

S Dedoncker PSDE 5029 FINAL

Ass 2

by Scott Dedoncker

FILE	12968_SCOTT_DEDONCKER_S_DEDONCKER_PSDE_5029_FINAL_ASS_2_157347_956996663.DOCX		
TIME SUBMITTED	26-APR-2018 09:22AM (UTC+0100)	WORD COUNT	5877
SUBMISSION ID	87506439	CHARACTER COUNT	32095

HND Electrical & Electronic Engineering

PSDE 5029: Virtual Instrumentation & Control



Assignment 2

NI Labview Magnitude Comparator

Scott Dedoncker

Valero Energy Electrical Apprentice

Date of Submission: 25/04/18

Word Count: 5641

Contents

Introduction	3
1. Research.....	4
1.1. NI DAQmx Data Acquisition Icons in LabVIEW.....	4
1.2. 8-Bit magnitude comparator	6
1.3. Analog to Digital Conversion.....	7
1.4. Transducers and Load Cells.....	8
3. Analysis of the Signal Conditioning Circuits	11
3.1. 8-bit Magnitude Comparator.....	11
3.2. Instrumentation Amplifier	13
4. Practical Build and Test via NI Elvis Board	15
4.1. Comparator to Labview .vi Simulation.....	15
4.2. 8-Bit Comparator with Binary Conversion LabVIEW .vi	20
5. Adapt The Design to Fit New Purposes and Analysis of Excel Data	25
6. Conclusions	27
Works Cited.....	28
Appendix	29

Figure 1 - Simple Magnitude Comparator (Poudel, n.d.)	6
Figure 2 - Comparator Truth Table (Poudel, n.d.)	6
Figure 3 - Arduino Board (Nate, n.d.)	7
Figure 4 - LCKD 1KG Compression Load Cell	8
Figure 5 - Equal To	9
Figure 6 - Less Than	10
Figure 7 - Greater Than	10
Figure 8 - Adapted Comparator	11
Figure 9 - LED 1 Equal	11
Figure 10 - LED 0 Greater Than	12
Figure 11 - LED 2 Less Than	12
Figure 12 - Comparator 2	13
Figure 13 - Comparator 1	13
Figure 14 - Comparator Results	14
Figure 15 - Voltage out against Load	14
Figure 16 - Second Converted 8-bit Magnitude Comparator	15
Figure 17 - .vi One	16
Figure 18 - .vi One Interface	16
Figure 19 - Voltage to Binary Test 1	17
Figure 20 - Voltage to Binary Test 2	18
Figure 21 - Voltage to Binary Test 3	19
Figure 22 - .vi to FPGA Build	20
Figure 23 - Comparison .vi Interface	21
Figure 24 - 8-Bit Magnitude comparator Adapted	21
Figure 25 - Comparison Simulation 1	22
Figure 26 - Comparison Simulation 2	23
Figure 27 - Comparison Simulation 3	24
Figure 28 - FPGA connected to the NI Elvis II Board	24
Figure 29 - Excel Comparison .vi	25
Figure 30 - 8-bit Magnitude Comparator .vi	26
Figure 31 - Data Logging	26

Introduction

The overall purpose of this assignment is to investigate, by researching and performing practical experiments on how an 8-bit digital magnitude comparator works, the circuit build behind it and how it interfaces with other programs. This will involve research into the following key areas of this topic:

- NI DAQmx Data Acquisition Icons used in LabVIEW .vi
- Digital Magnitude Comparators
- Analogue to Digital Converters
- Load Cell Transistors

The research will need to be detailed, structured well and coherent. It must also show a very detailed and good understanding of the points raised.

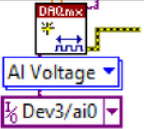
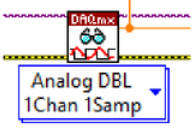
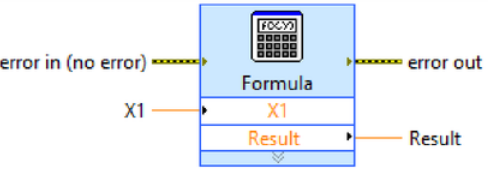
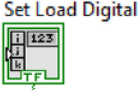
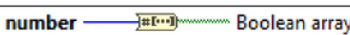
The knowledge gained from the research section of the assignment must be studied and then applied to a set of practical experiments involving a LabVIEW magnitude comparator. This would involve:

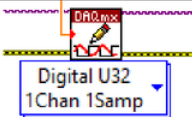
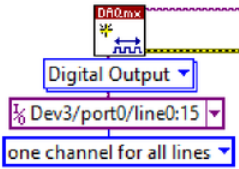
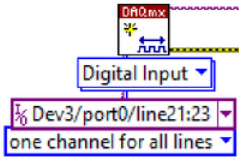
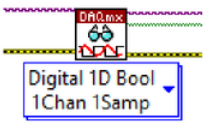
- Building a LabVIEW .vi that interfaces to an FPGA via a NI ELVIS II board to produce a digital comparison of two 8-bit digital signals.
- Allow this digital output to be presented in the forms of either greater than, equal to, or less than.
- Build an 8-bit magnitude comparator in Multisim and adjust it to suit different purposes
- Test run the comparator on the Nexys 4 DDR FPGA board.
- Design and test a signal conditioning circuit in Multisim to simulate a LCKD 1kg Load Cell.
- Measure the final LabVIEW .vi using Excel.

1. Research

1.1. NI DAQmx Data Acquisition Icons in LabVIEW

National instruments software is a platform designed in order to build and simulate visual programming language. This specific program is designed for data acquisition purposes by building files named “.vi”. This software incorporates but is not limited to aspects of dataflow programming and graphical programming. Within this assignment as block diagrams were used to control the “.vi”. These block diagrams used various components such as DAQmx components to control and make the circuit function. These components are explained below.

DAQmx Icons	Description
	Create Channel – used to create channels that measure voltage. Can be used with the AI custom voltage to gain measurements using internal or scaled excitation voltage.
	Read (Analog Double 1 channel sample) – used to read a single floating-point value from an input that has a single analog input channel.
	Formula – inputted formulae from the users can be used to calculate mathematical functions. Very similar to a scientific calculator.
	Analogue Input – returns a simple date sample. Works in double values 64 bit.
	Set Load Digital Output – opens an array to allow Boolean data to be inputted.
	Clear Task – Clears the current task. This .vi also releases any resources reserved for this task. Once this task is used and cleared it needs to be created again.
	Error Out – Contains error information.
	Number to Boolean Array – Converts an integer or fixed-point number to a Boolean array. These arrays can have 8, 16, 32 or 64 elements to them.
	Feedback Node – stores information in loop iterations for the next vi.

 DAQmx Write (Digital U32 1 channel sample)	DAQmx Write (Digital U32 1 channel sample) – Writes an integer in a single 32 bit form to a task that includes a single digital output channel.
 DAQmx Create Channel (DO-Digital Output)	DAQmx Create Channel (DO-Digital Output) – This Vi creates channels to generate digital signals for outputs.
 DAQmx Create Channel (DI-Digital Input)	DAQmx Create Channel (DI-Digital Input) – This Vi creates channels to measure digital signals for inputs.
 DAQmx Read (Digital 1D Boolean 1 Channel Sample)	DAQmx Read (Digital 1D Boolean 1 Channel Sample) – This Vi reads a boolean array containing values that contain a single digital input channel
Wait (ms)  milliseconds to wait	Wait (Milliseconds) – This task waits for a set amount of milliseconds entered into it from the left leg.

Very good detailed research on DAQmx icons used in assignment

1.2. 8-Bit magnitude comparator

For the practical experiment in this an assignment, it will involve building an utilising an 8-bit magnitude comparator. An 8-bit magnitude comparator is a type of device capable of high-speed arithmetic and logic comparison functions. It takes two 8-bit values and assesses them before outputting on one of three output flags. In this assignments case those flags would be greater than, equal to or less than. To summarise these devices are built for taking a random value (fig.1 A), and comparing it to a set value (fig.1 B).

(Texas Instruments, 1995)

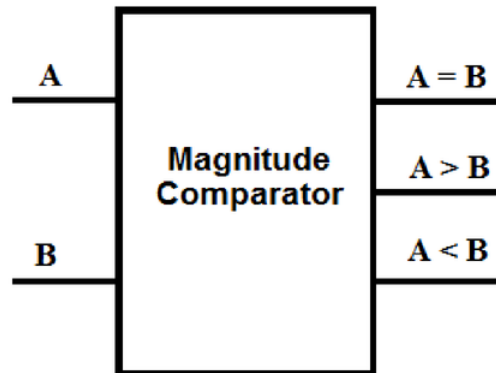


Figure 1 - Simple Magnitude Comparator (Poudel, n.d.)

While the example in figure 1 is simple, it can be better understood using a truth table to derive all possible outcomes. This can be seen in figure 2. The table shows 4 inputs for A and B, and the three possible outcomes each pair of values could have, with A being the generated value and B being the comparator set value. The first pair of 8 bit binary values both have A and B equalling to 12, so the output would be A=B. The second pair of values has A = 10 and B = 17, therefore the output is A being less than B, or A<B. The third pair of values inputs A as 15 and B to 5, therefore the output is A being greater than B, or A>B. The third pair of values just demonstrates the equal to output once more.

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

Good detail

Figure 2 - Comparator Truth Table (Poudel, n.d.)

How does the device make the comparison?

1.3. Analog to Digital Conversion

Standard microcontrollers use different voltages to interpret binary inputs. For example a microcontroller will differentiate a binary value using an input signal of either 0v or 5v. So a binary value of 0 is shown by 0v, and binary 1 by 5v. However due this is not always the case. What if the microcontroller detects an input of 3v or 1.5v? Is that a binary value of 1 or of 0? Because of this issue it is common practise for microcontrollers to detect and measure varying values such as analog inputs. Most microcontrollers are able to analyse these analog signals using an ADC to make a programmable decision.

An ADC, or an analog to digital converter is a common part of a microcontroller that takes a varying voltage detected on an analog input pin, and converts it into a digital number. Most modern microcontrollers commit a set amount of 5/6 pins to analog to digital conversion. An example of this can be seen on the Arduino microcontroller in figure 3 below. This board commits 6 pins labelled A0 to A5 to take in a varying voltage.

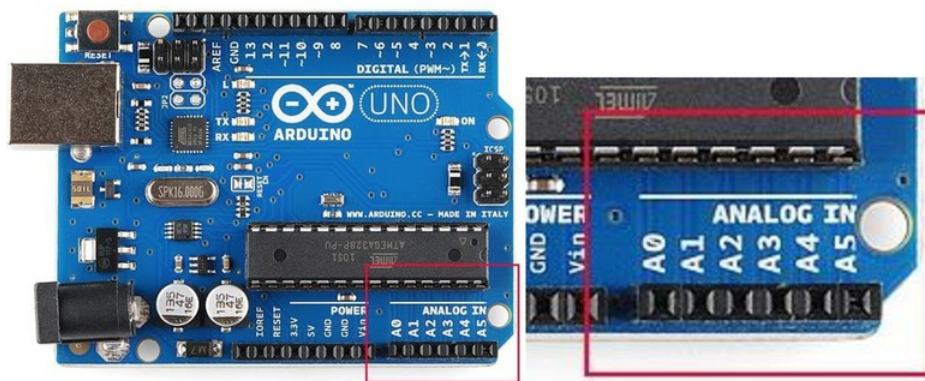


Figure 3 - Arduino Board (Nate, n.d.)

Most microcontrollers have the ability to detect 8-bit inputs. This would mean that a binary value up to 256 could be interpreted. The Arduino has a 10-bit ADC, so it can interpret values up to 1024. There are also ADC's available that can interpret full 16-bit binary values, so that is a binary value of up to 65,536 that can be read by that ADC. For the practical part of this assignment an 8-bit ADC would be used on the FPGA board. *Very good detail*

There are various methods in which an ADC works. The most common way is to take the input analog voltage and have it charge up an internal capacitor on the microcontroller. This capacitor is then discharged on an internal resistor and the time that it takes to discharge is measured in clock cycles. The number of clock cycles equals is what is used to determine the binary value from the analog input.

Interesting research shown...

1.4. Transducers and Load Cells

A transducer is a type of electrical component used in a variety of circuit set-ups. They are typically used for taking/measuring an energy/force, and converting it into another form. Some examples of transducers are megaphones, microphones, pressure plates/sensors, but for this assignment one that will be focussed on in particular is the load cell transducer.

A load cell is a type of transducer that takes a measurement or detects a force and converts that force into a variable in the form of an electrical output. They can also be referred to as strain gauge load cells, and that is their typical informal name across the load/weight measuring industry. They can be sectioned into groups of either hydraulic load cells, pneumatic load cells, strain-gauge load cells, piezoresistive load cells, Induction load cells, magnetostrictive load cells, and compression load cells. One load cell to look at in particular is a compression load cell known as the LCKD-1KG load cell.



Figure 4 - LCKD 1KG Compression Load Cell

This type of load cell uses its pressure plate to measure a weight and converts it into an electrical variable. This cell can measure loads from 0 to 1kg (1000g). It works on outputting a varying analog voltage between 0 and 5 volts.

*Not enough detail shown as to how the load
cell functions*

2. Design of Assignment Requirements; the Signal Conditioning Circuit

For this task an 8-bit magnitude comparator was designed and simulated using Multisim in order to determine a greater than, equal to or less than difference between two values. The first test performed was to see if the 8-bit magnitude comparator would compare the two values for A and B and recognise that they were both equal. The Binary values for A and B are set to equal values of 163. The circuit recognises this and displays the equal LED.

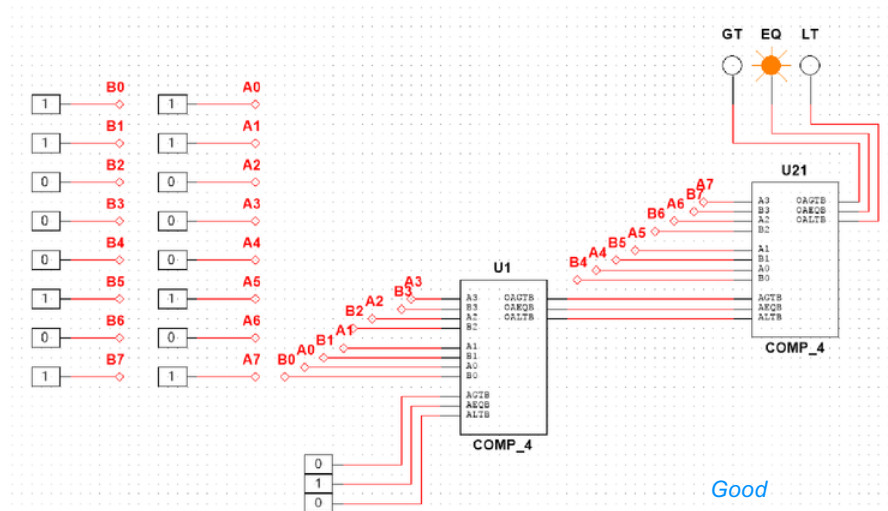


Figure 5 - Equal To

The second test was to see if the 8-bit magnitude comparator would recognise that the A value was less than the B value. A binary value of 162 was entered for value A, while a binary value of 163 remained for B. The comparator recognised this and displayed the less than LED.

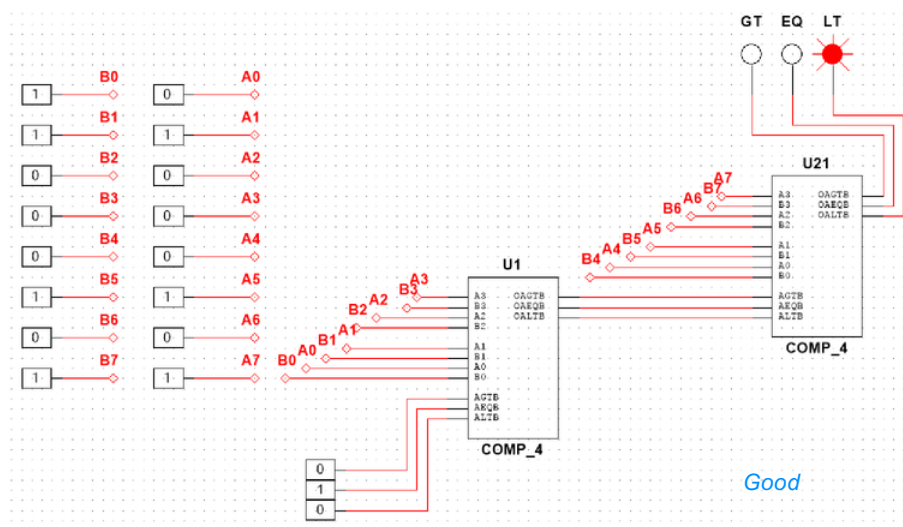


Figure 6 - Less Than

The third test performed was to see if the 8-bit magnitude comparator would recognise a value of A being greater than B. The A value was set to 163 while the B value was set to 162 for this test. The comparator recognised the difference of A being greater than B and in turn displayed the greater than LED.

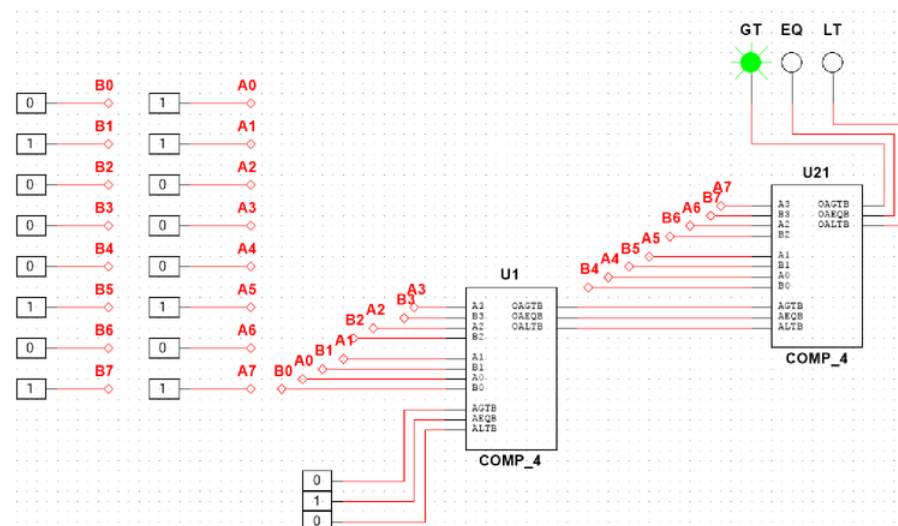


Figure 7 - Greater Than

Good understanding shown how to cascade 2X4 bit Magnitude comparators

3. Analysis of the Signal Conditioning Circuits

3.1. 8-bit Magnitude Comparator

The 8-bit magnitude comparator was then adapted so that it could be simulated on the Nexys4 FPGA board. The digital highs and lows as well as the LED's were swapped out for their PLD components counterparts which were switches (inputs and outputs) and LED's.

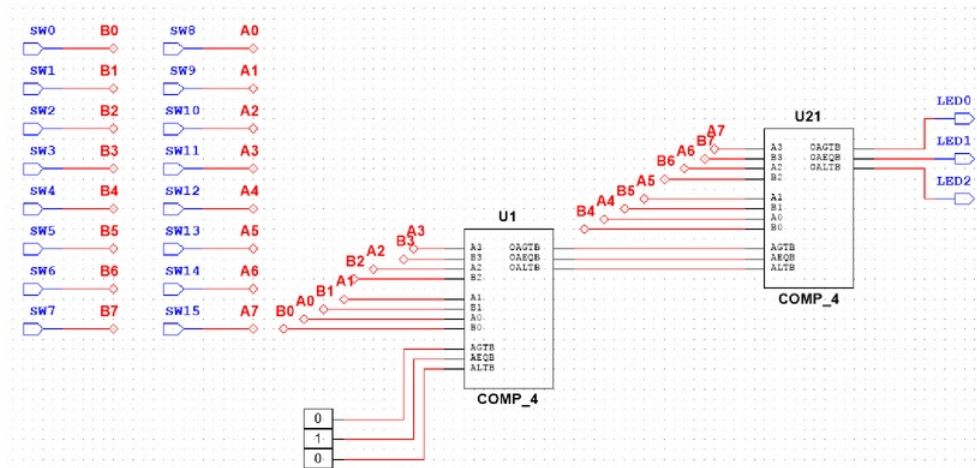


Figure 8 - Adapted Comparator

The first test was to see if the two values for A and B would be recognised by the FPGA board as equal. A binary value of 163 was entered using the switches. The switches read with switch 0 being the least significant and on the far right of the FPGA, and switch 15 being the most significant and on the far left. Switches 0-7 show the binary value for B, and switches 8-15 show the binary value for A. The photo below shows the two values entered on the switches, and the middle LED of LED1 lighting up displaying that the values are equal.

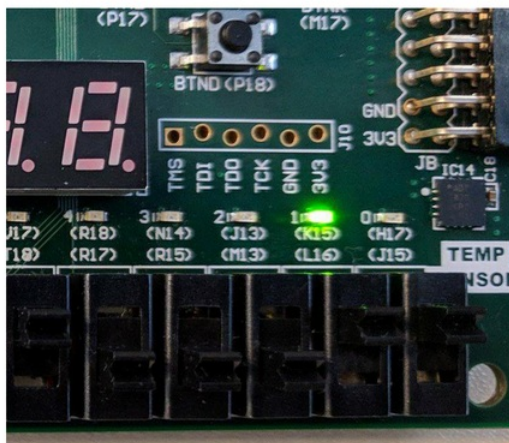


Figure 9 - LED 1 Equal

3.2. Instrumentation Amplifier

For this task a signal conditioning instrumentation amplifier was designed in Multisim to give an analogue output of 0 – 5V over the full load range when the excitation voltage (VEE) is 5V. The first screenshot in figure 13 demonstrates the circuit running at 50% on all 4 potentiometers. It gave an V(rms) reading of 19.9mv. The potentiometers were then varied with R1 and R4 increasing in increments of 5% to 100%, and R2 and R3 decrementing in steps of 5% to 0%. The results were recorded for every 5% change. An example of the potentiometers changing is shown in figure 12.

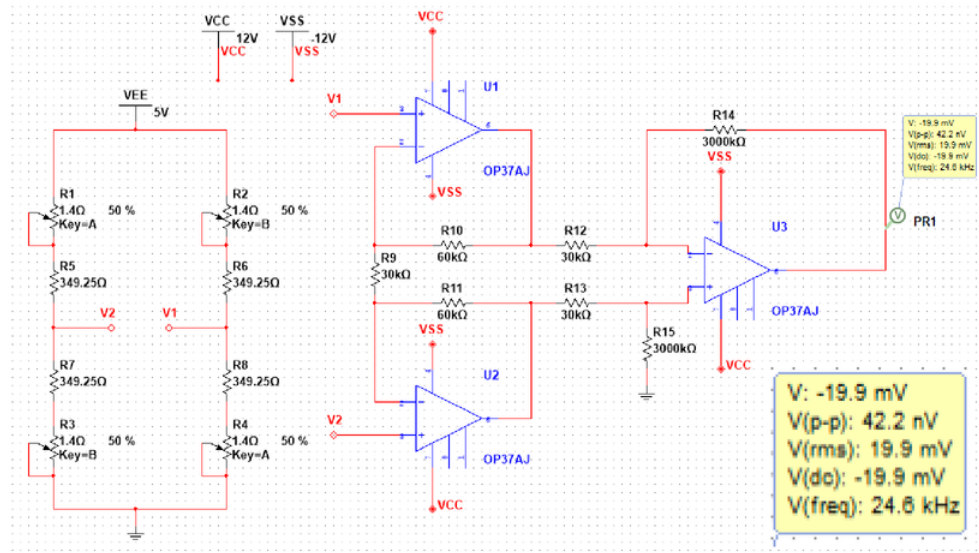


Figure 13 - Comparator 1

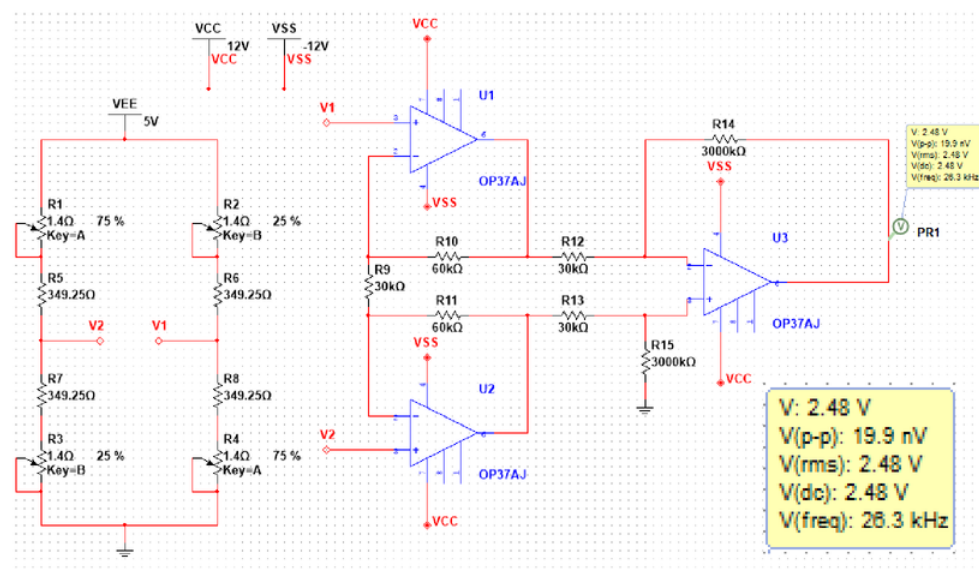


Figure 12 - Comparator 2

Good detail shown in relation to the schematic diagram but why choose the specific resistance values?
What is the gain of the instrumentation amplifier and why?

Figure 14 shows the table of results for incrementing the potentiometers in 5% intervals to 0% for R1 and R4, and to 100% for R2 and R3. This showed that as the weight in grams increased from 0g to 1000g or 1kg, the volt ranged steadily increased from 19.9mv at the start, effectively 0, to 4.98v by 1000g, essentially equalling 5v with rounding. This increase in voltage as load weight is increased is also shown visually in the graph in figure 15. This proves that using this type of comparator and performing a simulation like so can demonstrate how an external force can affect an electrical value. It demonstrates the process of taking a physical weight and transferring it into a range of between 0v and 5v. This could then be used to convert into a digital value of the weight for use in an electrical process situation.

R1/R4 (%)	R2/R3 (%)	Grms	V Out (rms)
50	50	0	19.9 mV
55	45	100	480 mV
60	40	200	980 mV
65	35	300	1.48 V
70	30	400	1.98 V
75	25	500	2.48 V
80	20	600	2.98 V
85	15	700	3.48 V
90	10	800	3.98 V
95	5	900	4.48 V
100	0	1000	4.98 V

Figure 14 - Comparator Results

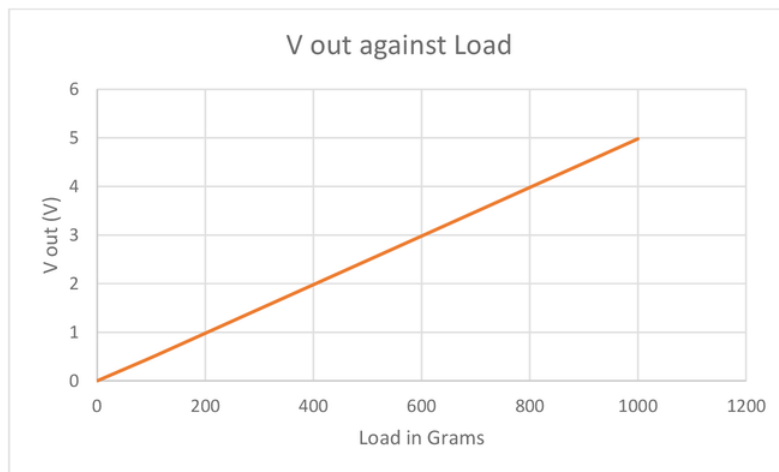


Figure 15 - Voltage out against Load

Good graphical analysis

4. Practical Build and Test via NI Elvis Board

4.1. Comparator to Labview .vi Simulation

The next part of this assignment involved taking the earlier generated 8-bit comparator from MultiSim, and building a .vi program to interface with once the MultiSim program has been uploaded to the FPGA. Once again this involved comparing two 8-bit digital signals and determining whether one is greater than, less than or equal to. The first step of this task was to take the 8-bit magnitude comparator from MultiSim and adjust its input/output pins to accommodate the .vi interface. This can be seen in figure 16 with the pins being named "JB".

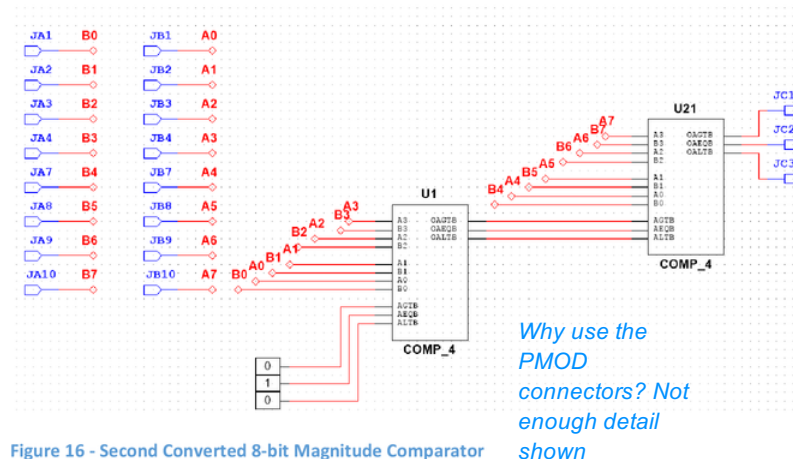


Figure 16 - Second Converted 8-bit Magnitude Comparator

Labview was used to create the .vi program. This build uses the DAQmx data acquisition tools earlier explored in the research section of this assignment. This build works by initially taking an analog voltage of between 0 and 5 volts and is read by the .vi program. From there the (if any) old reading is sent to the error out tool, while the current reading is detected by the analogue input tool and processed by the .vi's formula tool. This formula then takes the read voltage and calculates what the appropriate digital output should be for the process (divides by 5×256). In this case it would be the 8-bit binary conversion of a number. This is then sent to the digital right tool to output the digital value interpreted from the input voltage. This input value would be known as signal 1. This .vi can be seen in figure 17.

Figure 18 shows the accompanying interface in Labview. This is what will be used for simulation purposes in this task. The interface shows the analogue inputted value, the digital counterpart, and its 8-bit display. There is also a stop button.

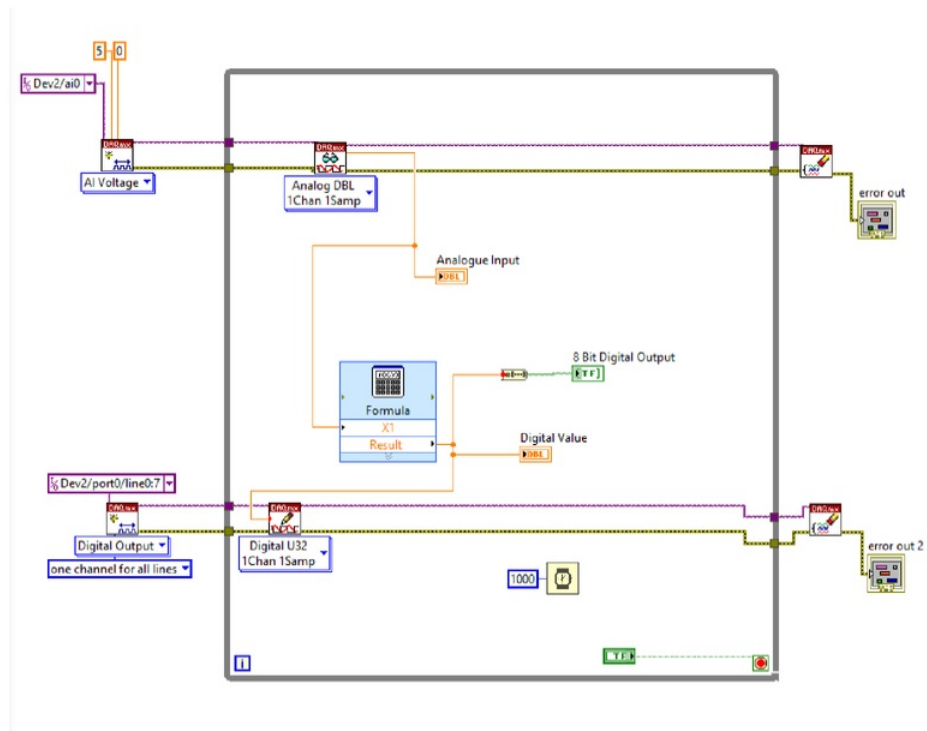


Figure 17 - .vi One

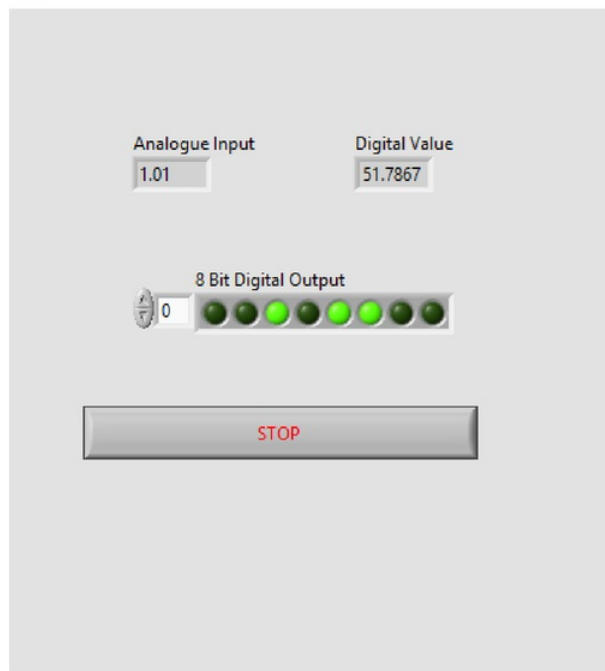


Figure 18 - .vi One Interface

Before interfacing with the 8-bit magnitude comparator could be done, the .vi needed to be tested for functionality. The first test involved using Labview instruments to simulate a 1v input into the .vi. An analogue input of 1.01v was inputted into the .vi and converted into a digital value of 51.8. The binary value of 52 was then shown on the 8-bit digital output LED's by illuminating the 3rd, 5th and 6th LED's, proving the system.

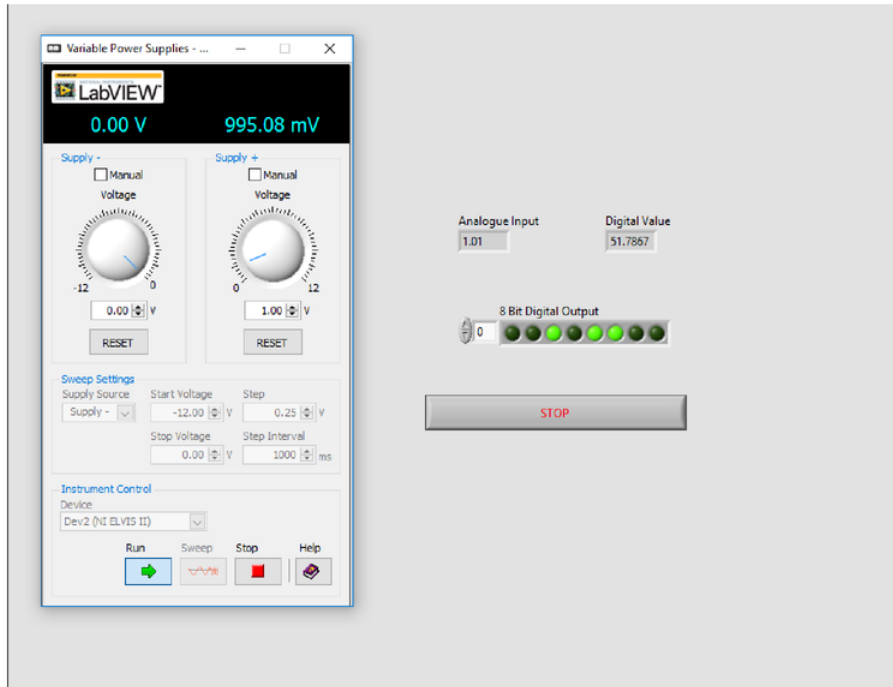


Figure 19 - Voltage to Binary Test 1

Why a value of 52? Show $(1.01/5) * 256 = 52$

The second test had a voltage of 2.5v inputted into the .vi. This converted to a digital value of 128.2, rounding closest to 128. This was then shown in binary on the LED's by illuminating the 8th bit of the binary LED display.

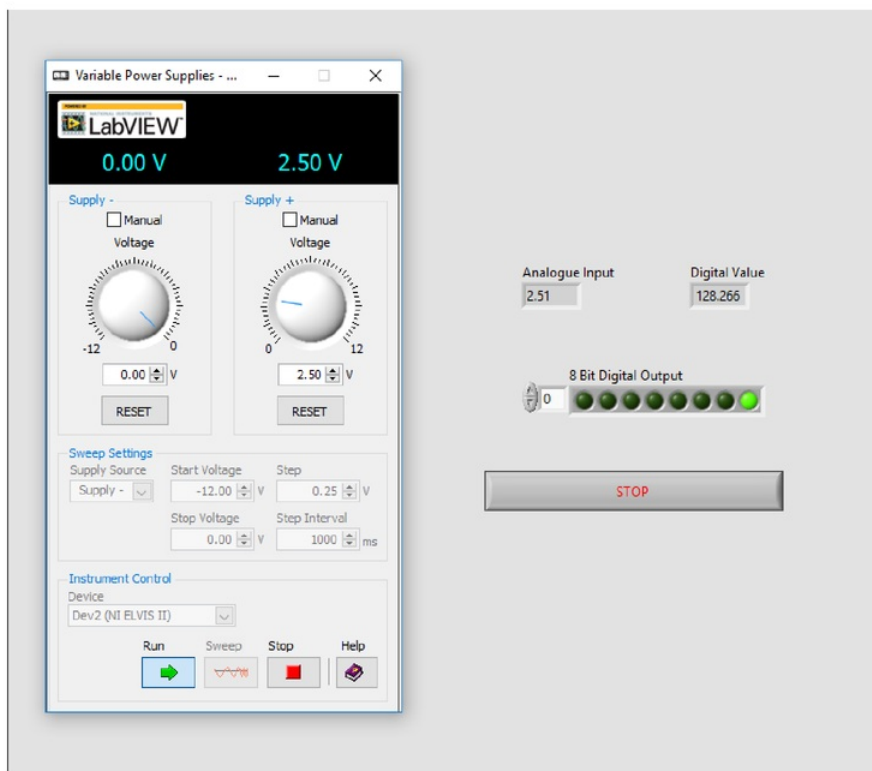


Figure 20 - Voltage to Binary Test 2

Again where is the mathematical analysis?

The second test had a voltage of 4.5v inputted into the .vi. This converted to a digital value of 230.2, rounding closest to 230. This was then shown in binary on the LED's by illuminating the 2nd, 3rd, 6th, 7th and 8th bits of the binary LED display.

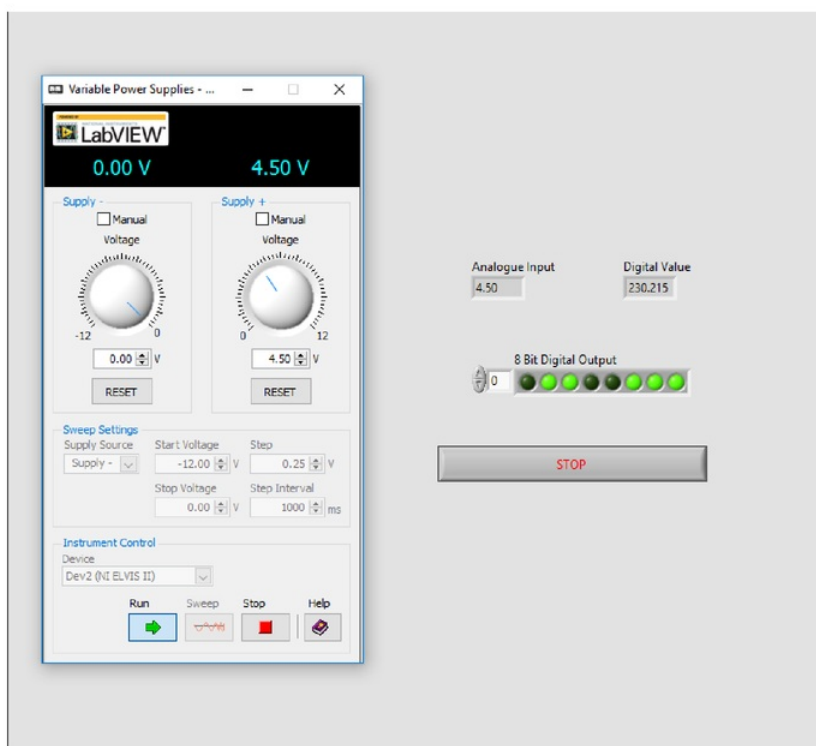


Figure 21 - Voltage to Binary Test 3

4.2. 8-Bit Comparator with Binary Conversion LabVIEW .vi

The next step in this assignment was to take the .vi and convert it so that it could be used with the 8-bit magnitude comparator in MultiSim. This .vi worked as an expanded version of the previous build to accommodate its new purpose. The new .vi can be seen in figure 22. This .vi works by first having an analog voltage between 0 and 5v being inputted and read as the load input. This analog value is then inputted into two formula tools. Formula uses the equation of $\frac{x1}{5 \times 256}$ and then placing it in the actual load output array. This is the digitised value ready for comparison. Formula 2 applies the equation of $\frac{x1}{5 \times 1000}$. This value is then placed in the load output (grams) tool. The first external digital output tool then takes the 8-bit converted value and inputs it into a digital write tool. Formula 3 applies the equation $\frac{x1}{1000 \times 256}$ and formula 4 uses the equation $x1 \times 256$. The bottom rail of the program takes the new value in grams and compares it with the set value using the read tool once more. This then gives an output of either greater than, less than or equal too depending on the comparison of the two values.

incorrect formula

incorrect formula

incorrect formula

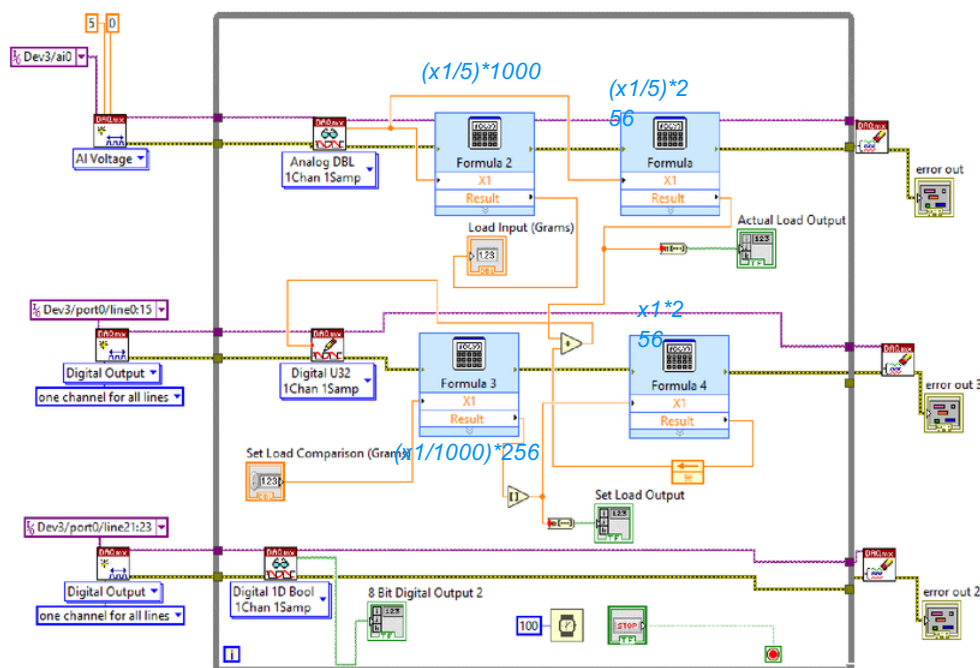


Figure 22 - .vi to FPGA Build

The .vi's interface can be seen in figure 23 and is what is used to display the two binary values and how they compare to one another. This interface shows each value in digital and binary form, and has three enlarged LED's to display either greater than, equal to or less than. It also has a stop button to stop the program. The 8-bit magnitude comparator from MultiSim was downloaded to the FPGA on the NI Elvis 11 board for this task. This circuit is what would take the set value and load value and compare them.

Incorrect formulae shown.... Why use the 'round to nearest' icon?

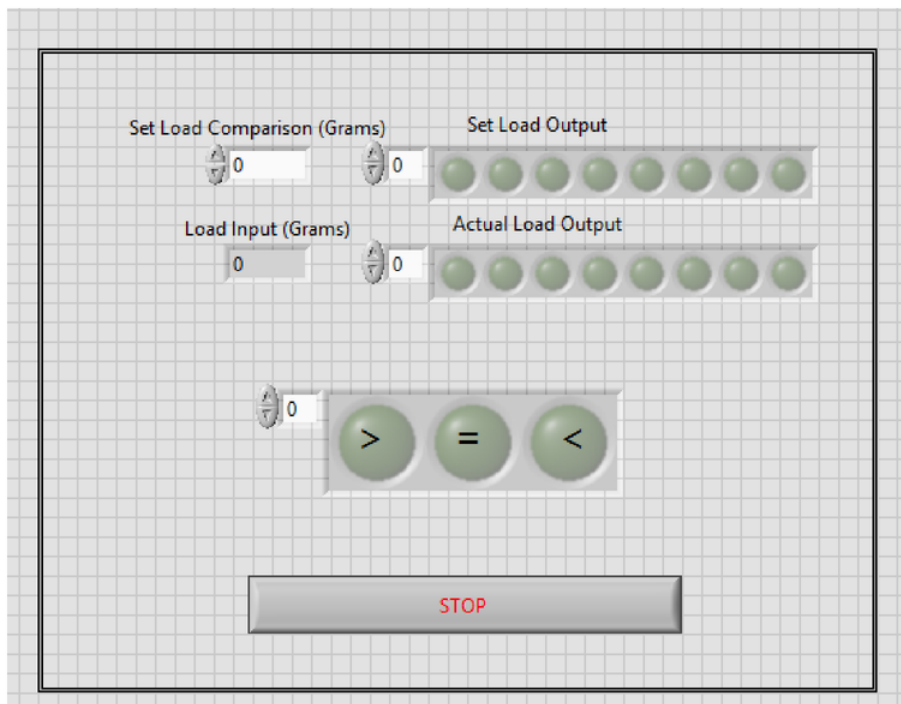


Figure 23 - Comparison .vi Interface

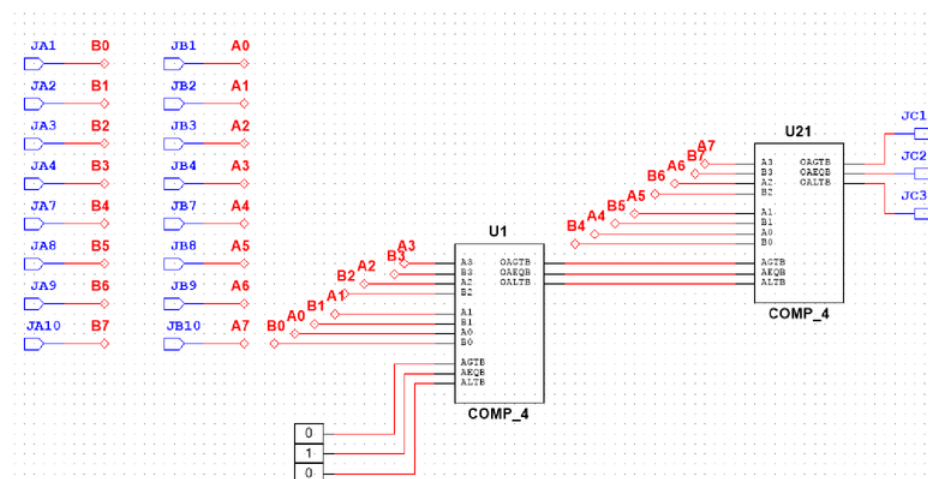


Figure 24 - 8-Bit Magnitude comparator Adapted

The first test once again used the Labview tools to simulate an analogue voltage input. The set value was entered as 250 grams and then the variable power supply tool was used to input a varied load value for comparison. The first test shows that the set value of 250 grams is greater than load value of 130 grams (0.64 V). This in turn showed the greater than LED illuminate.

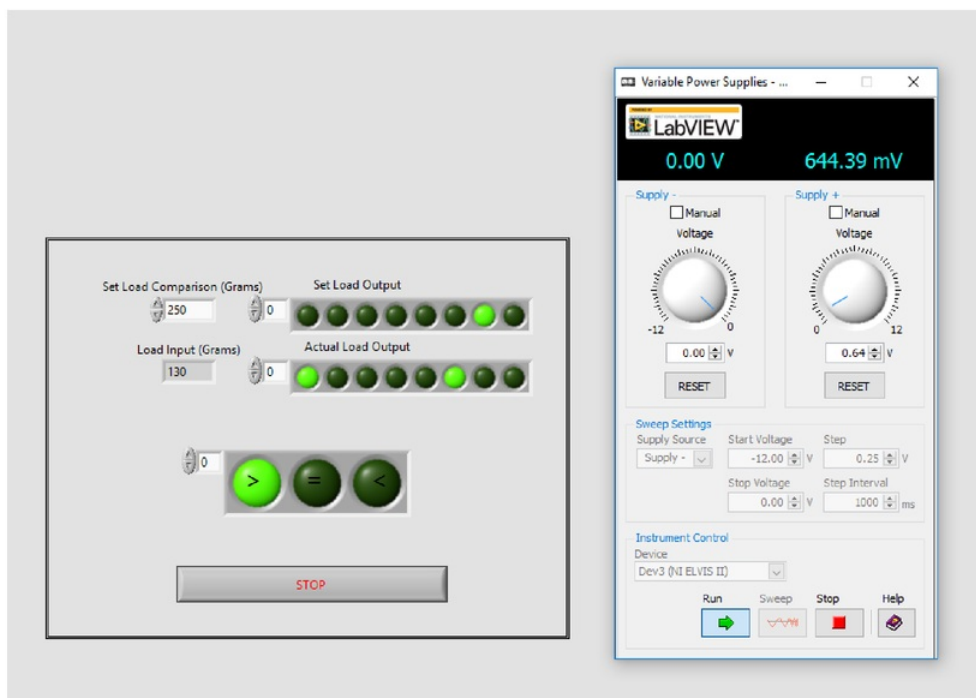


Figure 25 - Comparison Simulation 1

The second test shows that the set value of 250 grams is equal to load value of 250 grams (1.25 V). This in turn illuminated the equal to LED as the load value was equal to the set value.

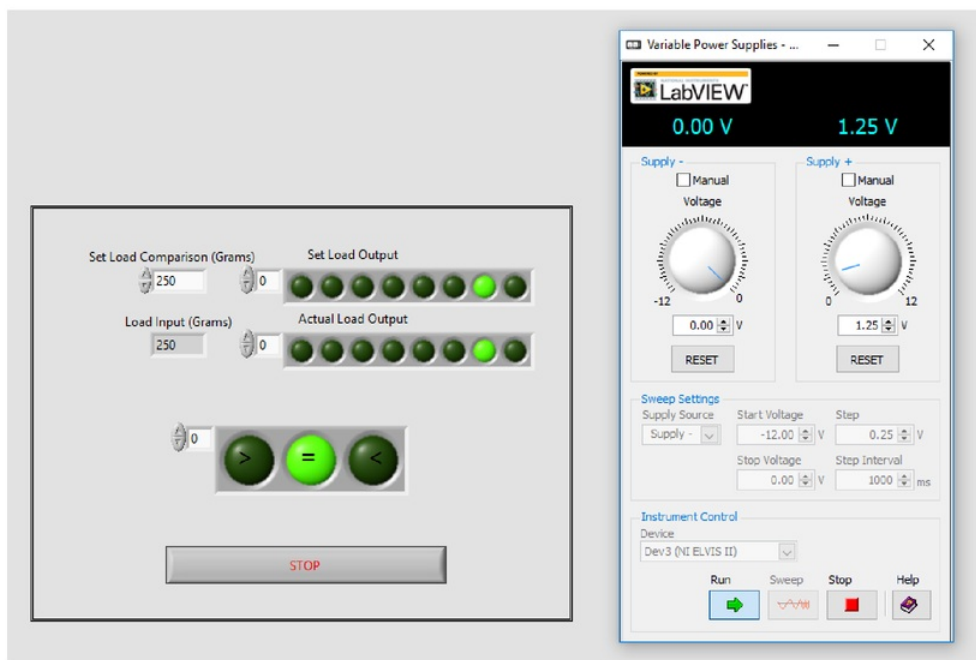


Figure 26 - Comparison Simulation 2

The third test shows the set value of 250 grams is less than load value of 811 grams (4.05 V). This in turn illuminated the less than LED, as the set value was less than the load value.

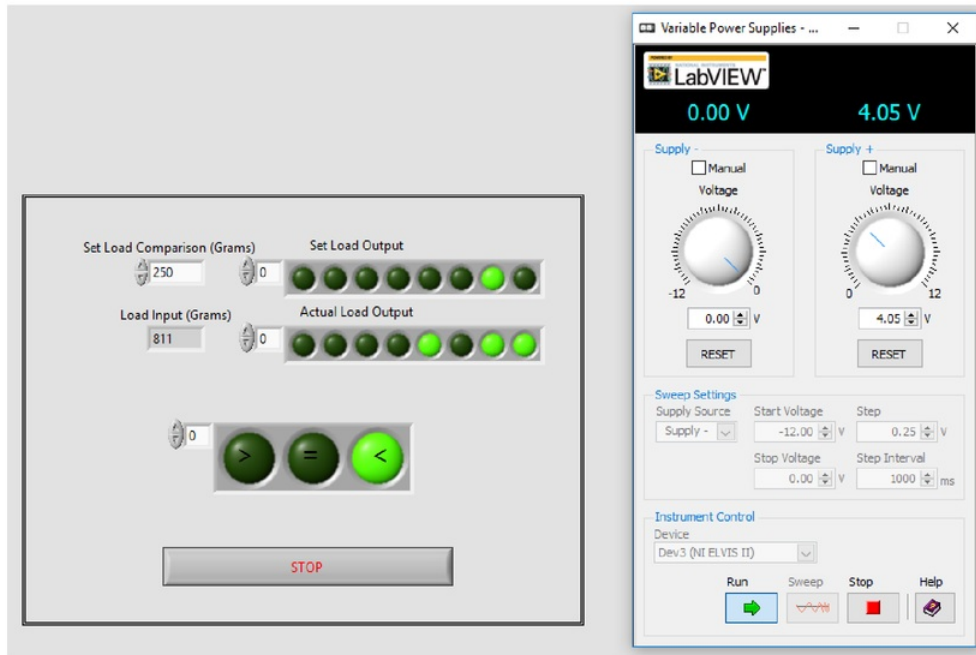


Figure 27 - Comparison Simulation 3

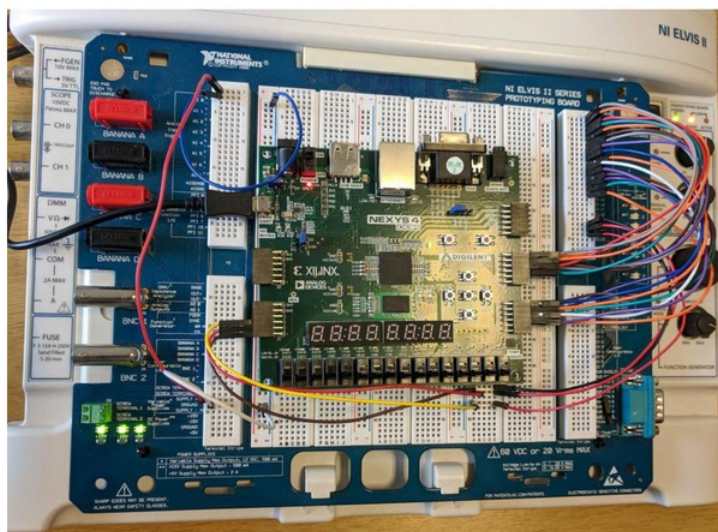


Figure 28 - FPGA connected to the NI Elvis II Board

Good set of practical results analysed correctly

Practical circuit shown but not analysed well

5. Adapt The Design to Fit New Purposes and Analysis of Excel Data

The final part of this assignment was to monitor the .vi over the load cells full measurable range of 0 to 5v. This is to see how load range has an effect over a period of time on the outputting values. For this test the .vi needed to be adjusted once again from the previous build. This .vi had write to measurement tools added onto the end of the first and last rail of the block design build. The first two write to measure tools took the digital value and wrote it as a grams value fluctuating between 0 and 1000g. The second three write to measure tools were used to compare whether the load value was less than, equal to or greater than the set value and slowly write each value over the range in this comparison. On the third rail each input value (less than, equal to or greater than) is also multiplied by 200. This is so it can be on a visible scale on the graph in an excel spread sheet. *Good*

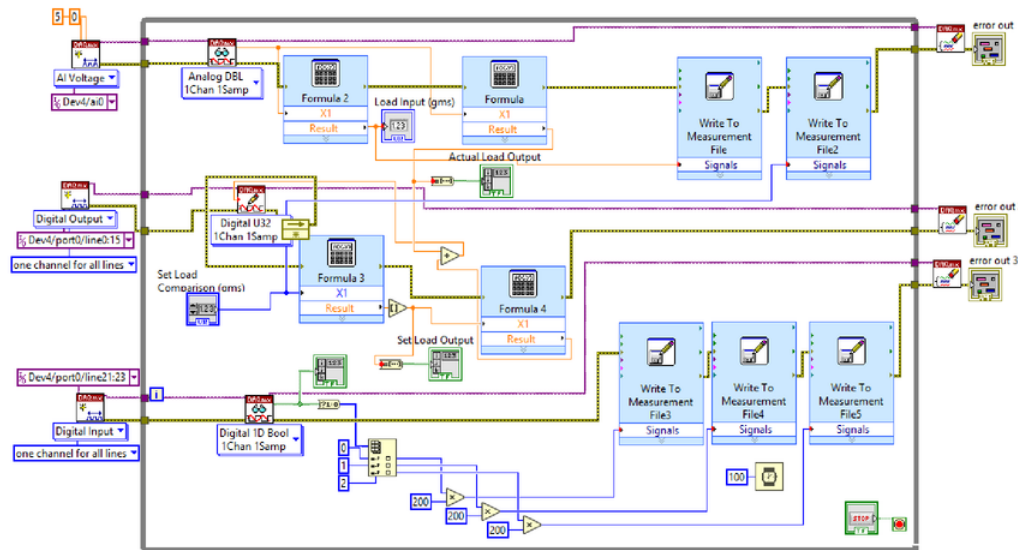


Figure 29 - Excel Comparison .vi

5 sets of data is being logged by this .vi over the full time period:

- Load input in grams
- Set Load Comparison in grams
- Load input Under set load (<)
- Load input Equal set load (=)
- Load input Over set load (>)

Why use the 'index array' for the digital inputs to the LabVIEW 'vi' from the FPGA?

The write to measurement tool is able to monitor all of this data over the 0 to 5v range and input it into excel in a tabular form, as well as displaying the data in the form of a graph.

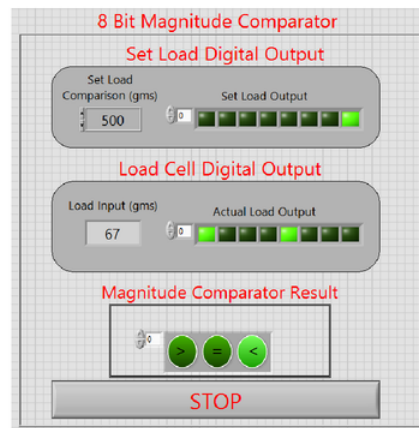


Figure 30 - 8-bit Magnitude Comparator .vi

The full table of excel data from the load cell can be found in the appendix of this assignment. Below in figure 31 shows the graphical table of all the data given by the load cell over the range of 0 to 5v.. The set load value can be seen as being set on the graph at 500 grams. Then the light blue load line depicts the analogue input, or load increasing overtime. The three plots along the bottom of the graph indicate the time periods for which the set load is more than, equal to, and less than the variable analogue load. Each of these plots would actually equal either 1 or 0 depending on when their criteria meets the current load to set load comparison, however because the .vi multiplied all the values by 200, it means that a time period of load 200 = 1, and a time period of load 0 = 0. The dark blue plot shows the set load being more than the abalgoue load and therefore having a logic = 1 up until about 12 seconds, then it switches to logic 0 and the equal criteria equals logic 1 until about 14 seconds. From here the less than logic is switched to 1 for the remainder of the 20 second time period.

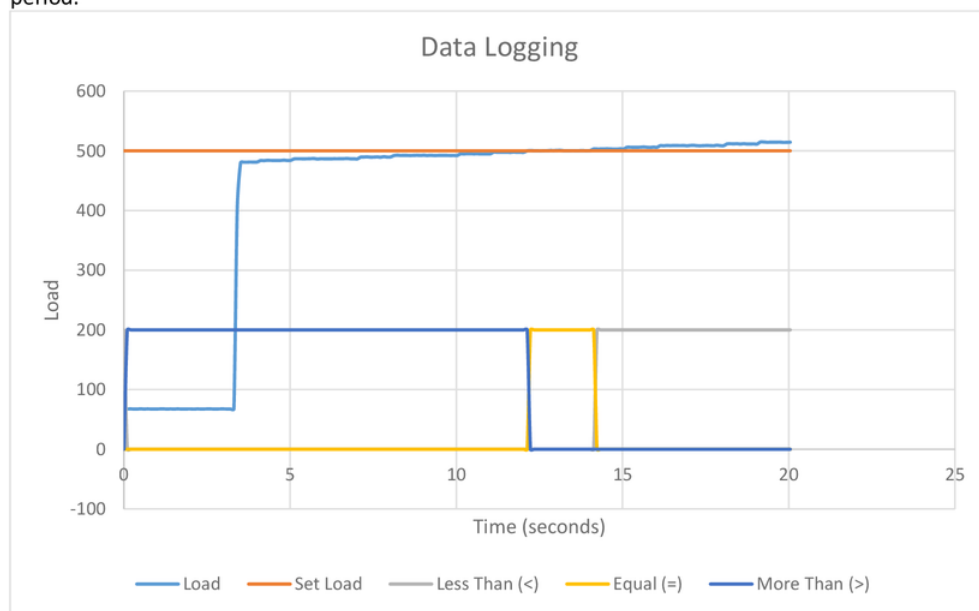


Figure 31 - Data Logging

6. Conclusions

The overall objective of this assignment was to investigate, through methods of research and carrying out practical simulations and experiments, an 8-bit magnitude comparator, NI DAQmx tools, analog to digital conversion, load cells, LabVIEW .vi's, and the relationship between them all. Each section was thoroughly researched with references to various sources covering a broad range of topics on related to the subject. These sources were both internet and book based. The first section of the research listed various DAQmx components from the LabVIEW system and broke down what their purpose was for in the circuit. This was important to understand as they would be used in the practical later on in the assignment.

The next section of the research delved into what a comparator was, specifically an 8-bit magnitude comparator. It was explained how the comparator takes a set value and a measured value, both of which being 8-binary values and comparing them in order to derive a comparison between the two. A truth table was also shown with examples of various scenarios that can occur within the comparators decision making process.

The next part of the research looked into what an ADC, or analog to digital converter was. It explained how analog to digital conversion works using a varying voltage between 0 and 5v, and how most modern day microcontrollers have ADC's installed on them.

The final part of the research looked at load cell transducers and how they operate in conjunction to comparators. Specifically the LCKD-1Kg and its operating specifications.

Once the research had been carried out, the practical experiments and simulations needed to be completed. The practical's were carried out as follows:

- An 8-bit magnitude comparator was built using Multisim and downloaded to a Nexys 4 DDR FPGA board. Three outputs of less than, equal too and greater than were produced on the board shown by LED's.
- A signal conditioning circuit was designed in Multisim to simulate a LCKD 1kg Load Cell. It was tested then over the full range of a 0 to 5v load cell to see the different measurements from resistors as the weight (in grams) increased on the load cell. This data was then produced on a graph
- The next step was building a LabVIEW .vi that interfaced with an FPGA via a NI ELVIS II board to produce a digital comparison of two 8-bit digital signals. The digital output from this system needed to be showing whether the measured value was less than, equal to, or greater than the set value. This worked as was expected and three sets of results were generated for it.

*Good, general overall conclusion given with
no reflection*

Works Cited

Nate, n.d. *Analog to Digital Conversion*. [Online]

Available at: <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>

[Accessed 15 04 2018].

Ohnhäuser, F., 2015. *Analog-Digital Converters for Industrial Applications Including an Introduction to Digital-Analog Converters*. illustrated ed. s.l.:Springer.

Pelgrom, M. J., 2010. *Analog-to-Digital Conversion*. illustrated ed. s.l.:Springer Science & Buisness Media.

Poudel, B., n.d. *Verilog for Beginners: 8-Bit Magnitutde Comparator*. [Online]

Available at: <https://esrd2014.blogspot.co.uk/p/8-bit-magnitude-comparator.html>

Texas Instruments, 1995. *SN54AS885, SN74AS885 8-BIT MAGNITUDE COMPARATORS*, Texas: Texas Instruments Incorporated.

Vahid, F., 2010. *Digital Design with RTL Design, Verilog and VHDL*. 2, illustrated ed. s.l.:John Wiley & Sons.

Appendix

Comparator load cell written to Excel

Time	Load	Set Load	Less Than (<)	Equal (=)	More Than (>)
0	67.284681	500	200	0	0
0.100266	67.4457	500	0	0	200
0.199924	67.606719	500	0	0	200
0.301181	67.349088	500	0	0	200
0.405635	67.413496	500	0	0	200
0.501966	67.70333	500	0	0	200
0.603161	67.381292	500	0	0	200
0.70295	67.574515	500	0	0	200
0.803218	67.477903	500	0	0	200
0.903486	67.477903	500	0	0	200
1.002779	67.4457	500	0	0	200
1.103018	67.606719	500	0	0	200
1.202709	67.606719	500	0	0	200
1.302976	67.349088	500	0	0	200
1.40271	67.574515	500	0	0	200
1.502984	67.316884	500	0	0	200
1.603242	67.606719	500	0	0	200
1.703844	67.4457	500	0	0	200
1.803015	67.381292	500	0	0	200
1.904285	67.477903	500	0	0	200
2.004548	67.510107	500	0	0	200
2.104818	67.284681	500	0	0	200
2.205081	67.574515	500	0	0	200
2.305414	67.4457	500	0	0	200
2.405024	67.4457	500	0	0	200
2.505288	67.606719	500	0	0	200
2.606558	67.606719	500	0	0	200
2.708307	67.381292	500	0	0	200
2.808483	67.413496	500	0	0	200
2.907747	67.574515	500	0	0	200
3.009016	67.574515	500	0	0	200
3.109282	67.413496	500	0	0	200
3.20955	67.349088	500	0	0	200
3.308863	67.606719	500	0	0	200
3.408866	402.818524	500	0	0	200
3.510169	479.787267	500	0	0	200
3.6094	480.850019	500	0	0	200
3.708662	480.946633	500	0	0	200
3.808935	481.172065	500	0	0	200
3.910507	480.978838	500	0	0	200
4.00977	481.300884	500	0	0	200
4.110454	483.651821	500	0	0	200
4.212708	483.587412	500	0	0	200
4.311974	484.038277	500	0	0	200
4.413243	483.909458	500	0	0	200

4.513512	483.845049	500	0	0	200
4.613776	483.877254	500	0	0	200
4.716049	483.877254	500	0	0	200
4.816315	484.295914	500	0	0	200
4.917583	483.812844	500	0	0	200
5.01785	484.070481	500	0	0	200
5.116656	486.389215	500	0	0	200
5.217927	486.711261	500	0	0	200
5.317244	486.679056	500	0	0	200
5.41697	486.582442	500	0	0	200
5.518268	486.711261	500	0	0	200
5.618506	486.872284	500	0	0	200
5.717772	487.065512	500	0	0	200
5.818048	486.292601	500	0	0	200
5.918096	486.679056	500	0	0	200
6.018496	486.614647	500	0	0	200
6.117192	486.614647	500	0	0	200
6.218463	486.485828	500	0	0	200
6.317726	486.743466	500	0	0	200
6.417752	486.679056	500	0	0	200
6.51819	486.872284	500	0	0	200
6.619283	486.872284	500	0	0	200
6.719549	487.097717	500	0	0	200
6.818815	486.646852	500	0	0	200
6.920115	486.646852	500	0	0	200
7.020351	486.453624	500	0	0	200
7.121173	489.352041	500	0	0	200
7.220437	489.609678	500	0	0	200
7.319701	489.835111	500	0	0	200
7.419977	489.384246	500	0	0	200
7.52031	489.48086	500	0	0	200
7.621517	489.770702	500	0	0	200
7.719778	489.287632	500	0	0	200
7.820044	490.157157	500	0	0	200
7.921325	489.513064	500	0	0	200
8.021593	489.674088	500	0	0	200
8.121373	491.960618	500	0	0	200
8.22164	492.604711	500	0	0	200
8.321908	492.218255	500	0	0	200
8.422175	492.379278	500	0	0	200
8.522439	492.250459	500	0	0	200
8.621703	492.153845	500	0	0	200
8.72197	492.314869	500	0	0	200
8.822251	492.508097	500	0	0	200
8.923167	492.250459	500	0	0	200
9.023466	492.572506	500	0	0	200
9.122857	492.121641	500	0	0	200
9.22298	492.443687	500	0	0	200
9.323228	492.153845	500	0	0	200

9.423495	492.18605	500	0	0	200
9.522761	492.411483	500	0	0	200
9.62304	492.347073	500	0	0	200
9.723297	492.250459	500	0	0	200
9.824565	492.153845	500	0	0	200
9.923926	492.18605	500	0	0	200
10.024093	492.121641	500	0	0	200
10.124051	494.794627	500	0	0	200
10.225324	494.826832	500	0	0	200
10.324708	495.052264	500	0	0	200
10.425215	495.084469	500	0	0	200
10.52545	494.923446	500	0	0	200
10.624715	495.02006	500	0	0	200
10.725985	494.826832	500	0	0	200
10.827282	494.762422	500	0	0	200
10.926522	494.923446	500	0	0	200
11.025781	494.987855	500	0	0	200
11.126199	497.821866	500	0	0	200
11.227604	497.596433	500	0	0	200
11.327734	497.499819	500	0	0	200
11.428629	497.499819	500	0	0	200
11.527893	497.564228	500	0	0	200
11.62816	497.43541	500	0	0	200
11.729463	497.85407	500	0	0	200
11.830697	497.499819	500	0	0	200
11.92976	497.564228	500	0	0	200
12.028998	497.564228	500	0	0	200
12.131269	499.947374	500	0	0	200
12.230533	500.527058	500	0	200	0
12.331802	500.172807	500	0	200	0
12.429783	500.108397	500	0	200	0
12.530048	499.979579	500	0	200	0
12.631322	500.205011	500	0	200	0
12.731743	500.269421	500	0	200	0
12.83188	500.591468	500	0	200	0
12.932948	500.462649	500	0	200	0
13.03117	500.720286	500	0	200	0
13.131989	500.849105	500	0	200	0
13.231255	500.237216	500	0	200	0
13.332555	500.172807	500	0	200	0
13.432796	500.205011	500	0	200	0
13.532054	500.237216	500	0	200	0
13.633322	500.043988	500	0	200	0
13.732736	500.366035	500	0	200	0
13.831849	500.33383	500	0	200	0
13.932117	500.205011	500	0	200	0
14.032383	500.591468	500	0	200	0
14.132174	503.167842	500	0	200	0
14.232399	503.071228	500	200	0	0

14.332551	503.071228	500	200	0	0
14.433296	503.200047	500	200	0	0
14.533535	503.36107	500	200	0	0
14.632736	503.200047	500	200	0	0
14.733979	502.910205	500	200	0	0
14.835247	502.878	500	200	0	0
14.936516	503.42548	500	200	0	0
15.035783	503.167842	500	200	0	0
15.136046	506.001855	500	200	0	0
15.236316	506.03406	500	200	0	0
15.338592	505.937446	500	200	0	0
15.436872	505.905241	500	200	0	0
15.536141	506.098469	500	200	0	0
15.638379	506.420516	500	200	0	0
15.737643	506.001855	500	200	0	0
15.83792	505.744217	500	200	0	0
15.937846	506.195083	500	200	0	0
16.039112	505.937446	500	200	0	0
16.138099	508.674845	500	200	0	0
16.238379	508.513822	500	200	0	0
16.339671	508.964688	500	200	0	0
16.437016	508.739255	500	200	0	0
16.539264	508.964688	500	200	0	0
16.63953	508.932483	500	200	0	0
16.737802	508.803664	500	200	0	0
16.839062	508.932483	500	200	0	0
16.94024	508.481617	500	200	0	0
17.041539	509.029097	500	200	0	0
17.145456	509.19012	500	200	0	0
17.241712	508.64264	500	200	0	0
17.340991	508.481617	500	200	0	0
17.440242	508.70705	500	200	0	0
17.542526	508.932483	500	200	0	0
17.640775	508.739255	500	200	0	0
17.741041	508.578231	500	200	0	0
17.841308	509.351144	500	200	0	0
17.943605	508.674845	500	200	0	0
18.042847	508.674845	500	200	0	0
18.144144	511.541064	500	200	0	0
18.245416	511.863112	500	200	0	0
18.344623	511.702088	500	200	0	0
18.444739	511.895316	500	200	0	0
18.546037	511.412246	500	200	0	0
18.646242	511.927521	500	200	0	0
18.746515	511.44445	500	200	0	0
18.846807	511.412246	500	200	0	0
18.946322	511.573269	500	200	0	0
19.046591	511.315631	500	200	0	0
19.146072	514.697127	500	200	0	0

19.246481	514.793741	500	200	0	0
19.34761	514.31067	500	200	0	0
19.449646	514.439489	500	200	0	0
19.545914	514.471694	500	200	0	0
19.647237	514.536103	500	200	0	0
19.745988	514.536103	500	200	0	0
19.845251	514.085237	500	200	0	0
19.946239	514.246261	500	200	0	0
20.046408	514.471694	500	200	0	0

S Dedoncker PSDE 5029 FINAL Ass 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

55/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

Text Comment. Very good detailed research on DAQmx icons used in assignment

PAGE 7

Text Comment. Good detail

Text Comment. How does the device make the comparison?

PAGE 8

Text Comment. Very good detail

Text Comment. Interesting research shown...

PAGE 9

Text Comment. Not enough detail shown as to how the load cell functions

PAGE 10

Text Comment. Good

PAGE 11

Text Comment. Good

Text Comment. Good understanding shown how to cascade 2X4 bit Magnitude comparators

PAGE 12

PAGE 13

Text Comment. Good set of practical results from the FPGA board which have been analysed well

PAGE 14

Text Comment. Good detail shown in relation to the schematic diagram but why choose the specific resistance values?

Text Comment. What is the gain of the instrumentation amplifier and why?

PAGE 15

Text Comment. Good graphical analysis

PAGE 16

Text Comment. Why use the PMOD connectors? Not enough detail shown

PAGE 17

PAGE 18

Text Comment. Why a value of 52? Show $(1.01/5) * 256 = 52$

PAGE 19

Text Comment. Again where is the mathematical analysis?

PAGE 20

PAGE 21

Text Comment. incorrect formula

Text Comment. incorrect formula

Text Comment. incorrect formula

Text Comment. $(x1/5)*1000$

Text Comment. $(x1/5)*256$

Text Comment. $x1*256$

Text Comment. $(x1/1000)*256$

Text Comment. Incorrect formulae shown.... Why use the 'round to nearest' icon?

PAGE 22

PAGE 23

PAGE 24

PAGE 25

Text Comment. Good set of practical results analysed correctly

Text Comment. Practical circuit shown but not analysed well

PAGE 26

Text Comment. Good

Text Comment. Why use the 'index array' for the digital inputs to the LabVIEW 'vi' from the FPGA?

PAGE 27

PAGE 28

Text Comment. Good, general overall conclusion given with no reflection

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

RESEARCH-INFORMED LITERATURE

-  Analogue / Digital Converters: Good detail shown on types of ADC available. (3/5) Load Cells: Not enough detail shown as to how the load cell functions (2/5) Digital Magnitude Comparators: How does the device make the comparison? (2/5) NI DAQmx in LabVIEW: Very good detailed research on DAQmx icons used in assignment (4/5)

11

KNOWLEDGE AND UNDERSTANDING

-  Design of the signal conditioning circuit for LCKD-1Kg Load Cell: Good detail shown in relation to the schematic diagram but why choose the specific resistance values? (2/5) Design of LabVIEW 'vi': LabVIEW 'vi' shown in report but incorrect formula values shown. Why use the 'round to nearest' icon? ((4/10) Design of 4/8-bit Magnitude Comparator: Good understanding shown how to cascade 2X4 bit Magnitude comparators. Why use the PMOD connectors? Not enough detail shown (6/10) Re-design to suit new applications: Why use the 'index array' for the digital inputs to the LabVIEW 'vi' from the FPGA? (3/5)

15

ANALYSIS

-  Analysis of signal conditioning circuit: Correct graphical analysis shown over the full load range

17

(3/5) Analysis of MultiSim Results: Good set of simulated and FPGA results analysed correctly (4/5)
Analysis of NI LabVIEW / FPGA results: Good set of results obtained and analysed satisfactorily
(6/10) Analysis of data logging files in EXCEL: Good graphical results shown and analysed correctly (4/5)

PRACTICAL APPLICATION AND DEP.



Connection diagram shown between the NI ELVIS unit, NI LabVIEW and the FPGA board via the PMOD connectors but not analysed (4/10)

4

SKILLS FOR PROFESSIONAL PRAC.



Deployment of appropriate media: Good use made of pictorial evidence and screen dumps in design and analysis. (3/5) Report layout and readability: A bit confused between design and analysis on all sections. Works cited could have been more varied (2/5) Section 9: Appropriate conclusion: Good, general overall conclusion given with no reflection (3/5)

8