# 1809797

*by* Lourens Brits


| | | | |
|---|---|---|---|
| FILE | 16633_LOURENS_BRITS_1809797_169043_1071860394.PDF (5.56M) | | |
| TIME SUBMITTED | 24-MAY-2021 05:56PM (UTC+0100) | WORD COUNT | 9781 |
| SUBMISSION ID | 154642996 | CHARACTER COUNT | 61296 |

| | |
|---|---|
| Course | : HND Electrical and Electronic Engineering |
| Module | : SGEE5007 |
| Assignment | : Element 1, 2, 3 - VICS Magnitude Comparator |
| Name | : Lourens Brits |
| University number | : 1809797 |
| Date submitted | : 24/05/2012 |
| Word count | : n/a |

## Contents


Table of Figures ..... 3

Overall Introduction ..... 6

1.1 Research ..... 7

1.1.1 Digital design methods using VHDL and IP integration ..... 7

1.1.2 NI DAQmx ..... 7

1.1.3 Digital Magnitude Comparators ..... 9

1.1.4 Analogue to Digital Converters ..... 12

1.1.5 LCKD Load Cell ..... 14

1.1.6 Instrumentation and Signal Conditioning ..... 15

1.2 Design Requirements ..... 16

1.2.1 4-Bit Adder ..... 16

1.3 Implementation ..... 18

1.3.1 Half Adder ..... 18

1.3.1.1 Half Adder Testing and analysis ..... 18

1.3.2 4-Bit Adder Design ..... 20

1.3.2.1 4-Bit Adder Testing and analysis ..... 21

1.3.3 4-Bit Magnitude Comparator ..... 23

1.3.3.1 4-Bit Magnitude Comparator Testing and analysis ..... 24

1.3.4 8 Bit Magnitude Comparator ..... 25

1.4 Testing and Analysis ..... 26

1.4.1 4-Bit Magnitude Comparator ..... 26

1.4.2 8-Bit Magnitude Comparator ..... 29

1.5 Evaluation ..... 30

2.1 Element 2 Introduction ..... 31

2.2 Design Requirements ..... 31

2.2.1 The Load Cell and Signal Conditioning ..... 31

2.2.1.1 The Load Cell ..... 31

2.2.1.2 Signal Conditioning ..... 32

2.2.2 Prototypes – 10 and 12-Bit ADC ..... 34

2.2.2.1 10 Bit Analogue to Digital Converter ..... 34

2.2.2.2 12 Bit Analogue to Digital Converter ..... 35

2.2.2.3 8-Bit Analogue to Digital Converter ..... 36

2.3 Implementation ..... 38

2.3.1 8-Bit Magnitude Comparator ..... 38
2.4 Testing and Analysis ..... 41
2.4.1 Load Cell and Signal ..... 41
2.4.2 10-Bit ADC testing ..... 42
2.4.3 12-Bit ADC testing ..... 44
2.4.4 8-Bit ADC testing ..... 45
2.4.5 Magnitude Comparator Testing ..... 46
2.5 Evaluation ..... 49
3.1 Element 3 Introduction ..... 50
3.2 Design Requirements ..... 50
3.3 Implementation ..... 51
3.4 Testing and Analysis ..... 53
3.5 Conclusion and Analysis ..... 56
Overall Conclusion ..... 57
References ..... 58
4. Appendix ..... 60
Appendix - Element 2 ..... 60
Appendix Element **3** ..... 69

*A good level of detail in the table of contents*

## Table of Figures

Figure 1 - Typical DAQ system (National Instruments, 2021) ..... 7
Figure 2 - Block Diagram using DAQmx (National Instruments, 2021) ..... 8
Figure 3 - DAQmx Block Diagram Icon functions (National Instruments, 2021) ..... 8
Figure 4 - 1-Bit Digital Comparator (Aspencore, 2021) ..... 9
Figure 5 - Digital Comparator Truth Table (Aspencore, 2021) ..... 9
Figure 6 - 4-Bit Magnitude Comparator (Dialog Semiconductor, 2018) ..... 10
Figure 7 - 4-Bit Magnitude Comparator Truth Table (Dialog Semiconductor, 2018) ..... 11
Figure 8 - Two 4-Bit Comparators Cascaded (Dialog Semiconductor, 2018) ..... 11
Figure 9 - ADC quantisation of an analogue signal (Gudino, 2018) ..... 12
Figure 10 - ADC Architecture - Speed vs. Resolution (Dewesoft, 2020) ..... 12
Figure 11 - Typical SAR Block Diagram (Dewesoft, 2020) ..... 13
Figure 12 - LCKD Load Cell (OMEGA, 2021) ..... 14
Figure 13 - Wheatstone bridge (Transducer Techniques, 2020) ..... 14
Figure 14 - Half Adder Logic Diagram (Kuphaldt, 2007) ..... 16
Figure 15 - Half Adder Truth Table (Kuphaldt, 2007) ..... 16
Figure 16 - Half Adder Ladder Logic (Kuphaldt, 2007) ..... 16
Figure 17 - Full Adder (Kuphaldt, 2007) ..... 17

Figure 18 - Full Adder Truth Table (Ashenden, 2008) ..... 17
Figure 19 - 4-Bit Adder (Ashenden, 2008) ..... 17
Figure 20 - Half Adder Build ..... 18
Figure 21 - Half Adder Behavioural Simulation ..... 18
Figure 22 - Half Adder Simulation Results ..... 19
Figure 23 - 4-Bit Adder Design ..... 20
Figure 24 - 4-Bit Adder VHD Coding ..... 20
Figure 25 - 4-Bit Adder Diagram ..... 21
Figure 26 - 4-Bit Adder Test Bench Code ..... 21
Figure 27 - 4-Bit Adder Test Simulation Results ..... 22
Figure 28 - 4-Bit Magnitude Comparator ..... 23
Figure 29 - 4-Bit Magnitude Comparator Behavioural Simulation Code ..... 24
Figure 30 - 4-Bit Magnitude Comparator Simulation Results ..... 24
Figure 31 - 4-Bit Adder & 4-Bit Mag Comparator ..... 25
Figure 32 - 8-Bit Magnitude Comparator ..... 25
Figure 33 - Test 1 - 4-Bit Magnitude Comparator ..... 26
Figure 34 - Test 2 - 4-Bit Magnitude Comparator ..... 27
Figure 35 - Test 3 - 4-Bit Magnitude Comparator ..... 28
Figure 36 - Test 1 8-Bit Magnitude Comparator ..... 29
Figure 37 - Test 2 8-Bit Magnitude Comparator ..... 29
Figure 38 - Test 3 8-Bit Magnitude Comparator ..... 30
Figure 39 - Load Cell Construction ..... 32
Figure 40 - Differential Amp Isolated for Tuning ..... 33
Figure 41 - 10-Bit ADC ..... 34
Figure 42 - Formula Block and Number to Boolean Array Description ..... 34
Figure 43 - 10-Bit ADC Formula Block ..... 35
Figure 44 - 12-Bit ADC 4095 Boolean Value ..... 35
Figure 45 - 12-Bit ADC ..... 36
Figure 46 - 8-Bit ADC with Analogue I/P and Digital Output ..... 37
Figure 47 - 8-BIT ADC Circuit, HMI and Variable Power Supply ..... 37
Figure 48 - 8-Bit Magnitude Comparator Modified ..... 38
Figure 49 - Nexys4 DDR FPGA Pmod Ports ..... 38
Figure 50 - Final LabVIEW Program ..... 39
Figure 51 - NI ELVIS II Board with Nexys4 DDR Board Setup ..... 40
Figure 52 - Panel View HMI ..... 40
Figure 53 - Testing - Load Cell Circuit Linear Response Graph ..... 41
Figure 54 - Testing 10-Bit ADC - Full Scale ..... 42
Figure 55 - Testing 10-Bit ADC - 50% ..... 43
Figure 56 - Testing - 12-Bit ADC 50% ..... 44
Figure 57 - Testing - 12-Bit ADC 100% ..... 45
Figure 58 - Testing - 8-Bit ADC 2.56v & 4.90v ..... 45
Figure 59 - Testing - Magnitude Comparator - Less Than ..... 47
Figure 60 - Testing - Magnitude Comparator - Equal To ..... 47
Figure 61 - Testing - Magnitude Comparator - Greater Than ..... 48
Figure 62 - Datalogger Block Configuration ..... 50

Figure 63 - Full Program ..... 51
Figure 64 - File 1 and 2 ..... 51
Figure 65 - File 3, 4 & 5 ..... 52
Figure 66 - Sample Timer ..... 52
Figure 67 – Testing - 200g Setpoint Log Graph ..... 53
Figure 68 - Testing - 200g HMI Test ..... 53
Figure 69 – Testing - 500g Setpoint Log ..... 54
Figure 70 - Testing - 500g HMI Test ..... 54
Figure 71 - Testing - 700g Setpoint Log Graph ..... 55
Figure 72 - Testing - 700g HMI Test ..... 55

Appendix Figure 1 - Multisim Circuit of Load Cell and Amplifier ..... 60
Appendix Figure 2 - Measuring Cell Specifications ..... 60
Appendix Figure 3 - Buffer Amp Configuration ..... 61
Appendix Figure 4 - Differential Op-amp - Unit gain ..... 61
Appendix Figure 5 - Combined Load-cell and Amplifier Circuit ..... 62
Appendix Figure 6 - Standard Resistor Value Table (Kuphaldt, 2007) ..... 62
Appendix Figure 7 Load Cell Testing 5V ..... 63
Appendix Figure 8 - Load Cell Testing 4.5V ..... 64
Appendix Figure 9 - Load Cell Testing 4V ..... 64
Appendix Figure 10 - Load Cell Testing 3.5V ..... 65
Appendix Figure 11 - Load Cell Testing 3V ..... 65
Appendix Figure 12 - Load Cell Testing 2.5V ..... 66
Appendix Figure 13 - Load Cell Testing 2V ..... 66
Appendix Figure 14 - Load Cell Testing 1.5V ..... 67
Appendix Figure 15 - Load Cell Testing 1V ..... 67
Appendix Figure 16 - Load Cell Testing 0.5V ..... 68
Appendix Figure 17 - Load Cell Testing 0V ..... 68
Appendix Figure 18 - 8-Bit ADC Formula Block Equation ..... 69
Appendix Figure 19 - Data Logger File Configuration File 1 ..... 69
Appendix Figure 20 - Data Logger File Configuration File 2 ..... 70
Appendix Figure 21 - Data Logger File Configuration File 3 ..... 70
Appendix Figure 22 - Data Logger File Configuration File 4 ..... 71
Appendix Figure 23 - Data Logger File Configuration File 5 ..... 71
Appendix Figure 24 - Raw data for datalogging - Generated text files ..... 71
Appendix Figure 25 - Data Log File Sample - File 1 ..... 72
Appendix Figure 26 - Data Log File Sample - File 2 ..... 72
Appendix Figure 27 - Data Log File Sample - File 3 ..... 72
Appendix Figure 28 - Data Log File Sample - File 4 ..... 73
Appendix Figure 29 - Data Log File Sample - File 5 ..... 73
Appendix Figure 30 - Load Cell 200g Data Log ..... 74
Appendix Figure 31 - Load Cell 500g Data Log ..... 75
Appendix Figure 32 - Load Cell 700g Data Log ..... 76

*A comprehensive list of figures*

*An extensive appendix with all*

# VICS Magnitude Comparator

## Overall Introduction

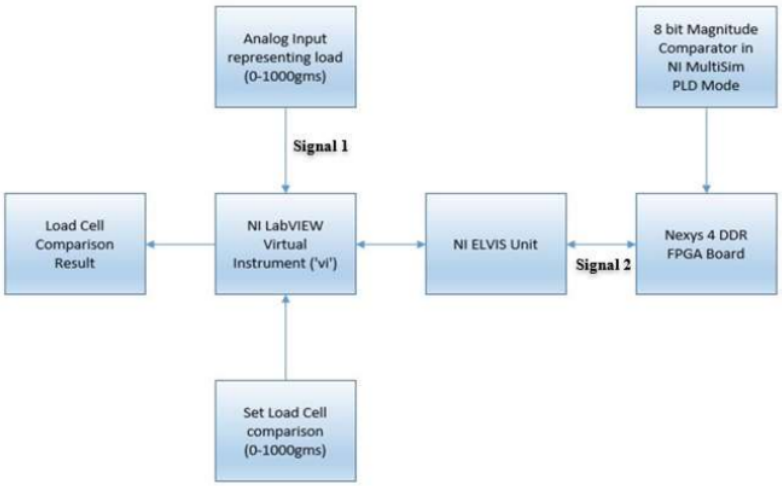
The aim of the following three elements is to design, create and test a magnitude comparator shown in the image below.

The objective will be carried out in three elements where element one will be concerned with research into the various elements of the magnitude comparator to establish a thorough understanding of each sub-system. Furthermore element 1 will cover the design, creation and testing of a combinational logic system, consisting of a 4-bit adder, magnitude comparator and 8-bit adder, using VHDL and IP integration and implementation using the Nexys 4 DDR FPGA board.

Element 2 concerns itself with the design, creation and testing of an instrument circuit consisting of a load measuring cell and instrument amplifier producing an output of signal of 0-5v, which will be converted to an 8-bit signal through the use of an analogue to digital converter. A secondary 8-bit signal will be created using LabVIEW, along with the 0-5v scaled 8-bit signal, will be compared using an 8-bit comparator using the ELVIS II development board.

*This as an interface with the FPGA*

Finally, in element 3 the load value will be converted into an actual load value, which will be fed into a data logger. The data will be extracted using .csv text files and presented on a standard excel document.




*Introduction covers most tasks.*

## 1.1 Research

The following research was carried out to enable a thorough understanding of the systems, methods, software, hardware and instrumentation that will be utilised in the practical design, construction and implementation of a combinational logic system on chip.

### 1.1.1 Digital design methods using VHDL and IP integration

VHDL (Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language), is a language used to describe the behaviour of digital circuits and is defined by IEEE standards. VHDL is utilised to design hardware, and has the ability to create test parameters to enable thorough testing and behavioural analysis of the hardware. (Intel, 2017)

VHDL is a versatile hardware programming language; due to this versatility it is the preferred language to use when programming FPGA (field programmable gate arrays), this is due to the amount of logic gates and flip-flops that can be integrated in these systems. The traditional method of drawing logic gates is not practicable; therefore, VHDL and Verilog are used. (Haskell, 2009)

Xilinx have created the first plug and play IP integration design environment in their Vivado design suite, this enables a graphical, correct-by-construction design development flow. This is in the form of a re-usable block that contains a whole system instead of a single function. By creating interconnectable sub-systems, the whole program is less complex and easily interpreted. (National Instruments, 2021)

Good explanation but need support ing image

### 1.1.2 NI DAQmx

DAQ or data acquisition is the process of measuring analogue signals in the form of voltage or current derived by a sensor or transducer, amongst many other physical properties, to ascertain pressure, temperature or flow to name a few with a computer. Figure 1 below shows a typical DAQ system, consisting of a sensor, DAQ device and computer (National Instruments, 2021).

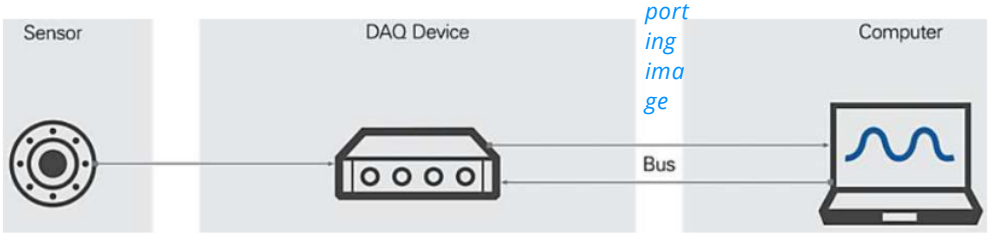



*Figure 1 - Typical DAQ system (National Instruments, 2021)*

NI DAQ was the first-generation multifunction data acquisition device driver from National Instruments. NI DAQmx is the newest version of this driver, utilising a new architecture that is

not harmonious with the old DAQ driver versions and incorporates many new functions and development tools. (National Instruments, 2021)

Figure 2 shows a typical block diagram that has been built using the DAQmx driver. This example shows how complex systems can be illustrated, simulated and diagnosed on a computer. Figure 3 gives a brief explanation of the icon functions.

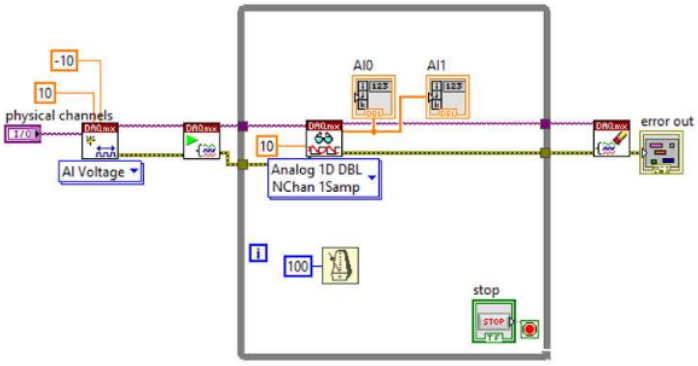



*Figure 2 - Block Diagram using DAQmx (National Instruments, 2021)*

| Labview Icon | Function |
|---|---|
| task/channels in, error in, task out, error out<br>DAQmx Start Task.vi | Sets the task for the rest of the function. |
| **DAQmx Read (Analog 1D DBL NChan 1Samp).vi**<br>task/channels in, timeout, error in, task out, data, error out<br>DAQmx Read.vi | Reads single floating point sample from each channel, from one or multiple inputs, analogue or digital. |
| **DAQmx Create Channel (AI-Voltage-Basic).vi**<br>input terminal configuration, minimum value, maximum value, task in, physical channels, name to assign, units, error in, custom scale name, task out, error out<br>DAQmx Create Channel.vi | Creates Channels to measure voltage from Elvis II DAQ board for instance |
| **DAQmx Clear Task.vi**<br>task in, error in, error out<br>DAQmx Clear Task.vi | Releases resources that are reserved in current task, then clears the task. |
| 100<br>Timer Block | Sets timing for syncrhonisation within the loop. |

*Figure 3 - DAQmx Block Diagram Icon functions (National Instruments, 2021)*

*Good explanations and example of main DAQ function blocks in LabVIEW*

### 1.1.3 Digital Magnitude Comparators

By comparing the magnitude of any two binary words, a magnitude comparator can determine whether one binary word is greater than, less than or equal to another. This is particularly useful in microcontrollers or CPU's, where binary numbers relate to process variables like temperature or flow measurement as an example.

An example of a basic 1-bit digital comparator is illustrated in figure 4, this will aid in the understanding of larger cascaded magnitude comparators.

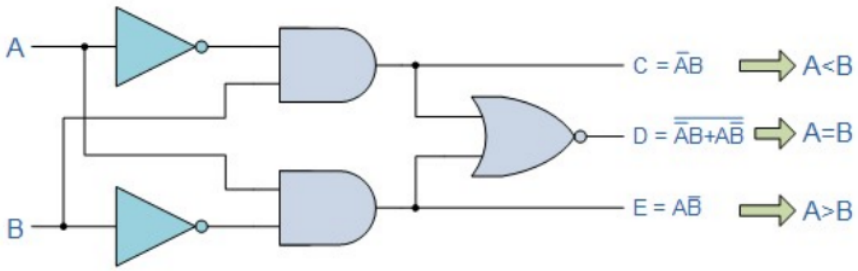



*Figure 4 - 1-Bit Digital Comparator (Aspencore, 2021)*

Through the use of combination logic, the various input states of A or B can be compared and represented on one of the three output terminals as A<B, A=B or A>B. The various states and output values are shown in figure 5 truth table. From this it is clear to see that the magnitude of each input is compared to the other, i.e. 1 or 0 (Ashenden, 2008).

| Inputs | | Outputs | | |
|---|---|---|---|---|
| B | A | A > B | A = B | A < B |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

*Figure 5 - Digital Comparator Truth Table (Aspencore, 2021)*

When B and A are both 0, or equal the A=B output is logic state 1, as for the inputs being either 1 or 0, A<B or A>B will be at state 1 respectively and in the case of both inputs being 1, A=B is true.

Figure 6 shows a 4-bit magnitude comparator logic diagram, constructed with logic gates to carry out the comparison function. There are different logic gate combinations and configurations that will achieve the same results.

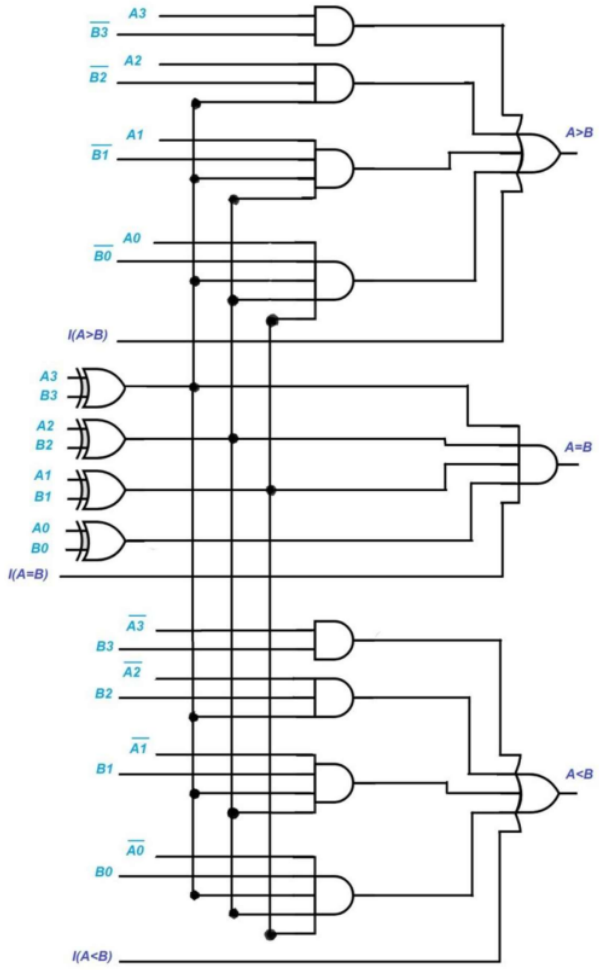



*Figure 6 - 4-Bit Magnitude Comparator (Dialog Semiconductor, 2018)*

In this example the first block is responsible for the determination of the A>B condition by using four AND gates into an OR gate, the second is for the A=B condition using XOR gates into a AND gate and the third for A<B, using AND gates into an OR.

Figure 6 shows the truth table for the 4-Bit Magnitude comparator in figure 5, where all inputs are compared and the resultant comparison output condition show.

| INPUTS | | | | | | | OUTPUTS | | |
|---|---|---|---|---|---|---|---|---|---|
| Comparing | | | | Cascading | | | | | |
| A3 B3 | A2 B2 | A1 B1 | A0 B0 | A < B | A = B | A > B | A < B | A = B | A > B |
| A3 > B3 | X | X | X | X | X | X | 0 | 0 | 1 |
| A3 = B3 | A2 > B2 | X | X | X | X | X | 0 | 0 | 1 |
| A3 = B3 | A2 = B2 | A1 > B1 | X | X | X | X | 0 | 0 | 1 |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 > B0 | X | X | X | 0 | 0 | 1 |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | 0 | 0 | 1 | 0 | 0 | 1 |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | 0 | 1 | 0 | 0 | 1 | 0 |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | 1 | 0 | 0 | 1 | 0 | 0 |
| A3 < B3 | X | X | X | X | X | X | 1 | 0 | 0 |
| A3 = B3 | A2 < B2 | X | X | X | X | X | 1 | 0 | 0 |
| A3 = B3 | A2 = B2 | A1 < B1 | X | X | X | X | 1 | 0 | 0 |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 < B0 | X | X | X | 1 | 0 | 0 |

*Figure 7 - 4-Bit Magnitude Comparator Truth Table (Dialog Semiconductor, 2018)*

Magnitude comparators can be cascaded together to enable larger than 4-bit binary numbers to be compared, an example of two 4-Bit Magnitude comparators is illustrated in figure 8. The most significant bits will initially be compared after which a sequence will be followed which is set in the controlling device.

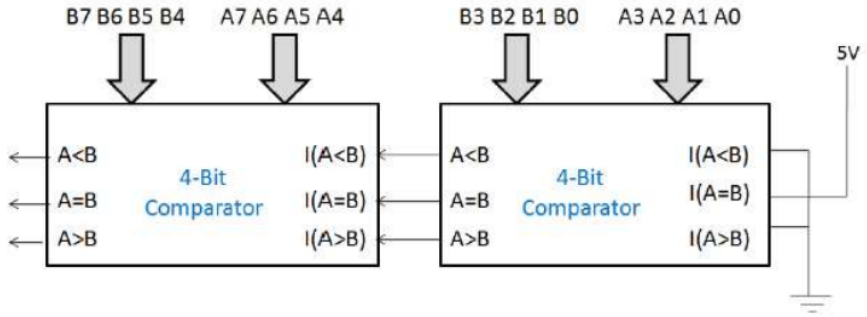



*Figure 8 - Two 4-Bit Comparators Cascaded (Dialog Semiconductor, 2018)*

*Good explanations and examples of the comparator*

### 1.1.4 Analogue to Digital Converters

Analogue to digital converters or ADC's enable varying amplitude sinewave signals, or analogue signals to be converted into a binary value of 1 or 0, this is achieved through sampling the analogue signal at defined intervals. The sampling intervals are known as the sampling rate. ADC's have various resolutions and are governed by the Nyquist criterion, which states that the minimum sampling frequency has to be twice that of the highest sampled analogue signal and is expressed as $f_N \geq 2f_m$, where $f_m$ is the input signal frequency (Gudino, 2018).

Figure 9 shows an analogue signal, in blue, with a square wave in red following the waveform. The square wave is the ADC sampling and quantising the analogue signal, where quantisation is the approximation of the sampled waveform into a binary value at each step.

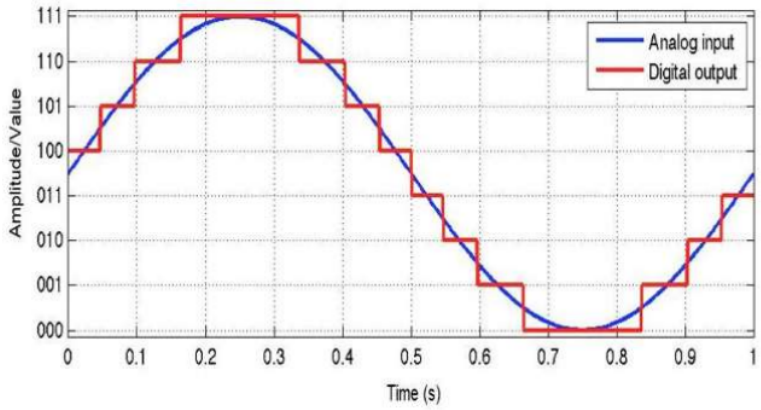



*Figure 9 - ADC quantisation of an analogue signal (Gudino, 2018)*

*Good explanation with a supporting diagram for quantisation. Good use of comparison table here.*

Figure 10 shows the various architectures utilised to perform analogue to digital conversion. It is apparent that the flash architecture is the fastest due to the lower resolution at the cost of high power consumption and highest cost. SAR exhibits the best balance between resolution, speed and cost (Dewesoft, 2020).

| ADC Type | Pros | Cons | Max Resolution | Max Sample Rate | Main Applications |
|---|---|---|---|---|---|
| **Successive Approximation (SAR)** | Good speed/resolution ratio | No inherent anti-aliasing protection | 18 bits | 10 MHz | Data Acquisition |
| **Delta-sigma (ΔΣ)** | High dynamic performance, inherent anti-aliasing protection | Hysteresis on unnatural signals | 32 bits | 1 MHz | Data Acquisition, Noise & Vibration, Audio |
| **Dual Slope** | Accurate, inexpensive | Low speed | 20 bits | 100 Hz | Voltmeters |
| **Pipelined** | Very fast | Limited resolution | 16 bits | 1 GHz | Oscilloscopes |
| **Flash** | Fastest | Low bit resolution | 12 bits | 10 GHz | Oscilloscopes |

*Figure 10 - ADC Architecture - Speed vs. Resolution (Dewesoft, 2020)*

Successive approximation evidently exhibits the best all round performance with excellent balance between speed and resolution and can convert a wide range of signals with very good fidelity. SAR designs are cheap, stable and reliable and can be configured to operate in a low end card that shares inputs as well as in multiplexed ADC devices that utilises individual ADC's for each channel. The drawback of SAR's are the lack of anti-aliasing filters, but this can be overcome by adding AAF to the input of the ADC by the DAQ system. (Dewesoft, 2020).

Figure 11 illustrates a typical SAR block diagram and functions as stated by Dewesoft; "An on-board DAC creates an analog reference voltage equal to the digital code output of the sample and holds a circuit. Both of these are fed into a comparator which sends the result of the comparison to the SAR. This process continues for "n" successive times, with "n" being the bit resolution of the ADC itself, until the closest value to the actual signal is found" (Dewesoft, 2020).

*Research demonstrates a sound knowledge with ADC's*

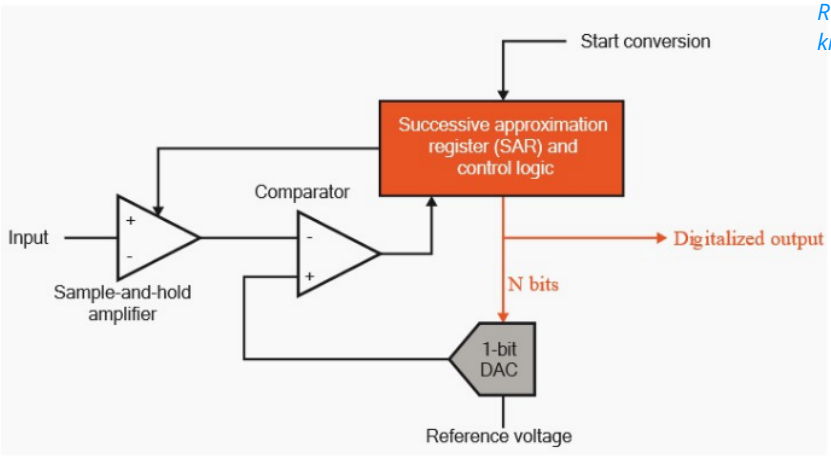



*Figure 11 - Typical SAR Block Diagram (Dewesoft, 2020)*

### 1.1.5 LCKD Load Cell

A load cell (Figure 11) is a device or transducer that measures force that is exerted onto it and produces a voltage, which is relative to the amount of force exerted. The principle of operation is that there are four strain gauges in the form of a Wheatstone bridge (Figure 13) which balances two legs of a bridge circuit. When the force is applied to the strain gauge the deformation is measured and produces an electrical signal or voltage. (HBM, 2020)

Figure 12 shows the LCKD series sub miniature load cell, with a range from 1kg to 1000 lbs, which is constructed from stainless steel and features temperature compensation, whilst operating on a 5v DC supply. (OMEGA, 2021)



*Figure 12 - LCKD Load Cell (OMEGA, 2021)*

*Explanation relevant in this portfolio. A good understanding demonstrated here.*

The Wheatstone bridge acts as a two parallel voltage divider circuits, where in the case of figure 13, R1 and R2 are a pair and R3 and R4 are a pair. The increase in forge applied to the strain gauge, causes resistance variations in the Wheatstone bridge between the two pairs of voltage dividers, this produces the measured output relative to amount of force applied. (Transducer Techniques, 2020)

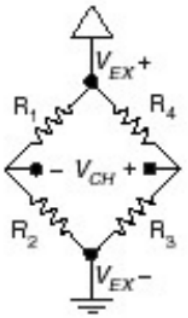



*Figure 13 - Wheatstone bridge (Transducer Techniques, 2020)*

### 1.1.6 Instrumentation and Signal Conditioning

Signal conditioning usually consists of amplification, electrical isolation, linearization, cold junction compensation and excitation. Signal conditioning is utilised to change a type of electronic signal into another form, to enable it to be read by conventional instrumentation (Omega, 2021)

Table 1 show an example of the functions that are carried out by a load cell signal conditioner.

| Load Cell Signal Conditioner Functions | |
|---|---|
| Technique | Purpose |
| Filtering | Used to reduce noise on load cell signal |
| Signal Conversion | Analogue to digtal conversion takes place to enable modern data evaluation |
| Amplifying | Either increase resolution or signal to noise ratio to achieve amplification |
| Linearisation | Detects unusual voltage from the cell output and structures the signal accordingly |
| Excitation | Similarly to RTD's, strain gauges need an external source to operate |

*Table 1 - Load Cell Signal Conditioning (Omega, 2021)*

*Stages involved in the signal conditioning process are identified*

## 1.2 Design Requirements

### 1.2.1 4-Bit Adder

A 4-Bit adder was created by firstly creating a half adder and then combining the two half-adders to create a full adder after which four full adders were combined. The combinational logic was designed and implemented using Xilinx Vivado software and the VHDL programming language.

Figure 14 shows the logic diagram for a half adder, utilising an XOR and AND gate in a parallel configuration. In this case ∑ is the low order output or SUM and $C_{out}$ is the carry out. A half adder is very useful for adding one binary digit (Kuphaldt, 2007).

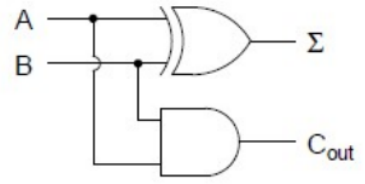



*Figure 14 - Half Adder Logic Diagram (Kuphaldt, 2007)*

Figure 15 illustrates the truth table for the half adder, with the four input variations and output results

| A | B | Σ | $C_{out}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

*Figure 15 - Half Adder Truth Table (Kuphaldt, 2007)*

A ladder logic representation can be seen in figure 16.

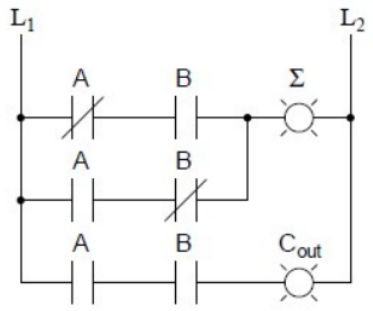



*Figure 16 - Half Adder Ladder Logic (Kuphaldt, 2007)*

To construct a full adder, two half adders can be cascaded together, as shown in figure 17. Here the second half adder has the carry bit into it as well as ∑ from the first half adder, this enables the addition of more than one binary bit.

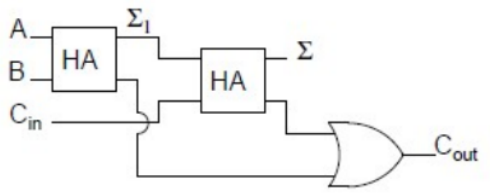


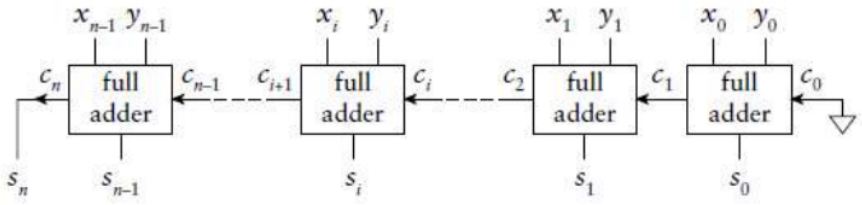

*Figure 17 - Full Adder (Kuphaldt, 2007)*

The full adder truth table can be seen in in figure 18.

| $x_i$ | $y_i$ | $c_i$ | $s_i$ | $c_i + 1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

*Figure 18 - Full Adder Truth Table (Ashenden, 2008)*

By cascading four full adders a 4-BIT adder can be created as shown in figure 19.




*4 bit adder components have been identified here. .*

*Figure 19 - 4-Bit Adder (Ashenden, 2008)*

## 1.3 Implementation

### 1.3.1 Half Adder

Figure 20 shows the behavioural architecture build for the half adder.

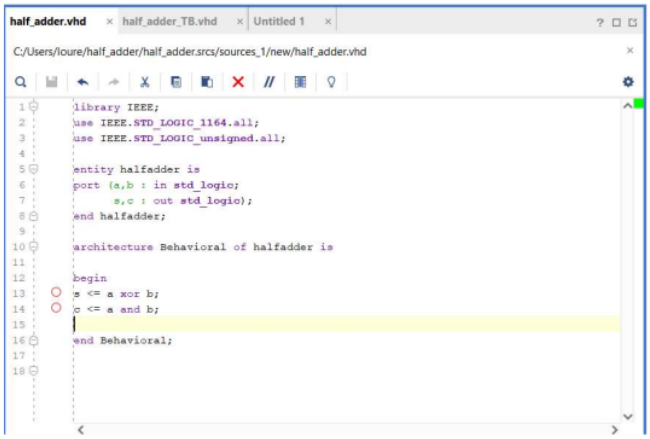



*Figure 20 - Half Adder Build*

#### 1.3.1.1 Half Adder Testing and analysis

Once the half adder was built, a behavioural simulation was carried out as shown on figure 21.

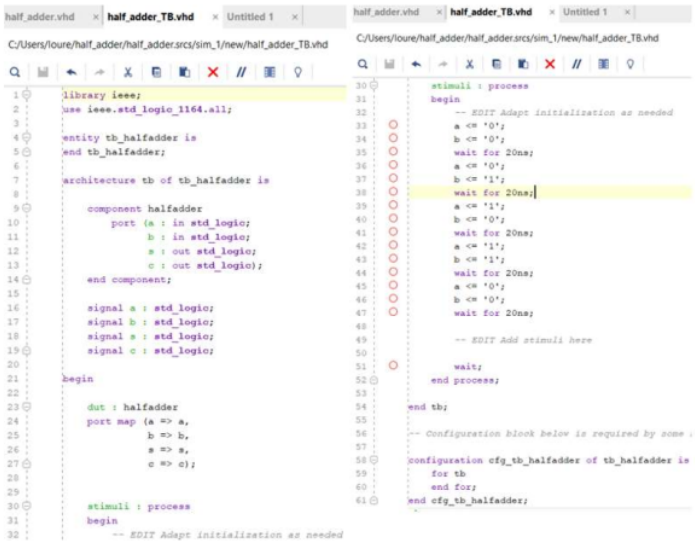



*Figure 21 - Half Adder Behavioural Simulation*

The behavioural simulation results are shown in figure 22.




*Figure 22 - Half Adder Simulation Results*

The same tests were carried out for the full adder and found to behave as desired, therefore the 4-bit adder was built by combining four full adders that consisted of two half adders each, figure 22 shows the 4-Bit adder construction in the Vivado VHDL software.

*A good assumption made here as the full adder will be tested in the 4 bit adder.*

### 1.3.2 4-Bit Adder Design

The four bit adder shows a more complex design, with the half adder inheritance apparent. By combining the four individual full adders, a 4-Bit adder is constructed.

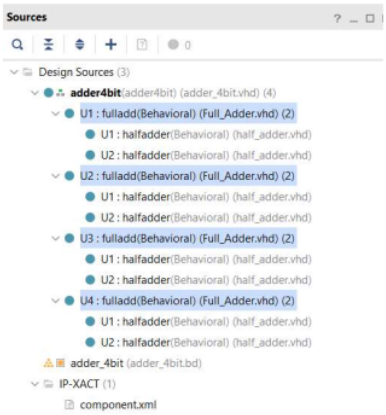
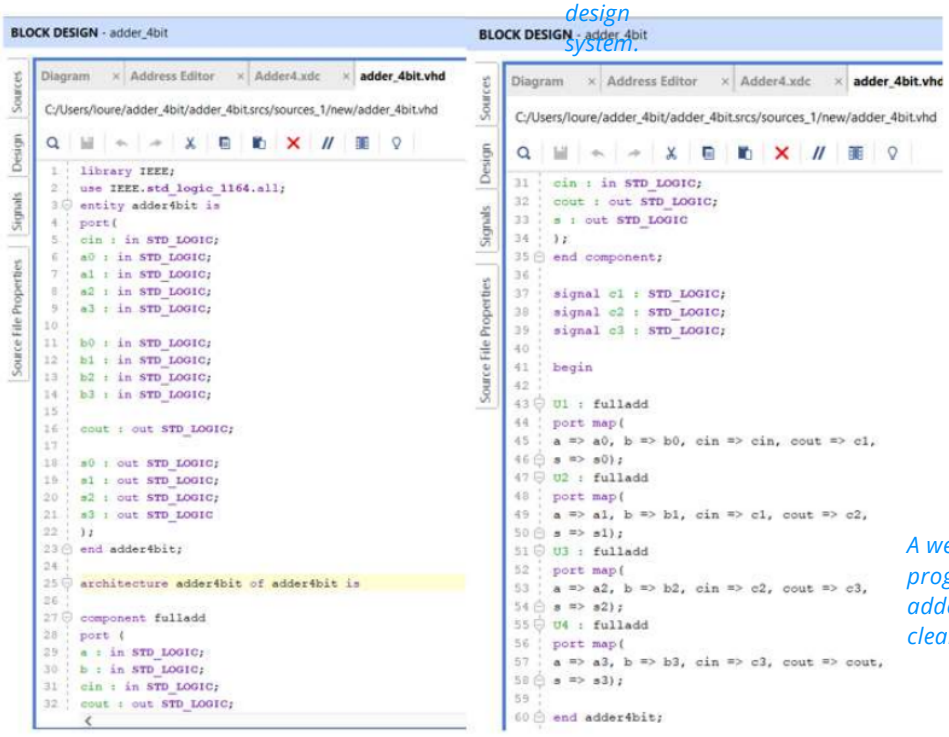



*Figure 23 - 4-Bit Adder Design*

*Hierarchy and inheritance has been identified which is an essential part of understanding for this design system.*

Figure 24 shows the logic layout of the 4-Bit adder, defining the hardware descriptions and ports as well as the port maps, with all operations held in the constraint files.

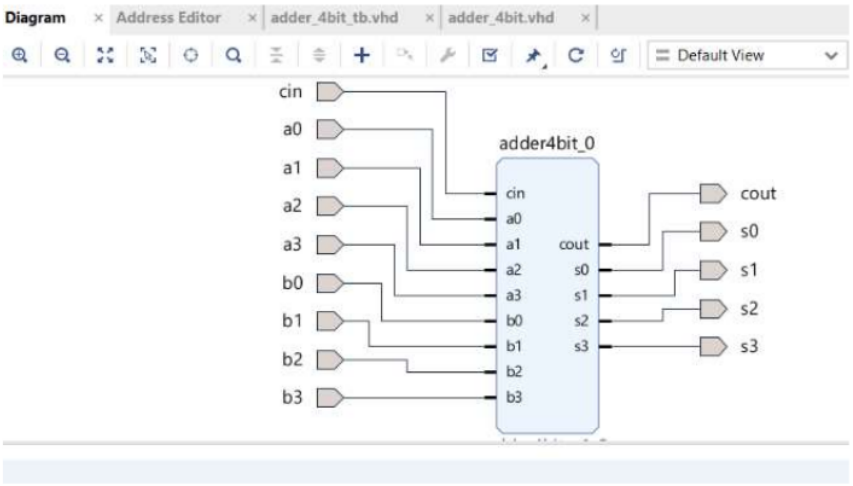



*A well-structured program with the full adder components clearly shown*

*Figure 24 - 4-Bit Adder VHD Coding*

### 1.3.2.1 4-Bit Adder Testing and analysis

The 4-Bit adder diagram is shown in figure 25, showing outputs and inputs matching the coding.




*IP design is correct*

*Figure 25 - 4-Bit Adder Diagram*

Figure 26 shows the test bench code that will be used for the simulation and testing. The expected result is to add two 4-bit binary numbers.

```
64      -- Put initialisation code here
65
66      a0 <= '1';
67      a1 <= '0';
68      a2 <= '0';
69      a3 <= '0';
70      b0 <= '1';
71      b1 <= '0';
72      b2 <= '0';
73      b3 <= '0';
74      cin <= '0';
75      wait for 10 ns;
76      a0 <= '1';
77      a1 <= '1';
78      a2 <= '1';
79      a3 <= '0';
80      b0 <= '0';
81      b1 <= '0';
82      b2 <= '1';
83      b3 <= '0';
84      cin <= '0';
85      wait for 10 ns;
```

*Good set of test data*

*Figure 26 - 4-Bit Adder Test Bench Code*

In figure 27 the simulation results are shown, it is clear to see that the code is functioning as required, with the binary addition being successful.

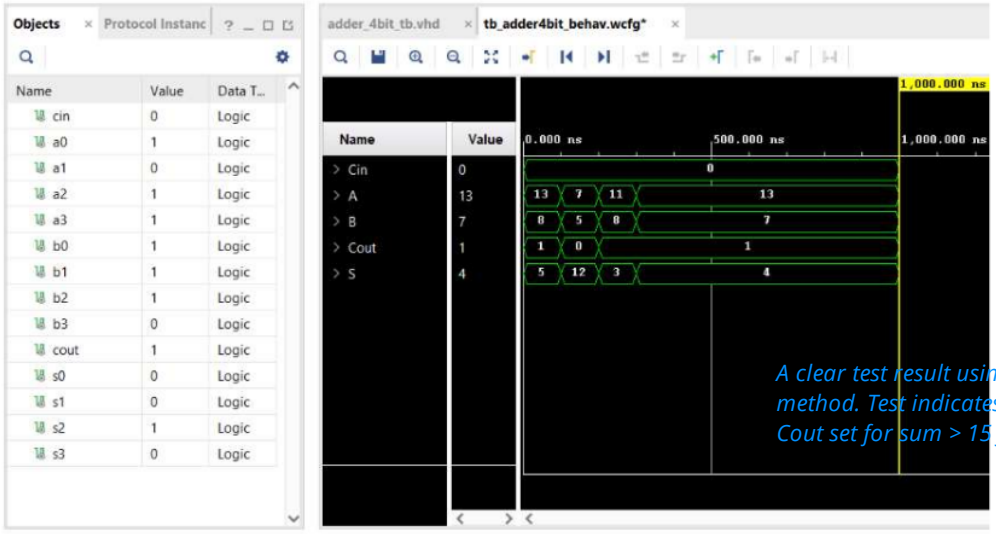



*Figure 27 - 4-Bit Adder Test Simulation Results*

### 1.3.3 4-Bit Magnitude Comparator

A 4-Bit Magnitude comparator was created, as shown in figure 28. The output from the 4-Bit adder is connected to the comparator with outputs OAGTB and OALTB set low and OAEQB set high. With this configuration the comparator will determine if the A is greater than B, A is less than B or if A and B are equal. To achieve the combination of the two systems were IP integrated using the IP wrapper function.

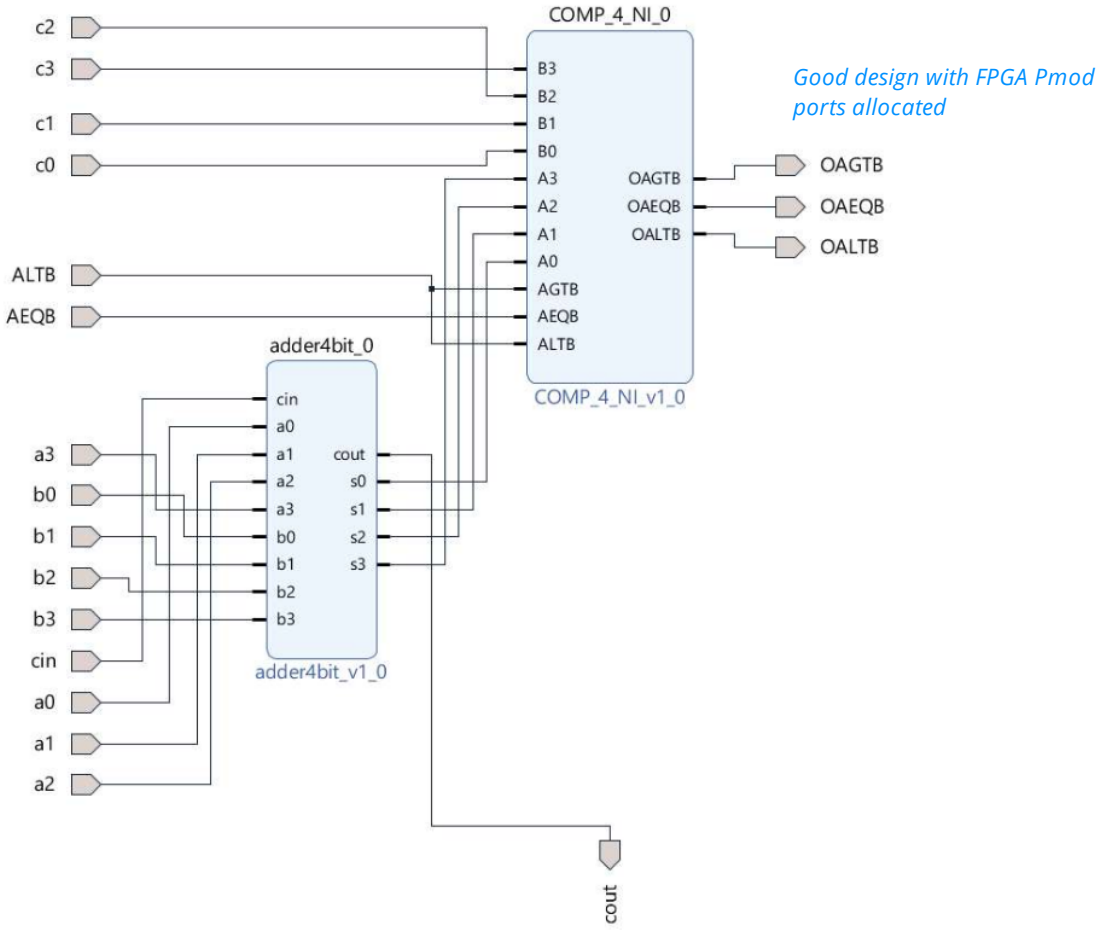



*Figure 28 - 4-Bit Magnitude Comparator*

### 1.3.3.1 4-Bit Magnitude Comparator Testing and analysis.

Using the same methods as before, the 4-Bit magnitude Comparator test bench was constructed to enable analysis of the system.

```
    dut : COMP_4_NI
    port map (B3    => B3,
              B2    => B2,
              B1    => B1,
              B0    => B0,
              A3    => A3,
              A2    => A2,
              A1    => A1,
              A0    => A0,
              AGTB  => AGTB,
              AEQB  => AEQB,
              ALTB  => ALTB,
              OAGTB => OAGTB,
              OAEQB => OAEQB,
              OALTB => OALTB);

    stimuli : process
    begin
        -- EDIT Adapt initialization as needed
        B3 <= '1';
        B2 <= '1';
        B1 <= '0';
        B0 <= '0';
        A3 <= '1';
        A2 <= '0';
        A1 <= '0';
        A0 <= '0';
        AGTB <= '0';
        AEQB <= '1';
        ALTB <= '0';
        Wait for 100ns;
        B3 <= '1';
        B2 <= '0';
        B1 <= '0';
        B0 <= '0';
        A3 <= '1';
        A2 <= '0';
        A1 <= '0';
        A0 <= '0';
        AGTB <= '0';
        AEQB <= '1';
        ALTB <= '0';
        Wait for 100ns;
        B3 <= '1';
        B2 <= '1';
        B1 <= '1';
        B0 <= '0';
        A3 <= '1';
        A2 <= '1';
        A1 <= '1';
        A0 <= '0';
        AGTB <= '0';
        AEQB <= '1';
        ALTB <= '0';
        Wait for 100ns;

        -- EDIT Add stimuli here

        wait;
    end process;
```

*The 4 bit comparator VHDL was generated from the PLD design. This must be stated.*

*Figure 29 - 4-Bit Magnitude Comparator Behavioural Simulation Code*

The simulation results in figure 29 shows the greater than, less than and equal to sequence, indicating the comparator is functioning correctly.

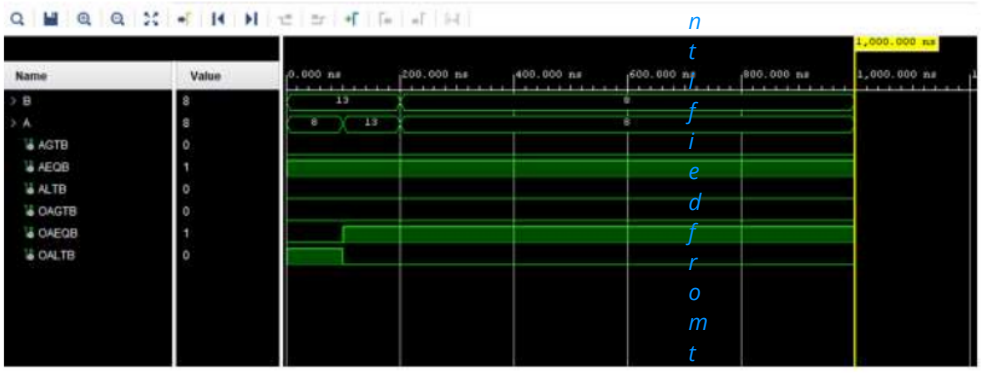



*Figure 30 - 4-Bit Magnitude Comparator Simulation Results*

*Test indicates correct operation of comparator*

### 1.3.4 8 Bit Magnitude Comparator

To enable the creation and testing of the 8-Bit magnitude comparator Multisim is used.

Figure 34 shows the two 4-Bit magnitude comparators cascaded in Vivado, before being converted to PLD and uploaded to Multisim

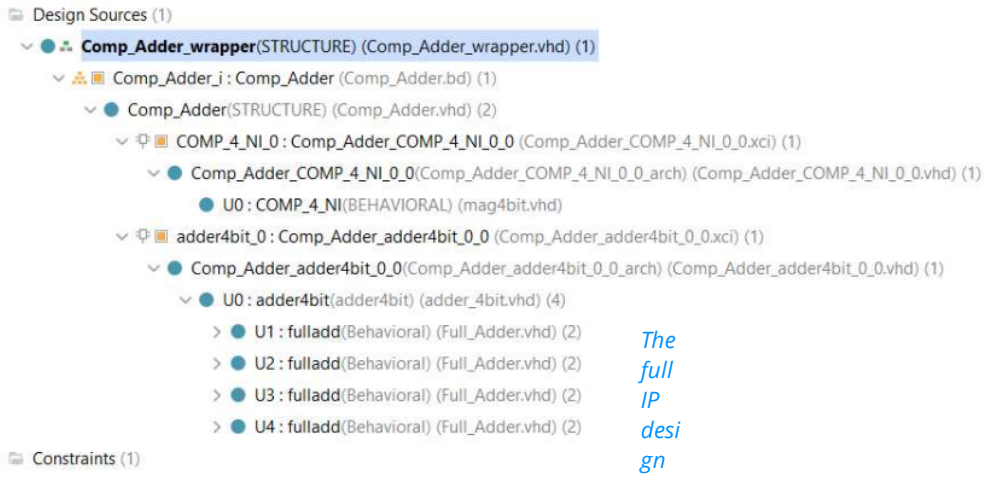
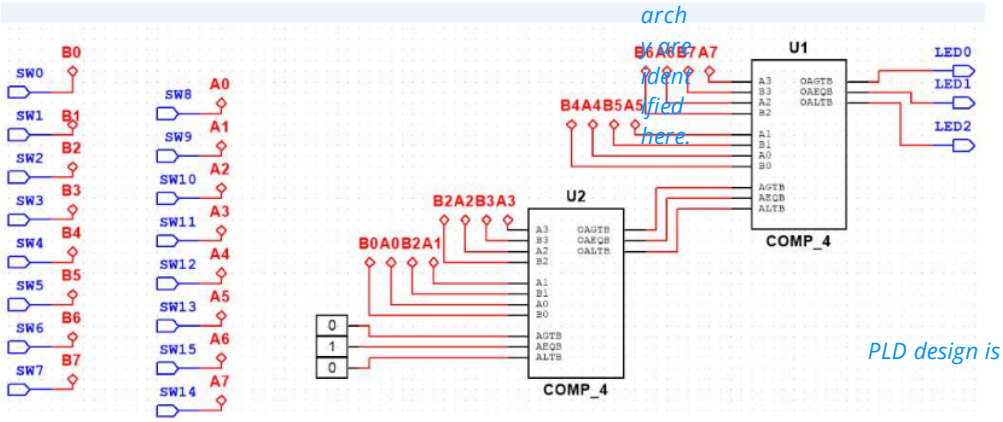



*Figure 31 - 4-Bit Adder & 4-Bit Mag Comparator*

*The full IP design components hierarchy are identified here.*

Figure 32 shows the cascaded comparators uploaded to Multisim.



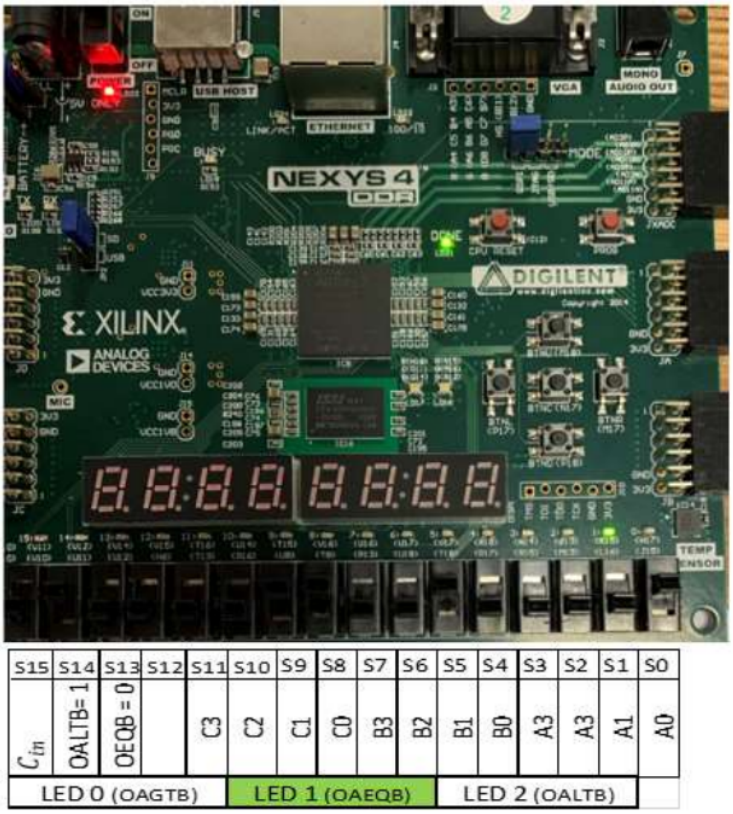

*PLD design is correct*

*Figure 32 - 8-Bit Magnitude Comparator*

## 1.4 Testing and Analysis

### 1.4.1 4-Bit Magnitude Comparator

The 4-Bit Magnitude comparator was converted to PLD and uploaded to the Nexus 4DDR test board. The switches marked S15 – S0 are labelled and the resultant LED tag is illuminated below the switch allocations. By selecting the switch high or low position, the intended binary sum can be achieved. The results are shown below on the results table. The switches in figure 33 are arrange to produce binary number 0011 (3) for the set point, 0001 for A & 0010 for B, resulting in the comparator adding A & B values to give 0011 (3), therefore energising LED 1 – OAEQB ( A equal to B), as expected.




| S15 | S14 | S13 | S12 | S11 | S10 | S9 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{in}$ | OALTB= 1 | OEQB = 0 | | C3 | C2 | C1 | C0 | B3 | B2 | B1 | B0 | A3 | A3 | A1 | A0 |
| LED 0 (OAGTB) | | | | | LED 1 (OAEQB) | | | | | LED 2 (OALTB) | | | | | |

*Test indicates the sum value is equal to the set value*

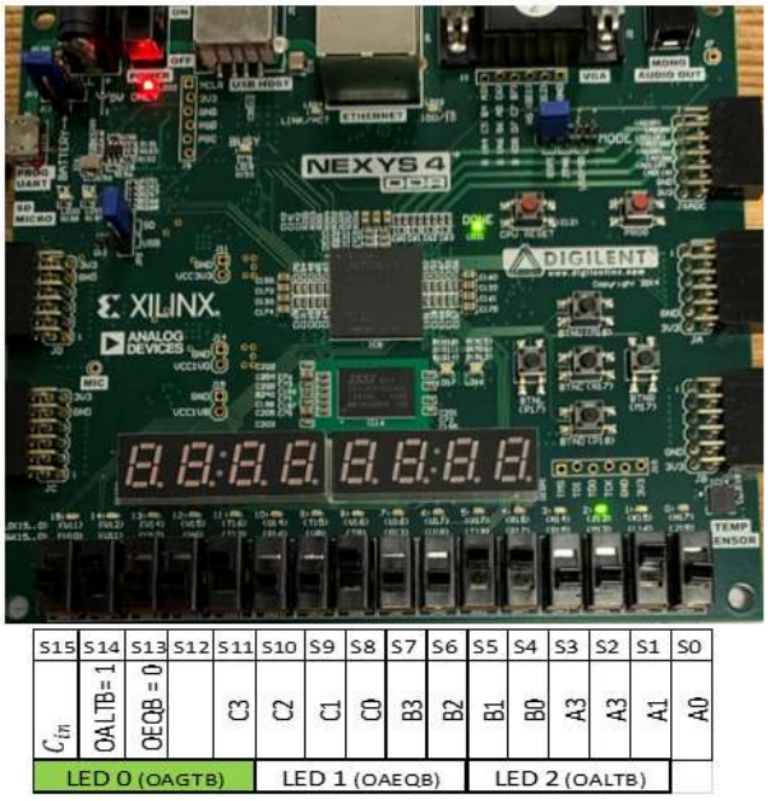
*Figure 33 - Test 1 - 4-Bit Magnitude Comparator*

Practical test 1 results.

| | Set Value | SUM A | SUM B | Total | LED |
|---|---|---|---|---|---|
| TEST 1 | 0011 (3) | 1 | 2 | 3 | OAEQB |

*Table 2 - Test 1 results 4-Bit Magnitude Comparator*

In test 2 (figure 34) the C switches are arranged in a manner that produces binary number 0100 (4), with A = 0100 and B = 0100, as expected the comparator energises LED 0 (OAGTB), as the combined value for A & B is 8, which is greater than the set point of 4. Results are shown in table 3.




| S15 | S14 | S13 | S12 | S11 | S10 | S9 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{in}$ | OALTB= 1 | OEQB = 0 | | C3 | C2 | C1 | C0 | B3 | B2 | B1 | B0 | A3 | A3 | A1 | A0 |
| LED 0 (OAGTB) | | | | | LED 1 (OAEQB) | | | | | LED 2 (OALTB) | | | | | |

*Test indicates the sum value is greater than set value.*

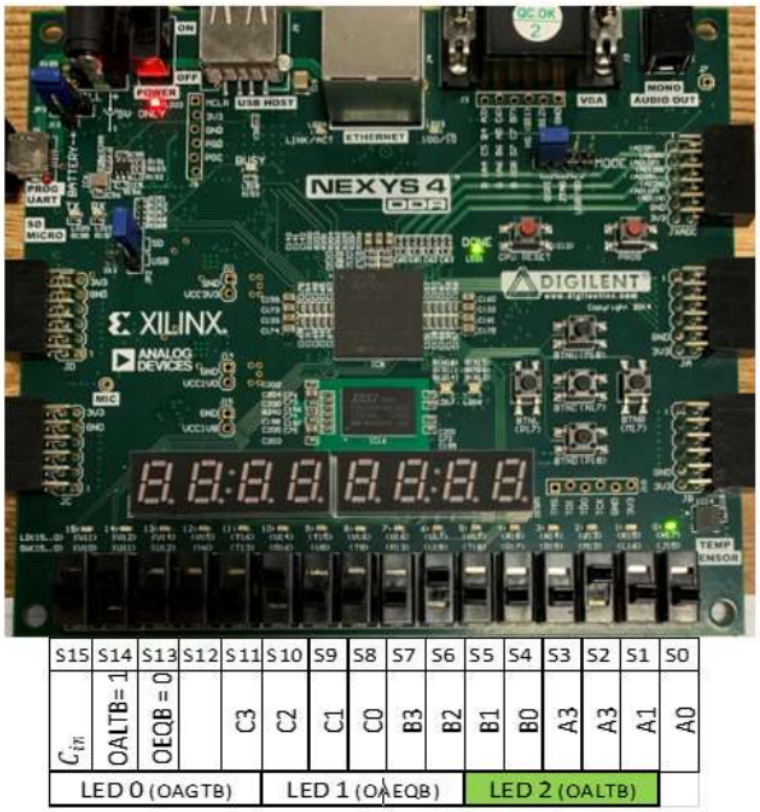
*Figure 34 - Test 2 - 4-Bit Magnitude Comparator*

Practical test 2 results.

| | Set Value | SUM A | SUM B | Total | LED |
|---|---|---|---|---|---|
| TEST 1 | 0011 (3) | 1 | 2 | 3 | OAEQB |
| TEST 2 | 0100 (4) | 4 | 4 | 8 | OAGTB |

*Table 3 - Test 2 Results - 4-Bit Magnitude Comparator*

Test 3 (figure 35) results confirmed that the 4-Bit Magnitude Comparator is functioning correctly with the set point set to 1001 (9) and the combined value of A and B being 5. LED 2 (OALTB) energised as expected, as 5 is less than 9 as shown in table 4.




*Test indicates the sum value is less than set value.*

*Figure 35 - Test 3 - 4-Bit Magnitude Comparator*

Practical test 3 results.

| | Set Value | SUM A | SUM B | Total | LED |
|---|---|---|---|---|---|
| TEST 1 | 0011 (3) | 1 | 2 | 3 | OAEQB |
| TEST 2 | 0100 (4) | 4 | 4 | 8 | OAGTB |
| TEST 3 | 1001 (9) | 2 | 3 | 5 | OALTB |

*Good use of tables in this section*

*Table 4 - Test 3 Results - 4-Bit Magnitude Comparator*

The three tests conducted confirms that the 4-Bit magnitude comparator is functioning as intended.

### 1.4.2 8-Bit Magnitude Comparator

Similar tests will be carried out on the 8-Bit magnitude comparator. Test 1 was conducted with switches S15 - S8 is designated to input B and S7 – S0 to input A. Both switches A and B are configured to produce a binary value of 00000111 (7), this results in LED 1 (OAEQB) energising, indicating that the values are equal. This confirms the correct operation of the system.

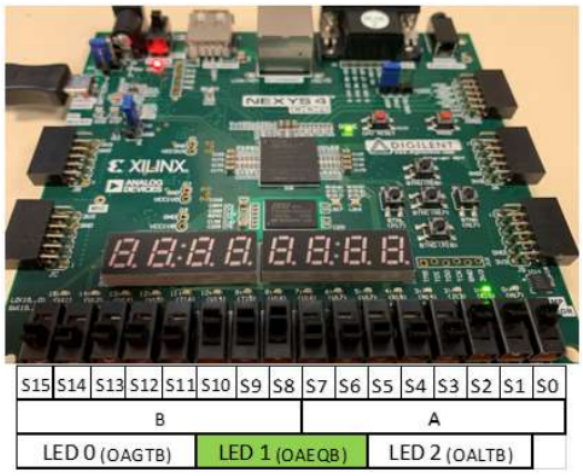



*Figure 36 - Test 1 8-Bit Magnitude Comparator*

*Test indicates A = B*

In test 3, input A is configured to give the binary value 128 and B to give 49, this resulted in LED 2 (OAGTB) to energise, which is correct as A is larger than B, therefore this test confirms correct operation.

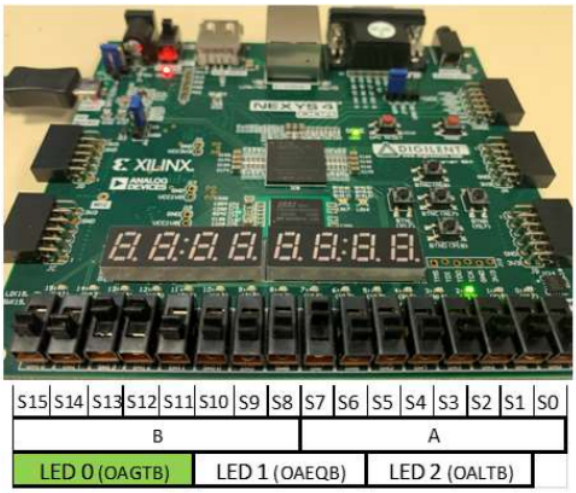



*Figure 37 - Test 2 8-Bit Magnitude Comparator*

*Test indicates A > B.*

Test 3 resulted in LED 2 (OALTB) energising as expected, as input A is set to binary value 5 and B to value 9. This test concludes that the system is functioning as expected.

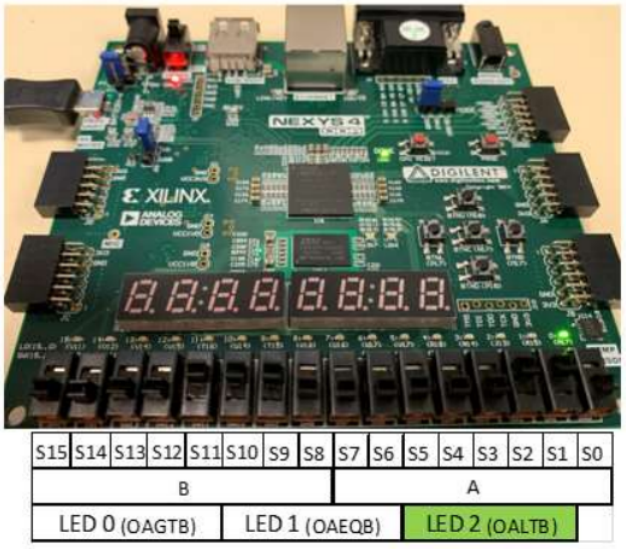



*Figure 38 - Test 3 8-Bit Magnitude Comparator*

*Test indicating A < B.*

## 1.5 Evaluation

The research conducted enabled a thorough understanding of the underlying principles associated with the design, build and testing of the systems. The use of VDHL language and IP integration proved to be a building block of the adder, 4 and 8-bit magnitude comparator. IP integration enabled the 4-bit magnitude comparator to be constructed by combining a 4-bit adder and comparator.

Once the 4-bit magnitude comparator was designed, built and tested, an 8bit magnitude comparator could be constructed. This highlights the versatility of this system, where cascading systems together, through the use of IP integration, larger bit capacities can be achieved. The hierarchy and inherent building block design ensures these systems are robust, as long as the initial design stages are executed, simulated and tested using Vivado.

The testing of the systems concluded that they all functioned correctly, therefore the overall design requirements were met and the project was an overall success and that the magnitude comparators are now ready to be integrated into an ADC.

*Good evaluation. The role of the FPGA could have been included.*

# Element 2

## 2.1 Element 2 Introduction

In this element of the portfolio, an instrument measuring system will be designed, built and tested using Multisim, with the aim of the circuit to weigh 1000 grams of weight over a 0-5v range.

This will be achieved by converting an analogue signal into 8-bit digital signals which will be compared to digital setpoint signal to ascertain if they are equal to set value, greater than set value or less than set value. The comparison signal will originate from a load cell, as covered in element one research and be conditioned and scaled to be 0-5v analogue signal.

This instrument circuit amplifier will be constructed using OP37AJ operational amplifiers (virtual) using the National Instruments' Multisim software for simulation and testing. The load cell is constructed using four fixed value resistors and four variable resistors, which simulate the individual load cells, excited by a 5-volt supply and arranged in a manner that will act as a Wheatstone bridge, as described in the research part of element 1.

The potential difference between the load cell output terminals V1 and V2 are fed into the comparator arranged op-amps in the instrument amplifier circuit where a scaled 0-5-volt output is produced.

The produced analogue signal will be converted to 8-Bit binary, after which it will be fed to the comparator.

*Introduction covers most reqirements*

## 2.2 Design Requirements

The circuit will be required to function form a 5-volt supply and be capable of measuring 1000g with the load cell output scaled to 2mV per volt, which will result in a full load output of 10mV, derived by the following equation; $5Vx\,{}^{2mV}/_{V} = 10mV$, after which the scaled value needs to be conditioned to enable a 0-5V final output.

The complete circuit can be seen in appendix figure 1 in which both elements are clearly distinguishable with the load cell on the left and the amplifier for signal conditioning on the right.

### 2.2.1 The Load Cell and Signal Conditioning

#### 2.2.1.1 The Load Cell

The load cell is shown in figure 39, in which the arrangement of resistors clearly indicates a Wheatstone-Bridge configuration, with fixed resistors R11, R13, R14 and R15, all at a set value of 349.25Ω and the four variable resistors notated as R8, R9, R10 and R12.

The Wheatstone-Bridge operates through the variation of diagonal pairs of variable resistors, i.e., pair 1 - R8 and R10 and pair 2 R9 and R12, with the differences in resistance pairs resulting in a differential voltage created on output pins V1 and V2 which is fed to the signal conditioning element of the circuit. The resting state resistance of the variable resistors is 350 Ω.

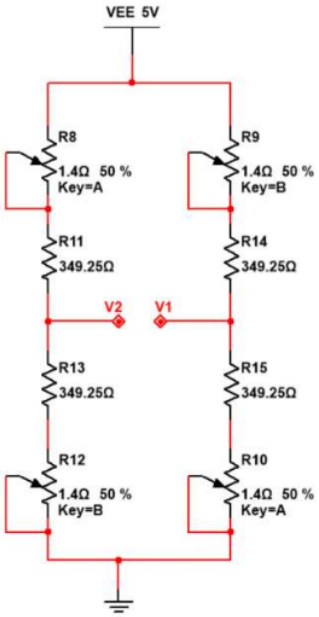



*Figure 39 - Load Cell Construction*

### 2.2.1.2 Signal Conditioning

The signal conditioning element will be responsible for receiving the load cell differential voltage from outputs V1 and V2, which will require a buffer amp or voltage follower with a suitable gain to increase the voltage before being fed to the differential op amp, which will have a suitable gain as well. The combination of these three op amps and associated circuit components will create the instrument amplifier, which will satisfy the project objectives.

Appendix figure 3 shows the buffer amplifier configuration which will be elaborated upon. As previously discussed, these two op amps will apply the most gain to the load cell output signal. The gain set through the use of the correct resistors labelled as R5, R6 and R7 and is calculated using the following equation; $A = 1 + \frac{R_5+R_6}{R_7}$. It is clear that through the use of this resistor configuration unity gain cannot be achieved, R5 and R6 will normally have the same resistance value with the loadcell differential voltage output apparent across R7. This configuration will ensure that both operational amplifier gain settings are the same and therefore will not change the relationship of voltage produced across V1 and V2.

Appendix figure 4 shows the differential op-amp with unity gain which will be elaborated upon. The voltages fed into R1 and R2 are still identical to the load cell differential voltages produces,

*This should now be high gain*

with gain applied by the buffer amp. The values for R1, R2, R3 and R4 are all the same at 1KΩ, therefore it is clear that this amp will provide unity gain. The gain calculation is as follows; $A = -\frac{R3}{R1}$, as $R_1 = R_2$ and $R_3 = R_4$.

The output voltage can now be determined through the use of the following formula;

$$V_{ou} = (V_2 - V_1)x\frac{R3}{R1}$$

The combination of these two circuits will create the instrument amplifier, as shown in appendix figure 5. The combined gain calculation to achieve the required 0-5V output from the maximum 10mV output from the load cell requires a circuit gain of 500, which is derived through $\frac{5V}{10x10^{-3}V} = 500$. This gain will we achieved through ensuring the buffer amp has a gain of 400 and the differential amp a gain of 100.

*Required gain is idnetified*

*This should be buffer unity gain of 5 and opamp gain of 100*

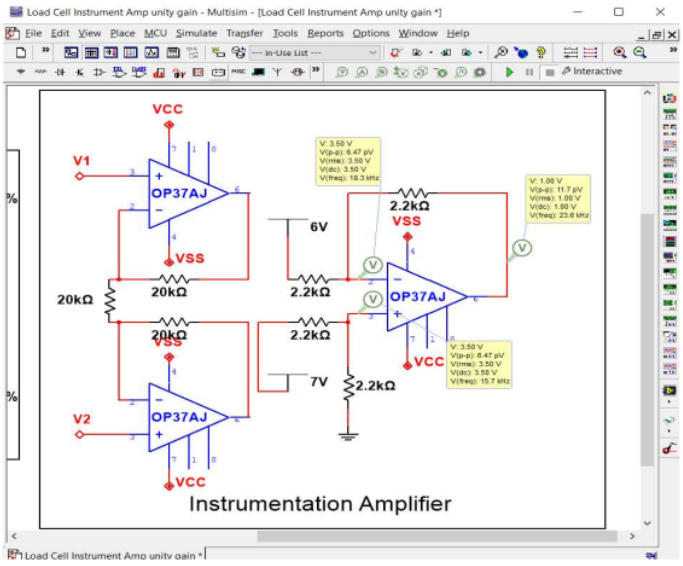
To achieve the required gain from the amplifiers some tuning is required and carried out by varying the resistor pair values accordingly. This has been carried out and the correct resistors have been selected using standard value resistors, of which the chart can be seen in appendix figure 6.

Figure 40 shows the differential amplifier isolated from the amplifier circuit; this isolation enables the amp performance to be analysed to ensure correct operation.

The inputs are loaded with 6 volts on the negative terminal and 7 volts on the positive terminal, which results in 1 volt output with unity resistance. The differential amp (OP37AJ) can be seen to have equal input and feedback resistances, therefore applying unity gain.

*A good explanation*

Gain calculation $V_{out} = \frac{R_{feedback}}{R_{in}} = \frac{R_f}{R_{in}} = \frac{2.2K}{2.2K} = 1$




*Principle of unity gain is realised*

*Figure 40 - Differential Amp Isolated for Tuning*

Testing will be carried out to ensure the circuit is functioning correctly. Test results will be illustrated in section 2.4 and evaluations carried out accordingly.

### 2.2.2 Prototypes – 10 and 12-Bit ADC

The following prototypes will evaluate 10 and 12-bit ADC's utilising the 0-5v DC signal created by the instrument amplifier, disregarding the frequencies of the instrument output signal. The operating principles have been covered in element 1 and will therefore not be covered again.

#### 2.2.2.1 10 Bit Analogue to Digital Converter

The ADC is created using LabVIEW software from National Instruments and is shown as well as the HMI panel in figure 41 below and consists of an analogue input, scaled representation of the input, formula block and 10-Bit digital output as well as a number to Boolean array shown in figure 42.

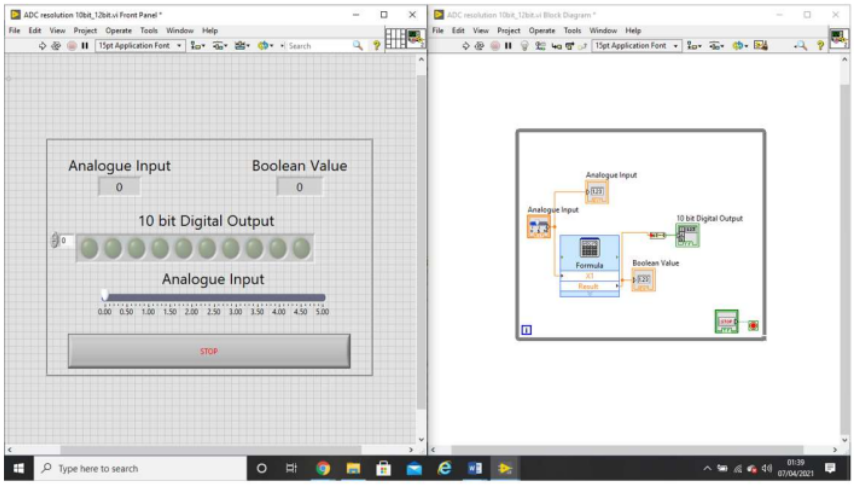
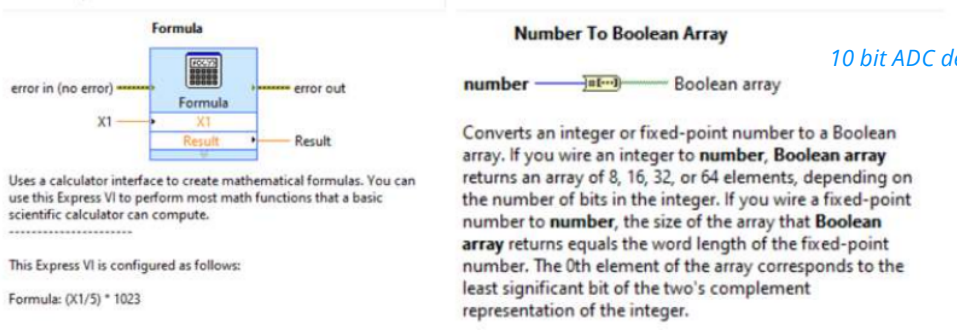



*Figure 41 - 10-Bit ADC*

The equation configuration command, figure 42, enables the ADC formula to be entered for 10-Bit resolution and takes the form of; $\frac{X_1}{V_{ref.max}}(2^n)$ , where $X_1$ = formula block analogue input, $V_{ref.max}$ = ADC maximum scale and the resolution in bits as n.




*10 bit ADC design is correct*

*Figure 42 - Formula Block and Number to Boolean Array Description*

Referring to figure 43, X is the analogue input signal, 5v is the maximum scale and 1023 is the ADC resolution as 10-Bit.

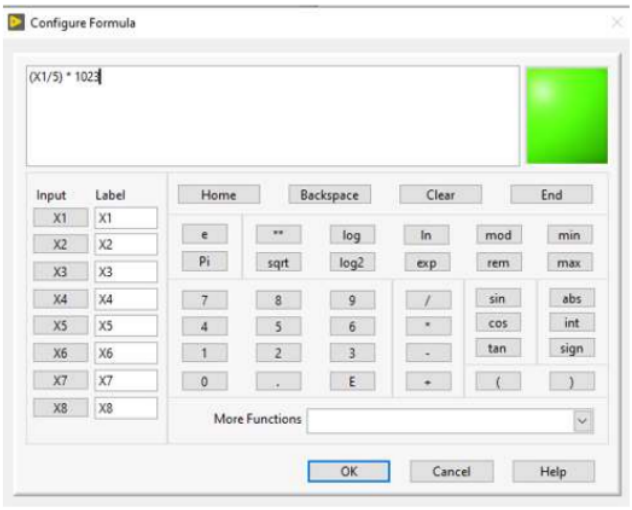
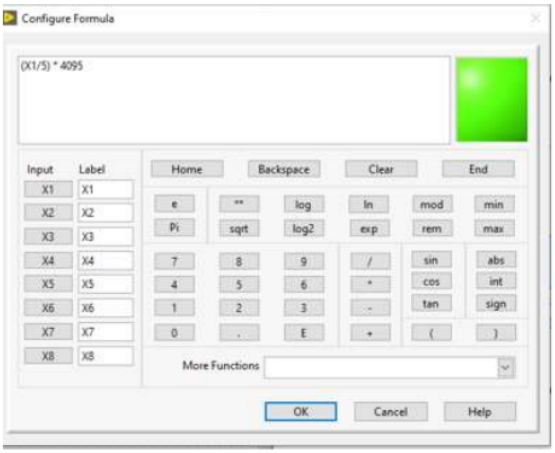



*Figure 43 - 10-Bit ADC Formula Block*

Testing of the 10-Bit ADC is conducted in section 2.4.2, which confirms correct operation.

### 2.2.2.2 12 Bit Analogue to Digital Converter

The 10-Bit ADC has now been changed to 12-Bit Resolution but manipulating the formula block once again, as shown in figure 44.




*12 bit ADC set in formula block*

*Figure 44 - 12-Bit ADC 4095 Boolean Value*

The formula now shows 4095 resolution, with the same maximum scale 0-5v analogue input.

Figure 45 shows the HMI amended to show the extra 2 lights in the array, 12 lights represent 12-Bits, the circuit construction stays the same.

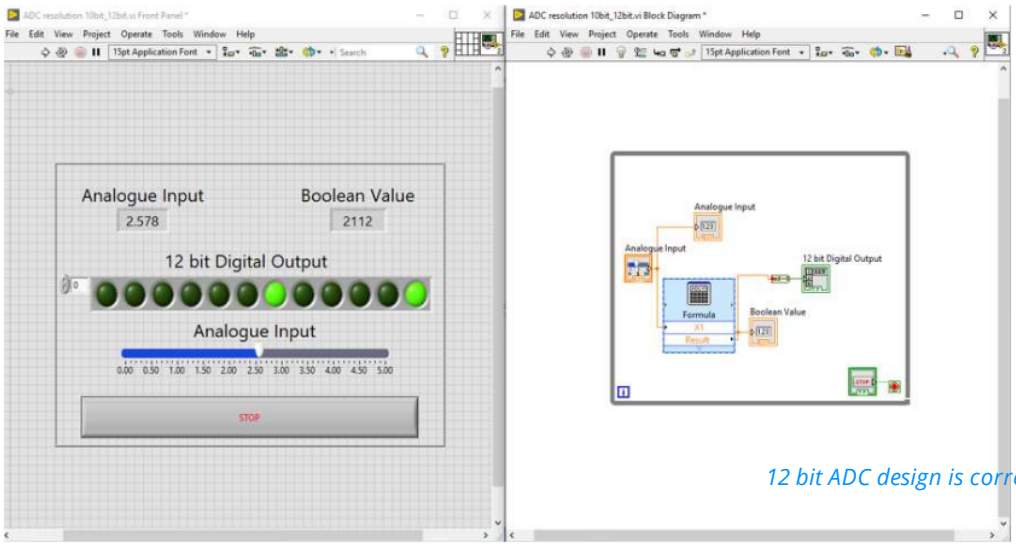



*Figure 45 - 12-Bit ADC*

Tests were carried out to confirm correct operation and can be seen in section 2.4.3, which confirms the 12-Bit ADC is functioning correctly.

### 2.2.2.3 8-Bit Analogue to Digital Converter

In line with the objectives, an 8-Bit ADC with a range of 0-5V will be utilised, which will enable the seamless connectivity of the loadcell and instrument amplifier as well as the 8-Bit magnitude comparator.

Building upon the knowledge gained so far on ADC's, the ADC circuit used in section 2.2.2.1 and 2.2.2.3 will be altered by adding an analogue circuit as shown in figure 46. A 0-5V analogue input is shown in to top left corner with the voltage limits clearly visible. The data read block reads the voltage which is fed t the formula block. Once the formula has been executed, the result is rounded and fed to the DO port, ready to be connected to the 8-Bit magnitude comparator.

The addition of a variable power supply on the analogue input enables testing of the formula block which is configured with the equation $\frac{X1}{5}(256)$, as shown in appendix figure 18.

*8 bit ADC design with formula block is correct. All the io DAQ function blocks are configured*

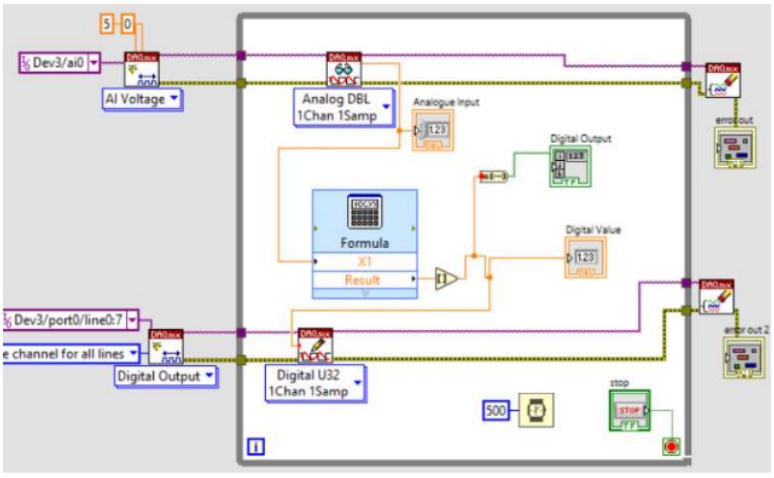
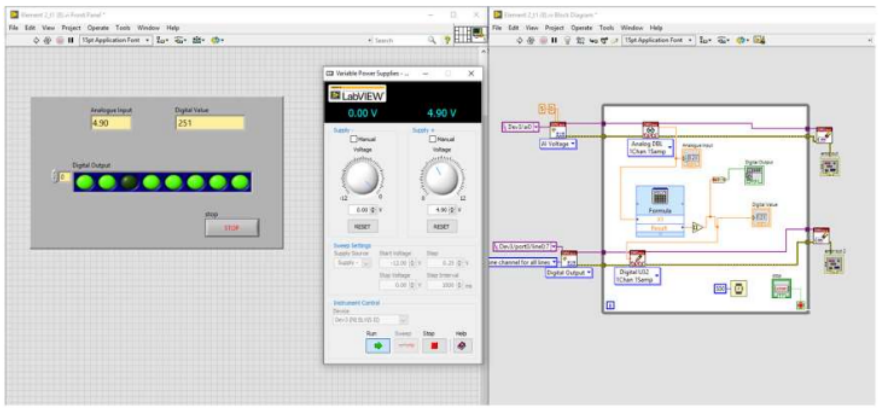



*Figure 46 - 8-Bit ADC with Analogue I/P and Digital Output*

Figure 47 shows the 8-Bit ADC and associated circuitry, the variable power supply as well as the HMI front panel which will be used to confirm correct operation.

The variable power supply is incorporated in the LabVIEW software and will act as a virtual input and simulate the loadcell and instrument amplifier output.




*Test for 8 bit ADC is good which proves*

*program configuration*

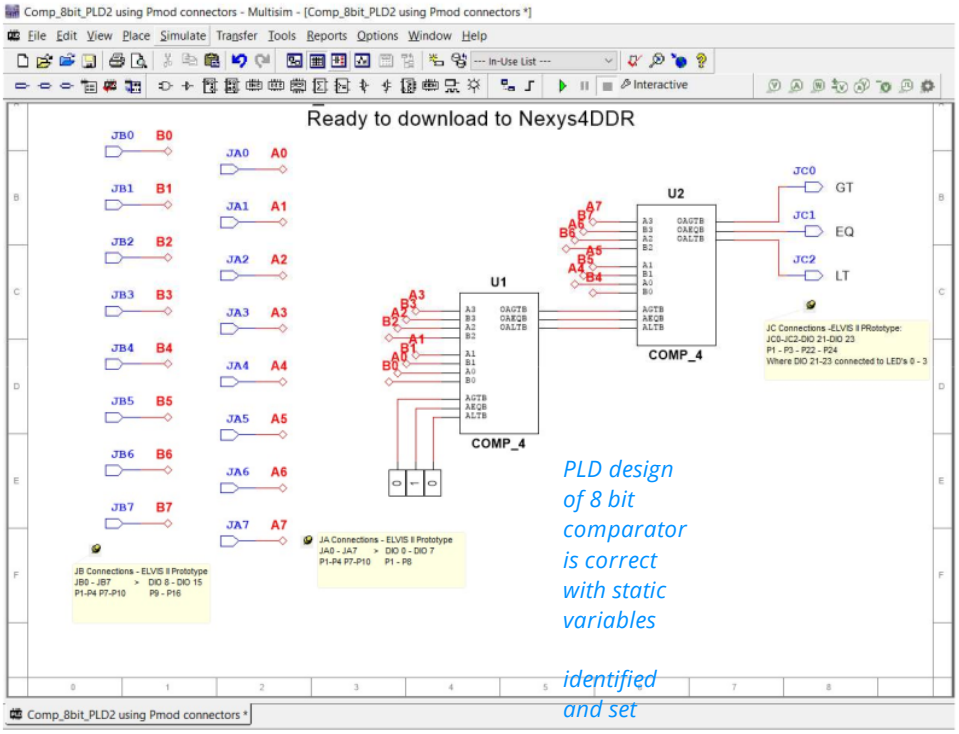
*Figure 47 - 8-BIT ADC Circuit, HMI and Variable Power Supply*

Testing of the 8-Bit ADC was carried out in section 2.4.4, where a conclusion can be made that it functions as intended.

## 2.3 Implementation

### 2.3.1 8-Bit Magnitude Comparator

Further development carried out on the 8-Bit magnitude comparator created in element one enables the comparator to be connected to the inputs and outputs of the Nexys4 DDR FPGA board, the modifications can be seen in figure 48. The changes consist of data set B connected to the JB connector, data set A is connected to JA connector and the LED output signals connected to JC connector. All changes are made on Multisim.




*PLD design of 8 bit comparator is correct with static variables*

*identified and set*

*Figure 48 - 8-Bit Magnitude Comparator Modified*

Figure 49 shows the Nexus4 DDR PMOD ports front view.

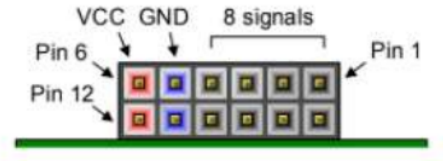



*Figure 49 - Nexys4 DDR FPGA Pmod Ports*

*FPGA Pmod ports are identified*

Figure 50 shows the complete program used in LabVIEW, the addition of three more formula blocks and a digital input apparent. The DO block needed to be amended to utilise 16 outputs instead of 8 as before, this will enable two sets of 8-Bit data to be sent to the Nexys4 DDR for comparison.

The DI block added enables the three Nexys4 DDR board inputs, greater than, equal to and less than to be received through lines 21-23, port 0.

Finally each formula block has its own formula which will enable the conversions and scaling to take place.

*Explanation gives an overview but the operation is not explained*

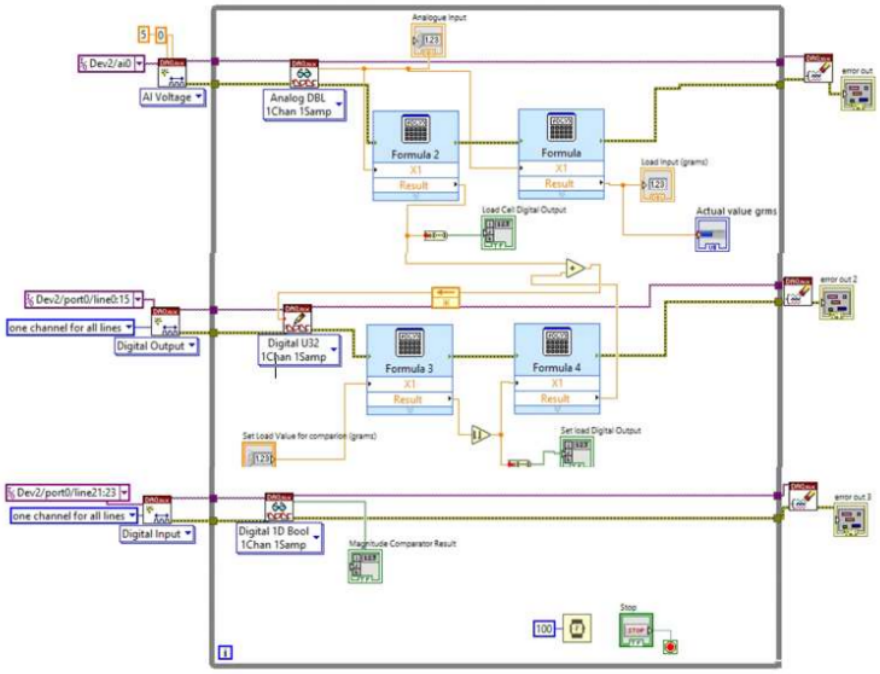



*Figure 50 - Final LabVIEW Program*

Table 5 illustrates the equation held in each formula block ad what function it serves.

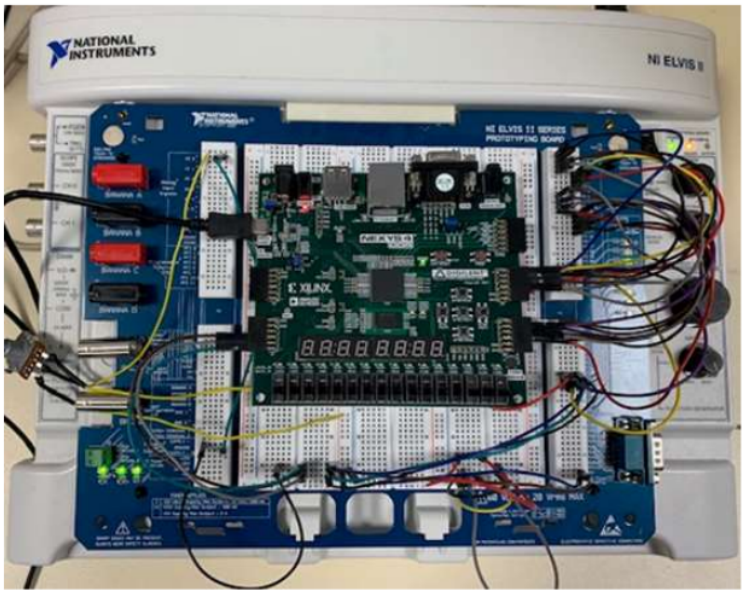
| | Formula Block Name | | | |
|---|---|---|---|---|
| | Formula | Formula 2 | Formula 3 | Formula 4 |
| Equation | $\frac{X_1}{5}(1000)$ | $\frac{X_1}{5}(256)$ | $\frac{X_1}{1000}(256)$ | $X_1(256)$ |
| Function | Voltage input conversion | Analogue voltage coversion | Reference setpoint conversion | Boolean value conversion |

*Good use of table.*

*Table 5 - Formula Block Functions*

*Formula block 4 needs to be explained in more detail a this*

Figure 51 shows the ELVIS II board and Nexys4 DDR board assembled ready to test the system. The boards are connected with short jumper wired connectors, Micro USB for Nexys4 DDR power supply and the potentiometer used to change the magnitude comparator state.




*Good image of equipment setup*

*Figure 51 - NI ELVIS II Board with Nexys4 DDR Board Setup*

Figure 52 shows the panel view HMI created using LabView. The HMI has been designed to function in an ergonomic manner, which shows the set load value, analogue input, load input, set load DO, load cell DO, actual value and result derived from the magnitude comparator.




*Front panel design is good*

*Figure 52 - Panel View HMI*

The complete system has been tested with results and test procedure published and analysed in section 2.4.5.

## 2.4 Testing and Analysis

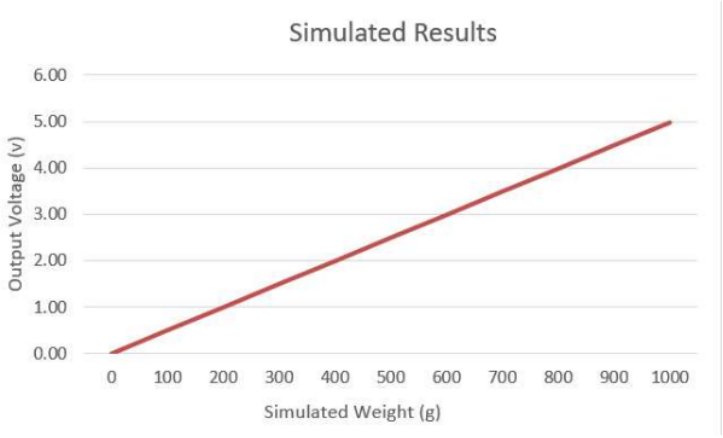
### 2.4.1 Load Cell and Signal

Test results for the combined circuit is shown in figure 53 with the table of results illustrated in Table 6.

The test is carried out by varying the resistors in opposing pairs, i.e., top left and bottom right increases whist top right and bottom left decreases, which will simulate an applied weight and therefore produce a differential voltage on outputs V1 & 2. When all resistances are equal distributed at 50%, the load cell simulated weight will be equal to zero and with 100% / 0% distribution the weight will be 1000g.

The response is linear and matches the expected results very closely, where each 5% increase in resistance of opposing variable resistors result in a simulated increase of 100 grams of weight on the cell, which produced an increase of 0.5v output.

The small discrepancies in expected and simulated result highlighted in table 6 are apparent throughout the measurements which is shown to be 0.01V initially. The real error voltage only becomes apparent once the 0g weight is simulated, the resultant voltage is -0.00587v, which leads to a conclusion that the circuit error is -0.00587v throughout the measurement range and therefore can be negated, or compensated for. The lack of resolution of the higher voltage outputs, results in the error to seem higher than it actually is.




*Good resultant linear graph which shows that the error is acceptable*

*Figure 53 - Testing - Load Cell Circuit Linear Response Graph*

It is clear to see that the tests results confirm the correct operation of the circuit, with an increase of 100g resulting in an increase of 0.5v, which satisfies the objectives of this section of element 2. Further evaluation and analysis carried out confirm that the resistor values derived were correct, resulting in an appropriate amount of gain being applied at the correct stages of the circuit.

| % Resistance Applied (Ω) | Expected Result | | Simulated Result | | Error (v) |
|---|---|---|---|---|---|
| | Weight (g) | Output Voltage (v) | Weight (g) | Output Voltage (v) | |
| 100 | 1000 | 5.0 | 1000 | 4.99 | -0.01 |
| 95 | 900 | 4.5 | 900 | 4.49 | -0.01 |
| 90 | 800 | 4.0 | 800 | 3.99 | -0.01 |
| 85 | 700 | 3.5 | 700 | 3.49 | -0.01 |
| 80 | 600 | 3.0 | 600 | 2.99 | -0.01 |
| 75 | 500 | 2.5 | 500 | 2.49 | -0.01 |
| 70 | 400 | 2.0 | 400 | 1.99 | -0.01 |
| 65 | 300 | 1.5 | 300 | 1.49 | -0.01 |
| 60 | 200 | 1.0 | 200 | 0.994 | -0.006 |
| 55 | 100 | 0.5 | 100 | 0.494 | -0.006 |
| 50 | 0 | 0 | 0 | -0.00587 | -0.00587 |

*Table 6 – Testing - Load cell Circuit Test Results*

*Good table of results*

### 2.4.2 10-Bit ADC testing

Figure 54 shows the first test conducted on the 10-Bit ADC. Here the range programmed, in line with the objective, is 0-5, with the maximum Boolean value attainable with 10-Bit resolution being 1023. From the results show a conclusion can be made that the 10-BIT ADC is functioning correctly at full scale or 5v analogue input.

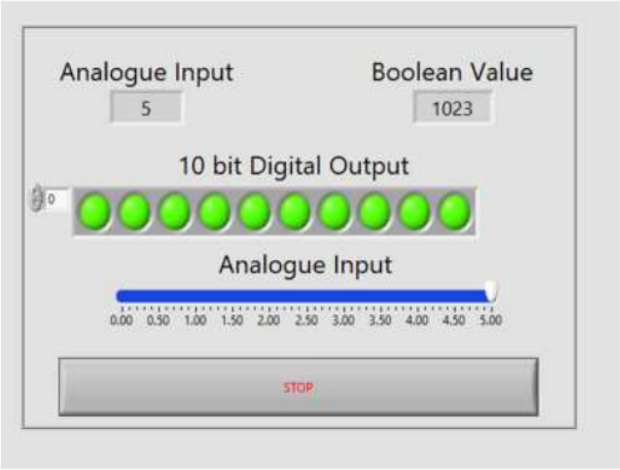



*Figure 54 - Testing 10-Bit ADC - Full Scale*

*Good test which shows the max value held steady. If the max raw value is set at 1024*

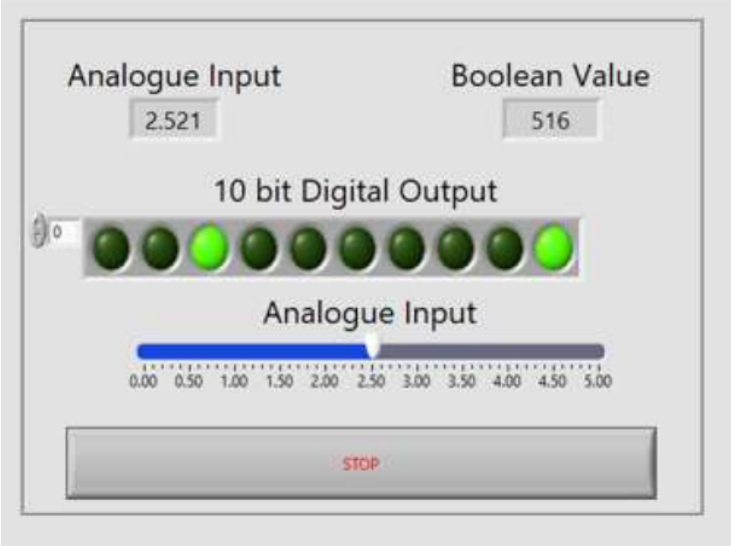
Table 7 shows the corresponding truth table, which confirms correct operation.

1+2+4+8+16+32+64+128+256+512 = 1023 - Correct

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

*Table 7 - Testing - 10-Bit ADC Truth Table - Full Scale*

Figure 55 shows the analogue input at approximately 2.5v, with the matching Boolean value close to 50% of the range.




*Good test showing value at half way point.*

*Figure 55 - Testing 10-Bit ADC - 50%*

Table 8 shows the corresponding truth table, which confirms correct operation.

4+512 = 516 - Correct

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 5 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |

*Table 8 - Testing - 10-Bit ADC Truth Table - 50%*

The results can be check with the equation $\frac{X_1}{V_{ref.max}}(2^n)$

Figure 55 = $\frac{2.521}{5}(1023) = 515.7966$ rounded to give LED array output of 516.

Convert back to an analogue signal;

$$\frac{515.79}{1023}(5) = 2.5209v$$

*Calc is good for proof*

Results in an error 0.00003v which is negligible.

### 2.4.3 12-Bit ADC testing

Figure 56 shows the input is set to approximately 50% with the resulting Boolean value matching. The truth table in table 9 conforms that the operation is correct.

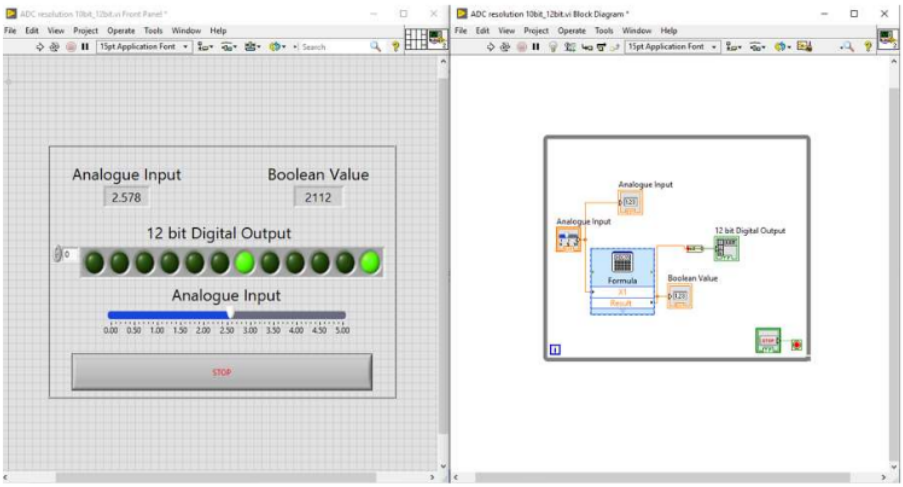



*Figure 56 - Testing - 12-Bit ADC 50%*

64+2048 = 2116 - Correct

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

*Table 9 - Testing - 12-Bit ADC Truth Table - 50%*

The results can be checked with the equation $\frac{X_1}{V_{ref.max}}(2^n)$

Figure 56 = $\frac{2.578}{5}(4095) = 2111.382$ rounded to give LED array output of 2112

Convert back to an analogue signal;

$$\frac{2111.382}{4095}(5) = 2.5775v$$

Results in an error 0.00046v which is negligible.

Figure 57 shows the 12-Bit ADC full-scale test, with 5v on the analogue input the output shows a Boolean value of 4095, with the results confirmed in the truth table in table 10.

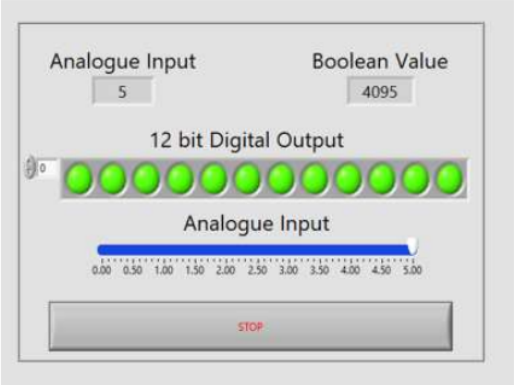



*Figure 57 - Testing - 12-Bit ADC 100%*

*Good test for 12 bit ADC indicating how LabVIEW handles the ADC arrays by a simple calculation and expansion of array.*

1+2+4+8+16+32+64+128+256+512+1024+2048 = 4095 - Correct

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

*Table 10 - Testing - 12-Bit ADC Truth Table - 100%*

## 2.4.4 8-Bit ADC testing

The image shown in figure 58 illustrates both tests carried out on the 8-Bit ADC with the variable power supplies set to 2.56V and 4.90V respectively.

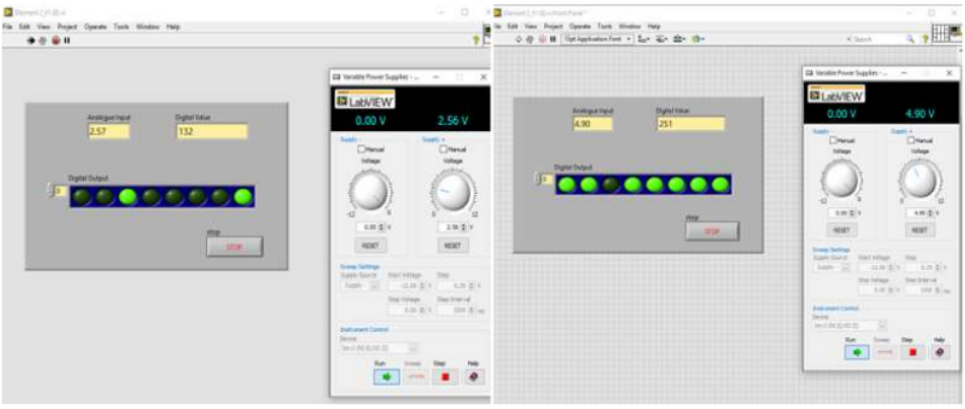



*Figure 58 - Testing - 8-Bit ADC 2.56v & 4.90v*

*Good practical test using virtual input signal from virtual P.S.*

The first digital output is 132 derived from a 2.57V virtual input and the second is 251 derived from a 4.90V virtual input. To confirm these values are correct and therefore the ADC is performing correctly the truth tables below were used.

4+128 = 132 - Correct

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

*Table 11 - Testing - 8-Bit ADC Truth Table - 50%*

There is a small error apparent calculated to be 0.008V, which is negligible.

1+2+8+16+32+64+128 = 251 - Correct

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

*Table 12 - Testing - 8-Bit ADC Truth Table - 100%*

There is a small error apparent calculated to be 0.002V, which is negligible.

*A good test indicating a small error which is acceptable.*

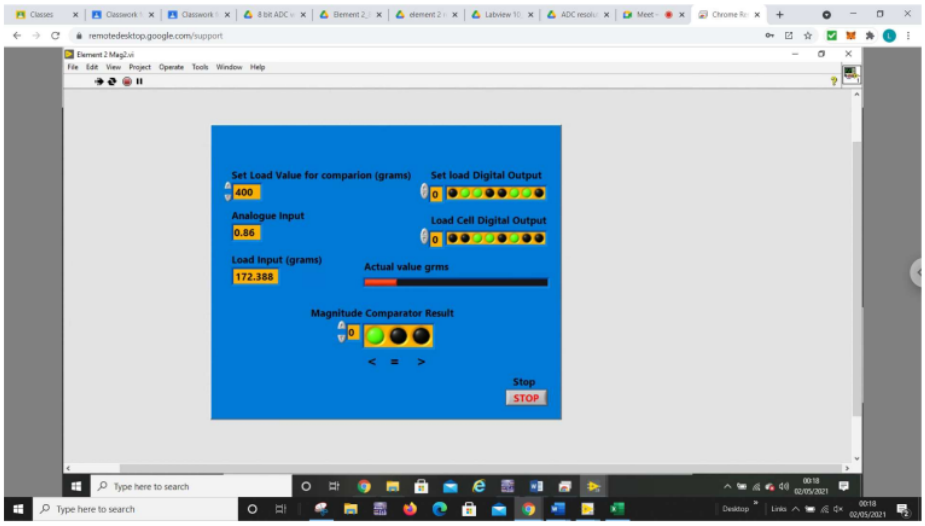
In conclusion all three ADC worked as required, albeit some very small errors which were calculated and found to be insignificant.

### 2.4.5 Magnitude Comparator Testing

The complete program tests carried through manually adjusting the input load with the potentiometer enabled three states to be evaluated, where the setpoint loads of 400, 200, and 200 has been used with the potentiometer loads varied greater than, equal to and less than to simulate the load cell circuit operation.

Test images can be seen below in figures

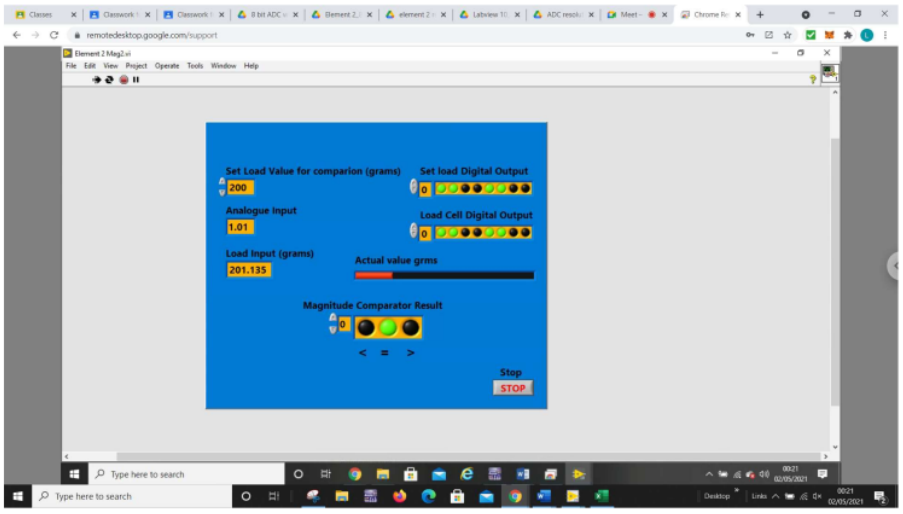
Less Than proven with the load input set lower than the setpoint value.




*Test is correct for value less than setpoint of 400grms*

*Figure 59 - Testing - Magnitude Comparator - Less Than*

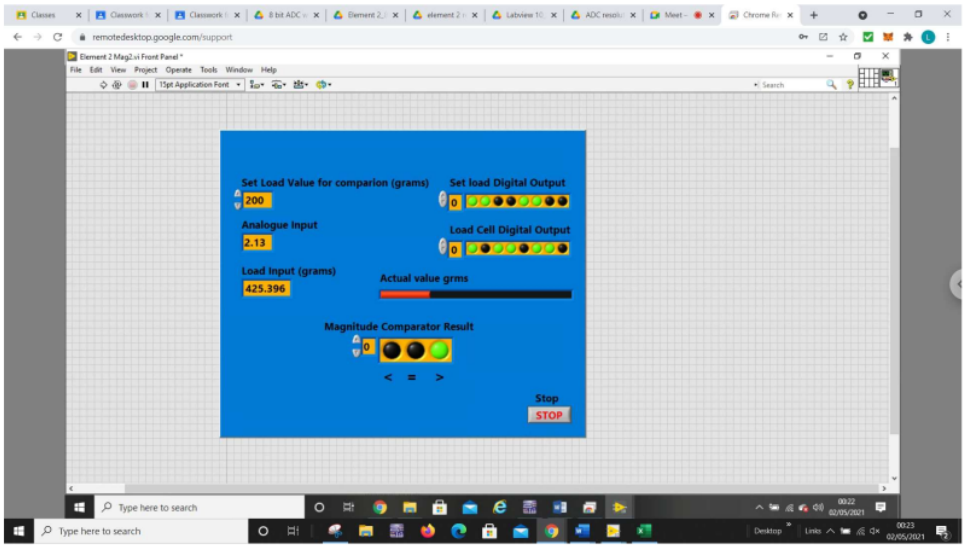
Equal to proven with the load input value set equal to the setpoint value




*Test is correct for value equal to setpoint of 200grms*

*Figure 60 - Testing - Magnitude Comparator - Equal To*

Greater than proven with the load input value set greater than the setpoint value




*Test is correct for value greater than setpoint of 200grms*

*Figure 61 - Testing - Magnitude Comparator - Greater Than*

The three tests conducted in the images above confirm that the complete system is functioning correctly. Seamless integration between the 8-Bit ADC and 8-Bit Magnitude comparator has been proven to be a success.

## 2.5 Evaluation

The researched load cell theory proved to be valuable during the simulation setup and testing of the load cell, of which the 0-10mv signal was successfully conditioned to 0-5v using the instrument amplifier. The stability, accuracy and repeatability of the measuring cell circuit was more than acceptable, proven by the linear response graph, with no self-excitation of the amplifier or any noise of concern. The two prototype analogue-to-digital converters were found to be very effective in converting the 0-5v analogue signal to a binary value, with the LabVIEW software making the task very user-friendly. Some error was apparent, but found to be insignificant.

The successful creation of the 8-Bit ADC enabled the progression onto the next element, the combination of the load cell circuit, 8-Bit ADC and 8-Bit magnitude comparator. The magnitude comparator was easily adapted to integrate into the ELVIS II board and the tests results confirmed that the whole circuit functioned as intended, exhibiting stability and insignificant error.

In conclusion, it has been deemed possible to create ADC scaling within LabVIEW, using the formula blocks provided in the program, to successfully compare values.

Further improvements to the system could include using dedicated connector cables and ensuring that the environmental impact of creating the system is kept to a minimum, through the selection of low-carbon footprint components.

# Element 3

## 3.1 Element 3 Introduction

Element 3 concerns itself with data recording and storage. Through the addition of data recording blocks or dataloggers, the LabVIEW circuit used in element 2, can be adapted to record 5 parameters from the system.

The parameters to be recorded are; Load input (grams), Set load comparison (grams), Load input less than set load, Load input equal to set load and load input more than set load. The stored data will be obtained from the system through using CSV text files.

*Introduction covers the practical task*

## 3.2 Design Requirements

The recorders feed the comparator binary values through multipliers which increase the gain by 200 for each data stream.

The datalogger file configuration screen can be seen in figure 62 below. The block allows the user to select the save location, format (text, binary, excel etc.), header allocation and the option to rename or overwrite existing files.

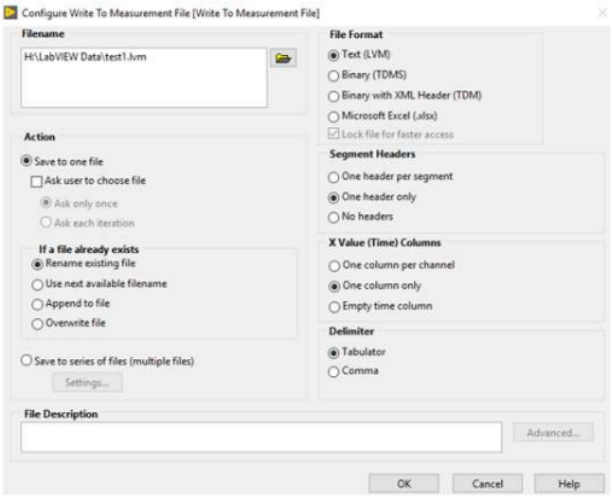



*Figure 62 - Datalogger Block Configuration*

*Text file configuration example shows the extension and file extension of .lvm*

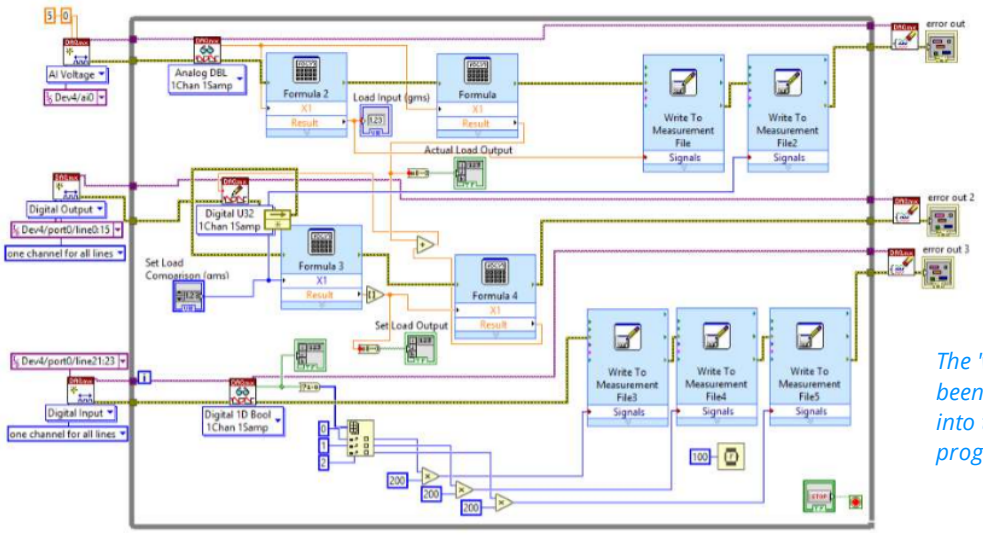
For the purpose of this element, the configuration in figure 62 will be used, where the save location is designated as H:\LabVIEW Data\test1.lvm, where the files will be stored as text files to be manually converted to excel which will enable all saved data to be graphically represented. Appendix figures 19-23 show the file configuration of all four dataloggers.

Appendix figure 24 shows the .lvm files saved in the appropriate location, confirming that the save address has been implemented correctly.

## 3.3 Implementation

Figure 63 shows the datalogger elements added to the established comparator circuit on LabVIEW.

The addition of the dataloggers enables the potentiometer analogue input and comparator outputs to be saved for future scrutiny.

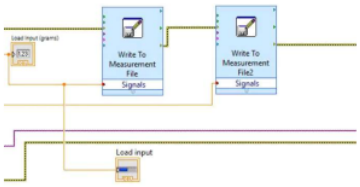



*The 'write files' have been implemented into the main program correctly.*

*Figure 63 - Full Program*

The individual elements of the program are elaborated upon below.

Referring to figure 64; File 1 is a .lvm text file and logs load input value as numeric value, where file 2 reads the setpoint integer (load setpoint comparison) in grams.




*Functions of file 1 and 2 are explained*

*Figure 64 - File 1 and 2*

Referring to figure 65, the first block is the converter that converts Boolean to integer, the second is the index array for elements 0,1 & 2 (these are from inputs 21, 22, 23). These values are multiplied by 200 to aid in graph scaling.

*Array and scaling fu*

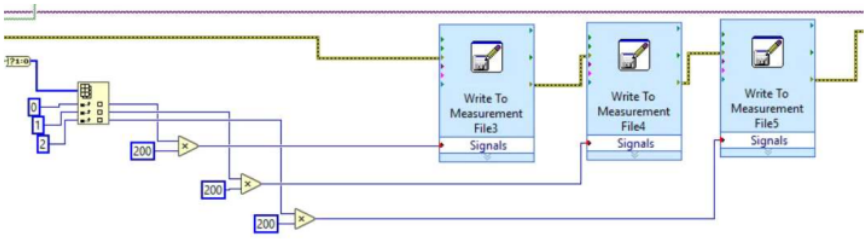



*Figure 65 - File 3, 4 & 5*

The Data logger blocks functions are; File 3 (less than), File 4 (equal to), File 5 (Greater than)

Individual File settings can be seen in appendix*** *Files 3-5 are explained*

Image 66 below shows the sample timer, to ensure a reasonable amount of data is sampled (in milliseconds). Too low = too much data / too high = not representative.




*Timer function is explained*

*Figure 66 - Sample Timer*

See appendix figures 25-29 for .lvm samples from datalogging on file 1,2,3,4,5.

## 3.4 Testing and Analysis

The first test conducted is the 200g simulated setpoint. Figure 67 shows the data logger test result graph – see appendix figure 30 for data logs.

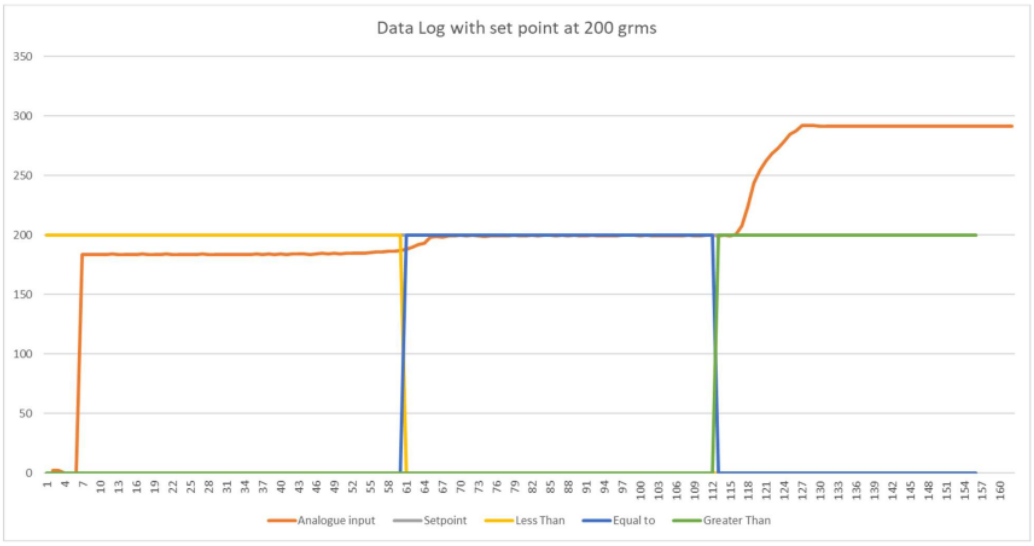



*Figure 67 – Testing - 200g Setpoint Log Graph*

The graph in figure 67 shows the analogue input (orange line), which is the simulated weight to initially be below the setpoint of 200, when set at 133.62g, with the yellow (less than) line at a high value which is correct. Thereafter the yellow line drops off to a low value and the blue (equal to) line goes high, which is when the setpoint matches the simulated load value, which is correct. Finally, the blue line drops to a low value, at which point the green line goes high to indicate the simulate value is higher than the setpoint.

Front panel display test results confirm correct operation in line with the data log graph. Note that there is a small error of 2.966g on the simulated weight value during the equal to test.

Equal to | Less than | More than

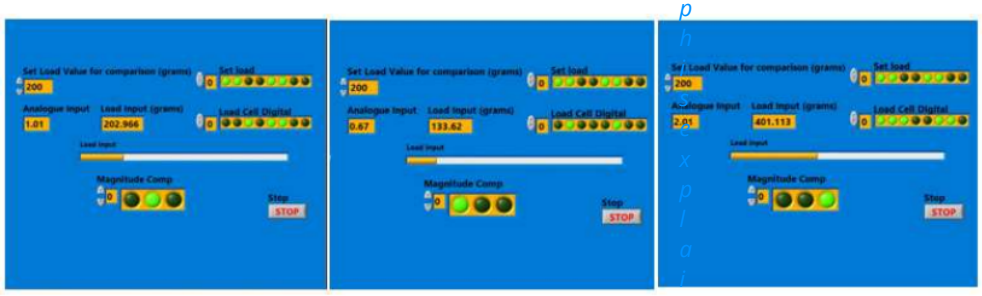


*Figure 68 - Testing - 200g HMI Test*

*Good set of results for comparator setting*

The second test conducted is the 500g simulated setpoint. Figure 69 shows the data logger test result graph – see appendix figure 31 for data logs.




*Figure 69 – Testing - 500g Setpoint Log*

The graph in figure 69 shows the analogue input (orange line), which is the simulated weight to initially be below the setpoint of 500, when set at 341.723g, with the yellow (less than) line at a high value which is correct. Thereafter the yellow line drops off to a low value and the blue (equal to) line goes high, which is when the setpoint matches the simulated load value of 500g, which is correct. Finally, above 500g, the blue line drops to a low value, at which point the green line goes high to indicate the simulated value is higher than the setpoint, which is correct.

Front panel display test results confirm correct operation in line with the data log graph. Note that there is a small error of 2.099g on the simulated weight value during the equal to test.

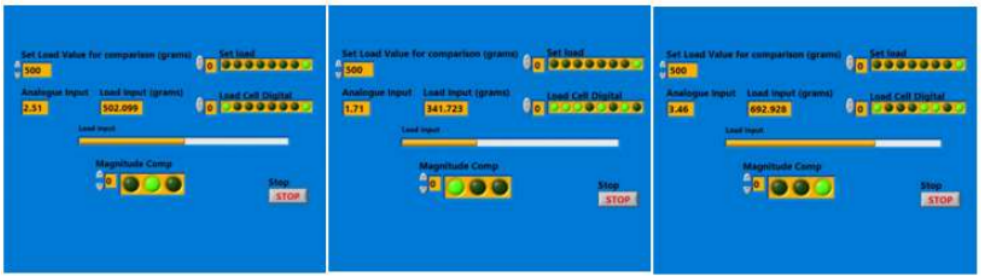



*Figure 70 - Testing - 500g HMI Test*

*Good set of results on front panel indicating all 3 conditions*

The third test conducted is the 700g simulated setpoint. Figure 71 shows data logger test result graph – see appendix figure 32 for data logs.

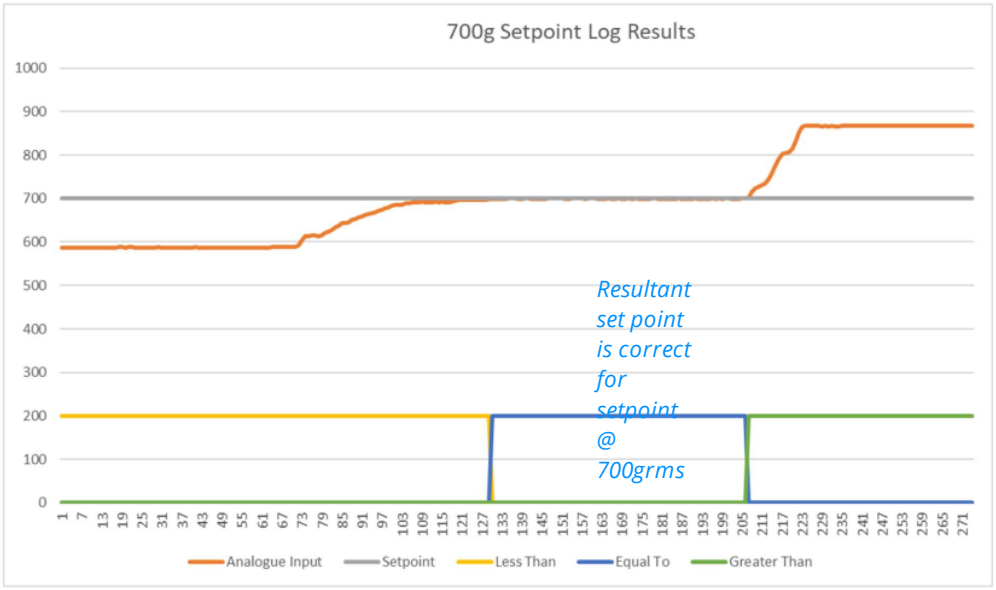



*Figure 71 - Testing - 700g Setpoint Log Graph*

The graph in figure 71 shows the analogue input (orange line), which is the simulated weight to initially be below the setpoint of 700, when set at 447.495g, with the yellow (less than) line at a high value which is correct. Thereafter the yellow line drops off to a low value and the blue (equal to) line goes high, which is when the setpoint matches the simulated load value of 500g, which is correct. Finally, above 700g, the blue line drops to a low value, at which point the green line goes high to indicate the simulated value is higher than the setpoint, which is correct

Front panel display test results confirm correct operation in line with the data log graph. Note that there is a small error of 1.408g on the simulated weight value during the equal to test.

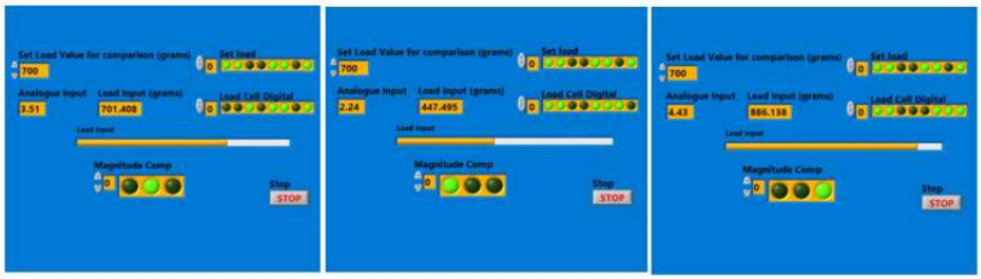



*Figure 72 - Testing - 700g HMI Test*

*Results are correct for all 3 condition on front panel*

## 3.5 Conclusion and Analysis

The tests conducted confirms that the element objectives have been met. The 8-Bit ADC successfully converted the analogue weight signal to a digital signal, after which the magnitude comparator successfully distinguished between the setpoint value and simulated weight value and ascertained if the simulated weight is less than, equal to or higher than the setpoint value. The data logging elements were added and tested and found to be functioning correctly, with an appropriate sampling interval to represent an accurate dataset. This element has highlighted how data logging can be implemented into industrial or manufacturing systems to improve reliability, efficiency and control.

Error has been evident throughout the system, starting with very small errors in the load cell signal conditioning element, which was deemed to be insignificant. Care must be taken to ensure the aforementioned error does not become amplified, when further elements are added to the design. Furthermore, quantisation error was apparent during the datalogging tests “equal to” simulation, this is due to the 8-Bit resolution of the ADC. Further improvements to the system can be made through utilising an ADC with higher resolution, like the 10-bit prototype, which would require further circuit amendments.

*Good evaluation*

# Overall Conclusion

The research conducted enabled a thorough understanding of the basic principles needed to achieve the goals of the three elements completed. Knowledge gained on the integration and programming of Elvis II and Nexys4 DDR boards to communicate with each other and accept analogue input signals can be applied to work environments as transferrable skills.

The programming of the boards using LabVIEW and Multisim illustrates the versatility of these systems and thereby their usefulness in industry. The manner in which the boards are programmed enable a visual link to be established to the user which greatly increases the understanding of the topic.

## References


Ashenden, P., 2008. *Digital Design.* 1st ed. London: Elsevier.

Ashenden, P., 2008. *Digital Design - An Embedded Systems Approach.* 1st ed. London: Elsevier .

Aspencore, 2021. *Electronics-Tutorials.* [Online] Available at: https://www.electronics-tutorials.ws/combination/comb_8.html [Accessed 03 01 2021].

Belleman, J., 2008. *Analogue to digital,* Geneva: CERN Switzerland.

Dewesoft, 2020. *Dewesoft.com.* [Online] Available at: https://dewesoft.com/daq/types-of-adc-converters [Accessed 04 01 2021].

Dialog Semiconductor, 2018. *Digital Magnitude / Identity Comparator.* Swindon: Dialog Semiconductor.

Gudino, M., 2018. *Arrow Electronics.* [Online] Available at: https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters [Accessed 03 01 2021].

Haskell, R., 2009. *Introduction to Digital Design.* 1st ed. Rochester: LBE Books.

HBM, 2020. *The Working Principle of a Compression Load Cell.* [Online] Available at: https://www.hbm.com/en/7325/the-working-principle-of-a-compression-load-cell/ [Accessed 04 01 2021].

Intel, 2017. *Intel.com.* [Online] Available at: https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/reference/glossary/def_vhdl.htm [Accessed 19 12 2020].

Kuphaldt, T., 2007. *Lessons In Electric Circuits.* 4th ed. Washington: s.n.

National Instruments, 2021. *NI.com.* [Online] Available at: https://www.ni.com/en-gb/shop/data-acquisition.html [Accessed 02 01 2021].

Omega, 2021. *Sensor Signal Conditioning.* [Online] Available at: https://www.omega.co.uk/prodinfo/signal-conditioners.html [Accessed 16 01 2021].

OMEGA, 2021. *Subminiature Load Cell LCKD Series Rugged Industrial Design.* [Online] Available at: https://www.omega.co.uk/pptst/LCKD.html [Accessed 05 01 2021].

OMEGA, 2021. *Why Load Cell Signal Conditioner is Critical to a Strain Gauge Setup.* [Online] Available at: https://www.omega.com/en-us/resources/load-cell-signal-conditioners [Accessed 06 01 2021].

OMEGA, n.d. [Online].

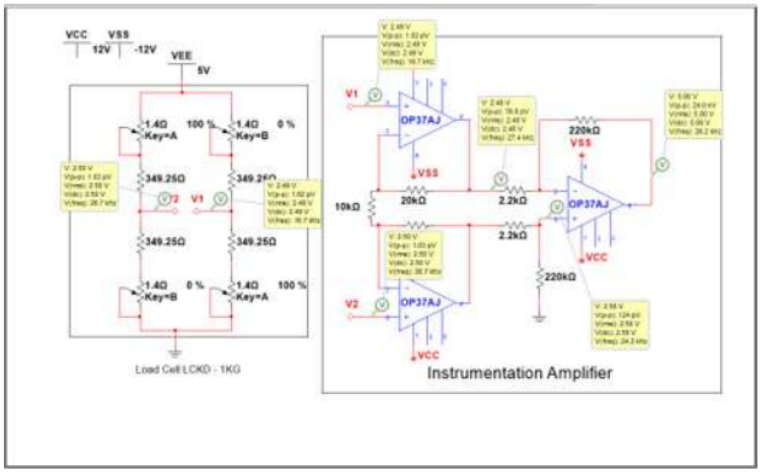
Transducer Techniques, 2020. *Quarter Bridge, Hals Bridge and Full Wheatstone Bridge Strain Gauge Load Cell configurations.* [Online] Available at: https://www.transducertechniques.com/wheatstone-bridge.aspx#:~:text=The%20Wheatstone%20bridge%20is%20the,two%20parallel%20voltage%20divider%20circuits.&text=The%20Wheatstone%20bridge%20configuration%20is,physical%20change%20in%20the%20specimen. [Accessed 05 01 2021].

Transducer Techniques, 2020. *TransducerTechniques.com.* [Online] Available at: https://www.transducertechniques.com/wheatstone-bridge.aspx#:~:text=The%20Wheatstone%20bridge%20is%20the,two%20parallel%20voltage%20divider%20circuits.&text=The%20Wheatstone%20bridge%20configuration%20is,physical%20change%20in%20the%20specimen. [Accessed 04 01 2021].

*Good number of relevant sources*

# 4. Appendix

## Appendix - Element 2



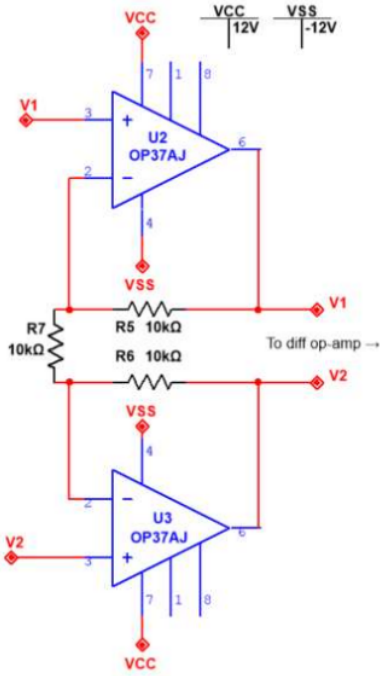

*Appendix Figure 1 - Multisim Circuit of Load Cell and Amplifier*

Measuring Cell Specifications

**SPECIFICATIONS OF LCKD-1KG**
**Excitation:** 5 Vdc, 7 Vdc max
**Output:** 2 mV/V nominal
**5-Point Calibration:** 0%, 50%, 100%, 50%, 0%
**Linearity:** ±0.25% FSO
**Hysteresis:** ±0.25% FSO
**Repeatability:** ±0.10% FSO
**Zero Balance:** ±2% FSO
**Operating Temp Range:** -54 to 121°C (-65 to 250°F)
**Compensated Temp Range:** 16 to 71°C (60 to 160°F)
**Thermal Effects:**
**Span:** ±0.018% FSO/°C
**Zero:** ±0.009% FSO/°C
**Safe Overload:** 150% of capacity
**Ultimate Overload:** 300% of capacity
**Bridge Resistance:** 350 Ω min
**Full Scale Deflection:** 0.025 to 0.076 mm (0.001 to 0.003")
**Electrical Connection:** 1.5 m (5') 4-conductor insulated cable with temperature compensation board
**Weight:** <14 g (<0.5 oz)
**Protection Class:** IP54

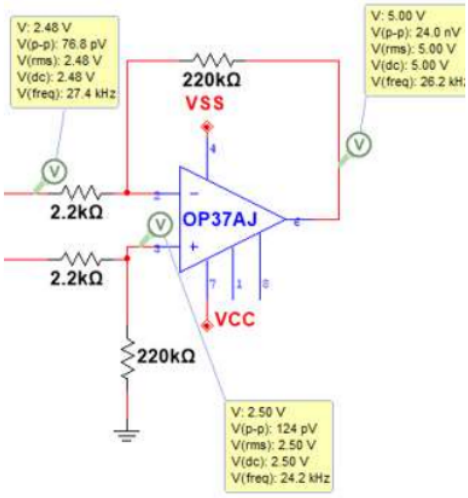
*Appendix Figure 2 - Measuring Cell Specifications*

Buffer amplifier configuration



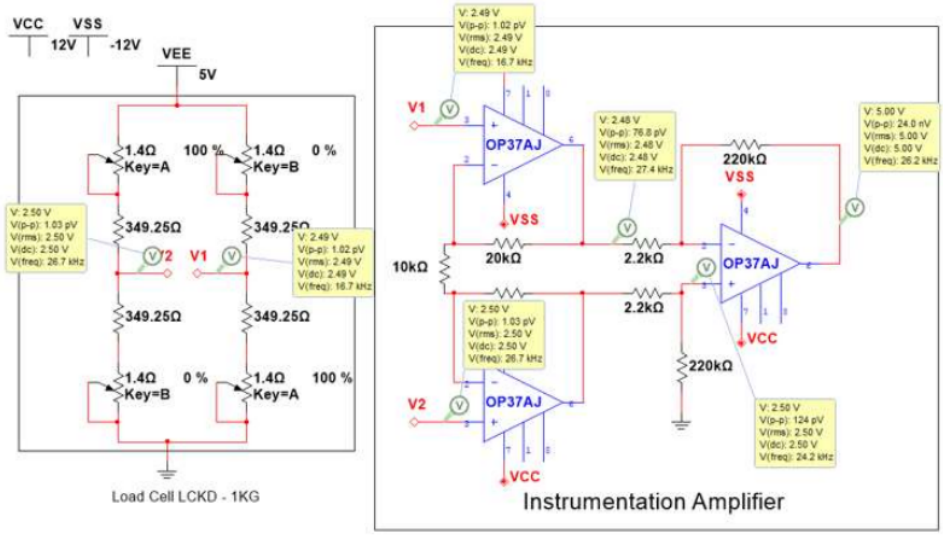

*Appendix Figure 3 - Buffer Amp Configuration*

Differential op-amp with unity gain




*Appendix Figure 4 - Differential Op-amp - Unit gain*

Combined circuit




*Appendix Figure 5 - Combined Load-cell and Amplifier Circuit*

Standard Resistor Value table

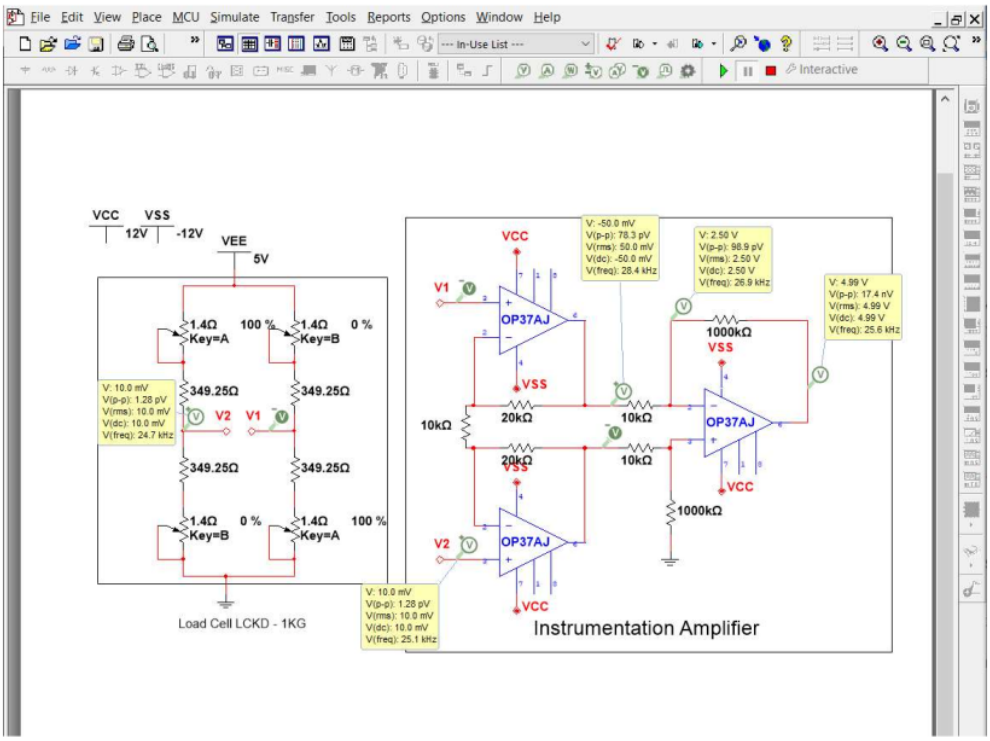
| Standard Resistor Values (±5%) | | | | | | |
|---|---|---|---|---|---|---|
| 1.0 | 10 | 100 | 1.0K | 10K | 100K | 1.0M |
| 1.1 | 11 | 110 | 1.1K | 11K | 110K | 1.1M |
| 1.2 | 12 | 120 | 1.2K | 12K | 120K | 1.2M |
| 1.3 | 13 | 130 | 1.3K | 13K | 130K | 1.3M |
| 1.5 | 15 | 150 | 1.5K | 15K | 150K | 1.5M |
| 1.6 | 16 | 160 | 1.6K | 16K | 160K | 1.6M |
| 1.8 | 18 | 180 | 1.8K | 18K | 180K | 1.8M |
| 2.0 | 20 | 200 | 2.0K | 20K | 200K | 2.0M |
| 2.2 | 22 | 220 | 2.2K | 22K | 220K | 2.2M |
| 2.4 | 24 | 240 | 2.4K | 24K | 240K | 2.4M |
| 2.7 | 27 | 270 | 2.7K | 27K | 270K | 2.7M |
| 3.0 | 30 | 300 | 3.0K | 30K | 300K | 3.0M |
| 3.3 | 33 | 330 | 3.3K | 33K | 330K | 3.3M |
| 3.6 | 36 | 360 | 3.6K | 36K | 360K | 3.6M |
| 3.9 | 39 | 390 | 3.9K | 39K | 390K | 3.9M |
| 4.3 | 43 | 430 | 4.3K | 43K | 430K | 4.3M |
| 4.7 | 47 | 470 | 4.7K | 47K | 470K | 4.7M |
| 5.1 | 51 | 510 | 5.1K | 51K | 510K | 5.1M |
| 5.6 | 56 | 560 | 5.6K | 56K | 560K | 5.6M |
| 6.2 | 62 | 620 | 6.2K | 62K | 620K | 6.2M |
| 6.8 | 68 | 680 | 6.8K | 68K | 680K | 6.8M |
| 7.5 | 75 | 750 | 7.5K | 75K | 750K | 7.5M |
| 8.2 | 82 | 820 | 8.2K | 82K | 820K | 8.2M |
| 9.1 | 91 | 910 | 9.1K | 91K | 910K | 9.1M |

*Appendix Figure 6 - Standard Resistor Value Table (Kuphaldt, 2007)*

Load Circuit test results

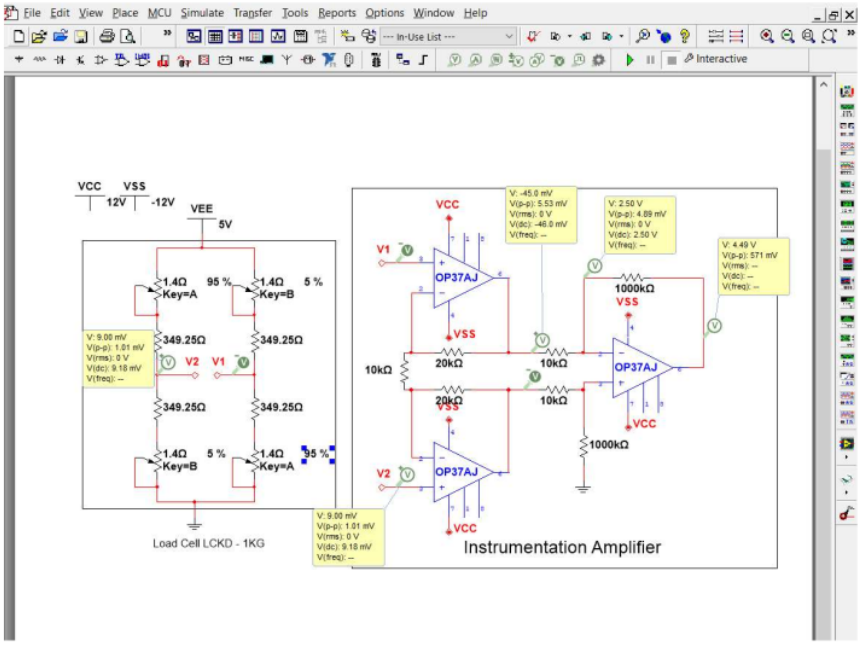
Circuit testing

100-0 4.99 V



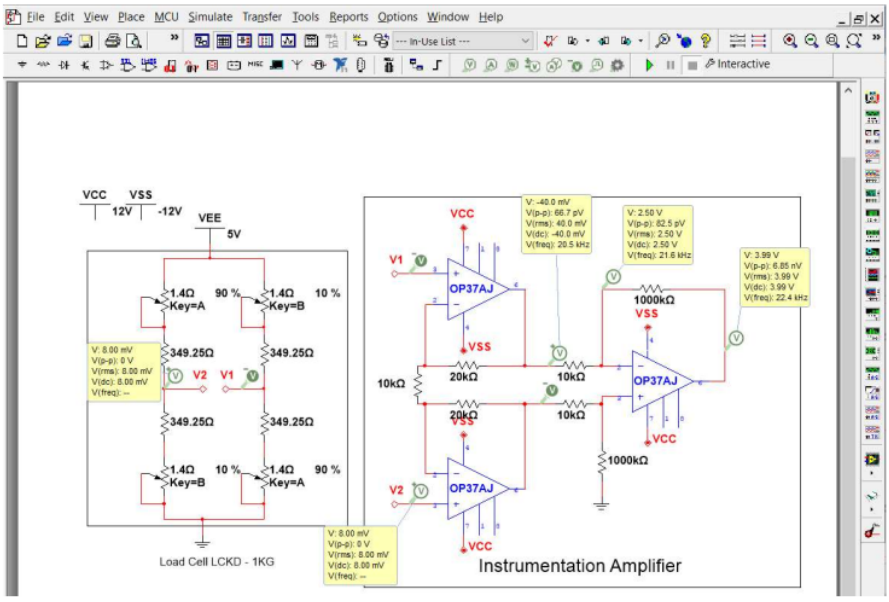

*Appendix Figure 7 Load Cell Testing 5V*

95-5 4.49V



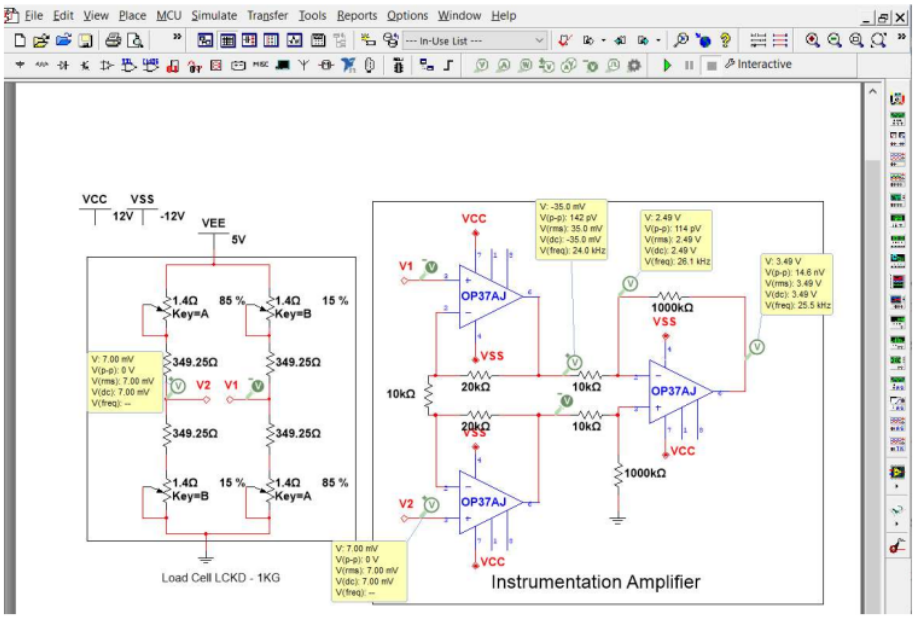

*Appendix Figure 8 - Load Cell Testing 4.5V*

90-10 3.99V



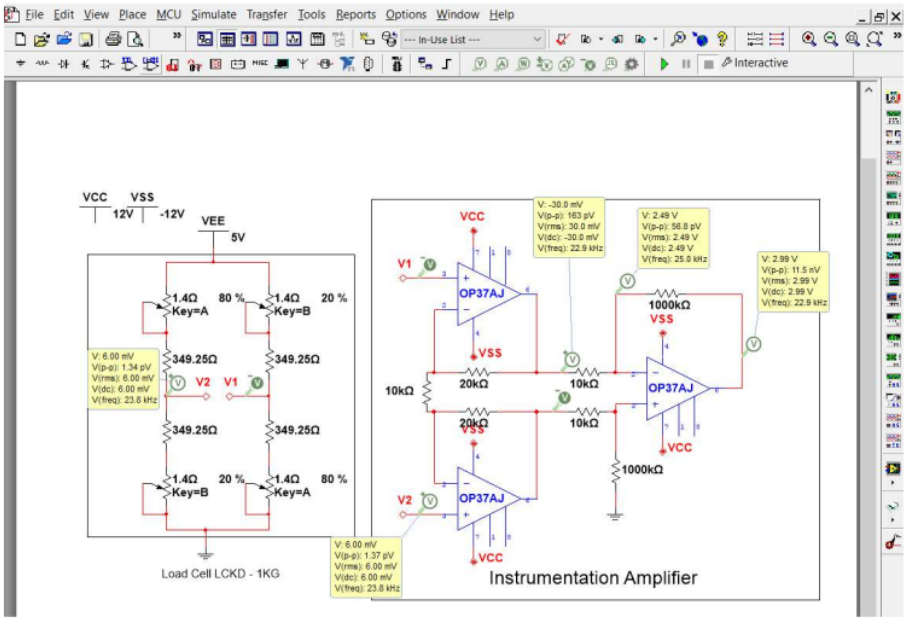

*Appendix Figure 9 - Load Cell Testing 4V*

85-15 3.49V



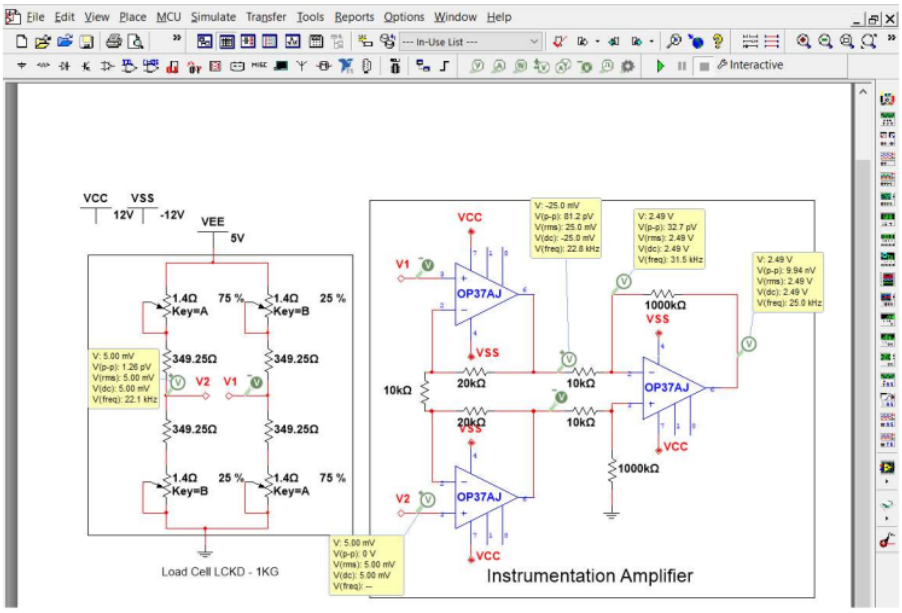

*Appendix Figure 10 - Load Cell Testing 3.5V*

80-20 2.99V



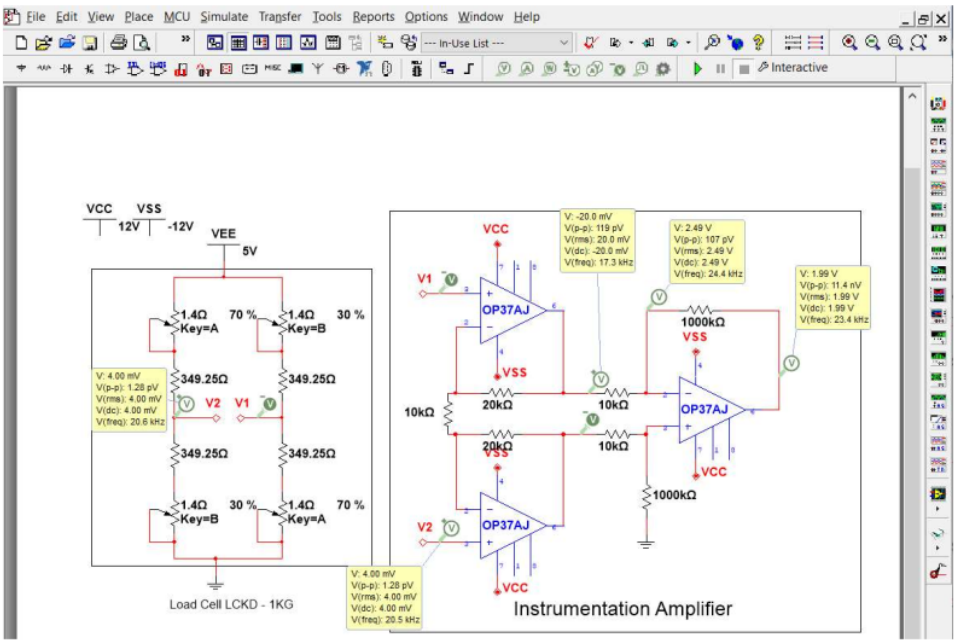

*Appendix Figure 11 - Load Cell Testing 3V*

75-25 2.49V



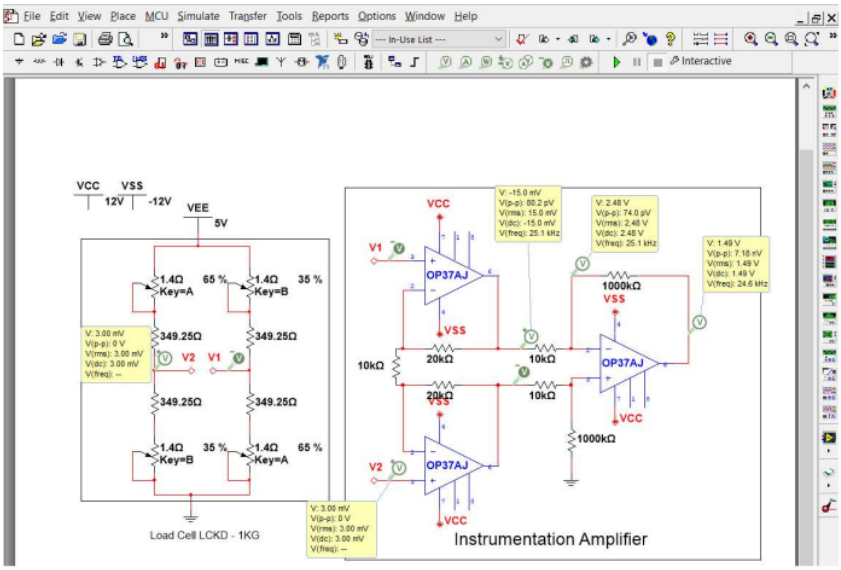

*Appendix Figure 12 - Load Cell Testing 2.5V*

70-30 1.99V



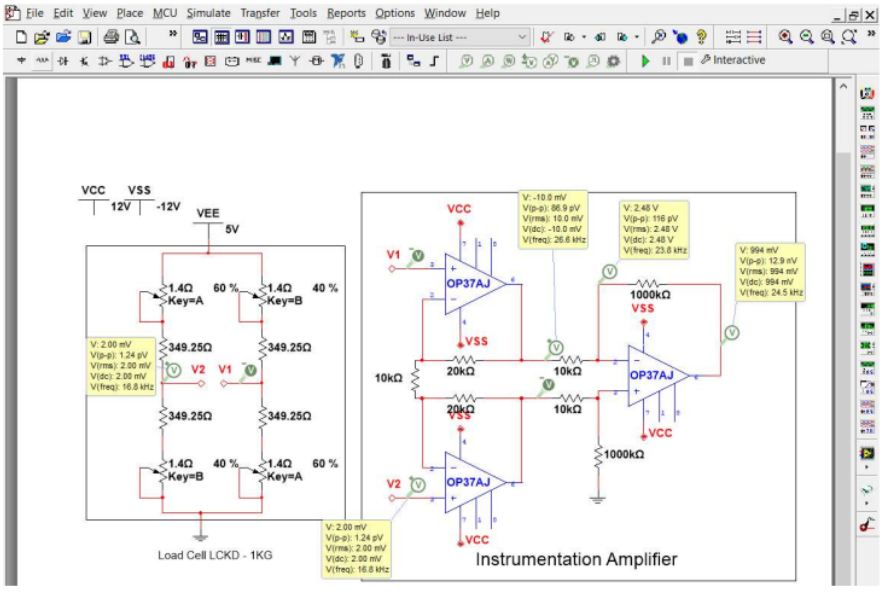

*Appendix Figure 13 - Load Cell Testing 2V*

65-35 1.49V



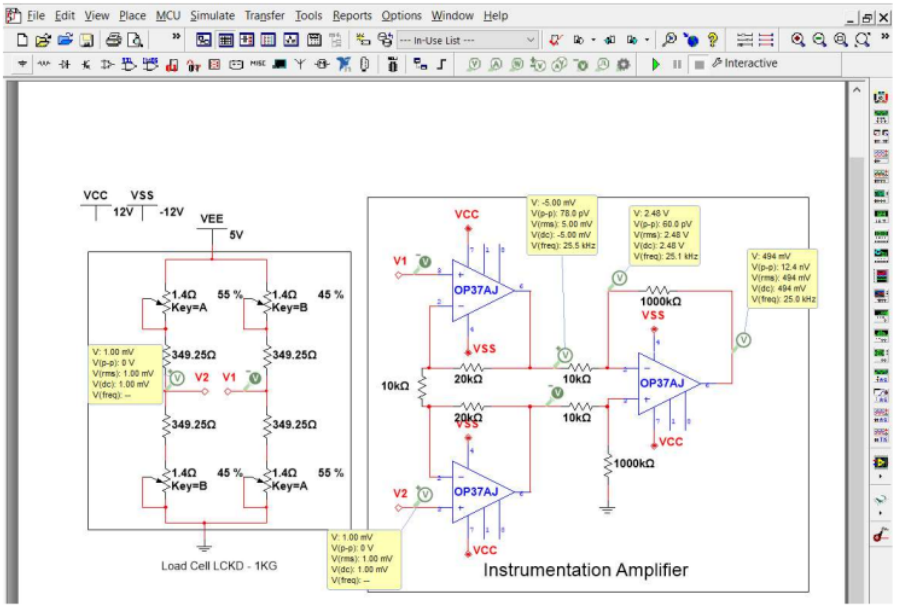

*Appendix Figure 14 - Load Cell Testing 1.5V*

60-40 – 994mV



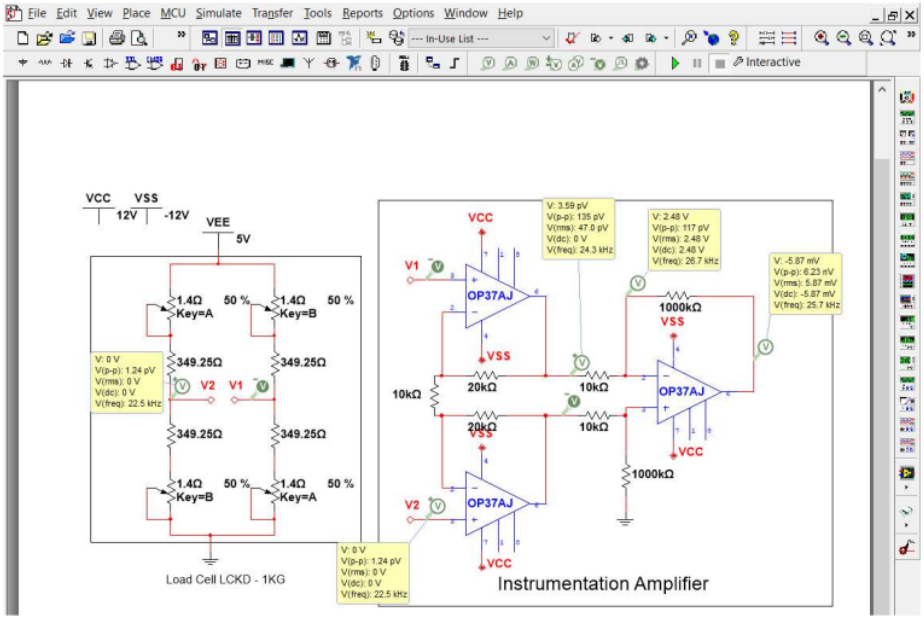

*Appendix Figure 15 - Load Cell Testing 1V*

55-45 – 494mV




*Appendix Figure 16 - Load Cell Testing 0.5V*

50-50 - –5.87mV (0v)

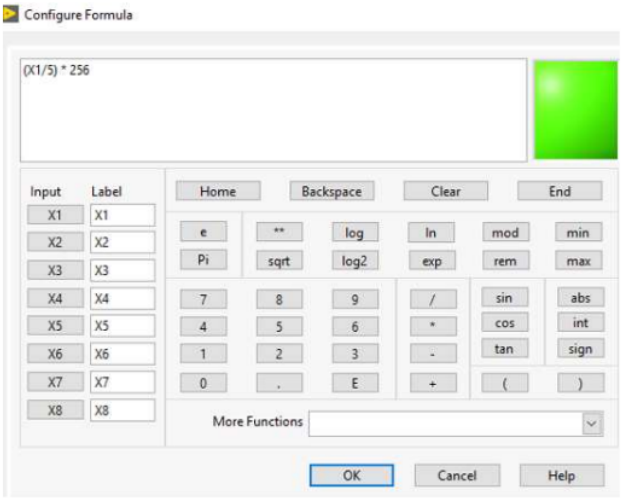



*Appendix Figure 17 - Load Cell Testing 0V*



*Full testing of amplifier is*

8-Bit ADC Formula Block equation



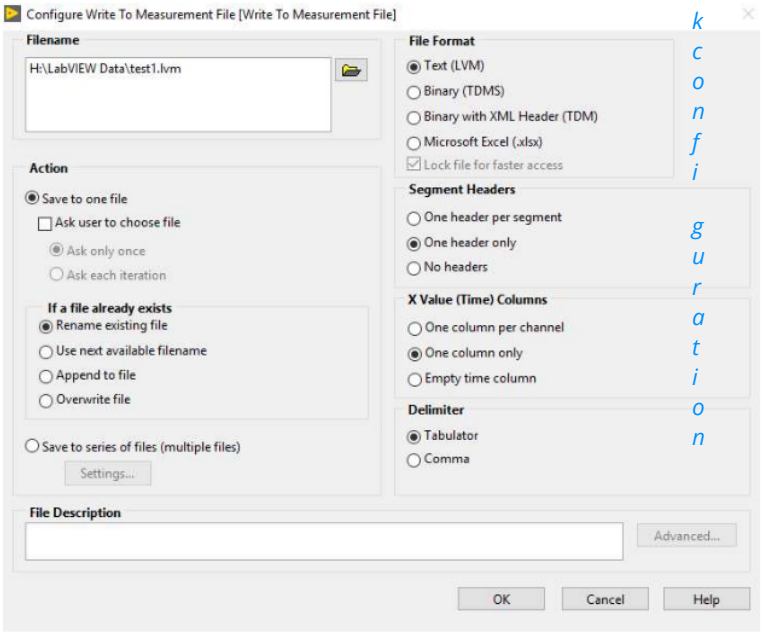

*Appendix Figure 18 - 8-Bit ADC Formula Block Equation*

*8bit ADC formula block configuration*

## Appendix Element **3**

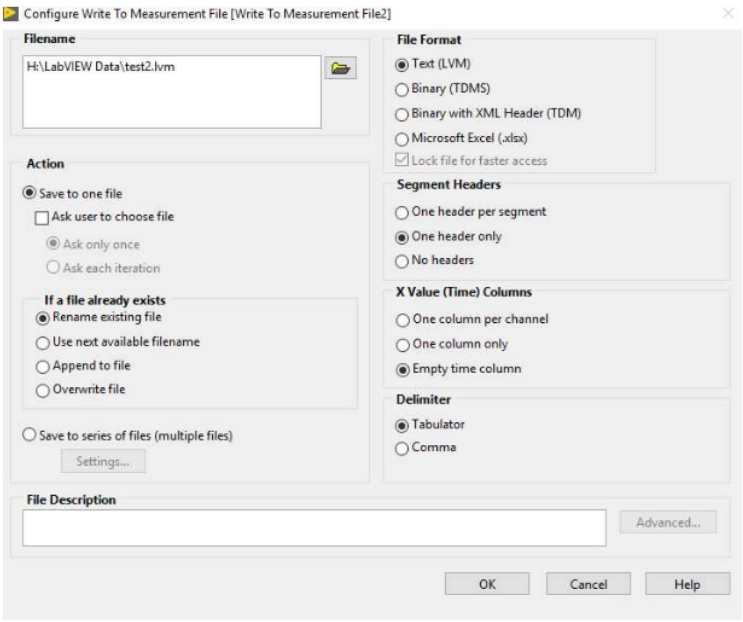
Data Logger file configurations;

File 1



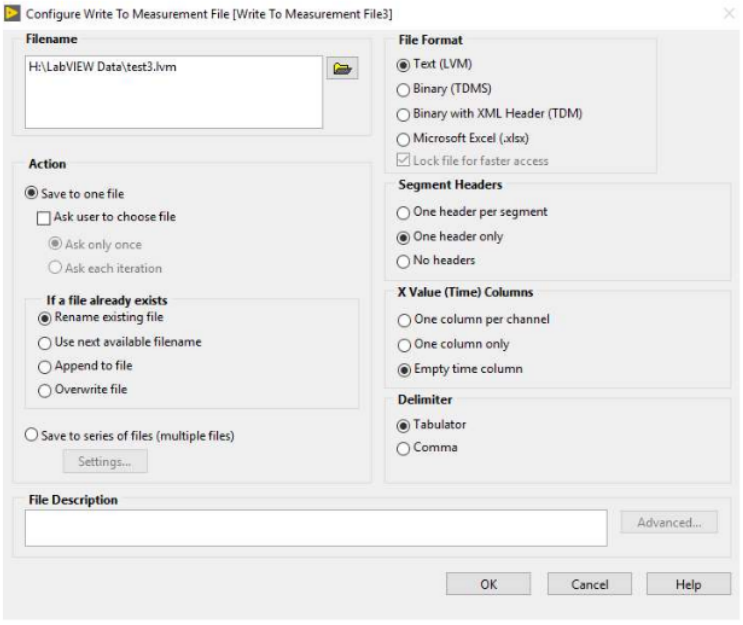

*Appendix Figure 19 - Data Logger File Configuration File 1*

File 2



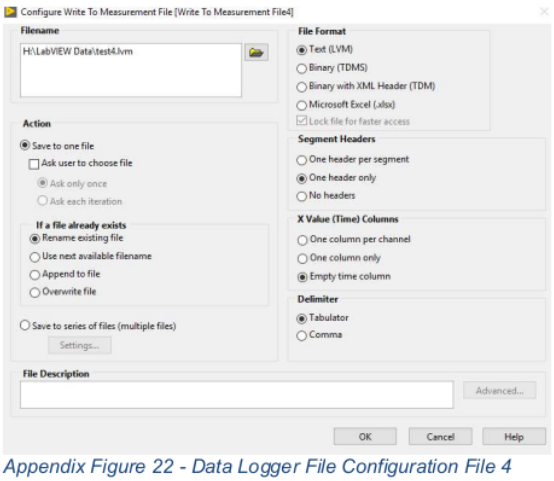

*Appendix Figure 20 - Data Logger File Configuration File 2*

File 3



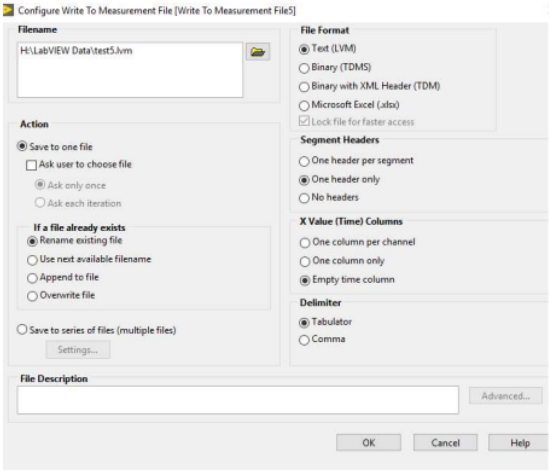

*Appendix Figure 21 - Data Logger File Configuration File 3*

File 4



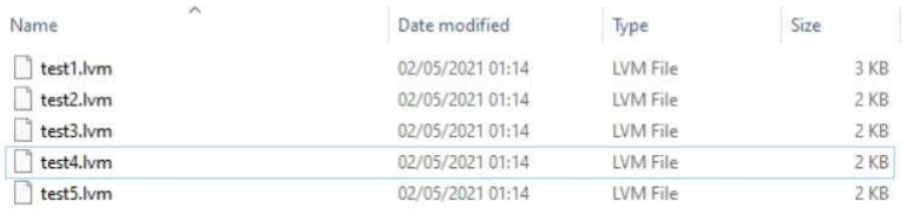

*Appendix Figure 22 - Data Logger File Configuration File 4*

File 5




*Appendix Figure 23 - Data Logger File Configuration File 5*

Raw data for datalogging - Generated text files

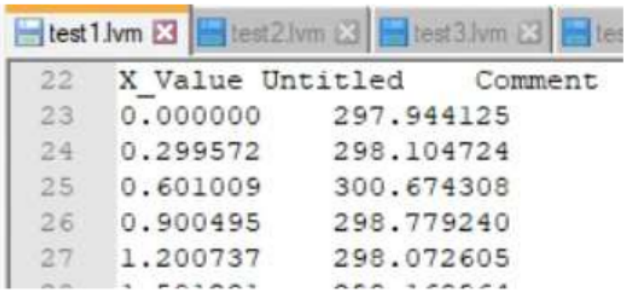



*Appendix Figure 24 - Raw data for datalogging - Generated text files*



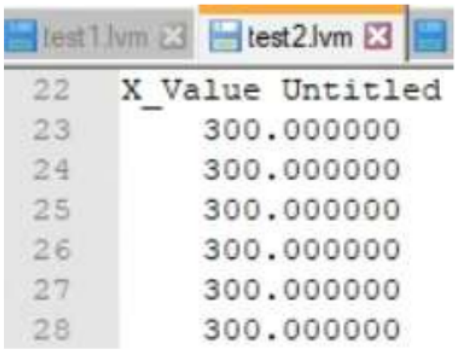
*Text file configurations are demonstrated*

File 1 sample



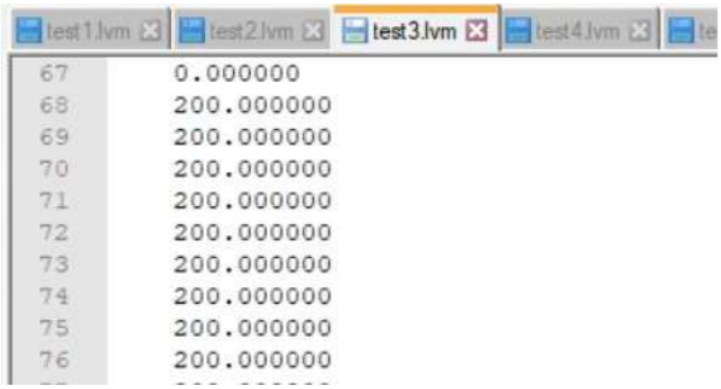

*Appendix Figure 25 - Data Log File Sample - File 1*

File 2 Sample



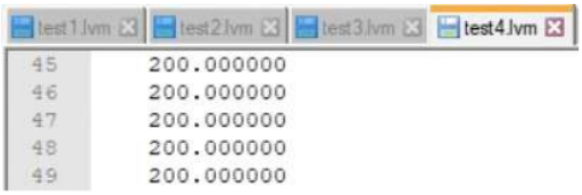

*Appendix Figure 26 - Data Log File Sample - File 2*

File 3 sample



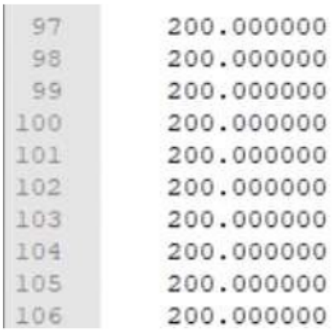

*Appendix Figure 27 - Data Log File Sample - File 3*

File 4 Sample




*Appendix Figure 28 - Data Log File Sample - File 4*

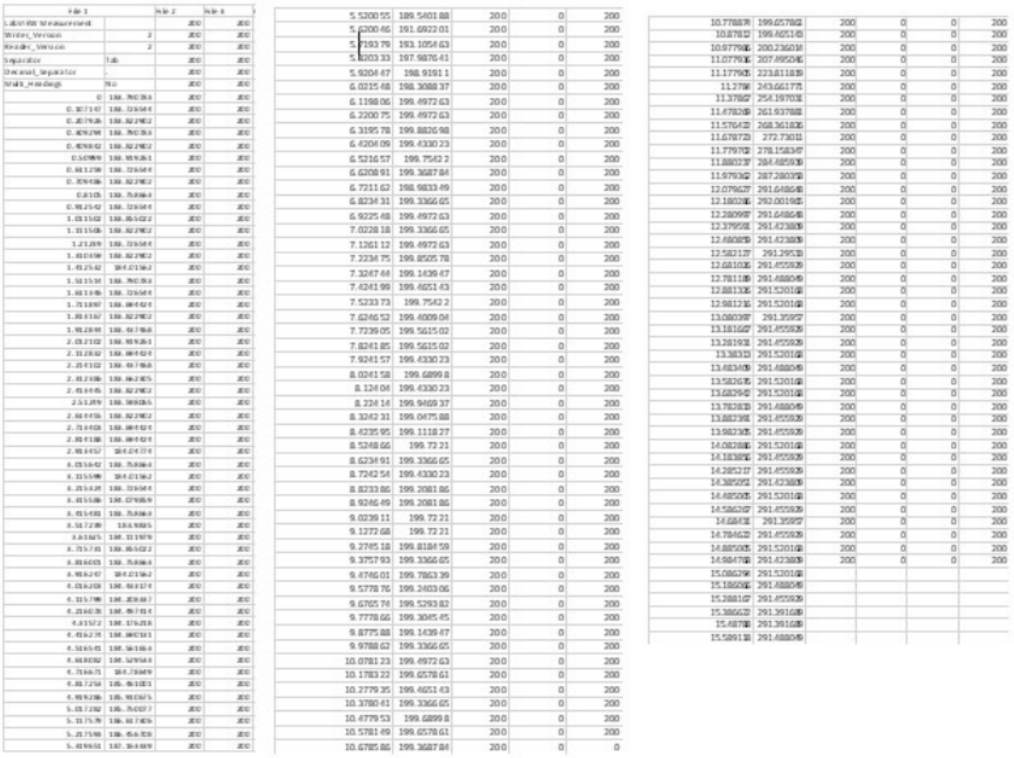
File 5 sample




*Raw value samples of text file data*

*Appendix Figure 29 - Data Log File Sample - File 5*

Load Cell 200g data log

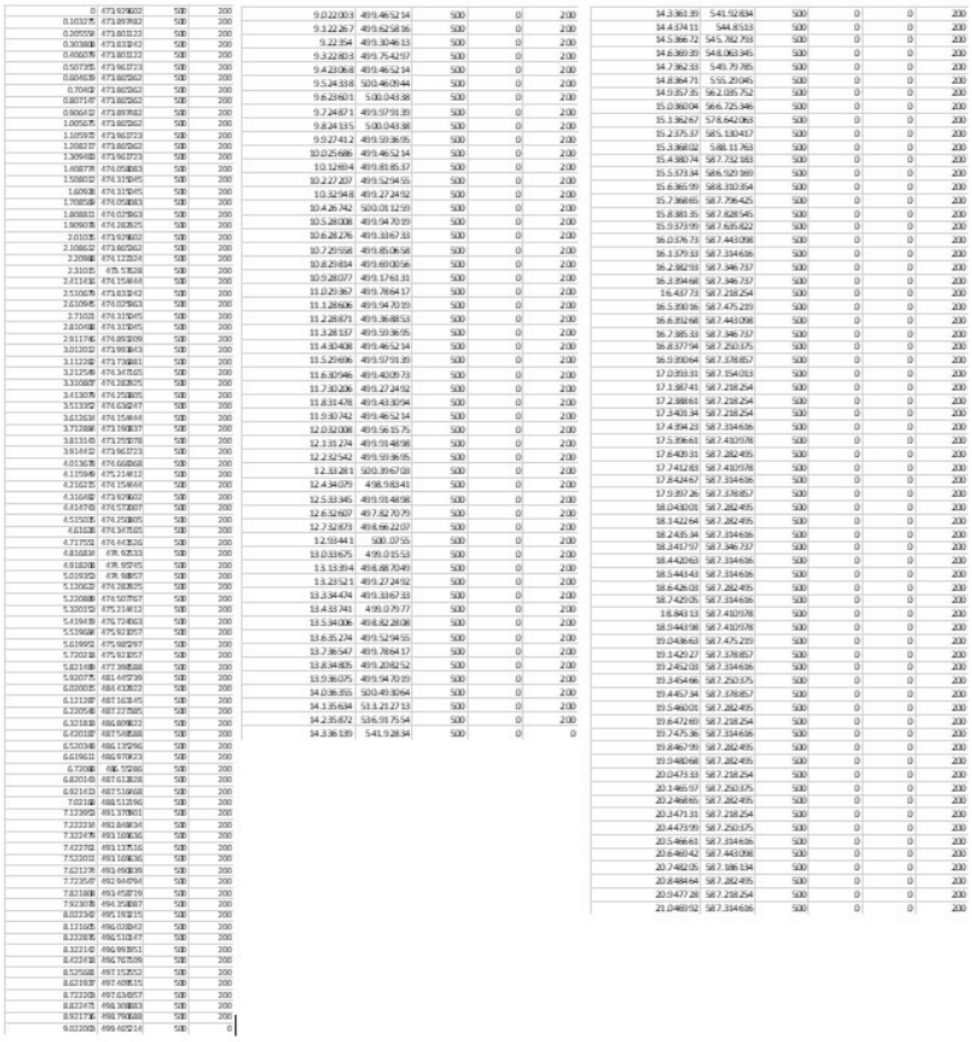


| | | | | | |
|---|---|---|---|---|---|
| 5.5200 55 | 189.5401 88 | 200 | 0 | 0 | 200 |
| 5.6200 46 | 191.6922 01 | 200 | 0 | 0 | 200 |
| 5.7193 79 | 193.1056 03 | 200 | 0 | 0 | 200 |
| 5.8203 33 | 197.9876 41 | 200 | 0 | 0 | 200 |
| 5.9204 47 | 198.9191 1 | 200 | 0 | 0 | 200 |
| 6.0215 48 | 198.3088 37 | 200 | 0 | 0 | 200 |
| 6.1198 06 | 199.4972 63 | 200 | 0 | 0 | 200 |
| 6.2200 75 | 199.4972 63 | 200 | 0 | 0 | 200 |
| 6.3195 78 | 199.8826 98 | 200 | 0 | 0 | 200 |
| 6.4204 09 | 199.4330 23 | 200 | 0 | 0 | 200 |
| 6.5216 57 | 199. 7542 2 | 200 | 0 | 0 | 200 |
| 6.6208 91 | 199.3687 84 | 200 | 0 | 0 | 200 |
| 6.7211 62 | 198.9833 49 | 200 | 0 | 0 | 200 |
| 6.8234 31 | 199.3366 65 | 200 | 0 | 0 | 200 |
| 6.9225 48 | 199.4972 63 | 200 | 0 | 0 | 200 |
| 7.0228 18 | 199.3366 65 | 200 | 0 | 0 | 200 |
| 7.1261 12 | 199.4972 63 | 200 | 0 | 0 | 200 |
| 7.2234 75 | 199.8525 78 | 200 | 0 | 0 | 200 |
| 7.3247 44 | 199.1439 47 | 200 | 0 | 0 | 200 |
| 7.4241 99 | 199.4651 43 | 200 | 0 | 0 | 200 |
| 7.5233 73 | 199. 7542 2 | 200 | 0 | 0 | 200 |
| 7.6246 52 | 199.4009 04 | 200 | 0 | 0 | 200 |
| 7.7239 05 | 199.5615 02 | 200 | 0 | 0 | 200 |
| 7.8241 85 | 199.5615 02 | 200 | 0 | 0 | 200 |
| 7.9241 57 | 199.4330 23 | 200 | 0 | 0 | 200 |
| 8.0241 58 | 199. 6899 8 | 200 | 0 | 0 | 200 |
| 8. 124 04 | 199.4330 23 | 200 | 0 | 0 | 200 |
| 8. 224 14 | 199.9469 37 | 200 | 0 | 0 | 200 |
| 8. 3242 31 | 199.0475 88 | 200 | 0 | 0 | 200 |
| 8. 4235 95 | 199. 1118 27 | 200 | 0 | 0 | 200 |
| 8. 5248 66 | 199. 72 21 | 200 | 0 | 0 | 200 |
| 8. 6234 91 | 199. 3366 65 | 200 | 0 | 0 | 200 |
| 8. 7242 54 | 199.4330 23 | 200 | 0 | 0 | 200 |
| 8. 8233 86 | 199. 2081 86 | 200 | 0 | 0 | 200 |
| 8. 9246 49 | 199. 2081 86 | 200 | 0 | 0 | 200 |
| 9. 0239 11 | 199. 72 21 | 200 | 0 | 0 | 200 |
| 9. 1272 68 | 199. 72 21 | 200 | 0 | 0 | 200 |
| 9. 2745 18 | 199. 8184 59 | 200 | 0 | 0 | 200 |
| 9. 3757 93 | 199. 3366 65 | 200 | 0 | 0 | 200 |
| 9. 4746 01 | 199. 7863 39 | 200 | 0 | 0 | 200 |
| 9. 5778 76 | 199. 2403 06 | 200 | 0 | 0 | 200 |
| 9. 6765 74 | 199. 5293 82 | 200 | 0 | 0 | 200 |
| 9. 7778 66 | 199. 3045 45 | 200 | 0 | 0 | 200 |
| 9. 8775 88 | 199. 1439 47 | 200 | 0 | 0 | 200 |
| 9. 9788 62 | 199. 3366 65 | 200 | 0 | 0 | 200 |
| 10. 0781 23 | 199. 4972 63 | 200 | 0 | 0 | 200 |
| 10. 1783 23 | 199. 0578 61 | 200 | 0 | 0 | 200 |
| 10. 2779 35 | 199. 4651 43 | 200 | 0 | 0 | 200 |
| 10. 3780 41 | 199. 3366 65 | 200 | 0 | 0 | 200 |
| 10. 4779 53 | 199. 6899 8 | 200 | 0 | 0 | 200 |
| 10. 5781 49 | 199. 0578 61 | 200 | 0 | 0 | 200 |
| 10. 6785 86 | 199. 3687 84 | 200 | 0 | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 10.778878 | 199.657862 | 200 | 0 | 0 | 200 |
| 10.878712 | 199.465145 | 200 | 0 | 0 | 200 |
| 10.977916 | 200.236016 | 200 | 0 | 0 | 200 |
| 11.077916 | 207.495246 | 200 | 0 | 0 | 200 |
| 11.177965 | 223.811839 | 200 | 0 | 0 | 200 |
| 11.278 | 243.661771 | 200 | 0 | 0 | 200 |
| 11.37827 | 254.197031 | 200 | 0 | 0 | 200 |
| 11.47826 | 261.937888 | 200 | 0 | 0 | 200 |
| 11.57642 | 268.361836 | 200 | 0 | 0 | 200 |
| 11.67872 | 272.73011 | 200 | 0 | 0 | 200 |
| 11.77972 | 278.158347 | 200 | 0 | 0 | 200 |
| 11.88027 | 284.485936 | 200 | 0 | 0 | 200 |
| 11.97936 | 287.280372 | 200 | 0 | 0 | 200 |
| 12.07962 | 291.648648 | 200 | 0 | 0 | 200 |
| 12.18028 | 292.001945 | 200 | 0 | 0 | 200 |
| 12.28099 | 291.648648 | 200 | 0 | 0 | 200 |
| 12.37955 | 291.423809 | 200 | 0 | 0 | 200 |
| 12.48068 | 291.423809 | 200 | 0 | 0 | 200 |
| 12.58217 | 291.29513 | 200 | 0 | 0 | 200 |
| 12.68102 | 291.455939 | 200 | 0 | 0 | 200 |
| 12.78118 | 291.488049 | 200 | 0 | 0 | 200 |
| 12.88133 | 291.520149 | 200 | 0 | 0 | 200 |
| 12.98123 | 291.520149 | 200 | 0 | 0 | 200 |
| 13.08039 | 291.35957 | 200 | 0 | 0 | 200 |
| 13.18166 | 291.455939 | 200 | 0 | 0 | 200 |
| 13.28193 | 291.455939 | 200 | 0 | 0 | 200 |
| 13.38313 | 291.520149 | 200 | 0 | 0 | 200 |
| 13.48349 | 291.488049 | 200 | 0 | 0 | 200 |
| 13.58267 | 291.520149 | 200 | 0 | 0 | 200 |
| 13.68294 | 291.520149 | 200 | 0 | 0 | 200 |
| 13.78283 | 291.488049 | 200 | 0 | 0 | 200 |
| 13.88239 | 291.455939 | 200 | 0 | 0 | 200 |
| 13.98235 | 291.455939 | 200 | 0 | 0 | 200 |
| 14.08288 | 291.520149 | 200 | 0 | 0 | 200 |
| 14.18389 | 291.455939 | 200 | 0 | 0 | 200 |
| 14.28527 | 291.455939 | 200 | 0 | 0 | 200 |
| 14.38509 | 291.423809 | 200 | 0 | 0 | 200 |
| 14.48503 | 291.520149 | 200 | 0 | 0 | 200 |
| 14.58627 | 291.455939 | 200 | 0 | 0 | 200 |
| 14.6843 | 291.35957 | 200 | 0 | 0 | 200 |
| 14.78462 | 291.455939 | 200 | 0 | 0 | 200 |
| 14.88503 | 291.520149 | 200 | 0 | 0 | 200 |
| 14.98478 | 291.423809 | 200 | 0 | 0 | 200 |
| 15.08629 | 291.520149 | | | | |
| 15.18609 | 291.488049 | | | | |
| 15.28816 | 291.455939 | | | | |
| 15.38662 | 291.391649 | | | | |
| 15.4878 | 291.391649 | | | | |
| 15.58913 | 291.488049 | | | | |

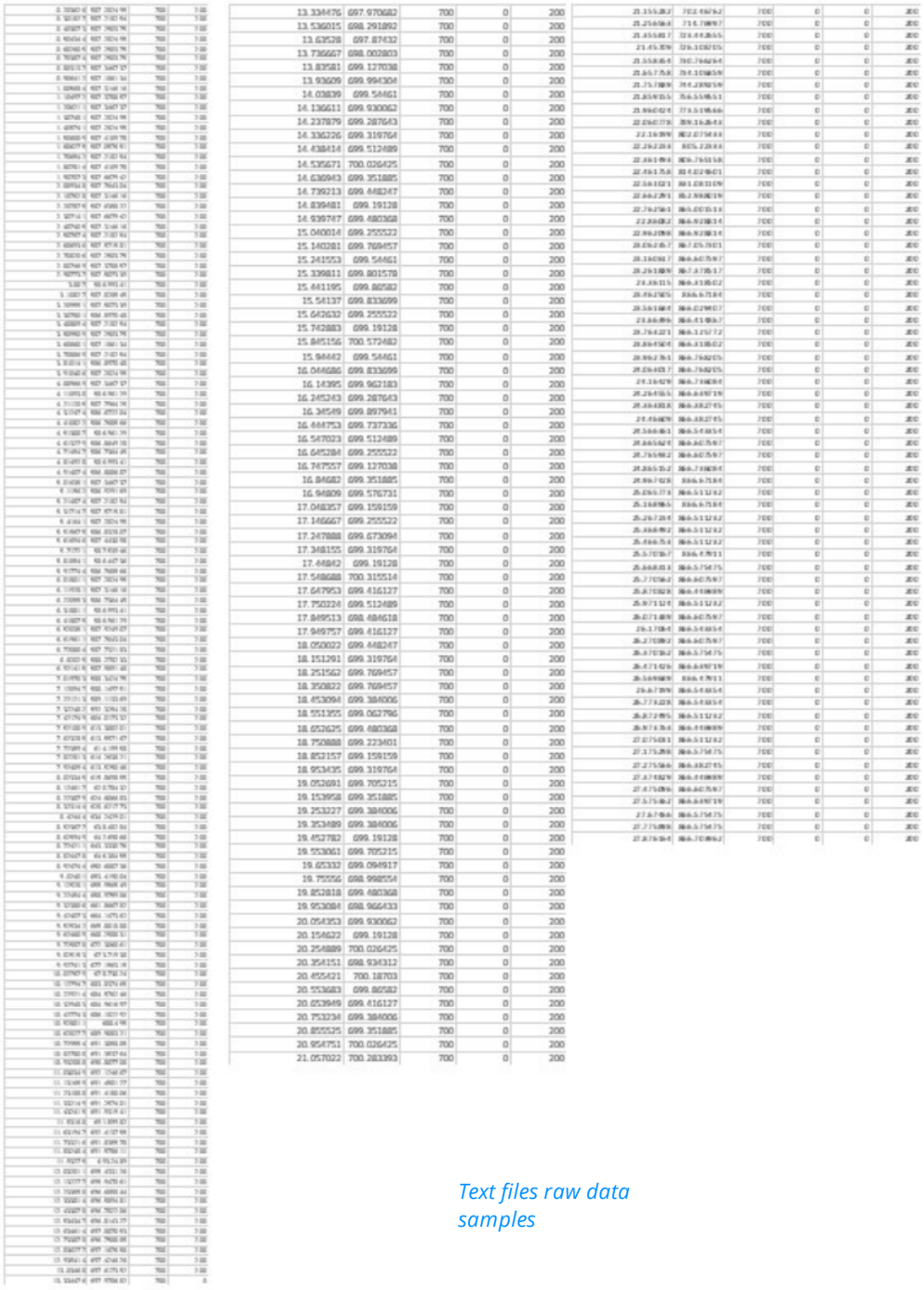
*Appendix Figure 30 - Load Cell 200g Data Log*

Load Cell 500g data log



*Appendix Figure 31 - Load Cell 500g Data Log*

Load Cell 700g data log



| | | | | |
|---|---|---|---|---|
| 13.334476 | 697.970682 | 700 | 0 | 200 |
| 13.536015 | 698.291892 | 700 | 0 | 200 |
| 13.63528 | 697.87432 | 700 | 0 | 200 |
| 13.736667 | 698.002803 | 700 | 0 | 200 |
| 13.83581 | 699.127038 | 700 | 0 | 200 |
| 13.93609 | 699.994304 | 700 | 0 | 200 |
| 14.03839 | 699.54461 | 700 | 0 | 200 |
| 14.136611 | 699.930062 | 700 | 0 | 200 |
| 14.237879 | 699.287643 | 700 | 0 | 200 |
| 14.336226 | 699.319764 | 700 | 0 | 200 |
| 14.438414 | 699.512489 | 700 | 0 | 200 |
| 14.535671 | 700.026425 | 700 | 0 | 200 |
| 14.636943 | 699.351885 | 700 | 0 | 200 |
| 14.739213 | 699.448247 | 700 | 0 | 200 |
| 14.839481 | 699.19128 | 700 | 0 | 200 |
| 14.939747 | 699.480368 | 700 | 0 | 200 |
| 15.040014 | 699.255522 | 700 | 0 | 200 |
| 15.140281 | 699.769457 | 700 | 0 | 200 |
| 15.241553 | 699.54461 | 700 | 0 | 200 |
| 15.339811 | 699.801578 | 700 | 0 | 200 |
| 15.441195 | 699.86582 | 700 | 0 | 200 |
| 15.54137 | 699.833699 | 700 | 0 | 200 |
| 15.642632 | 699.255522 | 700 | 0 | 200 |
| 15.742883 | 699.19128 | 700 | 0 | 200 |
| 15.845156 | 700.572482 | 700 | 0 | 200 |
| 15.94442 | 699.54461 | 700 | 0 | 200 |
| 16.046686 | 699.833699 | 700 | 0 | 200 |
| 16.14395 | 699.962183 | 700 | 0 | 200 |
| 16.245243 | 699.287643 | 700 | 0 | 200 |
| 16.34549 | 699.897941 | 700 | 0 | 200 |
| 16.444753 | 699.737336 | 700 | 0 | 200 |
| 16.547023 | 699.512489 | 700 | 0 | 200 |
| 16.645284 | 699.255522 | 700 | 0 | 200 |
| 16.747557 | 699.127038 | 700 | 0 | 200 |
| 16.84682 | 699.351885 | 700 | 0 | 200 |
| 16.94809 | 699.576731 | 700 | 0 | 200 |
| 17.048357 | 699.159159 | 700 | 0 | 200 |
| 17.146667 | 699.255522 | 700 | 0 | 200 |
| 17.247888 | 699.673094 | 700 | 0 | 200 |
| 17.348155 | 699.319764 | 700 | 0 | 200 |
| 17.44842 | 699.19128 | 700 | 0 | 200 |
| 17.548688 | 700.315514 | 700 | 0 | 200 |
| 17.647953 | 699.416127 | 700 | 0 | 200 |
| 17.750224 | 699.512489 | 700 | 0 | 200 |
| 17.849513 | 698.484618 | 700 | 0 | 200 |
| 17.949757 | 699.416127 | 700 | 0 | 200 |
| 18.050022 | 699.448247 | 700 | 0 | 200 |
| 18.151291 | 699.319764 | 700 | 0 | 200 |
| 18.251562 | 699.769457 | 700 | 0 | 200 |
| 18.350822 | 699.769457 | 700 | 0 | 200 |
| 18.453094 | 699.384006 | 700 | 0 | 200 |
| 18.551355 | 699.062796 | 700 | 0 | 200 |
| 18.652625 | 699.480368 | 700 | 0 | 200 |
| 18.750888 | 699.223401 | 700 | 0 | 200 |
| 18.852157 | 699.159159 | 700 | 0 | 200 |
| 18.953435 | 699.319764 | 700 | 0 | 200 |
| 19.052691 | 699.705215 | 700 | 0 | 200 |
| 19.153958 | 699.351885 | 700 | 0 | 200 |
| 19.253227 | 699.384006 | 700 | 0 | 200 |
| 19.353489 | 699.384006 | 700 | 0 | 200 |
| 19.452782 | 699.19128 | 700 | 0 | 200 |
| 19.553061 | 699.705215 | 700 | 0 | 200 |
| 19.65332 | 699.094917 | 700 | 0 | 200 |
| 19.75556 | 698.998554 | 700 | 0 | 200 |
| 19.852818 | 699.480368 | 700 | 0 | 200 |
| 19.953084 | 698.966433 | 700 | 0 | 200 |
| 20.054353 | 699.930062 | 700 | 0 | 200 |
| 20.154622 | 699.19128 | 700 | 0 | 200 |
| 20.254889 | 700.026425 | 700 | 0 | 200 |
| 20.354151 | 698.934312 | 700 | 0 | 200 |
| 20.455421 | 700.18703 | 700 | 0 | 200 |
| 20.553683 | 699.86582 | 700 | 0 | 200 |
| 20.653949 | 699.416127 | 700 | 0 | 200 |
| 20.753234 | 699.384006 | 700 | 0 | 200 |
| 20.855525 | 699.351885 | 700 | 0 | 200 |
| 20.954751 | 700.026425 | 700 | 0 | 200 |
| 21.057022 | 700.283393 | 700 | 0 | 200 |

*Text files raw data samples*

*Appendix Figure 32 - Load Cell 700g Data Log*

1809797

GRADEMARK REPORT

FINAL GRADE

69/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

**Text Comment.** A good level of detail in the table of contents

PAGE 4

PAGE 5

**Text Comment.** A comprehensive list of figures

**Text Comment.** An extensive appendix with all test requirements

PAGE 6

**Text Comment.** This as an interface with the FPGA

**Text Comment.** Introduction covers most tasks.

PAGE 7

**Text Comment.** Good explanation but needs a supporting image

PAGE 8

**Text Comment.** Good explanations and example of main DAQ function blocks in LabVIEW

PAGE 9

PAGE 10

PAGE 11

**Text Comment.** Good explanations and examples of the comparator

PAGE 12

**Text Comment.** Good explanation with a supporting diagram for quantisation. Good use of comparison table here.

PAGE 13

**Text Comment.** Research demonstrates a sound knowledge with ADC's

PAGE 14

**Text Comment.** Explanation relevant in this portfolio. A good understanding demonstrated here.

PAGE 15

**Text Comment.** Stages involved in the signal conditioning process are identified

PAGE 16

PAGE 17

**Text Comment.** 4 bit adder components have been identified here. .

PAGE 18

PAGE 19

**Text Comment.** A good assumption made here as the full adder will be tested in the 4 bit adder.

PAGE 20

**Text Comment.** Hierarchy and inheritance has been identified which is an essential part of understanding for this design system.

**Text Comment.** A well-structured program with the full adder components clearly shown

PAGE 21

**Text Comment.** IP design is correct

**Text Comment.** Good set of test data

PAGE 22

**Text Comment.** A clear test result using virtual bus method. Test indicates the carry Cout set for sum > 15 for 4 bit.

PAGE 23

**Text Comment.** Good design with FPGA Pmod ports allocated

PAGE 24

**Text Comment.** These settings are identified from the PLD design

**Text Comment.** The 4 bit comparator VHDL was generated from the PLD design. This must be stated.

**Text Comment.** Test indicates correct operation of comparator

PAGE 25

**Text Comment.** The full IP design components hierarchy are identified here.

**Text Comment.** PLD design is correct

PAGE 26

**Text Comment.** Test indicates the sum value is equal to the set value

PAGE 27

**Text Comment.** Test indicates the sum value is greater than set value.

PAGE 28

**Text Comment.** Test indicates the sum value is less than set value.

**Text Comment.** Good use of tables in this section

PAGE 29

**Text Comment.** Test indicates A = B

**Text Comment.** Test indicates A > B.

PAGE 30

**Text Comment.** Test indicates A < B.

**Text Comment.** Good evaluation. The role of the FPGA could have been included.

PAGE 31

**Text Comment.** Introduction covers most reqirements

PAGE 32

**Text Comment.** This should now be high gain

PAGE 33

**Text Comment.** Required gain is idnetified

**Text Comment.** This should be buffer amp gain of 5 and op amp gain of 100

**Text Comment.** A good explanation

**Text Comment.** Principle of unity gain is realised

PAGE 34

**Text Comment.** 10 bit ADC design is correct

PAGE 35

**Text Comment.** 12 bit ADC set in formula block

PAGE 36

**Text Comment.** 12 bit ADC design is correct

PAGE 37

**Text Comment.** 8 bit ADC design with formula block is correct. All the io DAQ function blocks are configured

**Text Comment.** Test for 8 bit ADC is good which proves program configuration

PAGE 38

**Text Comment.** PLD design of 8 bit comparator is correct with static variables identified and set

**Text Comment.** FPGA Pmod ports are identified

PAGE 39

**Text Comment.** Explanation gives an overview but the operation is not explained

**Text Comment.** Good use of table.

**Text Comment.** Formula block 4 needs to be explained in more detail a this block is the key to the data transfer to the FPGA 8 bit comparator

PAGE 40

**Text Comment.** Good image of equipment setup

**Text Comment.** Front panel design is good

PAGE 41

**Text Comment.** Good resultant linear graph which shows that the error is acceptable

PAGE 42

**Text Comment.** Good table of results

**Text Comment.** Good test which shows the max value held steady. If the max raw value is set at 1024 the max value of the system will overshoot.

PAGE 43

**Text Comment.** Good test showing value at half way point.

**Text Comment.** Calc is good for proof

PAGE 44

PAGE 45

**Text Comment.** Good test for 12 bit ADC indicating how LabVIEW handles the ADC arrays by a simple calculation and expansion of array.

**Text Comment.** Good practical test using virtual input signal from virtual P.S.

PAGE 46

**Text Comment.** A good test indicating a small error which is acceptable.

PAGE 47

**Text Comment.** Test is correct for value less than setpoint of 400grms

**Text Comment.** Test is correct for value equal to setpoint of 200grms

PAGE 48

**Text Comment.** Test is correct for value greater than setpoint of 200grms

PAGE 49

PAGE 50

**Text Comment.** Introduction covers the practical task

**Text Comment.** Text file configuration example shows the x axis set and file extension of .lvm

PAGE 51

**Text Comment.** The 'write files' have been implemented into the main program correctly.

**Text Comment.** Functions of file 1 and 2 are explained

**Text Comment.** Array and scaling function for graph is explained

PAGE 52

**Text Comment.** Files 3-5 are explained

**Text Comment.** Timer function is explained

PAGE 53

**Text Comment.** Resultant graph is explained

**Text Comment.** Good set of results for comparator settings

PAGE 54

**Text Comment.** Resultant graph for setpoint @ 500grms is good

**Text Comment.** Good set of results on front panel indicating all 3 conditions

PAGE 55

**Text Comment.** Resultant set point is correct for setpoint @ 700grms

**Text Comment.** Results are correct for all 3 condition on front panel

PAGE 56

**Text Comment.** Good evaluation

PAGE 57

PAGE 58

PAGE 59

**Text Comment.** Good number of relevant sources

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

**Text Comment.** Full testing of amplifier is present

PAGE 69

**Text Comment.** 8 bit ADC formula block configuration

PAGE 70

PAGE 71

**Text Comment.** Text file configurations are demonstrated

PAGE 72

PAGE 73

**Text Comment.** Raw value samples of text file data

PAGE 74

PAGE 75

PAGE 76

**Text Comment.** Text files raw data samples

PAGE 77

PAGE 78

GRADING FORM: CSG-RUBIC

| LOURENS BRITS | **0** |
|---|---|

**RESEARCH-INFORMED LITERATURE**

Research section has covered all requirements in reasonable detail. There are sections where images would have supported the explanations. Good relevant sources used.

**6/10**

**KNOWLEDGE AND UNDERSTANDING**

E1-Design and use of magnitude comparator with digital inputs has been created but explanations are not clear in parts. The 4 bit comparator PLD design is missing. Test methods used are good and followed a logical sequence. 7/10 E2- There are parts missing for the instruments amp and minor errors. The LabVIEW final design for the analogue load cell input is not explained in detail. The 16 bit word operation for the FPGA is not explained. The shift left and add parts are missing and this is the key for data transfer. 6/10 E3-Data logging has ben fully explained and a good knowledge of this section is demonstrated. 710

**20/30**

**ANALYSIS**

E1-Test and results are good . The test for the 4 bit comparator is missing but all other tests are present. 3/5 E2-The results from LabVIEW are correct . There are some minor errors in the explanations of amplifier gains section. 6.5/10 E3-Data logging using LabVIEW methods has ben designed and tested for operational requirements. 7.5/10

**17/25**

**PRACTICAL APPLICATION AND DEP.**

E1-Prototypes for the comparator have been designed using logical methods. PLD design methods used were successfully implemented using the FPGA development board. 3.5/5 E2-Design and build of LabVIEW interface for fpga followed a good development sequence. The dev sections were tested at designated stages to prove design. The data was shifted and added successfully. 3.5/5 E3-Data logging using LabVIEW methods has ben designed and tested for operational requirements. 8/10

**15/20**

**SKILLS FOR PROFESSIONAL PRAC.**

A good structure to the report and logical sequence to the tasks..

**11/15**