCLUDE ENCLOS
                   37  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   39          INCLUDE REGDEF          REGISTER DEFENITIONS
                   58          INCLUDE CNSTNT          CONSTANTS
                  191          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  492          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  722          INCLUDE JMPMAC          JUMP MACROS
                  756          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  771          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000            1147          RSECT   MSGEN
                 1148  ********************************************************************************
```

```
0000'                1150  MSGEN
0000'  C0CB          1151        MOV     LINK,R3          SAVE LINKER
0002'  04C5          1152        CLR     CSP              BLOCK-OF-1000-WORDS OFFSET = 0.
0004'  0204 0002*    1153        LI      CLM,MEND
0008'  09B4          1154        SRL     CLM,11           COMPUTE MEMORY SUM LIMIT
000A'  0A14          1155        SLA     CLM,1
000C'  C804 0002*    1156        MOV     CLM,@MEND        SAVE LIMIT FOR CHKSM
0010'                1157  2$
0010'  0201 0800     1158        LI      CNT,>800               1 KWORD CONSTANT
0014'                1159  4$
0014'  06A0 0001*    1160        BL      @MSSBR           SUM IN VAL
0018'  0500          1161        NEG     VAL              TWO'S COMPLEMENT THE SUM
001A'  C940 0003*    1162        MOV     VAL,@MSBUF(CSP)  STORE IT
001E'  05C5          1163        INCT    CSP              FOR NEXT BLOCK
0020'  8805 0002*    1164        C       CSP,@MEND
0024'  1AF5          1165        JL      2$               LOOP
0026'  1606          1166        JNE     9$               EXIT
                     1167  *     LAST BLOCK
0028'  C085          1168        MOV     CSP,TMP
002A'  0AA2          1169        SLA     TMP,10           * 1K
002C'  0201 0002*    1170        LI      CNT,MEND               LAST WORD TO BE SUMMED
0030'  6042          1171        S       TMP,CNT          NO. OF WORDS TO BE SUMMED
0032'  10F0          1172        JMP     4$
0034'                1173  9$
0034'  C2C3          1174        MOV     R3,LINK          RESTORE LINKER
0036'  045B          1175        RT
                     1176  ********************************************************************************
                     1177        END
```

```
                        1          IDT     PIDO
                        2
                        3          SUBTTL  INITIALIZE DATA OUTPUT
                        4
                      · 5  *       CALLING SEQ:    CALL    @PIDO
                        6
                        7  *-------------------------------------------------------------------+
                        8  *                                                                   !
                        9  *       PIDO SETS ALL OUTPUT POINTERS (TO THE START OF LAST CYCLE
                       10  *       IN OUTPUT-BUFFERS), CYCLE-, SUBFRAME- AND FRAME-COUNT,
                       11  *       ACTIVATES AND STARTS ALL (DFDR, AUX AND 429) OUTPUTS,
                       12  *       SENDS SYNC-CODE TO PLAY-BACK AND ENABLE PLAY-BACK SYNC
                       13  *       INTERRUPT.
                       14  *                                                                   !
                       15  *-------------------------------------------------------------------+
                       16  *       VERSION : 1
                       17  *       PROGRAMMED BY : M.CONSTANTINIDES
                       18  *       CHECKED BY : M.CONSTANTINIDES
                       19
                       20          INTERN  PIDO
                       21  * GLOBAL AREA:
                       22          EXTERN  DFWRC           DFDR WORD COUNT (POINTER) (SCRT
                       23
        =0000          24  SCR     EQU     R0      =       SCRATCH
                       25
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      709          INCLUDE JMPMAC          JUMP MACROS
                      743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1135          RSECT   PIDO
                     1136  ********************************************************************************
    0000'            1137  PIDO
                     1138+         MOVI    C55*2,DFWRC             SET DF-WR-COUNT TO LAST CYCLE
    0000'  0200 006E 1140A         LI      R0,C55*2
    0004'  C800 0001* 1141A         MOV     R0,@DFWRC
                     1143  * ACTIVATE AUX
                     1144
                     1145  * ACTIVATE 429
                     1146
                     1147  * ACTIVATE DFDR PLAY-BACK SYNC INT
                     1148  * ACTIVATE DFDR OUTPUT INTERRUPT
                     1149+         ENABLE  5,                      ENABLE DFDR INT VIA 9901
    0008'  020C 0700 1150A         LI      CRU,C9901
    000C'  1D 05     1151A         SBO     5
    000E'  045B      1152          RT
                     1153  ********************************************************************************
                     1154          END
```

```
                        1          IDT     'NICHTS'
                        2
                        3          SUBTTL   'NOTHING'
                        4
                        5  *       CALLING SEQ:    CALLWP  @NICHTS
                        6  *                       (OR UNUSED INTERRUPT)
                        7
                        8  *-------------------------------------------------------------------+
                        9  *                                                                   !
                       10  *       NICHTS SERV (SPECIALY) ALL INTERRUPTS WHICH ARE
                       11  *       NOT USED SO THAT IN THE CASE OF OCCURENCE OF AN
                       12  *       UNDESIRED INTERRUPT, IT DOES NOT GET LOST.
                       13  *                                                                   !
                       14  *-------------------------------------------------------------------+
                       15  *       VERSION : 1
                       16  *       PROGRAMMED BY : M.CONSTANTINIDES
                       17  *       CHECKED BY: M.CONSTANTINIDES
                       18
                       19          INTERN  NICHTS
                       20          EXTERN  RTSIN1
                       21
          =0000        22          RSECT   NICHTS
                       23
                       24  ********************************************************************
                       25  *
     0000'             26  NICHTS
                       27
     0000'  06A0 0001* 28          BL      @RTSIN1
     0004'  0380       29          RTWP                    RETURN
                       30
                       31  ********************************************************************
                       32
                       33          END
```

```
                     1          IDT    PIHW
                     2
                     3          SUBTTL  INITIALIZE H/W
                     4
                     5  *       CALLING SEQ:    CALL    @PIHW
                     6
                     7  *-------------------------------------------------------------+
                     8  *                                                             !
                     9  *       PIHW EXTINGUISHES FRONT PANEL 'DFDR FAIL', 'CAUTION'
                    10  *       AND 'FAIL' LAMPS, SET FRONT-PANEL DISPLAY TO '0' AND
                    11  *       DETERMINES THE ENGINE TYPE.
                    12  *                                                             !
                    13  *-------------------------------------------------------------+
                    14  *       VERSION : 1
                    15  *       PROGRAMMED BY : N.CONSTANTINIDES
                    16  *       CHECKED BY : N.CONSTANTINIDES
                    17
                    18
                    19          INTERN  PIHW
                    20  * REFERD MODULES:
                    21          EXTERN  DCDRQ
                    22
                    23  * GLOBALS :
                    24          EXTERN  D0      = 0 (ROM)
                    25
=0000               26  CNT     EQU     R0      =       SCRATCH
=0001               27  PTR     EQU     R1      =       SCRATCH
=0004               28  DDT     EQU     R4      =       SCRATCH
                    29  *               R5      =       SCRATCH
=000A               30  VSU     EQU     R10     =       VERSION NO (OUTPUT)
                    31
                    32          INCLUDE ENCLOS
                    34  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    36          INCLUDE REGDEF          REGISTER DEFENITIONS
                    55          INCLUDE CNSTNT          CONSTANTS
                   183          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   484          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   714          INCLUDE JMPMAC          JUMP MACROS
                   748          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   763          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1139          RSECT   PIHW
                  1140  ********************************************************************************
```

```
0000'              1142  PIHW
0000'  C80B 0000'  1143          MOV     LINK,@LINKZ             SAVE LINKER
                   1144+         XTOFPL                          EXTINQUISH FRONT-PANNEL LAMPS, FP=0
0004'  020C 05E0   1145A         LI      CRU,CRUFPL              GET FRONT PANEL LAMPS CRU ADDR
0008'  1E 00       1146A         SBZ     ZERO                    EXTINQ 'DFDR FAIL'
000A'  1E 01       1147A         SBZ     C1                      EXTINQ 'FAIL'
000C'  1E 02       1148A         SBZ     C2                      EXTINQ 'CAUTION'
000E'  C820 0002*  1149A         MOV     @D0,@CFPDSP             FRONT-PANEL DISPLAY =  0
0012'  FF80
0014'  1E 03       1150          SBZ     C3                      DISABLE U-CIRCUTE (LOW FOR EVER !!)
0016'  C2E0 0000'  1151          MOV     @LINKZ,LINK             RESTORE LINKER
001A'  045B        1152          RT
                   1153  ********************************************************************************
                   1154  * PRICVATE AREA
                   1155+         LOCR    PRIV,LINKZ              LINK-SAVE AREA
0000'  =0002       1158A LINKZ   BSS     2
                   1159  *                                       TO DETERMINE ENGINE TYPE
                   1160          END
```

```
                        1          IDT    POWDN
                        2          SUBTTL  POWER DOWN INTERRUPT SERVICE ROUTINE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: POWDN                          AUTH: N.COSTANTINIDES     *
                        6  * VERSION:1                            DATE: 9 APR 82
                        7  *                                                                  *
                        8  * FUNCTION: POWER DOWN INTERRUPT SERVICE ROUTINE.
                        9  *              THIS MODULE WILL BE EXECUTED WHEN POWER IS
                       10  *              LOST.
                       11  *                                                                  *
                       12  * CALLING MODULES: INTERRUPT #1                                    *
                       13  *                                                                  *
                       14  * CALLING SEQ:
                       15  *                                                                  *
                       16  * INPUTS:
                       17  *                                                                  *
                       18  * OUTPUTS:
                       19  *                                                                  *
                       20  * MODULES REFERENCED:
                       21  *                                                                  *
                       22  * WORKSPACE AREA:
                       23  *                                                                  *
                       24  * REGISTERS MODIFIED:
                       25  *                                                                  *
                       26  * VERSION HISTORY:
                       27  *                                                                  *
                       28  ********************************************************************
       =0000           29         RSECT   POWDN
                       30  *** CALL NAME
                       31         INTERN  POWDN
                       32  *** VARIABLES REFERENCED
                       33  *** CONSTANTS REFERENCED
                       34  *** TABLES REFERENCED
                       35  *** MODULES REFERENCED
                       36  *** LIBRARY
                       37         INCLUDE REGDEF          REGISTER DEFINITIONS
                       56         INCLUDE CNSTNT          CONSTANTS
                      184  *** REGISTERS DEFINITION
                      185  ********************************************************************
0000' 058B            186  POWDN   INC     R11             JUST FOR CHECKOUT
0002' 020C 0700       187         LI      CRU,C9901       SET CRU REG WITH 9901 ADDRESS
0006' 1E 10           188  HOLD    SBZ     16              HOLD MICRO PROCESSOR
0008' 10FE            189         JMP     HOLD            LOOP FOR EVER UNTIL NO MORE POWER
                      190  ********************************************************************
                      191         END
```

No errors detected

```
                        1
                        2          IDT     PON
                        3
                        4          SUBTTL  POWER ON
                        5
                        6  *       CALLING SEQ:    POWER-ON INTERRUPT (INT 0)
                        7
                        8  *-------------------------------------------------------------------+
                        9  *                                                                   |
                       10  *       PON OPERATS ON POWER-ON ENTRY.
                       11  *       IT INITIALIZES H/W, S/W, INTERRUPTS, DATA-IN AND -OUT AND
                       12  *       DOES SOME POWER ON DIAGNOSTIC.
                       13  *                                                                   |
                       14  *       VERSION HISTORY:                                              |
                       15  *                                                                     |
                       16  *-------------------------------------------------------------------+
                       17  *       CALLING MODULE: H/W
                       18  *       VERSION : 2
                       19  *       PROGRAMMED BY : M.CONSTANTINIDES
                       20  *       MODIFIED : 26-MAY-1983 M.CONSTANTINIDES
                       22          INTERN  PON
                       23  * VARIABLES REFERENCED
                       24          EXTERN  FRONT,CYPFRC
                       25          EXTERN  RAMBEG,RAMEND,POTRAM
                       26  * REFERD MODULES:
                       27          EXTERN  EXTIVE
                       28          EXTERN  ICINIT
                       29          EXTERN  MSGEN
                       30          EXTERN  PIHW,PISW,PINT
                       31          EXTERN  PIDA,PIDO
                       32          EXTERN  IDENT,RTSDL1
                       33  *** REGISTER DEFINITION:
        =0007          34  PINIT   EQU     R7      ANY POWER INTERRUPT INIT FLAG (RITFR)
        =0008          35  POCALF  EQU     R8      POWER-ON-CALIBRATION-FLAG
        =0009          36  POR9    EQU     R9
        =000A          37  VRSMSK  EQU     R10     VERSION NUMBER
                       38
                       39  *** LIBRARY
                       40          INCLUDE ENCLOS
                       42  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       44          INCLUDE REGDEF          REGISTER DEFENITIONS
                       63          INCLUDE CNSTNT          CONSTANTS
                      191          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      492          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      722          INCLUDE JMPMAC          JUMP MACROS
                      756          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      771          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1147          RSECT   PON
                     1148  ********************************************************************************
    0000'            1149  PON
    0000'  06A0 000F*  1150          BL      @RTSDL1         RTS
```

```
0004' 0300 0000  1151          LIMI   ZERO          DISABLE ALL INTERRUPTS
0008' 0200 0002  1152          LI     R0,2          SET COUNTER TO 2 INITIAL VALUE
000C' C800 0000' 1153          MOV    R0,@CNT
                 1154
                 1155  ***
                 1156  *** CHECK POWER TRANSIENT OR TRUE POWER ON (SWITCH TO ON) INTERRUPT
                 1157  *** RAM IS PROTECTED FOR THE 1ST 200 MSEC OF POWER OFF PERIOD.
                 1158  *** PREDEFINED PATTERN WILL BE STORED IN THE RESERVED RAM(HEX LOC D000).
                 1159  *** EVERY POWER ON INTERRUPT THE PATTERN IN THE RAM WILL BE CHECKED
                 1160  *** IF PATTERN IS VALID, IT IS DEFINED AS A POWER TRANSIENT.
                 1161  *** IF PATTERN IS INVALID, THEN IT IS A TRUE POWER ON TYPE.
                 1162
0010' 0200 0003  1163          LI     R0,3          NUMBER OF PATTERN
0014' 0201 0094' 1164          LI     R1,POTPAT     PATTERN START ADDRESS
0018' 0202 0005* 1165          LI     R2,POTRAM     RAM START ADDRESS
001C' 8CB1       1166  20$     C      *R1+,*R2+
001E' 160F       1167          JNE    22$           JIF RAM IS BAD (TRUE POWER ON)
0020' 0600       1168          DEC    R0
0022' 15FC       1169          JGT    20$           LOOP BACK FOR MORE CHECK
                 1170  *
                 1171  *  RAM IS GOOD
                 1172  *  CORRECT FRAME/SUBFRAME COUNTER TO START AT NEXT SUBFRAME
                 1173  *
0024' C0A0 0001* 1174  99$     MOV    @FRCNT,R2     SAVE FRAME/SUBFRAME COUNTER
0028' 0201 0003* 1175          LI     R1,RAMBEG     START ADDR OF RAM EXCLUDING POWER ON WP
002C' 04F1       1176  21$     CLR    *R1+          CLEAR RAM WORD
002E' 0281 0004* 1177          CI     R1,RAMEND     END OF RAM ?
0032' 1AFC       1178          JL     21$           LOOP IF NOT
0034' 0720 0002* 1179          SETO   @CYPFRC       RESET CYCLE PER FRAME COUNTER
0038' C802 0001* 1180          MOV    R2,@FRCNT     RESTORE FRAME/SUBFRAME COUNTER
003C' 1012       1181          JMP    60$           RAM IS GOOD (POWER TRANSIENT)
                 1182  *
                 1183  *  RAM IS BAD. RESTORE PATTERN
                 1184  *
003E' 0201 0094' 1185  22$     LI     R1,POTPAT     PATTERN START ADDRESS
0042' 0202 0005* 1186          LI     R2,POTRAM     RAM START ADDRESS
0046' CCB1       1187          MOV    *R1+,*R2+     PUT 1ST PATTERN IN POWER ON TRANS RAM WORDS
0048' CCB1       1188          MOV    *R1+,*R2+     PUT 2ND
004A' CCB1       1189          MOV    *R1+,*R2+     PUT 3RD
                 1190
                 1191  *** CLEAR ALL RAM WORDS FROM RAMBEG TO RAMEND (POTRAM IS NOT IN THIS AREA)
                 1192
004C' 0201 0003* 1193          LI     R1,RAMBEG     START ADDRESS OF RAM EXCLUDING POWER ON WP
0050' 04F1       1194  50$     CLR    *R1+          CLEAR RAM WORD
0052' 0281 0004* 1195          CI     R1,RAMEND     END OF RAM ?
0056' 1AFC       1196          JL     50$           LOOP IF NOT
0058' 0720 0002* 1197          SETO   @CYPFRC       RESET CYCLE PER FRAME COUNTER
005C' 0720 0001* 1198          SETO   @FRCNT        RESET FRAME/SUBFRAME COUNTER
0060' 1000       1199          JMP    60$
                 1200
                 1201  *** COMMON ENTRY
                 1202
```

```
                    1203+ 60$   CALL   PIHW          INITIATE H/W
0062' 06A0 0009$    1210A         BL   @PIHW
                    1212+       CALL   PISW          INITIATE S/W
0066' 06A0 000A$    1219A         BL   @PISW
                    1221+       CALL   IDENT         SELECT A/ ENGINE
006A' 06A0 000E$    1228A         BL   @IDENT
006E' 0701          1230        SETO   R1
0070' C041          1231        MOV    R1,R1
0072' 1602          1232        JNE    64$           JIF NO NEED GENERATE SUM CHECK IN OPER. PROG.
                    1233+       CALL   MSGEN           TO GENERATE SUM-VALUES INTO EPROM
0074' 06A0 000B$    1240A         BL   @MSGEN
                    1242+ 64$   CALL   PINT          INITIALIZE INTERRUPTS
0078' 06A0 000B$    1249A         BL   @PINT
                    1251+       CALL   PIDA          INITIATE DATA INPUT
007C' 06A0 000C$    1258A         BL   @PIDA
0080' 0620 0000'    1260        DEC    @CNT          DECREMENT COUNTER
0084' 16CF          1261        JNE    99$           JIF FIRST TIME ROUND
                    1262
                    1263  ********************************************************************
                    1264  *       THE ABOVE SECTION IS REPEATED ONCE BECAUSE IT WAS DISCOVERED
                    1265  *       THAT 'SOMETHING' HAD TO BE INITIALIZED TWICE, OTHERWISE THE
                    1266  *       DITS W/A TEST WOULD FAIL. FOR SOME UNKNOWN REASON THES SEEMS
                    1267  *       TO WORK.
                    1268  ********************************************************************
                    1269
                    1270+       CALL   PIDO          INITIATE DATA OUTPUT
0086' 06A0 000D$    1277A         BL   @PIDO
                    1279  *     CALL   PODPIN        INITIALIZE DFDR PLAYBACK
                    1280
                    1281  *** INITIALIZE INTER-CPU SOFTWARE AND HARDWARE
                    1282+       CALL   ICINIT
0088' 06A0 0007$    1289A         BL   @ICINIT
                    1291  ***   PINIT=-1 INDICATES POWER ON INTERRUPT. RTIFR MODULE USES THIS FLAG
                    1292  ***   TO DELAY ONE SUBFRAME BEFORE STARTING DATA TRANSMISSION TO CPU#2.
008E' 0707          1293        SETO   PINIT         SET IT
                    1294  *** GO TO EXECUTIVE
                    1295+       CALLWP EXTIVE        ENTER BACK-GROUND
0090' 0420 0006$    1297A         BLWP @EXTIVE
                    1299  ********************************************************************
                    1300  *** POWER ON TRANSIENT FIXED PATTERN
0094' AAAA          1301  POTPAT DATA   >AAAA         1ST PATTERN
0096' 1234          1302        DATA   >1234         2ND   "
0098' 5678          1303        DATA   >5678         3RD   "
                    1304
                    1305+       LOCR   PRIV,CNT
0000' =0002         1308A CNT   BSS    2
                    1309
                    1310  *
                    1311        END
```

```
                    1          IDT     PINT
                    2
                    3          SUBTTL  INITIALIZE INTERRUPTS
                    4
                    5  *       CALLING SEQ:    CALL    @PINT
                    6
                    7  *---------------------------------------------------------------
                    8  *                                                              !
                    9  *       PINT INITIALIZES 3-THRU-11 EXCEPT INTERRUPT 7 (INTER CPU)
                   10  *       VIA THE 9901 (PROGRAMMABLE SYSTEMS INTERFACE).
                   11  *       REAL TIME INTERRUPT (INT 3) IS INITIALIZED FOR 15.616 MSEC
                   12  *       WHICH EXACTLY AS LONG AS DFDR INTERRUPT (INT 5).
                   13  *                                                              !
                   14  *       VERSION HISTORY:
                   15  *
                   16  *---------------------------------------------------------------
                   17  *       VERSION : 1
                   18  *       PROGRAMMED BY : N.CONSTANTINIDES
                   19  *       CHECKED BY : N.CONSTANTINIDES
                   20
                   21          INTERN  PINT
                   22  * GLOBAL AREA:
                   23  *       (RAM)
                   24          EXTERN  RTPCYC          RT-PER-CYCLE COUNT
                   30          INCLUDE REGDEF          REGISTER DEFENITIONS
                   49          INCLUDE CNSTNT          CONSTANTS
                  182          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  483          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  713          INCLUDE JMPMAC          JUMP MACROS
                  747          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  762          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                 1139          RSECT   PINT
      =0000      1140  ********************************************************************
                 1141  PINT
                 1142          SETO    @RTPCYC                 DONT START DATA-ACQ YET
0000'
0000'  0720 0001 1143  *
                 1144+         RESET   INT6,
                                 CLR   @CADCLR                 RESET ADC-INT
                 1147A         RESET   INT10,
0004'  04E0 FFBB 1155+           MOV   @CD11OP,R0             RESET DM3-1 INT
                 1161A         RESET   INT11,
0008'  C020 FF84 1166+           MOV   @CD21OP,R0             RESET DM3-2 INT
                 1175A         LI      CRU,C9901               9901 CRU ADDR
000C'  C020 FF88 1177          LDCI    @INTIM,ZERO             INTERVAL-TIMER = 15.6 MSEC
0010'  020C 0700 1178                          DFDR (INT 5) OCCURES BEFOR RESETING IT !!!
0014'  3020 0000'1179  *                       ENABLE ONLY RT, ADC
                 1180  *       LDR     @INTEN,ZERO             INITIALIZE INT 3, 6.
                 1181          LIM     INTMSK                  SET MASK
0018'  3020 0002'1182                          FROM NOW ON THE R-T INTERRUPT IS ACTIVE
001C'  0300 000D 1183  *       R
                 1184  ********************************************************************
0020'  045B      1185          #DAT
                 1186+
```

```
                    1188  *PRIVATE DATA AREA
0000' 05B9          1189  INTIM  DATA  >5B9          INTERVAL TIMER FOR 15.6 MSEC
0002' 004A          1190  INTEN  DATA  >04A          INTR 1, 3, 6, ENABLE
                    1191
                    1192        END
```

```
                     1          IDT     RMSG1
                     2          SUBTTL  RECEIVE MSG TYPE #1 SUBROUTINE
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: RMSG1                          AUTH: N.COSTANTINIDES     *
                     6  * VERSION: 1                           DATE: 8-FEB-84            *
                     7  *                                                                  *
                     8  * FUNCTION:     RECEIVE MSG TYPE #1 SUBROUTINE.                   *
                     9  *               TRANSFERS DOC DATA FROM CPU #2 TO CPU #1          *
                    10  *               DOC DATA BUFFER.                                  *
                    11  *                                                                  *
                    12  * CALLING MODULES: ICMD=INTER CPU MESSAGE DISPATCHER               *
                    13  *                                                                  *
                    14  * CALLING SEQ: BL @RMSG1                                          *
                    15  *                                                                  *
                    16  * INPUTS:       R1=RECEIVE BUFFER START ADDRESS                   *
                    17  *                                                                  *
                    18  * OUTPUTS:      CPU#2 DOC DATA TO DOC DATA BUFFER                 *
                    19  *               TMA-BIT8 = 1 FOR MSG #1 REC OK STATUS             *
                    20  *               1ST WORD OF RC MSG BUFFER=0 (MAKE IT AVAILABLE)   *
                    21  *                                                                  *
                    22  * MODULES REFERENCED:                                             *
                    23  *                                                                  *
                    24  * WORKSPACE AREA:       CALLER'S                                  *
                    25  *                                                                  *
                    26  * REGISTERS MODIFIED:   R0 TO R9                                  *
                    27  *                                                                  *
                    28  ********************************************************************
       =0000        29          RSECT   RMSG1
                    30  *** CALL NAME
                    31          INTERN  RMSG1
                    32  *** VARIABLES REFERENCED
                    33          EXTERN  SYDOCB,TMA
                    34  *** CONSTANTS REFERENCED
                    35          EXTERN  D15,B8
                    36  *** MODULES REFERENCED
                    37  *** TABLES REFERENCED
                    38  *** REGISTERS DEFINITION
                    39  ********************************************************************
0000'               40  RMSG1
                    41  * COPY INTER CPU RECEIVE MSG BUFFER IN DOC DATA BUFFER
0000' C0C1          42          MOV     R1,R3           SAVE RECEIVER BUFFER START ADDRESS
0002' 0200 0005     43          LI      R0,5            5 NUMBER OF DATA WORDS
                    44  *
                    45  *       DATA WORDS ARE (IN ORDER)
                    46  *       1. DAY
                    47  *       2. MONTH
                    48  *       3. FLT NUMBER MSH
                    49  *       4. FLT NUMBER LSH
                    50  *       5. FLT LEG
                    51  *
                    52          AI      R1,4            ADD 2 OVERHEAD
0006' 0221 0004
```

RECEIVE MSG TYPE #1 SUBROUTINE RMSG1.SRC

```
000A' 0205 0001*     53          LI     R5,SYDOCB      CPU#1 DOC DATA BUFFER START ADDRESS
000E' CD71           54  10$     MOV    *R1+,*R5+      MOVE ONE WORD AT A TIME
0010' 0600           55          DEC    R0             DECREMENT WORD COUNTER
0012' 15FD           56          JGT    10$            LOOP
                     57  *
0014' 04D3           58          CLR    *R3            MAKE RECEIVER MSG BUFFER AVAILABLE
                     59  *
                     60  *   SET MSG #1 RC OK BIT FOR CPU #2
                     61  *   WHICH WILL BE SET IN MSG #0
0016' E820 0004*     62          SOC    @B8,@TMA       SET MSG TYPE #1 REC OK BIT
001A' 0002*
                     63  *
001C' 045B           64          RT
                     65  ********************************************************************************
                     66          END
```

No errors detected

```
                    1          IDT    ICMD
                    2          SUBTTL  INTER CPU MSG DISPATCHER JOB
                    3  ********************************************************************************
                    4  *                                                                              *
                    5  * NAME: ICMD                          AUTH: N.COSTANTINIDES   *
                    6  * VERSION: 1                          DATE: 13-JUL-82              *
                    7  *                                                                              *
                    8  * FUNCTION:     INTER-CPU MESSAGE DISPATCHER JOB.                              *
                    9  *               THE ICMD IS CALLED BY THE EXTIVE WHEN THIS JOB IS              *
                   10  *               PLACED IN THE JOB QUEUE BY THE INTER-CPU RECEIVE               *
                   11  *               INTERRUPT.  THE ICMD CHECKS THE MESSAGE TYPE, IF               *
                   12  *               IT IS LEGAL, TRANSFERS CONTROL TO THE MESSAGE                  *
                   13  *               SUBROUTINE LISTED IN THE RECEIVE MESSAGE TYPE                  *
                   14  *               SUBROUTINE ADDRESS TABLE (ICMDTB).                             *
                   15  *                                                                              *
                   16  *                                                                              *
                   17  * CALLING MODULES: EXTIVE                                                     *
                   18  *                                                                              *
                   19  * CALLING SEQ: BLWP @R1                                                       *
                   20  *                                                                              *
                   21  * INPUTS: R0 = RECEIVE MESSAGE BUFFER START ADDRESS                           *
                       *                                                                              *
                       * OUTPUTS: CALLS MESSAGE TYPE SUBROUTINE (RMSG0 TO RMSG@)                     *
                   24  *                                                                              *
                   25  * MODULE REFERENCED: RMSG0 TO RMSG@                                           *
                   26  *                                                                              *
                   27  * WORKSPACE AREA: WPJB                                                        *
                   28  *                                                                              *
                   29  * REGISTERS MODIFIED: R0,R1,R2                                                *
                   30  *                                                                              *
                   31  * VERSION HISTORY:                                                            *
                   32  *                                                                              *
                   33  ********************************************************************************
      =0000        34          RSECT   ICMD
                   35  *** CALL NAME
                   36          INTERN  ICMD
                   37  *** TABLES REFERENCED
                   38          EXTERN  ICMDTB,ICMMAX
                   39  *** MODULES REFERENCED
```

```
                    41  ********************************************************************************
      =0000'        42  ICMD    EQU     $
0000' C05B          43          MOV     *R13,R1                 GET MSG BUF PTR FROM CALLER'S R0
0002' C091          44          MOV     *R1,R2                  GET 1ST WORD OF MSG
0004' 0242 00FF     45          ANDI    R2,>FF
0008' 8802 0002*    46          C       R2,@ICMMAX              ACTUAL MSG TYPE .VS. MAX TYPE
000C' 1203          47          JLE     10$                     JIF LEGAL MSG TYPE
                    48  *** ILLEGAL MESSAGE TYPE. JUST IN CASE IT IS ALREADY CAPTURED IN ICRC (ISR)
000E' 04D1          49          CLR     *R1             MAKE IC RC MSG BUFFER AVAILABLE
0010' 0580          50          INC     R0                      KEEP TRACK OF ERROR
0012' 1004          51          JMP     ICMEXT                  EXIT
                    52  *** LEGAL MESSAGE TYPE. CALL MSG TYPE SUBROUTINE.
0014' 0A12          53  10$     SLA     R2,1                    MSG TYPE * 2
0016' C022 0001*    54          MOV     @ICMDTB(R2),R0          GET MSG TYPE SUBR
001A' 0690          55          BL      *R0                     CALL MSG TYPE SUBR WITH R1=BUF PTR
001C' 0380          56  ICMEXT  RTWP                            RETURN TO EXECUTIVE
                    57  ********************************************************************************
                    58          END

No errors detected
```

```
                        1          IDT     ICBSEL
                        2          SUBTTL  INTER-CPU RECEIVE MSG BUFFER SELECTION
                        3  ********************************************************************************
                        4  *                                                                              *
                        5  * NAME: ICBSEL                  AUTH: N.COSTANTINIDES   *
                        6  * VERSION: 1                    DATE: 17 DEC 81                             *
                        7  *                                                                              *
                        8  * FUNCTION:     IT SELECTS THE AVAILABLE RECEIVE MESSAGE BUFFER FOR     *
                        9  *               THE NEXT MESSAGE. IF BUFFER IS NOT AVAILABLE IT         *
                       10  *               INCREMENTS THE 'BUFFER NOT AVAIALBLE' ERROR AND         *
                       11  *               SELECTS THE LAST BUFFER LISTED IN THE INTER-CPU RECEIVER*
                       12  *               ADDRESS TABLE (ICRBTB).                                 *
                       13  *               THE MSB OF FIRST WORD IN THE BUFFER INDICATES WHETHER   *
                       14  *               THE BUFFER IS AVAILABLE (LOGIC 0) OR NOT AVAILABLE      *
                       15  *               (LOGIC 1).                                              *
                       16  *                                                                       *
                       17  *                                                                       *
                       18  * CALLING MODULES:      ICRC    = INTER-CPU RECEIVE INTERRUPT          *
                       19  *                       ICRERR  = INTER-CPU RECEIVE ERROR              *
                       20  *                                                                       *
                       21  * CALLING SEQ: BL @ICBSEL                                              *
                       22  *                                                                       *
                       23  * INPUTS: NONE                                                         *
                       24  *                                                                       *
                       25  * OUTPUTS: ICRBSA = RECEIVE MSG BUFFER START ADDRESS                    *
                       26  *                                                                       *
                       27  * MODULE REFERENCED: NONE                                              *
                       28  *                                                                       *
                       29  * WORKSPACE AREA: WPIC                                                 *
                       30  *                                                                       *
                       31  * REGISTERS MODIFIED: R0,R1,R6                                         *
                       32  *                                                                       *
                       33  * VERSION HISTORY:                                                     *
                       34  *                                                                       *
                       35  ********************************************************************************
      =0000            36          RSECT   ICBSEL
                       37  *** CALL NAME
                       38          INTERN  ICBSEL
                       39  *** VARIABLES REFERENCED
                       40          EXTERN  ICRER9,ICRBSA
                       41  *** CONSTANTS REFERENCED
                       42          EXTERN  BIT0
                       43          EXTERN  D0
                       44  *** TABLES REFERENCED
                       45          EXTERN  ICRBTB
                       46  *** REGISTER DEFINITION
                       47  ********************************************************************************
      =0000'           48  ICBSEL  EQU     $
0000' 0201 0005*       49          LI      R1,ICRBTB               INTER-CPU RECEIVE BUFFER ADDRESS TABLE
0004' C031             50          MOV     *R1+,R0                 NUMBER OF BUFFER ADDRESSES
                       51  * FIND AVAILABLE RECEIVE MSG BUFFER
0006' C831 0002*       52  10$     MOV     *R1+,@ICRBSA            GET BUFFER START ADDRESS
```

```
000A' C1A0 0002$    53          MOV     @ICRBSA,R6
000E' 8816 0004$    54          C       $R6,@D0             MSB OF 1ST WORD = 1 FOR BUSY
0012' 1505          55          JGT     ICB10               JIF BUFFER AVILABLE (+ VALUE)
0014' 1304          56          JEQ     ICB10               JIF BUFFER AVAILABLE (+0)
0016' 0600          57          DEC     R0
0018' 15F6          58          JGT     10$                 LOOP
                    59  $ BUFFER NOT AVAIALBLE.
001A' 05A0 0001$    60          INC     @ICRER9             MSG BUFFER NOT AVAILABLE ERROR
                    61  $ BUFFER AVAILABLE
001E' C1A0 0002$    62  ICB10   MOV     @ICRBSA,R6
0022' E5A0 0003$    63          SOC     @BIT0,$R6           SET TO BUSY
                    64  $ EXIT
0026' 045B          65          RT                          RETURN
                    66  $
                    67          END
```

No errors detected

```
                     1          IDT     IDENT
                     2          SUBTTL  IDENTIFY AIRCRAFT TYPE
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: IDENT.SRC                        AUTH: N. CONSTANTINIDES *
                     6  * VERSION: 1                             DATE: 3-MAY-1983        *
                     7  *                                                                  *
                     8  * FUNCTION:     READS THE IDENT DISCRETES FROM THE DISCRETE      *
                     9  *               CHANNELS 82 AND 83. FOR P&W VERSION (CH 82)      *
                    10  *               IT SETS A WORD, SSEG TO 0. FOR GE VERSION        *
                    11  *               (CH 83) SETS SSEG TO 2.                          *
                    12  *                                                                  *
                    13  * CALLING MODULES: PON                                           *
                    14  *                                                                  *
                    15  * CALLING SEQ: BL @IDENT                                         *
                    16  *                                                                  *
                    17  * INPUTS:       NONE                                             *
                    18  *                                                                  *
                    19  * OUTPUTS:      SSEG WORD                                        *
                    20  *                                                                  *
                    21  * MODULES REFERENCED: DCDRQ                                      *
                    22  *                                                                  *
                    23  * WORKSPACE AREA: CALLER'S                                       *
                    24  *                                                                  *
                    25  * REGISTERS MODIFIED: R0,R1,R2,R3,R4,R5                          *
                    26  *                                                                  *
                    27  * VERSION HISTORY:
                    28  *                                                                  *
                    29  ********************************************************************
        =0000       30          RSECT   IDENT
                    31  *** CALL NAME
                    32          INTERN  IDENT
                    33  *** VARIABLES REFERENCED
                    34          EXTERN  SSEG
                    35  *** CONSTANTS REFERENCED
                    36          EXTERN  D2
                    37  *** TABLES REFERENCED
                    38  *** MODULES REFERENCED
                    39          EXTERN  DCDRQ
                    40  *** LIBRARY
                    41          INCLUDE ENCLOS
                    43  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    45          INCLUDE REGDEF          REGISTER DEFENITIONS
                    64          INCLUDE CNSTNT          CONSTANTS
                   192          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   493          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   723          INCLUDE JMPMAC          JUMP MACROS
                   757          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   772          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                  1148  *** REGISTERS DEFINITION
                  1149  ********************************************************************
    0000'         1150  IDENT
```

```
                    1151
0000' C80B 0000'    1152          MOV   R11,@LINKZ          SAVE LINK
0004' 04E0 0001*    1153          CLR   @SSEG               CLEAR WORD
0008' 04C2          1154          CLR   R2                  CLEAR TABLE POINTER
000A' 04C3          1155          CLR   R3                  CLEAR DATA REGISTER
                    1156  *
000C' C162 003C'    1157          MOV   @TAB(R2),R5         CHANNEL ADDRESS
0010' 0201 0726     1158          LI    R1,>0726
0014' 06A0 0003*    1159          BL    @DCDRQ              READ CHANNEL
0018' E0C4          1160          SOC   R4,R3               SET BIT IN R3
001A' 05C2          1161          INCT  R2                  INCREMENT POINTER
                    1162  *
001C' C162 003C'    1163          MOV   @TAB(R2),R5         ADDRESS OF CHANNEL 83
0020' 0201 0726     1164          LI    R1,>0726
0024' 06A0 0003*    1165          BL    @DCDRQ              READ CHANNEL
0028' 0A14          1166          SLA   R4,1                MOVE BIT TO 2ND POS
002A' E0C4          1167          SOC   R4,R3               SET BIT IN R3
                    1168  *
002C' 0283 0002     1169          CI    R3,2                CHECK RESULT
0030' 1102          1170          JLT   10$                 JIF <2 (PW)
0032' 05E0 0001*    1171          INCT  @SSEG
                    1172
0036' C2E0 0000'    1173  10$     MOV   @LINKZ,R11          RESTORE LINK
003A' 045B          1174          RT
                    1175
                    1176  ********************************************************************
003C' 4900          1177  TAB     DATA  >4900
003E' 4A00          1178          DATA  >4A00
0040' 4B00          1179          DATA  >4B00
                    1180
                    1181+         LOCR  PRIV,LINKZ
0000' =0002         1184A LINKZ   BSS   2
                    1185
                    1186          END
```

No errors detected

```
                          1          IDT     IOB2SC
                          2
                          3          SUBTTL  CONVERT IOB TO 2'S COMPLEMENT
                          4
                          5  *       CALLING SEQ:    CALL    @IOB2SC
                          6
                          7  *-------------------------------------------------------------+
                          8  *                                                             |
                          9  *       IOB2SC CONVERTS 14-BIT IOB DATA INTO 12-BIT 2'S COMPLEMENT.
                         10  *       VOLT            -10     -5      0       5       10
                         11  *       IOB (HEX)       3FFF    3000    2000    1000    0
                         12  *       2'SC (HEX)      2001    3000    0       1000    2000
                         13  *                                                             |
                         14  *-------------------------------------------------------------+
                         15  *       VERSION : 1
                         16  *       PROGRAMMED BY : N.CONSTANTINIDES
                         17  *       CHECKED BY : N.CONSTANTINIDES
                         18
                         19
                         20          INTERN  IOB2SC
                         21
                         22
        =0000            23  SCR     EQU     R0      =       SCRATCH
        =0004            24  VAL     EQU     R4      =       IOB DATA(INPUT); 12-BIT 2'S COMPL (OUTPUT)
                         25
                         26
        =0000           157          RSECT   IOB2SC
                        158  ********************************************************************************
    0000'               159  IOB2SC
    0000'  0A24         160          SLA     VAL,C2          14-BIT LEFT-JUSTIF (WORK WITH 14 BIT DATA)
    0002'  0504         161          NEG     VAL             VAL = - VAL
    0004'  0224 B000    162          AI      VAL,CB000H      VAL = VAL + 2000(H4) (CARRY AWAY)
    0008'  0824         163          SRA     VAL,C2          REGAIN 16-BIT/2 DATA
    000A'  045B         164          RT
                        165  ********************************************************************************
                        166
                        167          END
```

```
                        1          IDT     JBQUE
                        2          SUBTTL  JOB QUEUE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: JBQUE                          AUTH: M.COSTANTINIDES     *
                        6  * VERSION: 1                           DATE: 23-OCT-81           *
                        7  *                                                                  *
                        8  * FUNCTION:     IT QUEUES THE BACKGROUND JOB IN THE JOB BUFFER   *
                        9  *               (JBBUF).  IT PLACES THE JOB NUMBER AND THE       *
                       10  *               BUFFER START ADDRESS IN THE QUEUE.  INPUT QUEUE  *
                       11  *               POINTER(JBIPTR) IS INCREMENTED BY 4 BYTES.       *
                       12  *               THE CALLER'S PC IS INCREMENTED BY 2 BYTES TO     *
                       13  *               SKIP OVER THE 'DATA' STATEMENT. JOB ARE          *
                       14  *               LISTED IN THE JOB LIST TABLE(JBTBL).             *
                       15  *               THE QUEUED JOB WILL BE EXECUTED BY THE EXTIVE.   *
                       16  *                                                                  *
                       17  * CALLING MODULES: REAL TIME ISR, INTER-CPU RECEIVER ISR,         *
                       18  *                                                                  *
                       19  * CALLING SEQ:  BLWP @JBQUE     JOB QUEUE                      *
                       20  *               DATA    N       JOB NUMBER (0 TO N)            *
                       21  *                                                                  *
                       22  * INPUT:        R0=BUFFER START ADDRESS                          *
                       23  *                                                                  *
                       24  * OUTPUT:       JOB NUMBER AND BUFFER POINTER IN JOBBUF BUFFER   *
                       25  *                                                                  *
                       26  * MODULES REFERENCED: NONE                                       *
                       27  *                                                                  *
                       28  * WORKSPACE AREA: WPEX                                           *
                       29  *                                                                  *
                       30  * REGISTERS MODIFIED: R8,R14                                     *
                       31  *                                                                  *
                       32  ********************************************************************
        =0000          33         RSECT   JBQUE
                       34  *** CALL NAME
                       35         INTERN  JBQUE
                       36  *** VARIABLES REFERENCED
                       37         EXTERN  JBBBUF,JBEBUF
                       38         EXTERN  WPEX
                       39  *** REGISTER DEFINITION
        =0008          40  JBIPTR  EQU     R8              INPUT POINTER FOR JBBUF
        =0009          41  JBOPTR  EQU     R9              OUTPUT POINTER FOR JBBUF
        =000D          42  WP      EQU     R13             WORKSPACE POINTER
                       43  ********************************************************************
  0000' 0003*          44  JBQUE   DATA    WPEX            WORKSPACE AREA
  0002' 0004'          45         DATA    JB
                       46  *
  0004' 0300 0001      47  JB      LIMI    1               DISABLE ALL INTERUPTS
  0008' 0288 0002*     48         CI      JBIPTR,JBEBUF   INPUT POINTER .VS. END OF JOB BUFFER
  000C' 1A02           49         JL      20$             JIF NOT AT THE END
  000E' 0208 0001*     50         LI      JBIPTR,JBBBUF   SET TO BEG OF JOB BUFFER
                       51  *
  0012' CE3E           52  20$     MOV     *R14+,*JBIPTR+  PLACE JOB NUMBER IN QUEUE,BUMP CALLER'S PC.
```

```
0014'  CE1D    53          MOV    *R13,@JBIPTR+  PLACE BUFFER START ADDRESS
               54  *                             CALLER'S R0 IN QUEUE.
0016'  0380    55          RTWP                  RETURN
               56  *
               57          END
```

No errors detected

```
                    1          IDT     JOB8PS
                    2          SUBTTL  BACKGROUND JOB
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: JOB8PS.SRC                      AUTH: N.COSTANTINIDES     *
                    6  * VERSION: 1                            DATE: 5-APRIL-1982        *
                    7  *                                                                  *
                    8  * FUNCTION:THIS JOB IS PERFORMED 8 TIMES PER SECOND, INITIATED     *
                    9  *          BY THE REAL TIME ISR.                                   *
                   10  *          THE FOLLOWING JOBS ARE PERFORMED:                       *
                   11  *               1. DMX #1 & #2 DISCRETE ACQUISITION                *
                   12  *                  AND OUTPUT VIA DESTINATION TABLE                *
                   13  *                                                                  *
                   14  * CALLING MODULES: BACKGROUND                                      *
                   15  *                                                                  *
                   16  * CALLING SEQ: BL @JOB8PS                                         *
                   17  *                                                                  *
                   18  * INPUTS:                                                         *
                   19  *                                                                  *
                   20  * OUTPUTS:                                                        *
                   21  *                                                                  *
                   22  * MODULES REFERENCED:                                             *
                   23  *                                                                  *
                   24  * WORKSPACE AREA:                                                 *
                   25  *                                                                  *
                   26  * REGISTERS MODIFIED:                                             *
                   27  *                                                                  *
                   28  * VERSION HISTORY:                                                *
                   29  *                                                                  *
                   30  ********************************************************************
=0000              31         RSECT   JOB8PS
                   32  ### CALL NAME
                   33         INTERN  JOB8PS
                   34  ### VARIABLES REFERENCED
                   35  ### CONSTANTS REFERENCED
                   36  ### TABLES REFERENCED
                   37  ### MODULES REFERENCED
                   38         EXTERN  DC1ACQ
                   39  ### LIBRARY
                   40         INCLUDE ENCLOS
                   42  ###    ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   44         INCLUDE REGDEF          REGISTER DEFENITIONS
                   63         INCLUDE CNSTNT          CONSTANTS
                  191         INCLUDE SUBMAC          FUNCTIONAL MACROS
                  492         INCLUDE MSCMAC          MISCELLANEOUS MACROS
                  722         INCLUDE JMPMAC          JUMP MACROS
                  756         INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                  771         INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                 1147  ### REGISTERS DEFINITION
```

```
                    1149  ********************************************************************************
                    1150  *
0000'               1151  JOB8PS
                    1152  *
                    1153  *** DMX #1 DISCRETE ACQUISITION AND OUTPUT
                    1154  *
                    1155+         CALLXD  DC1ACQ
0000' 2CC0          1161A         XOP     R0,3
                    1167  *
0002' 0380          1168          RTWP
                    1169          END
```

No errors detected

```
                    1          IDT     JOB4PS
                    2          SUBTTL  BACKGROUND JOB
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: JOB4PS.SRC                      AUTH: N.COSTANTINIDES     *
                    6  * VERSION: 2                            DATE: 27-MAY-1983         *
                    7  *                                                                  *
                    8  * FUNCTION:BACKGROUND JOB CALLED BY EXECUTIVE                      *
                    9  *          THIS JOB IS PERFORMED 4 TIMES PER SECOND, INITIATED     *
                   10  *          BY THE REAL TIME ISR.                                   *
                   11  *          THE FOLLOWING JOBS ARE PERFORMED:                       *
                   12  *               1. MEMORY SUM CHECK ONCE EVERY SEC                 *
                   13  *                  ( 1 KWORD EVERY SEC )                           *
                   14  *               2. MONITORS EAROM TASK FOUR PER SEC (EAMON)        *
                   15  *                  MONITORS ERASE/WRITE PROCESS                    *
                   16  *                  INITIATE WRITE                                  *
                   17  *               3. MONITORS SYSTEM STATUS 1 PER SEC (SYSMON)       *
                   18  *                  CONVERTS ERROR BIT IN SYEBF OF BUFFER TO        *
                   19  *                       ERROR CODE WHICH WILL BE STORED IN         *
                   20  *                       TEMPORARY EAROM STATUSBUFFER AND SUPER-     *
                   21  *                       FRAME STATUS BUFFER                        *
                   22  *               4. PANEL LED DISPLAY & LAMP CONTROL 1 PER SEC      *
                   23  *                       (SYSOUT)                                   *
                   24  *                  WHEN READ SWITCH IS DEPRESSED IT CONTROLS       *
                   25  *                       LED DISPLAY & LAMPS                        *
                   26  *               5. EAROM ERASE DISCRETE TEST                       *
                   27  *                                                                  *
                   28  * CALLING MODULES: EXTIVE                                         *
                   29  *                                                                  *
                   30  * CALLING SEQ: BLWP @JOB4PS                                       *
                   31  *                                                                  *
                   32  * INPUTS: JB4CYL = JOB 4 PER SEC COUNTER (0,1,2,3)                *
                   33  *                                                                  *
                   34  * OUTPUTS:                                                        *
                   35  *       JB4CYL = JOB 4 PER SEC COUNTER                            *
                   36  *                                                                  *
                   37  * MODULES REFERENCED:                                             *
                   38  *               EAMON   =       EAROM MONITOR                     *
                   39  *               SYSMON  =       SYSTEM STATUS MONITOR             *
                   40  *               SYSOUT  =       LAMPS & DISPLAY                   *
                   41  *               MSCHK   =       MEMORY SUM CHECK                  *
                   42  *               EABTST  =       EAROM ERASE BIT TEST              *
                   43  *                                                                  *
                   44  * WORKSPACE AREA: WPJB                                            *
                   45  *                                                                  *
                   46  * REGISTERS MODIFIED: R9                                          *
                   47  *                                                                  *
                   48  * VERSION HISTORY:                                                *
                   49  *       VERSION 2 ; SCR 000050                                    *
                   50  *                                                                  *
                   51  ********************************************************************
=0000              52         RSECT   JOB4PS
```

```
                     53  ### CALL NAME
                     54          INTERN  JOB4PS
                     55  ### VARIABLES REFERENCED
                     56  ### CONSTANTS REFERENCED
                     57  ### TABLES REFERENCED
                     58  ### MODULES REFERENCED
                     59          EXTERN  DFBTST
                     60          EXTERN  EAMON,EABTST
                     61          EXTERN  MSCHK,SUPBFX
                     62          EXTERN  SYSMON,SYSOUT
                     63  ### LIBRARY
                     64          INCLUDE ENCLOS
                     66  ###     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                     68          INCLUDE REGDEF          REGISTER DEFENITIONS
                     87          INCLUDE CNSTNT          CONSTANTS
                    215          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    516          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    746          INCLUDE JMPMAC          JUMP MACROS
                    780          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    795          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1171  ### REGISTERS DEFINITION
=0009              1172  JB4CYL  EQU     R7              JOB %'S COUNTER, INCREMENTED EVERY JOB4PS CALL  (0 TO 3)
=000A              1173  RSCTR   EQU     R10             READ SWITCH COUNTER (SYSOUT)
```

```
                    1175  ********************************************************************
                    1176  *
                    1177+ JOB4PS  ENABLE  C17            RESET INTEGRATOR EVERY 250 MSEC
0000' 020C 0700     1178A         LI      CRU,C9901
0004' 1D 11         1179A         SBO     C17
0006' 1E 11         1180          SBZ     C17
                    1181  *** INCREMENT FROM CYCLE COUNTER EVERY CALL
0008' 0589          1182          INC     JB4CYL
000A' 0249 0003     1183          ANDI    JB4CYL,3        COUNTER 0 TO 3 ONLY
                    1184  ********************************************************************
                    1185  *** EAROM MONITOR, 4 PER SEC
000E' 0420 0002*    1186          BLWP    @EAMON
0012' 06A0 0005*    1187          BL      @SUPBFX
                    1188  ***
0016' 0289 0001     1189          CI      JB4CYL,1
001A' 1606          1190          JNE     20$             JIF NOT SECOND CYCLE
                    1191  ********************************************************************
                    1192  *   2ND CYCLE OF 4 CYCLE
                    1193  *** SYSTEM MONITOR, 1 PER SEC
001C' 06A0 0006*    1194          BL      @SYSMON
                    1195  *** PANEL LED DISPLAY AND LAMPS, 1 PER SEC,
0020' 06A0 0007*    1196          BL      @SYSOUT
0024' 06A0 0001*    1197          BL      @DFBTST         DFDR BITE
                    1198  ***
0028' 0289 0002     1199  20$     CI      JB4CYL,2
002C' 1602          1200          JNE     30$             JIF NOT 3RD CYCLE
                    1201  ********************************************************************
                    1202  *   3RD CYCLE OF 4 CYCLE
                    1203  *** DFDR PLAYBACK SYNC CODE SEARCH COMMAND
002E' 06A0 0003*    1204          BL      @EABTST         TEST EAROM ERASE DISCRETE
0032' 0289 0003     1205  30$     CI      JB4CYL,3
0036' 1602          1206          JNE     JOB4EX          JIF NOT 4TH CYCLE
                    1207  ********************************************************************
                    1208  *   4TH CYCLE OF 4 CYCLE
                    1209  *** MEMORY SUM CHECK, 1 PER SEC
                    1210  *
0038' 06A0 0004*    1211          BL      @MSCHK
003C' 0380          1212  JOB4EX  RTWP
                    1213  ********************************************************************
                    1214          END
```

No errors detected

```
                        1          IDT     MSCHK
                        2
                        3          SUBTTL  MEMORY CHECK-SUM
                        4
                        5  *       CALLING SEQ:    CALL    @MSCHK
                        6
                        7  *---------------------------------------------------------------------+
                        8  *                                                                      !
                        9  *       MSCHK TEST EPROM (CHECK-SUM) IN THE BACKGROUND.
                       10  *       CALLING MODULE: JOB4PS
                       11  *                                                                      !
                       12  *---------------------------------------------------------------------+
                       13  *       VERSION : 1
                       14  *       PROGRAMED BY : N.CONSTANTINIDES
                       15  *       CHECKED BY   : N.CONSTANTINIDES
                       16
                       17          INTERN  MSCHK
                       18  *** VARIABLES REFERENCED:
                       19          EXTERN  MSIX,MSERR
                       20  *** CONSTANTS REFERENCED
                       21          EXTERN  BO
                       22  * REFERD MODULES:
                       23          EXTERN  MSSBR
                       24          EXTERN  SYSER,SYSOK
                       25  *** TABLES REFERENCE:
                       26          EXTERN  MEND,MSBUF
        =0000          27  SUM     EQU     R0      =       SCRATCH
        =0001          28  CNT     EQU     R1      =       SCRATCH
        =0005          29  MSP     EQU     R5      =       SCRATCH
                       30  *               R2,R3,R4        =       SCRATCH
                       31
                       32          INCLUDE ENCLOS
                       34  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       36          INCLUDE REGDEF          REGISTER DEFENITIONS
                       55          INCLUDE CNSTNT          CONSTANTS
                      188          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      489          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      719          INCLUDE JMPMAC          JUMP MACROS
                      753          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      769          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1144          RSECT   MSCHK
                     1145  ********************************************************************************
  0000  C0CB          1146  MSCHK   MOV     LINK,R3         SAVE LINK REG
  0002  C160 0001*    1147          MOV     @MSIX,MSP       MEMORY SUM INDEX(0,2,4,...)
  0006  0201 0800     1148          LI      CNT,>800        1 KWORD CONSTANT
  000A  8805 0007*    1149          C       MSP,@MEND       MEMORY SUM LIMIT
  000E  1A02          1150          JL      2$
  0010  1304          1151          JEQ     3$              JUMP IF LAST K WORD SECTION
  0012  04C5          1152          CLR     MSP             RESET  TO START FROM 1ST KWORD
                      1153  *     1ST THRU ONE BEFORE LAST K WORD BLOCK
  0014                1154  2$
                      1155+         CALL    MSSBR           SUM 1 KWORD BLOCK. RESULT IN R0 (SUM)
```

```
00[illegible]  06A0 0004*  1162A          BL    @MSSBR
0018'  1007        1164          JMP    5$            CHEK RESULT
                   1165  *     LAST BLOCK
0014'              1166  3$
0016'  C095        1167          MOV    MSP,R2        POINTER
0018'  0AA2        1168          SLA    R2,10         * 1K
001A'  0201 0007*  1169          LI     CNT,MEND      LAST WORD TO BE SUM
001E'  6042        1170          S      R2,CNT        NO. OF WORDS TO BE SUMMED
                   1171+         CALL   MSSBR         SUM THE REST OF WORDS
0020'  06A0 0004*  1178A          BL    @MSSBR
0024'              1180  5$
0026'  A025 0008*  1181          A      @MSBUF(MSP),SUM
0028'  1303        1182          JEQ    7$            JUMP IF SUM OK
                   1183  *  FAIL CHECK SUM TEST
002A'  E825 0003*  1184          SOC    @B0(MSP),@MSERR       SET ERROR BIT FOR THIS BLOCK
002E'  0002*
0030'  C050 0000'  1185          MOV    @ERCOD,R1             ERROR CODE
0034'  06A0 0005*  1186          BL     @SYSER                SET ERROR BIT IN SYEBF
0038'  100A        1187          JMP    10$
                   1188  *  PASS CHECK SUM FOR THIS BLOCK
003A'  4825 0003*  1189  7$      SZC    @B0(MSP),@MSERR RESET ERROR BIT FOR THIS BLOCK
003E'  0002*
0040'  C020 0002*  1190          MOV    @MSERR,R0
0044'  1604        1191          JNE    10$           JIF SOME BLOCK IS FAILING
                   1192  *  PASS MEMORY CHECK SUM FOR ALL BLOCKS
0046'  C060 0000'  1193          MOV    @ERCOD,R1             ERROR CODE
004A'  06A0 0006*  1194          BL     @SYSOK        RESET ERROR BIT IN SYEBF
004E'  05C5        1195  10$     INCT   MSP           BUMP POINTER FOR NEXT ITERATION
0050'  C805 0001*  1196          MOV    MSP,@MSIX     RESTORE MEMORY SUM INDEX
0054'  C2C3        1197          MOV    R3,LINK       UNSAVE LINK REG
0056'  045B        1198          RT
                   1199  ********************************************************************************
                   1200+         PRVDAT
0058'  0101        1202  ERCOD   DATA   >101           CHECK-SUM ERROR CODE
                   1203          END
```

```
                     1          IDT     MSGEN
                     2
                     3          SUBTTL MEMORY SUM WORD GENERATION
                     4
                     5  *       CALLING SEQ:    CALL    @MSGEN
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   !
                     9  *       GENMS GENERATES MEMORY SUM WORDS FOR EVERY 1K WORD BLOCK
                    10  *       AND STORE THEM IN MEMORY SUM BUFFER (MSBUF) [IN EPROM].
                    11  *       IT IS USED ONLY AT SYSTEM INTEGRATION LEVEL !!
                    12  *                                                                   !
                    13  * CALLING MODULE: PON (POWER ON ISR)                               !
                    14  *-------------------------------------------------------------------+
                    15  *       VERSION : 2
                    16  *       PROGRAMED BY : N.CONSTANTINIDES
                    17  *       CHECKED BY : N.CONSTANTINIDES
                    18
                    19          INTERN  MSGEN
                    20  * REFERD MODULES:
                    21          EXTERN  MSSBR
                    22  * GLOBALS:
                    23  *       (ROM)
                    24          EXTERN  MEND            ,2 EPROM END (START OF CHECK-SUM TABLE)
                    25  *                               ALSO CHECK-SUM LIMIT
                    26          EXTERN  MSBUF           CHEK-SUM-TABLE
                    27  *       (RAM)
                    28
=0000               29  VAL     EQU     R0      =       SCRATCH
=0001               30  CNT     EQU     R1      =       SCRATCH
=0002               31  TMP     EQU     R2      =       SCRATCH
=0004               32  CLM     EQU     R4      =       SCRATCH
=0005               33  CSP     EQU     R5      =       SCRATCH
                    34  *
                    35          INCLUDE ENCLOS
                    37  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    39          INCLUDE REGDEF          REGISTER DEFENITIONS
                    58          INCLUDE CNSTNT          CONSTANTS
                   191          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   492          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   722          INCLUDE JMPMAC          JUMP MACROS
                   756          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   771          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000             1147          RSECT   MSGEN
                  1148  ********************************************************************************
```

```
0000'                1150  MSGEN
0000'  C0CB          1151        MOV     LINK,R3         SAVE LINKER
0002'  04C5          1152        CLR     CSP             BLOCK-OF-1000-WORDS OFFSET = 0.
0004'  0204 0002*    1153        LI      CLM,MEND
0008'  09B4          1154        SRL     CLM,11          COMPUTE MEMORY SUM LIMIT
000A'  0A14          1155        SLA     CLM,1
000C'  C804 0002*    1156        MOV     CLM,@MEND       SAVE LIMIT FOR CHKSM
0010'                1157  2$
0010'  0201 0800     1158        LI      CNT,>800              1 KWORD CONSTANT
0014'                1159  4$
0014'  06A0 0001*    1160        BL      @MSSBR          SUM IN VAL
0018'  0500          1161        NEG     VAL             TWO'S COMPLEMENT THE SUM
001A'  C940 0003*    1162        MOV     VAL,@MSBUF(CSP) STORE IT
001E'  05C5          1163        INCT    CSP             FOR NEXT BLOCK
0020'  8805 0002*    1164        C       CSP,@MEND
0024'  1AF5          1165        JL      2$              LOOP
0026'  1606          1166        JNE     9$              EXIT
                     1167  *     LAST BLOCK
002B'  C085          1168        MOV     CSP,TMP
002A'  0AA2          1169        SLA     TMP,10          * 1K
002C'  0201 0002*    1170        LI      CNT,MEND              LAST WORD TO BE SUMMED
0030'  6042          1171        S       TMP,CNT         NO. OF WORDS TO BE SUMMED
0032'  10F0          1172        JMP     4$
0034'                1173  9$
0034'  C2C3          1174        MOV     R3,LINK         RESTORE LINKER
0036'  045B          1175        RT
                     1176  ********************************************************************************
                     1177        END
```

```
                        1          IDT     PIDO
                        2
                        3          SUBTTL  INITIALIZE DATA OUTPUT
                        4
                        5  *       CALLING SEQ:    CALL    @PIDO
                        6
                        7  *---------------------------------------------------------------*
                        8  *                                                               !
                        9  *       PIDO SETS ALL OUTPUT POINTERS (TO THE START OF LAST CYCLE
                       10  *       IN OUTPUT-BUFFERS), CYCLE-, SUBFRAME- AND FRAME-COUNT,
                       11  *       ACTIVATES AND STARTS ALL (DFDR, AUX AND 429) OUTPUTS,
                       12  *       SENDS SYNC-CODE TO PLAY-BACK AND ENABLE PLAY-BACK SYNC
                       13  *       INTERRUPT.
                       14  *                                                               !
                       15  *---------------------------------------------------------------*
                       16  *       VERSION : 1
                       17  *       PROGRAMMED BY : M.CONSTANTINIDES
                       18  *       CHECKED BY : M.CONSTANTINIDES
                       19
                       20          INTERN  PIDO
                       21  * GLOBAL AREA:
                       22          EXTERN  DFWRC           DFDR WORD COUNT (POINTER) (SCOTT)
                       23
        =0000          24  SCR     EQU     R0      =       SCRATCH
                       25
                       31          INCLUDE REGDEF          REGISTER DEFENITIONS
                       50          INCLUDE CNSTNT          CONSTANTS
                      178          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      479          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      709          INCLUDE JMPMAC          JUMP MACROS
                      743          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      758          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1135          RSECT   PIDO
                     1136  ********************************************************************************
    0000'            1137  PIDO
                     1138+         MOVI    C55*2,DFWRC             SET DF-WR-COUNT TO LAST CYCLE
    0000'  0200 006E 1140A         LI      R0,C55*2
    0004'  C800 0001* 1141A         MOV     R0,@DFWRC
                     1143  * ACTIVATE AUX
                     1144
                     1145  * ACTIVATE 429
                     1146
                     1147  * ACTIVATE DFDR PLAY-BACK SYNC INT
                     1148  * ACTIVATE DFDR OUTPUT INTERRUPT
                     1149+         ENABLE  5,                      ENABLE DFDR INT VIA 9901
    0008'  020C 0700 1150A         LI      CRU,C9901
    000C'  1D 05     1151A         SBO     5
    000E'  045B      1152          RT
                     1153  ********************************************************************************
                     1154          END
```

```
                        1          IDT     NICHTS
                        2
                        3          SUBTTL   NOTHING
                        4
                        5  *       CALLING SEQ:    CALLWP  @NICHTS
                        6  *                       (OR UNUSED INTERRUPT)
                        7
                        8  *-------------------------------------------------------------------+
                        9  *                                                                   !
                       10  *       NICHTS SERV (SPECIALY) ALL INTERRUPTS WHICH ARE
                       11  *       NOT USED SO THAT IN THE CASE OF OCCURENCE OF AN
                       12  *       UNDESIRED INTERRUPT, IT DOES NOT GET LOST.
                       13  *                                                                   !
                       14  *-------------------------------------------------------------------+
                       15  *       VERSION : 1
                       16  *       PROGRAMMED BY : M.CONSTANTINIDES
                       17  *       CHECKED BY: M.CONSTANTINIDES
                       18
                       19          INTERN  NICHTS
                       20          EXTERN  RTSIN1
                       21
          =0000        22          RSECT   NICHTS
                       23
                       24  ********************************************************************
                       25  *
      0000'            26  NICHTS
                       27
      0000'  06A0 0001* 28          BL      @RTSIN1
      0004'  0380       29          RTWP                    RETURN
                       30
                       31  ********************************************************************
                       32
                       33          END
```

```
                     1          IDT    PIHW
                     2
                     3          SUBTTL  INITIALIZE H/W
                     4
                     5  *       CALLING SEQ:    CALL    @PIHW
                     6
                     7  *-------------------------------------------------------------------+
                     8  *                                                                   |
                     9  *       PIHW EXTINGUISHES FRONT PANEL 'DFDR FAIL', 'CAUTION'
                    10  *       AND 'FAIL' LAMPS, SET FRONT-PANEL DISPLAY TO '0' AND
                    11  *       DETERMINES THE ENGINE TYPE.
                    12  *                                                                   |
                    13  *-------------------------------------------------------------------+
                    14  *       VERSION : 1
                    15  *       PROGRAMMED BY : M.CONSTANTINIDES
                    16  *       CHECKED BY : M.CONSTANTINIDES
                    17
                    18
                    19          INTERN  PIHW
                    20  * REFERD MODULES:
                    21          EXTERN  DCDRQ
                    22
                    23  * GLOBALS :
                    24          EXTERN  DO      = 0 (ROM)
                    25
=0000               26  CNT     EQU     R0      =       SCRATCH
=0001               27  PTR     EQU     R1      =       SCRATCH
=0004               28  DDT     EQU     R4      =       SCRATCH
                    29  *               R5      =       SCRATCH
=000A               30  VSU     EQU     R10     =       VERSION NO (OUTPUT)
                    31
                    32          INCLUDE ENCLOS
                    34  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    36          INCLUDE REGDEF          REGISTER DEFENITIONS
                    55          INCLUDE CNSTNT          CONSTANTS
                   183          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   484          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   714          INCLUDE JMPMAC          JUMP MACROS
                   748          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   763          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
=0000              1139          RSECT   PIHW
                   1140  ********************************************************************************
```

```
0000'                1142  PIHW
0000'  C80B 0000'    1143        MOV     LINK,@LINKZ          SAVE LINKER
                     1144+       XTOFPL                       EXTINQUISH FRONT-PANNEL LAMPS, FP=0
0004'  020C 05E0     1145A       LI      CRU,CRUFPL           GET FRONT PANEL LAMPS CRU ADDR
0008'  1E 00         1146A       SBZ     ZERO                 EXTINQ 'DFDR FAIL'
000A'  1E 01         1147A       SBZ     C1                   EXTINQ 'FAIL'
000C'  1E 02         1148A       SBZ     C2                   EXTINQ 'CAUTION'
000E'  C820 0002$    1149A       MOV     @D0,@CFPDSP          FRONT-PANEL DISPLAY =  0
0012'  FF80
0014'  1E 03         1150        SBZ     C3                   DISABLE U-CIRCUTE (LOW FOR EVER !!)
0016'  C2E0 0000'    1151        MOV     @LINKZ,LINK          RESTORE LINKER
001A'  045B          1152        RT
                     1153  ********************************************************************************
                     1154  * PRICVATE AREA
                     1155+       LOCR    PRIV,LINKZ           LINK-SAVE AREA
0000'  =0002        1158A LINKZ   BSS     2
                     1159  *                                  TO DETERMINE ENGINE TYPE
                     1160        END
```

```
                      1          IDT     POWDN
                      2          SUBTTL  POWER DOWN INTERRUPT SERVICE ROUTINE
                      3  ********************************************************************
                      4  *                                                                  *
                      5  * NAME: POWDN                          AUTH: N.COSTANTINIDES     *
                      6  * VERSION:1                            DATE: 9 APR 82
                      7  *                                                                  *
                      8  * FUNCTION: POWER DOWN INTERRUPT SERVICE ROUTINE.
                      9  *              THIS MODULE WILL BE EXECUTED WHEN POWER IS
                     10  *              LOST.
                     11  *                                                                  *
                     12  * CALLING MODULES: INTERRUPT #1                                    *
                     13  *                                                                  *
                     14  * CALLING SEQ:
                     15  *                                                                  *
                     16  * INPUTS:
                     17  *                                                                  *
                     18  * OUTPUTS:
                     19  *                                                                  *
                     20  * MODULES REFERENCED:
                     21  *                                                                  *
                     22  * WORKSPACE AREA:
                     23  *                                                                  *
                     24  * REGISTERS MODIFIED:
                     25  *                                                                  *
                     26  * VERSION HISTORY:
                     27  *                                                                  *
                     28  ********************************************************************
       =0000         29          RSECT   POWDN
                     30  *** CALL NAME
                     31          INTERN  POWDN
                     32  *** VARIABLES REFERENCED
                     33  *** CONSTANTS REFERENCED
                     34  *** TABLES REFERENCED
                     35  *** MODULES REFERENCED
                     36  *** LIBRARY
                     37          INCLUDE REGDEF          REGISTER DEFINITIONS
                     56          INCLUDE CNSTNT          CONSTANTS
                    184  *** REGISTERS DEFINITION
                    185  ********************************************************************
0000' 058B           186  POWDN   INC     R11             JUST FOR CHECKOUT
0002' 020C 0700      187          LI      CRU,C9901       SET CRU REG WITH 9901 ADDRESS
0006' 1E 10          188  HOLD    SBZ     16              HOLD MICRO PROCESSOR
0008' 10FE           189          JMP     HOLD            LOOP FOR EVER UNTIL NO MORE POWER
                    190  ********************************************************************
                    191          END
```

No errors detected

```
                        1
                        2          IDT     PON
                        3
                        4          SUBTTL  POWER ON
                        5
                        6  *       CALLING SEQ:    POWER-ON INTERRUPT (INT 0)
                        7
                        8  *-------------------------------------------------------------+
                        9  *                                                             |
                       10  *       PON OPERATS ON POWER-ON ENTRY.
                       11  *       IT INITIALIZES H/W, S/W, INTERRUPTS, DATA-IN AND -OUT AND
                       12  *       DOES SOME POWER ON DIAGNOSTIC.
                       13  *                                                             |
                       14  *       VERSION HISTORY:                                        |
                       15  *                                                               |
                       16  *-------------------------------------------------------------+
                       17  *       CALLING MODULE: H/W
                       18  *       VERSION : 2
                       19  *       PROGRAMMED BY : M.CONSTANTINIDES
                       20  *       MODIFIED : 26-MAY-1983 M.CONSTANTINIDES
                       22          INTERN  PON
                       23  * VARIABLES REFERENCED
                       24          EXTERN  FRCNT,CYPFRC
                       25          EXTERN  RAMBEG,RAMEND,POTRAM
                       26  * REFERD MODULES:
                       27          EXTERN  EXTIVE
                       28          EXTERN  ICINIT
                       29          EXTERN  MSGEM
                       30          EXTERN  PIHW,PISW,PINT
                       31          EXTERN  PIDA,PIDO
                       32          EXTERN  IDENT,RTSDL1
                       33  *** REGISTER DEFINITION:
        =0007          34  PINIT   EQU     R7      ANY POWER INTERRUPT INIT FLAG (RITFR)
        =0008          35  POCALF  EQU     R8      POWER-ON-CALIBRATION-FLAG
        =0009          36  POR9    EQU     R9
        =000A          37  VRSMSK  EQU     R10     VERSION NUMBER
                       38
                       39  *** LIBRARY
                       40          INCLUDE ENCLOS
                       42  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                       44          INCLUDE REGDEF          REGISTER DEFENITIONS
                       63          INCLUDE CNSTNT          CONSTANTS
                      191          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      492          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      722          INCLUDE JMPMAC          JUMP MACROS
                      756          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      771          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
        =0000        1147          RSECT   PON
                     1148  ********************************************************************************
    0000'            1149  PON
    0000'  06A0 000F*  1150          BL      @RTSDL1         RTS
```

```
0004' 0300 0000  1151          LIMI   ZERO          DISABLE ALL INTERRUPTS
0008' 0200 0002  1152          LI     R0,2          SET COUNTER TO 2 INITIAL VALUE
000C' C800 0000' 1153          MOV    R0,@CNT
                 1154
                 1155  ***
                 1156  *** CHECK POWER TRANSIENT OR TRUE POWER ON (SWITCH TO ON) INTERRUPT
                 1157  *** RAM IS PROTECTED FOR THE 1ST 200 MSEC OF POWER OFF PERIOD.
                 1158  *** PREDEFINED PATTERN WILL BE STORED IN THE RESERVED RAM(HEX LOC D000).
                 1159  *** EVERY POWER ON INTERRUPT THE PATTERN IN THE RAM WILL BE CHECKED
                 1160  *** IF PATTERN IS VALID, IT IS DEFINED AS A POWER TRANSIENT.
                 1161  *** IF PATTERN IS INVALID, THEN IT IS A TRUE POWER ON TYPE.
                 1162
0010' 0200 0003  1163          LI     R0,3          NUMBER OF PATTERN
0014' 0201 0094' 1164          LI     R1,POTPAT     PATTERN START ADDRESS
0018' 0202 0005* 1165          LI     R2,POTRAM     RAM START ADDRESS
001C' 8CB1       1166  20$     C      *R1+,*R2+
001E' 160F       1167          JNE    22$           JIF RAM IS BAD (TRUE POWER ON)
0020' 0600       1168          DEC    R0
0022' 15FC       1169          JGT    20$           LOOP BACK FOR MORE CHECK
                 1170  *
                 1171  *  RAM IS GOOD
                 1172  *  CORRECT FRAME/SUBFRAME COUNTER TO START AT NEXT SUBFRAME
                 1173  *
0024' C0A0 0001* 1174  99$     MOV    @FRCNT,R2     SAVE FRAME/SUBFRAME COUNTER
0028' 0201 0003* 1175          LI     R1,RAMBEG     START ADDR OF RAM EXCLUDING POWER ON WP
002C' 04F1       1176  21$     CLR    *R1+          CLEAR RAM WORD
002E' 0281 0004* 1177          CI     R1,RAMEND     END OF RAM ?
0032' 1AFC       1178          JL     21$           LOOP IF NOT
0034' 0720 0002* 1179          SETO   @CYPFRC       RESET CYCLE PER FRAME COUNTER
0038' C802 0001* 1180          MOV    R2,@FRCNT     RESTORE FRAME/SUBFRAME COUNTER
003C' 1012       1181          JMP    60$           RAM IS GOOD (POWER TRANSIENT)
                 1182  *
                 1183  *  RAM IS BAD. RESTORE PATTERN
                 1184  *
003E' 0201 0094' 1185  22$     LI     R1,POTPAT     PATTERN START ADDRESS
0042' 0202 0005* 1186          LI     R2,POTRAM     RAM START ADDRESS
0046' CCB1       1187          MOV    *R1+,*R2+     PUT 1ST PATTERN IN POWER ON TRANS RAM WORDS
0048' CCB1       1188          MOV    *R1+,*R2+     PUT 2ND
004A' CCB1       1189          MOV    *R1+,*R2+     PUT 3RD
                 1190
                 1191  *** CLEAR ALL RAM WORDS FROM RAMBEG TO RAMEND (POTRAM IS NOT IN THIS AREA)
                 1192
004C' 0201 0003* 1193          LI     R1,RAMBEG     START ADDRESS OF RAM EXCLUDING POWER ON WP
0050' 04F1       1194  50$     CLR    *R1+          CLEAR RAM WORD
0052' 0281 0004* 1195          CI     R1,RAMEND     END OF RAM ?
0056' 1AFC       1196          JL     50$           LOOP IF NOT
0058' 0720 0002* 1197          SETO   @CYPFRC       RESET CYCLE PER FRAME COUNTER
005C' 0720 0001* 1198          SETO   @FRCNT        RESET FRAME/SUBFRAME COUNTER
0060' 1000       1199          JMP    60$
                 1200
                 1201  *** COMMON ENTRY
                 1202
```

```
                    1203+ 60$    CALL   PIHW          INITIATE H/W
0062' 06A0 0009$    1210A        BL     @PIHW
                    1212+        CALL   PISW          INITIATE S/W
0066' 06A0 000A$    1219A        BL     @PISW
                    1221+        CALL   IDENT         SELECT A/ ENGINE
006A' 06A0 000E$    1228A        BL     @IDENT
006E' 0701          1230         SETO   R1
0070' C041          1231         MOV    R1,R1
0072' 1602          1232         JNE    64$           JIF NO NEED GENERATE SUM CHECK IN OPER. PROG.
                    1233+        CALL   MSGEN           TO GENERATE SUM-VALUES INTO EPROM
0074' 06A0 0008$    1240A        BL     @MSGEN
                    1242+ 64$    CALL   PINT          INITIALIZE INTERRUPTS
0078' 06A0 000B$    1249A        BL     @PINT
                    1251+        CALL   PIDA          INITIATE DATA INPUT
007C' 06A0 000C$    1258A        BL     @PIDA
0080' 0620 0000'    1260         DEC    @CNT          DECREMENT COUNTER
0084' 16CF          1261         JNE    99$           JIF FIRST TIME ROUND
                    1262
                    1263  ********************************************************************
                    1264  *      THE ABOVE SECTION IS REPEATED ONCE BECAUSE IT WAS DISCOVERED
                    1265  *      THAT 'SOMETHING' HAD TO BE INITIALIZED TWICE, OTHERWISE THE
                    1266  *      DITS W/A TEST WOULD FAIL. FOR SOME UNKNOWN REASON THES SEEMS
                    1267  *      TO WORK.
                    1268  ********************************************************************
                    1269
                    1270+        CALL   PIDO          INITIATE DATA OUTPUT
0086' 06A0 000D$    1277A        BL     @PIDO
                    1279  *      CALL   PODPIN        INITIALIZE DFDR PLAYBACK
                    1280
                    1281  *** INITIALIZE INTER-CPU SOFTWARE AND HARDWARE
                    1282+        CALL   ICINIT
008A' 06A0 0007$    1289A        BL     @ICINIT
                    1291  ***    PINIT=-1 INDICATES POWER ON INTERRUPT. RTIFR MODULE USES THIS FLAG
                    1292  ***    TO DELAY ONE SUBFRAME BEFORE STARTING DATA TRANSMISSION TO CPU#2.
008E' 0707          1293         SETO   PINIT         SET IT
                    1294  *** GO TO EXECUTIVE
                    1295+        CALLWP EXTIVE        ENTER BACK-GROUND
0090' 0420 0006$    1297A          BLWP @EXTIVE
                    1299  ********************************************************************
                    1300  *** POWER ON TRANSIENT FIXED PATTERN
0094' AAAA          1301  POTPAT DATA   >AAAA         1ST PATTERN
0096' 1234          1302         DATA   >1234         2ND    "
0098' 567B          1303         DATA   >567B         3RD    "
                    1304
                    1305+        LOCR   PRIV,CNT
0000' =0002         1308A  CNT   BSS    2
                    1309
                    1310  *
                    1311         END
```

```
                        1          IDT     PINT
                        2
                        3          SUBTTL  INITIALIZE INTERRUPTS
                        4
                        5  *       CALLING SEQ:    CALL    @PINT
                        6
                        7  *---------------------------------------------------------------+
                        8  *                                                               !
                        9  *       PINT INITIALIZES 3-THRU-11 EXCEPT INTERRUPT 7 (INTER CPU)
                       10  *       VIA THE 9901 (PROGRAMMABLE SYSTEMS INTERFACE).
                       11  *       REAL TIME INTERRUPT (INT 3) IS INITIALIZED FOR 15.616 MSEC
                       12  *       WHICH EXACTLY AS LONG AS DFOR INTERRUPT (INT 5).
                       13  *                                                               !
                       14  *       VERSION HISTORY:
                       15  *
                       16  *---------------------------------------------------------------+
                       17  *       VERSION : 1
                       18  *       PROGRAMMED BY : N.CONSTANTINIDES
                       19  *       CHECKED BY : N.CONSTANTINIDES
                       20
                       21          INTERN  PINT
                       22  * GLOBAL AREA:
                       23  *       (RAM)
                       24          EXTERN  RTPCYC          RT-PER-CYCLE COUNT
                       30          INCLUDE REGDEF          REGISTER DEFENITIONS
                       49          INCLUDE CNSTNT          CONSTANTS
                      182          INCLUDE SUBMAC          FUNCTIONAL MACROS
                      483          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                      713          INCLUDE JMPMAC          JUMP MACROS
                      747          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                      762          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
       =0000         1139          RSECT   PINT
                     1140  ********************************************************************
0000'                1141  PINT
0000'  0720 0001*    1142          SETO    @RTPCYC                 DONT START DATA-ACQ YET
                     1143  *
                     1144+         RESET   INT6,                   RESET ADC-INT
0004'  04E0 FFB8     1147A           CLR   @CADCLR
                     1155+         RESET   INT10,                  RESET DM3-1 INT
0008'  C020 FF84     1161A           MOV   @CD1IOP,R0
                     1166+         RESET   INT11,                  RESET DM3-2 INT
000C'  C020 FF88     1175A           MOV   @CD2IOP,R0
0010'  020C 0700     1177          LI      CRU,C9901               9901 CRU ADDR
0014'  3020 0000'    1178          LDCR    @INTIM,ZERO             INTERVAL-TIMER = 15.6 MSEC
                     1179  *                       DFOR (INT 5) OCCURES BEFOR RESETING IT !!!
                     1180  *                       ENABLE ONLY RT, ADC
0018'  3020 0002'    1181          LDCR    @INTEN,ZERO             INITIALIZE INT 3, 6.
001C'  0300 000D     1182          LIMI    INTMSK                  SET MASK
                     1183  *                       FROM NOW ON THE R-T INTERRUPT IS ACTIVE
0020'  045B          1184          RT
                     1185  ********************************************************************
                     1186+         PRVDAT
```

```
                    1188  *PRIVATE DATA AREA
0000' 05B9          1189  INTIM  DATA  >5B9          INTERVAL TIMER FOR 15.6 MSEC
0002' 004A          1190  INTEN  DATA  >04A          INTR 1, 3, 6, ENABLE
                    1191
                    1192        END
```

```
                     1          IDT     RMSG1
                     2          SUBTTL  RECEIVE MSG TYPE #1 SUBROUTINE
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: RMSG1                          AUTH: N.COSTANTINIDES     *
                     6  * VERSION: 1                           DATE: 8-FEB-84            *
                     7  *                                                                  *
                     8  * FUNCTION:     RECEIVE MSG TYPE #1 SUBROUTINE.                   *
                     9  *               TRANSFERS DOC DATA FROM CPU #2 TO CPU #1          *
                    10  *               DOC DATA BUFFER.                                  *
                    11  *                                                                  *
                    12  * CALLING MODULES: ICMD=INTER CPU MESSAGE DISPATCHER               *
                    13  *                                                                  *
                    14  * CALLING SEQ: BL @RMSG1                                          *
                    15  *                                                                  *
                    16  * INPUTS:       R1=RECEIVE BUFFER START ADDRESS                   *
                    17  *                                                                  *
                    18  * OUTPUTS:      CPU#2 DOC DATA TO DOC DATA BUFFER                 *
                    19  *               TMA-BIT8 = 1 FOR MSG #1 REC OK STATUS             *
                    20  *               1ST WORD OF RC MSG BUFFER=0 (MAKE IT AVAILABLE)   *
                    21  *                                                                  *
                    22  * MODULES REFERENCED:                                             *
                    23  *                                                                  *
                    24  * WORKSPACE AREA:       CALLER'S                                  *
                    25  *                                                                  *
                    26  * REGISTERS MODIFIED:   R0 TO R9                                  *
                    27  *                                                                  *
                    28  ********************************************************************
       =0000        29         RSECT   RMSG1
                    30  *** CALL NAME
                    31         INTERN  RMSG1
                    32  *** VARIABLES REFERENCED
                    33         EXTERN  SYDOCB,TMA
                    34  *** CONSTANTS REFERENCED
                    35         EXTERN  D15,BB
                    36  *** MODULES REFERENCED
                    37  *** TABLES REFERENCED
                    38  *** REGISTERS DEFINITION
                    39  ********************************************************************
0000'               40  RMSG1
                    41  * COPY INTER CPU RECEIVE MSG BUFFER IN DOC DATA BUFFER
0000'  C0C1        42         MOV     R1,R3           SAVE RECEIVER BUFFER START ADDRESS
0002'  0200 0005   43         LI      R0,5            5 NUMBER OF DATA WORDS
                    44  *
                    45  *      DATA WORDS ARE (IN ORDER)
                    46  *      1. DAY
                    47  *      2. MONTH
                    48  *      3. FLT NUMBER MSH
                    49  *      4. FLT NUMBER LSH
                    50  *      5. FLT LEG
                    51  *
0006'  0221 0004    52         AI      R1,4            ADD 2 OVERHEAD
```

RECEIVE MSG TYPE #1 SUBROUTINE RMSG1.SRC

```
000A' 0205 0001*    53          LI      R5,SYDOCB       CPU#1 ROC DATA BUFFER START ADDRESS
000E' CD71          54  10$     MOV     *R1+,*R5+       MOVE ONE WORD AT A TIME
0010' 0600          55          DEC     R0              DECREMENT WORD COUNTER
0012' 15FD          56          JGT     10$             LOOP
                    57  *
0014' 04D3          58          CLR     *R3             MAKE RECEIVER MSG BUFFER AVAILABLE
                    59  *
                    60  *   SET MSG #1 RC OK BIT FOR CPU #2
                    61  *   WHICH WILL BE SET IN MSG #0
0016' E820 0004*    62          SOC     @B8,@TMA        SET MSG TYPE #1 REC OK BIT
001A' 0002*
                    63  *
001C' 045B          64          RT
                    65  ********************************************************************************
                    66          END
```

No errors detected

```
                        1          IDT     RMSG2
                        2          SUBTTL  RECEIVE MSG TYPE #1 SUBROUTINE
                        3  ********************************************************************
                        4  *                                                                  *
                        5  * NAME: RMSG2                          AUTH: N.COSTANTINIDES     *
                        6  * VERSION: 1                           DATE: 22-JUN-82           *
                        7  *                                                                  *
                        8  * FUNCTION:     RECEIVE MSG TYPE #2 SUBROUTINE.                   *
                        9  *               RECEIVE SYSTEM ERROR MESSAGES FROM CPU #2         *
                       10  *               TO CPU #1 SYSTEM ERROR BUFFER                     *
                       11  *                                                                  *
                       12  * CALLING MODULES: ICMD=INTER CPU MESSAGE DISPATCHER               *
                       13  *                                                                  *
                       14  * CALLING SEQ: BL @RMSG2                                          *
                       15  *                                                                  *
                       16  * INPUTS:       R1=RECEIVE BUFFER START ADDRESS                   *
                       17  *                                                                  *
                       18  * OUTPUTS:      ERROR MSG TO ERROR MSG BUFFER                     *
                       19  *               TMA-BIT9 = 1 FOR MSG #2 REC OK STATUS BIT         *
                       20  *               1ST WORD OF RC MSG BUFFER = 0 (MAKE IT AVAILABLE)
                       21  *                                                                  *
                       22  * MODULES REFERENCED:   NONE                                      *
                       23  *                                                                  *
                       24  * WORKSPACE AREA:       CALLER'S                                  *
                       25  *                                                                  *
                       26  * REGISTERS MODIFIED:   R0 TO R9                                  *
                       27  *                                                                  *
                       28  ********************************************************************
       =0000           29          RSECT   RMSG2
                       30  *** CALL NAME
                       31          INTERN  RMSG2
                       32  *** VARIABLES REFERENCED
                       33          EXTERN  SYEBF2,TMA
                       34  *** CONSTANTS REFERENCED
                       35          EXTERN  D15,B9
                       36  *** MODULES REFERENCED
                       37  *** TABLES REFERENCED
                       38  *** REGISTERS DEFINITION
                       39  ********************************************************************
  0000'                40  RMSG2
                       41  * COPY INTER CPU RECEIVE ERROR MSG IN ERROR MSG BUFFER
  0000'  C0C1          42          MOV     R1,R3           SAVE RECEIVER BUFFER START ADDRESS
  0002'  0200 0008     43          LI      R0,8            8 NUMBER OF DATA WORDS
  0006'  0221 0004     44          AI      R1,4            ADD 2 OVERHEAD
  000A'  0205 0001*    45          LI      R5,SYEBF2       CPU#2 SYSTEM ERROR BUFFER START ADDRESS
  000E'  CD71          46  10$     MOV     *R1+,*R5+       MOVE ONE WORD AT A TIME
  0010'  0600          47          DEC     R0              DECREMENT WORD COUNTER
  0012'  15FD          48          JGT     10$             LOOP
                       49  *
  0014'  04D3          50          CLR     *R3             MAKE RECEIVER MSG BUFFER AVAILABLE
                       51  *
                       52  *   SET MSG #2 RC OK BIT FOR CPU #2
```

```
                        53   *  WHICH WILL BE SET IN MSG #0
    0016'  E820 0004*   54          SOC     @B9,@TMA        SET MSG TYPE #2 REC OK BIT
    001A'  0002*
                        55   *
    001C'  045B         56          RT
                        57   ********************************************************************************
                        58          END
```

No errors detected

```
                    1          IDT     RTIW64
                    2          SUBTTL  SET DFDR WORD 64
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: RTIW64.SRC                        AUTH: N.CONSTANTINIDES  *
                    6  * VERSION: 1                              DATE: 16-MAY-1983       *
                    7  *                                                                  *
                    8  * FUNCTION: THIS MODULE PUTS SUPERFRAME DATA IN EVERY SUBFRAME     *
                    9  *           WORD 64.                                               *
                   10  *                        SUBFRAME #1 = FRAME COUNT (FRCNT/2)       *
                   11  *                        SUBFRAME #2 = STATUS ERROR (SYSTWB)       *
                   12  *                        SUBFRAME #3 = DOC DATA (SYDOCB)           *
                   13  *                        SUBFRAME #4 = CAL DATA (SYCALB)           *
                   14  *          IT ALSO MOVES FRAME COUNT TO AIDS BUFFER.               *
                   15  *                                                                  *
                   16  * CALLING MODULES: RTIFR (REAL TIME INTERRUPT                      *
                   17  *                                                                  *
                   18  * CALLING SEQ: BL @RTIW64                                         *
                   19  *                                                                  *
                   20  * INPUTS:                                                         *
                   21  *               R5 = FRAME/SUBFRAME COUNTER                       *
                   22  *               OUTBB1 = START ADDRESS OF AIDS BUFFER             *
                   23  *                                                                  *
                   24  * OUTPUTS:      DFDR WORD 64 = SUPER FRAME DATA                   *
                   25  *                                                                  *
                   26  * MODULES REFERENCED: NONE                                        *
                   27  *                                                                  *
                   28  * WORKSPACE AREA: RTW                                             *
                   29  *                                                                  *
                   30  * REGISTERS MODIFIED:   R0,R1,R2                                  *
                   31  *                                                                  *
                   32  * VERSION HISTORY:                                                *
                   33  *                                                                  *
                   34  ********************************************************************
=0000              35         RSECT   RTIW64
                   36  *** CALL NAME
                   37         INTERN  RTIW64
                   38  *** VARIABLES REFERENCED
                   39         EXTERN  DFDQ
                   40         EXTERN  OUTBB1
                   41         EXTERN  SYDOCB,SYSTWB,SYCALB
                   42  *** CONSTANTS REFERENCED
                   43         EXTERN  D0
                   44  *** TABLES REFERENCED
                   45  *** MODULES REFERENCED
                   46  *** LIBRARY
                   47  *** REGISTERS DEFINITION
=0005              48  FRCNT   EQU     R5              FRAME/SUBFRAME COUNTER
```

```
                        50  ***
          =015A         51  ICL137  EQU     173*2           LOCATION OF DFIR FRAME COUNTER IN
                        52  *                               CPU 2 BUFFER
                        53  ****************************************************************
                        54  *   PUT SUPERFRAME DATA:EVERY SUBFRAME WORD #64
     0000'  C045        55  RTIW64  MOV     FRCNT,R1        FRAME/SUBFRAME COUNTER
     0002'  0241 0003   56          ANDI    R1,3            ONLY THE SUBFRAME COUNT NEEDED
     0006'  0A11        57          SLA     R1,1            CREATE BRANCH OFFSET VIA SUBFRAME * 2
     0008'  C005        58          MOV     FRCNT,R0
     000A'  0920        59          SRL     R0,2            R0 FOR FRAME COUNT
                        60  *   MOVE DFIR FRAME COUNT IN AIDS BUFFER
     000C'  C0A0 0002*  61          MOV     @OUTBB1,R2      START ADDR OF PRESENT AIDS BUFFER
     0010'  C880 015A   62          MOV     R0,@ICL137(R2)  MOVE FRAME COUNT IN AIDS BUFFER(FOR CPU #2)
                        63  *
     0014'  C085        64          MOV     FRCNT,R2        FRAME/SUBFRAME COUNT
     0016'  0242 003C   65          ANDI    R2,)3C          4 BITS OF SUPERFRAME COUNT
     001A'  0912        66          SRL     R2,1            R2 FOR SUPERFRAME COUNT * 2
                        67  *
     001C'  0461 0020'  68          B       @100$(R1)
     0020'  100E        69  100$    JMP     150$
     0022'  1006        70          JMP     130$
     0024'  1008        71          JMP     140$
     0026'  1000        72          JMP     120$
                        73  *
     0028'  C822 0003*  74  120$    MOV     @SYDOCB(R2),@DFDB+126   SUBF 3 WD 64 = IOC DATA
     002C'  0000"
     002E'  100A        75          JMP     202$
     0030'  C800 0000"  76  130$    MOV     R0,@DFDB+126            SUBF 1 WD 64 = FRAME COUNT
     0034'  1007        77          JMP     202$
     0036'  C822 0004*  78  140$    MOV     @SYSTWB(R2),@DFDB+126           SUBF 2 WD 64 = STATUS ERROR
     003A'  0000"
     003C'  1003        79          JMP     202$
     003E'  C822 0005*  80  150$    MOV     @SYCALB(R2),@DFDB+126   SUBF 4 WD 64 = CAL DATA
     0042'  0000"
                        81  *
     0044'  045B        82  202$    RT
                        83          END
```

```
                     1          IDT     RTIDA
                     2
                     3          SUBTTL  START DATA ACQ CYCLE
                     4
                     5  *       CALLING SEQ:    CALL    @RTIDA
                     6
                     7  *-----------------------------------------------------------------+
                     8  *                                                                 |
                     9  *       RTIDA GETS ANALOG, DISCRETE, DITS #1 AND DITS #2 PARAM-
                    10  *       COUNT, SENDS DITS, ANALOG ADDRESSES, AND RESET ADC-INTR
                    11  *       TO INITIATE ANALOG AND DITS DATA ACQUISTION, IN THE NEW
                    12  *       CYCLE.
                    13  *                                                                 |
                    14  *-----------------------------------------------------------------+
                    15  *       CALLING MODULE  RTISR
                    16  *       VERSION : 2
                    17  *       PROGRAMMED BY : M.CONSTANTINIDES
                    18  *       MODIIED: 4-MAY-1983 M.CONSTANTINIDES
                    20
                    21          INTERN  RTIDA
                    22  * REFERD MODULES:
                    23          EXTERN  AMSA
                    24          EXTERN  DRSAW
                    25  * GLOBAL AREA:
                    26  *       (RAM)
                    27  *                               AN, DC, DR1 AND DR2 PARAM-COUNT
                    28          EXTERN  ANPCNT,DR1PCT,DR2PCT
                    29  *                               AN, DR1 AND DR2 WP ADDR
                    30          EXTERN  ANW,DR1W,DR2W
                    31          EXTERN  RTW             REAL TIME INT WP
                    32          EXTERN  DR1ARD
                    33          EXTERN  DR2ARD
                    34          EXTERN  DASFLG,DFDD
                    35          EXTERN  SSEG
                    36  *
                    37  *       (ROM)
                    38  *                               AN-, DC-, DR1- AND DR2-PARAM-QUANT TABLE
                    39          EXTERN  ANPQT,DR1PQP,DR1PQG,DR2PQP,DR2PQG
                    40          EXTERN  DR1PNT          DR1-PARAM-NUMBER-TABLE
                    41          EXTERN  DR2PNT          DR2-PARAM-NUMBER-TABLE
                    42  *
                    43  *       TABLES REFERENCED
                    44  *
                    45          EXTERN  DR1A1P,DR1A1G,DR2A1P,DR2A1G,DR1I4P,DR1I4G,DR2I4P,DR2I4G
                    46          EXTERN  ANCIT,ANARD
                    47  *
=0000               48  TMP     EQU     R0      =       SCRATCH
=0008               49  CYPFRC  EQU     R8      =       CYCLE-PER-FRAME COUNT
                    50
                    51          INCLUDE ENCLOS
                    53  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
```

```
                     55          INCLUDE REGDEF          REGISTER DEFENITIONS
                     74          INCLUDE CNSTNT          CONSTANTS
                    202          INCLUDE SUBMAC          FUNCTIONAL MACROS
                    503          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    733          INCLUDE JMPMAC          JUMP MACROS
                    767          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    782          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000    1158          RSECT   RTIDA
```

```
006E' 160B         1214          JNE    80$                    JIF GE
                   1215+         MOVBRJ @DR2PQP(CYPFRC),TMP    BITS #1 QTY
0070' D028 0012*   1217A           MOVB @DR2PQP(CYPFRC),TMP
0074' 0980         1218A           SRL  TMP,C8
0076' C800 0005*   1220          MOV    TMP,@DR2PCT            R8 OF DR1W W.A
007A' C048         1221          MOV    CYPFRC,R1
007C' 0A11         1222          SLA    R1,1                   CHANGE TO EVEN WORD ADDR
007E' C821 001C*   1223          MOV    @DR2I4P(R1),@DR2ARD
0082' 000B*
0084' 100A         1224          JMP    81$
0086'              1225  80$
                   1226+         MOVBRJ @DR2PQG(CYPFRC),TMP    BITS #1 QTY
0086' D028 0013*   1228A           MOVB @DR2PQG(CYPFRC),TMP
008A' 0980         1229A           SRL  TMP,C8
008C' C800 0005*   1231          MOV    TMP,@DR2PCT            R8 OF DR1W W.A
0090' C048         1232          MOV    CYPFRC,R1
0092' 0A11         1233          SLA    R1,1                   CHANGE TO EVEN WORD ADDR
0094' C821 001D*   1234          MOV    @DR2I4G(R1),@DR2ARD
0098' 000B*
                   1235
                   1236  *** START ANALOG DATA ACQUISITION EVERY NEW CYCLE
009A' 02E0 0006*   1237  81$     LWPI   ANW                    WP = TO AN DATA-ACQ WP
                   1238+         RESET  INT6,                  RESET ADC INTERRUPT
009E' 04E0 FFB8    1241A           CLR  @CADCLR
                   1249+         CALL   ANSA                   SEND NEXT ANALOG ADDR TO START ADC INT
00A2' 06A0 0001*   1256A           BL   @ANSA
                   1258  *** START BITS #1 DATA ACQUISITION
00A6' 02E0 0007*   1259          LWPI   DR1W                   WP = TO BITS #1 DATA-ACQ WP
                   1260  *  BITS #1 OFFSET TABLE INDEX
00AA' C820 000E*   1261          MOV    @SSEG,@SSEG
00AE' 000E*
00B0' 1603         1262          JNE    50$
00B2' C1E0 0016*   1263          MOV    @DR1A1P,R7
00B6' 1002         1264          JMP    51$
00B8' C1E0 0017*   1265  50$     MOV    @DR1A1G,R7             DR1DA. 4 SUBF PARA OFFSET TABLE START ADDR.
                   1266
                   1267+ 51$     CALL   DRSAW,<R4,=,CD1IOP,X>,<R5,=,DR1PNT,X>
00BC' 0204 FF84    1277C           LI   R4,CD1IOP
00C0' 0205 0014*   1293D           LI   R5,DR1PNT
00C4' 06A0 0002*   1308A           BL   @DRSAW
                   1310  *** START BITS #2 DATA ACQUISITION
00C8' 02E0 0008*   1311          LWPI   DR2W                   WP = TO BITS #2 DATA-ACQ WP
                   1312  *** BITS #2 OFFSET TABLE INDEX
00CC' C820 000E*   1313          MOV    @SSEG,@SSEG
00D0' 000E*
00D2' 1603         1314          JNE    90$
00D4' C1E0 0018*   1315          MOV    @DR2A1P,R7
00D8' 1002         1316          JMP    91$
00DA' C1E0 0019*   1317  90$     MOV    @DR2A1G,R7             DR1DA. 4 SUBF PARA OFFSET TABLE START ADDR.
                   1318
                   1319+ 91$     CALL   DRSAW,<R4,=,CD2IOP,X>,<R5,=,DR2PNT,X>
00DE' 0204 FF88    1329C           LI   R4,CD2IOP
```

```
                    1160  ********************************************************************************
                    1161  *  EVERY NEW CYCLE START AN 8 DITS DATA ACQUISITION
0000'               1162  RTIDA
0000'  C80B 0000'   1163          MOV     LINK,@LINKZ     SAVE LINKER
                    1164  *** CLEAR NEXT 8 DFDR OUTPUT WORDS EVERY NEW CYCLE.
                    1165  ***     THIS IS NECESSARY FOR DESTINATION ROUTINE (DSTINE OR DSTINW)
0004'  0200 0008    1166          LI      R0,8            NUMBER OF WORDS TO BE CLEARED
0008'  C048         1167          MOV     CYPFRC,R1       CYCLE PER FRAME COUNTER
000A'  0241 0007    1168          ANDI    R1,7            SAVE CYCLE PER SEC COUNT ONLY (8 CYCLES/SEC)
000E'  0A41         1169          SLA     R1,4            R1=INDEX OF 1ST DFDR OUTPUT WORD OF NEW CYCLE
0010'  1601         1170          JNE     10$             JIF 1ST CYCLE OF NEW SUBFRAME
0012'  05C1         1171          INCT    R1              BUMP INDEX TO SKIP OVER SYNC CODE WORD
0014'  0221 000D*   1172  10$     AI      R1,DFOO         START ADDR OF DFDR OUTPUT
0018'  04F1         1173  12$     CLR     *R1+            CLEAR ALL 8 WORDS
001A'  0600         1174          DEC     R0
001C'  15FD         1175          JGT     12$             LOOP 8 TIMES
                    1176  *** ANALOG SET UP                               PAR-CNT = AN, PARAM-COUNT
001E'  04E0 000C*   1177          CLR     @DASFLG         TO DO AN CAL OR BITE
                    1178+         MOVBRJ  @ANPQT(CYPFRC),TMP      ANALOG QTY
0022'  D028 000F*   1180A           MOVB  @ANPQT(CYPFRC),TMP
0026'  0980         1181A           SRL   TMP,8
0028'  C800 0003*   1183          MOV     TMP,@ANPCNT             RB OF ANW W.A
002C'  D028 001E*   1184          MOVB    @ANCIT(CYPFRC),TMP      AN CYCLE INDEX TO GET DATA FROM OFFSET TABLES
0030'  0880         1185          SRA     TMP,8
0032'  C800 001F*   1186          MOV     TMP,@ANARO              R10 OF ANALOG SET EVERY CYCLE
                    1187  *** DITS #1 SET UP
0036'  C820 000E*   1188          MOV     @SSEG,@SSEG             CONFIGURATION
003A'  000E*
003C'  160B         1189          JNE     70$                     JIF GE
                    1190+         MOVBRJ  @DR1PQP(CYPFRC),TMP     DITS #1 QTY
003E'  D028 0010*   1192A           MOVB  @DR1PQP(CYPFRC),TMP
0042'  0980         1193A           SRL   TMP,8
0044'  C800 0004*   1195          MOV     TMP,@DR1PCT             RB OF DR1W W.A
0048'  C048         1196          MOV     CYPFRC,R1
004A'  0A11         1197          SLA     R1,1                    CHANGE TO EVEN WORD ADDR
004C'  C821 001A*   1198          MOV     @DR1I4P(R1),@DR1ARO
0050'  000A*
0052'  100A         1199          JMP     71$
0054'               1200  70$
                    1201+         MOVBRJ  @DR1PQG(CYPFRC),TMP     DITS #1 QTY
0054'  D028 0011*   1203A           MOVB  @DR1PQG(CYPFRC),TMP
0058'  0980         1204A           SRL   TMP,8
005A'  C800 0004*   1206          MOV     TMP,@DR1PCT             RB OF DR1W W.A
005E'  C048         1207          MOV     CYPFRC,R1
0060'  0A11         1208          SLA     R1,1                    CHANGE TO EVEN WORD ADDR
0062'  C821 001B*   1209          MOV     @DR1I4G(R1),@DR1ARO
0066'  000A*
                    1210
                    1211  *** DITS #2 SET UP
0068'               1212  71$
0068'  C820 000E*   1213          MOV     @SSEG,@SSEG             CONFIGURATION
006C'  000E*
```

```
00E2' 0205 0015* 1345D          LI    R5,DR2PNT
00E6' 06A0 0002* 1360A          BL    @DRSAW
                 1362
00EA' 02E0 0009* 1363          LWPI   RTW            RESET WP TO RT WP
00EE' C2E0 0000' 1364          MOV    @LINKZ,LINK    RESTORE LINKER
00F2' 045B       1365          RT
                 1366  ********************************************************************************
                 1367  * PRICVATE AREA
                 1368+         LOCR   PRIV,LINKZ            LINK-SAVE AREA
0000' =0002      1371A LINKZ   BSS    2
                 1372          END
```

```
                    1          IDT     RTIFR
                    2
                    3          SUBTTL  INITIALIZE FRAME
                    4
                    5  *       CALLING SEQ:    CALL    @RTIFR
                    6
                    7  *-------------------------------------------------------------->
                    8  *                                                              !
                    9  *       RTIFR SETS THE CYCLE-PER-FRAME-, CYCLE-, SUBFRAME-, FRAME-
                   10  *       AND SUPER-FRAME-COUNT, PUT THE SYNC-CODE INTO DFDR BUFFER
                   11  *       IN EACH NEW SUB-FRAME, GETS THE PARAM-ARRAY-OFFSET FOR ALL
                   12  *       PARAMETER TYPE (ANALOG, DISCRETE AND DITS).
                   13  *                                                              !
                   14  *-------------------------------------------------------------->
                   15  *       CALLING MODULE  RTISR
                   16  *       PROGRAMMED : N.CONSTANTINIDES
                   17  *       CHECKED    : N.CONSTANTINIDES
                   18
                   19          INTERN  RTIFR
                   20
                   21  * GLOBAL AREA:
                   22  *       (RAM)
                   23  *                               AN-, DC-, DR1- AND DR2-ARRAY-OFFSET
                   24          EXTERN  DFDQ            DFDR-OUTPUT-QUEUE
                   25          EXTERN  DFWRC           DFDR OUTPUT WORD COUNTER
                   26          EXTERN  PINIT           ANY POWER INTERRUPT INIT FLAG
                   27          EXTERN  ICQER           INT CPU OUTPUT QUE FULL ERRM
                   28          EXTERN  OUTBR,ICQBAB,ICQBAC,OUTBB1,DPBUFF,DPFLPB
                   29          EXTERN  ICIFL6
                   30  *       (ROM)
                   31          EXTERN  SYCODT          SYNC-CODE TABLE
                   32          EXTERN  ICQBAA          INT CPU OUT BUFF ADDR TABLE
                   33          EXTERN  ICQBKA          INT CPU OUT ARG-BLOCK ADDR
                   34          EXTERN  D1,D7,D3
                   35  *
                   36  *  MODULES REFERENCED
                   37          EXTERN  ICQMSG
                   38          EXTERN  RTIW64
                   39          EXTERN  SYSOK,SYSER,SYET02
                   40  *
=0001              41  WFFS    EQU     R1      =       SCRATCH
=0004              42  ICQBAI  EQU     R4      =       INT CPU OUTPUT BUFF ADDR TABLE INDEX
=0005              43  FRCNT   EQU     R5      =       FRAME/SUBFRAME CTR(INC AT 1ST CYCLE OF SUBFRAME)
                   44  *       EQU     R7      =       FLAG FOR CALLER
=0008              45  CYPFRC  EQU     R8      =       CYCLE PER FRAME COUNT(0 TO 31).
=0009              46  RTPCYC  EQU     R9      =       RT SUB CYCLE-PER-CYCLE COUNT (8 TO 1).
                   47          INCLUDE ENCLOS
                   49  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   51          INCLUDE REGDEF          REGISTER DEFENITIONS
                   70          INCLUDE CNSTNT          CONSTANTS
                  203          INCLUDE SUBMAC          FUNCTIONAL MACROS
                  504          INCLUDE MSCMAC          MISCELLANEOUS MACROS
```

```
                    1161  ********************************************************************
0000' C28B          1162  RTIFR   MOV     LINK,R10        SAVE LINK REG
0002' 0588          1163  10$     INC     CYPFRC          BUMP CYCLE PER FRAME COUNTER
0004' 0248 001F     1164          ANDI    CYPFRC,>1F      JUST IN CASE
0008' 2620 0010*    1165          CZC     @D7,CYPFRC
                    1166+         DOIF    ,EQ,,,          EXIT IF NOT TIME FOR NEW SUBFRAME
000C' 1649          1336E             JNE   91$
                    1472  *** NEW SUBFRAME
000E' C820 0002*    1473          C       @DFWRC,@D55X2   DFDR O/P WORD COUNTER
0012' 0048'
0014' 1402          1474          JHE     15$             JIF SYNCHRONIZED WITH DFDR O/P
0016' 0703          1475          SETO    R3              SET TO DONT START DATA ACQ FOR CALLER
0018' 1044          1476          JMP     100$            EXIT
001A'               1477  15$
                    1478  *** INCREMENT FRAME/SUBFRAME COUNTER
001A' 0585          1479          INC     FRCNT           BUMP FRAME/SUBFRAME COUNTER
001C' 0245 3FFF     1480          ANDI    FRCNT,>3FFF     CLEAR OVERFLOW BITS
                    1481  *** INSERT SUPER FRAME DATA IN DFDR WD #64
0020' 06A0 0013*    1482          BL      @RTIW64         PUT SUPER FRAME DATA IN DFDR WD #64
                    1483  *** TIME TO SEND DATA MSG TO CPU 2
0024' C020 0003*    1484          MOV     @PINIT,R0       ANY POWER INTERRUPT INIT FLAG
                    1485+         DOIF    ,NE,,,          DO IF ANY POWER INTERRUPT
0028' 1303          1656E             JEQ   92$
                    1791  *   DO NOT SEND DATA TO CPU 2
002A' 04E0 0003*    1792          CLR     @PINIT
002E' 1020          1793          JMP     93$
                    1794+         ENDBLK
0030'               1930E 92$
                    2014  ***  MOVE 64 WORDS OF DFDR OUTPUT DATA INTO INTER-CPU O/P BUFFER. 1111 USEC REAL TIME USED
0030' 0200 0040     2015          LI      R0,64           NUMB. OF WORDS
0034' C060 0005*    2016          MOV     @OUTBB,R1       DFDR O/P BUFFER S.A
0038' C0A4 0006*    2017          MOV     @ICOBAB(ICOBAI),R2      INT-CPU OUTPUT BUFF S.A
003C' CCB1          2018  20$     MOV     *R1+,*R2+       MOV 64 WORDS
003E' 0600          2019          DEC     R0
0040' 15FD          2020          JGT     20$
                    2021  **** MOVE 64 WORDS OF DFDR PLAYBACK DATA INTO INT-CPU OUTPUT BUFFER
                    2022  *** 900 USEC OF REAL TIME  USED
                    2023  *       MOV     @DPBUFF,R1      GET DFDR PB BUFFER POINTER
                    2024  *       LI      R0,64           NUMBER OF WORDS
                    2025  *       MOV     @DFFLPB(R1),R1  DFDR PLAYBACK BUFFER S.A
                    2026  *       MOV     @ICOBAB(ICOBAI),R2      INTER-CPU O/P BUFFER S.A
                    2027  *30$    MOV     *R1+,*R2+       64 WORDS
                    2028  *       DEC     R0
                    2029  *       JGT     30$
                    2030  *** MOVE 8 WORDS OF SYSTEM ERROR BUFFER TO INTER-CPU OUTPUT BUFFER
0042' 06A0 0016*    2031          BL      @SYET02         MOVE SYEBF TO INTER-CPU BUFFER
                    2032  ***
0046' C064 000E*    2033          MOV     @ICOBKA(ICOBAI),R1      ADDRESS OF ARG BLOCK
004A' 06A0 0012*    2034          BL      @ICOMSG         QUE DATA MESSAGE
004E' C060 000B*    2035          MOV     @ICIFLG,R1      GET IC INTERRUPT FLAG
0052' 160E          2036          JNE     199$
                    2037+         DOIF    ,EQ,,,          DO IF QUE FULL
```

INITIALIZE FRAME RTIFR.SRC

```
                    734     INCLUDE JMPMAC          JUMP MACROS
                    768     INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    783     INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
          =0000     1159    RSECT   RTIFR
```

```
0034' 160D          2215E                JNE   93$
0036' 05A0 0004*    2343           INC     @ICOER          KEEP COUNT
003A' 8820 0004*    2344           C       @ICOER,@D3
003E' 0011*
                    2345+          DOIF    ,GT,,,,         DO IF ERROR
0040' 1501          2531F             JGT  $+4
0042' 1004          2532F             JMP  94$
0044' C060 00AA'    2653           MOV     @ERCOD,R1       INTER-CPU COMM ERROR
0048' 06A0 0015*    2654           BL      @SYSER          INP REG: R1. SET ERROR BIT
                    2655+          ENDBLK
004C'               2799E  94$
004C' 1007          2875           JMP     110$
                    2876+          ELSEDO
004E' 1006          3149E              JMP  95$
0070'               3275E  93$
                    3352   ** INTER-CPU COMM OPERATIONAL
0072' 04E0 0004*    3353   199$    CLR     @ICOER          CLEAR INTER-CPU QUEUE FULL COUNTER
0076' C060 00AA'    3354           MOV     @ERCOD,R1       INTER-CPU COMM ERROR CODE
007A' 06A0 0014*    3355           BL      @SYSOK          INP REG: R1. RESET ERROR BIT.
                    3356+          ENDBLK
007E'               3504E  95$
                    3576   *   TIME TO SWITCH INTER-CPU O/P BUFFER
0080' 05C4          3577   110$    INCT    ICOBAI          BUMP INT-CPU O/P BUFFER ADDR TABLE INDEX
0082' 0284 0002     3578           CI      ICOBAI,2
0086' 1201          3579           JLE     40$             JIF     ITS LIMIT CHECK OK
0088' 04C4          3580           CLR     ICOBAI          ELSE, RESET
008A' C824 000D*    3581   40$     MOV     @ICOBAA(ICOBAI),@OUTBB1         NEW INT-CPU O/P BUFFER
008E' 0008*
                    3582
                    3583   ***     PUT NEXT SYNC CODE IN DFDR OUTPUT BUFFER
                    3584
0090' C045          3585           MOV     FRONT,R1
0092' 0241 0003     3586           ANDI    R1,3            SAVE SUBFRAME COUNT ONLY
0096' 0A11          3587           SLA     R1,1             * 2
0098' C821 000C*    3588           MOV     @SYCODT(R1),@DFOR       SYNC CODE IN 1ST WORD OF BUFFER
009C' 0001*
                    3589
                    3590   *** CHECK IF NEW FRAME
                    3591
                    3592+          DOIF    R1,EQ,,,,       IF SUBFRAME COUNT = 0,
009E' C041          3726B             MOV  R1,R1
00A0' 1601          3782E               JNE   96$
                    3898
                    3899   *** NEW FRAME
                    3900
00A2' 04CB          3901           CLR     CYPFRC          RESET 32 CYCLE COUNTER
                    3902+          ENDBLK          END DOIF
00A6'               4054E  96$
                    4122+          ENDBLK          END DOIF
00AA'               4254E  91$
00AA' 04C3          4342           CLR     R3              SET TO DATA ACQ FOR THE CALLER
00AC' C2CA          4343   100$    MOV     R10,LINK        UNSAVE LINK REG
```

```
00[illegible] C0C3    4344          MOV     R3,R3           FOR CALLER
00[illegible] 045B    4345          RT
              4346    ********************************************************************
00[illegible] 006E    4347  D55X2   DATA    55*2
00[illegible] 0501    4348  ERCOD   DATA    )501            INTER-CPU ERROR CODE
              4349
              4350          END
```

```
                    1          IDT     RTISR
                    2          SUBTTL  REAL TIME ISR (INTERRUPT 3)
                    3  ********************************************************************
                    4  *                                                                  *
                    5  * NAME: RTISR                          AUTH: N.COSTANTINIDES     *
                    6  * VERSION: 2                           DATE: 26-MAY-1983         *
                    7  *                                                                  *
                    8  * FUNCTION:                                                       *
                    9  *       RTISR SETS TIME-COUNT, RT-PER-CYCLE-COUNT AND AT THE     *
                   10  *       START OF A NEW CYCLE  (EVERY 125 MSEC),  INITIALIZES     *
                   11  *       THE DATA ACQUISITION.                                    *
                   12  *       THE DISCRETE PARAMETER ACQUISITION INITIATED EVERY CYCLE  *
                   13  *       BY QUEUING JOBSFS IN THE JOB QUE.                        *
                   14  *       IT ALSO CHECKS THE FRONT PANEL SWICH AND RESET THE WATCH  *
                   15  *       DOG TIMER (EVERY 16 MSEC).                               *
                   16  *       START INTER-CPU TRANSMISSION                             *
                   17  *                                                                  *
                   18  * CALLING MODULES: INTERRUPT 3                                    *
                   19  *                                                                  *
                   20  * CALLING SEQ: INTERRUPT 3                                        *
                   21  *                                                                  *
                   22  * INPUTS:                                                         *
                   23  *                                                                  *
                   24  * OUTPUTS:                                                        *
                   25  *                                                                  *
                   26  * MODULES REFERENCED: DCACQ = DISCRETE DATA ACQUISITION           *
                   27  *                     RTIFR=CYCLE & SUBFRAME FUNCTION             *
                   28  *                     RTIDA=INITIALIZE DATA ACQUISITION           *
                   29  *                                                                  *
                   30  * WORKSPACE AREA: RTW                                             *
                   31  *                                                                  *
                   32  * REGISTERS MODIFIED: R0,R9                                       *
                   33  *                                                                  *
                   34  * VERSION HISTORY:                                                *
                   35  *                                                                  *
                   36  ********************************************************************
=0000              37          RSECT   RTISR
                   38  ### CALL NAME
                   39          INTERN  RTISR
                   40  ### VARIABLES REFERENCED
                   41  ### CONSTANTS REFERENCED
                   42          EXTERN  B0
                   43  ### TABLES REFERENCED
                   44  ### MODULES REFERENCED
                   45          EXTERN  JBQUE,RTSIM1
                   46          EXTERN  RTIFR,RTIDA,ICSTX
                   47  ### LIBRARY
                   48          INCLUDE ENCLOS
                   50  ###     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                   52          INCLUDE REGDEF          REGISTER DEFENITIONS
                   71          INCLUDE CNSTNT          CONSTANTS
                  199          INCLUDE SUBMAC          FUNCTIONAL MACROS
```

```
                    500          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                    730          INCLUDE JMPMAC          JUMP MACROS
                    764          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                    779          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                   1155  *** REGISTERS DEFINITION
=0004              1156  ICOBAI  EQU     R4      =       INT-CPU O/P BUFFER ADDR TABLE INDEX
=0005              1157  FRCNT   EQU     R5      =       FRAME/SUBFRAME COUNT
=0006              1158  TIMEM   EQU     R6      =       MSW OF TIME
=0007              1159  TIMEL   EQU     R7      =       LSW
=0007              1160  CLOCK   EQU     R7
=0008              1161  CYPFRC  EQU     R8      =       CYCLE PER FRAME COUNT
=0009              1162  RTPCYC  EQU     R9      =       RT-SUB CYCLE-PER-CYCLE COUNT
                   1163  ********************************************************************************
```

```
                    1151  *** DOC DATA
                    1152
0004' 04C7          1153          CLR     R7
0006' C1E0 0005*    1154          MOV     @LATM1,R7
000A' 0967          1155          SRL     R7,6
000C' C807 0005*    1156          MOV     R7,@LATM1
0010' 04C7          1157          CLR     R7
0012' C1E0 0006*    1158          MOV     @LONGM1,R7
0016' 0967          1159          SRL     R7,6
0018' C807 0006*    1160          MOV     R7,@LONGM1
001C' 0201 0004*    1161          LI      R1,SYDOCB
0020' 0202 001E     1162          LI      R2,30
0024'               1163  30$
0024' C1E2 0004*    1164          MOV     @SYDOCB(R2),R7
0028' 0247 00FF     1165          ANDI    R7,>00FF
002C' C887 0004*    1166          MOV     R7,@SYDOCB(R2)
0030' EBA2 0040'    1167          SOC     @TAB(R2),@SYDOCB(R2)    MOVE COUNTER TO DOC DATA BUFFER
0034' 0004*
0036' 0642          1168          DECT    R2
0038' 16F5          1169          JNE     30$
                    1170
003A' C2E0 0000'    1171          MOV     @LINKZ,R11
003E' 045B          1172          RT
                    1173  ********************************************************************************
0040' 0000          1174  TAB     DATA    >0000
0042' 0100          1175          DATA    >0100
0044' 0200          1176          DATA    >0200
0046' 0300          1177          DATA    >0300
0048' 0400          1178          DATA    >0400
004A' 0500          1179          DATA    >0500
004C' 0600          1180          DATA    >0600
004E' 0700          1181          DATA    >0700
0050' 0800          1182          DATA    >0800
0052' 0900          1183          DATA    >0900
0054' 0A00          1184          DATA    >0A00
0056' 0B00          1185          DATA    >0B00
0058' 0C00          1186          DATA    >0C00
005A' 0D00          1187          DATA    >0D00
005C' 0E00          1188          DATA    >0E00
005E' 0F00          1189          DATA    >0F00
                    1190  *
                    1191+         LOCR    PRIV,LINKZ
0000' =0002         1194A LINKZ   BSS     2
                    1195
                    1196          END

No errors detected
```

```
                     1          IDT     SUPBFX
                     2          SUBTTL  FILL SUPERFRAME BUFFERS
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: SUPBFX.SRC                    AUTH: N.COSTANTINIDES      *
                     6  * VERSION: 1                          DATE: 30-MAY-1983          *
                     7  *                                                                  *
                     8  * FUNCTION:     THIS ROUTINE WILL FILL THE CALIBRATION AND       *
                     9  *               DOC DATA BUFFERS FROM THE DFDR OUTPUT BUFFER      *
                    10  *               PRIOR TO OUTPUTTING TO WD 64 OF THE DFDR.        *
                    11  *                                                                  *
                    12  * CALLING MODULES: JOB4PS                                         *
                    13  *                                                                  *
                    14  * CALLING SEQ: BL @SUPBFX                                         *
                    15  *                                                                  *
                    16  * INPUTS: NONE                                                    *
                    17  *                                                                  *
                    18  * OUTPUTS: OUTPUT TO CALIBRATION AND DOC DATA BUFFERS             *
                    19  *                                                                  *
                    20  * MODULES REFERENCED:                                             *
                    21  *                                                                  *
                    22  * WORKSPACE AREA: CALLER'S                                        *
                    23  *                                                                  *
                    24  * REGISTERS MODIFIED: R0,R1,R2                                    *
                    25  *                                                                  *
                    26  * VERSION HISTORY:                                                *
                    27  *                                                                  *
                    28  ********************************************************************
       =0000        29          RSECT   SUPBFX
                    30  *** CALL NAME
                    31          INTERN  SUPBFX
                    32  *** VARIABLES REFERENCED
                    33          EXTERN  SYCALB,OPL,GWM,SYDOCB,LATM1,LONGM1,GM
                    34  *** CONSTANTS REFERENCED
                    35  *** TABLES REFERENCED
                    36  *** MODULES REFERENCED
                    37  *** LIBRARY
                    38          INCLUDE ENCLOS
                    40  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    42          INCLUDE REGDEF          REGISTER DEFENITIONS
                    61          INCLUDE CNSTNT          CONSTANTS
                   189          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   490          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   720          INCLUDE JMPMAC          JUMP MACROS
                   754          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   769          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                  1145  *** REGISTERS DEFINITION
                  1146  ********************************************************************
  0000'              1147  SUPBFX
  0000'  C80B 0000'  1148          MOV     R11,@LINKZ              SAVE LINK REGISTER
                     1149
                     1150
```

```
                    102   ********************************************************************************
                    103   *
                    104   *               AN, CAL, BT DATA ACQ VECTORS
0080'               105   ADCBDV
0080'  0013* 0014*  106   ANACQV  DATA    ANW,ANACQ       ANALOG DATA ACQ
0084'  0018* 0019*  107   CALAQV  DATA    CALBTW,CALAQ    CALIBRATION DATA ACQ
0088'  0016* 0017*  108   BTACQV  DATA    CALBTW,BTACQ    BITE DATA ACQ
                    109   *       ARC-TANGENT VECTOR
008C'  001C* 001D*  110   ATANV   DATA    ATW,ATAN
                    111   *
                    112   ********************************************************************************
                    113           END
```

No errors detected

```
                        77  ********************************************************************************
                        78  *
                        79  * XOP VECTORS
    0040'  FFFF FFFF    80  XOPVCT  DATA   -1,-1   XOP 0 = MICRO TERMINAL OUTPUT
    0044'  FFFF FFFF    81          DATA   -1,-1   XOP 1 = MICRO TERMINAL INPUT
    0048'  0012* 0011*  82          DATA   SYSW,SYSER      XOP 2 = SET ERROR STATUS WORD
    004C'  0016* 0015*  83          DATA   DC1W,DC1ACQ     XOP 3 = DISCRETE DATA ACQ
    0050'  FFFF FFFF    84          DATA   -1,-1           XOP ??
    0054'  FFFF FFFF    85          DATA   -1,-1           XOP ??
    0058'  FFFF FFFF    86          DATA   -1,-1           XOP ??
    005C'  FFFF FFFF    87          DATA   -1,-1
    0060'  FFFF FFFF    88          DATA   -1,-1   XOP 8 = WRITE 1 HEX DIGIT
    0064'  FFFF FFFF    89          DATA   -1,-1   XOP 9 = HEX # INPUT
    0068'  FFFF FFFF    90          DATA   -1,-1   XOP 10 = HEX # OUTPUT
    006C'  FFFF FFFF    91          DATA   -1,-1   XOP 11 = ECHO
    0070'  FFFF FFFF    92          DATA   -1,-1   XOP 12 = WRITE CHARACTER
    0074'  FFFF FFFF    93          DATA   -1,-1   XOP 13 = READ CHARACTER
    0078'  FFFF FFFF    94          DATA   -1,-1   XOP 14 = MESSAGE OUT
    007C'  FFFF FFFF    95          DATA   -1,-1   XOP 15
                        96  *
                        97  *
                        98  *
                        99  *
                       100  ********************************************************************************
```

```
                        38
                        39  *                       INTERRUPT VECTORS
                        40  *                       ----------------
                        41
0000'                   42  INTR
0000'  0002* 0001*      43  IN0VCT  DATA    PONW,PON                POWER-ON
                        44  *
0004'  0004* 0003*      45  IN1VCT  DATA    PDNW,POWDN      POWER-DOWN
                        46  *
0008'  001B* 001A*      47  IN2VCT  DATA    NICHTW,NICHTS   4S FRAM
                        48  *
000C'  0006* 0005*      49  IN3VCT  DATA    RTW,RTISR       REAL TIME
                        50  *
0010'  001B* 001A*      51  IN4VCT  DATA    NICHTW,NICHTS   AUX
                        52  *
0014'  0008* 0007*      53  IN5VCT  DATA    DFW,DFISR       DFDR
                        54  *
0018'  000A* 0009*      55  IN6VCT  DATA    ADCW,ADCISR     ADC DATA READY
                        56  *
001C'  000F* 0010*      57  IN7VCT  DATA    WPIC,ICISR      INTER CPU
                        58  *
0020'  001B* 001A*      59  IN8VCT  DATA    NICHTW,NICHTS   DFDR PB SYNC
                        60  *
0024'  001B* 001A*      61  IN9VCT  DATA    NICHTW,NICHTS   DFDR PB WORD
                        62  *
0028'  000C* 000B*      63  INAVCT  DATA    DR1W,DR1ISR     DITS#1
                        64  *
002C'  000E* 000D*      65  INBVCT  DATA    DR2W,DR2ISR     DITS#2
                        66  *
0030'  001B* 001A*      67  INCVCT  DATA    NICHTW,NICHTS   SPARE
                        68  *
0034'  001B* 001A*      69  INDVCT  DATA    NICHTW,NICHTS   SPARE
                        70  *
0038'  001B* 001A*      71  INEVCT  DATA    NICHTW,NICHTS     -
                        72  *
003C'  001B* 001A*      73  INFVCT  DATA    NICHTW,NICHTS   EXTERNAL PRINTER 9902
                        74  *
                        75  ********************************************************************************
```

```
                     1          IDT     'ROMV'
                     2
                     3          SUBTTL  'INTERRUPT, XOP AND OTHER VECTORS, AND WORKSPACE AREA'
                     4  *
                     5  *       FILE:   ROMV.SRC
                     6  *       DATE:   17-NOV-1982
          =0000      7          RSECT   ROMV
                     8  *
                     9  *
                    10  *
                    11  ********************************************************************************
                    12  *
                    13          INTERN  ADCBDV,ANACOV,CALAQV,BTACQV
                    14          INTERN  ATANV
                    15          INTERN  INOVCT
                    16  ********************************************************************************
                    17  *
                    18  *
                    19          EXTERN  PON,PONW
                    20          EXTERN  POWDN,PDNW
                    21          EXTERN  RTISR,RTW
                    22          EXTERN  DFISR,DFW
                    23          EXTERN  ADCISR,ADCW
                    24          EXTERN  DR1ISR,DR1W
                    25          EXTERN  DR2ISR,DR2W
                    26          EXTERN  WPIC,ICISR
                    27          EXTERN  SYSER,SYSW
                    28          EXTERN  ANW,ANACQ
                    29          EXTERN  DC1ACQ,DC1W
                    30          EXTERN  BTACQ,CALBTW
                    31          EXTERN  CALAQ
                    32          EXTERN  NICHTS,NICHTW
                    33          EXTERN  ATW,ATAN
                    34
                    35  *
                    36  ********************************************************************************
```

```
00CE' 000A* 000B*  186  JBTBL  DATA  WPJB,ICMD      1ST JOB, RECEIVE MSG DISPATCHER (BY ICRC)
00D2' 000A* 000D*  187         DATA  WPJB,JOB8PS    2ND JOB, 8 TIMES PER SEC JOB (BY RTI)
00D6' 000A* 000C*  188         DATA  WPJB,JOB4PS    3RD JOB, 4 TIMES PER SEC JOB (BY RTI)
                   189  *
                   190  *** DFDR PLAYBACK DATA BUFFER START ADDRESS
                   191  *
00DA' 000E*        192  DFFLPB DATA  DFBUF1         START ADDRESS OF BUFF #1
00DC' 000F*        193         DATA  DFBUF2         START ADDRESS "   "   #2
                   194  *
                   195  *** DFDR SYNC CODE TABLE
                   196  *
00DE' 0247         197  DFSYNT DATA  >247           SUBFRAME #1
00E0' 05B8         198         DATA  >5B8           SUBFRAME #2
00E2' 0A47         199         DATA  >A47           SUBFRAME #3
00E4' 0DB8         200         DATA  >DB8           SUBFRAME #4
                   201  ********************************************************************************
                   202  *** SYSTEM ERROR TABLE
                   203  *
                   204  *       THE TABLE DEFINES THE ERROR TYPE CODE FOR THE SYSTEM
                   205  *       ERROR CODE FOR EACH WORD LOCATED IN THE SYSTEM ERROR BUFFER.
                   206  *       IT ALSO SPECIFIES THE SPARE WORDS.
                   207  *
                   208  *              ERR TYPE     FOR ERROR WORD OF SYST ERR BUFF(SYERF)
                   209  *
00E6' 0000         210  SYETBL DATA  >0000          1
00E8' 0100         211         DATA  >0100          2
00EA' 0200         212         DATA  >0200          3
00EC' 0200         213         DATA  >0200          4
00EE' 0400         214         DATA  >0400          5
00F0' 0500         215         DATA  >0500          6
00F2' FFFF         216         DATA  >FFFF          7      NO ERROR CODE
00F4' FFFF         217         DATA  >FFFF          8      SPARE
00F6' 0800         218         DATA  >0800          9
00F8' 0900         219         DATA  >0900          10
00FA' 0A00         220         DATA  >0A00          11
00FC' FFFF         221         DATA  >FFFF          12     SPARE
00FE' FFFF         222         DATA  >FFFF          13     SPARE
0100' FFFF         223         DATA  >FFFF          14     SPARE
0102' FFFF         224         DATA  >FFFF          15     SPARE
0104' FFFF         225         DATA  >FFFF          16     SPARE
                   226  *
                   227  ********************************************************************************
                   228  * EAROM
0106' 0010         229  EATC   DATA  16            250 MSEC CONSTANT FOR EATMR (250/15.625)
                   230  *
                   231
                   232         END
```

No errors detected

```
                    138  *
                    139  *       VERSION-TABLE
0088'               140  VRTBL
                    141  *               VERSION #1 : B767 P&W (NOT USED)
0088'  FFFF FFFF    142          DATA    -1,-1,-1,-1             DR1-4 SPEED+MODE (DR3, DR4 NOT USED)
008C'  FFFF FFFF
                    143
                    144  *               VERSION #2 : 767 G.E.
0090'  0000 00C0    145          DATA    0,>C0,-1,-1             DR1-4 SPEED+MODE (DR3, DR4 NOT USED)
0094'  FFFF FFFF
                    146
                    147  *               VERSION #3 : B757 R.R. (NOT USED)
0098'  FFFF FFFF    148          DATA    -1,-1,-1,-1             DR1-4 SPEED+MODE (DR3, DR4 NOT USED)
009C'  FFFF FFFF
                    149
                    150  *               VERSION #4 : B757 P&W (NOT USED)
00A0'  FFFF FFFF    151          DATA    -1,-1,-1,-1             DR1-4 SPEED+MODE (DR3, DR4 NOT USED)
00A4'  FFFF FFFF
                    152  *
                    153  **********************
                    154  *
                    155  *               OUTPUT BUFFERS ADDRESS TABLE
                    156  *
00A8'  0001*        157  OUTBAT  DATA    DFDQ            DFDR OUTPUT QUEUE ADDR
00AA'  0004*        158          DATA    ICOB1A          INT-CPU O/P     FLIP/FLOP BUFFER
00AC'  0002*        159          DATA    AXOB            AUX OUTPUT BUFFER ADDR
00AE'  0003*        160          DATA    DX10B           DITS #1 OUTPUT BUFFER ADDR
                    161  *
                    162  *   INTER-CPU OUTPUT BUFFER START ADDRESS TABLES
                    163  *   TO ACQUIRE DATA REF BY RTIFR, USED BY DSTIME, INDEXED BY ICOBAI
                    164  *
00B0'  0004*        165  ICOBAA  DATA    ICOB1A          START ADDRESS OF AIDS #1 BUFFER
00B2'  0005*        166          DATA    ICOB2A            "     "     "  "  #2   "
00B4'  0006*        167  ICOBAB  DATA    ICOB1B          START ADDRESS OF DFDR O/P #1 BUFFER
00B6'  0007*        168          DATA    ICOB2B            "     "         "    "  #2    "
00B8'  0008*        169  ICOBAC  DATA    ICOB1C          START ADDRESS OF DFDR P/B #1 BUFFER
00BA'  0009*        170          DATA    ICOB2C            "     "     "   "    "  #2    "
                    171  *
                    172  *   SENDING DATA TO AIDS CPU, REF BY RTIFR, USED BY ICOMSG
                    173  *
00BC'  00C0'        174  ICOBKA  DATA    ICOBK1                  IST BLOCK ADDRESS
00BE'  00C6'        175          DATA    ICOBK2                  2ND   "     "
00C0'  0006*        176  ICOBK1  DATA    ICOB1B                  START ADDRESS OF IST BUFFER
00C2'  0000         177          DATA    0                       MSG TYPE
00C4'  00F8         178          DATA    173+8+1+2+64            NUMBER OF DATA WORDS
00C6'  0007*        179  ICOBK2  DATA    ICOB2B                  START ADDRESS OF 2ND BUFFER
00C8'  0000         180          DATA    0                       MSG TYPE
00CA'  00F8         181          DATA    173+8+1+2+64            NUMBER OF DATA WORDS
                    182  *
                    183  *** LIST OF JOBS EXECUTED BY EXTIVE
                    184  *
00CC'  0002         185  JBMAX   DATA    2               NO. OF JOBS LESS 1
```

```
004B' 0020          86          DATA    >20
004A' 0040          87          DATA    >40
004C' 0080          88          DATA    >80
004E' 0100          89          DATA    >100
0050' 0200          90          DATA    >200
0052' 0400          91          DATA    >400
0054' 0800          92          DATA    >800
0056' 1000          93          DATA    >1000
0058' 2000          94          DATA    >2000
005A' 4000          95          DATA    >4000
005C' 8000          96          DATA    >8000
                    97  *
                    98  *** MASK TABLE
                    99  *
005E' FFFE         100  DAINTB  DATA    >FFFE
0060' FFFC         101          DATA    >FFFC
0062' FFF8         102          DATA    >FFF8
0064' FFF0         103          DATA    >FFF0
0066' FFE0         104          DATA    >FFE0
0068' FFC0         105          DATA    >FFC0
006A' FF80         106          DATA    >FF80
006C' FF00         107          DATA    >FF00
006E' FE00         108          DATA    >FE00
0070' FC00         109          DATA    >FC00
0072' F800         110          DATA    >F800
0074' F000         111          DATA    >F000
0076' E000         112          DATA    >E000
0078' C000         113          DATA    >C000
007A' 8000         114          DATA    >8000
007C' 0000         115          DATA    >0000
                   116  *
                   117  ********************************************************************************
                   118  *
                   119  *              ALL PARAMETERS
                   120  *              === ==========
                   121  ********************
                   122  *
                   123  *       SYNC CODES
                   124  *
007E'              125  SYCODT
007E' 0247         126          DATA    >247    SYNC-CODE FOR SF 1 (1107 OCT)
0080' 05B8         127          DATA    >5B8    SYNC-CODE FOR SF 2 (2670 OCT)
0082' 0A47         128          DATA    >A47    SYNC-CODE FOR SF 3 (5107 OCT)
0084' 0DB8         129          DATA    >DB8    SYNC-CODE FOR SF 4 (6670 OCT)
                   130  *
                   131  ********************
                   132  *
                   133  *       2 OR 3 MAN-CREW FLAG
                   134  *
0086' 0003         135  MNCRWF  DATA    3               NOW 3-MAN !!
                   136  *
                   137  ********************
```

```
                    34  ********************************************************************************
                    35  *
                    36  *                       COMMON
                    37  *                       ======
                    38  *
                    39  **********************
                    40  *
                    41  *
                    42  *               CONSTANTS
                    43  *               =========
0000' 0000          44  D0      DATA    0
0002' 0001          45  D1      DATA    1
0004' 0002          46  D2      DATA    2
0006' 0003          47  D3      DATA    3
0008' 0004          48  D4      DATA    4
000A' 0005          49  D5      DATA    5
000C' 0007          50  D7      DATA    7
000E' 0009          51  D9      DATA    9
0010' 000A          52  D10     DATA    10
0012' 000E          53  D14     DATA    14
0014' 000F          54  D15     DATA    15
0016' 0010          55  D16     DATA    16
0018' 001D          56  D29     DATA    29
001A' FFFF          57  DMINUS  DATA    -1
001C' 0001          58  B0      DATA    >1
001E' 0002          59  B1      DATA    >2
0020' 0004          60  B2      DATA    >4
0022' 0008          61  B3      DATA    >8
0024' 0010          62  B4      DATA    >10
0026' 0020          63  B5      DATA    >20
0028' 0040          64  B6      DATA    >40
002A' 0080          65  B7      DATA    >80
002C' 0100          66  B8      DATA    >100
002E' 0200          67  B9      DATA    >200
0030' 0400          68  B10     DATA    >400
0032' 0800          69  B11     DATA    >800
0034' 1000          70  B12     DATA    >1000
0036' 2000          71  B13     DATA    >2000
0038' 4000          72  B14     DATA    >4000
003A' 8000          73  B15     DATA    >8000
003C' 003F          74  X3F     DATA    >003F
003E' 00FF          75  XFF     DATA    >00FF
                    76  *
                    77  *
                    78  ********************************************************************************
                    79  *
                    80  *** SIGN TABLE
                    81  *
0040' 0002          82  DRSGTB  DATA    >2
0042' 0004          83          DATA    >4
0044' 0008          84          DATA    >8
0046' 0010          85          DATA    >10
```

```
                        1          IDT     ROMD
                        2
                        3          SUBTTL  'DATA AREA'
                        4
                        5  *       FILE:   ROMD.SRC
                        6  *       DATE:   17-MAY-1983
                        7
          =0000         8          RSECT   ROMD
                        9  ********************************************************************************
                       10          INTERN  VRTBL,MNCRWF
                       11          INTERN  OUTBAT
                       12          INTERN  ICOBAA,ICOBAB,ICOBAC,ICOBKA,ICOBK1,ICOBK2
                       13          INTERN  JBMAX,JBTBL
                       14          INTERN  DRSGTB,DAIMTB
                       15          INTERN  DPSYNT,DPFLPB
                       16  *
                       17          INTERN  D3,D5,D4,D1,D2,D0,D29,D7,D9,D10,D14,D16,D15,DMINUS
                       18          INTERN  B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15
                       19          INTERN  SYCODT,X3F,XFF,EATC
                       20          INTERN  SYETBL
                       21
                       22
                       23  *
                       24  ********************************************************************************
                       25  *
                       26          EXTERN  DF0D,AX0B,DX10B,ICOB1A
                       27          EXTERN  ICOB2A,ICOB1B,ICOB2B,ICOB1C,ICOB2C
                       28          EXTERN  WPJB,ICMD,JOB4FS,JOB8FS
                       29          EXTERN  DPBUF1,DPBUF2
                       30  *
                       31  ********************************************************************************
                       32  *
```

```
                    521  *                                                                       *
                    522  *************************************************************************
                    523  *
                    524  *              GENRAL
                    525  *
0D64'  =0002        526  SSEG    BSS     2               ENGINE CONFIG
                    527  *
0D66'  =0002        528  CLRFLG  BSS     2               EAROM CLEAR FLAG
0D68'  =0002        529  EAFLG   BSS     2               EAROM FLAG
0D6A'  =0002        530  BITCTR  BSS     2               BITE COUNTER
0D6C'  =0002        531  CALCTR  BSS     2               CAL COUNTER
0D6E'  =0002        532  SYNFLG  BSS     2               SYNCRO CAL FLAG
                    533  ********************************************
                    534  *
                    535  *
                    536  *       END OF RAM USED BY POWER ON ISR
0D70'               537  RAMEND
                    538          END
```

No errors detected

```
                    469  *
                    470  *              DITS # 1 OUTPUT BUFFER
                    471  *
09FA'  =0080        472  DX1OB   BSS     128
                    473  *
                    474  ********************************************
                    475  *
                    476  *              DITS # 2 OUTPUT BUFFER
                    477  *
0A7A'  =0080        478  DX2OB   BSS     128
                    479  *
                    480  ********************************************************************************
                    481  *
                    482  *  DFDR PLAYBACK BUFFER
                    483  *
0AFA'  =0080        484  DFBUF1  BSS     64*2    BUFFER 1
0B7A'  =0080        485  DFBUF2  BSS     64*2    BUFFER 2
0BFA'  =0002        486  DFFBAI  BSS     2
                    487  *                                                                              *
                    488  ********************************************************************************
                    489  *
                    490  *              CALIBRATION BUFFERS
                    491  *              ----------- -------
                    492  *
                    493  ********************************************
                    494  *
                    495  *              SIG. GND. CAL. BUFFER
                    496  *
0BFC'  =005A        497  VHOSBF  BSS     90
                    498  *
                    499  ********************************************
                    500  *
                    501  *              REF. GND. CAL. BUFFER
                    502  *
0C56'  =005A        503  VHORBF  BSS     90
                    504  *
                    505  ********************************************
                    506  *
                    507  *              SIG. CAL FACTOR BUFFER
                    508  *
0CB0'  =005A        509  KSBUF   BSS     90
                    510  *
                    511  ********************************************
                    512  *
                    513  *              REF. CAL. FACTOR BUFFER
                    514  *
0D0A'  =005A        515  KRBUF   BSS     90
                    516  *
                    517  ********************************************
                    518  *
                    519  *
                    520  ********************************************************************************
```

```
041A'  =0002        417  SYCW0   BSS    2                          W81
041C'  =0002        418  SYCW5   BSS    2                          W82
041E'  =0002        419  SYCW25  BSS    2                          W83
                    420  *
                    421  *      DOC DATA
                    422  *
0420'               423  SYDOCB
0420'  =0002        424  SYDAY   BSS    2                          W84
0422'  =0002        425  SYDMTH  BSS    2                          W85
0424'  =0002        426  SYDFMH  BSS    2                          W86
0426'  =0002        427  SYDFLH  BSS    2                          W87
0428'  =0002        428  SYDFLL  BSS    2                          W88
042A'  =0002        429  GWH     BSS    2                          W89
042C'  =0002        430  GWL     BSS    2                          W90
042E'  =0002        431  LATM1   BSS    2                          W91
0430'  =0002        432  LATM2   BSS    2                          W92
0432'  =0002        433  LATL    BSS    2                          W93
0434'  =0002        434  LONGM1  BSS    2                          W94
0436'  =0002        435  LONGM2  BSS    2                          W95
043B'  =0002        436  LONGL   BSS    2                          W96
043A'  =0002        437  ACIDNT  BSS    2                          W97
       =0002        438          BSS    2                          W98
       =0002        439          BSS    2                          W99
       =0002        440          BSS    2                          W100
0442'  =0002        441  EABTF   BSS    2      FOR EAROM CLEAR     W101
                    442  ************************************************
                    443  * INTER-CPU COMMUNICATION O/P BUFFERS
                    444  * LISTED IN ICOBAA,ICOBAB,ICOBAC
                    445  *      FIRST BUFFER
       =0004        446          BSS    4      MSG TYPE, NO OF DATA WORDS
0448'  =0080        447  ICOB1B  BSS    64*2          DFDR   OUTPUT DATA
04C8'  =0190        448  ICOB1A  BSS    200*2         AIDS DATA
       =0004        449          BSS    2*2           SPARE
       =0002        450          BSS    2             SUM CHECK
065E'  =0080        451  ICOB1C  BSS    64*2          PB
                    452  *      SECOND BUFFER
       =0004        453          BSS    4      MSG TYPE, NO OF DATA WORDS
06E2'  =0080        454  ICOB2B  BSS    64*2          DFDR O/P DATA
0762'  =0190        455  ICOB2A  BSS    200*2         AIDS DATA
       =0004        456          BSS    2*2           SPARE
       =0002        457          BSS    2             SUM CHECK
08FB'  =0080        458  ICOB2C  BSS    64*2          PB
                    459  *
0978'  =0002        460  ICOER   BSS    2      INT-CPU O/P QUE FULL ERROR COUNTER
                    461  *
                    462  ************************************************
                    463  *
                    464  *              AUX OUTPUT BUFFER
                    465  *
097A'  =0080        466  AXOB    BSS    128
                    467  *
                    468  ************************************************
```

```
                    365  *
02AC'  =0002        366  OUTBB   BSS    2       ADDRESS DFOQ FOR DFDR OUTPUTS (ONE BUFFER)
02AE'  =0002        367  OUTBB1  BSS    2       ADDRESS ICOB1A/ICOB2A FOR INT-CPU OUTPUTS (FLIP/FLOP BUFF)
       =0002        368          BSS    2       ADDRESS SPARE
       =0002        369          BSS    2       ADDRESS SPARE
                    370  *                                                                    *
                    371  ************************************************************************
                    372  *  SYSTEM ERROR BUFFER
02B4'  =0010        373  SYEBF   BSS    8*2     CPU 1 ERROR BIT WORD
02C4'  =0010        374  SYEBF2  BSS    8*2     CPU 2 ERROR BIT WORD
                    375  ************************************************************************
                    376  *       EAROM BUFFERS
                    377  * EATSBF : ERROR CODE WAITING TO BE STORED IN EAROM
02D4'  =0020        378  EATSBF  BSS    16*2    TEMP EAROM SYST STATUS BUFFER
                    379  * EAIMBF : 1ST 16 WORDS FOR SYSTEM STATUS BUFFER
02F4'  =0080        380  EAIMBF  BSS    64*2    IMAGE OF EAROM MEMORY BUFFER
                    381  *EASADR : PRESENT EAROM ADDR  FOR SYST STATUS BUFFER IN EAROM
0374'  =0002        382  EASADR  BSS    2       PRESENT SYSTEM STATU'S EAROM ADDRESS (0 TO 15)
                    383  *
0376'  =0002        384  EATBUF  BSS    2       EAROM CLEAR DISCRETE BUFFER
0378'  =0002        385  EACFLG  BSS    2       EAROM CLEAR FLAG
                    386  *                                                                    *
                    387  ************************************************************************
                    388  *
                    389  *               OUTPUT BUFFERS
                    390  *               ------ -------
                    391  *
                    392  ************************************************
                    393  *
                    394  *               DFDR OUTPUT QUEUE
                    395  *
037A'  =0080        396  DFOQ    BSS    128
03FA'  =0002        397  VCPR    BSS    2       5V. POT. REF. (AS WORD 65)
       =0002        398          BSS    2       DUMMY (WORD 66)
03FE'  =0002        399  DFBTF   BSS    2       DFDR BITE DISCRETE 4 SPS (WORD 67 BIT 0 )
                    400  *
                    401  *       CALIBRATION DATA
                    402  *
0400'               403  SYCALB
0400'  =0002        404  OPL     BSS    2       OIL PRESS L             W68
0402'  =0002        405  OPR     BSS    2       OIL PRESS R             W69
0404'  =0002        406  OTL     BSS    2       OIL TEMP L              W70
0406'  =0002        407  OTR     BSS    2       OIL TEMP R              W71
0408'  =0002        408  OQL     BSS    2       OIL QTY L               W72
040A'  =0002        409  OQR     BSS    2               R               W73
040C'  =0002        410  VIBL    BSS    2                               W74
040E'  =0002        411  VIBR    BSS    2                               W75
0410'  =0002        412  N2L     BSS    2                               W76
0412'  =0002        413  N2R     BSS    2                               W77
0414'  =0002        414  APURPM  BSS    2                               W78
0416'  =0002        415  APUEGT  BSS    2                               W79
0418'  =0002        416  ZERO1   BSS    2                               W80
```

```
025A'  =0002        313  BPSTF   BSS    2
                    314  *
                    315  ********************************************************************************
                    316  *                                                                              *
                    317  ********************************************************************************
                    318  *
                    319  *              SYSTEM BUFFERS
                    320  *              ------ -------
                    321  *
                    322  ***********************************************
                    323  *
                    324  *              SYSTEM STATUS BUFFER 16 WORDS.
                    325  *
025C'  =0020        326  SYSTWB  BSS    32             16 STATUS WORDS
                    327  *
                    328  ***********************************************
                    329  *
                    330  *              SYSTEM CAL BUFFER 16 WORDS.
                    331  *SYCALB
                    332  *SYCMU  BSS    24             12 WORD PARAMETER DATA
                    333  *ZERO1  BSS    2
                    334  *SYCW0  BSS    2              SYSTEM CAL WORD FOR GND
                    335  *SYCW5  BSS    2              SYSTEM CAL WORD FOR 5 V
                    336  *SYCW25 BSS    2              SYSTEM CAL WORD FOR 25 V
                    337  *
                    338  ***********************************************
                    339  *
                    340  *              SYSTEM DOC BUFFER 16 WORDS.
                    341  *SYDOCB
                    342  *SYDAY  BSS    2              DAY
                    343  *SYDMTH BSS    2              MONTH
                    344  *SYDFMH BSS    2              FLIGHT # MS-H
                    345  *SYDFLH BSS    2              FLIGHT # LS-H
027C'  =0002        346  GM      BSS    2              GROSS WT MSH
027E'  =0002        347  GL      BSS    2              GROSS WT LSH
0280'  =0002        348  LAT1    BSS    2              LAT MSH 1
0282'  =0002        349  LAT2    BSS    2              LAT MSH 2
0284'  =0002        350  LAT     BSS    2              LAT LSH
0286'  =0002        351  LONG1   BSS    2              LONG MSH 1
0288'  =0002        352  LONG2   BSS    2              LONG MSH 2
028A'  =0002        353  LONG    BSS    2              LONG LSH
       =0002        354          BSS    2              A/C IDENT
       =0006        355          BSS    3*2            SPARE
                    356  *
                    357  ************************************************************
                    358  * EXECUTIVE BUFFER
                    359  *
0294'  =0018        360  JBBBUF  BSS    6*4     ROOM FOR 6 JOBS (4 BYTES PER JOB)
02AC'                361  JBEBUF
                    362  ********************************************************************************
                    363  *
                    364  *** LIST OF DESTINATION BUFFER USED BY DSTINE
```

```
          =000A        261          BSS     10      R11-R15
                       262  ***********************************************
                       263  * BACKGROUND JOB WORKSPACE POINTER
                       264  *
    01D0' =000C        265  WPJB    BSS     6*2     R0-R5 SCRATCH
          =0002        266          BSS     2       R6
          =0002        267          BSS     2       R7
          =0002        268          BSS     2       R8
    01E2' =0002        269  JB4CYL  BSS     2       R9      JOB4PS COUNTER (0 TO 3 RANGE)
    01E4' =0002        270  RSCTR   BSS     2       R10     READ SWITCH COUNTER (SYSOUT)
          =000A        271          BSS     5*2     R11-R15
                       272  ***********************************************************************
                       273  *       BACKGROUND JOB WORKSPACE AREA
                       274  *       FOR EAROM (EAMON,EAWRIT)
    01F0' =0008        275  WPJB1   BSS     4*2     R0-R3   SCRATCH
          =0002        276          BSS     2       R4      SCRATCH
          =0002        277          BSS     2       R5      SCRATCH
    01FC' =0002        278  EATMR   BSS     2       R6      EAROM TIMER
    01FE' =0002        279  EAWDP   BSS     2       R7      WRITE DATA WORD POINTER
    0200' =0002        280  EAADR   BSS     2       R8      ADDRESS
    0202' =0002        281  EACNT   BSS     2       R9      CONTROL REG IMAGE
    0204' =0002        282  EAMODE  BSS     2       R10     MODE
          =000A        283          BSS     5*2     R11-R15
                       284
                       285  *
                       286  ***********************************************
                       287  *
                       288  *       NICHTS (NO NEED FOR R0-R12)
          =01F6'       289  NICHTW  EQU     $-26
    0210' =0006        290  NICHTR  BSS     6       R13-R15
                       291  *
                       292  ***********************************************
                       293  *
                       294  *       BITE & CALIBRATION
    0216' =000C        295  CALBTW  BSS     12      R0-R5
    0222' =0002        296  CALAP   BSS     2       CAL-MUX-ADDR-PTR (R6)
    0224' =0002        297  CALSN   BSS     2       CAL-SLOT # (R7)
    0226' =0002        298  CNVFLG  BSS     2       CONV-FLAG (R8)
    0228' =0002        299  CBR9    BSS     2       R9
    022A' =0002        300  BTADP   BSS     2       BT-ADDR-PTR (R10)
    022C' =000A        301  CALBTR  BSS     10      R11-R15
                       302  *
                       303  ***********************************************
                       304  *
                       305  ***********************************************
                       306  *
                       307  *       ARC-TAN "ATAN"
    0236' =0020        308  ATW     BSS     32
                       309  *
                       310  ***********************************************
                       311  *
    0256' =0004        312  BGW     BSS     4
```

```
                    209  ********************************************************************************
                    210  *                                                                              *
                    211  ********************************************************************************
                    212  *
                    213  *               XOP WORK-SPACEES
                    214  *
                    215  ***********************************************
                    216  *
                    217  *               SYSER/XOP 2 (SET ERROR STATUS)
                    218  *               (USING ONLY R9-R15)
       =0150'       219  SYSW    EQU     0-18
 0162' =0002        220  SYSR9   BSS     2               R9
 0164' =0002        221  SYSTBO  BSS     2               SYSTEM STATUS BUFFER OFFSET (R10)
 0166' =000A        222  SYSR    BSS     10              R11-R15
                    223  *
                    224  ********************************************************************************
                    225  *
                    226  *               DISCRETE #1 DA (XOP 3)
 0170' =000C        227  DC1W    BSS     12      R0-R5
 017C' =0002        228  DC1FLG  BSS     2       DC1-DA FLAG (R6)
 017E' =0002        229  DC1OA   BSS     2       DC1-TABLES-OFFSET-ARRAY (R7)
 0180' =0002        230  DC1PCNT BSS     2       DC1-PARAM-COUNT (R8)
 0182' =0002        231  DC1TOF  BSS     2       DC1-TABLES-OFFSET (R9)
 0184' =0002        232  DC1ARO  BSS     2       DC1-ARRAY-OFFSET (R10)
 0186' =000A        233  DC1R    BSS     10      R11-R15
                    234  *
                    235  *
                    236  ********************************************************************************
                    237  ********************************************************************************
                    238  *
                    239  *       OTHER WORK-SPACES
                    240  *
                    241  ***********************************************
                    242  *               ANALOG DA
                    243  *
 0190' =000E        244  ANW     BSS     14      R0-R6
 019E' =0002        245  ANOA    BSS     2       AN-TABLES-OFFSET-ARRAY (R7)
 01A0' =0002        246  ANPCNT  BSS     2       AN-PARAM-COUNT (R8)
 01A2' =0002        247  ANTOF   BSS     2       AN-TABLES-OFFSET (R9)
 01A4' =0002        248  ANARO   BSS     2       AN-ARRAY-OFFSET (R10)
 01A6' =000A        249  ANR     BSS     10      R11-R15
                    250  ****************************************************************
                    251  * EXECUTIVE WORKSPACE AREA FOR EXTIVE,JBQUE,MSCHK
                    252  *
 01B0' =0002        253  WPEX    BSS     2       R0
       =000A        254          BSS     10      R1-R5
 01BC' =0002        255  MSERR   BSS     2       R6      MEMORY SUMCHK ERROR
                    256  *                               B0(LSB)=1ST K BLOCK.....B15(MSB)=16TH K BLOCK
 01BE' =0002        257  MSIX    BSS     2       R7      MEMORY SUM INDEX(OFOR1STK,2FOR2NDK,..) IN MSCHK
 01C0' =0002        258  JBIPTR  BSS     2       R8      JOB QUEUE INPUT POINTER USED IN JBQUE
 01C2' =0002        259  JBOPTR  BSS     2       R9      JOB QUEUE OUTPUT POINTER USED IN JBQUE
       =0002        260          BSS     2       R10
```

```
00F6' =000A        157  DR2R   BSS   10     R11-R15
                   158  *
                   159  ***********************************************
                   160  *
                   161  *              DITS #3 (INT 10)
0100' =000E        162  DR3W   BSS   14     R0-R6
010E' =0002        163  DR3OA  BSS   2      DR3-TABLES-OFFSET-ARRAY (R7)
0110' =0002        164  DR3PCT  BSS   2      DR3-PARAM-CNT (R8)
0112' =0002        165  DR3TO   BSS   2      DR3-TABLES-OFFSET (R9)
0114' =0002        166  DR3ARO  BSS   2      DR3-ARRAY-OFFSET (R10)
0116' =000A        167  DR3R   BSS   10     R11-R15
                   168  *
                   169  ***********************************************
                   170  *
                   171  *              429 #1 AND 2 (INT 11)
                   172  *
0120' =0002        173  DWXF   BSS   2      DW-TX INT FLAG
                   174  *
                   175  ***********************************************
                   176  *
                   177  *              PLAY-BACK FRAME SYNC (INT 12)
0122' =0008        178  DFSYN  BSS   8      R0-R3
012A' =0002        179  DFFLFA BSS   2      R4
012C' =0002        180  DFWD64 BSS   2      R5
012E' =0002        181  DFSNOI BSS   2      R6
0130' =0002        182  DFBUFF BSS   2      R7
      =0010        183        BSS   16     R8-R15
                   184  *
                   185  ***********************************************
                   186  *
                   187  *              PLAY-BACK WORD SYNC (INT 13)
0142' =0008        188  DFWRD  BSS   8      R0-R3
014A' =0002        189  DFWNOI BSS   2      R4
014C' =0002        190  DFSCTR BSS   2      R5
014E' =0002        191  DFSCTI BSS   2      R6
0150' =0002        192  DFBUFP BSS   2      R7
0152' =0002        193  DFBUFI BSS   2      R8
0154' =0002        194  DFMODE BSS   2      R9
0156' =0002        195  DFWORD BSS   2      R10
      =000A        196        BSS   10     R11-R15
                   197  *
                   198  ***********************************************
                   199  *
                   200  *              MILTOP PRINTER (R14)
                   201  * (NOT YET)
                   202  *
                   203  ***********************************************
                   204  *
                   205  *              EXTERNAL PRINTER 9902 (INT 15)
                   206  * (NOT YET)
                   207  *
                   208  *
```

```
          =0002        105          BSS     2              (R10)
    0076' =000A        106  RTR     BSS     10             R11-R15
                       107  *
                       108  ***********************************************
                       109  *
                       110  *              AUX (INT 4)
                       111  * (NOT YET)
                       112  *
                       113  ***********************************************
                       114  *
                       115  *              DFDR (INT 5)
    0080' =0010        116  DFW     BSS     16      R0-R7
    0090' =0002        117  DFLWR   BSS     2       DF-LAST-WORD (R8)
    0092' =0002        118  DFWRC   BSS     2       DF-WORD-CNT (R9)
    0094' =0002        119  DFAILC  BSS     2       DF-FAIL-CNT (R10)
    0096' =000A        120  DFR     BSS     10      R11-R15
                       121  *
                       122  ***********************************************
                       123  *
                       124  *              ADC (INT 6)
    00A0' =000E        125  ADCW    BSS     14      R0-R6
    00AE' =0002        126  ADCR7   BSS     2       R7
    00B0' =0002        127  ADCR8   BSS     2       R8
    00B2' =0002        128  DASFLG  BSS     2       DAS-FLAG (R9)
    00B4' =0002        129  ADCR10  BSS     2       R10
    00B6' =000A        130  ADCR    BSS     10      R11-R15
                       131  *
                       132  ***********************************************
                       133  *
                       134  *              INTER CPU (INT 7)
                       135  * (NOT YET)
                       136  *
                       137  ***********************************************
                       138  *
                       139  *              DITS #1 (INT 8)
    00C0' =000C        140  DR1W    BSS     12      R0-R5
    00CC' =0002        141  DR1UP   BSS     2       DR1-DITS PORT UPDATE STATUS
    00CE' =0002        142  DR1OA   BSS     2       DR1-TABLES-OFFSET-ARRAY (R7)
    00D0' =0002        143  DR1PCT  BSS     2       DR1-PARAM-CNT (R8)
    00D2' =0002        144  DR1TO   BSS     2       DR1-TABLES-OFFSET (R9)
    00D4' =0002        145  DR1ARO  BSS     2       DR1-ARRAY-OFFSET (R10)
    00D6' =000A        146  DR1R    BSS     10      R11-R15
                       147  *
                       148  ***********************************************
                       149  *
                       150  *              DITS #2 (INT 9)
    00E0' =000C        151  DR2W    BSS     12      R0-R5
    00EC' =0002        152  DR2UP   BSS     2       DR2-DITS PORT UPDATE STATUS (R6)
    00EE' =0002        153  DR2OA   BSS     2       DR2-TABLES-OFFSET-ARRAY (R7)
    00F0' =0002        154  DR2PCT  BSS     2       DR2-PARAM-CNT (R8)
    00F2' =0002        155  DR2TO   BSS     2       DR2-TABLES-OFFSET (R9)
    00F4' =0002        156  DR2ARO  BSS     2       DR2-ARRAY-OFFSET (R10)
```

```
                    53          INTERN  JBBBUF,JBEBUF
                    54          INTERN  RAMBEG,RAMEND
                    55          INTERN  DWXF,GM,GL,LAT1,LAT2,LAT,LONG1,LONG2,LONG
                    56  *
                    57  ********************************************************************************
                    58  *
                    59  * POWER ON TRANSIENT RAM WORDS. TO BE USED TO DETECT
                    60  * POWER TRANSIENT OR POWER TURN ON.
                    61  * HEX LOCATION D000 IS RESERVED FOR THIS PURPOSE.
0000'  =0020        62  POTRAM  BSS     32
                    63  *                                                                              *
                    64  ********************************************************************************
                    65  *
                    66  *               INTERRUPT'S WORK-SPACE
                    67  *
                    68  ************************************************
                    69  *
                    70  *               POWER ON  (INT 0)
0020'  =000E        71  PONW    BSS     14              R0-R6
002E'  =0002        72  PINIT   BSS     2               R7 ANY POWER INTERRUPT INIT FLAG (RTIFR)
0030'  =0002        73  POCALF  BSS     2               POWER-ON-CALIBRATION-FLAG (R8)
0032'  =0002        74  POR9    BSS     2               R9
0034'  =0002        75  VRSMSK  BSS     2               VERSION-# (R10)
0036'  =000A        76  PONR    BSS     10              R11-R15
                    77  *
                    78  ************************************************
                    79  *
                    80  * BEGINNING OF RAM USED BY POWER ON ISR
0040'               81  RAMBEG
                    82  ************************************************
                    83  *
                    84  *               POWER DOWN (INT 1)
0040'  =0012        85  PDNW    BSS     9#2
0052'  =0002        86  ANSFLG  BSS     2       REG 9
       =000C        87          BSS     6#2
                    88  *
                    89  ************************************************
                    90  *
                    91  *               4-SEC FRAME (INT 2)
                    92  * (NOT USED)
                    93  *
                    94  ************************************************
                    95  *
                    96  *               REAL-TIME (INT 3)
0060'  =0008        97  RTW     BSS     8               R0-R3
0068'  =0002        98  ICOBAI  BSS     2               INT-CPU O/P BUFFER ADDR TBL INDEX (R4)
006A'  =0002        99  FRCNT   BSS     2               FRAME COUNT (R5)
006C'  =0002       100  TIMEM   BSS     2               MSW OF TIME (R6)
006E'              101  CLOCK
006E'  =0002       102  TIMEL   BSS     2               LSW (R7)
0070'  =0002       103  CYPFRC  BSS     2               CYCLE-PER-FRAME COUNT (R8)
0072'  =0002       104  RTPCYC  BSS     2               RT-PER-CYCLE COUNT (R9)
```

```
                    1   *
                    2   *       FILE : RAM.SRC
                    3   *
                    4   *
                    5   *
          =0000     6           RSECT   RAM
                    7   *
                    8   *
                    9   *
                   10   ******
                   11   *           CPU#1.
                   12   *       ALL RAM DEFINITION
                   13   *****
                   14   *
                   15   *       VERSION HISTORY:
                   16   *
                   17   *
                   18   *
                   19   ********************************************************************************
                   20   *                                                                              *
                   21           INTERN  PONW,POCALF,VRSMSK,PONW,POTRAM
                   22           INTERN  RTW,FRCNT,CYPFRC,RTPCYC,ICOBAI,EASADR
                   23           INTERN  TIMEM,TIMEL,CLOCK,CLRFLG,SSEG
                   24           INTERN  DFBTF,DFW,DFLWR,DFWRC,DFAILC
                   25           INTERN  ADCW,DASFLG,ANSFLG,ATW,ZERO1,SYNFLG
                   26           INTERN  BGW,BITCTR,CALCTR,DPSTF
                   27           INTERN  DR1W,DR1ARD,DR1UP,DR1DA,DR1PCT,DR1TO
                   28           INTERN  DR2W,DR2ARD,DR2UP,DR2DA,DR2PCT,DR2TO
                   29           INTERN  DR3W,DR3ARD,DR3UP,DR3DA,DR3PCT,DR3TO
                   30           INTERN  DPSYN,DPFLPA,DPWD64,DFBTF,DFOQ
                   31           INTERN  DPWRD,DPSCTR,DPSCTI,DPBUFF,DPBUFF,DPBUFI,DPMODE,DPWORD
                   32           INTERN  DPBUF1,DPBUF2,DFPBAI,DPSNOI,DPWNOI
                   33           INTERN  EATSBF,EAIMBF,EASADR,EACFLG,EATBUF,EABTF
                   34           INTERN  DC1W,DC1FLG,DC1ARD,DC1DA,DC1PCNT,DC1TOF,EAFLG
                   35           INTERN  AMW,ANARD,ANDA,AMPCNT,ANTOF
                   36           INTERN  CALBTW,CALAP,CNVFLG,CALSN,BTADP
                   37           INTERN  MSIX,MSERR,SYDFLL
                   38           INTERN  MICHTW
                   39           INTERN  OUTBB,OUTBB1,PINIT
                   40           INTERN  OPL,OPR,OTL,OTR,OQL,OQR,
                   41           INTERN  SYSTWB,SYERF,SYERF2,SYSW,SYSTBO
                   42           INTERN  SYCALB,SYCW5,SYCW25,SYCWO
                   43           INTERN  SYDOCB,SYDAY,SYDMTH,SYDFMH,SYDFLH
                   44           INTERN  VCPR,AXOB,DX10B,DX20B
                   45           INTERN  KSBUF,KRBUF,VMOSBF,VMORBF
                   46           INTERN  ICOB1A,ICOB1B,ICOB2A,ICOB2B
                   47           INTERN  ICOER,ICOB1C,ICOB2C
                   48           INTERN  JBIPTR,JBOPTR
                   49           INTERN  WFEX,JBIPTR,JBOPTR,WPJB,WPJB1
                   50           INTERN  VIBL,VIBR,APURPM,APUEGT
                   51           INTERN  GWM,GWL,LATM1,LATM2,LATL,LONGM1,LONGM2,LONGL
                   52           INTERN  ACIDNT,EABTF
```

```
0000'              1165  RTISR
0000'  06A0 0003*  1166        BL      @RTSIN1
0004'  020C 0700   1167        LI      CRU,C9901       RESET 9901 REAL TIME INTERRUPT
0008'  1E 00       1168        SBZ     0
000A'  1D 03       1169        SBO     C3
000C'  0587        1170        INC     CLOCK           LSW CLOCK = CLOCK + 1 (1 = 15.625 MS)
000E'  1601        1171        JNE     10$
0010'  0586        1172        INC     TIMEM           MSW OF TIME
0012'  0609        1173  10$   DEC     RTPCYC          RT-PER-CYCLE-CNT (INIT TO 1 AT WD 55, DFDR ISR)
                   1174+       DOIF    ,EQ,,,,         IF RT-PER-CYCLE-CT = 0,
0014'  1610        1344E           JNE   91$
                   1480  *** 1ST SUB CYCLE OF EVERY CYCLE
0016'  0209 0008   1481          LI    RTPCYC,C8               RT-PER-CYCLE-CT = 8
                   1482+         CALL  RTIFR             INITIALIZE FRAME
001A'  06A0 0004*  1489A         BL    @RTIFR
001E'  110B        1491          JLT   91$             JIF NOT SYNCHRONIZED WITH DFDR O/P WORD #55
                   1492  *                            (ONLY FOR NEW S/F)
                   1493+         CALL  RTIDA             INITIALIZE DATA ACQ
0020'  06A0 0005*  1500A         BL    @RTIDA
                   1502  *
0024'  0420 0002*  1503        BLWP    @JBQUE          QUE JOB 8 PS
0028'  0001        1504        DATA    1               JOB #2
002A'  2220 0001*  1505        COC     @BO,CYPFRC      2ND 4TH 6TH & 8TH CYCLE
                   1506+       DOIF    ,EQ,            QUE IF EQUAL
002E'  1603        1680E           JNE   92$
0030'  0420 0002*  1812          BLWP  @JBQUE          QUE JOB4PS
0034'  0002        1813          DATA  2               JOB #3
                   1814+       ENDBLK
0036'              1950E  92$
                   2034+       ENDBLK                  IFEND
0036'              2162E  91$
                   2254  *
                   2255  *** INTER-CPU START MESSAGE TRANSMISSION
0036'  06A0 0006*  2256        BL      @ICSTX
003A'  0380        2257        RTWP
                   2258  ********************************************************************************
                   2259        END
```

```
                         1          IDT     SYDSP
                         2          SUBTTL  SYSTEM ERROR DISPLAY
                         3  ********************************************************************
                         4  *                                                                  *
                         5  * NAME: SYDSP                          AUTH: N.COSTANTINIDES     *
                         6  * VERSION: 1                           DATE: 19-MAY-1982         *
                         7  *                                                                  *
                         8  * FUNCTION: THE ROUTINE IS CALLED BY SYSOUT TO OUTPUT THE         *
                         9  *          SYSTEM ERROR CODE TO THE 3 LED DISPLAYS, FROM THE      *
                        10  *          EAROM IMAGE BUFFER                                     *
                        11  *                                                                  *
                        12  * CALLING MODULES: SYSOUT                                         *
                        13  *                                                                  *
                        14  * CALLING SEQ:BL @SYDSP                                           *
                        15  *                                                                  *
                        16  * INPUTS: EAROM IMAGE BUFFER (EAIMBF)                             *
                        17  *                                                                  *
                        18  * OUTPUTS: ERROR CODES ON LED DISPLAY                             *
                        19  *                                                                  *
                        20  * MODULES REFERENCED: NONE                                        *
                        21  *                                                                  *
                        22  * WORKSPACE AREA:WPJB                                             *
                        23  *                                                                  *
                        24  * REGISTERS MODIFIED: R1,R0                                       *
                        25  *                                                                  *
                        26  * VERSION HISTORY:                                                *
                        27  *                                                                  *
                        28  ********************************************************************
       =0000            29          RSECT   SYDSP
                        30  *** CALL NAME
                        31          INTERN  SYDSP
                        32  *** VARIABLES REFERENCED
                        33          EXTERN  EAIMBF
                        34  *** CONSTANTS REFERENCED
                        35          EXTERN  D3
                        36  *** TABLES REFERENCED
                        37  *** MODULES REFERENCED
                        38  *** LIBRARY
                        39          INCLUDE CNSTNT
                       167  *** REGISTERS DEFINITION
       =000A           168  RSCTR   EQU     R10     =       READ SWITCH COUNTER
                       169  ********************************************************************
  0000'                170  SYDSP
  0000'  C04A          171          MOV     RSCTR,R1                READ SWITCH COUNTER
  0002'  2460 0002*    172          CZC     @D3,R1                  LOWER 2 BITS = 0 IF 4 SEC UP
  0006'  160B          173          JNE     100$                    JIF NOT TIME TO DISPLAY NEXT ERROR CODE
                       174  ***  4 SECS HAVE ELAPSED; DISPLAY NEXT ERROR CODE
  0008'  0921          175          SRL     R1,2                    READ SWITCH COUNTER
  000A'  0241 000F     176          ANDI    R1,>F
  000E'  0601          177          DEC     R1                      LESS 1
  0010'  0241 000F     178          ANDI    R1,>F                   CLEAR UPPER 12 BITS JUST IN CASE
  0014'  0A11          179          SLA     R1,1                    *2 = INDEX
```

```
                    180   *
0016'  C021 0001*   181         MOV     @EAIMBF(R1),R0          GET ERROR CODE FROM IMAGE OF EAROM
001A'  C800 FF80    182         MOV     R0,@CFPDSP              PUT     ERROR CODE IN 3 LED'S
                    183   *
001E'  045B         184   100*  RT
                    185   ********************************************************************************
                    186         END
```

No errors detected

```
                         1          IDT     SYBTOC
                         2          SUBTTL  SETS UPERROR TYPE CODE
                         3  ********************************************************************
                         4  *                                                                  *
                         5  * NAME: SYBTOC                          AUTH: N.COSTANTINIDES    *
                         6  * VERSION: 1                            DATE: 25-JUN-1982        *
                         7  *                                                                  *
                         8  * FUNCTION: CHECKS THE 16 ERROR WORDS AND SETS UP THE ERROR       *
                         9  *           TYPE CODE FOR THE SYSTEM ERROR CODE FOR EACH           *
                        10  *           WORD LOCATED IN THE SYSTEM ERROR BUFFER, USING         *
                        11  *           THE SYSTEM ERROR TYPE TABLE SYETBL                     *
                        12  *                                                                  *
                        13  * CALLING MODULES: SYSMON                                         *
                        14  *                                                                  *
                        15  * CALLING SEQ: BL @SYBTOC                                         *
                        16  *                                                                  *
                        17  * INPUTS: R4    =       WORD COUNTER (INDEX FOR SYETBL & SYEBF) *
                        18  *         R6    =       BIT COUNTER (NO OF BITS TESTED IN R7)   *
                        19  *         R7    =       ERROR BIT WORD (WORD FROM SYEBF)        *
                        20  *         R1    = 0     IF 1ST INITIAL CALL                     *
                        21  *               = NOT 0 IF NOT 1ST INITIAL CALL                 *
                        22  *                                                                  *
                        23  * OUTPUTS: R1   =       ERROR CODE                              *
                        24  *          R1   =       0 IF NO ERROR CODE IN SYEBF BUFFER      *
                        25  *          R4   =       WORD COUNTER                            *
                        26  *          R6   =       BIT COUNTER                             *
                        27  *          R7   =       ERROR BIT WORD                          *
                        28  *                                                                  *
                        29  * MODULES REFERENCED: NONE                                        *
                        30  *                                                                  *
                        31  * WORKSPACE AREA:WPJB                                             *
                        32  *                                                                  *
                        33  * REGISTERS MODIFIED:R0,R1,R4,R6,R7                               *
                        34  *                                                                  *
                        35  * VERSION HISTORY:                                                *
                        36  *                                                                  *
                        37  ********************************************************************
        =0000           38          RSECT   SYBTOC
                        39  *** CALL NAME
                        40          INTERN  SYBTOC
                        41  *** VARIABLES REFERENCED
                        42          EXTERN  SYEBF
                        43  *** CONSTANTS REFERENCED
                        44  *** TABLES REFERENCED
                        45          EXTERN  SYETBL
                        46  *** MODULES REFERENCED
                        47  *** LIBRARY
                        48  *** REGISTERS DEFINITION
                        49  ********************************************************************
    0000'               50  SYBTOC
    0000'  C041         51          MOV     R1,R1           CHECK IF 1ST CALL
    0002'  1302         52          JEQ     10$             JIF IT IS
```

```
0004' 04C1          53          CLR     R1              RESET IT
0006' 100C          54          JMP     22$             CONTINUE TO CHECK REST OF ERROR
0008' 0284 001E     55  10$     CI      R4,15$2
000C' 1517          56          JGT     50$                     JIF NO MORE CHECKING
000E' C024 0002$    57          MOV     @SYETBL(R4),R0          ERROR TYPE OR ERROR WORD
0012' 1504          58          JGT     20$                     JIF LEGITIMATE ERROR TYPE
0014' 1303          59          JEQ     20$
                    60  $$$ COME HERE IF NO ERROR IN THIS WORD
0016' 04C6          61          CLR     R6                      INIT    BIT COUNTER
0018' 05C4          62          INCT    R4                      BUMP WORD COUNTER
001A' 10F6          63          JMP     10$                     LOOP BACK
001C'               64  20$
001C' C1E4 0001$    65          MOV     @SYEBF(R4),R7           GET ERROR WORD FROM SYSTEM ERROR BUFF
                    66  $$$ CHECK ERROR BIT IN ERROR WORD
0020' 0586          67  22$     INC     R6                      BUMP BIT COUNTER
0022' 0917          68          SRL     R7,1                    PUT ERROR BIT IN CPU CARRY STATUS BIT
0024' 1807          69          JOC     40$                     JIF ERROR DETECTED
0026' 1303          70          JEQ     26$                     JIF NO ERROR IN THE REST OF BITS
0028' 0286 0010     71          CI      R6,16
002C' 11F9          72          JLT     22$                     JIF ALL 16 BITS NOT CHECKED
002E' 04C6          73  26$     CLR     R6                      INIT BIT COUNTER
0030' 05C4          74          INCT    R4                      BUMP WORD COUNTER
0032' 10EA          75          JMP     10$                     LOOP FOR MORE
                    76  $$$
0034' C046          77  40$     MOV     R6,R1                   BIT COUNTER
0036' E064 0002$    78          SOC     @SYETBL(R4),R1          R1 = ERROR CODE FOR CALLER
003A' 1001          79          JMP     100$
                    80  $$$
003C' 04C1          81  50$     CLR     R1                      NO ERROR FLAG FOR CALLER
003E' C041          82  100$    MOV     R1,R1
0040' 045B          83          RT
                    84  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
                    85          END
```

No errors detected

```
                     1          IDT     SYSER
                     2
                     3          SUBTTL  SYSTEM ERROR SETUP
                     4
                     5
                     6  *-------------------------------------------------------------------+
                     7  *                                                                   !
                     8  *       THIS SUBROUTINE IS CALLED BY AN ERROR CODE WHEN A SYSTEM
                     9  *       ERROR IS DETECTED. THE ERROR CODE WILL BE CONVERTED TO WORD
                    10  *       AND BIT NUMBERS WHICH WILL BE USED TO SET AN ERROR BIT IN THE
                    11  *       SYSTEM ERROR BUFFER.
                    12  *       THE INPUT ERROR CODE FORMAT IS
                    13  *
                    14  *       BIT NO: 11  10  9  8  7  6  5  4  3  2  1  0  (LSB)
                    15  *               N2  N2  N2 N2 N1 1N N1 N1 N1 N1 N1 N1
                    16  *
                    17  *       WHERE N1 = BIT NO (1 TO 32)
                    18  *             N2 = WORD NO (0 TO 15) OF SYST ERROR BUFFER
                    19  *
                    20  *       IF THE BIT NO IS GREATER THAN 16, THE WORD NO IS THE
                    21  *       NEXT WORD OF THE DECODED ONE
                    22  $
                    23  *       CALLING SEQ: BL @SYSER
                    24  *       INPUTS: R1 = ERROR CODE
                    25  *       OUTPUTS : SETS BIT N1 OF WORD NUMBER N2 IN SYSTEM ERROR BUFFER
                    26  *                                                                   !
                    27  *-------------------------------------------------------------------+
                    28  *       VERSION : 2
                    29  *       PROGRAMMED BY : N.CONSTANTINIDES
                    30
                    31
                    32         INTERN  SYSER
                    33  * GLOBAL AREA:
                    34         EXTERN  SYEBF          SYSTEM ERROR BUFFER (RAM)
                    35         EXTERN  B0,B5          ROM
                    36  *
        =0000       37         RSECT   SYSER
```

```
                    39  ********************************************************************************
0000'               40  SYSER
                    41  ***  CONVERT ERROR CODE TO BIT NUMBER AND WORD NUMBER
                    42  ***  OF SYSTEM ERROR BUFFER
0000'  C001         43          MOV     R1,R0                   R0 FOR WORD NUMBER
0002'  0601         44          DEC     R1                      DECR BIT NUMBER
0004'  0241 001F    45          ANDI    R1,>1F                  SAVE BIT NO ONLY
0008'  0A11         46          SLA     R1,1                    *2 (=0 TO 62 RANGE)
                    47  *
000A'  0240 0F00    48          ANDI    R0,>F00                 SAVE WORD NO ONLY
000E'  0970         49          SRL     R0,7                    RJ AND *2 (=0 TO 30 RANGE)
0010'  0220 0001*   50          AI      R0,SYEBF                ADD START ADDRESS OF SYSTEM ERROR BUF
                    51  ***
0014'  0281 001E    52          CI      R1,15*2                 CHECK BIT NO IF TWO WORD TYPE
0018'  1203         53          JLE     10$                     JIF NOT 2 WORD CODE
                    54  ***  SELECT NEXT WORD OF SYSTEM ERROR BUFFER AND BIT NUMBER
001A'  6060 0003*   55          S       @B5,R1                  LESS 32 BYTES = NEW BIT NUMBER
001E'  05C0         56          INCT    R0                      2ND WORD
                    57  *** SET ERROR BIT IN SYSTEM ERROR BUFFER(SYEBF)
0020'  E421 0002*   58  10$     SOC     @B0(R1),*R0
                    59  ***
0024'  045B         60          RT
                    61  ********************************************************************************
                    62          END
```

No errors detected

```
                         1          IDT     SYLMP
                         2          SUBTTL  TURN ON/OFF LAMPS
                         3  ********************************************************************
                         4  *                                                                  *
                         5  * NAME: SYLMP                          AUTH: M.COSTANTINIDES     *
                         6  * VERSION:                             DATE: 10-MAY-1983         *
                         7  *                                                                  *
                         8  * FUNCTION: TURNS ON DFDAU FAIL LAMP IF A BIT IS SET IN 2ND       *
                         9  *           WORD OF SYSTEM ERROR BUFFER.                           *
                        10  *           TURNS OFF DFDAU FAIL LAMP IF NOT SET.                  *
                        11  *           TURNS ON DFDAU CAUTION LAMP IF ERROR IN 3RD,5TH        *
                        12  *           9TH & 10TH WORDS OF SYSTEM ERROR BUFFER                *
                        13  *           TURNS OFF DFDAU CAUTION LAMP IF NOT SET                *
                        14  *                                                                  *
                        15  * CALLING MODULES: SYSOUT                                        *
                        16  *                                                                  *
                        17  * CALLING SEQ: BL @SYLMP                                         *
                        18  *                                                                  *
                        19  * INPUTS: SYSTEM ERROR BUFFER                                    *
                        20  *                                                                  *
                        21  * OUTPUTS: LAMPS                                                 *
                        22  *                                                                  *
                        23  * MODULES REFERENCED:NONE                                        *
                        24  *                                                                  *
                        25  * WORKSPACE AREA: WPJB                                           *
                        26  *                                                                  *
                        27  * REGISTERS MODIFIED:   R0,R1                                    *
                        28  *                                                                  *
                        29  * VERSION HISTORY:                                               *
                        30  *                                                                  *
                        31  ********************************************************************
          =0000         32          RSECT   SYLMP
                        33  *** CALL NAME
                        34          INTERN  SYLMP
                        35  *** VARIABLES REFERENCED
                        36          EXTERN  SYEBF
                        37  *** CONSTANTS REFERENCED
                        38  *** TABLES REFERENCED
                        39  *** MODULES REFERENCED
                        40  *** LIBRARY
                        41          INCLUDE REGDEF
                        60          INCLUDE CNSTNT
                       188  *** REGISTERS DEFINITION
                       189  ********************************************************************
0000' 020C 05E0         190  SYLMP   LI      CRU,CRUFPL          FRONT PANEL CRU ADDRESS
0004' 0201 0001*        191          LI      R1,SYEBF            SYSTEM ERROR BUFFER
0008' 05C1              192          INCT    R1                  BUMP TO 2ND WORD
000A' C031              193          MOV     *R1+,R0             AND BUMP TO 3RD WORD
000C' 1302              194          JEQ     10$                 JIF NO ERROR IN 2ND WORD
                        195  ***  DFDAU FAILURE
000E' 1D 01             196          SBO     1                   TURN ON DFDAU FAIL LAMP
0010' 1001              197          JMP     20$
```

```
                    198  ***  NOT DFDAU FAILURE
0012' 1E 01         199  10$     SBZ     1                   TURN OFF
                    200  ***
0014' C021 0006     201  20$     MOV     @6(R1),R0           BIT#0 6TH WORD(CPU#1/CPU#2 COMM ERR)
0018' 0240 0001     202          ANDI    R0,>1                CLEAR OTHER BITS IN 6TH WORD
001C' E031          203          SOC     *R1+,R0             3RD WORD OF SYSTEM ERROR BUFFER
001E' E031          204          SOC     *R1+,R0             4TH WORD
0020' E031          205          SOC     *R1+,R0             5TH WORD
0022' 0221 0006     206          AI      R1,6                BUMP TO 9TH WORD
0026' E031          207          SOC     *R1+,R0             9TH WORD
0028' E011          208          SOC     *R1,R0              10TH WORD
002A' C000          209          MOV     R0,R0
002C' 1302          210          JEQ     30$                 JIF NO ERROR
                    211  ***  DFDAU CAUTION
002E' 1D 02         212          SBO     2                   TURN ON
0030' 1001          213          JMP     100$
                    214  ***  NOT DFDAU CAUTION FAILURES
0032' 1E 02         215  30$     SBZ     2                   TURN OFF
                    216  *
0034' 045B          217  100$    RT
                    218  ********************************************************************************
                    219          END
```

No errors detected

```
                    1          IDT     SYSOK
                    2
                    3          SUBTTL  SYSTEM ERROR SETUP
                    4
                    5
                    6  *-------------------------------------------------------------------+
                    7  *                                                                   |
                    8  *       THIS SUBROUTINE IS CALLED BY AN ERROR CODE WHEN A SYSTEM
                    9  *       ERROR IS DETECTED. THE ERROR CODE WILL BE CONVERTED TO WORD
                   10  *       AND BIT NUMBERS WHICH WILL BE USED TO SET AN ERROR BIT IN THE
                   11  *       SYSTEM ERROR BUFFER.
                   12  *       THE INPUT ERROR CODE FORMAT IS
                   13  *
                   14  *       BIT NO: 11  10  9  8  7  6  5  4  3  2  1  0  (LSB)
                   15  *               N2  N2  N2 N2 N1 1N N1 N1 N1 N1 N1 N1
                   16  *
                   17  *       WHERE N1 = BIT NO (1 TO 32)
                   18  *             N2 = WORD NO (0 TO 15) OF SYST ERROR BUFFER
                   19  *
                   20  *       IF THE BIT NO IS GREATER THAN 16, THE WORD NO IS THE
                   21  *       NEXT WORD OF THE DECODED ONE
                   22  *
                   23  *       CALLING SEQ: BL @SYSOK
                   24  *       INPUTS: R1 = ERROR CODE
                   25  *       OUTPUTS : SETS BIT N1 OF WORD NUMBER N2 IN SYSTEM ERROR BUFFER
                   26  *                                                                   |
                   27  *-------------------------------------------------------------------+
                   28  *       VERSION : 2
                   29  *       PROGRAMMED BY : N.CONSTANTINIDES
                   30
                   31
                   33  * GLOBAL AREA:
                   34         EXTERN  SYEBF          SYSTEM ERROR BUFFER (RAM)
                   35         EXTERN  B0,B5          ROM
                   36  *
      =0000        37         RSECT   SYSOK
```

```
                    39  ********************************************************************************
0000'               40  SYSOK
                    41  ***  CONVERT ERROR CODE TO BIT NUMBER AND WORD NUMBER
                    42  ***  OF SYSTEM ERROR BUFFER
0000'  C001         43          MOV     R1,R0                   R0 FOR WORD NUMRER
0002'  0601         44          DEC     R1                      DECR BIT NUMBER
0004'  0241 001F    45          ANDI    R1,>1F                  SAVE BIT NO ONLY
0008'  0A11         46          SLA     R1,1                    * 2 (=0 TO 62 RANGE)
                    47  *
000A'  0240 0F00    48          ANDI    R0,>F00                 SAVE WORD NO ONLY
000E'  0970         49          SRL     R0,7                    RJ AND *2 (=0 TO 30 RANGE)
0010'  0220 0001*   50          AI      R0,SYEBF                ADD START ADDRESS OF SYSTEM ERROR BUF
                    51  ***
0014'  0281 001E    52          CI      R1,15*2                 CHECK BIT NO IF TWO WORD TYPE
0018'  1203         53          JLE     10$                     JIF NOT 2 WORD CODE
                    54  ***  SELECT NEXT WORD OF SYSTEM ERROR BUFFER AND BIT NUMBER
001A'  6060 0003*   55          S       @B5,R1                  LESS 32 BYTES = NEW BIT NUMBER
001E'  05C0         56          INCT    R0                      2ND WORD
                    57  *** CLEAR ERROR BIT IN SYSTEM ERROR BUFFER(SYEBF)
0020'  4421 0002*   58  10$     SZC     @B0(R1),*R0
                    59  ***
0024'  045B         60          RT
                    61  *
                    62  ********************************************************************************
                    63          END
```

No errors detected

```
                     1          IDT     SYSOUT
                     2          SUBTTL
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME:SYSOUT                          AUTH: N.COSTANTINIDES     *
                     6  * VERSION: 2                           DATE: 10-MAY-1983         *
                     7  *                                                                  *
                     8  * FUNCTION: CHECKS IF READ SWITCH IS DEPRESSED. IF NOT IT SETS    *
                     9  *           THE READ SWITCH COUNTER TO -1 AND CALLS SYLMP           *
                    10  *           (TURNS ON/OFF THE CAUTION AND FAIL LAMPS AS PER        *
                    11  *           SYSTEM ERROR BUFFER.)                                  *
                    12  *           IF THE SWITCH HAS BEEN DEPRESSED FOR LESS THAN         *
                    13  *           4 SECS IT PERFORMS A LAMP TEST. IF SWITCH IS           *
                    14  *           DEPRESSED FOR MORE THAN 4 SEC, IT ENDS THE LAMP        *
                    15  *           TEST AND CALLS SYDSP  (DISPLAYS SYSTEM ERROR CODE      *
                    16  *           FROM EAROM IMAGE)                                      *
                    17  *           ITERATION RATE = 1 PER SEC                             *
                    18  *                                                                  *
                    19  * CALLING MODULES: JOB4PS                                         *
                    20  *                                                                  *
                    21  * CALLING SEQ: BL @SYSOUT                                         *
                    22  *                                                                  *
                    23  * INPUTS:       READ SWITCH                                       *
                    24  *                                                                  *
                    25  * OUTPUTS:      FRONT PANEL LAMPS                                 *
                    26  *               FRONT PANEL LED DISPLAYS                          *
                    27  *                                                                  *
                    28  * MODULES REFERENCED:SYLMP = FRONT PANEL LAMP TURN ON/OFF         *
                    29  *                    SYDSP = FRONT PANEL 3 LED DISPLAY            *
                    30  *                                                                  *
                    31  * WORKSPACE AREA: WPJB                                            *
                    32  *                                                                  *
                    33  * REGISTERS MODIFIED:R0,R3                                        *
                    34  *                                                                  *
                    35  * VERSION HISTORY:                                                *
                    36  *                                                                  *
                    37  *                                                                  *
                    38  ********************************************************************
      =0000         39          RSECT   SYSOUT
                    40  *** CALL NAME
                    41          INTERN  SYSOUT
                    42  *** VARIABLES REFERENCED
                    43  *** CONSTANTS REFERENCED
                    44  *** TABLES REFERENCED
                    45  *** MODULES REFERENCED
                    46          EXTERN  SYLMP,SYDSP
                    47  *** LIBRARY
                    48          INCLUDE CNSTNT
                   176  *** REGISTERS DEFINITION
      =000A        177  RSCTR   EQU     R10     =       READ SWITCH COUNTER
                   178  ********************************************************************
0000' C0CB         179  SYSOUT  MOV     R11,R3                  SAVE LINK
```

```
0002' 020C 05E0    180          LI    R12,CRUFPL          FRONT PANEL CRU ADDRESS
                   181  ***  READ "READ SWITCH"
0006' C020 FF94    182          MOV   @CMDFWA,R0          READ SWITCH
000A' 1108         183          JLT   20$                 JIF READ SWITCH DEPRESSED
                   184  ***  READ SWITCH NOT DEPRESSED
000C' C28A         185          MOV   RSCTR,RSCTR         READ SWITCH COUNTER
000E' 1102         186          JLT   10$                 JIF READ SWITCH WAS DEPRESSED BEFORE
0010' 070A         187          SETO  RSCTR               SET READ SW CTR TO NOT DEPRESSED
0012' 1010         188          JMP   30$
0014' 070A         189  10$     SETO  RSCTR               READ SWICH COUNTER = -1
0016' 06A0 0001$   190          BL    @SYLMP              LAMP TO ON/OFF
001A' 1015         191          JMP   100$                EXIT
                   192  ***  READ SWITCH DEPRESSED
001C'              193  20$
001C' 058A         194          INC   RSCTR               BUMP READ SWITCH COUNTER
001E' 028A 0004    195          CI    RSCTR,4
0022' 1B0D         196          JH    40$                 JIF NOT LAMP TEST PERIOD
0024' 1307         197          JEQ   30$                 JIF END OF LAMP TEST
                   198  ***  LAMP TEST PERIOD
0026' 1D 00        199          SBO   0                   TURN ON DFDR FAIL LAMP
0028' 1D 01        200          SBO   1                    *    * DFDAU  *   *
002A' 1D 02        201          SBO   2                    *    *   *  CAUTION LAMP
                   202  *
002C' C820 004A'   203          MOV   @D888,@CFPDSP       FRONT PANEL LED TO 888
0030' FFB0
0032' 1009         204          JMP   100$                EXIT
                   205  ***  END OF LAMP TEST
0034' 1E 00        206  30$     SBZ   0                   TURN OFF
0036' 1E 01        207          SBZ   1                    *    *
0038' 1E 02        208          SBZ   2                    *    *
003A' 04E0 FFB0    209          CLR   @CFPDSP             FRONT PANEL LED TO 000
                   210  ***  DISPLAY ERROR CODE ON LED
003E'              211  40$
003E' 06A0 0002$   212          BL    @SYDSP              DISPLAY SYSTEM ERROR CODE
                   213  ***  TURN ON LAMPS
0042' 06A0 0001$   214          BL    @SYLMP              TURN ON/OFF LAMPS
                   215  *
0046' C2C3         216  100$    MOV   R3,R11              USAVE LINK
0048' 045B         217          RT
                   218  ************************************************************
004A' 0888         219  D888    DATA  >888
                   220          END
```

No errors detected

```
                1          IDT     SYSMON
                2          SUBTTL  SYSTEM MONITOR
                3  ********************************************************************
                4  *                                                                  *
                5  * NAME: SYSMON.SRC                      AUTH: N.COSTANTINIDES      *
                6  * VERSION:                              DATE: 25-JUN-1982          *
                7  *                                                                  *
                8  * FUNCTION: THIS ROUTINE IS CALLED ONCE PER SECOND TO CHECK        *
                9  *           IF NEW ERROR HAS BEEN GENERATED OR DELETED.            *
               10  *           THE SYSTEM ERROR BUFFER (SYEBF) CONTAINS THE           *
               11  *           CURRENT ERROR STATUS. THE FIRST 8 WORDS OF THE SYEBF   *
               12  *           ARE RESERVED FOR ERRORS GENERATED BY CPU #1 AND THE    *
               13  *           LAST 8 FOR CPU #2.                                     *
               14  *           ALL ERRORS LOCATED IN THE SYEBF WILL BE CONVERTED TO   *
               15  *           A PREDEFINED ERROR CODE. WHENEVER A NEW SUPERFRAME     *
               16  *           IS DETECTED THE SUPERFRAME STATUS BUFFER (SYSTWB)       *
               17  *           WILL BE CLEARED AND ANY ERROR CODE DECODED FROM THE     *
               18  *           SYEBF WILL BE STORED IN THIS BUFFER. WHEN IT IS NOT     *
               19  *           A NEW SUPERFRAME THE ERROR CODE WILL BE STORED IN       *
               20  *           THE SYSTWB PROVIDED THIS ERROR CODE IS NOT IN THE       *
               21  *           SYSTWB, IT WILL ALSO BE STORED IN THE TEMPORARY        *
               22  *           EAROM SYSTEM STATUS BUFFER (EATSBF) IF THE             *
               23  *           ERROR CODE IS NOT ALREADY IN THE EAROM.                *
               24  *                                                                  *
               25  * CALLING MODULES: JOB4PS                                          *
               26  *                                                                  *
               27  * CALLING SEQ: BL @SYSMON                                          *
               28  *                                                                  *
               29  * INPUTS: SYEBF =       SYSTEM ERROR BUFFER                        *
               30  *       FRCNT  = FRAME/SUBFRAME COUNTER                            *
               31  *                                                                  *
               32  * OUTPUTS: EATSBF       =       TEMP EAROM SYSTEM STATUS BUFFER    *
               33  *          SYSTWB       =       SUPERFRAME SYSTEM STATUS BUFFER    *
               34  *                                                                  *
               35  * MODULES REFERENCED: SYBTOC                                       *
               36  *                                                                  *
               37  * WORKSPACE AREA: WPJB                                             *
               38  *                                                                  *
               39  * REGISTERS MODIFIED: R3,R1,R4,R5,R6,R0,R8                         *
               40  *       R3      =       JUMP REGISTER                              *
               41  *       R4      =       WORD COUNTER (INDEX FOR SYETBL)            *
               42  *       R5      =       ERROR COUNTER (MAX OF 16)                  *
               43  *       R6      =       BIT COUNTER (NO. OF BITS TESTED IN R7)     *
               44  *       R7      =       ERROR BIT WORD (WORD FROM SYEBF)           *
               45  *                                                                  *
               46  * VERSION HISTORY:                                                 *
               47  *                                                                  *
               48  ********************************************************************
=0000          49         RSECT   SYSMON
               50  ### CALL NAME
               51         INTERN  SYSMON
               52  ### VARIABLES REFERENCED
```

```
                        53          EXTERN  FRCNT
                        54          EXTERN  SYSTWB
                        55          EXTERN  EAIMBF
                        56          EXTERN  EATSBF
                        57  *** CONSTANTS REFERENCED
                        58          EXTERN  X3F
                        59  *** TABLES REFERENCED
                        60  *** MODULES REFERENCED
                        61          EXTERN  SYBTOC
                        62  *** LIBRARY
                        63          INCLUDE ENCLOS
                        65  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                        67          INCLUDE REGDEF          REGISTER DEFENITIONS
                        86          INCLUDE CNSTNT          CONSTANTS
                       214          INCLUDE SUBMAC          FUNCTIONAL MACROS
                       515          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                       745          INCLUDE JMPMAC          JUMP MACROS
                       779          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                       794          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
                      1170  *** REGISTERS DEFINITION
                      1171  ********************************************************************
0000' C80B 0000'      1172  SYSMON  MOV     R11,@LINK1                      SAVE LINK
0004' 0203 0002       1173          LI      R3,2                    SELECT NOT NEW SUPERFRAME FUNCTION
0008' C020 0001*      1174          MOV     @FRCNT,R0               FRAME/SUBFRAME COUNTER
000C' 2420 0005*      1175          CZC     @X3F,R0
0010' 1607            1176          JNE     20$                     JIF NOT NEW SUPERFRAME
                      1177  ***  NEW SUPERFRAME
0012' 0201 001E       1178          LI      R1,15*2
0016' 04E1 0002*      1179  10$     CLR     @SYSTWB(R1)             CLEAR SUPERFRAME SYSTEM STAT BUFFER
001A' 0641            1180          DECT    R1
001C' 18FC            1181          JOC     10$
001E' 04C3            1182          CLR     R3                      SELECT NEW SUPERFRAME FUNCTION
                      1183  ***
0020' 04C4            1184  20$     CLR     R4                      INIT WORD COUNTER FOR SYBTOC
0022' 04C6            1185          CLR     R6                       '    BIT    '     ' SYBTOC
0024' 04C5            1186          CLR     R5                      INIT ERROR COUNTER
0026' 04C1            1187          CLR     R1                      INDICATE 1ST CALL TO SYBTOC
                      1188  ***  CHECK & CONVERT ERROR BIT TO CODE
0028' 06A0 0006*      1189  30$     BL      @SYBTOC         INPUTS: R1,R4,R6,R7  OUTPUTS: R1 = E CODE
002C' 133D            1190          JEQ     100$                    EXIT IF NO ERROR IN SYST ERROR BUFFER
                      1191  ***
002E' 0463 0032'      1192          B       @40$(R3)
0032' 1001            1193  40$     JMP     50$                     NEW SUPERFRAME FUNCTION
0034' 1007            1194          JMP     60$                     OTHERS
                      1195  ********************************************************
                      1196  ***  NEW SUPERFRAME FUNCTION
0036' C941 0002*      1197  50$     MOV     R1,@SYSTWB(R5)          STORE ERROR CODE IN SUPERFRAME SS BUF
003A' 05C5            1198          INCT    R5                      BUMP ERROR COUNTER
003C' 0285 0020       1199          CI      R5,16*2
0040' 11F3            1200          JLT     30$                     LOOP IF ALL 16 ERROR CODES ARE GENERATED
0042' 1032            1201          JMP     100$                    EXIT
                      1202  ********************************************************
```

```
                    1203  ***  OTHER THAN NEW SUBFRAME FUNCTION
0044'               1204  60$
                    1205  ********************************************************
                    1206  ***  STORE ERROR CODE (R1) IN SS BUFFER
0044'  0200 0010    1207        LI    R0,16                 LOOP COUNT OF 16 WORDS
0048'  0202 0002*   1208        LI    R2,SYSTWB             SUPER FRAME SYSTEM STATUS BUFFER
004C'  BCB1         1209  72$   C     R1,*R2+               CHECK ERROR CODE WITH SSS BUFFER
004E'  1311         1210        JEQ   80$                   JIF    ALREADY IN SSS BUFFER
0050'  0600         1211        DEC   R0
0052'  15FC         1212        JGT   72$                   LOOP IF ALL 16 WORDS NOT CHECKED
                    1213  ***  ERROR CODE IS NOT IN SUPERFRAME SS BUFFER
0054'  C0A0 0001*   1214        MOV   @FRCNT,R2             FRAME/SUBFRAME COUNTER
0058'  0242 003C    1215        ANDI  R2,>3C                SAVE SUPERFRAME COUNT ONLY
005C'  0912         1216        SRL   R2,1                  R,J AND *2
005E'  C022 0002*   1217  76$   MOV   @SYSTWB(R2),R0        CHECK SUPERFRAME SS BUFFER IF AVAILABLE
0062'  1305         1218        JEQ   78$                   JIF WORD AVAILABLE
0064'  05C2         1219        INCT  R2                    BUMP,WORD INDEX
0066'  02B2 0020    1220        CI    R2,16*2
006A'  11F9         1221        JLT   76$                   JIF NOT END OF SSS BUFFER
006C'  1002         1222        JMP   80$                   JIF WORD NOT AVAILABLE
006E'  C881 0002*   1223  78$   MOV   R1,@SYSTWB(R2)        STORE ERROR CODE
                    1224  ********************************************************
                    1225  * IF ERROR CODE NOT IN IMMAGE OF EAROM SS BUFFER ,STORE IN TEMP EAROM SS BUFF
0072'  0200 0010    1226  80$   LI    R0,16                 LOOP COUNT OF 16 WORDS
0076'  0202 0003*   1227        LI    R2,EAIMBF             IMAGE OF EAROM SYSTEM STATUS BUFFER
007A'  8C81         1228  82$   C     R1,*R2+               CHECK ERROR CODE IN IMAGE OF EAROM SS BUFF
007C'  1311         1229        JEQ   90$                   JIF ALREADY IN BUFFER
007E'  0600         1230        DEC   R0
0080'  15FC         1231        JGT   82$                   LOOP IF ALL 16 WORDS CHECKED
                    1232  ***  ERROR CODE NOT IN EAROM. STORE IN TEMP EAROM BUFFER. R1 = ERROR CODE
0082'  0708         1233        SETO  R8                    ASSUME NO ROOM IN TEMP EAROM BUFFER
0084'  04C2         1234        CLR   R2
0086'  C022 0004*   1235  86$   MOV   @EATSBF(R2),R0        CHECK TEMPORARY EAROM SS BUFFER
008A'  8001         1236        C     R1,R0
008C'  1309         1237        JEQ   90$                   JIF ALREADY IN TEMP EAROM SS BUFF
008E'  C202         1238        MOV   R2,R8                 SAVE AVAILABLE WORD OF TEMP EAROM BUFF
0090'  05C2         1239        INCT  R2                    BUMP WORD INDEX
0092'  0282 0020    1240        CI    R2,16*2
0096'  11F7         1241        JLT   86$                   JIF NOT END OF TEMP E SS BUFFER
                    1242  *
0098'  C208         1243        MOV   R8,R8
009A'  1102         1244        JLT   90$                   JIF NO ROOM IN TEMP EAROM BUFFER
009C'  CA01 0004*   1245        MOV   R1,@EATSBF(R8)        STORE ERROR CODE IN TEMP EAROM BUFF
                    1246  ***
00A0'  0585         1247  90$   INC   R5                    BUMP ERROR COUNTER
00A2'  0285 0010    1248        CI    R5,16
00A6'  11C0         1249        JLT   30$                   LOOP IF NOT 16 ERRORS
                    1250  ***
00A8'  C2E0 0000'   1251  100$  MOV   @LINK1,R11                   USAVE LINK
00AC'  045B         1252        RT
                    1253  ********************************************************************
                    1254+       LOCR  PRIV,LINK1
```

```
    0000' =0002         1257A LINK1   BSS     2
                        1258          END

    No errors detected
```

```
                         1          IDT     SYET02
                         2          SUBTTL  TRANSFER ERROR CODES TO INT-CPU BUFFER
                         3  ********************************************************************
                         4  *                                                                  *
                         5  * NAME: SYET02.SRC (TWA)              AUTH: N.COSTANTINIDES      *
                         6  * VERSION: 2                          DATE:   19-MAY-1983        *
                         7  *                                                                  *
                         8  * FUNCTION:     THE SYET02 IS CALLED BY THE REAL TIME ISR AT A    *
                         9  *               RATE OF 1 TIMES PER SECOND.                        *
                        10  *               THE DITS#3-1 CHANNEL UPDATE ERROR WORD AND         *
                        11  *               DITS#3-2 CHANNEL UPDATE ERROR WORD WILL            *
                        12  *               BE STORED IN THE 7TH AND 8TH LOCATION OF SYSTEM    *
                        13  *               ERROR BUFFER, RESPECTIVELY.                        *
                        14  *               THE SYET02 WILL TRANSFER SYSTEM ERROR WORDS         *
                        15  *               LOCATED IN SYEBF TO THE INTER-CPU BUFFER.          *
                        16  *                                                                  *
                        17  * CALLING MODULES: RTIFR                                          *
                        18  *                                                                  *
                        19  * CALLING SEQ: BL @SYET02                                         *
                        20  *                                                                  *
                        21  * INPUTS: SYEBF = SYSTEM ERROR BUFFER                             *
                        22  *         DR1UP = DITS#1 CHANNEL UPDATE ERROR WORD                *
                        23  *         DR2UP = DITS#2 CHANNEL UPDATE ERROR WORD                *
                        24  *                                                                  *
                        25  * OUTPUTS: SYSTEM ERROR WORDS IN INTER-CPU BUFFER                 *
                        26  *                                                                  *
                        27  * MODULES REFERENCED:   NONE                                      *
                        28  *                                                                  *
                        29  * WORKSPACE AREA: CALLER'S                                        *
                        30  *                                                                  *
                        31  * REGISTERS MODIFIED: R0,R1,R2                                    *
                        32  *                                                                  *
                        33  * VERSION HISTORY:                                                *
                        34  *                                                                  *
                        35  ********************************************************************
           =0000        36          RSECT   SYET02
                        37  *** CALL NAME
                        38          INTERN  SYET02
                        39  *** VARIABLES REFERENCED
                        40          EXTERN  DR1UP,DR2UP
                        41          EXTERN  ICOBAI,ICOBAA
                        42          EXTERN  SYEBF
                        43  *** CONSTANTS REFERENCED
                        44  *** TABLES REFERENCED
                        45  *** MODULES REFERENCED
                        46  *** LIBRARY
                        47  *** REGISTERS DEFINITION
                        48  ********************************************************************
0000'                   49  SYET02
                        50  ***
0000'  C820 0001*       51          MOV     @DR1UP,@SYEBF+12        DITS#3-1 CHANNEL UPDATE ERROR
0004'  0000'
```

```
0006' CB20 0002*     52          MOV     @DR2UP,@SYEBF+14        DITS#3-2 CHANNEL UPDATE ERROR
000A' 0000*
                     53    *** INIT
000C' C060 0003*     54          MOV     @ICOBAI,R1              PRESENT INTER-CPU BUFFER INDICATOR
0010' C0A1 0004*     55          MOV     @ICOBAA(R1),R2          PRESENT INTER-CPU BUFFER S.A AND
0014' A0A0 0034'     56          A       @ICESW,R2               ADD OFFSET = POINTS TO ERROR WORD
0018' 0200 0008      57          LI      R0,8                    NUMBER OF ERROR WORDS TO BE TRANSFERED
001C' 0201 0005*     58          LI      R1,SYEBF                SYSTEM ERROR BUFFER (1ST 8 WORDS TO CPU #2)
                     59    *** MOVE SYSTEM ERROR WORDS TO INTER-CPU BUFFER
0020' CCB1           60    10*   MOV     *R1+,*R2+
0022' 0600           61          DEC     R0
0024' 15FD           62          JGT     10*                     LOOP 8 TIMES
                     63    *** INITIALIZE DITS CHANNEL UPDATE REGISTERS TO ALL BAD CHANNELS.
0026' 0200 00FF      64          LI      R0,>00FF                INITIAL VALUE OF CHANNEL UPDATE WORD
002A' C800 0001*     65          MOV     R0,@DR1UP               INIT DITS#3-1 CHANNEL UPDATE REGISTER
002E' C800 0002*     66          MOV     R0,@DR2UP               INIT DITS#3-2 CHANNEL UPDATE REGISTER
                     67    ********************************************************************************
0032' 045B           68          RT                              RETURN
                     69    ********************************************************************************
                     70    *  PRIVATE DATA AREA
0034' 016A           71    ICESW DATA    1B1*2                   OFFSET TO ERROR WORD OF INTER-CPU BUFFER
                     72          END
```

No errors detected

```
                     1          IDT     TBLOFS
                     2          SUBTTL  GET NEXT PARAMETER TABLES OFFSET
                     3  ********************************************************************
                     4  *                                                                  *
                     5  * NAME: TBLOFS.SRC                      AUTH: N.COSTANTINIDES     *
                     6  * VERSION: 2                            DATE: 10-MAR-82           *
                     7  *                                                                  *
                     8  * FUNCTION: ACCORDING TO THE VERSION MASK TBLOFS FINDS THE         *
                     9  *           NEXT PARAMETER'S TABLES-OFFSET (IF THERE ARE MORE      *
                    10  *           PARAMETERS TO ACQUIRE).                                *
                    11  *                                                                  *
                    12  * CALLING MODULES: ANSA = ANALOG ACQ                               *
                    13  *                  DRSA = DITS ACQ                                 *
                    14  *                                                                  *
                    15  * CALLING SEQ: CALL @TBLOFS                                        *
                    16  *                                                                  *
                    17  * INPUTS:                                                          *
                    18  *         R7=TABLE OFFSET ARRAY TABLE START ADDR(ANOA)              *
                    19  *         R8=NUMBER OF PARAMS TO PROCESS IN THIS CYCLE(ANPCNT)     *
                    20  *         R10=OFFSET OF TABLE-OFFSET-ARRAY TABLE(ANARO)            *
                    21  *                                                                  *
                    22  * OUTPUTS: R9=OFFSET OF TABLE OFFSET ARRAY TABLE                   *
                    23  *                                                                  *
                    24  * MODULES REFERENCED: NONE                                         *
                    25  *                                                                  *
                    26  * WORKSPACE AREA: CALLER'S                                         *
                    27  *                                                                  *
                    28  * REGISTERS MODIFIED: R0,R1,R8,R9,R10                              *
                    29  *                                                                  *
                    30  * VERSION HISTORY:                                                 *
                    31  *                                                                  *
                    32  *                                                                  *
                    33  *                                                                  *
                    34  ********************************************************************
=0000               35          RSECT   TBLOFS
                    36  *** CALL NAME
                    37          INTERN  TBLOFS
                    38  *** VARIABLES REFERENCED
                    39          EXTERN  VRSMSK
                    40  *** CONSTANTS REFERENCED
                    41  *** TABLES REFERENCED
                    42  *** MODULES REFERENCED
                    43  *** LIBRARY
                    44          INCLUDE ENCLOS
                    46  ***     ENCLOS.SRC FILE CONTAINS FOLLOWING LIBRARY FILES:
                    48          INCLUDE REGDEF          REGISTER DEFENITIONS
                    67          INCLUDE CNSTNT          CONSTANTS
                   195          INCLUDE SUBMAC          FUNCTIONAL MACROS
                   496          INCLUDE MSCMAC          MISCELLANEOUS MACROS
                   726          INCLUDE JMPMAC          JUMP MACROS
                   760          INCLUDE BLKMAC          OTHER MACROS (BY D. SCOTT)
                   775          INCLUDE LBLMAC          HANDLES MACROS AUTOMATICALLY
```

```
                1151   *** REGISTERS DEFINITION
=0000           1152   TMP     EQU     R0      =       SCRATCH
=0001           1153   MSK     EQU     R1      =       SCRATCH
=0002           1154   VMT     EQU     R2      =       PARAM-VERSION-MASK-TABLE (INPUT)
=0007           1155   POA     EQU     R7      =       PARAM-OFFSET- ARRAY (INPUT)
=0008           1156   PCNT    EQU     R8      =       PARAM COUNT (INPUT,OUTPUT)
=0009           1157   TOF     EQU     R9      =       PARAM TABLES OFFSET (OUTPUT)
=000A           1158   POP     EQU     R10     =       PARAM ARRAY OFFSET (INPUT,OUTPUT)
```

```
                    1160  ********************************************************************
0000'               1161  TBLOFS
0000'  0608         1162          DEC   PCNT                    PARAM-CNT = PARAM-CNT - 1.
                    1163+         DOIF  ,GE,,,,                 IF PARAM-COUNT >= 0,
0002'  1105         1339E           JLT   91$
0004'  05BA         1469            INC   POP                       PARM-ARRAY-OFS = PARM-ARY-OFS + 1
0006'  C007         1470            MOV   POA,TMP                   GET NEXT OFFSET FROM ARRAY TABLE
0008'  A00A         1471            A     POP,TMP
                    1472+           MOVBRJ *TMP,TOF
000A'  D250         1474A           MOVB  *TMP,TOF
000C'  0989         1475A           SRL   TOF,8
                    1477+           ENDBLK                        IFEND
000E'               1609E 91$
000E'  045B         1697          RT
                    1698  ********************************************************************
                    1699          END
```

No errors detected

REFERENCES

The following is a list of references used. Most of them are specifically mentioned in the text, while others were pertinent and informative sources which were found to be helpful during the development of the project.

1. *"ARINC Characteristic 717 Specifications" :*
   Airline Electronics Industry Committee. Published by Aeronautical Radio Incorporated, Maryland, USA, 1979.

2. *"ARINC Characteristic 429 Specifications" :*
   Airline Electronics Industry Committee. Published by Aeronautical Radio Incorporated, Maryland, USA, 1979.

3. *"Texas Instruments 9900 Family Systems Design Handbook" :*
   First Edition. William D. Simpson, Gerald Luecke, Don L. Cannon, David H. Clemens, Texas Instruments Engineering Staff, Houston, USA, 1978. Library Of Congress Catalog Number: 78-058005.

4. *"Structured Analysis and System Specifications" :*
   Tom De Marco. Published by Yourdon Incorporated, New York, USA, 1978.

5. *"Real-Time Microprocessor Systems" :*
   First Edition. Stephen Savitzky. Published by Prentice-Hall, USA, 1982.

6. *"Current Practices in Software Development" :*
   First Edition. David King. Published by Yourdon Incorporated, New York, USA, 1984.

7. *"DEC RSX-11M+ Operating System Reference Manual" :*
   Digital Equipment Corporation, Massachusetts, USA, 1979.

8. *"TMS9900 Assembly Language Reference Manual" :*
   Second Edition. Texas Instruments, Houston, Texas, USA 1978.

9. *"TMS9900 Cross-Assembler Version 10.34 Reference Manual" :*
   Boston Systems Office Incorporated, Massachussetts, USA, 1978.

10. *"Boston Systems Office Cross Linkage Editor Version 5.01" :*
    Reference Manual : Boston Systems Office Incorporated, Massachussetts, USA, 1978.

11. *"Boston Systems Office Code Converter Version 3.17" :*
    Reference Manual : Boston Systems Office Incorporated, Massachussetts, USA, 1978.

12. *"Boeing 757/767 Airborne Computer Software Standards" :*
Boeing Aircraft Company, Seattle, USA, 1978,
Document Number D6T10740-1.

13. *"Techniques of Program Structure and Design" :*
Edward Yourdon. Published by Prentice Hall Incorporated,
USA, 1975.

14. *"FAA Intrument Flight Handbook" :*
United States Department Of Transportation,
Federal Aviation Administration, USA, 1980.

15. *"IRIG Telemetry Standard 106-80" :*
Telemetry Group, Range Commander's Council,
White Sands Missile Centre,New Mexico, USA, 1980.

16. *"The Practical Guide to Structured Systems Design" :*
Meilir Page-Jones. Published by Yourdon Incorporated,
New York, USA, 1980.

17. *"Writings Of The Revolution" :*
Selected Readings on Software Engineering.
Edited By Edward Yourdon. Published by Yourdon Incorporated,
New York, USA, 1982.

18. *"Digital Expandable Data Acquisition and Recording System" :*
Airlines Electronic Engineering Committee,
Letter 78-114/AIDS-107, September 1978.

19. *"Mark 33 Digital Information Transfer System (DITS)" :*
Airlines Electronic Engineering Committee,
Letter 77-079/SAI-46, September 1978.

20. *"Universal Flight Data Recorder" :*
Product Specification. Sunstrand Data Control,
Washington, USA, 1980.

21. *"16-bit Microprocessor performs like a minicomputer" :*
Alan Lofthus, Deene Ogden.
Electronics, May 1976.

22. *"Data Base Management Systems" :*
First Edition. Alfonso F. Cardenas.
Published by Allyn and Bacon Incorporated, USA, 1979.

23. *"Numerical Analysis" :*
Second Edition. Richard L. Burden, J. Douglas Faires.
Published by Prindle, Weber and Scmidt, Boston, 1981.

24. *"An Introduction to Data Structures with Applications" :*
First Edition. Jean-Paul Tremblay, Paul G. Sorenson.
Published by McGraw-Hill, USA, 1976

25. *"Numerical Mathematics In Computing" :*
First Edition. Ward Cheney, David Kincaid.
Published by Brooks/Cole, USA, 1980.

26. *"Programming Microprocessor Interfaces For Control And Instrumentation" :*
Michael Andrews. Published by Prentice-Hall, USA, 1980.

27. *"Logical Design of a Modular Data Base Machine" :*
Nima Aghelvi.
Published by Prentice-Hall, USA, 1980.