Mason Morgan

by Mason Morgan

FILE	13695_MASON_MORGAN_MASON_MORGAN_169043_1629976719.PDF (5.56M)		
TIME SUBMITTED	20-MAY-2021 11:10AM (UTC+0100)	WORD COUNT	8965
SUBMISSION ID	154057544	CHARACTER COUNT	55735

Course:	HND Electrical and Electronic engineering
Module:	SGEE 5007 VICS
Assignment:	NI LabVIEW Magnitude Comparator
Name:	Mason Richard Morgan
University number:	1809802
Date submitted:	20th MAY 2021.
Word count:	4098

Table of Contents

1. Introduction 4
2. Research 4
   a. Digital design methods using VHDL and IP integration 4
   b. NI DAQmx data acquisition used in LabVIEW 6
   c. Digital Magnitude Comparators 11
   d. Analogue to Digital Convertors 15
   e. Load Cells (namely LCKD-1Kg) 16
   f. Instrumentation and Signal Conditioning 18
3. Design requirements 19
   a. Half Adder 19
   b. Full Adder 20
4. Implementation 21
   a. 4-bit Adder and Comparator implementation 21
   b. 8-bit Adder and Comparator implementation 26
5. Testing and Analysis 28
   a. Testing and Analysis of the 4-bit adder 28
   b. Testing and Analysis of the 4-bit adder 34
6. Evaluation 37

References 38

This should be the 8 bit comparator

Content layout and detail is good

Table of Figures

Figure 1: VHDL Code 5
Figure 2: Block diagram with multiple FB's 6
Figure 3: "Add" function block 6
Figure 4: Analogue input voltage function block 7
Figure 5: DAQmx read function block 7
Figure 6: NI DAQmx Clear Icon function block 8
Figure 7: Digital Input function block 8
Figure 8: Digital Outputfunction block 8
Figure 9: Digital Read function block 9
Figure 10: Error out function block 9
Figure 11: Feedback node function block 9
Figure 12: Formula function block 10
Figure 13: 4-Bit Magnitude Comparator 11
Figure 14: 8-bit magnitude comparator block diagram 12
Figure 15: Truth table for an 8-bit magnitude comparator 13

Figure 16: Timing diagram 13
Figure 17: 8-Bit Digital Magnitude Comparator 14
Figure 18: Analogue to digital signals 15
Figure 19: ADC type comparison 16
Figure 20: Load Cells LCKD-1Kg 17
Figure 21: Signal Conditioning System 18
Figure 22: Signal conditioning used in industry 18
Figure 23: Half Adder design development 19
Figure 24: Full Adder Functional Diagram 20
Figure 25: Full Adder Truth Table 20
Figure 26: Full Adder Logic Diagram 20
Figure 27: Block diagram and logic diagram of a 4-bit adder 21
Figure 28: Hierarchy System for a 4-bit adder 22
Figure 29: Inputs and outputs for physical communication 22
Figure 30: 4-bit adder VHDL code part 1 23
Figure 31: 4-bit adder VHDL code part 2 23
Figure 32: 4-bit adder VHDL code part 3 23
Figure 33: Combination of 4-bit adder and magnitude comparator 24
Figure 34: Multisim implemented magnitude comparator and 4-bit adder 25
Figure 35: Hierarchy for an 8-bit adder 26
Figure 36: 8-bit magnitude comparator // multisim 27
Figure 37: Hierarchy for a 4-bit adder 28
Figure 38: Simulation results of a 4-bit adder 29
Figure 39: 4-bit adder testbench coding for simulation part 1 29
Figure 40: 4-bit adder testbench coding part 2 30
Figure 41: Evaluation table (Greater than) 31
Figure 42: Greater than nexys DDR board 31
Figure 43: Calculations for equal to (AEQB) test 32
Figure 44: AEQB implemented onto nexys board 32
Figure 45: ALTB calculations 33
Figure 46: ALTB implemented onto nexys board 33
Figure 47: 8-Bit Functional Testing AEQB 34
Figure 48: 8-bit nexys board for ALTB values 35
Figure 49: 8-bit mag comparator for values of AGTB 36

Good list of figures

1. Introduction

Element 1 is focussing on designing and implementing a combinational system using both the vivado and multisim software's. Once a finished program has been constructed using these software's they will be written to a Nexys 4 DDR FPGA development board.

Using the two software's listed above both a 4-bit adder and a magnitude comparator will be designed and combined together to write onto the board.

Reasonable introduction

2. Research

To complete the upcoming tasks successfully it was useful to carry out research into the relevant topics listed as subtitles below. Once the knowledge has been gained then this can be used moving forwards.

a. Digital design methods using VHDL and IP integration

VHDL is a very used language and is not case sensitive. VHDL stands for Very High-Speed IC description Language. It is a Hardware Description programming language used to design hardware systems such as FPGA and is an alternative to Verilog. VHDL can also be used to design low level systems like gates to high level systems like in Verilog.
A program in VHDL is known as a VHDL model and each VHDL model has two components:

- entity
- architecture

Entity represents the interface specification of the component in the model and the architecture represents the internal structure or implementation of the entity.
A design entity is a basic element in a VHDL model. It can have different levels of abstraction like gate, printed circuit board or the entire system. The different levels of abstraction are classified into:

- Behavioural
- Dataflow
- Structural

In VHDL, a design entity is a module (as in software system) with its inputs and output which are referred as ports.

Waveforms



Logic Operation – OR Gate

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Symbol:




```
VHDL Code:
Library ieee;
use ieee.std_logic_1164.all;

entity or1 is
   port(x,y:in bit ; z:out bit);
end or1;

architecture virat of or1 is
begin
   z<=x or y;
end virat;
```

Figure 1: VHDL Code

(point, 2020)

Figure 1 above shows come VHDL language used along with its outputting Waveform and logic operation.

Intellectual Property integration is most often used in electronic chips. The use of IP integration can be a reusable component. This particular chip can perform up to several high-speed complex functions, whilst being able to retain the ability to configure for other uses.

Good explanation with a good example

An IP example would have helped here

b. NI DAQmx data acquisition used in LabVIEW

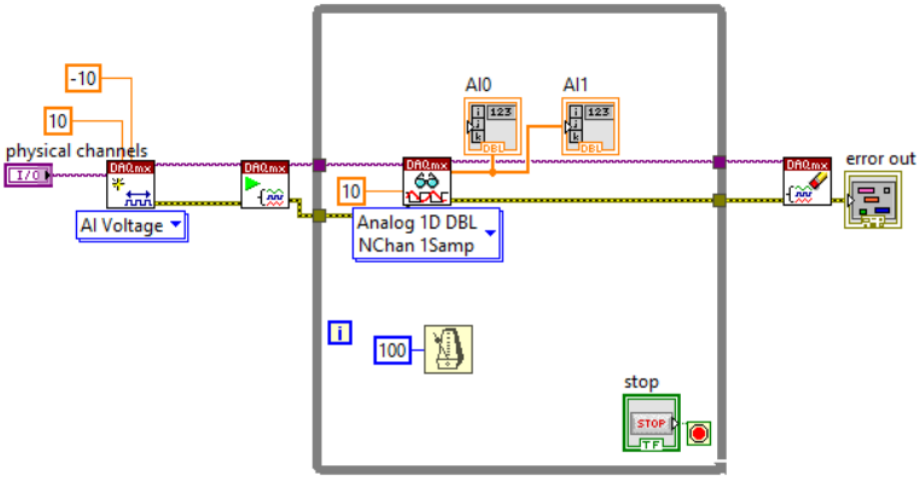


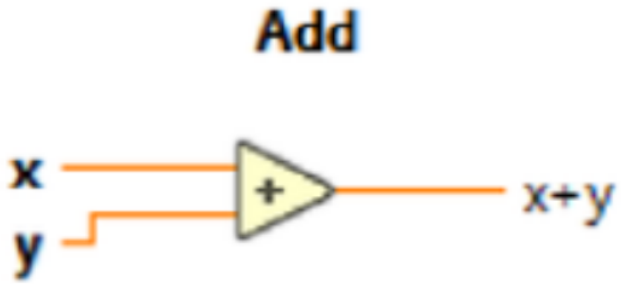
Figure 2: Block diagram with multiple FB's

(Instruments, 2020)

Shown by figure 1 a block diagram holds several different function blocks that can be used.

Figure 2 below shows a unique function block noted as "Add". The purpose of the add function is to take a couple of inputs I, e. x and y and sums them together to produce an outcome in one equation.

Figure 3: "Add" function block



(Instruments, 2020)

Figure 3 shows an image of an “analogue input voltage”. This is used to measure voltage and uses external excitation to scale the voltage.

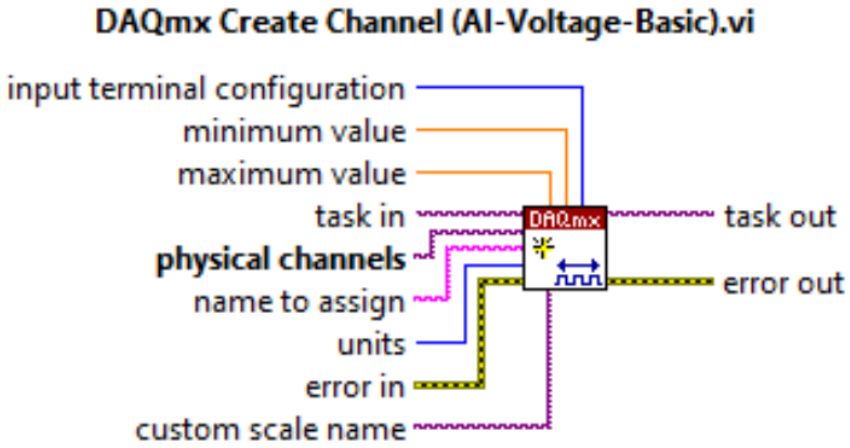
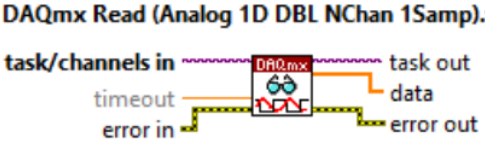


Figure 4: Analogue input voltage function block

(Instruments, 2020)



Reads a single floating-point sample from each channel in a task that contains one or more analog input channels.

Figure 5: DAQmx read function block

(Instruments, 2020)



Clears the task. Before clearing, this VI aborts the task, if necessary, and releases any resources the task reserved. You cannot use a task after you clear it unless you recreate the task.

Figure 6: NI DAQmx Clear Icon function block

(Instruments, 2020)

Digital Input

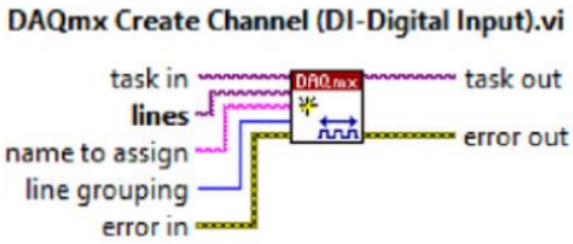


Figure 7: Digital Input function block

(Instruments, 2020)

The digital input measures the digital signal. Digital lines can be separated into multiple channels or combined into single channels.

Digital Output

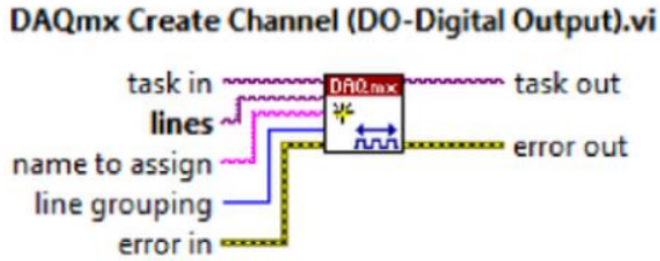


Figure 8: Digital Outputfunction block

(Instruments, 2020)

The function block “Digital Output” is very similar to the input version however it is used for outputs. It can create single or multiple channels.

Digital Read function block shown by figure 8 has the capability to read an array of Boolean values from single digital input channels.

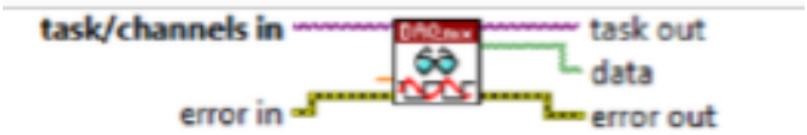


Figure 9: Digital Read function block

(Instruments, 2020)



Figure 10: Error out function block

(Instruments, 2020)

Error out shown by figure 9 is used to provide standard error information within the program

Feedback node stores data from one loop iteration or VI execution to the next. This can be seen in figure 10.



Figure 11: Feedback node function block

(Instruments, 2020)

Good coverage of the DAQ function blocks and the formula block

The formula function block is used to create utilize calculations wherever they may be needed.

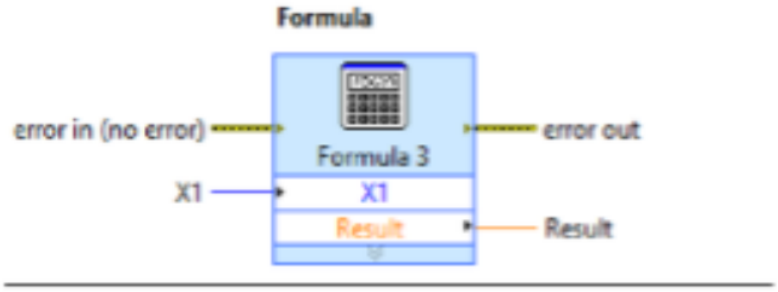


Figure 12: Formula function block

(Instruments, 2020)

c. Digital Magnitude Comparators

Shown by figure 13 below is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. A circuit can be designed to have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition. (Anon., 2020)

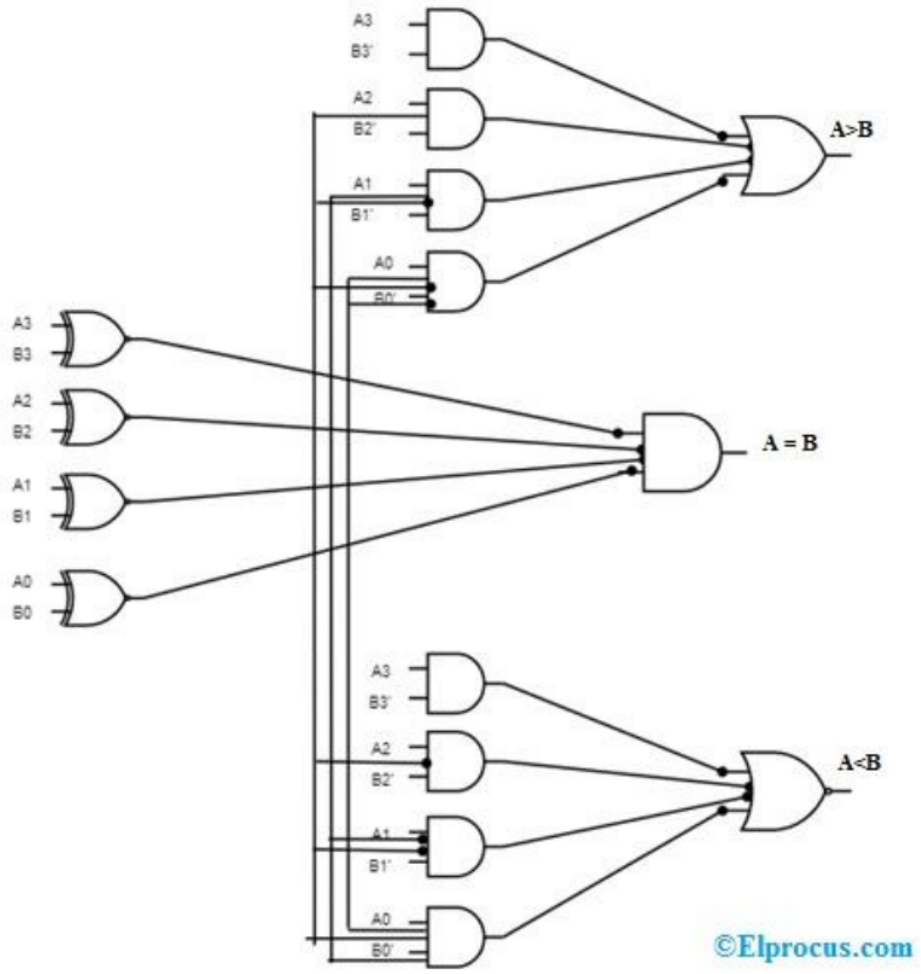


Figure 13: 4-Bit Magnitude Comparator

(Anon., 2019)

A comparator that compares two binary numbers (each number having 4 bits) and produces three outputs based on the relative magnitudes of given binary bits is called a 4-bit magnitude comparator.

The input bits can be termed as **A = A3 A2 A1 A0** and **B = B3 B2 B1 B0**
The output is **A> B** in the cases of
A3 = 1 and **B3 = 0**
A3 = B3 and **A2 = 1, B2 = 0**
A3 = B3 and **A2= B2** and **A1 = 1** and **B1 = 0**
A3 = B3 and **A2= B2** and **A1 = B**1 and A0 = 1 and **B0 = 0**
And **A>B** can be expressed as
A>B = A3B3' + (A3 Ex-Nor B3) A2B2' + (A3 Ex-Nor B3) (A2 Ex-Nor B2) A1B1' + (A3 Ex-Nor B3) (A2 Ex-Nor B2) (A1 Ex-Nor B1) A0B0'
While

A<B = A3'B3 + (A3 Ex-Nor B3) A2'B2 + (A3 Ex-Nor B3) (A2 Ex-Nor B2) A1'B1 + (A3 Ex-Nor B3) (A2 Ex-Nor B2) (A1 Ex-Nor B1) A0'B0
And similarly, A=B can be expressed as

A=B = (A3 Ex-Nor B3) (A2 Ex-Nor B2) (A1 Ex-Nor B1)(A0 Ex-Nor B0)

Mostly, 4-bit comparators are in the form of IC's and the IC 7485 is widely used. Data comparison can be performed by grounding A>B, A<B and A+B inputs to the Vcc terminal. Furthermore, this integrated circuit performs a cascading operation where it helps for cascading multiple comparators.

An 8-bit magnitude comparator compares the two 8-bit values and produces a 1-bit flag as result, which indicates that the first value is either greater than or less than or equal to the second value. The block diagram of a comparator is shown by figure 15.

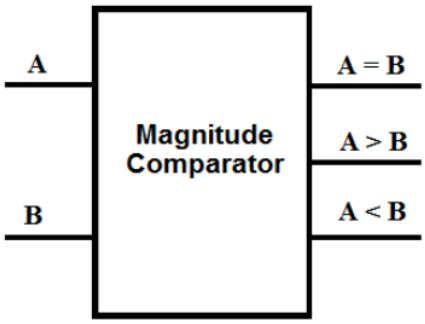


Figure 14: 8-bit magnitude comparator block diagram

(Anon., 2020)

Figure 15 displays a truth table for a magnitude comparator. A and B are the two 8-bit input signals to the comparator which compares the magnitude of these the two signals.

Inputs		Outputs		
A	B	A<B	A=B	A>B
00001100	00001100	0	1	0
00001010	00010001	1	0	0
00001111	00000101	0	0	1
00011000	00011000	0	1	0

Figure 15: Truth table for an 8-bit magnitude comparator

(Anon., 2020)

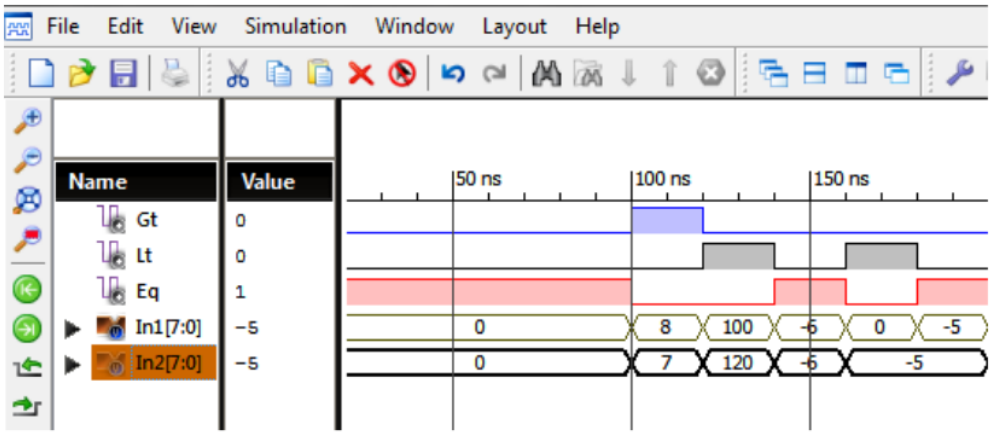
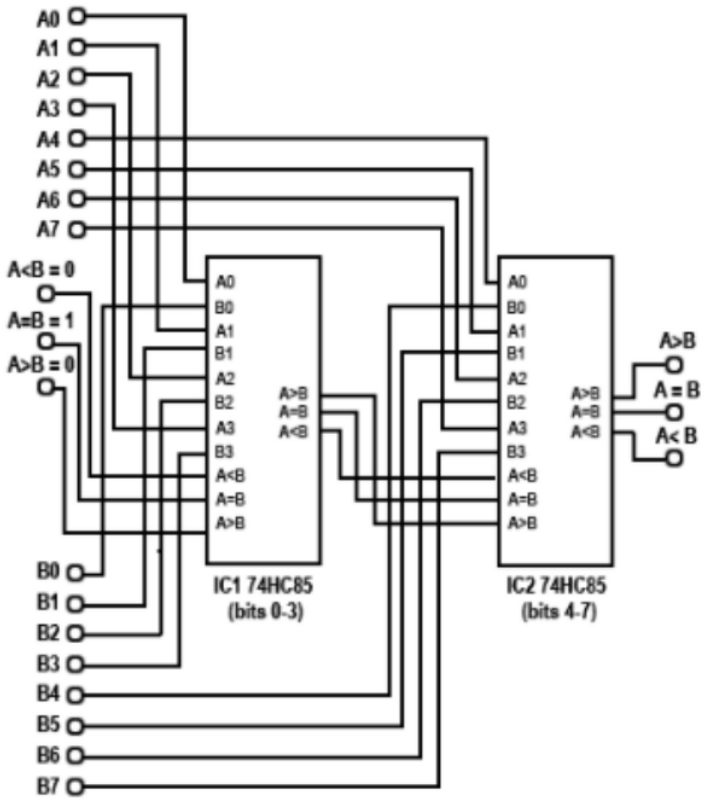


Figure 16: Timing diagram

(Anon., 2020)

Figure 16 displays a timing diagram for a magnitude comparator



Good explanation with relevant sources

Figure 17: 8-Bit Digital Magnitude Comparator

(Coates, 2020)

An 8-bit magnitude comparator as shown by figure -17 compares the two 8-bit values and produce a 1-bit flag as a result which indicates that the first value is either greater than or less than or equal to the second value.
Verilog treats reg data type as unsigned integer number of specified widths. Therefore, the comparison 0 and -5 leads to (0<-5). This is because 2's complement representation of -5 is 251 and 0<251
(Electronics-Tutorial, 2020)

d. Analogue to Digital Convertors

Use capital letter

analog signals are signals that have a continuous sequence with continuous values (there are some cases where it can be finite). These types of signals can come from sound, light, temperature, and motion. Digital signals are represented by a sequence of discrete values where the signal is broken down into sequences that depend on the time series or sampling rate (more on this later). The easiest way to explain this it through a visual! Figure 18 shows a great example of what analog and digital signals look like.

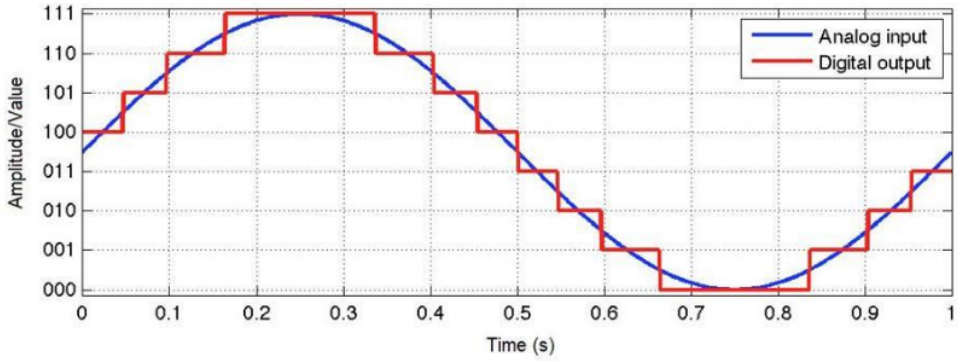


Figure 18: Analogue to digital signals

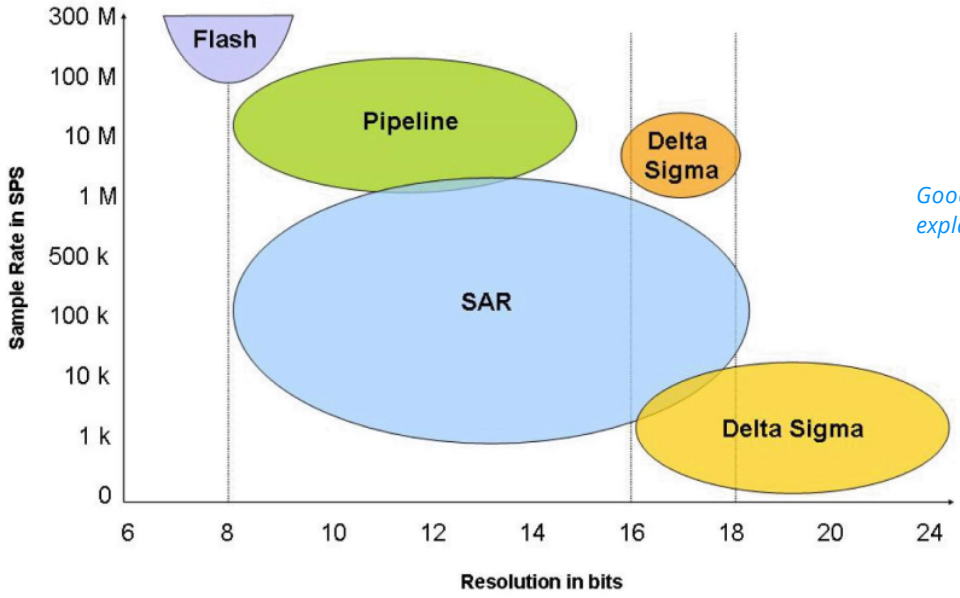
(Gudino, 2016)

Figure 18 shows a continuous anaolg signal turning into a digital signal.

ADCs follow a sequence when converting analog signals to digital. They first sample the signal, then quantify it to determine the resolution of the signal, and finally set binary values and send it to the system to read the digital signal. Two important aspects of the ADC are its sampling rate and resolution. (Anon., 2020)

Many circuit designs have been tried in the development of anaolg-to-digital convertors (ADC's) The field has been narrowed to four basic topologies: flash, successive-approximation (SAR), pipeline, and sigma-delta converters. Each has its strong and weak points.

The optimum device for a particular system depends on the application. More specifically, the optimum device depends on what needs to be done with the data. Applications requiring waveform reconstruction have different requirements from those applications performing calculations or decisions based on a snapshot reading.



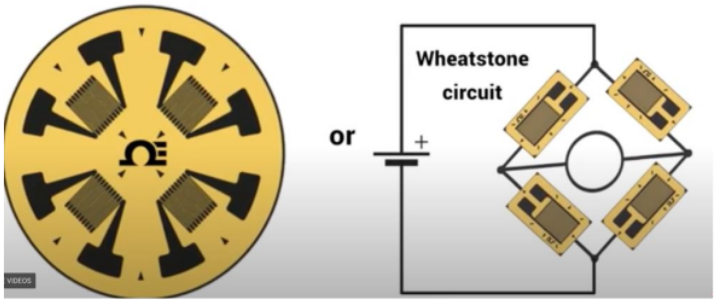
Good effort with the explanation of ADC's

Figure 19: ADC type comparison

(Blair, 2005)

e. Load Cells (namely LCKD-1Kg)

A load cell is a transducer that measures force and outputs this as an electrical signal.
The most common load cells contain strain gauges to perform the sensing. Conversion of force is achieved by measuring physical deformation of the internal strain gauge. Either a purpose designed strain gauge or strain gauges in a Wheatstone formation can be used, see figure 21 attached.



Load cells come in many shapes and sizes, some of these types are:

- S-shaped, for tension and compression. Applications include suspended loads.
- Miniature compression, ideal for restricted spaces
- Tension links, for measuring loads on cables and support rods.
- Bending beam, low-profile construction

LCKD/LCMKD Series compression load cells are designed to measure load ranges from 1 kg to 1000 lb. Rugged all stainless-steel construction and high-performance strain gages ensure superior linearity and stability. Temperature compensation is achieved through a miniature circuit board in the cable. These cells are designed to be mounted on a smooth, flat surface. (OMEGA, 2020)



Load cell is explained

Figure 20: Load Cells LCKD-1Kg
(OMEGA, 2020)

f. Instrumentation and Signal Conditioning

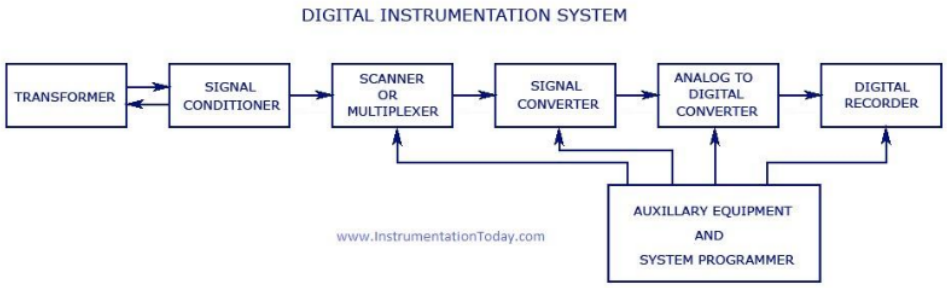


Figure 21: Signal Conditioning System

Good information on signal conditioning

(John, 2020)

Signal conditioning is a process of data acquisition, and an instrument called a signal conditioner is used to perform this process. This instrument converts one type of electrical or mechanical signal (input-signal) into another (output-signal). The purpose is to amplify and convert this signal into an easy to read and compatible form for data-acquisition or machine-control.

A signal conditioner helps to provide precise measurements, which are essential for accurate data acquisition and machine-control. These instruments can perform an additional number of different functions.

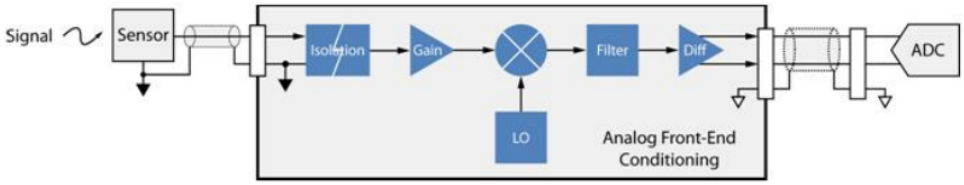


Figure 22: Signal conditioning used in industry

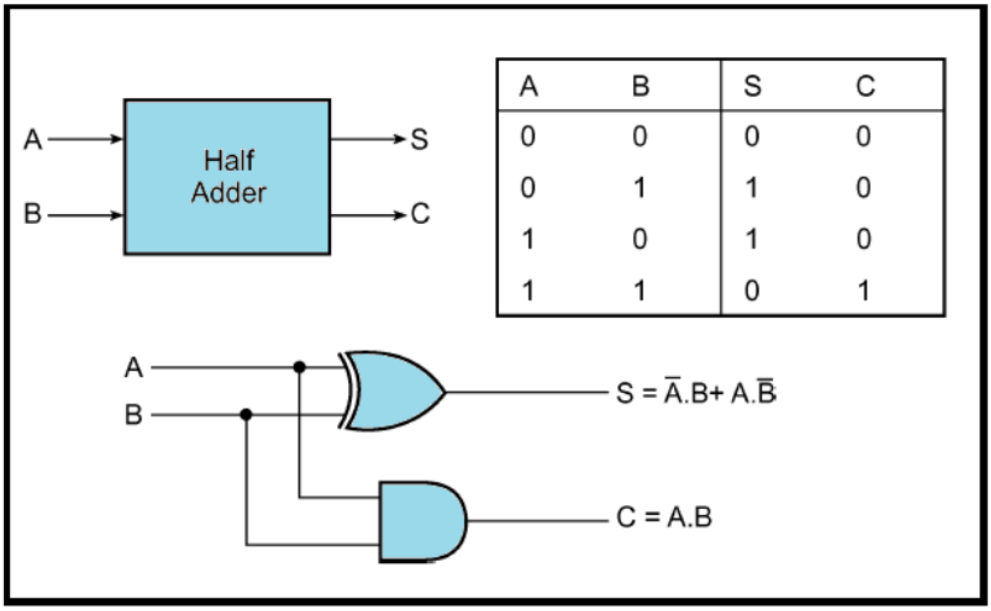
(Nutaq, 2020)

3. Design requirements

a. Half Adder

The design requirements for this stage of the task required us to build a Half Adder. The Vivado software will be used to design a combinational logic system using both VHDL and IP integration.

This starting stage is required to in time get to a more developed result being a 4-bit adder. Before progressing with the construction of the Half Adder, the system must be designed and both a truth table and logic diagram must be developed. The below image (Figure 23) shows a function block for a half adder along with a truth table and a logic diagram.



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 23: Half Adder design development

(HUB, 2020)

b. Full Adder

Once a Half Adder has been created it is then duplicated to create the full adder. The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs.
The first two inputs are A and B, and the third input is an input carry as C-IN. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next. (Focus, 2020)

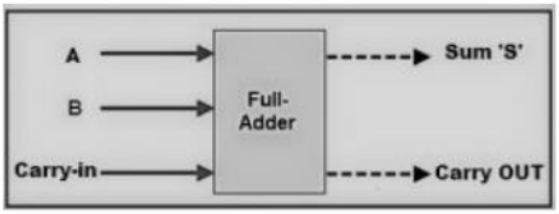


Figure 24: Full Adder Functional Diagram

(Focus, 2020)

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 25: Full Adder Truth Table

(Focus, 2020)

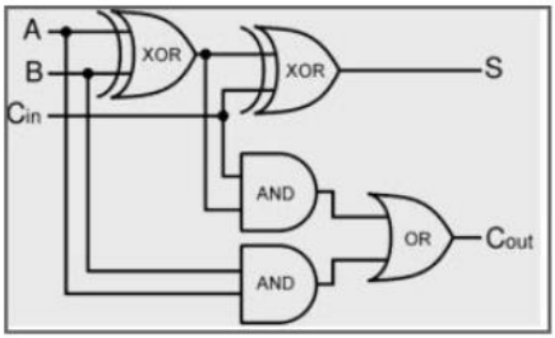


Figure 26: Full Adder Logic Diagram

(Focus, 2020)

4. Implementation

a. 4-bit Adder and Comparator implementation

To construct the 4-bit adder we needed to make sure that a full adder had been complete. To make a 4-bit Adder 4 separate Full Adders needed to be combined. A finished result is shown by figure 27.

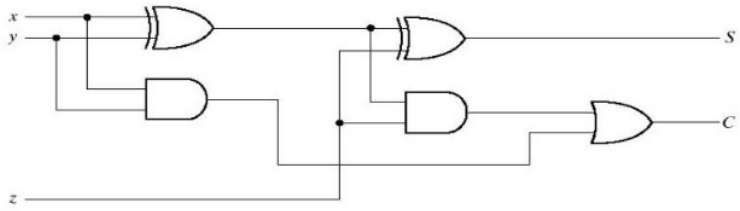


Fig. 4-8 Implementation of Full Adder with Two Half Adders and an OR Gate

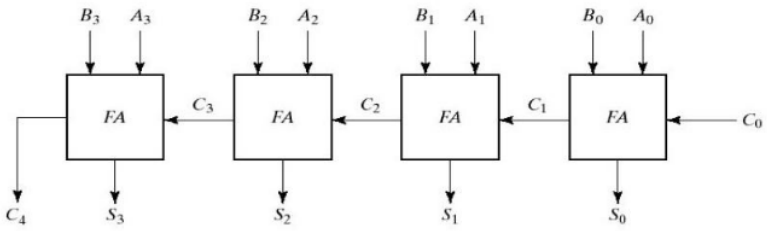


Fig. 4-9 4-Bit Adder

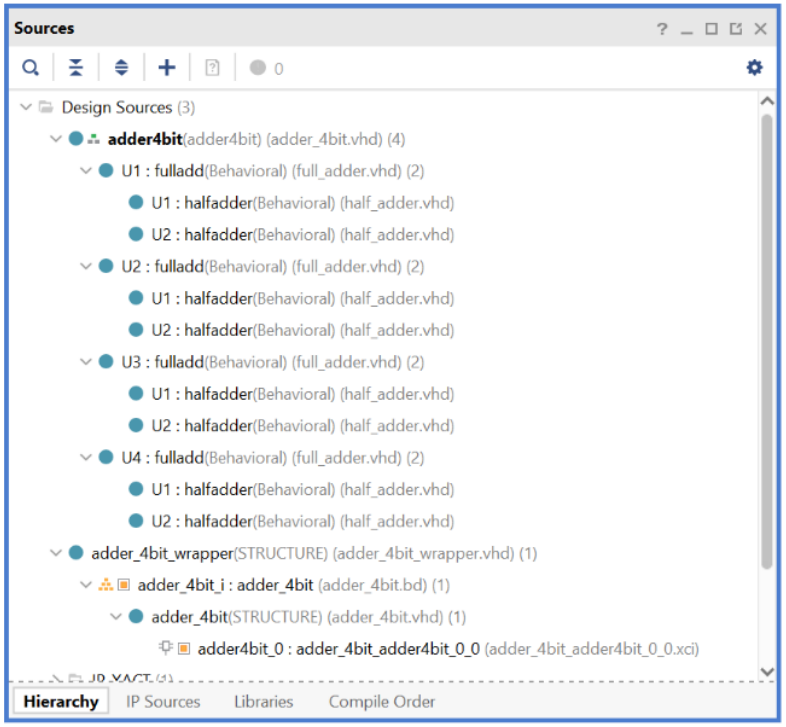
Figure 27: Block diagram and logic diagram of a 4-bit adder

(Serve, 2020)

The purpose of the half adder and full adder is explained

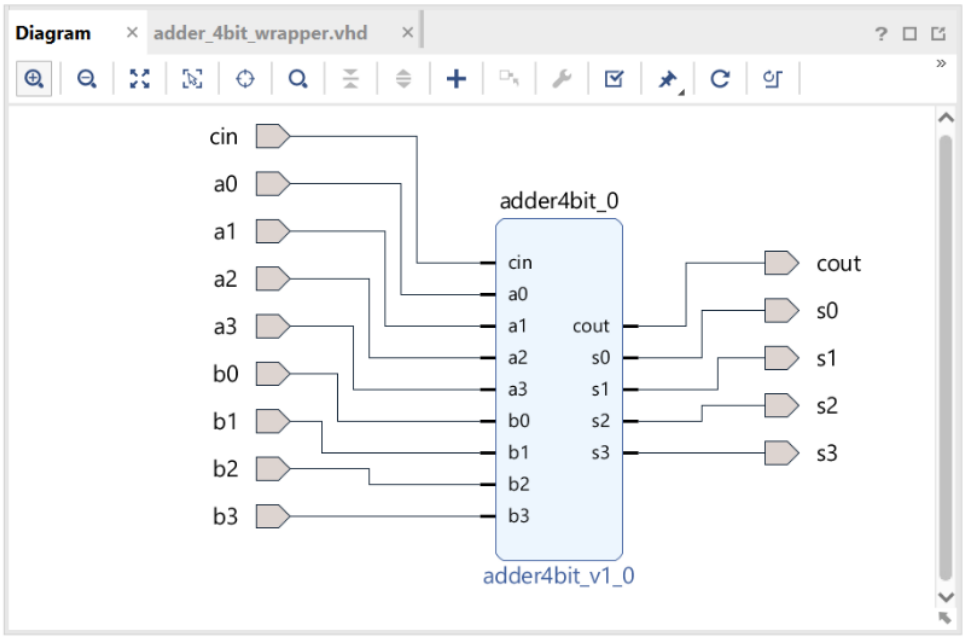
From the Vivado software it will use a hierarchy system to arrange the full adders and half adders to provide the required outputs of a 4-bit adder. See Figure 28 below for the hierarchy system used for this application.
In figure 29 a screenshot has been captured of a few inputs and outputs being regulated through a microchip used on Vivado, this Is later used to download to a NEXYS DDR 4 board.

Hierarchy and in here



Hierarchy of 4 bit adder is identified

Figure 28: Hierarchy System for a 4-bit adder



IP design is good

Figure 29: Inputs and outputs for physical communication

The particular code that had to be written can be seen by figure 30. This is where the VHDL language has been used to design the 4-bit adder.

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity adder4bit is
4  port(
5  cin : in STD_LOGIC;
6  a0 : in STD_LOGIC;
7  a1 : in STD_LOGIC;
8  a2 : in STD_LOGIC;
9  a3 : in STD_LOGIC;
10 b0 : in STD_LOGIC;
11 b1 : in STD_LOGIC;
12 b2 : in STD_LOGIC;
13 b3 : in STD_LOGIC;
14 cout : out STD_LOGIC;
15 s0 : out STD_LOGIC;
16 s1 : out STD_LOGIC;
17 s2 : out STD_LOGIC;
18 s3 : out STD_LOGIC
19 );
20 end adder4bit;
21 architecture adder4bit of adder4bit is
22 component fulladd
23 port (
24 a : in STD_LOGIC;
25 b : in STD_LOGIC;
26 cin : in STD_LOGIC;
27 cout : out STD_LOGIC;
28 s : out STD_LOGIC
29 );
30 end component;
31
32 signal c1 : STD_LOGIC;
33 signal c2 : STD_LOGIC;
34 signal c3 : STD_LOGIC;
35 begin
36 U1 : fulladd
37 port map(
38 a => a0, b => b0, cin => cin, cout => c1,
39 s => s0);
40 U2 : fulladd
41 port map(
42 a => a1, b => b1, cin => c1, cout => c2,
43 s => s1);
44 U3 : fulladd
45 port map(
46 a => a2, b => b2, cin => c2, cout => c3,
47 s => s2);
48 U4 : fulladd
49 port map(
50 a => a3, b => b3, cin => c3, cout => cout,
51 s => s3);
52
53 end adder4bit;
```

Figure 30: 4-bit adder VHDL code part 1

Figure 31: 4-bit adder VHDL code part 2

4 bit adder IP and code have been created and the designs are in good order.

Figure 32: 4-bit adder VHDL code part 3

As displayed by figures 30, 31, 32 above all the VHDL code has been written. The first stage of the code (lines 1 – 20) show the entity detailing what each port will be located as.

It was also important to create a 4-bit magnitude comparator. Shown by figure 35 below an input/output of the comparator is created by combining the 4-bit adder, with outputs 'AGTB', 'AEQB' and 'ALTB' corresponding to an LED display that will illuminate on the NEXYS 4 DDR board. The outputs that have been set up and displayed previously in figure 29 are used as links to combine to inputs into the magnitude comparator. An image of this is displayed by figure 33.

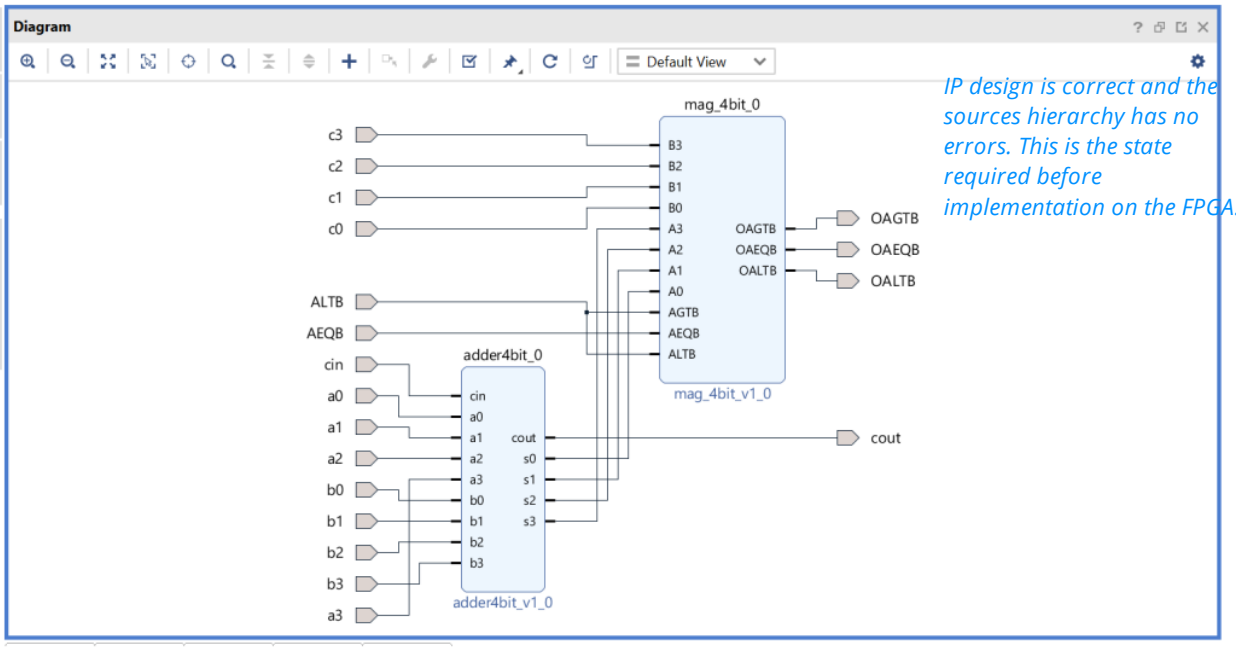


Figure 33: Combination of 4-bit adder and magnitude comparator

This circuit could also be implemented on MultiSim. See figure 34 below.

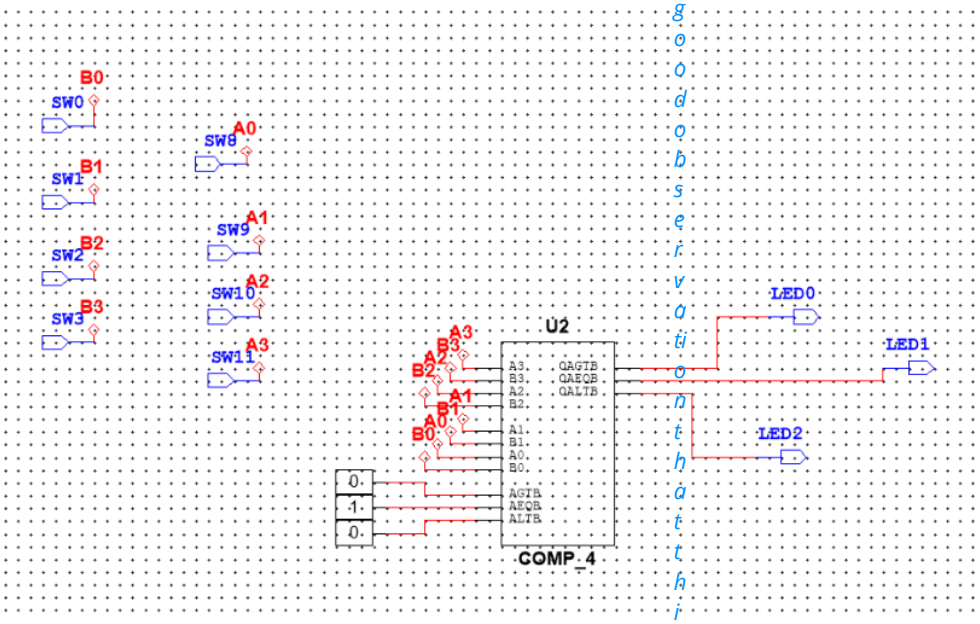


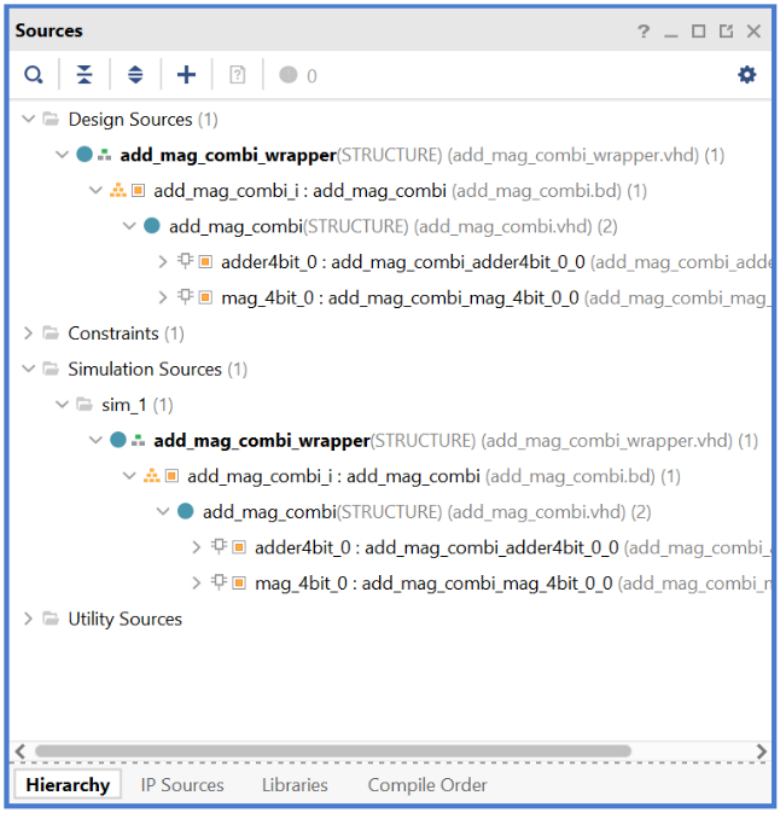
Figure 34: Multisim implemented magnitude comparator and 4-bit adder

We used the Multisim software to perform an operational test. This then enabled us to determine if the circuit for performing properly by manually forcing different inputs on and off.

A good observation that is another alternative method for lesscom

b. 8-bit Adder and Comparator implementation

The 8-bit magnitude comparator was then implemented and tested using VHDL with FPGA Nexys 4 DDR development board.
The 8-bit was constructed by combining two 4-bit adders and comparators. This could only be done once the 4-bit adder and comparator had been tested first.
The hierarchy was then implemented packaging the program.



This is in the wrong place

Figure 35: Hierarchy for an 8-bit adder

Again, to perform a functional test to ensure the circuit worked correctly it was re-designed on multisim. This allowed us to force the input values on and off. Expander inputs are set with a constant digital high or low to allow for the 'compare' function of the design to work.

Static conditions for PLD design are identified for the test data

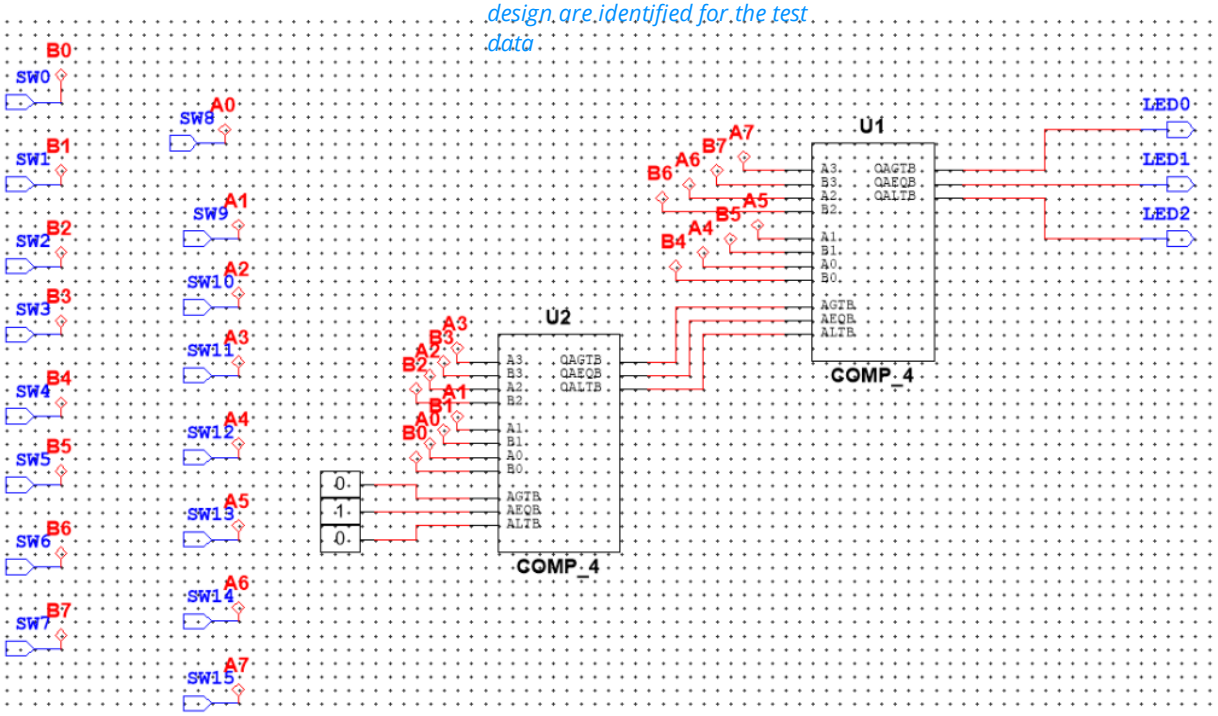


Figure 36: 8-bit magnitude comparator // multisim

As shown by figure 36, AGTB and ALTB are set as 0 meaning they are digital low, and AEQB is let at value '1' meaning it is set as a digital high.

5. Testing and Analysis

At this stage, all sim and practical results have been implemented and recorded. On completion of each design stage, each version of the adders was subsequently tested to verify the correct design and functionality.

a. Testing and Analysis of the 4-bit adder

Captured by figure 37 below is the complete hierarchy of the 4-bit adder. Once the hierarchy had been developed the simulation stage could be developed, please see figure 38.

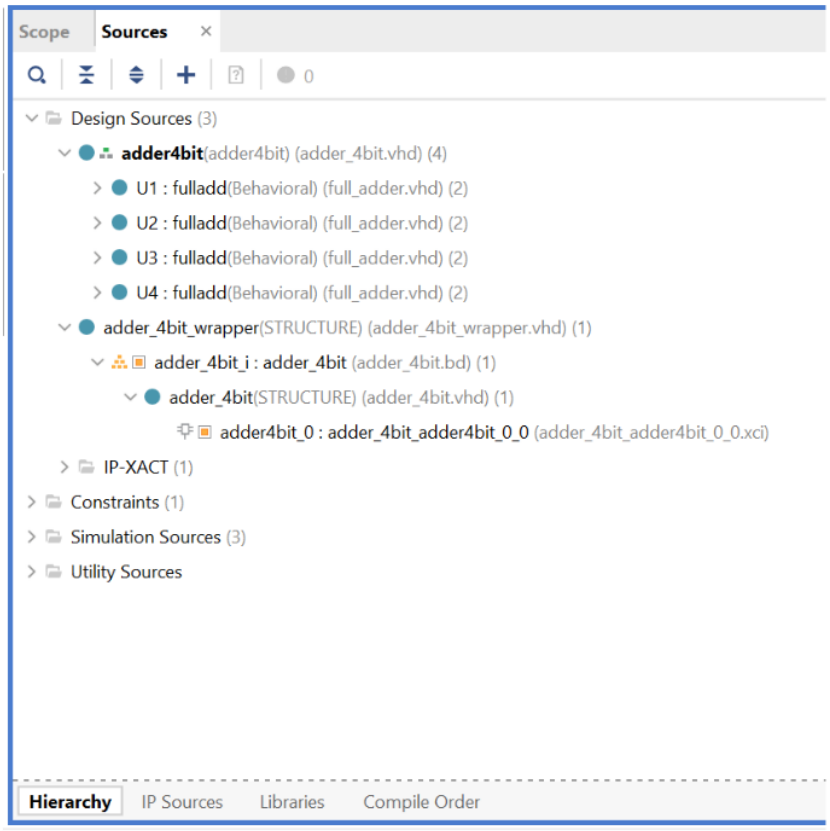


Figure 37: Hierarchy for a 4-bit adder

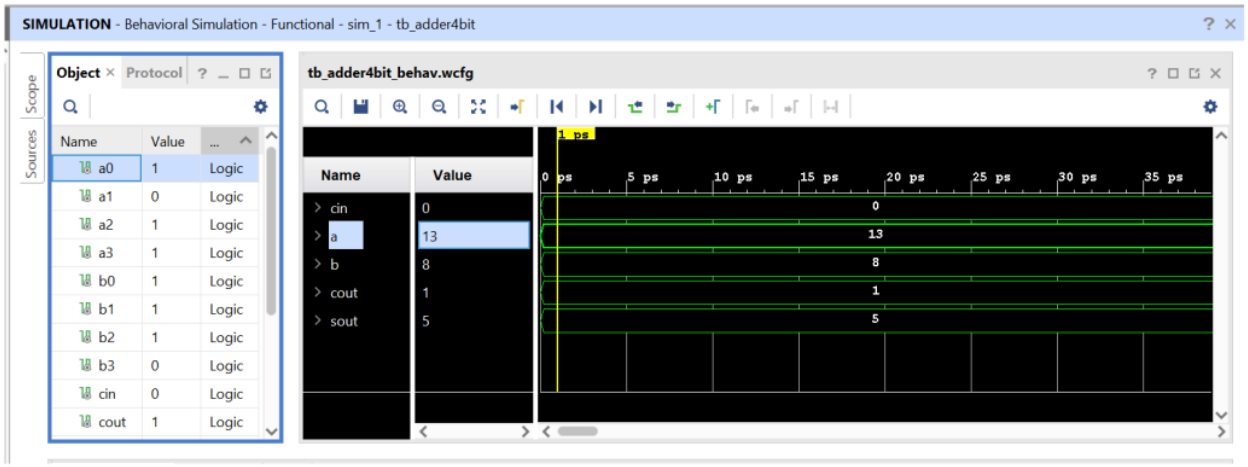


Figure 38: Simulation results of a 4-bit adder

Test is correct but should be for more than one value

Figure 37 shows the testing and simulation of the 4-bit adder using the vivado software. This stage required for all inputs to be grouped and placed into a 'virtual bus'.

```
58
59      stimuli : process
60      begin
61          -- EDIT Adapt initialization as needed
62          cin <= '0';
63          a0 <= '1';
64          a1 <= '0';
65          a2 <= '1';
66          a3 <= '1';
67          b0 <= '0';
68          b1 <= '0';
69          b2 <= '0';
70          b3 <= '1';
71  wait for 100 ns;
72          cin <= '0';
73          a0 <= '1';
74          a1 <= '1';
75          a2 <= '1';
76          a3 <= '0';
77          b0 <= '1';
78          b1 <= '0';
79          b2 <= '1';
80          b3 <= '0';
```

Test data in test bench is correct

Figure 39: 4-bit adder testbench coding for simulation part 1

```
81   wait for 100 ns;
82           cin <= '0';
83           a0 <= '1';
84           a1 <= '1';
85           a2 <= '0';
86           a3 <= '1';
87           b0 <= '0';
88           b1 <= '0';
89           b2 <= '0';
90           b3 <= '1';
91   wait for 100 ns;
92           cin <= '0';
93           a0 <= '1';
94           a1 <= '0';
95           a2 <= '1';
96           a3 <= '1';
97           b0 <= '1';
98           b1 <= '1';
99           b2 <= '1';
100          b3 <= '0';
101  wait for 100 ns;
102          -- EDIT Add stimuli here
103
```

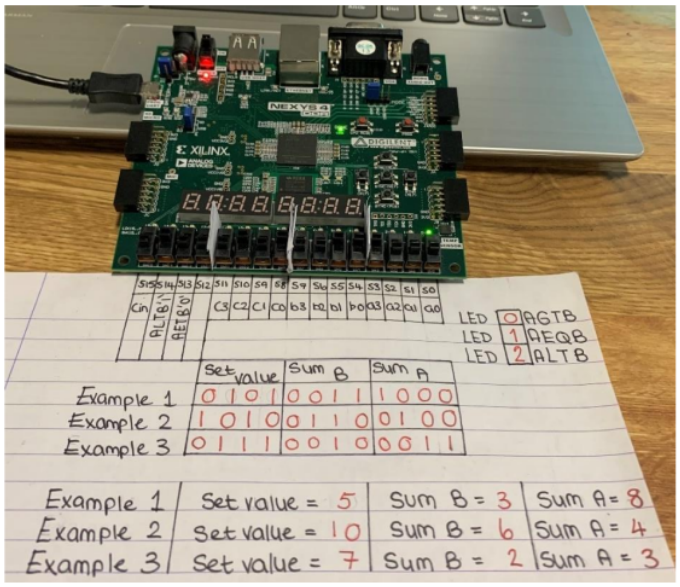
This test data is not tested

Figure 40: 4-bit adder testbench coding part 2

Figure 38 and 39 show the values of all the inputs at different intervals. Once these values were written they could then be seen in the simulation.

After the coding had been complete and the simulation had been grouped and was successful the block diagram created earlier could now be downloaded onto the NEXYS 4 DDR board. When the same circuit was made on the multisim software the values could be recorded. Now the values could be recorded physically through the interactive board.

Figure 40 below shows how the test was implemented and what action was taking to record the data needed.



The test must have a heading.

Figure 41: Evaluation table (Greater than)

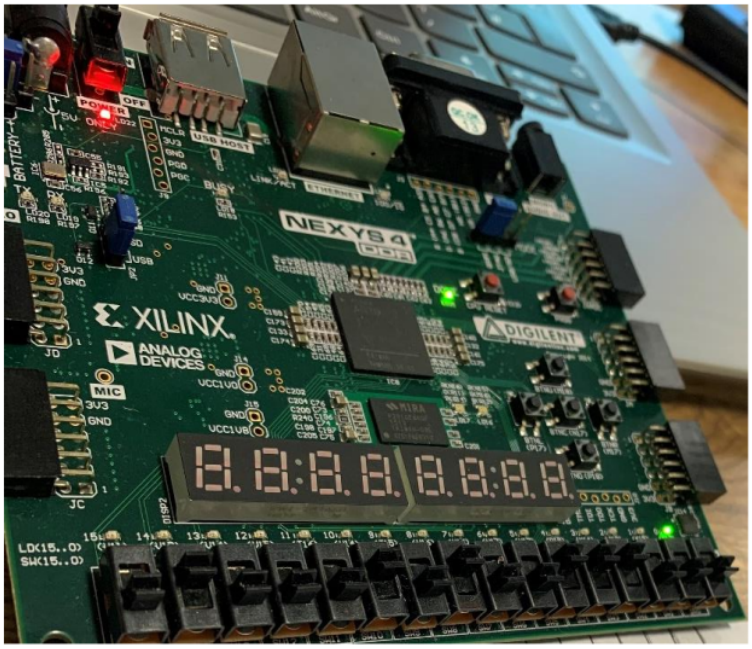


Figure 42: Greater than nexys DDR board

Figure 40 and 41 display the physical test conducted once the program had been downloaded to the nexys ddr board. To keep track of what values needed to be met a table was created to compare calculated values to practical values.
Through writing the expected values on paper the same value to be reached on the board, if the correct LED corresponded to the calculated value on paper then the calculations and program were correct.

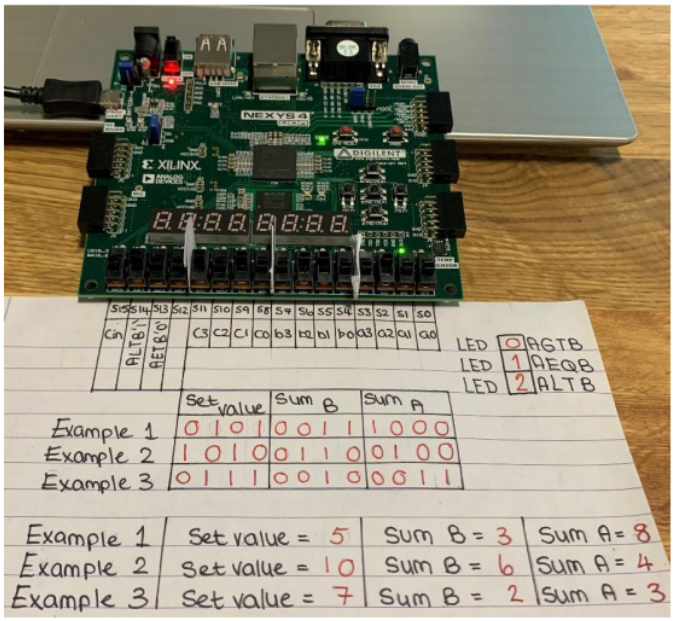


Figure 43: Calculations for equal to (AEQB) test

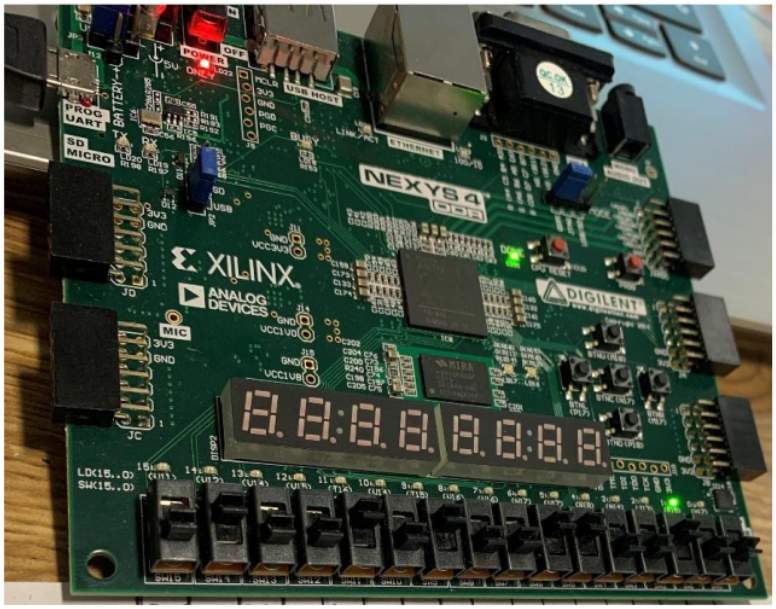


Figure 44: AEQB implemented onto nexys board

The test was conducted again as shown by figure 42 and 43. This time the calculations were made for an equal to value which have been implemented on the nexys board for clarification. The toggle switches on the board were used to derive a final value.

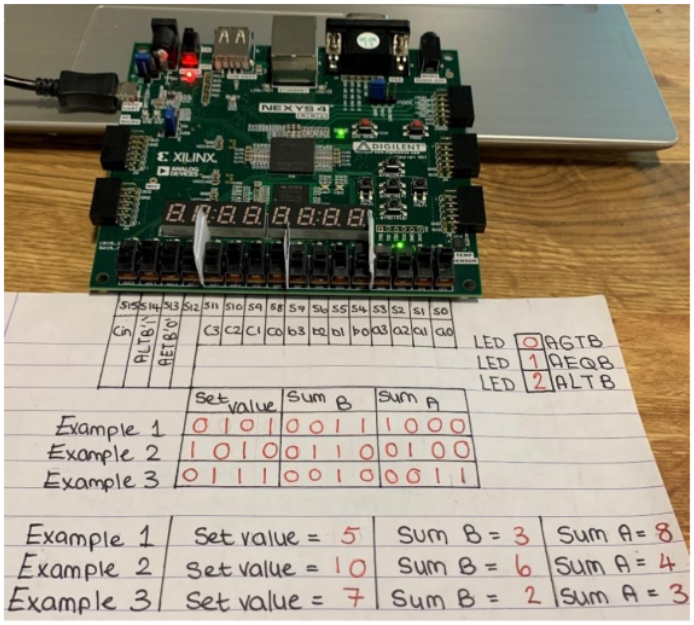


Figure 45: ALTB calculations

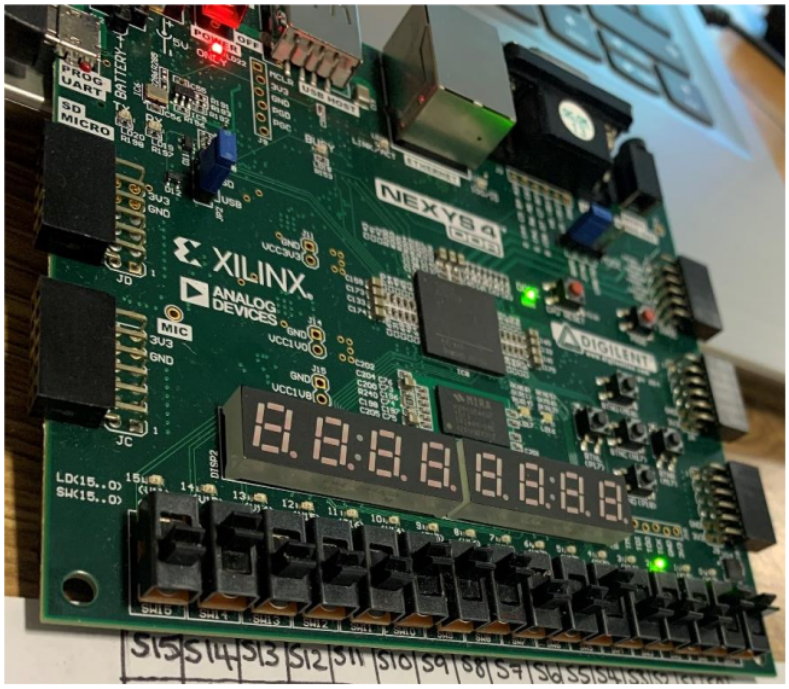
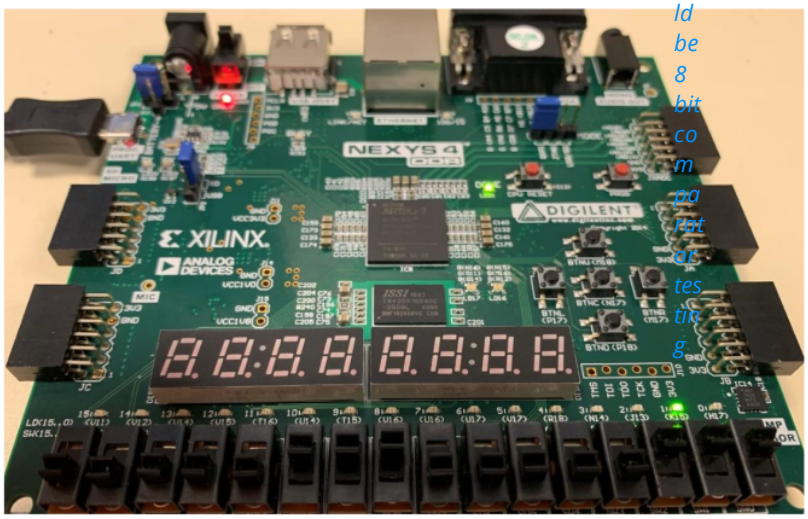


Figure 46: ALTB implemented onto nexys board

Calculations can be shown by figure 44 for a 'more than' value. Figure 45 then implements the calculated value onto the nexys ddr board shown by an LED.

b. Testing and Analysis of the 4-bit adder

This should be 8 bit compartor testing



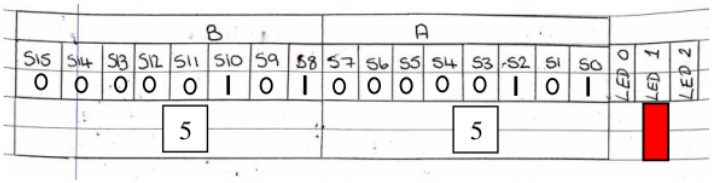
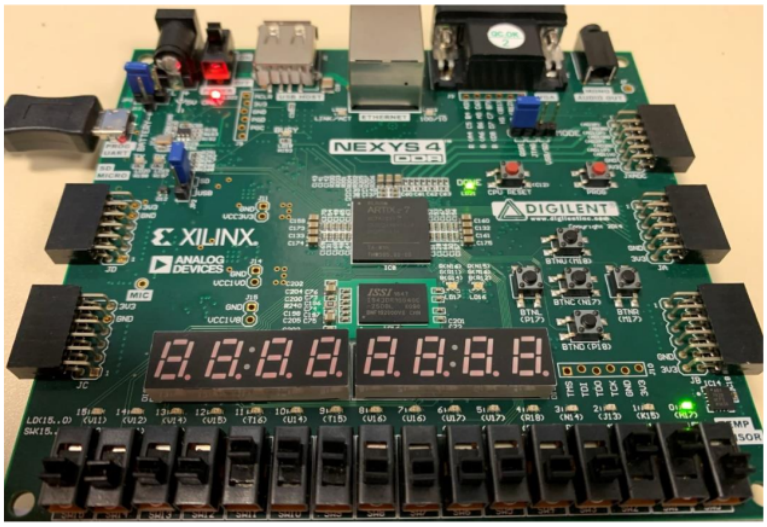


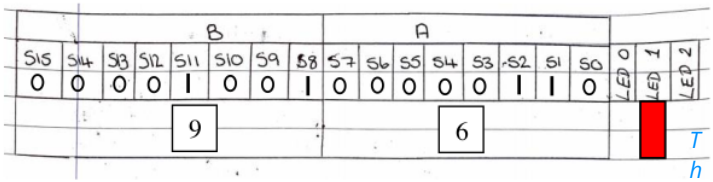
Figure 47: 8-Bit Functional Testing AEQB

Figure 47 shows both tested and calculated values for values equal to or AEQB. A final value of 5 has been used for both A and B.



The calculated values show a value of A at 9 and a value of B at 6.

Figure 48: 8-bit nexys board for ALTB values



This is incorrect: LED0 should be

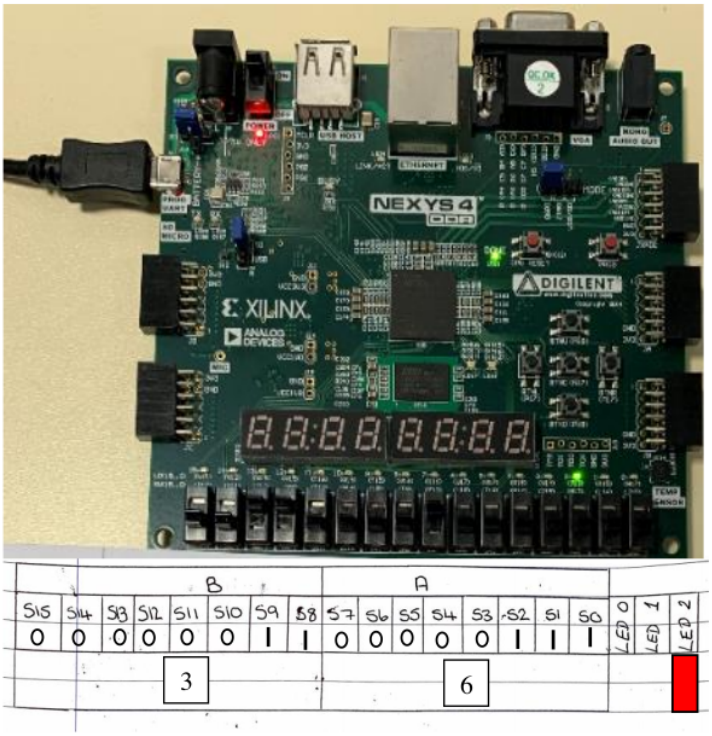


Figure 49: 8-bit mag comparator for values of AGTB

Figure 49 displays calculated and tested values for values of A which are greater than B.

6. Evaluation

Through use of both the vivado and multisim software a 4-bit magnitude comparator and an 8-bit magnitude comparator have been successfully implemented. The circuits were both tested on multisim and downloaded to the nexys 4 DDR board. By doing this both sets of results could be evaluated and compared for accuracy. The prototype systems designed, built, and tested show true results.

To reach the stage of both the 4-bit and 8-bit circuits the early stages had to be completed first, this included the half adder and full adder designs.

The half adder was implemented using VHDL language and then simulated in the vivado software. The research came very useful as it already gave an insight into how it was constructed. Through the construction of the half adder the full adder could then be made. This was done by combining two half adders to make a full adder. Again, using the VHDL language the full adder was constructed and simulated.

The 4-bit adder worked very well when we were able to combine all 4 full adders together. Once the 4-bit adder had been generated it could then be simulated within the Vivado program. On success of this we could then implement this system on the multisim program and conduct tests. Once successful bitstream was used to download the constructed program onto the nexys 4 DDR board where more tested were conducted.
The same process was used for the 8-bit adder where two 4-bit adders were combined and again the multisim circuit was built and the vivado program communicated to the nexys 4 DDR board where we could conduct more tests.

This should be 8 bit comparator and two 4 bit comparators

References

Analogue_Devices, 2020. *www.analog.com*. [Online]
Available at: https://www.analog.com/en/education/education-library/articles/high-perf-electrocardiogram-signal-conditioning.html#:~:text=An%20ECG%20measures%20the%20voltage,50%20Hz%2F60%20Hz%20interference.
[Accessed 24 11 2020].

Anon., 2019. *What is a Digital Comparator and Magnitude Comparator*. [Online]
Available at: https://www.elprocus.com/digital-comparator-and-magnitude-comparator/
[Accessed December 2020].

Anon., 2020. *8-bit magnitude comparator*. [Online]
Available at: https://esrd2014.blogspot.com/p/8-bit-magnitude-comparator.html
[Accessed 10 December 2020].

Anon., 2020. *How ADC's work*. [Online]
Available at: https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters
[Accessed December 2020].

Anon., 2020. *Magnitude Comparator in Digital Logic*. [Online]
Available at: https://www.geeksforgeeks.org/magnitude-comparator-in-digital-logic/
[Accessed 05 December 2020].

Anon., 2020. *Magnitude Comparator in Digital Logic*. [Online]
Available at: https://www.geeksforgeeks.org/magnitude-comparator-in-digital-logic/
[Accessed December 2020].

Arar, S., 2019. *www.allaboutcircuits.com*. [Online]
Available at: https://www.allaboutcircuits.com/technical-articles/understanding-amplitude-quantization-error-analog-digital-converters/
[Accessed 25 11 2020].

Arar, S., 2019. *www.allaboutcircuits.com*. [Online]
Available at: https://www.allaboutcircuits.com/technical-articles/learn-about-sar-adcs-architecture-applications-support-circuitry/
[Accessed 25 11 2020].

Aspencore, 2020. *www.edn.com*. [Online]
Available at: https://www.edn.com/conditioning-techniques-for-real-world-sensors/
[Accessed 24 11 2020].

Aspencore, 2020. *www.electronics-tutorials.ws*. [Online]
Available at: https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html
[Accessed 23 11 2020].

Blair, J., 2005. *www.semanticscholar.org*. [Online]
Available at: https://www.semanticscholar.org/paper/ADC-parameters-and-characteristics-Rapuano-Daponte/952379e8f821e84cca562688fa71ad20cccaf376
[Accessed 25 11 2020].

Coates, E., 2020. *www.learnabout-electronics.org*. [Online]
Available at: https://learnabout-electronics.org/Digital/dig43.php
[Accessed 22 11 2020].

Electronics-Tutorial, 2020. *www.electronics-tutorials.ws*. [Online]
Available at: https://www.electronics-tutorials.ws/combination/comb_8.html
[Accessed 22 11 2020].

Focus, E. P., 2020. *Electronics Project Focus*. [Online]
Available at: https://www.elprocus.com/half-adder-and-full-adder/#:~:text=Difference%20between%20Half%20Adder%20and%20Full%20Adder%20,are%20thr%20...%20%202%20more%20rows%20
[Accessed 3 12 2020].
Gudino, M., 2016. *www.davidswiston.blogspot.com*. [Online]
Available at: https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters
[Accessed 28 11 2020].
HUB, E., 2020. *ELECTRONICS HUB,* s.l.: https://www.electronicshub.org/binary-adder-and-subtractor/.
IEEE, 2004. *www.ieeexplore.ieee.org*. [Online]
Available at: https://ieeexplore.ieee.org/document/1434582
[Accessed 25 11 2020].
Instruments, N., 2020. *DAQmx icons,* s.l.: National Instruments.
John, 2020. *Instrumentation Today*. [Online]
Available at: http://www.instrumentationtoday.com/instrumentation-system/2011/07/
[Accessed 11 December 2020].
Kester, W., 2005. *www.analog.com*. [Online]
Available at: https://www.analog.com/en/analog-dialogue/articles/the-right-adc-architecture.html
[Accessed 25 11 2020].
Nutaq, 2020. *Nutaq*. [Online]
Available at: http://www.nutaq.com/blog/analog-digital-—-part-5-signal-conditioning
[Accessed December 2020].
OMEGA, 2020. *Engineering 360*. [Online]
Available at: https://datasheets.globalspec.com/ds/10/OMEGAEngineering/B00081CC-8698-4663-954A-63250E724ADE
[Accessed December 2020].
OMEGA, 2020. *OMEGA*. [Online]
Available at: https://www.omega.co.uk/pptst/LCKD.html
[Accessed December 2020].
point, T., 2020. *Tutorials point*. [Online]
Available at: https://www.tutorialspoint.com/vlsi_design/vlsi_design_vhdl_introduction.htm
[Accessed 20 12 2020].
Serve, S., 2020. *Hardware Description Language*. [Online]
Available at: https://www.slideserve.com/efrat/hardware-description-language-introduction-powerpoint-ppt-presentation
[Accessed 04 12 2020].
Spiegel, J. V. d., 2001. *www.seas.upenn.edu*. [Online]
Available at: https://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
[Accessed 24 11 2020].
Xie, S., 2013. *www.analog.com*. [Online]
Available at: https://www.analog.com/en/analog-dialogue/articles/successive-approximation-adcs.html
[Accessed 25 11 2020].

A good number of sources

Course:	HND Electrical and Electronic engineering
Module:	SGEE 5007 VICS
Assignment:	NI LabVIEW Magnitude Comparator
Name:	Mason Richard Morgan
University number:	1809802
Date submitted:	18th MAY 2021.
Word count:	3195

This should indicate that this section is element 2

Contents

1. Introduction 43
2. Load cells and signal conditioning 45
   2.1. Unity gain of instrument amplifier 45
   2.2. Instrument amplifier with bridge for load cell sensor 46
   2.3. Practical connections to the Elvis II DAQmx board 49
3. Prototypes 50
   3.1. 10-bit resolution ADC LabVIEW 50
   3.2. 12-bit resolution ADC LabVIEW 53

3.3. 8-bit ADC 55
4. Digital Magnitude Comparator 59
4.1. Nexys 4 DDR FPGA Development board 59
4.2. Magnitude Comparator Setup 60
4.3. LabVIEW 61
4.4. Test Results 64
5. Evaluation 66
6. Appendix 67
6.1. Load cells and signal conditioning 67

Table of Figures

Figure 1: Block diagram of system 43
Figure 2: Unity Gain 45
Figure 3: Ranges of Load Cells 46
Figure 4: Schematic diagram for a compression load cell 46
Figure 5: Load Cell graph derived from table 1. 48
Figure 6: Practical setup 49
Figure 7: 10-bit digital output at a value of 0 50
Figure 8: 10-bit digital output at a value of 2.5 51
Figure 9: 10-bit digital output at a value of 5 52
Figure 10: Formula block equation 52
Figure 11: 12-bit digital output at a value of 2.5 53
Figure 12: Formula block for a 12-bit resolution ADC 53
Figure 13: 12-bit digital output at a value of 5 54
Figure 14: 8-bit resolution ADC at 2.5 55
Figure 15: Formula for an 8-bit ADC 55
Figure 16: 8-bit ADC with a value of 5 56
Figure 17: 8-bit ADC with a value of 1 56
Figure 18: Elvis board relating to a value of 57
Figure 19: Nexys 4 DDR electrical connections 59
Figure 20: Practical setup of Nexys board and Elvis II board 60
Figure 21: Block diagram program for magnitude comparator 61
Figure 22: Formula block no. 1 62
Figure 23: Formula block no. 2 62
Figure 24: Formula block no. 3 63
Figure 25: Formula block no. 4 63
Figure 26: Less Than 64
Figure 27: Equal to 65
Figure 28: More than 65
Figure 29: percentage between 55-45 67

Table of Tables

Table 1: Data table for altering resistance outputting voltage. 47
Table 2: Boolean equivalent value for an analogue value of 2.5 51
Table 3: 12-bit analogue value of 2.5 53

Table 4: 12-bit analogue value of 5 54
Table 5: 8-bit resolution ADC with a value of 2.5 55
Table 6: Boolean value for figure 16 57
Table 7: Boolean value for figure 17 58

Good use of tables

1. Introduction

During this stage of the report the LabVIEW program will be used for a great deal of the topics. The NI Elvis II will be used to construct the 8-bit magnitude comparator which was constructed within element 1 to provide a suitable overview of the capabilities on the NI Elvis II board.
The construction of the system has been displayed below by figure 1.

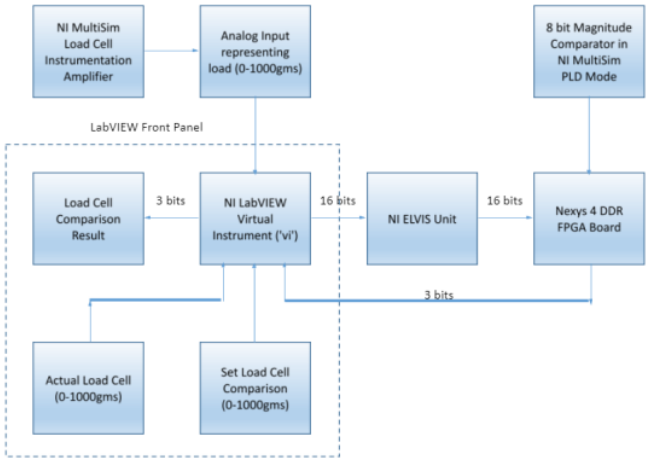


Figure 50: Block diagram of system

During this stage of the task an investigation will also be carried out into the effect of altering the weight upon a load cell. To list the activities involved in this, they can be broken down as:

- The unity gain of the instrument amplifier which will involve the CCT and an explanation of values in the cct.

Further work into this report will also include Instrument amplifier with bridge for load cell sensor which can be broken down as:

- Covering load cells data

Tests will later be carried out on the instrument amp to get the linear line response graph.

- Separate data will be compiled and merged into the data graph for simplicity.

The next stage of the task involves Prototypes. The following prototypes will be built using the LabVIEW software and the below circuits will be constructed:

- 10 and 12bit ADC prototypes
- 8-bit ADC with virtual analogue input.

Information will then be collected for the Digital Magnitude Comparator which will include:

- Images from the Nexys 4 DDR FPGA development board

- Images of the Magnitude Comparator setup
- Screenshots of Front panel view and block design within LabVIEW program
- Explanation of the formula blocks
- Greater than set value screenshot.
- Less than set value screenshot.
- Equal to set value screenshot.

Good introduction

2. Load cells and signal conditioning

2.1. Unity gain of instrument amplifier

Figure 2 below presents an instrument amplifier with unity gain. The instrument amplifier should allow the input to be equal to the output, as far as level is concerned.

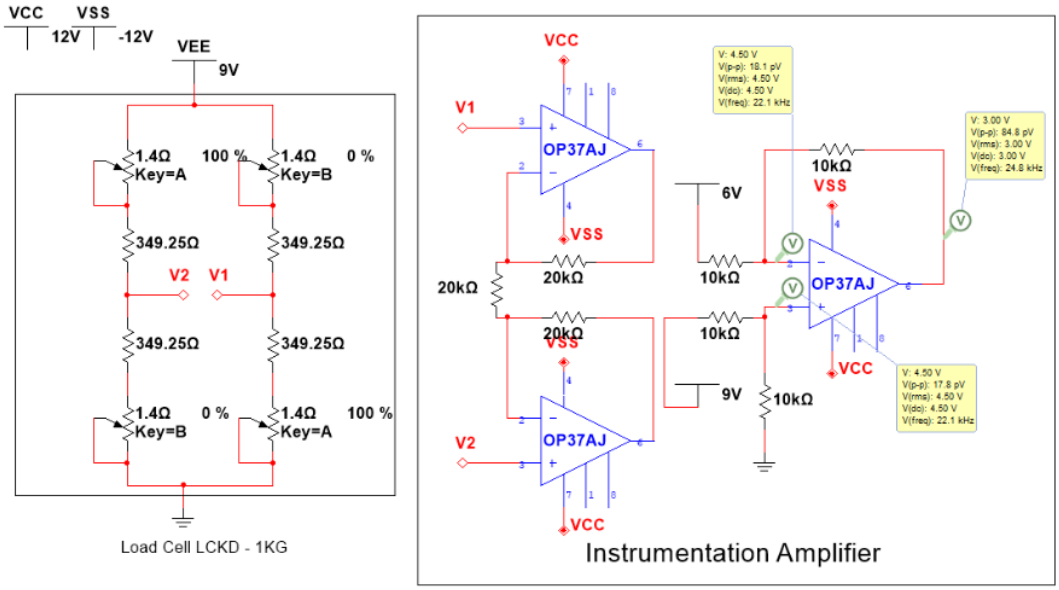
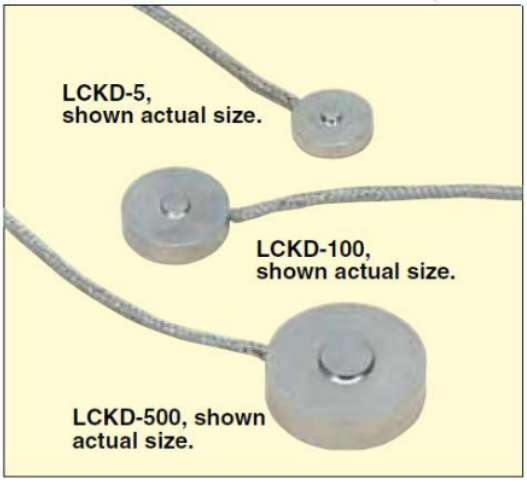


Figure 51: Unity Gain

Unity gain to show operation of op amp

2.2. Instrument amplifier with bridge for load cell sensor

Figure 2 shows a range of Load Cells. These load cells are small in size however are very powerful. These particular load cells are defined as OMEGA's LCKD series sub-miniture



compression load cells.

Figure 52: Ranges of Load Cells

These load cells are capable of measuring load ranges between 1kg – 1000lb. Ensuring that the load cells remain intact and free from damage once a heavy load is applied they are constructed from rugged stainles steel and high performance strain gases.
Typically the load cells are best applied and mounted to a flat, smooth surface.

Figure 53: Schematic diagram for a compression load cell

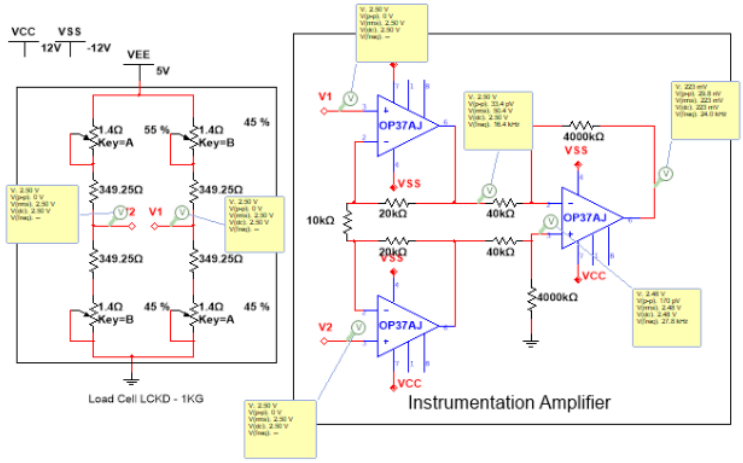


Figure 4 shows the electrical schematic diagram for a Compression Load Cell. The following data can be taken from the Load Cell, these being:
OUTPUT: 2mV / V nominal

LCKD – 1KG
EXCITATION: 5Vdc, 7Vdc maximum
UNSTRESSED RESISTANCE 350ohms

To check the gain of the amplifier:
Gain = (1 + (2*40/20)) * (4000/40) = 500
Buffer gain = (1 + 2 * (40/20)
Diff amp gain = (1000 / 10)
The buffer gain * diff amp gain = 5 * 100 = 500

The 10mV range should be referred to here which would justify the gain of 500.

Gain of instrument amp is established

Data table for altering resistance outputting voltage		
Resistance %	Grams	Voltage (V)
55 - 45	100	0.223
60 - 40	200	0.973
65 - 35	300	1.47
70 – 30	400	1.97
75 - 25	500	2.47
80 – 20	600	2.97
85 - 15	700	3.47
90 – 10	800	3.97
95 – 5	900	4.47
100 – 0	1000	4.97

Table 1: Data table for altering resistance outputting voltage.

Good table of results from practical test of load sim with instrument amp

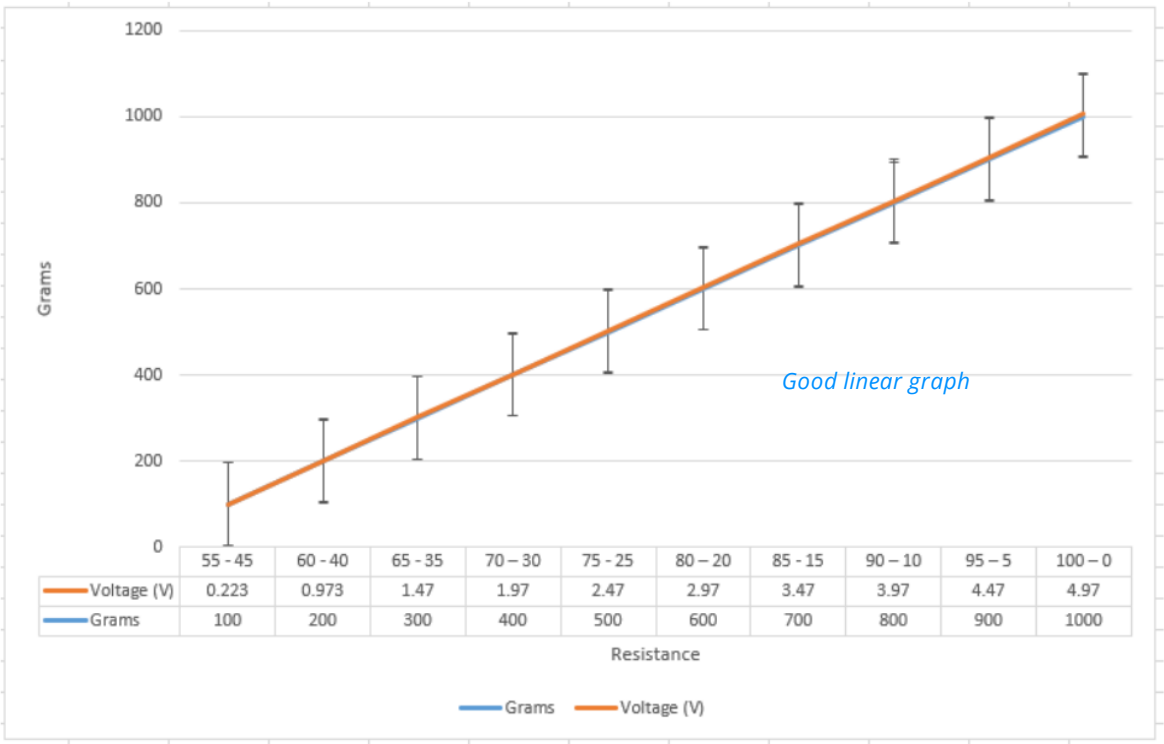


Figure 54: Load Cell graph derived from table 1.

The different values for the Vrms can be seen above in figure 6 however the circuit constructions have been placed in the appendix and referenced as "6.1. Load cells and signal conditioning".

2.3. Practical connections to the Elvis II DAQmx board

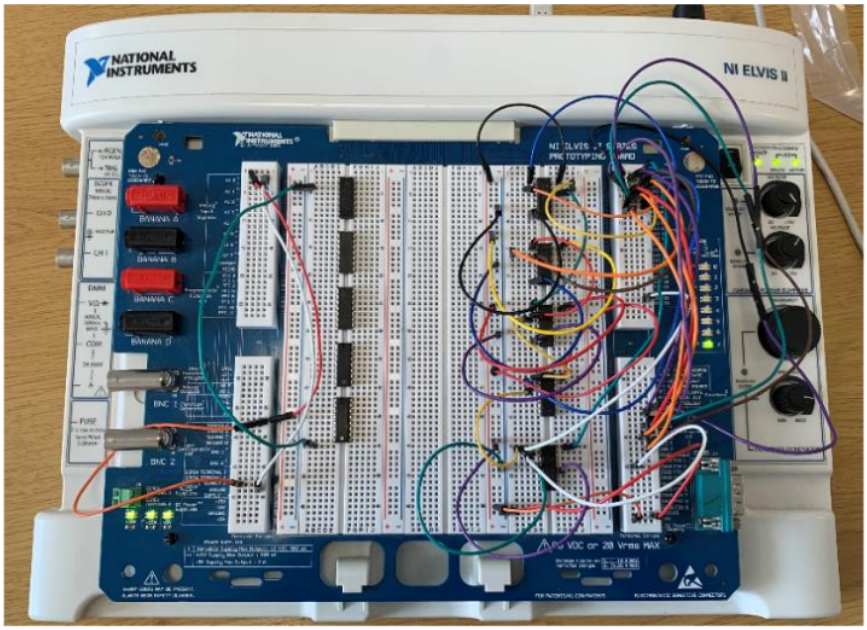


Figure 55: Practical setup

Labview analysis with Magnitude Comparator
Test 1: A = B on comparator
Set load = 567 grms
Load = [A in(V) / 5] * 1000 = 566g
Digital value = [566 / 1000] * 256 = 144
144 = 10010001

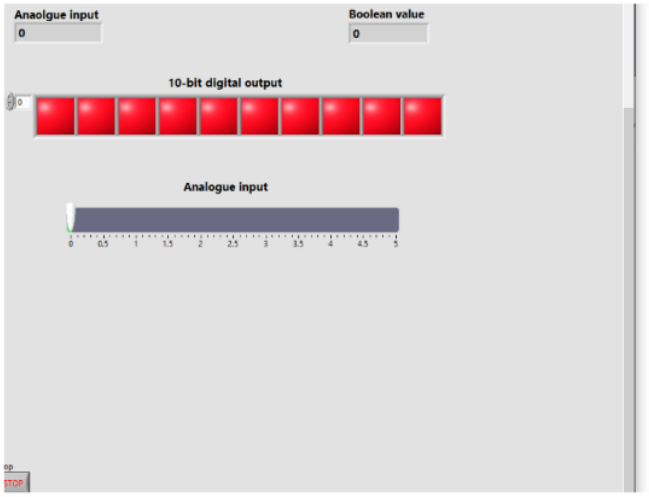
Test 2: A > B on comparator
Load = [4.55 / 5] * 1000 = 910g
Digital value = [567 / 1000] * 256 = 232
232 = 11101001

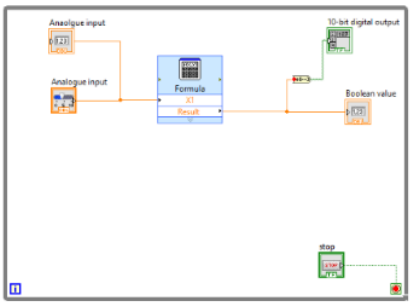
Test 3: A < B on comparator *Part missing*

3. Prototypes

3.1. 10-bit resolution ADC LabVIEW

Figure 7-9 below represent a 10-bit digital output resolution ADC. The software which was used to construct these images is called LabVIEW, LabVIEW makes the process of building this interactive technology very easy with simple human interface with little programming





needs.

Figure 56: 10-bit digital output at a value of 0

Figure 7 shows the human interface or the interactive screen on the left with the program along with all the program blocks on the right. Figure 7 has been placed here to show what response you get at an analogue input of 0.

Figure 8 has a analogue input value of 2.5. Visually the value has been displayed in different forms on the interactive screen. The screen shows a digital analogue value to 5d.p and a slider counter which varies the values from 0 – 5. The Analogue input value can then be outputted and displayed by the 10-bit digital output counter which utilises coloured blocks to indicate the value.

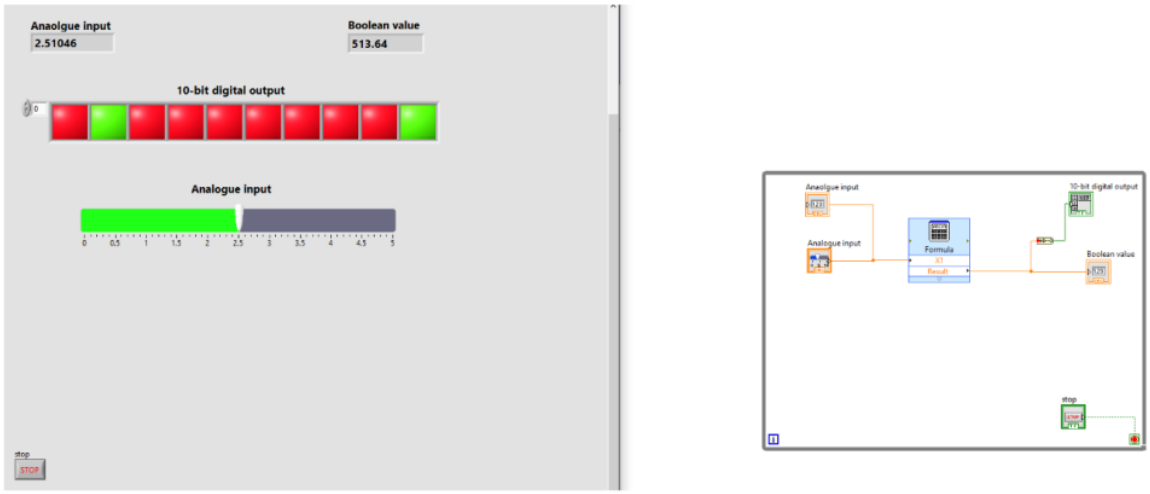


Figure 57: 10-bit digital output at a value of 2.5

From this the Boolean value for the digital output can be calculated, see below for a proven example:

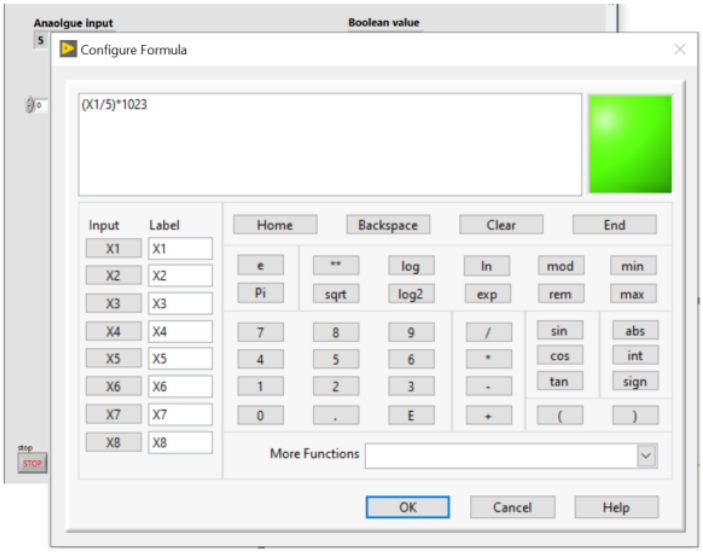
1	2	4	8	16	32	64	128	256	512
0	1	0	0	0	0	0	0	0	1

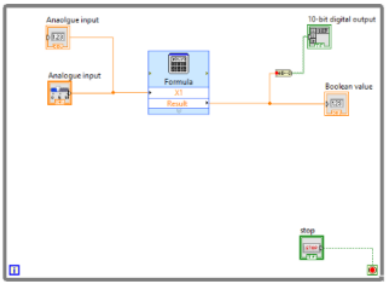
Table 2: Boolean equivalent value for an analogue value of 2.5

With a 10-bit word the range is 0 – 1023. From table 2 the total amount bits will add up to 1023, from this value the number of red counters need to be deducted, thus:
1023 – 509 = 514

Another new value has been displayed for figure 9, this value being 5. This is the maximum value that counter can make. Results show that all green indicators are illuminated meaning they all have a value of 1. From this the Boolean value can be derived however due to there not being 0 values then only the maximum Boolean will be displayed, this being 1023.

Figure 58: 10-bit digital output at a value of 5





10 bit ADC design is good

Figure 59: Formula block equation

Figure 10 shows the formula used within the function block.

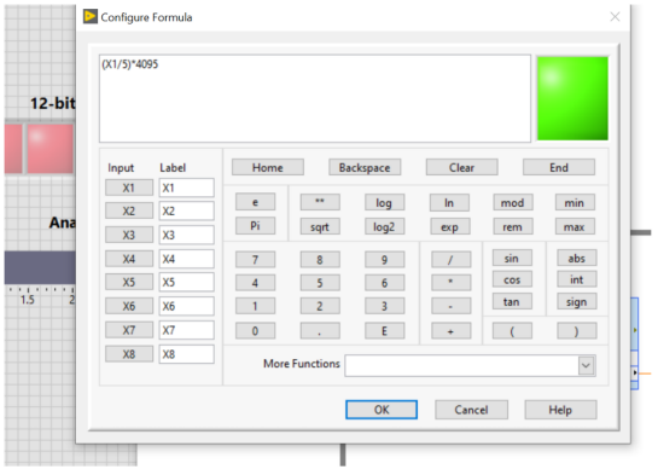
3.2. 12-bit resolution ADC LabVIEW

The ADC's resolution has now been changed from a 10-bit to a 12-bit which can be seen by figure 11. Visual differences show that rather than 10 illuminated blocks as they were before they are now 12 illuminated blocks to suit the 12-bit ADC resolution.



Figure 60: 12-bit digital output at a value of 2.5

As well as the illuminated blocks changing the formula has also been altered, this can be seen by figure 12.



12 bit design is good

Figure 61: Formula block for a 12-bit resolution ADC

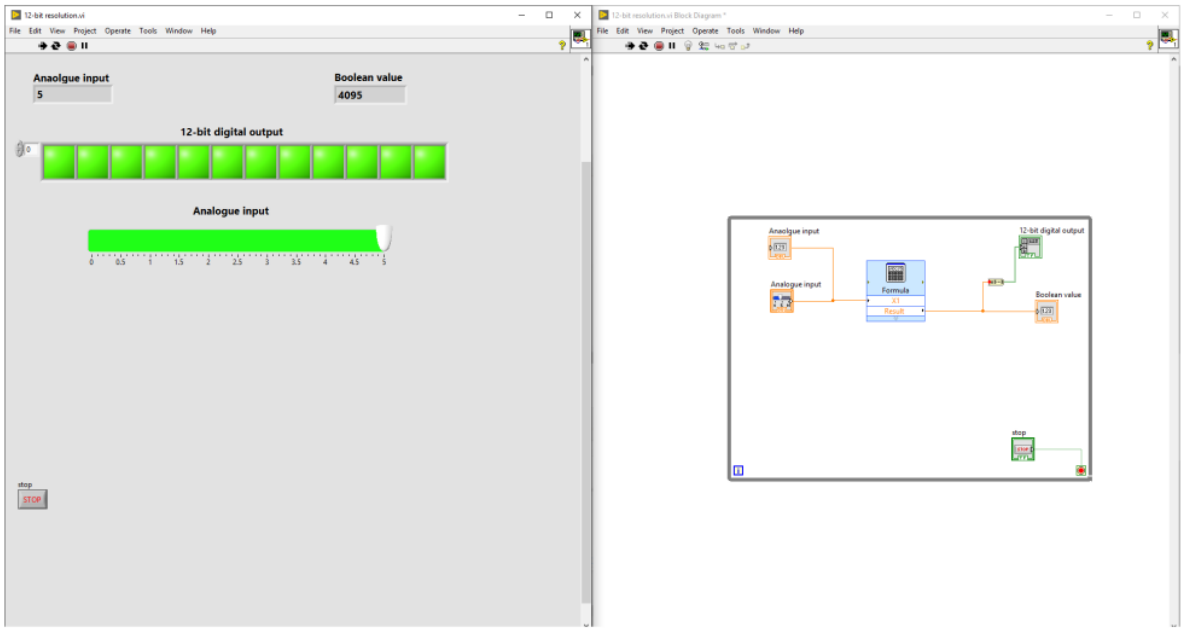
1	2	4	8	16	32	64	128	256	512	1024	2048
0	0	0	0	0	0	0	0	0	0	0	1

Table 3: 12-bit analogue value of 2.5

Another value can be determined for a 12-bit analogue value of 2.5, this being 2048. See above table 3.

Now that another 2 lights have been added to the diagram the Boolean equation will have to change accordingly. Thus, through the additional 2 bits added there will now be a Boolean range between 0 – 4095.

Figure 62: 12-bit digital output at a value of 5



The digital output seen in figure 13 above can be proven below:

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

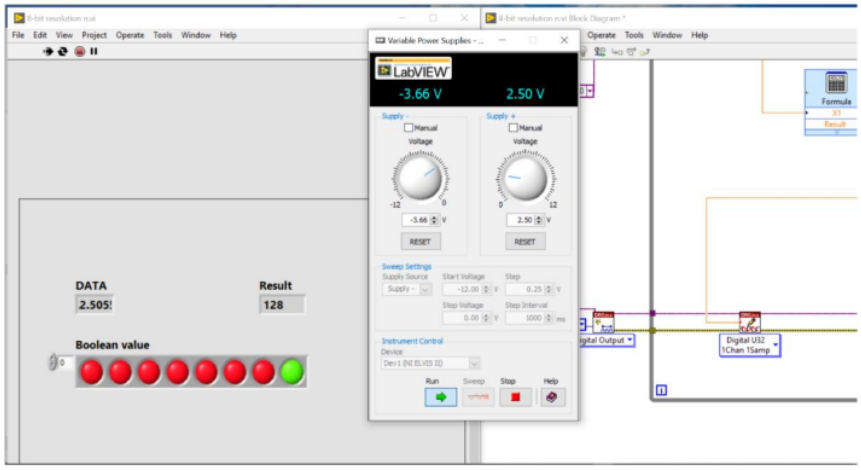
Table 4: 12-bit analogue value of 5

Table 3 shows that when all the blocks are energised meaning they are of a value of 1 then all the lights will be illuminated, and a maximum value can be determined, this value is 4095 which is also shown in the formula above in figure 12.

12bitADCtestisgood

3.3. 8-bit ADC

An 8-bit ADC resolution has now been determined and can be shown by figure 14. Illustrated by figure 14 the analogue input is no longer a slider like the 10-bit and 12-bit but instead a potentiometer that can regulate the input voltage.

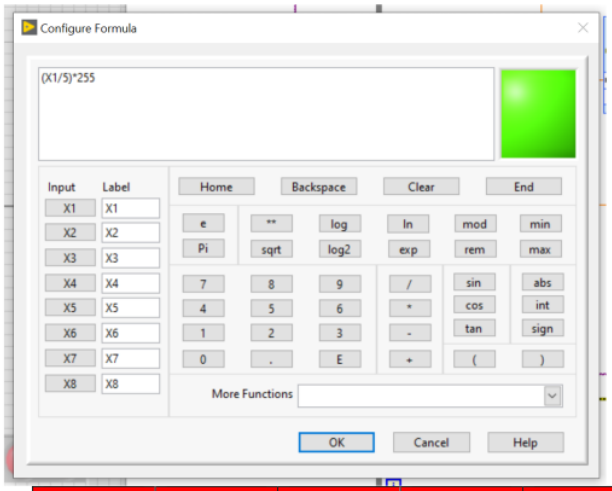


this does not use the POT; this uses a virtual PS as shown

Figure 63: 8-bit resolution ADC at 2.5

The formula for the 8-bit ADC can be shown by figure 15 below.

Figure 64: Formula for an 8-bit ADC



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

Table 5: 8-bit resolution ADC with a value of 2.5

The formula represents the following:
X1: Analogue input signal

5: Maximum voltage scale
256: 8-bit resolution

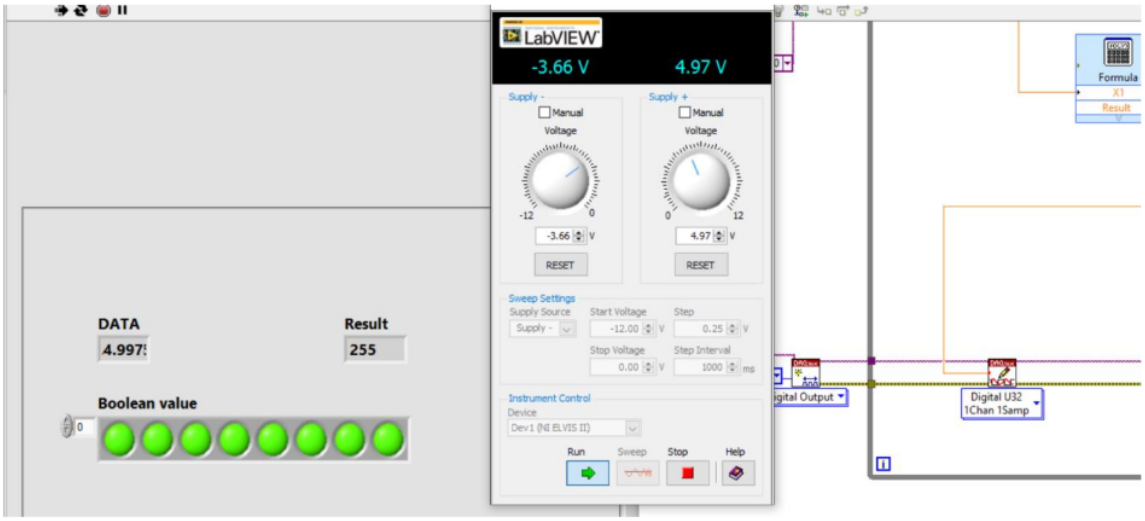


Figure 65: 8-bit ADC with a value of 5

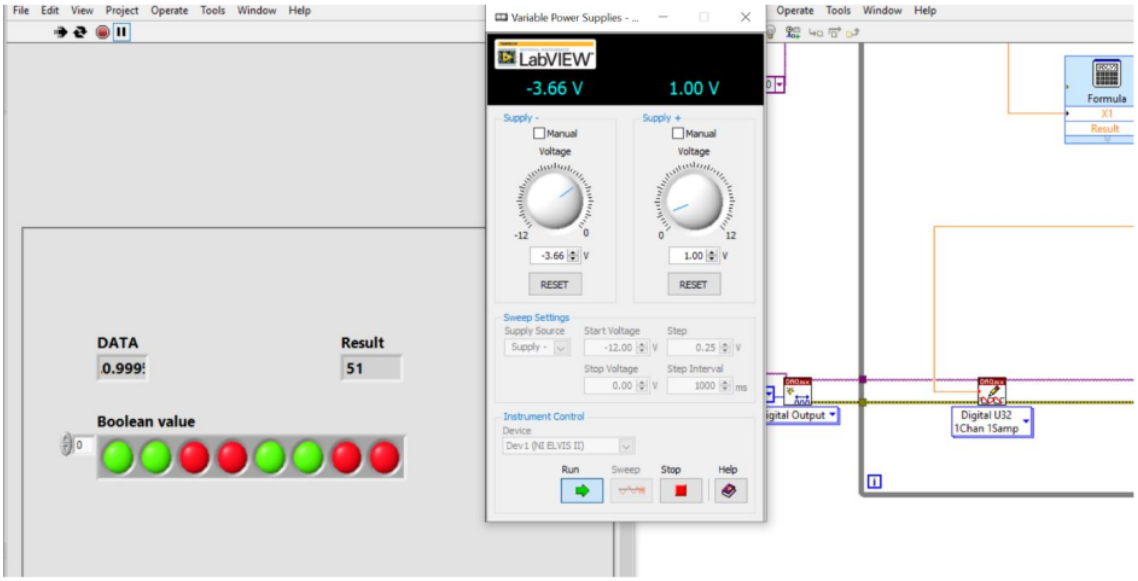


Figure 66: 8-bit ADC with a value of 1

Figures 16 and 17 are further example of the 8-bit resolution ADC but with different data values. The LabVIEW software is compatible with the NI Elvis II board and the board has been displayed below with its corresponding lights seen on screenshot of the LabVIEW program in figure 17.

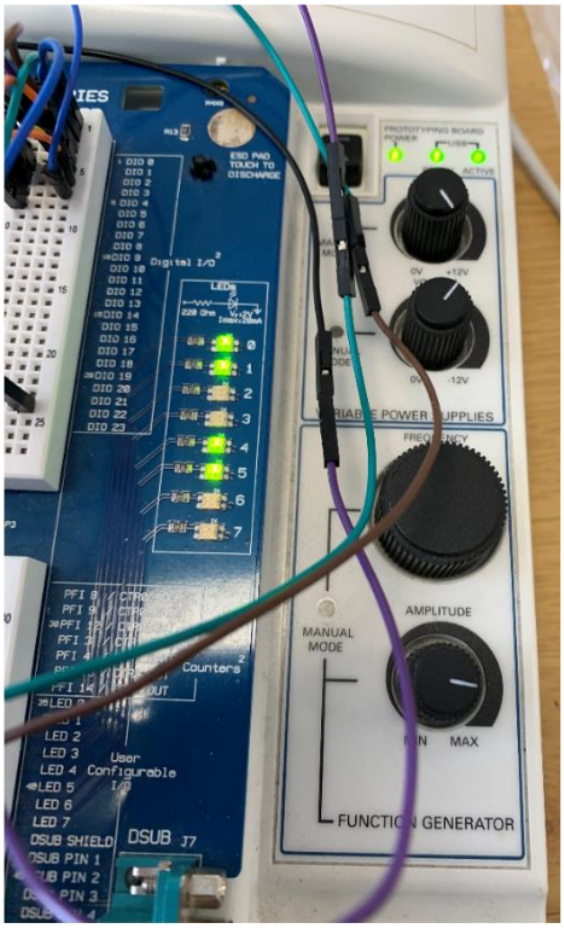


Figure 67: Elvis board relating to a value of

1	2	4	8	16	32	64	128
1	1	1	1	1	1	1	1

Table 6: Boolean value for figure 16

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

Table 7: Boolean value for figure 17

Table 7 represents the Elvis board image shown by figure 18. Clearly, the Boolean value displayed is:

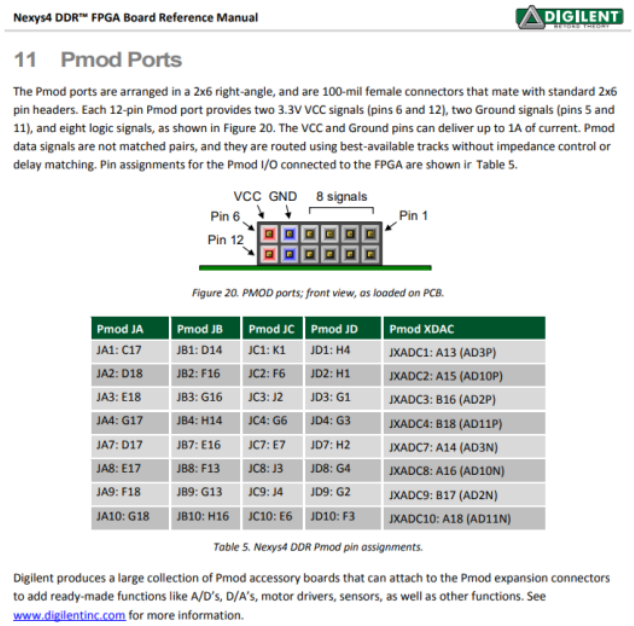
$1 + 2 + 16 + 32 = 51$
$255 - 51 = 204$

8 bit ADC tests show the design is worki ng

4. Digital Magnitude Comparator

4.1. Nexys 4 DDR FPGA Development board

This small but powerful board can communicate with MATLAB software meaning all programs written can be downloaded onto the Nexys board.



Nexys4 DDR™ FPGA Board Reference Manual

DIGILENT

11 Pmod Ports

The Pmod ports are arranged in a 2x6 right-angle, and are 100-mil female connectors that mate with standard 2x6 pin headers. Each 12-pin Pmod port provides two 3.3V VCC signals (pins 6 and 12), two Ground signals (pins 5 and 11), and eight logic signals, as shown in Figure 20. The VCC and Ground pins can deliver up to 1A of current. Pmod data signals are not matched pairs, and they are routed using best-available tracks without impedance control or delay matching. Pin assignments for the Pmod I/O connected to the FPGA are shown ir Table 5.

Figure 20. PMOD ports; front view, as loaded on PCB.

Pmod JA	Pmod JB	Pmod JC	Pmod JD	Pmod XDAC
JA1: C17	JB1: D14	JC1: K1	JD1: H4	JXADC1: A13 (AD3P)
JA2: D18	JB2: F16	JC2: F6	JD2: H1	JXADC2: A15 (AD10P)
JA3: E18	JB3: G16	JC3: J2	JD3: G1	JXADC3: B16 (AD2P)
JA4: G17	JB4: H14	JC4: G6	JD4: G3	JXADC4: B18 (AD11P)
JA7: D17	JB7: E16	JC7: E7	JD7: H2	JXADC7: A14 (AD3N)
JA8: E17	JB8: F13	JC8: J3	JD8: G4	JXADC8: A16 (AD10N)
JA9: F18	JB9: G13	JC9: J4	JD9: G2	JXADC9: B17 (AD2N)
JA10: G18	JB10: H16	JC10: E6	JD10: F3	JXADC10: A18 (AD11N)

Table 5. Nexys4 DDR Pmod pin assignments.

Digilent produces a large collection of Pmod accessory boards that can attach to the Pmod expansion connectors to add ready-made functions like A/D's, D/A's, motor drivers, sensors, as well as other functions. See www.digilentinc.com for more information.

FPGA ports are identified

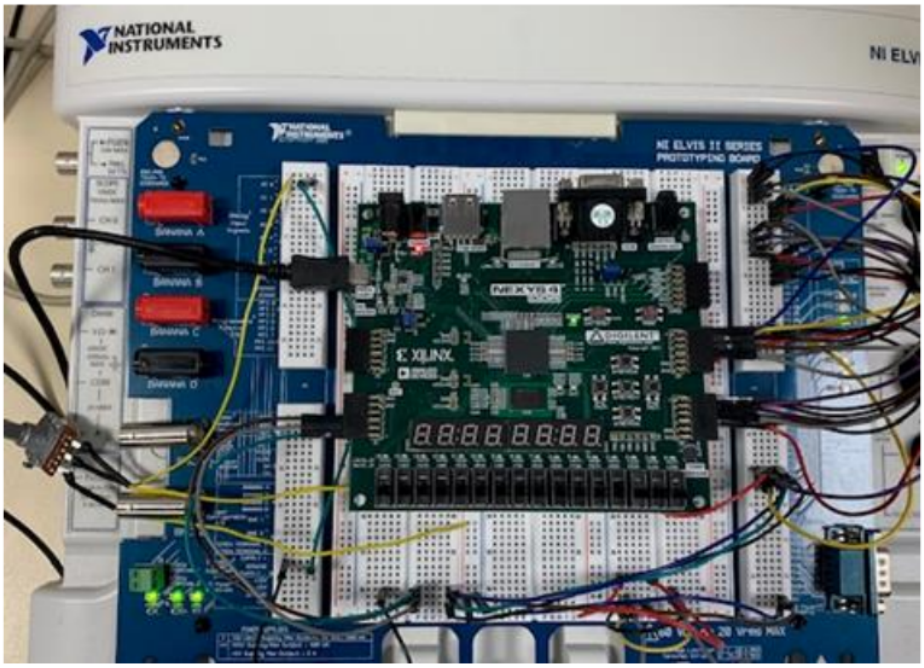
Figure 68: Nexys 4 DDR electrical connections

The FPGA will also be used as a magnitude comparator and will work with the LabVIEW program also. The board can be mounted to the Elvis II board, but it is critical that the correct connections are made thus figure 19 can be followed to minimise any errors.

4.2. Magnitude Comparator Setup

Spelling

The complete setup of the Nexys bard integrated into the Elvis II board can be seen by figure 20 below.



Good image showing equipment setup

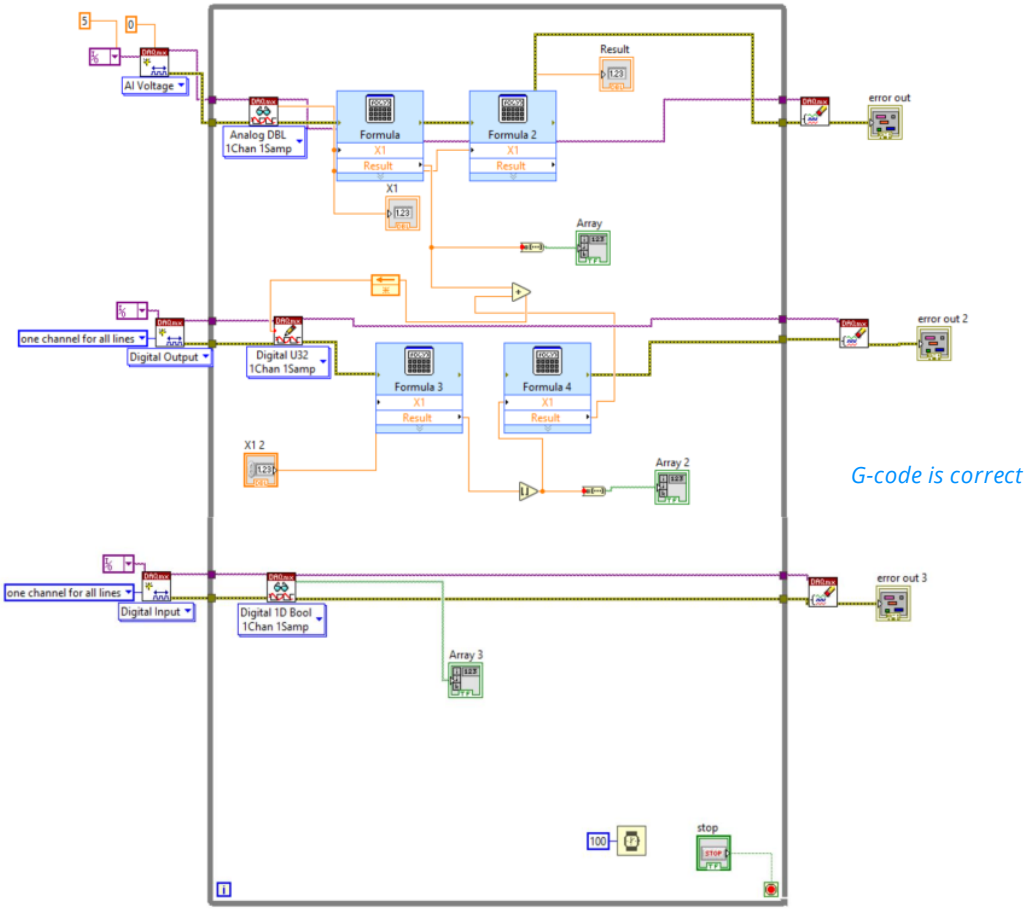
Figure 69: Practical setup of Nexys board and Elvis II board

To alter the voltage accordingly through the setup in figure 20 a potentiometer has been included for a manual increase or decrease.

4.3. LabVIEW

As mentioned previously, the LabVIEW program was utilised to create a program that could be downloaded to the final practical setup shown by figure 20. Figure 21 shows a series of block functions which have all been combined to make one program. There are 4 separate formula blocks integrated into the program and screenshots have been taken of each formula and can be seen by figure 22 – 26.

Figure 70: Block diagram program for magnitude comparator



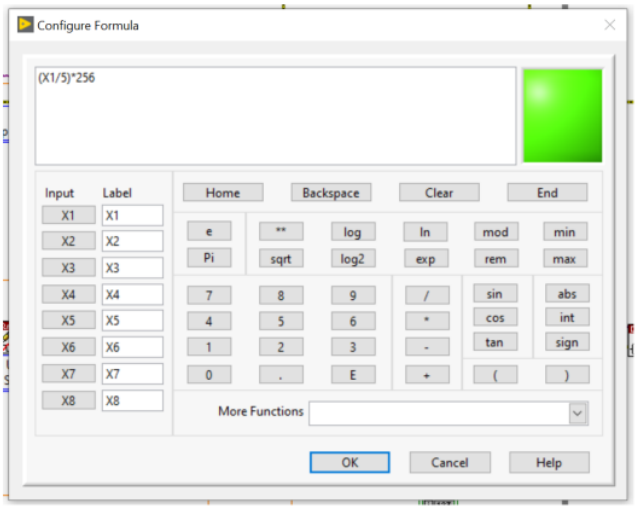


Figure 71: Formula block no. 1

The importance of formula 1 is to convert the analogue input into a gram value. Clearly shown by figure 22 the formula uses the analogue input by the maximum analogue input (5) and multiplies by 1000 which converts to grams.

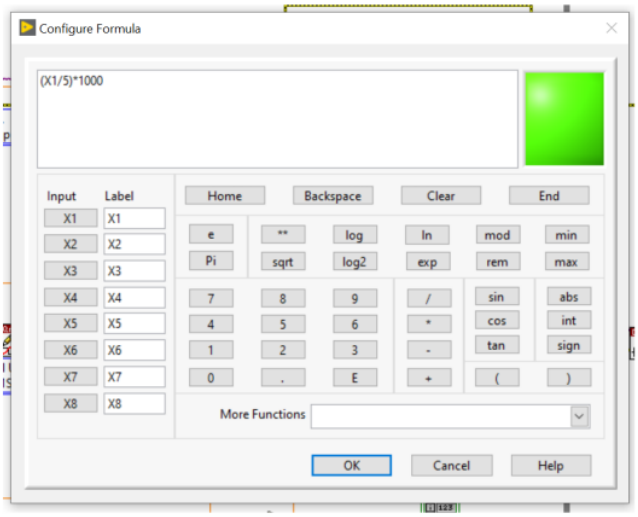


Figure 72: Formula block no. 2

Formula 2 converts the analogue input (5) and then multiplies this figure by 256 which is the resolution which in turn converts into binary.

This is not accurate

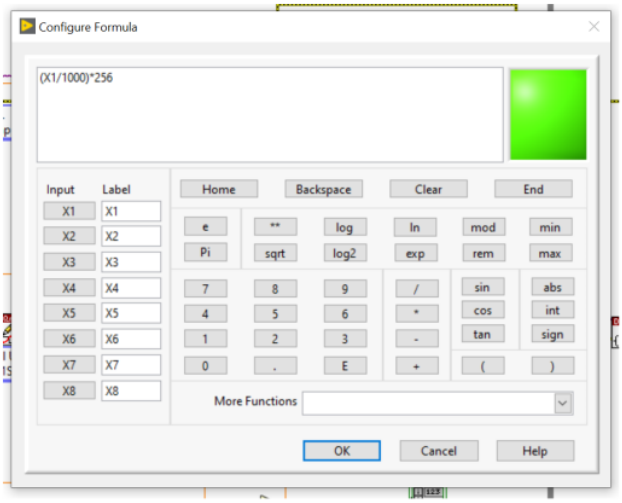


Figure 73: Formula block no. 3

Formula 3 converts the set gram value into a digital 8-bit word output. To do this as shown by figure 24 the gram value is divided by the maximum gram value, which is 1000, this is then multiplied by the resolution.

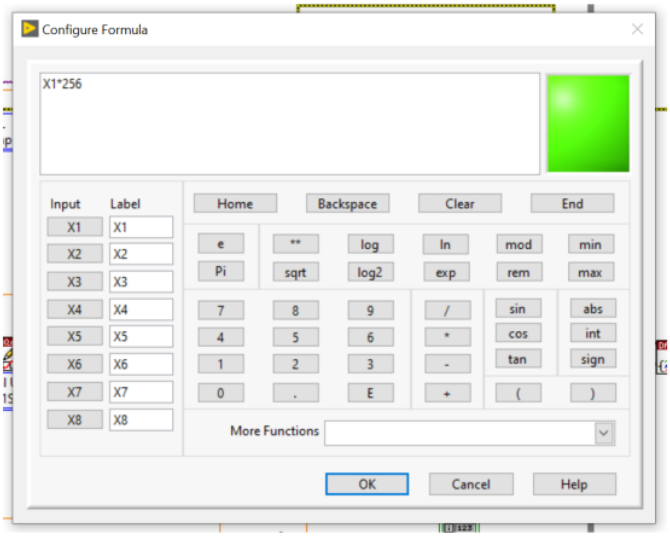


Figure 74: Formula block no. 4

This block should be explained. This is where the 8 bits are converted to 16 bits

by multiplying the first 8 bits by 256. Shift right 8 bits

4.4. Test Results

At this stage, the program has been downloaded to the Elvis II board and the input load can be manually adjusted through use of the potentiometer which can be seen in figure 20.
As shown below 3 separate tests were carried out, this includes a test for a less than value, an equal to value and a more than value. These tests ensure that the magnitude comparator works accordingly. The set load value has been changed in each case to determine an easier access to the desired value. The Load input will be according to the output it gives, meaning that a load input for an output equal to value will be less than a more than load input value but more than a less than load input value.



Figure 75: Less Than

Figure 26 shows that the load input set less than the constant set load, the magnitude comparator has determined these values less than.

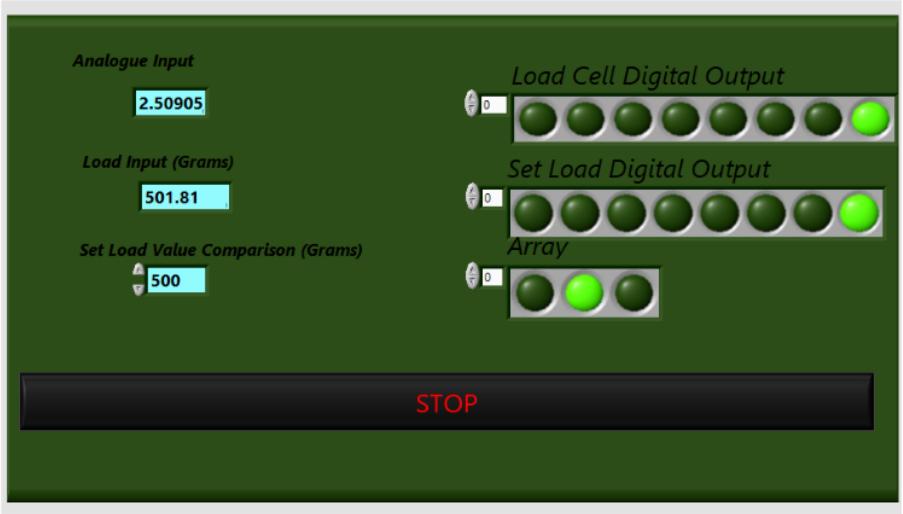


Figure 76: Equal to

Figure 27 shows that the load input is set equal to the constant set load, the magnitude comparator has determined the set load value equal to the set value.

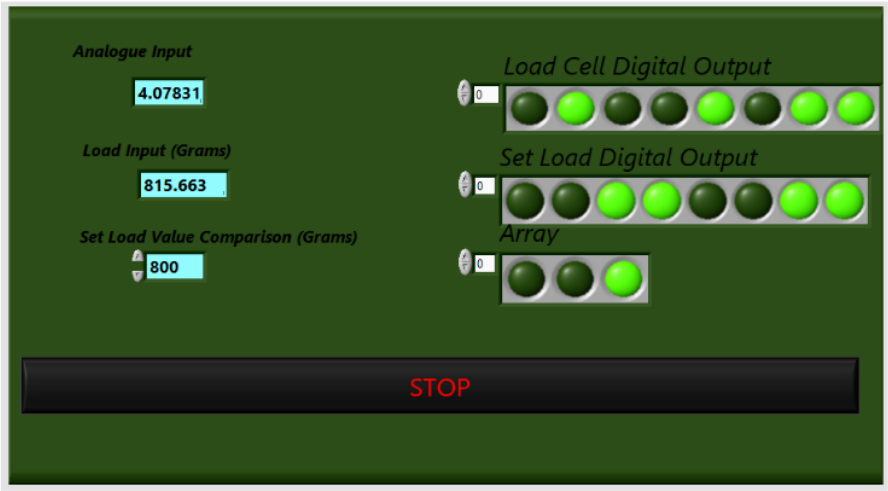


Figure 77: More than

Tests are good

Figure 28 shows that the load input is set more than the constant set load, the magnitude comparator has a set load value which is higher than the set value. As seen by the set load value and the array LED indicators this is correct.

5. Evaluation

In evaluation the portfolio presented the following. The load cells and signal conditioning showed an instrument amplifier with unity gain, the instrument was presented as a wiring schematic. Later, further studies were carried out into load cells and how they can be implemented into the system. A table of data was compiled together which showed altering the percentage in different stages. The results showed that when the resistance of the circuit posed a greater difference the larger the weight and the higher the voltage, a line graph was plotted from these results.

Later, three tests were carried out on the Elvis II board. The tests included the LabVIEW analysis with the Magnitude Comparator which carried out a first test for A = B on comparator the second was A > B on comparator and the third showed A < B on comparator. Results for these tests can be seen on page 10.

Different prototypes were carried out during this portfolio. A 10-bit 12-bit and 8-bit were constructed, and screenshots were taken of each ADC. Boolean expressions could be derived from the data collected which could be compared against the digital value displayed on the corresponding screenshots. The difference between the 8-bit and the 10 – 12-bit showed that rather than using a slider counter a digital potentiometer was utilised. Figure 18 captures a screenshot of the Elvis II board LED's which corresponded to the 8-bit resolution ADC figure 17, this proved accuracy between simulated and practical results and proved to be successful.

Later the Digital Magnitude Comparator integrated a nexys 4 DDR board into the Elvis II board. This was done to carry out manual tests of determining values of less than, equal to and more than. To do this a suitable program was constructed which complied with the nexys 4 DDR and the Elvis II board, figure 21 shows the program for this. Screenshots were captured for these three different states and have been shown by figure 26, 27 and 28.

A reasonable evaluation

6. Appendix

6.1. Load cells and signal conditioning

55 – 45

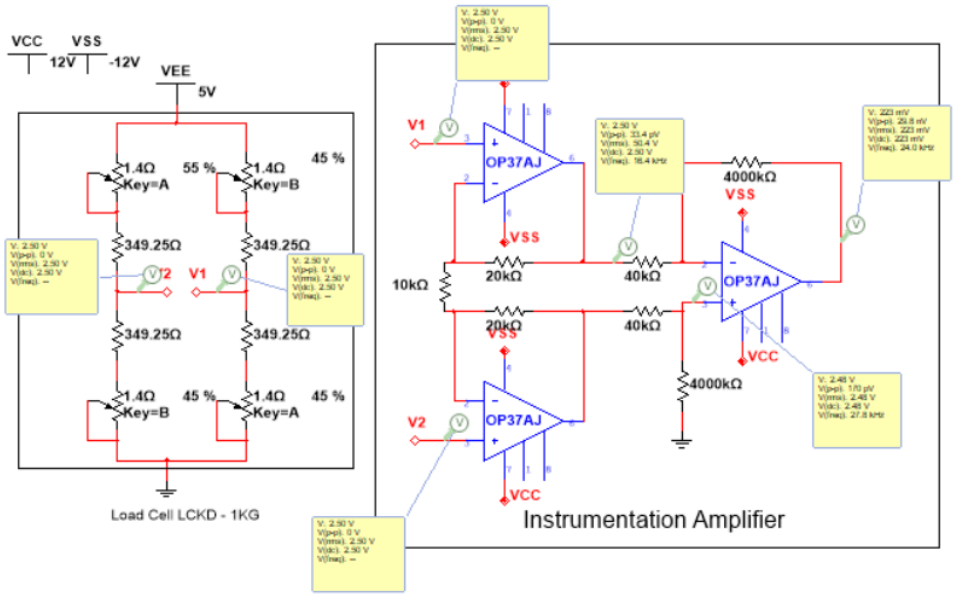
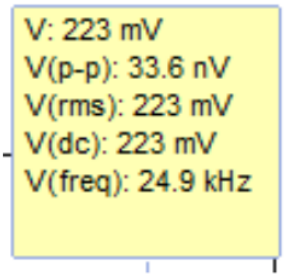
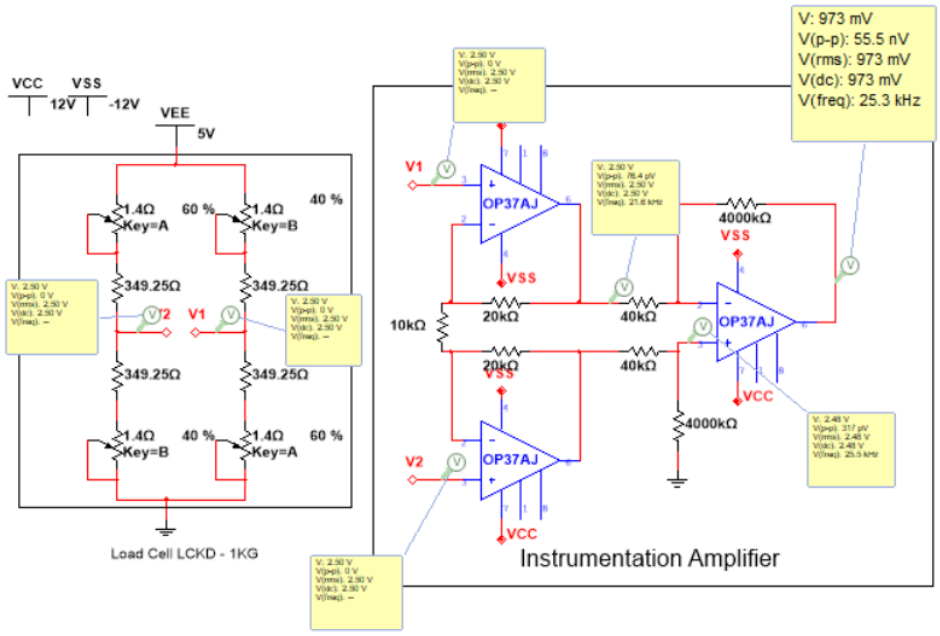


Figure 78: percentage between 55-45

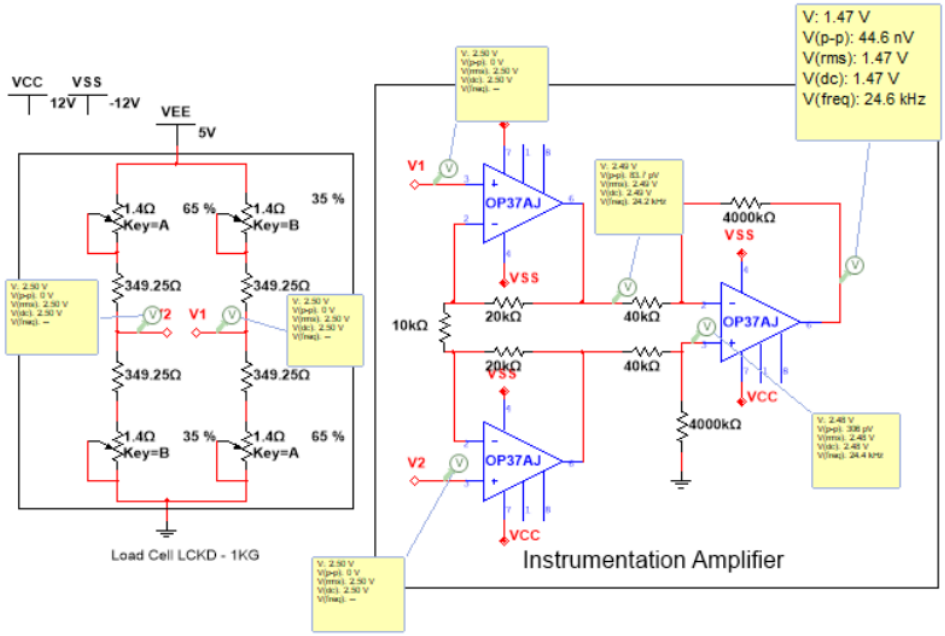
Zoom in of V results on the output



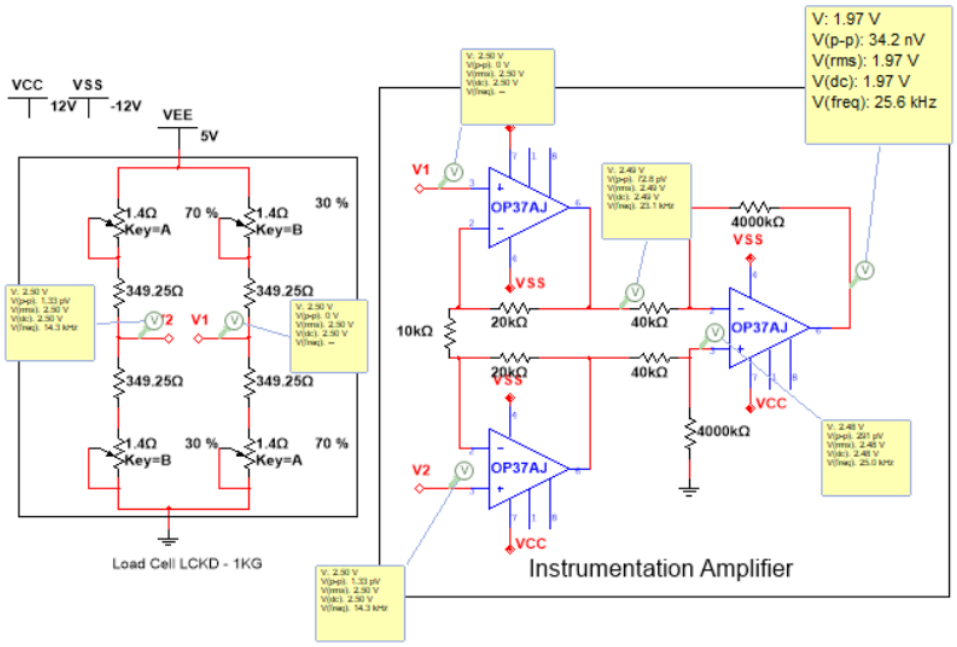
60 – 40


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
60 %
1.4Ω
Key=B
40 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
40 %
1.4Ω
Key=A
60 %
Load Cell LCKD - 1KG
V: 973 mV
V(p-p): 55.5 nV
V(rms): 973 mV
V(dc): 973 mV
V(freq): 25.3 kHz
V1
OP37AJ
VSS
4000kΩ
VSS
10kΩ
20kΩ
40kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

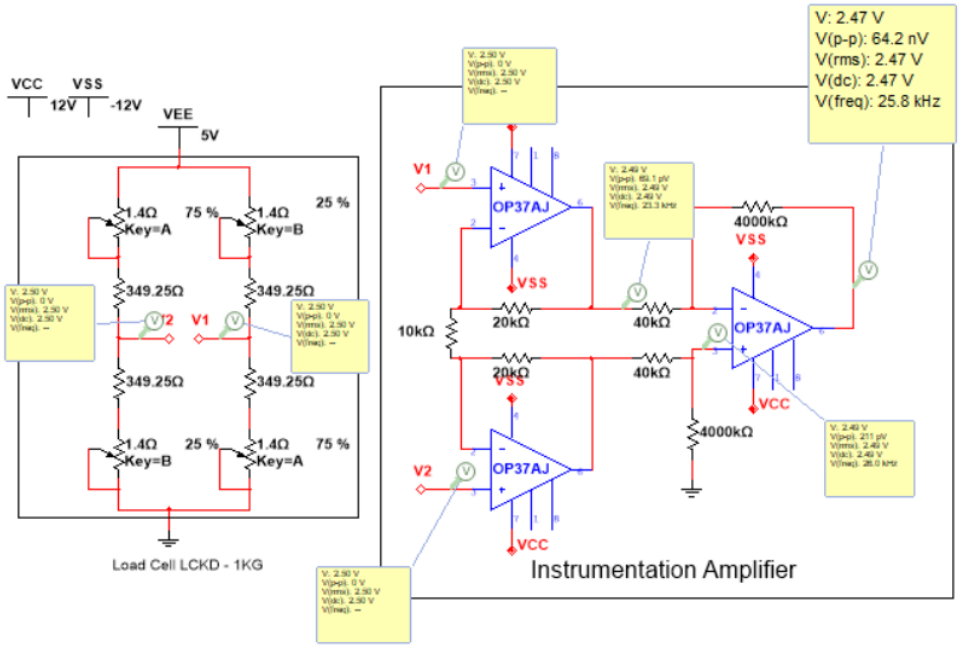
65 – 35


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
65 %
1.4Ω
Key=B
35 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
35 %
1.4Ω
Key=A
65 %
Load Cell LCKD - 1KG
V: 1.47 V
V(p-p): 44.6 nV
V(rms): 1.47 V
V(dc): 1.47 V
V(freq): 24.6 kHz
V1
OP37AJ
VSS
4000kΩ
VSS
10kΩ
20kΩ
40kΩ
OP37AJ
20kΩ
VSS
40kΩ
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

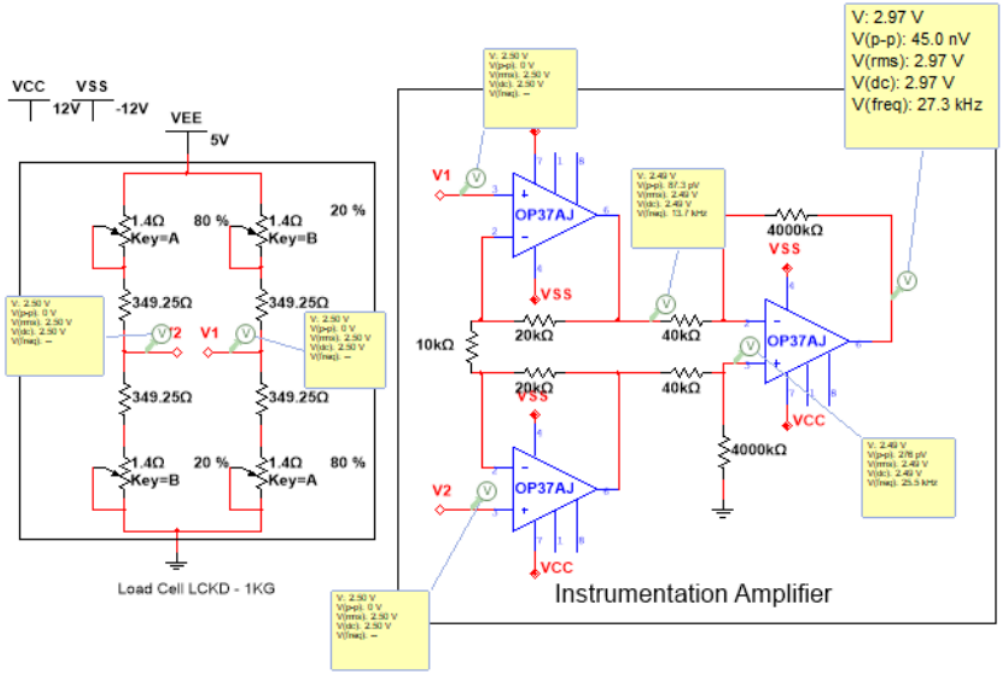
70 – 30


VCC
12V
VSS
-12V
VEE
5V
1.4Ω
Key=A
70 %
1.4Ω
Key=B
30 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
30 %
1.4Ω
Key=A
70 %
Load Cell LCKD - 1KG
V: 1.97 V
V(p-p): 34.2 nV
V(rms): 1.97 V
V(dc): 1.97 V
V(freq): 25.6 kHz
V1
OP37AJ
VSS
10kΩ
20kΩ
40kΩ
4000kΩ
OP37AJ
VSS
VCC
20kΩ
40kΩ
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

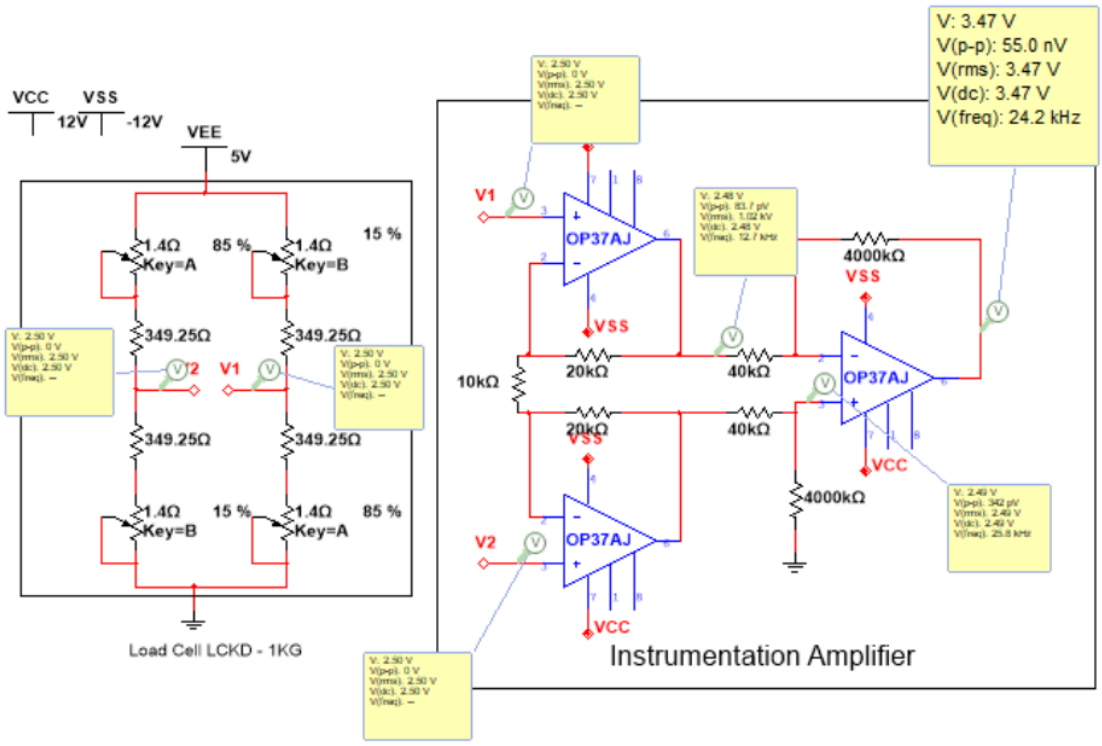
75 – 25


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
75 %
1.4Ω
Key=B
25 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
25 %
1.4Ω
Key=A
75 %
Load Cell LCKD - 1KG
V: 2.47 V
V(p-p): 64.2 nV
V(rms): 2.47 V
V(dc): 2.47 V
V(freq): 25.8 kHz
V1
OP37AJ
VSS
4000kΩ
VSS
10kΩ
20kΩ
40kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

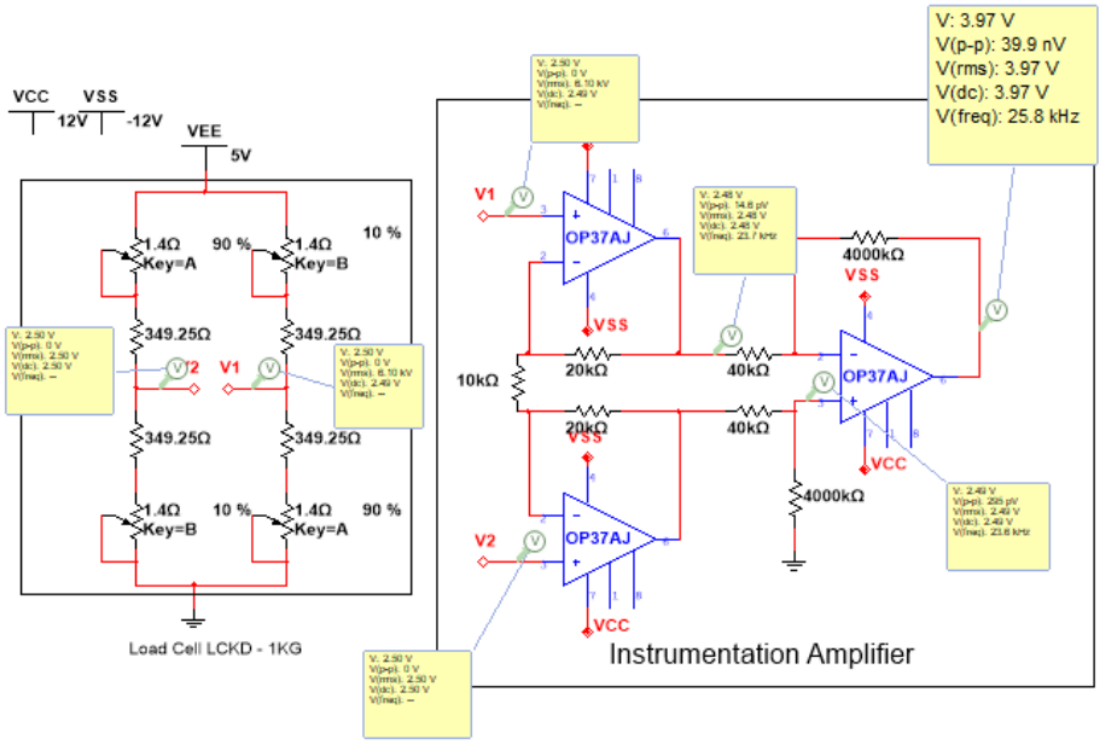
80 – 20


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
80 %
1.4Ω
Key=B
20 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
20 %
1.4Ω
Key=A
80 %
Load Cell LCKD - 1KG
V: 2.97 V
V(p-p): 45.0 nV
V(rms): 2.97 V
V(dc): 2.97 V
V(freq): 27.3 kHz
V1
OP37AJ
VSS
4000kΩ
VSS
10kΩ
20kΩ
40kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

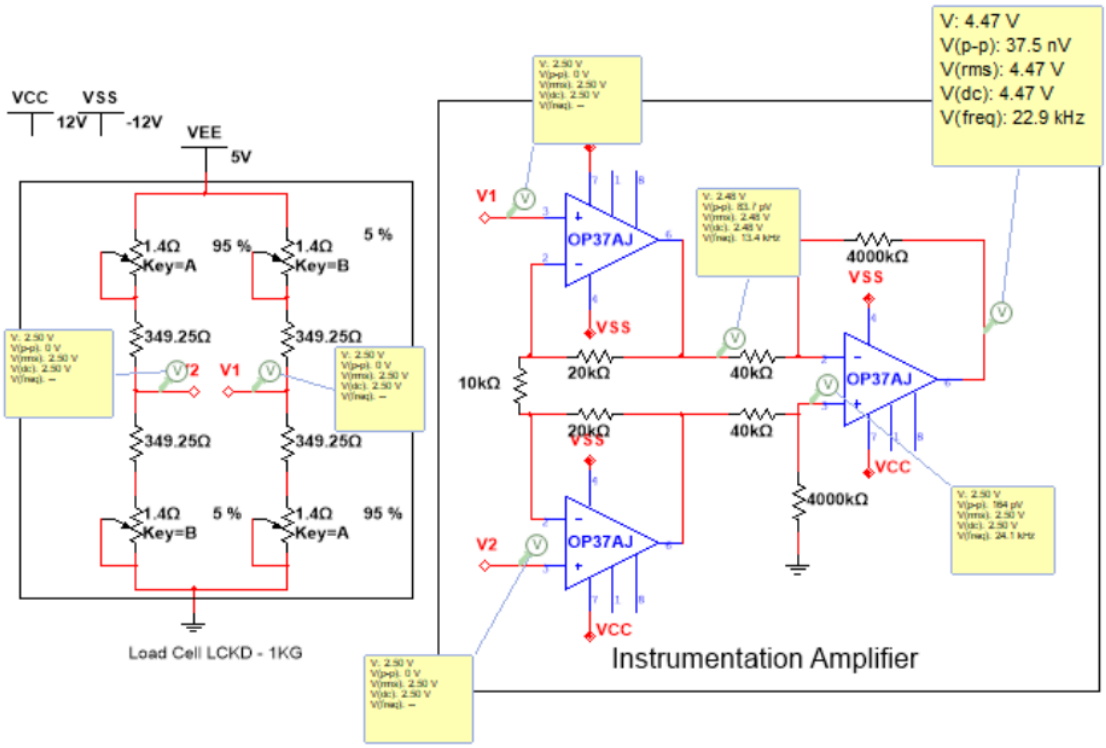
85 – 15


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
85 %
1.4Ω
Key=B
15 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
15 %
1.4Ω
Key=A
85 %
Load Cell LCKD - 1KG
V: 3.47 V
V(p-p): 55.0 nV
V(rms): 3.47 V
V(dc): 3.47 V
V(freq): 24.2 kHz
V1
OP37AJ
VSS
20kΩ
40kΩ
4000kΩ
VSS
10kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

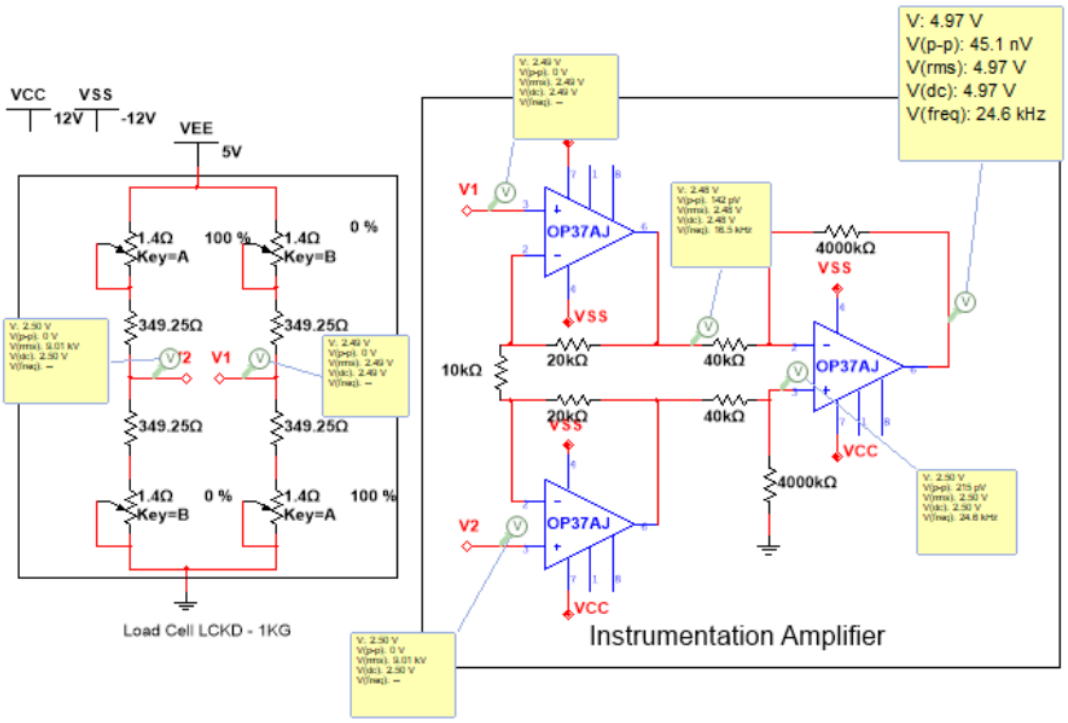
90 – 10


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
90 %
1.4Ω
Key=B
10 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
10 %
1.4Ω
Key=A
90 %
Load Cell LCKD - 1KG
V1
OP37AJ
VSS
20kΩ
40kΩ
4000kΩ
VSS
10kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier
V: 3.97 V
V(p-p): 39.9 nV
V(rms): 3.97 V
V(dc): 3.97 V
V(freq): 25.8 kHz

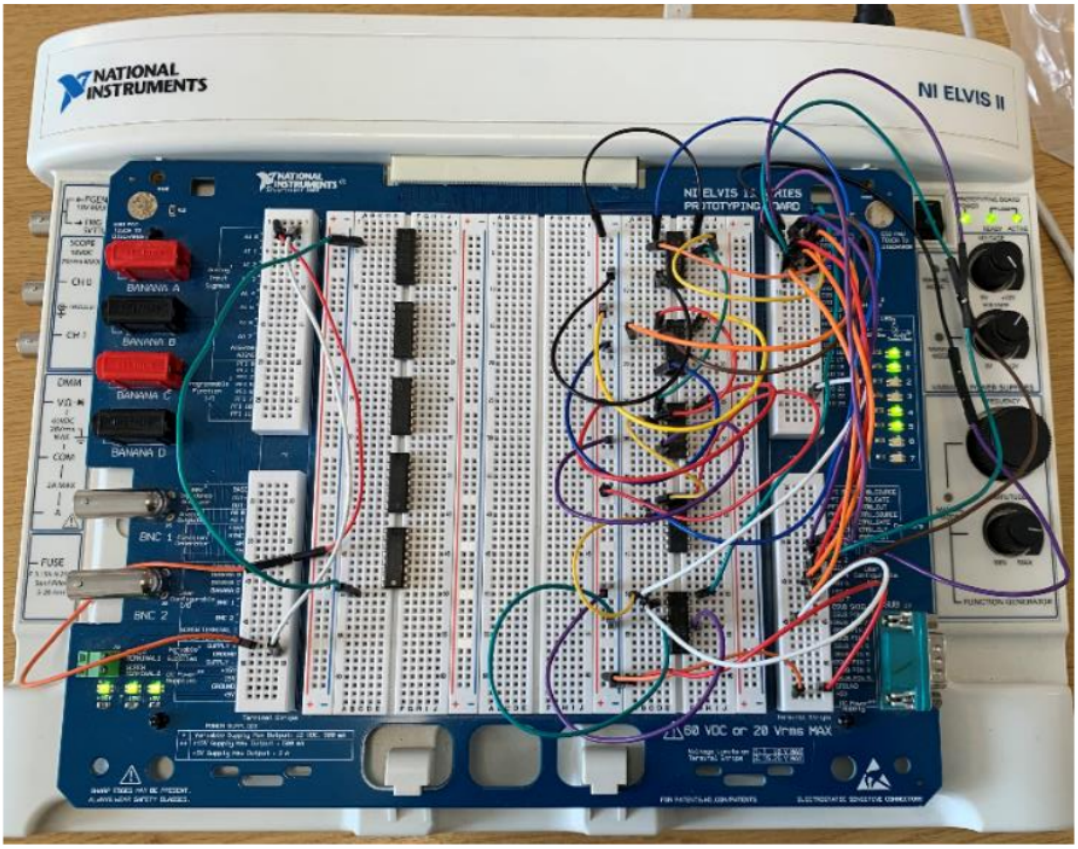
95 – 5


VCC
VSS
12V
-12V
VEE
5V
1.4Ω
Key=A
95 %
1.4Ω
Key=B
5 %
349.25Ω
349.25Ω
V2
V1
349.25Ω
349.25Ω
1.4Ω
Key=B
5 %
1.4Ω
Key=A
95 %
Load Cell LCKD - 1KG
V: 4.47 V
V(p-p): 37.5 nV
V(rms): 4.47 V
V(dc): 4.47 V
V(freq): 22.9 kHz
V1
OP37AJ
VSS
4000kΩ
VSS
10kΩ
20kΩ
40kΩ
OP37AJ
20kΩ
40kΩ
VSS
VCC
4000kΩ
V2
OP37AJ
VCC
Instrumentation Amplifier

100 – 0



All instrument amplifier tests are present



DAQ board test indicates the accuracy of the test

Course:	HND Electrical and Electronic engineering
Module:	SGEE 5007 VICS
Assignment:	NI LabVIEW Magnitude Comparator E3
Name:	Mason Richard Morgan
University number:	1809802
Date submitted:	20th MAY 2021.
Word count:	1379

Contents

Table should have more details to point to certain significant tasks.

1. Introduction 80
2. Design requirements 81
3. Implementation 82
4. Testing and Analysis 84
   - 4.1. 200-gram SetPoint 84
   - 4.2. 500-gram Setpoint 85
   - 4.3. 700-gram Setpoint 86
5. Conclusions and Recommendations 88

Appendix 89

System setup 89

Excel data for 200-, 500- and 700-gram setpoints 91

Figure 1: LabVIEW for data recording 81
Figure 2: Log text files 82
Figure 3: File location save 83
Figure 4: Setpoint for 200grams in the form of a line graph 84
Figure 5: HMI screen displaying Less than value. 84
Figure 6: Setpoint for 500-grams in the form of a line graph 85
Figure 7: HMI screen displaying equal to value. 86
Figure 8: 700-gram Setpoint 86
Figure 9: HMI screen displaying More than value. 87
Figure 10: Integrating Nexys board into Elvis II board 89
Figure 11: Less than Setpoint 89
Figure 12: Equal to Setpoint 90
Figure 13: Greater than Setpoint 90
Figure 14: Logged data for 200-gram setpoint 91
Figure 15: Data logged for 500-gram setpoint 92
Figure 16: Data logged for 700-gram setpoint 93

List of figures is good

1. Introduction

During this stage of the report the following activities occurred:
The NI LabVIEW design will need to output the actual Load cell value in grams. It will record this data by converting into the form of data logging files.
A program made up of function blocks must be constructed from the LabVIEW software. The program will be used to record live data. The information that must be logged are:

- Load input
- Set Load Comparison
- Load input Under set load
- Load input Equal set load
- Load input Over set load

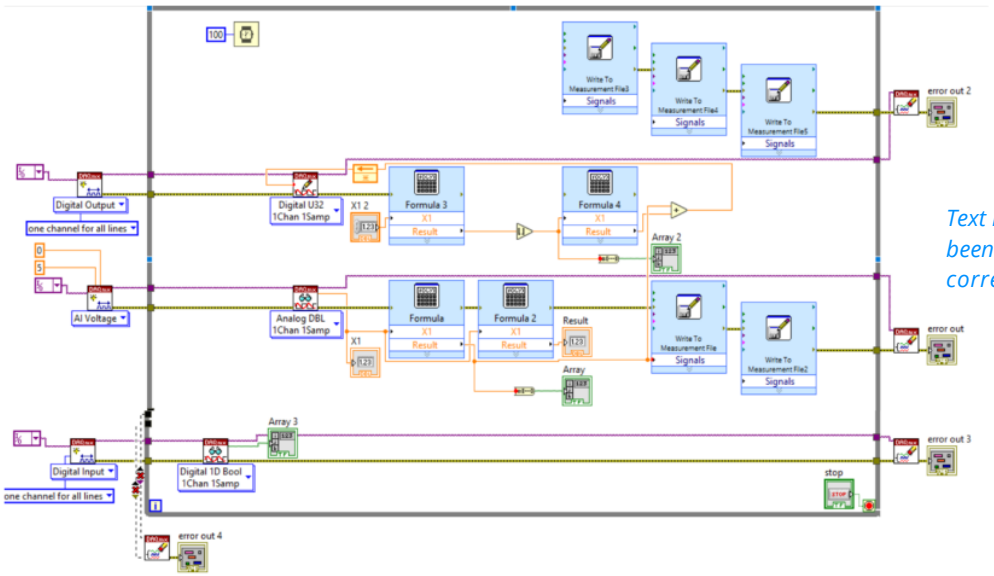
Microsoft excel will be used to construct graphs from the live data collected previously. By the end of this element all data from the system should be obtained using CSV text files and presented in an excel document and in the form of graphs for simplicity.
HMI screens can be constructed to vary the current making the load value either:

- Less than
- Equal to
- More than

Introducti on is good

2. Design requirements

A program can be seen below illustrated by figure which has been constructed on LabVIEW. The program uses multiple function blocks to combine to make a data logging program. The objective with this program is to record data based on the earlier set point magnitude comparator.



Text blocks have been implemented correctly

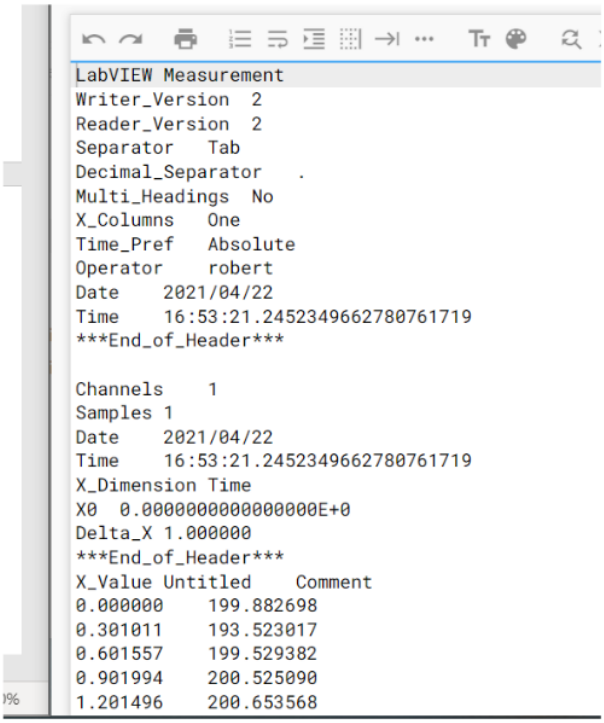
Figure 79: LabVIEW for data recording

This program can save all data logged over a specific time of running the software into a separate text file. An advantage of this data logging method means that it uses a potentiometer which means that all comparator values can be logged against the set point which will keep a record. The three file blocks in figure 1 are three comparator outputs which represent Less than, Equal to and greater than.

3. Implementation

Figure 2 illustrates a text file block specifically from file 1. To explain the process happening it could be said that inputs are being taken and saved in a specific location.
There are multiple files at once taking data, File 1 is using the input from the potentiometer which is used as the load. File 2 is taking the input from the set point and stays constant. Information from Figure 2 can be extracted and used to plot data graphs. Some of the useful information being recorded can be seen below, these are:

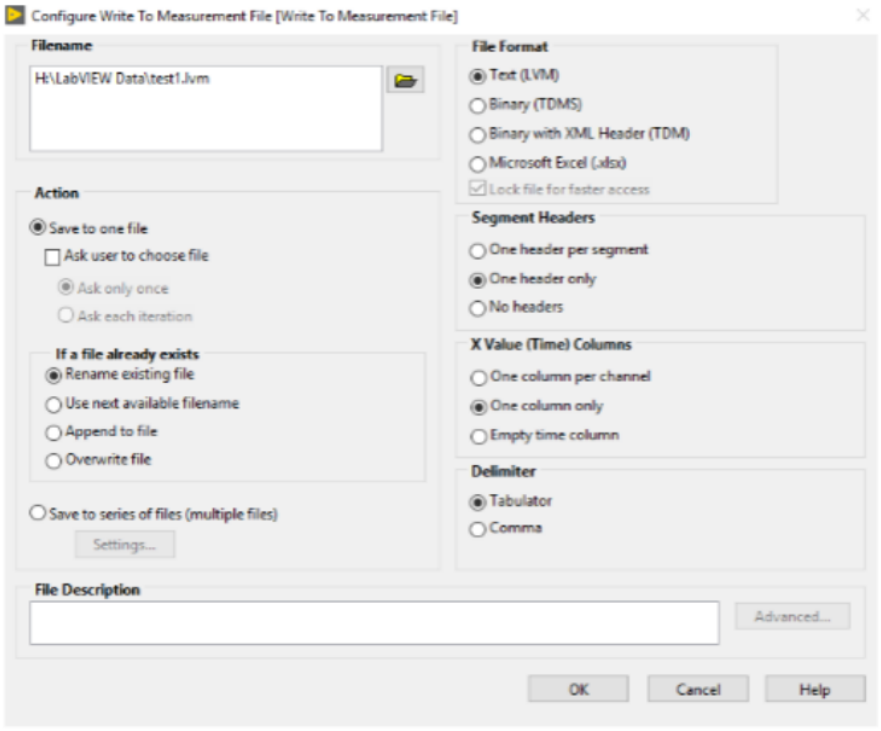
- Load Input (grams)
- Set Load Comparison (grams)
- Load input Under set load
- Load input Equal set load
- Load input Over set load



```
LabVIEW Measurement
Writer_Version  2
Reader_Version  2
Separator   Tab
Decimal_Separator   .
Multi_Headings  No
X_Columns   One
Time_Pref   Absolute
Operator    robert
Date    2021/04/22
Time    16:53:21.2452349662780761719
***End_of_Header***

Channels    1
Samples 1
Date    2021/04/22
Time    16:53:21.2452349662780761719
X_Dimension Time
X0  0.00000000000000000E+0
Delta_X 1.000000
***End_of_Header***
X_Value Untitled    Comment
0.000000    199.882698
0.301011    193.523017
0.601557    199.529382
0.901994    200.525090
1.201496    200.653568
```

Figure 80: Log text files



Text file example is correct and shows the x axis set

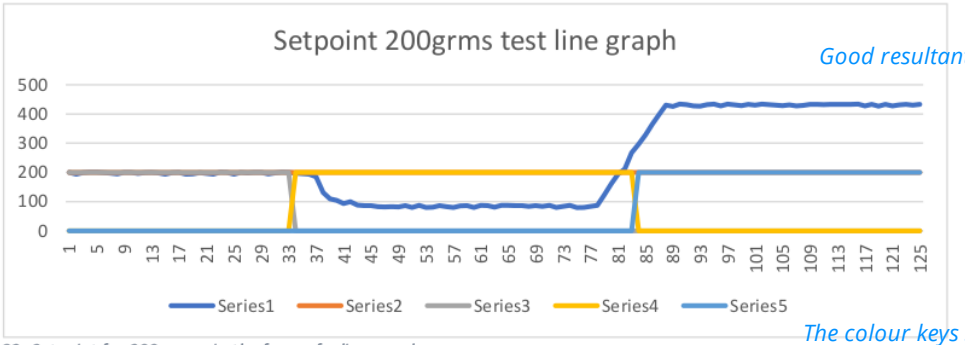
Figure 81: File location save.

5 different text files were saved to a know location so that all data that was recorded was recorded into these text files. The selected file format is a text LVM file which is required to record the data in this case.

4. Testing and Analysis

The below graphs (figure 4, 6 and 8) use setpoints of 200-grams, 500-grams, and 700-grams to determine values which are Less than, Equal to and More than. HMI screens have been designed which correspond to these 3 separate values.

4.1. 200-gram SetPoint

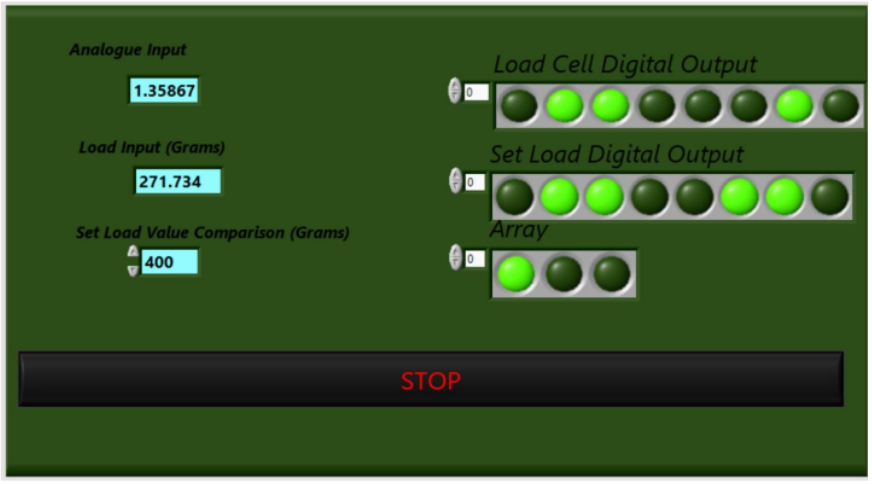


Good resultant graph

The colour keys should be labelled correctly

Figure 82: Setpoint for 200grams in the form of a line graph

Figure 4 clearly shows the logged data comprised into a graph for a Setpoint of 200-grams. The graph legends show the different trends across a timescale of 16 seconds. The graph measures the analogue input, the setpoint, the Less than values, the Equal to value and the More than values. The yellow trend presents the Less than trend. Once the Less than trend climbs to 200grams the equal to and more than values drop to 0. The duration the yellow less than trend is at its optimum value is for 5.2 seconds.



Good test indication on front panel display

Figure 83: HMI screen displaying Less than value.

Figure 5 presents the HMI screen for the Less than value. Also displayed on the screen are the bits outputted which can be seen for the setpoint and digital representation for the analogue input.
An formula can be equated from the data in the table to determine the analogue input ensuring it is correct:

(271.734 / 400) * 2 = 1.35867

4.2. 500-gram Setpoint

Figure 6 shows the data logged for a 500 gram load setpoint. The orange trend labelled as series 2 and is the load input whilst the grey trend or series 3 is the setpoint. The anaolgue input from the potentiometer imediately logs a figure of 500grams which will later switch between 0 – 500.

It can be seen that the blue trend or series 5 is the Equal to value thus when the Equal to value goes high the less than and more values go low and vice versa. At approximately 14.2 seconds the input increases past the setpoint which represents the the more than value which is displayed by the green trend or series 6.

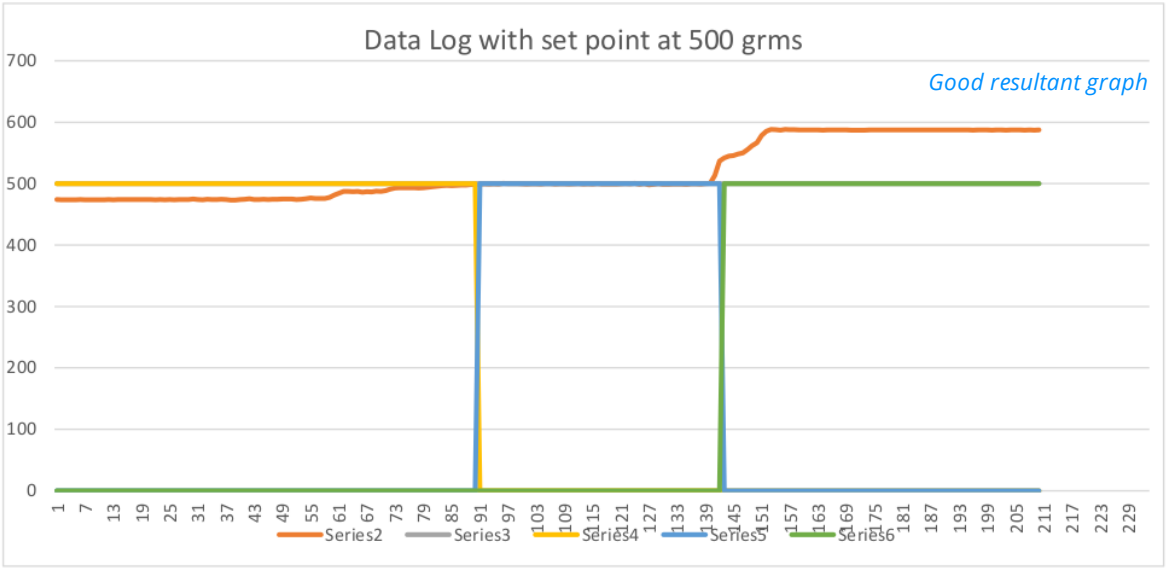


Figure 84: Setpoint for 500-grams in the form of a line graph

Figure 7 shows the interactive HMI screen for the Equal to value

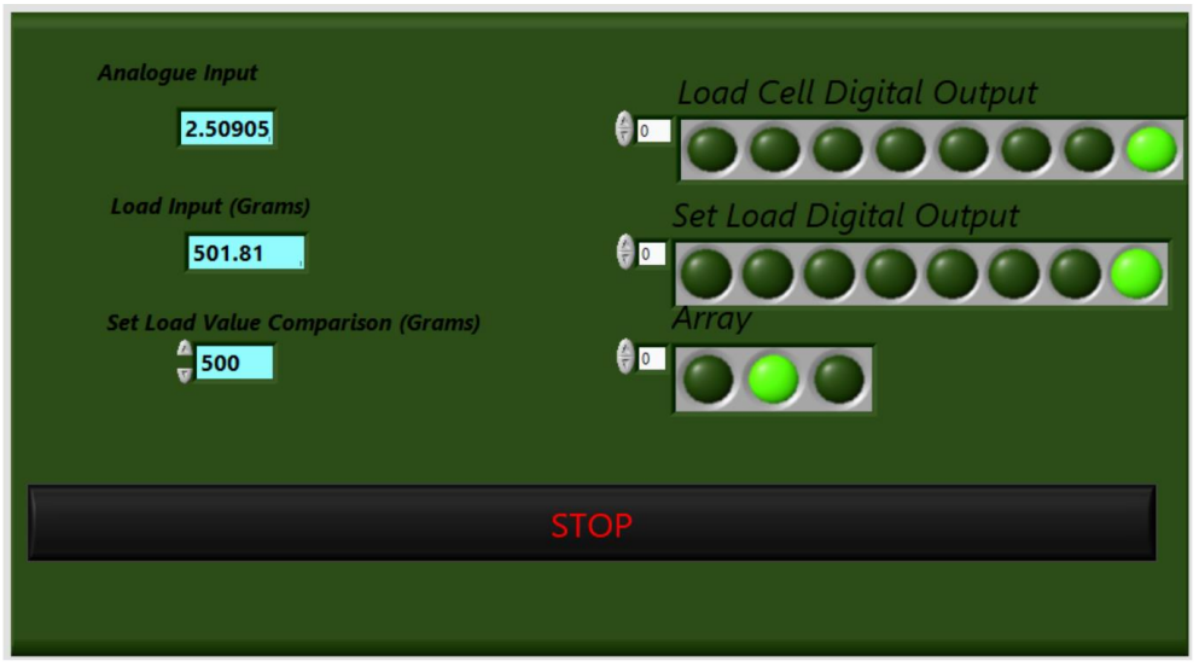


Figure 85: HMI screen displaying equal to value.

4.3. 700-gram Setpoint

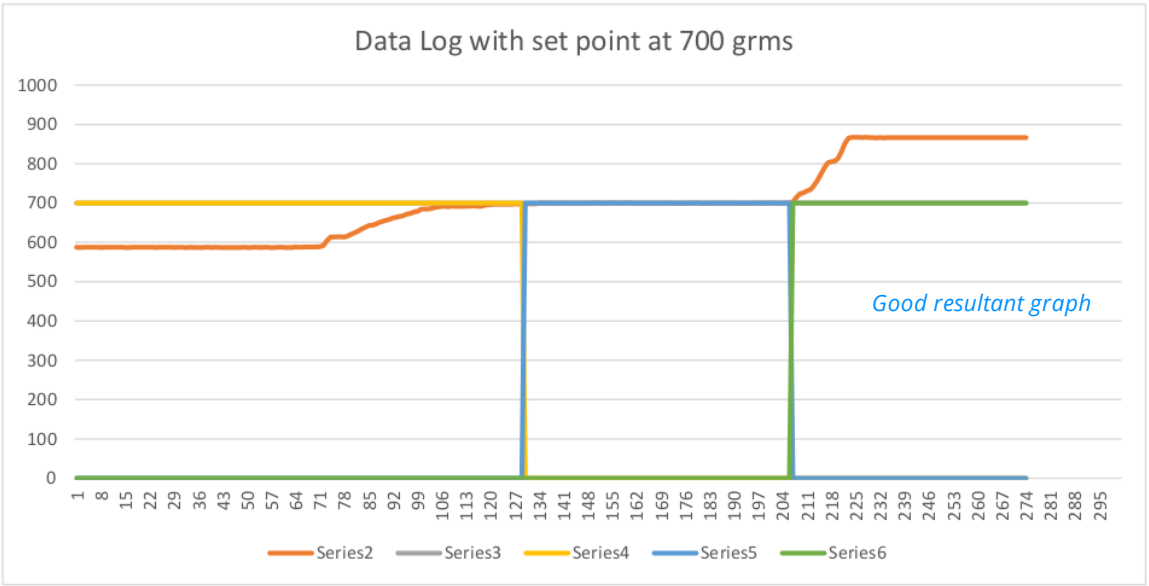


Figure 86: 700-gram Setpoint

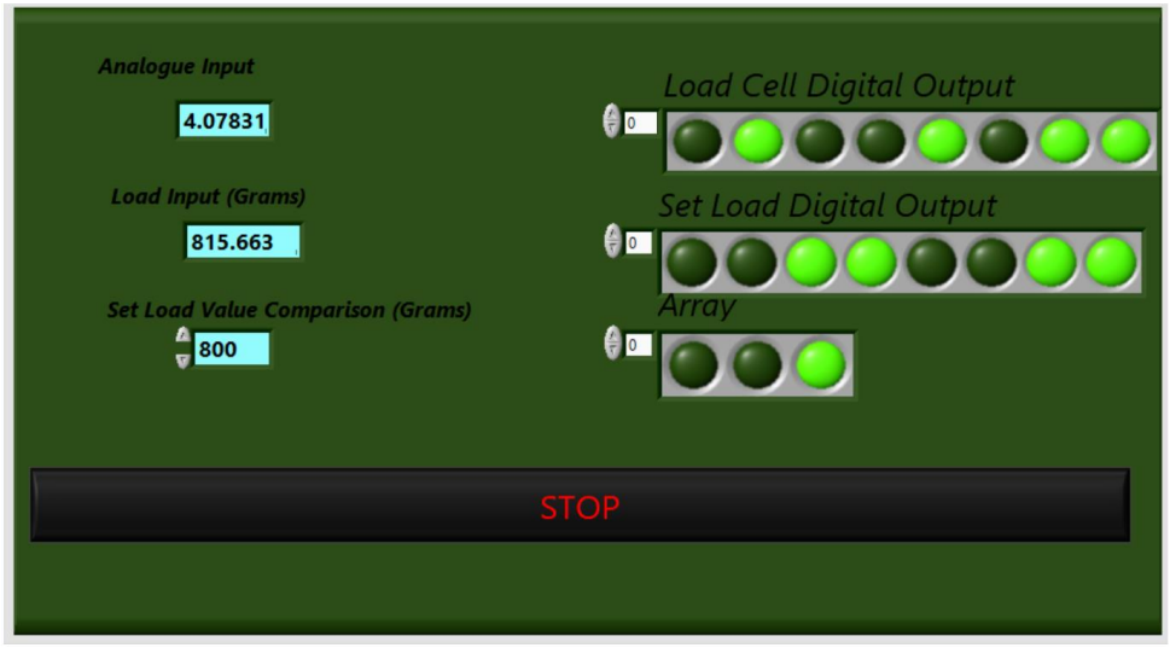


Figure 87: HMI screen displaying More than value.

Figure 9 presents the data logged for the More than setpoint. Results show that the orange trend is the load input which is increased from below the set point to be equal to the grey trend and then more than than the grey trend. The more than trend can be spotted by the green line or series 6, the green more than trend only becomes active once the equal to value drops to 0.

Figure 10 presents the More than value on the HMI screen presented by a green LED.

Graph of resultant outputs and front panel displays are all good.

5. Conclusions and Recommendations

To conclude the results and practical tests carried out proved to be a success. The research that was carried out proved to be a great help when developing the simulations and implementation of boards and software downloads. Through use of the LabVIEW software programs were successfully created to integrate into the electronic boards. Through costruction of the HMI screens visual measures were put in place to analyse real life data whilst putting protocols in place to be able to extract the live data and be able to read it later. It quickly become clear that the data collected from the practical tests needed to be converted into a quicker, easier means of reading and the graphs constructed for the Less than, Equal to and More than values were very useful indeed.
Through use of programming during this report skills have been improved and knowledge can now be taken forward.

Conclusion is reasonable

Appendix

System setup

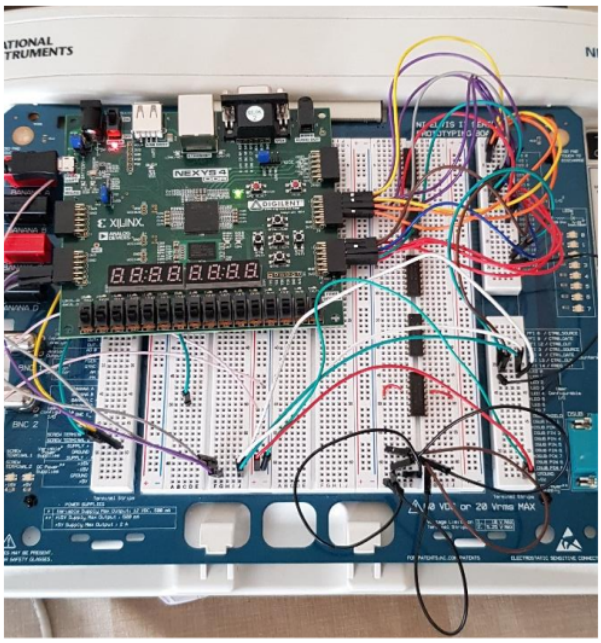
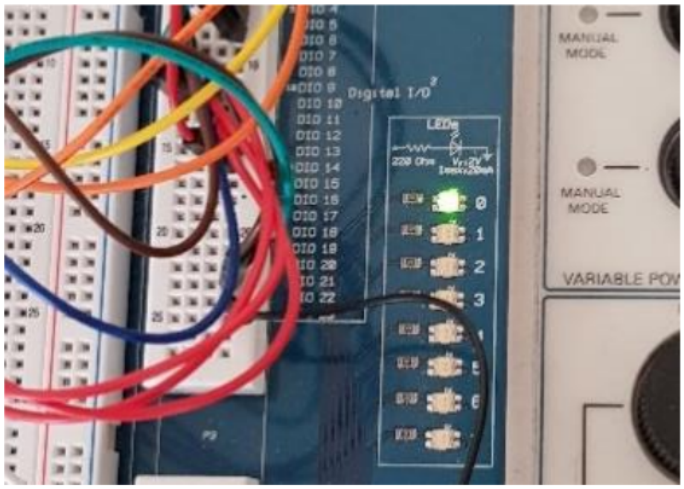


Figure 88: Integrating Nexys board into Elvis II board

Digital output test for < = > conditions.

Less than set point value

Figure 89: Less than Setpoint



Equal to the set point value

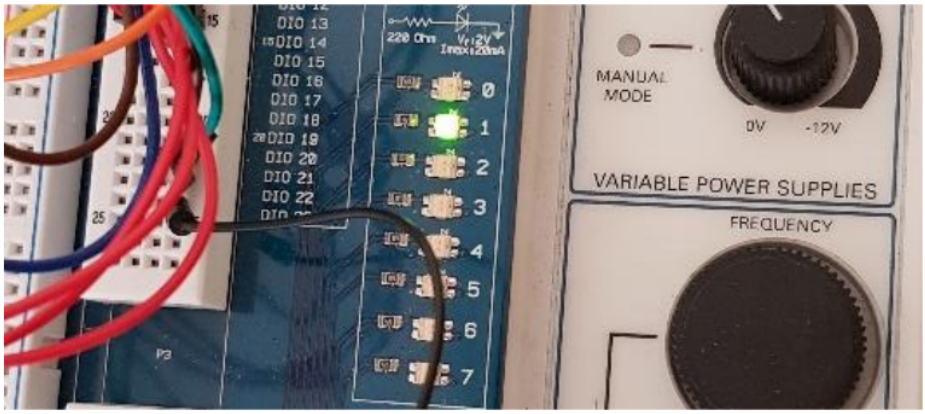
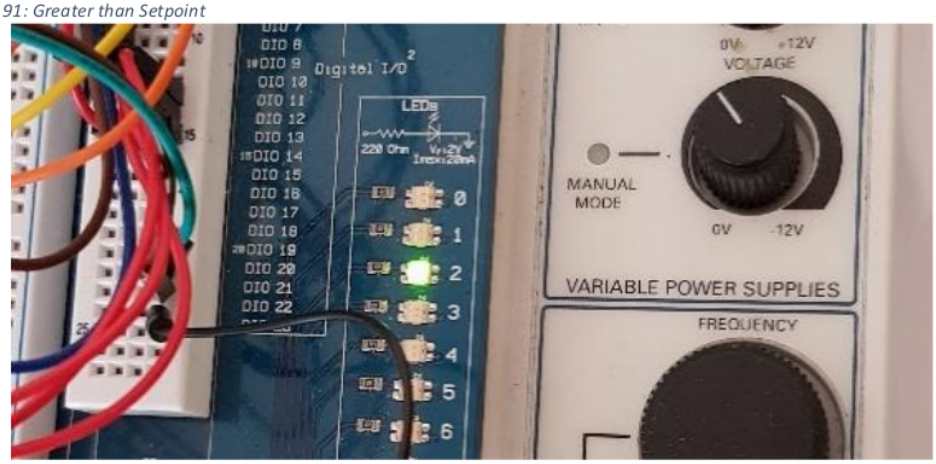


Figure 90: Equal to Setpoint

Greater than set point value

Figure 91: Greater than Setpoint



DAQ board indicating correct results is a good feature in the appendix. This should be referred to from the main report

Excel data for 200-, 500- and 700-gram setpoints

1	LabVIEW Measurement		200	200
2	Writer_Version	2	200	200
3	Reader_Version	2	200	200
4	Separator	Tab	200	200
5	Decimal_Separator	.	200	200
6	Multi_Headings	No	200	200
7	0	183.7908	200	200
8	0.107147	183.7265	200	200
9	0.207926	183.8229	200	200
10	0.309294	183.7908	200	200
11	0.409842	183.8229	200	200
12	0.50999	183.9193	200	200
13	0.611259	183.7265	200	200
14	0.709486	183.8229	200	200
15	0.8105	183.7587	200	200
16	0.912542	183.7265	200	200
17	1.011502	183.855	200	200
18	1.111506	183.8229	200	200
19	1.21239	183.7265	200	200
20	1.310359	183.8229	200	200
21	1.412532	184.0156	200	200
22	1.511514	183.7908	200	200
23	1.611346	183.7265	200	200
24	1.711897	183.6944	200	200
25	1.813167	183.8229	200	200
26	1.912844	183.4375	200	200
27	2.012102	183.9193	200	200
28	2.112832	183.6944	200	200
29	2.214102	183.4375	200	200
30	2.312386	183.6623	200	200
31	2.413445	183.8229	200	200
32	2.51249	183.5981	200	200
33	2.614455	183.8229	200	200
34	2.713403	183.6944	200	200
35	2.814188	183.6944	200	200
36	2.913457	184.0477	200	200
37	3.015642	183.7587	200	200
38	3.115599	184.0156	200	200
39	3.215324	183.7265	200	200
40	3.315586	184.0799	200	200
41	3.415481	183.7587	200	200
42	3.517239	183.9835	200	200
43	3.61625	184.112	200	200
44	3.715731	183.855	200	200
45	3.816001	183.7587	200	200
46	3.916247	184.0156	200	200
47	4.016203	184.4332	200	200
48	4.115799	184.2083	200	200
49	4.216078	184.4974	200	200
50	4.31572	184.1762	200	200
51	4.416274	184.6901	200	200
52	4.516541	184.5617	200	200
53	4.618082	184.5295	200	200
54	4.716671	184.7865	200	200
55	4.817253	185.461	200	200
56	4.919286	185.9107	200	200
57	5.017282	185.7501	200	200
58	5.117579	186.6173	200	200
59	5.217593	186.4567	200	200
60	5.319651	187.1633	200	200

Figure 92: Logged data for 200-gram setpoint

Good example of the raw data from the text files

Figure 93: Data logged for 500-gram setpoint.

1	0	473.929602	500	500	0	0
2	0.103275	473.897482	500	500	0	0
3	0.205554	473.801122	500	500	0	0
4	0.303808	473.833242	500	500	0	0
5	0.406079	473.801122	500	500	0	0
6	0.507355	473.961723	500	500	0	0
7	0.604619	473.865362	500	500	0	0
8	0.70402	473.865362	500	500	0	0
9	0.807147	473.865362	500	500	0	0
10	0.906412	473.897482	500	500	0	0
11	1.005675	473.865362	500	500	0	0
12	1.105972	473.961723	500	500	0	0
13	1.208217	473.865362	500	500	0	0
14	1.309483	473.961723	500	500	0	0
15	1.408774	474.058083	500	500	0	0
16	1.508012	474.315045	500	500	0	0
17	1.60928	474.315045	500	500	0	0
18	1.708569	474.058083	500	500	0	0
19	1.808811	474.025963	500	500	0	0
20	1.909078	474.282925	500	500	0	0
21	2.01035	473.929602	500	500	0	0
22	2.108612	473.865362	500	500	0	0
23	2.20988	474.122324	500	500	0	0
24	2.31015	473.57628	500	500	0	0
25	2.411416	474.154444	500	500	0	0
26	2.510679	473.833242	500	500	0	0
27	2.610945	474.025963	500	500	0	0
28	2.71021	474.315045	500	500	0	0
29	2.810488	474.315045	500	500	0	0
30	2.911746	474.893209	500	500	0	0
31	3.012012	473.993843	500	500	0	0

	A	B	C	D	E	F
1	0.203626	587.282495	700	700	0	0
2	0.301827	587.218254	700	700	0	0
3	0.403073	587.250375	700	700	0	0
4	0.504344	587.282495	700	700	0	0
5	0.602605	587.250375	700	700	0	0
6	0.703874	587.250375	700	700	0	0
7	0.803137	587.346737	700	700	0	0
8	0.906412	587.186134	700	700	0	0
9	1.005684	587.314616	700	700	0	0
10	1.104972	587.378857	700	700	0	0
11	1.206211	587.346737	700	700	0	0
12	1.307481	587.282495	700	700	0	0
13	1.405741	587.282495	700	700	0	0
14	1.506009	587.410978	700	700	0	0
15	1.606275	587.057651	700	700	0	0
16	1.706542	587.218254	700	700	0	0
17	1.807814	587.410978	700	700	0	0
18	1.907073	587.667942	700	700	0	0
19	2.009348	587.764304	700	700	0	0
20	2.107628	587.314616	700	700	0	0
21	2.207875	587.635822	700	700	0	0
22	2.307141	587.667942	700	700	0	0
23	2.407405	587.314616	700	700	0	0
24	2.507674	587.218254	700	700	0	0
25	2.606936	587.571581	700	700	0	0
26	2.708206	587.250375	700	700	0	0
27	2.807469	587.378857	700	700	0	0
28	2.907737	587.507339	700	700	0	0

Figure 94: Data logged for 700-gram setpoint

Mason Morgan

GRADEMARK REPORT

FINAL GRADE

61 /100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

Text Comment. This should be the 8 bit comparator

Text Comment. Content layout and detail is good

PAGE 3

Text Comment. Good list of figures

PAGE 4

Text Comment. Reasonable introduction

PAGE 5

Text Comment. Good explanation with a good example

Text Comment. An IP example would have helped here

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

Text Comment. Good coverage of the DAQ function blocks and the formula block

PAGE 11

PAGE 12

PAGE 13

PAGE 14

Text Comment. Good explanation with relevant sources

PAGE 15

Text Comment. Use capital letter

PAGE 16

Text Comment. Good effort with the explanation of ADC's

PAGE 17

Text Comment. Load cell is explained

PAGE 18

Text Comment. Good information on signal conditioning

PAGE 19

PAGE 20

PAGE 21

Text Comment. The purpose of the half adder and full adder is explained

Text Comment. Hierarchy and inheritance is identified here which are the building blocks of the 4bit adder

PAGE 22

Text Comment. Hierarchy of 4 bit adder is identified

Text Comment. IP design is good

PAGE 23

Text Comment. 4 bit adder IP and code have been created and the designs are in good order.

PAGE 24

Text Comment. IP design is correct and the sources hierarchy has no errors. This is the state required before implementation on the FPGA.

PAGE 25

Text Comment. A good observation that this is another alternative method for less complex circuits.

PAGE 26

Text Comment. This is in the wrong place

PAGE 27

Text Comment. Static conditions for PLD design are identified for the test data

PAGE 28

PAGE 29

Text Comment. Test is correct but should be for more than one value

Text Comment. Test data in test bench is correct

PAGE 30

Text Comment. This test data is not tested

PAGE 31

Text Comment. The test must have a heading.

PAGE 32

PAGE 33

PAGE 34

Text Comment. This should be 8 bit comparator testing

PAGE 35

Text Comment. This is incorrect: LED 0 should be on.

PAGE 36

PAGE 37

Text Comment. This should be 8 bit comparator and two 4 bit comparators

PAGE 38

PAGE 39

Text Comment. A good number of sources

PAGE 40

Text Comment. This should indicate that this section is element 2

PAGE 41

Text Comment. Table should have more detail

Text Comment. Good list of figures

PAGE 42

Text Comment. Good use of tables

PAGE 43

PAGE 44

Text Comment. Good introduction

PAGE 45

Text Comment. Unity gain to show operation of op amp

PAGE 46

PAGE 47

Text Comment. The 10mV range should be referred to here which would justify the gain of 500.

Text Comment. Gain of instrument amp is established

Text Comment. Good table of results from practical test of load sim with instrument amp

PAGE 48

Text Comment. Good linear graph

PAGE 49

Text Comment. Part missing

PAGE 50

PAGE 51

PAGE 52

Text Comment. 10 bit ADC design is good

PAGE 53

Text Comment. 12 bit design is good

PAGE 54

Text Comment. 12 bit ADC test is good showing the max setting is held

PAGE 55

Text Comment. this does not use the POT; this uses a virtual PS as shown

PAGE 56

PAGE 57

PAGE 58

Text Comment. 8 bit ADC tests show the design is working

PAGE 59

Text Comment. FPGA ports are identified

PAGE 60

Text Comment. Spelling

Text Comment. Good image showing equipment setup

PAGE 61

Text Comment. G-code is correct

PAGE 62

Text Comment. This is not accurate

PAGE 63

Text Comment. This block should be explained. This is where the 8 bits are converted to 16 bits by multiplying the first 8 bits by 256. Shift right 8 bits

PAGE 64

PAGE 65

Text Comment. Tests are good

PAGE 66

Text Comment. A reasonablke evaluation

Text Comment. A reasonable evaluation

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

Text Comment. All instrument amplifier tests are present

PAGE 77

Text Comment. DAQ board test indicates the accuracy of the test

PAGE 78

PAGE 79

Text Comment. Table should have more details to point to certain significant tasks.

Text Comment. List of figures is good

PAGE 80

Text Comment. Introduction is good

PAGE 81

Text Comment. Text blocks have been implemented correctly

PAGE 82

PAGE 83

Text Comment. Text file example is correct and shows the x axis set

PAGE 84

Text Comment. Good resultant graph

Text Comment. The colour keys should be labelled correctly

Text Comment. Good test indication on front panel display

PAGE 85

Text Comment. Good resultant graph

PAGE 86

Text Comment. Good resultant graph

PAGE 87

Text Comment. Graph of resultant outputs and front panel displays are all good.

PAGE 88

Text Comment. Conclusion is reasonable

PAGE 89

PAGE 90

Text Comment. DAQ board indicating correct results is a good feature in the appendix. This should be referred to from the main report

PAGE 91

Text Comment. Good example of the raw data from the text files

PAGE 92

PAGE 93

GRADING FORM: CSG-RUBIC

MASON MORGAN	**0**

RESEARCH-INFORMED LITERATURE

Research section has covered all requirements. 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. There are headings missing in many areas. There are some minor errors / confusions in the explanations . Test methods used are good but did not followed a logical sequence. 6/10 E2- There are parts missing for the instruments amp and minor errors in the explanations. The LabVIEW design for the analogue load cell input is explained and the design is a good effort. The 16 bit word operation for the fpga is not explained. The shift right and add parts are missing and this is the key to how the data bits are organised for the FPGA. The design is operational but needs to be explained. 6/10 E3-Dat logging has ben fully explained and a good knowledge of this section is demonstrated. There are minor parts missing. 6/10

18/30

ANALYSIS

E1-Test and results have some errors and some tests are missing. Some parts are not clear as to what is being tested. The evaluation is reasonable. 3/5 E2-The results from LabVIEW are correct . There are some tests which do not make sense. 6/10 E3-Data logging using LabVIEW methods has ben designed and tested for operational requirements. 6/10

15/25

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 board. 3/5 E2-Design and build of LabVIEW interface for FPGA followed a good development sequence. The development sections were tested at specific stages to prove design. The data was shifted right and added successfully. 3/5 E3-Data logging using LabVIEW methods has ben designed and tested for operational requirements. 7/10

13/20

SKILLS FOR PROFESSIONAL PRAC.

Report had some redundant content and headings were missing that caused confusion.

9/15