

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

# **The Development of an On-Chip-Metering Solution**

**A dissertation prepared by:**      **David Kaplen      B.Sc. (Elec. Eng.), UCT**  
**Department of Electrical Engineering**  
**University of Cape Town**


**Prepared for:**                      **Prof. J. Tapson**  
**Department of Electrical Engineering**  
**University of Cape Town**

**This dissertation is prepared in full fulfillment of the requirements for the degree  
of a Master of Science in Electrical Engineering.**

**March 2001**

## Declaration

I hereby declare that this dissertation is my own work, and unless otherwise stated, all work herein has been completed by myself. Furthermore, to my knowledge this work has not been submitted before for any degree or examination in any other University. It is being submitted for the degree of Master of Science in Electrical Engineering at the University of Cape Town.

  
..... [D.E.Kaplen]

Candidate

Match 2001

University of Cape Town

## Acknowledgements

I would like to thank the following people for their invaluable contribution towards this project:

Prof. J. Tapson, my thesis supervisor, for all his assistance and guidance throughout this project.

Len Schaller, Project Team Leader, Energy Measurements Ltd., for all his assistance throughout this project.

Shaun Parsens, Development Engineer, Energy Measurements Ltd., for his technical advise and assistance.

The team from the UCT Power Machines Laboratory, particularly Chris, for all their assistance during the testing phases of the project.

Sam, for all his advice, assistance and inspiration, without which this project would probably not have been possible.

My family, Mom, Dad, Lyndi and Wendy, for all their absolute love and support. Their encouragement and belief in me has made it possible for me to get this far.

Ayesha, for all her unconditional love and support through the many long hours. Your strength and motivation has been invaluable in overcoming the many obstacles.

## Terms of Reference

Working in conjunction with a Technical Design Team from EML, and as stipulated by them, the scope of this dissertation is to:

- investigate the fundamental theory behind power and energy measurement
- investigate and provide a comparative analysis of various power measurement techniques, including commercially available solutions. This investigation includes a comparative analysis of aspects such as required processor overhead, signal conditioning requirements, cost, accuracy and compliance with IEC1036 regulations.
- investigate the viability of implementing each of the investigated techniques on a MCU-based system for the purpose of OCM
- design and developed a proposed OCM solution, including a firmware algorithm and hardware prototype based on the findings of the investigation
- test the proposed solution and determine its compliance with IEC1036 specifications
- compile a complete report of all findings and results

## Synopsis

Energy Measurements Ltd (EML), a joint venture partnership between Siemens and Spescom, manufacture prepaid electricity utility meters for both the local and international markets. Under the brandname CASHPOWER 2000, EML produces single and polyphase prepayment utility meters.

Currently, these meters currently utilise a separate module for the measuring of electrical energy. In order to reduce component costs, EML proposed the energy measurement be conducted by the onboard Microcontroller Unit (MCU), a term known as On-Chip-Metering (OCM). It is envisioned that this would quickly translate in an increase in revenue.

However, a major concern regarding this has been the increase in the required processor overhead. The CASHPOWER 2000 embedded MCU would be required to conduct all the present metering functionality in addition to the energy measurement. This, together with the cost analysis and compliance with the stipulated IEC1036 regulations, constitute the key criteria in determining the projects viability.

This dissertation represents the investigative and development stages of a prototype algorithm and accompanying peripheral hardware as a possible solution for OCM. As part of the preliminary research, several examples of digital power and energy measurement techniques were investigated. A comparative analysis of these was performed to facilitate the development of a unique solution based on the research conducted. This completed, a prototype was developed and preliminary testing was conducted to determine its compliance with the stipulated regulations for a class 2 meter, as per IEC1036 specifications.

During the course of the investigation it was found that methods of energy and power measurement using continuous analogue signals did not comply with the cost constraints and were eliminated as possible solutions. Furthermore, methods involving the computation of RMS values of current and voltage were also not viable as these require more processor overhead than is acceptable.

The employed method involves the computation of instantaneous power information by multiplying digitised samples of voltage and of current and then averaging to yield average power information. This in turn is used to generate output pulses proportional to the expired energy.

A two-stage gain design employing automatic ranging was utilised to increase the effective operating range. The analogue input signals to both channels are pre-conditioned prior to sampling to remove any high-frequency noise and to provide correct scaling. All analogue offsets in the two current signals are removed via an offset-nulling feedback configuration in the signal conditioning circuitry.

The required processor overhead for the developed algorithm was measured as approximately 0.115 MIPS. Motorola, the MCU manufacturer, stipulate the device may operate at up to around 2.08 MIPS with a MCU Bus frequency of 8.33MHz. Thus the required overhead constitutes about 6% of the processor time. The memory requirements are 1.036 kB of non-volatile ROM and just 22 bytes of volatile RAM.

Various software and hardware-based simulations were conducted to determine the optimal system parameters, and also to provide an indication of the system accuracy in the presence of ambient noise, line harmonics, a dynamic power factor, and line frequency fluctuations. Hardware testing was performed using purely resistive and resistive-inductive loads over a range of power factors and loads.

It was found that the developed solution did not conform to the required IEC1036 specifications for a class 2 meter. The output error results exceeded the  $\pm 2\%$  limits when measuring loads towards the lower operating range, particularly for non-unity power factors. Thus preliminary testing indicated that the OCM project might not in fact be viable. However, the validity of these results is questionable since appropriate testing equipment was not available, and the reference devices used contain excessively high uncertainties.

In light of this, it is recommended that the proposed solution be tested on site at EML using the appropriate testing equipment to determine the overall viability of OCM.

## Glossary

Static watt-hour meter	Meter in which current and voltage act on solid-state (electronic) elements to produce an output proportional to watt-hours
OCM	On-Chip-Metering, the measurement of consumer energy expenditure using an embedded Microcontroller Unit for prepaid utility meters
MCU	Microcontroller Unit, an Integrated Circuit containing a Central Processing Unit and various peripherals such as input/output ports, embedded non-volatile and volatile memory, embedded Analogue-to-Digital Converter, etc
ADC	Analogue-to-Digital Converter, a device used to convert a continuous analogue signal to discrete digitised samples
MUX	Multiplexor, a device used to switch one of several inputs (either digital or analogue) to a single output
RAM	Random Access Memory, volatile memory that may be updated during run-time
ROM	Read Only Memory, non-volatile memory that is pre-programmed and cannot be changed during run-time
RMS	Root-Mean-Square
S/H	Sample and Hold, a term describing the capturing and temporary storing of a signal
LCD	Liquid Crystal Display
BCD	Binary-Coded-Decimal, a method of storing decimal values using the base 2
DC	Direct Current
AC	Alternating Current
ASIC	Application Specific Integrated Circuit, a dedicated device designed exclusively for a distinct function
PWM	Pulse Width Modulation
RC	Resistor-Capacitor
PGA	Programmable Gain Amplifier, an analogue amplifier the gain of which is digitally programmable
HPF	High Pass Filter, a filter that attenuates all frequencies beyond its 3dB corner frequency
LPF	Low Pass Filter, a filter that attenuates all frequencies below its 3dB corner frequency
MIPS	Millions of Instructions per Second, a term used to quantify the performance speed of a processor
IEC	International Electrotechnical Commission
ANSI	American National Standards Institute

# Table of Contents

<b>Declaration.....</b>	<b>ii</b>
<b>Acknowledgements.....</b>	<b>iii</b>
<b>Terms of Reference.....</b>	<b>iv</b>
<b>Synopsis.....</b>	<b>v</b>
<b>Glossary .....</b>	<b>vii</b>
<b>List of Illustrations.....</b>	<b>xiv</b>
<b>List of Tables.....</b>	<b>xvii</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 Problem Statement.....	1
1.2 Definition of OCM.....	1
1.3 Background to OCM Project.....	1
1.4 Significance of Immediate Study.....	2
1.5 Scope of Project.....	2
1.6 Limitations of Project.....	3
<b>2 Power and Energy Measurement Theory.....</b>	<b>4</b>
2.1 Mathematical Analysis of Power and Energy Measurement.....	4
2.1.1 Power Analysis.....	5
2.1.2 Energy Analysis.....	7
2.2 Power, Energy and RMS Measurement Using Discrete Values.....	8
2.3 Influence of Non-Unity Power Factors and Nonsinusoidal Signals.....	9
2.3.1 Non-Unity Power Factor Influences.....	9
2.3.2 Validity of Power Calculation in the Presence of Harmonic Distortions.....	11

2.4	Definition of System Error.....	13
<b>3</b>	<b>Microcontroller Considerations.....</b>	<b>14</b>
<b>4</b>	<b>Investigated Techniques.....</b>	<b>17</b>
4.1	Digital Wattmeter Using a Sampling Method.....	17
4.1.1	Overview.....	17
4.1.2	Methodology.....	17
4.1.3	System Hardware.....	19
4.1.3.1	Signal Conditioning.....	19
4.1.3.2	Preamplifiers.....	19
4.1.3.3	Timing Pulse Generator.....	20
4.1.3.4	ADCs.....	20
4.1.3.5	Computing Circuitry.....	21
4.1.4	Software.....	21
4.1.5	Measurement Procedure.....	21
4.1.6	Results.....	22
4.2	Digital Sampling Laboratory Wattmeter.....	24
4.2.1	Overview.....	24
4.2.2	Configuration.....	24
4.2.3	Digital Sampling Technique.....	25
4.2.4	Analogue Signal Conditioning.....	26
4.2.5	Sampling Interval Generator.....	27
4.2.6	Results.....	30
4.3	Digital Power Metering Manifold.....	31
4.3.1	RMS-Value Calculating Algorithm.....	31
4.3.2	Real Power Calculation.....	32
4.4	Design of a Microprocessor-Based Digital Wattmeter.....	33
4.4.1	Overview.....	33
4.4.2	System Hardware.....	33
4.4.3	System Operation.....	35
4.4.4	Power Measurement Algorithm.....	36
4.5	Fuzzy Based Adaptive Digital Power Metering Using a Genetic Algorithm.....	38
4.5.1	Overview.....	38
4.5.2	Genetic Algorithm Overview.....	39
4.5.3	Fuzzy Knowledge Base Tuning Using a Genetic Algorithm.....	40
4.5.4	Fuzzy-Based Adaptive Power Metering Algorithm.....	41
4.5.5	Results.....	42
4.6	RMS Value Method.....	43
4.6.1	Overview.....	43
4.6.2	Obtaining the Input RMS Values.....	44
4.6.2.1	RMS-to-DC Producing Transducers.....	44

4.6.2.2	<i>RMS-to-DC ASICs</i> .....	46
4.6.2.3	<i>On-Board RMS Calculations Using Discrete Samples</i> .....	47
4.6.3	Obtaining the Power Factor.....	51
<b>4.7</b>	<b>Reduced Scan Principle</b> .....	<b>55</b>
4.7.1	Overview.....	55
4.7.2	Power Calculation Method.....	56
4.7.3	Inherent Error Compensation.....	57
4.7.4	Results of Simulated Data.....	58
<b>4.8</b>	<b>Power Measurement Using Rectified Inputs</b> .....	<b>60</b>
4.8.1	Overview.....	60
4.8.2	Increased Resolution.....	61
4.8.3	Obtaining Rectified Inputs.....	62
4.8.4	Rectification Test Results.....	63
<b>4.9</b>	<b>Microchip Microcontroller Based Watt-Hour Meter Reference Design</b> ..	<b>65</b>
4.9.1	PICREF-3 Overview.....	65
4.9.2	System Overview.....	66
4.9.3	Hardware Overview.....	67
4.9.4	Microcontroller Input/Output.....	69
4.9.5	Signal Measurements.....	71
4.9.6	Power Calculations.....	72
4.9.7	Firmware Details.....	73
4.9.8	Test Results.....	74
<b>4.10</b>	<b>Analog Devices AD7755 Metering ASIC</b> .....	<b>75</b>
4.10.1	Overview.....	75
4.10.2	Signal Conditioning.....	77
4.10.2.1	<i>External Analogue Signal Conditioning</i> .....	78
4.10.2.2	<i>Internal Analogue Signal Conditioning</i> .....	79
4.10.3	Analogue to Digital Conversion.....	80
4.10.4	HPF and Offset Effects.....	80
4.10.5	Power Calculation Methodology.....	81
4.10.6	Outputs.....	82
4.10.7	System Errors.....	83
4.10.8	Operating Range.....	83
<b>5</b>	<b>Rejected Techniques</b> .....	<b>84</b>
5.1	A Fuzzy Based Adaptive Power Metering System Utilising a Genetic Algorithm.....	85
5.2	The Reduced Scan Principle.....	85
5.3	Power Measurements Using Rectified Inputs.....	87
5.4	Power Measurement Using RMS Values of Current and Voltage.....	87
5.4.1	On-Board RMS Calculation Using Discrete Samples.....	88

5.4.2	Power Factor Calculation.....	88
5.5	The PICREF-3 System - A Microchip MCU Based Approach.....	92
<b>6</b>	<b>Techniques Considered Further.....</b>	<b>93</b>
6.1	Considerations for OCM Implementation Based on the AD7755.....	95
6.1.1	Peripherals.....	96
6.1.2	Sampling - Analogue to Digital Conversion.....	96
6.1.3	Internal Signal Conditioning.....	97
6.1.4	Power Calculation Methodology.....	97
6.1.5	Outputs.....	97
6.2	Digital Wattmeter Using a Sampling Method.....	98
6.2.1	Analogue Signal Conditioning.....	98
6.2.2	Sampling Timing and Resolution.....	99
6.2.3	Power Computation Algorithm.....	100
6.3	Digital Power Metering Manifold.....	100
6.4	Digital Sampling Laboratory Wattmeter.....	101
<b>7</b>	<b>Preliminary OCM Design Considerations.....</b>	<b>102</b>
7.1	System Tradeoffs.....	103
7.2	Transducer Considerations.....	104
7.2.1	Current Transformer.....	106
7.2.2	Current Shunt.....	110
7.2.3	Voltage (Potential) Transformer.....	112
7.2.4	Resistive Network as a Voltage Divider.....	113
7.2.5	RMS Value Producing Current and Voltage Transducers.....	115
7.3	Preliminary Conclusions Regarding Final OCM Design.....	116
<b>8</b>	<b>Proposed OCM Solution.....</b>	<b>117</b>
8.1	Overview.....	117
8.1.1	Test Version.....	117
8.1.2	Final OCM Solution.....	118
8.2	Peripherals.....	120
8.2.1	Two Stage Current Gain.....	120
8.2.2	Input Signal Offset Compensation.....	121
8.2.3	Timing Parameters.....	122
8.2.4	Output Frequency Generation.....	126

<b>8.3</b>	<b>Algorithm.....</b>	<b>128</b>
8.3.1	Overview.....	128
8.3.2	Processor and Memory Overhead.....	131
<b>8.4</b>	<b>Hardware.....</b>	<b>133</b>
8.4.1	OCM Signal Conditioning.....	133
8.4.1.1	<i>Current-Signal Processing Hardware.....</i>	<i>136</i>
8.4.1.2	<i>Choice of Operational Amplifiers.....</i>	<i>139</i>
8.4.1.3	<i>Gain and Phase Response.....</i>	<i>141</i>
8.4.1.4	<i>Voltage-Signal Processing Hardware.....</i>	<i>143</i>
8.4.1.5	<i>Anti-Alias Filters.....</i>	<i>144</i>
8.4.2	Power Supply.....	146
8.4.3	Noise and Interference Reduction.....	147
8.4.3.1	<i>Grounding.....</i>	<i>148</i>
8.4.3.2	<i>EMI Shielding.....</i>	<i>150</i>
<b>9</b>	<b>Microsoft Excel Simulations.....</b>	<b>152</b>
9.1	Input Gain Settings.....	153
9.2	Sampling Parameters.....	156
9.3	Digital Signal Production.....	157
9.4	Simulation Macros.....	161
9.5	Results.....	162
<b>10</b>	<b>Testing Results.....</b>	<b>169</b>
10.1	Tests Using Simulated Inputs.....	169
10.1.1	Scope of Simulated Testing.....	170
10.1.2	Sampled Signals.....	171
10.1.3	Output Results.....	173
10.1.4	Possible Sources of Error.....	178
10.2	Testing Using Actual Loads.....	179
10.2.1	Test Results.....	179
10.2.2	Possible Sources of Error.....	182

11	Main Findings.....	184
12	Main Conclusions.....	185
13	Recommendations.....	186
14	Bibliography.....	187

**Annexure A - Hardware Schematics**

**Annexure B - Firmware/Software Listing**

**Annexure C - Test Equipment Specifications**

University of Cape Town

## List of Illustrations

<b>Figure 2.1:</b>	<i>Instantaneous Power Flow</i>	5
<b>Figure 2.2(a):</b>	<i>Power Flow Circuit Under Steady State</i>	6
<b>Figure 2.2(b):</b>	<i>Phasor Representation of Signals in Figure 2.2(a)</i>	7
<b>Figure 2.3(a):</b>	<i>Real Power Information from Instantaneous Power for PF = 1</i>	10
<b>Figure 2.3(b):</b>	<i>Real Power Information from Instantaneous Power for PF = 0.5</i>	10
<b>Figure 3.1:</b>	<i>A Graphical Representation of Actual 8-bit ADC Dynamic Error</i>	16
<b>Figure 4.1:</b>	<i>Functional Block Diagram of Digital Wattmeter</i>	19
<b>Figure 4.2:</b>	<i>Basic Configuration of Digital Sampling Wattmeter</i>	25
<b>Figure 4.3:</b>	<i>Analog Signal Conditioning Circuitry for Laboratory Wattmeter</i>	27
<b>Figure 4.4:</b>	<i>Sampling Pulse Generator</i>	29
<b>Figure 4.5:</b>	<i>MCU-Based System Block Diagram</i>	34
<b>Figure 4.6:</b>	<i>Flow Diagram for the Wattmeter Measurement Process</i>	35
<b>Figure 4.7:</b>	<i>Block Diagram for the Power-to-Frequency Converter</i>	36
<b>Figure 4.8:</b>	<i>The General Concept of a Genetic Algorithm</i>	39
<b>Figure 4.9:</b>	<i>An Example of the Difference in Peak Values for V and I</i>	51
<b>Figure 4.10:</b>	<i>Graph showing Sampling points for Reduced Scan Method</i>	57
<b>Figure 4.11:</b>	<i>Proposed Rectification Process Block Diagram</i>	61
<b>Figure 4.12:</b>	<i>Rectified Input Signal</i>	61
<b>Figure 4.13:</b>	<i>Developed Voltage Rectification Circuit Schematic</i>	62
<b>Figure 4.14:</b>	<i>Developed Current Rectification Circuit Schematic</i>	63
<b>Figure 4.15:</b>	<i>PICREF-3 Watt-Hour Meter Block Diagram</i>	66
<b>Figure 4.16:</b>	<i>High-voltage AC Electronic Block Diagram</i>	68
<b>Figure 4.17:</b>	<i>Small Signal Analog/Digital V-sense Electronics Block Diagram</i>	68
<b>Figure 4.18:</b>	<i>Small Signal Analog/Digital I-sense Electronics Block Diagram</i>	69
<b>Figure 4.19:</b>	<i>Microcontroller I/O Block Diagram</i>	70
<b>Figure 4.20:</b>	<i>Measurement Loop Block Diagram</i>	73
<b>Figure 4.21:</b>	<i>Functional Block Diagram (AD7755)</i>	75
<b>Figure 4.22:</b>	<i>Test Circuit for AD7755</i>	77
<b>Figure 4.23:</b>	<i>Typical Connection for AD7755 Channel 1</i>	78
<b>Figure 4.24:</b>	<i>Typical Connections for AD7755 Channel 2</i>	79
<b>Figure 4.25:</b>	<i>Maximum Signal Levels for AD7755 Channel 1 (Gain=1)</i>	79
<b>Figure 5.1:</b>	<i>Current Signal with Strong 3<sup>rd</sup> Harmonic Content</i>	90
<b>Figure 5.2:</b>	<i>Input Waveforms with DC Offset in Current Channel</i>	91

<b>Figure 7.1:</b>	<i>Typical Connections for Standard Current Transformer</i>	106
<b>Figure 7.2:</b>	<i>Typical Connections for High-Precision Transformer</i>	107
<b>Figure 7.3:</b>	<i>Typical Shunt Connection</i>	110
<b>Figure 7.4:</b>	<i>Typical Connection for a Voltage Transformer</i>	112
<b>Figure 7.5:</b>	<i>Typical Resistive Voltage Divider</i>	113
<b>Figure 8.1:</b>	<i>Functional Block Diagram of Final OCM Solution</i>	119
<b>Figure 8.2:</b>	<i>Unipolar Signal Offset Compensation</i>	121
<b>Figure 8.3:</b>	<i>Output Pulse Generation Mechanism</i>	124
<b>Figure 8.4:</b>	<i>Final OCM Algorithm Flow Diagram</i>	129/130
<b>Figure 8.5:</b>	<i>Complete Hardware System</i>	135
<b>Figure 8.6:</b>	<i>Shielded Circuitry</i>	135
<b>Figure 8.7:</b>	<i>Overview of Current Signal Conditioning Hardware</i>	136
<b>Figure 8.8:</b>	<i>Input-Current Signal Conditioning Schematic</i>	137
<b>Figure 8.9:</b>	<i>Shunt Coupling</i>	137
<b>Figure 8.10:</b>	<i>Simulated Circuit using MicroSim Schematic</i>	141
<b>Figure 8.11:</b>	<i>Gain Response for Simulated Circuit</i>	142
<b>Figure 8.12:</b>	<i>Schematic Representing the Voltage Division Circuitry</i>	144
<b>Figure 8.13:</b>	<i>Aliasing Effects</i>	145
<b>Figure 8.14:</b>	<i>MCU ADC Interfacing</i>	146
<b>Figure 8.15:</b>	<i>Bipolar Power Supply Schematic</i>	147
<b>Figure 8.16:</b>	<i>Equivalent Shunt Circuit</i>	148
<b>Figure 8.17:</b>	<i>Actual Shunt Circuitry</i>	149
<b>Figure 8.18:</b>	<i>Effects of Parasitic Shunt Inductance</i>	150
<b>Figure 9.1:</b>	<i>Input Parameters for Microsoft Excel Simulation Worksheet</i>	152
<b>Figure 9.2:</b>	<i>Graph Depicting Simulated Current Signals with a 3A Load Current</i>	155
<b>Figure 9.3:</b>	<i>Graph Depicting Simulated Current Signals with a 7A Load Current</i>	155
<b>Figure 9.4:</b>	<i>Graph Depicting Simulated Current Signals with a 60A Load Current</i>	156
<b>Figure 9.5:</b>	<i>Continuous Signals Produced by Microsoft Simulation Worksheet</i>	159
<b>Figure 9.6:</b>	<i>Discrete Signals Produced by Microsoft Excel Simulation Worksheet</i>	160
<b>Figure 9.7:</b>	<i>Optimal System Simulated Results</i>	164
<b>Figure 9.8:</b>	<i>Optimal System Simulated Results with a 10-bit ADC</i>	166
<b>Figure 9.9:</b>	<i>Simulated Results for an 8-bit ADC and a 1024Hz Sampling Frequency</i>	167
<b>Figure 9.10:</b>	<i>Simulated Results for a 10-bit ADC and a 1024Hz Sampling Frequency</i>	168
<b>Figure 10.1:</b>	<i>Sampled Signals using Simulated Inputs</i>	171
<b>Figure 10.2:</b>	<i>Sampled Signals without Offset Nulling</i>	172
<b>Figure 10.3:</b>	<i>System Output Error Using Simulated Inputs</i>	173

<b>Figure 10.4:</b>	<i>Non-unity Power Factor Signals</i>	176
<b>Figure 10.5:</b>	<i>Linearity Comparison - Low Gain Channel</i>	177
<b>Figure 10.6:</b>	<i>Linearity Comparison - High Gain Channels</i>	177
<b>Figure 10.7:</b>	<i>Load Configuration Used for Testing Purposes</i>	179
<b>Figure 10.8:</b>	<i>OCM Test Version Output Error for PF = 1</i>	181
<b>Figure 10.9:</b>	<i>Captured Waveforms</i>	183

University of Cape Town

## List of Tables

<b>Table 4.1:</b>	<i>Results for the System Taken at Full Scale</i>	22
<b>Table 4.2:</b>	<i>Factors Contributing to Overall Uncertainty at the 99% Confidence Interval</i>	30
<b>Table 4.3:</b>	<i>Number of Samples, Sampling Period and Measurement Period</i>	37
<b>Table 4.4:</b>	<i>CR4310 RMS Transducer Specifications</i>	45
<b>Table 4.5:</b>	<i>CR4310 Input Range and Accuracy</i>	45
<b>Table 4.6:</b>	<i>Comparison of e Value using Empirical and Computed Methods</i>	59
<b>Table 4.7:</b>	<i>Output Error using Reduced Scan on Simulated Data</i>	59
<b>Table 4.8:</b>	<i>Simulated Test Results for Rectified Signals</i>	64
<b>Table 4.9:</b>	<i>Channel 1 Gain and Corresponding Maximum Input Signal</i>	80
<b>Table 5.1:</b>	<i>Summary – Rejected Techniques</i>	84
<b>Table 6.1:</b>	<i>Summary – Techniques Considered Further</i>	94
<b>Table 7.1:</b>	<i>Summary of Transducer Considerations</i>	105
<b>Table 7.2:</b>	<i>ANSI Current Transformer Accuracy for Various Burdens</i>	108
<b>Table 8.1</b>	<i>Required Processor Overhead Breakdown</i>	132
<b>Table 8.2</b>	<i>Phase Response of Simulated Circuit</i>	145
<b>Table 10.1</b>	<i>Output Results using Simulated Inputs</i>	174
<b>Table 10.2</b>	<i>Output Results for Non-Unity Power Factor using Simulated Inputs</i>	175
<b>Table 10.3</b>	<i>Results using OCM Test Version with Actual Loads</i>	180
<b>Table 10.4</b>	<i>OCM Final Version Test Results</i>	181

## Section 1: Introduction

An investigation into the viability of developing a unique microcontroller-based energy measurement technique and integrating it with the next generation consumer prepaid electricity utility meters is presented in this dissertation. A comparative analysis of various power and energy measurement techniques is also presented, as this provided the basis for developing the proposed solution.

### 1.1 Problem Statement

To conduct a comparative analysis of possible techniques for the implementation of On Chip Metering (OCM), and to develop a proposed solution based on the findings of the analysis.

### 1.2 Definition of OCM

The CASHPOWER units from EML currently utilise a Toshiba MCU as the core of their operation. This device controls all aspects of the meter, including the LCD display, keypad scanning, the tokenless credit transfer encryption, etc, but does not conduct the actual energy measuring. It may prove viable to eliminate the metering ASIC and peripheral circuitry by conducting energy measurement directly onboard the MCU. This defines the essence of On Chip Metering (OCM).

### 1.3 Background to OCM Project

Pre-paid electricity metering solutions were introduced into both the local and international markets through Energy Measurements Ltd (EML), a joint venture partnership between Siemens and Spescom. Under the brand name CASHPOWER 2000, EML produces single and polyphase prepayment utility meters, using a keypad based credit transfer mechanism.

EML are constantly seeking new ways to reduce production costs while maintaining a high level of quality. The OCM project was proposed as one possible way to achieve both of these requirements.

## 1.4 Significance of Immediate Study

The pre-payment meters currently produced by EML utilise a separate module for the metering of electrical energy. Should On Chip Metering (OCM) become viable, EML can expect a decrease in manufacturing costs through a decreased component count, which quickly translates to an increase in revenue collection.

The significance of the research conducted for this report is that a comparative assessment of possible OCM solutions and commercially available solutions is attained, facilitating the development phase of the project. The significance of the solution proposed in this dissertation is that it provides an indication of the processor and hardware requirements of a working prototype. Furthermore, the results of the preliminary testing conducted further facilitate the viability investigation.

## 1.5 Scope of Project

Working in conjunction with a Technical Design Team from EML, and as stipulated by them, the scope of this dissertation is to:

- investigate the fundamental theory behind power and energy measurement
- investigate and provide a comparative analysis of various power measurement techniques, including commercially available solutions. This investigation includes a comparative analysis of aspects such as required processor overhead, signal conditioning requirements, cost, accuracy and compliance with IEC1036 regulations.
- investigate the viability of implementing each of the investigated techniques on a MCU-based system for the purpose of OCM
- design and developed a proposed OCM solution, including a firmware algorithm and hardware prototype based on the findings of the investigation
- test the proposed solution and determine its compliance with IEC1036 specifications

- compile a complete report of all findings and results

## 1.6 Limitations of Project

The main limitations of this project included:

- Time constraints. Research for the OCM project began on approval of the project proposal in March 2000 and was to be concluded in February 2001.
- Financial constraints
- Limited access to appropriate testing equipment
- Limited access to technical literature (mainly IEEE Transaction Journals, the University Library, and the Internet)
- Limited consultation with EML. With the company based in Isando, Gauteng and the investigation conducted at the University of Cape Town, communication with EML was restricted to telephonic consultations and a single visit to the plant in November 1999 to conduct a presentation of preliminary research.

## Section 2 - Power and Energy Measurement Theory

The purpose of this chapter is twofold: (1) to briefly review some of the basic theory, definitions and concepts that are essential to the measurement of electrical energy and (2) to introduce simplifying assumptions that allow for easier evaluation of power analysis along with effects of using these assumptions.

An attempt has been made to use Institute of Electrical and Electronics Engineers (IEEE) and International Electrotechnical Commission (IEC) standard letter and graphic symbols as much as possible. Moreover, the units used belong to the International System of Units (SI). Unless otherwise stated, lowercase letters are used to represent instantaneous values of quantities that may vary as a function of time, and uppercase letters are used to represent either the average or rms values. Uppercase letters in bold represent complex values or vectors. The positive direction of a power or current quantity is denoted explicitly by respective arrows in the various schematics and diagrams. The voltage at any node is defined with respect to the circuit analogue ground, for example  $v_a$  is the voltage at node  $a$  with respect to ground. Differential voltages are indicated denoted by including the two nodes' references as follows:  $v_{ab}$  refers to the voltage of node  $a$  with respect to node  $b$ , where  $v_{ab} = v_a - v_b$ .

### 2.1 Mathematical analysis of power and energy measurement

All methods for measuring both real and complex power and corresponding energy have their fundamental roots based on the equations relating power, energy, voltage and current discussed in this theoretical section.

### 2.1.1 Power Analysis

The following figure represents a power flow between two alternating voltage (AC) single-phase subcircuits:

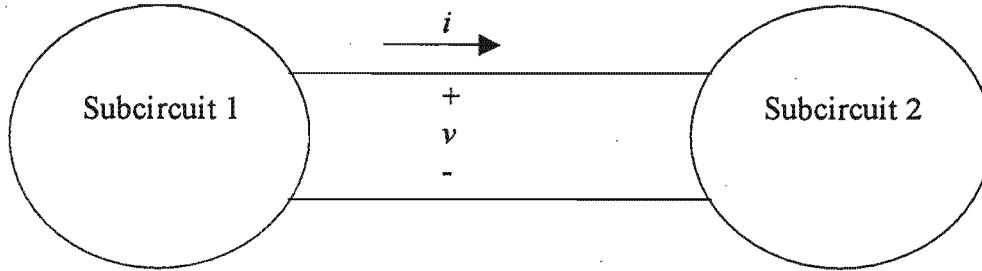


Figure 2.1 – Instantaneous power flow

By definition<sup>[1]</sup>, the instantaneous power flow from subcircuit 1 to subcircuit 2 is:

$$p(t) = v(t)i(t) \quad [2.1]$$

If  $v$  and  $i$  waveforms repeat with a time period of  $T$  [sec] then the average power flow from subcircuit 1 to subcircuit 2 can be calculated as:

$$P_{ave} = \frac{1}{T} \int_0^T p(t).dt = \frac{1}{T} \int_0^T v(t)i(t).dt \quad [2.2]$$

In the below schematic, the voltage source represents subcircuit 1 of Figure 2.1 above, and subcircuit 2 is represented by a resistive inductive load under steady state conditions.

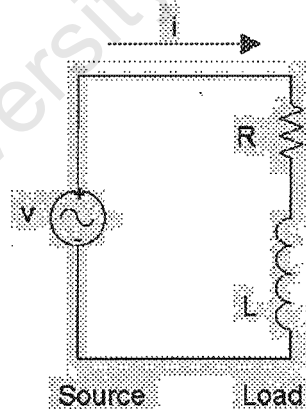


Figure 2.2(a) – Power flow circuit under steady state

Disregarding line distortions, spikes and harmonics, the voltage and current signals present above are as follows:

$$v_{inst}(t) = \sqrt{2}V_{RMS} \cos(\omega t) \quad \text{and} \quad i_{inst}(t) = \sqrt{2}I_{RMS} \cos(\omega t - \phi) \quad [2.3]$$

, where  $\omega$  = line frequency in  $\text{rad.s}^{-1}$  and  $\phi$  = phase delay in radians.

Conversely, the rms values of  $V$  and  $I$  may be calculated from the instantaneous ones as follows [1]:

$$V_{RMS} = \sqrt{\frac{1}{T} \int_0^T (v_{inst})^2 dt} \quad \text{and} \quad I_{RMS} = \sqrt{\frac{1}{T} \int_0^T (i_{inst})^2 dt} \quad [2.4]$$

Since both these instantaneous values vary sinusoidally with time at the same frequency, they can be represented in a complex plane by means of the projection of the rotating phasors to the horizontal real axis, as shown in Figure 2.2(b). These phasors rotate with an angular frequency  $\omega$ , and are represented by:

$$\mathbf{V} = V e^{j\theta} = V \quad \text{and} \quad \mathbf{I} = I e^{-j\phi} \quad [2.5]$$

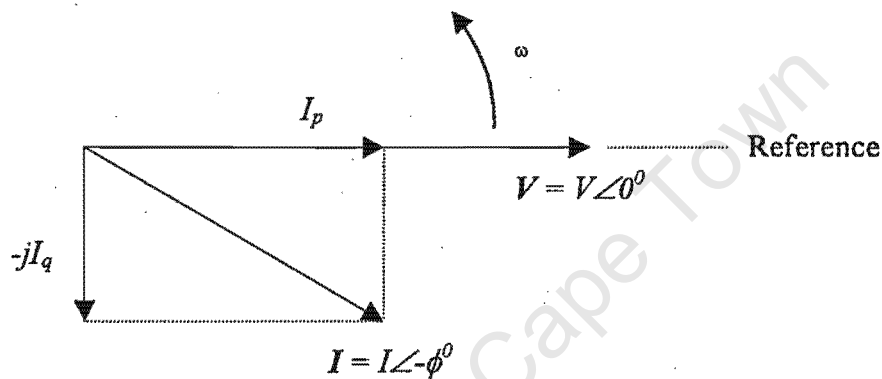


Figure 2.2(b) – Phasor representation of signals in Figure 2.2(a)

From this it is clear that the power delivered to the load is complex, consisting of two parts: an imaginary component arising from the imaginary current  $I_q$ , and a real component arising from the real current  $I_p$ .

By definition [1], the complex power delivered from the source to the load is:

$$\mathbf{S} = \mathbf{V} \mathbf{I}^* = V e^{j\theta} \times I e^{j\phi} = VI e^{j\phi} \quad [2.6]$$

The magnitude of this (in volt-amperes) provides the apparent power, as follows:

$$S = VI \text{ [VA]} \quad [2.7]$$

The real average power (in watts) is defined as:

$$P = \text{Re}[\mathbf{S}] = VI \cos \phi \text{ [W]} \quad [2.8]$$

which is expressed as a product of the rms voltage  $V$  and the current component  $I_p = I \cos \phi$ , which is the component in phase with the voltage in Figure 2.3(b). In the diagram, only  $I_p$  ( $= I \cos \phi$ ) is responsible for the power transfer, and not the out-of-phase component,  $I_q$ . The out-of-phase current component yields the reactive power (in volt-ampere-reactive).

Defining the complex power  $S = P + jQ$  and using Equations 2.6 and 2.7 above, this results in:

$$Q = VI \sin \phi \quad [\text{VAR}] \quad [2.9]$$

With the sign of the phase lag  $\phi$  as defined Equation 2.3, the inductive load shown in Figure 2.2(a) has a positive value of  $\phi$ . In accordance with Equation 2.9, an inductive load draws positive, or lagging, vars. Conversely a capacitive load draws negative, or leading, vars. In this instance, the load supplies positive vars to the electrical source.

For the purpose of power measurement for OCM, the physical significance of  $S$ , and  $Q$  as individual components are disregarded, however together they constitute the real power  $P$ , which represents the rate of useful work being performed as well as the power losses.

### 2.1.2 Energy Analysis

In order for Electricity Utility Service Providers to bill consumers according to their respective tariff schemes, it is necessary to measure the energy consumption of each consumer. In South Africa, the majority of household consumers are billed on a regular basis according to a Flat Rate scheme, as opposed to a Time Of Use or Peak Usage Scheme.

This introduces the need for an integral record (in watt-hours or kilowatt-hours) of the real power with respect to time, which denotes the total energy expenditure over a given period as follows:

$$\text{Energy Expenditure} = \int_0^{T_a} P \cdot dt \quad [\text{Wh}] \text{ or } [\text{kWh}]$$

$T_a$  represents the billing period. [2.10]

This scheme is also employed in prepaid services, with the exception that credit tokens are used to represent the prepaid credit and energy expenditure is assessed at discrete regular intervals.

## 2.2 Power, Energy and RMS measurement using discrete values

The power and energy calculations defined in Sections 2.1.1 and 2.1.2 are true only for continuous voltage and current signals. The evaluation of these values by analogue means is therefore legitimate. However, as part of the scope of On Chip Metering, the power and energy calculations need to be performed using digital means. The power and energy calculations must thus be performed using discrete sampled values of the continuous voltage and current signals.

For the purpose of this derivation, the following assumptions have been made:

- Sampling is instantaneous i.e. sample time is zero seconds
- Concurrent sampling is possible i.e. no phase delay exists between concurrent samples
- Quantisation error is zero i.e. sampling resolution is assumed to be infinite
- A unity power factor exists, and all noise, harmonics and line distortions have been disregarded

The influence of harmonics, line distortions and non-unity power factors on the below power calculations is discussed in Section 2.3. The effects of the former assumptions are discussed separately within each of the relevant sections of this report.

According to Equation 2.2, the real average power in a system may be calculated by averaging the instantaneous power over some time period. By definition [2] the mathematical equivalent of the integration operation in this equation is as follows:

$$P_{ave(real)} = \frac{1}{T} \int_0^T v.i.dt = \frac{1}{T} \sum_{n=0}^{\infty} \Delta t.v_n.i_n \quad \text{as } \Delta t \rightarrow 0 \quad [2.11]$$

Where  $\Delta t$  is the time between adjacent discrete values of  $v$  and  $i$ . For a finite number of samples and sample period,  $N$  and  $\Delta t$  respectively, the above results in an approximation for  $P$  as follows:

$$P_{ave(real)} \approx \frac{1}{T} \sum_{n=0}^N \Delta t \cdot v_n i_n \quad [2.12]$$

Thus the real average power in a system may be approximated from  $N$  number of discrete samples of the voltage and current signals over a time period of  $T$ . Since  $\Delta t$  is defined by  $T$  and  $N$ , the value of  $\frac{\Delta t}{T}$  is simply  $\frac{1}{N}$ . Thus Equation 2.12 is reduced to:

$$P_{ave(real)} \approx \frac{1}{N} \sum_{n=0}^N v_n i_n \quad [2.13]$$

This has the same effect as low-pass filtering a digital signal representing instantaneous power to provide average, or real, power information. The smaller the number of samples the voltage and current's multiplication is averaged over, the closer this equation relates to the instantaneous power.

Similarly, the line RMS voltage and current may be calculated using discrete values and the mathematical equivalent of the integration operation in Equation [2.4] as follows [2]:

$$V_{RMS} = \sqrt{\frac{1}{T} \sum_{n=0}^{\infty} \Delta t \times (v_n)^2} \quad \text{and} \quad I_{RMS} = \sqrt{\frac{1}{T} \sum_{n=0}^{\infty} \Delta t \times (i_n)^2} \quad \text{as } \Delta t \rightarrow 0$$

Again, for a finite number of samples, this simplifies to:

$$V_{RMS} \approx \sqrt{\frac{1}{N} \sum_{n=0}^N (v_n)^2} \quad \text{and} \quad I_{RMS} \approx \sqrt{\frac{1}{N} \sum_{n=0}^N (i_n)^2} \quad [2.14]$$

## 2.3 Influence of non-unity power factors and nonsinusoidal signals

### 2.3.1 Non-unity power factor influences

The power factor in a system is a measure of the ratio of real power to apparent power<sub>[1]</sub>. For sinusoidal waveforms, it may be calculated as  $\cos \phi$ . The method used to extract the real power information from the instantaneous power signal as defined by Equation 2.13 is still valid even when the voltage and current signals are not in phase [3]. This can be seen by relating Equations 2.8 and 2.13:

$$P_{ave(real)} = VI \cos \phi \approx \frac{1}{N} \sum_{n=0}^N v_n i_n \quad [2.15]$$

The low pass filtering process denoted by the above relationship is clarified in Figures 2.3(a) and 2.3(b).

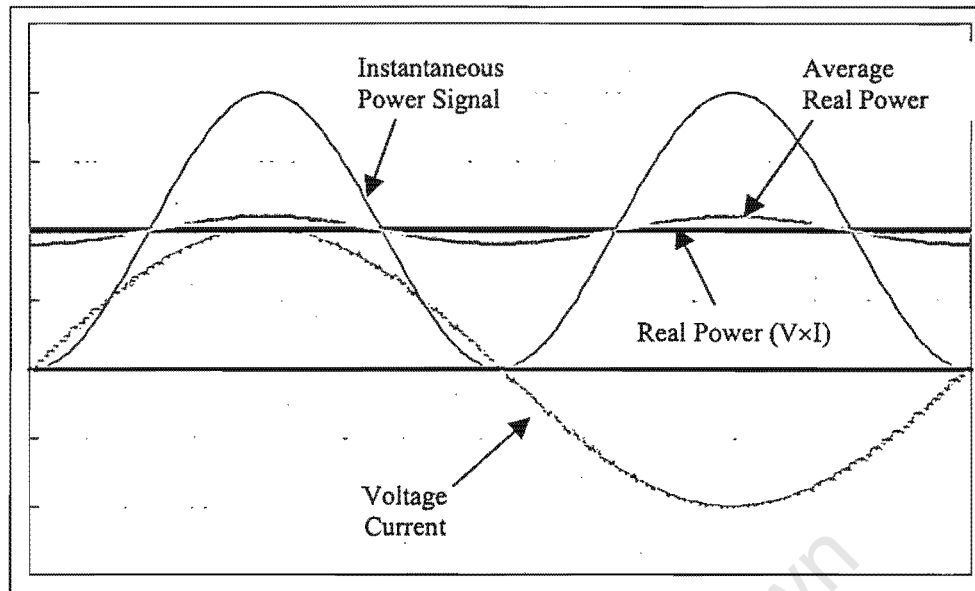


Figure 2.3(a) – Real Power information from Instantaneous Power for  $PF = 1$

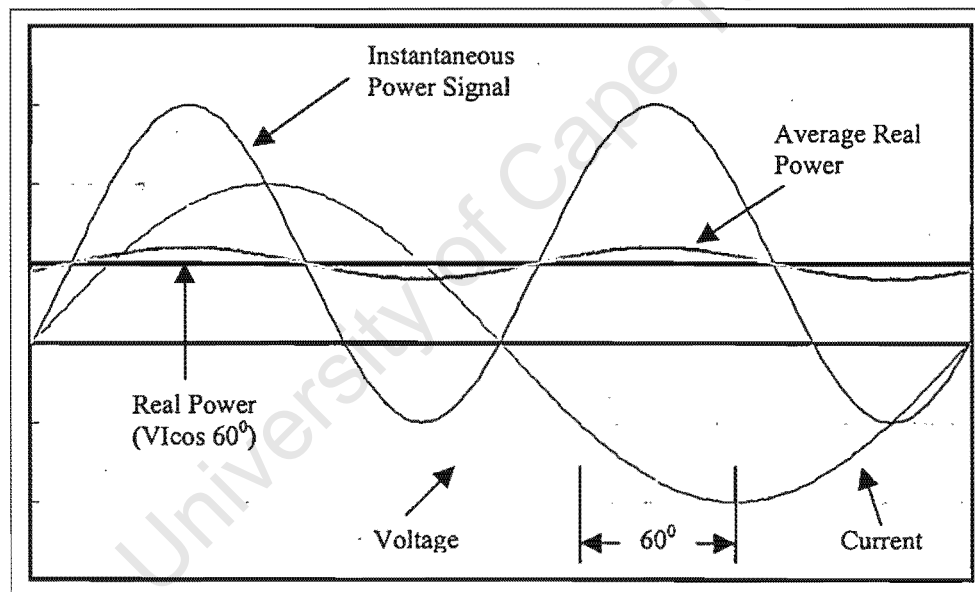


Figure 2.3(b) – Real Power information from Instantaneous Power for  $PF = 0.5$

In Figure 2.3(a), the real average power given by Equation 2.8 is simply  $V \times I$ , since the power factor is zero. In Figure 2.3(b) this value is  $V \times I \times \cos 60^\circ$ . The average real power signal, given by Equation 2.13, has been derived by low-pass filtering the instantaneous power signal, and provides a close approximation to the real power for both power factors.

### 2.3.2 Validity of power calculation in the presence of harmonic distortions

This real power calculation method also holds true for nonsinusoidal current and voltage waveforms<sup>[3]</sup>. All voltage and current waveforms in practical applications will contain a harmonic content to some extent, though usually the voltage to a lesser degree. The introduction of these distortions may be attributed to several factors, though mainly through non-linear loads, such as switch-mode power supplies and inductive loads. The amount of distortion in the voltage or current waveform is quantified by means the Total Harmonic Distortion (THD).

Using the Fourier Transform theorem, the instantaneous voltage and current waveforms can be expressed in terms of their Fourier (harmonic) components as follows <sup>[3]</sup>:

$$v(t) = V_0 + v(t)_1 + \sum_{h=2}^{\infty} v(t)_h \quad [2.16(a)]$$

where:

$v(t)$  is the instantaneous voltage

$V_0$  is the DC or average value

$v(t)_1$  is the fundamental (line frequency) voltage component

$v(t)_h$  is the voltage component of harmonic  $h$

or in terms of their RMS values :

$$v(t) = V_0 + \sqrt{2} \times \sum_{h=1}^{\infty} V_h \times \sin(h\omega t + h\alpha) \quad [2.16(b)]$$

where:

$v(t)$  is the instantaneous voltage

$V_0$  is the DC or average value

$V_h$  is the rms value of the voltage harmonic  $h$

$h\alpha$  is the phase angle of the voltage harmonic  $h$

Similarly,

$$i(t) = I_0 + i(t)_1 + \sum_{h=2}^{\infty} i(t)_h \quad [2.17(a)]$$

where:

$i(t)$  is the instantaneous voltage

$I_0$  is the DC or average value

$i(t)_1$  is the fundamental (line frequency) current component

$i(t)_h$  is the current component of harmonic  $h$

and also,

$$i(t) = I_0 + \sqrt{2} \times \sum_{h=1}^{\infty} I_h \times \sin(h\omega t + h\beta) \quad [2.17(b)]$$

where:

$i(t)$  is the instantaneous current

$I_0$  is the DC or average value

$I_h$  is the rms value of the current harmonic  $h$

$h\beta$  is the phase angle of the current harmonic  $h$

Assuming that no DC component exists in either the current or voltage waveforms and using Equations 2.16(b) and 2.17(b), the real power  $P$  can also be expressed in terms of its fundamental real power ( $P_1$ ) and harmonic real power ( $P_H$ ).

$$P = P_1 + P_H$$

where:

$$P_1 = V_1 \times I_1 \cos \phi_1 \quad \text{and} \quad [2.18]$$

$$\phi_1 = \alpha_1 - \beta_1 \quad (\text{i.e. the phase angle for the fundamental real power})$$

and

$$P_H = \sum_{h=1}^{\infty} V_h \times I_h \cos(h\phi) \quad \text{and} \quad [2.19]$$

$$\phi_h = \alpha_h - \beta_h \quad (\text{i.e. the phase angle of the harmonic real power})$$

As can be seen from Equation 2.19, a harmonic real power component is generated for every harmonic, provided that particular harmonic is present in both the voltage and current waveforms. The power factor calculation has been shown to be accurate in the case of a pure sinusoid by Equation 2.15, and therefore the harmonic real power must also correctly account for power factor since by the Fourier Transform Theorem it is made up of a series of pure sinusoids[3].

For the same reason, the power calculation is also valid for current and voltage waveforms with harmonic components. Moreover, if the voltage waveform is assumed to be purely sinusoidal, the current components at harmonic frequencies do not contribute to the average real power, as stated in Section 2.1.

The number of harmonics accounted for by sampling is dependent on the voltage and current channel bandwidths. These in turn are dependent on the respective sampling frequencies, and are further discussed in the relevant OCM technique sections.

## 2.4 Definition of System Error

The respective errors described in this report are calculated as follows:

$$\text{Error}[\%] = \frac{\text{True Value} - \text{Measured Value}}{\text{True Value}} \times 100 [\%]$$

,where the True Value denotes that value measured by the reference testing equipment, or calculated mathematically if possible.

### Section 3 - Microcontroller Considerations

The purpose of this chapter is to highlight several key issues regarding the chosen MCU for OCM implementation. The present CASHPOWER utility meters currently operate using a Toshiba MCU as the core component for all metering operations other than power measurement. These, however, do not contain an onboard A/D converter, which is the primary feature required for OCM. Consequently, EML have chosen to use the H8/3827 Hitachi MCU for the purpose of implementing OCM, should it prove viable. Based on research to date, this device contains all the necessary features required for OCM.

Since the Technical Design Team from EML is based in Isando, Gauteng, support from them and access to their particular technical equipment, including any software, is severely limited. For this reason, it was decided that the OCM algorithm and hardware prototype be developed on the Motorola 68HC908GP32 Microcontroller Unit (MCU) using the Motorola M68ICS08GP In-Circuit Simulator Kit, which has kindly been made available by Motorola. The developed firmware was designed to work independently from the existing metering software, as a standalone module. The remaining MCU functionality; the LCD display driver, credit token encryption, et-cetera, are to be designed and implemented by EML.

The actual MCU to be used for the final OCM product, should it prove viable, is the Hitachi H8/3827, as stipulated by EML. The architectures of this device and of the Motorola 68HC908GP20 are similar, facilitating the integration of the developed firmware with the Hitachi MCU.

actual data recorded using the Motorola In-Circuit Simulator. The input was varied between the ADC reference rails of 0.001V to 4.933V with approximately 5 % increments to obtain a total of 21 samples.

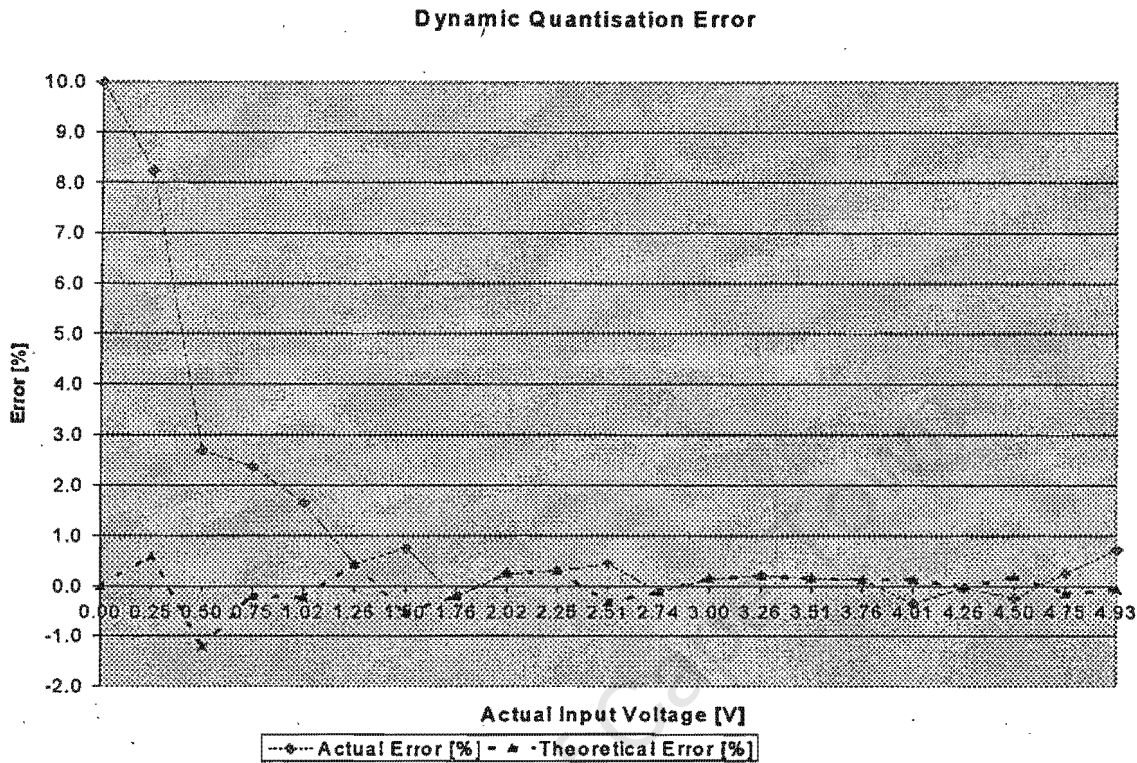


Figure 3.1 – A graphical representation of the actual 8-bit ADC dynamic error

From this data it is clear that a higher error is introduced when registering the lower range of input voltages. Theoretically, an 8bit ADC should be capable of registering an input voltage of 0.39 % of the maximum reference rail, using Equation 3.2. For a reference voltage of 4.933V this corresponds to the Motorola ADC detecting a minimum voltage of 0.019V. However, as shown empirically, the actual registered value for voltages as low as this will in all probability contain a high error.

Conversion times of the two MCU ADCs are both software adjustable. The Motorola ADC conversion time is determined as follows:

$$\text{Conversion time} = \frac{16 \text{ to } 17 \text{ ADC cycles}}{\text{ADC frequency}}$$

Motorola stipulates that for optimum accuracy the clock speed of the ADC should be around 1MHz, though higher values may be used. Using this value, the minimum conversion time for this device would be around 16 $\mu$ s.

## Section 4: Investigated Techniques

The various power and energy measurement techniques investigated are presented in this section.

### 4.1 Digital Wattmeter Using a Sampling Method.

The data presented in Section 4.1 is a review on a paper by R. S. Turgel entitled “Digital Wattmeter Using a Sampling Method” [4].

#### 4.1.1 Overview

The system described by Turgel measures average (real) power by sampling voltage and current at predetermined intervals and integrating them numerically. This method differs from those available prior to the release of the paper, which included electrodynamic, electrostatic, thermal, and Hall effect wattmeters, all of which make use of analog multiplication of current and voltage performed by a physical process.

The response of the system agrees with that of a standard wattmeter within 0.02% from DC through to 1kHz, with the exception of zero power factor measurements.

This method is in accordance with the theoretical analysis in Section 2.2 of this report.

#### 4.1.2 Methodology

The fundamental equation for calculating the average electric power is

$$P = \frac{1}{T} \int_0^T vi \cdot dt \quad [4.1.1]$$

For sinusoidal quantities, T, is chosen to be a multiple of the period of  $v$  or  $I$  [4],

$$T = \frac{n}{f} \quad [4.1.2]$$

where  $n$  is a positive integer and  $f$  is the input frequency.

The method proposed by Turgel is directly related to this formula. Instantaneous values of voltage and current are measured simultaneously, and from sets of these

measurements the average power is computed using digital methods. Because the number of measurements is necessarily finite, the integral of Equation 4.1.1 is replaced by a summation and the time interval  $T$  by the number of measured points,

$$P \approx \frac{1}{N} \sum_{j=1}^N v_j i_j \quad [4.1.3]$$

In Equation 4.1.2 the time over which the integration extends is related to the period of the voltage or current waveforms. Similarly, the time taken for  $N$  samples must be such that the interval between samples is an integral fraction of the period, and all such intervals must be equal to one another [4]. It has been shown [5] that under these conditions that with at least 3 samples per cycle, the summation will theoretically be exact. In practice, more samples are needed to reduce the effects of random noise and fluctuations in sample timing. Moreover, samples must be taken at definite intervals and not at random. Random sampling does not produce rapid convergence in the numerical integration process and is therefore not suited for high accuracy measurement.

In order to determine how many samples would suffice for a power measurement of good accuracy (below 0.5%), Turgel utilised a computer simulation of the measurement process. The study took into account the effects of input noise, timing uncertainty, presence of harmonics, and errors due to finite resolution. The known value of the integral given by Equations 4.1.1 and 4.1.2 was compared with the computed summation of Equation 4.1.3. The acceptable tolerance was set at 100 ppm and the presence of random noise with a peak value of 100 ppm was postulated. From this study, Turgel drew the following conclusions:

- Not more than 512 sample points in total are required
- A timing uncertainty of 50ns is tolerable (for input frequencies up to 1kHz)
- The percentage error due to systematic timing errors is proportional to the percentage deviation of the timing base period from the period of the input waveform
- A resolution corresponding to 15 bits for the measured values of current and voltage is adequate for the stipulated output error (0.02% at full scale input)
- The sampling rate must be at least 40kHz to resolve frequency components up to 10kHz.

### 4.1.3 System Hardware

A functional block overview of the system developed by Turgel is shown below.

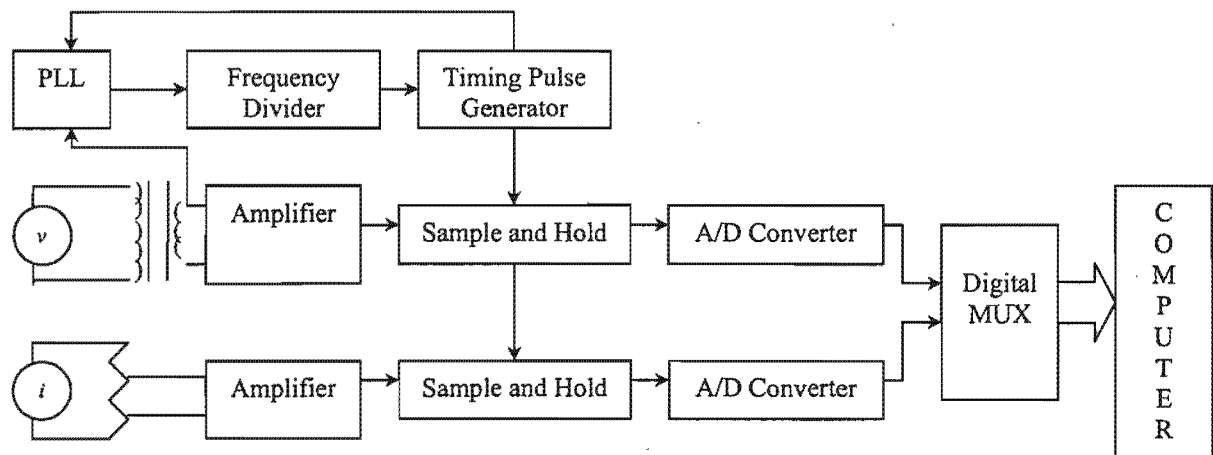


Figure 4.1: Functional Block diagram of the digital wattmeter

#### 4.1.3.1 Signal Conditioning

The current signal is obtained from a specially constructed bifilar shunt with a time constant of 2ns. At 5A, the voltage drop across the shunt is 0.5V, thus keeping the dissipation to 2.5W. The voltage signal is obtained from a two-stage step-down transformer with a 200-1 ratio which has negligible ratio and phase corrections. Two similar transformers are used, one for the frequency range of 50 to 400 Hz, the other for the range 400 to 2000Hz. The secondary voltage is 0.5V.

#### 4.1.3.2 Preamplifiers

These two input signals are fed into high-impedance instrumentation amplifiers which permit the shunt output and transformer secondary to work essentially on open circuit. The amplifier gain is fixed at a nominal value of 13.5 and is flat within 50 ppm from dc to 5kHz. This gain raises the 0.5V inputs to the level required by the sample and hold circuitry,  $\pm 10\text{V}$  peak. An important consideration is to ensure that both preamplifiers introduce the same phase shift respectively. At 10kHz the difference in phase shift between the two was less than  $600\mu\text{rad}$ . Moreover, the effects of small remaining differences in phase shift can be cancelled by averaging measurements with the two channels interchanged.

#### 4.1.3.3 Timing Pulse Generator

A critical part of the measurement process is the timing of the sampling. This is brought into a definite relationship with the input signal by means of a phase-lock circuit, as shown in Figure 4.1. Part of the input voltage signal is fed to a high-impedance instrumentation amplifier with good common mode rejection which acts as a buffer to isolate the analog measuring circuit from the digital circuitry. The amplifier gain is chosen so that the phase lock will work reliably over an input voltage range from 0.1 to 1.0 V.

The frequency of the voltage signal appearing at the output of the amplifier is then compared to a subharmonic of the pulse repetition frequency in a phase sensitive detector which in turn controls the frequency of the sample timing pulse oscillator. The subharmonic is obtained by means of a binary frequency divider which has 9 manually selectable ratios. These range from a ratio of 4 to 512. Therefore, since the timing pulse oscillator operates over a band from 20kHz to 40kHz the subharmonic generated by the highest ratio setting (512) ranges from 40 to 80Hz. It follows that, for instance, for an input of 60Hz there are 512 sampling points per cycle, while for an input of 6kHz there are only 4 samples per cycle. The ADC's however, always operate at the maximum possible rate consistent with the phase-lock requirements from 20kHz to 40kHz.

There is a tradeoff between capture range and stability in phase-lock circuits. Since in this application stability is of overriding importance, the loop filter circuits have been selected to provide high stability. As a consequence the capture range is fairly narrow, however for nominal frequencies between 40Hz and 100Hz the oscillator requires no external trimming to bring it into a lock.

#### 4.1.3.4 Analog-to-Digital ADCs

The outputs of the preamplifiers are fed to two parallel S/H circuits and captured samples are held long enough for the A/D conversion. The two resulting digital numbers are then multiplexed to a common 15-bit parallel output. This configuration is preferable to the one where an analog multiplexor feeds signals to a single channel.

It avoids the additional errors introduced by analog multiplex switching circuits through noise and offset voltages.

The 15-bit output of the ADCs exists as a two's complement binary number, therefore containing 14-bits of resolution plus 1 sign bit. The nominal conversion time is around 15 $\mu$ s, however due to various settling times and the time required to transfer the output data, a minimum of 25 $\mu$ s is required between conversion cycles. The maximum throughput rate is therefore 40kHz.

#### **4.1.3.5 Computing Circuitry**

Since the ADC data occupy only 15 bits, a 16-bit processor is sufficient to handle the computations. Because the input data is multiplexed, the 16<sup>th</sup> bit of the computer-generated word is used as a channel indicator. The processor was linked to a teleprinter to provide readable outputs.

#### **4.1.4 Software**

All routines were written by Turgel in assembly language using 16-bit integer arithmetic. With the technology available at the time of release (1974) the time taken for a typical power measurement was around 1 second. Specifics regarding the various algorithms were not published.

Certain software-based diagnostic checks are performed with each power measurement cycle. These include a test for the ADC channel number, the sequence in which data is stored in memory, and finally a check for clipping of the sampled signals indicating an input voltage outside the nominal range.

#### **4.1.5 Measurement Procedure**

Wherever possible, measurement procedures relied on external standards in order to be independent of such circuit parameters as drift, offset, gain and internal reference source stability. For each update of the measured power the following process takes place:

- Each sequence starts with a "zero reading" where the inputs are connected via a small resistor to analog ground. This reading is used as an offset correction reference to subsequent data.

- Next is the “calibration voltage” obtained from a Zener reference source stable to within a few parts per million per month. These measurements are the average of 512 readings on each channel and both positive and negative calibration voltages are used to compensate for a slight asymmetry in the response of the S/H and ADC combination. This data is used as a gain correction reference for subsequent data.
- 512 Samples of voltage and current are then taken and stored in memory as previously described.
- The appropriate pair of calibration constants from the first two steps is used depending on the sign of the measured value and the measurement channel.
- The power calculation is computed using Equation 4.1.3 as previously described.
- The result is displayed on the system output (a Teleprinter) and the sequence begins again.

It is assumed that in the brief interval between calibration and measurement the circuit parameters change only by negligible amounts, and that the computed value will be independent of offsets and the gains of the amplifiers and ADCs [4].

#### 4.1.6 Results

To investigate the performance of the system, Turgel made measurements in comparison with an electrodynamic wattmeter. Table 4.1 below displays the results obtained for two values of input frequency at nominal inputs 100V and 5A. A dynamic range analysis was not provided.

Frequency [Hz]	Power Factor	Difference from Comparison Standard (% of full scale)	Comparison Standard	Uncertainty of Standard (% of full scale)
50	1	0.014	Wattmeter	0.01
		-0.004	rms <sub>a</sub>	0.005
	0.5 lead	Wattmeter	0.02	
	0.5 lag	Wattmeter	0.02	
60	1	0.008	Wattmeter	0.02
		-0.018	rms <sub>a</sub>	0.005
	0.5 lead	Wattmeter	0.02	
	0.5 lag	Wattmeter	0.02	

**Table 4.1:** Results for the system taken at full-scale input

<sup>a</sup> rms value for each channel was computed by simulated means[4]

It was found that certain test instruments, such as oscilloscopes, when attached at critical points modified the system response because of loading, ground loops, or pickup<sup>[4]</sup>. Furthermore, under some conditions the sensitivity of the null meter used to compare the inputs of the phase-lock circuit was insufficient for optimum adjustment of the oscillator.

University of Cape Town

## 4.2 Digital Sampling Laboratory Wattmeter

The data in Section 4.2 is a review of all relevant data presented in a paper by A.C. Corney and R.T. Pullman entitled “Digital Sampling Laboratory Wattmeter” [6].

### 4.2.1 Overview:

The digital sampling wattmeter described by Corney *et al* in this section is based on the concept pioneered by R.S. Turgel discussed in Section 4.1.

The laboratory multifunction wattmeter described below uses two 15-bit ADCs and achieves an overall power measurement accuracy of 0.015% of applied volt-amperes at mains frequencies over the range 100mA to 100A and 5-100V.

Power is measured by multiplying digitized samples of the current and voltage waveforms and, if the correct sampling strategy is used, relatively few samples per waveform cycle are required [6]. An inherent advantage of this system is that the digital multiplication gives complete independence between the current and voltage channels, allowing each to be calibrated independently.

This method is in accordance the theoretical analysis discussed in Section 2.2.

### 4.2.2 Configuration

The system block diagram presented as Figure 4.2 depicts the system operation. All processing was performed using an 8085-based single-board computer. Two respective signal conditioning boards were used to interface the required channels with respective external 15-bit hybrid ADCs. These boards comprise a selectable gain voltage amplifier and current-to-voltage converter for the voltage and current signals respectively. The gains are automatically switched by the processor so that the output signal levels are within the input range of the ADCs.

The two ADCs are simultaneously triggered at a sampling interval previously determined by the processor. The samples from each converter are multiplied together and the products are summed and then averaged to provide real power information.

During data acquisition and subsequent processing, checks are made to ensure that the signals fall within defined level and frequency limits. Checks are also made on frequency stability before the results are displayed.

The timing information from which the sampling period is determined is derived from the zero crossings of the voltage signal. If this signal is not detected, the processor checks for the presence of current and is present, derives its timing from this signals until the voltage signal is reestablished.

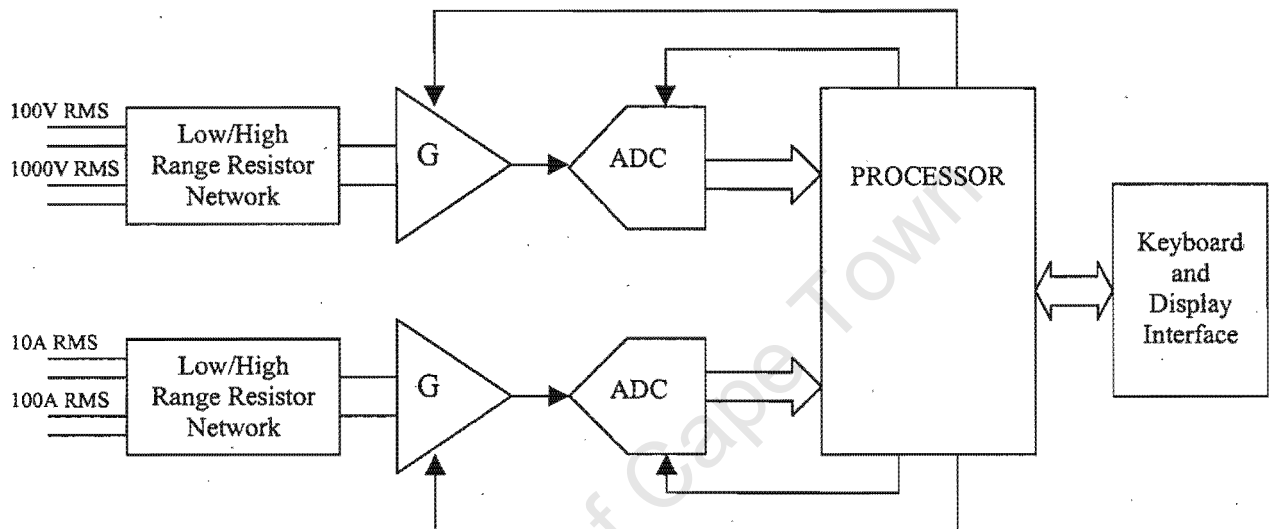


Figure 4.2: Basic configuration of digital sampling wattmeter

### 4.2.3 Digital Sampling Technique

The average power measured over an interval  $T$  is given by:

$$P = \frac{1}{T} \int_0^T V(t)I(t)dt \quad [4.2.1]$$

and this may be approximated by the summation:

$$P = \frac{1}{n} \sum_{i=1}^n V_i I_i \quad [4.2.2]$$

,where  $V_i$  and  $I_i$  are simultaneous samples of voltage and current equally spaced in time. The samples may be spread over  $m$  periods ( $m$  an integer) allowing the sampling interval to be increased accordingly. The conditions required for this are:

- 1) the sampling must occur over an integral number of waveforms, and
- 2) the waveform is stationary for the duration of measurement.

With these two conditions, the result is accurate for all power harmonics except those harmonics  $k$ , for which  $km/n$  is an integer [6]. The harmonic power terms arise from harmonics  $V_p$  and  $I_q$  in the voltage and current waveforms combining to produce power components at the  $(p+q)^{\text{th}}$  and  $(p-q)^{\text{th}}$  harmonics. Therefore if  $m$  and  $n$  have no factors in common, errors are limited to those voltage-current products where  $p+q$  or  $p-q$  is an integral multiple of  $n$ . Thus if the number of samples taken per period,  $n$ , is considerably larger than the number of periods sampled over,  $m$ , then only the higher order harmonics  $k$ , which may be considered negligible, contribute to the system error.

For low values of  $n$ , however, errors may not be negligible, particularly if both waveforms are non-sinusoidal or if a dc component is present on one waveform while the other contains significant distortion.

In addition to the power measurement, a watt-hour measurement capability is also provided. Whenever data is processed the clock is read and the time since the last measurement update is computed. The difference is multiplied by the latest power measurement and the result added to the accumulated watt-hour measurement. This approach assumes that the power remains essentially constant between updates which may be up to 2 seconds apart, depending in the signal frequency.

#### **4.2.4 Analogue Signal Conditioning**

The analogue conditioning circuitry developed by Corney *et al* is shown in simplified form in Figure 4.3. High and low ranges are provided for the voltage and current inputs, selectable by choice of the appropriate input terminals. The voltage ranges are 0-100V and 0-1000V, and the current ranges are 0-10A and 0-100A. Once the high or low range inputs have been selected, the microprocessor automatically selects the optimum gain setting for the voltage and current amplifiers. Nine gain settings are provided for the voltage circuit and 12 for the current circuit. The precision gain resistors are selected by means of solid-state switches in such a way that switch imperfections have no influence. Because calibration is achieved by means of software, trimmers are not required and the instrument can be kept reading correctly without losing track of the long-term calibration drift.

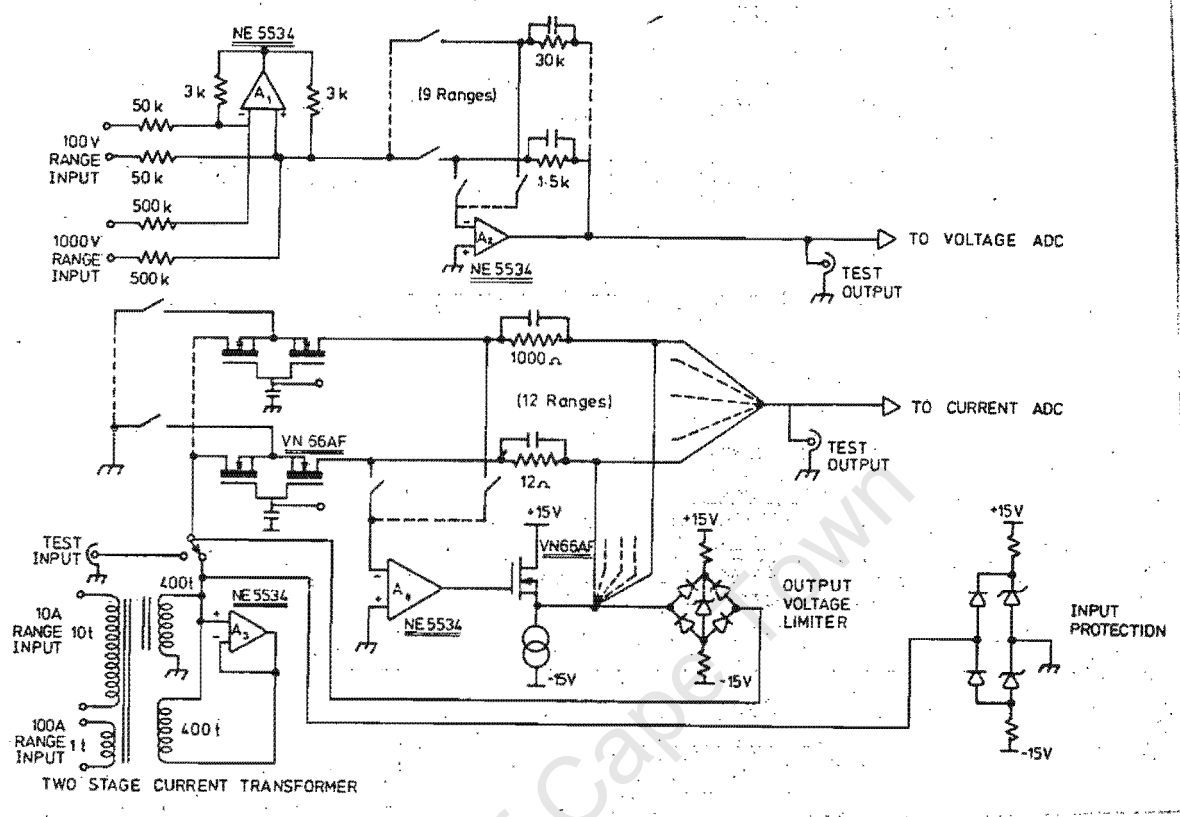


Figure 4.3: Analog signal conditioning circuitry for Laboratory Wattmeter

Comey, A. Pullman, R. (1987) "Digital Sampling Laboratory Wattmeter," IEEE Transactions on Instrumentation and Measurement, vol. IM-36, no. 1. Page 55.

Amplifier  $A_1$  acts as a current pump which performs the dual role of injecting the correct current into the input switch of  $A_2$  irrespective of the switch resistance, and of providing a differential input. An amplified two-stage current transformer provides accurate current scaling factors of 40:1 and 400:1. The transformer secondary current is passed into a current-to-voltage converter comprising amplifier  $A_4$ , transistor power booster and the switch selectable gain resistors. Some of the circuitry used to protect the instrument from overloads and transients are shown.

Phase matching between the current and voltage amplifiers is provided by a trimmer capacitor across each gain setting resistor. To minimise wideband noise, the analogue signals are fed to the ADC's via matched, passive, singly terminated, 3pole, low-pass Butterworth filters having a cut-off frequency of 20kHz.

The number of cycles  $m$  over which the samples are taken was set at 7 to allow sufficient time between samples at the upper frequency limit of 1kHz. The technique chosen for the generation of the sampling pulses was to measure the period between  $m$  successive positive-going zero crossings and calculate the interval  $m/n$ . This value is then loaded into a prestable down counter which generates a sampling pulse at each underflow.

The necessity for accurate sample timing has been pointed out [7] and the effect of imprecise sample timing due to finite sampling interval resolution, termed the “truncation error”, has been analysed by Stenbakken [8]. To quantify the truncation error for the chosen configuration, assume a nominal sampling interval of  $Tm/n$ . Now introduce a one-half clock-cycle timing error  $\varepsilon$  such that the true sampling interval is between  $Tm/n$  and  $Tm/n(1 \pm \varepsilon)$ . Following Stynbakken's analysis and assuming that sampling starts one sampling interval after a voltage zero crossing, an expression for the maximum truncation error is obtained as a fraction of the volt-amperes and is given by

$$e_T = \frac{\sin(2\pi.m.\varepsilon)}{n.\sin\left(\frac{2\pi.m}{n}\right)} \cdot \cos\left(\phi + 2\pi.m\left(\frac{1}{n} - \varepsilon\right)\right) \quad [4.2.3]$$

where  $\phi$  is the phase angle between the voltage and current waveforms. For this case with a clock frequency of 9.216 MHz, the timing error  $\varepsilon$  is  $\pm 1.1 \times 10^{-5}$  for a waveform period of 20ms.

The maximum truncation error for  $m = 7$  and  $n = 29$  is, therefore,  $\pm 0.0015\%$  and occurs at a phase angle of  $93^\circ$ . The truncation error at unity power factor is  $\pm 0.001\%$ .

The sampling pulses are generated as shown in Figure 4.4.

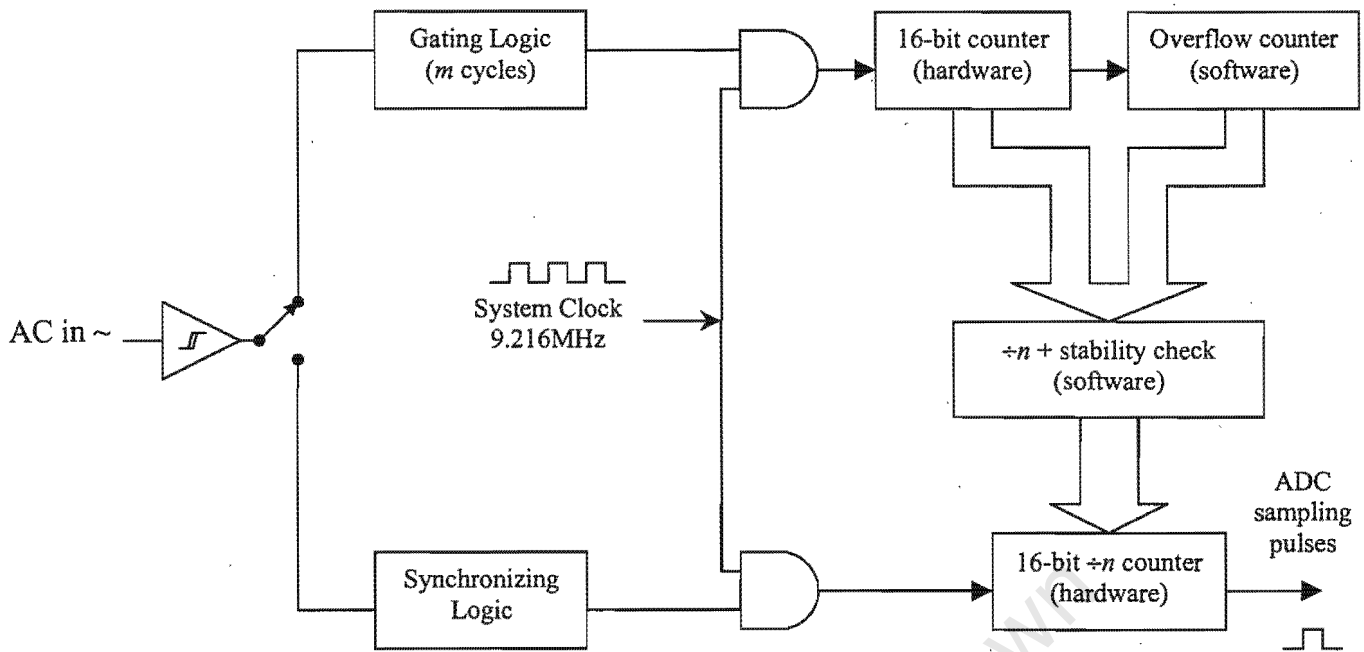


Figure 4.4: Sampling pulse generator

Clock pulses at 9.216MHz are gated to a 16-bit hardware counter for exactly  $m=7$  periods of the ac-coupled signal. The counter is permitted to overflow during this period and number of times overflow occurs is recorded by the software. The resultant count is an integer given by  $N_{\text{overflow}} \times 2^{16} + \text{residual count}$ . This is divided by  $n=29$  to give the sampling interval. If the sampling interval exceeds the allocated 16 bits the signal frequency is deemed too low and no sampling takes place. This corresponds to a lower frequency of 34 Hz from the stipulated values of  $n$  and  $m$ . Otherwise the sampling interval is tested for stability by repeating the measurement. If two consecutive results agree, a 16-bit down counter is loaded with the sampling interval immediately after the next positive-going zero crossing of the voltage signal. This counter is clocked at 9.216MHz from the same clock used to measure the period.

Using this strategy ensures that the sampling interval is insensitive to drifts in the clock frequency and that the first sample is always close to one sampling interval after a positive-going voltage zero crossing. It also ensures that the same number of samples per cycle  $n$  is taken regardless of fluctuations in line frequency.

### 4.2.7 Results

The error budget measured by Corney *et al* is displayed as Table 4.2 where it can be seen that the overall uncertainty at unity power factor is 0.0138% up to 400Hz.

	<b>Error (parts in <math>10^4</math> of measured power) PF = 1 and f = 40 – 400Hz</b>
AC/DC signal transfer	1.0
Linearity and adjustment of voltage ranges	0.5
Linearity and adjustment of current ranges	0.5
Noise and quantisation effects, Taken for a single reading	0.6
Sample timing error	0.2
Phase match between channels	N/A
<b>RMS Total uncertainty</b>	<b>1.38 (= 0.0138%)</b>

**Table 4.2:** Factors contributing to overall uncertainty at the 99% confidence interval

### 4.3 Digital Power Metering Manifold

The data presented in Section 4.3 is a brief review on a subsection of a paper by C.P. Young and M.J. Devany entitled "Digital Power Metering Manifold"<sup>[9]</sup>. In this paper, Young *et al* present a distributed approach to power metering on multiple branch circuits supplied from a common distribution panel. Although the bulk of the paper concerns a master-slave design in the distribution of data in a multiple power measurement system, a brief reference to information relevant to the current study is included.

#### 4.3.1 RMS-Value Calculating Algorithm

Young *et al* present a method of measuring the RMS value of a sampled signal that inherently compensates for any offsets in the signal. To summarise,  $N$  sampled data  $x_0, x_1, x_2, \dots, x_{N-1}$  are acquired from a signal  $X$  containing a non-zero offset, the RMS value of the signal  $X$  is given by

$$\begin{aligned} X_{rms} &= G_x \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} x_k^2 - \left( \frac{1}{N} \sum_{k=0}^{N-1} x_k \right)^2} \\ &= G_x \sqrt{x^2 - \bar{x}^2} \\ &= G_x \sigma_{xk} \end{aligned} \quad [4.3.1]$$

where  $\sigma_x$  is standard deviation and  $G_x$  is the scale factor to convert the computation result of a set of digital samples to the signal's rms value. The factors  $G_V$  and  $G_I$  for calculating rms voltage and current, respectively, are

$$\begin{aligned} G_V &= \frac{V_{rms,max} \sqrt{2}}{2^{B-1}} \\ G_I &= \frac{I_{rms,max} \sqrt{2}}{2^{B-1}} \end{aligned} \quad [4.3.2]$$

where  $V_{rms,max}$  and  $I_{rms,max}$  are maximum rms ratings associated with the voltage or current measurement system and  $B$  is the number of bits of the A/D converter. The two terms in Equation 4.3.2 above effectively scale the sampled data from integer samples to real instantaneous voltage and current values.

### 4.3.2 Real Power Calculation

Young *et al* also present a real power calculation that compensates for signal offsets. If  $N$  samples of voltage signals  $v_0, v_1, v_2, \dots, v_{N-1}$  and current signals  $i_1, i_2, i_3, \dots, i_{N-1}$  are acquired at the same sampling instants, then the real power  $P$  is calculated using Equation 4.3.2 and Equation 2.15 as follows:

$$\begin{aligned}
 P &= G_v G_i \left( \frac{1}{N} \sum_{k=0}^{N-1} v_k i_k - \left( \frac{1}{N} \sum_{k=0}^{N-1} v_k \right) \left( \frac{1}{N} \sum_{k=0}^{N-1} i_k \right) \right) \\
 &= G_v G_i (\overline{vi} - \overline{v}\overline{i})
 \end{aligned}
 \tag{4.3.3}$$

The terms in parenthesis represent the mean of the products and the products of the means of the voltage and current samples respectively [9].

The power measurement method described by Equation 4.3.3 is advantageous in that it compensates for scaling factors when converting from real data to digital integer values, and reduces errors introduced through offset components present in the input signals.

The inclusion of the latter term,  $\overline{v}\overline{i}$ , ensures that any offsets in the inputs signals do not contribute to the measured power. Young *et al* used this as an alternative to implementing a HPF in one or both of the input channels. An inherent advantage of this method is that the voltage and current samples, as well as the instantaneous power, may be summed itinerant, that is, during the sampling sequence. Furthermore, the implementation of a HPF in one channel would introduce an associated phase shift introducing the need for phase-shift compensation in the other. This is not the case here.

## 4.4 Design of a Microprocessor-Based digital Wattmeter

The data in Section 4.4 is a review on a paper by J.J. Hill and W.E. Alderson entitled “Design of a Microprocessor-Based Digital Wattmeter” [10].

### 4.4.1 Overview

Hill *et al* present a microprocessor-based implementation of a sampling wattmeter. The measurement process is in two parts. First, the microprocessor measures the voltage waveform period and computes the sampling interval and number of required samples. The measurement of average power then takes place over an integer number of cycles. Digitized samples of the voltage and current waveforms are multiplied together and the products accumulated over the measurement period. The sum is divided by the number of samples, scaled, and read out as a 4-digit display of average power. A full-scale accuracy of better than 0.5% was attained.

The power measurement method is based on that first described by Turgel [4]. The system developed by Hill *et al* uses a minimum of 30 samples from which to compute average power. For an input frequency of 50Hz a total of 65 samples are used, as shown in Table 4.3 in Section 4.4.4.

The system described by Hill *et al* attained a full-scale accuracy of better than 0.5%.

### 4.4.2 System Hardware

A system block diagram appears as Figure 4.5. This was based on a Motorola M68000 8-bit processor to which an  $8 \times 8$  bit hardware multiplier and analog input circuits were added. The hardware multiplier was necessary to compute the products of  $v_i$  in real time.

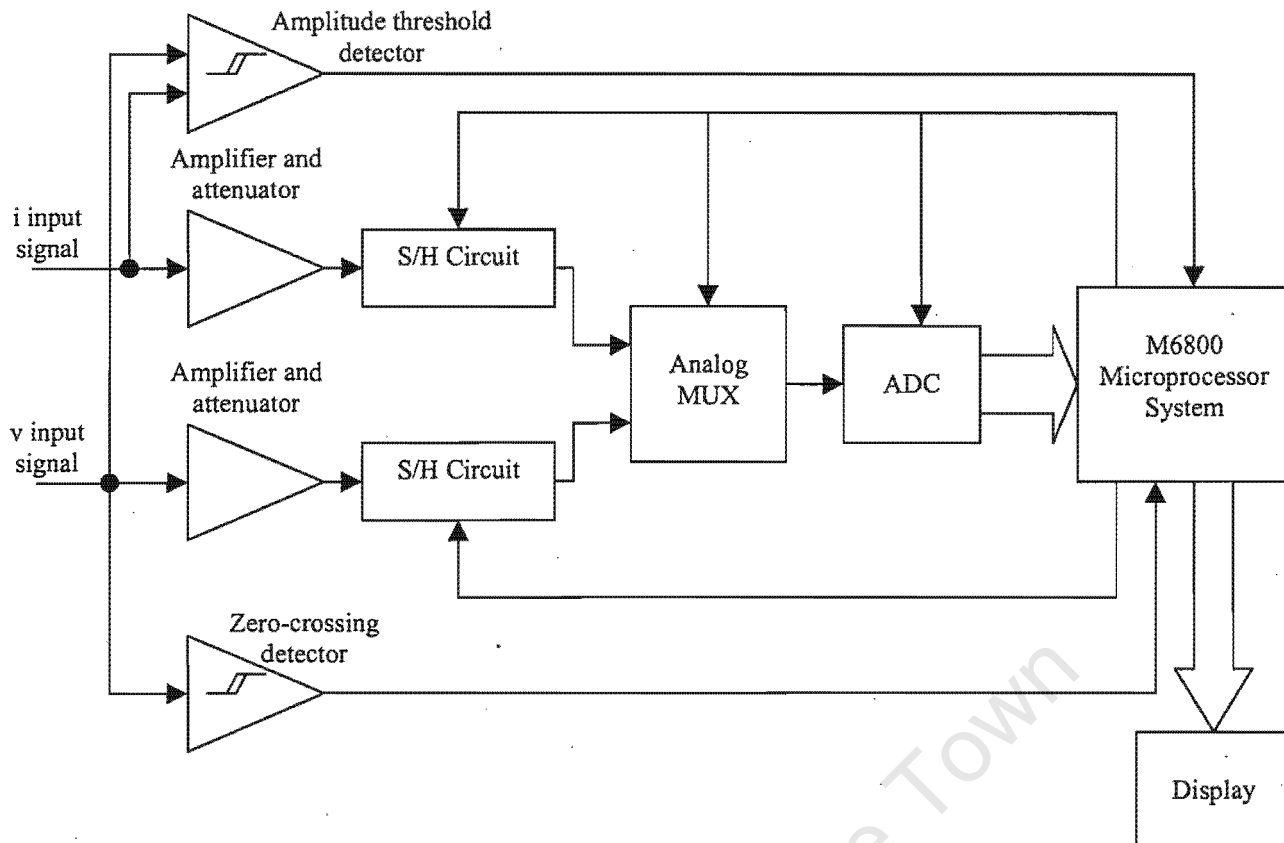


Figure 4.5: MCU-based system block diagram

Voltage signals representing load current and line voltage are supplied to the input circuitry. Two S/H circuits are used to enable concurrent capturing of the respective inputs. These samples are presented to a single 12-bit (11 bits + sign bit) ADC in a sequential manner using an analog multiplexer. A zero-crossing detection circuit inputs the voltage signal and provides a digital edge to the Microprocessor with each negative-going zero-crossing. An amplitude sensing circuit is used to disable the measurement process if either of the inputs exceeds the allowable input range.

To maintain synchronism, all peripherals are controlled via the Microprocessor under software control.

A number of potentiometers are provided in the analog input circuitry for front-end calibration. These consist of independent offset adjustments for the S/H circuits, and a gain and offset adjustment for the ADC.

### 4.4.3 System Operation

The wattmeter measurement process developed by Hill *et al* is depicted by Figure 4.6 below.

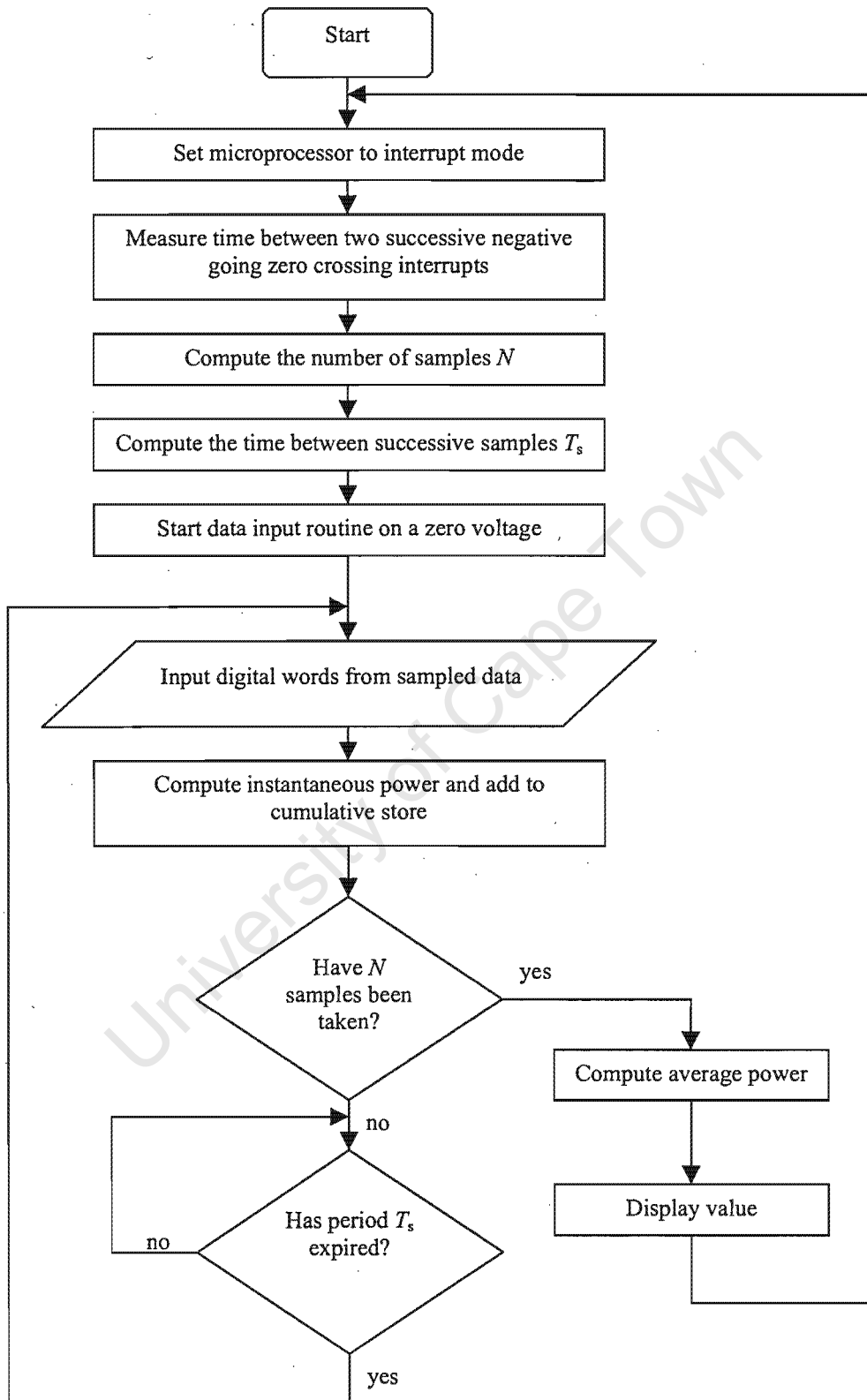


Figure 4.6: Flow diagram for the wattmeter measurement process

The sampling frequency must be selected such that  $N$  sampling pairs are equally spaced over the measurement interval [10]. This is achieved by ensuring the sampling frequency is locked to the input signal frequency. The zero-crossing detector is used to interrupt the microprocessor while executing a count program. The time between two interrupts is evaluated and this represents the period of the voltage waveform. The number of samples  $N$  and the measurement interval  $T$  are then computed.

Sample pairs are then read in from the ADC, multiplied together using the hardware multiplier, and the result added to the cumulative store before a further pair of samples is taken. After  $N$  sample pairs have been taken, the average value is computed, scaled, and displayed as a digital readout of average power.

#### 4.4.4 Power Measurement Algorithm

The measurement of average power is based upon a multiple interrupt driven system. An interrupt vector originating from the negative-going zero-crossing condition initiates the start of the measurement cycle. Initially the periodic time of the input signal must be obtained using a software-based timer. Provided that the input signal frequency is within the range of the instrument, the next interrupt from the same source enables the periodic time to be computed. The number of samples  $N$  required can then be obtained from the relationship

$$N = \text{int} \left[ \frac{N_c T_c}{T_d} \right] \text{ for } N \text{ an integer [10]} \quad [4.4.3]$$

where  $N_c$  is the count between two successive interrupts,  $T_c$  is the count cycle time, and  $T_d$  is the cycle time of the data input routine. If the value  $N$  is found to be less than 30, then the measurement interval is increased by the periodic time of the input signal and the number of samples recalculated. This process is repeated until  $N \geq 30$ .

The ratio  $\frac{2M}{N}$  is then tested and if the result is an integer, the measurement interval is increased by the periodic time of the input signal while maintaining the same number of samples  $N$ .

The time between successive samples,  $T_s$ , can then be computed by

$$T_s = \text{integer part of } [( \text{measurement interval in } \mu\text{s} ) / N ] + 1 \mu\text{s} \text{ [10]} \quad [4.4.4]$$

The digital words corresponding to each pair of samples are converted to unsigned BCD. The product of each pair is then obtained in stages using the 8-bit hardware multiplier to obtain partial products and assembling these to produce a result which is finally rounded to an accuracy of 16 bits. The product is then added to, or subtracted from, a 32-bit cumulative store depending on the sign associated with each pair of samples.

Table 4.3 below shows the number of sample pairs used to calculate average power, together with the corresponding sampling period and measurement period for three input frequencies.

Input Frequency [Hz]	Input Period [ms]	Number of samples $N$	Sampling period $T_s$ [ $\mu\text{s}$ ]	Measurement Period [ms]
10	100	328	305	100
50	20	65	308	20
100	10	32	313	10

*Table 4.3: Number of samples, sampling period and measurement period*

For this 12-bit instrument, an accuracy of  $\pm(0.2\% + 0.2\%$  of full scale) or  $\pm$ (the least significant digit) whichever is the greater is obtained for ac power measurements. At input frequencies below 110 Hz, the instrument takes three cycles of the input waveform to respond to step changes in power or frequency.

## 4.5 Fuzzy Based Adaptive Digital Power Metering Using a Genetic Algorithm

The data in Section 4.5 is a review on a paper by C.H. Kung, M.J Devaney, and C.M. Huang entitled “Fuzzy-Based Adaptive Digital Power Metering Using a Genetic Algorithm” [11].

### 4.5.1 Overview

Kung *et al* describe a fuzzy-based, adaptive approach to the metering of power and RMS voltage and current employing a genetic algorithm. The fuzzy-based adaptive metering engine adjusts the number of points per cycle to be processed and the location of these points. Adjustments are based on the optimal fuzzy rules constructed by a genetic algorithm to satisfy overall metering-error criteria under different operating environments while minimizing the number of points actually employed in the metering computation. This results in a reduction in the metering-computation effort, which frees up the processor for other metering tasks.

Generally, designing a fuzzy inferencing engine requires much trial-and-error effort in determining the fuzzy rules, and the derived rule base has to be refined when the operating condition is changed. By using a genetic algorithm to construct the optimal fuzzy rules, the trial-and-error effort can be significantly reduced.

The fuzzy-based adaptive metering algorithm was implemented on a Microcontroller-based power metering system that employs a multitasking operating system which exploits the efficiencies achieved by the reduced metering rate. The system was tested with a variety of actual and synthesized power-system waveforms and the experimental evaluations demonstrated high accuracy in the metered quantities.

The method used by Kung *et al* to measure the system power is the same as that developed by Young *et al* [9] presented in Section 4.3. The fundamental equation used

is

$$P = G_v G_I \left[ \frac{1}{N} \sum_{n=1}^N v_n i_n - (\bar{V})(\bar{I}) \right] \quad [4.5.1]$$

This equation gives the power irrespective of the direction of either signals [11].

#### 4.5.2 Genetic Algorithm Overview

A genetic algorithm is a search algorithm based on the mechanics of natural selection and natural genetics [11]. It combines survival of the fittest among string structures with a structured yet randomised information exchange to form a search algorithm with some of the innovative flair of human search. In genetic algorithms, each parameter to be optimized is represented by a string structure which is similar to the chromosome structure in natural genes.

A group of such strings is termed a population. Strings in a population, when mutated and/or crossed over, produce a new generation of strings. For each generation, all the populations are evaluated based on fitness. The individual with the higher fitness, or survival probability, will have more chances of survival. The general concept of a genetic algorithm is illustrated in the below figure.

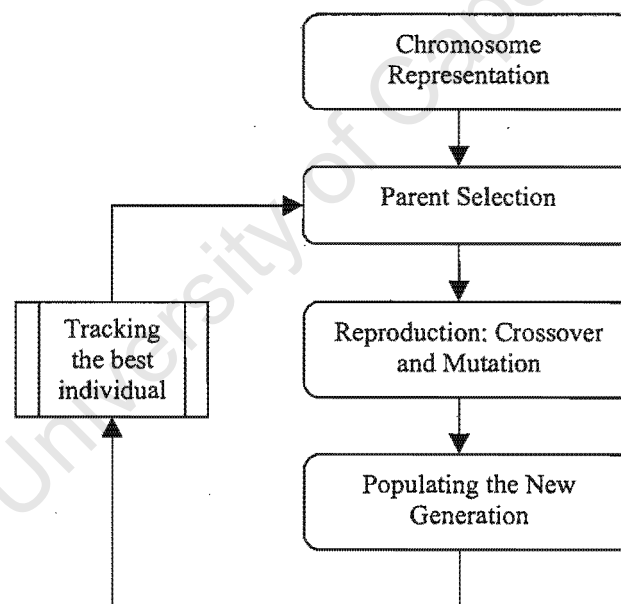


Figure 4.7: The general concept of a genetic algorithm

Each phenotype is a value decoded from a chromosome, and each chromosome can be represented by a binary string. For example, an 8-bit binary string is chosen to represent a chromosome with the phenotype restricted to values between – 40 and 40.

Therefore, the phenotype may take on one of 255 different values between 40 and 40.

In keeping with the ideas of natural selection, the individuals with a higher fitness are more likely to mate than individuals with lower fitness. The method used by Kung<sup>*t*</sup> *al* to select parents from the individuals is called the *roulette-wheel-method*, where each member of the population is given a sector whose size is proportional to the fitness of that individual. The wheel is “spun” and whichever individual comes up becomes a parent.

Once the parents have been selected, a random process of recombination in which each parent contributes part of its genetic structure to offspring. The crossover method used in this system is called the *single-point crossover*, where, based on the probability of crossover, partial exchange of characters between two strings is performed. The point of crossover is chosen randomly.

Every so often it becomes necessary to introduce a random alteration of the value of a string. This is done to ensure that a larger segment of the search vector is explored. In binary string representation, mutation occurs by simply flipping the state of a bit from 1 to 0 or vice versa.

Finally, the new generation is produced from the mating of parents. The method employed in this system mates enough parents so that enough children are produced to replace the entire population.

### 4.5.3 Fuzzy Knowledge Base Tuning Using a Genetic Algorithm

A fuzzy rule of a multiple input single output (MISO) fuzzy system can be represented as

$$\text{IF } x_1 \text{ is } A_{i1} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } B_I \quad [4.5.2]$$

, where  $1 \leq i \leq n$ . The determination of the two-dimensional set of input parameters  $A_{i1}.. A_{in}$  normally requires much trial-and-error effort. By using a genetic algorithm to set these parameters, or *fuzzy knowledge base*, the process is significantly reduced.

The exact details regarding this process is not within the scope of this investigation, however it is suffice to say that each output chromosome from the genetic algorithm may be used to represent a fuzzy rule base.

#### 4.5.4 Fuzzy-Based Adaptive Power Metering Algorithm

The general structure of metering algorithm developed by Kung *et al* is depicted in Figure 4.8. The actual power system voltage and current samples are acquired as a fixed rate of 64 samples per input cycle. The voltage and current time data are then decimated by factors of 1, 2, 4, or 8 depending upon the relative periodicity of the signal waveforms. The adaptive error criteria permits a higher decimation factor is required when the sampled data exhibits the aperiodic influences of disturbances or noise.

As shown Figure 4.8, a fuzzifier function is used to convert relative differences between the calculated even and odd groups of power measurements into fuzzy data before this information is used for the learning process.

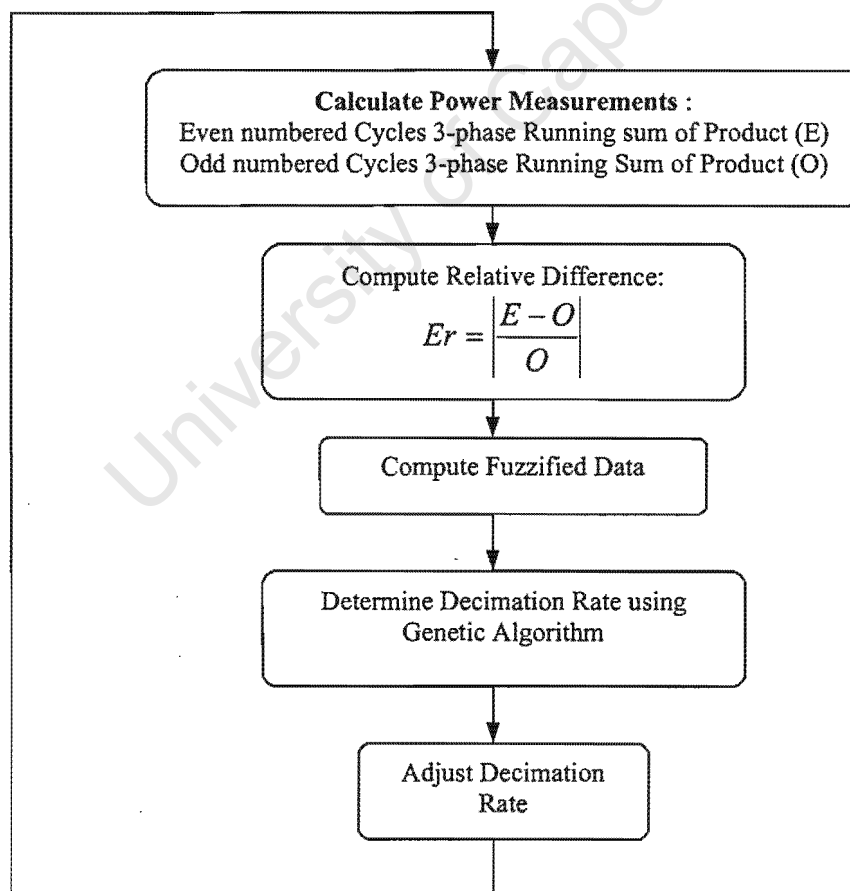


Figure 4.8: General structure of the fuzzy-based adaptive digital power metering

The rule base utilized in this application can be generalized as the following [11]

Rule 1: IF (*Er* is *Less*) THEN (*DF* is *Increased*)

Rule 2: IF (*Er* is *Greater*) THEN (*DF* is *Decreased*)

where *Er* and *DF* represent the *relative difference* and the *decimation factor*, respectively. The *less*, *greater*, *increased*, and *decreased*, are the linguistic values that represent *less than the lower bound*, *greater than the lower bound*, *increase the decimation factor*, and *decrease the decimation factor*, respectively.

The genetic algorithm is initiated every 1 minute to refine the fuzzy knowledge base and comply with the current situation of the metered power system signals.

#### 4.5.5 Results

The system was tested using an unloaded universal motor in a portable drill. The load was selected because of its relatively high aperiodic components. The results were compared against a meter with known accuracy (The Square D PowerLogic Circuit Monitor 2000).

Kung *et al* found that the developed system was about three times more stable than the circuit monitor. As the aperiod nature of the load was increased the relative difference between the voltage measurements of the two systems remained constant, whereas the difference in current and power measurements increased. This was attributed to the increase in the aperiodic components of the current signal.

Over a series of tests, the maximum relative difference between the two systems' power measurements was found to be 0.67% at a decimation factor of 8.

## 4.6 RMS value Method

### 4.6.1 Overview

According to Equation 2.8 of this report, the real power delivered to a load may be calculated using the RMS equivalents of the sinusoidal voltage and current waveforms and the corresponding power factor as follows:

$$\text{Let } v_{inst} = V_{RMS} \cos(\omega.t) \quad \text{and} \quad i_{inst} = I_{RMS} \cos(\omega.t + \phi) \quad [4.6.1]$$

,where  $\omega$  = line frequency in  $\text{rad.s}^{-1}$  and  $\phi$  = phase delay

$$\text{then the real power is : } P_{real} = V_{RMS} \cdot I_{RMS} \cdot \cos(\phi) \quad [4.6.2]$$

If external components are used to generate these RMS values and the power factor is calculated, a simple multiplication operand would produce the desired power quantity. However, if the use of external RMS-to-DC components is not viable for OCM, it may be possible to calculate these values on-board whilst still offering a reduction in the required overhead.

The nature of the calculation described by Equation 4.6.2 above offers the following advantages:

- Lower susceptibility to non-concurrent sampling of the two input channels through individual averaging of the respective inputs before they are integrated into the power calculation
- Lower susceptibility to any phase shift introduced between the two input channels by the signal conditioning circuitry, for the same reason as stated above
- If external RMS-to-DC ASICs or transducers are used, the measurement process is considerably simpler and requires less overhead than alternative methods
- These devices also eliminate the need for a bipolar power supply
- Certain RMS-to-DC producing ASICs are waveform independent, measuring true RMS values for highly distorted signals
- If required, power factor information is readily available

However, using the RMS values to calculate the system power also has several distinct disadvantages over alternative methods. These are discussed in the relevant sections below.

There exist several options for obtaining the RMS values as well as measuring the phase delay and calculating the corresponding power factor. The below sections concern the details of these methods and the viability of implementing them as an OCM solution. Both RMS calculating ASIC's and transducers are investigated, as well as a possible method for calculating these on-board from discrete samples. Two methods of measuring the phase delay and calculating the corresponding power factor are also considered.

#### **4.6.2 Obtaining the input RMS values**

There exist three distinct options for obtaining the RMS values of the input waveforms for the purpose of OCM ; (1) the utilisation of RMS-to-DC transducers, (2) RMS-to-DC ASICs, and (3) on-board calculation using discrete samples.

In order to obtain the values required for Equation 4.6.2 above, an RMS averaging process must be employed over a minimum period of one full mains cycle [1]. In a steady state condition, a longer refresh period will yield more accurate results due to the averaging process of the RMS calculation. However should the input fluctuate faster than this refresh period, the extended averaging will tend to attenuate the influence of these higher frequency components. Thus a trade off exists between averaging out the influence of random noise and quantisation error in discrete cases, and incorporating the higher frequency components of the waveform. The refresh periods of commercially available RMS calculating ASICs designed for a 50Hz system vary from 0.2 seconds up to 1 second.

##### **4.6.2.1 RMS-to-DC Producing Transducers**

RMS current-to-voltage and voltage-to-voltage transducers produce analogue output signals directly related to the RMS equivalent of the line current and/or voltage.

Generally, these contain an inherent error, usually stipulated over a dynamic range, which does not conform to IEC 1036 specifications for a Class 1 meter. Furthermore

they are relatively expensive compared with their counterparts, the RMS calculating ASICs.

One of these devices, the CR4310 from CR Magnetics [12], was considered as a possible solution. The CR4310 series current-to-voltage transducer produces a 0-5 VDC output signal that is directly proportional to the input AC current. The output signal is average-sensing, calibrated for RMS.

This transducer is primarily used with process control and industrial instrumentation equipment. The DC output signal may be directly connected to the high impedance analog input of a standard MCU without any signal conditioning. The CR4310 series measures up to 200% of full scale on a short time basis (1-minute or less) and 150% on a continuous basis at its rated accuracy. This conforms to IEC 1036 specifications concerning overvoltages and currents.

The below table contains an overview of the device specifications

Specification	Comment	Compliance with IEC1036
Accuracy	Depends on range and load resistance. Refer to Table 4.5 below.	Refer to Table 4.5 below.
Ripple	1% Max. peak ripple on output	Conforms
Calibrated Signal Out	0 – 5VDC	N/A
Response time	250ms Max, 10-90% of full scale	Conforms
Max. Signal Out	12VDC	N/A
Frequency	50/60Hz	Required Operating range specifications not supplied. Assume this does not conform
Insulation Class	600V, or 10kV Full Wave	N/A
Input Range	Type dependant, refer to Table 4.5 below	conforms
Load Resistance	Refer to Table 4.5 below	Conforms to maximum allowable transducer dissipation for class 1 meter
Operating Temperature Range	-30° to +60°	Conforms

Table 4.4: CR4310 RMS Transducer Specifications

The CR4310 series includes 10 respective transducers designed for a scope of input ranges, however only those conforming to the IEC specifications were considered.

Input Range [A <sub>RMS</sub> ]	Accuracy [% of FS]	Minimum Load Resistance[Ω]	Compliance with IEC1036
0-75	0.5	160K	Input range does not conform Accuracy conforms to class 2 Power dissipation conforms to class 1
0-100	0.5	160K	Input range conforms to class 1 Accuracy conforms to class 2* Power dissipation conforms to class 1

\* - Since IEC1036 specifications stipulate a maximum range of 0-80A, the specified accuracy will not conform to class 2 over this reduced full scale range.

**Table 4.5: CR4310 Input Range and Accuracy**

A cost analysis for these devices was unavailable at the time of this report, however due to the specialised nature of the device it is assumed they would offer no financial benefit over the AD7755 (refer to Section 4.10) currently used by EML.

#### 4.6.2.2 RMS-to-DC ASICs

These devices output a low DC voltage representing the RMS value of the input waveform. They may be used for both current and voltage inputs, and are easily interfaced with a MCU ADC with minimal signal conditioning. Several of these devices were briefly investigated, however as with their transducer counterparts above, they offer no financial benefit over the present measuring solution. For this reason only the most cost-effective device found was considered, the AD736 from Analog Solutions [13].

The AD736 is a low power, precision, true RMS-to-DC converter. It provides a maximum error of  $\pm 0.3 \text{ mV} \pm 0.3\%$  of a full-scale input range of 200mV with sine-wave inputs. This corresponds to a static error of below 0.15%. The error over a dynamic range is not stated, however this is assumed to be well below IEC1036 stipulation for a class 1 meter.

The AD736 computes the RMS value of both AC and DC input voltages. It can also be operated ac coupled by adding an external capacitor. In this mode, the AD736 resolves input signal levels of 100 mV RMS or less, despite variations in temperature or supply voltage. This is a benefit when offset errors are present in the transducer output. The device conforms to all relevant IEC specifications for a class 1 meter when used in conjunction with an appropriate current transformer and/or shunt [13].

An international retail price per unit was stated as \$5,90 (around R41,00), however the price per unit quota may be expected to drop by up to 50% for larger batches. This still does not offer any financial benefit over the present measuring solution.

#### 4.6.2.3 On-board RMS calculations using discrete samples

The two methods of obtaining RMS values of the input waveforms using external components described in section 4.6.2.1 and 4.6.2.2 above offer no financial benefit over the current solution. Therefore, if an RMS calculating method is to be viable for OCM, it is concluded that no specialised external components may be used other than a current shunt and several discrete signal conditioning components. All RMS calculations must therefore be conducted on-board the MCU.

Using the fundamental mathematical relationship between continuous and discrete signals, it may prove possible to calculate the RMS equivalents of the input waveforms on-board the MCU. A lengthy search produced several algorithms for performing such calculations, however these were all written in higher level languages requiring access to full math libraries. For the purpose of this investigation, the method described below was developed in order to assess the viability of conducting these calculations on-board.

As described in Section 2.2, the RMS values of the input signals may be calculated as follows:

$$V_{RMS} = \sqrt{\frac{\Delta t}{T} \sum_{n=0}^N v_n^2} = \sqrt{\frac{1}{N} \sum_{n=0}^N v_n^2} \quad \text{and} \quad I_{RMS} = \sqrt{\frac{\Delta t}{T} \sum_{n=0}^N i_n^2} = \sqrt{\frac{1}{N} \sum_{n=0}^N i_n^2} \quad [2] \quad [4.6.3]$$

, where once again  $\Delta t$  is the time between samples,  $T$  is the mains cycle period and  $N$  is the total number of samples per cycle.

Equation 4.6.3 provides an estimate for the RMS values of the input current and voltage waveforms regardless of the shapes of the input waveforms<sup>[2]</sup>. A simpler RMS calculating method exists for purely sinusoidal waveforms, however this method is highly susceptible to waveform shape<sup>[14]</sup>.

The peripheral characteristics such as sampling frequency, input range, quantization error, et cetera are not discussed in this section and for the purpose of the immediate investigation they are assumed perfect (i.e. a sampling frequency higher than the Nyquist criteria for the highest frequency component, a full scale input range, zero quantization error, etc).

The most complex operation of Equation 4.6.3 to compute on a MCU is the square-root function. The range of values to be evaluated covers several million depending on the number of cycles evaluated over and the sampling frequency, eliminating the use of a look-up table as a viable option. Thus, a square-root calculating algorithm remains the only alternative.

Although several dedicated methods for calculating the square root of an integer exist, it was decided to use Newton's Method. The reason for this is two fold: (1) the method is both simple and easy to implement using an MCU, and (2) designing the algorithm from scratch facilitates the calculation of the required processor overhead.

Newton's method is an iterative process that requires a variable number of loops in order to converge within the preset error limits. For this reason the maximum number of loops was limited to 20, although the system will generally converge with less.

Briefly, Newton's method for solving square roots is as follows [2]:

If one starts with a guess of  $r$  as the solution to  $\sqrt{x}$ , then a better guess will be

$\frac{r}{2} + \frac{x}{2r}$ . The theoretical proof for this is not within the scope of this report.

The pseudo-code on the ensuing page illustrates the algorithm developed for implementing Newton's method.

Function Sqrt(x) :=

```

ε = 1
{set the tolerance level to 1}

Max_It = 20
{set the maximum number of iterations to 20}

i = 0
{initilase a dummy variable to be used as a loop counter}

r = x/2
{set the first guess, r, to half the value being evaluated}

μ = r/2 + x/(2×r)
{from Newton's method, the second guess is μ}

while [(absolute value of [μ-r]>ε) and (i ≠ Max_It)]
{compute the following loop while the error is greater than the allowable tolerance and while
the number of iterations is less than the allowed maximum}
    r = μ {refresh the new guess from the updated one in the
           previous iteration}
    μ = r/2 + x/(2×r) {calculate the next "better" guess}
    i = i+1 {increment the number of iterations by one}

sqrt(x) = μ {the answer, either after 20 iterations, or the error is less
             than the tolerance}

```

The tolerance,  $\epsilon$ , has been set to 1 in order to allow for integer subtraction and addition, however this results in a loss of precision of 1 in 256, or 0.039%. One possible solution to better this is to multiply the value of  $x$  by a known square, for example 4, and then dividing the final solution by it's square root, i.e. 2. This effectively increases the value being computed reducing the error associated with a relatively large tolerance.

Using the above method, the square root of the summed squared samples of a simulated voltage waveforms was calculated, which when divided by the square root of the number of samples taken, provides RMS information according to Equation 4.6.3. The test parameters used are as follows (the input signal was generated using an HP 33120A Digital Signal Generator):

- VRMS = 1.768V (corresponds to a 5V ptp signal)
- Line frequency = 50Hz
- Sampling frequency  $\approx$  2000Hz
- Sampling ADC resolution– 8 bits

- Sampling ADC resolution – 8 bits
- Refresh period = 40 ms – corresponds to two full cycles

The input waveform was offset by 2.5V to provide a unipolar ADC input signal. The most significant bit of the respective sampled values was used as a sign bit.

Given these, the value of N was calculated as:

$$N = \frac{1}{50} \times 2000 \times \frac{0.04}{\frac{1}{50}} = 80$$

and the value to be evaluated was calculated as:

$$\sum_0^{80} (V_{\text{sampled}})^2 = 645748$$

The results of the square-rooting iteration were as follows:

- The initial guess, r, was half of the above: 322874.
- MCU Calculated solution: 803
- Actual Solution: 803.584
- Error: 0.068%
- Number of required iterations: 12

When divided by the square root of the number of samples (i.e. 8.944) this provides real RMS information according to Equation 4.6.3. However, since the written math functions, including the division, contain a discrete tolerance of 1, only integer values may be used. For this reason, the calculated values of 803 and 8.944 are both bit-shifted by one byte (multiplied by 256) in order to increase the overall resolution.

Thus the actual division operation computed is as follows:

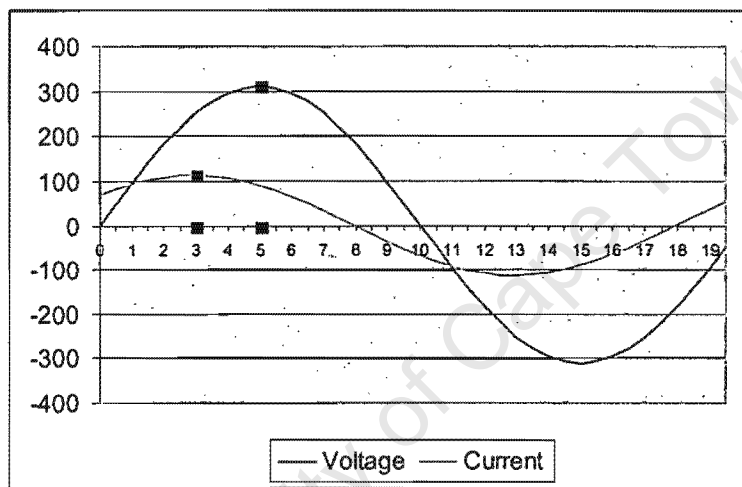
$$V_{RMS} \equiv \frac{803 \times 256}{8.944 \times 256} = \frac{205568}{2290} \approx 90$$

,which when multiplied with a scaling constant represents the RMS value of the input.

### 4.6.3 Obtaining the Power Factor

Several possible methods of measuring the phase difference and calculating the corresponding power factor were investigated, however only one is presented below. These operations were performed by the MCU from discrete samples of the input waveforms using the developed algorithm, the derivation of which is discussed below.

In order to solve Equation 4.6.2, the value of  $\cos(\phi)$  must be calculated. Since  $\phi$  is defined as the lag or lead (in degrees) between the two signals, it is directly proportional to the time difference between the peaks of the two respective signals, provided they contain no harmonics. This is clarified in Figure 4.9 below.



*Figure 4.9: An example of the difference in peak values for Voltage and Current*

The power factor has been set to 0.8 leading and the current thus leads the voltage by  $36.86^\circ$ . The x-axis depicts the time elapsed (in ms) for a frequency of 50Hz.

The peak value of the voltage signal occurs at 5ms and the current at 3 ms as shown. Let the time difference between the two peak values be  $\Delta t_p$ . In this case  $\Delta t_p = 2$  ms. The phase angle is therefore approximately:

$$\phi \approx 360^\circ \times 50 \text{ Hz} \times \Delta t_p = 36^\circ \quad [4.6.4]$$

The calculated power factor would then be  $\cos(36^\circ) = 0.809$ .

The actual phase calculation implemented uses the zero-crossing as a reference, however considerable errors may be introduced for signals containing offsets or

strong harmonic content. To compensate, an estimation of the actual zero-crossing point is made using the points around it. This is done by storing each respective sample of current and voltage in an array. After sampling 40 values of current and of voltage, corresponding to one full cycle period at the stipulated sampling frequency, a simple loop is run to find the sample closest to the negative to positive zero crossing point.

Each point in the array is checked until a positive value is found. Once this is completed, each adjacent sample is tested and if found to be positive, a variable is incremented. For each negative value found, the variable is decremented until the variable reaches zero. This is illustrated below:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Value	21	16	10	4	-2	1	3	-1	-5	-10	-16	-22	-29	-37

Variable	1	2	3	4	3	4	5	4	3	2	1	0		
----------	---	---	---	---	---	---	---	---	---	---	---	---	--	--

↑  
Start
↑  
Stop

The samples in this example illustrate a waveform with a strong 3<sup>rd</sup> harmonic component, however random noise may also produce the effect of a slight 'jitter' at the zero-crossing point. The samples begin as positive and the variable is incremented until the sample in index 4 is reached, where the variable is decremented. This continues until index 11 where the variable reaches zero. The zero crossing is then deemed to have occurred at the sample with an index value of half way between the starting and stopping points of the variable. In this example, the halfway mark between the start (index 0) and stop (index 11) is index 6. Thus the seventh element in the array marks the zero crossing point of the sampled wave. This method is employed in an attempt to reduce errors due to any jitter around the zero mark of non-sinusoidal waveforms.

This method is performed on both the current and voltage samples, providing two index values for the zero crossing points. The value  $\Delta n_p$  is defined as the difference in these array indices.

With a sampling frequency of 2 kHz, adjacent entries in the array correspond to the instantaneous values of the input signals sampled 500 $\mu$ s apart (the sampling period). Thus two values which are  $\Delta n_p$  indices apart will correspond to a sampled time difference of:  $\Delta t_p = 500\mu\text{s} \times \Delta n_p$  [s]. From Equation 4.6.4 this relates to a phase angle of:  $\phi = 360^0 \times 50 \times 0.0005 \times \Delta n_p = 9 \times \Delta n_p$ . Thus the power factor calculation is reduced to:

$$PF = \cos(9.\Delta n_p) \quad [4.6.5]$$

Two methods for calculating  $\cos(\phi)$  (once  $\phi$  is known) were attempted. The first involves a lookup table, which requires minimal processor time, but may use more memory. Furthermore, a lookup table works best with known discrete values and since  $\phi$  is a 'continuous' wide ranging variable, this may prove tedious to implement.

A more applicable method involves Taylor expansions. From these, it has been proven that the following approximation is true [2]:

$$\cos(x) \approx 1 - \frac{x^2}{2} + \frac{x^4}{24} \quad [4.6.6]$$

This calculation is easily performed and provides a very close approximation. One problem, however, is that this version of the Taylor expansion is valid for  $x$  in radians only. In order to keep a high precision it was more applicable to calculate the value of  $\phi$  in degrees. Therefore it was necessary to convert from degrees to radians in order to calculate  $\cos(\phi)$ . The power factor calculation was thus modified to:

$$PF = \cos(\phi) = 1 - \frac{1}{2} \left( 9\Delta n_p \times \frac{2\pi}{360} \right)^2 + \frac{1}{24} \left( 9\Delta n_p \times \frac{2\pi}{360} \right)^4 \quad [4.6.7]$$

The 32-bit division and multiplication subroutines developed for the purpose of this investigation work best when the inputs and outputs have an absolute integer value greater than one.

It is possible to adapt the division function to handle fractions, however a simpler method would be to multiply both the divisor and dividend by the same constant. The quotient may still lie between  $-1$  and  $1$ , however, and for this reason the remainder is computed to a further two bytes after the quotient has been calculated.

With this in mind the equation now used to calculate the power factor is (from the Equation 4.6.7):

$$PF = 1 - \frac{\Delta n_p^2 \times 256}{81.06 \times 256} + \frac{\Delta n_p^4 \times 256}{39421.37 \times 256} \quad [4.6.8]$$

The latter two terms in this equation produce fractional values, which are complex to represent in an 8-bit processor. This further complicates the addition and subtraction process required to resolve Equation 4.6.8 above. In order to overcome this, all the terms, including the resolved remainders, are further offset by two bytes (i.e. multiplied by  $256^2$ ). The PF will then be represented by an integer value as follows:

$$PF \equiv 255 - \frac{\Delta n_p^2 \times 256^2}{81.06 \times 256} + \frac{\Delta n_p^4 \times 256^2}{39421.37 \times 256}$$

The PF algorithm was tested for various input parameters. It was found that the method is accurate to around 0.72% for purely sinusoidal input signals. Signals containing strong harmonics or offsets, however, yielded considerably larger output errors (between 2% and 5%).

## 4.7 Reduced Scan Principle

### 4.7.1 Overview

The “Reduced Scan Principle” [15] is an interpolation method that alternately measures current and voltage at regular time intervals, and effectively multiplies each voltage sample by an interpolated current sample. While the standard power measurement method depends on the assumption that both channels are sampled simultaneously, the reduced scan principle interpolates data in such a manner as to allow for non-concurrent sampling.

The sampling occurs in an alternating fashion, that is, one channel is sampled at the stipulated sampling instant and then the other at the next sampling instant. Each current sample is used twice, once with the previous voltage sample and once with the following voltage sample. The results of these two operands are added together for each value of  $n$ .

This method introduces a small inherent error caused by the associated phase shift, and also through the linearity assumption of the input signals between adjacent samples. A “correction constant” is then used in an attempt to minimise these effects in the power calculation.

The reduced scan provides three distinct advantages over the standard technique

- Concurrent sampling of the two input channels is not necessary
- Only half the measurements for the same period are necessary, since every measured current or voltage value is used twice
- Only half the multiplication operations are necessary because two current values are added before multiplication

### 4.7.2 Power Calculation Method

The standard discrete average power calculation method described in Section 2.2 is

$$P = \frac{1}{N} \sum_{n=1}^N v_n i_n \quad [4.7.1]$$

,with  $v_n$  and  $I_n$  as the samples of voltage and current taken at time  $n$ , respectively.

The corresponding energy [in watts] measured over the time period represented by  $N$  samples is therefore

$$W = \sum_{n=1}^N v_n i_n \times \Delta t \quad [4.7.2]$$

, where  $\Delta t$  represents the time between adjacent samples. Thus the total time over which Equation 4.7.2 is evaluated corresponds to

$$T = \sum_{n=1}^N \Delta t = N \times \Delta t \quad [4.7.3]$$

The reduced scan method is based on Equation 4.7.1 with a small modification [15]

$$P_{red} = \frac{1}{N} \sum_{n=1}^N (v_n \times i_{n-1}) + \frac{1}{N} \sum_{n=1}^N (v_n \times i_{n+1})$$

,which may be reduced to

$$P_{red} = \frac{1}{N} \sum_{n=1}^N v_n \times (i_{n-1} + i_{n+1}) \quad [4.7.4]$$

, with current samples  $i_{n-1}$  and  $i_{n+1}$  taken at times  $n-1$  and  $n+1$  respectively. The terms parenthesis constitute twice the average value of the estimated current sample  $i_n$ :

$$i_{n-1} + i_{n+1} \approx 2i_n \quad [4.7.5]$$

This is a linear approximation to a non-linear input and forms the basis for the reduced scan.

Thus, in energy terms from Equations 4.7.3 through 4.7.5,

$$W_{red} = \sum_{n=1}^N v_n \times (i_{n-1} + i_{n+1}) \times \Delta t = T \sum_{n=1}^N v_n \times (i_{n-1} + i_{n+1}) \text{ [Watts]} \quad [4.7.6]$$

over a time period  $T$ . This is depicted in Figure 4.10.

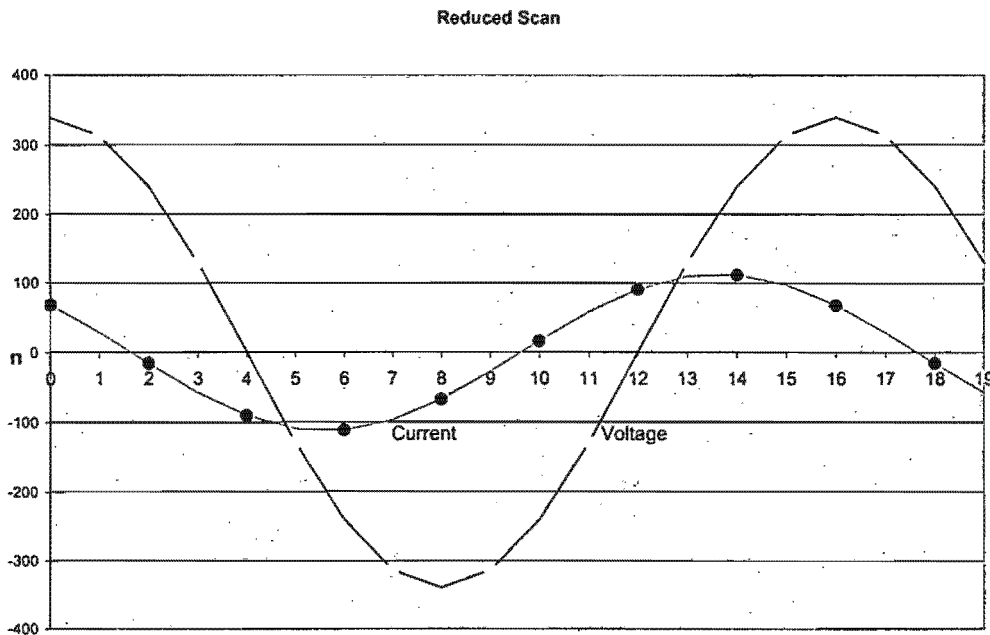


Figure 4.10: Graph showing sampling points for reduced scan method

The darker dots indicate the sampling points of the current waveform and the lighter of the voltage. Sampling begins at  $n = 0$  with the current waveform and  $I_n$  is obtained. As  $n \rightarrow n+1$ , the voltage is sampled and  $v_n$  is obtained, and the current sample becomes  $i_n \rightarrow i_{n-1}$ . At  $n = 2$ , the current is sampled again and temporarily termed as  $i_{n+1}$  so as to compute the first estimated instantaneous power value. The sampling process begins again and a cumulative sum of these “instantaneous” powers is made over a time period of  $T$ . This continues until  $n = N$ , upon which the result of the summed “instantaneous” power is multiplied with the elapsed time to provide energy information in watts according to Equation 4.7.6.

### 4.7.3 Inherent Error Compensation

The stipulated error calculation was derived from the inverse of the correction constant in Equation 4.7.8 below, and is as follows: [15]

$$e = [-\cos(\Delta t \times f \times 2\pi) \times 100] + 100 \quad [\%] \quad [4.7.7]$$

,with  $f$  as the mains frequency. The accuracy of this calculation is discussed in Section 4.7.4.

The correction constant,  $c$ , is normally included in the calibration phase and not used explicitly. It was stipulated as: [15]

$$c = \frac{1}{\cos(\Delta t \times f \times 2\pi)} \quad [4.7.8]$$

#### 4.7.4 Results of Simulated Data

In order to confirm this method using empirical means, an Excel worksheet was fabricated with variable parameters of input frequency, input rms values of current and voltage, power factor, sampling frequency, and total number of samples used. All calculations were performed using values accurate to at least the 10<sup>th</sup> decimal place where applicable, thus quantisation was not accounted for. The objective of the exercise was to gain a better arithmetic assessment of the effects of using interpolation methods, rather than simulate actual implementation on an MCU. For this reason, all output data from this worksheet discussed below is used in a purely comparative fashion, and in no way indicates values predicted for an MCU-based system.

It was confirmed that only the two parameters stipulated in Equation 4.7.7, the line frequency and sampling period, affect the output error (disregarding peripheral influences which would be present on a real system such as noise, dynamic line frequencies, distorted input signals, etc).

Equation 4.7.6 was implemented on a maximum number of input samples  $N = 1360$ , this corresponding to 34 cycles at a maximum sampling frequency of 2kHz. The total number of cycles sampled over was changed by altering the sampling or line frequency. It was found that changes in input amplitude or power factor had no influence on the output error, as predicted by Equation 4.7.7.

The inherent error,  $e$ , was calculated for each set of input parameters using Equation 4.7.7 and also using empirical means. Two of these results are displayed in Table 4.6.

$e$ computation method	Line Frequency [Hz]	Sampling Frequency [Hz]	Number of Samples $N$	Number of Cycles	$e$ value obtained [%]
Theoretical	50	500	50	5	19.0983
Empirical	50	500	50	5	21.13
Theoretical	50	2000	80	2	1.23%
Empirical	50	2000	80	2	0.02%

**Table 4.6:** Comparison of  $e$  value using empirical and computed methods

As can be seen, the theoretical error calculated using Equation 4.7.7 does not correspond exactly to the empirically found error. Since the correction constant  $c$  is derived from this, it too will not correct the computed output exactly.

Using the Excel worksheet, the corrected reduced scan result was compared with the theoretical energy derived from the input parameters. The following results were recorded with input parameters of  $V_{rms} = 240V$ ,  $I_{rms} = 100A$ , unity power factor and line frequency of 50Hz.

Sampling Frequency [Hz]	Number of samples $N$	Number of Cycles	Time Period $T$ [s]	Reduced scan output [Whr]	Calculated Output [Whr]	Error [%]
500	200	20	0.4	2.66666	2.65333	0.5
2000	400	10	0.2	1.33333	1.33	0.25
6000	600	5	0.1	0.66666	0.66555	0.17

**Table 4.7:** Output error using reduced scan on simulated data

The data was measure explicitly over integral cycles simulating ideal sampling.

It stands to reason that since the interpolation method is a linear approximation, a lower sampling frequency will reduce the accuracy of the linear estimate. Clearly, the linear approximation of a non-linear function becomes increasing inaccurate as the time between successive samples is increased.

## 4.8 Power measurement using rectified inputs

### 4.8.1 Overview

The 10-bit and 8-bit ADCs on the Hitachi and Motorola MCUs provide lower resolutions than all the measuring solutions investigated. Unfortunately, these parameters cannot be improved, and if the on-board ADCs are to be used the decreased resolution will undoubtedly reduce output accuracy. One possible method of effectively increasing the output resolution of these converters is to full-wave rectify the input signals, thereby eliminating the need for a sign bit in the sampled word and increasing the measurement resolution by a factor of 2.

The polarity of the input samples is required to determine the power factor, and thus cannot be neglected. Instead, it is proposed that the polarity of each sample of current and voltage be measured using a comparator with an analog ground reference and digital output, and be represented by a sign-flag comprising of a single bit in a memory. This may then be integrated with a modified power calculation to incorporate signed values.

The system thus requires the use of four ADC inputs, two per input channel respectively. Each ADC input receives half the waveform for it's respective channel. The sampling of these channels may be firmware handled from the logic levels of the two digital inputs representing the polarity of the input waveforms.

The proposed system is presented in Figure 4.11.

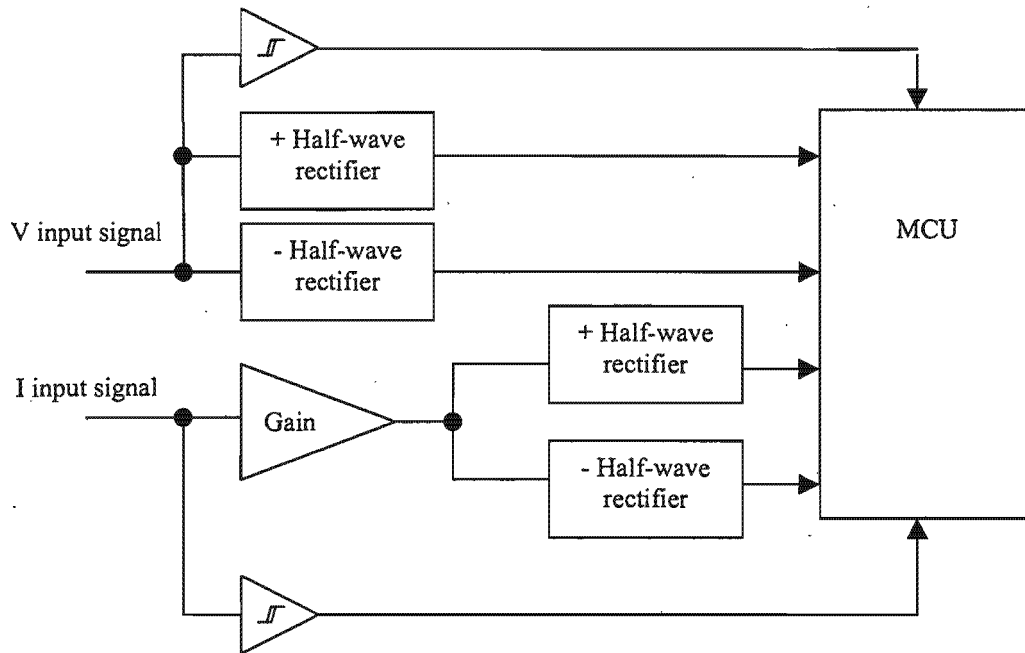


Figure 4.11: Proposed rectification process block diagram

The output of each analog ground-referenced comparator is a digital signal, the level of which defines the polarity of the input waveform, and is fed to a standard digital I/O line of the MCU. The channel bandwidth, sampling frequency, influences of harmonic distortion, temperature fluctuations and noise were not considered in this particular investigation.

#### 4.8.2 Increased Resolution

The theory behind the increased resolution obtained by using rectified inputs is explained with the aid of the figure below.

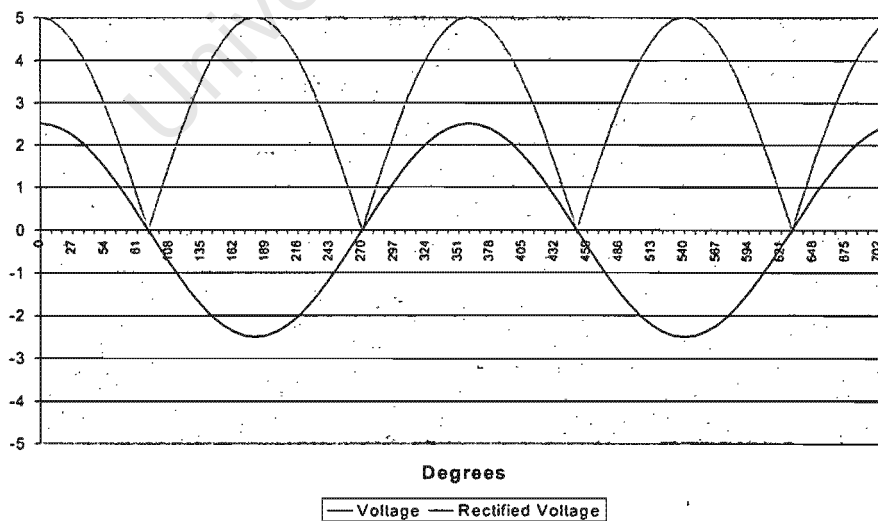


Figure 4.12: Rectified input signal

For bipolar signals, the ADC samples over the full range of  $-2.5\text{V}$  to  $2.5\text{V}$ . Thus each half cycle is measured using one sign bit and 9 bits for the Hitachi MCU and 7 bits for the Motorola MCU. The rectified signal, however, is unipolar eliminating the need for a sign bit. The full ADC range may now be used to measure each half cycle, effectively increasing the resolution by a factor of 2. The polarity of each sample obtained has an associated sign-flag elsewhere in memory, which defines its polarity.

### 4.8.3 Obtaining rectified inputs

Since the whole basis for the proposed method relies heavily on obtaining accurate rectification, a number of simulated tests (using MicroSim Spice Evaluation Version 8.0, July 1997) were performed before a modified power calculation was investigated in an attempt to find a viable solution. This analog rectification process proved more troublesome than was originally expected. Several variations of rectification circuits were both simulated and constructed, most of which proved unusable due to the 1.2 or so volts deadpan inherent in standard diode rectifiers.

The solution found to be most viable is presented below. These circuits were simulated, and then constructed and tested with simulated inputs.

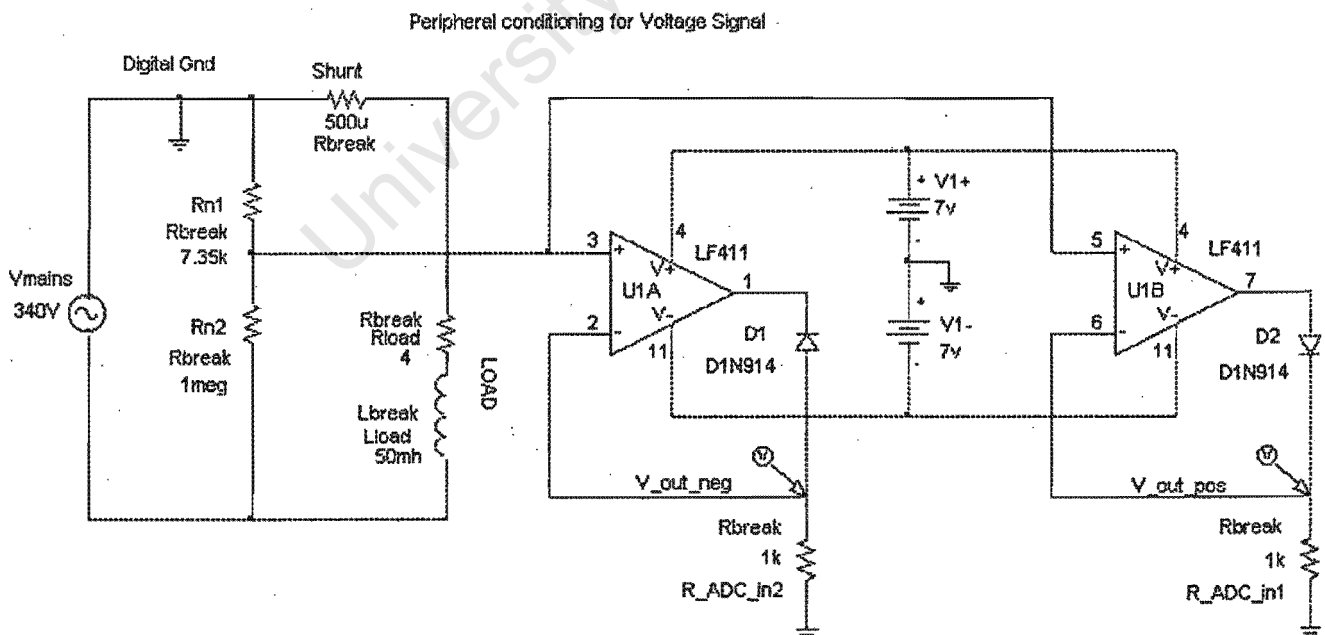


Figure 4.13: Developed voltage rectification circuit schematic

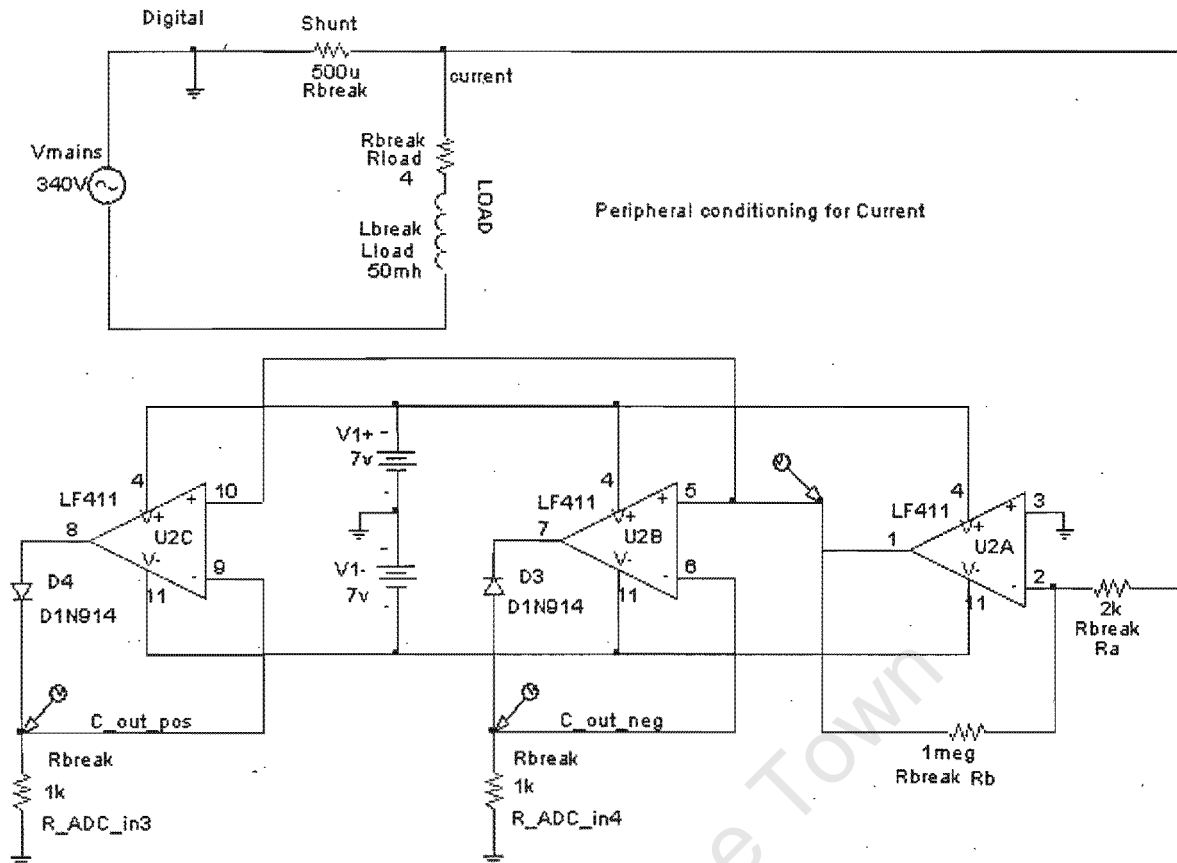


Figure 4.14: Developed current rectification circuit schematic

The exact theory behind these circuits is not within the scope of this report, but is available on request.

#### 4.8.4 Rectification Test results

Using the above circuits in a software-simulated environment, it was found that the system produces an unacceptable error. Still, the results obtained were the most accurate of all the attempted circuits. Several tests were performed, all of which contained errors above those allowed by IEC 1036 regulations.

In particular, the following results were recorded. The “input” was measured as the voltage signal present at the input of the two half-wave rectifiers of the voltage channel, and the output of the gain amplifier of the current channel respectively. The “output” is the respective voltage signal fed to the MCU. Error is presented as percentage of input value.

Input Channel	Input Signal polarity and Value [V]	Output Value [V]	Difference [mV]	Error [%]
Voltage (pos)	+ 4.15	+ 4.10	50	1.20%
Voltage (neg)	- 4.15	+ 4. 07	70	1.69%
Current (pos)	+ 4.70	+ 4.64	60	1.28%
Current (neg)	- 4.70	4.61	90	1.91%

Table 4.7: Simulated test results for rectified signals

The system was tested to determine if these errors were linear, in which case they may possibly be compensated for, however it was found that the error is significantly increased for smaller input values in a non-linear fashion. Furthermore, it was possible to notice a slight distortion on the output waveforms, particularly for input signals with amplitudes below 2V.

After considering possible modified power computation algorithms for this system, it is predicted that high-pass filtering of the input signals using digital means will still be possible. However, this algorithm may be significantly more time consuming than one using the standard method of power measurement.

## 4.9 Microchip Microcontroller Based Watt-Hour Meter Reference Design

### 4.9.1 PICCREF-3 Overview

The PICREF-3 Watt-Hour Meter (WHM) Reference Design [14] shows the use of a mixed signal microcontroller from Microchip in an AC power measurement application. The system is capable of monitoring and displaying power and energy consumption on worldwide power mains in the 90V to 264V range. The signal processing circuitry consisting of sensor, signal conditioning electronics, ADC and microcontroller is abbreviated by the use of the mixed signal microcontroller with its on-board ADC.

The mixed signal microcontroller used is the Microchip PIC16C924. This microcontroller has five ADC channels, two of which are used to digitize voltage and current signals. The microcontroller features of pulse width modulation (PWM) and direct liquid crystal display (LCD) drive are also utilized to further reduce cost and parts count. The PWM output feature is used with a single pole RC filter to provide a comparator reference with 10 bits of resolution. The direct LCD drive is used to drive an 8-digit 7-segment LCD for outputs.

All measurements and calculations are performed once per second. The current waveforms measured are linear for resistive and inductive loads and non-linear for switching power supplies. The current waveform is sampled during the positive current cycle with waveform symmetry assumed between positive and negative cycles, although full cycle measurement is possible with some hardware and firmware modifications.

It should be noted that only those aspects of the PICREF-3 design having pertinence to the present study have been reviewed in this section. All other technical documentation regarding this design, including the firmware and circuit schematics is provided on the CD accompanying this report for further review.

### 4.9.2 System Overview

Figure 4.15 represents the complete PICREF-3 system block diagram.

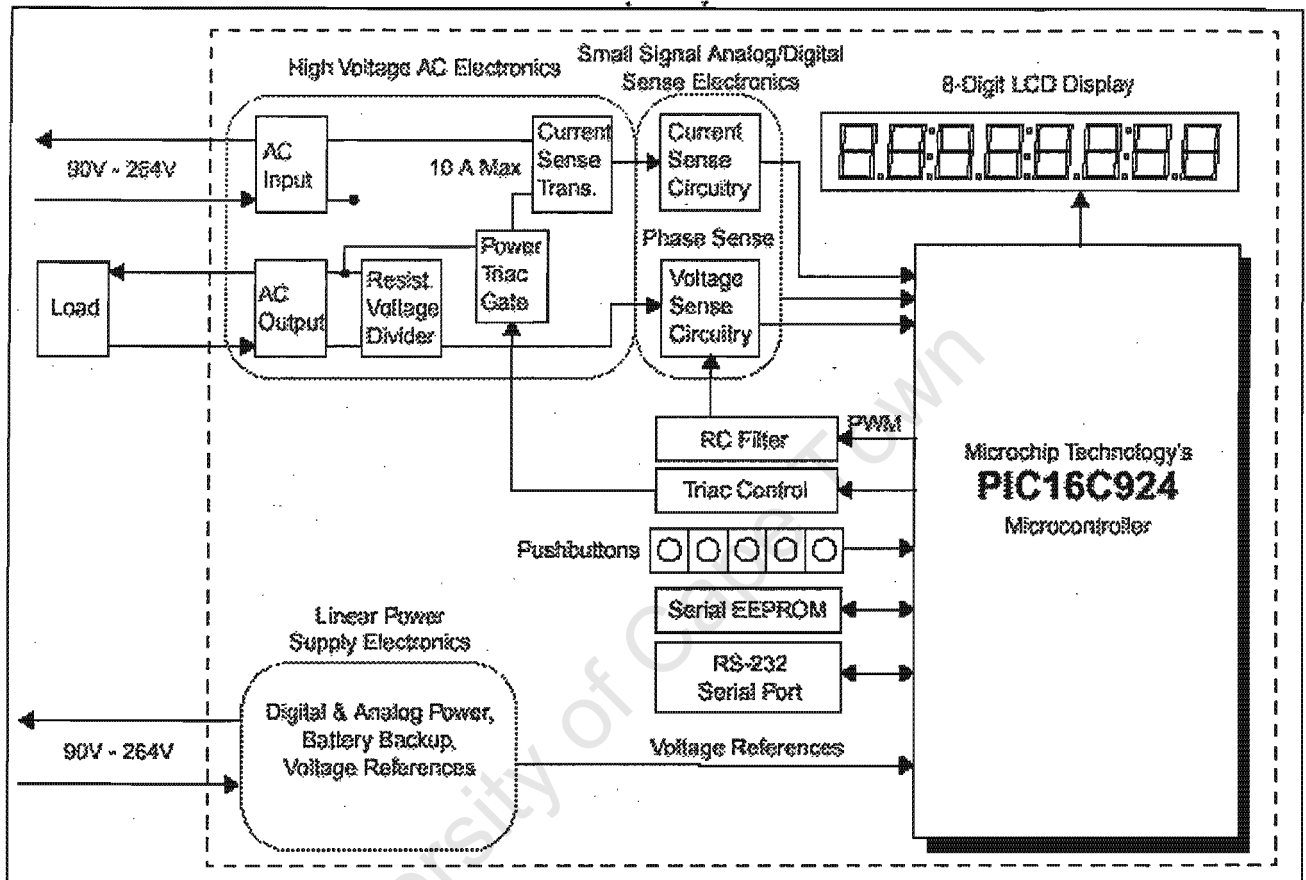


Figure 4.15: PICREF-3 Watt-Hour Meter Block Diagram

Microchip Technology, Inc. (1997) "Watt-Hour Meter Reference Design," PICREF\_3.pdf downloaded from <http://microchip.com>

The stipulated system is limited to measuring a maximum current of 10 A, however a maximum of up to 200A may be measured with minimal hardware modifications. Power line voltage (90V-264V) is sensed by the high voltage AC electronics. Balanced input voltage and current signals are provided to the Small Signal Analog/Digital Sense Electronics for input into the MCU.

The PIC16C924 uses the analog/digital voltage sense and analog/digital current sense to calculate the power and energy consumption of the load. In addition, the pulse width modulation (PWM) output of the microcontroller is filtered and used as a

inputs. The on/off state of the triac control circuit, which applies current to the power triac gate to effect on/off load current switching, is also controlled by the MCU.

A serial EEPROM is available to store the watt-hour count displayed at the time of a power outage. In addition, an RS-232 serial port is available for sending and receiving serial data. Firmware for these functions has not been implemented in this system, but may be developed if required. Linear power supply electronics provide general device power, voltage reference and battery backup.

The PIC16C924 source code is written in C. Firmware algorithms accomplish voltage and current measurements, voltage and current waveform phase shift measurements, real-time timekeeping, and calculation and display of power and energy consumption. In addition, there is code to display measurements on the LCD.

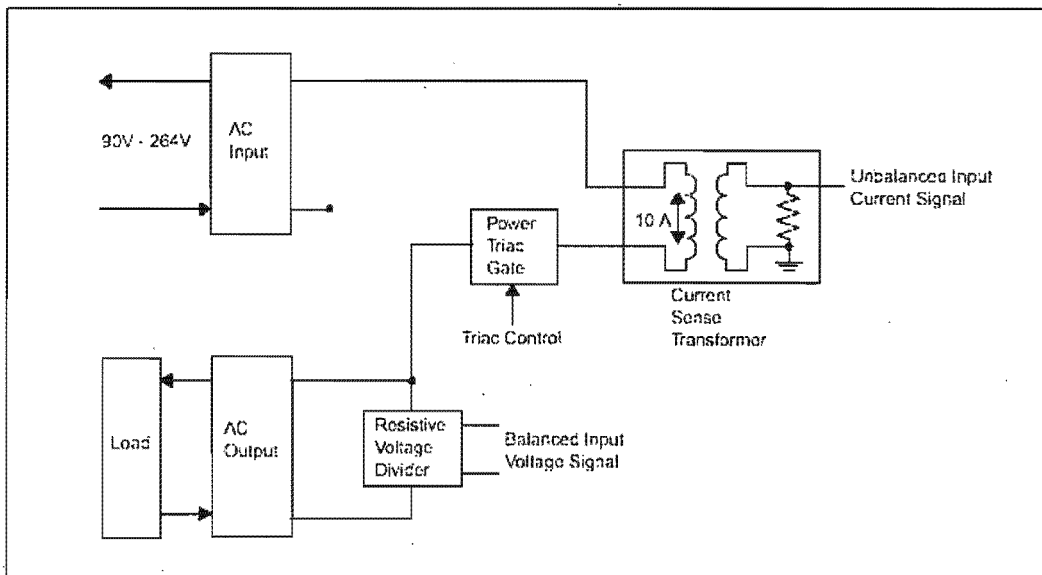
The Watt-Hour Meter (WHM) reference design has five modes of operation, determined by the PIC16C924 firmware: Reset, Entry, Measurement, Sleep, and Hibernate. All firmware regarding these is modular based (refer to accompanying CD).

#### **4.9.3 Hardware Overview**

This section describes the PICREF-3 hardware and how it functions in the system.

##### **Analog Signal Conditioning**

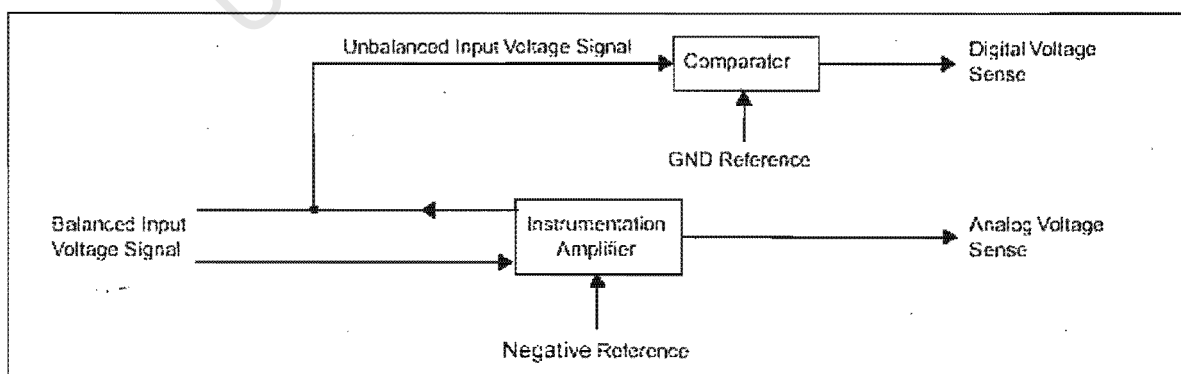
AC voltage (90V - 264V) is applied to the AC input and a load is connected to the AC output as shown in Figure 4.16 below. Line current flow is from the AC input to the AC output and through the load. This current continues through the triac gate and current sense transformer back to the AC input. It also flows through a resistive voltage divider via the AC output. The current sense transformer and resistive voltage divider provide low-voltage analog waveforms which replicate the large scale voltage and current waveforms at their respective sensor inputs.



**Figure 4.16:** High-Voltage AC Electronic Block Diagram

The balanced input voltage signal from the resistive divider is amplified by an instrumentation amplifier as shown in Figure 4.17 below. This amplifier accomplishes signal amplification and rejection of common mode noise. The signal at the amplifier output swings above and below a negative reference potential. The amplifier output is connected to the ADC converter input of the PIC16C924.

An unbalanced input voltage signal is tapped off of the balanced input voltage signal and connected to a comparator input. This comparator is used to compare the input signal to an analog ground reference, and its output goes high when the input signal exceeds this reference. This output provides digital voltage sense information and is connected to a digital I/O line of the PIC16C924.



**Figure 4.17:** Small Signal Analog/Digital voltage sense electronics block diagram

Similarly, the unbalanced input current signal from the sense transformer is amplified by an instrumentation amplifier as shown in Figure 4.18. This amplifier also accomplishes signal amplification and rejection of common mode noise and its output is connected to a second ADC of the PIC16C924

The unbalanced input current signal is also connected to a comparator input. This comparator is used to compare the input signal to a variable positive reference - the PWM reference. The comparator output also goes high when the input signal exceeds the reference and is connected to a digital I/O line of the MCU.

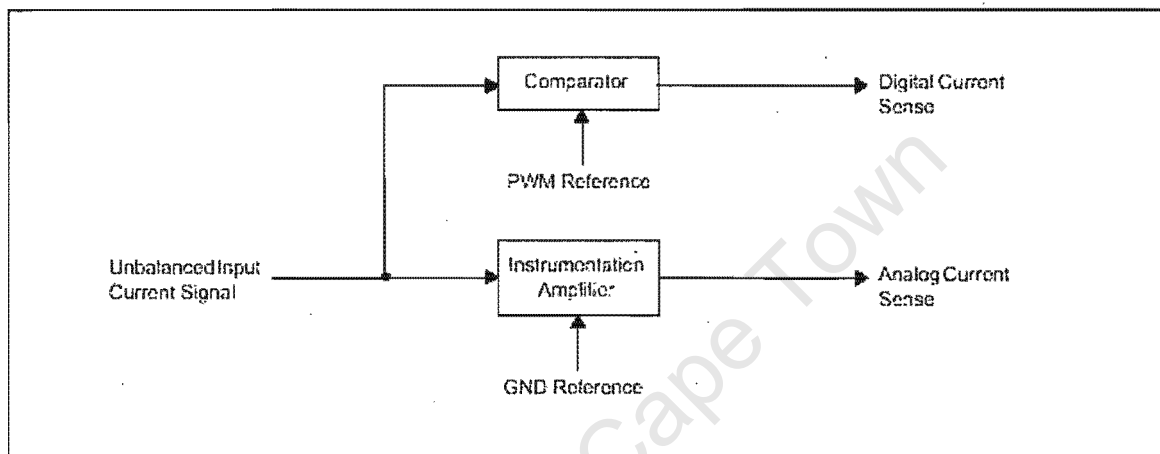


Figure 4.18: Small signal Analog/Digital current sense electronics block diagram

#### 4.9.4 Microcontroller Input/Output

The PIC16C924 is a “mixed-signal” microcontroller, or a microcontroller capable of analog and digital input/output as shown in Figure 4.18 below. This capability is used to control certain functions of the PICREF-3.

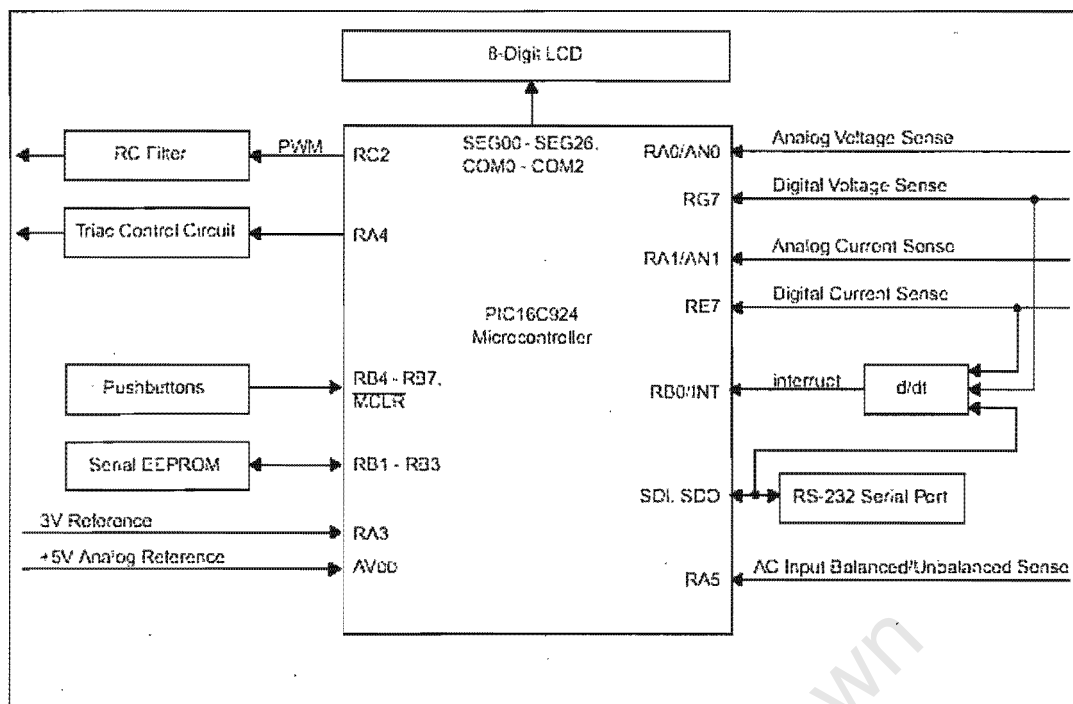


Figure 4.19: Microcontroller I/O block diagram

### Inputs

The on-board ADC converters of the PIC16C924 allow inputs of analog, as well as digital, voltage and current sense signals from the Small Signal Analog/Digital Sense Electronics (refer to Figure 4.18). Depending on user push-button input, these signals may be used for the display (LCD), calculation, or both. The pushbuttons provide digital logic signals to the MCU.

Voltages of +5V and +3V are provided for the MCU A/D converters as reference values, the latter providing better resolution.

The MCU has an external interrupt input (shown as RB0/INT in figure 4.21). This feature is used indirectly by the digital voltage sense line, digital current sense line, and serial communication input lines. Each of these signals is differentiated by identical circuits, and wire-OR'd together to obtain a single line input to the external interrupt. The differentiated digital voltage sense signal is used to wake the microprocessor from Sleep or Hibernate modes.

An A/C Input Balanced/Unbalanced Sense line provides the microcontroller with information on whether the input neutral line is grounded or ungrounded to support both polarized and unpolarised power grids.

### **Output**

The PIC16C924's feature of direct LCD drive capability facilitates the displaying of the relevant information. The user, through various pushbutton inputs, determines the information displayed.

Through the use of PWM and an RC filter, the PIC16C924 can vary the analog reference voltage to the current sense comparator (refer to figure 4.17) for use in true RMS current measurements. The microcontroller also controls the triac, through triac control circuitry, which may be turned on or off by pressing the START/STOP pushbutton. This will turn the current flowing to the load on or off.

### **Input/Output**

A Serial EEPROM is provided to store data in the event of a power loss. In addition, an RS-232 serial port is provided for communication with a computer. Firmware for these functions has not been provided, but may easily be developed.

## **4.9.5 Signal Measurements**

### **Current – True RMS Measurements**

True RMS measurements are highly dependent on waveform shape [14]. The watt-hour meter first determines waveform shape and then calculates true RMS values. The PICREF-3 provides algorithms for the fundamental shapes of linear resistive, linear inductive and narrow sinewave pulses.

The peak current-waveform amplitude sample is found in order to ensure that the comparator reference values do not exceed the waveform's peak amplitude. Current waveforms are then digitally scanned to determine waveform type. The digital scan is accomplished by varying the reference voltage, created by filtering the MCU's PWM output, of the digital current sense comparator (refer to Figure 4.15). As the comparator reference voltage is increased from ground to the maximum signal level

(in varying step sizes), digital pulses synchronized with the analog waveform zero crossings are created. A count of when the pulses are low and high is kept to create the digital current scan.

#### **4.9.6 Power Calculations**

The following power elements are calculated by the PICREF-3 system.

##### **Apparent Power**

RMS voltage and RMS current are simply multiplied to obtain apparent power.

##### **Power Factor**

Power factor is determined using a phase shift count and a look-up table. A firmware counter counts between rising edges of the voltage and current comparator outputs. The total phase shift count from two consecutive cycles in a one second measurement interval is divided by two to obtain an average phase shift count.

The average phase shift count is then applied to an algorithm which counts the number of 0.5 degree phase shift increments. The total number of increments is used to index into a cosine function offset look-up table. This table contains an integer offset which, when added to a long integer base number, yields a long integer cosine result of phase angles between 2° and 78.5°, which corresponds to power factors ranging from 0.999 to 0.199. A rolling two-sample average is used to obtain an average power factor reading.

The firmware counter is a 16-bit integer variable which has a phase shift per count resolution of better than 0.1° at all frequencies between 47 Hz and 63 Hz for the microcontroller oscillator operating at 8 MHz.

##### **True Power**

True power in Watts (W) is found by multiplying the apparent power by the power factor. This is based on the fundamental relationship,

$$P = V_{rms} \times I_{rms} \times \cos(\phi)$$

with  $\phi$  as the relative phase shift.

## Watt-Hours

A true power calculation is made once per second. Energy in joules is obtained by integrating true power in joules/sec to joules.

### 4.9.7 Firmware Details

The PIC16C924 firmware was written in C and consists of 24 separate modules, which are included on the CD accompanying this report. A flow diagram for the module most pertinent to this investigation, that of the measurement loop, is shown below in Figures 4.20.

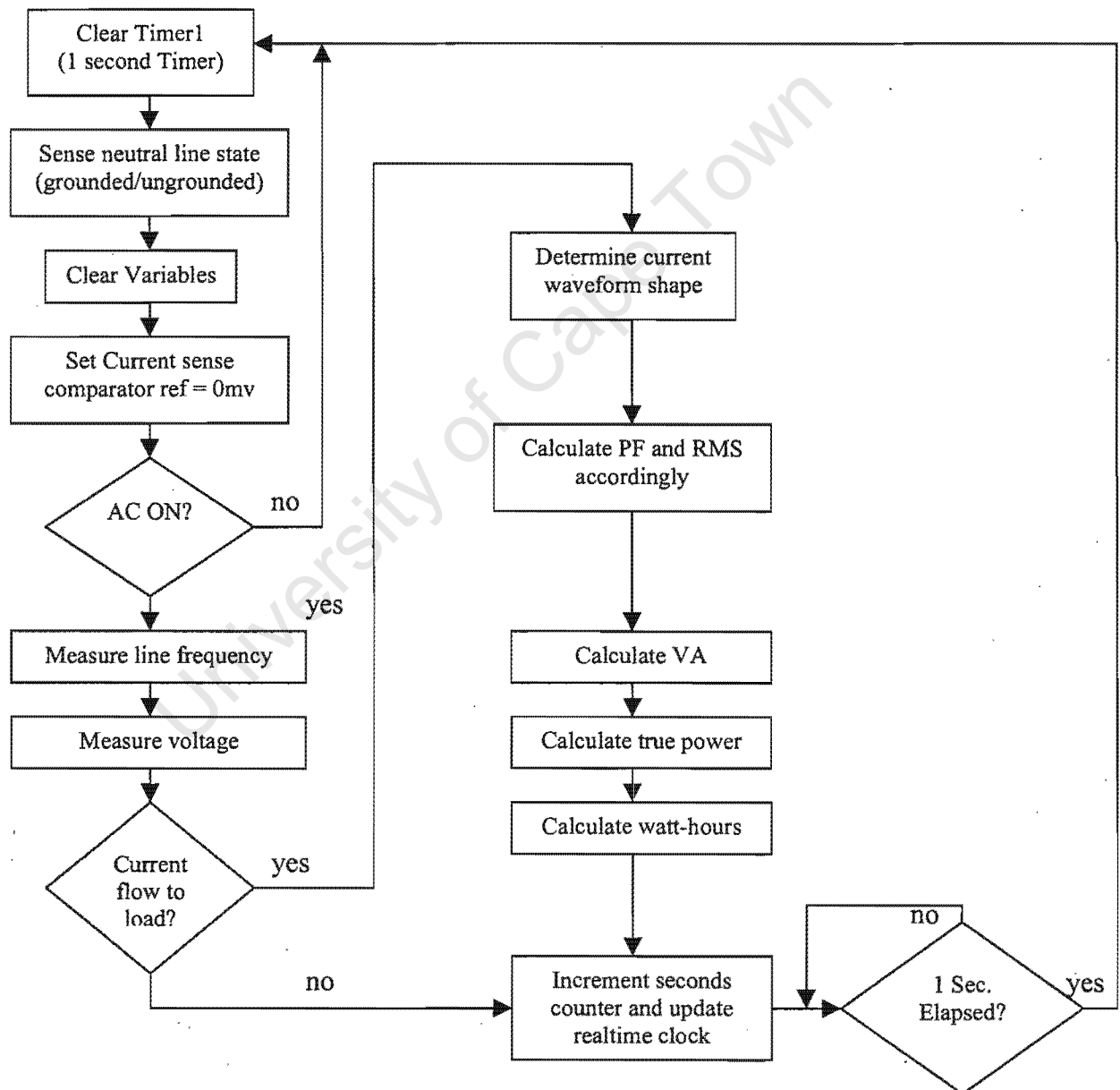


Figure 4.20: Measurement loop block diagram

#### **4.9.8 Test Results**

PICREF-3 Watt-Hour Meter measurements were compared to the measurements of a commercial watt-hour meter, the Yokogawa WT2010 Digital Power Meter, which has a basic measurement accuracy of 0.03%.

The system attained a power measurement accuracy at unity power factor for both a linear resistive load and a linear inductive load of 1.0% of reading plus 0.2% of full scale. Accuracy over a dynamic range is not provided. Also not provided are results for inputs with dynamic frequency and high frequency components. It is assumed these influences would decrease the overall accuracy.

University of Cape Town

## 4.10 Analog Devices AD7755 Metering ASIC

All data presented in the section was supplied courtesy of Analog Devices, Inc. in two PDF technical documents; (1) AD7755\_0.pdf, entitled “AD7755\* Energy Metering IC with Pulse Output” [3], and (2) AD7755\_T.pdf, entitled “AN-559 Preliminary Technical Data APPLICATION NOTE” [16]. Since this device is utilised as the energy measurement solution in the present generation CASHPOWER meters, a more in-depth study is presented. Furthermore, several other technical documents regarding the AD7755 standard operation testing results, testing results for susceptibility to electromagnetic interference and electrostatic discharge, implementation considerations and full schematic data for a test circuit were downloaded and are included on the CD accompanying this report for further review.

### 4.10.1 Overview

The below figure is an internal functional block diagram of the AD7755 device.

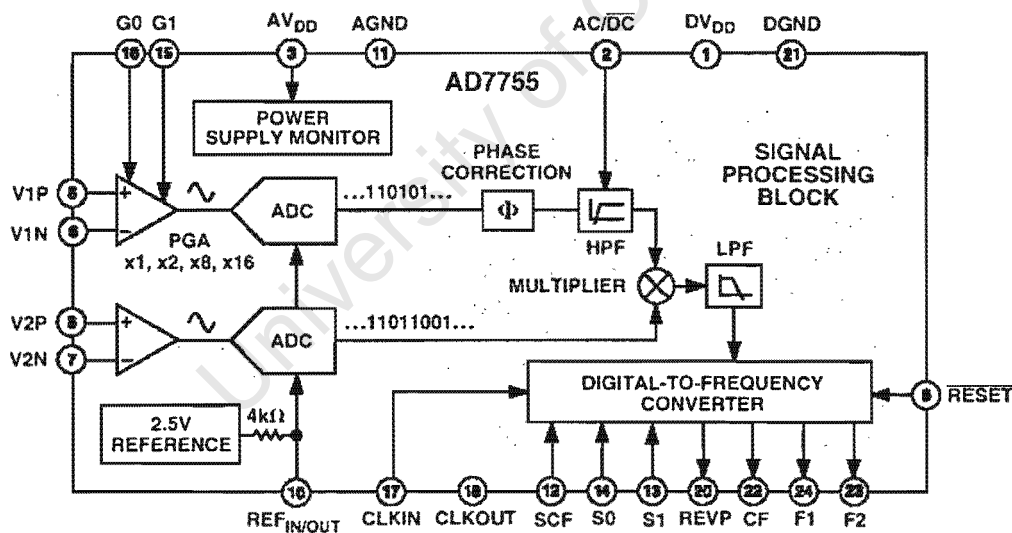


Figure 4.21 - Functional Block Diagram (AD7755)

The AD7755 Power-to-Frequency device is an Application Specific IC (ASIC) designed primarily for the measurement of single-phase electrical energy. The part specifications surpass the accuracy requirements as quoted in the IEC1036 standard. The only analog circuitry used in the AD7755 exists in the ADCs and reference circuit. All other signal processing (e.g. multiplication and filtering) is carried out in

the digital domain. This approach provides superior stability and accuracy over extremes in environmental conditions and over time.

The device accepts a pair of voltage input signals that represent the voltage and current of the power line. Internally, these signals are converted to the digital domain with oversampling ADCs. A fixed-function digital signal processor continuously multiplies the two signals, their product being proportional to instantaneous power, as described in Section 2.2. This digital signal is then low-pass filtered to provide real energy information. The input signals are derived either directly via a resistor network for the voltage and resistive shunt for the current, or through the use of voltage and current transformers respectively.

The AD7755 supplies average real power information as digital pulse frequencies on the two low frequency outputs (F1 and F2). These logic outputs may be used to drive an electromechanical counter directly or may be interfaced to an MCU. The Calibration Frequency (CF) logic output represents instantaneous real power information. This output is intended to be used for calibration purposes, but may also be interfaced to an MCU.

The AD7755 includes a power supply monitoring circuit, which ensures that it remains in a reset condition until the supply voltage reaches 4V. If the supply then falls below 4V, the AD7755 resets again and no pulses are issued on F1, F2 and CF.

Internal phase matching circuitry ensures that the voltage and current channels are phase matched. A High Pass Filter (HPF) may be activated in the current channel in to filter any DC component in the current signal. Furthermore, an internal no-load threshold ensures that the AD7755 does not exhibit any creep when no load is present.

Other features of the AD7755 include:

- Less than 0.1% Error Over a Dynamic Range of 500 to 1
- The Logic Output REVP Can Be Used to Indicate a Potential Miswiring or Negative Power

- Direct Drive for Electromechanical Counters and Two Phase Stepper Motors (F1 and F2)
- A PGA in the Current Channel Allows the Use of Small Values of Shunt and Burden Resistance
- Proprietary ADCs and DSP Provide High Accuracy over Large Variations in Environmental Conditions and Time
- ADC's accept bipolar signal inputs
- On-Chip Power Supply Monitoring
- On-Chip Creep Protection (No Load Threshold)
- On-Chip Reference  $2.5\text{ V} \pm 8\%$  ( $30\text{ ppm}/^{\circ}\text{C}$  Typical) with External Overdrive Capability
- Single 5 V Supply, Low Power ( $15\text{ }\mu\text{W}$  Typical)

#### 4.10.2 Signal Conditioning

The test circuit [16] supplied by Analog Devices as used in their evaluation for performance is presented below.

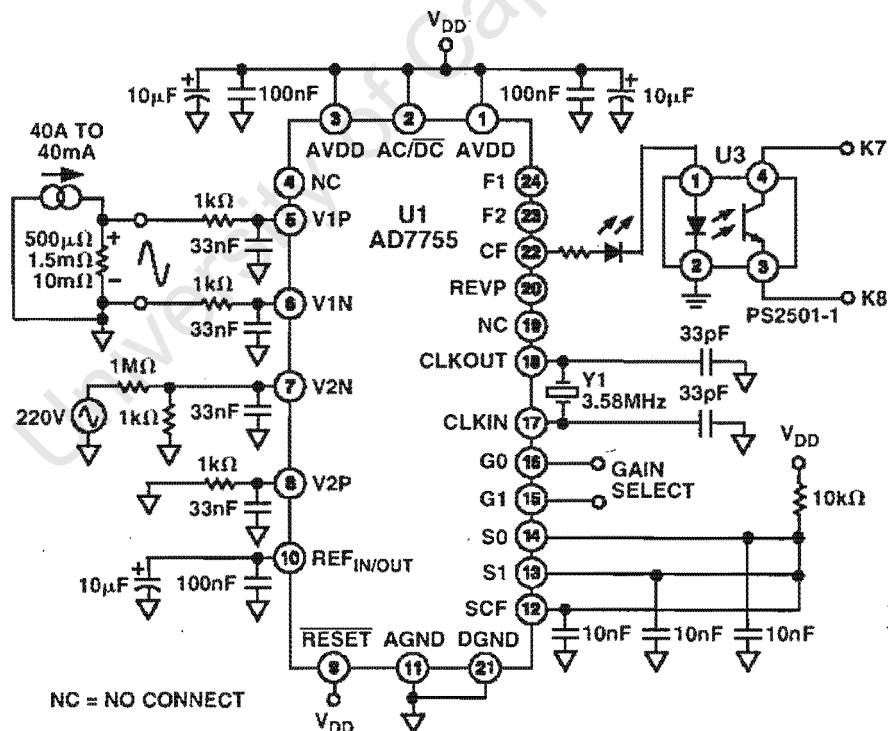


Figure 4.22: Test Circuit for AD7755

#### 4.10.2.1 External Analogue Signal Conditioning

Although the AD7755 operates off a unipolar supply, it accepts bipolar inputs on both the input channels. Both the current and voltage signal input channels, Channels 1 and 2 respectively, are fully differential. The input signals are obtained using either isolation transformers or resistive means; the voltage being scaled down using a network, and the current converted to a low-amplitude voltage signal via a shunt. These signals are read directly by the AD7755, eliminating the need for external amplifiers or filters.

As supplied by the manufacturers [18], a typical peripheral connection for Channel 1 is displayed in Figure 4.23.

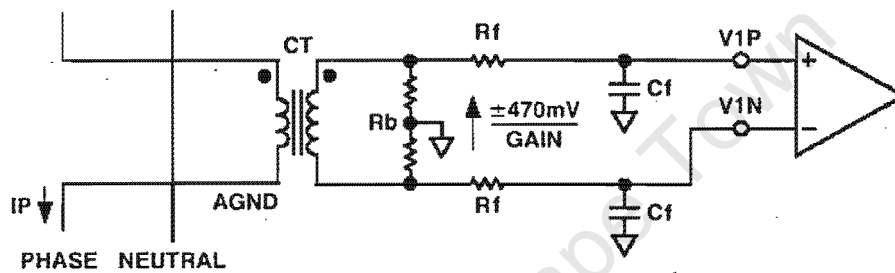


Figure 4.23: Typical Connection for AD7755 Channel 1

A current transformer (CT) is the current transducer selected for this example. The common mode voltage for this channel is analogue ground (AGND) and is derived by center tapping the burden resistor ( $R_b$ ) to AGND. This provides the complementary analogue input signals for the two Channel 1 inputs. The CT turns ratio and burden resistor are selected to give a peak differential voltage of  $\pm 470\text{mV}$  (divided by the input gain) at maximum load.

Figure 4.24 shows two typical connections for Channel V2 [18]. The first option uses a potential transformer (PT) to provide complete isolation from the mains voltage. In the second option the AD7755 is biased around the neutral wire, and a resistor divider is used to provide a voltage signal that is proportional to the line voltage. Adjusting the ratio of  $R_a$ ,  $R_b$  and  $V_R$  is also a convenient way of carrying out a gain calibration on the meter.

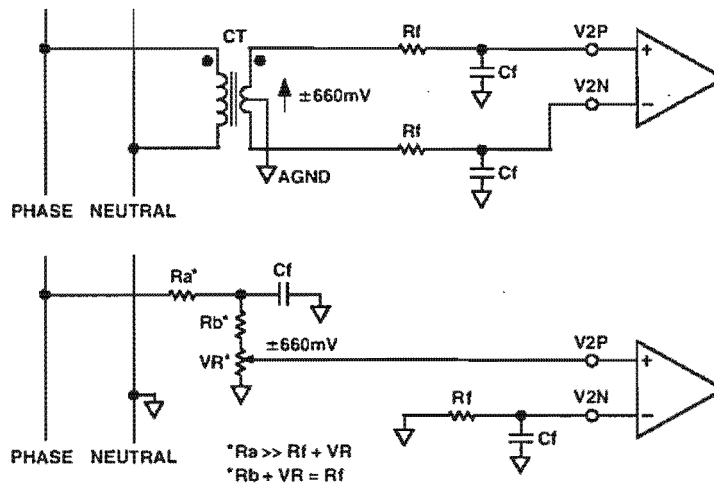


Figure 4.24: Typical Connections for AD7755 Channel 2

In all cases the filter capacitors  $C_f$  are included in an attempt to reduce high frequency noise on the input signals.

#### 4.10.2.2 Internal Analogue Signal Conditioning

The current input channel (Channel V1 in Figure 4.21) accepts a maximum peak differential signal of  $\pm 470\text{mV}$  (or  $330\text{mV}_{\text{RMS}}$  for a pure sinusoidal signal) and a maximum common mode signal of  $\pm 100\text{mV}$ , as shown below.

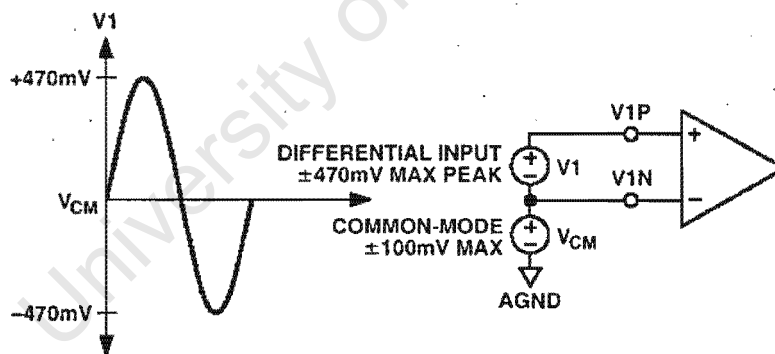


Figure 4.25: Maximum Signal Levels for AD7755 Channel 1, (Gain = 1)

This channel also has a programmable gain amplifier (PGA) with a user selectable gain of 1, 2, 8 or 16, set using two input pins G1 and G2. These gains facilitate easy transducer interfacing and the corresponding maximum permissible input signals are shown in Table 4.8.

Channel 1 Gain	Maximum Differential Signal
1	$\pm 470$ mV
2	$\pm 235$ mV
8	$\pm 60$ mV
16	$\pm 30$ mV

**Table 4.8 – Channel 1 Gain and corresponding Maximum input Signal**

Similarly, channel V2 is a fully differential voltage input. The maximum peak differential signal for this channel is  $\pm 660$  mV with a maximum common-mode voltage of 100mV, however best results are achieved using a common mode equal to AGND.

### **4.10.3 Analogue to Digital Conversion**

The use of two ADC's allows for concurrent sampling of the two input channels. These ADCs are 16-bit second order sigma-delta with an oversampling rate of 900kHz. The manufacturers have stipulated a maximum ADC offset error of  $\pm 15$ mV per channel. This corresponds to 3.19% of the full-scale current channel input signal and 2.27% of the full-scale voltage channel input signal. This further increases the need for a HPF in one of the two channels. Since the two ADCs operate independently of one another, the phase difference between the sampled signals is negligible and assumed to be zero.

The AD7755 features two input pins, S1 and S0, which allow the user to select one of four possible frequencies for analogue to digital conversion.

### **4.10.4 HPF and Offset Effects**

As depicted in Figure 4.21, Channel 1 contains a High Pass Filter (HPF). This is required since any offset in Channels 1 and 2 will contribute a DC component after multiplication [18]. Since this DC component is extracted by the LPF and used to generate the real power information, the offsets will have contributed a constant error to the real power calculation.

This problem is avoided by filtering through those low frequency components of just one channel, Channel 1, which are deemed DC. Thus the cutoff frequency for this filter is set to allow only varying components of the current signal to pass through.

The digital HPF in Channel 1 has an associated phase lead response. To offset this phase response and equalize the phase response between channels, a digital phase correction network is also placed in Channel 1. The phase correction network matches the phase to within  $\pm 0.1^\circ$  over a range of 45 Hz to 65 Hz and  $\pm 0.2^\circ$  over a range 40 Hz to 1 kHz. This ensures correct active harmonic power calculation even at low power factors.

#### 4.10.5 Power and Energy Calculation Methodology

As previously described the digital output of the low-pass filter after multiplication contains the real power information. However, since this LPF is not an ideal “brick wall” filter implementation, the output signal also contains attenuated components at the line frequency and its harmonics, i.e.,  $\cos(h\omega t)$  where  $h = 1, 2, 3, \dots$  etc.

The magnitude response of the filter is given by [18]:

$$|H(f)| = \frac{1}{1 + \left(\frac{f}{8.9\text{Hz}}\right)^2}$$

For a line frequency of 50 Hz this would give an attenuation of the  $2\omega$  (100 Hz) component of approximately  $-22$  dBs. The dominating harmonic will be at twice the line frequency, i.e.  $\cos(2\omega t)$  and this is due to the instantaneous power signal.

The instantaneous real power signal at the output of the CPF contains a significant amount of instantaneous power information. This signal is passed to the digital-to-frequency converter where it is integrated over time in order to produce an output frequency. The accumulation of this signal will suppress or average out any non-dc components in the instantaneous real power signal since the average value of a sinusoidal signal is zero. Hence the frequency generated by the AD7755 is proportional to the average real power [18].

The frequency output CF varies over time, even under steady load conditions. This frequency variation is primarily due to the  $\cos(2\omega t)$  component in the instantaneous real power signal. The output frequency on CF can be up to 2048 times higher than the frequency on F1 and F2. This higher output frequency is generated by accumulating the instantaneous real power signal over a much shorter time while converting it to a frequency. This shorter accumulation period means less averaging of the  $\cos(2\omega t)$  component.

As a consequence, some of this instantaneous power signal passes through the digital-to-frequency conversion, however where CF is used for calibration purposes, the frequency counter would normally average the frequency removing any ripple. If, however, CF is being used to measure energy, e.g., in a microprocessor-based application, the CF output must also be averaged to calculate power. Because the outputs F1 and F2 operate at a much lower frequency, a lot more averaging of the instantaneous real power signal is carried out in these signals than in the CF. The result is a greatly attenuated sinusoidal content and a virtually ripple-free frequency output.

The real power calculation also holds true for nonsinusoidal current and harmonic waveforms as well as for non-unity power factors [18]. For a mathematical analysis as to the validity of this, refer to Section 2.3.

#### **4.10.6 Outputs**

The AD7755 outputs real power information on the two outputs F1 and F2 in the form of active low pulses. The pulse rate at these outputs is relatively low, and is user settable ranging from 0 Hz to a maximum of 2.72 Hz. This means that the frequency at these outputs is generated from real power information accumulated over a relatively long period of time. The result is an output frequency that is proportional to the average real power.

The averaging of the real power signal is implicit to the digital-to-frequency conversion. The output pulse rate on CF can be up to 2048 times the pulse rate on F1

and F2, with a maximum frequency of 5.57kHz. Because of its relatively high pulse rate, the frequency at this logic output is proportional to the instantaneous real power. With much less averaging of the real power signal, the CF output is much more responsive to power fluctuations.

#### **4.10.7 System Errors**

Analog Devices [18] stipulate the device features an overall error of less than 0.1% over a dynamic range of 500 to 1. All system errors remain below this. Output error due to temperature fluctuations is attributed mainly to variations in the on-chip reference voltage, which features a typical temperature coefficient of 30 ppm/°C.

Furthermore, the AD7755 has been stringently tested and found to comply with *all* Class 1 IEC521 and IEC1036 specifications. A great deal of data regarding these tests has been published and is available on the CD accompanying this report.

#### **4.10.8 Operating Range**

Due to the ease of transducer interfacing, the AD7755 supports operation over a dynamic range of system parameters. The device complies with all IEC 1036 requirements when used on both 50 Hz and 60 Hz line frequencies.

Operation over the required current range also complies with these requirements. The AD7755 includes a “no load threshold” and “start-up current” feature that eliminates any creep effects in the meter. It is designed to issue a minimum output frequency when the line current drops below the preset threshold. Any load generating a frequency lower than this minimum will not cause a pulse to be issued on F1, F2 or CF. The minimum output frequency is given as 0.0014% of the full-scale output frequency for each of the F<sub>1-4</sub> frequency selections.

## Section 5: Rejected Techniques

The following techniques discussed in Section 4 were rejected as viable solutions for OCM implementation:

1. A fuzzy-based adaptive power metering system utilising a genetic algorithm (Section 4.5)
2. The Reduced Scan Principle method (Section 4.7)
3. Power measurement using rectified inputs (Section 4.8)
4. Power measurement using rms values of current and voltage (Section 4.6)
5. The PICREF-3 system – a Microchip MCU based approach (Section 4.9)

The justification for rejecting each method is briefly discussed in the ensuing sections. A brief summary of these details is provided in Table 5.1.

Measurement Technique	Major Failing Attributes
Fuzzy Based System	<ul style="list-style-type: none"> <li>• Processor overhead too large</li> <li>• Response too slow</li> </ul>
Reduced Scan Principle	<ul style="list-style-type: none"> <li>• Inherent error too large</li> <li>• Required sampling frequency too high – increased overhead</li> </ul>
Power Measurement using Rectified Inputs	<ul style="list-style-type: none"> <li>• Rectification error too large</li> <li>• Considerable signal processing required</li> <li>• Complex power evaluation algorithm</li> <li>• Large memory requirements</li> </ul>
Power Measurement using RMS Values	<ul style="list-style-type: none"> <li>• RMS transducers output error too large and also not financially viable</li> <li>• RMS-to-dc ASICs not financially viable</li> <li>• Processor overhead for RMS calculation method too large</li> <li>• Resolution of RMS calculation method too low</li> <li>• Power factor measurement unreliable and inaccurate for non-sinusoidal inputs</li> </ul>
PICREF – 3 System	<ul style="list-style-type: none"> <li>• Processor overhead too large</li> <li>• Power factor required</li> <li>• Signal conditioning requirements not financially viable</li> </ul>

**Table 5.1: Summary - rejected techniques**

## 5.1 A fuzzy based adaptive power metering system utilising a genetic algorithm [11]

Referring to Section 4.5, the described adaptive digital power metering algorithm is a dynamic metering solution which attempts to reduce required overhead in response to the aperiodic nature of the input waveforms. It remains unclear exactly how the extent of the aperiodic nature of the input waveforms is evaluated. If the decimation factor is high, the number of samples integrated into the metering algorithm is reduced. Thus for near-periodic waveforms, it appears as if the system reduces the accuracy of the measurement. This may prove accurate for periodic signals without higher frequency harmonics, however many examples of purely periodic waveforms containing large high frequency harmonics exist. If the effective sampling rate is reduced when measuring these waveforms, much of the high frequency components may be missed.

Many details regarding the algorithm are not provided by Kung *et al* [11], such as the processing time, refresh rate, resolution, response to input frequency fluctuations, etc. It is predicted that the processor overhead required to implement such a system would be considerably large - to the extent that the implementation of this system is not deemed viable as an OCM solution. Furthermore, the actual overhead saved may not justify the overhead required to implement the adaptive algorithm in an OCM application.

## 5.2 The Reduced Scan Principle method

The Reduced Scan Principle [15] is an interpolation method that alternately measures current and voltage at regular time intervals, and effectively multiplies each voltage sample by an interpolated current sample. From the results in Section 4.7 of the simulated input data, it is concluded that implementing this method will introduce inherent errors over and above those compensated for. Even with the compensation factor, the errors displayed in Table 4.6 are significant, especially at low sampling frequencies. It is predicted that these errors will increase significantly if the system is to be implemented on a hardware testbed.

Furthermore, the system is more susceptible to fluctuations in input frequency and sampling timing, as shown by Equation 4.7.7, than the standard computation technique. This was proved using the first example of Table 4.6 with a line frequency of 55Hz as opposed to 50Hz. The standard power calculation produced an error of 0.59% whilst the reduced scan produced an error increase, from that shown in the table of 0.5%, to 5.52%. Although this is greatly reduced at higher sampling frequencies, the increase in error is still significant and would very probably extend beyond the limits stipulated by IEC1036 regulations.

The reduction in processor overhead obtained by the reduced scan is the primary advantage of the system. However, the need to compensate for the phase shift error  $e$  by calibration with the constant  $c$  effectively reduces this.

Furthermore, the actual overhead saved is minimal considering the Motorola MCU's ADC is capable of convergence in just 12 to 16s. This amounts to a phase shift of less than  $0.29^\circ$  for a 50Hz system. Non-concurrent sampling tests were performed using the same waveform on both input channels. The channels were sampled using the MCU's multiplexed 8-bit ADC, and 149 out of 160 pairs of samples of the same 50Hz input sine wave corresponded. The remaining 11 were a maximum of 1 out of 255 (the quantization error) difference and are attributed to quantization error, rather than a difference due to phase shifting. It is noted that the Hitachi 10-bit ADC is capable of detecting smaller phase differences through its increased resolution, however the overall increase in error arising from a  $0.29^\circ$  phase error is well within acceptable limits.

The 50% reduction in the number of multiplication operations also reduces the overhead, but only minimally so. This is especially true for the Motorola where the samples exist as 8-bit bytes and may be multiplied in a simple routine requiring a maximum of 21 instructions (for signed operation) corresponding to 84 bus clock cycles to compute. This amounts to around  $20\mu\text{s}$  and may easily be conducted during the sampling period. The addition operation used as the alternative to this multiplication in the reduced scan method would require exactly the same memory space, but around half the processor time. The 10-bit resolution of the Hitachi ADC

converter means that two 10-bit words must be multiplied. Even if these were treated as 16-bit words, the time required to multiply them would be less than 80 $\mu$ s, or around 80 clock cycles.

In light of the above it is concluded that, although this method may be viable as an OCM solution using a high sampling frequency, the error introduced would be substantially higher than that of the standard method. Furthermore, the high sampling frequency required to minimize the inherent error may actually *increase* required overhead beyond that of the standard power measurement technique.

### 5.3 Power measurement using rectified inputs

A power measurement system using rectified inputs may allow for higher ADC converter resolution, as discussed in Section 4.8. However, considering the poor test results displayed in Table 4.7, and in light of the failed attempts to develop an accurate rectification process that remains cost-effective, the proposed method was not investigated further. Unless a more applicable means of rectifying the signals is obtained, the rectification output error is too large for the system to be viable as an OCM solution.

### 5.4 Power measurement using rms values of current and voltage

Several methods of using the rms values of the current and voltage to calculate the real power were investigated and are discussed in Section 4.6. None of these methods were found to be viable as an OCM solution due to the following reasons:

- The use of RMS-to-DC producing ASICs does not offer any financial benefit over the presently employed measurement method
- The use of RMS-to-DC transducers is not viable due to their relatively high cost and high inaccuracies
- The on-board RMS calculation method requires a large processor overhead
- The RMS calculation method is considerably more complex and time consuming than its discrete sample-utilizing counterpart

- The power factor calculation described in Section 4.6.3 uses interpolation, possibly introducing error through inaccurate conjecture in the presence of a large 3<sup>rd</sup> harmonic
- It is also highly susceptible to fluctuations in line frequency, requiring a line frequency close to 50Hz to remain accurate

Several of these factors are discussed in more detail below.

#### **5.4.1 On-board RMS calculation using discrete samples**

A considerable effort was made to develop a viable RMS calculating algorithm to compute the RMS values of both input waveforms. The method described in Section 4.6.2.3 was tested on the Motorola MCU using simulated inputs and is proven to work to an acceptable accuracy. However, this method requires a substantial amount of processor time. A reduction in overhead may be possible using a look-up table for the square-root function, however as mentioned before, the number of entries required to implement this would probably not be viable.

#### **5.4.2 Power Factor calculation**

The power factor calculation method described in Section 4.6.3 was also found to be not viable. This method was developed with the specific intention of minimising false zero-crossing detection and increasing output resolution.

Since the RMS calculating method described above may offer a viable solution, the viability of the overall system relies on the ability to measure the power factor accurately. Consequently, a thorough justification for rejecting the developed power factor measurement technique is provided. The analyses provided below are based on the results of the Motorola MCU-based implementation of this method with simulated inputs.

- *Poor resolution*

Even though the power factor measurement methodology is legitimate, the error induced with each calculation is considerably large. This is mainly due to the fact that each time a number is divided, the remainder is resolved, but is not utilized in the ensuing multiplication, subtraction and addition procedures. More advanced

math routines with higher resolutions may be developed at the cost of considerably more processor overhead.

- *Susceptibility to dynamic line frequency*

The power factor calculation relies heavily on the assumption that the line frequency remains at a constant 50Hz. Should this method of measurement be implemented, the line frequency will have to be tested on a regular basis, perhaps just before each phase difference calculation. The ensuing calculations will then automatically be adjusted for the measured line frequency. This is also performed at the cost of more processor overhead though.

- *Large processor overhead*

The processor overhead required for the described power factor calculation is considerably large. Since both the current and voltage channels must be sampled and tested in order to attain the value  $\Delta n_p$ , neither one may be used to continue the line sampling while the power factor is being determined. Although this only constitutes a maximum time period of one mains cycle, the RMS calculations utilize the same math functions as the power factor calculations and multiplexing of these operations is therefore impossible. Thus the power factor calculation must be completed before the RMS calculations may begin.

- *Zero-crossing detection highly dependant on wave shape*

Although three separate algorithms were developed and tested to detect the correct zero-crossing point, all failed to accommodate waveforms with a strong 3<sup>rd</sup> harmonic content consistently. This reason for this is illustrated in Figure 5.1.

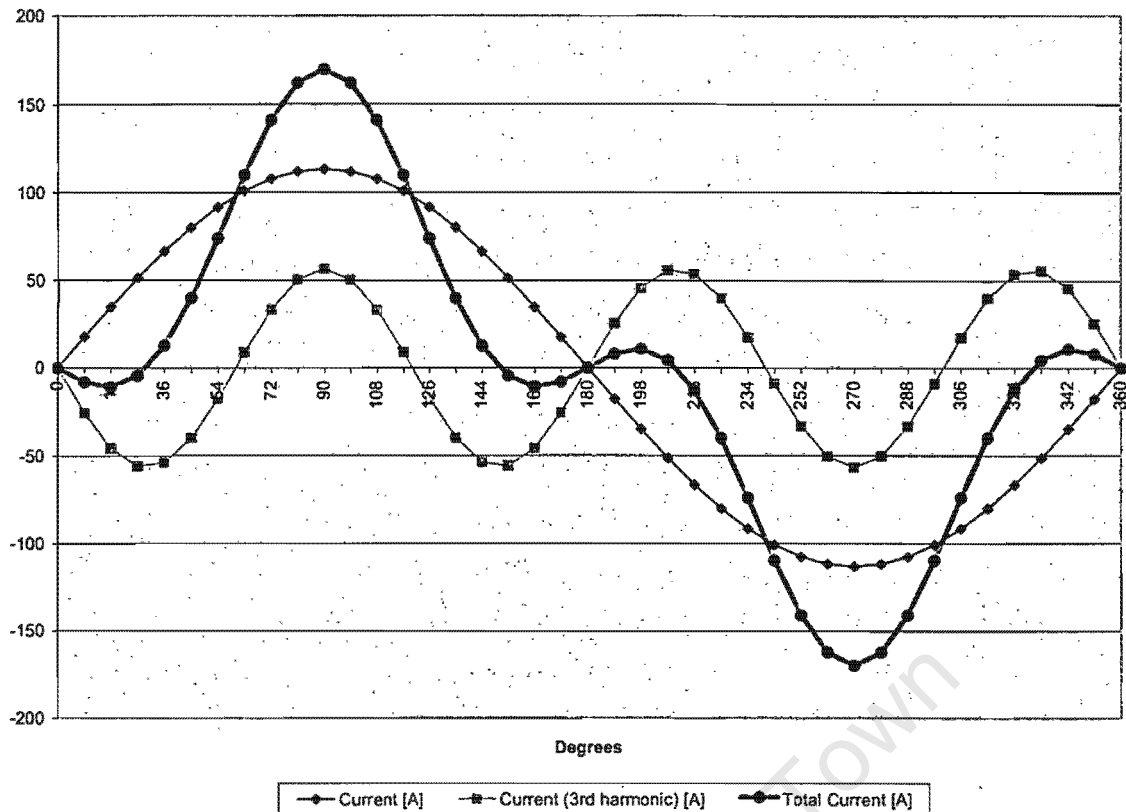


Figure 5.1: Current Signal with strong 3<sup>rd</sup> harmonic content

The 3<sup>rd</sup> harmonic depicted above has been accentuated for the purpose of clarity and such a strong 3<sup>rd</sup> harmonic content would very rarely occur.

As can be seen, the waveform sampled (depicted as Total Current above) crosses the zero-mark three times per half cycle. When determining the value of  $\Delta n_p$ , the starting point for the positive-verses-negative count begins at the sample closest to a negative-to-positive transition. For the example above, if this begins at the sample taken at  $36^0$  then correct zero-crossing detection will occur. However, if the count starts at the sample taken at  $180^0$  the zero crossing detection requires a further one and a half cycles to place the crossing point. This crossing point would then be offset by an undetermined amount which may lead to an incorrect phase shift calculation when compared to the second channel's crossing point. Thus a lagging power factor may be erroneously measured when in fact a leading one is present.

- *Input Offset on one channel provides erroneous phase shift calculation*

Should the two input channels contain different offsets, the phase shift calculation will be incorrect. This is depicted in Figure 5.2.

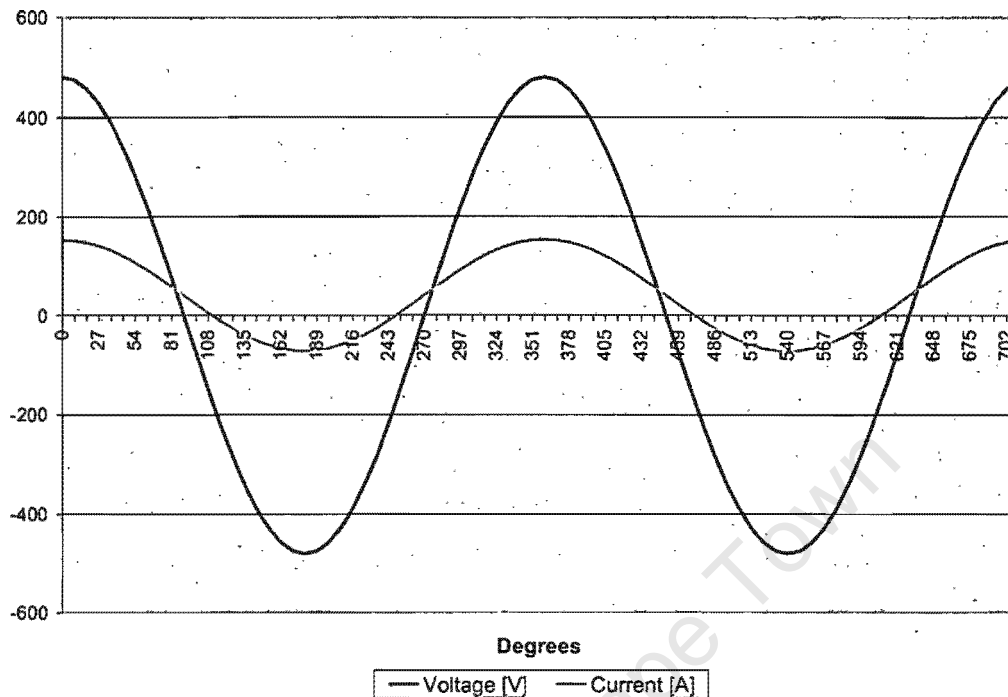


Figure 5.2: *Input waveforms with DC offset in current channel*

These two inputs are in fact in phase, but due to the offset in the current channel, a phase difference will be detected if the zero-crossings are used. In this case, detecting the peak of each signal would achieve better results, however the problems stated in the previous bulleted point above would arise for waveforms with strong harmonic content. One solution would be to high-pass filter the inputs before testing. Unfortunately, not only would this slow the power factor calculation process, but would require considerable RAM and processor time over and above that required for the RMS calculations.

In light of these points, the power factor calculation method is not viable as part of an OCM solution.

## 5.5 The PICREF-3 system – a Microchip MCU based approach

The PICREF-3 [14] system discussed in Section 4.9 utilizes a mixed signal MCU, the PIC16C924, as the core of a power metering solution. It was decided that an OCM solution based on this system is not viable. This mostly due to the large processor overhead requirements and the necessity to measure the power factor (refer to Section 5.4.4). Several other attributes also justify this decision.

The stipulated system measures the current during the positive cycle only and assumes waveform symmetry. This does not conform to IEC1036 regulations, since a load with half-wave rectification, such as a simple motor speed controller, would not be detected.

The analog signal conditioning is also considerably complex. The required current sense circuitry and voltage sense circuitry would probably increase manufacturing costs beyond those of the present metering solution. The use of PWM to calibrate the voltage and current sense circuitry would further increase processor overhead as well.

In light of the above it was decided that the required memory, processor overhead, and signal conditioning circuitry are too large for the system to be viable as an OCM solution.

## Section 6: Techniques considered further

Several aspects of the following methods from Section 4 were considered to be significant in the development of a viable OCM solution:

1. Analog Devices AD7755 Metering ASIC methodology (Section 4.10)
2. “Digital Wattmeter Using a Sampling Method” (Section 4.1)
3. “Digital Power Metering Manifold” (Section 4.3)
4. “Design of a Microprocessor-Based Digital Wattmeter” (Section 4.4)
5. “Digital Sampling Laboratory Wattmeter” (Section 4.2)

The particular aspects of these methods considered in the development of an OCM solution are summarised in the table on the next page. Some of these are also discussed in further detail in the ensuing sections.

Measurement technique	Applicable aspect considered	Reason for inclusion
AD7755 Methodology	General	<ul style="list-style-type: none"> <li>device currently used as measurement solution in CASHPOWER units</li> <li>overall operation conforms to IEC 1036 regulations</li> <li>calculates only the required quantities – instantaneous and average real power</li> </ul>
	Peripherals	<ul style="list-style-type: none"> <li>required analog signal conditioning viable</li> <li>compensates for phase shift, dc offsets</li> <li>transducer interface viable</li> </ul>
	Data acquisition method	<ul style="list-style-type: none"> <li>discrete sampling of inputs – only viable choice as concluded in Section 5.4</li> </ul>
	Power calculation method	<ul style="list-style-type: none"> <li>robust, proven and accurate</li> <li>easily implemented on MCU</li> <li>minimal processor overhead</li> </ul>
	Outputs	<ul style="list-style-type: none"> <li>output frequency as required for OCM</li> <li>calibration frequency also required for OCM calibration</li> </ul>
“Digital wattmeter using a Sampling Method”	General	<ul style="list-style-type: none"> <li>operation similar to AD7755</li> <li>overall accuracy complies with IEC1036</li> <li>calculates required quantity</li> </ul>
	Timing pulse generator	<ul style="list-style-type: none"> <li>possible method of line frequency compensation for OCM</li> </ul>
	Calibration voltage measurement	<ul style="list-style-type: none"> <li>possible method to compensate voltage drift due to temperature fluctuations</li> </ul>
“Digital Power metering Manifold”	Power metering algorithm	<ul style="list-style-type: none"> <li>inherent advantage of dc component compensation</li> <li>easily implemented on MCU</li> <li>minimal processor overhead</li> </ul>
“Design of a Microprocessor-Based Digital Wattmeter”	General	<ul style="list-style-type: none"> <li>methodology similar to AD7755</li> <li>developed system similar to OCM project</li> </ul>
	Measurement process	<ul style="list-style-type: none"> <li>stipulated algorithmic process similar to that required for OCM</li> <li>compensates for line frequency</li> </ul>
“Digital Sampling Laboratory Wattmeter”	Power measurement	<ul style="list-style-type: none"> <li>based on the paper mentioned above “Digital Wattmeter using a sampling method”</li> </ul>
	Signal conditioning	<ul style="list-style-type: none"> <li>dynamic input gain according to input</li> <li>(analog) phase matching described</li> </ul>
	Sampling timing	<ul style="list-style-type: none"> <li>compensates for line frequency</li> <li>methodology may be possible for OCM</li> </ul>

Table 6.1: Summary - techniques considered further

## 6.1 Considerations for OCM Implementation Based on the AD7755

The AD7755 Power-to-Frequency device is an Application Specific IC (ASIC) designed primarily for the measuring of single-phase electrical energy. Due to the relative simplicity of the real power calculation, and considering that aspects of the power system such as phase delay, power factor and rms voltage and current need not be measured, the viability of implementing this methodology on a Microcontroller based system is excellent. For this reason, the employed methodology was closely examined.

### 6.1.1 *Peripherals*

The methodology used to calculate the real power described in Section 10 requires that both the current and voltage signals are bipolar. Since the AD7755 accepts bipolar inputs, it is assumed that the device contains an internal charge-pump or similar device in order to obtain dual reference rails for the input gain amplifiers. Furthermore, the internal ADCs are assumed to operate with separate digital and analogue ground rails. The ADCs on both the Motorola MCU and the Hitachi MCU are unable to handle bipolar inputs and neither of the MCUs feature internal gain amplifiers to scale the analogue input signals.

Therefore, if the AD7755's methodology were to be mimicked, the bipolar voltage and current signals from the respective transducers would have to be amplified and converted to unipolar signals using external components. There exist two possible methods for accomplishing this: (1) to use separate digital and analog grounds, with the analog ground offset by + 2.5 V, to (2) to offset the input signals so that the entire waveform lies within the (positive) Microcontroller ADC reference range and use a single ground.

In order to achieve either of these, an external bipolar power supply is required. Possible methods for achieving this are briefly discussed below. Aspects such as the meter's power supply including the design, stability, reliability, cost, etc, are not within the scope of this report.

The need for a bipolar supply eliminates the possibility of utilizing the Capacitive-coupled power supply design used in the present Cashpower units. Instead, a modified version of this design must be developed to provide a bipolar supply. Other options include using an external charge pump or a center-tapped transformer. All of these methods introduce noise into both the analogue and digital modules of the system.

The use of a transformer provides a source of magnetic flux and may introduce harmonic distortions in the meter power supply. Furthermore, the power requirements of the meter itself would be slightly higher due to increased power losses. Also, production costs would probably increase due to the relatively high cost of the component and the added space requirement. However complete electrical isolation between the meter and the mains supply would be obtained, ensuring safer operation and a reduced risk of electrical shock.

The use of a charge pump requires less board space than a transformer, but high frequency noise is introduced through the switching action of the device. Most of these devices obtain excellent signal-to-noise ratios, attenuating the noise to acceptable levels.

The voltage and current signals may be obtained as recommended by Analog Devices for the AD7755 [16]. The scaling of these signals would have to be conducted using external differential amplifiers to ensure that the full-scale line voltage and current input signals correspond to the maximum specified line voltage and load current.

### **6.1.2 Sampling – Analogue to Digital Conversion**

The AD7755 contains two separate 16-bit ADCs which allow for concurrent sampling of both input signals. The Motorola and Hitachi MCUs feature just one ADC that multiplexes each of the respective inputs, introducing a slight phase error between the sampled input signals. The extent of this phase error is mostly dependent on the conversion time of the ADC. The maximum sampling rate of the Motorola ADC is provided as 1MHz for operation within the stipulated error limits. At this speed consecutive samples occur  $1\mu\text{s}$  apart. The phase error introduced is computed in Section 3 as  $0.018^\circ$ , corresponding to just 0.005% of the full cycle ( $360^\circ$ ). It was

predicted that due to the relatively low resolution of the Motorola ADC, this would not be detected. Later testing confirmed this (refer to Section 10.1.2).

### **6.1.3 Internal Signal Conditioning**

The HPF in Channel 1 and the LPF used to extract real power from the instantaneous power in the AD7755 may both prove viable for implementation on a MCU.

The phase lead associated with a HPF may introduce considerable phase error between the two signal channels. A possible method of overcoming this is to introduce an All-Pass filter with a similar associated phase shift in the voltage channel. An easier method of phase correction would be to time delay the influx of the high-passed digital current signal values into the multiplication process. As an example, the voltage sample of time  $t$  would be multiplied with the current sample of time  $t+1$  to provide correct instantaneous power.

Preliminary research on several digital HPF algorithms indicated the system would obtain a relatively slow response time. Furthermore, signals containing severe offsets may exhibit clipping due to these offsets being multiplied by the first and secondary gains. In light of these, it was decided to remove signal offsets using analog circuitry, thereby eliminating the need for a digital HPF (refer to Section 8.4.1.1).

### **6.1.4 Power Calculation Methodology**

The simplicity of the power calculation results in low overhead requirements and a robustness that cannot be found using analogue means. The multiplication process and low-pass filtering action used to extract the real power on the AD 7755 are relatively easy to implement. This method was used in the final OCM solution presented in Section 8.

### **6.1.5 Outputs**

The AD7755 outputs a digital pulse frequency directly proportional to energy expenditure. The digital pulses are generated by low-pass filtering the instantaneous power information and accumulating the result until it reaches a stipulated threshold.

When this threshold is exceeded an output pulse is generated. The threshold constant is dependent on the sampling frequency, input voltage and current gains, ADC resolutions and the quantity of energy determined by the Meter Constant corresponding to one pulse. This method of generating the output pulses was utilised in the final OCM solution and is discussed further in Section 8.

## 6.2 “Digital Wattmeter Using a Sampling Method”

Turgel [4] presents a method of measuring average power by sampling instantaneous voltage and currents from which the results are computed by numerical integration. Intercomparisons with an external power measurement devices agreed within 0.02% from dc up to 1kHz, with the possible exception of zero power factor.

It was found that several power and energy measuring solutions utilise methods strongly based on those described by Turgel, most notably that of the AD7755 previously discussed.

An important aspect of Turgel's design is that of offset sampling. As mentioned in Section 4.1.5, sampling begins with a "zero reading" [4], where the inputs are connected via a small resistor to ground. This reading is then used as an offset correction reference to subsequent data, and provides an ideal method to compensate for the effective offset introduced through separate analogue and digital grounds. This aspect of Turgel's design was incorporated in the final OCM solution presented in Section 8.

Several of Turgel's other design considerations also influenced the final OCM design, and are discussed further in the ensuing three subsections.

### 6.2.1 *Analogue Signal Conditioning*

Turgel does not include information regarding aspects of the input signals such as signal-to-noise ratios, influence of non-sinusoidal signals and influence of half-wave rectified current signals in the paper. All of these would certainly increase the overall system error.

The use of signal instrumentation amplifiers with high common-mode voltage immunity for preamplification in the described system decreases susceptibility to common-mode noise and offsets. The trade off, however, is at an increase in component cost, though many low-cost solutions offering moderate common-mode rejection ratios are available today.

Although the phase shift and ratio errors introduced by these devices is neglected, they may influence the overall accuracy if implemented. This influence may be negligible, however, as long as the *relative* phase shift between the channels remains minimal [16].

The choice of shunt in the described system allows a current signal of 0.5V rms to be produced at a full-scale load of 5A. For the purpose of OCM, the use of a shunt to produce this magnitude of signal would not be viable considering a load current of up to 80A is required to be measured. The power dissipation of the device would be up to

$$P_{shunt} = V_{rms} I_{rms} = 0.5 \times 80 = 40W$$

at unity power factor. This is unacceptable according to IEC1036 specifications [17]. Two alternatives exist though; (1) use a current signal transformer with a relatively low input/output ratio, or (2) use a lower impedance shunt and increase the preamplifier gain. Refer to Section 7 of this report for a further analysis of these alternatives.

### 6.2.2 Sampling Timing and Resolution

Although Turgel used external S/H circuits and A/D converters, these functions may easily be conducted by an MCU with on-board A/D converters. However, the sampling timing and resolution requirements of the described method may not be possible to implement without introducing external components, increasing costs.

The nominal refresh time of around 1 second indicates that, with the stipulated line frequency of 60Hz and the total number of samples ( $N = 512$ ) being taken over a single cycle, the power measurement is estimated for 59 cycles out of 60. Thus the system only measures the actual power 1.67% of the time and thus does not conform

to IEC1036 specifications. However, this slow refresh rate may be attributed to the use of a relatively slow processor. It should be noted that massive improvements have been achieved in the operating speeds of modern processors compared with those available in 1974.

The described system [4] uses a PLL with a binary frequency divider to generate the pulses used to initiate the S/H and A/D converters. Although this allows the system to be frequency independent, the implementation of the PLL would increase costs. The 15-bit A/D converters provide substantially higher resolution compared with the 10-bit and 8-bits of the Hitachi and Motorola A/D converters respectively.

### 6.2.3 Power Computation Algorithm

The computation of Equation 4.1.3 is performed by the processor at a rate of about once a second. As mentioned above, modern processors such as those of the Hitachi and Motorola MCUs operate at speeds that are able to conduct this computation in acceptable times (in the order of several hundred microseconds for large values of  $N$ ). Thus implementation of Equation 4.1.3 may well be viable as power calculation. The required arithmetic operands, multiplication, division, addition and subtraction, are easily implemented on an 8bit MCU, even for 16, 24 and 32-bit words.

In summary, the described algorithm is sound, robust and simple, and is currently used in several power and energy measurement devices available.

## 6.3 “Digital Power Metering Manifold”

The power metering algorithm described by Younget *al* [9] in their paper has a solid theoretical basis as proved in Section 2.2 of this report and in the paper by Turgel [4] discussed above.

The described algorithm (Equation 4.3.3) is primarily based on that proposed by Turgel, with a slight modification. The inclusion of the latter term,  $\bar{v}_i$ , of Equation 4.3.3 is an innovative approach that ensures any offset (dc) in the inputs signals do not contribute to the measured power. This is used as an alternative to implementing a

HPF in one or both of the input channels. An inherent advantage of this method is that the voltage and current samples, as well as the instantaneous power, may be summed itinerant, that is, during the sampling sequence. Furthermore, the implementation of a HPF in one channel would introduce an associated phase shift introducing the need for phase-shift compensation in the other. This is not the case here.

In spite of these advantages, this method was not used as the final OCM solution for three main reasons ; (1) any offsets in the input signals are multiplied by the pre-ADC amplifiers and often offset the resulting waveform so that it becomes unipolar, (2) the resulting waveform often exhibits clipping for the same reason, and (3) the increased processor overhead does not justify the increased accuracy, considering that a suitable means of removing the offsets using simple analogue means was found (refer to Section 8).

#### **6.4 “Digital Sampling Laboratory Wattmeter”**

The digital sampling wattmeter presented by Corney [6] (Section 4.2) is also based on the concept pioneered by Turgel [4].

Referring to Section 4.2, all of the signal conditioning for this system is conducted externally in an analogue domain and as shown in Figure 4.3, is extensive. The use of multiple input ranges for the current and voltage channels are not required for OCM as these remain constant for the stipulated utility meters. However, the use of the 9 solid-state switches for the voltage and 12 for the current range settings presents a possible solution to increasing the effective input range. A two-stage input gain design was developed for the final OCM solution presented in Section 8 loosely based on the multiple-input gain idea proposed by Corney [6].

## Section 7: Preliminary OCM Design Considerations

Several design aspects concerning the final OCM solution are presented in this section. In addition, several of the tradeoffs encountered whilst developing the design are also discussed.

Based on the research presented so far, it has been concluded that in order for OCM to be viable the following constraints must be met:

- The system must adhere to all relevant specifications as stipulated by IEC1036 regulations for a minimum of a class 2 metering unit (preferably class 1).
- The system must be financially viable – the present metering solution, including all signal conditioning components and manufacturing costs, amounts to a total of ± R12 per utility unit. The OCM solution, including all peripherals except the current transducer, should ideally cost less than this.
- The system must be robust and exhibit long term stability and reliability.

Any system not conforming to any of these constraints may automatically be excluded as a viable OCM solution.

In conjunction with these, the system should ideally adhere to the following constraints as well:

- The system must have a proven theoretical basis.
- All components used must be commercially available.
- All system outputs must be available in a form compatible with the present utility units.

During the course of this investigation it has further been concluded that the following parameters should be optimized as far as possible: ( in order of significance)

- cost

- processor overhead
- memory requirements
- number of external components (relates to overall cost)
- ease of calibration (relates to overall cost)

## 7.1 System Tradeoffs

It is important to note that, as with any design, there are tradeoffs between criteria that must be optimized. The three most fundamental tradeoffs regarding the OCM design are:

- **Cost versus Accuracy**

This is usually the fundamental tradeoff in any measurement system. Increasing the system accuracy, either through hardware or software, effectively results in an increase in production costs. Increasing the signal processing accuracy by using higher precision components, for example, results in an increase in component costs. Increasing the accuracy of software-based arithmetic operands, for example, increases the need for a more powerful processor, also resulting in an increase in component costs. The inverse is also true.

- **Accuracy versus Processor Overhead**

Increasing the precision or resolution of algorithmic modules increases the required overhead. Moreover, increasing the number of algorithmic modules or their complexity increases the required processor overhead.

- **Hardware optimization versus Firmware optimization**

This effectively resolves to a tradeoff between hardware intensity and processor overhead. A reduction in the required hardware results in an increase in the complexity of the firmware. This effectively results in an increased processor overhead. For example, if a hardware HPF is removed and implemented onboard, the firmware required to implement this would increase processor requirements. The inverse is also true.

## 7.2 Transducer Considerations

In order to develop the final OCM algorithm, it was important to decide what transducers would be used to attain the input signals. The primary function of a "transducer" or a "sensor" is as an energy converter<sup>[18]</sup>, converting information from one physical parameter, in this case a load current or line voltage, to another; usually a form that can be interfaced to the measurement system. In this report, the term "transducer" refers to a device that outputs an electrical signal, either a current or voltage, representing an input signal, also either voltage or current.

Four basic transducer types (i.e. those that do not perform any type of operation on the acquired signal) and two more complex types were considered. These are

- Current transformers
- Current shunts
- Voltage transformers
- Resistive network as a voltage divider
- RMS value producing current and voltage transducers

The main influences relating to the transducers described in this section are:

- External noise (EMI, resonance, et cetera)
- Dynamic frequency response
- Line harmonics
- Overvoltages and currents
- Voltage and current spikes
- Dynamic physical conditions of temperature and humidity

Table 7.1 on the ensuing page contains a summary of the key issues regarding these devices.

Transducer	Signal Processing Considerations	Accuracy	Main Sources of Error	Cost	Overall Viability Rating
Current Transformer	<ul style="list-style-type: none"> <li>Low impedance secondary load required</li> <li>Low output voltage – requires active gain</li> <li>Poor signal-to-noise ratio</li> <li>Provides isolation</li> </ul>	<ul style="list-style-type: none"> <li>1% to 3% for commercial transformers</li> <li>0.3% to 2.4% for ANSI class metering transformers</li> </ul>	<ul style="list-style-type: none"> <li>Secondary burden variance</li> <li>Output loading (high impedance)</li> <li>EMI</li> <li>Dynamic ratio error</li> <li>phase shift</li> </ul>	Medium to High	Medium
Current Shunt	<ul style="list-style-type: none"> <li>Very low impedance output-requires large input impedance</li> <li>No isolation provided</li> <li>Low output voltage – requires active gain</li> <li>Poor signal-to-noise ratio</li> </ul>	<ul style="list-style-type: none"> <li>Dynamic error dependant on temperature coefficient</li> <li>Generally better than required for present application</li> </ul>	<ul style="list-style-type: none"> <li>Shunt loading</li> <li>Temperature drift</li> <li>Induced noise</li> </ul>	Medium	Good
Voltage Transformer	<ul style="list-style-type: none"> <li>High impedance secondary load required</li> <li>Difficult to calibrate without external amplifier</li> <li>High voltage output – good signal-to-noise ratio</li> <li>Provides isolation</li> </ul>	<ul style="list-style-type: none"> <li>1% (best) for commercial transformers</li> <li>better than 1% for high-precision transformers</li> </ul>	<ul style="list-style-type: none"> <li>Secondary burden variance</li> <li>Output loading (low impedance)</li> <li>EMI</li> <li>Dynamic ratio error</li> <li>Phase error</li> </ul>	Medium to High	Medium
Voltage Divider	<ul style="list-style-type: none"> <li>High precision resistors required</li> <li>Proven method (currently used in Cashpower units)</li> <li>High output voltage possible – does not require active gain</li> <li>Good signal-to-noise ratio</li> <li>Easily calibrated</li> <li>Small physical size</li> <li>No isolation provided</li> </ul>	<ul style="list-style-type: none"> <li>Dynamic error dependant on temperature coefficient</li> <li>Typical dynamic error of between <math>200\mu\Omega/K</math> and <math>500\mu\Omega/K</math></li> <li>Tolerance may be compensated for during calibration</li> </ul>	<ul style="list-style-type: none"> <li>Loading of output (low impedance)</li> <li>Induced noise</li> <li>Minimal phase error (may be neglected)</li> </ul>	Low	Excellent
RMS Producing Transducers	<ul style="list-style-type: none"> <li>Provide rms information</li> <li>Direct interfacing with MCU possible</li> <li>No further analog signal processing required</li> </ul>	<ul style="list-style-type: none"> <li>Nominal accuracy of 0.5% of full-scale</li> <li>Dynamic error not provided</li> </ul>	<ul style="list-style-type: none"> <li>Not considered</li> </ul>	Very high	Not viable

Table 7.1: Summary of transducer considerations

### 7.2.1 Current Transformer

A current transformer is primarily used to obtain a small current signal representing a larger one for measurement purposes. A major inherent advantage of transformers (both current and voltage) is that isolation between the primary and secondary circuits is obtained.

For lower accuracy measurements, the primary of a current transformer is usually connected in series with the load, in contrast to the line-to-line connection for potential and power transformers. The secondary winding is usually connected to a low-impedance load to reduce power consumption. This is depicted as:

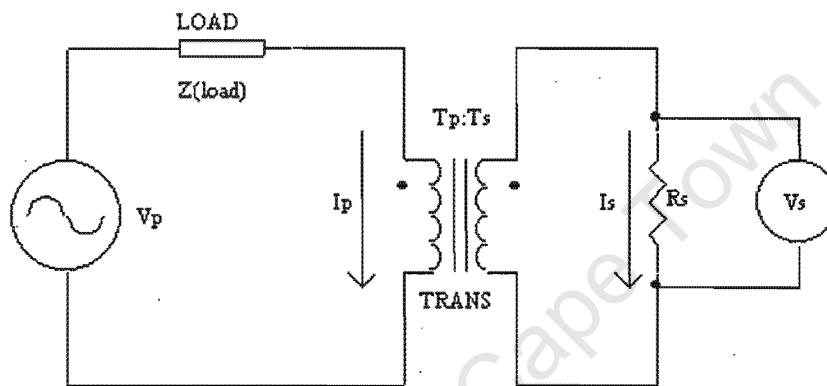


Figure 7.1: Typical connections for standard current transformer

where  $V_p$  and  $V_s$  are the primary and secondary voltages,  $I_p$  and  $I_s$  are the primary and secondary (complex) currents,  $R_s$  is a low impedance secondary load, and  $T_p:T_s$  is the transformer ratio.

Thus, the voltage across the secondary load,  $V_s$ , is directly proportional to the primary load current according to

$$\begin{aligned} V_s &= I_{R_s} \times R_s \\ &= \frac{T_p}{T_s} \times I_p \times R_s \end{aligned} \quad [7.1]$$

The transformer primary and secondary resistance and inductance have not been shown for the purpose of simplicity.

These types of current transformers are stipulated either by the turns ratio of the secondary winding to the primary, or by the nominal nameplate (input-output) ratio. The latter refers to the ratio of the input signal to the output and is correct for a given load. Thus for the example depicted by Figure 7.1, the turns ratio would be  $T_s:T_p$  (e.g. 50:5) and the nameplate ratio would be the numerical value of  $I_{p(\text{nominal})}:I_{s(\text{nominal})}$ , usually in integer values (e.g. 55:5). There exist several ways of obtaining desired current transformer ratios, however these are not within the scope of this report.

Another type of current transformer used in higher accuracy solutions is the “windowed” transformer, so called because the primary current is passed through a “window” in the core of the transformer. This device is not required to connect in series with the load current. Although more costly than their counterparts, these transformer produce more accurate results and also obtain true isolation between the primary and secondary circuits. A typical connection for these devices is depicted below.

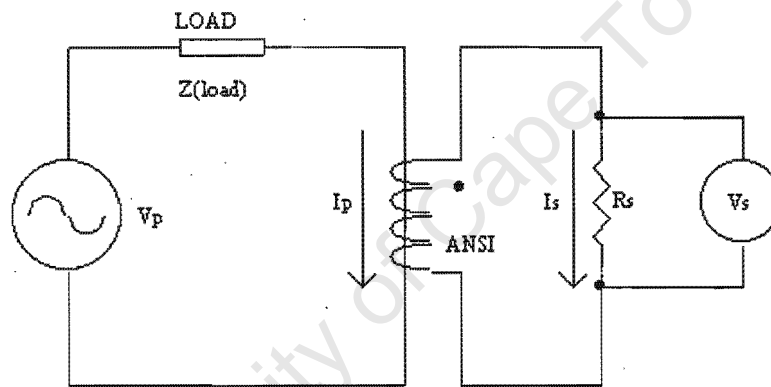


Figure 7.2: Typical Connections for high-precision transformer

The primary current passes via a wire conductor through the core of this transformer, and a secondary current is subsequently induced in the windings.

A particular family of these devices was investigated, the ANSI metering class transformers from CR Magnetics, Inc. (refer below).

## Current Transformer Accuracy

The primary criteria to consider when implementing a current transformer for high accuracy measurements is the burden placed on the secondary of the transformer. This is the impedance of the load that is connected to the transformer and significantly effects the dynamic error of the device. This value is generally given in ohms or VA (volt-amps).

Typical accuracy ratings for standard current transformers generally range between  $\pm 1\%$  to  $\pm 3\%$  at full-scale for burdens ranging from 0.5VA and above. Accuracies in this range do not comply with IEC1036 regulations and these transformers were therefore not considered.

The “windowed” transformer series, however, feature more acceptable errors. As an example, typical accuracy ratings for the ANSI Class Metering Current Transformers from CR Magnetics [19] are:

Turns Ratio	ANSI Metering Class @ 60Hz					(Burden)
	B0.1	B0.2	B0.5	B0.9	B1.8	
50:5	0.3%	0.6%	0.9%	1.2%	2.4%	(Accuracy)

**Table 7.2:** ANSI Current Transformer Accuracy for various burdens

The headings at the top of the table **B0.1** through **B1.8** organize the accuracy of the transformer according to the burden placed on the secondary. For example **B0.1** means a burden of 0.1 ohms is placed on the secondary winding.

The accuracies listed under the burden values are given in percent and are for a full-scale reading. Percent accuracy means that the reading received from the transformer at the burden listed will be within the percentage given of ideal. Hence, a 50 to 5 turns ratio transformer with a primary current of 50 Amps will output 5 amps +/- 0.3% into a 0.1 ohm secondary burden. The current in the secondary circuit will be somewhere between 4.985 and 5.015 amps.

It is critical to understand that the accuracy ratings are for a full-scale reading. This accuracy will only be maintained from 20% full scale and up. Thus the transformer should be selected so that the majority of readings will be within the 20 to 100% full-scale range

From Table 7.2 it is clear that increased loading of the secondary current decreases its accuracy. Thus for the purpose of power metering, the secondary burden should be as low as possible to maintain a high accuracy. Furthermore, the dynamic error of the transformer is highly dependent on the stability of this secondary burden. Low value resistors, such as those displayed in Table 7.2, are highly susceptible to temperature fluctuations, and will reduce the accuracy of the readings. For this reason, very low-tolerance resistors must be employed.

Phase errors introduced by these devices is minimal (data not available). Should phase errors be introduced, however, these may be compensated for using digital phase correction techniques. Phase shift error is frequency dependent and dynamic error may reduce the effectiveness of the compensation. However, due to the relatively low operating frequency range stipulated in IEC regulations, the amount of dynamic phase error introduced may probably be neglected.

Similarly, the dynamic ratio error introduced through fluctuations in frequency is minimal and may also be neglected. Current transformer bandwidths are above those required for power metering applications with the IEC1036 specified fundamental frequency range (45Hz to 65Hz) and subsequent harmonics (up to the 7<sup>th</sup> harmonic).

Further sources of error regarding current transformers include external magnetic and electromagnetic fields, high frequency input signals and fluctuations in ambient temperature. The exact influence of these factors has not yet been considered, and will be investigated further should this device be recommended as the current transducer.

### **Current Transformer Cost**

Standard commercial current transformers from CR Magnetics, Inc. with operating parameters conforming to the range of line voltage and load current nominal to

metering applications, but with unacceptably low accuracy ratings, were found to cost around \$22 to \$26 per unit (CR Magnetics Commercial Current Transformer Range). The more applicable ANSI transformers ranged from \$120 per unit and above.

### 7.2.2 Current Shunt

A current shunt may be considered to be a low-impedance purely resistive device (i.e. it contains minimal inductance that may be neglected). The present generation CASHPOWER units from EML currently use current shunts to obtain voltage signals relating to the load current.

A typical connection for a current shunt is as follows.

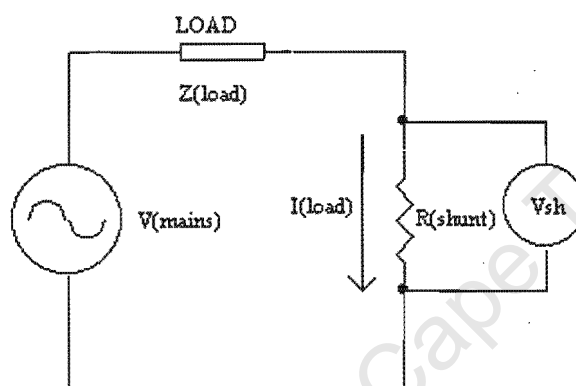


Figure 7.3: Typical shunt connection

where  $R_{shunt}$  is the shunt impedance (in  $\Omega$ ) and  $V_{shunt}$  is the voltage across the shunt proportional to the load current  $I_{load}$ , according to

$$V_{shunt} = I_{load} \times R_{shunt} \quad [7.2]$$

#### Shunt Accuracy

If the device is considered to be purely resistive, a reasonable assumption [16], then no phase error will be introduced between the measured voltage  $V_{shunt}$  and the load current  $I_{load}$ .

The critical source of errors regarding shunt operation is impedance drift due to ambient and self-induced temperature fluctuations [16]. The influence of shunt resistance fluctuation is assessed on the ensuing page.

For the purpose of error analysis, the load impedance is also considered to be purely resistive, thus  $Z_{load} = R_{load}$ . Let  $R_e$  be the resistance error of the shunt, and  $R_{shunt}$  represent the *nominal* value of the shunt resistance. Thus, from Figure 7.3, and noting that  $I_{load} = I_{shunt}$ ,

$$I_{shunt} = \frac{V_{mains}}{R_{load} + (R_{shunt} \pm R_x)} \quad [7.3]$$

and thus, from Equations 7.2 and 7.3

$$V_{shunt} = I_{shunt} \times (R_{shunt} \pm R_x) \quad [7.4]$$

and combining Equations 7.3 and 7.4

$$V_{shunt} = \frac{V_{mains} \times (R_{shunt} \pm R_x)}{R_{load} + (R_{shunt} \pm R_x)} \quad [7.5]$$

For a full-scale load of 80A rms at 240V rms, the load impedance is a (minimum) value of approximately  $3\Omega$ . A nominal shunt value of around  $0.5m\Omega$  is assumed, and the resistance error due to temperature fluctuation of  $R_x$  is extremely small compared with the load resistance and shunt resistance. Thus the value of  $R_x$  in the denominator term above may be neglected for simplicity. Thus Equation 7.5 is reduced to

$$\begin{aligned} V_{shunt} &\approx \frac{V_{mains} \times (R_{shunt} \pm R_x)}{R_{load} + R_{shunt}} \\ &= \frac{V_{mains} \times R_{shunt}}{R_{load} + R_{shunt}} (1 \pm R_x) \\ &= V_{shunt(nom)} \times R_{shunt} (1 \pm R_x) \end{aligned} \quad [7.6]$$

where  $V_{shunt(nom)}$  is the nominal value of the voltage output arising from a nominal shunt resistance.

Thus the percentage output error arising from a variance of  $R_x$  in the nominal shunt resistance is (from Equation 7.6)

$$\mathcal{E}_{R_x} = \left| 1 - \frac{V_{shunt}}{V_{shunt(nom)}} \right| \times 100 = |1 - R_{shunt} (1 \pm R_x)| \times 100 \quad [7.7]$$

and since  $R_x$  may be stated in percentage terms of  $R_{shunt}$ , it is clear from Equation 7.7 that the output error is directly related to the percentage error of the shunt value from its nominal value. Thus

$$\text{output error [\%]} = \text{shunt variance [\%]}$$

It should be noted that this relationship also applies to the secondary burden of the current transformer discussed in Section 7.2.1, since both of these are linear systems.

The shunt resistance fluctuation due to temperature may be expressed either in terms of percent increase per degree [ $+\%/^{\circ}\text{C}$ ] or in terms of parts per million (ppm). Specific data for the particular shunts required this report was not found. However, shunt value fluctuations are usually very low [16] providing an output accurate enough to comply with IEC1036 specifications.

Prices for these devices are unknown, however it is assumed that they are more cost-effective than the current transformers previously mentioned as these are presently used in the CASHPOWER units.

### 7.2.3 Voltage (Potential) Transformer

A voltage transformer is mostly used to provide a scaled voltage signal from the secondary windings proportional to the voltage across its primary windings. An inherent advantage of this device is it provides electrical isolation between the primary and secondary circuits. A typical connection for a voltage transformer is depicted below.

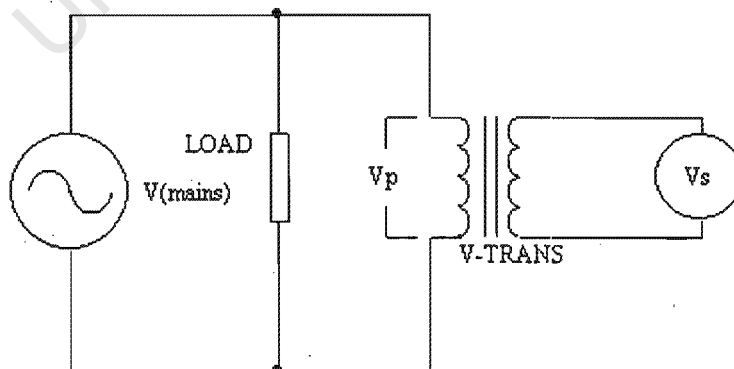


Figure 7.4: Typical connection for a voltage transformer

As with the current transformer, the device is stipulated according to a turns ratio. The output should be fed into a high impedance load to reduce loading which causes erroneous results.

Voltage transformer bandwidths are above those required for power metering applications with the IEC1036 specified fundamental frequency range (45Hz to 65Hz) and subsequent harmonics (up to the 7<sup>th</sup> harmonic).

Main sources of error introduced by voltage transformers are similar to current transformers. These include electromagnetic and electric fields, high-frequency spikes, fluctuations in ambient temperature and input frequency<sub>[16]</sub>. Accuracies for these devices are also similar to the current transformers, ranging from 1% and worse, although high-precision versions are also available at increased costs.

#### 7.2.4 Resistive network as a voltage divider

A voltage divider comprising of two or more resistors may also be used to scale the line voltage to more acceptable limits. A typical connection is as follows:

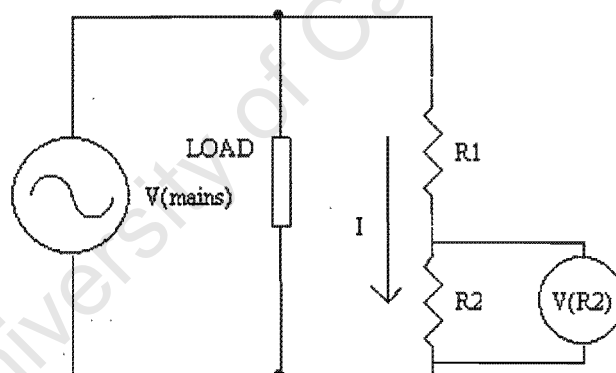


Figure 7.5: Typical resistive voltage divider

The output voltage,  $V_{R2}$  is related to the line voltage,  $V_{mains}$ , by:

$$V_{R2} = \frac{V_{mains} R2}{(R1 + R2)} \quad [7.8]$$

In order to reduce loading of the output voltage,  $V_{R2}$ , it should be fed into a high-impedance load. Decreasing the load resistance effectively decreases the resistance of  $R2$ , changing the characteristics of the divider and producing erroneous results.

### Voltage Divider Accuracy

If it can be assumed that the divider consists of impedances that are purely resistive, that is, they contain no inductance, the output will contain no phase error. All resistors contain some degree of inductance, however, and a marginal phase shift is theoretically introduced. This is usually minimal, and for the application at hand may be neglected.

The two critical sources of error introduced when using a voltage divider such as the one depicted in figure 10.5 are resistor tolerance and fluctuations due to temperature.

Resistor tolerance is usually stated over a dynamic range, that is, over the complete range of rated power dissipation. For example, a  $1\text{k}\Omega$  resistor with a 1% tolerance is rated as  $1\text{k}\Omega \pm 1\% (\pm x\%/^\circ)$ .

Tolerances may be compensated for during calibration simply by adjusting either  $R1$  or  $R2$  in order to obtain the desired voltage  $V_{R2}$ . However, this will be accurate for the nominal ambient temperature only. Temperature drifts change the resistance of *both* resistors respectively. For simplicity, this change may be regarded as a linear one, thus if both resistors in the divider are chosen with the same temperature coefficient, the ratio between them will theoretically remain constant. This is illustrated in the example below:

Let the temperature coefficient of both resistors in Figure 10.5 be  $0.05\%/^\circ$ . Thus, Equation 7.8 may be modified to give

$$V_{R2} = \frac{V_{\text{mains}} R2(1 \pm 0.0005/^\circ)}{(R1(1 \pm 0.0005/^\circ) + R2(1 \pm 0.0005/^\circ))} \quad [7.9]$$

Removing the common term,  $(1 \pm 0.0005/^\circ)$  from the denominator and numerator resolves back to Equation 7.8. Therefore if the temperature coefficient of both the resistors can be considered to be both equal and linear, the output voltage will remain constant.

In practice, however, these assumptions do not hold true, and error arising from dynamic temperature does occur. Actual numerical values for this error may be determined using empirical means, however it is predicted they will remain within tolerable limits using high precision resistors.

The bandwidth of even the cheapest resistors is many times greater than that required for the application at hand.

Other sources of error introduced through resistive voltage dividers are induced noise and (minimal) phase shift arising from the inherent inductance. Long term drift of the resistors may be neglected for the same reason as the temperature drift, that is, the drift of *both* resistors is ideally the same.

The use of high-precision resistors as a voltage divider is more cost-effective than a voltage transformer.

### **7.2.5 RMS value producing current and voltage transducers**

As mentioned in Section 4.6 of this report, transducers containing all the necessary embedded components to produce output voltages and currents proportional to the rms values of the line voltage and load current, respectively, are commercially available. The characteristics of one class of these devices from CR Magnetics are provided in Section 4.6.2.1. The cost of these devices excluded them as viable options and they were not considered further.

### 7.3 Preliminary Conclusions Regarding Final OCM Design

Based on the research presented to this point the following conclusions regarding an OCM solution have been drawn:

- The system cannot use any external rms-to-dc components, including rms transducers and ASICs
- The on-board analog multiplexed MCU A/D converter is the sole analog-to-digital interface that may be used to obtain digital representation of the input signals
- All power calculations must be performed by the MCU
- The system must use a resistor-divider network to obtain the input voltage signal (refer to section 10)
- The system must use a current shunt to obtain a voltage signal proportional to current (refer to section 10)
- The system must present the output as a digital pulse frequency directly proportional to average real power
- Calibration of the input signal amplification gain must be performed

## Section 8: Proposed OCM Solution

Based on the research presented in the previous sections, a final OCM design was developed. The design is presented in this section in three parts, namely an overview of the entire design, a breakdown of the OCM firmware algorithms, and the peripheral hardware. Although several aspects of the design pertain to both the hardware and firmware modules, they are discussed separately in Sections 8.2, 8.3 and 8.4 with cross-references made where applicable.

### 8.1 Overview

#### 8.1.1 Test Version

As part of the terms of reference for this project specified by EML, the system is required to output a digital pulse train, which when integrated represents expired energy. Although the system is not required to display any information regarding the sampled signals, line frequency, power factor, etc, it became necessary during the development stages to monitor these parameters, both as a diagnostic tool for error analysis, and also to verify several assumptions regarding offsets, signal distortions, etc.

For this reason *two* separate algorithms were developed, using the same signal conditioning and power supply circuitry. One of these, hereafter referred to as the test version, does not output a digital frequency, but relays data directly to a host PC via an opto-isolated serial communications link using the standard RS232 protocol. A dedicated Borland C (Version 3.1) program was written for the data logging that outputs files read directly by Microsoft. The data is then displayed graphically and various checks are performed to confirm the results computed by the MCU.

The uploaded data consists of 10 fundamental cycles of 41 samples of voltage and current, and the resulting real power information. Thus a total of 410 samples of voltage and of current are uploaded, and 10 values representing the real power over one cycle for each reading. The 10 values of real power are then averaged and the

result, later converted to real power in watts using an Excel macro, is used to calculate the output error for the respective reading.

The test version was mostly used as a diagnostic tool and to measure the system accuracy, and is not considered the official OCM solution. The algorithm contains many functions and modules not outlined in the terms of reference, and is also not optimised. Furthermore, it does not directly output any information regarding expired energy, merely instantaneous and real power, however the latter is directly related to expired energy when integrated over time, and may therefore be used to determine the system error.

In light of this, the test version algorithm is not discussed in detail in this report. All of the actual power measurement parameters such as sampling frequency, high and low-gain crossover points, mathematical algorithms, are the same as those used in the official OCM solution, hereafter referred to as the final version. References are made in Sections 10.1 and 10.2 (the testing results) to some of the data obtained using the test version, along with several plots of the sampled signals. A full schematic of the isolated coupling used for the serial communications link with the host PC is presented in Annexure A.

### **8.1.2 Final OCM Solution**

The final OCM solution is primarily based on that of the AD7755<sub>[16][3]</sub> discussed in Section 4.10 and 6.1, in that voltage and current samples are multiplied to produce instantaneous power information. A cumulative summer then adds this to a running sum, effectively averaging the instantaneous power information to provide average power. This is then used to generate output pulses according to the meter constant.

A functional block diagram overview of the system is presented as Figure 8.1, and all schematic diagrams are provided in Annexure A. All signal conditioning circuitry is discussed in Section 8.4.

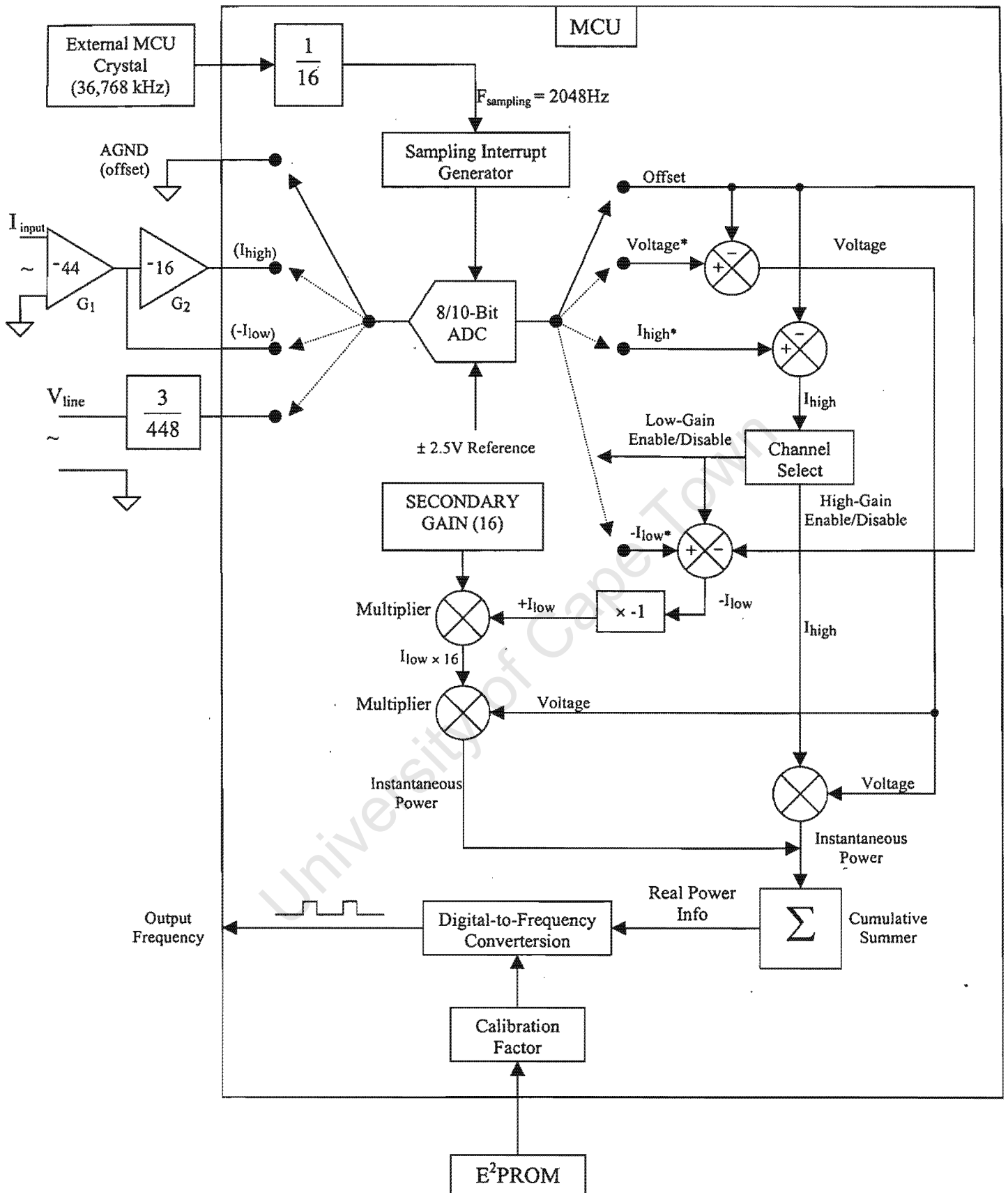


Figure 8.1: Functional block diagram of final OCM solution

## 8.2 Peripherals

### 8.2.1 Two Stage Current Gain

The system contains four input channels, a high-gain (low range) current channel, a low-gain (high range) current channel, an offset channel, and a single stage voltage channel. A two-stage gain design was chosen for the current signal to increase the effective range of operation. The resolution, however, remains the same for the low and high gain channels respectively, and is determined by the MCU ADC resolution. Using the 8-bit ADC on the Motorola MCU, a maximum resolution of  $2^{-8} = 0.39\%$  of full scale is possible for each channel respectively. However, the algorithm is auto-ranging and often uses both gain channels to produce the digitised current signal. When this occurs, the lower range of the digitised waveform contains a resolution *higher* than is possible when using a single gain channel.

The two current signals,  $I_{low}$  and  $I_{high}$  in Figure 8.1, correspond to the outputs of the first and second stage gains respectively. When the load current exceeds a maximum threshold, the system automatically samples from the low-gain channel. The smaller of the two,  $I_{low}$ , is used to sample the current signal for larger load currents (starting at around 4.7A), and vice versa. The crossover point from the high-gain to the low-gain channel is set to a discrete sampled value of  $\pm 120$  on the high-gain channel and the first stage gain is set to 44, the second to 16. These values were chosen after conducting many simulations, discussed in Section 9, to provide the optimal crossover point and maximum resolution.

In terms of the load current, the crossover point is calculated as follows:

The maximum allowable input ADC voltage swing is  $\pm 2.5V$ . This corresponds to discrete sampled values of  $+127$  and  $-128$ , using a two's complement convention, however a maximum discrete sampled swing of  $\pm 127$  is assumed for simplicity. Thus a scaling factor of  $\pm 2.5V / \pm 127$  exists between the input and the sampled value. The second to first stage crossover voltage is thus  $\pm 120 \times 2.5/127 = 2.362V$ .

With  $G_2 = 16$  and  $G_1 = 44$ , this corresponds to a shunt output voltage of

$$V_{shunt} = \frac{2.362}{44 \times 16} = 3.36mV.$$

This corresponds to an rms value of  $V_{shunt(rms)} = \frac{3.36mV}{\sqrt{2}} = 2.37mV$  for sinusoidal waveforms. Thus channel crossover occurs at a load current of

$$I_{load(rms)} = \frac{2.37mV}{500\mu\Omega} = 4.745A$$

All load currents below this value are sampled exclusively using the high-gain channel. Currents above this are sampled using both channels and the secondary gain of 16 is accounted for in firmware. This is discussed further in Section 8.#.(algorithm)

### 8.2.2 Input Signal Offset Compensation

Since the MCU ADC is unipolar and the input signals are necessarily bipolar (refer to Section 8.4), separate analogue and digital grounds are used. The analogue ground (AGND) is held constant at +2.5V with respect to the digital (DGND), and the digital supply rails are thus  $\pm 2.5V$  with respect to analogue ground. In order to convert the (unipolar) sampled signals to bipolar ones, the analogue ground is also sampled and regarded as an offset, which is subtracted from them, as shown in Figure 8.2.

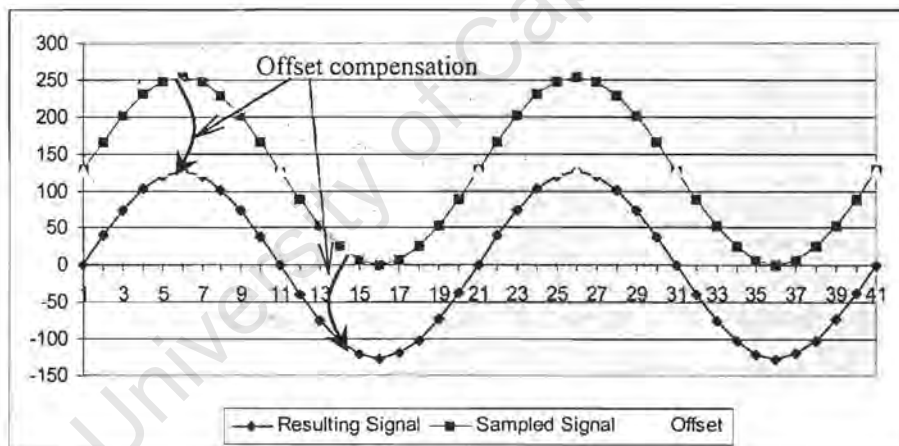


Figure 8.2: Unipolar signal offset compensation

Since the relative difference between the two grounds is assumed to remain reasonably constant, AGND is only sampled once per fundamental cycle (50Hz), saving on processor overhead.

As is seen in Figure 8.1, the low gain current signal is inverted. This is compensated for using firmware and is discussed in Section 8.3.

### 8.2.3 Timing Parameters

Several sampling parameters were determined using Excel simulations (Section 9) of the developed system to provide optimum results. The trade-offs regarding the choice of sampling frequency and average power refresh rate are also discussed in this section.

Sampling timing is interrupt driven using a Timebase Module, which is an internal clock interrupt source featured on the Motorola MCU. The MCU analog multiplexer is also timed using these interrupts. A standard timer/counter may also be used, however the Timebase Module was chosen because of its ease of use and convenient choice of frequencies. The Timebase Module requests interrupts at a frequency derived by dividing down the external crystal frequency. A crystal frequency of 32,768kHz was chosen as this provided a convenient reference frequency for both the internal Bus Clock and the Timebase interrupts.

The MCU Bus Clock frequency is set to 4.9152MHz using an internal PLL frequency multiplier. Since the Motorola MCU requires on average around four Bus Clock cycles per instruction, a minimum of 1MIPS is ensured, although the Motorola MCU is capable of a Bus Clock frequency of over 8MHz, corresponding to over 2MIPS. The Hitachi MCU is also capable of running around 2MIPS, however, since one of the main considerations of the OCM algorithm is to reduce required processor overhead as much as possible, a worst-case scenario of 1MIPS was assumed. This is discussed further in Section 8.3.

The Timebase Module is capable of providing interrupts at the following frequencies:

Division Factor	Interrupt Frequency [Hz]
32,768	1
8192	4
2048	16
128	256
64	512
32	1024
16	2048
8	4096

For reasons stated in Section 9, the sampling frequency was chosen as 2048Hz. Thus, with a fundamental frequency of 50Hz, a total of 41 sample pairs are obtained per fundamental cycle.

It should be noted that the output pulse generation does *not* rely on sampling occurring over a complete cycle. The energy measurement calculation, completed by counting the number of output pulses according to the meter constant, is not reliant on obtaining average power exclusively over one fundamental cycle. If this were the case, the line frequency would have to be measured and the sampling frequency adjusted accordingly to ensure the system did not erroneously detect offsets in the sampled signals, as stated in Sections 4.1 [4] and 4.2 [6]. Instead, the final OCM version continuously computes instantaneous real power for every sample pair and continuously updates the output frequency accordingly. Thus the meter is *not* line frequency dependent when measuring expired energy.[16][3]

However, the test version described before does *not* sample continuously and is considerably reliant on the line frequency. Ten sets of average power, each computed exactly over a single mains cycle, are computed. Thus the test version is only accurate for 50Hz line frequency and was only tested at 50Hz, though it may easily be modified for other frequencies as well.

As discussed in Section 2, all line harmonics within the input bandwidth will also be incorporated in the power calculation [16]. The selected sampling frequency results in a Nyquist frequency of 1024Hz, however the input bandwidth is limited to below 800Hz by the anti-aliasing filters (refer to Section 8.4). IEC1036 specifications stipulate the meter must accurately measure up to and including the 5<sup>th</sup> harmonic [17]. Thus theoretically, a minimum sampling frequency of  $50\text{Hz} \times 2 \times 5 = 500\text{Hz}$  is required, however it was found that oversampling was required to reduce output error to acceptable limits.

Since a single multiplexed ADC was used, a slight phase error between the current and voltage signals is introduced through non-concurrent sampling. In order to reduce this as much as possible, both current channels are sampled every Timebase interrupt,

even though one is not required. This is done since the channel selection mechanism requires several clock cycles to compute which channel to use. By sampling both and then checking which is required, the phase difference between them is minimised. Furthermore, the order in which the voltage and current channels are sampled helps to reduce phase error. The high gain current channel is sampled first, followed immediately by the voltage channel, and then the low gain current channel. Thus with a maximum ADC settling time of  $16 \text{ s}$  and a nominal line frequency of  $50\text{Hz}$ , a maximum phase difference of  $\pm 0.29^\circ$  is introduced between the voltage and current channels, depending on which current channel is used. Unfortunately, this phase error cannot be eliminated completely unless an interpolation method, such as the Reduced Scan method presented in Section 4.7, is used. However, in certain cases where both current channels are used, this phase error is reduced through averaging.

### 8.2.4 Output Frequency Generation

The output frequency generation is achieved using the same mechanism as the AD7755<sub>[16][3]</sub>, where a running sum (cumulative summer) of the instantaneous powers is constantly checked against a stipulated threshold. When the running sum exceeds this threshold, an output pulse is generated. Thus one pulse is generated for a stipulated amount of energy measured. This is clarified with the aid of Figure 8.3 below.

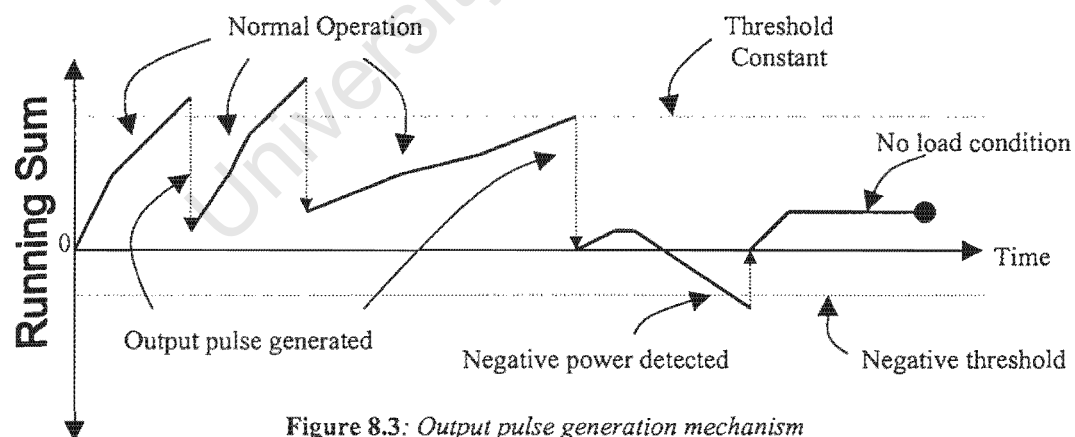


Figure 8.3: Output pulse generation mechanism

Three different load cases are shown above. In the first (the first three curves), the load draws power from the line and a positive energy consumption is measured. As may be seen, when the running sum exceeds the threshold a pulse is produced. The threshold constant is then subtracted from the running sum to begin the cycle again. In

this way, any excess power measured before a pulse was produced is incorporated in the ensuing pulse. This mechanism allows the MCU to check the running sum periodically, instead of with every sample pair, reducing processor overhead. In the final OCM version, the running sum is checked only once per every 41 sample pairs (corresponding to a fundamental cycle at 50Hz).

In the second case, a negative power is detected, possibly resulting from an extremely poor power factor. In this case, the meter should disconnect the supply to the load and periodically test the load to determine if the problem has been amended. The implementation of this function is not within the scope of this project. For the purpose of this project, when negative power is detected, the system simply sets an LED and halts the output frequency.

In the last case, the running sum was not found to increase substantially in a given time limit, indicating a no-load condition. Normally, the output frequency would be halted and the load periodically tested until a minimum current was detected, however this function is not within the scope of this project.

The threshold constant is dependent on several system parameters, including the Meter Constant (MC). The maximum allowable Meter Constant of 4000 imp/kWhr, as stipulated by EML, is used to maximise the meter resolution and to speed up the calibration process. The threshold constant (TC) is calculated in several steps:

Firstly, each impulse corresponds to stipulated energy expenditure:

$$MC = 4000 \frac{\text{imp}}{\text{kWhr}} = \frac{4000}{1000 \times 60 \times 60} \frac{\text{imp}}{\text{W sec}} = 0.00111 \frac{\text{imp}}{\text{W sec}} \quad [8.1]$$

Since (theoretically) 0.011 impulses are produced for every Watt-second, a single impulse corresponds to  $1/0.0011 = 900 \text{ W.sec}$ . Thus the threshold constant should correspond to a measured energy of 900 W.sec.

Secondly, since the maximum output frequency will be produced for the maximum allowable current and voltage at unity power factor, these two parameters also determine the threshold constant. They are required in order to convert 900W.sec into

its discrete value according to the voltage and current ADC resolution and the sampling frequency. This is done as follows:

The scale factors relating discrete digitised values to actual voltage and current are

$$K_V = \frac{\sqrt{2} \times V_{\max(rms)}}{2^{N_V-1}} \text{ for the voltage,} \quad [8.2]$$

and

$$K_I = \frac{\sqrt{2} \times I_{\max(rms)}}{2^{N_I-1}} \text{ for the current} \quad [8.3]$$

where  $N_V$  and  $N_I$  is the voltage and current ADC resolution in bits. Because of the two-stage gain design with a secondary gain of 16, the maximum possible discrete value produced for the current channel is  $\pm 2^{12-1}$  (an effective 12 bits, with the extra 4 bits added by the secondary gain). Thus  $N_I = 12$ , and  $N_V = 8$ .

Therefore, using the standard power calculation of  $P = VI$  (assuming unity power factor), the scale factor relating digitised instantaneous power information (discrete V sample multiplied by discrete I sample) to actual instantaneous power in watts is

$$K_P = K_V \times K_I. \quad [8.4]$$

Thirdly, if the instantaneous power for each sample pair is multiplied by the sampling period ( $T_s$ ), the energy over  $T_s$  is found according to

$$E_{T_s} = \int_0^{T_s} P \cdot dt = T_s \times P \text{ [W]}. \quad [8.5]$$

Thus finally, the threshold constant may be found by relating the digitised voltage and current samples to the calculated impulse threshold of 900W.sec.

$$900W \cdot \text{sec} = TC \times K_P \times T_s \quad [8.6]$$

Using [8.2] through [8.6] and noting that  $I_{\max(rms)} = 80A$  and  $V_{\max(rms)} = 264V_{[17]}$  and  $F_s = 1/T_s = 2048$ , the threshold constant is calculated as

$$TC = \frac{900 \times 2048}{\frac{\sqrt{2} \times 80}{2^{12-1}} \times \frac{\sqrt{2} \times 264}{2^{8-1}}} \approx 114390011 \rightarrow$$

With the stipulated meter constant of 4000imp/kWhr the required output frequency for any given real power (regardless of power factor or waveshape) is calculated as follows:

$$MC = 0.0011 \frac{\text{imp}}{W \cdot \text{sec}} \text{ (from [8.1])}$$

Thus 0.0011 pulses per second are produced for every watt measured.

Noting that  $\frac{1}{1\text{sec}} = 1\text{Hz}$  and power is measured in watts, it is clear that the output frequency is simply

$$F_{\text{output}} = 0.0011 \times P \quad [\text{Hz}]$$

,where P is power in watts. For the given parameters of voltage and current, the range of output frequencies required is from 0.0047Hz, for  $I_{\text{min}} = 0.08\text{A}$ ,  $V_{\text{min}} = 176\text{V}$  and  $\text{PF}_{\text{min}} = 0.3$ , to 23.47Hz, for  $V_{\text{max}} = 264\text{V}$ ,  $I_{\text{max}} = 80\text{A}$  and  $\text{PF} = 1$ .

For the actual developed system, the output is merely toggled instead of producing a pulse, effectively dividing the frequency by two. To compensate, the threshold constant is divided by two. This was done to conserve processor overhead, as a minimum allowable output pulse high-time of 3ms is required by the testing equipment on site at EML. The equipment does not specify a maximum high-time.

The output frequency is used to drive a calibration LED, which is directly read by the testing equipment at EML. However, purely for testing purposes conducted at the University, an opto-isolated coupling was used to isolate the frequency meter from the system.

## 8.3 Algorithm

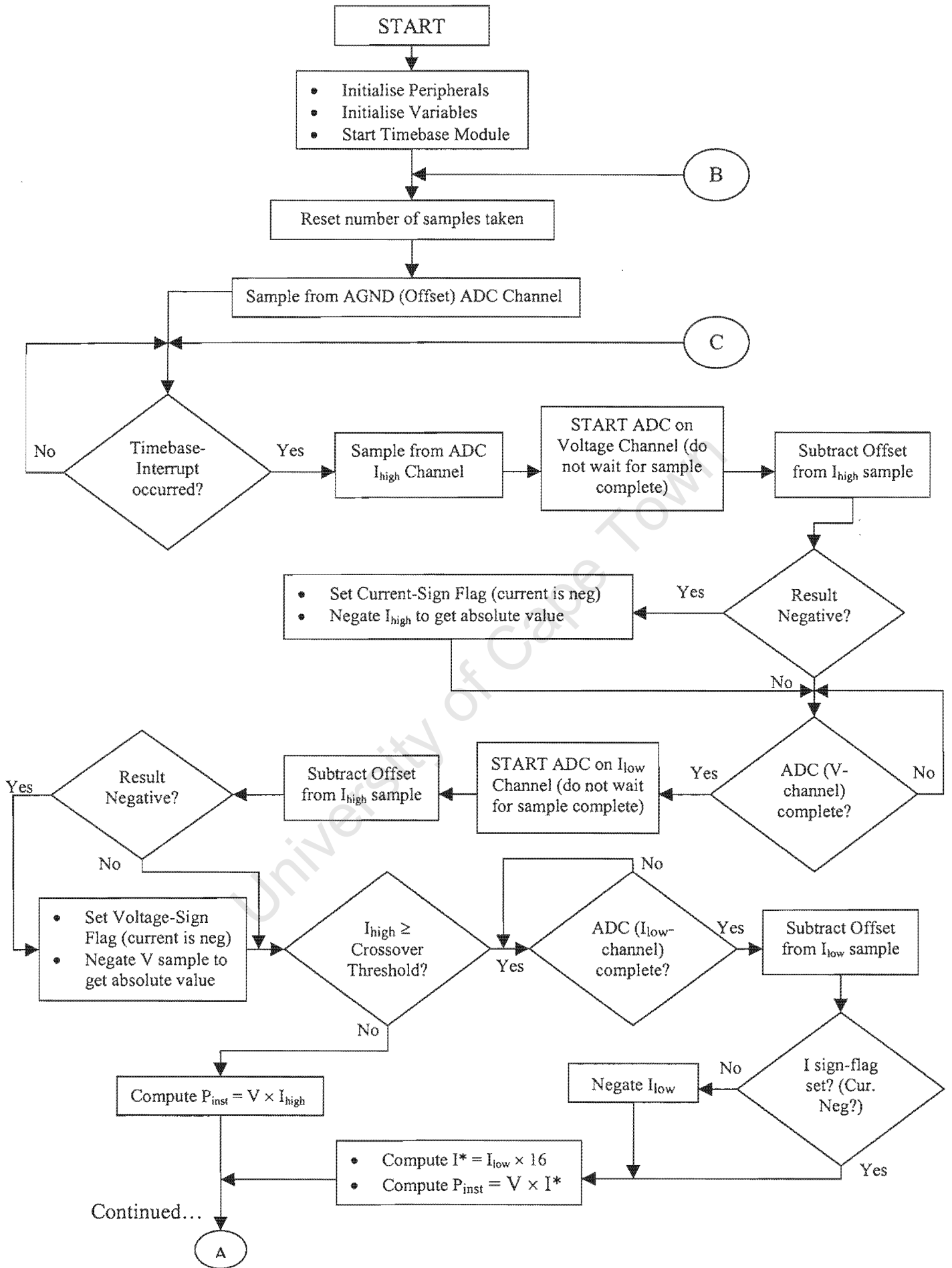
The OCM algorithm was developed in a modular approach in an attempt to facilitate its integration with the existing EML metering firmware. As mentioned before, two versions were created, one exclusively for testing and as a diagnostic tool and the other as the final proposed OCM solution. The former of these is very briefly discussed whilst the latter is presented in this Section in more detail. The firmware for both these is attached as Annexure B.

### 8.3.1 Overview

The complete algorithmic flow diagram for the final OCM algorithm is presented as Figure 8.4. All firmware code was written in assembler and assembled using a dedicated 68HC908 assembler in a development environment specifically for the 68HC908GPZ family of MCUs. Although the metering firmware in the current generation CASHPOWER Utility Meters is written in C and cross compiled to assembler, coding in assembler allows for optimising the processor and memory overhead. Furthermore, calculating the exact number of MIPS required for the algorithm is considerably easier to calculate using the assembler coding.

Simplicity was a key aspect in the algorithm design, mainly to ensure a robust solution requiring as little processor overhead as possible. Several steps were taken to increase the algorithm speed, thereby reducing the required processor overhead. As an example, direct memory addressing was used in all the mathematical operands, as opposed to indirect addressing which requires 1 to 2 extra bus cycles per instruction to compute. Another time saving mechanism was introduced by initiating the ADC and computing various instructions in its stipulated minimum convergence time of 16  $\mu$ s.

In addition, no ADC conversion-complete interrupts are generated, as these may not always occur exactly after 16  $\mu$ s depending on the input voltage. Furthermore, ADC interrupts require a separate Interrupt Service Routine (ISR), further increasing the response time. Instead, a flag is set upon convergence and program flow is easily controlled.



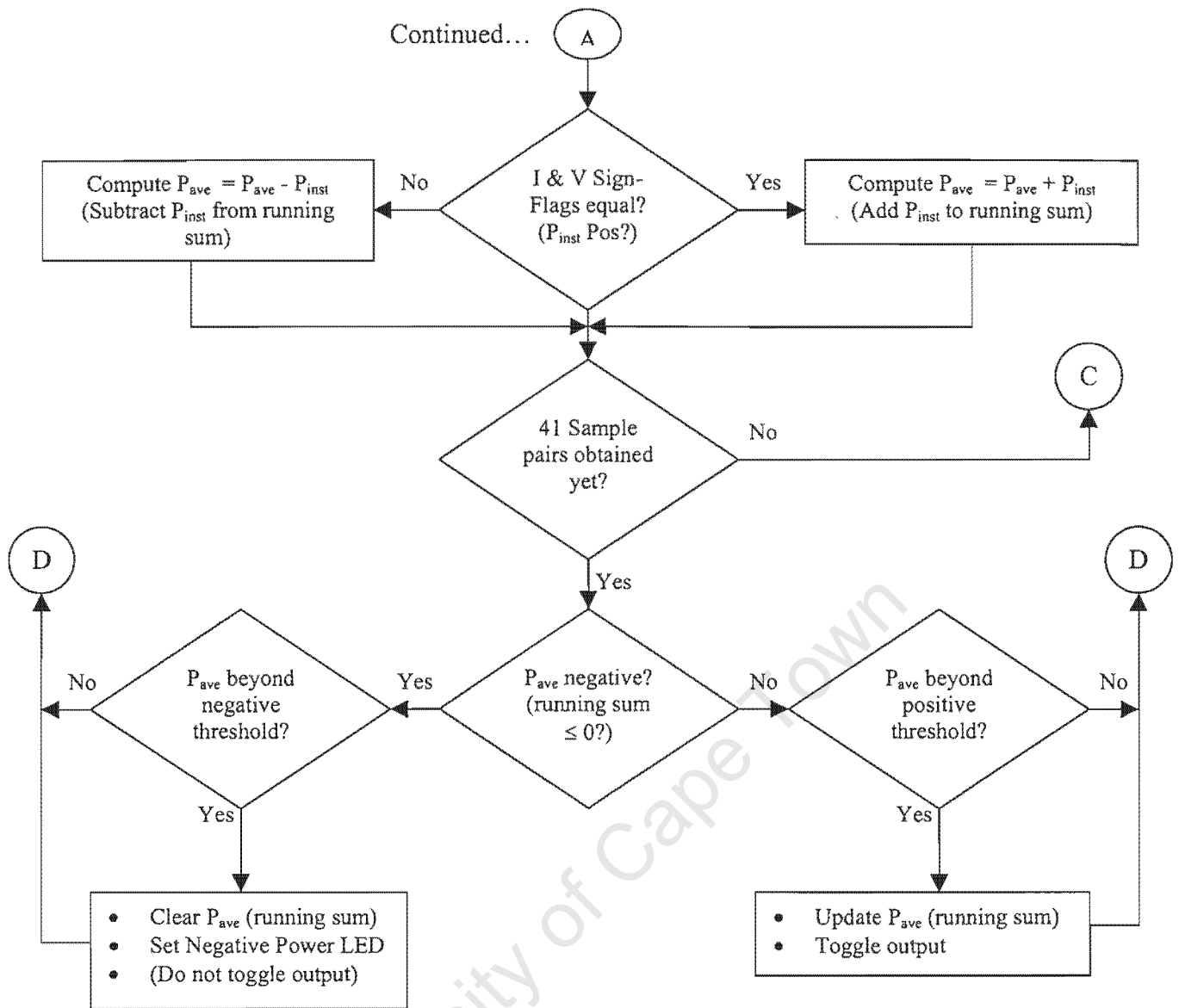


Figure 8.4: Final OCM algorithm flow diagram

Upon a hardware reset, the PLL and Bus Clock, Timebase Module, and the ADC clock speed and conversion status are initialised. Once these peripherals have been set up and the Timebase Interrupts enabled, all variables are initialised and sampling begins.

Before the input signal sampling commences, the analogue ground reference (offset) is sampled. This occurs once every 41 Timebase Interrupts as discussed in Section 8.2.3. Once a Timebase Interrupt Request has been processed, setting an interrupt-received flag, the high-gain current channel is sampled, yielding a sample referred to as  $I_{high}$  in Figure 8.4. Immediately thereafter, the ADC is initiated on the

Voltage channel whilst the offset is removed from  $I_{\text{high}}$ . The result is checked for polarity and stored in an absolute-sign flag format.

Once the voltage channel is sampled and the ADC initiated on  $I_{\text{low}}$ , channel selection is conducted. If  $I_{\text{high}}$  is below the stipulated crossover threshold of  $\pm 120$ , the instantaneous power is computed by multiplying the current and voltage sample in an  $8 \times 8$ -bit operation. If  $I_{\text{high}}$  was lower than the crossover threshold,  $I_{\text{low}}$  is scaled up by the secondary gain of 16 in an  $8 \times 8$ -bit operand and the result multiplied with the voltage sample in a  $16 \times 8$ -bit operation. The output of this also represents instantaneous power.

This is then added to or subtracted from the running sum accordingly, producing average power information. Once the average power has been computed over 41 samples, it is either compared with the pulse threshold value calculated in Section 8.2.4, or a negative power threshold value. If the average power exceeds the pulse threshold constant, the output is toggled. If it exceeds the negative value, a negative (net) power is detected and the average power is reset. A negative-power LED indicator is also set.

Once this has been completed, the cycle begins again by sampling from the analogue ground (offset) channel.

### **8.3.2 Processor and Memory Overhead**

In order to determine the viability of OCM, the relevant persons at EML have requested a complete processor overhead breakdown. The exact processor overhead required by the described algorithm depends on the load being measured, as may be seen in Figure 8.4. Thus the number of instructions required per second [in MIPS] is merely an average and was obtained directly from the embedded firmware listed in Annexure B (the number of Bus Cycles required for each respective instruction is placed in parenthesis next to the instruction). The estimated overhead breakdown is presented in Table 8.1.

Algorithm Module	Approximate Number of Bus Cycles Required	Corresponding MIPS for 4.194 MHz Bus Frequency on 68HC908GP32 MCU	Corresponding MIPS for 8.388 MHz Bus Frequency on 68HC908GP32 MCU
Offset Sampling	20 cycles + 16 s once every 20ms	0.0018	0.0009
Main Sampling Loop	186 cycles + 48 s(max) once every 488.28 s	0.225280	0.112640
Summation Updating	53 cycles every 20ms	0.002650	0.001325
Output Pulse Generation	37 cycles every 43.27ms (at maximum load)	0.000851	0.000426
<b>TOTAL</b>	-	<b>0.230581</b>	<b>0.1154255</b>

**Table 8.1:** *Required processor overhead breakdown*

The data presented above was calculated for a worst-case scenario, i.e. under maximum load with both current channels used every mains cycle.

The 68HC908GP32 Motorola MCU guarantees a minimum of around 2 MIPS with a Bus frequency of 8.388 MHz. The OCM algorithm requires less than 0.12 MIPS. This corresponds to approximately 6% of the processor overhead, and the processor is thus free for around 84% of the time.

The final OCM algorithm used in this project requires 1.036kb of non-volatile ROM (obtained from the assembled Hexadecimal file) and just 21 bytes of volatile RAM, although extensive use is made of the two accumulators which do not require additional memory space.

## 8.4 Hardware

The hardware developed for this project includes the peripheral signal conditioning circuitry, the power supply, an opto-isolated RS232 serial link to a host PC and an opto-isolated output calibration link. The former two modules are discussed below. The latter two were developed purely for debugging and testing purposes and are not discussed in detail. All schematics used in both the final and test versions of the final project are included as Annexure A. Several datasheets supplied by the respective manufacturers have also been included on the CD accompanying this report.

Should OCM prove viable, a suitable power supply will have to be developed as this is not within the scope of this project. Furthermore, all protection circuitry, including that of the power supply and input signals, is also not within the scope of this project and has therefore not been included.

### 8.4.1 OCM Signal Conditioning

In order to assure both high accuracy and stability, it is imperative that the MCU ADC is presented with the appropriate signals. As with most mixed signal applications, a tradeoff exists between a hardware intensive and a firmware intensive design. A hardware intensive design results in a decreased processor overhead at the cost of an increased component count and therefore a higher financial cost. Furthermore, analog signal conditioning is more susceptible to external interference, drift, offsets, et-cetera, than digital signal processing. Thus initially it was decided that the majority of the signal processing should be conducted in the digital domain.

However, once the actual firmware was developed it was found that the larger signals all contained clipping due to large offsets present in the input signals. Furthermore removing any small DC offsets in the sampled signals by high-pass filtering them introduced an inherent phase error. Whilst compensating for the phase error was easily accomplished by byte-shifting one signal relative to the other, this required storing a minimum of 82 8-bit samples for every power calculation for a sampling frequency of 2048Hz. The increased processor overhead required was also substantial.

It was thus decided to conduct all filtering using analogue means. Since the pre-sampled signal has to be amplified externally, incorporating the filters resulted in a minimal increase in component cost.

The particular requirements of the OCM signal conditioning circuitry are listed below. These are discussed in more detail in Sections 8.4.1.1 to 8.4.1.5.

- Signal scaling - The signal obtained from the shunt must be amplified in a two-stage design (refer to Section 8.2.1). The voltage signal may be obtained using a resistor divider network across the load and shunt. Both of these must be scaled to provide the maximum allowable MCU ADC signal at maximum load.
- High-frequency noise attenuation - The optimum signal-to-noise ratios of the amplified current signals should be maintained by low-pass filtering the input signal and removing any high-frequency noise.
- Offset nulling - Any DC components in the two ADC input current signals must be removed before sampling.
- Common-mode noise attenuation - Any common-mode noise induced on the pre-amplified current signal must be removed using a differential amplifier design.
- Anti-aliasing - Anti-aliasing filters must be employed to prevent mirror frequencies appearing in the sampled signals.
- MCU ADC input protection - The MCU inputs must be protected against overvoltages and current and voltage spikes.

The complete developed system is shown below.

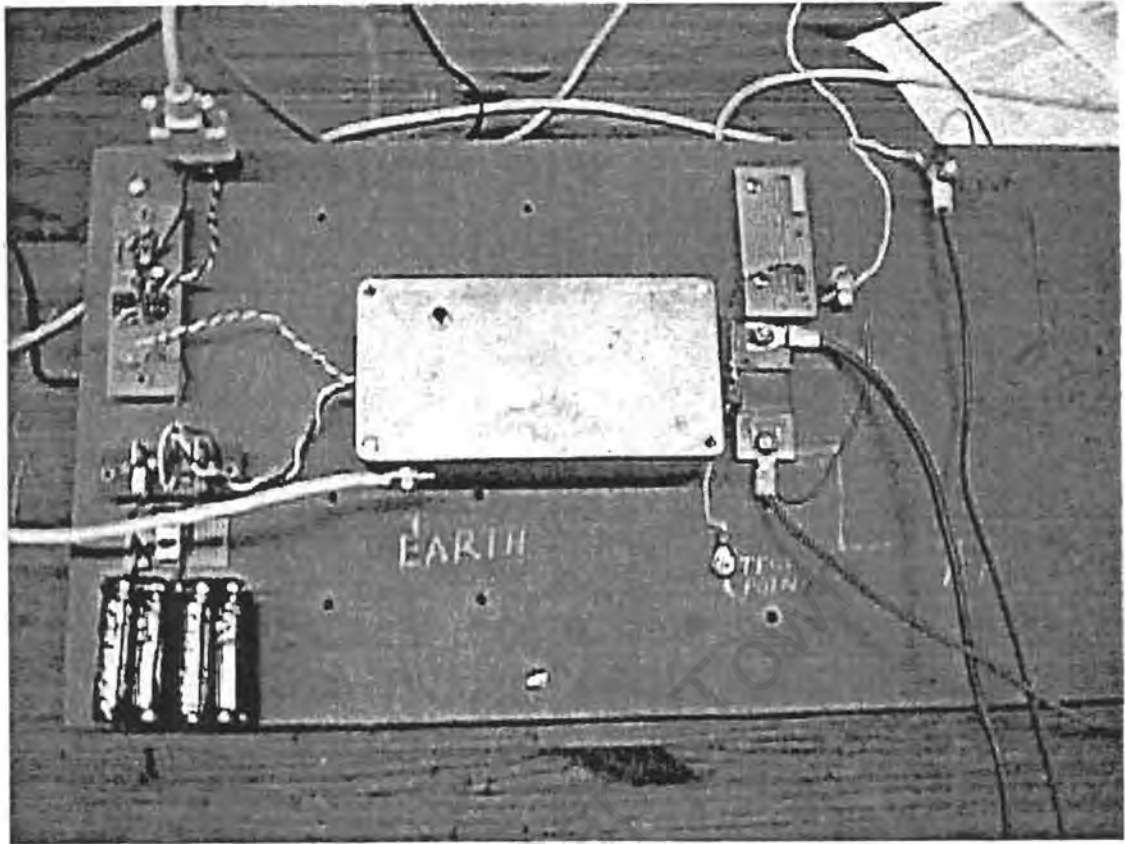


Figure 8.5: Complete hardware system

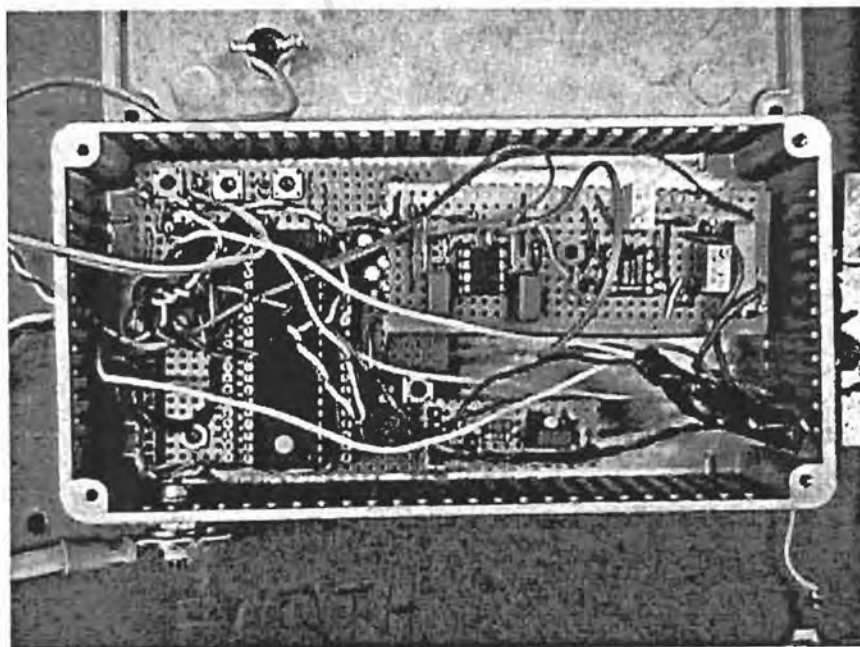


Figure 8.6: Shielded circuitry

### 8.4.1.1 Current-Signal Processing Hardware

The below diagram depicts the OCM current signal conditioning hardware.

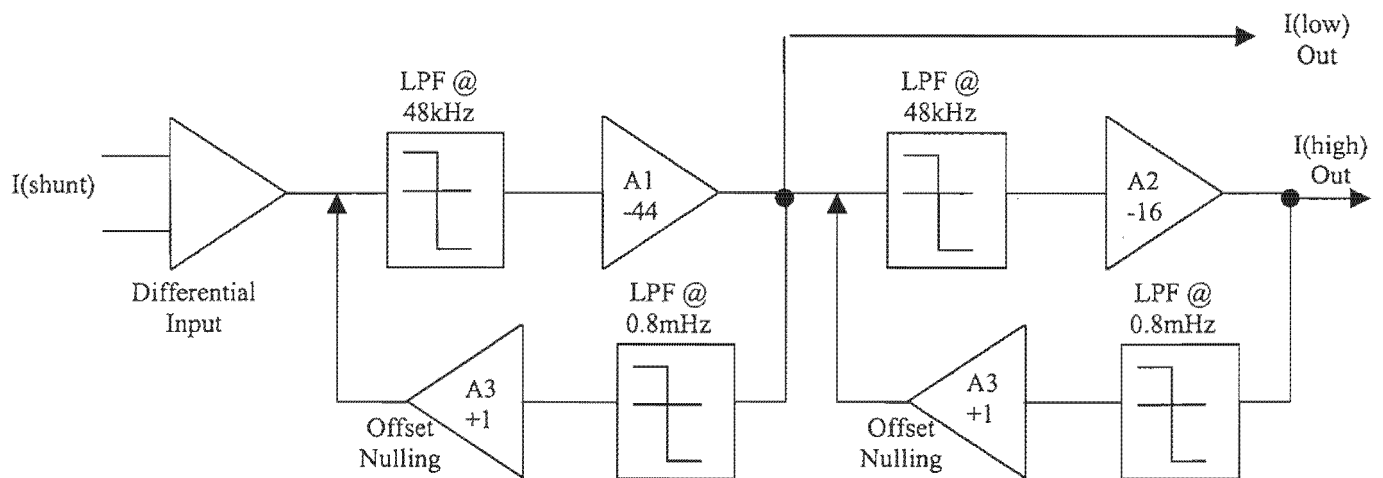


Figure 8.7: Overview of current signal conditioning hardware

The differential operation is applied with reference to ground and is incorporated in the inverting amplifier A1. Both low-pass filters were included to prevent oscillation in the feedback loop and also to attenuate high-frequency noise. The first stage gain of -44 provides the appropriate signal for the high-current (low-gain) ADC input, and the second stage gain of -16 for the low-current (high-gain) input. Offset nulling is achieved using a low-pass filtering negative feedback loop with a very low cut-off frequency.

Since both amplifiers are inverting, the low-gain signal is inverted and the high-gain signal re-inverted to the original polarity. This discrepancy is accounted for using firmware.

Using the bipolar power supply discussed in Section 8.4.2, the output signals are truly bipolar with respect to analogue ground, whilst remaining unipolar with respect to digital ground.

The circuitry schematic is presented as Figure 8.11 on the ensuing page and all anti-aliasing circuitry is presented in Figure 8.17.

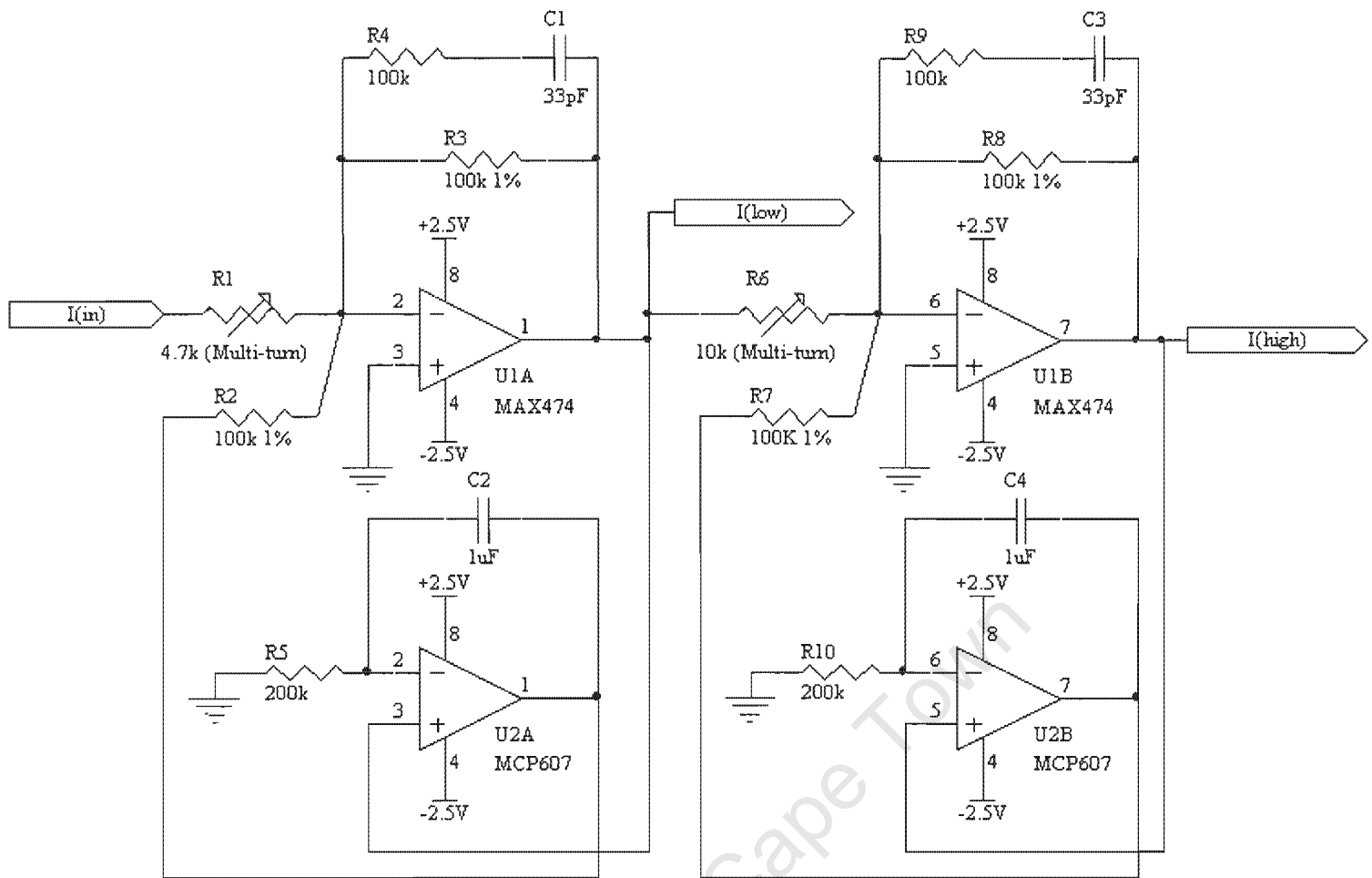


Figure 8.11: Input-current signal conditioning schematic

The shunt coupling is depicted below.

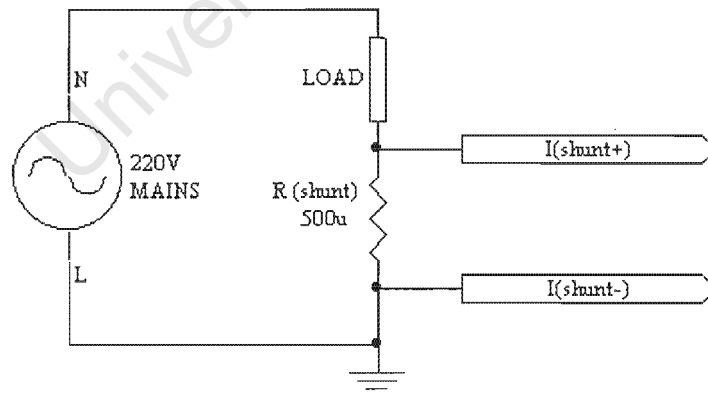


Figure 8.12: Shunt Coupling

The automatic nulling [20] is conducted by the negative low-pass filtering feedback loop using C2, R5, U2A and C4, R10 and U2B. The average value of the outputs of U1A and U1B (A1 and A2 in Figure 8.10) is fed back to the inverting summer and subtracted from the input signal, effectively removing the offset from the output. This method is far more cost effective and more accurate than removing offsets using manual adjustment or high-pass filters. The increased component count is justified by considering the time required and relative inaccuracy in nulling offsets manually using a potentiometer. Furthermore, a manual adjustment allows for static offset compensation, whilst the above method removes dynamic offsets continuously.

The second option, implementing a high pass filter before amplification, introduces an inherent phase error and also attenuates the already low signal. The extent of this phase error was tested using a simple RC high-pass filter with a 1.59Hz cut-off frequency. It was found that the phase error at 50Hz was approximately  $4.7^\circ$ , and more than 5% of the original offset was still passed. Aside from this, the low-gain operational amplifier introduces an offset as well. Although this may be negligible, it is amplified in the second stage by a factor of 16 and effects the high-gain output substantially. Actual sampled data from this experiment is presented in Section 10.1.

A minor trade-off regarding the cut-off frequency of the two integrators is between response speed and accuracy. A lower cut-off frequency ensures true DC filtering but with a slower response to changes in the offset. Two further considerations are voltage drift and capacitor saturation, which may occur if the RC time constant is too large. A reasonable compromise was found with a -3dB corner frequency of 0.8mHz which results in a response speed of 1.25s. It is assumed that changes in the load current offset will occur slower than this, however even if this is not the case the error introduced in the remaining time is minimal.

In order to prevent oscillation and to filter high-frequency noise, the first stage gain is low-pass filtered using R4 and C1, and the secondary gain by R9 and C3. These

provide a cut-off frequency of around  $\frac{1}{2\pi.R.C} = 48kHz$ .

Although the anti-aliasing filters would ordinarily filter out high frequencies, removing these in the amplification phase improves the signal-to-noise ratio and reduces the slew-rate required from the operational amplifier conducting the offset nulling.

The first stage gain is set by adjusting the multi-turn resistor R1, and the second by adjusting R6. It is imperative that the secondary gain be as close to 16 as possible (for the fundamental frequency of 50Hz) in order to ensure that crossover of the two signals occurs at the correct point. It should be noted that once testing commenced (refer to Section 10.1) the main source of output error was found to be the secondary gain, which was exceedingly difficult to set to exactly 16 using the potentiometer.

Unfortunately, due to the offset nulling and low-pass filtering, the gain could not be set accurately using the standard formula of

$$Gain = -\frac{R_{feedback}}{R_{input}}$$

, hence the need for the potentiometer. Furthermore, the 1% tolerance of the feedback resistors, R3 and R8 respectively, is *not* good enough for output accuracies below the  $\pm 2\%$  threshold, further increasing the need for manual adjustment.

#### 8.4.1.2 Choice of Operational Amplifiers

The choice of operational amplifiers for the amplification and offset nulling is critical. Fortunately, due to the relatively low fundamental frequency of 50Hz and low gain, aspects such as slew rate and gain bandwidth product need not be considered. Also, the very low shunt impedance of  $500\mu\Omega$  allows low input impedances to be used with negligible signal attenuation. Even entry-level operational amplifiers surpass all these requirements.

However, the signal-to-noise ratio of the shunt's output is expected to be very poor due to the relatively small signal produced and amplifiers U1A and U1B (A1 and A2 in Figure 8.10) therefore require good common-mode rejection ratios. Since any offsets in the outputs of these two amplifiers are effectively removed through automatic nulling, these two amplifiers do not require a very low voltage offset rating. However, amplifiers U2A and U2B do require low offset ratings since the offsets

present in the low and high-gain signals correspond directly to the offsets induced by these two amplifiers. Thus these operational amplifiers' voltage offsets should be as low as possible to maintain a high accuracy.

All the devices should feature rail-to-rail output in order to prevent clipping under nominal conditions for the high-gain channel and heavy load conditions for the low-gain channel. If this is not possible, the relevant devices must be supplied with rails beyond the  $\pm 2.5V$  of the MCU ADC.

Another important consideration in choosing the operational amplifiers is cost. As stated in the OCM project requirements of section 1, the entire OCM implementation should cost less than R12,50. Other relevant considerations include operating temperature range, temperature coefficient, operating current

In light of the above, the device chosen to perform the offset nulling is the MCP607 from Microchip. This is a dual operational amplifier solution featuring rail-to-rail output and a very low offset voltage of  $\pm 250\mu V$ (max). The device is rated from  $-40^{\circ}C$  to  $+85^{\circ}C$  with a nominal temperature coefficient of  $1.8\mu V/^{\circ}C$ . Incidentally, the device features a common-mode rejection ratio of 91dB (typical) which, in conjunction with the mentioned parameters, makes it an attractive choice for the inverting amplifiers as well. Unfortunately, the MCP607 is a relatively new release and was not widely available at the time of this project. A single sample was provided for this project by EML, courtesy of Microchip, but attempts to attain another failed.

Preliminary pricing for the MCP607 indicates it may be viable option at around R3 to R4 per unit in mass batches. The quad packaged version, the MCP609, is priced just marginally higher. The device is also available in a SMD package, requiring less PCB space. Thus it is recommended that this device (the MCP609) be used for all four operational amplifiers (A1 through A4 in Figure 8.10), should the OCM project be developed further. A full datasheet (MCP607.pdf) for this family of operational amplifiers is available on the attached CD accompanying this report for further referencing.

Since neither a second MCP607 nor a MCP609 was available, the MAX474 was chosen to perform the filtering and amplification. This device features a rail-to-rail output swing to within  $\pm 50\text{mV}$  and a relatively poor input offset voltage of  $\pm 2\text{mV}(\text{max})$ . However, due to the offset nulling configuration this offset does not effect the output, as previously mentioned. All the other device characteristics are within the required limits.

#### 8.4.1.3 Gain and Phase Response

The first half of the circuit depicted in Figure 8.11 was simulated using MicroSim Schematics (Evaluation Version 8.0, 1998) in order to determine the phase and gain response. Unfortunately, the evaluation version of this program allows a restricted number of nodes which was exceeded when attempting to simulate the entire circuit. Thus the circuit depicted in Figure 8.13 was used instead.

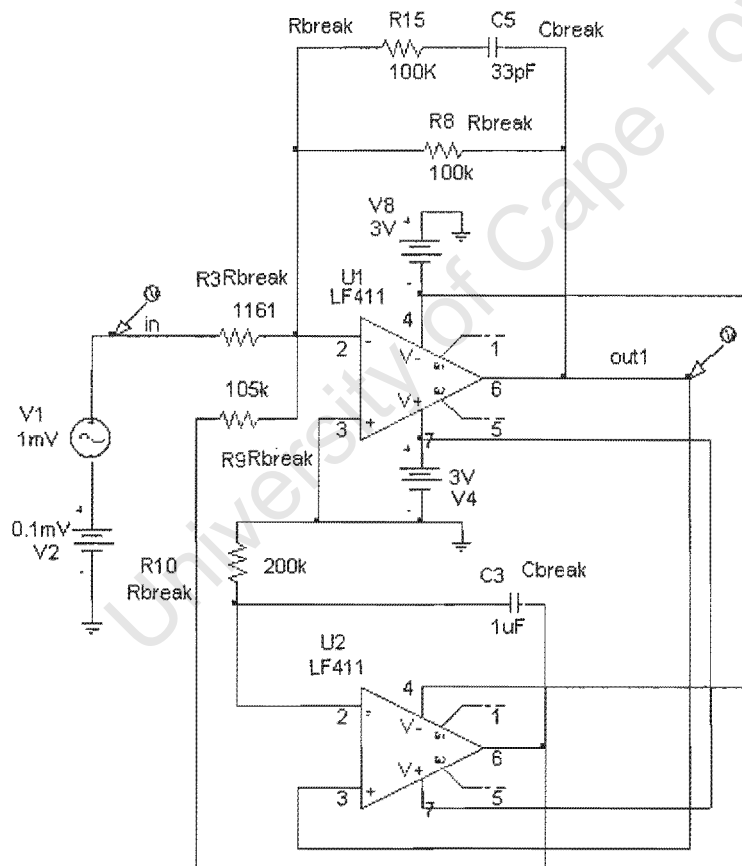


Figure 8.13: Simulated Circuit using MicroSim Schematics

Since models for the MCP607 and MAX474 were not available, the LF411 was used instead. This is a JFET input operational amplifier with a maximum voltage offset of

500 $\mu$ V, thus making it a reasonable substitute for investigating the phase and gain response. The device does not feature rail-to-rail output swing and slightly higher supply rails of  $\pm 3$ V were used to compensate. The input signal is simulated using a 1mV amplitude voltage source with a 100 $\mu$ V offset.

The resulting gain response for the frequency range 10Hz to 1MHz is depicted below.

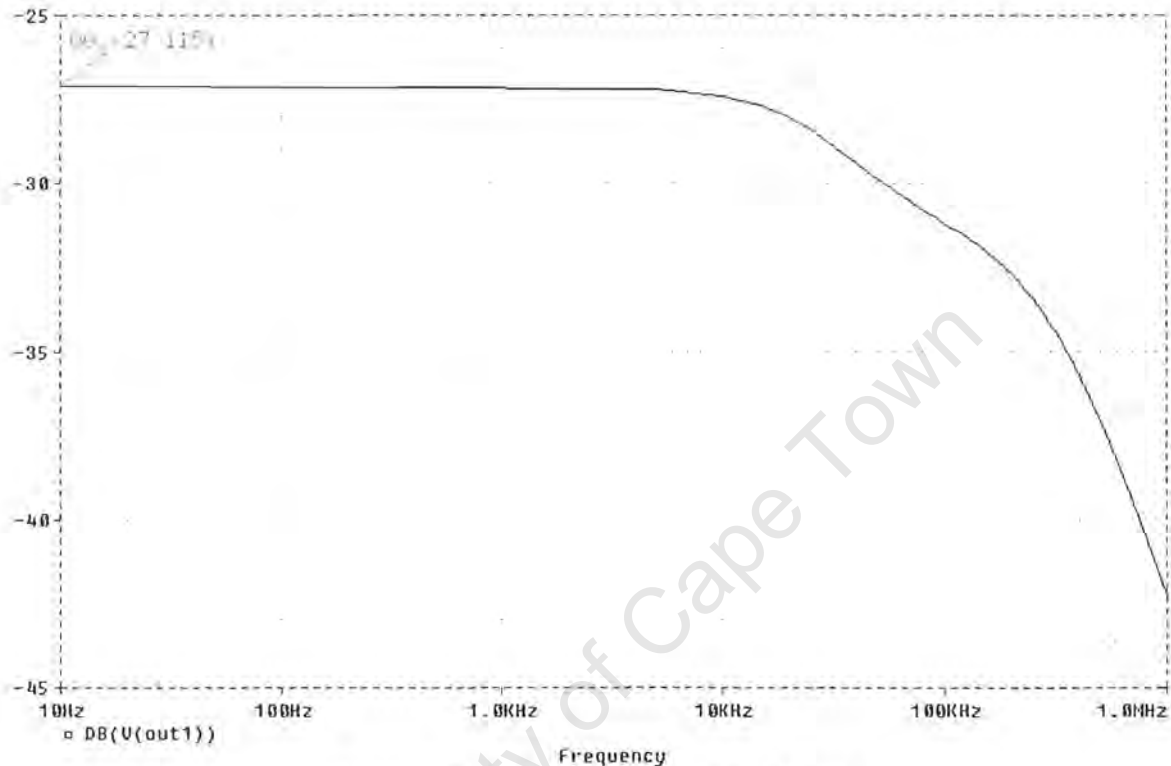


Figure 8.14: Gain response for simulated circuit

As may be seen, the gain is satisfactory for the required frequency range of 50Hz to 250Hz, and attenuates near the calculated cut-off frequency of 48kHz. Using the intercept value in Figure 8.14 the gain at 10Hz is calculated as

$$gain_{10\text{Hz}} = 2 \times 10^{\frac{27.115}{20}} = 45.37$$

The version of MicroSim used does not allow plotting of the phase response when sweeping the frequency. Consequently, the phase shift at the fundamental frequency as well as the 3<sup>rd</sup>, 5<sup>th</sup> and 7<sup>th</sup> harmonics were measured discretely and the results are presented in Table 8.2.

Frequency (f)	Time shift (t <sub>s</sub> )	Corresponding Phase shift (φ)	Resulting power output error at stipulated frequency
50Hz	0s	0°	0
150Hz	55 s	2.97°	0.134%
250Hz	100 s	9.00°	1.23%
350Hz	104 s	13.1°	2.60%

**Table 8.2:** Phase response of simulated circuit

The output error was calculated as follows:

$$\phi = \frac{t_s}{\frac{1}{f}} \times 360^\circ$$

and

$$error[\%] = (1 - \cos(\phi)) \times 100$$

Since any phase shift will result in an erroneous power factor measurement of  $\cos(\phi)$ , the measured output will be accurate to this value. The increasing error at higher frequencies has a lesser effect on the overall output since these harmonics are usually much smaller than the fundamental frequency.

#### 8.4.1.4 Voltage-Signal Processing Hardware

The signal representing the line voltage is acquired using a resistor-network voltage divider, as depicted in Figure 8.15. The required voltage division is achieved using four series connected 100kΩ 1/8W resistors (R1 through R4) and a 4.7kΩ multi-turn trimming potentiometer (R5) for fine adjustment. The theoretically ideal value for R5 is calculated as follows:

$$V_{mains(max)amp} = \sqrt{2} \times 264V = +373.4V \quad [17]$$

$$V_{ADC(in)(max)amp} = +2.5V \text{ (maximum allowable ADC input)}$$

$$\frac{R5}{R1 + R2 + R3 + R4} = \frac{2.5V}{373.4V} \approx \frac{3}{448} \text{ (as calculated using Excel Simulations)}$$

and therefore

$$R5 = \frac{400k \times 2.5}{373.4} = 2.678k\Omega$$

Motorola stipulate that the MCU ADC inputs should be presented with a source impedance of less than 10kΩ in order to avoid attenuation through loading. Hence the choice of resistor values.

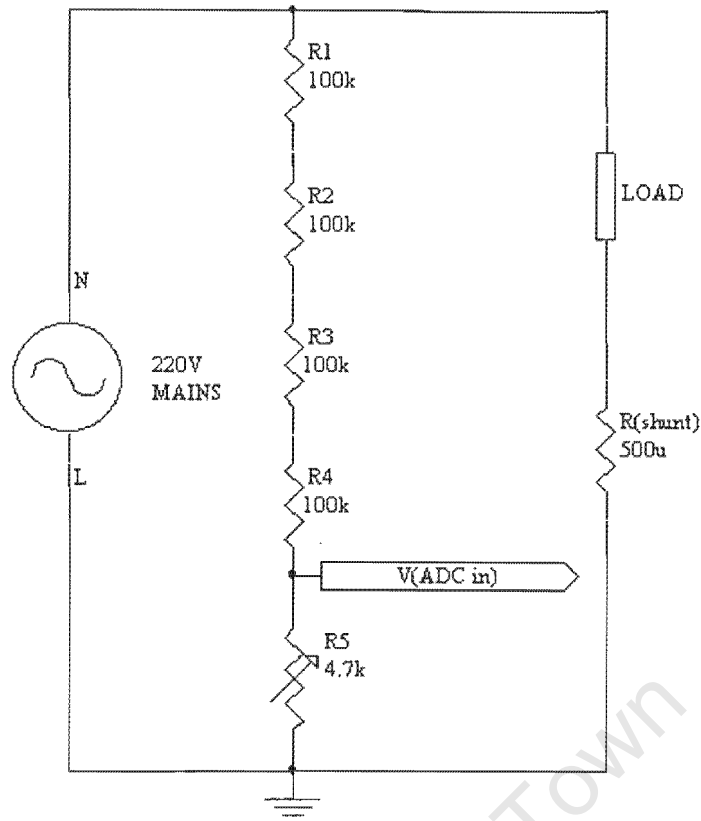


Figure 8.15: Schematic representing the voltage division circuitry

Resistors R1 through R4 are used instead of a single 400kΩ resistor in order to protect against flashover since the flashover rating of a standard carbon-film resistor is around 200VDC. Furthermore, these four resistors share most of the power dissipation, allowing cheaper, lower rated resistors to be used. The total maximum power dissipation through R1 to R5 is:

$$P_{R1 \rightarrow R5} = \frac{V_{\max}^2}{R} = \frac{264^2}{402678} = 173mW$$

Thus R1 through R4 should be able to dissipate a maximum power of around

$$P_{R1 \rightarrow R4} = \frac{173mW}{4} = 43.3mW$$

In light of this, 1/8W resistors were chosen and surpass the required rating comfortably.

#### 8.4.1.5 Anti-Alias Filters

The Anti-alias filters are low-pass filters that are placed before the analog inputs of any ADC. They are required in order to prevent a possible distortion due to sampling called aliasing<sub>[16]</sub>. Figure 8.16 on the ensuing page illustrates the effects of aliasing.

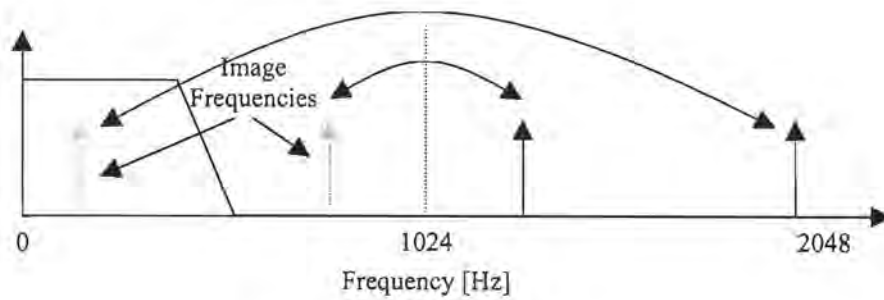


Figure 8.16: *Aliasing Effects*

This illustrates how frequency components above the Nyquist frequency get imaged or folded back down below 1024Hz. In the example shown it can be seen that only frequencies near the sampling frequency will move into the low frequency band of interest. Thus a simple RC LPF may be used to attenuate these high frequencies and prevent distortion in the band of interest. This single pole filter exhibits a roll off of -20dBs/dec for frequencies above the cut-off frequency.

Since these filters exhibit a phase response, it is imperative that all the anti-aliasing filters be matched in order to ensure no phase error is introduced between the channels. Even so, phase mismatch can easily occur due to poor component tolerances in the LPF. Since the resulting error is more pronounced at lower cut-off frequencies, these should be set as high as possible<sup>[16]</sup>. However, the corner frequency should not be set too high, as this could allow enough high frequency components to be aliased and so cause accuracy problems in a noisy environment. The chosen frequency of around 800Hz using a 200k $\Omega$  1% resistor and a 100nF 5% capacitor is a reasonable compromise.

The resulting ADC interface is depicted in Figure 8.17.

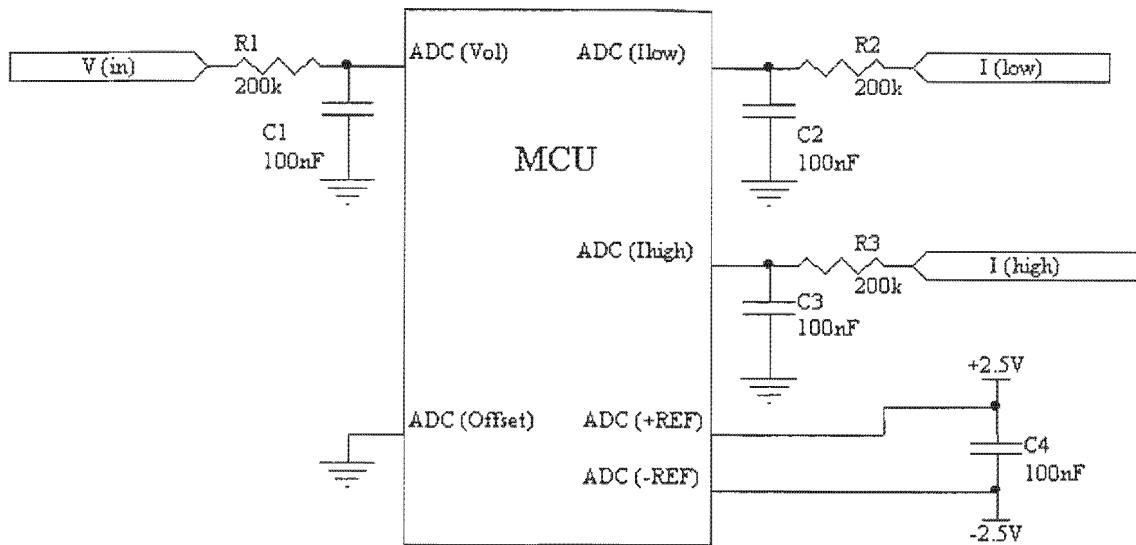


Figure 8.17: MCU ADC interfacing

### 8.4.2 Power Supply

Due to the bipolar nature of the input signals, a bipolar power supply is required. It is noted once again that the development of the power supply is *not* within the scope of this project. However, in order to test the system using real loads, an isolated bipolar supply was necessary. Although some of the standard workbench power supplies available in the University laboratories are isolated, it was decided to construct an independent one for the sake of accuracy and convenience.

It is imperative that the supply be held as close to  $\pm 2.5\text{V}$  as possible, since the MCU ADC utilises these as reference voltages for the A/D conversion. Should a slight offset exist in the supply, all the input signals will appear to contain an offset, and a regulated supply is therefore necessary. The supply constructed for the purpose of this project is depicted in Figure 8.18.

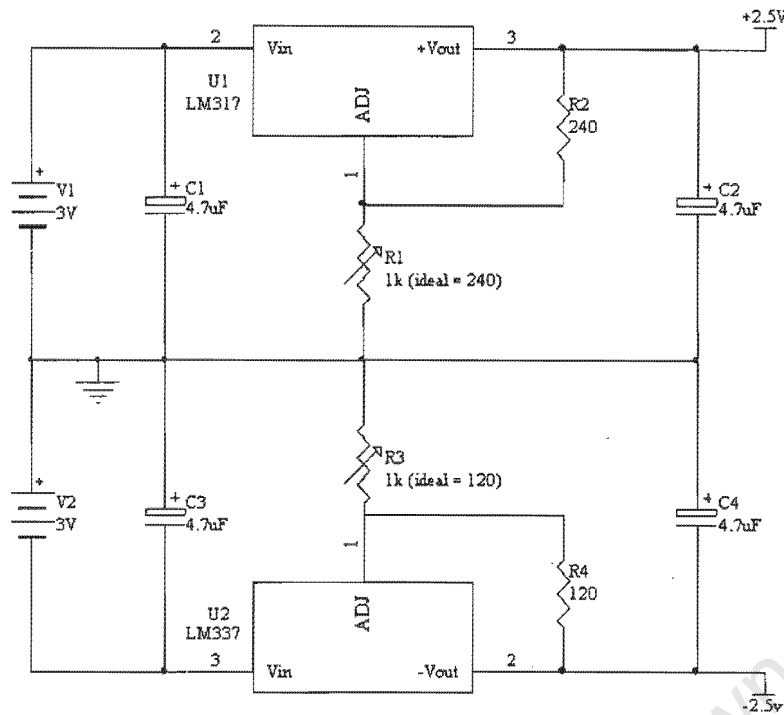


Figure 8.18: Bipolar power supply schematic

The LM317 and LM337 are standard voltage regulators, the output of which is adjusted using R1 and R3 respectively. These devices can supply up to 1.5 A, many times more than is required. Note that capacitors C2 and C4 are *not* the smoothing capacitors, which are employed physically closer to the MCU and peripheral circuitry, and are included mainly for de-coupling. The multi-turn potentiometers R1 and R3 were used instead of static resistors since the latter contain tolerances unacceptable for the purpose at hand. The ideal calculated values are in parenthesis beside R1 and R3 respectively.

Six 1.5V cells were used as V1 and V2 respectively, and the ground point is connected to Mains Live. Thus the entire system floats at a nominal 220V and is not isolated. For this reason the necessary safety precautions had to be adhered to in order to avoid electrocution.

### 8.4.3 Noise and Interference Reduction

In order to reduce the output error as much as possible, various design steps were taken to reduce susceptibility to both external and internal interference and noise. The two most important of these are correct grounding and EMI shielding which are discussed below. In conjunction, all circuitry was adequately de-coupled. The

circuitry was constructed on Veriboard and several plates between the high-voltage inputs and the low current-signal inputs were grounded to reduce induced noise between them.

#### 8.4.3.1 Grounding

All analogue ground nodes in the system were connected to a common point to reduce induced noise and increase the effectiveness of the common-mode rejection, with the exception of the negative output of the shunt. Grounding of this type was not necessary for the digital ground as sufficient de-coupling was included near all the relevant digital devices.

One place where correct grounding is essential is on the shunt. There are three connections to the shunt. One pair of connections provides the current sense inputs, of which one is connected to ground *indirectly* through the current signal differential input of A1 in Figure 8.10, and the third connection is the ground reference for the rest of the system. The equivalent circuit [16] for the shunt is shown in Figure 8.19.

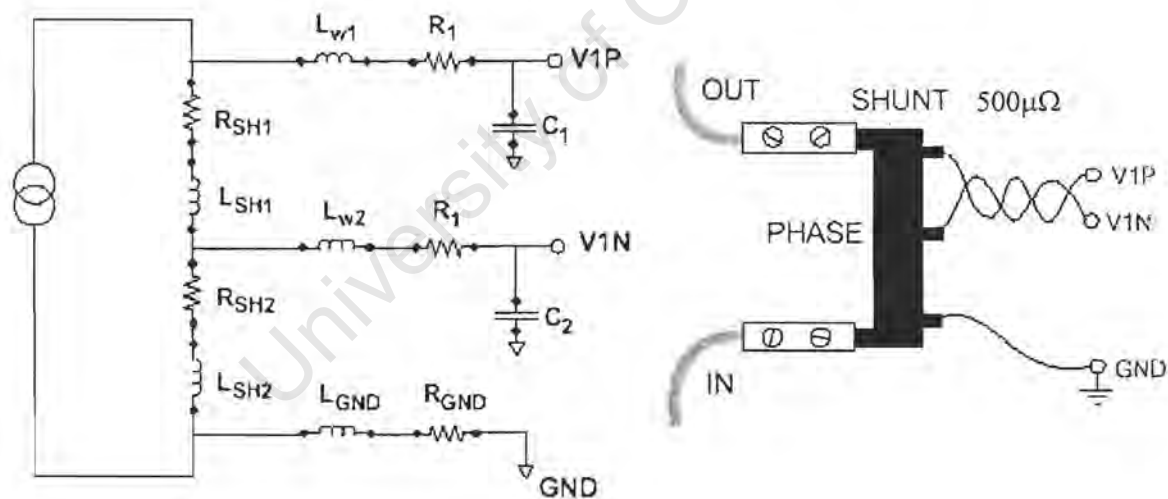


Figure 8.19: Equivalent shunt circuit[16]

The actual shunt used is shown below.

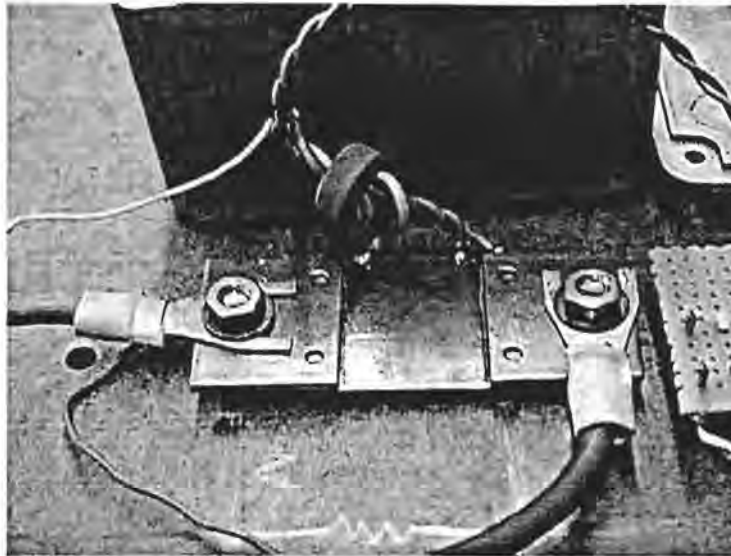


Figure 8.20: Actual shunt circuitry

As may be seen, a twisted wire pair was used to connect the differential output to the peripheral circuitry. The ground reference connection is also visible.

The actual shunt resistance is shown in Figure 8.19 as  $R_{SH1}$  ( $= 500\mu\Omega$ ).  $R_{SH2}$  is the resistance between the  $V_{IN}$  input terminal and the system ground reference point. This resistance is much smaller than  $R_{SH1}$  and thus the ground reference and  $V_{IN}$  are essentially at the same potential, except for the common-mode noise induced on the twisted wire pair connecting  $V_{IP}$  and  $V_{IN}$  to the current-signal conditioning circuitry.

When used at low frequencies the shunt can be considered as a purely resistive element with no significant reactive elements [16]. The shunt used in this project was considered to be purely resistive and the parasitic inductance was not compensated for. This is a reasonable assumption since the buffering action of the operational amplifiers reduces the effect of the inductance considerably. In addition, the effect of the inductance is considerable only at very low shunt resistances (in the order of  $200\mu\Omega$ ) and at high input frequencies (above the 7<sup>th</sup> harmonic for a 50Hz system) [16].

The latter claim was confirmed by Analogue Devices [16] by plotting the phase and magnitude response of an anti-alias filter network with and without (dashed) a parasitic inductance of 2nH. This is presented as Figure 8.21 below.

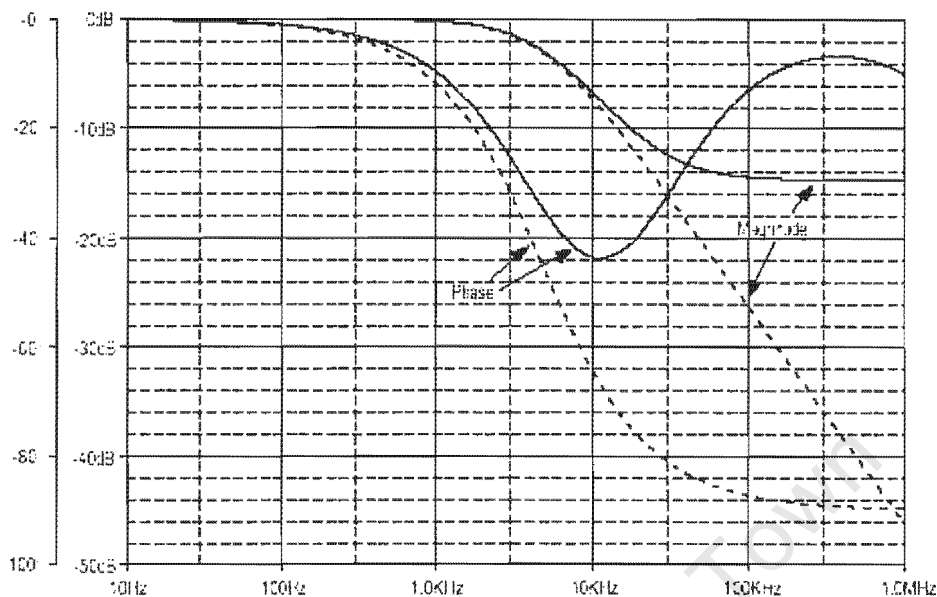


Figure 8.21: Effect of the parasitic shunt inductance

As can be seen from the plot, both the gain and phase response of the network are effected, but only at frequencies beyond those in the considered range for this project. It is predicted that the addition of a buffer between the shunt output signal and the anti-aliasing filter would reduce the effects shown in the figure above.

#### 8.4.3.2 EMI Shielding

The entire system was housed in a grounded metallic enclosure to shield against ambient EMI, as shown in Figure 8.9. Before this was done on the prototype, the induced noise increased the output error beyond acceptable limits. Furthermore, a twisted wire pair was used to connect the shunt to the peripheral circuitry (refer to Figure 8.20) in an attempt to reduce non-common-mode noise.

The effects of EMI, particularly on the current signal, are considerable. The input of the first current channel was viewed on a digital oscilloscope with no load connected. The output was seen to contain noise calculated by the oscilloscope of approximately  $780\mu\text{V}_{\text{rms}}$  at a fundamental frequency of 50Hz. Fortunately, most of this was common-mode and was rejected by the differential input, however the high-gain

output still contained noise of around  $1\text{mV}_{\text{rms}}$ , which corresponds to around 5.12% of the minimum voltage detectable by the Motorola MCU ADC. Fortunately, the gain settings ensure that even at the minimum load current the ADC is presented with a signal several fold higher than this and the overall effect of the noise is substantially reduced to more acceptable levels.

University of Cape Town

## Section 9: Microsoft Excel Simulations

A Microsoft Excel worksheet was used to assess the influence of both internal and external factors before conducting hardware testing. Once the optimal set of parameters was found using this worksheet, the system was built and tested. The Excel worksheet proved valuable as a timesaving mechanism and allowed the cost-effective testing of various combinations of parameters. The entire worksheet has been included on the CD accompanying this report.

Simulations using data in the discrete domain were conducted using various Visual Basic Macros created for this purpose. The input parameters considered in the simulations are depicted in a screen capture of the Parameters sheet of the spreadsheet below.

Ambient Conditions	
Static Power Factor	1
Line Frequency [Hz]	50
Ambient Temperature [deg]	23

Voltage Signal Parameters	
Nominal Line Voltage [Vrms]	220
Minimum Line Voltage [Nominal - %]	20.00%
Corresponding Minimum Voltage [Vrms]	176
Maximum Line Voltage [Nominal + %]	20.00%
Corresponding Maximum Voltage [Vrms]	264
Voltage Division Ratio	3/448
Voltage Offset [mV]	10

Current Signal Parameters	
Nominal Load Current [Arms]	20
Minimum Load Current [% of Nominal]	0.40%
Corresponding Minimum Current [Arms]	0.08
Maximum Load Current [% of Nominal]	400.00%
Corresponding Maximum Current [Arms]	80
Shunt resistance [uOhms]	500
Injected Differential Random Noise (pk) [mV]	0.01

Current Signal Gain Factors	
Number of Gain Stages	2
Stage1 Gain	43.82869189
Stage 2 Gain	16
Stage 1 Crossover (Lower Hysteresis) [bit]	4
Stage 2 Crossover (Higher Hysteresis) point	-120

Sampling Parameters	
Sampling Resolution [bits]	8
Number of Samples per Cycle	40.96
Number of Cycles Incorporated	10
Conversion Time [uS]	16
Effective Sampling Frequency [Hz]	2048
Total Sample Pairs Incorporated	410
Relative Phase Error [deg]	0.288

Operational Amplifier Parameters	
Offset [uV]	200
Offset Temperature Coefficient [uV/deg]	10

Pulse Generation Parameters	
Scale Factor	0.161132813
Meter Constant [imp/kVhr]	4000
Summed Threshold Value	5719505.455

Accumulated Gain	
43.82869189	701.2590703

Effective Crossover Points (Abs)	
4	16
-120	120

Figure 9.1: Input Parameters for Microsoft Excel Simulation Worksheet

All user changeable parameters are highlighted in blue and calculated system attributes are in green. This setup allows for various parameter adjustments and also calculates the necessary calibration factors for a wide range of inputs, thereby facilitating the adaptation of the final product to suit various operating environments.

## 9.1 Input Gain Settings

The voltage and current channel gain settings were calculated to allow for the maximum possible ADC input range.

The voltage channel gain setting, expressed as a division ratio in cell F13 in Figure 9.1, was calculated to provide a maximum voltage swing of  $\pm 2.5\text{V}$  at the maximum stipulated RMS line voltage of  $264\text{V}$  [17]. The calculated ratio of  $3/448$  is achieved using several series-connected resistors and a multi-turn precision potentiometer for fine-tuning on the actual hardware platform (refer to Section 8.4).

The first stage gain (cell F 27) can be calculated manually to achieve the best possible resolution from the input signal parameters. At the maximum possible load current, the low-gain ADC input should be presented the maximum (unclipped) allowable signal with a  $\pm 2.5\text{V}$  amplitude. However, this calculation was performed using the Microsoft Excel iteration function, Goalseek, and for the stipulated system parameters of  $I_{Max} = 80\text{A}$ ,  $R_{Shunt} = 500\mu\Omega$ ,  $V_{dd} = +2.5\text{V}$  and  $V_{ss} = -2.5\text{V}$  the first stage gain was calculated as  $43.8$  ( $\approx 44$ ).

Similarly, the secondary gain is set so that the output signal produced from this (the second) channel corresponds to the maximum allowable input voltage swing of the MCU ADC exactly at the stage 2 gain-to-stage 1 gain crossover value (currently set to 120). Thus an (unclipped) signal with amplitude  $\pm 2.5\text{V}$  will be presented to the high-gain ADC input at the point where the low-gain channel ADC reads a sampled value of 120.

The final current signal used in the instantaneous power calculation is a combination of these two channels. Using a two-stage gain design effectively increases the input

signal range, but does *not*, however, increase the overall resolution. The final current signal is thus comprised of up to 256 different discrete variables with a maximum possible range of - 2048 ( $= -128 \times 16$ ) to + 2032 ( $= +127 \times 16$ ) with the system operating under maximum load current.

With these gain settings, the MCU ADC is presented a signal with  $\pm 0.028\text{V}$  amplitude (assuming no offset exists) at the stipulated minimum current of  $0.08\text{A}$  [17]. This corresponds to discrete sampled values between -2 and +2 for an 8-bit ADC. It should be noted that the actual system is *not* expected to measure load currents of this low amplitude due to the relatively low resolution of the 8-bit Motorola ADC as discussed in Section 3. Load currents of this magnitude will probably be assumed as no-load conditions, however the 10-bit ADC of the Hitachi MCU may be more successful.

A simple amplitude comparator is used as the switching mechanism to determine which channel's sample is to be used in the instantaneous power calculation. The high-gain sample is compared with a static crossover constant (set to 120). If the sample is below this, the high-gain channel's sample is used. If the sample is above this value, the low-gain sample is used with the appropriate secondary gain scaling (set to 16 - or 4 bits). Thus before the input signal rises beyond the clipping threshold of the high-gain channel, the system automatically uses the low-gain channel. The input signal cannot exceed the supply range of the MCU when clipping occurs since the operational amplifiers are powered off the same supply. Thus damage to the MCU ADC channels is avoided at high load currents.

The following graphs depict three cases for input loads taken from the simulation. A small DC offset has been injected, however switching occurs in the same manner when no offset is present.

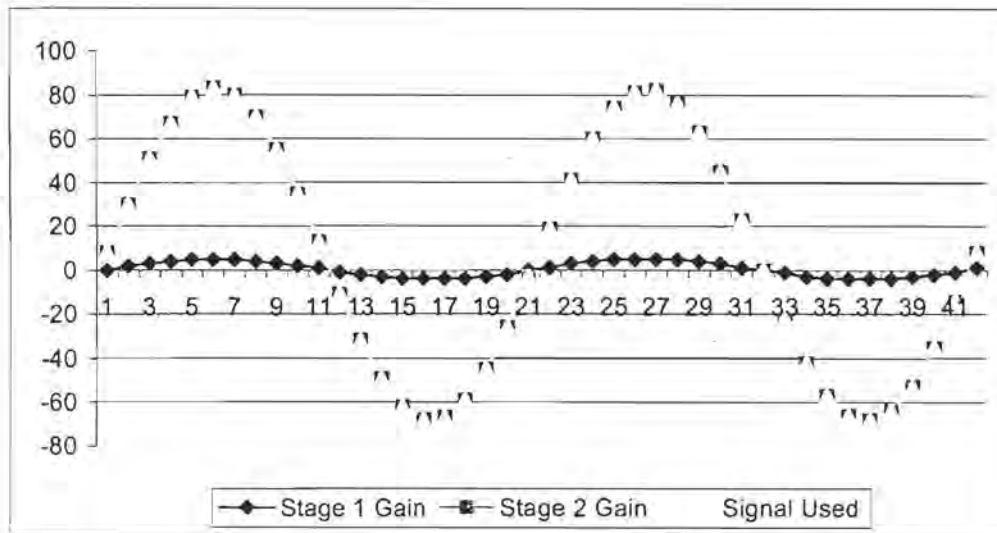


Figure 9.2: Graph depicting simulated current signals with a 3A load current

In the first case, no clipping occurs on the high-gain channel and the signal does not exceed the switching threshold value of 120. Thus the final signal is comprised entirely of the high-gain channel samples. The load current for this example was set to 3A with all other parameters set to nominal.

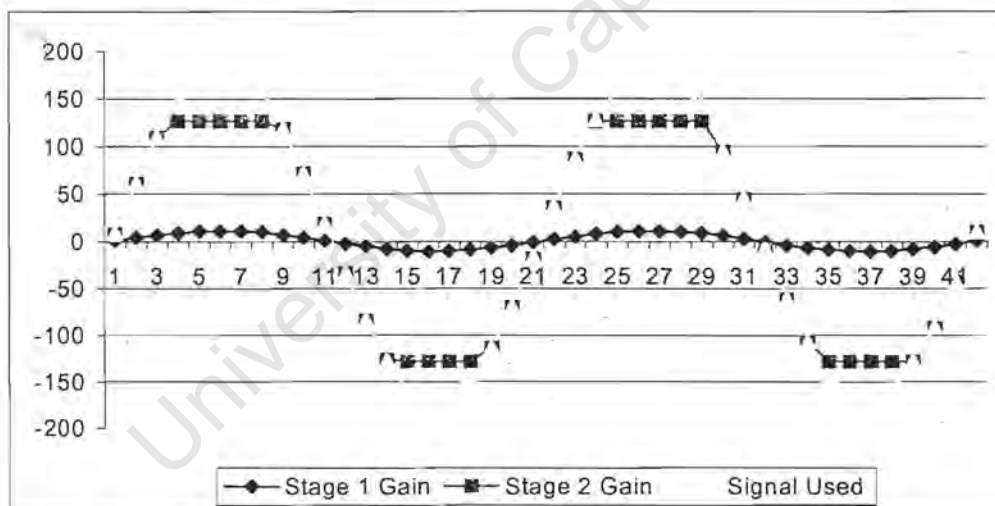


Figure 9.3: Graph depicting simulated current signals with a 7A load current

In this case, the high-gain channel signal exhibits clipping of the peak values and the signal crosses the switching threshold for 5 samples on the positive cycle. The final signal is thus comprised of both the channel's signals with a load current of 7A.

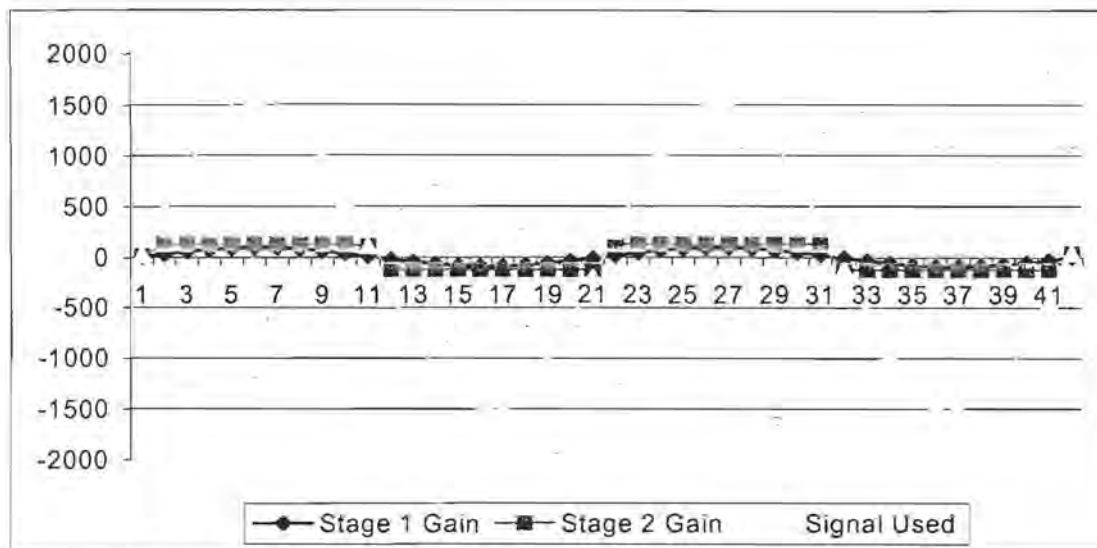


Figure 9.4: Graph depicting simulated current signal with a 60A load current

In this example, the final signal is comprised entirely of the low-gain signal with the load current set to 60A.

## 9.2 Sampling Parameters

Two of the main parameters assessed when conducting the simulations are the sampling frequency and ADC resolution. The influence of these parameters on the overall system accuracy is complex to determine stochastically, especially when considering noisy input signals containing multiple harmonics and when measuring over a large dynamic range. Using simulated data to assess the influence of these parameters greatly facilitated the determination of their optimal values.

The ADC conversion time was incorporated in the signal calculations as a relative phase error between the voltage and current channels. The Motorola MCU ADC conversion time of  $16\mu\text{s}$  corresponds to a  $0.288^\circ$  phase error at 50Hz. This error is increased for the 3<sup>rd</sup> and 5<sup>th</sup> harmonics, which must both be incorporated in the energy calculation [17], but was assumed to be static for the simulations.

The main trade-off regarding the sampling frequency is between accuracy and processor overhead. Oversampling at higher frequencies increases accuracy at the cost of more calculations required per time unit.

The sampling cycle period refers to the number of mains cycles incorporated in the *power* calculation. Average real power is *not* required as an output for the purpose of OCM. The energy measurement method described in Section 8.2.4 simply requires a running sum of the instantaneous power information over a single mains cycle, and is used to generate a pulsed output proportional to the system energy. It was anticipated that testing the actual system using the pulsed output would be extremely time consuming due to the relatively low output frequencies produced when measuring small loads. It was thus decided to use the real power information for testing and calibration instead.

Real power information measured over just a single cycle is available, but is relatively inaccurate. Real power taken over several cycles produces more consistent and reliable results, since influences such as random noise and quantisation error are generally averaged out. Moreover, the system may be highly erroneous when measuring rectified or burst-fired loads.

The trade-off regarding the sampling cycle period is between increased accuracy, through better averaging, and system response speed. Averaging over a longer period results in the system being less responsive to changes in the actual power. With this in mind it was found that averaging over 10 mains cycles was a reasonable compromise.

#### Additional Parameters

Other parameters considered were ambient temperature, temperature coefficients for the two operational amplifiers, injected white-band random noise, voltage and current signal offsets, the meter constant and operational amplifier internal offsets. The nominal values used for these parameters are presented in Figure 9.1 in their respective cells. All these, with the exception of the meter constant, were used in the discrete signal production described in the below section.

### 9.3 Digital Signal Production

The discrete digital signals representing the sampled data are calculated using a rounding process mimicking the ADC quantisation, the resolution of which is determined from the number of ADC bits. In addition the quantised signals incorporate the following factors: input gain settings, phase errors, random noise,

offset errors, clipping, ambient temperature, dynamic range, power factor, line frequency, sampling frequency etc. Also incorporated are the channel switching mechanism for the two current channels, the respective scaling factors for voltage and current channels, the output scaling factor, the instantaneous power information for each sample, and the accumulated instantaneous power information (running summation information) used in the output pulse production. A two's complement convention is assumed throughout, and thus the negative range of the discrete variables is one digit larger than the positive. This is shown in cells J11 and J22 of Figure 9.6 as an example.

The total number of discrete values calculated for each simulation corresponds to the total number of samples used in the real power calculation (corresponds to cell L8 in Figure 9.1). That is, only the exact number of values used in each power calculation is calculated. This was done to alleviate the PC memory requirements when running a simulation and also as a time saving device. Previous attempts at running a simulation which calculated all the sample points, even those not used in the calculation, tended to result in the program crashing and even when this did not occur, simulations took up to several minutes to complete.

Figure 9.5 is a screen capture of the "continuous" voltage and current signals calculated from the input data. These signals have not been quantised and contain no clipping. The current signal is that produced directly from the shunt and does not incorporate any gain.

File Edit View Insert Format Tools Data Window Help					
A	B	C	D	E	F
1	Actual Power	1100	voltage scale factor	2.916815472	Full Screen X Close Full Screen
2	Ave Power [int]	6647.282927	current scale factor	0.055242717	
3	Scale Factor	0.161132813	Cal Factor	1.028514556	
4	Measured Power	1101.026047			
5	Error	-0.093%	Input Cell ->	5	
6					
7			Continuous Variables		
8	Sample Number	Time [ms]	Phase Angle [deg]	Nominal-Voltage Signal [V]	Current Signal [A]
9	0	0	0	0.01	0.000215968
10	1	0.9765625	17.578125	0.639179061	0.001289681
11	2	1.953125	35.15625	1.209600399	0.002257427
12	3	2.9296875	52.734375	1.667993551	0.003024398
13	4	3.90625	70.3125	1.971550136	0.003536936
14	5	4.8828125	87.890625	2.091921635	0.003728062
15	6	5.859375	105.46875	2.017866804	0.003597465
16	7	6.8359375	123.046875	1.75630147	0.003147866
17	8	7.8125	140.625	1.331652675	0.002439036
18	9	8.7890625	158.203125	0.783577487	0.001496561
19	10	9.765625	175.78125	0.163259507	0.000449021
20	11	10.7421875	193.359375	-0.471371059	-0.000636416
21	12	11.71875	210.9375	-1.061047384	-0.001635777
22	13	12.6953125	228.515625	-1.550700822	-0.002469766
23	14	13.671875	246.09375	-1.894603658	-0.003037177
24	15	14.6484375	263.671875	-2.060639521	-0.003317201
25	16	15.625	281.25	-2.033302668	-0.003265544
26	17	16.6015625	298.828125	-1.81514603	-0.002884267
27	18	17.578125	316.40625	-1.426542802	-0.002232371
28	19	18.5546875	333.984375	-0.90378383	-0.00132509
29	20	19.53125	351.5625	-0.295688488	-0.000295859
30	21	20.5078125	369.140625	0.340954465	0.000785475
31	22	21.484375	386.71875	0.94669027	0.001797434
32	23	22.4609375	404.296875	1.46495052	0.002678417
33	24	23.4375	421.875	1.847335967	0.003327788
34	25	24.4140625	439.453125	2.058136432	0.0036858
35	26	25.390625	457.03125	2.077665697	0.003704983
36	27	26.3671875	474.609375	1.904099965	0.003413434
37	28	27.34375	492.1875	1.553648178	0.002808757
38	29	28.3203125	509.765625	1.059038299	0.001958199
39	30	29.296875	527.34375	0.466460917	0.000962476
40	31	30.2734375	544.921875	-0.168744401	-0.000121102
41	32	31.25	562.5	-0.787257151	-0.001171651

Figure 9.5: Continuous Signals Produced by Microsoft Simulation Worksheet

The next figure is the remainder of the same sheet and depicts the quantised signals from those in the above figure. The clipping action of the high-gain channel can be seen in cells J11 through J17, and the channel switching has forced the output signal for these samples to use the low-gain channel with added gain as seen in cells K11 through K17.

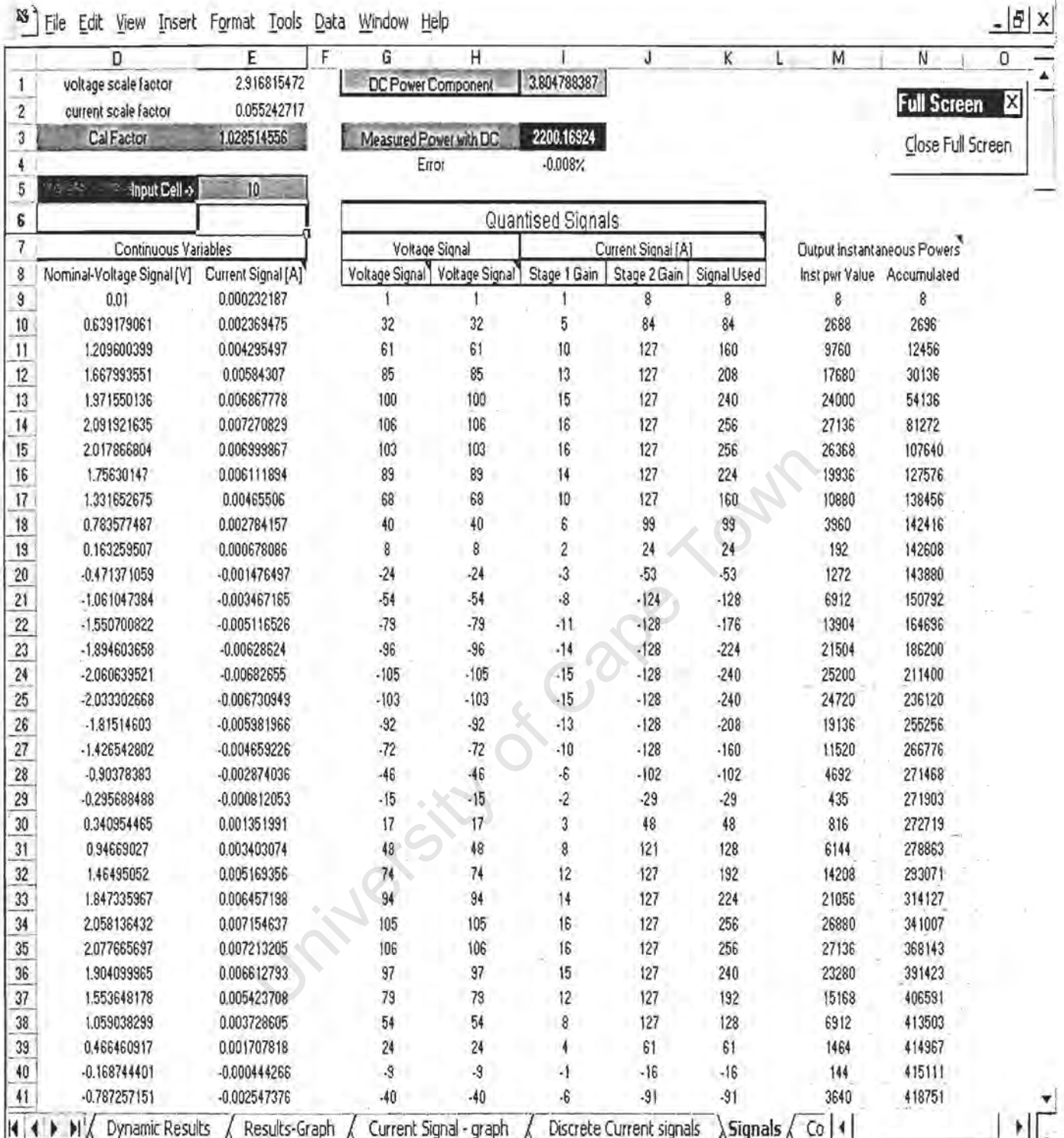


Figure 9.6: Discrete Signals Produced by Microsoft Excel Simulation Worksheet

## 9.4 Simulation Macros

Several Excel macros were created to conduct the various calibration and output accuracy calculations. The running of these macros constitutes the mathematical "simulation" of a system test for the stipulated input parameters.

The three main macros are:

- **Calibration Macro** - This macro is run on the Signals worksheet and automatically calculates the calibration factors for the real power and energy outputs according to the nominal values of current, voltage, power factor, frequency, etc. Thus the system is calibrated to provide an optimal accuracy for a single set of nominal inputs. Due to the injected random noise and quantisation action, actual calibration factors differ with each running. This macro may be called using the shortcut key "ctrl+f" and must be called before running either of the two macros listed below.
- **Current Sweep Macro** - In order to assess the system accuracy over a full dynamic range, the input current must be swept between the absolute minimum and maximum values. Using a pre-defined function called Scenario, this macro sweeps the input current according to a logarithmic scale using 28 automatically calculated points. Scenario allows the use of lookup tables when using formulas with varying inputs. Each output error arising from the input points is stored and displayed graphically. All other input parameters are held static, thus allowing the user to alter several input parameters, including the power factor unlike the macro described below, and still conduct a full current sweep.

Due to the injected random noise and the empirical nature of the quantisation mechanism used in the signal production, each running of this macro produces a unique set of results, even when all the input parameters remain unchanged. Thus in order to obtain an indication of the worst-case scenario regarding output accuracies, the macro should be run several times. This macro may be called using the shortcut key "ctrl+c".

- Full Sweep Macro - This macro calls the current sweep macro described above with several different parameters injected with each call. In order to obtain a more complete assessment of the system Shaun Parsens of the Technical Design from EML advised the calculation of output errors for three different power factor settings over the full dynamic load current range. The three power factors suggested were 0.5 leading, unity, and 0.5 lagging. Furthermore, the offset values for the input current signal should be variable according to the actual signal being sampled. A relatively low offset may be assumed at low load currents, whilst heavier loads are assumed to contain a higher offset. Thus the offset is swept from 50% to 150% of the user defined injected offset (cell F23 in Figure 9.1) along with the swept current values.

Three sets of outputs are thus recorded, one for each power factor, and are displayed graphically (refer to Figure 9.7). This macro may be called using the shortcut key "ctrl+s".

## 9.5 Results

Various system parameters were tested using the above macros in order to determine their optimal values. The resulting values are listed in the paragraph below. The factors considered in defining the optimal setup include accuracy, dynamic range, processor overhead, resolution, and system response speed. Where applicable, either the stipulated or optimal value found is listed in parenthesis.

### Ambient Parameters

- Meter Constant (4000 impulses per kWh – maximum allowable)
- Power factor (swept between 0.3 lagging and 0.3 leading)
- Line frequency (50 Hz nominal)
- Ambient temperature (23<sup>0</sup> C nominal, swept between -20<sup>0</sup> C and 40<sup>0</sup> C)

### Line Voltage Parameters

- Nominal RMS line voltage (220V)
- Minimum line voltage according to IEC1036 specifications (20% below nominal corresponding to 176V)

- Maximum line voltage according to IEC1036 specifications (20% above nominal corresponding to 264V)
- Voltage division ratio corresponding to the voltage channel input resistor divider network (3/448 – optimal value)
- Injected offset (DC) voltage, possibly arising from large rectified loads such as burst-fired controlled heating elements common on household cooking stoves (swept between 0.01mV and 0.1mV on ADC input voltage)

#### Load Current Parameters

- Nominal RMS load current (20A)
- Minimum load current according to IEC1036 specifications (0.4% of nominal - corresponding to 0.08A)
- Maximum load current according to IEC1036 specifications (400% of nominal - corresponding to 80A)
- Shunt resistance (500  $\mu\Omega$ )
- Injected differential evenly distributed random noise on ADC current input signal (0.01mV peak)

#### Current Channel Dual Gain Parameters

- Stage 1 gain (44)
- Stage 2 gain (16 – corresponds to sampled value shifted by 4 bits)
- Stage 1-to-2 crossover point (120 - discrete sampled decimal point of stage 1 gain sampled value)

#### Sampling Parameters

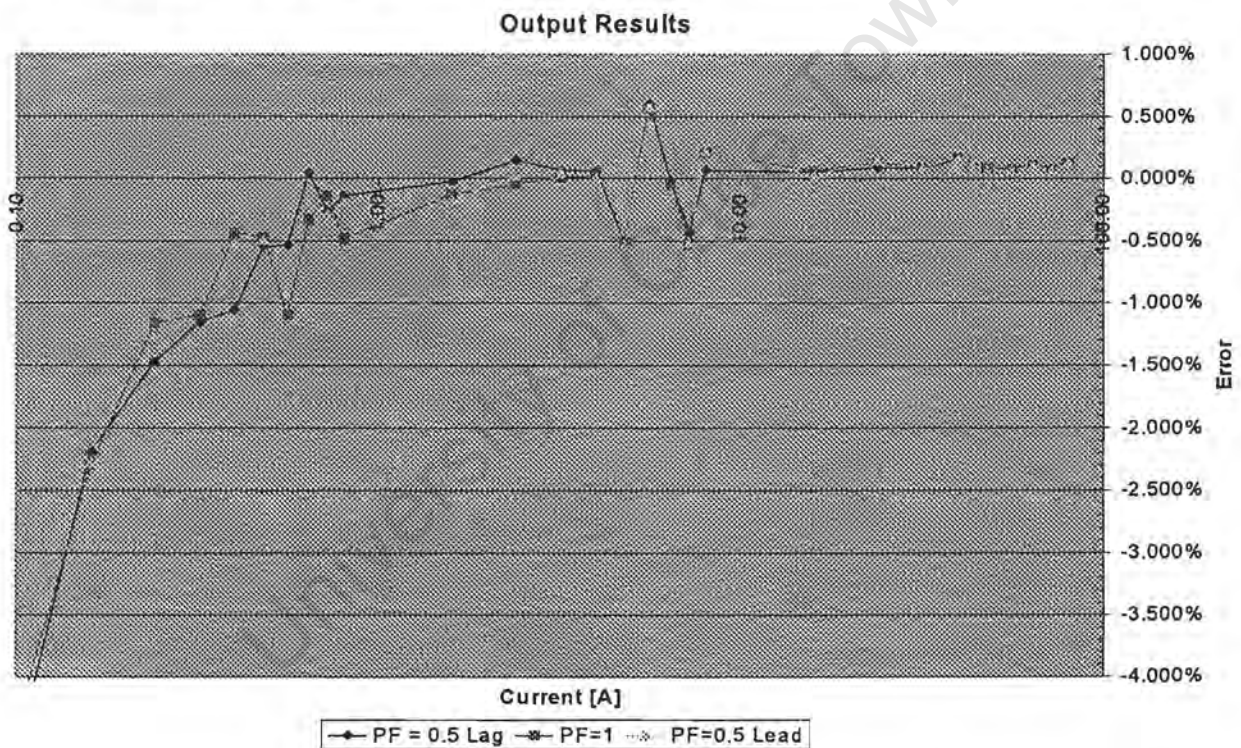
- Sampling resolution (8-bits for Motorola MCU and 10-bits for Hitachi MCU)
- Number of samples per (mains) fundamental cycle (40.96)
- Number of fundamental cycles incorporated (10 – corresponds to an effective sampling frequency of 2048 Hz)
- ADC conversion time (16 $\mu$ s for Motorola MCU)

## Operational Amplifiers Parameters

- Inherent offset voltage ( $20\mu\text{V}$  for MSP407)
- Offset temperature coefficient ( $1\mu\text{V}/\text{deg}$  for MSP407)

The responses depicted in the three examples below represent the output error over a dynamic range of 1000 to 1 measured from 28 points along a logarithmic scale. All system attributes were set to the optimal values described above, except where indicated.

In the example below all parameters were set to their optimal values as described above with an 8-bit ADC assumed.



**Figure 9.7: Optimal System Simulated Results**

As may be seen the system does not exhibit results consistent with a class 2 meter [17]. The error crosses the  $-2\%$  level for all three values of power factor towards the lower end of the scale. The sharp drop-off at lower currents is mostly attributed to the injected DC offset in the current signal and the inherent offset present in the operational amplifiers. This was confirmed by removing the offsets altogether and observing the result which did not cross the  $\pm 3\%$  level and only crossed the  $\pm 2\%$  level at the two lowest current settings. Furthermore, a negative DC signal was

injected into the current signal, as opposed to the above example, and the error *rose* beyond the +2% level. This is consistent with the measured signal containing a slight negative offset.

The slight oscillatory nature of the output error between 4A and 8A is attributed to the switching action between the two channels. In this current range both the two current channels' signals constitute the final current signal significantly and the constant switching between them increases the output error. Due to the secondary gain of 16, any random interference in the low-gain channel, which may toggle the least significant bit erroneously, will effectively cause 16 times as much error as an erroneously toggled bit in the high-gain channel. This problem is greatly reduced outside of the 4A to 8A current range due to the averaging nature of the power calculation. Outside of this range, the final signal is mainly derived from either the low or high gain channels, and thus the quantisation error is averaged out to a greater extent.

The error introduced through this switching action, hereafter referred to as the switching error, is a trade-off against a lessened operating range. A single gain system offers more consistent accuracies at the cost of a smaller operating range. For this reason it was decided not to introduce further gain channels.

Figure 9.8 depicts the results obtained using a 10-bit ADC, as opposed to the 8-bit version used above, with all other parameters set the same.

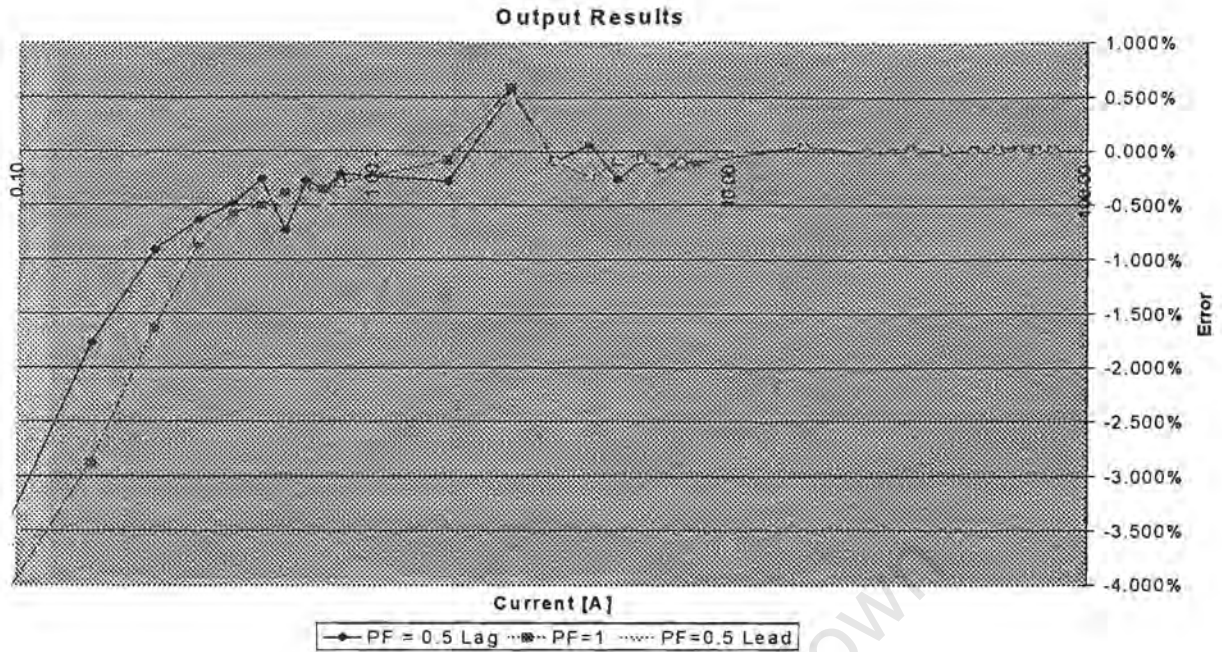


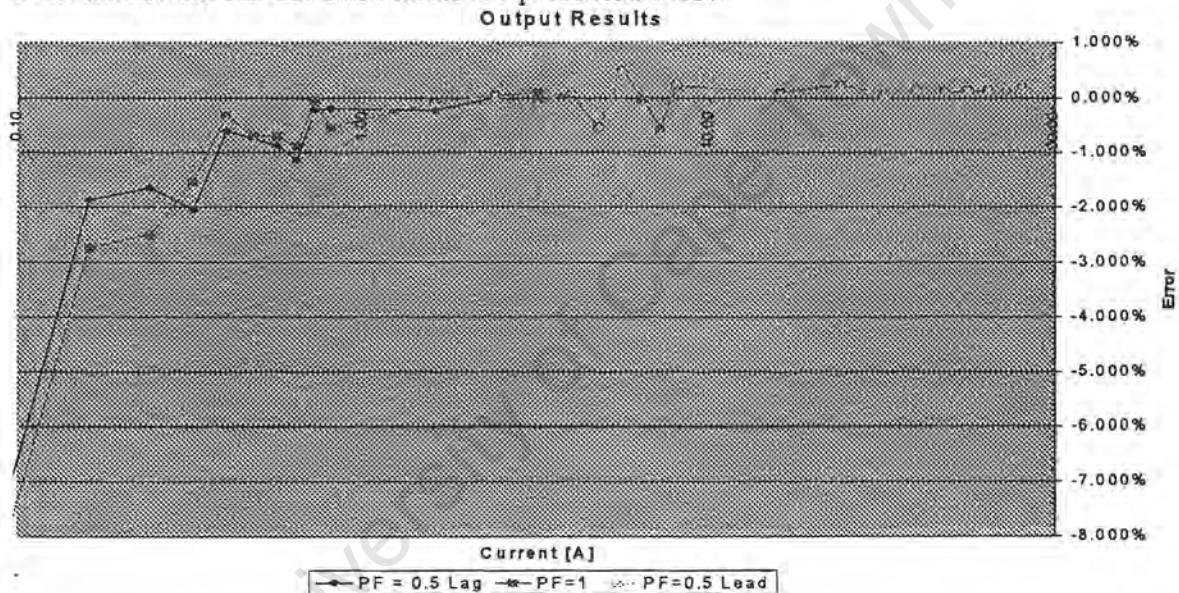
Figure 9.8: Optimal System Simulated Results with a 10-bit ADC

Surprisingly, the 10-bit version did not exhibit considerably better results as expected. The drop-off towards the lower end of the scale is still substantial, however the middle to upper range results are slightly improved. The switching error is lessened and has also shifted between 2A and 4A. This is expected since the extra gain of 16 constitutes just 4 bits shift out of 10 for each sample, as opposed to the 4 out of 8 for the first example. Thus the low-gain and high-gain samples for the 10-bit ADC retain a closer hamming distance than the 8-bit version. Unfortunately, the 10-bit ADC system does not conform to IEC1036 specifications for a Class 2 meter, as the error still drops below the -2% level. Several attempts made to reduce the error to within class 2 limits, by increasing the sampling frequency, increasing the secondary gain, and reducing the injected offset to just 5mV on the ADC input signals, all failed.

The sampling frequency of 2048 Hz in the first example above corresponds to a Nyquist frequency of around 1kHz. IEC1036 specifications<sup>[17]</sup> stipulate the system must include frequencies up to the 5<sup>th</sup> harmonic before any signal attenuation is introduced by the peripheral circuitry. Thus on a 50Hz system, all signals up to and including 250Hz must be included. Theoretically, a sampling frequency with a Nyquist frequency just beyond this value should include the required harmonics, however simulated testing showed that reducing the sampling frequency results in an

increased overall error. Thus the sampling frequency should be kept as high as possible, but with a trade-off between accuracy and processor overhead. Furthermore, the attenuation and phase shift introduced by the anti-aliasing filters must also be considered. With an absolute minimum sampling frequency of at least 500Hz required to include the 5<sup>th</sup> harmonic, no room exists for the anti-aliasing filter attenuation. Consequently it was decided that the minimum allowable sampling frequency should be at least 1kHz, thus leaving between 250Hz and 500Hz for the anti-aliasing cut-off frequency.

In light of this, the system was tested using a 1024Hz sampling frequency on both the 8-bit and 10-bit ADCs. The results are presented below



**Figure 9.9:** Simulated Results using an 8-Bit ADC and a 1024Hz Sampling Frequency

As may be seen the drop-off is increased substantially and the switching error to a lesser extent.

The results from using a 10-bit ADC with a reduced sampling frequency, presented as Figure 9.10, are just marginally worse than those depicted in Figure 9.8 (increased frequency). It was anticipated that this difference would be far more appreciable using an actual hardware system.

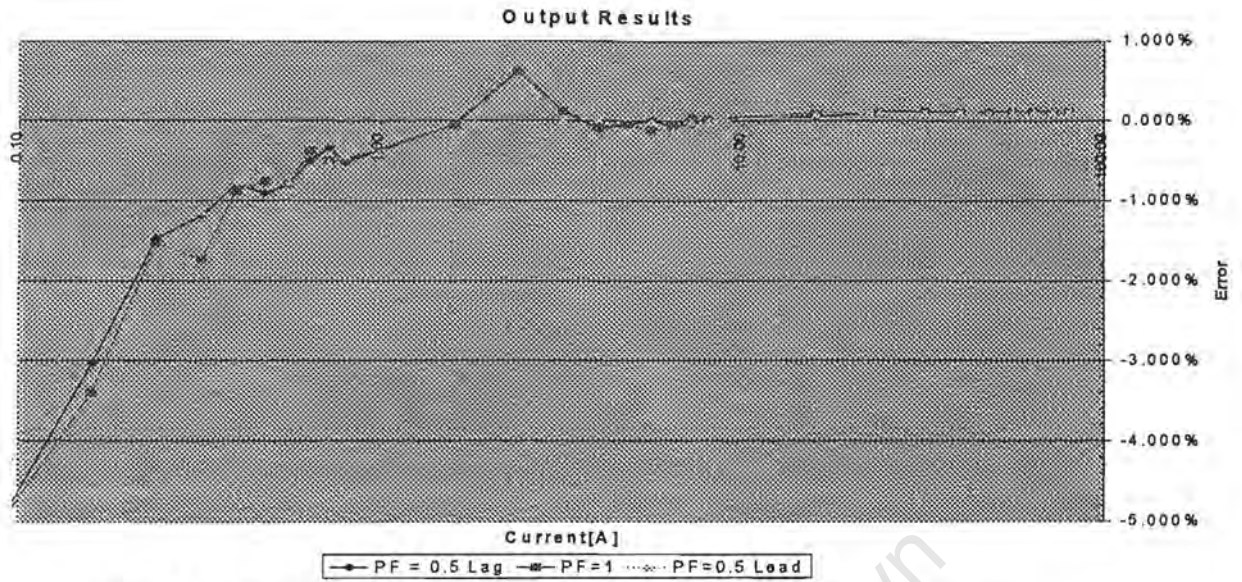


Figure 9.10: Simulated Results using a 10-Bit ADC and a 1024Hz Sampling Frequency

The validity of all the simulation results is further discussed in Section 10.

## Section 10: Testing Results

Hardware tests were performed on the developed system described in Section 8 with several parameters set using the results of the simulations discussed in Section 9. As mentioned before, one of the main limitations of this project is the lack of appropriate testing equipment. The necessary equipment required to create loads with the required power factor, injected offset and various noise conditions was not available.

Furthermore, the range of (stable) loads required to test both the upper and lower operating range of the system was not sufficient. Consequently, simulated input signals were created and used to test the developed system over these conditions. Several of the results obtained using simulated inputs are presented in Section 10.1. Some of these were later verified using actual loads and these results are presented in Section 10.2.

All relevant test equipment specifications are provided as Annexure C.

### 10.1 Tests Using Simulated Inputs.

The simulated signals were created using two Hewlett Packard Digital Signal Generators (model number HP 33120A). The output frequency of each of these was verified using the generators themselves. The injected white-band noise amplitude and any injected DC offsets were also created using these devices.

The relative phase difference between the signals was monitored using a Hewlett Packard Digital Oscilloscope (model number HP 54620). All (purely sinusoidal) signals were measured using a true-RMS calculating Hewlett Packard Digital Multimeter (model number HP 34401A). Unfortunately, it was not possible to phase-lock the two signal generators for a unity power factor simulation and this had to be performed manually using the oscilloscope as a reference. Consequently, a marginal phase error may have been introduced.

The output frequency generated by the final OCM solution was measured using a second oscilloscope (the same model as described above).

### 10.1.1 *Scope of Simulated Testing*

The hardware-based simulations were conducted using 19 test points over a (simulated) current range from around  $800\text{mA}_{\text{rms}}$  to  $80\text{A}_{\text{rms}}$  with the voltage signal held at a constant  $240\text{V}_{\text{rms}}$ . This current range was also tested for 0.5 lagging and 0.5 leading power factors using 10 test points.

Although IEC1036 specifications specify a current range starting from  $0.08\text{A}_{\text{rms}}$  [17], the relatively small voltage signals required to simulate currents as low as these could not be created or measured accurately using the available test equipment. Using a  $500\mu\Omega$  shunt a  $0.08\text{A}$  current would produce a voltage signal of just  $40\mu\text{V}_{\text{rms}}$ . An attempt made to create a signal of this magnitude using a high-precision resistor voltage divider did not produce a stable signal. The relatively strong ambient noise present in the laboratory tended to overshadow the signal, even with the appropriate EMI shielding and grounding. Introducing a second low-pass filter in conjunction with the anti-aliasing filter did *not* attenuate this noise substantially, since this contained a strong 50Hz component. Furthermore, the True-RMS meter's reading tended to fluctuate wildly when trying to measure these signals. Unfortunately, as seen in Section 9, the highest errors occur in the lower current range. In light of all this, the system error cannot be verified beyond the tested points.

In conjunction to sweeping the current and power factor, testing was also performed on signals containing offsets and white-band noise, however these had little effect on the output.

Outputs from the tests were obtained using the test version of the developed OCM solution, which uses an opto-isolated serial link between the MCU and the host PC as described in Section 8.4. The data uploaded from the MCU includes 10 sets of 41 samples of current and voltage for each point tested, and the corresponding average power information. The final OCM version, which outputs a digital frequency directly proportional to the measured energy, was not tested as the relatively low output

frequencies could not be measured accurately enough with the available equipment. However, should EML pursue the OCM project further, the frequency generation will be formally tested using the appropriate equipment present at the company's premises.

### 10.1.2 Sampled Signals

The signals captured by the MCU were uploaded and displayed graphically in order to assess the extent of influences such as offsets, noise and phase errors. The figures below were created from actual (unedited) data uploaded from the MCU and depict the real values utilised in the real power calculation performed on-board.

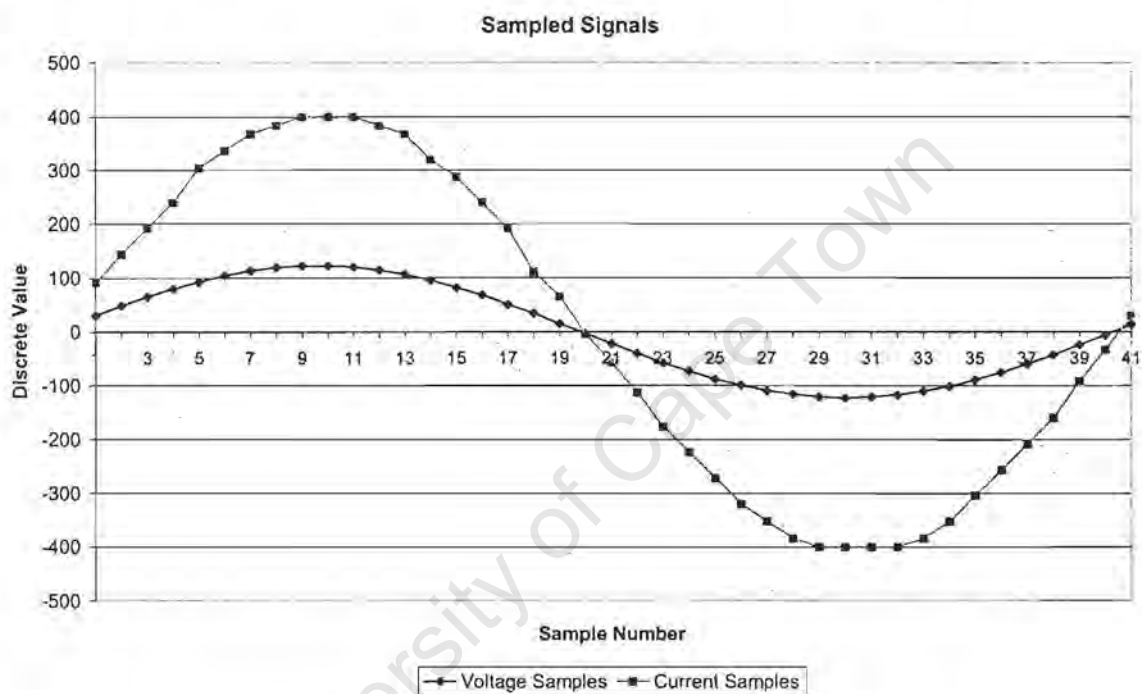


Figure 10.1: Sampled signals using simulated inputs

The signals presented in Figure 10.1 were measured using a simulated current of  $16.038A_{rms}$  and a  $240V_{rms}$  voltage with a unity power factor and no injected noise or offset.

As may be seen, the sampled signals are purely sinusoidal and in phase, indicating that the anti-aliasing filters are closely matched. No indication of mirror frequencies due to aliasing exists, confirming the cut-off frequency of the anti-aliasing filters is sufficient. The current signal does not appear to contain any significant offset, indicating that offset nulling is sufficient.

As a comparison, the below figure depicts a measurement taken at a slightly higher current rating before the offset nulling was included in the signal processing hardware.

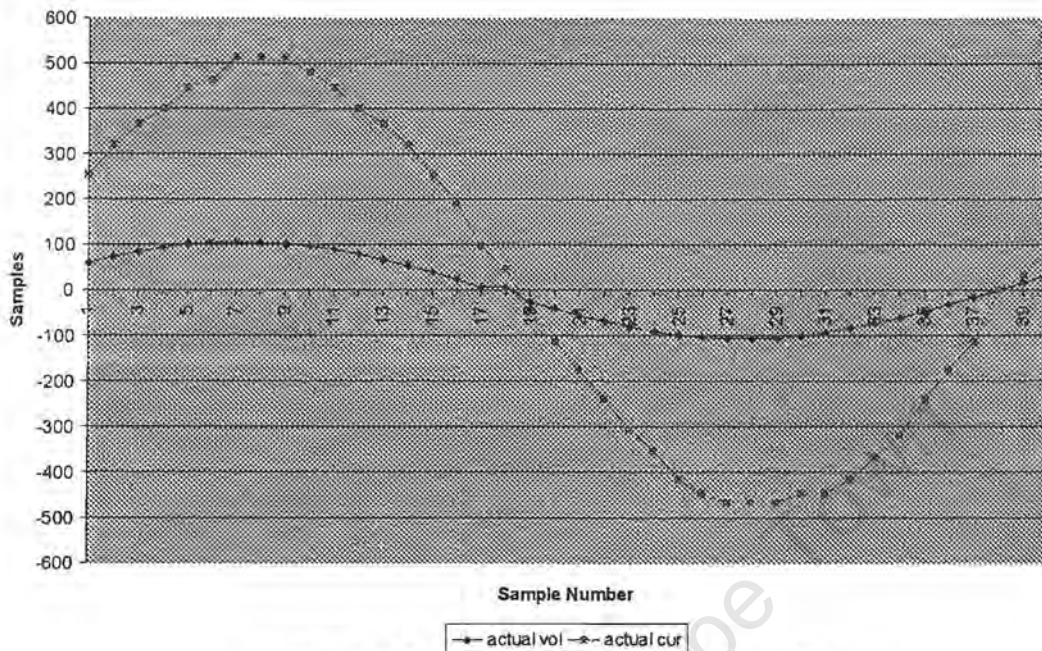


Figure 10.2: Sampled Signals without offset nulling

As is easily seen, a considerable offset is present in the current signal. Several tests were performed over varying power factors before the offset nulling was included and the results of these were found to be highly erroneous. A 16.2% error was calculated for a current of 1.61A at unity power factor, and even worse errors were found for non-unity cases.

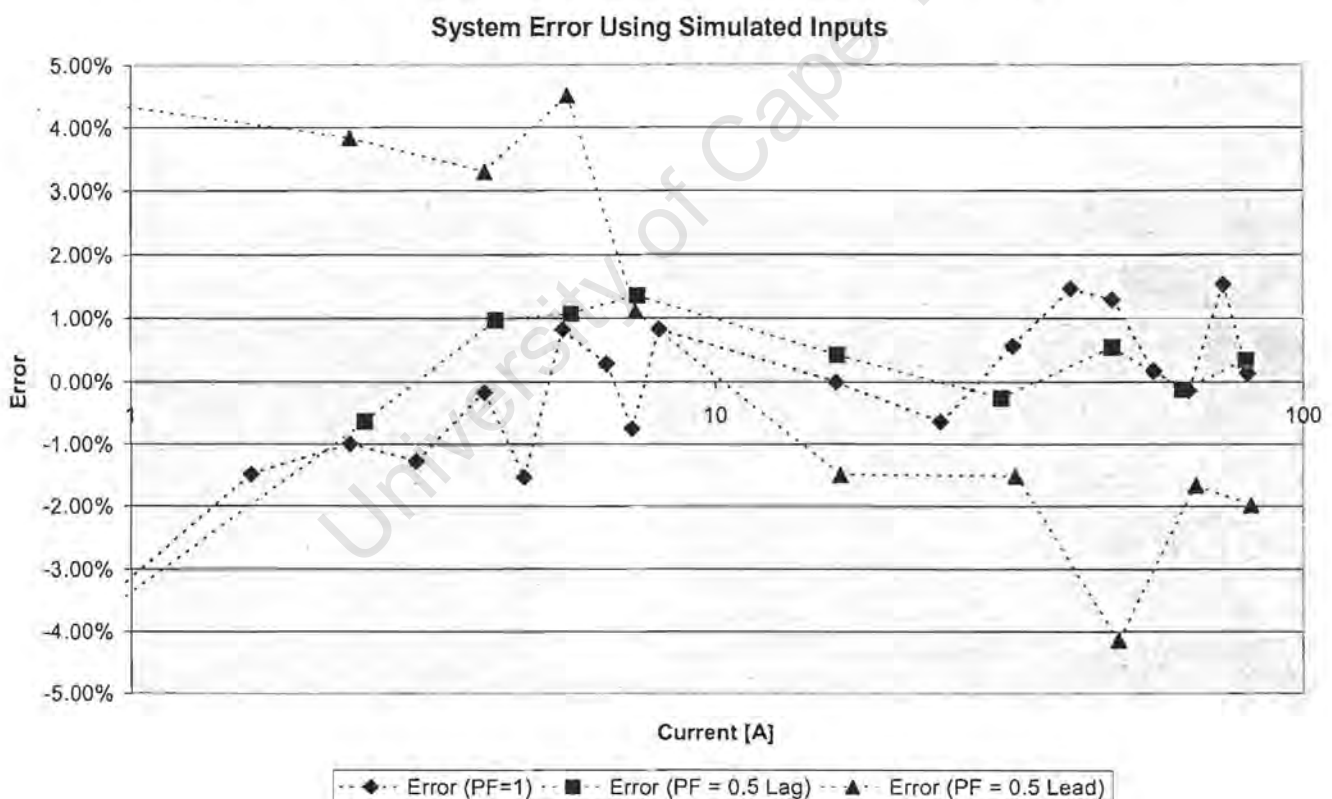
Several tests were performed to determine the sources of the offset. The signal generator was directly connected to the oscilloscope and the observed output was observed to contain a small offset at low signal values. The oscilloscope itself also contains an inherent offset, measured using an independently tested DC signal (using the multimeter described above). This was subtracted from the offset observed from the output of the signal generator. Over an average of 10 readings, it was concluded that the signal generator introduced a varying offset between  $4\mu\text{V}$  and  $200\mu\text{V}$  for outputs between  $0.8\text{mV}_{\text{rms}}$  and  $4\text{mV}_{\text{rms}}$ . This corresponds to roughly 5% of the signal's RMS value. This offset is amplified by the two operational amplifiers and influences the sampled signals considerably as seen in Figure 10.2. Furthermore, the operational amplifiers introduce offsets as well, however the effects of these are marginal. The

manufacturer stipulated offset of  $200\mu\text{V}_{\text{max}}$  constitutes just 1.05% of the maximum 19mV resolution of the 8-bit Motorola MCU ADC. To confirm this, a known DC signal, measured using the multimeter described above, was fed to the two amplifiers and the high-gain output measured to contain an offset of around  $902\mu\text{V}$ , corresponding to 4.7% of the maximum resolution of the 8-bit MCU ADC. Thus for the example in Figure 10.1, the offset introduced by the operational amplifiers constitutes about 0.8% of the sampled signal amplitude.

In light of this, it was decided to include the offset nulling circuitry described in Section 8.4 at the cost of an increased component count. The resulting outputs were found to be far more accurate, as shown in Section 10.1.3.

### 10.1.3 Output Results

The complete set of output errors obtained using simulated inputs is depicted below.



**Figure 10.3:** System output error using simulated inputs

The current is presented on a logarithmic scale for convenience and the three curves have not been assumed to be continuous due to their distinct fluctuating nature and are therefore depicted as dotted lines.

For reasons stated above, the full current range could not be tested. The first set of data points taken at around 800mA are not displayed in Figure 10.3 since the multimeter's resolution was found to be inadequate when measuring low current signals of around  $400\mu\text{V}_{\text{rms}}$ . Thus the relatively large errors associated with them are not verified and are attributed to measurement error rather than system error. The dotted curves to these points have been included however to indicate the sharp increase in error observed at low current values.

The complete set of output details for the unity power factor measurements are presented in Table 10.1.

	Cal Factor	0.003782856				
	Line Voltage [Vrms]	240				
UNITY POWER FACTOR						
Point	Load Current [Arms]	Shunt Voltage[mVrms]	Recorded Summed Value	PF = 1	Actual Power [W]	Error (PF=1)
0	0.836	0.418	55038	208.2008129	200.64	-3.77%
1	1.6	0.8	103012	389.6795331	384	-1.48%
2	2.36	1.18	151218	572.0358757	566.4	-1.00%
3	3.06	1.53	196609	743.7434795	734.4	-1.27%
4	4.02	2.01	255490	966.481807	964.8	-0.17%
5	4.7	2.35	302752	1145.267134	1128	-1.53%
6	5.48	2.74	344820	1304.404308	1315.2	0.82%
7	6.52	3.26	412509	1560.462029	1564.8	0.28%
8	7.22	3.61	461491	1745.753867	1732.8	-0.75%
9	8.028	4.014	505111	1910.762034	1926.72	0.83%
10	16.038	8.019	1017517	3849.12	3849.12	0.00%
11	24.16	12.08	1542596	5835.418097	5798.4	-0.64%
12	32.04	16.02	2021302	7646.293825	7689.6	0.56%
13	40.14	20.07	2509374	9492.599781	9633.6	1.46%
14	47.2	23.6	2955915	11181.79995	11328	1.29%
15	55.6	27.8	3521481	13321.25453	13344	0.17%
16	63.8	31.9	4053190	15332.63296	15312	-0.13%
17	72.96	36.48	4557705	17241.14041	17510.4	1.54%
18	80.26	40.13	5085085	19236.14286	19262.4	0.14%

*Table 10.1 Output results using simulated inputs*

The shunt's output voltage was simulated using the above mentioned function generator and a resistor divider. This was measured using the multimeter and the equivalent load current was calculated from the stipulated shunt impedance of  $500\mu\Omega$ .

The system was calibrated at the 10<sup>th</sup> current point of 16.038A using the Excel iteration function called Goalseek, hence the zero output error for this point.

It should be noted that the lowest current simulated of 0.836A cannot be accurately verified due to the low resolution of the testing equipment below 500 $\mu$ Vrms.

The full set of results for the non-unity tests is presented in Table 10.2.

LAGGING POWER FACTOR					Power Factor	0.477
Point	Load Current [Arms]	Shunt Voltage[mVrms]	Recorded Summed Value	Recorded Power	Actual Power [W]	Error (PF = Lag)
0	0.826	0.413	25984	98.29372294	94.56048	-3.95%
2	2.5	1.25	76138	288.0190685	286.2	-0.64%
4	4.2	2.1	125871	476.1518319	480.816	0.97%
6	5.66	2.83	169451	641.008684	647.9568	1.07%
8	7.36	3.68	219705	831.1123152	842.5728	1.36%
10	16.116	8.058	485654	1837.15901	1844.95968	0.42%
12	30.6	15.3	928451	3512.196173	3503.088	-0.26%
14	47.22	23.61	1421234	5376.323161	5405.7456	0.54%
16	62.2	31.1	1884614	7129.222843	7120.656	-0.12%
18	79.74	39.87	2405065	9098.013884	9128.6352	0.34%
LEADING POWER FACTOR					Power Factor	0.543
Point	Load Current [Arms]	Shunt Voltage[mVrms]	Recorded Summed Value	Recorded Power	Actual Power [W]	Error (PF = Lead)
0	0.84	0.42	27658	104.6262234	109.4688	4.42%
2	2.358	1.179	78120	295.5166886	307.29456	3.83%
4	4.02	2.01	133924	506.615169	523.8864	3.30%
6	5.58	2.79	183571	694.4226067	727.1856	4.51%
8	7.32	3.66	249368	943.3231643	953.9424	1.11%
10	16.33	8.165	570921	2159.711768	2128.1256	-1.48%
12	32.32	16.16	1130171	4275.273828	4211.9424	-1.50%
14	48.42	24.21	1737115	6571.255408	6310.0944	-4.14%
16	65.6	32.8	2297450	8690.921866	8548.992	-1.66%
18	81.436	40.718	2860910	10822.40975	10612.73952	-1.98%

**Table 10.2: Output results for non-unity power factor using simulated inputs**

The recorded summed values for each test point were calculated as the average instantaneous power over ten fundamental cycles to provide real average power information. The oscilloscope's phase-difference calculation function is performed using data sampled from an 8-bit ADC and is thus only accurate to around 1.4°. In light of this, the two power factors of 0.477 Lagging and 0.543 Leading were calculated from the sampled data, respectively, as shown on the next page.

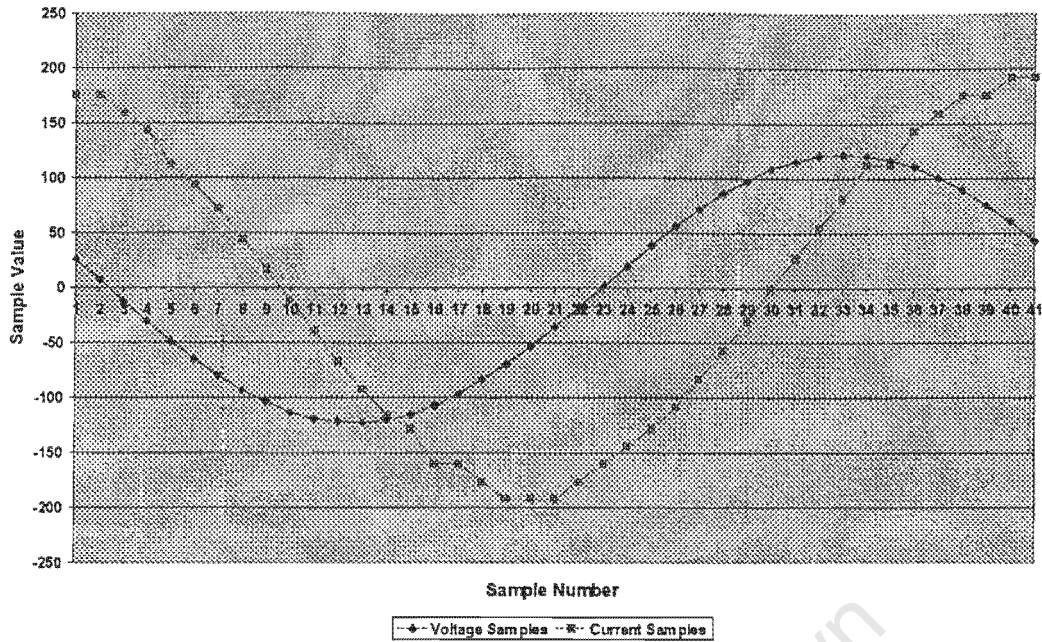


Figure 10.4: Non-unity power factor signals

Since no offsets appear in either of the two signals, the zero-crossing points were used as a reference. In the example above, the rising current waveform crosses at around sample 30 and the voltage at sample 23. The power factor for this example is thus:

$$PF = \cos(\phi), \text{ where } \phi = 360 \times \frac{\Delta t}{T_s}$$

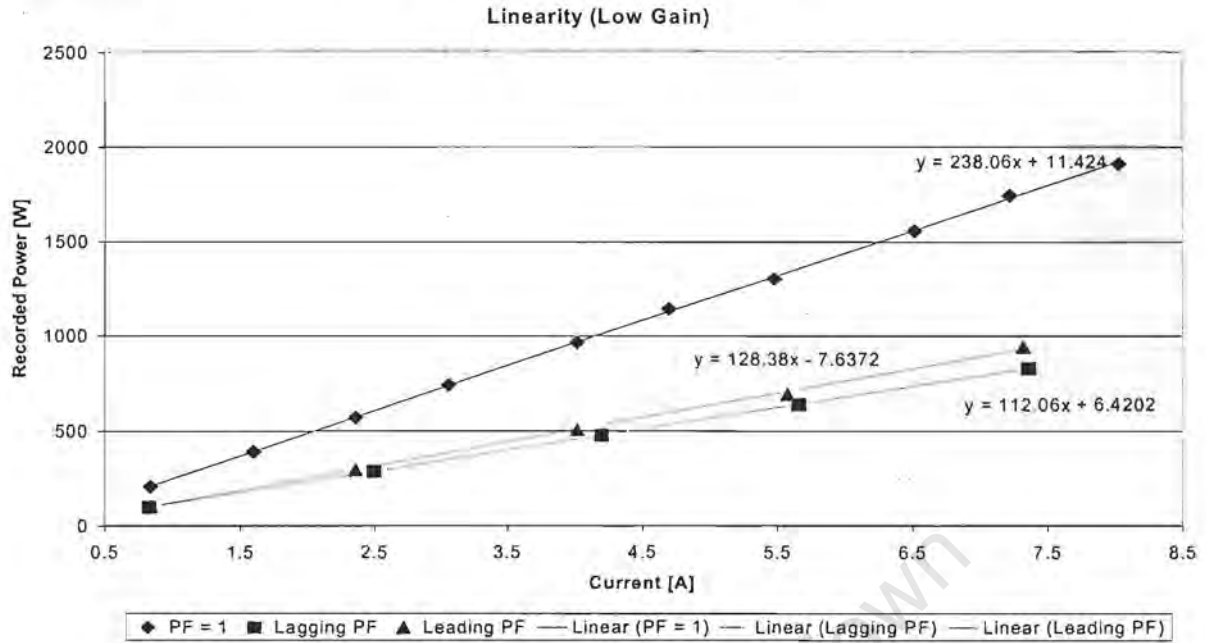
, and  $\Delta t$  is the time difference between the two zero-crossing points and  $T_s$  is the fundamental period. With a sampling rate of 2048 Hz and a 50Hz fundamental frequency, the 7 sampling points difference between the two signals corresponds to a power factor of:

$$PF = \cos\left(360 \times \frac{7}{2048} \times 50\right) = \cos(61.52) = 0.477 \text{ (lagging)}$$

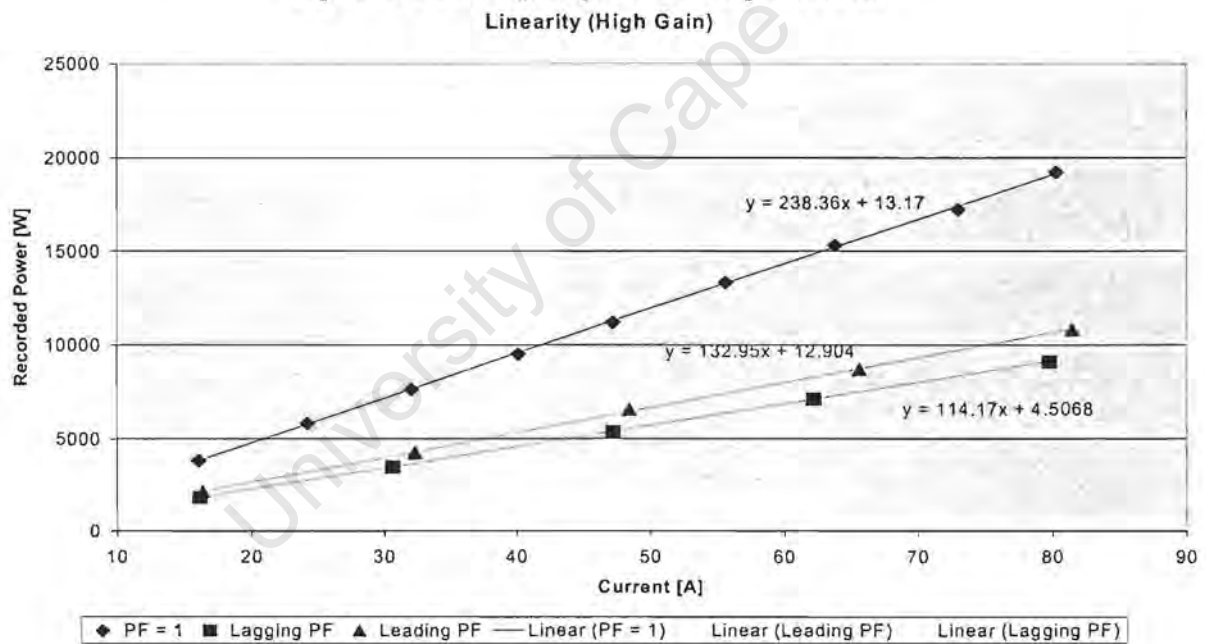
The leading power factor was calculated in the same way.

This method of determining the true power factor is valid since no noticeable phase difference appeared in the unity power factor case (refer to Figure 10.2 as an example), and the voltage and current sampling is thus assumed to be concurrent.

In conjunction to the output error comparison, a linearity analysis was also performed on the output data. This is displayed in the two graphs below.



**Figure 10.5: Linearity comparison - Low gain channel**



**Figure 10.6: Linearity comparison - High gain channel**

Optimal linear trendlines calculated by Excel were added and the resulting equations are shown. The linearity of the two channels is clear, however the slopes of the three curves do not match exactly between the two gain channels, particularly for the non-unity power factor cases. This confirms that the secondary gain was not exactly 16, and even more so for the leading power factor case, whose slopes jump from 128 to 133.

#### 10.1.4 Possible Sources of Error

As shown in Table 10.2 and in Figure 10.3, the error at several leading power factor test points was considerably larger than anticipated in the simulations discussed in Section 9. Since these errors mostly pertain to low currents, the signals of which are sampled by the high gain channel, it was anticipated that the secondary gain had altered. This was confirmed once the linearity test between the two gain channels was performed.

Later testing showed that the gain was closer to 15.8, producing the sharp increase in error at lower currents. Unfortunately the gain of the secondary amplification is highly sensitive, even using a multiturn potentiometer. This proved to be the most problematic aspect of the OCM design.

Another main source of error is attributed to quantisation, particularly for the *larger* currents. This is because toggling the least significant bit of a high gain sample effectively increases or decreases the sample by 16, as opposed to 1 for a low gain sample. This has an even greater effect at the crossover point between the two channels. Consequently, the fluctuations most apparent between 4A and 8A in Figure 10.3 are attributed to switching error, as described in Section 9.

The last current simulated of around 80A produced a slight clipping (around three samples per half period) in the current channel, indicating that the first stage gain was slightly higher than 44, however the errors at this current were still within the  $\pm 2\%$  limit.

Other sources of error include random noise (below the anti-aliasing filter cut-off frequency of 800Hz) and measurement error.

## 10.2 Testing Using Actual Loads

The operating range tested using actual loads was significantly smaller than that simulated, due to a limited range of loads available. Testing was completed using purely resistive and resistive-inductive loads on both the final and test versions of the OCM solution. A three-phase load box, with two phases placed in parallel, was used to create the resistive loads, and the inductive loads were created by adding a variable power inductor in series. These two load configurations used are depicted below.

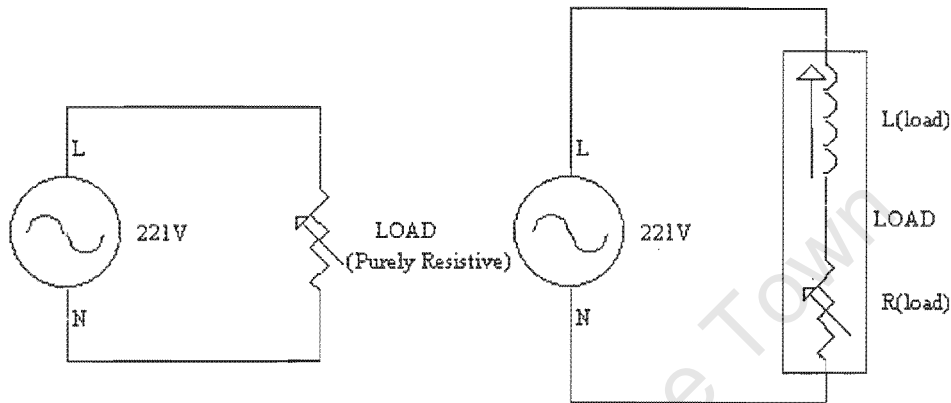


Figure 10.7: Load configuration used for testing purposes

The load box was comprised of 16 separate heating elements with up to four connected at a time. The rated range using this configuration is from 200W to 4.8kW in discrete intervals at a line voltage of 200V. Thus as with the simulated testing, the lower operating range, which is predicted to generate the highest errors, could not be tested. In an attempt to circumvent this, a  $250\mu\Omega$  shunt was used instead of a  $500\mu\Omega$  for which the system was originally designed. Since this has the same effect as halving the load, it was effectively possible to test system for smaller loads.

All test results were verified using a Fluke 43 Power Quality Analyzer. The relevant specifications for this device are presented in Annexure C.

### 10.2.1 Test Results

The complete set of test results using the test OCM version with actual loads is presented in Table 10.3. The unusual range of power factors is partly due to the limited inductance available, but mostly due to the discrete values of the resistive loads. The power factor was varied by adjusting the ratio between the resistive

impedance and the inductor impedance, thereby changing the ratio between the active and reactive powers. It was thus difficult to attain a full range of loads for a constant power factor and vice versa. In addition, the results for all the non-unity power factor tests cannot be accurately verified for reasons explained in Section 10.2.2.

		Cal. Factor				
		Line Voltage [Vrms]	7.08394E-06			
			221.1			
Result Number	Power [kW]	Power Factor	Current [Arms]	summed value	Recorded power [kW]	Error
1	0.12	1	0.542740841	16797	0.11898902	0.84%
2	0.23	1	1.040253279	33015	0.233876436	-1.69%
3	0.37	1	1.673450927	53155	0.376547083	-1.77%
4	0.47	1	2.125734962	67855	0.480681071	-2.27%
5	0.72	1	3.256445047	102768	0.728002834	-1.11%
6	0.95	1	4.296698327	134571	0.953293528	-0.35%
7	1.055	1	4.771596563	150376	1.065255275	-0.97%
8	1.165	1	5.269109	165480	1.172251176	-0.62%
9	1.4	1	6.331976481	197630	1.4	0.00%
10	1.625	1	7.349615559	227608	1.612362496	0.78%
11	1.845	1	8.344640434	261233	1.850560138	-0.30%
12	2.06	1	9.317051108	284778	2.017351617	2.07%
13	2.275	1	10.28946178	317025	2.245787583	1.28%
14	0.18	0.44	1.850252868	25973	0.183991297	-2.22%
15	0.21	0.83	1.144333099	31512	0.223229267	-6.30%
16	0.23	0.63	1.651195681	31468	0.222917573	3.08%
17	0.22	0.44	2.261420172	31483	0.223023832	-1.37%
18	0.25	0.69	1.638710269	34483	0.244275667	2.29%
19	0.42	0.59	3.219649058	57289	0.405832111	3.37%
20	0.43	0.83	2.34315825	57386	0.406519253	5.46%
21	0.43	0.66	2.946699012	58950	0.417598543	2.88%
22	0.45	0.74	2.750375885	61366	0.434713353	3.40%

Table 10.3: Results using OCM test version with actual loads

Due to the smaller range, the outputs were calibrated for a 6.33A current at unity power factor.

The unity power factor results may be compared with those obtained from the spreadsheet simulations and the test performed using simulated inputs. With a current range of between 0.54A to 10.3A a 2.07% maximum error was recorded, as compared to -3.77% for simulated inputs (however this was largely attributed to measuring error). The spreadsheet results predicted substantially lower errors than those obtained for both sets of tests, as shown in Figure 9.7.

In addition, the non-unity power factor results were considerably worse than those obtained using simulated input. Several possible reasons for this are discussed in Section 10.2.2. A graphical representation of the unity power factor results is displayed below.

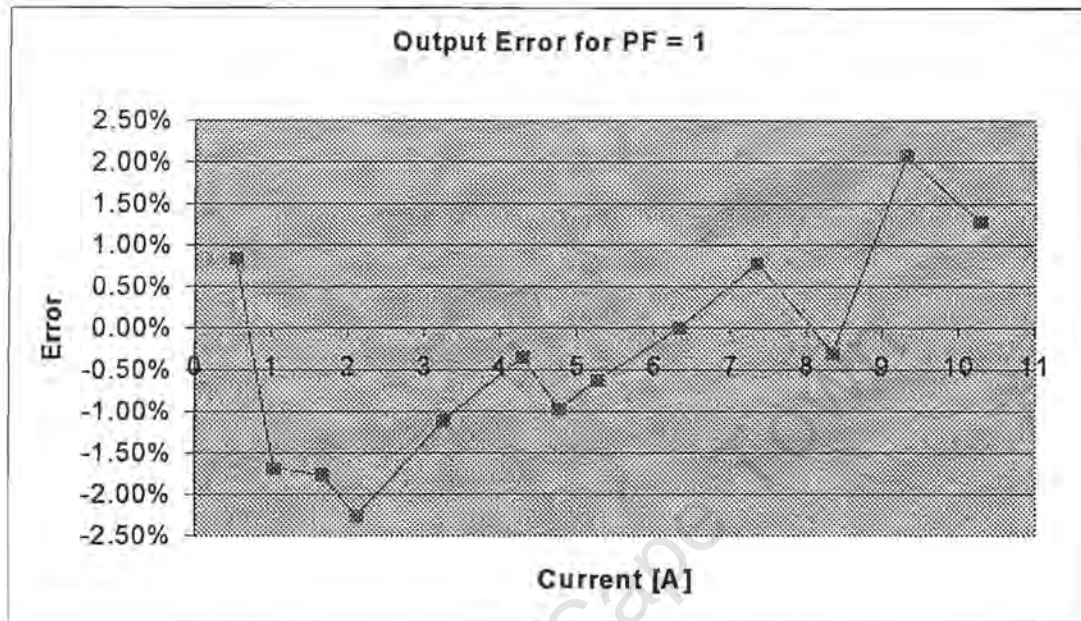


Figure 10.8: OCM test version output error for PF = 1

The final OCM version was also tested for several loads, though under unity power factor only. The corresponding results obtained are displayed below.

	Line Voltage [Vrms]	221.1				
Result Number	Power [kW]	Power Factor	Current [Arms]	Required Frequency [Hz]	Actual Frequency [Hz]	Error
1	0.12	1	0.542740841	0.132	0.139887922	5.64%
2	0.23	1	1.040253279	0.253	0.272264079	7.08%
3	0.37	1	1.673450927	0.407	0.424201791	4.06%
4	0.47	1	2.125734962	0.517	0.546749178	5.44%
5	0.72	1	3.256445047	0.792	0.816803117	3.04%
6	0.95	1	4.296698327	1.045	1.060622881	1.47%
7	1.055	1	4.771596563	1.1605	1.181280803	1.76%
8	1.165	1	5.269109	1.2815	1.297676294	1.25%
9	1.4	1	6.331976481	1.54	1.54	0.00%
10	1.625	1	7.349615559	1.7875	1.738098745	-2.84%
11	1.845	1	8.344640434	2.0295	1.982516151	-2.37%
12	2.06	1	9.317051108	2.266	2.232486778	-1.50%
13	2.275	1	10.28946178	2.5025	2.456966341	-1.85%

Table 10.4: OCM final version test results

The output frequency was measured using the HP Digital Oscilloscope mentioned before, however because of its limited resolution at low frequencies, the oscilloscope's frequency measurement output was not used. Instead, the period of each frequency was measured and the frequency calculated. All inputs were measured using the Fluke 43 as before.

As may be seen in Table 10.4, these results are considerably worse than all the results obtained before. It was anticipated that results obtained for non-unity power factors would be even worse based on the results of Table 10.3. Several possible sources of these errors are discussed in Section 10.2.2.

### **10.2.2 Possible Sources of Error**

In conjunction to the sources of error mentioned in Section 10.1.4, the largest source of error for testing conducted using actual loads is measurement error. Firstly, the Fluke 43 has a numerical display of just 3 significant figures for powers below 10kW. Since all the testing was conducted using loads much less than this, they may only be verified to three significant figures, in conjunction to the stipulated tolerance of  $\pm 4\%$  of full scale. As an example of how this effects the output error, a one unit difference in the least significant digit corresponds to a change in output error of over 5% for a power factor of 0.63 lagging (determined by altering result number 16 in Table 10.3).

Moreover, the power factor reading on the Fluke 43 has a tolerance of  $\pm 0.04$ , corresponding to up to  $\pm 8\%$  at a power factor of 0.5 (refer to Annexure C). Thus, as the power factor drops from unity so the uncertainty of the reading increases, possibly explaining the relatively poor results obtained at low power factors.

Lastly, the Fluke 43 current probe exhibits a tolerance of  $\pm 5\%$  of the displayed reading + 0.3A. Considering the relatively low currents measured, the influence of this tolerance is significant. Furthermore, all uncertainties of the actual meter itself must be added to the uncertainty of the current probe, as specified by the manufacturer (refer to Annexure C). The result is an unacceptably high overall uncertainty. Unfortunately, this meter was the best available for the purpose at hand.

Aside from these issues, another major attribution to the increase in errors may lie in the poor quality of the line voltage. The tests were performed in the University's Machines Laboratory which is situated adjacent to a computer laboratory containing over sixty computers. As a result of their switched-mode power supplies, the line voltage in the Laboratory is often severely distorted, containing several harmonics and exhibiting severe clipping. An example of a particularly poor waveform was captured using the OCM software for analysis on the first day of testing and is shown below.

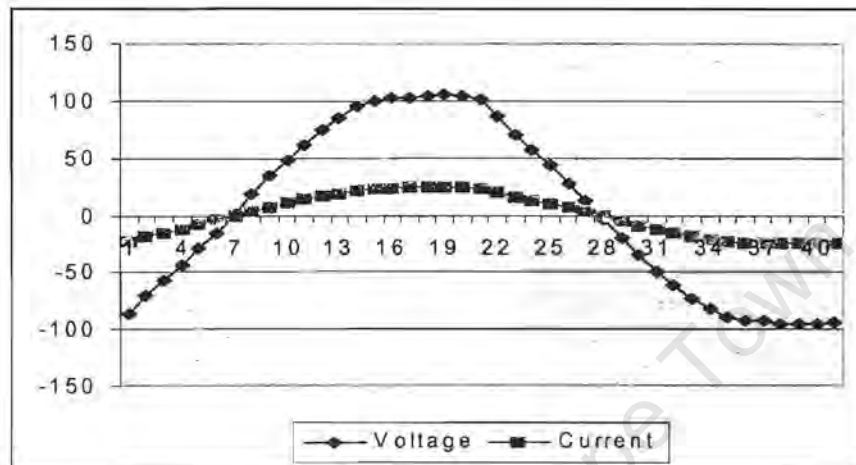


Figure 10.8: Captured Waveforms

According to the Fluke 43, the line voltage exhibited a 4.8% Total Harmonic Distortion, 99.8% of which was at the fundamental frequency, though considerable 3<sup>rd</sup>, 5<sup>th</sup> and 7<sup>th</sup> harmonics were also observed. As may be seen in Figure 10.8, and was confirmed on the Fluke 43 scope, the voltage peaks are clipped and the waveform contains a slight offset.

It should be noted, however, that the designed OCM system *should* inherently measure power correctly under these conditions, as described in Section 2 [16]. Unfortunately, this cannot be confirmed due to the relatively poor results obtained under non-unity power factor condition.

## Section 11: Main Findings

Based on research conducted during this investigation and presented in this report, the following was found:

1. Many alternative techniques for power measurement exist.
2. Many commercially available solutions for power measurement exist.
3. The measurement of power and energy has previously been accomplished using discrete digital samples of voltage and current in an MCU-based system.
4. None of the considered techniques comply with the stipulated OCM constraints
5. All the considered techniques are essentially based on the fundamental theory behind power and energy measurement.
6. All power measurement techniques using RMS values of current and voltage require considerably more processor overhead and peripheral components than is viable.
7. A firmware-intensive approach is more cost effective than a hardware-intensive approach.
8. A firmware-intensive approach is more processor intensive than a hardware-intensive approach.
9. The proposed OCM solution hardware is more cost-effective than the present CASHOPWER metering solution.
10. The processor overhead required to implement the proposed OCM solution is minimal.
11. Simulations of the proposed OCM solution yield output errors not compliant with IEC1036 specifications.
12. The developed OCM solution yields output errors not compliant with IEC1036 specifications. However, the validity of these results is questionable considering the inadequacy of the testing equipment.

These and other findings regarding the project are discussed in each relevant section of this report in more detail.

## Section 12: Main Conclusions

Based on research conducted during this investigation and the findings of this report the following main conclusions are made:

1. None of the power measurement techniques investigated are viable for OCM.
2. A unique power measurement technique must be employed in order to comply with the stipulated constraints.
3. Power measurement using RMS values of current and voltage is not viable for OCM.
4. The use of discrete digital samples of voltage and current is the only viable power calculation for OCM.
5. The respective MCU multiplexing A/D converters may be considered to provide concurrent sampling, of at least two input channels, for the stipulated sampling frequency and ADC resolution.
6. Zero-crossing detection, and subsequently, power factor measurement is not reliable using an MCU with the stipulated constraints of sampling frequency and processor overhead.
7. The proposed OCM solution complies with the stipulated cost restraints, and the processor overhead requirements are within viable limits.
8. The overall viability of the proposed OCM solution is at present undetermined due to an inadequacy of the available testing equipment.

These and other conclusions made are discussed in each relevant section of this report in more detail.

## Section 13: Recommendations

Based on the research and consequent conclusions presented in this report, the following recommendations are made:

1. The proposed OCM algorithm must be implemented on the H8/3827 Hitachi MCU, with the relevant modifications made to accommodate its 10-bit ADC.
2. The modified algorithm must then be integrated with the peripheral circuitry presented in Section 8.4.
3. The resulting system must be tested on site at EML in Isando, using the appropriate testing equipment, for IEC1036 compliance.
4. Should the resulting system be found to comply with IEC1036 specifications, it must be integrated with the existing CASHPOWER metering software. This in turn must be independently tested for IEC1036 compliance.
5. Should the system be found not to comply with IEC1036 specifications at any stage, it should either be modified by the relevant persons at EML using the research presented in this report, or the OCM project should be rejected completely. This is at the sole discretion of the relevant persons at EML.

## Section 14: Bibliography

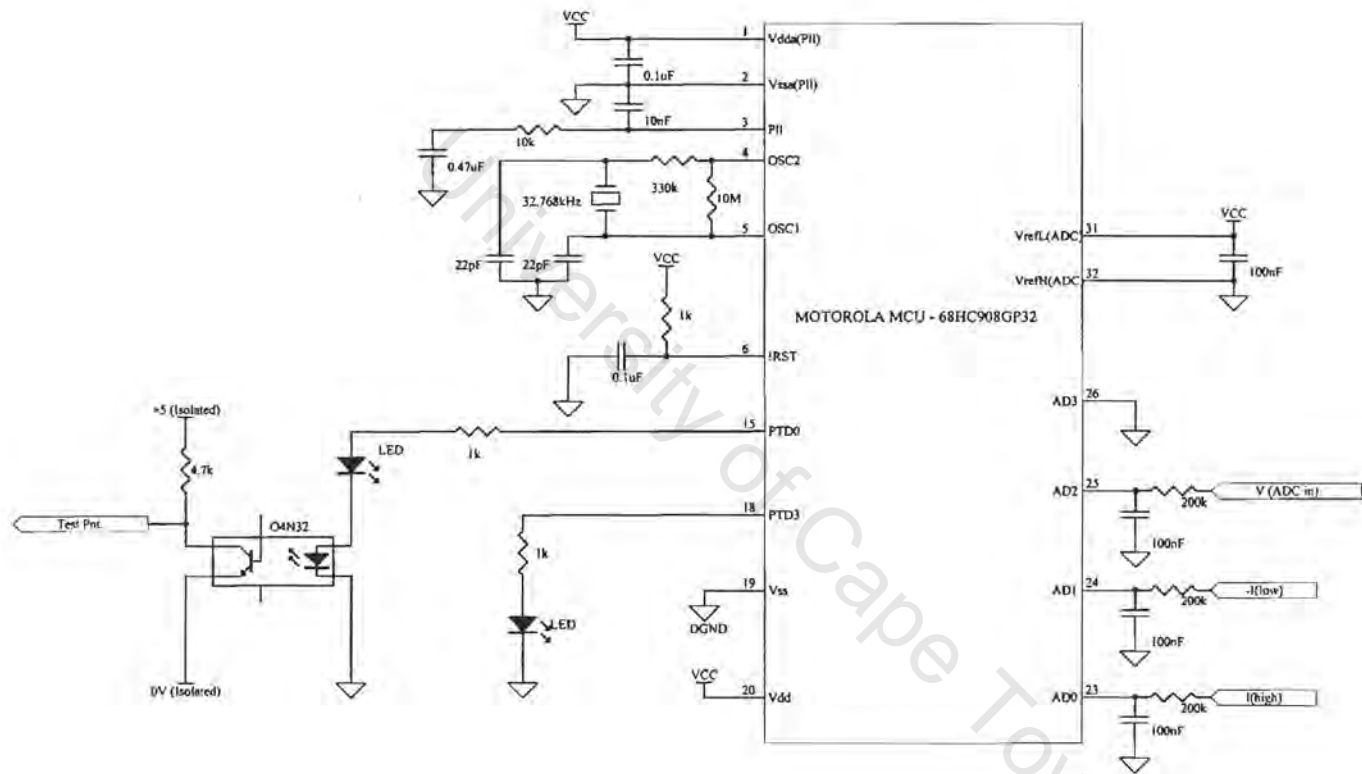
1. Mohan, N. Underland, T & Robbins, W. (1984) "Power Electronics," John Wiley & Sons Inc., Second Edition. Pages 34-36.
2. Hughes-Hallet, D. Gleason, A, et al. (1994) "Calculus," John Wiley & Sons Inc., International Edition. Pages 159, 297-299, 600.
3. Analog Devices, Inc. (1999) "Energy Metering IC with Pulse Output," AD7755\_0.pdf downloaded from <http://www.analog.com>
4. Turgel, R. (1974) "Digital Wattmeter Using a Sampling Method," IEEE Transactions on Instrumentation and Measurement, vol. IM-23, no. 4. Pages 337-341.
5. Davis, P. Rabinowitz, R. (1967) "Numerical Integration," in Turgel, R. (1974) "Digital Wattmeter Using a Sampling Method," IEEE Transactions on Instrumentation and Measurement, vol. IM-23, no.4. Page 337.
6. Corney, A. Pullman, R. (1987) "Digital Sampling Laboratory Wattmeter," IEEE Transactions on Instrumentation and Measurement, vol. IM-36, no. 1. Pages 54-59.
7. Dix, C. (1982) "Calculated Performance of a Digital Sampling Wattmeter Using Systematic Sampling," in Corney, A. Pullman, R. (1987) "Digital Sampling Laboratory Wattmeter," IEEE Transactions on Instrumentation and Measurement, vol. IM-36, no. 1. Pages 56.
8. Stenbakken, G. (1984) "A Wideband Sampling Wattmeter," in Corney, A. Pullman, R. (1987) "Digital Sampling Laboratory Wattmeter," IEEE Transactions on Instrumentation and Measurement, vol. IM-36, no. 1. Pages 56-57.
9. Young, C. Devaney, M. (1998) "Digital Power Metering Manifold," IEEE Transactions on Instrumentation and Measurement, vol. 47, no. 1. Pages 224-228.
10. Hill, J. Alderson, W. (1981) "Design of a Microprocessor-Based Digital Wattmeter," IEEE Transactions on Industrial Electronics and Control Instrumentation, vol IECI-28, no. 3. Pages 180-184.
11. Kung, C. Devaney, M. Huang, C & Kung, C. (1998) "Fuzzy-Based Adaptive Digital Power Metering Using a Genetic Algorithm," IEEE Transactions on Instrumentation and Measurement, vol. 47, no. 1. Pages 183-188.

12. CR Magnetics, Inc. "Precision AC Current to Voltage Transducer," 4310\_a.pdf, ancr4310.pdf downloaded from <http://www.crmagnetics.com>
13. Analog Devices, Inc. "AN-268 Application Note," an268.pdf downloaded from <http://www.analog.com>
14. Microchip Technology, Inc. (1997) "Watt-Hour Meter Reference Design," PICREF\_3.pdf downloaded from <http://microchip.com>
15. Energy Measurement Limited Information Pack. (1996) "Metering Application Report," Texas Instruments.
16. Colins, A. (1999) "AN-559 Preliminary Technical Data Application Note," Analog Devices, Inc. AD7755\_T.pdf downloaded from <http://www.analog.com>
17. International Electrotechnical Commission 1036 Specification (1996) "Alternating Current Static Watt-Hour Meters for Active Energy (Classes 1 and 2)," The South African Bureau of Standards, Second Edition.
18. Bartholomew, D. (1963) "Electrical Measurements and Instrumentation," Allyn & Bacon Inc., Pages 112, 116, 119-120, 230.
19. CR Magnetics, Inc. "Commercial & Metering Class Current Transformers," ancrct\_2.pdf downloaded from <http://www.crmagnetics.com>
20. Shrire, S. (1997) "EEE359W - Electronic Components, Circuits and Modules," Course Notes, University of Cape Town. Pages 1-20, 55, 128-139, 225-228, 230-238.

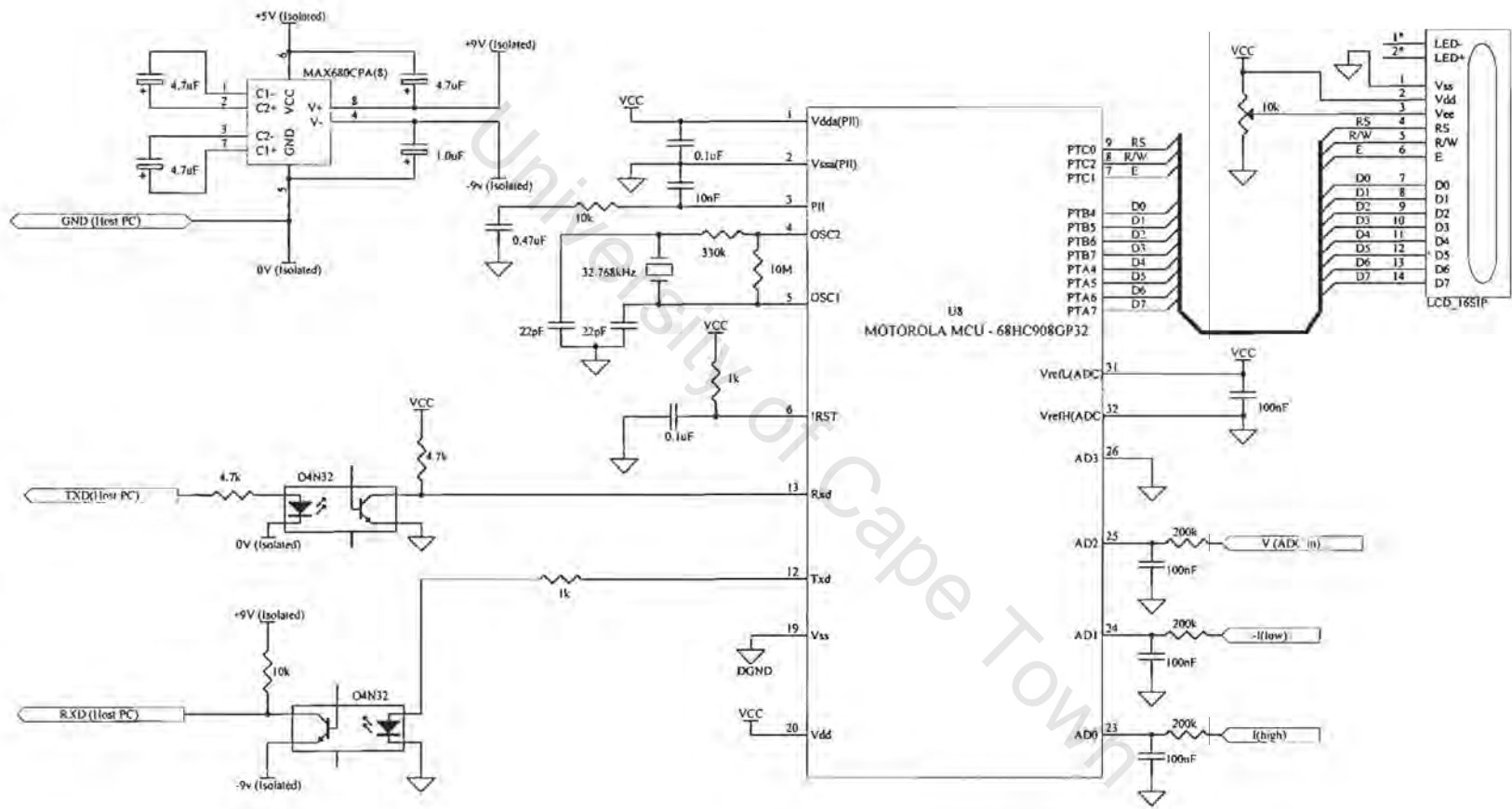
## Annexure A: Hardware Schematics

All the circuitry schematics for the developed hardware are presented in this annexure.

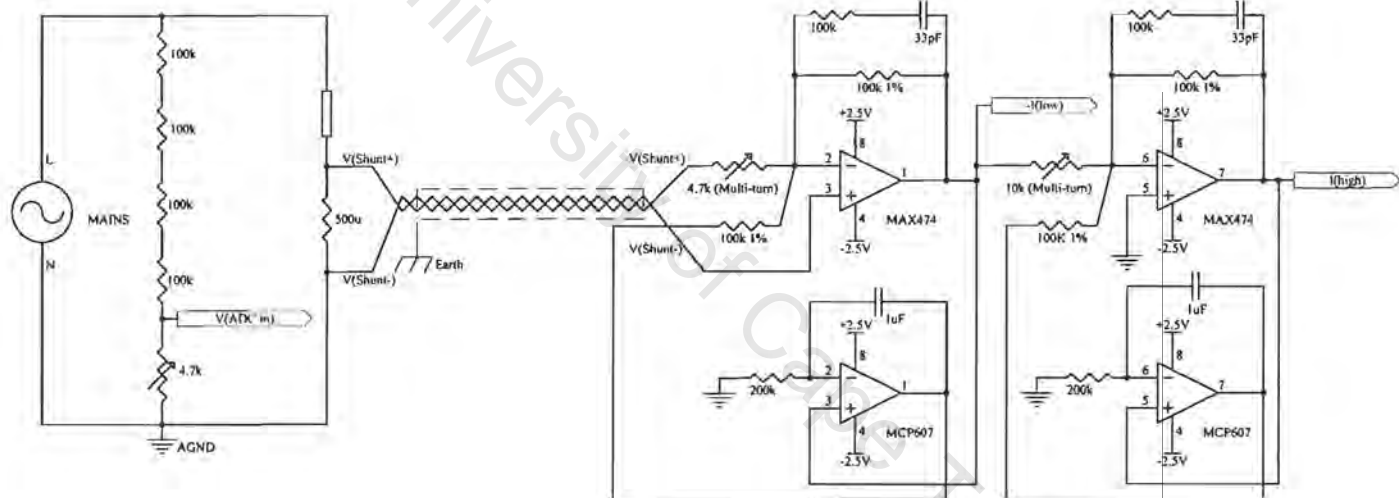
University of Cape Town



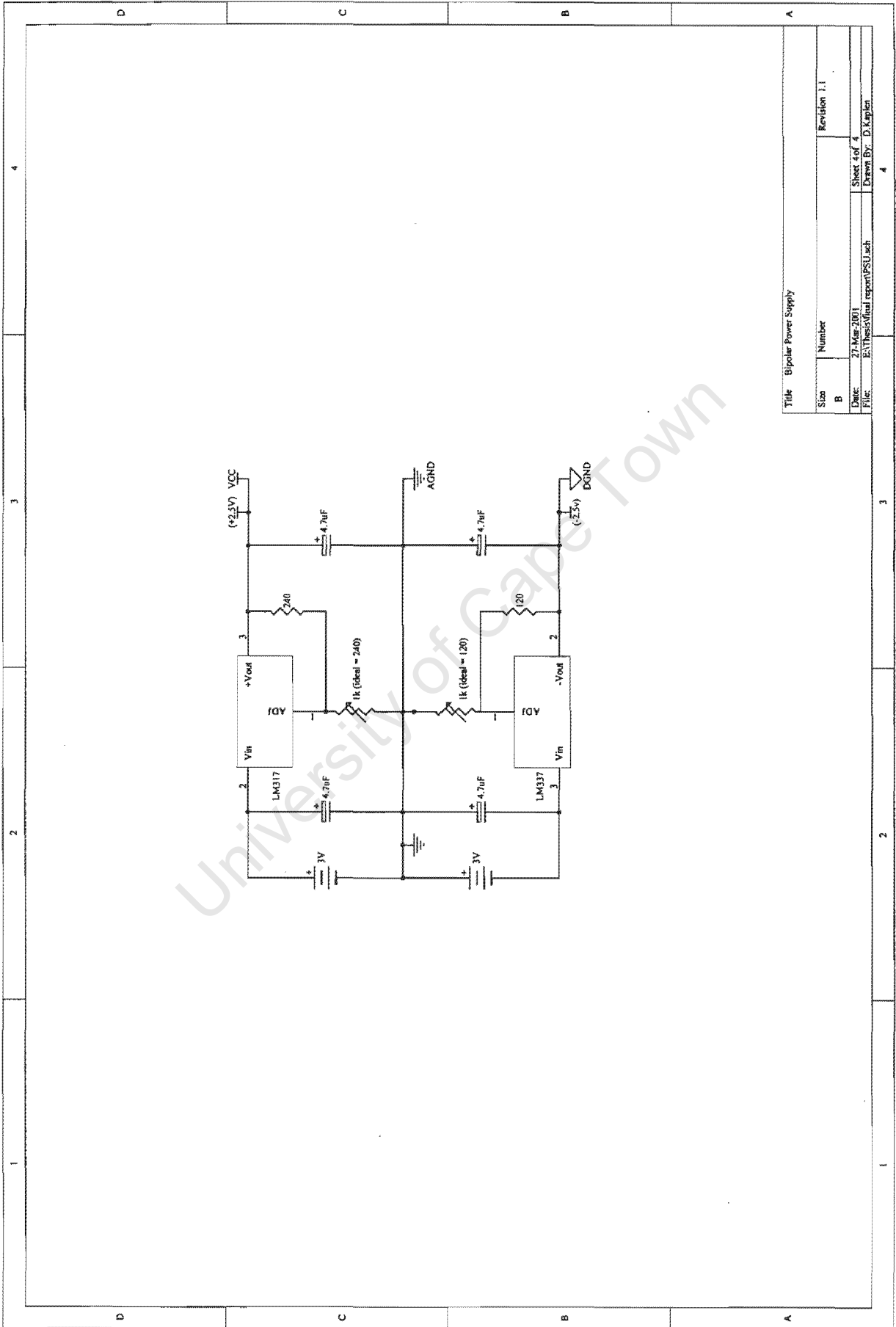
Title: Final OCM Version MCU		
Size: B	Number:	Revision: 1.3
Date: 27-Mar-2001	Sheet 1 of 4	
File: E:\Thesis\...Final Version MCU.sch	Drawn By: D.Kaplan	



Title Test OCM Version MCU		
Size B	Number	Revision 1.3
Date:	27-Mar-2001	Sheet 2 of 4
File:	E:\Thesis\1Test Version MCU.sch	Drawn By: D.Kaplan



Title OCM Peripheral Signal Conditioning		
Size B	Number	Revision 3.2
Date: 27-Mar-2001	Sheet 3 of 4	
File: E:\Thesis\Signal Conditioning Sch	Drawn By: D.Kaplan	



Title: Bipolar Power Supply	
Size: B	Number: Revision 1.1
Date: 27-Mar-2001	Sheet 4 of 4
File: E:\Thesis\final report\PSU.sch	Drawn By: D.K.Kaplan

## Annexure B: Firmware/Software Listing

The MCU firmware for both the final OCM solution and the test version are presented. Also included is the data logger Borland C software coding.

### **Final OCM Solution Firmware**

;FINAL SOLUTION - NO TESTING PROCEDURES

;Firmware written and compiled by D.Kaplen

;Filename: OCM\_Final.asm

;Include Files: gp20regs.inc - general register declarations

; peripherals\_fin.inc - init routines for peripherals

;MCU: Motorola 68HC908GPZ

;NOTE:

;Variables stored as absolute value with sign flag

;Includes output pulse generation

;No LCD used

;No serial comms used

;Nulling offset included

----- MEMORY ASSIGNMENTS -----

RamStart EQU \$0040 ;from 0040 to B000 gives approx 45kBytes ram

RomStart EQU \$B000 ;from B000 to FFFE gives approx 20kBytes rom

ResetVector EQU \$FFDC

\$Include 'gp20regs.inc' ;general peripheral register declarations

----- NUMERICAL CONSTANTS -----

ADC\_GND\_CHANNEL EQU \$03 ;define ADC channels

ADC\_V\_CHANNEL EQU \$02

ADC\_I\_HIGH\_CHANNEL EQU \$00

ADC\_I\_LOW\_CHANNEL EQU \$01

CROSSOVER\_POINT EQU \$78 ;equal to 120 decimal

SECONDARY\_GAIN EQU !16 ;decimal 16

TOT\_NUM\_SAMPLES EQU !41 ;total number of samples per channel at sampling freq. of 2048 Hz

SUM\_THRESH\_MS EQU \$00 ;summed threshold constant (32 bits unsigned)

SUM\_THRESH\_NSH EQU \$57 ;these are all used for pulse generation

SUM\_THRESH\_NSL EQU \$45

SUM\_THRESH\_LS EQU \$D1

MIN\_THRESH\_NSH EQU \$E0 ;the third byte of 32 bit constant - other bytes not  
;required. Used to detect negative power beyond threshold  
;and stop output

org RamStart

----- DECLARE VARIABLES -----

ADC\_Offset ds 1

```
ADC_Voltage      ds 1
ADC_Current_MS   ds 1
```

```
Inst_Temp        ds 1
Inst_Pwr_MS      ds 1
Inst_Pwr_NS      ds 1
Inst_Pwr_LS      ds 1
```

```
Voltage_Sign_Flag ds 1
Current_Sign_Flag ds 1
```

```
Summed_Pwr_LS    ds 1
Summed_Pwr_NSL   ds 1
Summed_Pwr_NSH   ds 1
Summed_Pwr_MS    ds 1
Sample_Num       ds 1
```

```
Temp_Pwr_MS      ds 1
Temp_Pwr_NSH     ds 1
Temp_Pwr_NSL     ds 1
Temp_Pwr_LS      ds 1
```

```
Temp_Vol_Variable ds 1
Temp_Cur_Variable ds 1
```

```
TB_ISR_flag      ds 1          ;to indicate a timebase interrupt
                                   ;bit 0 is set on each interrupt
```

```
org RomStart
```

```
;------ INITIALISE PERIPHERALS -----;
```

```
main_init:
  mov  #$FF,DDRA          ;set port A as output
  mov  #$F0,DDRB          ;set port B high nibble as output, low as input
  bset 0,DDRD             ;set port C pin 4 as output for output frequency
  bset 3,DDRD             ;set port C pin 3 as output for negative power indicator
  mov  #1,CONFIG1         ;disable COP reset
  jsr  init_clk           ;initialise clock rate
  jsr  init_TB            ;initialise Timebase Module
  mov  #$20,ADCLK         ;Bus Clock/4 for ADC clock register (=approx 1 MegHz)
```

```
;------ INITIALISE VARIABLES -----;
```

```
  clra                   ;clear accumulators
  clr  Voltage_Sign_Flag ;clear sign flags
  clr  Current_Sign_Flag
  clr  Summed_Pwr_LS     ;clear running summation
  clr  Summed_Pwr_NSL
  clr  Summed_Pwr_NSH
  clr  Summed_Pwr_MS
  clr  TB_ISR_flag
  cli                               ;enable global interrupts
```

```
***** MAIN ROUTINE *****
```

```
main:
  bset 1,TBCR             ;start timebase (interrupt to provide pulses for sampling)(4)
  mov  #TOT_NUM_SAMPLES,Sample_Num ;total number of samples taken per sampling loop(4)
```

```

Offset_test:                ;determine offset factor - sample gnd channel

    mov  #ADC_GND_CHANNEL,ADSCR    ;disable interrupt, wait for set flag, single conversion(4)
                                           ;sample from GND Channel

ADC_Offset_Wait:
    brclr 7,ADSCR,ADC_Offset_Wait    ;wait for conversion flag bit to be set(max 16us+4)
    mov  ADR,ADC_Offset            ;conversion complete, store value in variable(4)

***** SAMPLING LOOP *****
Sampling_Loop:

TB_wait:                    ;(interrupt service routine uses 4+4+3)
    brclr 0,TB_ISR_flag,TB_wait    ;wait here for interrupt flag to be set(4)
                                           ;DO NOT conduct procedure within interrupt service routine

    bclr 0,TB_ISR_flag            ;to get here, a TB interrupt has occurred(4)

    mov  #ADC_I_HIGH_CHANNEL,ADSCR    ;disable ADC interrupt, wait for set flag,(4)
                                           ;single conversion - sample from High Gain Input

ADC_I_High_Wait:
    brclr 7,ADSCR,ADC_I_High_Wait    ;wait for conversion flag bit to be set(max=16uS)
    lda  ADR                    ;conversion complete, store value in accumulator(3)

    mov  #ADC_V_CHANNEL,ADSCR        ;disable interrupt, wait for set flag,(4)
                                           ;START a single conversion - sample from Voltage ADC input

    sub  ADC_Offset                ;subtract offset while ADC is converging(3)
                                           ;result now in accumulator a
    tax                                     ;copy high-gain current value to x accumulator(1)
    bcc  Clear_I_Sign_Flag          ;branch if current positive or zero - set flag to zero(3)
    bset 0,Current_Sign_Flag        ;else current is negative, set flag(4)
    negx                             ;and obtain absolute value (x accumulator holds current)(1)
    bra  ADC_V_Wait                ;branch(3)

Clear_I_Sign_Flag:
    clr  Current_Sign_Flag          ;if get here then current is positive(3)

ADC_V_Wait:                  ;wait for conversion compete from above
    brclr 7,ADSCR,ADC_V_Wait        ;(max < 16us + 4)
    lda  ADR                    ;grab sample(3)

    mov  #ADC_I_LOW_CHANNEL,ADSCR    ;sample low-gain-input in case it is needed(4)

    sub  ADC_Offset                ;subtract offset from Voltage channel while ADC converging(3)
    bcc  Clear_V_Sign_Flag          ;branch if voltage positive (carry bit clear)(3)
    bset 0,Voltage_Sign_Flag        ;else it was negative(4)
    nega                             ;obtain voltage absolute value(1)
    sta  ADC_Voltage                ;store absolute value in variable(3)
    bra  Channel_Selection          ;(3)

Clear_V_Sign_Flag:
    clr  Voltage_Sign_Flag          ;voltage was positive(3)
    sta  ADC_Voltage                ;copy absolute result to variable(3)

Channel_Selection:
    txa                             ;get (absolute)current value back(1)
    sub  #CROSSOVER_POINT          ;check High-gain-input still in accumulator(3)

```

```

bcc   ADC_I_Low_Wait           ;branch if carry bit clear i.e. if >= than pos crossover point,
                                   ;then use low-gain input(3)
                                   ;else use high-gain - no branching required

```

\*\*\*\*\*High-gain Inst Power Calculation\*\*\*\*\*

```

High_Gain_Inst_Pwr_Cal:         ;else use the high-gain input
clr   Inst_Pwr_MS               ;only required 16-bits(3)
lda   ADC_Voltage               ;recall this is in absolute value format(3)
mul                                       ;obtain instantaneous power (absolute value)(5)
sta   Inst_Pwr_LS               ;(3)
stx   Inst_Pwr_NS               ;these are the absolute bytes of result(3)
lda   Voltage_Sign_Flag         ;(3)
cbeq  Current_Sign_Flag,Pos_Inst_Pwr ;if signs are equal then output will be positive(5)
bra   Neg_Inst_Pwr             ;else they are negative(3)

```

\*\*\*\*\* Low Gain Current Channel Sampling \*\*\*\*\*

```

ADC_I_Low_Wait:                 ;wait for ADC conversion complete (low-gain-input)
brclr 7,ADSCR,ADC_I_Low_Wait    ;(max < 16us+4)
lda   ADR                       ;(3)

sub   ADC_Offset                ;subtract offset from low-gain-input -> result in accumulator(3)

brset 0,Current_Sign_Flag,Current_Flag_Set ;branch if high-gain input was negative(5)
nega                                       ;else it was positive(3) - negate this sample
bra   Low_Gain_Inst_Pwr_Cal ;(3)

```

```

Current_Flag_Set:               ;if get here then current is negative
                                   ;but do NOT negate - not necessary for low gain
                                   ;no branching required

```

\*\*\*\*\* Low-Gain Inst Power Calculation \*\*\*\*\*

```

Low_Gain_Inst_Pwr_Cal:         ;recall that low-gain current value is in acc
ldx   #SECONDARY_GAIN          ;load gain(2)
mul                                       ;multiply gain with (absolute) current value(5)
                                   ;the sign of this operation is in Current_Sign_Flag...
                                   ;...since gain is always positive
                                   ;absolute value of current now in X:A

stx   ADC_Current_MS           ;copy MS byte of current to variable(3)
                                   ;a holds LS byte of current

ldx   ADC_Voltage              ;load signed voltage(3)
mul                                       ;multiply ADC_Current_LS with (absolute)voltage(5)
sta   Inst_Pwr_LS              ;this is LS of Inst Pwr(3)
stx   Inst_Temp                ;make a copy of MSByte of partial result(3)

ldx   ADC_Voltage              ;load signed voltage(3)
lda   ADC_Current_MS           ;load ADC_Current_MS byte into a accumulator(3)

mul                                       ;multiply ADC_Current_MS with (unsigned)voltage(5)
                                   ;this value should be offset by 256 (1 byte)
                                   ;now need to add the two partial multiplication results
                                   ;therefore a must be added with Inst_Temp(3)
add   Inst_Temp                ;this is NS of Inst Pwr(3)
sta   Inst_Pwr_NS              ;this is NS of Inst Pwr(3)
txa                                       ;transfer x to a(1)
adc   #0                        ;add the carry from the previous result(2)
                                   ;Inst_Pwr_MS is now in a accumulator

```

```

    sta  Inst_Pwr_MS          ;(3) no branching required...
Pwr_polarity_test:
    lda  Voltage_Sign_Flag    ;(3)
    cbeq Current_Sign_Flag,Pos_Inst_Pwr ;if signs are equal then output will be positive(5)
    bra  Neg_Inst_Pwr        ;else they are negative(3)

```

\*\*\*\*\* Summation updating \*\*\*\*\*

```

Pos_Inst_Pwr:
    clc                      ;clear carry bit(1)
    lda  Summed_Pwr_LS       ;load LS byte to be added to(3)
    add  Inst_Pwr_LS         ;add to LS byte of inst pwr(3)
    sta  Summed_Pwr_LS       ;update byte(3)

    lda  Summed_Pwr_NSL      ;load NS byte to be added to(3)
    adc  Inst_Pwr_NS         ;add with carry from partial result(3)
    sta  Summed_Pwr_NSL      ;update byte(3)

    lda  Summed_Pwr_NSH      ;(3)
    adc  Inst_Pwr_MS         ;add with carry from partial result(3)
    sta  Summed_Pwr_NSH      ;(3)

    lda  Summed_Pwr_MS       ;(3)
    adc  #0                  ;add carry from partial result(2)
    sta  Summed_Pwr_MS       ;(3)

    bra  Summation_updated   ;(3)

```

```

Neg_Inst_Pwr:
    clc                      ;clear carry bit(1)
    lda  Summed_Pwr_LS       ;load LS byte to be subtracted from(3)
    sub  Inst_Pwr_LS         ;subtract LS byte of inst pwr(3)
    sta  Summed_Pwr_LS       ;update byte(3)

    lda  Summed_Pwr_NSL      ;load NS byte to be subtracted from(3)
    sbc  Inst_Pwr_NS         ;subtract with carry from partial result(3)
    sta  Summed_Pwr_NSL      ;update byte(3)

    lda  Summed_Pwr_NSH      ;(3)
    sbc  Inst_Pwr_MS         ;subtract with carry from partial result(3)
    sta  Summed_Pwr_NSH      ;(3)

    lda  Summed_Pwr_MS       ;(3)
    sbc  #0                  ;subtract carry from partial result(2)
    sta  Summed_Pwr_MS       ;(3)

```

```

Summation_Updated:
    dbnz Sample_Num,Not_Finished ;branch if not finished sampling(4)
    ;else check Summation for output pulse
    bra  Finished_Sampling

```

```

Not_Finished:
    jmp  Sampling_Loop       ;need this since a branch instruction is out of range(4)

```

\*\*\*\*\* END OF SAMPLING LOOP \*\*\*\*\*

\*\*\*\*\* PULSE GENERATION \*\*\*\*\*

```

Finished_Sampling:
    ;first check if summed inst_PWR is negative

```

```

brset 7,Summed_Pwr_MS,Ave_Pwr_Negative ;branch if summed inst_power is negative(5)
;else it was positive - no branching required

Ave_Pwr_Positive: ;test if beyond summed threshold value

clc ;clear carry bit(1)

lda Summed_Pwr_LS ;load LS byte of running sum(3)
sbc #SUM_THRESH_LS ;subtract threshold LS byte with carry(3)
sta Temp_Pwr_LS ;store result as temp for later update if required(3)

lda Summed_Pwr_NSL ;load NSL byte of running sum(3)
sbc #SUM_THRESH_NSL ;subtract threshold NSL byte with carry(3)
sta Temp_Pwr_NSL ;store result as temp for later update if required(3)

lda Summed_Pwr_NSH ;load NSH byte of running sum(3)
sbc #SUM_THRESH_NSH ;subtract threshold NSH byte with carry(3)
sta Temp_Pwr_NSH ;store result as temp for later update if required(3)

lda Summed_Pwr_MS ;load MS byte of running sum(3)
sbc #SUM_THRESH_MS ;subtract threshold MS byte with carry(3)
sta Temp_Pwr_MS ;store result as temp for later update if required(3)

bcc Toggle_output ;if carry clear, then running sum was(4)
;larger than threshold - toggle the output
;else continue
jmp main ;start again(4)

Toggle_output:
mov Temp_Pwr_LS,Summed_Pwr_LS ;new values for running sum(4)
mov Temp_Pwr_NSL,Summed_Pwr_NSL ;(4)
mov Temp_Pwr_NSH,Summed_Pwr_NSH ;(4)
mov Temp_Pwr_MS,Summed_Pwr_MS ;(4)

bclr 3,PTD ;clear negative power indicator
brset 0,PTD,LEDwasset ;branch if LED was ON(5)
bset 0,PTD ;else LED was OFF, set it now(4)
jmp main ;LED toggled, continue(4)

LEDwasset:
bclr 0,PTD ;LED was ON, clr it now(4)
jmp main ;LED toggled, continue(4)

Ave_Pwr_Negative: ;test if beyond the minimum threshold value to stop timer
;note - since running sum is negative, MS bit of MS byte
;will be SET and need to subtract from threshold value
;therefore MS bit of MS byte of threshold must be SET
;with 1 byte of 24 bits used, the minimum threshold is
;set to FF 00 00 / 2 = -8 355 840

lda #MIN_THRESH_NSH ;third byte of minimum threshold(2)
sbc Summed_Pwr_NSH ;unsigned subtract NSH byte of running sum(3)
;running sum IS signed -> negative
bcc Beyond_Minimum ;if carry clear, running sum was less than threshold(4)
;else it was not - continue without resetting running sum
;(4)
jmp main ;(4)

Beyond_Minimum:
clr Summed_Pwr_LS ;reset these - begin sampling from scratch again(4)

```

```

clr Summed_Pwr_NSL ;DO NOT toggle output(4)
clr Summed_Pwr_NSH ;(4)
clr Summed_Pwr_MS ;(4)
bset 3,PTC ;set negative power indicator(4)
jmp main ;start again ;(4)

```

```

;-----INCLUDE FILES-----

```

```

$Include 'peripherals_fin.inc'

```

```

***** END OF MAIN *****

```

```

*****
* DUMMY_ISR - Dummy Interrupt Service Routine. *
* Returns nothing from interrupt. *
*****

```

```

dummy_isr:

```

```

    rti ; return
;System reset vector.
    org ResetVector

```

```

    dw TB_ISR ; Time Base Vector - the only ISR used for Final Version
    dw dummy_isr ; ADC Conversion Complete
    dw dummy_isr ; Keyboard Vector
    dw dummy_isr ; SCI Transmit Vector
    dw dummy_isr ; SCI Receive Vector
    dw dummy_isr ; SCI Error Vector
    dw dummy_isr ; SPI Transmit Vector
    dw dummy_isr ; SPI Receive Vector
    dw dummy_isr ; TIM2 Overflow Vector
    dw dummy_isr ; TIM2 Channel 1 Vector
    dw dummy_isr ; TIM2 Channel 0 Vector
    dw dummy_isr ; TIM1 Overflow Vector
    dw dummy_isr ; TIM1 Channel 1 Vector
    dw dummy_isr ; TIM1 Channel 0 Vector
    dw dummy_isr ; PLL Vector
    dw dummy_isr ; IRQ1 Vector
    dw dummy_isr ; SWI Vector
    dw main_init ; Reset Vector

```

```

***** END OF OCM_FINAL.ASM *****

```

### **Final Solution Peripherals**

```

*****PERIPHERAL MODULES*****

```

```

;Filename: peripheral_fin.inc
;Written by D.Kaplan
; To be included under ROM
; Includes:
; (1) BUS clock via PLL settings
; (2) Timebase interrupt setup and service routine

```

```

*****INTERNAL PLL - BUS CLOCK GENERATOR*****

```

```

init_clk:
; generates a 4.9152MHz bus clock from external 32768Hz crystal
; for 4.9152MHz external - do not call init_clk

```

```

;NOTE:

```

```

;In order to change the clock generator PLL settings, ensure that the PLL first off -
;(PLLON=0). Once PLL off, ensure that it is not driving the CPU bus -(BCS=0)
;Also set the prescaler and VCO power of two bits, all done by:

```

```

    mov #%00000010,PCTL

```

;Now we must. These registers tell the PLL what factor  
 ;the clock must be multiplied by. This factor is determined using a look up chart in the  
 ;documentation.

```

    mov #$02,PMSH          ;set up the PLL multiplier registers - for 4.1943MHz
    mov #$58,PMSL
;   mov  #$03,PMSH          ;for 8.388MHz - not used
;   mov  #$D1,PMSL

    mov #$80,PMRS         ;set VCO of PLL for 4.1943MHz operation
;   mov  #$D0,PMRS         ;for 8.388MHz - not used

    bset 5,PCTL           ;PLL setup complete - turn it on

    nop                   ;allow PLL to stabilise

    bset 4,PCTL           ;select PLL as Bus Clock source

    rts                   ;return
  
```

\*\*\*\*\*TIME BASE VECTOR\*\*\*\*\*

```

Init_TB:                   ;initialise Timebase Module
    bset 6,TBCR           ;set bits : TBR2,TBR1 for freq of 2048 HZ
    bset 5,TBCR
    bclr 4,TBCR          ;clr bits : TBR0 for freq of 2048
    bset 3,TBCR          ;clear timebase interrupt flag
    bset 2,TBCR          ;set TBIE i.e. enable timebase interrupt
    bclr 1,TBCR          ;to start counting, need to set TBON (bit 1,TBCR)
    rts                   ;return
;-----
TB_ISR:                    ;time base interrupt service routine
    bset 3,TBCR          ;clear timebase interrupt flag by setting bit 3 of TB control register
    bset 0,TB_ISR_flag   ;use this as an interrupt flag
    rti                  ;do not perform the interrupt routine here, but in main loop
  
```

\*\*\*\*\*END OF PERIPHERALS\_FIN.INC\*\*\*\*\*

### **OCM Test Version Firmware**

The test version contains many more modules than the final version, none of which effect the power calculation. This code is *not* optimised.

;OCM TEST SOLUTION ASSEMBLER CODE

```

; Filename:      OCM_test.asm
; Include Files: gp20regs.inc - register definitions for MCU
;               LCD_regs.inc - register definitions for intelligent LCD
;               peripherals.inc - peripherals' subroutines (serial comms, timebase, etc)
;               LCD_Subroutines.inc - LCD subroutines for various output displays
  
```

;NOTE: All test routines included in this module including:

```

;   LCD intelligent display
;   serial communications
;   single cycle testing
;   multiple cycle testing
; all of which are included exclusively as testing routines and must be removed
  
```

; for the final solution. Furthermore, the output frequency generation HAS NOT been  
 included in this test version. All outputs to be verified using the developed BC program.

;Variables stored as absolute value with sign flag

```
RamStart EQU $0040 ;from 0040 to B000 gives approx 45kBytes ram
RomStart EQU $B000 ;from B000 to FFFE gives approx 20kBytes rom
ResetVector EQU $FFDC
```

```
$Include 'gp20regs.inc'
$Include 'LCD_regs.inc'
```

```
-----
MSG1 EQU $E000
MSG2 EQU $E0A0
MSG3 EQU $E0C0
MSG4 EQU $E0D0
SAMPLING_DONE EQU $E0F0

ADC_GND_CHANNEL EQU $03 ;define adc channels
ADC_V_CHANNEL EQU $02
ADC_I_HIGH_CHANNEL EQU $00
ADC_I_LOW_CHANNEL EQU $01

CROSSOVER_POINT EQU $78 ;!120
SECONDARY_GAIN EQU !16
TOT_NUM_SAMPLES EQU !41 ;total number of sample pairs at sampling freq of 2048 Hz

CONTINUE_KEY EQU $63 ;corresponds to ascii 'c' for continue from host PC

    org RamStart

LCD_HIGH EQU PTA ;Port A reserved for High nibble of 8 bit LCD data
LCD_LOW EQU PTB ;Port B reserved for Low nibble of 8 bit LCD data

ADC_Offset ds 1
ADC_Voltage ds 1
ADC_Current_MS ds 1
Inst_Temp ds 1
Inst_Pwr_MS ds 1
Inst_Pwr_NS ds 1
Inst_Pwr_LS ds 1
Voltage_Sign_Flag ds 1
Current_Sign_Flag ds 1
Summed_Pwr_LS ds 1
Summed_Pwr_NSL ds 1
Summed_Pwr_NSM ds 1
Summed_Pwr_MS ds 1
Sample_Num ds 1

Temp_Vol_Variable ds 1
Temp_Cur_Variable ds 1

Sample_Pointer_MS ds 1
Sample_pointer_LS ds 1
Zero ds 1
decode_var ds 5 ;for units, tens, hundreds, thousands and ten thousands
TB_ISR_flag ds 1 ;to indicate a timebase interrupt - bit 0 set on each interrupt
RXD_byte ds 1
```

```

reg0    ds 1      ;use a byte as a type of register for LCD display
var1    ds 1      ;variable for multiple use
XH      ds 1      ;xhigh byte variable
XL      ds 1      ;xlow byte variable
YH      ds 1      ;yhigh byte variable
YL      ds 1      ;ylo w byte variable
Z1      ds 1      ;zhigh byte for 16 bit arithmetic answer
Z0      ds 1      ;zlow byte for 16 bit arithmetic answer

Sample_Array ds 328 ;maximum RAM allowable for testing samples
                ;use 16-bit words for each sample - thus 40 samples per channel
                ;160 bytes array. e.g. to address the third element (byte)
                ; lda samplearray+3 ; loads into accumulator.

```

```
org RomStart
```

---

```

;
main_init:
mov  #$FF,DDRA      ;set port A as output
mov  #$FF,DDRB      ;set port B as output
mov  #$00,DDRC      ;set port C as input
bset 0,DDR D        ;set port C pin 4 as output
mov  #1,CONFIG1     ;disable COP reset
jsr  init_clk       ;initialise clock rate
jsr  init_sci       ;initialise serial port
jsr  init_TB        ;initialise Timebase Module
mov  #$20,ADCLK     ;Bus Clock/4 for ADC clock register (=approx 2 MegHz)

main:
bclr 5,SCC2         ;mask RXD interrupts for now
clra
clrx
clr  Voltage_Sign_Flag
clr  Current_Sign_Flag
clr  Summed_Pwr_LS
clr  Summed_Pwr_NSL
clr  Summed_Pwr_NSM
clr  Summed_Pwr_MS
clr  Zero

clr  Sample_Pointer_MS
clr  Sample_Pointer_LS

mov  #Sample_Array,Sample_Pointer_LS ; point to an element in this array
mov  #TOT_NUM_SAMPLES,Sample_Num
cli  ;enable global interrupts

jsr  shortpulse     ;allow LCD module to settle
jsr  resetLCD       ;initialise LCD module
jsr  longpulse
jsr  initLCD
jsr  longpulse
ldhx #MSG4          ;display init message on LCD
jsr  wrLCDstr

jsr  RXD_Wait       ;wait for serial input before proceeding

mov  #clrDsp,reg0   ;Clear LCD before continuing
jsr  wrLCDcom
mov  #homeCur,reg0

```

```

jsr  wrLCDcom

*****Main Sampling Loop*****

Offset_test:                ;determine offset factor - sample gnd channel

    mov  #ADC_GND_CHANNEL,ADSCR ;disable interrupt, wait for set flag, single conversion
                                ;sample from GND Channel

ADC_Offset_Wait:
    brclr 7,ADSCR,ADC_Offset_Wait ;wait for conversion flag bit to be set
    mov  ADR,ADC_Offset          ;conversion complete, store value in variable

Sampling_Loop:              ;- main loop for sampling channels and calculating power*****
    bset 1,TBCR                ;start timebase (interrupt to provide pulses for sampling)

TB_wait:
    brclr 0,TB_ISR_flag,TB_wait ;wait here for interrupt flag to be set, MAKE A NOTE OF WHY
                                ;HERE AND NOT IN THE INTERRUPT SERVICE ROUTINE NBNBNB

    bclr 0,TB_ISR_flag         ;to get here, a TB interrupt has occurred

    mov  #ADC_I_HIGH_CHANNEL,ADSCR ;disable interrupt, wait for set flag,(4)
                                ;single conversion - sample from ADC(2) - High Gain Input

ADC_I_High_Wait:
    brclr 7,ADSCR,ADC_I_High_Wait ;wait for conversion flag bit to be set(max=16uS)
    lda  ADR                    ;conversion complete, store value in accumulator(3)

    mov  #ADC_V_CHANNEL,ADSCR    ;disable interrupt, wait for set flag,(4)
                                ;single conversion - sample from ADC(1) - Voltage
    sub  ADC_Offset              ;subtract offset while ADC is converging(3)
                                ;result now in a accumulator
    tax                                ;copy current value to x accumulator(1)
    bcc  Clear_I_Sign_Flag        ;branch if current positive or zero (carry buit clear)- set flag to zero(3)
    bset 0,Current_Sign_Flag      ;else current is negative, set flag(4)
    negx                           ;and obtain absolute value (x accumulator hold current)(1)
    bra  ADC_V_Wait              ;branch(3)

Clear_I_Sign_Flag:
    clr  Current_Sign_Flag        ;if get here then current is positive(3)

ADC_V_Wait:                  ;wait for conversion compete from above
    brclr 7,ADSCR,ADC_V_Wait      ;(max 16us)
    lda  ADR                      ;(3)

    mov  #ADC_I_LOW_CHANNEL,ADSCR ;sample low-gain-input in case it is needed(4)

    sub  ADC_Offset              ;subtract offset from Voltage channel while ADC converging(3)
    bcc  Clear_V_Sign_Flag        ;branch if voltage positive (carry bit clear)(3)
    bset 0,Voltage_Sign_Flag      ;else it was negative(4)
    nega                           ;obtain voltage absolute value(1)
    sta  ADC_Voltage              ;store absolute value in variable(3)
    bra  Channel_Selection        ;(3)

Clear_V_Sign_Flag:
    clr  Voltage_Sign_Flag        ;voltage was positive(3)
    sta  ADC_Voltage              ;copy absolute result to variable(3)

Channel_Selection:
    txa                            ;get (absolute)current value back(1)
    sub  #CROSSOVER_POINT         ;check High-gain-input still in accumulator(3)

```

```

    bcc  ADC_I_Low_Wait      ;branch if carry bit clear i.e.if >= than pos crossover point, use low-gain
input(3)
                                ;else use high-gain - no branching required

***** High-gain Inst Power Calculation *****

High_Gain_Inst_Pwr_Cal:      ;else use the high-gain input
    clr  Inst_Pwr_MS        ;only required 16-bits(3)

;this section of code is for testing purposes only - remove for final solution
;NOTE: due to sign bit & absolute value format, only absolute values for each sample will be
;transmitted

    stx  Temp_Cur_Variable ;save current value for now

    ldhx Sample_Pointer_MS  ;load h:x pointer to next available space in array
    mov  ADC_Voltage,x+     ;store absolute value of 8-bit Voltage sample next
    mov  Zero,x+           ;store MSbyte of current - since high gain used, only 1 byte
    mov  Temp_Cur_Variable,x+ ;store absolute value of 8-bit current sample next
    clc
    lda  Sample_Pointer_LS
    add  #03
    sta  Sample_Pointer_LS
    lda  Sample_Pointer_MS
    adc  #00
    sta  Sample_Pointer_MS

    idx  Temp_Cur_Variable

After_Serial_Vol:          ;take this out for final solution

    lda  ADC_Voltage        ;recall this is in absolute value format(3)
    mul  ;obtain instantaneous power (absolute value)(5)
    sta  Inst_Pwr_LS        ;(3)
    stx  Inst_Pwr_NS        ;these are the absolute bytes of result(3)
    lda  Voltage_Sign_Flag  ;(3)
    cbeq Current_Sign_Flag,Pos_Inst_Pwr ;if signs are equal then output will be positive(5)
    bra  Neg_Inst_Pwr       ;else they are negative(3)

***** Continue Sampling if required *****

ADC_I_Low_Wait:           ;wait for ADC conversion complete (low-gain-input)
    brclr 7,ADSCR,ADC_I_Low_Wait ;(max 16us)
    lda  ADR                ;(3)
    sub  ADC_Offset         ;subtract offset from low-gain-input -> result in accumulator(3)

    brset 0,Current_Sign_Flag,Current_Flag_Set ;branch if high-gain input was negative(5)
    nega                                     ;else this sample is negative (they are inverted)
    bra  Low_Gain_Inst_Pwr_Cal ;(3)

Current_Flag_Set:        ;if get here then current is negative
    nega                 ;obtain low-gain current absolute value(1)
                                ;no branching required

***** Low-Gain Inst Power Calculation *****

Low_Gain_Inst_Pwr_Cal:   ;recall low-gain current value in a
    idx  #SECONDARY_GAIN ;load gain(2)
    mul  ;multiply gain with (absolute) current value(5)
                                ;the sign of this operation is in Current_Sign_Flag

```

```

                                ;since gain is always positive
                                ;absolute current now in X:A
psha ;push LS byte of current
pshx ;push MS byte of current

; this section is for testing purposes only

stx Temp_Cur_Variable ;save current value for now

ldhx Sample_Pointer_MS ;load h:x pointer to next available space in array
mov ADC_Voltage,x+ ;store absolute value of 8-bit Voltage sample next
mov Temp_Cur_Variable,x+ ;store MSbyte of current - since high gain used, only 1 byte
sta Temp_Cur_Variable
mov Temp_Cur_Variable,x+ ;store LS byte of current
clc
lda Sample_Pointer_LS
add #03
sta Sample_Pointer_LS
lda Sample_Pointer_MS
adc #00
sta Sample_Pointer_MS

;end of testing procedure

pulx ;MS byte of current
pula ;LS byte of current
stx ADC_Current_MS ;copy MS byte of current to variable(3)
;a holds LS byte of current

ldx ADC_Voltage ;load signed voltage(3)
mul ;multiply ADC_Current_LS with (absolute)voltage(5)
sta Inst_Pwr_LS ;this is LS of Inst Pwr(3)
stx Inst_Temp ;make a copy of MSByte of partial result(3)

ldx ADC_Voltage ;load signed voltage(3)
lda ADC_Current_MS ;load ADC_Current_MS byte into a accumulator(3)

mul ;multiply ADC_Current_MS with (unsigned)voltage(5)
;this value should be offset by 256 (1 byte)
;now need to add the two partial multiplication results
add Inst_Temp ;therefore a must be added with Inst_Temp(3)
sta Inst_Pwr_NS ;this is NS of Inst Pwr(3)
txa ;transfer x to a(1)
adc #0 ;add the carry from the previous result(2)
;Inst_Pwr_MS is now in a accumulator
sta Inst_Pwr_MS ;(3)

lda Voltage_Sign_Flag ;(3)
cbeq Current_Sign_Flag,Pos_Inst_Pwr ;if signs are equal then output will be positive(5)
bra Neg_Inst_Pwr ;else they are negative(3)

***** Summation updating *****
Pos_Inst_Pwr:
clc ;clear carry bit(1)
lda Summed_Pwr_LS ;load LS byte to be added to(3)
add Inst_Pwr_LS ;add to LS byte of inst pwr(3)
sta Summed_Pwr_LS ;update byte(3)

lda Summed_Pwr_NSL ;load NS byte to be added to(3)
adc Inst_Pwr_NS ;add with carry from partial result(3)

```

```

sta Summed_Pwr_NSL          ;update byte(3)

lda Summed_Pwr_NSM          ;(3)
adc Inst_Pwr_MS             ;add with carry from partial result(3)
sta Summed_Pwr_NSM          ;(3)

lda Summed_Pwr_MS           ;(3)
adc #0                      ;add carry from partial result(2)
sta Summed_Pwr_MS           ;(3)

jmp Summation_updated       ;(2)

```

## Neg\_Inst\_Pwr:

```

clc                          ;clear carry bit(1)
lda Summed_Pwr_LS           ;load LS byte to be subtracted from(3)
sub Inst_Pwr_LS             ;subtract LS byte of inst pwr(3)
sta Summed_Pwr_LS           ;update byte(3)

lda Summed_Pwr_NSL         ;load NS byte to be subtracted from(3)
sbc Inst_Pwr_NS             ;subtract with carry from partial result(3)
sta Summed_Pwr_NSL         ;update byte(3)

lda Summed_Pwr_NSM         ;(3)
sbc Inst_Pwr_MS             ;subtract with carry from partial result(3)
sta Summed_Pwr_NSM         ;(3)

lda Summed_Pwr_MS          ;(3)
sbc #0                      ;subtract carry from partial result(2)
sta Summed_Pwr_MS          ;(3)

jmp Summation_updated       ;(2)

```

## test2:

```

jmp Sampling_Loop

```

## Summation\_Updated:

```

lda Summed_Pwr_MS          ;for testing
lda Summed_Pwr_NSM
lda Summed_Pwr_NSL
lda Summed_Pwr_LS

dbnz Sample_Num,test2      ;branch if not finished sampling
                           ;else check Summation for output pulse
                           ;no branching required

```

## Finished\_Sampling:

```

bclr 1,TBCR                ;finished sampling - stop the timebase module
                           ;first check if summed inst_PWR is negative

lda Summed_Pwr_MS          ;for testing
lda Summed_Pwr_NSM
lda Summed_Pwr_NSL
lda Summed_Pwr_LS
brset 7,Summed_Pwr_MS,Ave_Pwr_Negative ;branch if summed inst_power is negative
                           ;else it was positive - no branching required

```

## Ave\_Pwr\_Positive:

```

;note - at a later stage need to implement pulse generation here
nop

```

```

;for now - just display samples

Ave_Pwr_Negative:                ;detected negative power - do not update output

    mov  #clrDsp,reg0            ;clear display
    jsr  wrLCDcom
    mov  #homeCur,reg0
    jsr  wrLCDcom

    ldhx #SAMPLING_DONE          ;displays "Sampling done."
    jsr  wrLCDstr

    jsr  RXD_Wait                ;wait for RXD_Wait press

    mov  #clrDsp,reg0            ;clear display
    jsr  wrLCDcom
    mov  #homeCur,reg0
    jsr  wrLCDcom

*****Serial Output Transmission*****
    mov  #!41,YH                ;total number of sample pairs to be transmitted

;here the sample pairs are transmitted via the serial port
;they are first decoded and sent using up to five bytes per sample

    ldhx #Sample_Array

outloop:                          ;decode samples and send out serial port
    mov  #00,Z1                  ;first voltage
    mov  x+,Z0
    jsr  decode
    jsr  sci_senddecoded

    mov  x+,Z1                    ;next current
    mov  x+,Z0
    jsr  decode
    jsr  sci_senddecoded

    dbnz YH,outloop              ;continue

    mov  #$01,reg0
    jsr  send_data ;signal end of samples' transmission

;now need to send summed value - small problem have 32 (signed) bits to send, but only
;16 - bit decode function
;do the math in C and for now only send the ASCII format of the HEX numbers

    lda  Summed_Pwr_MS           ;send first nibble
    and  #$F0                    ;mask lower nibble
    nsa                      ;switch nibbles
    add  #$30 ;add 30hex to the number to convert it to ASCII
    sta  reg0
    jsr  send_data

    lda  Summed_Pwr_MS           ;lowr nibble
    and  #$0F
    add  #$30
    sta  reg0
    jsr  send_data

```

```

lda Summed_Pwr_NSM    ;send first nibble
and #$F0             ;mask lower nibble
nsa                  ;switch nibbles
add #$30             ;add 30hex to the number to convert it to ASCII
sta reg0
jsr send_data

```

```

lda Summed_Pwr_NSM    ;lowr nibble
and #$0F
add #$30
sta reg0
jsr send_data

```

```

lda Summed_Pwr_NSL    ;send first nibble
and #$F0             ;mask lower nibble
nsa                  ;switch nibbles
add #$30             ;add 30hex to the number to convert it to ASCII
sta reg0
jsr send_data

```

```

lda Summed_Pwr_NSL    ;lowr nibble
and #$0F
add #$30
sta reg0
jsr send_data

```

```

lda Summed_Pwr_LS     ;send first nibble
and #$F0             ;mask lower nibble
nsa                  ;switch nibbles
add #$30             ;add 30hex to the number to convert it to ASCII
sta reg0
jsr send_data

```

```

lda Summed_Pwr_LS     ;lowr nibble
and #$0F
add #$30
sta reg0
jsr send_data

```

```

mov #$01,reg0
jsr send_data ;signal end of summed value's transmission

```

```

jmp main ;start again

```

```

;Ave_Pwr_Negative:           ;summation is negative - reverse power present
    bset 0,PTD                ;set Negative-flag output LED

```

```

jmp main ;finished

```

```

shortpulse: ;approx 158us - used for LCD pulse command

```

```

    psha
    lda #!254

```

```

shortloop:
    nop
    dbnza shortloop
    pula
    rts

```

```
longpulse:    ;approx 40ms      - used for LCD init. and reset command
```

```
    psha
    lda    #!50
```

```
longloop:
    jsr   shortpulse
    dbnza longloop
    pula
    rts
```

---

```
button:      ;de-bounce ,etc for button on PTB pin 3
```

```
    brset 3,PTC,button ;pb4
    jsr   longpulse
    brset 3,PTC,button
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
    jsr   longpulse
```

```
    rts
```

---

```
RXD_Wait:    ;wait for RXD interrupt and correct byte received
```

```
    psha
```

```
RXD_Wait_Loop:
```

```
    lda    reg0 ;from RXD_ISR - reg0 loaded
    cbeqa #CONTINUE_KEY,RXD_Continue ;check received byte - if equal to "continue"...
                                           ;...key then continue
```

```
    bra   RXD_Wait_Loop
```

```
RXD_Continue:
```

```
    pula
    rts
```

---

```
$Include 'peripherals.inc'
```

```
$Include 'LCD_subroutines.inc'
```

---

```
org MSG1
db !11
db "Sampling..."
```

```
org MSG2
db !04
db "Done"
```

```
org MSG3
db !10
db "Reading..."
```

```
org MSG4
db !08
db "Ready..."
```

```
org SAMPLING_DONE
```

```
db !13
db "Sampling Done:"
```

```
*****
* DUMMY_ISR - Dummy Interrupt Service Routine.          *
* Just does a return from interrupt.                    *
*****
dummy_isr:
```

```
    rti        ; return
```

```
;System reset vector.
    org ResetVector
```

```
    dw TB_ISR    ; Time Base Vector
    dw dummy_isr ; ADC Conversion Complete
    dw dummy_isr ; Keyboard Vector
    dw dummy_isr ; SCI Transmit Vector
    dw RXD_ISR   ; SCI Receive Vector
    dw dummy_isr ; SCI Error Vector
    dw dummy_isr ; SPI Transmit Vector
    dw dummy_isr ; SPI Receive Vector
    dw dummy_isr ; TIM2 Overflow Vector
    dw dummy_isr ; TIM2 Channel 1 Vector
    dw dummy_isr ; TIM2 Channel 0 Vector
    dw dummy_isr ; TIM1 Overflow Vector
    dw dummy_isr ; TIM1 Channel 1 Vector
    dw dummy_isr ; TIM1 Channel 0 Vector
    dw dummy_isr ; PLL Vector
    dw dummy_isr ; ~IRQ1 Vector
    dw dummy_isr ; SWI Vector
    dw main_init ; Reset Vector
```

```
=====END OF OCM_TEST>ASM=====
```

### Test Version Include Files

```
*****PERIPHERAL MODULES*****
```

```
;Filename:      peripherals.inc
;Written by D.Kaplen for OCM_test.asm
```

```
; To be included under ROM
; Includes:
; (1) BUS clock via PLL settings
; (2) Serial Config & Subroutines - init_sci - initialise SCI
;         - send_data - send a byte
;         - wrSCIstr - send a string
;         - RXD_ISR - RXD interrupt service routine
```

```
*****INTERNAL PLL - BUS CLOCK GENERATOR*****
```

```
init_clk:
    mov #%00000010,PCTL
    mov #$02,PMSH ;for 4 meg
    mov #$58,PMSL
    mov #$80,PMRS ;for 4 meg
    bset 5,PCTL
    nop
    bset 4,PCTL
    rts
```

```
*****TIME BASE VECTOR*****
```

```

Init_TB:
    bset 6,TBCR      ;set bits : TBR2,TBR1 (for freq of 2048 HZ)
    bset 5,TBCR
    bclr 4,TBCR      ;clr bits : TBR0 (for freq of 2048)
    bset 3,TBCR      ;clear timebase interrupt flag
    bset 2,TBCR      ;set TBIE i.e. enable timebase interrupt
    bclr 1,TBCR      ;to start counting, need to set TBON (bit 1,TBCR)
rts
;-----
TB_ISR:      ;time base interrupt service routine
    bset 3,TBCR      ;clear timebase interrupt flag by setting bit 3 of TB control register
    bset 0,TB_ISR_flag ;use this as an interrupt flag
    rti              ;do not perform the interrupt routine here, but in main loop

*****SERIAL COMMS*****
init_sci:
;   baud rate = 9600
;   parity = none
;   1 stop bit
;   8 bit mode
;   null modem config - no handshake
;   Transmit % Receive both enabled. Receive generates CPU interrupt

    bset 0,CONFIG2    ;use internal clock to generate baud rate

;   With prescaler divisor (PD) and baud rate divisor (BD) is given as
;   baud rate = bus freq/(64*PD*BD)

    mov #%00000011,SCBR ;9600 baud
    mov #$40,SCC1      ;enable SCI
    mov #%0010000,SCC2 ;enable RXD interrupts
    bset 3,SCC2        ;set TXD enable bit
    bset 2,SCC2        ;set RXD enable bit

    rts
;-----
send_data:
;   clear the transmitter empty bit. The SCS1 register is
;   read only. The empty bit is automatically cleared by the read operation.
;   thus MCU waits until the transmitter is ready to send data.
;   inputs:    reg0 - 1 byte to be transmitted out the SCI
;   returns:   nothing
;   uses:     reg0
;   destroys:  SCI trans flags

wait:  brcclr 7,SCS1,wait ;wait for TXD ready flag

    mov reg0,SCDR        ;move data into data register of SCI
    rts
;-----
wrSCIstr:
;   writes a string out the serial port
;   address of the string is to be placed in H:X by calling program
;   length of string is the first character in string -
;   inputs:    address of string in H:X
;   returns:   nothing
;   uses:     reg0, H:X, a
;   destroys:  H:X, reg0, flags

```

```

    psha
    mov  X+,reg0    ;get length and point to first character
    lda  reg0       ;store length in acc

sci_loop:  mov  X+,reg0    ;current character = REG ; increment pointer
          jsr  send_data
          dbnza sci_loop   ;continue for all characters
          pula
          rts

```

```

;-----
sci_senddecoded:
;  sends a decoded word (16 bit) out the serial port
;  output is in ASCII character format for each decoded byte
;  inputs:  decode_var(0 through 4) - decoded word
;  returns:  nothing
;  uses:    reg0, decode_var(0..4), a
;  calls:   send_data
;  destroys: reg0, sci trans flags

```

```

    psha

```

```

    lda decode_var+4
    add #$30    ;add 30hex to the number to convert it to ASCII
    sta reg0
    jsr send_data

```

```

    lda decode_var+3
    add #$30    ;add 30hex to the number to convert it to ASCII
    sta reg0
    jsr send_data

```

```

    lda decode_var+2
    add #$30    ;add 30hex to the number to convert it to ASCII
    sta reg0
    jsr send_data

```

```

    lda decode_var+1
    add #$30    ;add 30hex to the number to convert it to ASCII
    sta reg0
    jsr send_data

```

```

    lda decode_var
    add #$30    ;add 30hex to the number to convert it to ASCII
    sta reg0
    jsr send_data

```

```

    pula
    rts

```

```

;-----
RXD_ISR:
;  RXD interrupt service routine - for now output to the LCD
;  places recieved byte in reg0
;  inputs:  one RXD byte from SCI
;  returns:  RXD_byte - received byte (in RAM)
;  uses:    RXD_byte
;  destroys:  RXD_byte, RXD rec flag

```

```

    bclr 5,SCC2      ;Mask all other RXD interrupts during RXD_ISR
    mov  SCS1,RXD_byte    ;need to read this to reset flag

```

```

mov SCDR,RXD_byte      ;get the data from the serial port output
mov RXD_byte,reg0
jsr wrLCDdata;

bset 5,SCC2            ;unmask RCD interrupts
jsr longpulse
rti

```

\*\*\*\*\*END OF PERIPHERALS.INC\*\*\*\*\*

\*\*\*\*\*HEADER FILE FOR 8 BIT 2 LINES 5X7 MATRIX LCD DISPLAY\*\*\*\*\*

```

; To be included with header file Gp20regs.inc
;Filename:      LCD_regs.inc
;Written by D.Kaplen

```

```

DCoffset EQU !126      ;40 , a !126 offset for now (FCB = form constant byte)
Line_freq EQU !50      ;line frequency in Hz
offCur EQU $0C
lineCur EQU $0E
blinkCur EQU $0D
combnCur EQU $0F
homeCur EQU $02
shLfCur EQU $00;$10
shRtCur EQU $00;$14

```

```

; display control instructions

```

```

clrDsp EQU $01
offDsp EQU $0A
onDsp EQU $0E
shLfDsp EQU $18
shRtDsp EQU $1C

```

```

Config EQU $38 ;config for 8 bit data 2 lines, 5x7 matrix display
entryMode EQU $06 ;increment data, do not shift display

```

\*\*\*\*\*END OF LCD\_REGS.INC\*\*\*\*\*

\*\*\*\*\* LCD SUB-ROUTINES FOR OCM\*\*\*\*\*

```

; Filename:      LCD_Subroutines.inc
;Written by D.Kaplen

```

```

;Note - aarious display routines were written for data testing and have since become obsolete. These are not included.

```

```

;; substitutions:

```

```

; 1. LCD_RS = 3,PTB
; 2. LCD_R/W = 2,PTB
; 3. LCD_E = 4,PTC
; Data - High nibble (D4..D7) = 4,PTA..7,PTA
; Data - Low nibble (D0..D3) = 4,PTB..7,PTB

```

```

;-----subroutine DECODE          :decodes Z1:Z0 to bcd-----
;inputs:      Z1:Z0 16 bit word to decode
;outputs:     decode_var(0 to 4) i.e. five bytes
;destroys:    a, flags, XL, XH

```

```

decode:          ;decodes XH:XL to bcd.

```

```

    clr decode_var      ;units
    clr decode_var+1    ;tens

```

```

    clr decode_var+2    ;hundreds
    clr decode_var+3    ;thousands
    clr decode_var+4    ;tenthousands
;need to check if 8 bit i.e. if msbyte if zero

word: tst   Z1    ;dispsmsbyte    ;check if msbyte is neg
      bge   begindecode    ;if positive, then jump to decode

      ldx   Z1    ;dispsmsbyte    ;need to make 16 bit word positive using 2's complement
      lda   Z0    ;displsbyte
      coma                ;1's complement both bytes
      comx
      add   #1    ;add 1 to lsbyte for 2's complement
      psha                ;push to stack
      txa                ;transfer msbyte to acc to add with carry
      adc   #0    ;add carry result to complete 2's complement
      tax                ;transfer back
      pula                ;clean up stack
      sta   Z0    ;displsbyte    ;store back again for decode
      stx   Z1    ;dispsmsbyte
      mov   #'-',reg0
      jsr   wrLCDdata    ;print a "-" before the value to indicate negative
      jmp   begindecode

;byte: tst   Z0    ;displsbyte    ;if get here, working with 8 bit code
;      bge   begindecode    ;need to test if it is negative
;      neg   Z0    ;displsbyte
;      mov   #'-',reg0
;      jsr   wrLCDdata

```

```

begindecode:
  brclr 0,Z0,bit1    ;branch to bit1 if bit0 in dispbyte clr
  inc decode_var    ;if bit0 set, incr units
bit1: brclr 1,Z0,bit2
  inc decode_var
  inc decode_var
bit2: brclr 2,Z0,bit3
  lda decode_var
  add #4
  sta decode_var
bit3: brclr 3,Z0,bit4
  lda decode_var
  add #8
  sta decode_var
bit4: brclr 4,Z0,bit5
  lda decode_var+1
  add #1
  sta decode_var+1
  lda decode_var
  add #6
  sta decode_var
bit5: brclr 5,Z0,bit6
  lda decode_var+1
  add #3
  sta decode_var+1
  inc decode_var
  inc decode_var
bit6: brclr 6,Z0,bit7
  lda decode_var+1

```

```

    add #6
    sta decode_var+1
    lda decode_var
    add #4
    sta decode_var
bit7: brclr 7,Z0,bit8
    inc decode_var+2
    lda decode_var+1
    add #2
    sta decode_var+1
    lda decode_var
    add #8
    sta decode_var

bit8: brclr 0,Z1,bit9
    lda decode_var+2
    add #2
    sta decode_var+2
    lda decode_var+1
    add #5
    sta decode_var+1
    lda decode_var
    add #6
    sta decode_var
bit9: brclr 1,Z1,bit10
    lda decode_var+2
    add #5
    sta decode_var+2
    lda decode_var+1
    add #1
    sta decode_var+1
    lda decode_var
    add #2
    sta decode_var
bit10: brclr 2,Z1,bit11
    lda decode_var+3
    add #1
    sta decode_var+3
    lda decode_var+2
    add #0
    sta decode_var+2
    lda decode_var+1
    add #2
    sta decode_var+1
    lda decode_var
    add #4
    sta decode_var
bit11 brclr 3,Z1,bit12
    lda decode_var+3
    add #2
    sta decode_var+3
    lda decode_var+2
    add #0
    sta decode_var+2
    lda decode_var+1
    add #4
    sta decode_var+1
    lda decode_var
    add #8
    sta decode_var

```

```

bit12: brclr 4,Z1,bit13
      lda decode_var+3
      add #4
      sta decode_var+3
      lda decode_var+2
      add #0
      sta decode_var+2
      lda decode_var+1
      add #9
      sta decode_var+1
      lda decode_var
      add #6
      sta decode_var
bit13: brclr 5,Z1,bit14
      lda decode_var+3
      add #8
      sta decode_var+3
      lda decode_var+2
      add #1
      sta decode_var+2
      lda decode_var+1
      add #9
      sta decode_var+1
      lda decode_var
      add #2
      sta decode_var
bit14: brclr 6,Z1,bit15
      lda decode_var+4
      add #1
      sta decode_var+4
      lda decode_var+3
      add #6
      sta decode_var+3
      lda decode_var+2
      add #3
      sta decode_var+2
      lda decode_var+1
      add #8
      sta decode_var+1
      lda decode_var
      add #4
      sta decode_var
bit15: brclr 7,Z1,rollover
      lda decode_var+4
      add #3
      sta decode_var+4
      lda decode_var+3
      add #2
      sta decode_var+3
      lda decode_var+2
      add #7
      sta decode_var+2
      lda decode_var+1
      add #6
      sta decode_var+1
      lda decode_var
      add #8
      sta decode_var
rollover:
lsd clra

```

```

add #$09
cmp decode_var
bge nsd1
lda decode_var
sub #!10
sta decode_var
inc decode_var+1
bra lsd
nsd1: clra
add #$09
cmp decode_var+1
bge nsd2
lda decode_var+1
sub #!10
sta decode_var+1
inc decode_var+2
bra nsd1
nsd2: clra
add #$09
cmp decode_var+2
bge nsd3
lda decode_var+2
sub #!10
sta decode_var+2
inc decode_var+3
bra nsd2
nsd3: clra
add #$09
cmp decode_var+3
bge done
lda decode_var+3
sub #!10
sta decode_var+3
inc decode_var+4
bra nsd3
done: rts

```

---

display: ;displays a 16 bit word after being decoded above

```

lda decode_var+4
add #$30 ;add 30hex to the number to convert it to ASCII
sta reg0
jsr wrLCDdata
lda decode_var+3
add #$30 ;add 30hex to the number to convert it to ASCII
sta reg0
jsr wrLCDdata
lda decode_var+2
add #$30 ;add 30hex to the number to convert it to ASCII
sta reg0
jsr wrLCDdata
lda decode_var+1
add #$30 ;add 30hex to the number to convert it to ASCII
sta reg0
jsr wrLCDdata
lda decode_var
add #$30 ;add 30hex to the number to convert it to ASCII
sta reg0

```

```

jsr  wrLCDdata
rts

;-----
; subroutine wrLCDdata
; write a data word to the LCD
;
; data must be placed in reg0 by calling program
;-----
wrLCDdata:

    bclr 4,PTC
    bset 3,PTB      ; select send Data
    bclr 2,PTB      ; select write operation
    jsr  shortpulse
    mov  reg0,LCD_HIGH ; load command into port
    rol  reg0      ;rotate left through carry
    rol  reg0      ;rotate left through carry
    rol  reg0      ;rotate left through carry
    rol  reg0      ;rotate left through carry
    mov  reg0,LCD_LOW
    bset 3,PTB
    bclr 2,PTB
    jsr  shortpulse

    jsr  pulseEwait ; pulse the Enable line
    rts            ;ret

;-----
; subroutine wrLCDstr
; display a text string at current cursor position
; address of the string is to be placed in H:X by calling program
; length of string is the first character in string - uses reg0

wrLCDstr:
    psha
    mov  X+,reg0      ;get length and point to first character
    lda  reg0         ;store length in acc

lcd_loop:  mov  X+,reg0 ;write current character to LCD
           jsr  wrLCDdata
           dbnza lcd_loop ;continue for all characters
           pula
           rts

;-----
; subroutine wrLCDcom
; write a command word to the LCD
;
; LCD_RS, LCD_RW, and r0 must be set by
; the calling program
;-----
wrLCDcom:

    bclr 4,PTC
    bclr 3,PTB      ; select send Command
    bclr 2,PTB      ; select Write operation
    jsr  shortpulse
    mov  reg0,LCD_HIGH ; load command into port
    rol  reg0 ;rotate left through carry
    rol  reg0 ;rotate left through carry

```

```

rol reg0 ;rotate left through carry
rol reg0 ;rotate left through carry
mov reg0,LCD_LOW
jsr shortpulse
bclr 3,PTB
bclr 2,PTB
jsr pulseWait ; pulse the Enable line

rts ;ret
;
;-----
; subroutine pulseWait
; generates a positive pulse on the LCD enable line.
; waits for the Busy Flag to clear before returning.
; input : none
; output : none
; destroys : LCD_RW, LCD_RS, LCD_DATA
pulseWait:
bclr 4,PTC
jsr shortpulse
bset 4,PTC ; pulse the Enable line
jsr shortpulse
bclr 4,PTC
rts
;
;-----
; subroutine initLCD - initialize the LCD
initLCD:
bclr 3,PTB ; LCD Register Select line
bclr 2,PTB ; Read / Write line
bclr 4,PTC ; Enable line

mov #$38,reg0 ; Function Set - 2 lines 5x7 fonts
jsr wrLCDcom ;call wrLCDcom

mov #$0F,reg0 ; display ON with bklinking curser
jsr wrLCDcom ;lcall wrLCDcom

mov #$06,reg0 ; set Entry Mode
jsr wrLCDcom ;lcall wrLCDcom ; increment cursor to right, no display shift

mov #$01,reg0 ; clear display, home cursor
jsr wrLCDcom ;lcall wrLCDcom
jsr longpulse

rts ;ret
;
;-----
; subroutine resetLCD - reset the LCD
; software version of the power on reset operation

resetLCD:
bclr 3,PTB ; LCD Register Select line
bclr 2,PTB ; Read / Write line
bclr 4,PTC ; Enable line

mov #$30,reg0 ; step 1
jsr wrLCDcom

mov #Config,reg0 ; FUNCTION SET
jsr wrLCDcom ;lcall wrLCDcom

```

```

mov #$08,reg0      ; display on cursor off
jsr wrLCDcom       ;lcall wrLCDcom

mov #$01,reg0      ; clear display,
jsr wrLCDcom       ;lcall wrLCDcom
jsr longpulse

mov #$02,reg0      ;home cursor
jsr wrLCDcom
jsr longpulse

mov #entryMode,reg0 ; set ENTRY MODE
jsr wrLCDcom       ;lcall wrLCDcom       ; increment cursor, do not shift display

rts                ;ret

```

\*\*\*\*\*END OF LCD\_SUBROUTINES.INC\*\*\*\*\*

## **Borland C Data Logging Software**

/\*Inter.c - An RS232 protocol Interrupt Driven Data Logger Comms Program

Written by D.Kaplen (09/2000) exclusively for data logging as part of a MSc dissertation project entitled "The Development of an On-Chip-Metering Solution"

```

*/
#include <conio.h>
#include <string.h>
#include <dos.h>
#include <stdio.h>
#include <errno.h>
#include <math.h>
#define INV_COM 34 //ASCII for ""

#define CR 10 //carriage return
#define LF 13 //line feed
#define PORT1 0x2F8 /* Port Address Goes Here - using COM2 for now */
/* Defines Serial Ports Base Address */
/* COM1 0x3F8 */
/* COM2 0x2F8 */
/* COM3 0x3E8 */
/* COM4 0x2E8 */

#define INTVECT 0x0C /* Com Port's IRQ here - using COM1 for now */

/* (Must also change PIC setting) */
int com_num;
int bufferin = 0, bufferout = 0;
unsigned int address[4], PORT1, INTVECT, /* Address of Port */
old_port0, old_port1, old_port2, old_port3, old_port4 ; //to save previous settings
unsigned int a,
i,
a_vari,
c=0,
b_vari,
file_flag = 0, //flag for file status
prog_int, //for the programmable interrupt
bongo;

```

```

FILE *out_file;
unsigned char ch, dummy;
unsigned char buffer[1025];

void interrupt (*oldport1isr)();

void portid(void)
{
    unsigned int far *ptraddr; /* Pointer to location of Port Addresses */
    ptraddr=(unsigned int far *)0x00000400;
    for (a = 0; a < 4; a++)
    {
        address[a] = *ptraddr;
        if (address[a] == 0)
            printf("No port found for COM%d \n",a+1);
        else
            printf("Address assigned to COM%d is %Xh\n",a+1,address[a]);
        *ptraddr++;
    }

    printf("\nSelect COM port => "); //user selectable comm port
    dummy = getche();
    com_num = dummy - '0'; //convert to integer
    while ((com_num<1)||((com_num>4)||((address[com_num-1]==0)))
    {
        printf("\nInvalid Port, please select another => ");
        dummy = getche();
        com_num = dummy - '0';
        if (dummy == 27) break;
    }
    PORT1 = address[com_num-1]; //select port address
    printf(" ... COM%c selected (%Xh)",dummy,PORT1);
    if (dummy == '1') {INTVECT = 0x0C; prog_int =0xEF; } //set comm address
    if (dummy == '2') {INTVECT = 0x0B; prog_int =0xF7; }
    if (dummy == '3') {INTVECT = 0x0C; prog_int =0xEF; }
    if (dummy == '4') {INTVECT = 0x0B; prog_int =0xF7; }
}

void interrupt PORT1INT() // Interrupt Service Routine (ISR) for PORT1
{
    int c;

    do { c = inportb(PORT1 + 5);
        if (c & 1)
        {
            buffer[bufferin] = inportb(PORT1);
            bufferin++;
            if (bufferin == 1024) bufferin = 0; //1k byte buffer
        }
    } while (c & 1);

    outportb(0x20,0x20);
}

void portsetup() //setup the chosen port for comms
{
    old_port0 = inportb(PORT1 + 0); //save port's previous settings
    old_port1 = inportb(PORT1 + 1);
    old_port2 = inportb(PORT1 + 2);
}

```

```

old_port3 = inportb(PORT1 + 3);
old_port4 = inportb(PORT1 + 4);

outportb(PORT1 + 1, 0); // disable port interrupts

oldportl_isr = getvect(INTVECT); // Save old Interrupt Vector for later recovery
setvect(INTVECT, PORT1INT); /* Set Interrupt Vector Entry */
/* COM1 - 0x0C */
/* COM2 - 0x0B */
/* COM3 - 0x0C */
/* COM4 - 0x0B */

/* PORT 1 - Communication Settings */

outportb(PORT1 + 3, 0x80); /* SET DLAB ON */
outportb(PORT1 + 0, 0x0C); /* Set Baud rate - Divisor Latch Low Byte */
/* Default 0x03 = 38,400 BPS */
/* 0x01 = 115,200 BPS */
/* 0x02 = 56,700 BPS */
/* 0x06 = 19,200 BPS */
/* 0x0C = 9,600 BPS */
/* 0x18 = 4,800 BPS */
/* 0x30 = 2,400 BPS */

outportb(PORT1 + 1, 0x00); /* Set Baud rate - Divisor Latch High Byte */
outportb(PORT1 + 3, 0x03); /* 8 Bits, No Parity, 1 Stop Bit */
outportb(PORT1 + 2, 0xC7); /* FIFO Control Register */
outportb(PORT1 + 4, 0x0B); /* Turn on DTR, RTS, and OUT2 */

outportb(0x21,(inportb(0x21) & prog_int)); //0xEF); /* Set Programmable Interrupt */
/* Controller */
/* COM1 (IRQ4) - 0xEF */
/* COM2 (IRQ3) - 0xF7 */
/* COM3 (IRQ4) - 0xEF */
/* COM4 (IRQ3) - 0xF7 */

outportb(PORT1 + 1, 0x01); /* Interrupt when data received */
}

int filestuff() /*opens a file, returns a 1 if error, else 0
{
    char *p_name, filename[20]; //max file name length
    // int ch;
    printf("\nOpen file?? ");
    c = getch();
    if (c == 'y')
    {
        printf("\nEnter Output Filename => ");
        if((p_name = strchr(fgets(filename, 20, stdin), '\n')) != NULL)
            *p_name = '\0'; //if filename contains newline (i.e. CR)
            //then replace it with \0 - NULL
        while ((out_file = fopen(filename, "w")) == NULL) //open output file
        {
            printf("\nUnable to open file %s: errno =%d", filename, errno);
            perror(" message ");
            printf("\nTry another (y/n)? ");
            c = getche();
            if (c != 'y')
                { file_flag = 1; printf("\n\nWarning! - file not opened...");
                return 1; //file not opened
            }
        }
    }
}

```

```

    }
}
printf("\nFile opened OK");
return 0; //file now opened okay!
}
else return 1;
}

void emulator()          //a terminal emulator for testing
{
    do {
        if (bufferin != bufferout)
        {
            bongo = buffer[bufferout]; //was ch
            bufferout++;
            if (bufferout == 1024) bufferout = 0;
            if (bongo != 252) printf("%c", bongo);
        }
        if (kbhit())
        {
            c = getch();
            outportb(PORT1, c);
        }
    } while (c != 27);
}

void main(void)
{
    char msg[] = "Voltage Samples";
    char msg2[] = "Current Samples";
    char msg3[] = "Summed Value";
    char Summed_Val_Out[13];
    long int Pwr_nibbles[9];

    long int Summed_Value;
    int i = 0;

    clrscr();
    printf("Motorola 68HC908 Based Serial Comms Data Logger\n");
    printf("Written by David Kaplen\nElectrical Engineering Department\n");
    printf("University of Cape Town\n\nDetecting COM Settings...\n\n");
    portid(); //identify port address and IRQ
    portsetup(); //set baud rate, parity, handshake, etc

    if( (file_flag = filestuff() ) == 0) //if file opened
    {
        printf("\nInputting data...\n");

        putc(INV_COM, out_file);
        while (msg[i]) putc(msg[i++], out_file);
        putc(INV_COM, out_file);
        putc(',', out_file);
        i = 0;
        putc(INV_COM, out_file);
        while (msg2[i]) putc(msg2[i++], out_file);
        putc(INV_COM, out_file);
        putc('\n', out_file);

        do

```

```

{
a = 0;
i = 0;
printf("\nPress any key to continue or esc to exit ");
c = getch();
if(c==27)
{ fclose(out_file);
printf("\nFile Closed OK");
outportb(PORT1 + 1 , 0); /* Turn off interrupts - Port1 */
outportb(0x21,(inportb(0x21) | 0x10)); /* MASK IRQ using PIC */
/* COM1 (IRQ4) - 0x10 */
/* COM2 (IRQ3) - 0x08 */
/* COM3 (IRQ4) - 0x10 */
/* COM4 (IRQ3) - 0x08 */

setvect(INTVECT, oldport1_isr); /* Restore old interrupt vector */
outportb(PORT1 + 0 ,old_port0); //Restore old port settings
outportb(PORT1 + 1, old_port1);
outportb(PORT1 + 2, old_port2);
outportb(PORT1 + 3, old_port3);
outportb(PORT1 + 4, old_port4);
printf("\nSerial Comms Closed OK");
return; /*if esc pressed, exit
}
else outportb(PORT1, c);

printf("\nWaiting for Samples\n");
c = getch(); //flag to continue
outportb(PORT1,c);
ch = 0; //initialise ch
do {
// if (kbhit())
// { c = getch();
// printf("%d",c);
// outportb(PORT1, c);}

if (bufferin != bufferout)
{
/* Here a series of 16 bit words is collected. Each word received as
5 separate bytes representing Ten thousands, thousands, hundreds, ten, units
These are concatenated into single strings resepectively. Before and after
each 5-byte segment, an inverted comma is placed. After the closing " of
each 5 byte segment, a newline is also placed. All this done to conform
to the csv format. After the specified number of received bytes, or
untill the ASCII code $01 is received, the file is closed.

*/

ch = buffer[bufferout];//was ch
bufferout++;
if (bufferout == 1024) bufferout = 0; //if get a byte
i = i++;
if (ch!=1)
{
if ((i==1)||(i==6)) putc(INV_COM,out_file); //inverted comma
if ((i!=1)||(i!=5)||(i!=6)||(i!=10)) printf("%c",ch);
if ((ch!=1)||(ch!=252)) putc(ch,out_file); //if not end of trans
if ((i==5)||(i==10)) putc(INV_COM,out_file); //inverted comma
if (i==5) {printf(" ");putc(',',out_file);}
if (i==10) {printf("\n"); putc('\n',out_file); i=0;}

```

```

    }
} while(ch!=1);
// printf("Waiting for Samples\n");
a = 0;
i = 0;
ch = 0;    //initialise
do {

    if (bufferin != bufferout)
    {
        ch = buffer[bufferout];
        bufferout++;
        if (bufferout == 1024) bufferout = 0;    //if get a byte

        if (ch=='0') Pwr_nibbles[i]=0;
        else if(ch=='1') Pwr_nibbles[i]=1;
        else if(ch=='2') Pwr_nibbles[i]=2;
        else if(ch=='3') Pwr_nibbles[i]=3;
        else if(ch=='4') Pwr_nibbles[i]=4;
        else if(ch=='5') Pwr_nibbles[i]=5;
        else if(ch=='6') Pwr_nibbles[i]=6;
        else if(ch=='7') Pwr_nibbles[i]=7;
        else if(ch=='8') Pwr_nibbles[i]=8;
        else if(ch=='9') Pwr_nibbles[i]=9;
        else if(ch==':') Pwr_nibbles[i]=10;
        else if(ch==';') Pwr_nibbles[i]=11;
        else if(ch=='<') Pwr_nibbles[i]=12;
        else if(ch=='=') Pwr_nibbles[i]=13;
        else if(ch=='>') Pwr_nibbles[i]=14;
        else if(ch=='?') Pwr_nibbles[i]=15;
        // printf("Pwr_nibble[%d] = %ld\n",i,Pwr_nibbles[i]);
        i = i++;
    }
} while(ch!=1);

if (Pwr_nibbles[0]<7) //if positive
{ Summed_Value = (Pwr_nibbles[0]*268435456) + (Pwr_nibbles[1]*16777216)
    + (Pwr_nibbles[2]*1048576) + (Pwr_nibbles[3]*65536)
    + (Pwr_nibbles[4]*4096) + (Pwr_nibbles[5]*256)
    + (Pwr_nibbles[6]*16) + (Pwr_nibbles[7]*1);
}

else //negative value detected
{ printf("\nNeg value detected"); //if summed value is negative
  for(i=0;i<8;i++) Pwr_nibbles[i] = 15-Pwr_nibbles[i]; //complement nibbles

  Pwr_nibbles[7] = Pwr_nibbles[7]+1;    //add 1 for two's complement conversion
  i=7;
  do
  {
    if (Pwr_nibbles[i]>15)
    { Pwr_nibbles[i] = 0;
      Pwr_nibbles[i-1] = Pwr_nibbles[i-1]+1;
    }
    i=i-1;
  } while(i>=0);

  Summed_Value =-( (Pwr_nibbles[0]*268435456) + (Pwr_nibbles[1]*16777216)

```

University of Cape Town

```

        + (Pwr_nibbles[2]*1048576) + (Pwr_nibbles[3]*65536)
        + (Pwr_nibbles[4]*4096) + (Pwr_nibbles[5]*256)
        + (Pwr_nibbles[6]*16) + (Pwr_nibbles[7]*1));
    }

    printf("\n\n Summed Value = > %ld",Summed_Value);

    i = 0;
    putc(INV_COM,out_file);
    while (msg3[i]) putc(msg3[i++],out_file);
    putc(INV_COM,out_file);
    putc(',',out_file);
    ltoa (Summed_Value,Summed_Val_Out,10);    //convert to string
    i = 0;
    putc(INV_COM,out_file);
    while (Summed_Val_Out[i]) putc(Summed_Val_Out[i++],out_file);
    putc(INV_COM,out_file);
    putc('\n',out_file);
} while(c!=27);    //infinite loop, stop when esc key

fclose(out_file);
printf("\n\nFile closed successfully");

outputb(PORT1, 'O');
outputb(PORT1, 'K');

}
else if(file_flag == 1)
    {printf("\nUse emulator??");
    if ((ch = getche())=='y')
        { printf("\nEmulator started...terminate with ESC\n\n");
        emulator();
        }
    }
outputb(PORT1 + 1 , 0); /* Turn off interrupts - Port1 */
outputb(0x21,(inportb(0x21) | 0x10)); /* MASK IRQ using PIC */
/* COM1 (IRQ4) - 0x10 */
/* COM2 (IRQ3) - 0x08 */
/* COM3 (IRQ4) - 0x10 */
/* COM4 (IRQ3) - 0x08 */

setvect(INTVECT, oldport1isr); /* Restore old interrupt vector */
outputb(PORT1 + 0 ,old_port0); //Restore old port settings
outputb(PORT1 + 1, old_port1);
outputb(PORT1 + 2, old_port2);
outputb(PORT1 + 3, old_port3);
outputb(PORT1 + 4, old_port4);
getch();

}

```

/\*NOTE - check to see if there is a character to receive and if there is remove it from the UART and place it in a buffer contained in memory. keep on checking the UART, in case FIFO's are enabled, so all data available at the time of interrupt is retrieved. The instruction outputb(0x20,0x20) tells the Programmable Interrupt Controller that the interrupt has finished. \*/

## **Annexure C: Test Equipment Specifications**

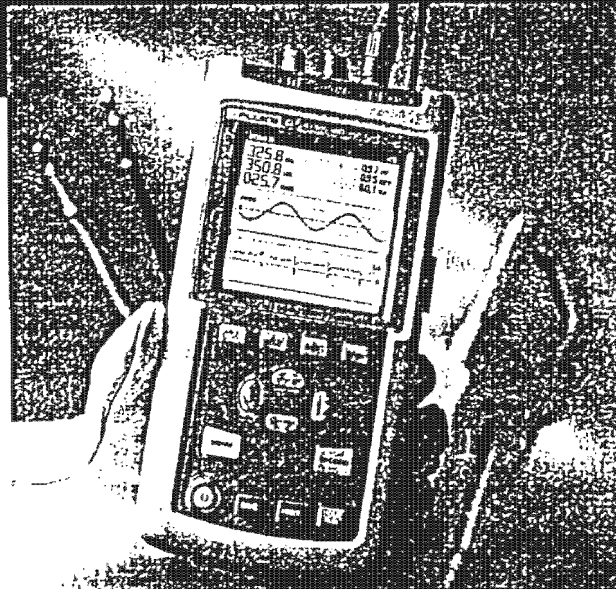
All relevant specifications for the respective testing equipment, as supplied by the manufacturers, used throughout this project are presented in this annexure.

University of Cape Town

**FLUKE®**

# Power Quality Analyzer

**43**



○ Users Manual

## Safety Specifications

### Safety Characteristics

Designed and tested for measurements on 600 V rms Category III, Pollution Degree 2 in accordance with:

- EN 61010.1 (1993) (IEC 1010-1)
- ANSI/ISA S82.01-1994
- CAN/CSA-C22.2 No. 1010.1-92 (including approval)
- UL3111-1 (including approval)

Installation Category III refers to distribution level and fixed installation circuits inside a building.

### ⚠ Maximum input voltage Input 1 and 2

Direct on inputs or with test leads TL24	(see Figure 11)
0 to 66 kHz	600 V rms
> 66 kHz	derating to 5 V rms
With Shielded Banana-to-BNC Adapter Plug BB120	(see Figure 11)
0 to 400 kHz	300 V rms
> 400 kHz	derating to 5 V rms

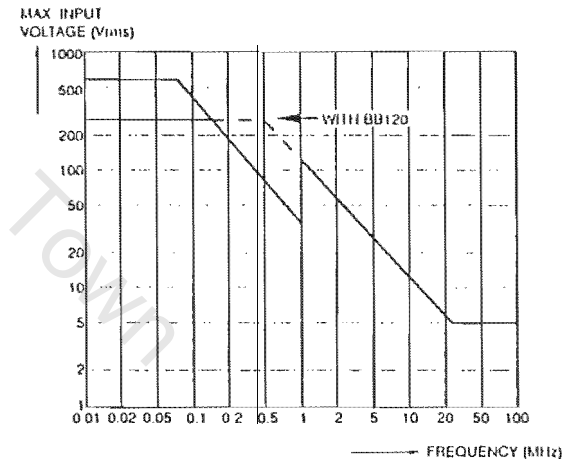


Figure 11. Max. Input Voltage v.s. Frequency

### ⚠ Maximum floating voltage

From any terminal to ground	
0 to 400 Hz	600 V rms

## Function Specifications

For all specifications, probe specifications must be added.

### Electrical functions

Specifications are valid for signals with a fundamental between 40 and 70 Hz.

Minimum input voltage ..... 4 V peak-peak

Minimum input current ..... 10 A peak-peak (1 mV/A)

Input bandwidth ..... DC to 15 kHz (unless specified otherwise)

#### Volts / Amps / Hertz

Readings ..... V rms (AC+DC), A rms (AC+DC), Hz

Voltage ranges (auto) ..... 5.000 V to 500.0 V, 1250 V

±(1 % + 10 counts)

Current ranges (auto) ..... 50.00 A to 500.0 kA, 1250 kA

±(1 % + 10 counts)

Frequency range ..... 10.0 Hz to 15.0 kHz

40.0 to 70.0 Hz ..... ±(0.5 % + 2 counts)

#### Power

Readings ..... Watt, VA, VAR, PF, DPF, Hz

Watt, VA, VAR ranges (auto) ..... 250 W to 250 MW, 625 MW, 1.56 GW

when selected: total (%): ..... ±(2 % + 6 counts)

when selected: fundamental (%f): ..... ±(4 % + 4 counts)

DPF ..... 0.00 to 1.00

0.00 to 0.25 ..... not specified

0.25 to 0.90 ..... ± 0.04

0.90 to 1.00 ..... ± 0.03

PF ..... 0.00 to 1.00

± 0.04

Frequency range ..... 10.0 Hz to 15.0 kHz

40.0 to 70.0 Hz ..... ±(0.5 % + 2 counts)

#### Harmonics

Number of harmonics ..... DC..21, DC..33, DC..51

#### Readings / Cursor readings

V rms / I rms ..... fund. ±(3 % + 2 counts)

31" ±(5 % + 3 counts)

51" ±(15 % + 5 counts)

Watt ..... fund. ±(5 % + 10 counts)

31" ±(10 % + 10 counts)

51" ±(30 % + 5 counts)

Frequency of fundamental ..... ± 0.25 Hz

## Current Probe

### ⚠ Safety Characteristics

Designed for measurements on 600 V rms Category III. Protection class II, double or reinforced insulation requirements in accordance with:

- IEC 1010-1
- ANSI/ISA S82
- CSA-C22.2 No.1010.1-92
- UL1244

### Electrical Specifications

Current range ..... 1A to 500 A rms

AC current over range limit ..... 700 A rms

Maximum 10 minutes, followed by removal from current carrying conductor for 30 minutes.

Output Signal ..... 1mV AC/A AC

### Accuracy

5 to 10 Hz

1 to 500 A ..... -3 dB typically

10 to 20 Hz

1 to 300 A ..... ± 5 %

300 to 400 A ..... ± 15 %

400 to 500 A ..... ± 25 %

20 to 45 Hz

1 to 500 A ..... ± 5 %

45 to 65 Hz

1 to 20 A ..... ±5 % of reading + 0.3 A

20 to 100 A ..... ±5 % of reading

± 3° phase shift

100 to 500 A ..... ±2 % of reading

± 5° phase shift

65 Hz to 3 kHz

1 to 50 A ..... ±(5 % + 0.4 A)

50 to 500 A ..... ± 5 %

Influence of temperature on accuracy ..... <0.15 % per 10 °C (18 °F)

### Altitude

During operation ..... 2.0 km (6560 feet)

While stored ..... 12 km (40 000 feet)

### Scope

#### Input Impedance

Input 1.....	1 M $\Omega$ // 12 pF ( $\pm$ 2 pF)
Input 2.....	1 M $\Omega$ // 10 pF ( $\pm$ 2 pF)

#### Horizontal

Time base modes (selectable)..... Normal, Single, Roll

#### Ranges (selectable within modes)

In Normal.....	5 s to 20 ns/div
In Single shot.....	5 s to 1 $\mu$ s/div
In Roll mode.....	60 s to 1 s/div

Time base error..... <  $\pm$ (0.4 % + 1 pixel)

#### Maximum sampling rate

10 ms to 60 s.....	5 MS/s
20 ns to 10 ms.....	25 MS/s

Trigger source (auto)..... Input 1 or Input 2

#### Vertical

Voltage ranges..... 5.0 mV/div to 500 V/div  
Trace accuracy.....  $\pm$ (1 % + 2 pixels)

#### Bandwidth Input 1 (voltage)

excluding test leads or probes.....	DC to 20 MHz (-3 dB)
with test leads TL24.....	DC to 1 MHz (-3 dB)
with 10:1 probe PM8918 (optional).....	DC to 20 MHz (-3 dB)
with shielded test leads STL120 (optional).....	DC to 12.5 MHz (-3 dB)
	DC to 20 MHz (-6 dB)
Lower transition point (AC coupling).....	10 Hz (-3 dB)

#### Bandwidth input 2 (current)

with Banana-to-BNC adapter.....	DC to 15 kHz
Lower transition point (AC coupling).....	10 Hz (-3 dB)

### Scope readings

The accuracy of all scope readings is valid from 18 °C to 28 °C with relative humidity up to 90 % for a period of one year after calibration. Add 0.1 x (the specified accuracy) for each °C below 18 °C or above 28 °C. More than one waveform period must be visible on the screen.

V DC, A DC.....  $\pm$ (0.5 % + 5 counts)

#### V AC and V AC+DC (True RMS) input 1

DC to 60 Hz.....	$\pm$ (1 % + 10 counts)
60 Hz to 20 kHz.....	$\pm$ (2.5 % + 15 counts)
20 kHz to 1 MHz.....	$\pm$ (5 % + 20 counts)
1 MHz to 5 MHz.....	$\pm$ (10 % + 25 counts)
5 MHz to 20 MHz.....	$\pm$ (30 % + 25 counts)

#### A AC and A AC+DC (True RMS) input 2

DC to 60 Hz.....	$\pm$ (1 % + 10 counts)
60 Hz to 15 kHz.....	$\pm$ (30 % + 25 counts)

#### Frequency (Hz), Pulse width, Duty cycle (2.0 % to 98.0 %)

1 Hz to 1 MHz.....	$\pm$ (0.5 % + 2 counts)
1 MHz to 10 MHz.....	$\pm$ (1 % + 2 counts)
10 MHz to 30 MHz.....	$\pm$ (2.5 % + 2 counts)

#### Phase (Input 1 to Input 2)

1 Hz to 400 Hz.....	$\pm$ 2°
---------------------	----------

#### Peak voltage

Peak max, Peak min.....	$\pm$ 5 % of full scale
Peak-peak.....	$\pm$ 10 % of full scale

#### Crest

Range.....	1.0 to 10.0
	$\pm$ (5 % + 1 counts)

## Miscellaneous

### Display

Useful screen area .....	72 x 72 mm (2.83 x 2.83 in)
Resolution .....	240 x 240 pixels
Backlight.....	Cold Cathode Fluorescent (CCFL)

### ⚠ Power

#### External

Power Adapter.....	PM8907
Input Voltage .....	10 to 21V DC
Power .....	5 W typical

#### Internal

Rechargeable Ni-Cd battery pack.....	BP120
Voltage range .....	4 to 6 V DC
Operating Time.....	4 hours
Charging Time.....	4 hours with Fluke 43 off 12 hours with Fluke 43 on
Refresh cycle.....	8 to 14 hours

### Memory

Number of screen memories.....	10
Number of transient memories (temporary) .....	40

### Mechanical

Height x width x depth.....	232 x 115 x 50 mm (9.1 x 4.5 x 2 in)
Weight (including battery pack).....	1.1 kg (2.5 lbs)

### Interface .....

Supported Printers .....HP Deskjet®, Laserjet®, PostScript and Epson FX80

Serial via PM9080 (optically isolated RS232 Adapter/Cable).

Parallel via PAC91 (optically isolated Print Adapter Cable, optional).

To PC .....

Serial via PM9080 (optically isolated RS232 adapter/cable),  
using SW43W (FlukeView® Power Quality Analyzer software).

## Electromagnetic Immunity

The Fluke 43, including standard accessories, conforms with the EEC directive 89/336 for EMC immunity, as defined by IEC1000-4-3, with the addition of the following tables.

### Disturbance with test leads TL24 or Current Clamp 801-500s

- Volts / amps / hertz
- Resistance, Capacitance
- Power
- Harmonics

Table 1

No visible disturbance	E = 3 V/m	E = 10 V/m
Frequency: 10 kHz - 27 MHz	(-)	(-)
Frequency: 27 MHz - 1 GHz	(-)	(-)

(-): no visible disturbance

### Disturbance with test leads TL24 In scope mode

- V AC+DC (True RMS)

Table 2

Disturbance less than 1 % of full scale	E = 3 V/m	E = 10 V/m
Frequency: 10 kHz - 27 MHz	2 V/div - 500 V/div	10 V/div - 500 V/div
Frequency: 27 MHz - 200 MHz	500 mV/div - 500 V/div	2 V/div - 500 V/div
Frequency: 200 MHz - 1 GHz	(-)	5 mV/div - 500 V/div

(-): no visible disturbance

Table 3

Disturbance less than 10 % of full scale	E = 3 V/m	E = 10 V/m
Frequency: 10 kHz - 27 MHz	1 V/div	5 V/div
Frequency: 27 MHz - 200 MHz	200 mV/div	1 V/div
Frequency: 200 MHz - 1 GHz	(-)	(-)

(-): no visible disturbance

Ranges not specified in Tables 2 and 3 may have a disturbance of more than 10 % of full scale.

# HP 33120A - Function / Arbitrary Waveform Generator

## WAVEFORMS

**Standard Waveforms:** Sine, Square, Triangle, Ramp, Noise, DC volts, Sine(x)/x, Negative Ramp, Exponential Rise, Exponential Fall, Cardiac

**Arbitrary Waveforms:**  
 Waveform Length: 8 to 16,000 points  
 Amplitude Resolution: 12 bits (including sign)  
 Sample Rate: 40 MSa / sec  
 Non-Volatile Memory: Four (4) 16k waveforms

## FREQUENCY CHARACTERISTICS

**Sine:** 100  $\mu$ Hz – 15 MHz  
**Square:** 100  $\mu$ Hz – 15 MHz  
**Triangle:** 100  $\mu$ Hz – 100 kHz  
**Ramp:** 100  $\mu$ Hz – 100 kHz  
**Noise (Gaussian):** 10 MHz bandwidth  
**Arbitrary Waveforms:**  
 8 to 8,192 points: 100  $\mu$ Hz – 5 MHz  
 8,193 to 12,287 points: 100  $\mu$ Hz – 2.5 MHz  
 12,288 to 16,000 points: 100  $\mu$ Hz – 200 kHz

**Resolution:** 10  $\mu$ Hz or 10 digits

**Accuracy:** 10 ppm in 90 days,  
 20 ppm in 1 year,  
 18°C – 28°C

**Temperature Coefficient:** < 2 ppm / °C

**Aging:** < 10 ppm / yr

## SINEWAVE SPECTRAL PURITY (into 50 $\Omega$ )

**Harmonic Distortion**  
 DC to 20 kHz: -70 dBc  
 20 kHz to 100 kHz: -60 dBc  
 100 kHz to 1 MHz: -45 dBc  
 1 MHz to 15 MHz: -35 dBc

**Total Harmonic Distortion**  
 DC to 20 kHz: 0.04%

**Spurious (non-harmonic)**  
 Output (DC to 1 MHz): < -65 dBc  
 Output (> 1 MHz): < -65 dBc + 6 dB/octave

**Phase Noise:** < -55 dBc in a 30 kHz band

## SIGNAL CHARACTERISTICS

**Square wave**  
 Rise/Fall Time: < 20 ns  
 Overshoot: < 4%  
 Asymmetry: 1% + 5 ns  
 Duty Cycle: 20% to 80% (to 5 MHz)  
 40% to 60% (to 15 MHz)

**Triangle, Ramp, Arb**  
 Rise/Fall Time: 100 ns (typical)  
 Linearity: < 0.1% of peak output  
 Settling Time: < 250 ns to 0.5% of final value  
 Jitter: < 25 ns

## OUTPUT CHARACTERISTICS (1)

**Amplitude (into 50 $\Omega$ ): (2)** 50 mVpp – 10 Vpp  
**Accuracy (at 1 kHz):**  $\pm$  1% of specified output  
**Flatness** (sine wave relative to 1 kHz)  
 < 100 kHz:  $\pm$  1% (0.1 dB)  
 100 kHz to 1 MHz:  $\pm$  1.5% (0.15 dB)  
 1 Mz to 15 MHz:  $\pm$  2% (0.2 dB) Ampl  $\geq$  3Vrms  
 1 Mz to 15 MHz:  $\pm$  3.5% (0.3 dB) Ampl < 3Vrms

**Offset (into 50 $\Omega$ ): (3)**  $\pm$  5 Vpk ac + dc  
**Accuracy: (4)**  $\pm$  2% of setting + 2 mV

**Output Impedance:** 50 ohms fixed

**Resolution:** 3 digits, Amplitude and Offset

**Output Units:** Vpp, Vrms, dBm

**Isolation:** 42 Vpk maximum to earth

**Protection:** Short-circuit protected  
 $\pm$  15 Vpk overdrive < 1 minute

(1) Add 1/10th of output amplitude and offset specification per °C for operation outside of 18°C to 28°C range (1-year specification).

(2) 100 mVpp – 20 Vpp amplitude into open-circuit load.

(3) Offset  $\leq$  2 X peak-to-peak amplitude.

(4) For square wave outputs, add 2% of output amplitude additional error.

## MODULATION CHARACTERISTICS

### AM Modulation

Carrier -3 dB Freq: 15 MHz (typical)  
 Modulation: Any internal waveform plus Arb  
 Frequency: 10 mHz to 20 kHz ( $\pm 0.05\%$  to 2.5 kHz, then decreases linearly to  $\pm 0.4\%$  at upper limit)  
 Depth: 0% to 120%  
 Source: Internal / External

### FM Modulation

Modulation: Any internal waveform plus Arb  
 Frequency: 10 mHz to 10 kHz ( $\pm 0.05\%$  to 500 Hz, then decreases linearly to  $\pm 0.8\%$  at upper limit)  
 Peak Deviation: 10 mHz to 15 MHz  
 Source: Internal Only

### Burst Modulation

Carrier Frequency: 5 MHz max.  
 Count: 1 to 50,000 cycles, or Infinite  
 Start Phase:  $-330^\circ$  to  $+330^\circ$   
 Internal Rate: 10 mHz to 50 kHz  $\pm 1\%$   
 Gate Source: Internal or External Gate (1)  
 Trigger Source: Single, External, or Internal Rate

### FSK Modulation

Frequency Range: 10 mHz to 15 MHz ( $\pm 0.05\%$  to 600 Hz, then decreases linearly to  $\pm 4\%$  at upper limit)  
 Internal Rate: 10 mHz to 50 kHz  
 Source: Internal / External (1 MHz max.)

## FREQUENCY SWEEP

Type: Linear or Logarithmic  
 Direction: Up or Down  
 Start F / Stop F: 10 mHz to 15 MHz  
 Time: 1 ms to 500 sec  $\pm 0.1\%$   
 Source: Single, External, or Internal

## REAR-PANEL INPUTS

External AM Modulation:  $\pm 5$  Vpk = 100% Modulation  
 5 k $\Omega$  Input Resistance

### External Trigger/FSK

Burst Gate: (1) TTL (low true)  
 Latency: 1.3  $\mu$ s  
 Jitter: 25 ns

## SYSTEM CHARACTERISTICS

### Configuration Times (2)

Function Change: (3) 80 ms  
 Frequency Change: (3) 30 ms  
 Amplitude Change: 30 ms  
 Offset Change: 10 ms  
 Select User Arr: 100 ms  
 Modulation Parameter Change: < 350 ms

### Arb Download Times over HP-IB:

Arb Length	Binary	ASCII Integer	ASCII Real (4)
16,000 points	8 sec	81 sec	100 sec
8,192 points	4 sec	42 sec	51 sec
4,096 points	2.5 sec	21 sec	26 sec
2,048 points	1.5 sec	11 sec	13 sec

### Arb Download Times over RS-232 at 9600 Baud: (5)

Arb Length	Binary	ASCII Integer	ASCII Real (6)
16,000 points	35 sec	101 sec	134 sec
8,192 points	18 sec	52 sec	69 sec
4,096 points	10 sec	27 sec	35 sec
2,048 points	6 sec	14 sec	18 sec

(1) Trigger source ignored when External Gate is selected.

(2) Time to change parameter and output the new signal.

(3) Modulation or sweep off.

(4) Times for 5-digit and 12-digit numbers.

(5) For 4800 baud, multiply the download times by two;  
 For 2400 baud, multiply the download times by four, etc.

(6) Time for 5-digit numbers. For 12-digit numbers,  
 multiply the 5-digit numbers by two.

# HP 34401A - True-RMS Calculating Digital Multimeter

## Chapter 1 Specifications DC Characteristics



### Measuring Characteristics

<b>DC Voltage</b>	
Measurement Method:	Continuously integrating, multi-slope III A/D converter.
A/D Linearity:	0.0002% of reading + 0.0001% of range
Input Resistance:	Selectable 10 M $\Omega$ or >10 G $\Omega$
0.1 V, 1 V, 10 V ranges	10 M $\Omega$ $\pm$ 1%
100 V, 1000 V ranges	< 30 pA at 25°C
Input Bias Current:	Copper alloy
Input Terminals:	1000 V on all ranges
Input Protection:	

<b>Resistance</b>	
Measurement Method:	Selectable 4-wire or 2-wire ohms. Current source referenced to LO input.
Max. Lead Resistance: (4-wire ohms)	10% of range per lead for 100 $\Omega$ , 1 k $\Omega$ ranges. 1 k $\Omega$ per lead on all other ranges.
Input Protection:	1000 V on all ranges

<b>DC Current</b>	
Shunt Resistor:	0.1 $\Omega$ for 1A, 3A. 5 $\Omega$ for 10 mA, 100 mA
Input Protection:	Externally accessible 3A, 250 V fuse Internal 7A, 250 V fuse

<b>Continuity / Diode Test</b>	
Response Time:	300 samples/sec with audible tone
Continuity Threshold:	Adjustable from 1 $\Omega$ to 1000 $\Omega$

<b>DC:DC Ratio</b>	
Measurement Method:	Input HI-LO / Reference HI-LO
Input HI-LO	100 mV to 1000 V ranges
Reference HI-Input LO	100 mV to 10 V ranges (autoranged)
Input to Reference	Reference LO to Input LO voltage < 2 V Reference HI to Input LO voltage < 12 V

<b>Measurement Noise Rejection</b>	
<b>60 Hz (50 Hz) [5]</b>	
DC CMRR	140 dB
<b>Integration Time</b>	
100 PLC / 1.67s (2s)	Normal Mode Rejection [6]
10 PLC / 167 ms (200 ms)	70 dB [7]
1 PLC / 16.7 ms (20 ms)	60 dB [7]
<1 PLC / 3 ms (800 $\mu$ s)	60 dB
	0 dB

### Operating Characteristics [8]

Function	Digits	Readings/s	Additional Noise Error
DCV, DCI, and Resistance	6 1/2	0.6 (0.5)	0% of range
	6 1/2	6 (5)	0% of range
	5 1/2	60 (50)	0.001% of range
	5 1/2	300*	0.001% of range [10]
	4 1/2	1000	0.01% of range [10]

<b>System Speeds [9]</b>	
Function Change	26/sec
Range Change	50/sec
Autorange Time	<30 ms
ASCII readings to RS-232	55/sec
ASCII readings to HP-IB	1000/sec
Max. Internal Trigger Rate	1000/sec
Max. External Trigger Rate to Memory	1000/sec
Max. External Trigger Rate to HP-IB	900/sec

**Autozero OFF Operation**  
Following instrument warm-up at calibration temperature  $\pm$ 1°C and <10 minutes, add 0.0002% range additional error + 5  $\mu$ V.

**Settling Considerations**  
Reading settling times are affected by source impedance, cable dielectric characteristics, and input signal changes.

**Measurement Considerations**  
HP recommends the use of Teflon<sup>®</sup> or other high-impedance, low-dielectric absorption wire insulation for these measurements.

- [1] Specifications are for 1-hour warm-up at 6 1/2 digits.
- [2] Relative to calibration standards.
- [3] 20% overrange on all ranges, except 1000 Vdc, 3 A range.
- [4] Specifications are for 4-wire ohms function, or 2-wire ohms using Math Null. Without Math Null, add 0.2  $\Omega$  additional error in 2-wire ohms function.
- [5] For 1 k $\Omega$  unbalance in LO lead.
- [6] For power-line frequency  $\pm$  0.1%.
- [7] For power-line frequency  $\pm$  1%, use 40 dB. For  $\pm$  3%, use 30 dB.
- [8] Readings speeds for 60 Hz and (50 Hz) operation, Autozero Off.
- [9] Speeds are for 4 1/2 digits, Delay 0, Autozero OFF, and Display OFF. Includes measurement and data transfer over HP-IB.
- [10] Add 20  $\mu$ V for dc volts, 4  $\mu$ A for dc current, or 20 m $\Omega$  for resistance.

Teflon is a registered trademark of E.I. duPont deNemours and Co.

## ■ DC Characteristics

### Accuracy Specifications $\pm$ ( % of reading + % of range ) [ 1 ]

Function	Range [ 3 ]	Test Current or Burden Voltage	24 Hour [ 2 ] 23°C $\pm$ 1°C	90 Day 23°C $\pm$ 5°C	1 Year 23°C $\pm$ 5°C	Temperature Coefficient 0°C – 18°C 28°C – 55°C
<b>DC Voltage</b>	100.0000 mV		0.0030 + 0.0030	0.0040 + 0.0035	0.0050 + 0.0035	0.0005 + 0.0005
	1.000000 V		0.0020 + 0.0006	0.0030 + 0.0007	0.0040 + 0.0007	0.0005 + 0.0001
	10.00000 V		0.0015 + 0.0004	0.0020 + 0.0005	0.0035 + 0.0005	0.0005 + 0.0001
	100.0000 V		0.0020 + 0.0006	0.0035 + 0.0006	0.0045 + 0.0006	0.0005 + 0.0001
	1000.000 V		0.0020 + 0.0006	0.0035 + 0.0010	0.0045 + 0.0010	0.0005 + 0.0001
<b>Resistance</b> [ 4 ]	100.0000 $\Omega$	1 mA	0.0030 + 0.0030	0.008 + 0.004	0.010 + 0.004	0.0006 + 0.0005
	1.000000 k $\Omega$	1 mA	0.0020 + 0.0005	0.008 + 0.001	0.010 + 0.001	0.0006 + 0.0001
	10.000000 k $\Omega$	100 $\mu$ A	0.0020 + 0.0005	0.008 + 0.001	0.010 + 0.001	0.0006 + 0.0001
	100.0000 k $\Omega$	10 $\mu$ A	0.0020 + 0.0005	0.008 + 0.001	0.010 + 0.001	0.0006 + 0.0001
	1.000000 M $\Omega$	5 $\mu$ A	0.002 + 0.001	0.008 + 0.001	0.010 + 0.001	0.0010 + 0.0002
	10.00000 M $\Omega$	500 nA	0.015 + 0.001	0.020 + 0.001	0.040 + 0.001	0.0030 + 0.0004
100.0000 M $\Omega$	500 nA // 10 M $\Omega$	0.300 + 0.010	0.800 + 0.010	0.800 + 0.010	0.1500 + 0.0002	
<b>DC Current</b>	10.00000 mA	< 0.1 V	0.005 + 0.010	0.030 + 0.020	0.050 + 0.020	0.002 + 0.0020
	100.0000 mA	< 0.6 V	0.01 + 0.004	0.030 + 0.005	0.050 + 0.005	0.002 + 0.0005
	1.000000 A	< 1 V	0.05 + 0.006	0.080 + 0.010	0.100 + 0.010	0.005 + 0.0010
	3.000000 A	< 2 V	0.10 + 0.020	0.120 + 0.020	0.120 + 0.020	0.005 + 0.0020
<b>Continuity</b>	1000.0 $\Omega$	1 mA	0.002 + 0.010	0.008 + 0.020	0.010 + 0.020	0.001 + 0.002
<b>Diode Test</b>	1.0000 V	1 mA	0.002 + 0.010	0.008 + 0.020	0.010 + 0.020	0.001 + 0.002
<b>DC:DC Ratio</b>	100 mV to 1000 V		( Input Accuracy ) + ( Reference Accuracy )			
			Input Accuracy = accuracy specification for the HI-LO input signal. Reference Accuracy = accuracy specification for the HI-LO reference input signal.			

#### Transfer Accuracy (typical)

$$\frac{(\text{24 hour \% of range error})}{2}$$

#### Conditions:

- Within 10 minutes and  $\pm$  0.5°C.
- Within  $\pm$  10% of initial value.
- Following a 2-hour warm-up.
- Fixed range between 10% and 100% of full scale.
- Using 6 1/2 digit slow resolution ( 100 PLC ).
- Measurements are made using accepted metrology practices.

Chapter 1 Specifications  
AC Characteristics



**Measuring Characteristics**

<b>Measurement Noise Rejection [ 8 ]</b>	
AC CMRR	70 dB

**True RMS AC Voltage**

Measurement Method: AC-coupled True RMS – measures the ac component of the input with up to 400 Vdc of bias on any range.

Crest Factor: Maximum 5:1 at full scale

**AC Filter Bandwidth:**

Slow	3 Hz – 300 kHz
Medium	20 Hz – 300 kHz
Fast	200 Hz – 300 kHz

Input Impedance: 1 MΩ ± 2%, in parallel with 100 pF

Input Protection: 750 V rms all ranges

**True RMS AC Current**

Measurement Method: Direct coupled to the fuse and shunt. AC-coupled True RMS measurement ( measures the ac component only ).

Shunt Resistor: 0.1 Ω for 1 A and 3 A ranges

Burden Voltage: 1 A range: < 1 V rms

3 A range: < 2 V rms

Input Protection: Externally accessible 3A, 250 V fuse  
Internal 7A, 250 V fuse

**Settling Considerations**

Applying >300 V rms (or >1 A rms) will cause self-heating in signal-conditioning components. These errors are included in the instrument specifications. Internal temperature changes due to self-heating may cause additional error on lower ac voltage ranges. The additional error will be less than 0.02% of reading and will generally dissipate within a few minutes.

**Operating Characteristics [ 9 ]**

Function	Digits	Reading/s	AC Filter
ACV, ACI	6½	7 sec/reading	Slow
	6½	1	Medium
	6½	1.6 [ 10 ]	Fast
	6½	10	Fast
	6½	50 [ 11 ]	Fast

**System Speeds [ 11 ], [ 12 ]**

Function or Range Change	5/sec
Autorange Time	< 0.8 sec
ASCII readings to RS-232	50/sec
ASCII readings to HP-IB	50/sec
Max. Internal Trigger Rate	50/sec
Max. External Trigger Rate to Memory	50/sec
Max. External Trigger Rate to HP-IB or RS-232	50/sec

- [ 1 ] Specifications are for 1-hour warm-up at 6½ digits, Slow ac filter, sinewave input.
- [ 2 ] Relative to calibration standards.
- [ 3 ] 20% overrange on all ranges, except 750 Vac, 3 A range.
- [ 4 ] Specifications are for sinewave input >5% of range. For inputs from 1% to 5% of range and <50 kHz, add 0.1% of range additional error. For 50 kHz to 100 kHz, add 0.13% of range.
- [ 5 ] 750 Vac range limited to 100 kHz or 8x10<sup>7</sup> Volt-Hz.
- [ 6 ] Typically 30% of reading error at 1 MHz.
- [ 7 ] For frequencies below 100 Hz, slow AC filter specified for sinewave input only.
- [ 8 ] For 1 kΩ unbalance in LO lead.
- [ 9 ] Maximum reading rates for 0.01% of ac step additional error. Additional settling delay required when input dc level varies.
- [ 10 ] For External Trigger or remote operation using default settling delay ( Delay Auto ).
- [ 11 ] Maximum useful limit with default settling delays defeated.
- [ 12 ] Speeds are for 4½ digits, Delay 0, Display OFF, and Fast AC filter.

## ■ AC Characteristics

### Accuracy Specifications $\pm$ (% of reading + % of range) [1]

Function	Range [3]	Frequency	24 Hour [2] 23°C $\pm$ 1°C	90 Day 23°C $\pm$ 5°C	1 Year 23°C $\pm$ 5°C	Temperature Coefficient 0°C – 18°C 28°C – 55°C
True RMS AC Voltage [4]	100.0000 mV	3 Hz – 5 Hz	1.00 + 0.03	1.00 + 0.04	1.00 + 0.04	0.100 + 0.004
		5 Hz – 10 Hz	0.35 + 0.03	0.35 + 0.04	0.35 + 0.04	0.035 + 0.004
		10 Hz – 20 kHz	0.04 + 0.03	0.05 + 0.04	0.06 + 0.04	0.005 + 0.004
		20 kHz – 50 kHz	0.10 + 0.05	0.11 + 0.05	0.12 + 0.05	0.011 + 0.005
		50 kHz – 100 kHz	0.55 + 0.08	0.60 + 0.08	0.60 + 0.08	0.060 + 0.008
		100 kHz – 300 kHz [6]	4.00 + 0.50	4.00 + 0.50	4.00 + 0.50	0.20 + 0.02
	1.000000 V to 750.000 V	3 Hz – 5 Hz	1.00 + 0.02	1.00 + 0.03	1.00 + 0.03	0.100 + 0.003
		5 Hz – 10 Hz	0.35 + 0.02	0.35 + 0.03	0.35 + 0.03	0.035 + 0.003
		10 Hz – 20 kHz	0.04 + 0.02	0.05 + 0.03	0.06 + 0.03	0.005 + 0.003
		20 kHz – 50 kHz	0.10 + 0.04	0.11 + 0.05	0.12 + 0.05	0.011 + 0.005
50 kHz – 100 kHz [5] 100 kHz – 300 kHz [6]		0.55 + 0.08 4.00 + 0.50	0.60 + 0.08 4.00 + 0.50	0.60 + 0.08 4.00 + 0.50	0.060 + 0.008 0.20 + 0.02	
True RMS AC Current [4]	1.000000 A	3 Hz – 5 Hz	1.00 + 0.04	1.00 + 0.04	1.00 + 0.04	0.100 + 0.006
		5 Hz – 10 Hz	0.30 + 0.04	0.30 + 0.04	0.30 + 0.04	0.035 + 0.006
		10 Hz – 5 kHz	0.10 + 0.04	0.10 + 0.04	0.10 + 0.04	0.015 + 0.006
	3.00000 A	3 Hz – 5 Hz	1.10 + 0.06	1.10 + 0.06	1.10 + 0.06	0.100 + 0.006
		5 Hz – 10 Hz	0.35 + 0.06	0.35 + 0.06	0.35 + 0.06	0.035 + 0.006
		10 Hz – 5 kHz	0.15 + 0.06	0.15 + 0.06	0.15 + 0.06	0.015 + 0.006

#### Additional Low Frequency Errors (% of reading)

Frequency	AC Filter		
	Slow	Medium	Fast
10 Hz – 20 Hz	0	0.74	—
20 Hz – 40 Hz	0	0.22	—
40 Hz – 100 Hz	0	0.06	0.73
100 Hz – 200 Hz	0	0.01	0.22
200 Hz – 1 kHz	0	0	0.18
> 1 kHz	0	0	0

#### Additional Crest Factor Errors (non-sinewave) [7]

Crest Factor	Error (% of reading)
1 – 2	0.05%
2 – 3	0.15%
3 – 4	0.30%
4 – 5	0.40%

#### Sinewave Transfer Accuracy (typical)

Frequency	Error (% of range)
10 Hz – 50 kHz	0.002%
50 kHz – 300 kHz	0.005%

#### Conditions:

Sinewave input.  
 Within 10 minutes and  $\pm$  0.5°C.  
 Within  $\pm$  10% of initial voltage and  $\pm$  1% of initial frequency.  
 Following a 2-hour warm-up.  
 Fixed range between 10% and 100% of full scale (and < 120 V).  
 Using 6 1/2 digit resolution.  
 Measurements are made using accepted metrology practices.

**Measuring Characteristics**

<b>Frequency and Period</b>	
Measurement Method:	Reciprocal-counting technique. AC-coupled input using the ac voltage measurement function.
Voltage Ranges:	100 mV rms full scale to 750 V rms. Auto or manual ranging.
Gate Time:	10 ms, 100 ms, or 1 sec

**Settling Considerations**  
 Errors will occur when attempting to measure the frequency or period of an input following a dc offset voltage change. The input blocking RC time constant must be allowed to fully settle (up to 1 sec) before the most accurate measurements are possible.

**Measurement Considerations**  
 All frequency counters are susceptible to error when measuring low-voltage, low-frequency signals. Shielding inputs from external noise pickup is critical for minimizing measurement errors.

**Operating Characteristics [ 5 ]**

Function	Digits	Reading/s
Frequency,	6½	1
Period	5½	9.8
	4½	80

<b>System Speeds [ 5 ]</b>	
Configuration Rates	14/sec
Autorange Time	<0.6 sec
ASCII readings to RS-232	55/sec
ASCII readings to HP-IB	80/sec
Max. Internal Trigger Rate	80/sec
Max. External Trigger Rate to Memory	80/sec
Max. External Trigger Rate to HP-IB or RS-232	80/sec

- [ 1 ] Specifications are for 1-hour warm-up at 6½ digits.
- [ 2 ] Relative to calibration standards.
- [ 3 ] 20% overrange on all ranges, except 750 Vac range.
- [ 4 ] Input > 100 mV.  
For 10 mV input, multiply % of reading error x10.
- [ 5 ] Speeds are for 4½ digits. Delay 0, Display OFF, and Fast AC filter.

■ **Frequency and Period Characteristics**

**Accuracy Specifications ± ( % of reading ) [ 1 ]**

Function	Range [ 3 ]	Frequency	24 Hour [ 2 ] 23°C ± 1°C	90 Day 23°C ± 5°C	1 Year 23°C ± 5°C	Temperature Coefficient 0°C – 18°C 28°C – 55°C
<b>Frequency, Period [ 4 ]</b>	100 mV to 750 V	3 Hz – 5 Hz	0.10	0.10	0.10	0.005
		5 Hz – 10 Hz	0.05	0.05	0.05	0.005
		10 Hz – 40 Hz	0.03	0.03	0.03	0.001
		40 Hz – 300 kHz	0.006	0.01	0.01	0.001

**Additional Low-Frequency Errors ( % of reading ) [ 4 ]**

Frequency	Resolution		
	6½	5½	4½
3 Hz – 5 Hz	0	0.12	0.12
5 Hz – 10 Hz	0	0.17	0.17
10 Hz – 40 Hz	0	0.2	0.2
40 Hz – 100 Hz	0	0.06	0.21
100 Hz – 300 Hz	0	0.03	0.21
300 Hz – 1 kHz	0	0.01	0.07
> 1 kHz	0	0	0.02

**Transfer Accuracy (typical)**  
 0.0005% of reading

**Conditions:**  
 Within 10 minutes and ± 0.5°C.  
 Within ± 10% of initial value.  
 Following a 2-hour warm-up.  
 For inputs > 1 kHz and > 100 mV.  
 Using 6½ digit slow resolution ( 1 second gate time ).  
 Measurements are made using accepted metrology practices.