

Sam Harries - Assignment 2

Final

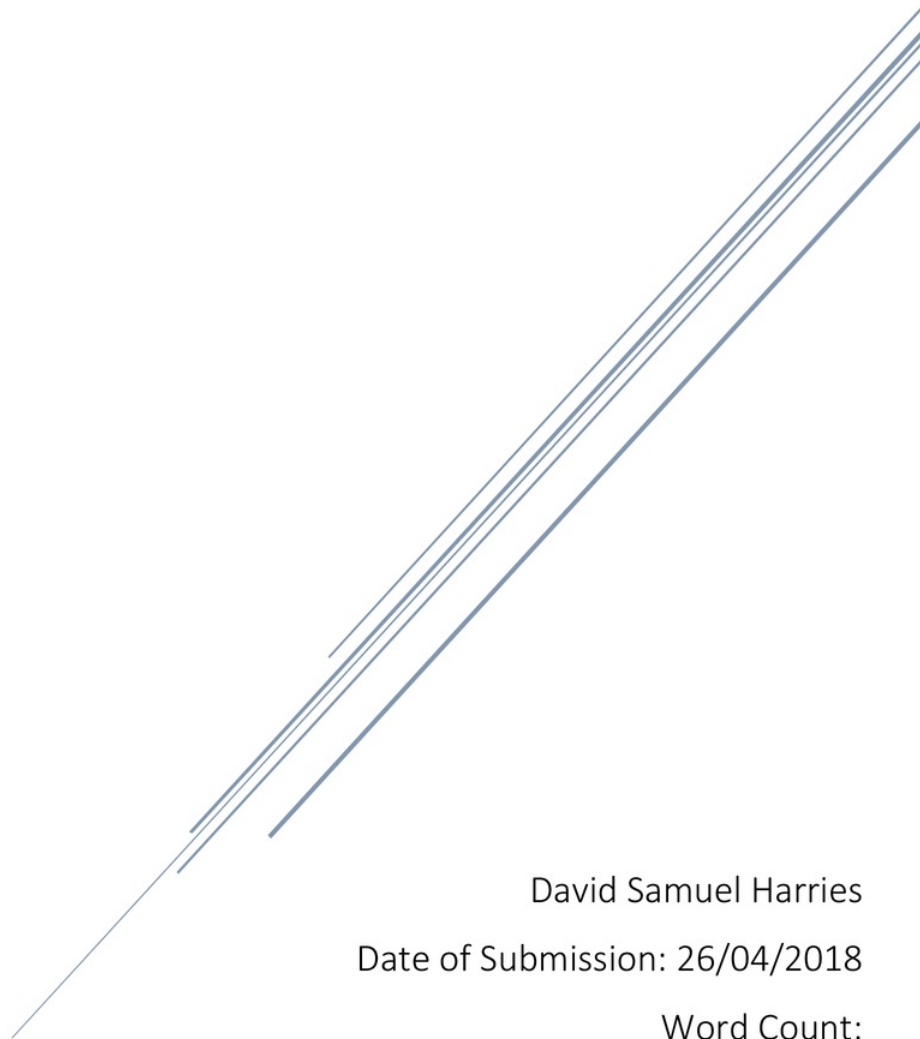
by David Harries

FILE	7309_DAVID_HARRIES_SAM_HARRIES_-_ASSIGNMENT_2_FINAL_157347_1913065437.DOCX		
TIME SUBMITTED	26-APR-2018 09:43AM (UTC+0100)	WORD COUNT	7192
SUBMISSION ID	87508234	CHARACTER COUNT	47055

HND Electrical and Electronic Engineering

PSDE5029

Assignment 2: NI LabVIEW Magnitude Comparator



David Samuel Harries

Date of Submission: 26/04/2018

Word Count:

TABLE OF CONTENTS

1	– Introduction	5
2	– Research	6
2.1	NI DAQmx Data Acquisition Icons	6
2.1.1	Add	6
2.1.2	Analogue Input Voltage	6
2.1.3	Analogue Read	7
2.1.4	Clear	7
2.1.5	Digital Input	7
2.1.6	Digital Output	8
2.1.7	Digital Read	8
2.1.8	Error Out	8
2.1.9	Feedback Node	8
2.1.10	Formula	9
2.1.11	Number to Boolean	9
2.1.12	Round to Nearest	9
2.1.13	Wait	9
2.1.14	While Loop	10
2.1.15	Write	10
2.2	Digital Magnitude Comparators	10
2.3	Analog to Digital Converters	12
2.3.1	Flash	13
2.3.2	Subranging	14
2.3.3	Successive Approximation	14
2.3.4	Sigma-Delta	15
2.3.5	Integrating ADCs	16
2.4	Load Cells	17
3	– Design of Assignment Requirements	19
3.1	8 Bit Magnitude Comparator Design	19
3.2	Load Cell Strain Gauge Design	21
3.3	LabVIEW ADC Design	23
4	– Analysis of Signal Conditioning Circuit, NI LabVIEW and Multisim Simulated Results	27
4.1	8 Bit Magnitude Comparator Simulated Test Results	27
4.2	Strain Gauge Load Cell Simulated Test Results	29
4.3	LabVIEW ADC Simulated Test	30

5	– Practical Build and Test via NI ELVIS Board.....	33
5.1	8 Bit Magnitude Comparator Practical Test.....	33
5.2	LabVIEW Program Test on the NI ELVIS II Board.....	34
5.2.1	Nexys4DDR FPGA Board.....	34
		35
5.2.2	NI ELVIS II Development Board.....	36
5.2.3	Practical Setup	36
6	– Analysis of NI LabVIEW / FPGA Results	38
7	– Adapt Design to Fit New Purposes or Applications	41
8	– Conclusions	43
9	– Appendices.....	44

TABLE OF FIGURES

Figure 1-1.....	5
Figure 2-1.....	6
Figure 2-2.....	6
Figure 2-3.....	7
Figure 2-4.....	7
Figure 2-5.....	7
Figure 2-6.....	8
Figure 2-7.....	8
Figure 2-8.....	8
Figure 2-9.....	8
Figure 2-10	9
Figure 2-11	9
Figure 2-12	9
Figure 2-13	9
Figure 2-14	10
Figure 2-15	10
Figure 2-16	10
Figure 2-17	11
Figure 2-18	12
Figure 2-19	13
Figure 2-20	13
Figure 2-21	14
Figure 2-22	14
Figure 2-23	15
Figure 2-24	16
Figure 2-25	16
Figure 2-26	17
Figure 2-27	17

Figure 3-1	19
Figure 3-2	19
Figure 3-3	20
Figure 3-4	21
Figure 3-5	21
Figure 3-6	22
Figure 3-7	22
Figure 3-8	23
Figure 3-9	24
Figure 3-10	25
Figure 3-11	26
Figure 3-12	26
Figure 4-1	27
Figure 4-2	28
Figure 4-3	28
Figure 4-4	29
Figure 4-5	30
Figure 4-6	30
Figure 4-7	31
Figure 4-8	31
Figure 5-1	33
Figure 5-2	33
Figure 5-3	34
Figure 5-4	34
Figure 5-5	35
Figure 5-6	36
Figure 5-7	37
Figure 6-1	38
Figure 6-2	39
Figure 6-3	39
Figure 7-1	41
Figure 7-2	42

1 – INTRODUCTION

The aims and objectives of this assignment are to use NI LabVIEW to interface with an FPGA using a NI ELVIS II development board to develop an 8 bit digital magnitude comparator. This comparator will compare the output of two 8 bit digital signals, one being the set value, and one being a variable. The system should indicate whether the variable signal is higher than, equal to, or lower than the set value.

The variable signal is generated via a 0 – 5V analogue input which represents a load applied to a LCKD-1kg load cell. This signal is sent through LabVIEW and converted into an 8 bit digital signal which is then output to the FPGA. The set signal is generated by a set value in LabVIEW. This signal represents a set value between 0 – 1kg which is then converted into an 8 bit signal ready for comparison on the FPGA.

For the comparison to be made, an 8 bit digital magnitude comparator must be designed in Multisim. This design will be first tested in a simulation, then tested practically by downloading it to the FPGA board. This comparator will be used to compare the two signals, and output 3 digital bits to LabVIEW.

Finally, the design must be altered to allow for data logging. By data logging the load cell input, and the digital comparator's outputs over time, a graph can be made to show the relationship between the two when compared to the set value. The final design should be structured as **Figure 1-1** shows.

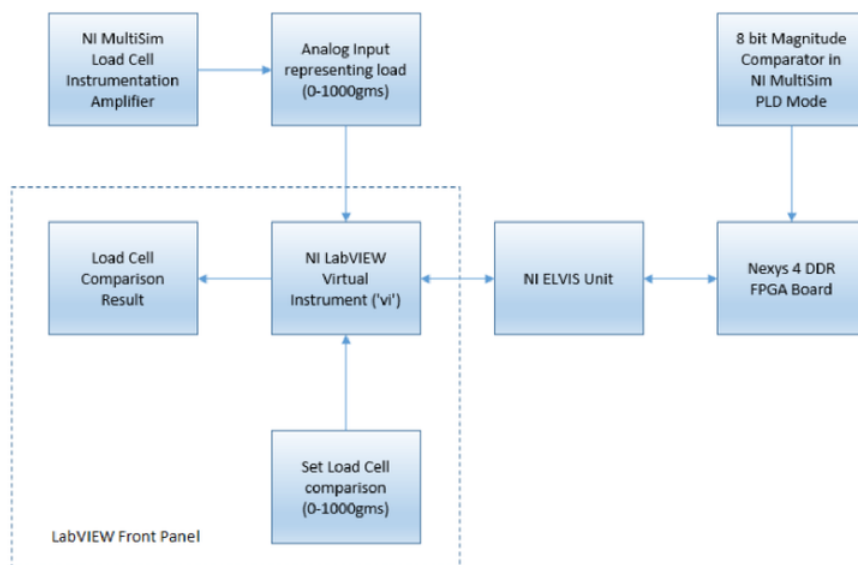


Figure 1-1

By the end of the assignment an application of appropriate theory to the analysis of practical aspects of industrial control systems should be achieved. An understanding of how software / hardware packages are used in industry for the control and test of automated systems are used, and why they are so useful when it comes to adapting designs to be used in various other applications.

Excellent introduction to the assignment

2 – RESEARCH

2.1 NI DAQMX DATA ACQUISITION ICONS

Ni DAQmx is driver software developed by national instruments that is used to communicate with and control data acquisition devices. DAQmx has an API (Application Programming Interface) that allows for the creation of applications in software such as LabVIEW. The API contains functions, classes and libraries. In LabVIEW the DAQmx driver operates in the form of icons. A selection of these will be discussed below. These DAQmx icons are all used within the later program.

2.1.1 Add

The add function is one of the simplest and acts like an AND gate. It simply takes two numerical values, x and y and calculates the sum of the two. This value is then transmitted on its output. The Add icon can be seen below.

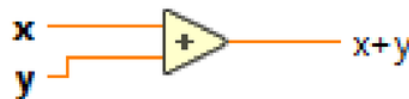


Figure 2-1

2.1.2 Analogue Input Voltage

The Analogue Input Voltage icon is used to create one or more channels to measure an analogue voltage. These channels are connected to the physical channels pin of the icon and will create a drop-down menu that can be used to find the correct channel. A maximum and minimum limit can be added to this function to ensure the input voltage stays within that range. **Figure 2-2** shows the Analogue Input Voltage icon.

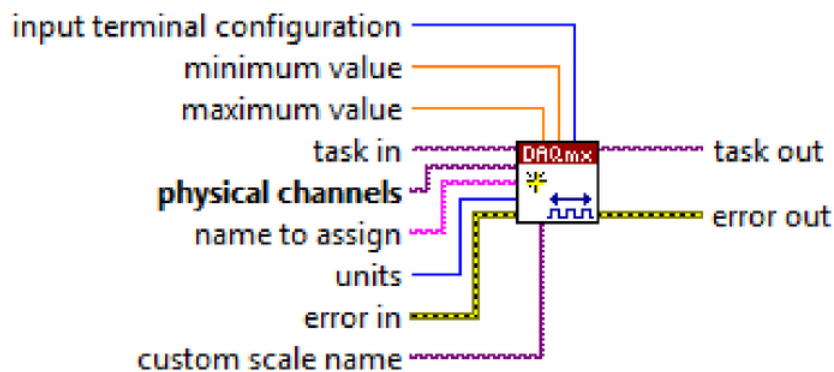


Figure 2-2

The task out terminal is used to represent the measurement of the analogue input and transmits the value of the voltage to other functions.

2.1.3 Analogue Read

The analogue read icon is used to read the input channel or task and create a floating-point sample. This sample is then output on the data terminal of the icon to later be used in calculations or other functions. The Analogue Read icon can be seen in **Figure 2-3**.

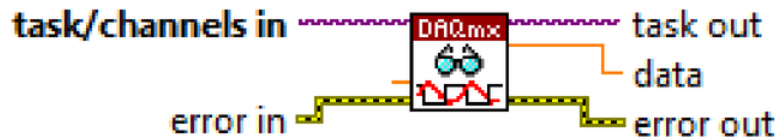


Figure 2-3

2.1.4 Clear

The Clear icon is used to terminate or clear the task. This is used at the end of a branch of functions to release any resources that the task has reserved. **Figure 2-4** shows the clear icon.



Figure 2-4

Once a task has been terminated it cannot be used anymore unless it is recreated. The error is output on the other side of the icon is connected to an Error Out icon which will later be discussed.

2.1.5 Digital Input

The Digital Input icon functions as its name suggests. It is used to create one or more channels to measure digital signals. This is done by selecting the channel on the lines terminal. Multiple lines can be grouped into one channel or separated into multiple channels. However, if a specific port is selected then they cannot be separated, and multiple digital input icons must be used. The Digital Input icon can be seen in **Figure 2-5**.

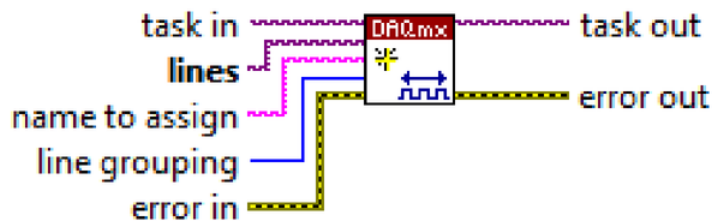


Figure 2-5

2.1.6 Digital Output

The Digital Output operates in the same way as the Digital Input, except it is used to generate signals rather than measure them. The icon for the Digital Output function can be seen in

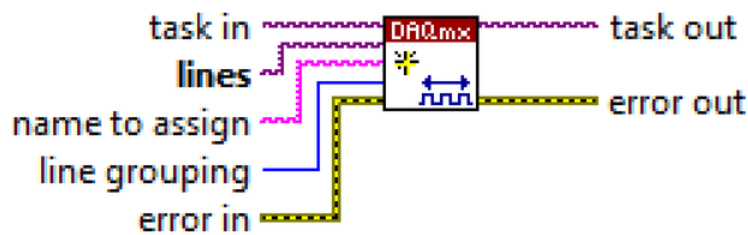


Figure 2-6

2.1.7 Digital Read

The Digital Read icon is used to read an array of Boolean values from a task. The task must contain a single digital input channel to be read. The values are then sent through the data channel. **Figure 2-7** shows the Digital Read icon.



Figure 2-7

2.1.8 Error Out

The Error Out icon is used to display error messages on a text box on the front panel. Errors can be detected anywhere through the LabVIEW design hence why every block icon has an error in, and error out terminal. No matter where an error occurs it will be sent to this icon and displayed to the user.

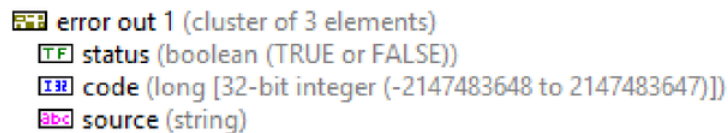


Figure 2-8

2.1.9 Feedback Node

A Feedback Node is used to store data from a VI execution or loop to the next.



Figure 2-9

2.1.10 Formula

The Formula icon is used to create mathematical formula that can be used to perform many different functions. The input can be a voltage, and the output is the result of the formula inside the block. The result is transmitted from the Result terminal. **Figure 2-10** shows the Formula icon.

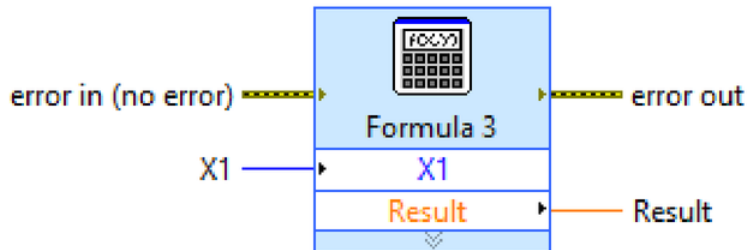


Figure 2-10

2.1.11 Number to Boolean

The Number to Boolean icon is used to convert an integer or fixed-point number, into a Boolean array. The size of the array is determined by the number of bits in the integer, or by the word length of the fixed-point number. **Figure 2-11** shows the Number to Boolean icon.



Figure 2-11

2.1.12 Round to Nearest

The Round to Nearest icon is used to round the number on the input to the nearest integer. If the number is exactly half way between two integers it rounds to the nearest even number. **Figure 2-12** shows the Round to Nearest icon.



Figure 2-12

2.1.13 Wait

The Wait icon is used to get the LabVIEW design to wait for the number of milliseconds assigned to the icon. This icon is used to synchronize activities or to force the current thread to obtain control of the CPU. This is done by assigning a value of 0 to the icon. **Figure 2-13** shows the Wait icon.



Figure 2-13

2.1.14 While Loop

The While Loop is used to make a design contained within it repeat until a certain condition is met. This loop can be stopped using a stop button. **Figure 2-14** shows a While Loop.



Figure 2-14

2.1.15 Write

A Write icon is used to write a single 32bit integer sample to a task that contains a single digital output channel.

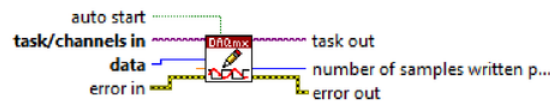


Figure 2-15

(Zone.ni.com, 2018) [Very good detailed research carried out on the DAQmx icons](#)

2.2 DIGITAL MAGNITUDE COMPARATORS

The purpose of a comparator is to compare a set of variables to a set of constants and output a signal which depends on the results of the comparison. In the case of a digital magnitude comparator, the magnitude of the two signals are compared, and the output signal indicates whether the variable signal is higher than, equal to or less than the constant signal.

A requirement for this assignment is to design an 8-bit magnitude comparator. An example of a 4 bit magnitude comparator is the 74HC/HCT85. A pinout diagram, logic symbol and IEC logic symbol is found on the components datasheet and can be seen in **Figure 2-16**.

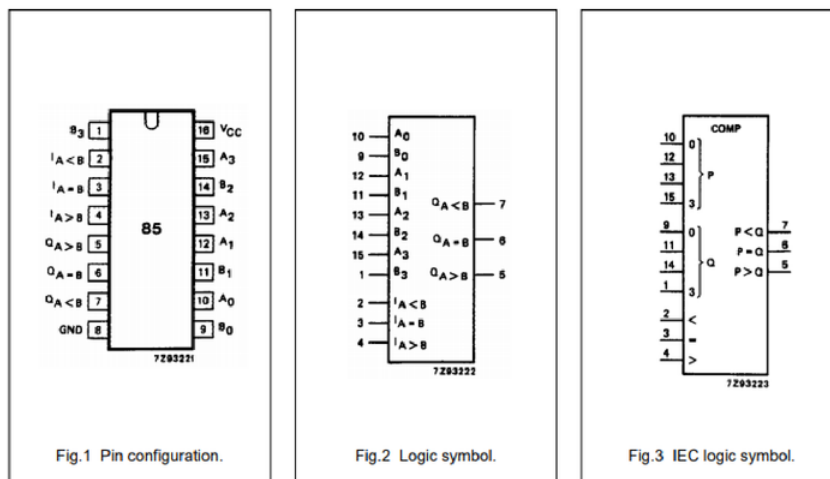


Figure 2-16

The datasheet also provides a function table for all the different possible combinations for the comparator. This can be found below.

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	I _{A>B}	I _{A<B}	I _{A=B}	Q _{A>B}	Q _{A<B}	Q _{A=B}
A ₃ >B ₃	X	X	X	X	X	X	H	L	L
A ₃ <B ₃	X	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	L	L	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	H	L	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	H	L	L	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	L	H	H	L

Table 1

(74HC85 4 bit Magnitude Comparator, 1990)

The logic diagram for the 74HC85 can be found in

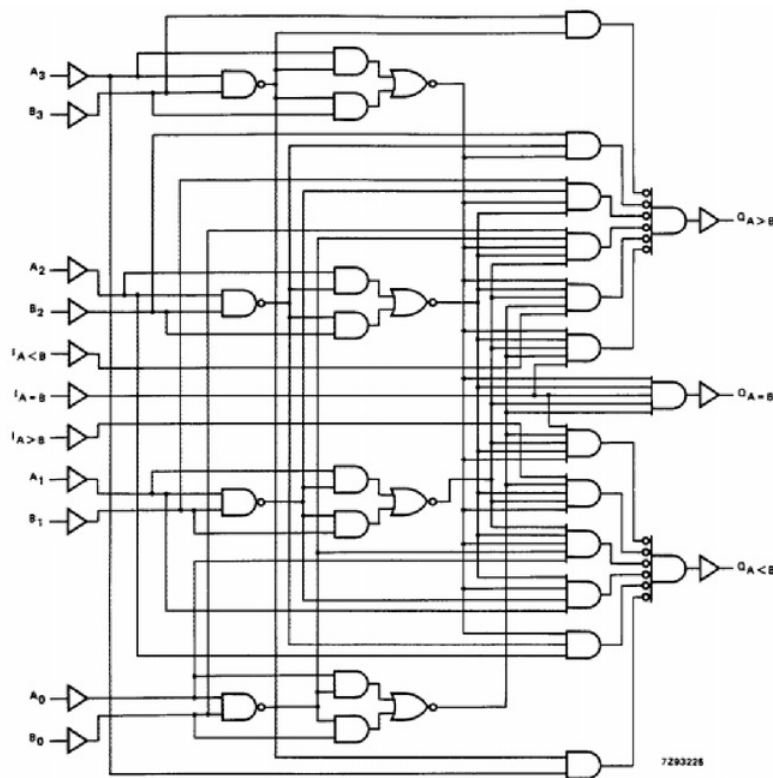


Figure 2-17

The datasheet also provides a diagram of how to turn a 4 bit magnitude comparator into a 12 bit comparator. This can be seen in **Figure 2-18**.

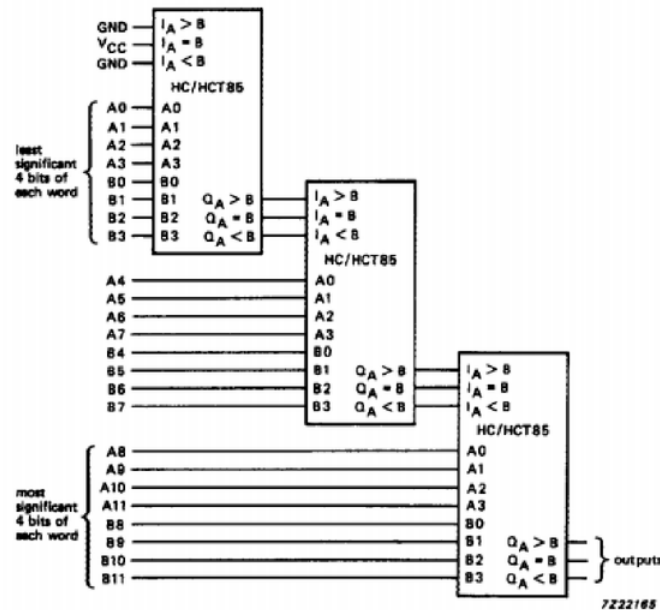


Figure 2-18

The above design shows that the 4 bit comparator can be cascaded to increase the number of bits being compared. For an 8 bit comparator, two 4 bit comparators must be cascaded in the configuration as shown in **Figure 2-18**.

When a comparator is comparing large binary or BCD numbers, it starts with comparing the most significant bit in both signals. It does this to save time and to operate more efficiently. If the MSB in signal 1 is equal to the MSB in signal 2, then it moves to the next most significant bit, and continues until it finds an inequality. If no inequalities are found, then the comparator outputs a signal indicating that both signals are equal.

However, if an inequality is found between the two signals, the output can be determined and the comparison between any lower bits is stopped. Digital comparators are often used in analogue to digital conversion systems.

Very good understanding shown on how the comparator functions

2.3 ANALOG TO DIGITAL CONVERTERS

Analogue to digital conversion is the process of taking an analogue signal and turning it into a digital signal so it can be used by a computer. Digital signals are superior to analogue signals when it comes to being susceptible to noise and interference. Digital signals are also much easier to recover than analogue signals.

This conversion is done by ADCs which come in several different architectures. The different architectures include:

- Integrating

- Sigma-delta
- Successive Approximation
- Flash
- Sub ranging

Each of these architectures have their advantages and disadvantages. The usefulness of these different types depends on two major factors, their sample rate, and their resolution. **Figure 2-19** shows a graphical comparison of sample rate vs resolution for the named ADCs.

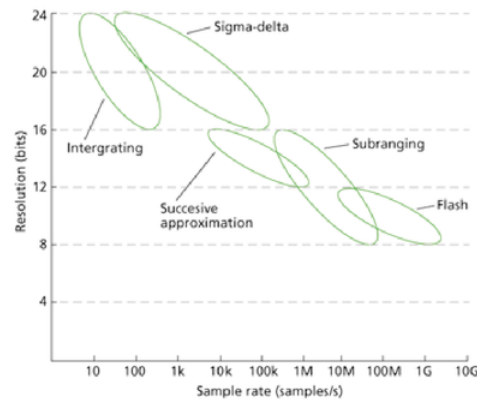


Figure 2-19

(Rowe, 2002)

2.3.1 Flash

The Flash ADC has the highest sample rate of them all, however it also has the lowest average resolution. This makes them useful for applications that do not require accurate conversion but do however value speed. Flash ADCs are very power hungry due to the sheer number of comparators needed to operate. And 8 bit flash ADC requires 255 comparators. **Figure 2-20** shows a 3 bit flash ADC.

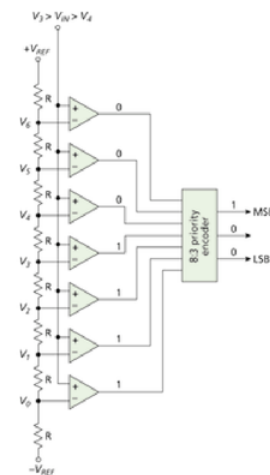


Figure 2-20

This ADC uses a bank of comparators used to compare the variable input voltage with a constant set voltage across the resistor network. This set voltage is often referred to as the reference voltage. The volt drop across the resistors starts at half of the LSB and increase in increments equal to that of one LSB per comparator. Each comparator outputs a value that represents one LSB.

As the input voltage is increased, the comparators change their output from 0 to 1 once the input voltage reaches the reference voltage.

2.3.2 Subranging

Subranging ADCs are slightly slower than flash ADCs, however they are capable of higher resolution. **Figure 2-21** shows a block diagram on an 8 bit subranging ADC.

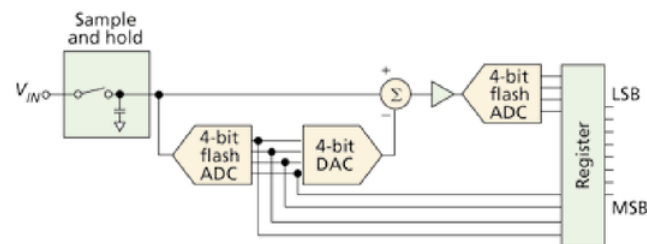


Figure 2-21

As shown, the 8 bit subranging ADC uses two 4 bit ADCs and a 4 bit DAC. When an analogue signal is input, the sample and hold amplifier stores a sample of the signal. The first stage's ADC converts the signal into digital data that gets sent to the register. At the same time the signal is reconverted back into an analogue signal which then gets subtracted from the sampled input signal from the sample and hold amplifier. The signal is then amplified and sent through the second ADC which outputs to the register where the final digital signal is output.

Subranging ADCs consume a lot less power than the flash ADC as an 8 bit subranging ADC only uses 30 comparators, instead of the flash's 255.

(Rowe, 2002), (Cao and Yan, 2008)

2.3.3 Successive Approximation

Successive approximation ADCs, often called SARs once again have slower sample rate than the previous two ADCs, however the resolution is once again increased.

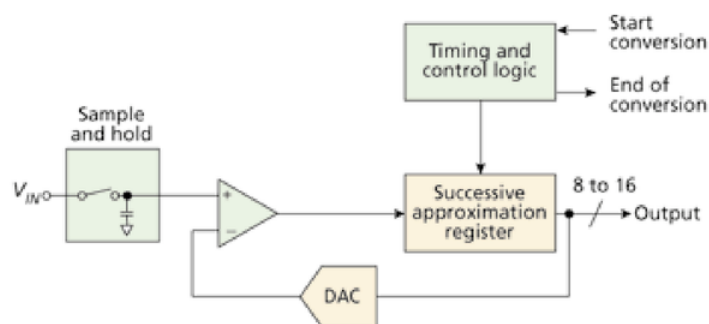


Figure 2-22

Figure 2-22 shows a successive approximation ADC block diagram. The principle of operation for this ADC is that it successively approximates each bit in a signal until the final digital signal is deciphered. When an analogue signal is input, the register sets its output so that every bit except the MSB is set to 0. When this is the case, the output is at half of its maximum. This output is sent to a comparator which compares the signal with the input signal and if the output is larger than the input, the comparator outputs a 1, and if it's smaller than the input then the comparator outputs a 0. This process repeats except the second most significant bit is output to 1, then the third MSB and so on.

This process repeats until every bit has been sent through the comparator, at which time the input signal would have been completely digitised. One cycle is required for each output bit, which means that N cycles are required for an N bit converter. A 16 bit converter would take twice as long to convert a signal than an 8 bit converter.

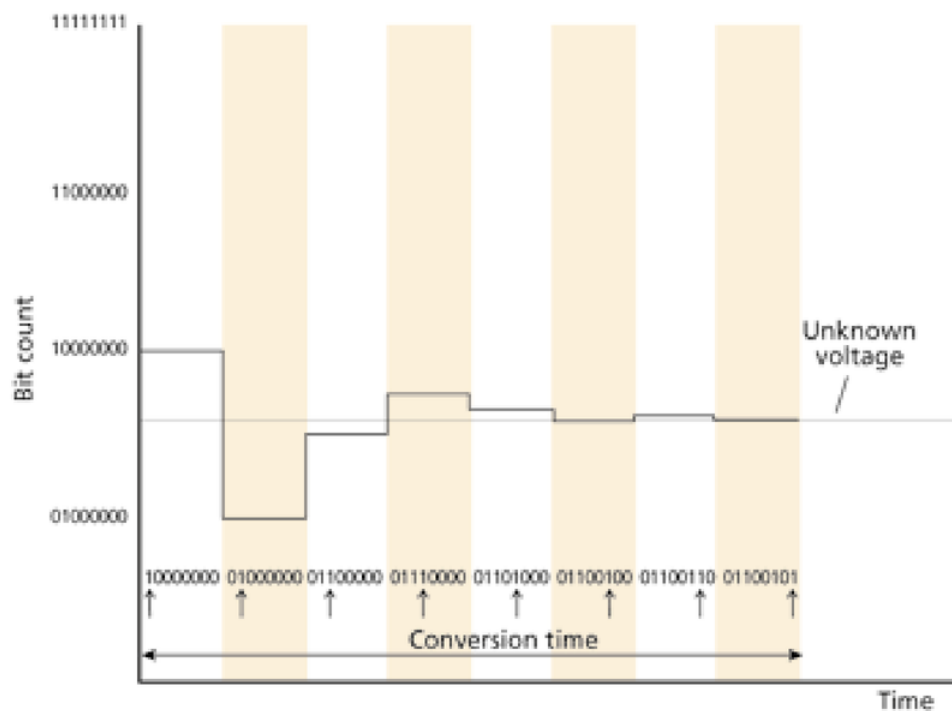


Figure 2-23

Figure 2-23 shows how a typical approximation cycle would be done to obtain an unknown input voltage. SAR converters are typically used in applications such as scopes and digital multi meters as speed isn't required, but accuracy is.

(Rowe, 2002)

2.3.4 Sigma-Delta

A sigma-delta converter is a slower but much more accurate ADC than the aforementioned ones. This is typically used for digitizing audio-frequency signals as they are very accurate.

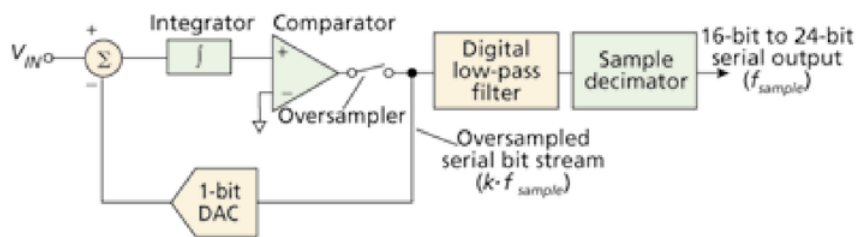


Figure 2-24

A sigma-delta ADC will oversample an input analogue signal and send it through an integrator. The integrator's output drives a comparator. The output from the comparator then in turn drives the DAC found in the feedback loop. Over many cycles the combination of the three components produces a serial bit stream which represents the oversampled analogue signal.

This serial bitstream then gets sent through a digital low-pass filter which removes frequencies at or above the Nyquist frequency. "The Nyquist frequency is a type of sampling frequency that uses signal processing that is defined as "half of the rate" of a discrete signal processing system. It is the highest frequency that can be coded for a particular sampling rate so that the signal can be reconstructed." (Techopedia.com, 2017)

The filtered signal is finally sent through a sample decimator which is used to remove the oversampled data. "In an ADC with 128X oversampling, the decimator will retain 1 bit for every 128 bits that it receives. The final output is a serial bit stream." (Rowe, 2002)

2.3.5 Integrating ADCs

Integrating ADCs are equally as accurate as sigma-delta ADCs but are slower and are therefore made slightly redundant. However, they still have their uses as they remove any power line frequency noise from the input signal. This happens because the input voltage charges a capacitor for a set amount of time. This time is typically 50Hz which is equal to 1 line cycle.

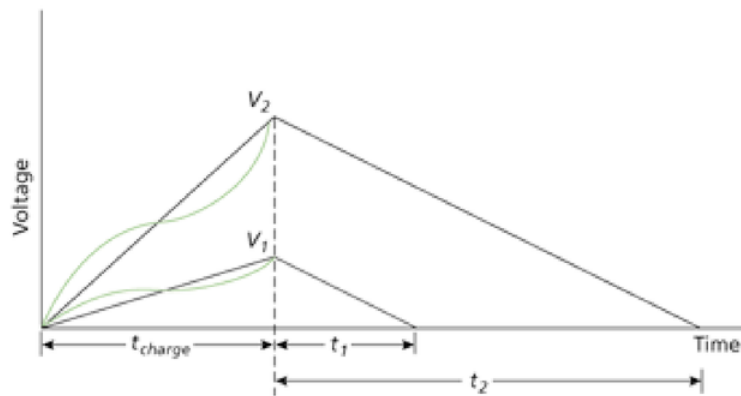


Figure 2-25

The signal is integrated over this time period which removes the line noise from the equation and doesn't get converted. Once the time is over, the capacitor discharges at a fixed rate. At the same time, a counter is made active which counts the ADC's output bits. The longer it takes

for the capacitor to discharge, the more the counter will count, resulting in higher accuracy at the cost of sample rate. Integrated ADCs are often used in high accuracy digital multi meters.

(Rowe, 2002), (Cao and Yan, 2008)

[Excellent detail on ADC research](#)

2.4 LOAD CELLS

"A load cell (or loadcell) is a transducer which converts force into a measurable electrical output." (Omega.co.uk, 2013) There are many different types of load cells such as hydraulic, pneumatic, strain gauge, inductive and reluctance and magnetostrictive. For this assignment a strain gauge is required to measure the weight of a subject between 0 and 1kg.

"A strain gauge is a long length of conductor arranged in a zigzag pattern on a membrane." (Rdpe.com, 2013) When the membrane is stretched, its resistance increases. Because of this they are often used in fours to create a Wheatstone Bridge configuration. When a weight or pressure is applied to the circuit, the resistance of a parallel pair of resistors increases, while the resistance of the other parallel pair of resistors decreases. **Figure 2-26** and **Figure 2-27** show the effect of pressure on each strain gauge when they extend and contract.

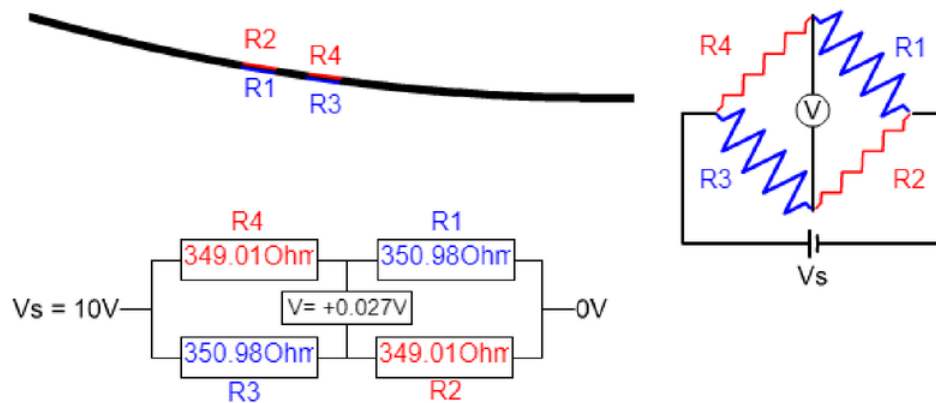


Figure 2-26

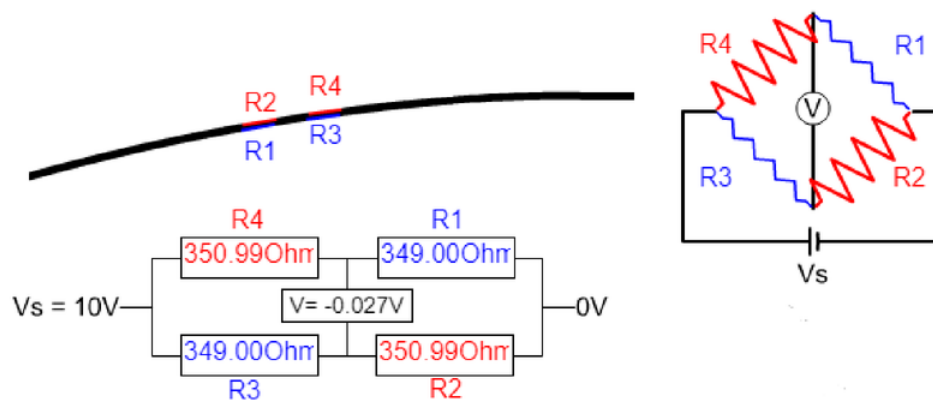


Figure 2-27

The voltage going through the Wheatstone bridge circuit can be calculated by:

$$V = \left(V_s \times \left(\frac{R1}{R1 + R4} \right) \right) - \left(V_s \times \left(\frac{R2}{R2 + R3} \right) \right)$$

Using this formula, and **Figures 2-26** and **2-27** it can be seen that when the beam is in downward tension, R1 and R3 are compressed making their resistance lower, whereas R2 and R4 increase in resistance. The output voltage can then be conditioned to produce an output that can be used to measure the force.

In this assignment the force would be the downward force produced by the weight of an object on the strain gauge.

Very good detail on how a load cell functions

3 – DESIGN OF ASSIGNMENT REQUIREMENTS

The initial requirements for this assignment are to design an 8 bit magnitude comparator, a strain gauge and an ADC. These designs can be found in this section.

3.1 8 BIT MAGNITUDE COMPARATOR DESIGN

The design of the magnitude comparator is simple as it was provided by the 74HC85 datasheet.

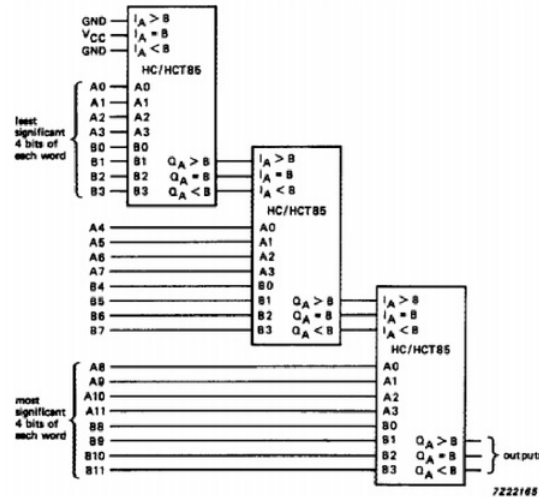


Figure 3-1

By following this design the comparator was built in Multisim. This circuit can be seen in

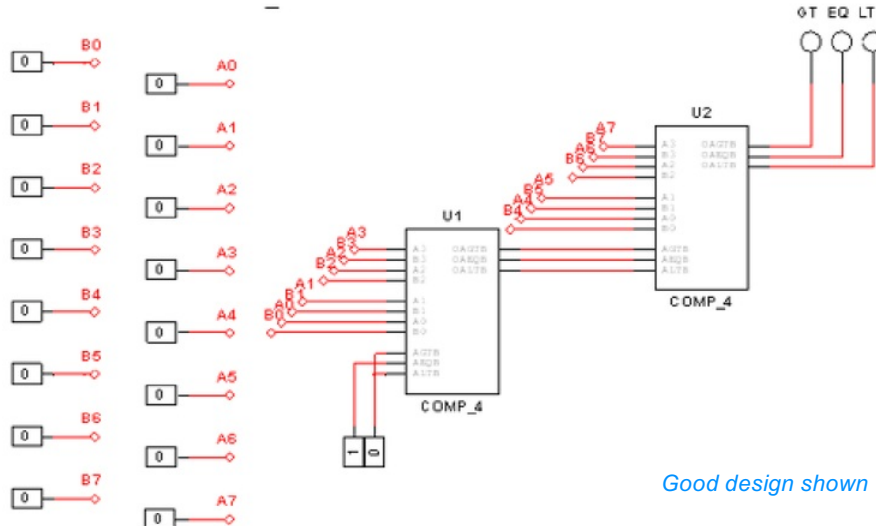


Figure 3-2

There are 8 A inputs and 8 B inputs. The A inputs will be used as the set signal, whereas the B inputs will be used as the variable signal. A0 to A3 and B0 to B3 are connected to the first 4 bit comparator. Cascaded off this comparator is another 4 bit comparator to make the overall circuit an 8 bit comparator. Connected to the second comparator is A4 to A7 and B4 to B7. The outputs of the second comparator are connected to three probes that will indicate whether the B signal is higher than, lower than or equal to the A signal. The initial output state is determined by the digital constants connected to the first comparator which indicate that A is equal to B as the digital constant 1 is connected to AEQB.

The simulated tests for this circuit can be found in **Section 4**. The next step is to adapt this circuit into a PLD circuit that can be downloaded to the FPGA on the nexys board for practical testing.

This adaption is done by replacing the inputs and outputs of the circuit shown in **Figure 3-2** with input and output addresses located on the Nexys4DDR schematic diagram it can be seen that there are switches and LEDs available to be used as inputs and outputs. These can be seen in **Figure 3-3**.

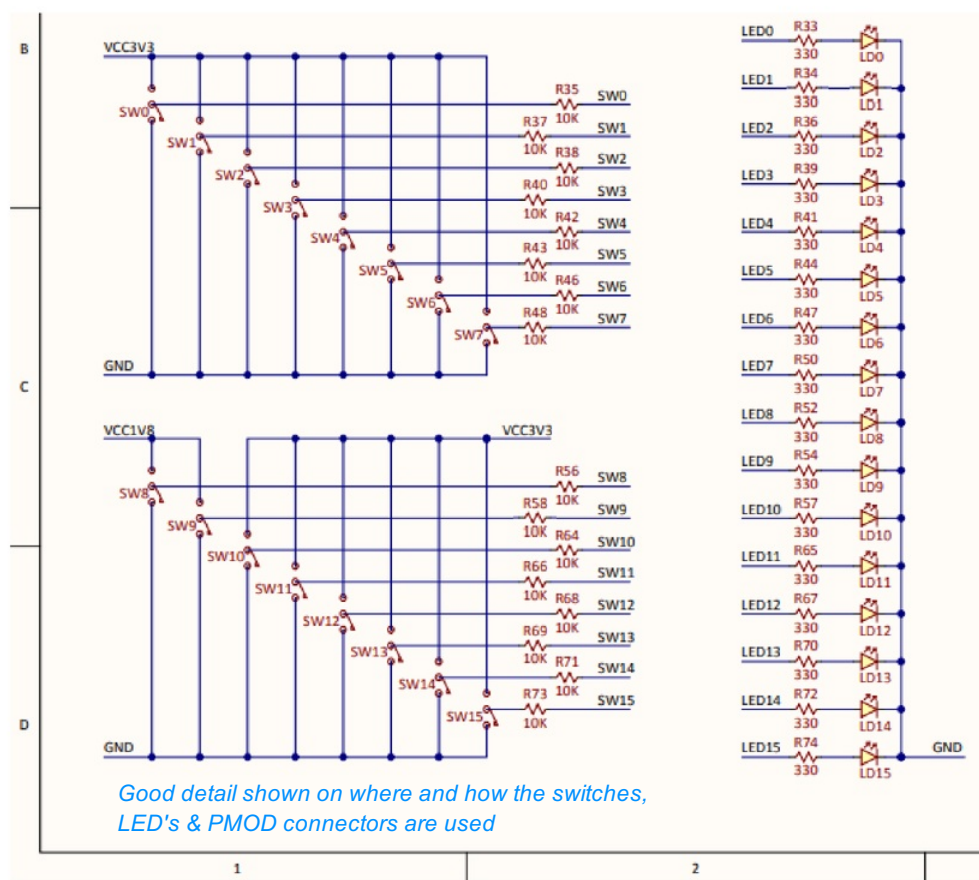


Figure 3-3

The inputs will be replaced with switches, and the outputs will be replaced with LEDs. These replacements can be seen in **Figure 3-4**.

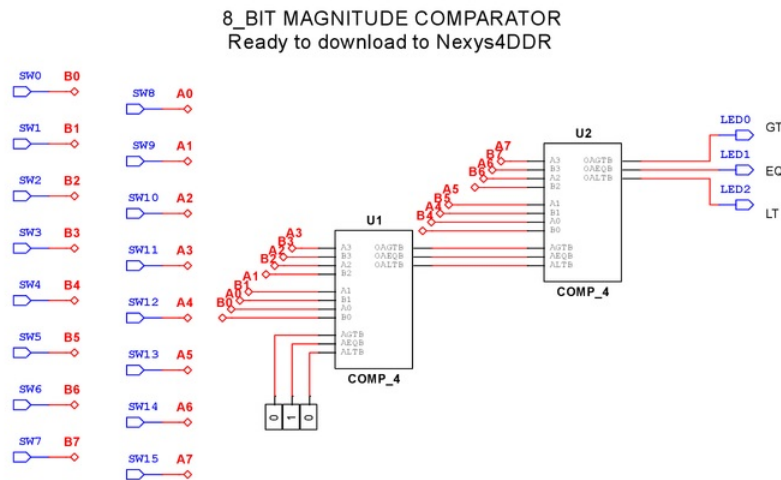


Figure 3-4

With these adjustments made, the circuit is ready to be tested. These practical tests can be found in **Section 5**.

3.2 LOAD CELL STRAIN GAUGE DESIGN

The next requirement was to design a strain gauge. As the researched showed a strain gauge consisted of a Wheatstone bridge configuration of four resistors with variable resistance. Specifically, the strain gauge must measure between 0 to 1kg. To find the required resistance of each of the bridge resistors the datasheet for the LCKD 1kg load cell was looked up. The specifications of this device can be found in **Figure 3-5**.

SPECIFICATIONS

Excitation: 5 Vdc, 7 Vdc max

Output: 2 mV/V nominal

5-Point Calibration: 0%, 50%, 100%, 50%, 0%

Linearity: $\pm 0.25\%$ FSO

Hysteresis: $\pm 0.25\%$ FSO

Repeatability: $\pm 0.10\%$ FSO

Zero Balance: $\pm 2\%$ FSO

Operating Temp Range: -54 to 121°C (-65 to 250°F)

Compensated Temp Range: 16 to 71°C (60 to 160°F)

Thermal Effects:

Span: $\pm 0.018\%$ FSO/°C

Zero: $\pm 0.009\%$ FSO/°C

Safe Overload: 150% of capacity

Ultimate Overload: 300% of capacity

Bridge Resistance: 350 Ω min

Full Scale Deflection: 0.025 to 0.076 mm (0.001 to 0.003")

Electrical Connection: 1.5 m (5') 4-conductor insulated cable with temperature compensation board

Weight: <14 g (<0.5 oz)

Protection Class: IP54

Figure 3-5

(LCKD Design, 2016)

As stated in the specifications for the LCKD 1kg load cell, the bridge resistance is 350Ω. The specification also states that the excitation should be 5V DC. Using this information, a strain gauge circuit can be designed. This can be seen in **Figure 3-6**.

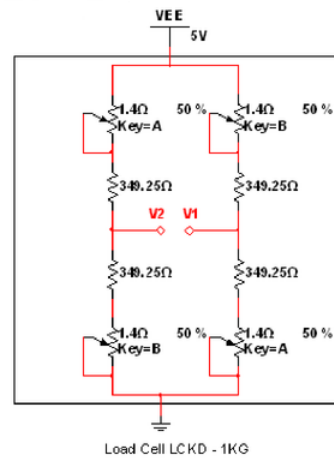


Figure 3-6

The output from the strain gauge is stated to be 2mV/V. In order to make this value readable and range from 0 to 5V where 0V represents 0 grams, and 5V represents 1kg, a gain of 500 is needed. This is proven by:

$$2mV \times A_v = 2mV \times 500 = 1V$$

For an output of 5V the equation would look like:

$$10mV \times 500 = 5V$$

Therefore, signal conditioning on the output of the strain gauge is required. To do this an instrumentation amplifier must be used. The typical design of an instrumentation amplifier can be seen in **Figure 3-7**.

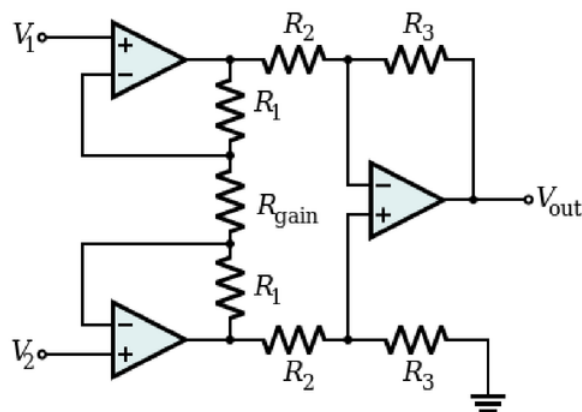


Figure 3-7

The values of the components can be calculated using the following formula:

$$\frac{V_{out}}{V_2 - V_1} = A_v = \left(1 + \frac{2R_1}{R_{gain}}\right) \times \frac{R_3}{R_2}$$

By using inputting values of: $R_{gain} = 20k\Omega$, $R_1 = 20k\Omega$, $R_2 = 10k\Omega$ and $A_v = 500$, The value of R_3 can be calculated.

$$500 = \left(1 + \frac{2(20000)}{20000}\right) \times \frac{R_3}{10000}$$

$$R_3 = 1666k\Omega$$

With these values the instrumentation amplifier can be designed. The design of this amplifier can be seen in **Figure 3-8**.

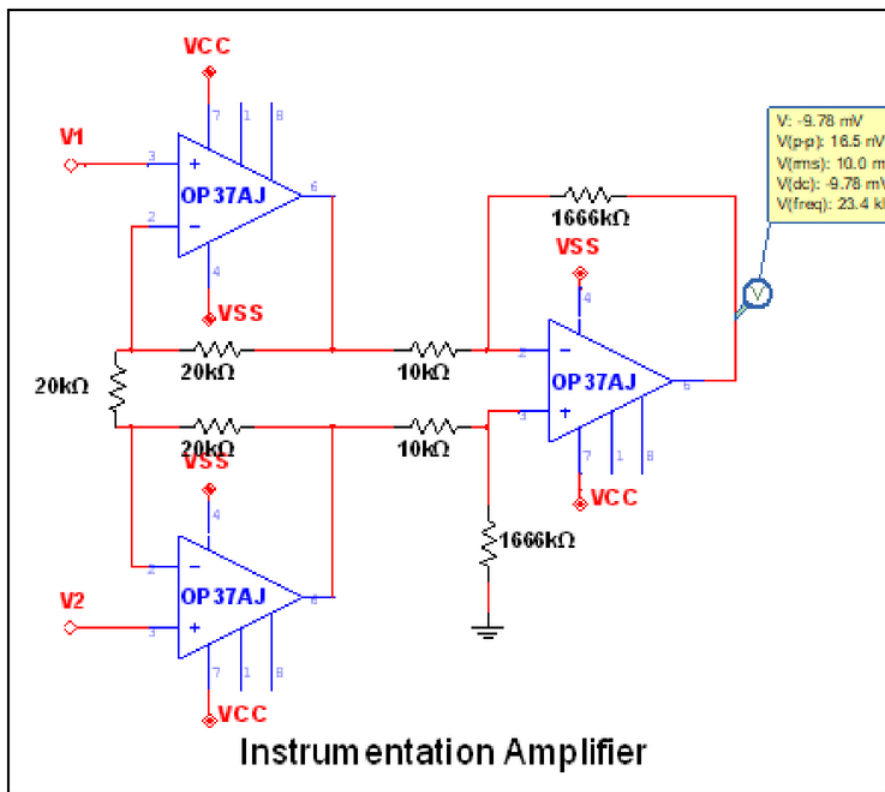


Figure 3-8

The strain gauge and the instrumentation amplifier will be tested together in **Section 4**.

Excellent detailed design shown for the Load cell and instrumentation amplifier

3.3 LABVIEW ADC DESIGN

Using the knowledge obtained in the research about DAQmx, the ADC could be designed. In order to create the ADC an analogue input was required which must be followed by an analogue read. The analogue input would be a power supply which would input a voltage that would be

converted into a floating-point sample by the analogue read. This floating-point would be the analogue input and would be displayed on the front panel as such. This value would then be fed to a formula box where the following formula would convert the analogue value into a digital value.

$$\left(\frac{\text{Analogue Input } X1}{5} \right) \times 256 = \text{Digital Value} \quad \text{Good}$$

This digital value would be displayed on the front panel alongside the analogue input. It is also converted into Boolean via the use of a number to Boolean icon. This Boolean expression is then displayed using an array to show the binary value on the front panel.

Both the analogue input and the digital output branches are cleared and contained within a while loop. A delay function of 1000 ms is added to ensure that the program doesn't use 100% of the CPU's resources while running. A stop button is added to the while loop, and error outs are added to the end of each branch. **Figure 3-9** shows the ADC 'vi' built in LabVIEW.

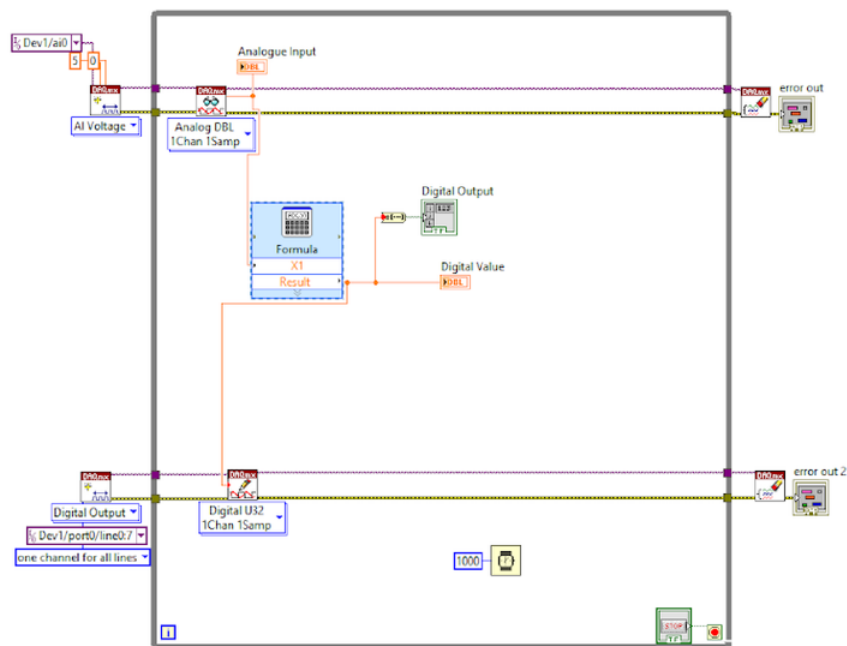


Figure 3-9

When connected to the FPGA the correct channels, ports and lines could be selected on the analogue input and output. This allows the LabVIEW virtual instrument to interface with the FPGA. The graphical interface also known as the front panel was also designed alongside this back-end program. This can be seen in

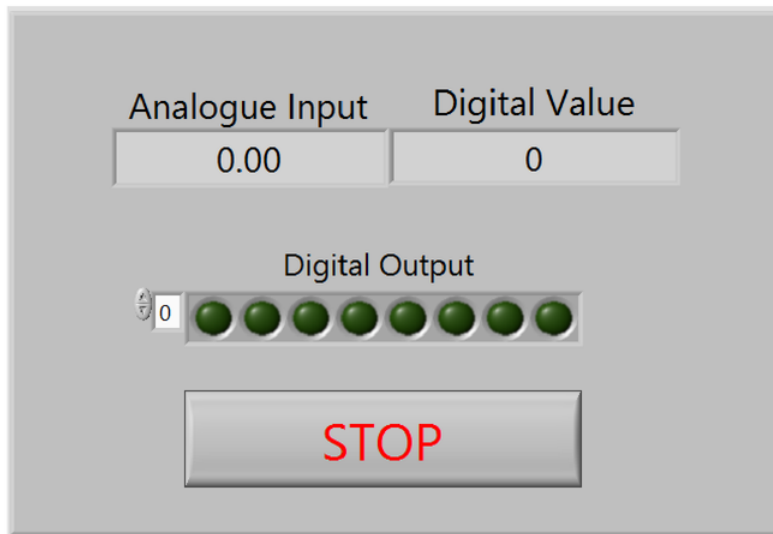


Figure 3-10

As previously stated, both the analogue input and digital value are displayed numerically on the front panel. Below them is the binary display for the digital output. Below that is the stop button for the whole program to be stopped.

The simulated test results can be found in **Section 4**. However, for this program to fulfil the specification, modifications must be made. Firstly, a third branch is added which has a digital input followed by a digital read. The read is connected to an array of 3 elements which will be used to display the comparison icons < = >. The input will be obtained from the output of the 8 bit magnitude comparator which will be downloaded to the FPGA. The digital input will be connected to lines 21 to 23 which will be where the output from the FPGA is connected. This branch is ended once again with a clear and an error out.

The next addition would be an extra formula added to the analogue input branch. This formula is used to calculate the weight of the subject on the strain gauge based on the input voltage. The formula can be seen below:

$$\left(\frac{\text{Analogue Input} \times 1}{5} \right) \times 1000 = \text{Load (gms)} \quad \text{Good}$$

This is then displayed on the front panel. The other formula on the analogue branch remains the same however it is relabelled to 'Actual Load Output'. This will display a binary value on the front panel still.

The digital branch has two formulae added. The first is used to convert the set load value which is input by the user, into a digital value. This can be seen below.

$$\left(\frac{\text{Set Load Value}}{1000} \right) \times 256 = \text{Digitized Set Load Value} \quad \text{Good}$$

This value is then used for two things; it is converted to Boolean form by a number to Boolean icon then stored in an array to be displayed on the front panel in binary form, and as the

input to the next formula. First it is rounded to the nearest integer by a round to nearest icon, then it enters the next formula block.

$$\text{Digitized Set Load Value} \times 256$$

This formula is used to keep the digitized set load value as a multiple of 256 while also making it high enough to not interfere with the other binary displays. From the formula block it is added to the actual load output by an add icon. The output of the add icon is connected to a write icon.

The additions and changes to the program can be seen in **Figure 3-11** and the redesigned front panel can be seen in **Figure 3-12**. The practical tests for this program can be found in **Section 5**.

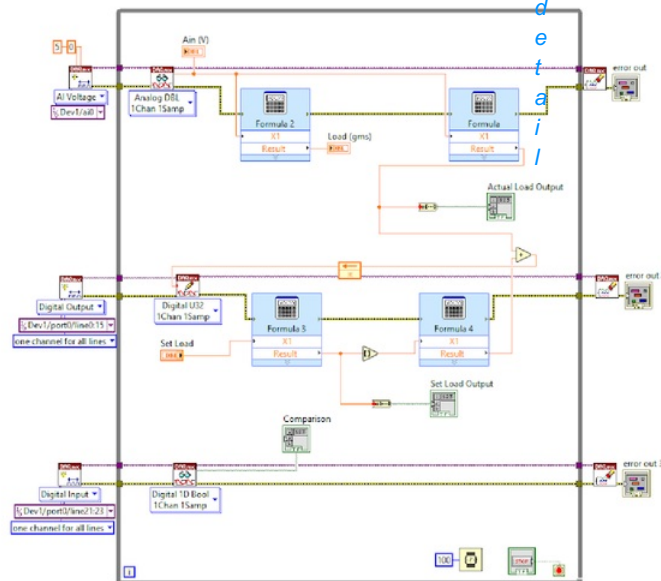


Figure 3-11

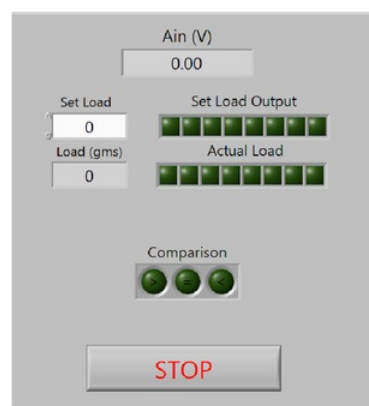


Figure 3-12

Excellent design shown which clearly shows understanding on how the LabVIEW 'vi' functions

4 – ANALYSIS OF SIGNAL CONDITIONING CIRCUIT, NI LABVIEW AND MULTISIM SIMULATED RESULTS

This section will discuss and analyse the simulated results of the previously designed circuits.

4.1 8 BIT MAGNITUDE COMPARATOR SIMULATED TEST RESULTS

The test procedure for this circuit is rather simple as all it requires is a set binary value to be put on the A inputs, and three different binary values to be put on the B inputs; one above, one below and one equal to A. An expected result would be to see the respective probes light up when A is less than, greater than and equal to B.

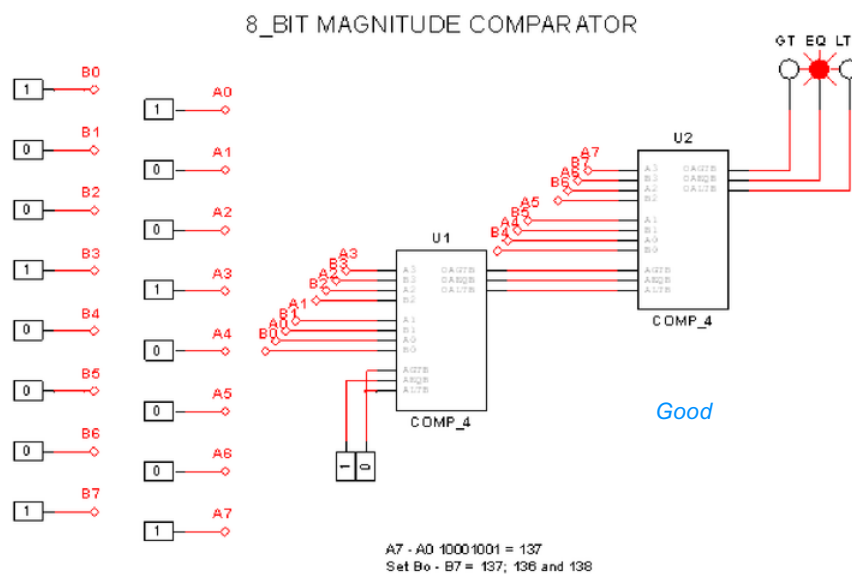


Figure 4-1

This first test has both A and B inputs set to 137, which is 10001001 in binary and can be seen in **Figure 4-1**. As expected, the EQ probe is lit up as both signals have an equal value. The next test is to see what happens when the B inputs are set to 136 which is 10001000 in binary. This can be seen in **Figure 4-2**.

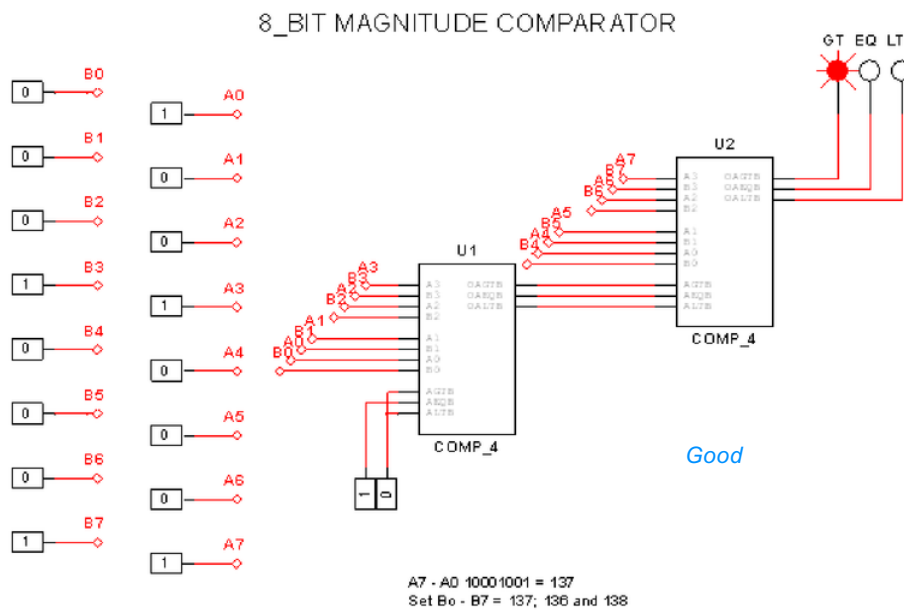


Figure 4-2

Once again, the result is expected. $A > B$ therefore the greater than probe lit up. The next test is to see what happens when B inputs are set to 138, which is 10001010 in binary.

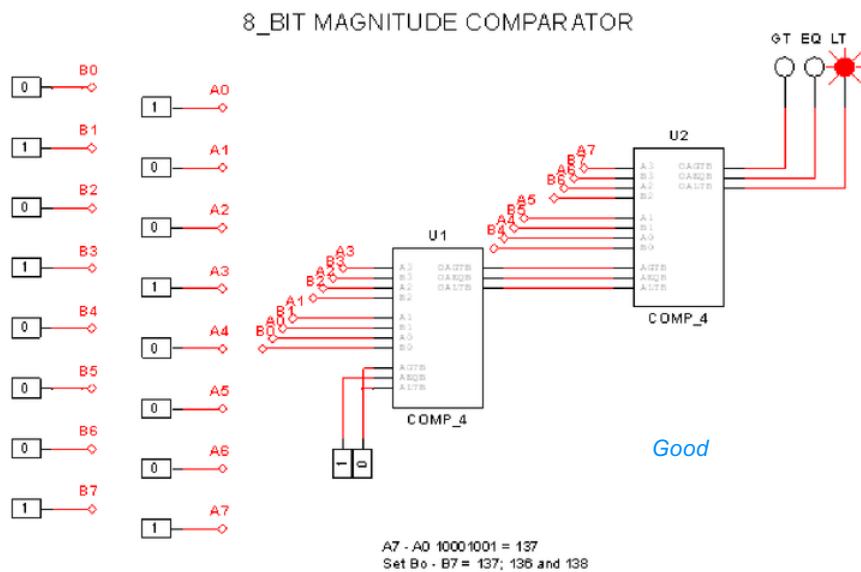


Figure 4-3

Figure 4-3 shows an expected result once again as $A < B$ and the less than probe is lit. This means that the 8 bit magnitude comparator works as the specification states.

Good set of MultiSim simulated results which have been correctly analysed

4.2 STRAIN GAUGE LOAD CELL SIMULATED TEST RESULTS

To test the strain gauge, a series of different loads are placed on the Wheatstone bridge of potentiometers to increase the voltage. The load is increase in 100g increments meaning 10 readings are obtained from the output of the instrumentation amplifier. An expected result would be to see a linear relationship between the load and the output voltage where the output voltage ranges from 0V to 5V. **Figure 4-4** shows both the Load Cell and the Instrumentation Amplifier connected to each other. A test table can be found below with the results from the 10 tests.

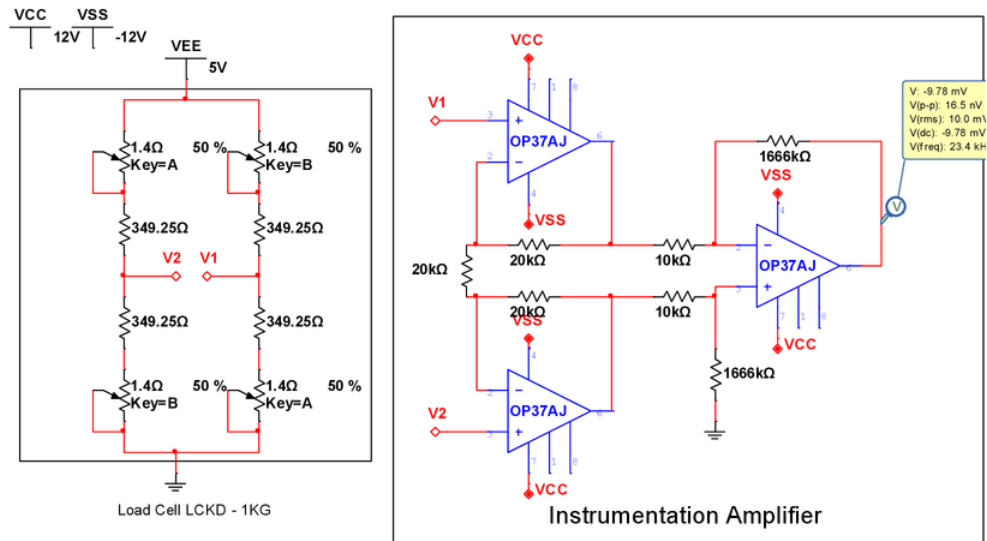


Figure 4-4

Load (gms)	Vout (V)
0	0
100	0.49
200	0.99
300	1.49
400	1.99
500	2.49
600	2.99
700	3.49
800	3.99
900	4.49
1000	4.99

Table 2

The results from Table 2 can produce a graph to show the relationship between Load and Vout. This can be seen in **Figure 4-5**.

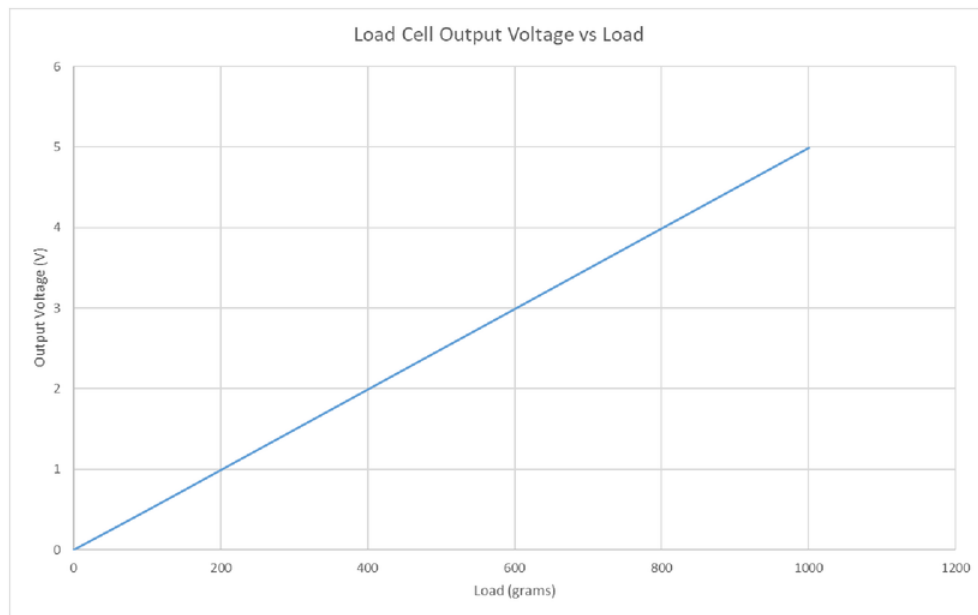


Figure 4-5

As expected, the relationship between load and output voltage was linear and ranged between 0V and 5V. This is as the specification stated and therefore the circuit works correctly.

Good graphical analysis shown

4.3 LABVIEW ADC SIMULATED TEST

The final test in this section is to test the ADC designed in LabVIEW. This is done by inputting three different analogue inputs into the program and checking to see if they are being converted into digital and binary values correctly. The first test can be seen below.

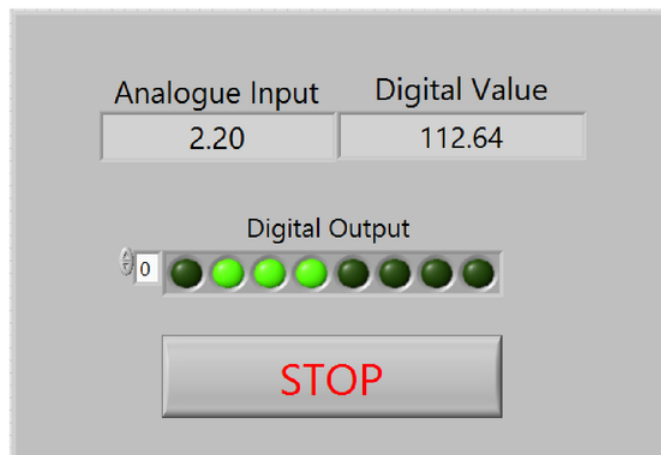


Figure 4-6

The analogue input signal is 2.2V. The conversion to a digital value can be seen as follows, along with the conversion from digital to binary.

$$\frac{2.2}{5} \times 256 = 112.64$$

$$112.64 = 01110000$$

Both results were accurate. The next test can be seen below.

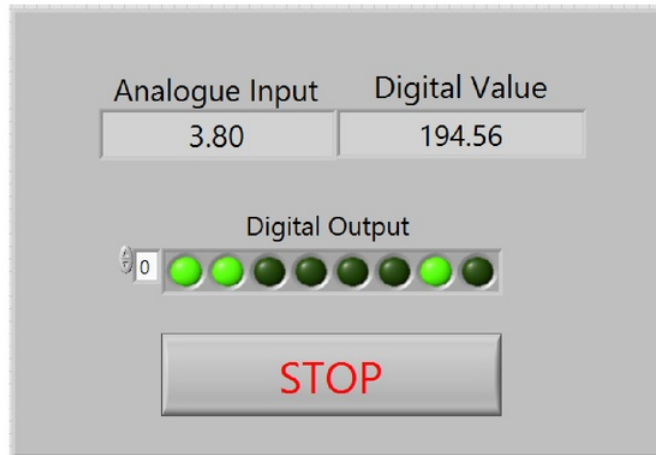


Figure 4-7

With an analogue input signal of 3.8V, the digital and binary value can be calculated.

$$\frac{3.8}{5} \times 256 = 194.56$$

$$194.56 = 11000010$$

Once again the results corresponded to the calculations. The final test can be found below.

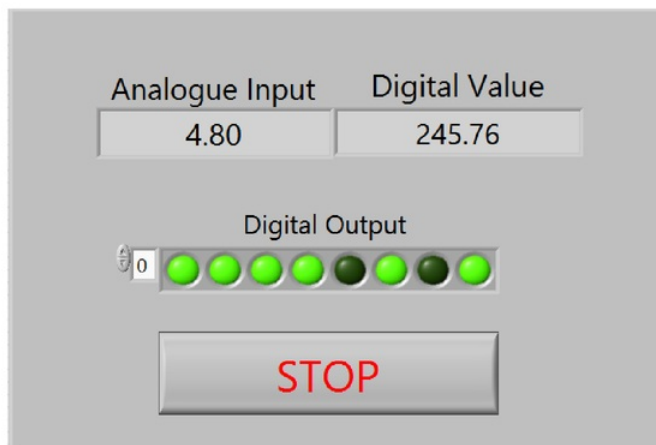


Figure 4-8

The calculations for this test can be found as follows.

$$\frac{4.8}{5} \times 256 = 245.76$$

$$245.76 = 11110101$$

All three tests were verified by calculations to prove that the ADC built in LabVIEW worked. The next step is to test the magnitude comparator and the adjusted LabVIEW program practically on the FPGA, and the NI ELVIS board.

Good analysis shown of 8 bit ADC results

5 – PRACTICAL BUILD AND TEST VIA NI ELVIS BOARD

This section will detail the testing of the magnitude comparator and the LabVIEW program on the nexys4DDR and NI ELVIS boards.

5.1 8 BIT MAGNITUDE COMPARATOR PRACTICAL TEST

As shown previously in section 3, the 8 bit magnitude comparator was redesigned to be able to be downloaded and tested to an FPGA by replacing the inputs and outputs. The PLD design can be seen in **Figure 5-1**.

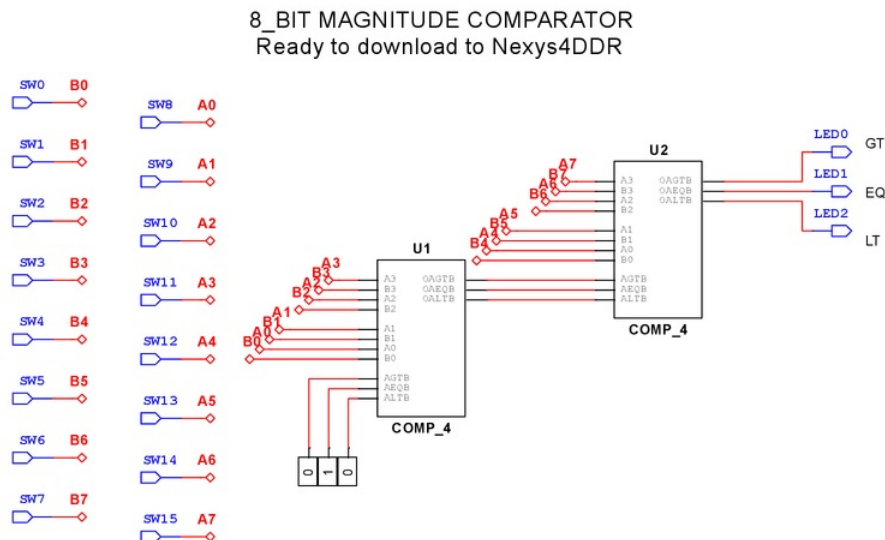


Figure 5-1

To practically test this, it must be connected to the PC and the circuit must be downloaded to the board. Once downloaded, the switches 0-7 (B Ports) will act as the set point signal, and the switches 8-15 (A Ports) will act as the variable signal. Three different variable signals will then be chosen, one greater than, one equal to and one less than the set point signal. LED0 should light up when the variable is greater than, LED1 should light up if the variable is equal to, and LED2 should light up when the variable is less than the set point signal.

The first test can be seen in **Figure 5-2** which shows both signals as equal. The binary value for both signals is 10001001 which is equal to 137. The expected result would be to have LED1 light up.

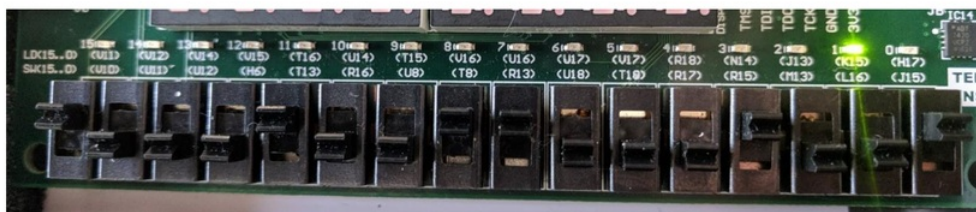
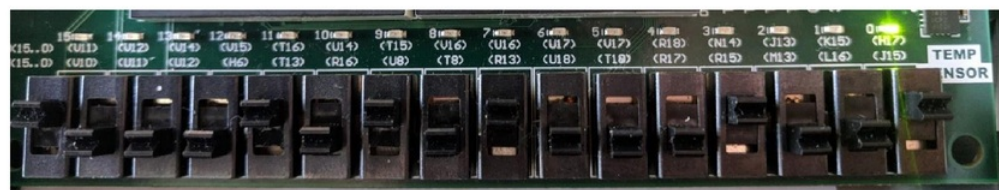


Figure 5-2

Good

Once again the expected result was correct. The final test checks to see if LED0 lights up when switches 8-15 are set to 10001010.



The final test proved that the magnitude comparator worked as intended. This allows the next test to be carried out.

5.2 LABVIEW PROGRAM TEST ON THE NI ELVIS II BOARD

5.2.1 Nexys4DDR FPGA Board

From the Nexys4DDR datasheet it can be seen that there are five sets of Pmod ports. These ports can be used to connect the FPGA to the ELVIS board. This will allow LabVIEW to communicate and interface with the FPGA. The details of the Nexys4DDR board can be seen in

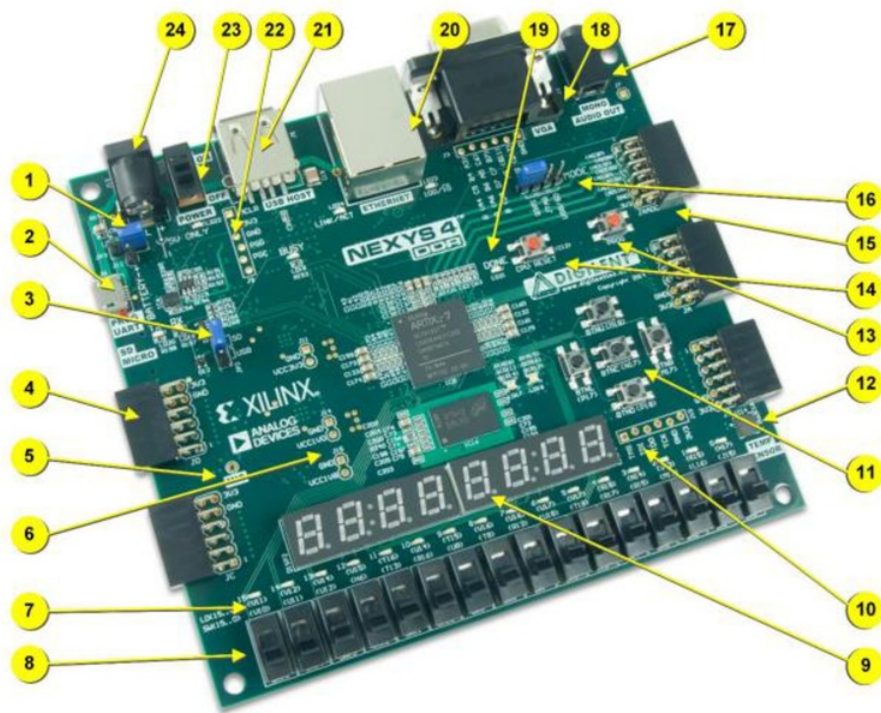


Figure 1. Nexys4 DDR board features.

Callout	Component Description	Callout	Component Description
1	Power select jumper and battery header	13	FPGA configuration reset button
2	Shared UART/ JTAG USB port	14	CPU reset button (for soft cores)
3	External configuration jumper (SD / USB)	15	Analog signal Pmod port (XADC)
4	Pmod port(s)	16	Programming mode jumper
5	Microphone	17	Audio connector
6	Power supply test point(s)	18	VGA connector
7	LEDs (16)	19	FPGA programming done LED
8	Slide switches	20	Ethernet connector
9	Eight digit 7-seg display	21	USB host connector
10	JTAG port for (optional) external cable	22	PIC24 programming port (factory use)
11	Five pushbuttons	23	Power switch
12	Temperature sensor	24	Power jack

Figure 5-5

(Nexys4™ FPGA Board Reference Manual, 2013)

5.2.2 NI ELVIS II Development Board

By looking at the NI ELVIS II datasheet, a parts locator diagram can be found for the prototyping board. This can be seen in **Figure 5-6**

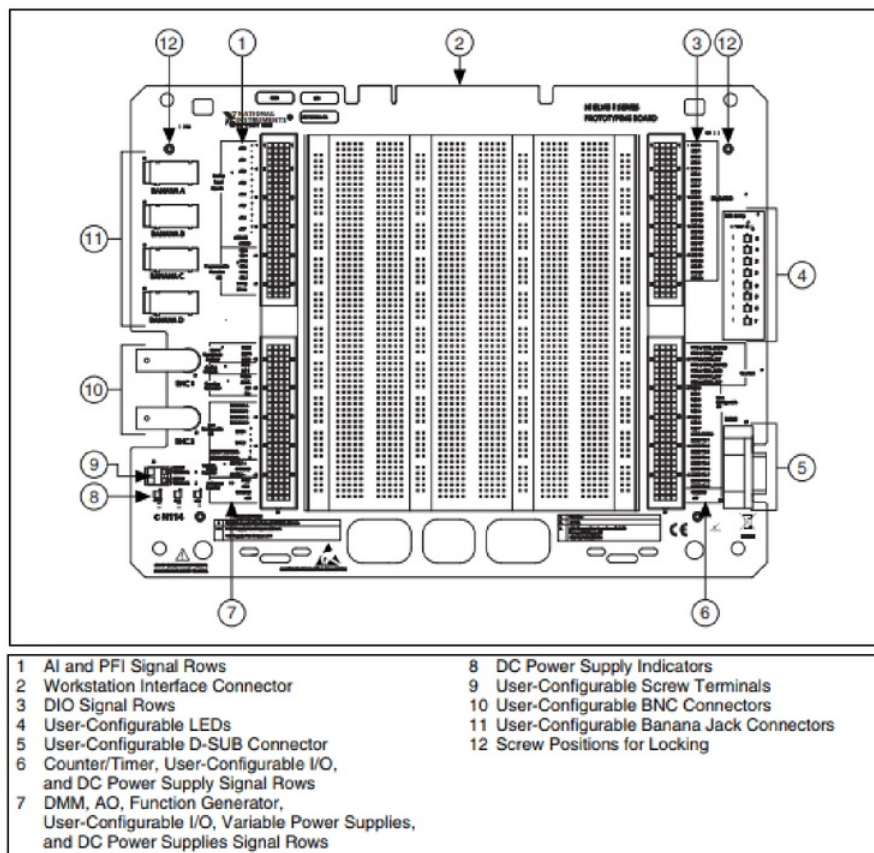


Figure 5-6

(NI Educational Laboratory Virtual Instrumentation Suite II Series (NI ELVISTM II Series) User Manual, 2011)

5.2.3 Practical Setup

After looking at both the Nexys and ELVIS boards, the practical could be set up. After connecting the prototyping board to the workstation, the FPGA needs to be interfaced to the prototyping board. This is done by using jumper cables going from the ELVIS board into the Pmod ports on the Nexys board.

The two signals are transferred from the ELVIS board to the Nexys board through I/O pins 0-15. I/O pins 0-7 is connected to Pmod ports JA, while 8-15 are connected to JB. The output of the FPGA comes from the Pmod ports JC and go back to I/O pins 21-23 on the ELVIS board. The connections can be seen in

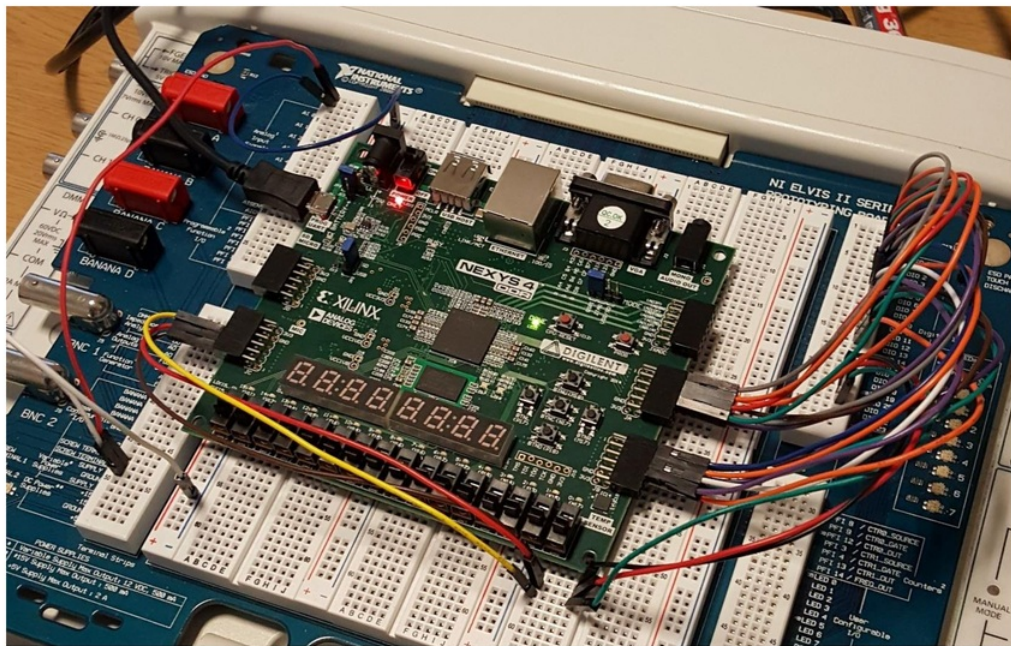


Figure 5-7

The magnitude comparator is then downloaded to the FPGA and the LabVIEW program is run. The analogue input is created by a virtual power supply which is part of the national instruments test bench. The test results for this practical can be found in **Section 6**.

[Very good detail on the practical setup of the NI ELVIS unit, the FPGA and the LabVIEW 'vi'](#)

6 – ANALYSIS OF NI LABVIEW / FPGA RESULTS

The test for the LabVIEW program interfacing with the FPGA magnitude comparator involves the same steps as the previous tests. Using the virtual variable power supply as the analogue input, three different voltages are used, above, below and equal to the set point which is selected in LabVIEW. The input and output are both displayed along, along with a numerical value of the load and set point and a binary value of the load and set point.

The first test is to test an equal value. The set load is set in LabVIEW to be 567 grams. The analogue input is increased until it the measured load equals the set load. At this point the comparison indicator should indicate equal and the binary values for both load values should be the same. This first test can be seen in **Figure 6-1**.



Figure 6-1

To ensure accuracy, calculations can be made:

$$Load = \frac{2.83}{5} \times 1000 = 566g$$

$$Digital Value = \frac{566}{1000} \times 256 = 144.896$$

$$144.896 = 10010001 \quad G$$

The only error during this test is that the binary indicators are the wrong way around, so the calculated binary value is mirrored. Everything else works as intended. This error is likely to persist through the next two tests but is an easy fix. The second test can be seen in **Figure 6-2**. For this test the input analogue voltage was increased so that the load exceeded the set load. An expected result would be to see a higher numerical load value, binary load value, and the comparison indicator showing greater than.

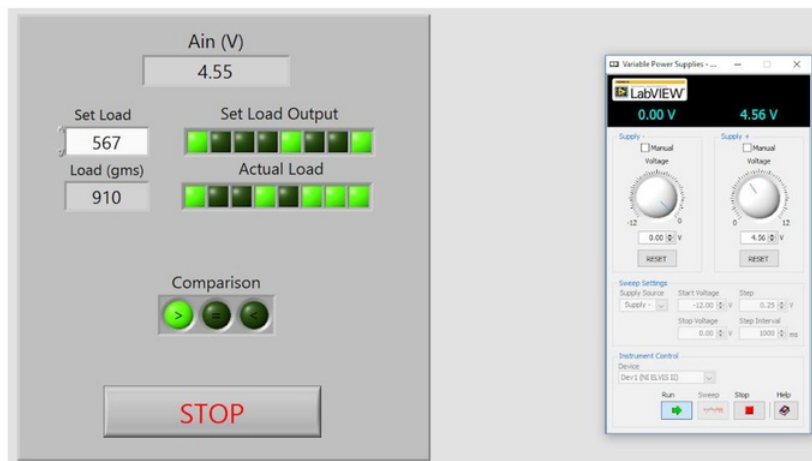


Figure 6-2

As expected the numerical value is larger than the set point and the comparison indicator shows greater than. The calculations to ensure accuracy can be seen below.

$$\text{Load} = \frac{4.55}{5} \times 1000 = 910g$$

$$\text{Digital Value} = \frac{566}{1000} \times 256 = 232.96$$

$$232.96 = 11101001 \quad \text{Good}$$

Just like the previous test, the only error was that the binary indicators are mirrored. Both the comparison indicator and the numerical load value are correct. The binary value is also correct, only the indicator is the wrong way around on the front panel. The third test involves changing the variable power supply to a lower value than the set point. The expected result here would be to have the comparison indicator show less than, and the numerical and binary value of the load should be lower than the set point. **Figure 6-3** shows the final test.

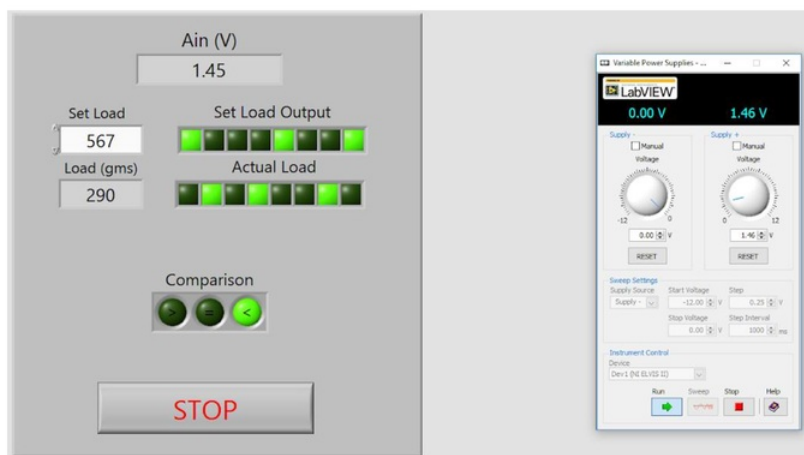


Figure 6-3

The comparison indicator shows that the value is less than, and the numerical and binary values show that they are also less than the set point. Once again calculations can be made to ensure the accuracy of the displayed results.

$$Load = \frac{1.45}{5} \times 1000 = 290g$$

$$Digital\ Value = \frac{290}{1000} \times 256 = 74.24$$

$$232.96 = 01001010$$

These three sets of results prove that the LabVIEW program and 8 bit magnitude comparator are interfacing with each other and operating as intended. The only error in the design is that the binary indicators are the wrong way around, meaning they display a mirrored version of the binary value where the MSB is on the right, and the LSB is on the left.

Very good set of practical results which have been analysed very well

7 – ADAPT DESIGN TO FIT NEW PURPOSES OR APPLICATIONS

This section will detail the adaptation of the previous circuit to enable data logging. The ability to log the data from the input load, the set load and the comparison outputs will enable the production of a graph which will show how each of the previously mentioned factors effect each other.

To modify the program to enable data logging, 'Write To Measurement File' icons were added to the parameters needed to be measured. These icons write text files which store the values to later be used in excel to produce a graph. These can be seen in **Figure 7-1**.

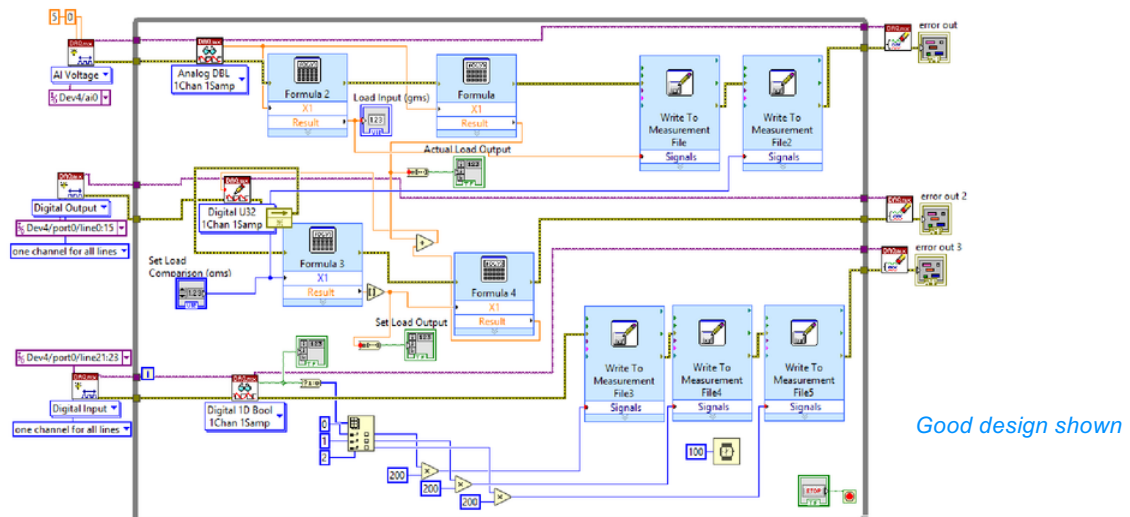


Figure 7-1

The top two write to measurement file icons are connected to the load input, and the set load. The bottom three are connected to multiply icons which are simply there to take the values to 200 when the comparison variable they are linked to is at a logic 1. When they are at a logic 0 they simply record 0s. This allows the three comparison variables to be viewed when plotted on a graph.

The input load is increased from 0g to just over 300g when the set point is set to 300g. When the input load is below the set point the measurement file linked to less than is at 200. When the input load becomes equal to the set point, the less than write file stops writing 200s to the text file, and starts writing 0s. At the same moment, the equal to write to measurement file starts writing 200s. This creates a change in the graph to indicate that the input load is equal to the set point.

When the input load exceeds the set load, the greater than write to measurement file starts writing 200s to the text file, and the equal to write to measurement file starts writing 0s again. This graph can be seen in

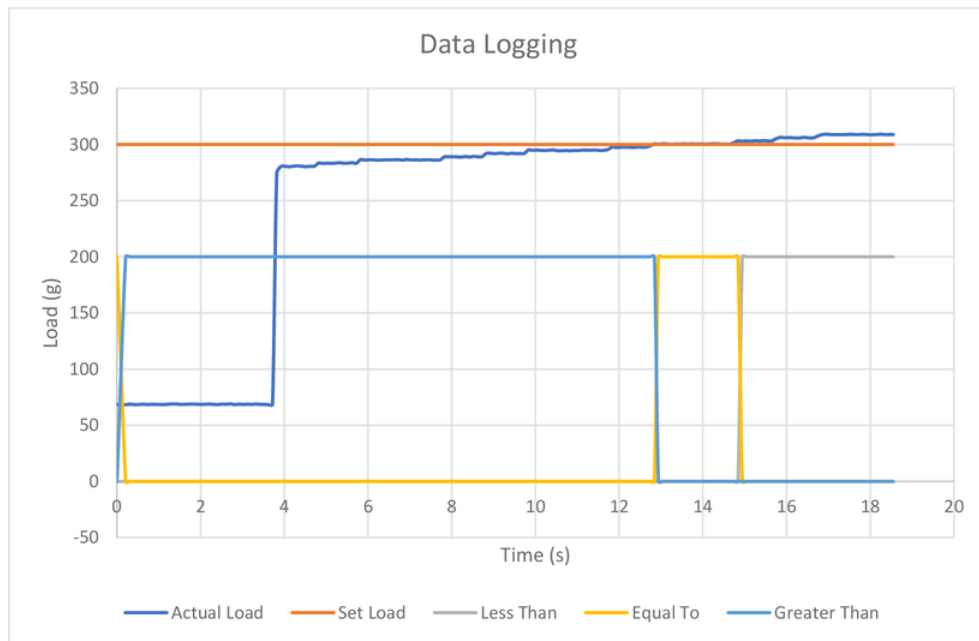


Figure 7-2

Good graphical analysis shown which has been explained well

8 – CONCLUSIONS

Data acquisition is key in industry as more data allows for more efficient processes which could result in energy, time and money saved.

The use of LabVIEW in conjunction with Multisim create a very powerful combination of design tools capable of designing almost any data acquisition system. With the ability to design systems like this, adapting designs to be applicable to other applications is made easy.

An example of another application could be to keep measure the level of a tank, rather than the weight of something. This design could easily be modified to be a compare the levels of liquids, or temperatures or almost any other variable.

Throughout the assignment the design process went well. Every design that was tested operated as the specification required. The use of datasheets was critical in the design of the magnitude comparator, strain gauge and instrumentation amplifier. They also played a critical role in the practical setup as it provided information about the architecture of both the Nexys and ELVIS boards.

The only problem to occur was that the binary value indicators on the final LabVIEW front panel were the wrong way around, which created a mirrored binary value. This problem is a simple fix however and did not cause the program to run incorrectly.

A recommendation for the assignment would be to obtain and use a strain gauge load cell in the practical testing to make sure that it interfaced correctly with the rest of the design.

Good, detailed conclusion given with some reflection

9 – APPENDICES

Zone.ni.com. (2018). NI-DAQmx Overview - NI-DAQ™mx Help - National Instruments. [online] Available at: <https://zone.ni.com/reference/en-XX/help/370466AD-01/daqhelp/nidaqoverview/> [Accessed 16 Apr. 2018].

74HC85 4 bit Magnitude Comparator. (1990). 1st ed. [ebook] Southhampton: Phillips Semiconductors, pp.3,4,5,8. Available at: <http://www.learnabout-electronics.org/Downloads/74HC85.pdf> [Accessed 13 Apr. 2018].

Rowe, M. (2002). How do ADCs work?. [online] EDN. Available at: <https://www.edn.com/design/test-and-measurement/4383626/How-do-ADCs-work-> [Accessed 13 Apr. 2018].

Techopedia.com. (2017). What is the Nyquist Frequency? - Definition from Techopedia. [online] Available at: <https://www.techopedia.com/definition/21075/nyquist-frequency> [Accessed 14 Apr. 2018].

Cao, Z. and Yan, S. (2008). Low-power high-speed ADCs for nanometer CMOS integration. Dordrecht: Springer Science + Business Media B.V.

Rdpe.com. (2013). How it Works - Strain Gauge Load Cell. [online] Available at: <http://www.rdpe.com/ex/hiw-sglc.htm> [Accessed 14 Apr. 2018].

Omega.co.uk. (2016). compression load cell, subminiature load cell. [online] Available at: <https://www.omega.co.uk/pptst/LCKD.html> [Accessed 15 Apr. 2018].

Nexys4™ FPGA Board Reference Manual. (2013). 2nd ed. [ebook] Pullman: Digilent, p.2. Available at: https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Supporting%20Material/Nexys4_RM_VB1_Final_3.pdf [Accessed 16 Apr. 2018].

NI Educational Laboratory Virtual Instrumentation Suite II Series (NI ELVISTM II Series) User Manual. (2011). 1st ed. [ebook] Austin: National Instruments, pp.2-3. Available at: <http://www.ni.com/pdf/manuals/374629c.pdf> [Accessed 16 Apr. 2018].

Sam Harries - Assignment 2 Final

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

77/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

Text Comment. Excellent introduction to the assignment

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

Text Comment. Very good detailed research carried out on the DAQmx icons

PAGE 11

PAGE 12

Text Comment. Very good understanding shown on how the comparator functions

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

Text Comment. Excellent detail on ADC research

PAGE 18

Text Comment. Very good detail on how a load cell functions

PAGE 19

Text Comment. Good design shown

PAGE 20

Text Comment. Good detail shown on where and how the switches, LED's & PMOD connectors are used

PAGE 21

PAGE 22

PAGE 23

Text Comment. Excellent detailed design shown for the Load cell and instrumentation amplifier

PAGE 24

Text Comment. Good

PAGE 25

Text Comment. Good

Text Comment. Good

PAGE 26

Text Comment. Excellent detail

Text Comment. Excellent design shown which clearly shows understanding on how the LabVIEW 'vi' functions

PAGE 27

Text Comment. Good

PAGE 28

Text Comment. Good

Text Comment. Good

Text Comment. Good set of MultiSim simulated results which have been correctly analysed

PAGE 29

PAGE 30

Text Comment. Good graphical analysis shown

PAGE 31

PAGE 32

Text Comment. Good analysis shown of 8 bit ADC results

PAGE 33

Text Comment. Good

PAGE 34

Text Comment. Good set of practical results and compared successfully with the MultiSim simulated results

PAGE 35

PAGE 36

PAGE 37

Text Comment. Very good detail on the practical setup of the NI ELVIS unit, the FPGA and the LabVIEW 'vi'

PAGE 38

Text Comment. Good

PAGE 39

Text Comment. Good

PAGE 40

Text Comment. Very good set of practical results which have been analysed very well

PAGE 41

Text Comment. Good design shown

PAGE 42

Text Comment. Good graphical analysis shown which has been explained well

PAGE 43

Text Comment. Good, detailed conclusion given with some reflection

PAGE 44

GRADING FORM: CSG-RUBIC

DAVID HARRIES

77

RESEARCH-INFORMED LITERATURE



Analogue / Digital Converters: Excellent detail on ADC research Load Cells: Very good detail on how a load cell functions Digital Magnitude Comparators: Very good understanding shown on how the comparator functions NI DAQmx in LabVIEW: Very good detailed research carried out on the DAQmx icons

16

KNOWLEDGE AND UNDERSTANDING



Design of the signal conditioning circuit for LCKD-1Kg Load Cell: Excellent detailed design shown for the Load cell and instrumentation amplifier (4/5) Design of LabVIEW 'vi': Excellent design shown which clearly shows understanding on how the LabVIEW 'vi' functions (8/10) Design of 4/8-bit

23

Magnitude Comparator: Good detail shown on where and how the switches, LED's & PMOD connectors are used on the FPGA. Good MultiSim design shown. (8/10) Re-design to suit new applications: Good design shown but not explained fully (3/5)

ANALYSIS



Analysis of signal conditioning circuit: Good graphical analysis shown (4/5) Analysis of MultiSim Results: Good set of MultiSim simulated results which have been correctly analysed. Good set of practical FPGA results and compared successfully with the MultiSim simulated results (4/5) Analysis of NI LabVIEW / FPGA results: Very good set of practical results which have been analysed very well (8/10) Analysis of data logging files in EXCEL: Good graphical analysis shown which has been explained well (3/5)

19

PRACTICAL APPLICATION AND DEP.



Build interface between LabVIEW and the FPGA board via the NI ELVIS 11 board: Very good detail on the practical setup of the NI ELVIS unit, the FPGA and the LabVIEW 'vi' (7/10)

7

SKILLS FOR PROFESSIONAL PRAC.



Deployment of appropriate media: Excellent use made of screen dumps and pictorial evidence in design and analysis. (4/5) Report layout and readability: Very good report structure which covered all aspects of the assignment (4/5) Appropriate conclusion: Excellent introduction to

12

the assignment. Good, detailed conclusion given
with some reflection (4/5)