

Assignment_2

by Joshua Beer

FILE	12940_JOSHUA_BEER_ASSIGNMENT_2_157347_457207123.DOCX		
TIME SUBMITTED	26-APR-2018 09:12AM (UTC+0100)	WORD COUNT	6929
SUBMISSION ID	87505675	CHARACTER COUNT	35654



NI LabVIEW
Magnitude
Comparator

VIRTUAL INSTRUMENTATION & CONTROL SYSTEMS

Joshua Beer

PSDE5029

26/4/18

Contents

.....	1
1. Introduction.....	5
2. Research	7
2.1 NI DAQmx data acquisition icons used in LabVIEW	7
2.2 Digital Magnitude Comparators.....	13
2.3 Analogue to Digital Converters	14
2.4 Load Cells (namely LCKD-1Kg)	15
(Omega, 2003-2013).....	16
3. Design of Assignment Requirements.....	17
3.1 MultiSim Comparator Design	17
3.2 MultiSim Signal Conditioning Design	18
3.3 NI LabVIEW Design	20
4. Analysis of MultiSim, Signal Conditioning Circuit and NI LabVIEW 'vi' Simulated Results	23
4.1 MultiSim Simulation Testing Results	23
4.2 Signal Conditioning Circuit Testing Results	25
4.3 NI LabVIEW 'vi' Simulation Testing Results.....	27
5. Practical Build and Test Via NI ELVIS Board.....	30
6. Analysis of NI LabVIEW / FPGA Results	32
7. Adapt Design to Fit New Purposes or Applications.....	35
8. Analysis of Data Logging EXCEL Results	37
9. Conclusions	38
10. Bibliography.....	40
Bibliography	40
11. Appendices.....	41
Data Logging.....	48

Figure 1 - DAQmx Create Channel Analogue Input.....	7
Figure 2 - DAQmx AI Example.....	7
Figure 3 - DAQmx Read Analogue	8
Figure 4 - DAQmx Formula	8
Figure 5 - DAQmx Output	8
Figure 6 - DAQmx Clear Task	9
Figure 7 - DAQmx Error Out	9
Figure 8 - DAQmx Number to Boolean Array.....	9
Figure 9 - DAQmx Feedback Node.....	10
Figure 10 - DAQmx Create Channel Digital Output	10
Figure 11 - DAQmx DO Example	10
Figure 12 - DAQmx Write Digital	11
Figure 13 - DAQmx Create Channel Digital Input	11
Figure 14 - DAQmx Read Digital	11
Figure 15 - DAQmx Magnitude Comparator	12
Figure 16 - DAQmx Wait	12
Figure 17 - DAQmx While Loop.....	12
Figure 18 - Digital Magnitude Comparator	13
Figure 19 - Load Cell	15
Figure 20 - MultiSim Comparator Design	17
Figure 21 - MultiSim Signal Conditioning Design.....	18
Figure 22 - NI LabVIEW 'vi' Design	20
Figure 23 - MultiSim Simulation Test; Equal.....	23
Figure 24 - MultiSim Simulation Test; Greater.....	24
Figure 25 - MultiSim Simulation Test; Less	24
Figure 26 - Signal Conditioning Simulation Testing Results Table	25
Figure 27 - Signal Conditioning Simulation Testing Results Graph	26
Figure 28 - NI LabVIEW 'vi' Simulation Test; Less	27
Figure 29 - NI LabVIEW 'vi' Simulation Test; equal.....	28
Figure 30 - NI LabVIEW 'vi' Simulation Test; Greater.....	29
Figure 31 - Practical Testing Via NI FPGA Board MultiSim	30
Figure 32 - FPGA and ELVIS Board Practical Set Up	31
Figure 33 - Practical Testing Via NI FPGA Board MultiSim: Greater.....	32
Figure 34 - Practical Testing Via NI FPGA Board MultiSim: Equal.....	33

Figure 35 - Practical Testing Via NI FPGA Board MultiSim: Less	34
Figure 36 - Adapted LabVIEW 'vi' Design.....	35
Figure 37 - Write to Measurement File.....	36
Figure 38 - Data Logging Graph.....	37

1. Introduction

The aims and objectives of this assignment are as follows:

Design a LabVIEW 'vi' that will interface to an FPGA via an NI ELVIS 2 board to give an 8 bit digital comparison between two 8 bit signals. This will be done by designing a MultiSim circuit which will compare the two signals and output 3 more signals whether the comparison is less than, equal to or more than the set value.

The first signal taken in will be an analogue signal which will be 0-5 V in range. This will represent the load in grams (0-1000) which will then be converted into the first 8 bit signal. This signal will be output to the ELVIS board and used to compare signals.

The second signal will be the set value in grams with the same range of 0-1000 grams, this will also be converted to an 8 bit signal and output to the EVLIS board where they can be compared.

The MultiSim program will be simulated before practically testing using the board to make sure the program works as should. These results will be recorded and used for analysis later on in the assignment.

Once simulated the program can be downloaded and practically tested on the Nexys 4 DDR FPGA board which will compare the two signals which at this point will be set using the switches on the board and will light up an LED depending whether the signals are less than, equal to or greater than the set point. Each outcome should be tested and results recorded.

These results should be recorded for analysis later on in the assignment.

Once the MultiSim program works correctly, the LabVIEW design should be made to read in the analogue signal and convert to the 8 bit value and also allow a value for the set load to be made.

This should then compare both of these digital output values and convert them so that the two 8 bit values stay separate and don't effect one another.

The digital output comparison should then be used as inputs and read by the program so they can be compared and output a less than, equal to or greater than signal.

These results should then be recorded for analysis later on in the assignment.

The final aim of this assignment is to adapt the design already made on LabVIEW so that the actual load cell value in grams over its full range is output. This should be compared to the set value and output the correct 1 of 3 signals.

These values should all be data logged using data logging files and using EXCEL a graph should be produced to show how the value of grams compared to the set point will affect the results.

2. Research

2.1 NI DAQmx data acquisition icons used in LabVIEW

NI DAQmx is used to create the program in LabVIEW that will be used as the comparator. This will include specific icons which will show the user what is going on in the program. These will be explained below:

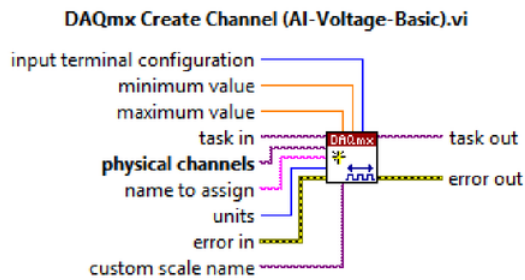


Figure 1 - DAQmx Create Channel Analogue Input

Figure 1 shows the create a channel icon which lets the user take an analogue input from an outside source and use this to output the signal into the program. The user can set a maximum and minimum value and the units if needed. It also allows for the physical channels in which the signal is coming from to be set, an example of this can be seen below in figure 2.

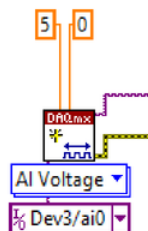


Figure 2 - DAQmx AI Example

Figure 2 shows the limits set between 0 and 5 volts and the units set to voltage with the analogue signal coming in from device 3, channel 0.

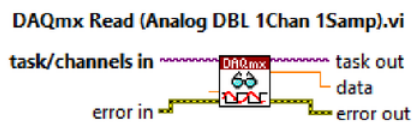


Figure 3 - DAQmx Read Analogue

Figure 3 shows the read an analogue signal icon. This icon is showing the user that an analogue signal from a specific channel is being read and the data from that channel is now output through the data part of the icon.

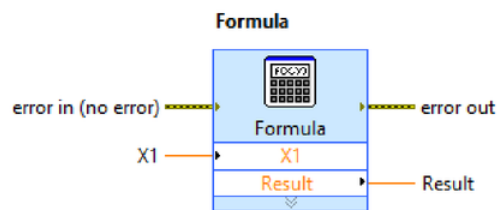


Figure 4 - DAQmx Formula

Figure 4 shows the formula icon in which the user can add a formula for any data input into X1 (as shown in figure). The formula will take place inside the icon and will then output the correct result which can be used in the program. The user must always put the input value to the formula as X1, for example $(X1/5)*256$.



Figure 5 - DAQmx Output

Figure 5 shows the output icon which will show the user the output value of any given point in the program, usually this is used after a formula to show the user the new output value calculated.

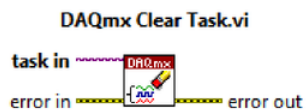


Figure 6 - DAQmx Clear Task

Figure 6 shows the clear task icon which shows the user that the task before hand has been cleared and releases and resources the task reserved first. The clear task uses the task in and error in which have started in the first icon used in the program and the error out.



Figure 7 - DAQmx Error Out

Figure 7 shows the error out icon which contains all the error information. The output provides standard error out functionality and this contains the error which has flowed through the program to this point.

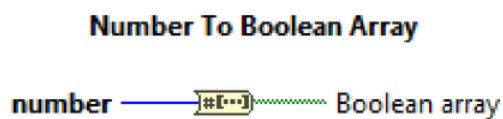


Figure 8 - DAQmx Number to Boolean Array

Figure 8 shows the icon which will convert a number into a Boolean array. This allows a numbered value to be converted into an array containing 8, 16, 32 or 64 bits. For example, if a value of 14.1 was entered into the number part, and 8 bit array would be made discarding the 0.1 and showing 00001110 as the array.

Feedback Node



Figure 9 - DAQmx Feedback Node

Figure 9 shows the feedback node icon. This icon shows the stored data from one loop and applies it to the next loop.

DAQmx Create Channel (DO-Digital Output).vi

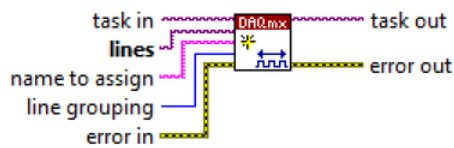


Figure 10 - DAQmx Create Channel Digital Output

Figure 10 shows the create channel for digital output icon. This icon shows the user that a digital output signal taken from the program is now used as either one digital channel or separate channels (lines as shown in figure). An example of this is shown below in figure 11.

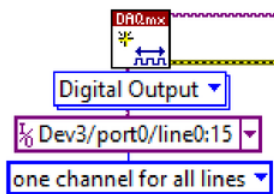


Figure 11 - DAQmx DO Example

Figure 11 shows the DO icon which takes its value from device 3, port 0 and separates the data into 16 different lines (0-15). This has one channel for all lines.

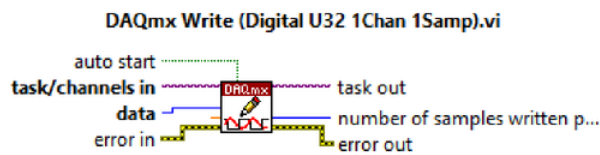


Figure 12 - DAQmx Write Digital

Figure 12 shows the write digital icon. It takes in the initial digital data from the channels and then uses the digital output data from the program, which has been calculated previous; and will write them into a single 32 bit unsigned integer sample for a single digital output channel.

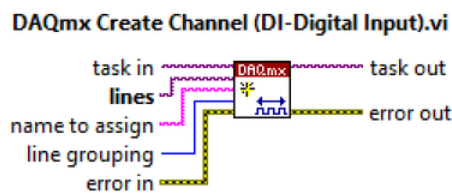


Figure 13 - DAQmx Create Channel Digital Input

Figure 13 shows the icon for creating a channel using a digital input. This shows the user that a channel to measure a digital signal is made. This is very similar to the digital output channels in figures 10 and 11.

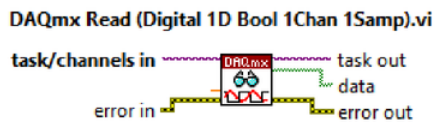


Figure 14 - DAQmx Read Digital

Figure 14 shows the read a digital signal icon. This icon is showing the user that a digital signal from a specific channel is being read and the data from that channel is now output through the data part of the icon.

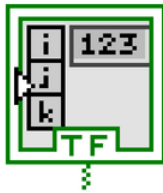


Figure 15 - DAQmx Magnitude Comparator

Figure 15 shows the icon for a magnitude comparator. This is the same icon as the output icon explained above as this is also an output but this differs in the fact that the data returns a 1D array of Boolean samples and each element of the array corresponds to a digital line in the channel. Therefore a comparison of each digital line can be made.

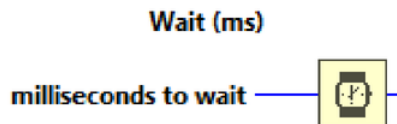


Figure 16 - DAQmx Wait

Figure 16 shows the wait icon. This shows the user that the program must wait the specific set time before returning a value into the program.

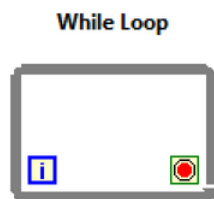


Figure 17 - DAQmx While Loop

Figure 17 shows the while loop icon. This repeats the code within its boundaries until a specific condition occurs.

Good overall understanding shown on the DAQmx icons used within the assignment

2.2 Digital Magnitude Comparators

Digital magnitude comparators use logic gates to determine whether or not the signal coming in is greater than, equal to or less than the signal it's being compared against.

(electronics-tutorials, 2018) States that "a *Magnitude Comparator* is a digital comparator which has three output terminals, one each for equality, $A = B$ greater than, $A > B$ and less than $A < B$ "

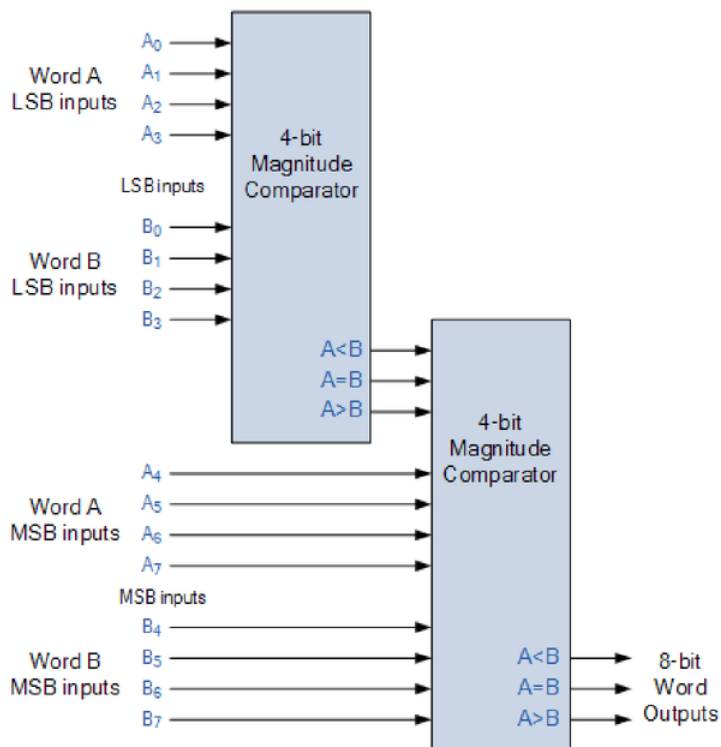


Figure 18 - Digital Magnitude Comparator (electronics-tutorials, 2018)

Figure 18 shows an 8 bit magnitude comparator and how it works using the cascade effect. This works by comparing the most significant bits (MSB) first to determine whether A is greater than, equal to or less than B. It does it in this order because the output would automatically be greater than or less than if A or B were different comparing the MSB first. *Good*

If the bits are equal it will cascade down bit by bit until it reaches the least significant bit (LSB), if these are also equal then the comparator will output the equal to output signal. *Good explanation given as to a 8 bit design*

An example of a digital magnitude comparators is the 74HC/HCT85 4-bit magnitude comparator. This is the same comparator as above but not used in cascade, it is useful in applications such as; process controllers and servo motor control.

(Phillips, 1990)

2.3 Analogue to Digital Converters

(microcontrollerslab, 2017) States "An analogue to digital converter (ADC) is an electronic device which converts varying analogue signals into digital signals so that they can easily be read by the digital devices. It has many applications in electronics projects."

So by taking an analogue signal which can vary between a set range, for example 4 - 20 mA or 0 – 5 Volts a digital value can be obtained from this. This allows the control system to be able to read the digital form and apply it.

Transducers are used to transform the analogue (physical) signal into a digital one, this is done by taking the value of the signal and dividing by the range of the signal and then multiplying it by the range of the digital value being used, for example an 8 bit digital value will multiply by 256 as this is how many different values you can have in 8 bits.

To increase the resolution of the analogue signal the digital range should be increased, so fewer samples per bit are used. This increases the accuracy of the digital value.

The most common types of analogue to digital converters available are:

1. Flash Analog to Digital converter.
2. Dual slope Analog to Digital converter.
3. Successive Approximation Analog to Digital Converter.

2.4 Load Cells (namely LCKD-1Kg)

Load cells are used to determine how much weight is pressing down on a given point in a circuit. These use change in resistance to calculate how much weight is being applied. The figure bellow's load cell has an unstressed resistance of 350 Ohms, therefore at 0 grams the load cell will be 350 Ohms.

Compression Load Cell
LCKD -1KG
Excitation: 5 Vdc, 7 Vdc maximum
Output: 2 mV/V nominal
Unstressed resistance 350Ω

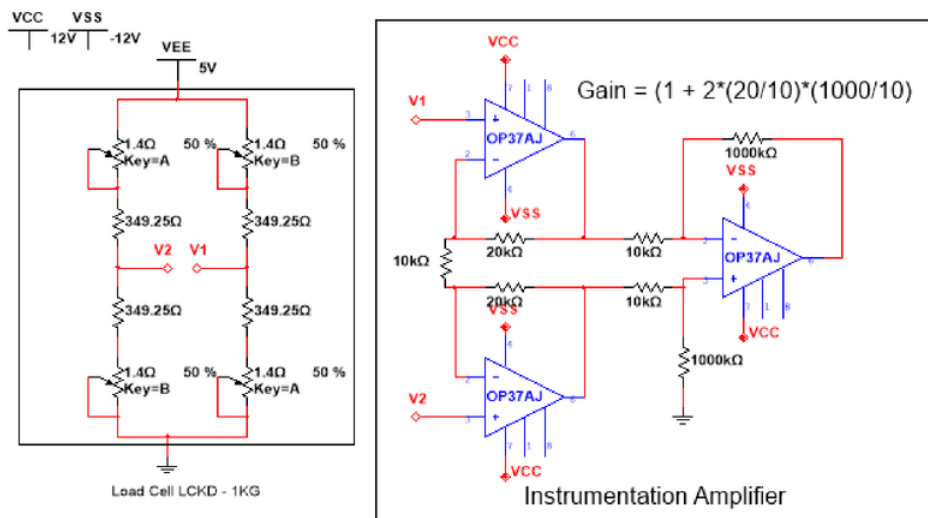


Figure 19 - Load Cell

Figure 19 shows a diagram of a load cell and how it works.

The load cells value is varied by changing the potentiometer values. By increasing the opposite resistance (top left and bottom right) and decreasing the other set of opposing resistors (top right and bottom left) the cell will be loaded and the grams increased.

The load cell will be equal at 50% and therefore 0 grams and fully loaded at 100% and 0% in opposite corners shown in the figure therefore being 1000 grams.

The gain required for the circuit in the assignment is 500, the following shows the maths behind this (using figure 19 values as an example):

Instrumentation Amplifier

$$\text{Buffer Gain} = (1 + (2 * 20K)/10K) = 5$$

$$\text{Diff Amp Gain} = 10000/10 = 100$$

Gain required:

$$\text{At 1Kg and 5 V excitation } V_2 - V_1 = 10\text{mV}$$

$$\text{At 1Kg } V_{\text{out}} = 5\text{V}$$

$$\begin{aligned} \text{Gain } G &= 5 / 0.01 \\ &= 500 \end{aligned}$$

(Omega, 2003-2013)

*Good explanation on how to signal condition a load cell but
not as to how they work!*

3. Design of Assignment Requirements

This part of the assignment will be showing the design requirements of the assignment and how these requirements are met with the designs being shown below.

3.1 MultiSim Comparator Design

The first part of the design requirements was to develop an 8 bit comparator. This was done on MultiSim.

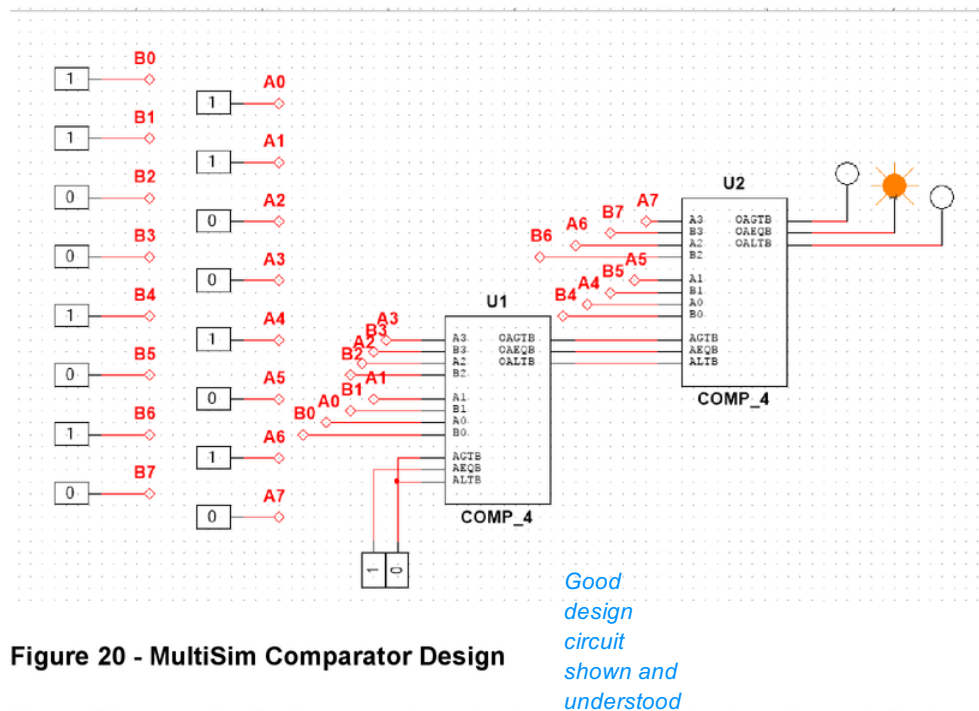


Figure 20 - MultiSim Comparator Design

Figure 20 shows the 8 bit comparator using two digital input values for each 8 bits. These can be seen on the less hand side of the figure and the inputs are labelled A0-A7 and B0-B7. These inputs feed into the main comparators as shown in the figure (U1 and U2).

The comparator will then output the final value whether A is greater than, less than or equal to the value set in B. This can be seen by the LED illuminating on the right hand side of the circuit.

This works by the program systematically checking to see if the A7 value is greater than, less than or equal to the B7 value, if the value is less than or greater than it will output the correct result on the comparator.

If the values of equal it will repeat this step until it finds a greater than or less than value further down the sequence. If still all values are equal at A0 and B0 and these values are also equal then the output generated will be the equal to one.

3.2 MultiSim Signal Conditioning Design

The next design stage was to make a signal conditioning unit, this was done on MultiSim and would give a linear voltage output for the range of 0-1000 grams.

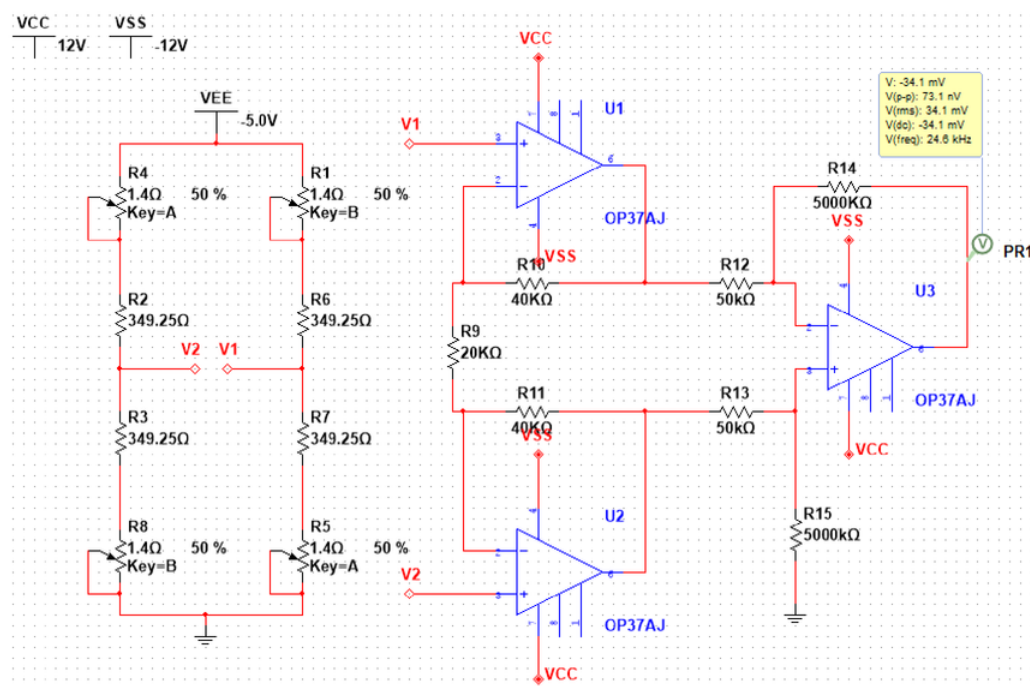


Figure 21 - MultiSim Signal Conditioning Design

Figure 21 above shows the design made for the signal conditioning. It was by having the load cell (left hand side of the figure) with a maximum voltage across it of 5V.

The load cells value is varied by changing the potentiometer values. By increasing R4 and R5 and decreasing R1 and R8 the cell will be loaded and the grams increased.

The load cell will be equal at 50% and therefore 0 grams and fully loaded at 100% and 0% in opposite corners shown in the figure therefore being 1000 grams.

This means the range for 0-1000 grams is 50-100%, so every 5% change is 100 grams.

The instrument amplifier is shown on the right hand side of the figure and will need to achieve a gain of 500. This is because at 1000 grams and 5 V supply $V_2 - V_1 = 10\text{mv}$, but to be able to get the whole 5 volts output a gain of 500 must be applied because $5/0.01 = 500$. *Good explanation given*

To check the gain of the amplifier the formula is as follows;

$$\text{Gain} = (1 + 2 \cdot (40/20) \cdot (5000/50)) = 500$$

$$\text{Buffer gain} = (1 + 2 \cdot (40/20))$$

$$\text{Diff amp gain} = (5000/50)$$

This is the buffer gain * the diff amp gain which is $5 \cdot 100$.

3.3 NI LabVIEW Design

The next part of the design stage was to use LabVIEW to create a design that will include the 8 bit comparator and compare a set load value (grams) against the actual load value (grams).

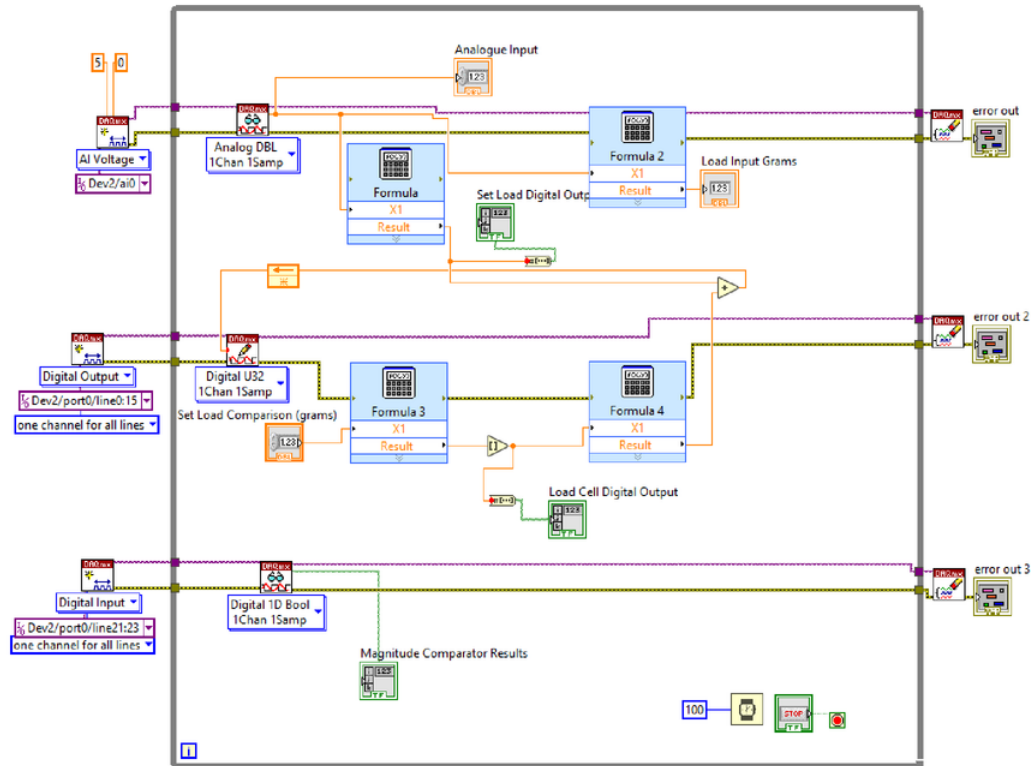


Figure 22 - NI LabVIEW 'vi' Design

Figure 22 shows the NI LabVIEW 'vi' design which will take in the analogue value and use this to compare against the set value and output a less than, equal to or greater than signal.

The icons used above have already been described in the research part of the assignment and therefore only the specifics of the design will be explained further.

The design is taking a 0 – 5V signal from the device off channel 0 and reading this value. It is taking this value and showing it as the analogue input value before being input into both formulas.

The first formula (shown in figure 20 as 'Formula') is $(X1/5)*256$, this is converting the analogue value into a digital value of 8 bits as if the X1 value is 5 it would be $5/5 = 1 * 256 = 256$ which would be full scale on an 8 bit digital scale. The second formula (shown in figure 20 as 'Formula 2') is $(X1/5)*1000$, this is converting the analogue signal into grams as if X1 was 2.5 (half scale) this would be $2.5/5 = 0.5 * 1000 = 500$ grams.

The output value from the grams is then taken and displayed as 'load input grams'.

This then ends the first line with the clear task.

The next line uses a digital output signal from channels 0-15 which is 16 digital outputs (this is because it is two 8 bit comparators). It then writes using the digital icon taking in the data from 'Formula' and 'Formula 4' which are added together.

The data into 'Formula 3' which is $(X1/1000)*256$ is taken from the set load comparison (grams) which is set by the user and is converting from grams to digital but only the first 8 bits of the comparator.

The data from 'Formula 3' must then be changed into a whole number as to not affect the second 8 bit values. It takes this whole value and uses it as the load cell digital output.

'Formula 4' which is $(X1)*256$ is taking the first 8 bit values and multiplying by 256 to create the second 8 bit comparator. For example is $256 * 256$ was used it would = 65536 which is the max value of 16 bits.

Again this then ends the second line with the clear task.

The third line starts with the digital input which is taking the inputs from channels 21-23 which will be the less than, equal to or greater than values. It then reads these values and the comparator will then output the correct value dependant on it being less than, equal to or greater than.

The design also has a wait 100 ms time where 100 ms must elapse before the value is put back into the program, this is so the load cell has a good sample rate.

The design also has a stop button allowing the user to stop the design at any time, and all of the design apart from the analogue/digital input and digital outputs along with the error outs and clear task are within the while loop.

Good, clear explanation of the design of the LabVIEW 'vi'

4. Analysis of MultiSim, Signal Conditioning Circuit and NI LabVIEW 'vi' Simulated Results

Using the designs above, simulations can be carried out to see if the designs worked correctly. This has been done and the results are shown below.

4.1 MultiSim Simulation Testing Results

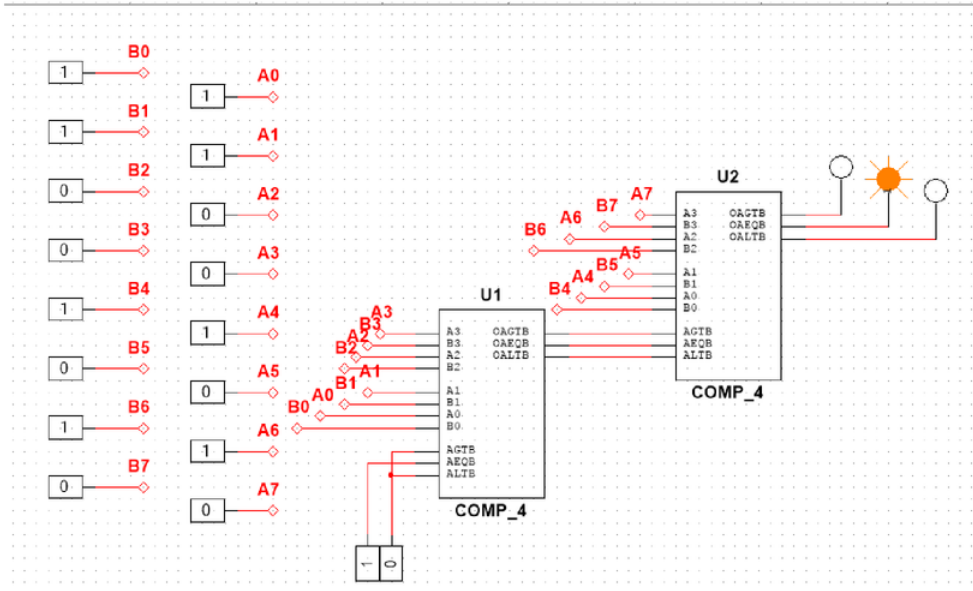


Figure 23 - MultiSim Simulation Test; Equal

Figure 23 shows the comparator simulation where the values of both 8 bit comparators are 83. This is because 01010011 is binary for 83 and as they are both equal values, the equal to LED is lit up.

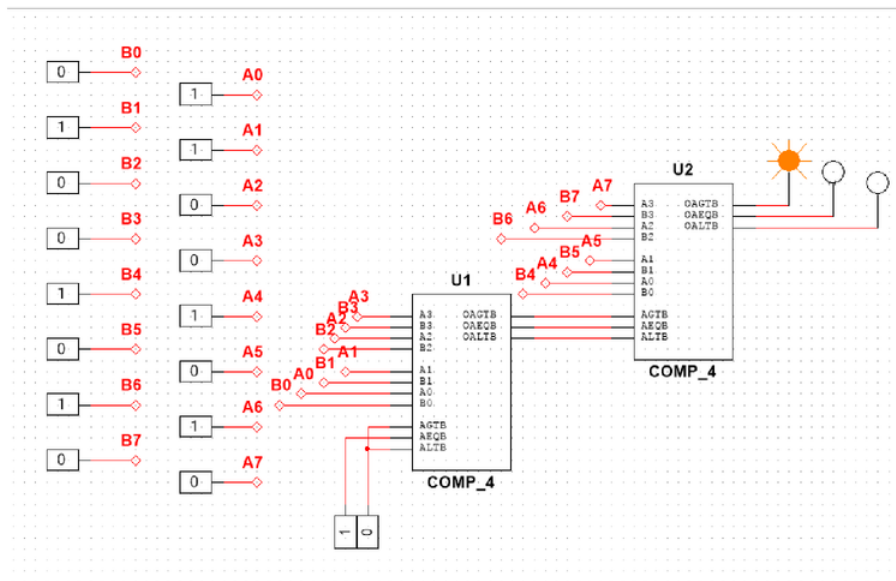


Figure 24 - MultiSim Simulation Test; Greater

Figure 24 shows that the values are now different as A value is 83 still but now B value is 82, this is because 01010010 in binary is 82. This means A is greater than B so therefore the greater than LED is lit up.

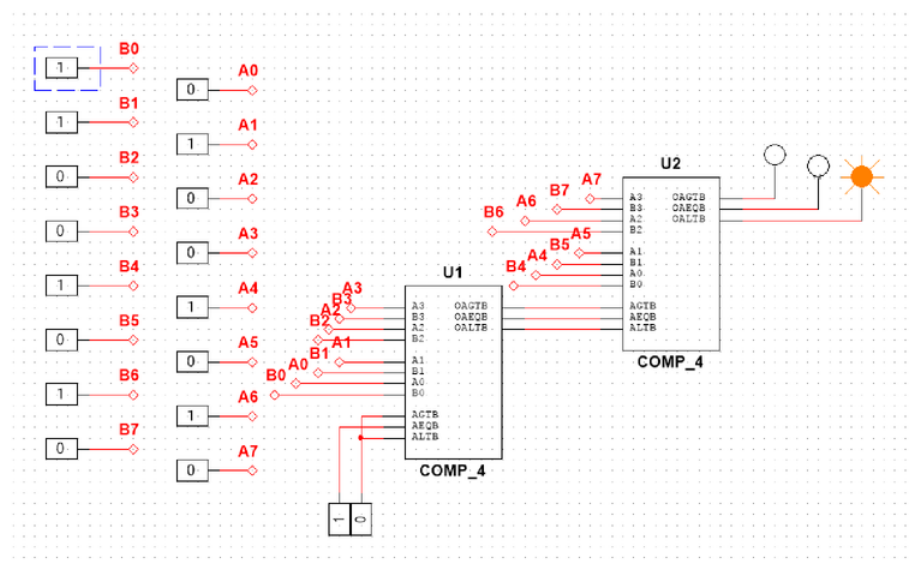


Figure 25 - MultiSim Simulation Test; Less

Figure 25 shows that the values are again different as A value is 82 but now B value is 83 again. This means A is less than B so therefore the less than LED is lit up.

This part of the design meets the specification set out in the aims and objectives and has been proved to work by carrying out simulation tests and the results have come back and shown to be correct.

Very clear understanding shown through the analysis of the Multisim results

4.2 Signal Conditioning Circuit Testing Results

Testing of the signal conditioning circuit can be seen in full in the appendix. This shows all the simulations from 50 up to 100%. The results of this testing are shown in the table below.

%	Grams	Voltage
50	0	0.0341
55	100	0.534
60	200	1.03
65	300	1.53
70	400	2.03
75	500	2.53
80	600	3.03
85	700	3.53
90	800	4.03
95	900	4.53
100	1000	5.03

Figure 26 - Signal Conditioning Simulation Testing Results Table

Figure 26 shows the results table for signal conditioning. This is increasing the load from 0 to 1000 grams in 100 gram steps by changing the potentiometer difference by 5% at a time. The table shows the voltage starts at 0.03 which is to be considered as zero as this error continues throughout the results and would be due to there being voltage leakage in the circuit and is amplified by the gain.

The results show a linear increase of 0.5V per 100 grams increase until at 1000 grams the voltage is full scale of 5 volts (as the range is 0-5 Volts). This graph can be seen in figure 27 below.

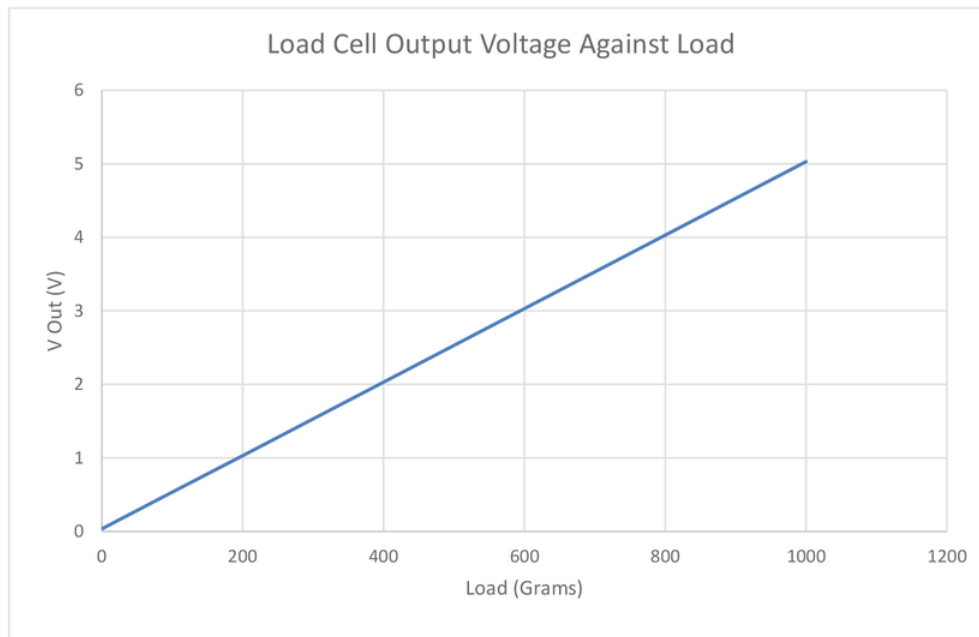


Figure 27 - Signal Conditioning Simulation Testing Results Graph

Figure 27 shows the linear increase in voltage output against the linear increase in load (grams).

This proves that the signal conditioning has worked correctly and the gain has made the scale of 0-5 volts correct and also made the scale linear according to the specification.

Correct graphical analysis shown and analysed correctly

4.3 NI LabVIEW 'vi' Simulation Testing Results

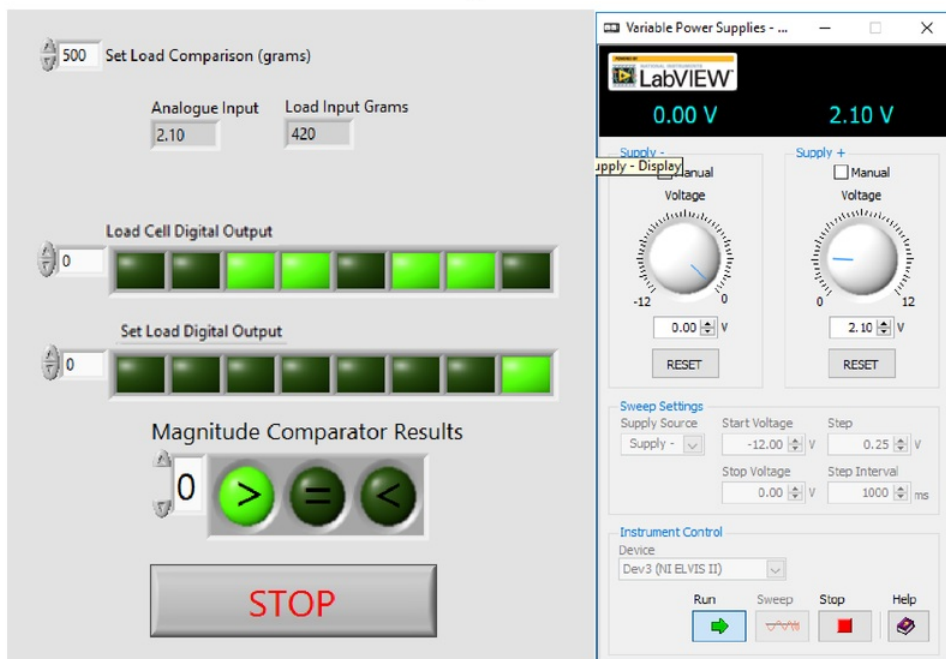


Figure 28 - NI LabVIEW 'vi' Simulation Test; Less

Figure 28 shows the test for the LabVIEW 'vi' where the load cell output is less than the set load output. It shows that the set load is 500 grams, as defined by the user and confirmed by the 8 bit comparator showing 10000000 which is 128 in binary and half of 256 which is 500 grams.

The load input is 420 grams and 2.1 Volts, every 0.1 volt is 20 grams therefore 2.1 volts is $21 \times 20 = 420$ grams. This is shown on the 8 bit comparator as 01101100 which is 108.

Therefore 420 grams is less than 500 grams, 2.1 volts is less than 2.5 volts and 108 is less than 128 which is what the comparator is using to give the less than output.

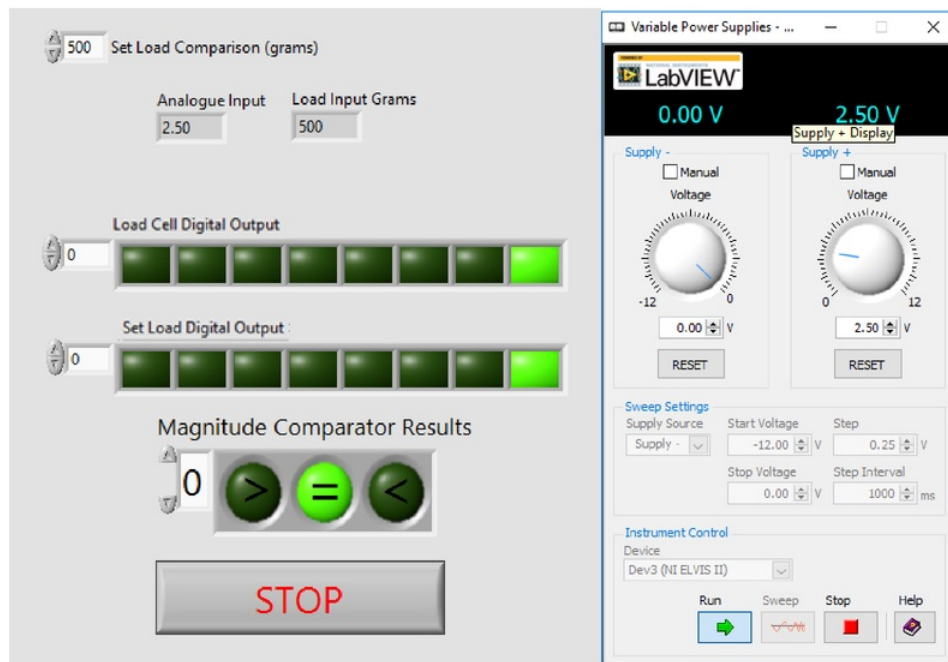


Figure 29 - NI LabVIEW 'vi' Simulation Test; equal

Figure 29 shows the test for the LabVIEW 'vi' where the load cell output is equal to the set load output. It shows that the set load is 500 grams still but now the load input is also 500 grams and 2.5 Volts This is shown by the 8 bit comparators both reading 10000000 which is 128 in binary.

Therefore because both the set load and the load cell output are the same weight and have the same digital output, the equal to is being outputted also to show the user they are equal.

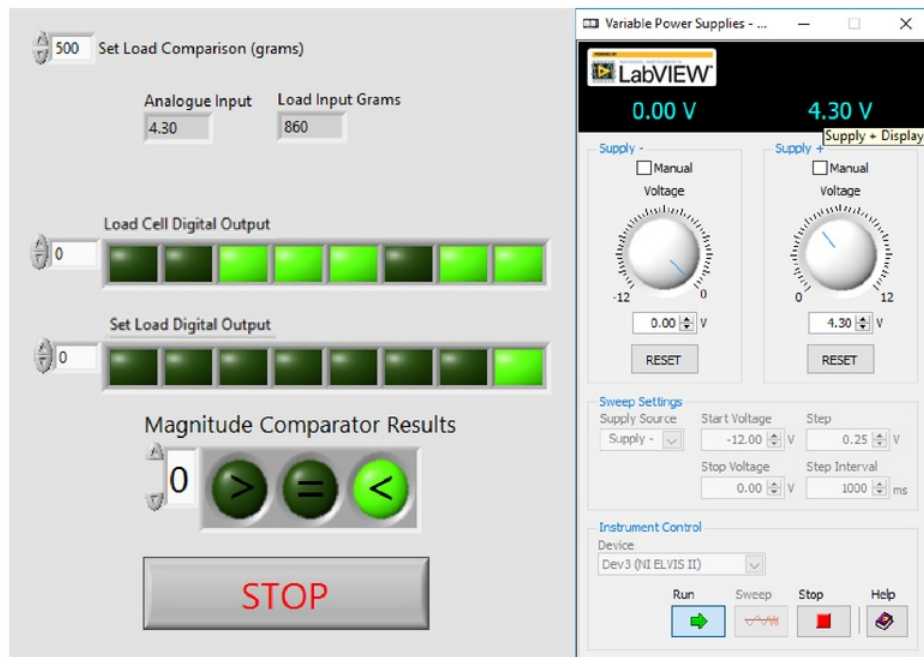


Figure 30 - NI LabVIEW 'vi' Simulation Test; Greater

Figure 30 shows the test for the LabVIEW 'vi' where the load cell output is greater than the set load output. It shows that the set load is 500 grams still, whereas the load input is now 860 grams and 4.3 Volts, every 0.1 volt is 20 grams as stated before 4.2 is double 2.1 as earlier calculated to be 420 grams so $* 2 = 840 + 20$ grams = 860 grams. This is shown on the 8 bit comparator as 11011100 which is 220.

Therefore 860 grams is greater than 500 grams, 4.3 volts is greater than 2.5 volts and 220 is greater than 128 which is what the comparator is using to give the greater than output.

This proves that the LabVIEW design is working along with the 8 bit comparator design which is inbuilt into it. Each test has been proved to give the correct output of less than, equal to or greater than.

Excellent analysis of LabVIEW results.

5. Practical Build and Test Via NI ELVIS Board

For this part of the assignment the designs implemented above will be tested practically via the NI ELVIS board. To be able to do this, the MultiSim circuit needs to be adjusted so as to run off the switches on the FPGA board and output to the LEDs of the board.

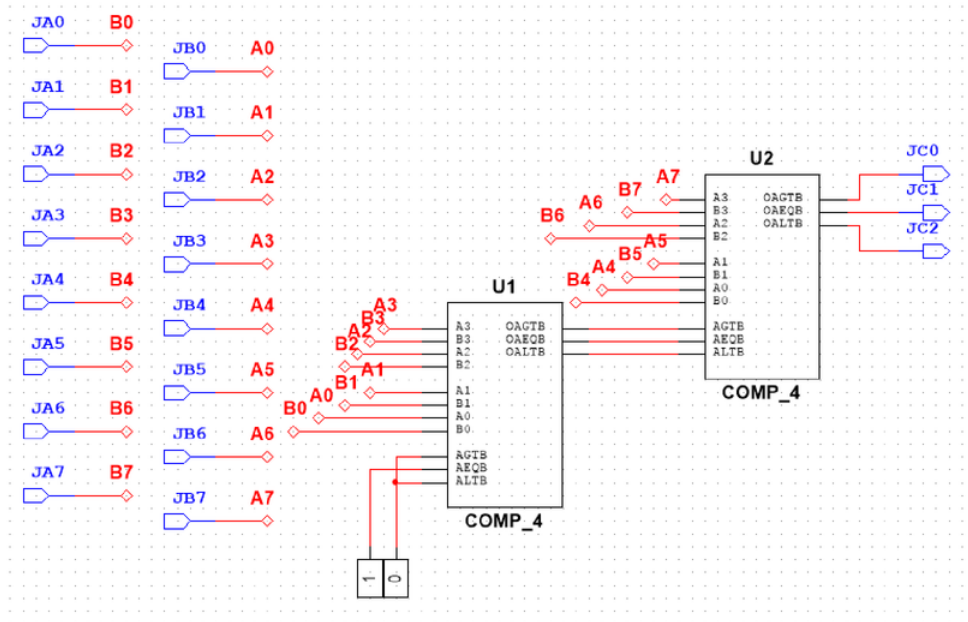


Figure 31 - Practical Testing Via NI FPGA Board MultiSim

Figure 31 shows the MultiSim circuit being converted to a PMOD circuit. This is done by replacing the inputs which could be set to on/off via MultiSim with physical switches which will be used by the board itself and changing the LEDs which would light up on MultiSim to physical LEDs.

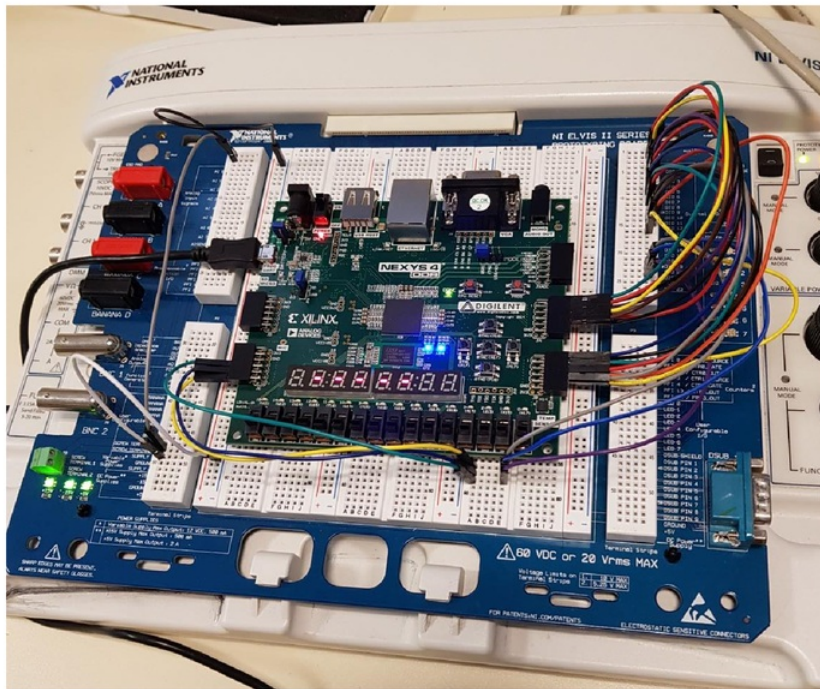


Figure 32 - FPGA and ELVIS Board Practical Set Up

Figure 32 shows the FPGA and ELVIS board set up with the two 8 bit comparators, analogue input which will take the set load and compare it with the actual load input in grams to output the greater than, equal to or less than values shown.

Not a totally clear explanation given

6. Analysis of NI LabVIEW / FPGA Results

Once the board has been built the results can be analysed below.

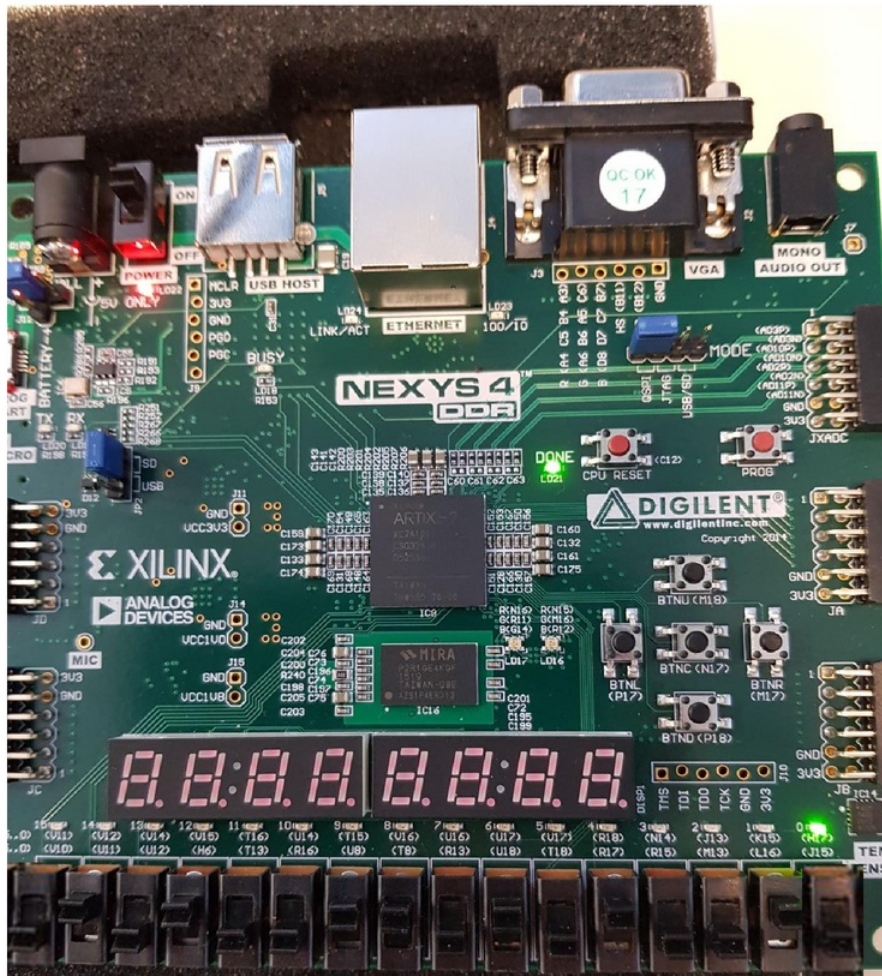


Figure 33 - Practical Testing Via NI FPGA Board MultiSim: Greater

Figure 33 shows the practical results for A being greater than B, that's because the binary values for A is 01010011 and the value for B is 01010010. A equals 83 where B equals 82, this is just like the simulation tested above with the same result of the LED lighting up to show greater than.

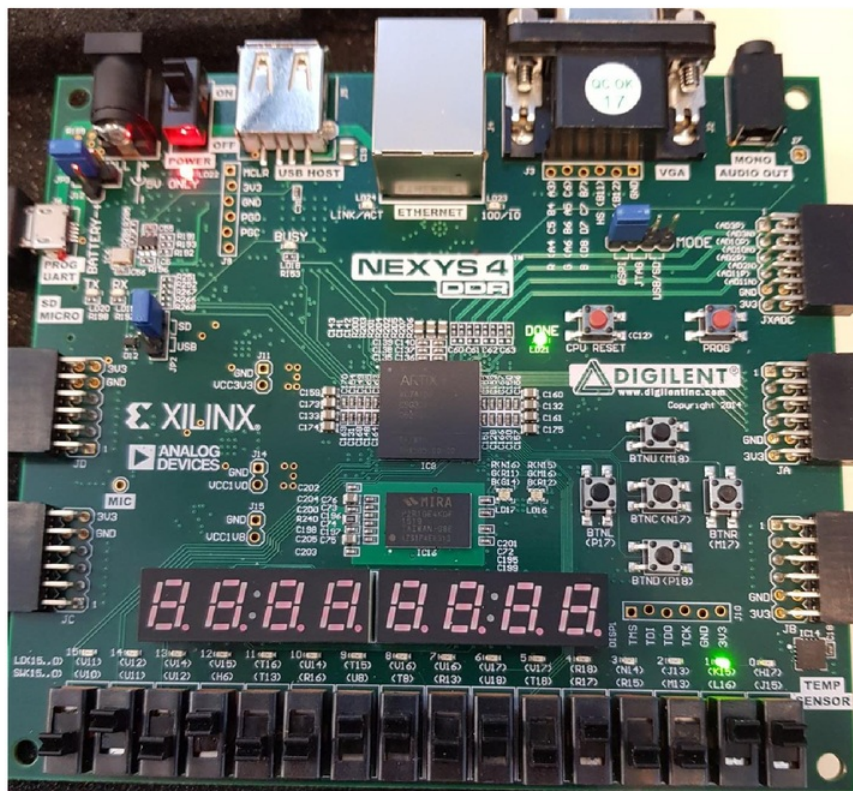


Figure 34 - Practical Testing Via NI FPGA Board MultiSim: Equal

Figure 34 shows the practical results for A being greater than B, that's because the binary values for A is 01010011 and the value for B is 01010011. A equals 83 where B also equals 83, this is just like the simulation tested above with the same result of the LED lighting up to show equal to.

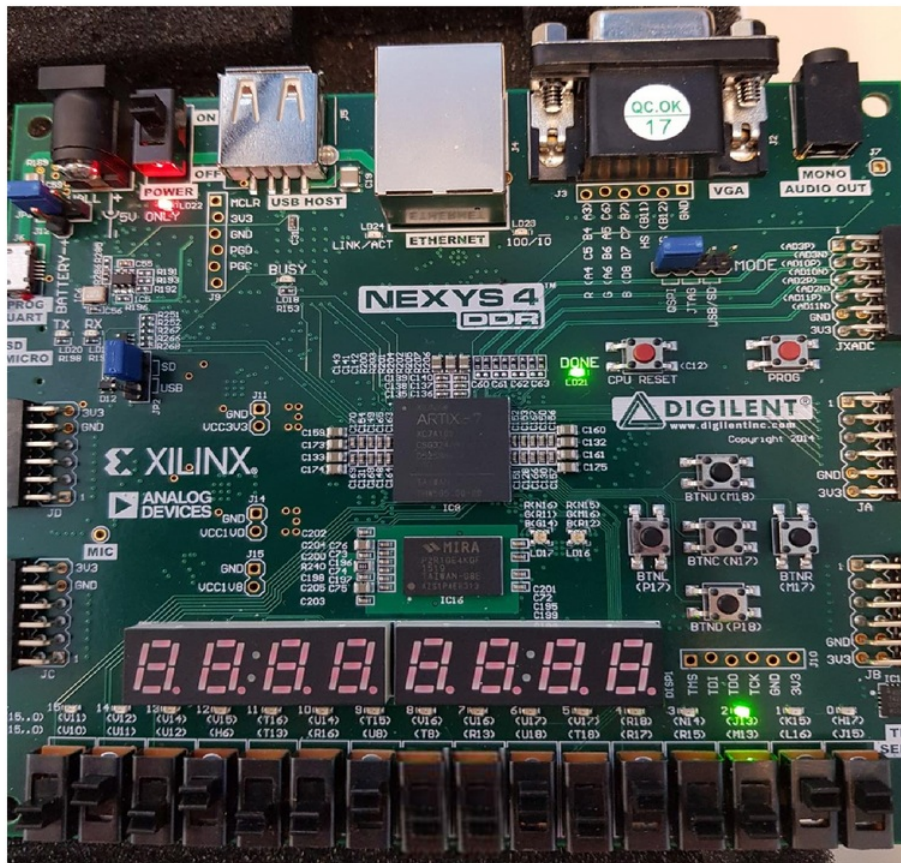


Figure 35 - Practical Testing Via NI FPGA Board MultiSim: Less

Figure 35 shows the practical results for A being less than B, that's because the binary values for A is 01010010 and the value for B is 01010011. A equals 82 where B equals 83, this is just like the simulation tested above with the same result of the LED lighting up to show less than.

Excellent pictorial representation of practical FPGA results and analysed well

7. Adapt Design to Fit New Purposes or Applications

The next part of the assignment is to adapt the LabVIEW design so that the data can be logged and compared on EXCEL.

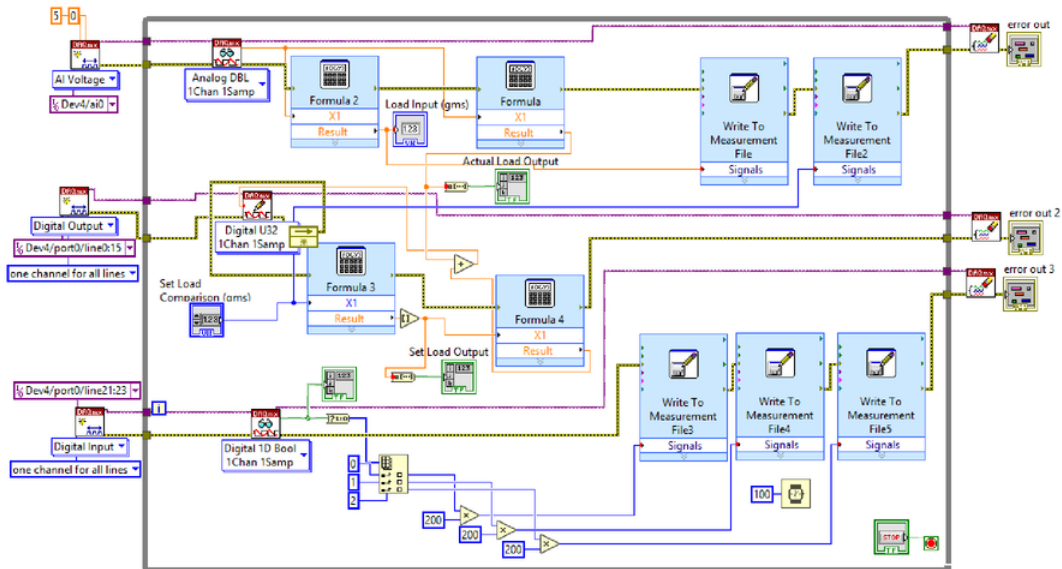


Figure 36 - Adapted LabVIEW 'vi' Design

Figure 36 shows the adapted design from figure 22.

The difference now being that the following information data is now logged:

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

This is done by taking the relevant data parts and inputting them into write to measurement files which will record every change in data over a set time (this can be found in the appendix under data logging).

The difference being that before the data is logged for the less than, equal to or greater than comparator; it is multiplied by 200.

Therefore a result of 1 will now show as 200 on the graph when logged as to be easily identifiable as to what's happening.

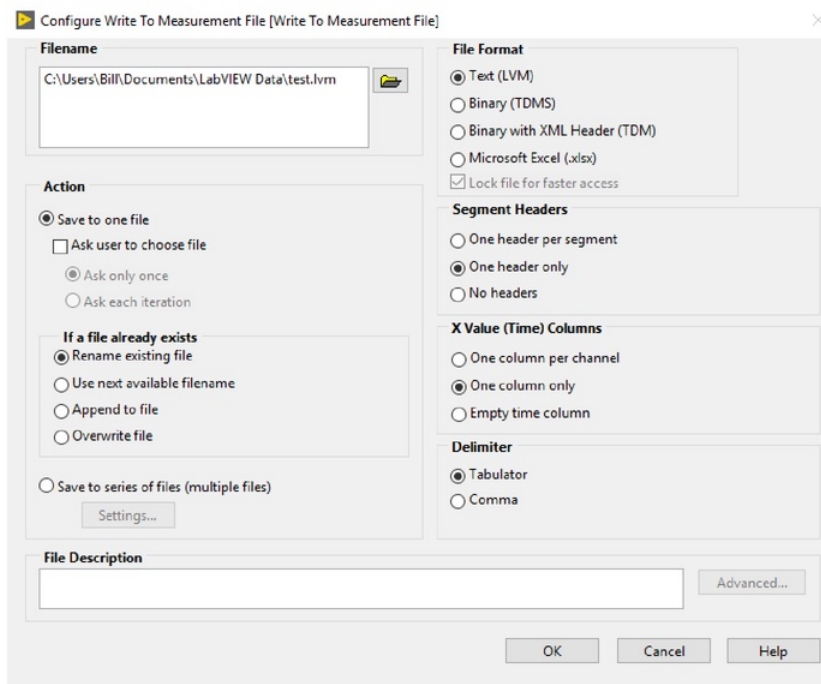


Figure 37 - Write to Measurement File

Figure 37 shows how the files are saved via the write to measurement file.

*Good LabVIEW block diagram shown and design is clear
but not complete*

8. Analysis of Data Logging EXCEL Results

This task will analyse the results taken from the data logging files shown above and in appendix.

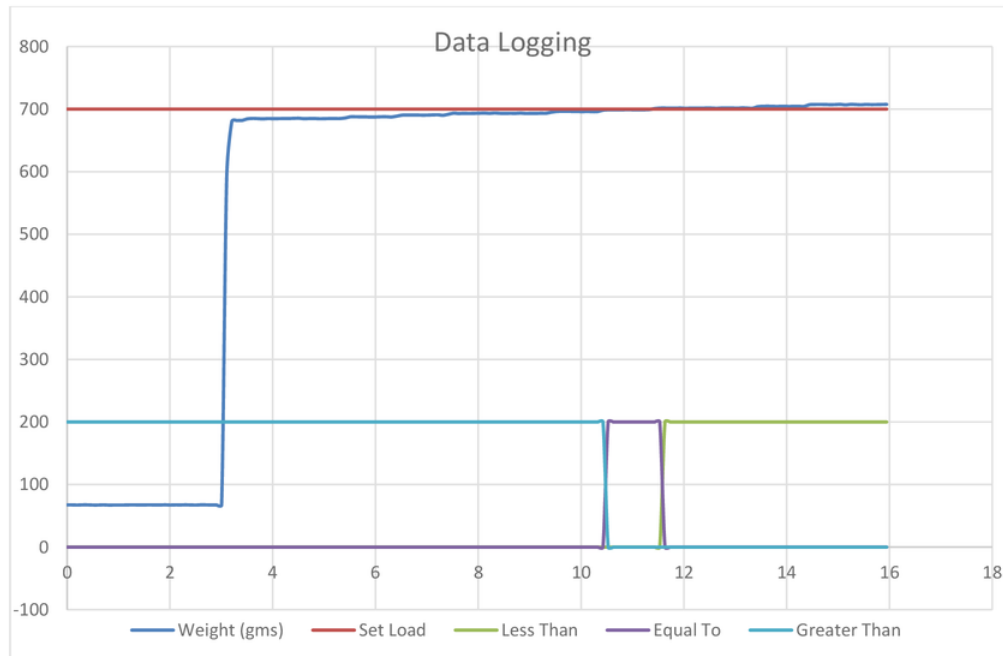


Figure 38 - Data Logging Graph

Figure 38 shows the data which has been logged and turned into a graph.

This shows how over time the weight in grams increases from below 700 to over 700 grams. The set point of 700 grams is greater than the actual load of grams to begin with and therefore the greater than line on the graph is steady at 200 as per the multiplication factor explained above.

Once the actual load reaches 700 grams (698.88 to be precise) the graph now shows equal to at 200 and greater than has dropped to zero. Once the load surpasses 700 grams (701.67 to be precise) the graph now shows less than at 200 and equal to has dropped off to 0.

This has proved once again that the digital comparator is working as per specification and outputting a greater than, equal to or less than signal where needed.

9. Conclusions

To be able to conclude the assignment, the aims and objectives set out in the introduction will be used to see if they have been met successfully or not.

The aims and objectives of this assignment were as follows:

Design a LabVIEW 'vi' that will interface to an FPGA via an NI ELVIS 2 board to give an 8 bit digital comparison between two 8 bit signals. This will be done by designing a MultiSim circuit which will compare the two signals and output 3 more signals whether the comparison is less than, equal to or more than the set value.

This was done successfully by designing the circuit and testing the design by simulating the results and then altering the design so it could be practically tested. They each gave out the correct result for the values input.

The first signal taken in will be an analogue signal which will be 0-5 V in range. This will represent the load in grams (0-1000) which will then be converted into the first 8 bit signal. This signal will be output to the ELVIS board and used to compare signals.

The second signal will be the set value in grams with the same range of 0-1000 grams, this will also be converted to an 8 bit signal and output to the EVLIS board where they can be compared.

Both of the above objectives were met by designing the LabVIEW circuit, as it was able to convert analogue to grams and grams to digital. This was also tested and simulated to prove the design.

Once simulated the program can be downloaded and practically tested on the Nexys 4 DDR FPGA board which will compare the two signals which at this point will be set using the switches on the board and will light up an LED depending whether the signals are less than, equal to or greater than the set point. Each outcome should be tested and results recorded.

This was tested practically and lit up the correct LED dependant on how the switches were set up, these used the same values as the simulated results as to make sure the design worked both practically and simulated.

Once the MultiSim program works correctly, the LabVIEW design should be made to read in the analogue signal and convert to the 8 bit value and also allow a value for the set load to be made.

This should then compare both of these digital output values and convert them so that the two 8 bit values stay separate and don't effect one another.

The digital output comparison should then be used as inputs and read by the program so they can be compared and output a less than, equal to or greater than signal.

As mentioned above, this part of the design was successful.

The final aim of this assignment is to adapt the design already made on LabVIEW so that the actual load cell value in grams over its full range is output. This should be compared to the set value and output the correct 1 of 3 signals.

These values should all be data logged using data logging files and using EXCEL a graph should be produced to show how the value of grams compared to the set point will affect the results.

The design was successfully adapted to allow the data to be logged and analysed. This was shown in the form of a graph and all the data can be seen in the annex.

In conclusion, the assignment has been a success as my knowledge of the subject matter has been improved and this has shown in my understanding of how the design works and interlinks with each other.

All the aims and objectives have successfully been met and all the simulated results match the practical ones. This is why the assignment as a whole has been a success.

Excellent conclusion given which does show some reflection

10. Bibliography

Bibliography

electronics-tutorials. (2018). *httpcombination/comb_8.html*. Retrieved April 2018, from <https://www.electronics-tutorials.ws>: https://www.electronics-tutorials.ws/combination/comb_8.html

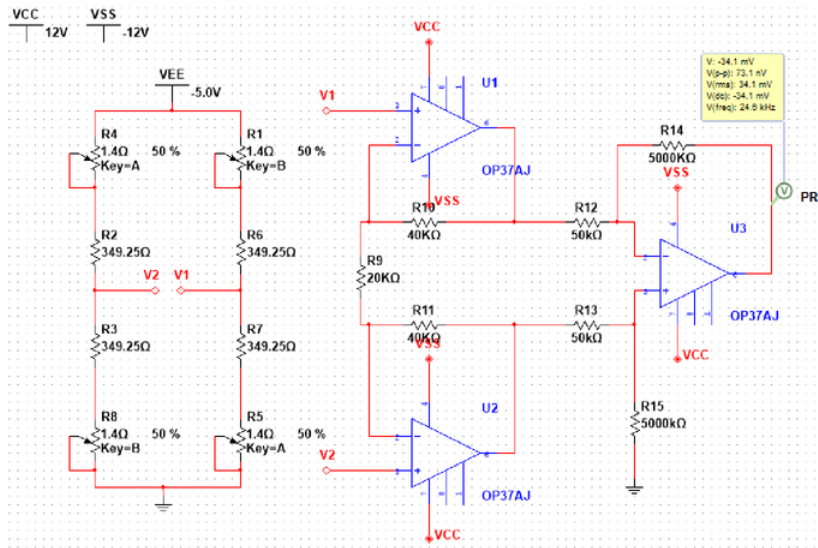
microcontrollerslab. (2017). *analog-to-digital-adc-converter-working/*. Retrieved April 2018, from microcontrollerslab.com: <http://microcontrollerslab.com/analog-to-digital-adc-converter-working/>

Omega. (2003-2013). *Sub Miniature Load Cell*. Retrieved September 2013, from <http://www.omega.com/pptst/LCKD.html>

Phillips. (1990). *4-bit magnitude comparator 74HC/HCT85*. Retrieved April 2018, from
<file:///C:/Users/new/Downloads/74HC85%204%20bit%20Magnitude%20Comparator.pdf>:
<file:///C:/Users/new/Downloads/74HC85%204%20bit%20Magnitude%20Comparator.pdf>

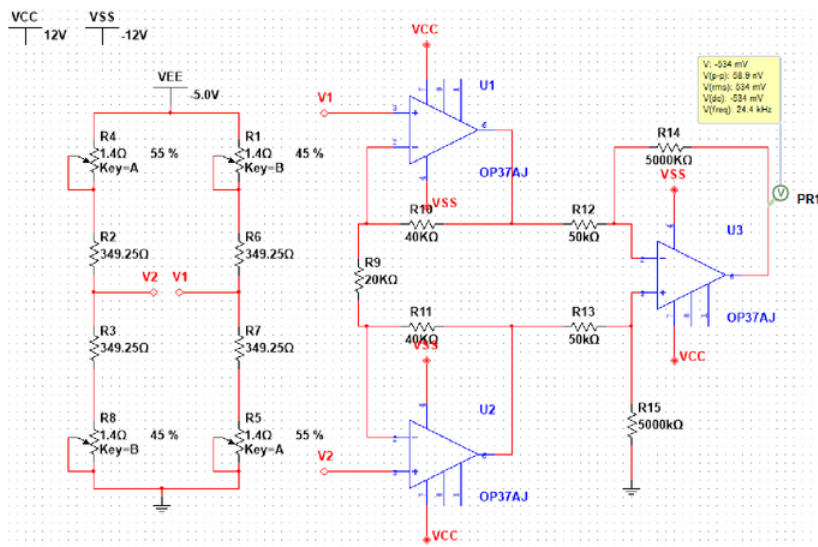
References could have been more varied...

11. Appendices



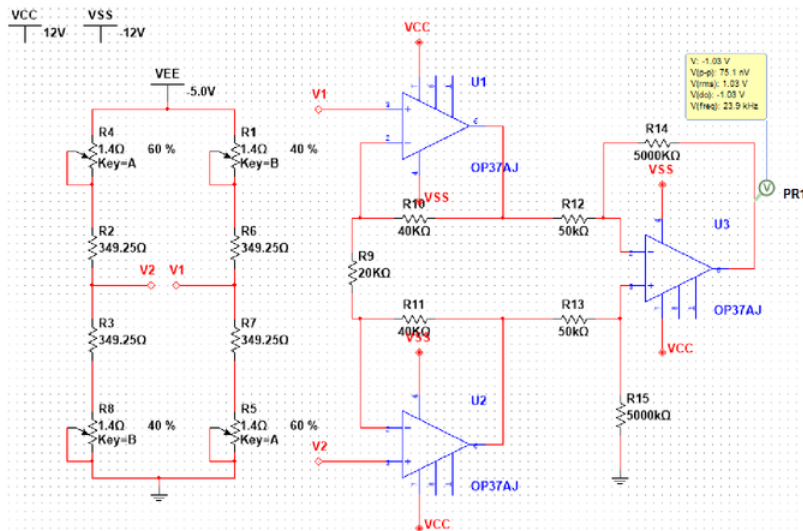
V: -34.1 mV
V(p-p): 22.4 nV
V(rms): 34.1 mV
V(dc): -34.1 mV
V(freq): 24.4 kHz

Vrms = 34.1 mV (basically zero) for equal load



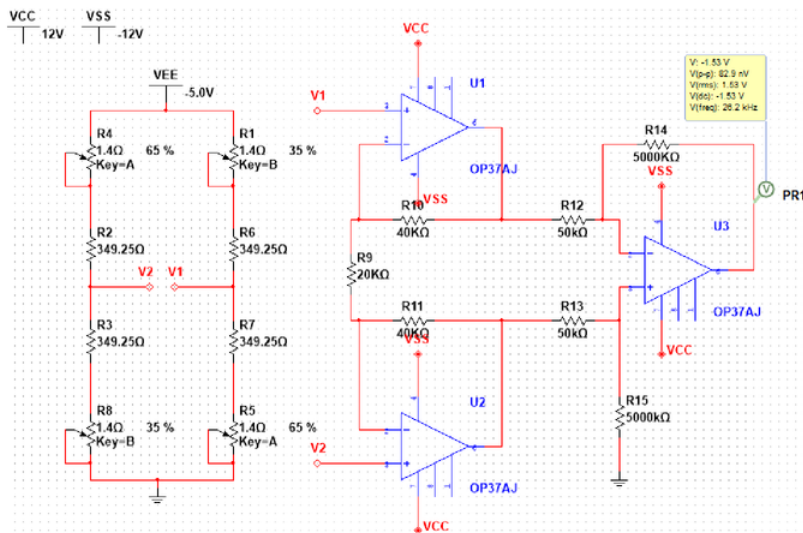
V: -534 mV
V(p-p): 68.7 mV
V(rms): 534 mV
V(dc): -534 mV
V(freq): 25.0 kHz

Vrms = 534 mV for load of 55% to 45%



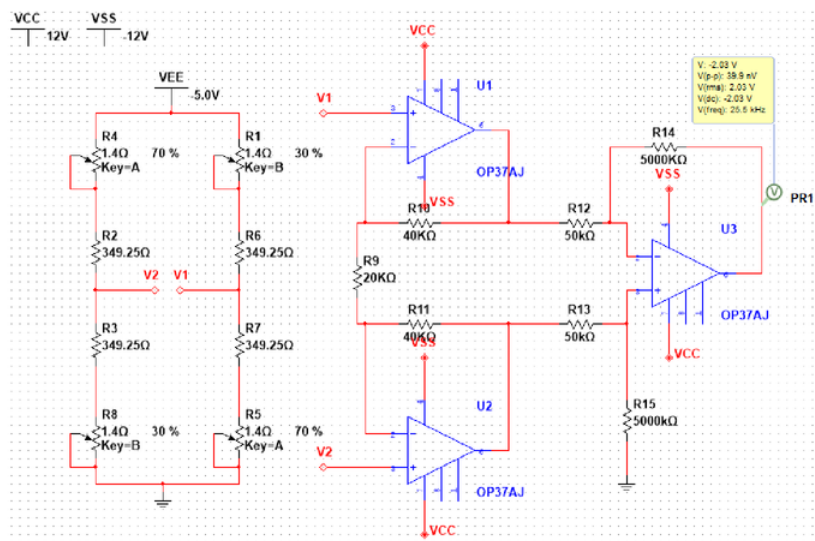
V: -1.03 V
V(p-p): 35.5 mV
V(rms): 1.03 V
V(dc): -1.03 V
V(freq): 24.3 kHz

Vrms = 1.03 V for load of 60% to 40%



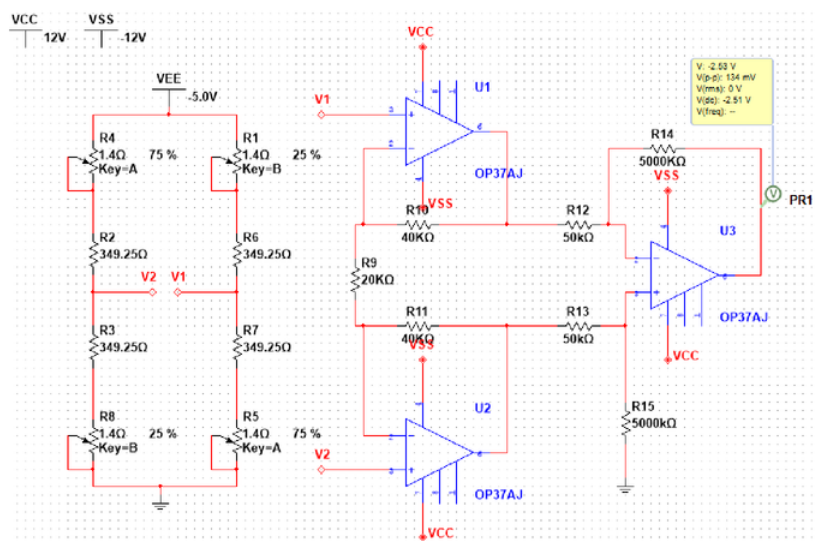
V: -1.53 V
V(p-p): 71.2 mV
V(rms): 1.53 V
V(dc): -1.53 V
V(freq): 20.4 kHz

Vrms = 1.53 V for load of 65% to 35%



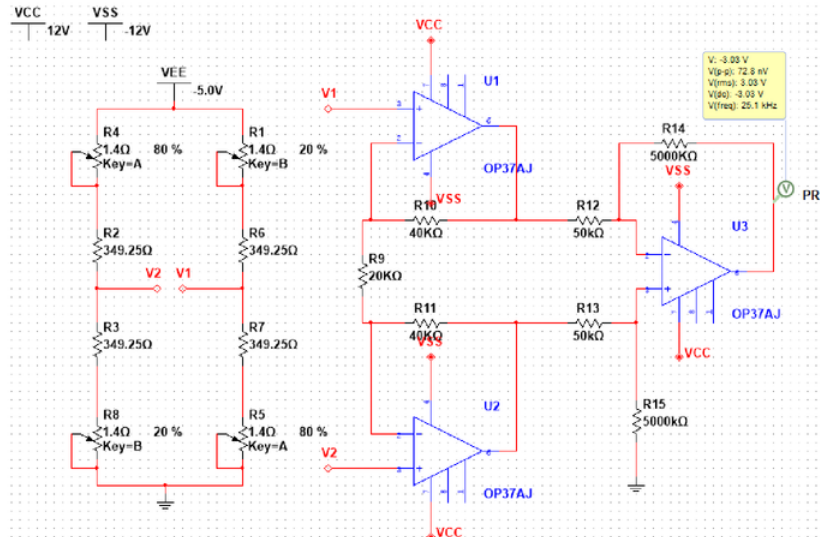
V: -2.03 V
V(p-p): 58.9 mV
V(rms): 2.03 V
V(dc): -2.03 V
V(freq): 25.6 kHz

Vrms = 2.03 V for load of 70% to 30%



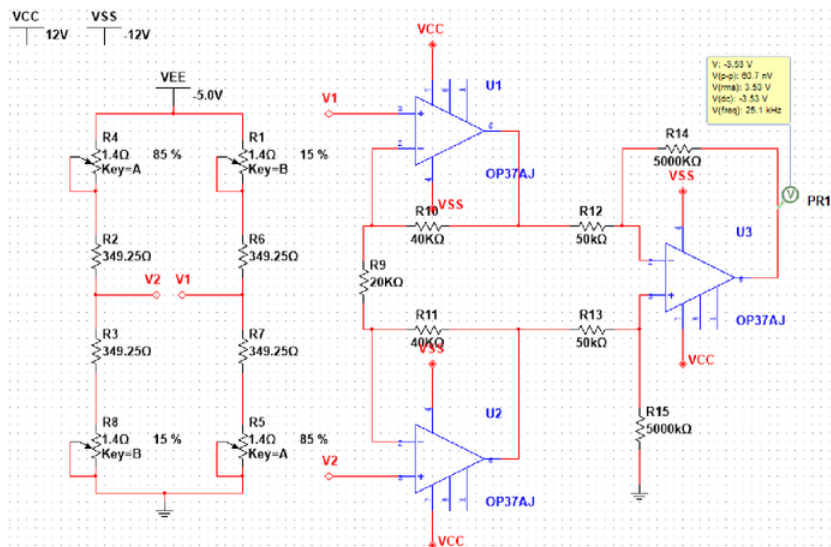
V: -2.53 V
V(p-p): 50.4 nV
V(rms): 2.53 V
V(dc): -2.53 V
V(freq): 25.2 kHz

Vrms = 2.53 V for load of 75% to 25%



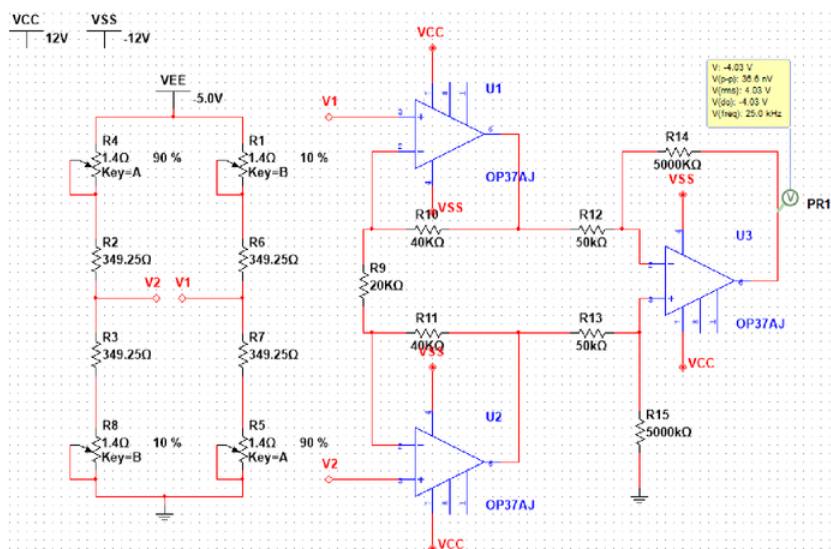
V: -3.03 V
V(p-p): 30.7 nV
V(rms): 3.03 V
V(dc): -3.03 V
V(freq): 25.8 kHz

Vrms = 3.03 V for load of 80% to 20%



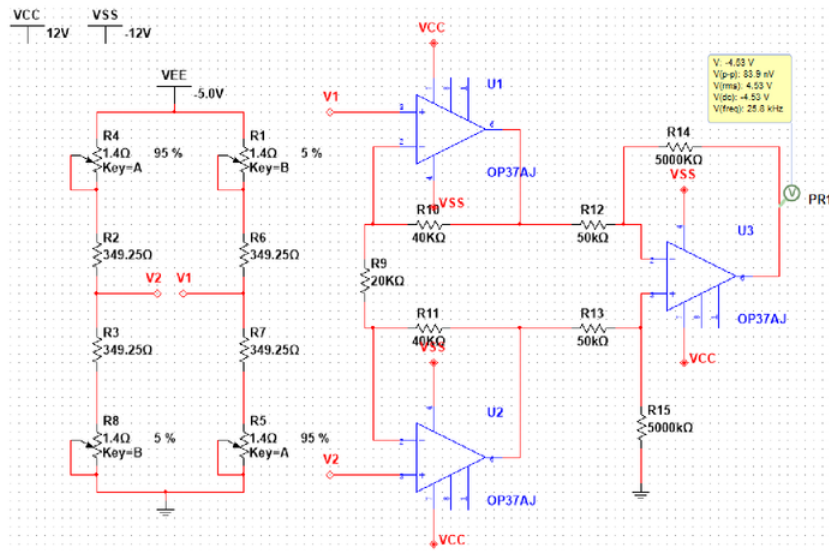
V: -3.53 V
V(p-p): 52.3 nV
V(rms): 3.53 V
V(dc): -3.53 V
V(freq): 25.5 kHz

Vrms = 3.53 V for load of 85% to 15%



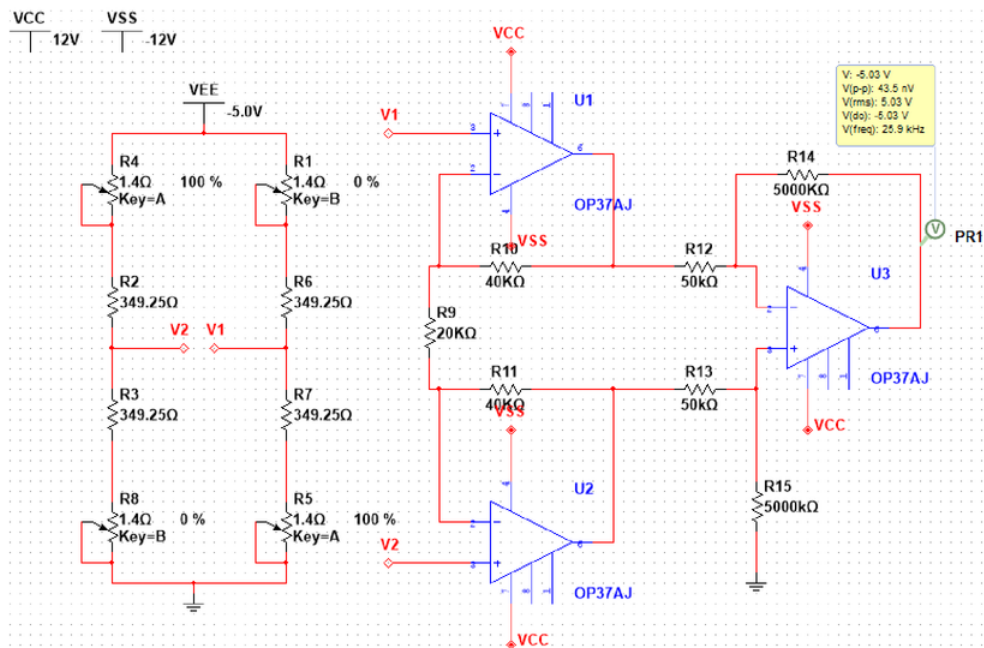
V: -4.03 V
V(p-p): 71.7 nV
V(rms): 4.03 V
V(dc): -4.03 V
V(freq): 28.2 kHz

Vrms = 4.03 V for load of 90% to 10%



V: -4.53 V
V(p-p): 51.1 nV
V(rms): 4.53 V
V(dc): -4.53 V
V(freq): 25.3 kHz

Vrms = 4.53 V for load of 95% to 5%



V: -5.03 V
V(p-p): 68.7 nV
V(rms): 5.03 V
V(dc): -5.03 V
V(freq): 26.0 kHz

Vrms = 5.03 V for load of 100% to 0%

Data Logging

Voltage	Weight (gms)	Set Load	Less Than	Equal To	Greater Than
0	67.510107	700	0	0	200
0.099267	67.4457	700	0	0	200
0.200534	67.252477	700	0	0	200
0.300829	67.510107	700	0	0	200
0.401067	67.510107	700	0	0	200
0.500824	67.220273	700	0	0	200
0.601107	67.220273	700	0	0	200
0.701358	67.477903	700	0	0	200
0.802628	67.188069	700	0	0	200
0.901891	67.220273	700	0	0	200
1.003164	67.252477	700	0	0	200
1.101422	67.284681	700	0	0	200
1.204697	67.4457	700	0	0	200
1.30296	67.510107	700	0	0	200
1.404228	67.316884	700	0	0	200
1.503653	67.381292	700	0	0	200

1.60393	67.510107	700	0	0	200
1.704891	67.349088	700	0	0	200
1.805181	67.349088	700	0	0	200
1.905424	67.542311	700	0	0	200
2.005602	67.284681	700	0	0	200
2.104866	67.4457	700	0	0	200
2.205133	67.413496	700	0	0	200
2.305402	67.542311	700	0	0	200
2.405666	67.381292	700	0	0	200
2.5044	67.220273	700	0	0	200
2.605673	67.574515	700	0	0	200
2.704936	67.510107	700	0	0	200
2.806204	67.316884	700	0	0	200
2.906472	67.381292	700	0	0	200
3.005855	67.316884	700	0	0	200
3.105223	591.280284	700	0	0	200
3.205387	680.295396	700	0	0	200
3.307687	681.744638	700	0	0	200
3.408335	682.066691	700	0	0	200
3.505758	684.417684	700	0	0	200
3.606026	685.126202	700	0	0	200
3.707297	684.868559	700	0	0	200
3.807596	684.482095	700	0	0	200
3.907008	684.93297	700	0	0	200
4.007588	684.804148	700	0	0	200
4.106852	684.868559	700	0	0	200
4.206118	684.997381	700	0	0	200
4.308387	684.997381	700	0	0	200
4.411547	685.287229	700	0	0	200
4.508726	685.416051	700	0	0	200
4.607958	684.771943	700	0	0	200
4.709228	684.965175	700	0	0	200
4.809494	685.061791	700	0	0	200
4.908762	684.675327	700	0	0	200
5.009125	684.804148	700	0	0	200
5.110296	684.997381	700	0	0	200
5.210562	684.997381	700	0	0	200
5.309826	684.997381	700	0	0	200
5.413101	685.931337	700	0	0	200
5.512558	687.734839	700	0	0	200
5.612826	687.79925	700	0	0	200
5.712091	687.670428	700	0	0	200
5.812358	687.638223	700	0	0	200
5.913656	687.348374	700	0	0	200
6.012214	687.509401	700	0	0	200
6.113487	687.79925	700	0	0	200

6.212748	687.831455	700	0	0	200
6.314018	687.477196	700	0	0	200
6.415454	688.797617	700	0	0	200
6.514618	690.343476	700	0	0	200
6.613867	690.60112	700	0	0	200
6.714203	690.568914	700	0	0	200
6.81539	690.504503	700	0	0	200
6.915752	690.375682	700	0	0	200
7.016265	690.440093	700	0	0	200
7.116532	690.762147	700	0	0	200
7.215796	690.890968	700	0	0	200
7.316065	690.182449	700	0	0	200
7.418503	691.79272	700	0	0	200
7.521601	693.531812	700	0	0	200
7.616859	692.952115	700	0	0	200
7.717876	693.241963	700	0	0	200
7.818142	693.048731	700	0	0	200
7.920424	693.209758	700	0	0	200
8.020682	693.467401	700	0	0	200
8.120948	693.692839	700	0	0	200
8.220212	693.241963	700	0	0	200
8.321482	693.821661	700	0	0	200
8.421849	693.306374	700	0	0	200
8.52073	693.177553	700	0	0	200
8.621107	693.306374	700	0	0	200
8.722247	693.209758	700	0	0	200
8.821501	693.596223	700	0	0	200
8.920794	693.080936	700	0	0	200
9.022201	693.177553	700	0	0	200
9.121466	693.177553	700	0	0	200
9.220727	693.402991	700	0	0	200
9.321088	693.435196	700	0	0	200
9.422458	695.013262	700	0	0	200
9.524559	696.237068	700	0	0	200
9.622925	696.430301	700	0	0	200
9.724095	696.591328	700	0	0	200
9.825357	696.333684	700	0	0	200
9.925624	696.301479	700	0	0	200
10.02513	696.01163	700	0	0	200
10.1254	696.333684	700	0	0	200
10.22767	695.915014	700	0	0	200
10.32793	696.237068	700	0	0	200
10.42671	698.298216	700	0	0	200
10.52703	698.877914	700	0	200	0
10.62925	698.877914	700	0	200	0
10.72851	698.942325	700	0	200	0

10.82885	699.006736	700	0	200	0
10.92905	699.135558	700	0	200	0
11.02931	698.910119	700	0	200	0
11.13058	698.845709	700	0	200	0
11.22985	698.97453	700	0	200	0
11.33116	699.393201	700	0	200	0
11.43138	700.874652	700	0	200	0
11.53112	701.679788	700	0	200	0
11.63239	701.744199	700	200	0	0
11.73466	701.647583	700	200	0	0
11.83293	701.615377	700	200	0	0
11.93322	701.776405	700	200	0	0
12.03346	701.357734	700	200	0	0
12.13373	701.647583	700	200	0	0
12.234	701.679788	700	200	0	0
12.33526	701.647583	700	200	0	0
12.43611	701.776405	700	200	0	0
12.53821	701.744199	700	200	0	0
12.63755	701.422145	700	200	0	0
12.73774	701.80861	700	200	0	0
12.83813	701.840816	700	200	0	0
12.93796	701.873021	700	200	0	0
13.03822	701.583172	700	200	0	0
13.13962	702.066254	700	200	0	0
13.23876	701.744199	700	200	0	0
13.33902	701.486556	700	200	0	0
13.43986	703.934171	700	200	0	0
13.54013	704.513869	700	200	0	0
13.6404	704.642691	700	200	0	0
13.73974	704.417253	700	200	0	0
13.84001	704.385047	700	200	0	0
13.94028	704.610485	700	200	0	0
14.04168	704.320636	700	200	0	0
14.14095	704.513869	700	200	0	0
14.24251	704.449458	700	200	0	0
14.34278	704.449458	700	200	0	0
14.44257	707.025896	700	200	0	0
14.54284	707.380156	700	200	0	0
14.64411	707.412362	700	200	0	0
14.74538	707.380156	700	200	0	0
14.84565	707.122512	700	200	0	0
14.94491	707.251334	700	200	0	0
15.04518	707.573389	700	200	0	0
15.14544	706.768252	700	200	0	0
15.2455	707.573389	700	200	0	0
15.34485	707.315745	700	200	0	0

15.4455	706.961485	700	200	0	0
15.54567	707.412362	700	200	0	0
15.64493	707.186923	700	200	0	0
15.7462	707.251334	700	200	0	0
15.84847	707.412362	700	200	0	0
15.94673	707.573389	700	200	0	0

Assignment_2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

71 /100

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

Text Comment. Good overall understanding shown on the DAQmx icons used within the assignment

PAGE 13

Text Comment. Good

PAGE 14

Text Comment. Good explanation given as to a 8 bit design

PAGE 15

PAGE 16

Text Comment. Good explanation on how to signal condition a load cell but not as to how they work!

PAGE 17

Text Comment. Good design circuit shown and understood

PAGE 18

PAGE 19

Text Comment. Good explanation given

PAGE 20

PAGE 21

PAGE 22

Text Comment. Good, clear explanation of the design of the LabVIEW 'vi'

PAGE 23

PAGE 24

PAGE 25

Text Comment. Very clear understanding shown through the analysis of the Multisim results

PAGE 26

Text Comment. Correct graphical analysis shown and analysed correctly

PAGE 27

PAGE 28

PAGE 29

Text Comment. Excellent analysis of LabVIEW results.

PAGE 30

PAGE 31

Text Comment. Not a totally clear explanation given

PAGE 32

Text Comment. Excellent pictorial representation of practical FPGA results and analysed well

Text Comment. Good LabVIEW block diagram shown and design is clear but not complete

Text Comment. Good analysis of graphical results

Text Comment. Excellent conclusion given which does show some reflection

Text Comment. References could have been more varied...

JOSHUA BEER**71****RESEARCH-INFORMED LITERATURE**

 Analogue / Digital Converters: Basic research shown (2/5) Load Cells: Good explanation on how to signal condition a load cell but not as to how they work! (2/5) Digital Magnitude Comparators: Good explanation given as to a 8 bit design (3/5) NI DAQmx in LabVIEW: Good overall understanding shown on the DAQmx icons used within the assignment (4/5)

11**KNOWLEDGE AND UNDERSTANDING**

 Design of the signal conditioning circuit for LCKD-1Kg Load Cell: Very good detail shown and understood (4/5) Design of LabVIEW 'vi': Good, clear explanation of the design of the LabVIEW 'vi' (8/10) Design of 4/8-bit Magnitude Comparator: Good design circuit shown and understood (7/10) Re-design to suit new applications: Good LabVIEW block diagram shown and design is clear but not complete (3/5)

22**ANALYSIS**

 Analysis of signal conditioning circuit: Correct graphical analysis shown and analysed correctly (4/5) Analysis of MultiSim Results: Very clear

20

understanding shown through the analysis of the Multisim results (4/5) Analysis of NI LabVIEW / FPGA results: Excellent analysis of LabVIEW results. Excellent pictorial representation of practical FPGA results and analysed well (8/10) Analysis of data logging files in EXCEL: Good analysis of graphical results (4/5)

PRACTICAL APPLICATION AND DEP.



Build interface between LabVIEW and the FPGA board via the NI ELVIS 11 board: Good pictorial evidence shown but the explanation given is not clear (6/10)

6

SKILLS FOR PROFESSIONAL PRAC.



Deployment of appropriate media: Excellent use made of pictorial evidence and screen dumps in design and analysis (4/5) Report layout and readability: Very good report which covered all aspects of the assignment well (4/5) Appropriate conclusion: Excellent conclusion given which does show some reflection (4/5)

12