

2017197

by Aeron Smitham

Submission date: 28-Feb-2023 11:04PM (UTC+0000)

Submission ID: 198186792

File name: 16679_Aeron_Smitham_2017197_181628_315881746.pdf (2.07M)

Word count: 5804

Character count: 29392

Course: HND Electrical and Electronic Engineering
Module: SGEE5007 Virtual Instrumentation and Control
Portfolio: Magnitude Comparator
Name: Aeron Smitham
University number: 2017197
Date submitted: 28/02/2023

Table of Contents

1. Element 1	4
1.1. Introduction	4
1.2. Research.....	4
1.2.1. Digital Design Methods	4
1.2.2. NI DAQmx with LabVIEW	5
1.2.3. Digital Magnitude Comparator	7
1.2.4. Load Cells 1kg.....	7
1.2.5. ADCs	8
1.2.6. Signal Conditioning	8
1.3. Design requirements.....	9
1.3.1. 4 Bit Adder	9
1.3.2. 4 Bit Mag Comp.....	11
1.3.3. 4 Bit Adder & 4Bit Mag Comp Combined	12
1.3.4. 8 Bit Mag Comp.....	13
1.4. Implementation	14
1.4.1. Combined 4 Bit Adder & 4 Bit Mag Comp.....	14
1.4.2. 8 Bit Mag Comp.....	15
1.5. Testing and Analysis.....	16
1.5.1. 4 Bit Adder	16
1.5.2. 4 Bit Mag Comparator.....	17
1.5.3. 4 Bit Adder & Comparator Combination FPGA	18
1.5.4. 8 Bit Mag Comp.....	20
1.6. Evaluation	22
2. Element 2	23
2.1. Introduction	23
2.2. Design Requirements	23
2.2.1. Load Cells	23
2.2.2. Digital Magnitude Comparator	24
2.3. Implementation	26
2.3.1. Load Cells	26
2.3.2. Digital Magnitude Comparator	27
2.4. Testing and Analysis.....	29
2.4.1. Load Cells	29
2.4.2. Digital Magnitude Comparator	31
2.5. Evaluation	34

3.	Element 3	35
3.1.	Introduction	35
3.2.	Design Requirements	35
3.3.	Implementation	37
3.4.	Testing and Analysis.....	38
3.5.	Conclusions	39
4.	References	40
5.	Appendix	41

Good detailed table of contents

List of figures should be included

1. Element 1

1.1. Introduction

In this element, an investigation into VHDL and IP integration is to be held. A system is to be built based on a packing machine which will use VHDL and IP integration with an FPGA to create a 4-Bit Adder and 4-Bit Comparator system.

1.2. Research

1.2.1. Digital Design Methods

An FPGA (Field-Programmable Gate Array) is an integrated circuit (IC) that can be programmed after it has been built. FPGAs are used for digital circuits and can be reprogrammed to implement new circuits and systems as needed.

An FPGA consists of an array of configurable logic blocks (CLBs) and programmable interconnects. The CLBs can be programmed to implement logic functions such as AND, OR, and NOT gates, as well as more complex functions such as counters and shift registers. The interconnects allow the CLBs to be connected to implement more complex circuits.

One of the key advantages of FPGAs is their flexibility. Unlike Application Specific Integrated Circuits (ASICs), which are specific to one function, FPGAs can be reprogrammed multiple times to implement various circuits and systems. This flexibility makes them ideal for prototyping and testing digital systems (FPGA Tutorial, 2020).

In this portfolio, the FPGA used will be an Artix 7. A sample of the VHDL code used can be seen in figure 1 and an IP circuit created can be seen in figure 2.

A good explanation of capabilities relevant to the requirements. Good supporting examples

```
36 U1 : fulladd
37 port map(
38   a => a0, b => b0, cin => cin, cout => c1,
39   s => s0);
```

Figure 1

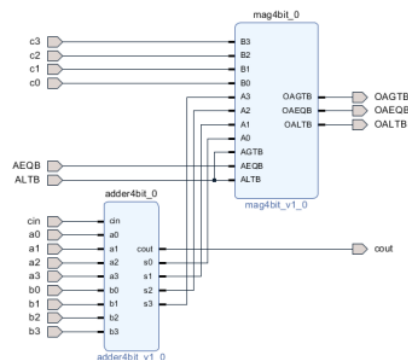


Figure 2

1.2.2. NI DAQmx with LabVIEW

NI-DAQmx is a software architecture developed by National Instruments (NI) that provides a common programming interface for data acquisition and control hardware. The software is used to interface with various types of sensors, instruments, and other hardware devices to acquire and control analogue and digital signals in real-time (National Instruments, 2022).

Figure 3 shows the create channel icon which can be used to measure digital and analogue IO, among other things. The minimum and maximum voltage levels can be set to suit. This function can be used to create a channel from an external source, such as a virtual power supply through LabVIEW and the ELVIS II DAQmx board.

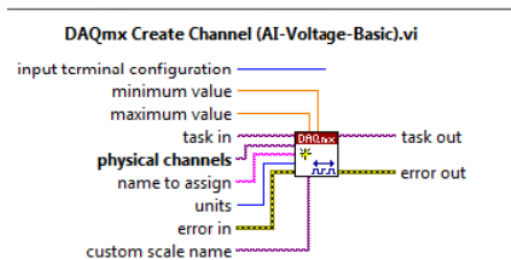
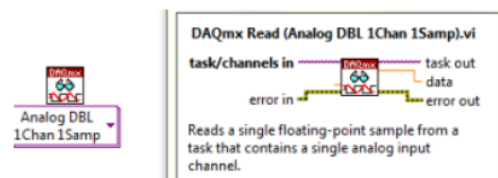


Figure 3

Figure 4 is the read version of the above function. This icon can read digital and analogue IO.



Good explanation with relevant supporting examples

Figure 4

Figure 5 shows the write function, which can be sued to write digital and analogue values to outputs for external devices.

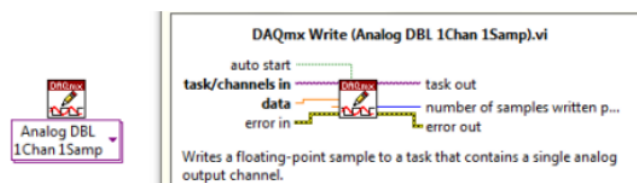


Figure 5

Figure 6 is a formula block which allows data from input signals to be converted to suit various requirements where a mathematical equation is needed – such as when to scale a signal.

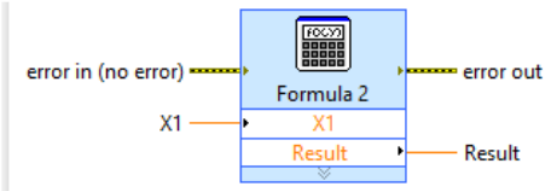


Figure 6

There are more critical parts to the DAQ that could have been included, and an example system could have enhanced the research findings. However, there are good significant parts covered.

1.2.3. Digital Magnitude Comparator

A digital magnitude comparator is a circuit that compares binary numbers and determines if they are equal or if one is greater than the other. The comparator outputs a binary signal indicating the result of the comparison. The magnitude comparator can be used in various applications, including numerical and scientific computations, digital signal processing, and computer arithmetic.

In digital electronics, the magnitude comparator is usually implemented using digital logic gates, such as AND, OR, and NOT gates, or using specialized integrated circuits (ICs) designed specifically for this purpose. The operation of the comparator is based on the comparison of the individual bits in the binary numbers, starting from the most significant bit (MSB) to the least significant bit (LSB). *Good explanation that differentiates a digital comparator*

The digital magnitude comparator is a fundamental building block in digital systems and plays an important role in many digital circuits, such as adders, subtractors, and binary search algorithms (Electronics Tutorial, 2018). *Good explanations, however, a supporting image example could have been included.*

1.2.4. Load Cells 1kg

A load cell is a type of transducer that is used to convert a force into an electrical signal. Load cells are widely used in a variety of applications, including weighing scales, process control, and force measurement.

A load cell typically consists of a metal element, such as a beam, that deforms when a force is applied to it. The deformation is proportional to the applied force, and is typically measured using strain gauges. Strain gauges are resistive elements that change their resistance in response to the applied force, causing a change in the electrical resistance of the load cell.

The electrical resistance change is then amplified and converted into a voltage or current signal, which can be read and processed by electronic instrumentation. The electronic instrumentation can be used to display the weight or force measurement, or to control other components in a system.

Load cells are available in many different shapes, sizes, and capacities, and are made from various materials, including stainless steel, aluminium, and titanium. They can be used in single-point or multi-point configurations, and are often used in combination with other components, such as amplifiers and signal conditioners, to provide a complete measurement system (Prielipp, 2019).

Good significant points included in the explanation. A supporting image as an example could have been included.

1.2.5. ADCs

An analogue to digital converter (ADC) is a device that changes an analogue signal, i.e., a current or voltage, into a digital the signal. This allows analogue signals to be analysed and processed by digital systems, such as computers and microcontrollers.

An ADC works by measuring the analogue input signal at intervals in time and encodes the resulting values into a digital format. The conversion process is usually performed in several stages, including analogue signal amplification, analogue to digital conversion, and signal conditioning.

ADCs are used in a variety of applications, including: data acquisition, communication systems and control systems. They are available in many different forms, including integrated circuits (ICs), standalone units, and modular components.

ADCs are characterized by several key performance parameters, such as resolution, sample rate, and accuracy. Resolution determines the number of bits in the digital output and thus the number of quantization levels in the signal. The sample rate determines the number of samples per second that can be taken, and the accuracy determines the degree to which the ADC accurately represents the analogue signal (Maloy-Smith, 2020).

Good explanation that could be improved with a supporting image

1.2.6. Signal Conditioning

Signal conditioning refers to the process of preparing an analogue signal for conversion into a digital signal via an ADC. The goal of signal conditioning is to improve the accuracy and quality of the analogue signal, so that it can be reliably converted into a digital signal.

Signal conditioning typically involves several steps, including: amplification, filtering, scaling. The signal conditioning process may also include additional steps, such as converting the signal from one form to another, like voltage to current, or adjusting the signal's frequency or phase.

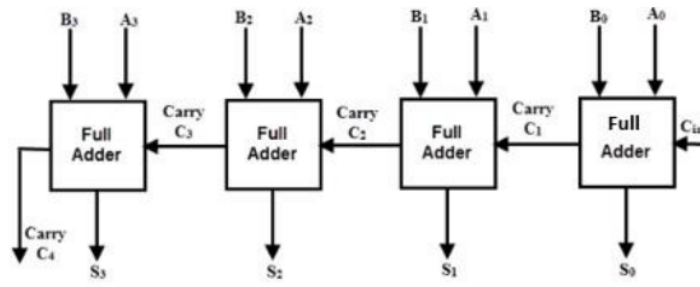
Signal conditioning is a critical step in the data acquisition process, as it ensures that the analogue signal is reliably and accurately converted into a digital signal that can be used for control and analysis (Dewesoft, 2021).

A block diagram of the stages described here should have been included.

The research contains good relevant points and sources, and is well balanced for each area. Supporting images / diagrams would have enhanced your explanations.

1.3. Design requirements

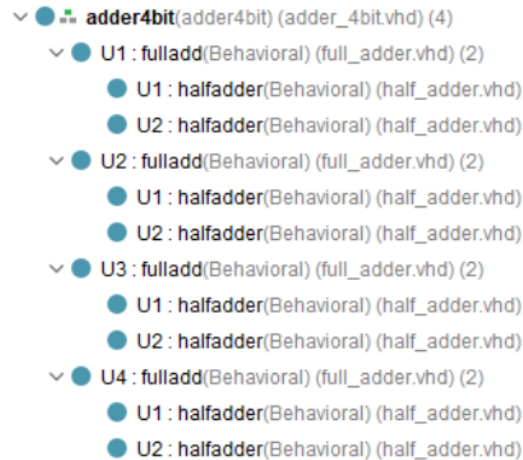
1.3.1. 4 Bit Adder



Good block diagram with all the components required for the VHDL development

Figure 7

Figure 7 is the block diagram for a 4-Bit Adder, showing 4 Full Adders and their respective inputs and outputs. The inputs are made up of: A_x , B_x and a 'Carry' C_{in} , while there are two outputs each: a 'sum' S_x and a 'Carry' C_x .



Good identification of hierarchy of building blocks for the VHDL

Figure 8

Figure 8 is the hierarchical model for a 4-Bit Adder showing how it is connected, with 4 Full Adders which each have 2 Half Adders.

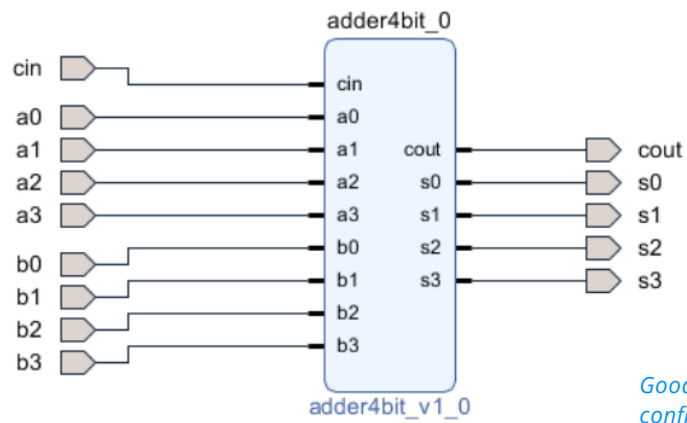
```

36 U1 : fulladd
37   port map(
38     a => a0, b => b0, cin => cin, cout => c1,
39     s => s0);
    
```

Figure 9

Figure 9 is the VHDL code for the first Full Adder in a 4-Bit Adder, signified as U1. All of its connections can be seen to be the first, with its 'a', 'b' and 's' being '0s', 'cin' is the first carry in, while the first carry out is 'c1'.

Good explanation and sample



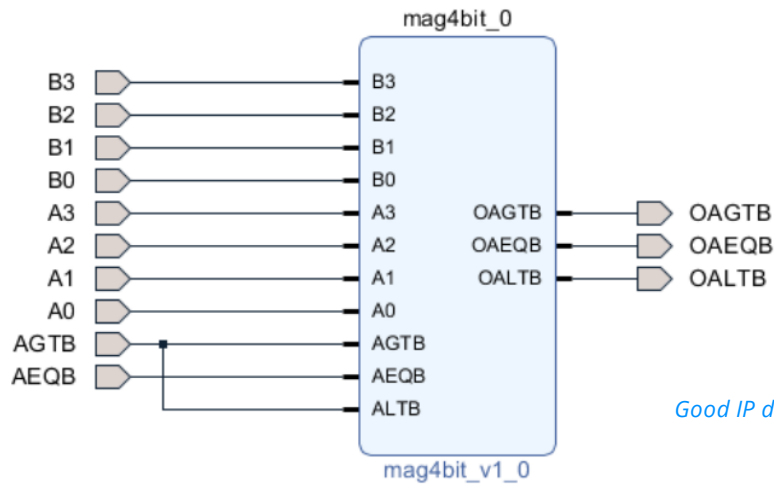
Good IP design with io configuration

Figure 10

Figure 10 shows the IP circuit generated from the VHDL code.

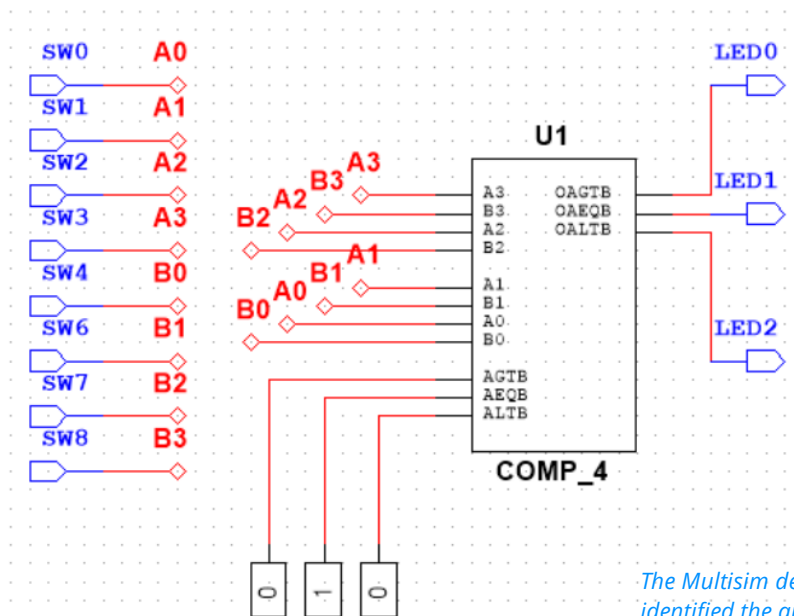
The program explanation should be included and the program listing should be in the appendix

1.3.2. 4 Bit Mag Comp



Good IP design with io configuration

Figure 11



The Multisim design is good and you have identified the auto generated VHDL that was taken and used in the Vivado ADE.

Figure 12

A Multi-Sim was created, figure 12, and the VHDL code was copied from this and inserted into the Vivado software, figure 11.

The IP design can be generated from the VHDL should have been pointed out.

1.3.3. 4 Bit Adder & 4Bit Mag Comp Combined

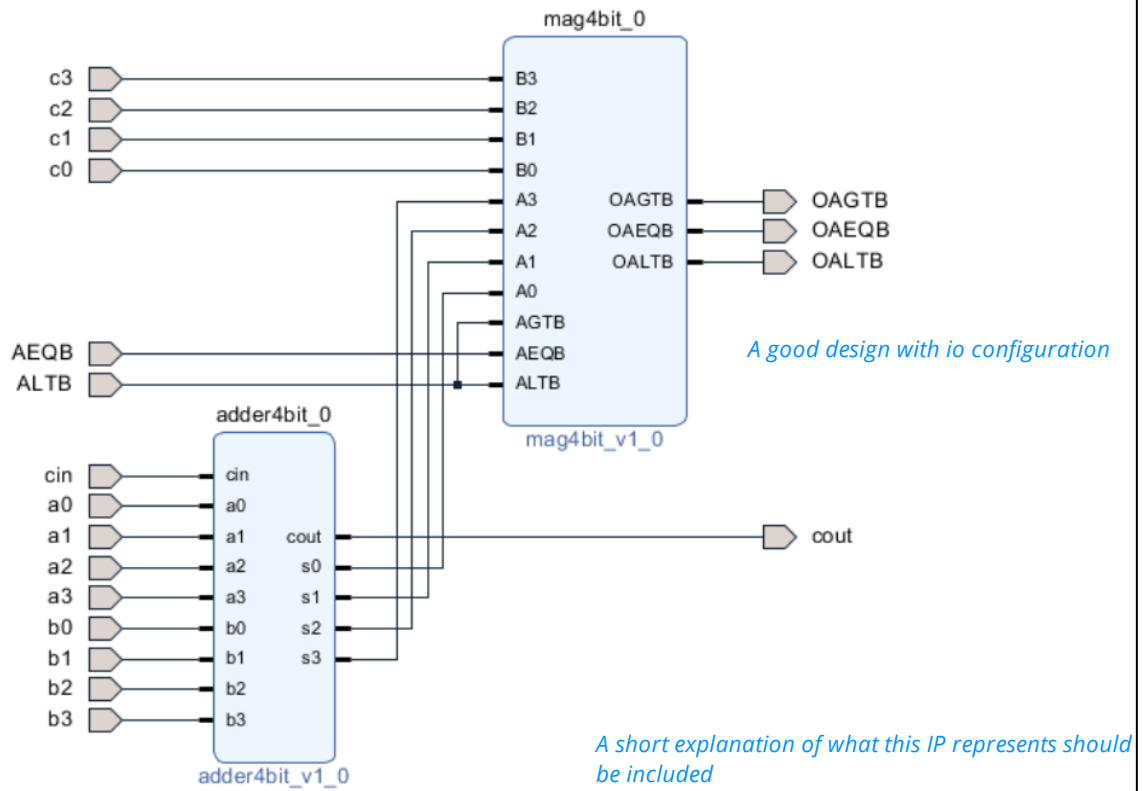
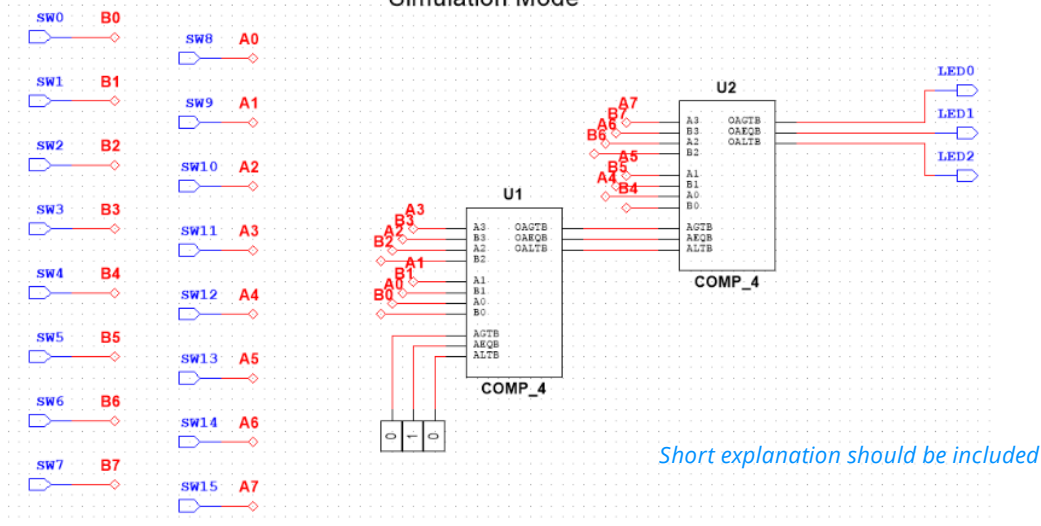


Figure 13

1.3.4. 8 Bit Mag Comp

8_BIT MAGNITUDE COMPARATOR Simulation Mode



Short explanation should be included

Figure 14

1.4. Implementation

1.4.1. Combined 4 Bit Adder & 4 Bit Mag Comp



Figure 15

In figure 15 the combined 4-Bit Adder and 4-Bit Magnitude Comparator on the NEXYS 4 FPGA can be seen. Testing is carried out via the switches and LEDs, which have been assigned in the code. LEDs 0, 1 and 2 represent 'A greater than C', 'A equal to C' and 'A less than C', respectively. The state of these LEDs is controlled by the switches. From right to left, the first four switches are the 4-Bit A's, followed by the 4-Bit Bs, then the 4-Bit Cs and the last four switches are settings for the FBGA. The setting switches can be set up to add the 4-Bit A and B together, and to compare them against the 4-Bit C. This comparison between A and B with C then controls the state of the 0, 1 and 2 LEDs. LED 15 is on when the binary value is above 15.

FPGA

Explanation is good and demonstrates a good understanding of the function

1.4.2. 8 Bit Mag Comp



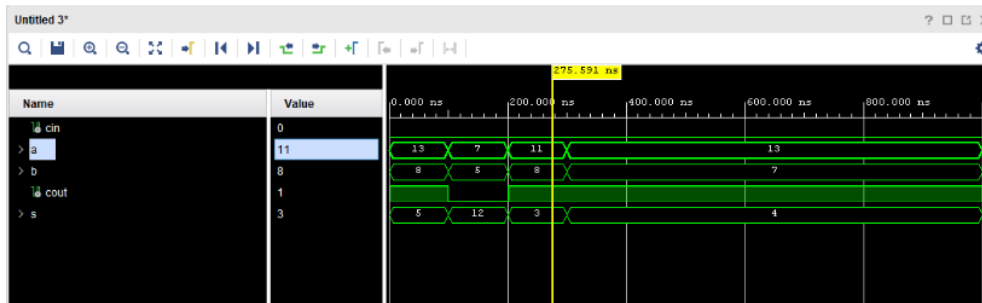
Figure 16

The 8-Bit Magnitude Comparator FPGA can be seen in figure 16. It works in much the same way as the 4-Bit Adder and Magnitude Comparator, but an 8-Bit A is compared with an 8-Bit B, and LEDs 0, 1, and 2 show if 'B is greater than A', 'B is equal to A' and 'B is less than A', respectively.

Explanation includes the significant points of the system

1.5. Testing and Analysis

1.5.1. 4 Bit Adder



Good clear
image of test
results

Figure 17

'A' and 'B' are 4-Bits – meaning the total value for each can be up to 15 and as you are adding 'A' and 'B' together, their total can be greater than 15. In this situation, 'cout' will become true – signifying the sum of the addition is greater than 15. In figure 17 at the point marked with the yellow line, we can see that 'As' decimal value is 11 and 'Bs' is 8, and that 'cout' had become true.

Very good explanation as the 'cout' state is significant

Figure 19 shows each bits value at the same time as figure 17. While figure 18 shows the code for the waveforms in figure 17.

cin	0	Logic
a0	1	Logic
a1	0	Logic
a2	1	Logic
a3	1	Logic
b0	1	Logic
b1	1	Logic
b2	1	Logic
b3	0	Logic
cout	1	Logic
s0	0	Logic
s1	0	Logic
s2	1	Logic
s3	0	Logic

Figure 19

```
59 stimuli : process
60 begin
61     -- EDIT Adopt 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 100ns;
72     cin <= '1';
73     a0 <= '1';
74     a1 <= '1';
75     a2 <= '1';
76     a3 <= '0';
77     b0 <= '1';
78     b1 <= '0';
79     b2 <= '1';
80     b3 <= '0';
81     wait for 100ns;
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 100ns;
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 100ns;
102    -- EDIT Add stimuli here
```

Good range of test data used

Figure 18

1.5.2. 4 Bit Mag Comparator

Good clear test results

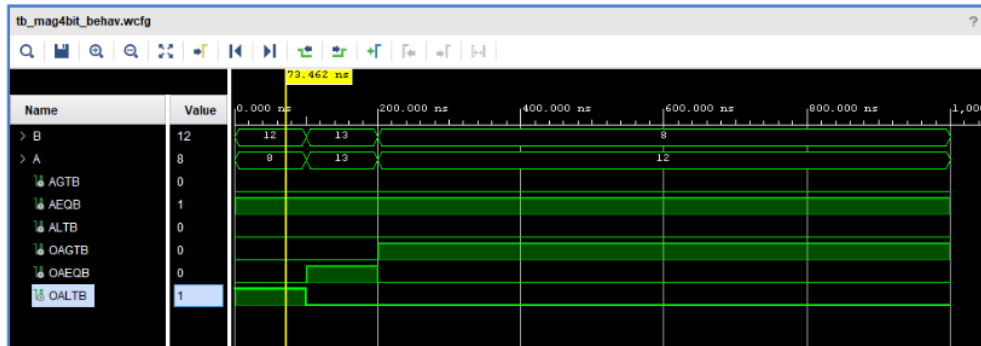


Figure 20

In figure 20 we can see the three different states, with 'A' being less than 'B', then the two being equal and then 'A' greater than 'B'. The waveforms were generated from the test bench code below in figure 21.

B3	1	Logic
B2	0	Logic
B1	0	Logic
B0	0	Logic
A3	1	Logic
A2	1	Logic
A1	0	Logic
A0	0	Logic
AGTB	0	Logic
AEQB	1	Logic
ALTB	0	Logic
OAGT	1	Logic
OAEQ	0	Logic
OALT	0	Logic

Figure 22

```

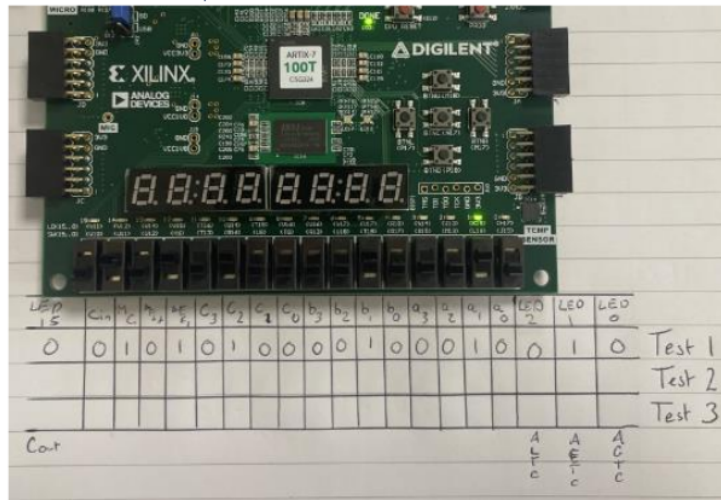
59 stimuli : process
60 begin
61     -- EDIT Adapt initialization as needed
62     B3 <= '1';
63     B2 <= '1';
64     B1 <= '0';
65     B0 <= '0';
66     A3 <= '1';
67     A2 <= '0';
68     A1 <= '0';
69     A0 <= '0';
70     AGTB <= '0';
71     AEQB <= '1';
72     ALTB <= '0';
73     wait for 100ns;
74     B3 <= '1';
75     B2 <= '1';
76     B1 <= '0';
77     B0 <= '1';
78     A3 <= '1';
79     A2 <= '1';
80     A1 <= '0';
81     A0 <= '1';
82     AGTB <= '0';
83     AEQB <= '1';
84     ALTB <= '0';
85     wait for 100ns;
86     B3 <= '1';
87     B2 <= '0';
88     B1 <= '0';
89     B0 <= '0';
90     A3 <= '1';
91     A2 <= '1';
92     A1 <= '0';
93     A0 <= '0';
94     AGTB <= '0';
95     AEQB <= '1';
96     ALTB <= '0';
97     wait for 100ns;
98     -- EDIT Add stimuli here

```

Good range of test data

Figure 21

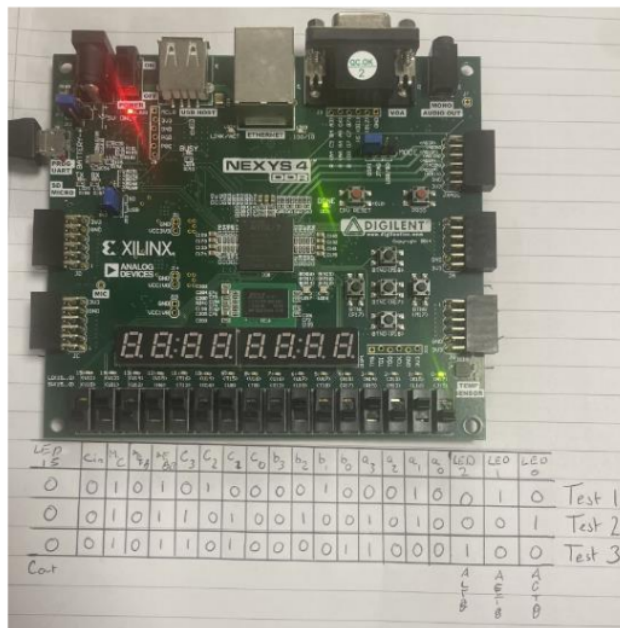
1.5.3. 4 Bit Adder & Comparator Combination FPGA



Good evidence of the 'equal' test state

Figure 23

Test 1 can be seen in figure 23, the comparison between A&B and C shows that LED 1, for A is equal to C is on. This is because the total decimal value of A is 4 – as A₁ is 2 and B₁ is 2 – and C has a value of 4.



Good test evidence

Figure 24

Test 2 can be seen in figure 24 and shows that A is greater than C – as LED 0 is on. This is due to A&B have a decimal value of 10, and C also has a decimal value of 6.

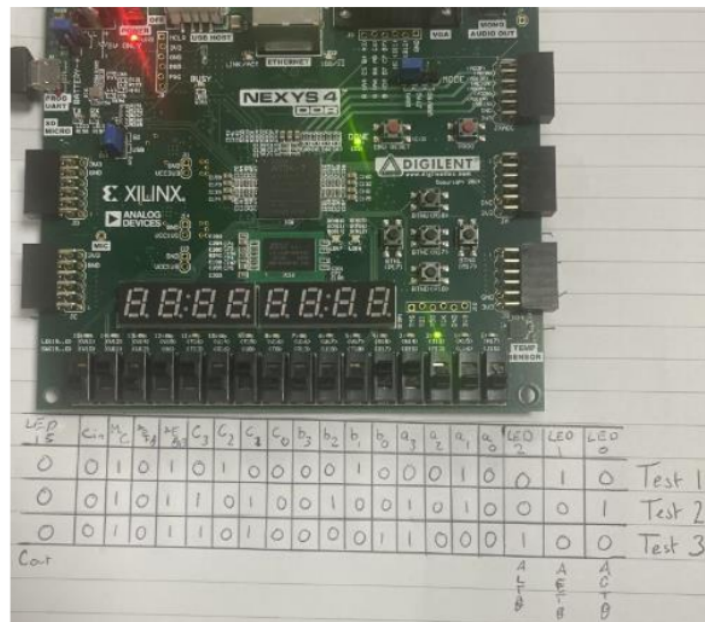


Figure 25

Test 3 can be seen in figure 25, the comparison shows A is less than C, as C has a decimal value of 10 while A has a value of 9.

All 3 test states have been successful

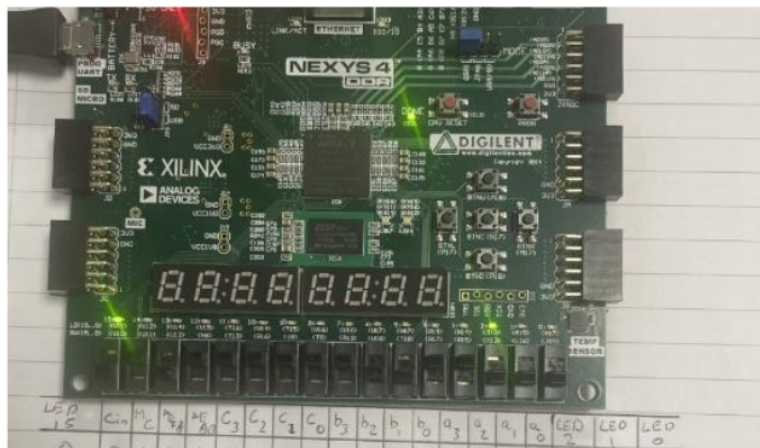
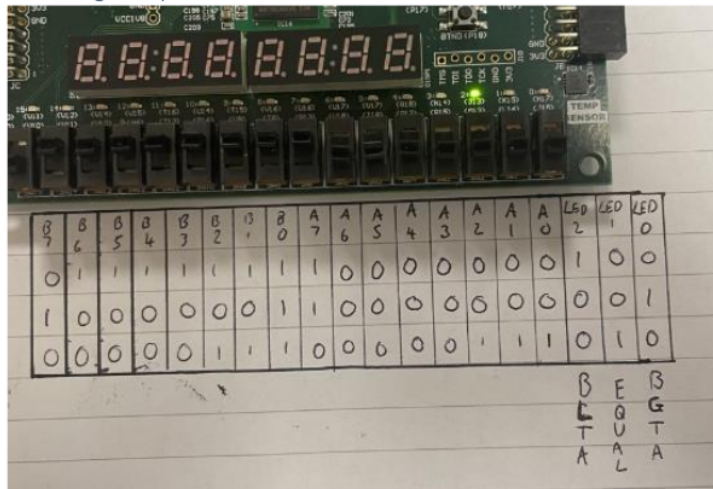


Figure 26

An additional test was carried out in figure 26 to show the Cout LED come on – LED 15. This is because the total of A is 17, above the maximum of 15 of the 4-Bit. In instances like this, the Cout LED will be on to signify a carry bit.

The 'cout' state is significant and is tested successfully here. This supports the explanation earlier in the implementation

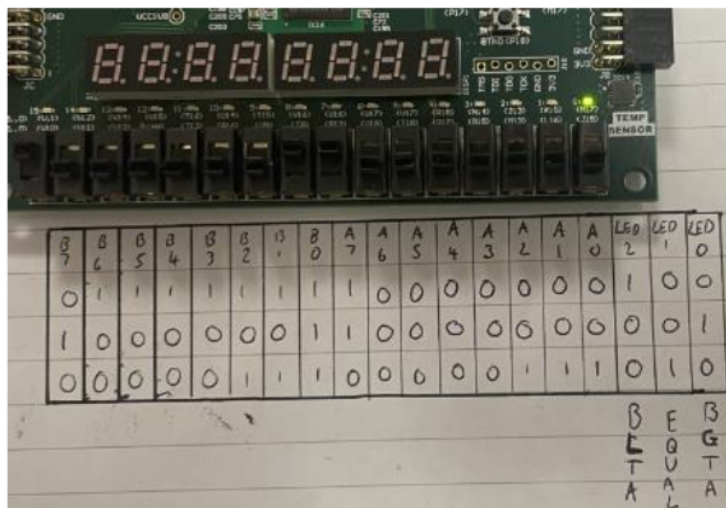
1.5.4. 8 Bit Mag Comp



Good test evidence

Figure 27

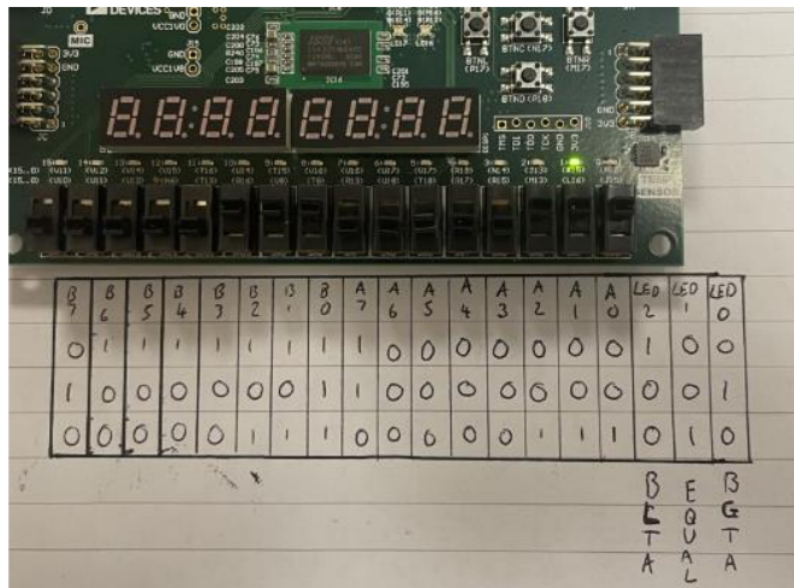
Test 1 can be seen in figure 27, the comparison between A and B shows that LED 2, for B is less than A is on. This is because the total decimal value of A is 128 – as A₇ is high – and B has a value of 127, as B₀ to B₆ are high.



Good test evidence

Figure 28

Test 2 can be seen in figure 28, the comparison between A and B shows that LED 0, for B is greater than A is on. This is because the total decimal value of A is 128 – as A₇ is high – and B has a value of 129, as B₀ and B₇ are high.



All 3 test states are identified and explained

Figure 29

Test 3 can be seen in figure 29, the comparison between A and B shows that LED 1, for B is equal to A is on. This is because the total decimal value of A is 7 – as A₀ to A₂ are high – and B also has a value of 7, as B₀ to B₂ are high.

1.6. Evaluation

FPGA

In this element, the VHDL prototypes were used in conjunction with the FBGA boards to generate a result from the IP system. The VHDL code was created through various means, such as writing the code, the test bench and through the Multi-Sim PLD. This code was then downloaded to the FPGA for testing.

The digital design and results of the 4-Bit Adder and 4-Bit Magnitude Comparator proves the reliability of the FPGA output data and the IP design can be generated and implemented for any system using interfacing.

The 8-Bit Comparator is designed and tested with the FPGA and the outputs are reliable. The 8-Bit Comparator system can be implemented with LabVIEW and DAQmx using I/O ports on the FBGA and DAQ-mx.

FPGA

If the input values into the FPGA are steady, an accurate and reliable output will be generated.

Evaluation covers significant points relevant for the next element

2. Element 2

2.1. Introduction

This element is focussed on the use of LabVIEW to design an interface for the FPGA that contains the 8-Bit Magnitude Comparator developed in Element 1, via the NI ELVIS II board.

This interface will then compare 2 8-Bit digital values to find if they are equal, less than or greater than one another. This will be displayed on the interface and FPGA via LED indicators which will be assigned via the PMOD ports.

Intro should include purpose of the load cell

2.2. Design Requirements

2.2.1. Load Cells

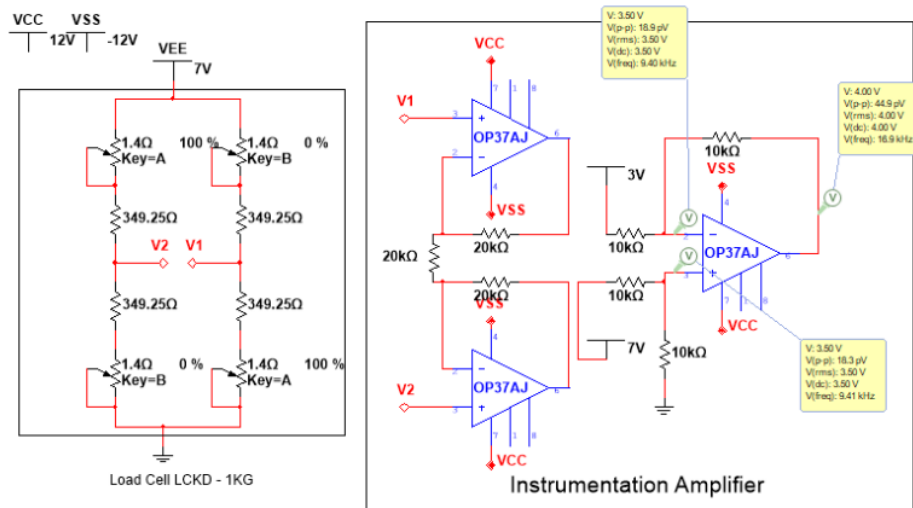


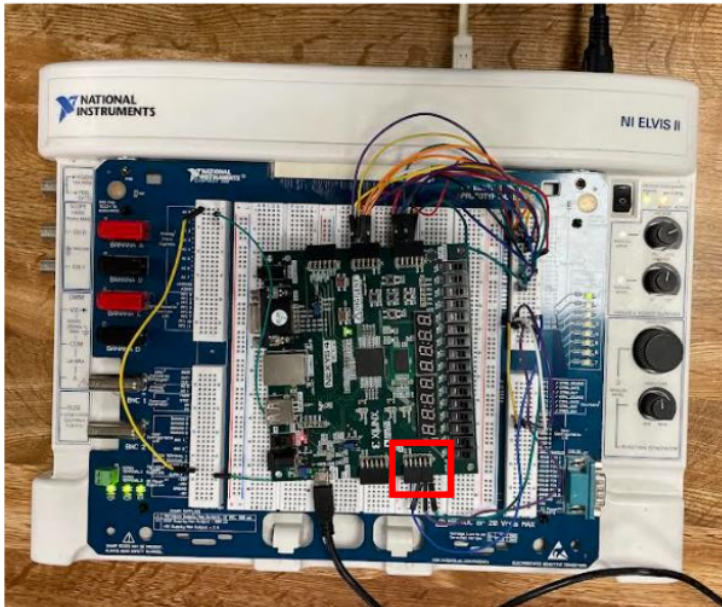
Figure 30

Good example of how the inst amp works using unity gain

As can be seen in figure 30, the difference between the positive and negative voltage into the OP37AJ can be seen on the output indicator showing 4V which is the difference between the voltages with a unity gain of 1.

The significance of the inst amp should be pointed out here!

2.2.2. Digital Magnitude Comparator



Good explanation of the interfacing technique

Figure 31

Figure 31 shows the set-up of the magnitude comparator FPGA and the NI ELVIS II board. The boards are connected together via the numerous cable links. This allows the comparison to be available and also the LEDs of the ELVIS board to light when the comparison is in one of its three states: equal too, less than or greater than. The PMOD ports for the LEDs can be seen in the red square in figure 31 and are pins JC 0-3. The PMOD connectors and pin assignments can be seen in figure 32, with PMODs JA, JB and JC used for this task.



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

Table 5. Nexys4 DDR Pmod pin assignments.				
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)

Figure 32

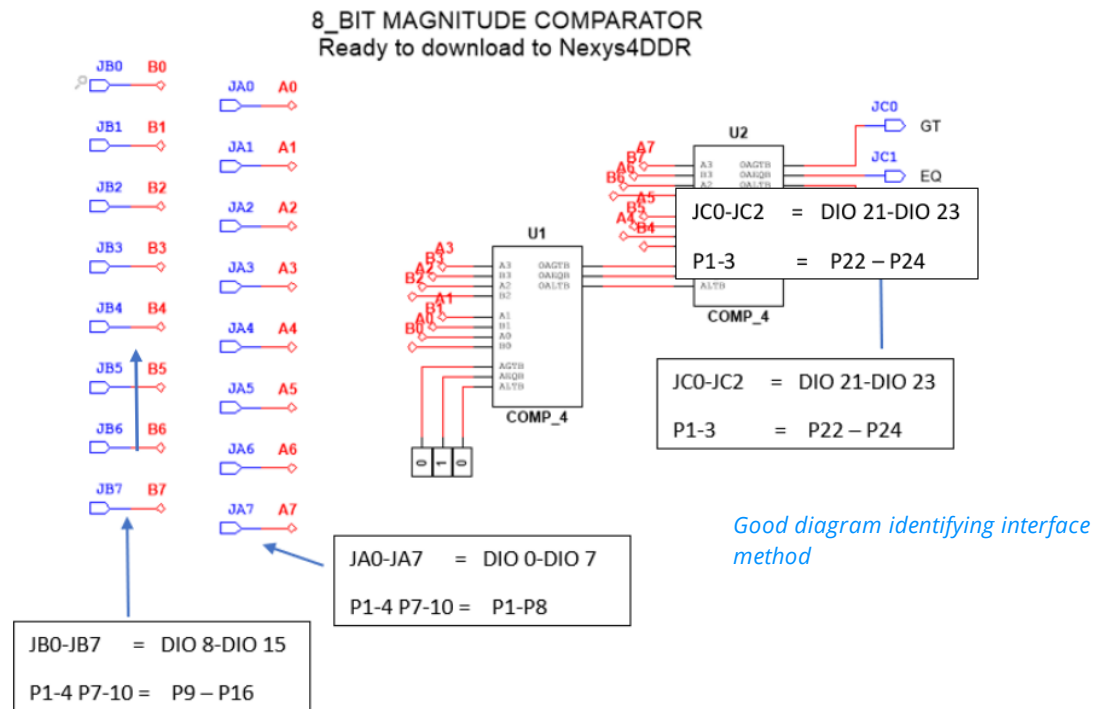


Figure 33

Figure 33 shows the FPGA connections and has their *the* PMOD port pin allocations and their *the* relevant ELVIS board connections labelled.

2.3. Implementation

2.3.1. Load Cells

In figure 34 the load cell has been added to the instrument amplifier. The unloaded load cell is at 0g and 0V as all four resistors are at 50%. To get a fully loaded load cell of 1000g and 5V, the top left and bottom right must be at 100% and the top right and bottom left must be at 0%, or vice-versa. Increasing and decreasing each respective set by 5% will change the voltage, and therefore the load in grams.

Good explanation of function procedure required

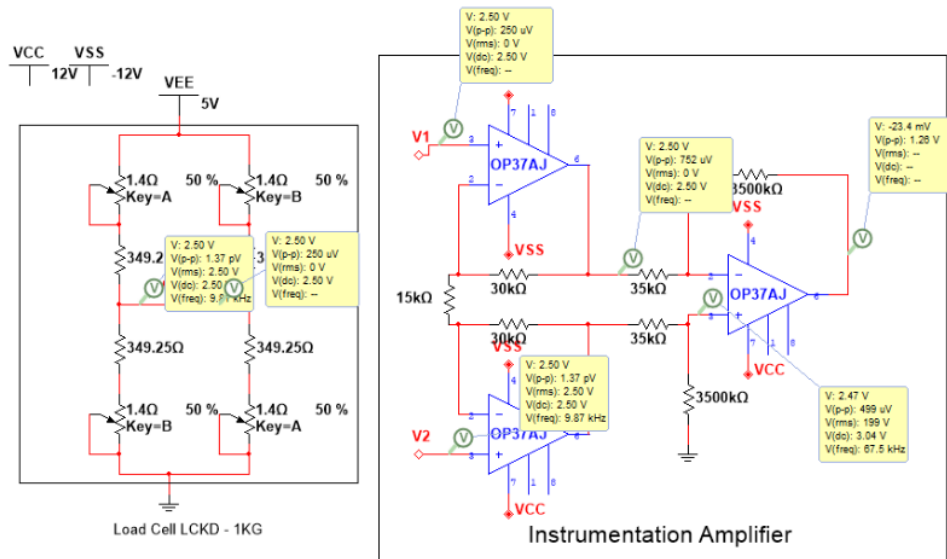


Figure 34

2.3.2. Digital Magnitude Comparator

As part of the implementation of the digital magnitude comparator, prototypes were designed to help gain an understanding of using LabVIEW.

In figure 35, the interface view can be seen on the left and the block view can be seen on the right. The block view shows how the system works, with a slider to be used as an analogue input and a display box showing the analogue value connected to the formula block which outputs to another display box with the Boolean equivalent and to the green indicators.

Figure 35 shows the analogue slider to be set at 2.5, which equates to a Boolean value of 512. The furthest right indicator is the only one lit as that bits value is 512. An explanation of this can be seen in table 1.

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

Table 1

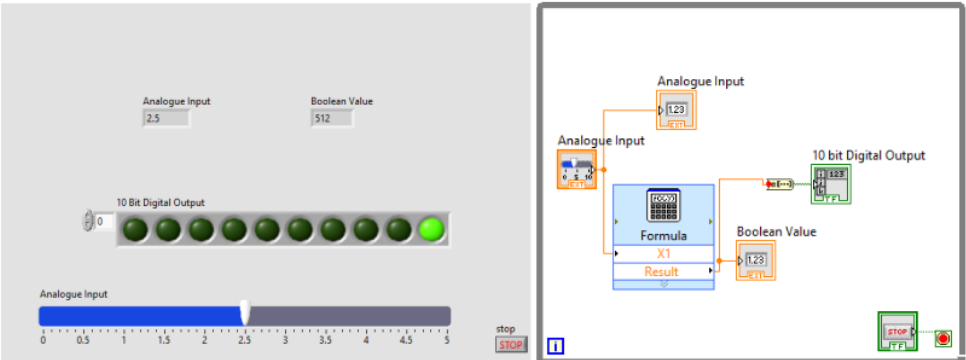


Figure 35

Should have a heading '10 bit ADC'

Another example of a 10-Bit digital output can be seen in figure 59, and a 12-Bit example can be seen in figure 60 – both in the appendix.

The evidence in the appendix verifies this

There should be an 8 bit ADC test here also!

There should be a heading here

The block view of the LabVIEW program can be seen in figure 36 There are four formula blocks which each have separate tasks. The four blocks are: Formula 1, Formula 2, Formula 3 and Formula 4.

Formula 1 is: $(X1/5) * 1000$. This is used to convert the analogue input into grams. To do this, the input 'X1' must be divided by the maximum input voltage of 5V, which is then multiplied by 1000 to get a gram value. As an example, if the maximum input of 5V is present then the equation will equal 1000g – proving to be true. $\frac{5}{5} * 1000 = 1000g$.

Formula 2 is: $(X1/5) * 256$. This converts the analogue input to a binary digital output. To do this, the analogue input is again divided by the maximum input value and is then multiplied by 256 – the full scale of an 8-bit decimal value.

Formula 3 is: $(X1/1000) * 256$. This converts the gram setpoint into an 8-bit digital output for display on the LabVIEW interface. The set gram value is divided by the maximum value of 1000g and is then multiplied by 256 to scale it.

Formula 4 is: $X1 * 256$. This shifts the bits over 8 places for a 16-bit transfer to the FPGA. The shifted 8-bit is for the setpoint value, and the added 8-bits which will go in the now free 8-bit space will be for the analogue input.

Good explanation of the significant blocks, including the shift left for 8 bits

The io should be explained here

Good block program layout

The io should be explained here

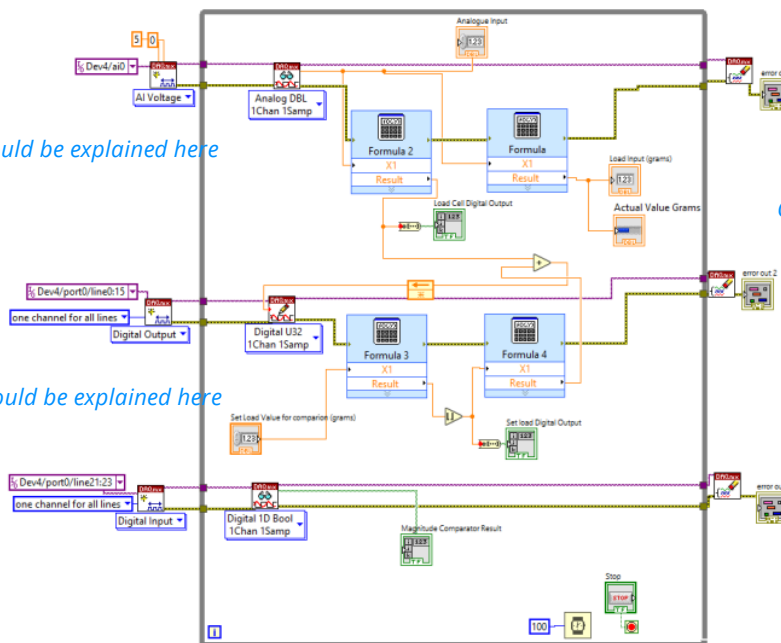


Figure 36

2.4. Testing and Analysis

2.4.1. Load Cells

$$\text{Buffer Gain} = \left(1 + 2 \times \left(\frac{35}{15} \right) \right) = 5$$

$$\text{Diff Amp Gain} = 3500 \div 35 = 100$$

$$\text{Total Gain} = 5 \times 100 = 500$$

Resistor values ahve been chosen correctly and test results indicate this.

% Resistance	Grams	Voltage	Figure
50	0	23.4mV	34
55	100	477mV	49
60	200	977mV	50
65	300	1.48V	51
70	400	1.98V	52
75	500	2.48V	53
80	600	2.98V	54
85	700	3.48V	55
90	800	3.98V	56
95	900	4.48V	57
100	1000	4.98V	58

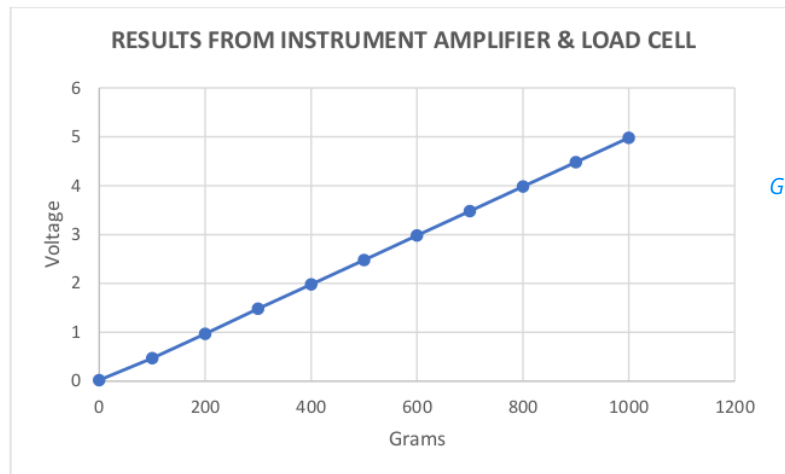
Good table of results

Table 2

Testing was completed by increasing the load from 0 – 1000 grams in 100-gram increments by altering the potentiometer difference by 5% each time – increasing two of the potentiometers and decreasing two of them.

Table 2 indicates the voltage starts at 0.023V, which is to be considered as zero. This is an error which is due to leakage voltage in the circuit which is being amplified by the gain.

The results give an accurate linear response of 0.5V per 100g, increasing to 1000 grams at full scale of 4.98V. The only increase which is not 0.5V is the step from 0g – 100g.



Good linear response

Figure 37

The graph in figure 37 shows a linear increase in voltage output as the load also increases in a linear fashion.

This proves that the signal conditioning is accurate, and the gain has made the scale of 0 – 5V, with the maximum value to be 4.98V and proves the scale to be linear which satisfies the required specification.

2.4.2. Digital Magnitude Comparator

Good use of table to log test results

Test No	State	Setpoint	LabVIEW Result	Figure	FPGA 8Bit Mag Comp Result	Figure	Pass/Fail
1	<	650	Sensor Value < SP (628g)	38	Sensor Value < SP	39	Pass
2	=	650	Sensor Value = SP (648g)	40	Sensor Value = SP	41	Pass
3	>	650	Sensor Value > SP (670g)	42	Sensor value > SP	43	Pass

Table 3

Good range of test results

3 tests were carried out with the 8-Bit magnitude comparator ($< = >$) and can be seen in table 3. The test results in LabVIEW were the same as the results from the FPGA. This proves the operation of each state in real time was reliable. The same test procedure can be implemented in element 3.

2.4.2.1. Less Than

In figures 38 and 39, we can see that the analogue input from the LabVIEW variable power supply is 3.13V, which is equal to a load input of 628g. As this is below the setpoint of 650g, the less than indicator is on and LED 0 is on, on the ELVIS board.

Explanation is consistent with test results

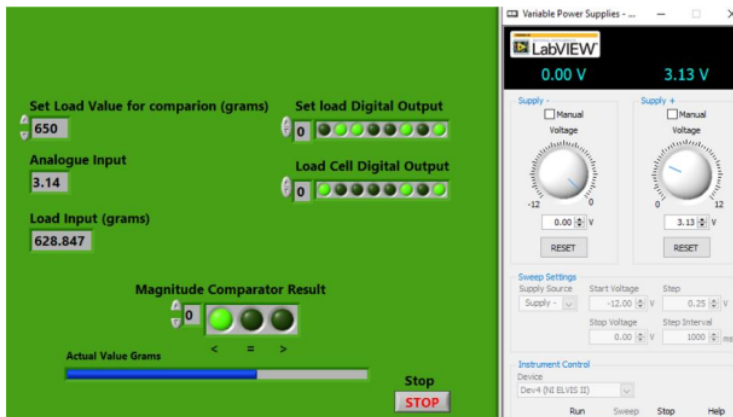


Figure 38



Figure 39

Good evidence

2.4.2.2. Equal To

In figures 40 and 41, the LabVIEW power supply is 3.23V, which equates to 648g. For the magnitude comparator this is good enough to equal 650g, as we can see the set load digital output and load cell digital output have the same indicators on and the equal to indicator is on.

LED1 is also on, on the ELVIS board.

Explanation is consistent with test results

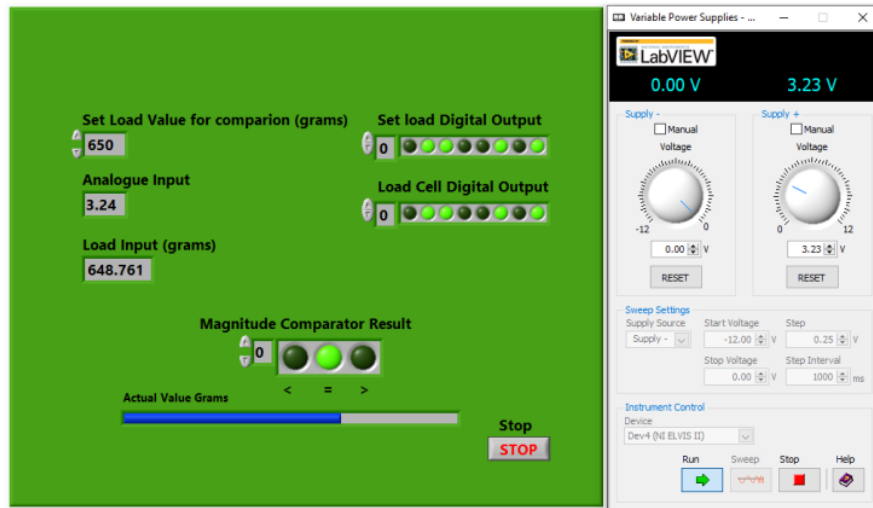


Figure 40

Good test evidence

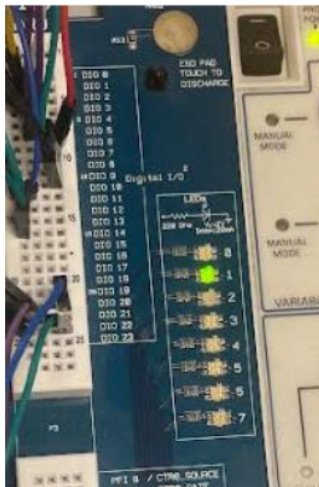


Figure 41

2.4.2.3. Greater Than

In figures 42 and 43, the LabVIEW power supply is 3.33V, which equates to 670g. As this is above the setpoint of 650g, the greater than indicator is on and LED 2 is on, on the ELVIS board.

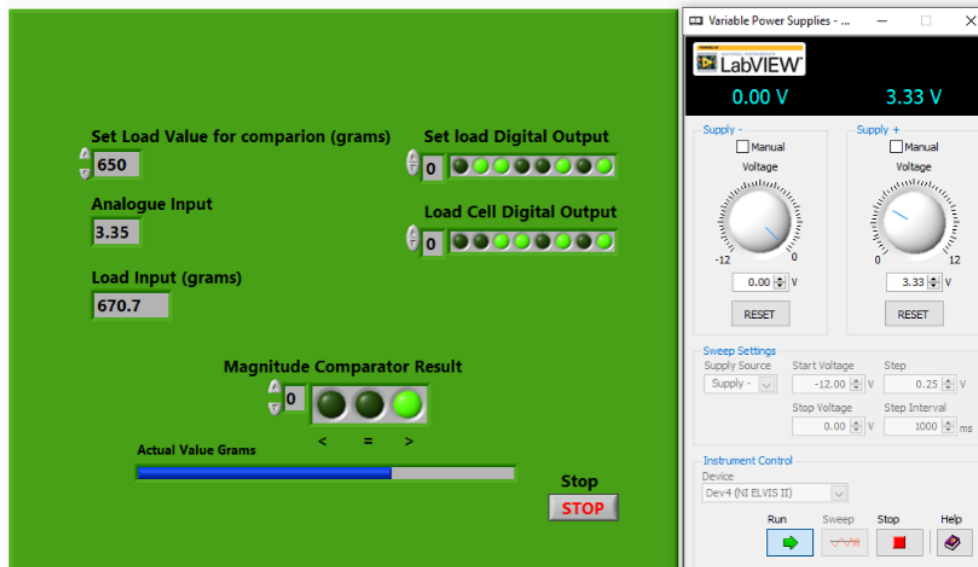


Figure 42

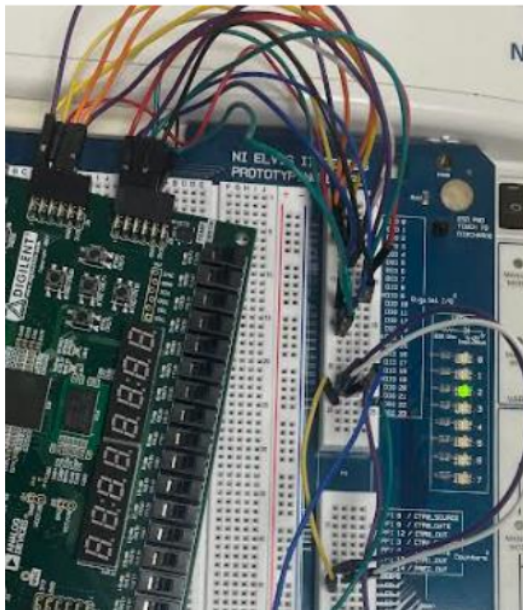


Figure 43

All 3 test states have been included with appropriate explanations and evidence

2.5. Evaluation

In this element, a LabVIEW 'vi' was designed to interface with the FPGA containing an 8-Bit magnitude comparator and two 8-Bit digital signals were compared to see if they were less than, equal to or greater than one another.

The first signal was an analogue input representing the load applied to a LCKG-1kg load cell which was then converted to an 8-Bit digital signal via an ADC. The second signal was the load setpoint in grams that was set via LabVIEW which was then converted to an 8-Bit digital signal. Both signals were then output to the FPGA. *Good recognition of purpose here*

The magnitude comparator is utilised to compare both converted digital signals as they would not have been able to compare the values in their original form.

The testing of the LabVIEW 'vi' and 8-Bit magnitude comparator system was successful as all requirements of the element were met as can be seen through the design, implementation and analysis of the system in this portfolio.

The purpose and functions of the system are identified in the evaluation

3. Element 3

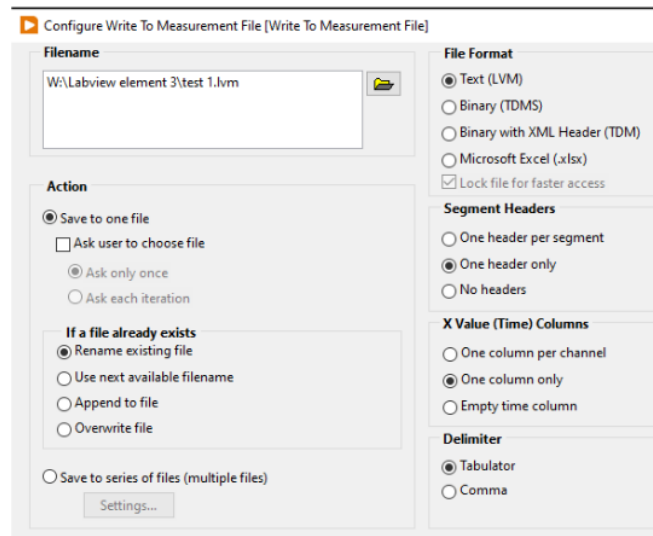
3.1. Introduction

In this element, an improved NI LabVIEW circuit will be designed to measure and display the load cell value in grams via a LabVIEW interface and a table and graphical comparison of the load cell values over its full range. *Grammatical errors here!*

3.2. Design Requirements

To modify the design to allow the comparison, measurements must be taken and logged. Measurements of the input, setpoint and the comparator outputs will be taken and logged to analyse. To do this, 'Write to Measurement Files' must be added to the NI LabVIEW block design. These files are text (LVM) files and will record automatically. *Good identification of requirements*

The first measurement file, 'Test 1', is for the analogue input of the potentiometer which is simulating the load. This file can be seen in figure 44, and is the only file to have a time column as the time will be needed as the input value changes. *Purpose is recognised*

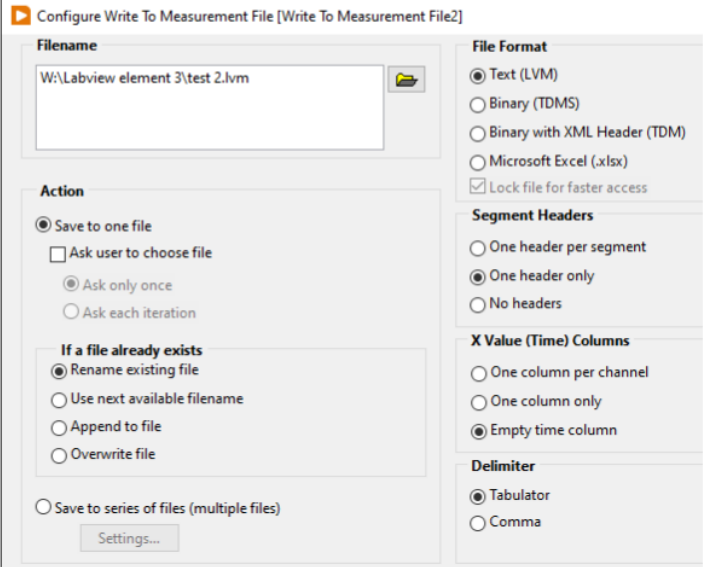


Good evidence of text file configuration

Figure 44

Figure 45 shows the second measurement file, 'Test 2', which is for the setpoint which will remain constant. This setpoint value will be compared against the 'Test 1' analogue input value. As the 'Test 1' value has a time value, there is no requirement for a time column in this measurement file.

This is the same for the other 3 measurement files, as they do not require a time column. The remaining files are for the compartor outputs: less than, equal to and greater than, when comparing the setpoint and actual value.



*Good sample evidence.
There should be some
reference to the other 3 files
in the appendix*

Figure 45

3.3. Implementation

Figure 46 shows the completed block design with the five 'Write to Measurement Files'. Each file for the comparison value is scaled by 300 to give a clearer representation on the graph results in the testing and analysis section.

Good explanation of the scale factor and its significance

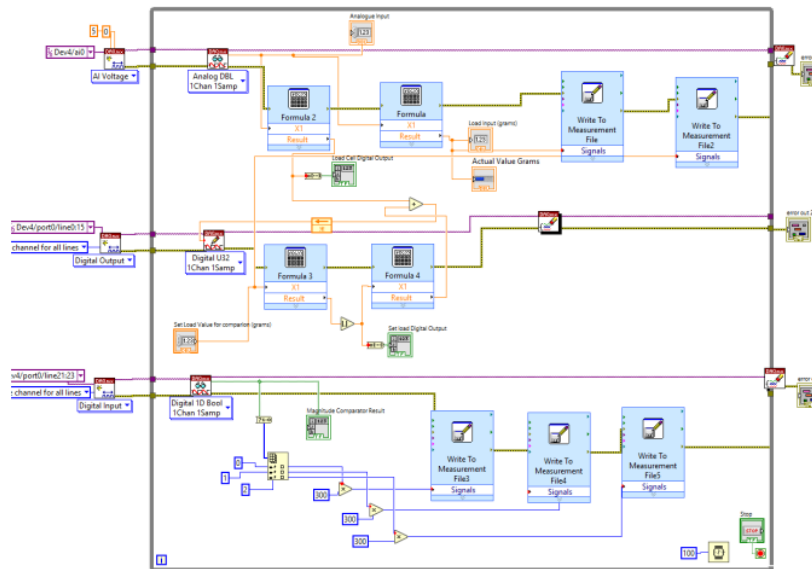


Figure 46

Figure 47 shows the layout of the text file for file 1. Under line 22, the time value will be beneath the 'X_Value' and the analogue input value will be beneath the 'Untitled' column. This will be replicated for the other 4 files and all the results will be gathered in an Excel document to create a graphical representation of the results. The full table of results can be seen in table 5 of the appendix.

Good sample and explanation

```
1 LabVIEW Measurement
2 Writer_Version 2
3 Reader_Version 2
4 Separator Tab
5 Decimal Separator .
6 Multi_Headings No
7 X_Columns One
8 Time_Pref Absolute
9 Operator as210694
10 Date 2023/02/02
11 Time 18:12:42.4299135208129882813
12 ***End_of_Header***
13
14 Channels 1
15 Samples 1
16 Date 2023/02/02
17 Time 18:12:42.4299135208129882813
18 X_Dimension Time
19 X0 0.000000000000000E+0
20 Delta_X 1.000000
21 ***End_of_Header***
22 X_Value Untitled Comment
```

Figure 47

3.4. Testing and Analysis

As can be seen in figure 48 we can see the results clearly. The analogue input value was first greater than the 650g setpoint, as the orange line is above the grey, and we can also see the green line at 300 – due to the scaling in the block design, and t this time the two other conditions are at 0.

The second test showed the two values to be equal to another at 40s, as only the grey line is visible and the equal to line is at 300. The third test shows that the analogue input is less than the setpoint at 68s, as this is the point when the less than line reaches 300.

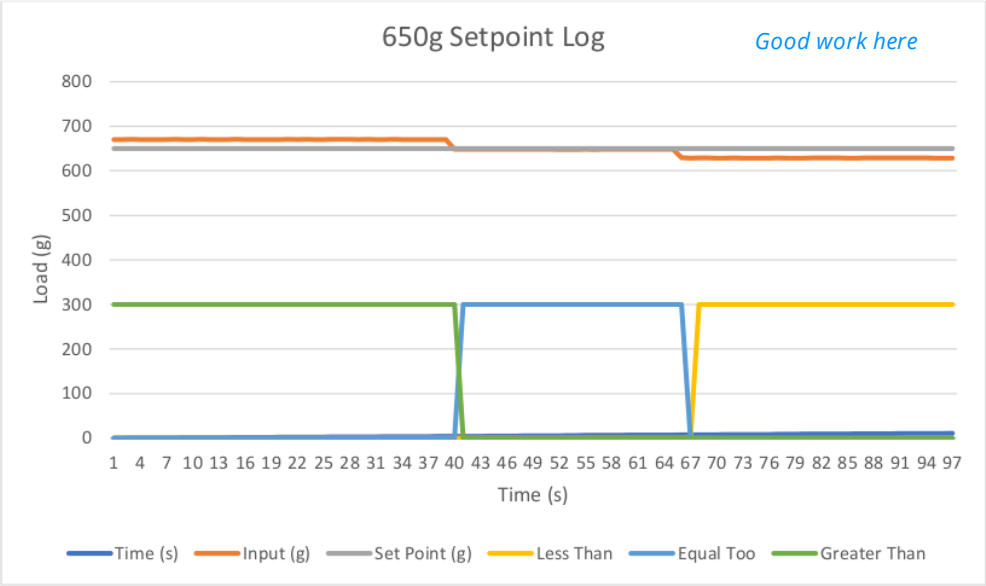


Figure 48

The same values for this test were used in Element 2 and as such, the LabVIEW interface results can be seen in Section 2.4. See table 4 for reference to tests.

Test	Figure
Less Than	38
Equal To	40
Greater Than	42

Table 4

Good reference to earlier test results

3.5. Conclusions

In this element, the research conducted in Element 1 was utilised further to adapt the NI LabVIEW circuit to output the load cell values over its full range and then compare these values in data logging files. This was completed by simulating an analogue input for the actual load and then comparing with the defined setpoint.

Testing was carried out over three different loads ensuring the correct response from the comparator was outputted – whether it was less than, equal to or greater than the setpoint. The setpoint was 650g, while the load values were 629g, 649g and 670g. The less than and equal too values were slightly off the expected 630g and 650g but were sufficient to give a correct binary and display response for their respective conditions.

Good practical point made here

The above issue – with the accuracy of the comparison – could be an area for improvement. The 8-bit comparator means that a value less than 1.24g than intended is seen as the same value. This could be eliminated by the introduction of a comparator with more bits.

A good recommendation of increased resolution

4. References

Dewesoft, 2021. *What Is Signal Conditioning or Signal Conditioner?*. [Online]
Available at: <https://dewesoft.com/daq/what-is-signal-conditioning>
[Accessed 19 01 2023].

Electronics Tutorial, 2018. *Digital Comparator and Magnitude Comparator Tutorial*. [Online]
Available at: https://www.electronics-tutorials.ws/combination/comb_8.html
[Accessed 09 01 2023].

FPGA Tutorial, 2020. *An Introduction to VHDL Data Types*. [Online]
Available at: <https://fpgatutorial.com/vhdl-types-and-conversions/>
[Accessed 06 01 2023].

Maloy-Smith, G., 2020. *Types of A/D Converters - The Ultimate Guide*. [Online]
Available at: <https://dewesoft.com/daq/types-of-adc-converters#main-adc-types>
[Accessed 11 01 2023].

National Instruments, 2022. *Learn 10 Functions in NI-DAQmx and Handle 80 Percent of Your Data Acquisition Applications*. [Online]
Available at: <https://www.ni.com/en-gb/support/documentation/supplemental/06/learn-10-functions-in-ni-daqmx-and-handle-80-percent-of-your-dat.html>
[Accessed 05 01 2023].

Prielipp, L., 2019. *WHAT IS A LOAD CELL?*. [Online]
Available at: <https://realpars.com/load-cell/>
[Accessed 10 01 2023].

Good number of sources used in research

5. Appendix

This should have a heading

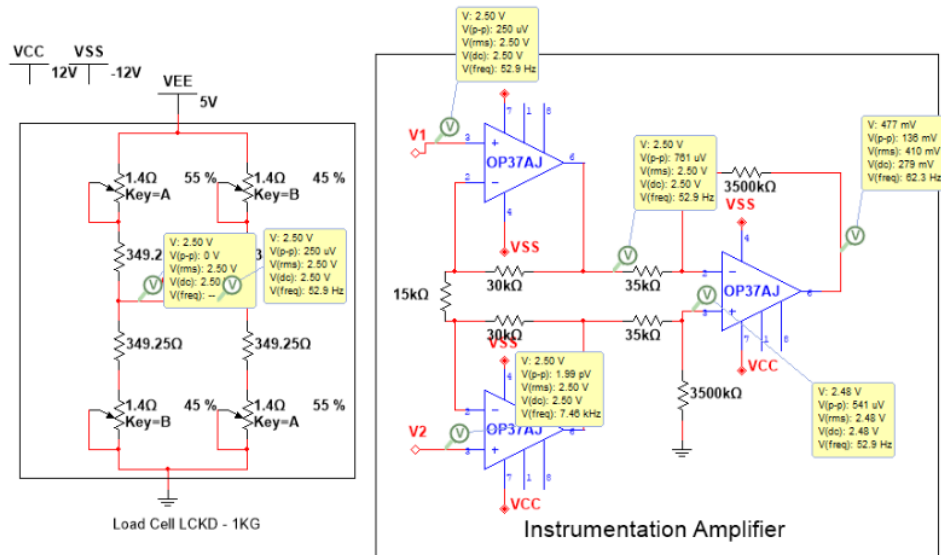


Figure 49

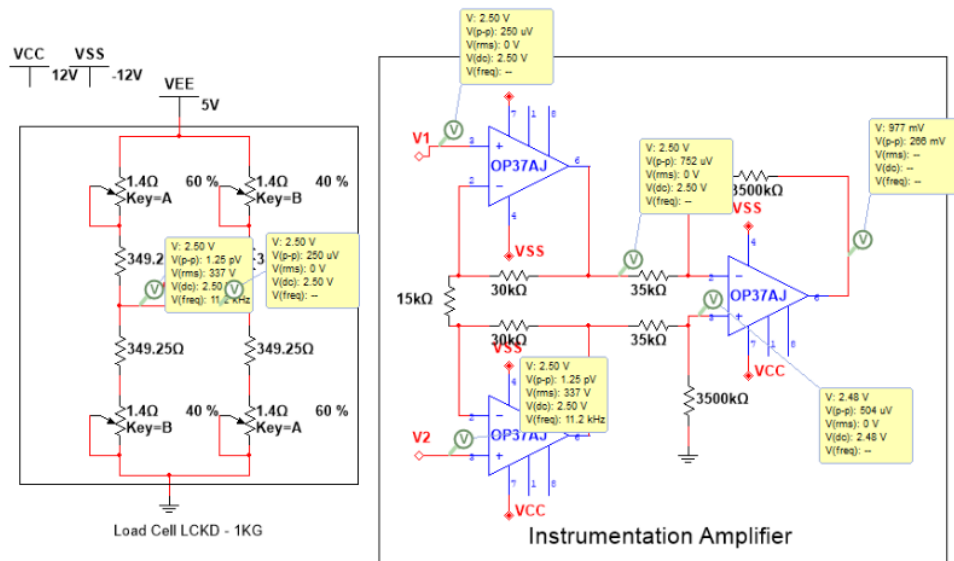


Figure 50

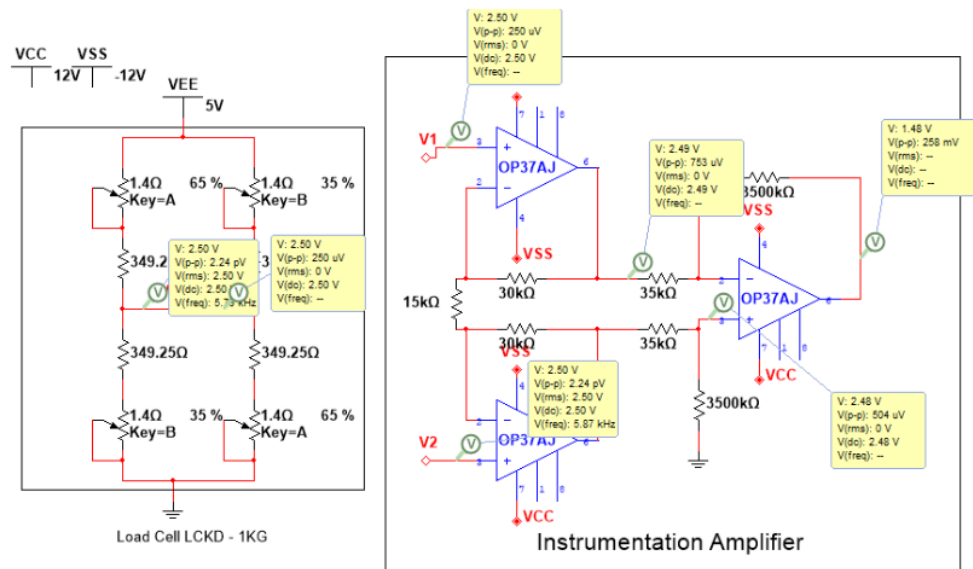


Figure 51

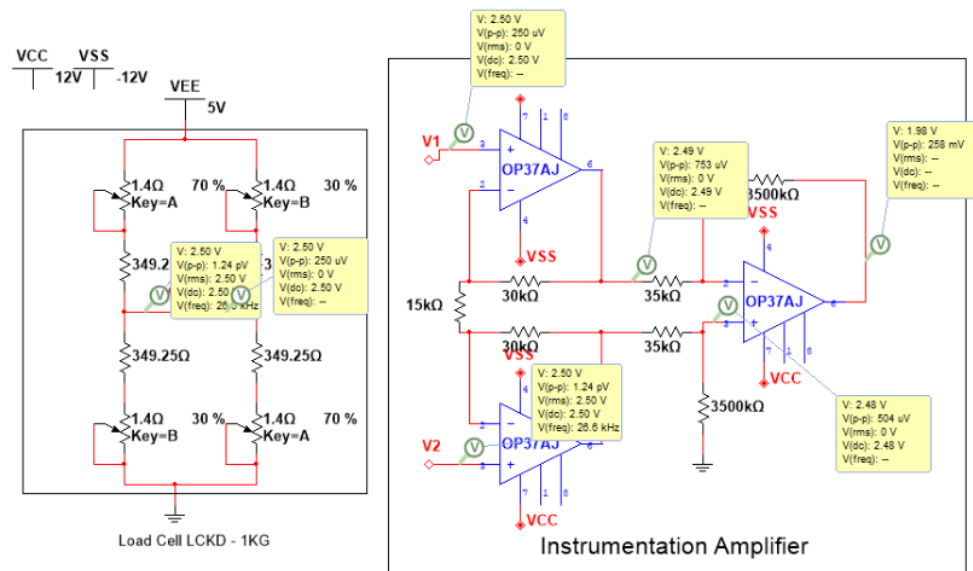


Figure 52

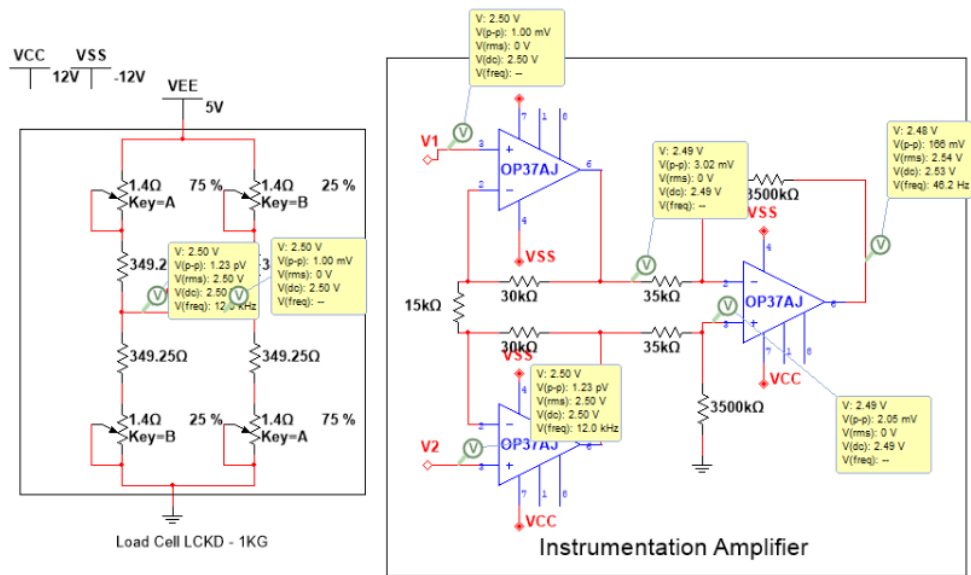


Figure 53

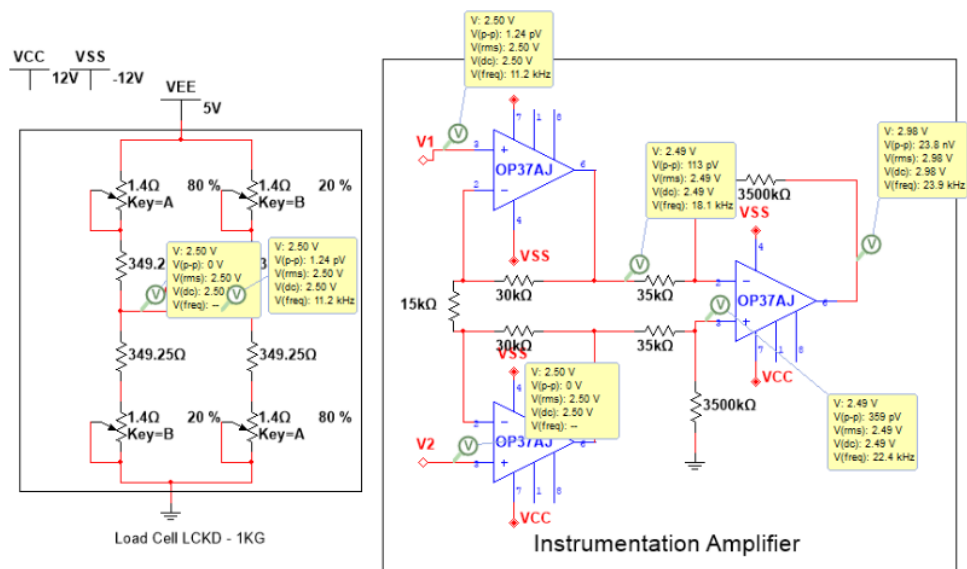


Figure 54

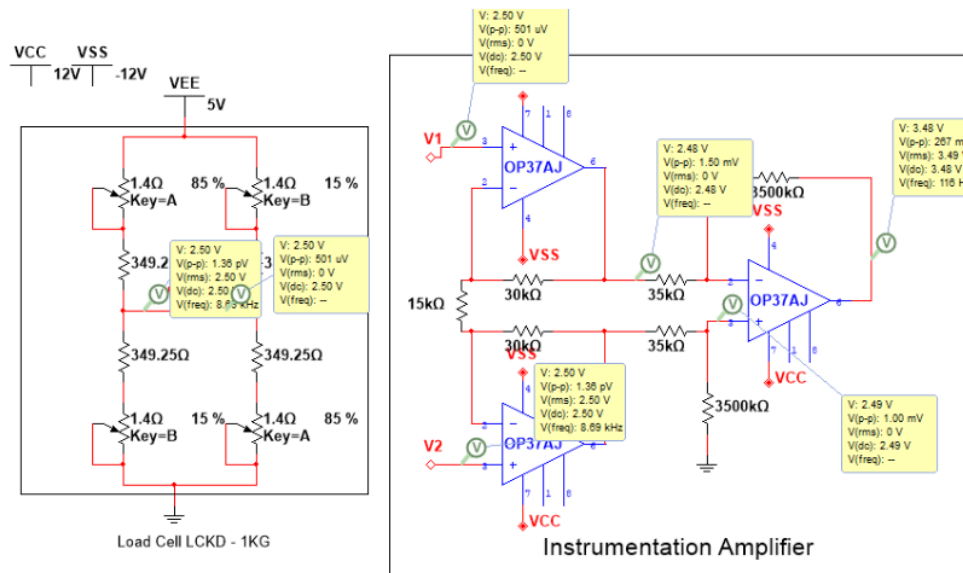


Figure 55

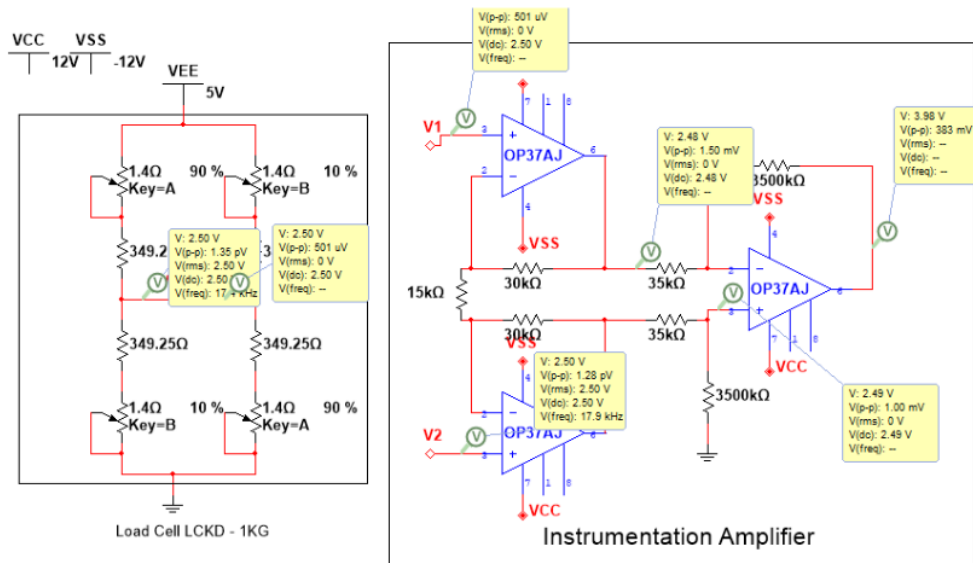


Figure 56

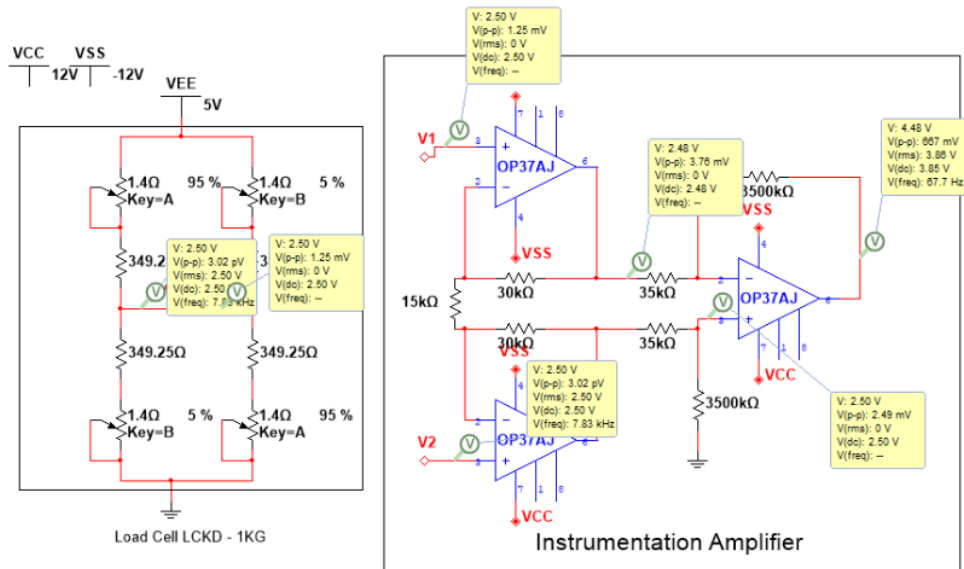


Figure 57

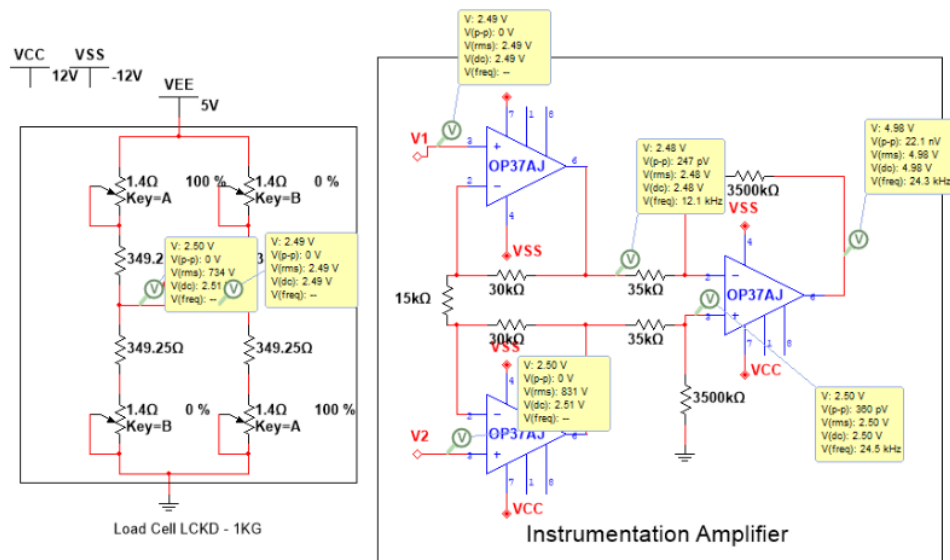


Figure 58

This should have a heading

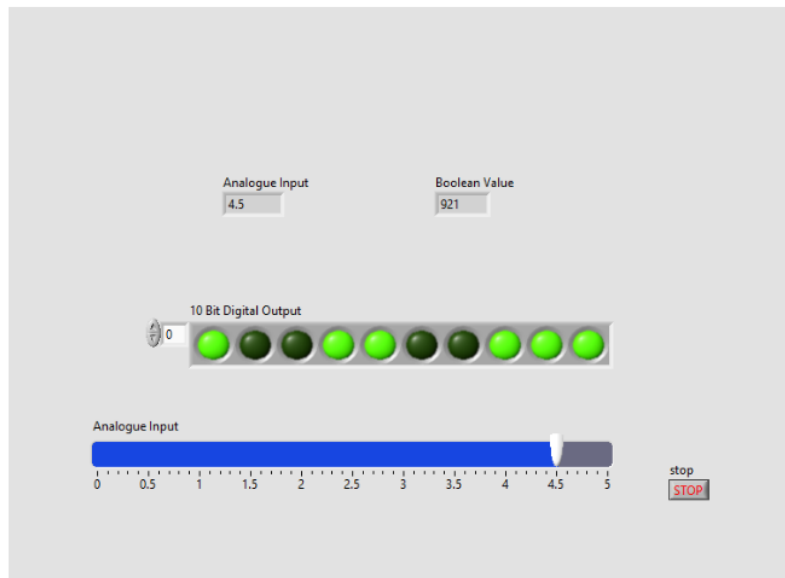


Figure 59

This should have a heading

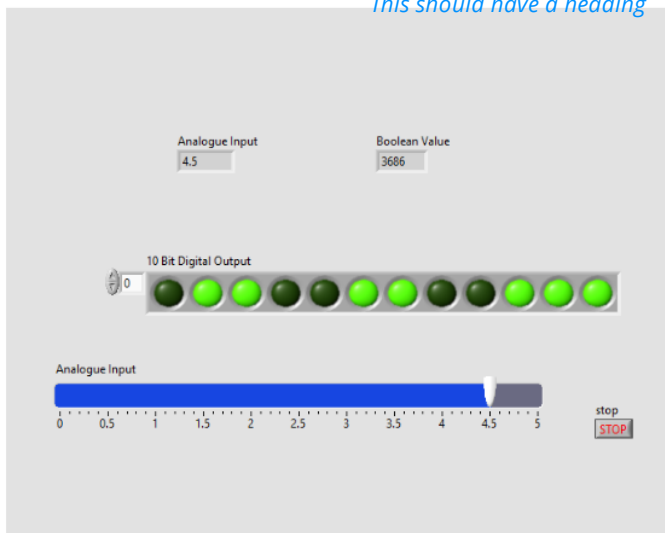


Figure 60

This data should have a heading

Time (s)	Input (g)	Set Point (g)	Less Than	Equal Too	Greater Than
0	670.0576	650	0	0	300
0.16965	670.1218	650	0	0	300
0.266183	670.4431	650	0	0	300
0.368674	670.3467	650	0	0	300
0.467892	670.0255	650	0	0	300
0.565715	670.154	650	0	0	300
0.666652	670.3788	650	0	0	300
0.766451	670.6679	650	0	0	300
0.867991	670.1861	650	0	0	300
0.96897	670.2824	650	0	0	300
1.066668	670.5394	650	0	0	300
1.169623	670.3788	650	0	0	300
1.267634	670.3146	650	0	0	300
1.370423	670.3788	650	0	0	300
1.469623	670.9249	650	0	0	300
1.570023	670.2824	650	0	0	300
1.670912	670.3788	650	0	0	300
1.771862	670.4109	650	0	0	300
1.871724	670.3467	650	0	0	300
1.972614	670.3467	650	0	0	300
2.074392	670.6358	650	0	0	300
2.173566	670.1861	650	0	0	300
2.274079	670.6679	650	0	0	300
2.375507	670.154	650	0	0	300
2.475115	670.3146	650	0	0	300

2.575801	670.8285	650	0	0	300
2.675559	670.5394	650	0	0	300
2.775176	670.7	650	0	0	300
2.876626	670.3788	650	0	0	300
2.977613	670.4752	650	0	0	300
3.076427	670.4109	650	0	0	300
3.178211	670.3467	650	0	0	300
3.27662	670.4752	650	0	0	300
3.378721	670.1218	650	0	0	300
3.480005	670.1861	650	0	0	300
3.579314	670.2824	650	0	0	300
3.679846	670.0897	650	0	0	300
3.778677	670.3467	650	0	0	300
3.880771	670.0255	650	0	0	300
3.978622	648.5366	650	0	0	300
4.078629	648.2476	650	0	300	0
4.180005	648.2154	650	0	300	0
4.280989	648.1833	650	0	300	0
4.380835	648.3118	650	0	300	0
4.482809	648.2476	650	0	300	0
4.580136	648.1512	650	0	300	0
4.755624	648.5688	650	0	300	0
5.03179	648.2476	650	0	300	0
5.147555	648.6009	650	0	300	0
5.246786	648.1191	650	0	300	0
5.34668	648.3118	650	0	300	0
5.448237	648.087	650	0	300	0
5.730018	648.087	650	0	300	0
5.826627	648.0548	650	0	300	0
6.227337	648.1512	650	0	300	0
6.325197	648.087	650	0	300	0
6.426229	648.3439	650	0	300	0
6.527839	648.4082	650	0	300	0
6.626104	648.1833	650	0	300	0
6.72667	648.4724	650	0	300	0
6.826627	648.2154	650	0	300	0
6.926645	648.4082	650	0	300	0
7.033267	648.5045	650	0	300	0
7.133685	648.4082	650	0	300	0
7.233734	648.5366	650	0	300	0
7.33295	629.7781	650	0	300	0
7.434175	628.6539	650	0	0	0
7.535439	628.7503	650	300	0	0
7.633184	628.7824	650	300	0	0
7.733217	628.5897	650	300	0	0
7.836627	628.686	650	300	0	0

7.936532	629.1357	650	300	0	0
8.03957	628.5254	650	300	0	0
8.13788	628.5897	650	300	0	0
8.239733	628.429	650	300	0	0
8.339849	628.6539	650	300	0	0
8.440152	629.0072	650	300	0	0
8.81157	628.5897	650	300	0	0
8.945683	628.4933	650	300	0	0
9.044046	628.5575	650	300	0	0
9.144114	628.9109	650	300	0	0
9.246944	629.0715	650	300	0	0
9.344689	629.1357	650	300	0	0
9.445683	628.9751	650	300	0	0
9.545445	628.6539	650	300	0	0
9.649185	628.4933	650	300	0	0
9.785329	629.0393	650	300	0	0
9.883257	629.1678	650	300	0	0
9.98471	628.8466	650	300	0	0
10.09287	628.8145	650	300	0	0
10.19117	628.9109	650	300	0	0
10.29298	628.9109	650	300	0	0
10.39188	629.0072	650	300	0	0
10.49998	628.7503	650	300	0	0
10.60082	628.6539	650	300	0	0
10.69896	628.4933	650	300	0	0
10.8009	628.686	650	300	0	0

Table 5

A good set of test data

FINAL GRADE

GENERAL COMMENTS

Instructor

63/100

PAGE 1

PAGE 2

PAGE 3

Text Comment. Good detailed table of contents

Text Comment. List of figures should be included

PAGE 4

Text Comment. A good explanation of capabilities relevant to the requirements. Good supporting examples

PAGE 5

Text Comment. Good explanation with relevant supporting examples

PAGE 6

Text Comment. There are more critical parts to the DAQ that could have been included, and an example system could have enhanced the research findings. However, there are good significant parts covered.

PAGE 7

Text Comment. Good explanation that differentiates a digital comparator

Text Comment. Good explanations, however, a supporting image example could have been included.

Text Comment. Good significant points included in the explanation. A supporting image as an example could have been included.

PAGE 8

Text Comment. Good explanation that could be improved with a supporting image

Text Comment. A block diagram of the stages described here should have been included.

Text Comment. The research contains good relevant points and sources, and is well balanced for each area. Supporting images / diagrams would have enhanced your explanations.

PAGE 9

Text Comment. Good block diagram with all the components required for the VHDL development

Text Comment. Good identification of hierarchy of building blocks for the VHDL

Text Comment. Good explanation and sample

PAGE 10

Text Comment. Good IP design with io configuration

Text Comment. The program explanation should be included and the program listing should be in the appendix

PAGE 11

Text Comment. Good IP design with io configuration

Text Comment. The Multisim design is good and you have identified the auto generated VHDL that was taken and used in the Vivado ADE.

Text Comment. The IP design can be generated from the VHDL should have been pointed out.

PAGE 12

Text Comment. A good design with io configuration

Text Comment. A short explanation of what this IP represents should be included

PAGE 13

Text Comment. Short explanation should be included

PAGE 14

Text Comment. FPGA

Text Comment. Explanation is good and demonstrates a good understanding of the function

PAGE 15

Text Comment. Explanation includes the significant points of the system

PAGE 16

Text Comment. Good clear image of test results

Text Comment. Very good explanation as the 'cout' state is significant

Text Comment. Good range of test data used

PAGE 17

Text Comment. Good clear test results

Text Comment. Good range of test data

PAGE 18

Text Comment. Good evidence of the 'equal' test state

Text Comment. Good test evidence

PAGE 19

Text Comment. All 3 test states have been successful

Text Comment. The 'cout' state is significant and is tested successfully here. This supports the explanation earlier in the implementation

PAGE 20

Text Comment. Good test evidence

Text Comment. Good test evidence

PAGE 21

Text Comment. All 3 test states are identified and explained

PAGE 22

Text Comment. FPGA

Text Comment. FPGA

Text Comment. Evaluation covers significant points relevant for the next element

PAGE 23

Text Comment. Intro should include purpose of the load cell

Text Comment. Good example of how the inst amp works using unity gain

Text Comment. The significance of the inst amp should be pointed out here!

PAGE 24

Text Comment. Good explanation of the interfacing technique

PAGE 25

Text Comment. Good diagram identifying interface method

Text Comment. the

Text Comment. the

PAGE 26

Text Comment. Good explanation of function procedure required

PAGE 27

Text Comment. Should have a heading '10 bit ADC'

Text Comment. The evidence in the appendix verifies this

Text Comment. There should be an 8 bit ADC test here also!

Text Comment. There should be a heading here

PAGE 28

Text Comment. Good explanation of the significant blocks, including the shift left for 8 bits

Text Comment. The io should be explained here

Text Comment. Good block program layout

Text Comment. The io should be explained here

PAGE 29

Text Comment. Resistor values have been chosen correctly and test results indicate this.

Text Comment. Good table of results

PAGE 30

Text Comment. Good linear response

PAGE 31

Text Comment. Good use of table to log test results

Text Comment. Good range of test results

Text Comment. Explanation is consistent with test results

Text Comment. Good evidence

PAGE 32

Text Comment. Explanation is consistent with test results

Text Comment. Good test evidence

PAGE 33

Text Comment. All 3 test states have been included with appropriate explanations and evidence

PAGE 34

Text Comment. Good recognition of purpose here

Text Comment. The purpose and functions of the system are identified in the evaluation

PAGE 35

Text Comment. Grammatical errors here!

Text Comment. Good identification of requirements

Text Comment. Purpose is recognised

Text Comment. Good evidence of text file configuration

PAGE 36

Text Comment. Good sample evidence. There should be some reference to the other 3 files in the appendix

PAGE 37

Text Comment. Good explanation of the scale factor and its significance

Text Comment. Good sample and explanation

PAGE 38

Text Comment. Good work here

Text Comment. Good reference to earlier test results

PAGE 39

Text Comment. Good practical point made here

Text Comment. A good recommendation of increased resolution

PAGE 40

Text Comment. Good number of sources used in research

PAGE 41

Text Comment. This should have a heading

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

Text Comment. This should have a heading

Text Comment. This should have a heading

PAGE 47

Text Comment. This data should have a heading

PAGE 48

PAGE 49

Text Comment. A good set of test data

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE



Research covers good significant points in all areas, with relevant sources used. there could have been improvements with supporting images in the areas indicated.

6/10

KNOWLEDGE AND UNDERSTANDING



Element 1 design requirements have been mostly expained with good evidence included. The 4 bit adder explanation is only part complete and there is no listing of the final program. There are significant points included and there is a good understanding demonstrated in the explanations and images. / diagrams. 5/10 Element 2 is mostly complete with consistent explanations with evidence including diagrams / images. The 8 bit ADC is missing and the LabVIEW program is only part explained. There is a good program design achieved and a clear understanding of the functions. Good interfacing methods have been included. 6.5/10 Element 3 content indicates a good understanding with the addition of text files into the LabVIEW program. 6.5/10

18/30

ANALYSIS



Element 1 Test results indicate a good method adopted using the virtual bus and is explained very well. The carry bit is explained and demonstrated. The evaluation does recognise the significance of the functions. 3.5/5 Element 2 test results are explained and include a good range of values. There are some parts missing (8 bit ADC). The evaluation is informative of the successes of the test results. 6/10 Element 3 test results are good and are explained consistently with the graph produced from results. Evaluation is good and includes a relevant recommendation. 6.5/10

16/25

PRACTICAL APPLICATION AND DEP.



Element 1 design and build is good and partly explained. The program for the 4 bit adder is only partly explained and the final program listing is missing. this should be in the appendix. However, the implementation was successful. 3/5 Element 2 interfacing methods used indicate a good knowledge gained about the components and their functions. The 8 bit ADC should have been included as part of the design process. 3/5 Element 3 data logging methods have been implemented with good explanation of functions and key points such as the scaler value for graphing. 7/10

13/20

SKILLS FOR PROFESSIONAL PRAC.



Element 1 covers requirements in good detail and in the correct sequence. The test evidence is very well presented. 3.5/5 Element 2 has some key headings missing. The sequence of the element is

10/15

good with good quality test evidence. 3/5 Element
3 has the requirements in good order and a good
graph of results. 3.5/5