

Assignment 2

by Emyr Dyer

FILE	7311_EMYR_DYER_ASSIGNMENT_2_157347_2001935557.DOCX		
TIME SUBMITTED	26-APR-2018 09:15AM (UTC+0100)	WORD COUNT	3650
SUBMISSION ID	87505946	CHARACTER COUNT	28036

HND Electrical & Electronic Engineering

NI LabVIEW Magnitude Comparator

PSDE5029 – Virtual Instrumentation and
Control Systems

ED067180

Submission Date: 26/04/2018

Word Count: 3585

CONTENTS

1	Introduction	1
2	Research.....	2
2.1	What is DAQmx?	2
2.1.1	Create Channel.....	2
2.1.2	Read	3
2.1.3	Write	3
2.1.4	Add	3
2.1.5	Feedback	4
2.1.6	Formula	4
2.1.7	Number to Boolean Array	4
2.1.8	Round to Nearest.....	4
2.1.9	While Loop	5
2.1.10	Wait.....	5
2.1.11	Error out.....	5
2.2	Digital Magnitude Comparators.....	6
2.3	Analogue to Digital Converters	7
2.4	Load Cells	8
3	Design of Assignment Requirements	10
3.1	8-bit Comparator	10
3.2	Signal Conditioning Unit.....	12
3.3	The LabVIEW vi.....	13
4	Analysis of Signal Conditioning Circuit, NI LabVIEW and MultiSim simulated results	16
4.1	Signal Conditioning Circuit	16
4.2	8-bit Comparator	17
4.3	The LabVIEW vi.....	19
5	Practical Build and Test via NI ELVIS board.....	21
6	Analysis of NI LabVIEW / FPGA results.....	23
6.1	FPGA.....	23
6.2	LabVIEW	25
7	Adapt Design to Fit New Purposes or applications.....	27
8	Analysis of Data Logging Excel results	29
9	Conclusions	30
10	Citations	31

11	Image Sources	32
----	---------------------	----

Figures

Figure 2.1: AI Voltage Basic.....	2
Figure 2.2: DI Digital Input	2
Figure 2.3: DO Digital Output.....	2
Figure 2.4: Analogue Single Channel/Single Sample.....	3
Figure 2.5: Boolean Single Channel/Single Sample.....	3
Figure 2.6: Digital Unsigned 32-bit Integer write.....	3
Figure 2.7: Add Function	3
Figure 2.8: Feedback Node	4
Figure 2.9: Formula	4
Figure 2.10: Number to Boolean Array	4
Figure 2.11: Round to Nearest	4
Figure 2.12: While Loop	5
Figure 2.13: Wait.....	5
Figure 2.14: Error out.....	5
Figure 2.15: Logic gate layout inside the 4-bit comparator.....	6
Figure 2.16: 12-bit Magnitude Comparator modified to 8-bit.....	7
Figure 2.17: LCKG load cell	8
Figure 2.18: Wheatstone Bridge Circuit.....	8
Figure 2.19: Strain Gauge.....	8
Figure 3.1: Simulation version of 8-bit Comparator	10
Figure 3.2: Downloadable version of FPGA 8-bit Comparator.....	11
Figure 3.3: xdc file	11
Figure 3.4: Multisim version of a Signal Conditioning Unit.....	12
Figure 3.5	12
Figure 3.6: First Iteration (ADC)	13
Figure 3.7: First Iteration (ADC) Front Panel.....	13
Figure 3.8: Addition of Digital Input, Comparator and Gram Conversion	14
Figure 3.9: Front Panel of vi	15
Figure 4.1: Instrumentation Amplifier	16
Figure 4.2: Vout over Load	17
Figure 4.3: A Less Than B	17
Figure 4.4: A Equal to B.....	18

Figure 4.5: A Greater Than B.....	18
Figure 4.6: ADC Test 1	19
Figure 4.7: ADC Test 2	19
Figure 4.8: ADC Test 3	20
Figure 5.1: Physical Build	21
Figure 5.2: Amended 8-bit Comparator	22
Figure 6.1: A Less Than B	23
Figure 6.2: A Equal To B	24
Figure 6.3: A Greater Than B.....	24
Figure 6.4: Less Than.....	25
Figure 6.5: Equal To.....	25
Figure 6.6: Greater Than	26
Figure 7.1: Data Logging Added	27
Figure 7.2: Configuration of the Data Logging tools	28
Figure 7.3: Supply adjustment	28
Figure 8.1: Output of LabVIEW vi.....	29
Figure 9.1: LattePanda Board running Windows 10 with 7" screen.....	30

1 INTRODUCTION

The ultimate objective is to build a virtual weighing scale system that gives an indicator of the load's weight being over or under a set value. To achieve this, an FPGA (Nexus 4) will be programmed in PLD using Multisim to develop the comparison function. To process the incoming and outgoing signals read from and sent to the FPGA through the ELVIS II board, an application will be developed in LabView 2017 developed by National Instruments.

Good, concise introduction....

2 RESEARCH

2.1 WHAT IS DAQMX?

DAQmx is National Instrument's set of tools used for developing Data Acquisition Applications which was released in 2003 (National Instruments Corporation, 2018). The tools used in the design for the Analogue to Digital Converter are the following.

2.1.1 Create Channel

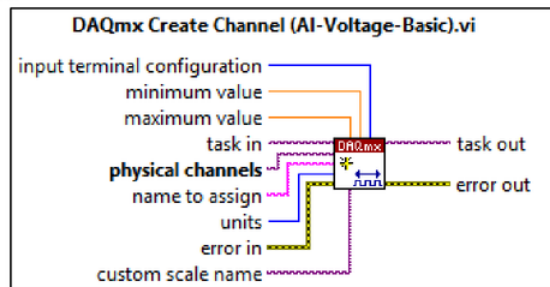


Figure 2.1: AI Voltage Basic

The Create Channel Tool is used to link to the connectors of the Elvis II prototyping board. The Analogue Voltage Input setting for the tool measures an external voltage source, with the measurable range fixed with a minimum and maximum value.

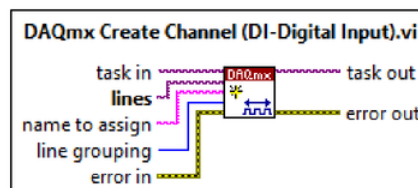


Figure 2.2: DI Digital Input

The Digital Input setting for the tool reads the ports on the board as lines, which can specify an individual port or a range of ports. To read multiple individual ports, the user needs to add additional channels for the other additional ports.

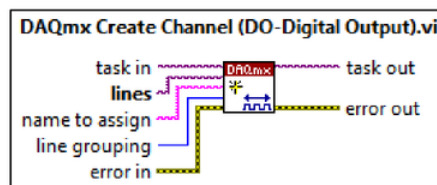


Figure 2.3: DO Digital Output

Functioning similarly to the Input setting of the channel, this version of the Channel writes to the specified ports on the Elvis II development board

2.1.2 Read

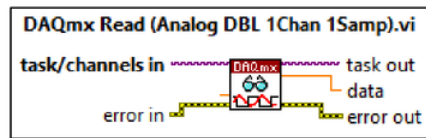


Figure 2.4: Analogue Single Channel/Single Sample

From an Analogue Input Voltage Channel, a voltage measurement with a single floating point is read and passed on to be processed by a mathematical operation be it a formula or data type conversion.

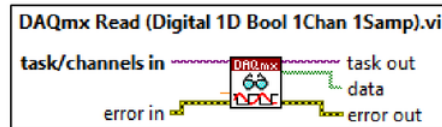


Figure 2.5: Boolean Single Channel/Single Sample

From a Digital Input Channel, a Boolean state is read and passed on to function to display the state, and then processed for data logging.

2.1.3 Write

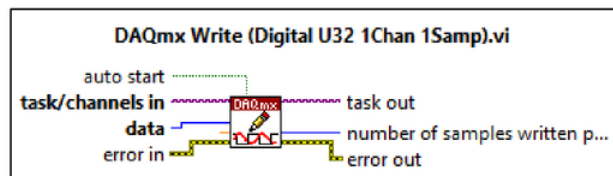


Figure 2.6: Digital Unsigned 32-bit Integer write

This tool reads in the processed integer and writes to the lines specified in the Digital Output Channel.

2.1.4 Add

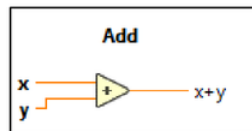


Figure 2.7: Add Function

Combines two inputs and joins values together into one output.

2.1.5 Feedback

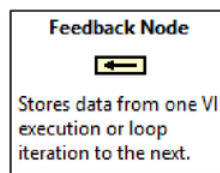


Figure 2.8: Feedback Node

The Feedback Node is used to pass data from one DAQmx function to another.

2.1.6 Formula

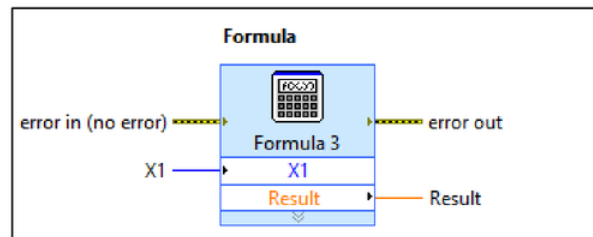


Figure 2.9: Formula

The formula function reads in a variable as X1, which is then processed by a calculation derived by the user and is then output to another function.

2.1.7 Number to Boolean Array

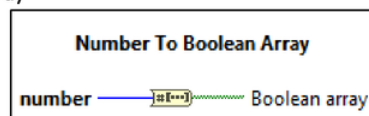


Figure 2.10: Number to Boolean Array

This function simply converts an integer to an array, the size of which is determined by the number of bits found in the integer.

2.1.8 Round to Nearest

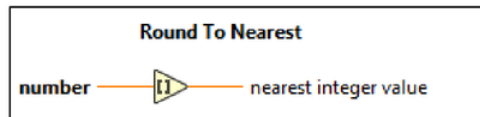


Figure 2.11: Round to Nearest

This function rounds the input number to the nearest even integer, which would be required for processing.

2.1.9 While Loop

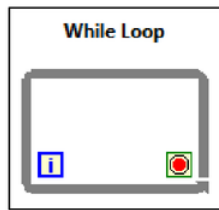


Figure 2.12: While Loop

The While Loop is used to execute any instructions contained within until a specified condition is satisfied, be it a fixed duration in time or a manual stop instruction.

2.1.10 Wait

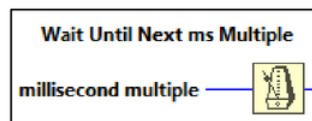


Figure 2.13: Wait

The wait function is required control the duration of the While Loop. With a longer wait, the sampling rate decreases. The time set is measured in milliseconds.

2.1.11 Error out

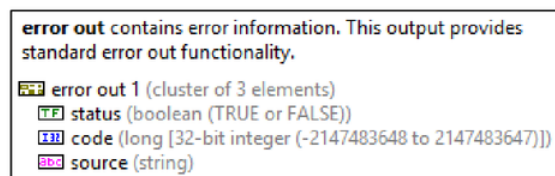


Figure 2.14: Error out

This is used for fault finding in the code or system, which triggers when an unexpected state or value is detected.

(Add the write to file function)

*You didn't add the 'write to measurement file'!
Good detail on DAQmx icons which is what the
brief asked for...*

2.2 DIGITAL MAGNITUDE COMPARATORS

Required for the design is an 8-bit comparator, used to compare two 8-bit words and determine their equality. To build an 8-bit, two 4-bit comparators are used.

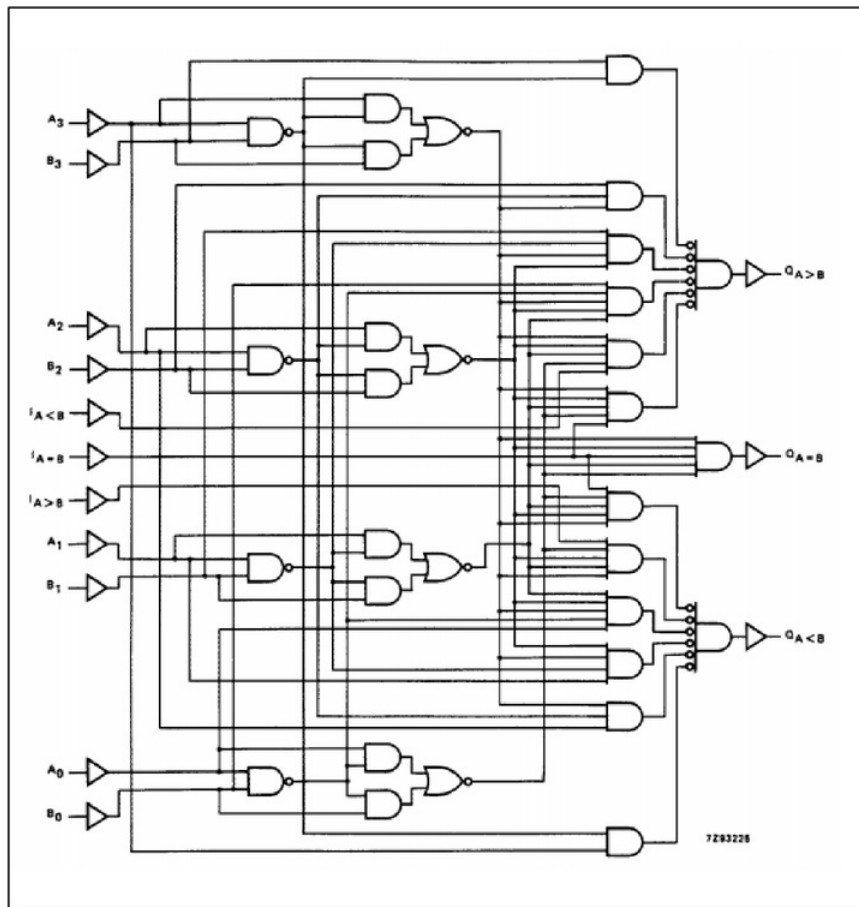


Figure 2.15: Logic gate layout inside the 4-bit comparator

By consulting a datasheet for a CMOS 4-bit comparator, a design is available for a 12-bit cascaded comparator, which can be modified into an 8-bit by removing the third comparator from the design (Nexperia, 1990).

Good schematic diagram shown but how does it work?

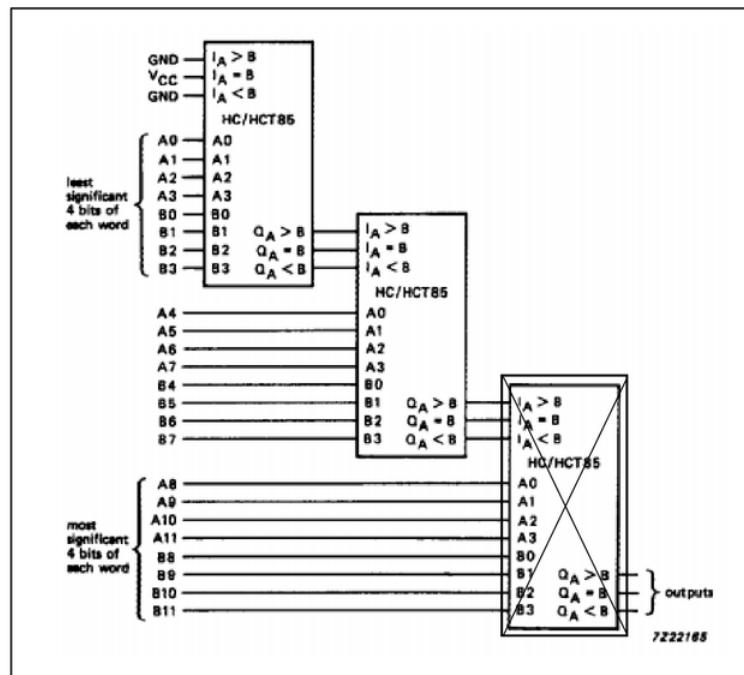


Figure 2.16: 12-bit Magnitude Comparator modified to 8-bit

Good detail

2.3 ANALOGUE TO DIGITAL CONVERTERS

Often a Controller (FPGA, Microcontroller, PLC) needs to control an analogue device which could be a motor or speaker for example. In the current age of computing, all internal processing is exclusively a digital process, so there needs to be some form of conversion to allow analogue signals to be processed.

To convert an analogue signal to a digital value, an ADC is used to convert the magnitude of an analogue voltage signal into a discrete digital value. Some of the more common ADCs available are 8-bit ($2^8 = 256$ levels), 10-bit ($2^{10} = 1,024$ levels) or 16-bit ($2^{16} = 65,536$ levels), so with a higher bit value, the more accurately an analogue signal can be converted to digital.

The converted value is a ratiometric value (Applied Measurements Ltd., 2018), meaning that when using a 10-bit ADC, a voltage of 0 to 5V covers a range of 0 to 1,023.

$$\frac{ADC \text{ Resolution}}{System \text{ Voltage}} = \frac{ADC \text{ Reading}}{Voltage \text{ Measured}} \quad \text{Good}$$

So, assuming an input voltage of say 3.14V is measured, the digital value represented is 642 (rounded down) assuming a 10-bit ADC is used.

$$\frac{1023}{5} = \frac{x}{3.14}$$

$$\frac{1023}{5} \times 3.14 = x = 642$$

Should be shown as $(V_{in} / V_{ref}) \times \text{number of bits}$ In this case $(3.14 / 5) \times 1024 = 643$ it

1
0
2
4
s
(
0
-
1

2.4 LOAD CELLS

For the system under design, a Sub-Miniature Industrial Compression Load Cell manufactured by Omega (1KG model) is considered. The design of the Load Cell is manufactured from stainless steel and is thermally compensated by a circuit contained in the cable linked to the cell, ensuring an accurate reading (OMEGA Engineering inc., 2018).

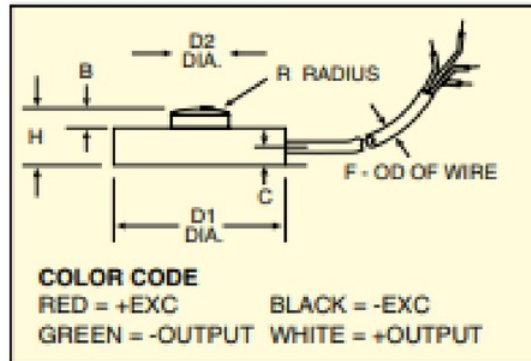


Figure 2.17: LCKG load cell

In 1843, English physicist Sir Charles Wheatstone developed what is known as the Wheatstone Bridge Circuit. When pressure is applied to the strain gauges, which are typically in pairs of two, resistance is altered. When compression is applied to the grid of the strain gauge, the area thickens, and the resistance is decreased. The change in resistance is normally linear, so a ratiometric relation between pressure and resistance should be consistent (Window, 1992).

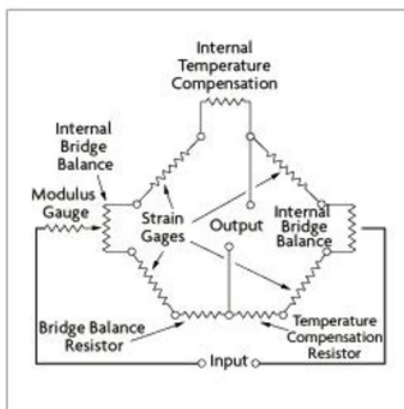


Figure 2.18: Wheatstone Bridge Circuit

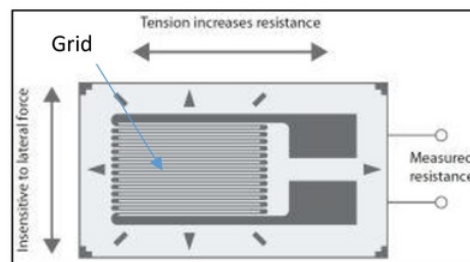


Figure 2.19: Strain Gauge

An amplifier would also be used with the Wheatstone Bridge to scale up the small signals produced by the Strain Gauges to produce a more measurable signal, which in turn is sent to a measuring device (FPGA, PLC, Microcontroller etc). Using a 5V excitation voltage instead of 3.3V improves the signal to noise ratio, however when increasing the voltage, self-heating can become an issue, so it is important not to reach the practical limit of the device (Frank Wenner, 1924).

When selecting a Load Cell (or any component), it is vital to refer to the specifications in the accompanying data sheets, as there are many factors that can adversely affect the measurements taken by the device. Factors to consider include Humidity (corrosion of wire), Lead-Wire Effects (Ω),

Temperature and Overloading. Temperature can alter the property of the material (rigidity, conductivity) and consequently result in altered sensitivity. Lead-Wire Effects refer to the length of the wire between the measuring device and the Load Cell, which in effect would add to the resistance, this can however be compensated for mathematically through software. When Overloading, the elasticity of the materials that make up the load cell could be permanently deformed, resulting in an inaccurate resistance per load produced.

Very good detailed research carried out on the type of load cell used in the assignment

3 DESIGN OF ASSIGNMENT REQUIREMENTS

3.1 8-BIT COMPARATOR

The first element to be designed was the 8-bit Comparator, of which there is a sim and FPGA version. The sim version was used for testing to ensure that the design would work in the final application, and the FPGA version is the same with the interactive Boolean variables replaced with inputs and the probes/indicators replaced with outputs.

The 8-bit comparator is built from two 4-bit comparators as found from the design found in research.

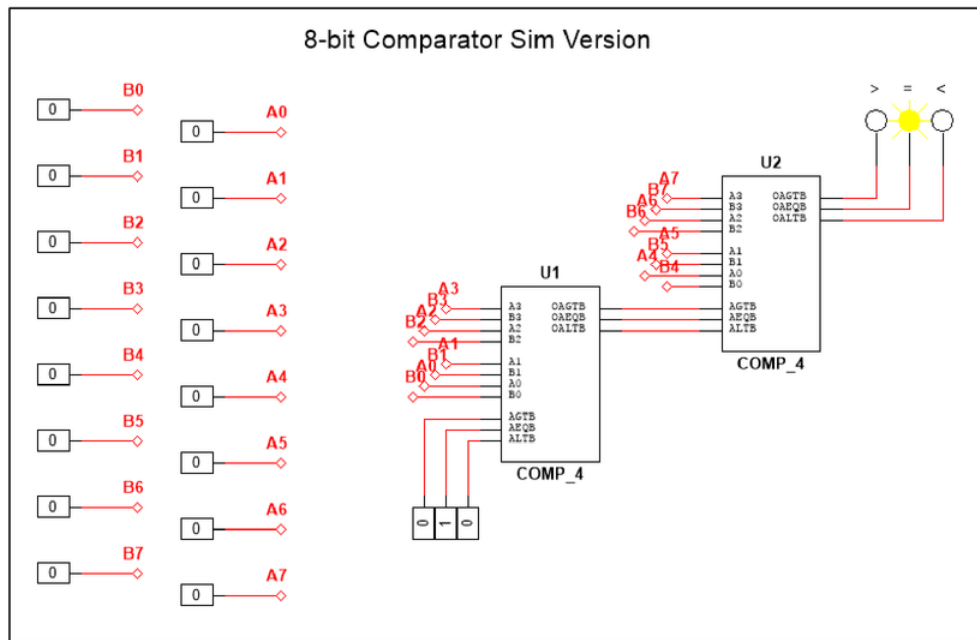


Figure 3.1: Simulation version of 8-bit Comparator

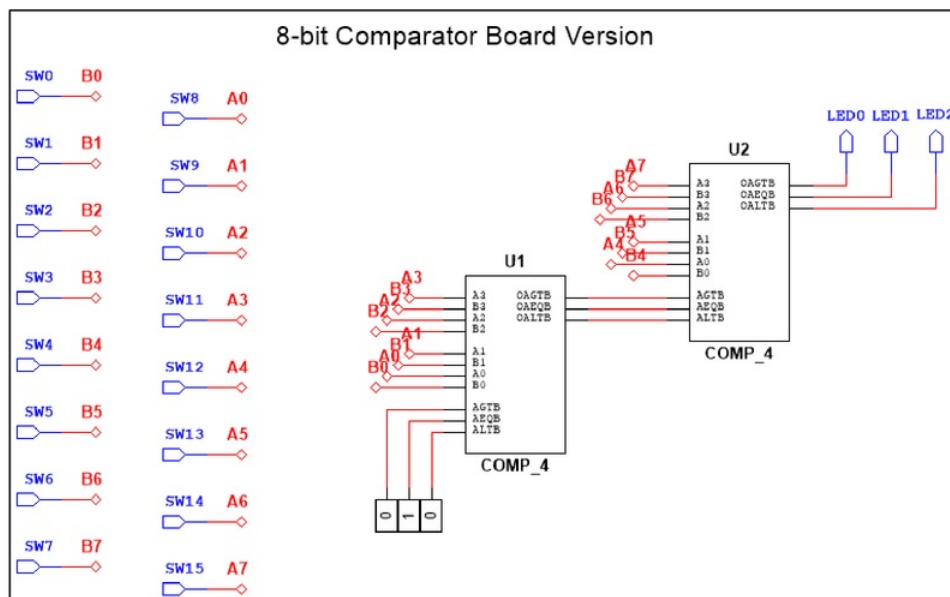


Figure 3.2: Downloadable version of FPGA 8-bit Comparator

The outputs for the Nexys 4 board used by the FPGA program are identified in a file called MultiSim.xdc. The file ties the ports, switches, and displays from the board to names that are readily more usable in programming.

```

1  ## This file is a general .xdc for the Nexys4 DDR Rev. C
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  ## Clock signal
7  set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports CLK100MHZ]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
8  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLK100MHZ];
9
10
11  ##Switches
12
13  set_property -dict { PACKAGE_PIN J15      IOSTANDARD LVCMOS33 } [get_ports SW0]; #IO_L24N_T3_R50_15 Sch=sw[0]
14  set_property -dict { PACKAGE_PIN L16      IOSTANDARD LVCMOS33 } [get_ports SW1]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
15  set_property -dict { PACKAGE_PIN M13      IOSTANDARD LVCMOS33 } [get_ports SW2]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
16  set_property -dict { PACKAGE_PIN R15      IOSTANDARD LVCMOS33 } [get_ports SW3]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
17  set_property -dict { PACKAGE_PIN R17      IOSTANDARD LVCMOS33 } [get_ports SW4]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
18  set_property -dict { PACKAGE_PIN T18      IOSTANDARD LVCMOS33 } [get_ports SW5]; #IO_L7N_T1_D10_14 Sch=sw[5]
19  set_property -dict { PACKAGE_PIN U18      IOSTANDARD LVCMOS33 } [get_ports SW6]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
20  set_property -dict { PACKAGE_PIN R13      IOSTANDARD LVCMOS33 } [get_ports SW7]; #IO_L5N_T0_D07_14 Sch=sw[7]
21  set_property -dict { PACKAGE_PIN T8       IOSTANDARD LVCMOS18 } [get_ports SW8]; #IO_L24N_T3_34 Sch=sw[8]
22  set_property -dict { PACKAGE_PIN U8       IOSTANDARD LVCMOS18 } [get_ports SW9]; #IO_25_34 Sch=sw[9]
23  set_property -dict { PACKAGE_PIN R16      IOSTANDARD LVCMOS33 } [get_ports SW10]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
24  set_property -dict { PACKAGE_PIN T13      IOSTANDARD LVCMOS33 } [get_ports SW11]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
25  set_property -dict { PACKAGE_PIN H6       IOSTANDARD LVCMOS33 } [get_ports SW12]; #IO_L24F_T3_35 Sch=sw[12]
26  set_property -dict { PACKAGE_PIN U12      IOSTANDARD LVCMOS33 } [get_ports SW13]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
27  set_property -dict { PACKAGE_PIN U11      IOSTANDARD LVCMOS33 } [get_ports SW14]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
28  set_property -dict { PACKAGE_PIN V10      IOSTANDARD LVCMOS33 } [get_ports SW15]; #IO_L21P_T3_DQS_14 Sch=sw[15]
29
30
31  ## LEDs
32
33  set_property -dict { PACKAGE_PIN H17      IOSTANDARD LVCMOS33 } [get_ports LED0]; #IO_L18P_T2_A24_15 Sch=led[0]
34  set_property -dict { PACKAGE_PIN K15      IOSTANDARD LVCMOS33 } [get_ports LED1]; #IO_L24F_T3_R51_15 Sch=led[1]
35  set_property -dict { PACKAGE_PIN J13      IOSTANDARD LVCMOS33 } [get_ports LED2]; #IO_L17N_T2_A25_15 Sch=led[2]
36  set_property -dict { PACKAGE_PIN N14      IOSTANDARD LVCMOS33 } [get_ports LED3]; #IO_L8P_T1_D11_14 Sch=led[3]
37  set_property -dict { PACKAGE_PIN R18      IOSTANDARD LVCMOS33 } [get_ports LED4]; #IO_L7P_T1_D09_14 Sch=led[4]
38  set_property -dict { PACKAGE_PIN V17      IOSTANDARD LVCMOS33 } [get_ports LED5]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
39  set_property -dict { PACKAGE_PIN H17      IOSTANDARD LVCMOS33 } [get_ports LED6]; #IO_L17P_T2_A14_D30_14 Sch=led[6]

```

Figure 3.3: xdc file

Very good detailed design shown for simulation and downloading to the FPGA

3.2 SIGNAL CONDITIONING CIRCUIT

To observe the characteristics of the Load Cell to be used, a Multisim model was built for simulation purposes, to determine the ratio grams to volts.

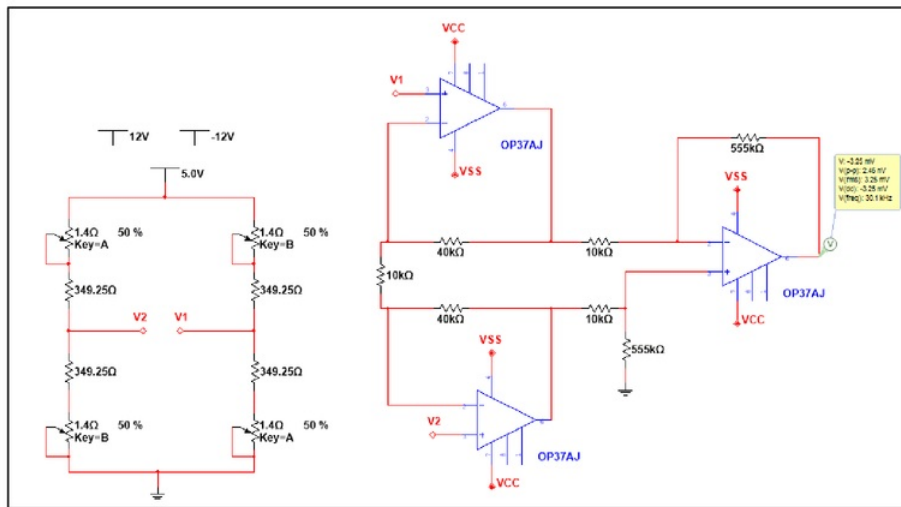


Figure 3.4: Multisim version of a Signal Conditioning Unit

As can be seen in Figure 4.1, A Wheatstone bridge as found through research in section 2.4, was built using potentiometers to simulate the strain gauges by alternating the resistors pairs' ohms by percentages of 1.4Ω. Linked to the simulated Load Cell is the Signal Conditioning Unit itself, which is made from three OpAmps and scales the signals to measurable levels that can be more accurately processed (2mV/V). The simulated circuit should produce a voltage range of 0V to 5V in proportion to 0g to 1000g. The gain in the design is determined through the following equation.

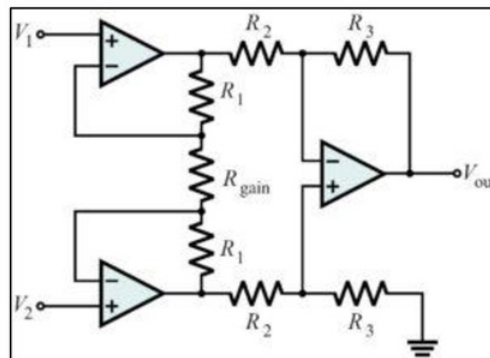


Figure 3.5: Amplifier Circuit

$$Gain = \left(1 + \left(\frac{2R_1}{R_{gain}} \right) \right) \frac{R_3}{R_2}$$

$$Gain = \left(1 + \left(\frac{2 \times 40k\Omega}{10k\Omega} \right) \right) \frac{555k\Omega}{10k\Omega} = 499.5$$

This design has a gain of nearly 500.

Good design shown but why is a gain of 500 required?

3.3 THE LABVIEW VI

The LabVIEW Virtual Instrument (vi) would go through many versions during development. The first version of the vi was developed to handle the Analogue to Digital conversion. An Analogue Channel was set up to read the measured voltage between 0V and 5V, which was then processed by the first Formula ($x_1/5 \times 256$) which would convert the analog voltage to a digital value and 8-bit binary number to be displayed on the Front Panel. The Calculated result would also be sent to Lines 0 to 7 on the Digital Port of the Elvis II prototyping board. On the front panel, there is a Stop pushbutton to end the while loop and a wait function that would set a half second wait period before running another loop.

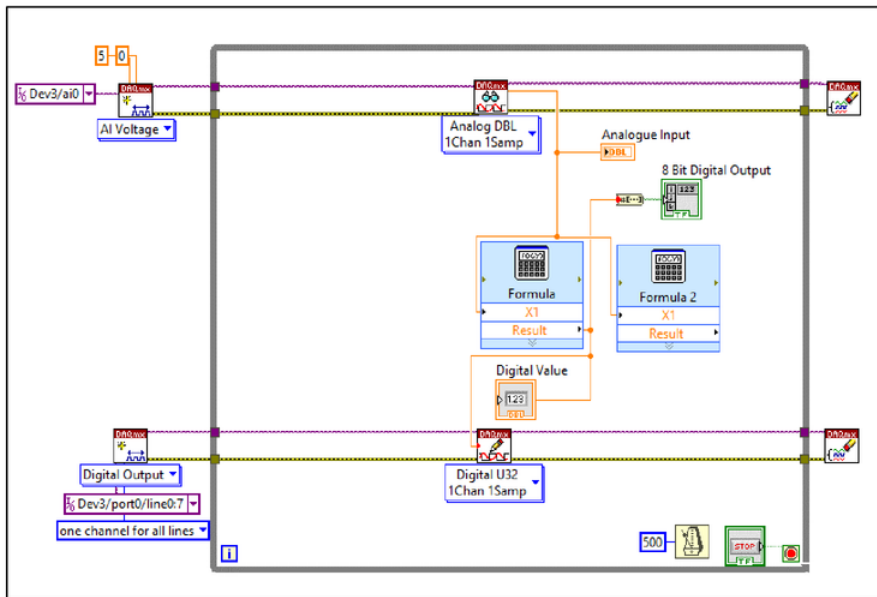


Figure 3.6: First Iteration (ADC)

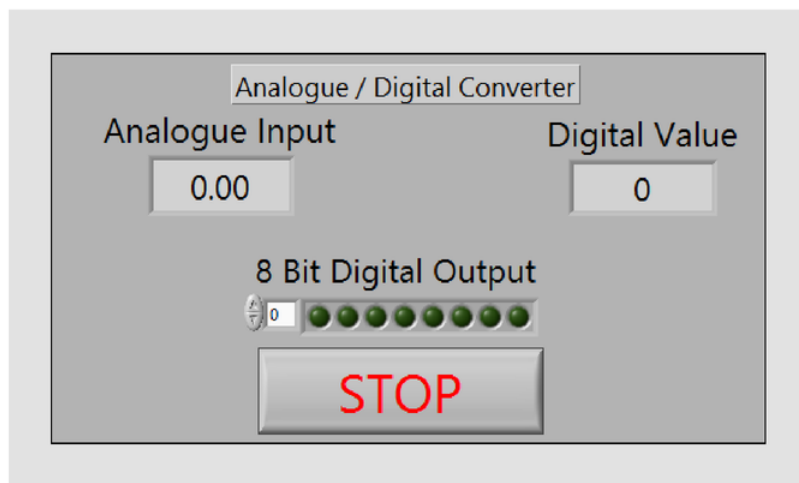


Figure 3.7: First Iteration (ADC) Front Panel

Building on the previous version, a 2nd Formula was added to change the measured input voltage into a measurement in grams ($x_1/5 \times 1000$), with the measurement displayed to the Front Panel. $(x_1/5)*1000$

A User-set control was added, which allowed a set-point to be added for comparison purposes. That set point was fed into the 3rd Formula ($x_1/1000 \times 256$) to be converted into a binary number (after rounding) and to be fed into the 4th Formula. The 4th Formula took the processed integer and multiplied it ($x_1 \times 256$) which had the effect of bumping along the integer, so that it could be added to the result of the 1st Formula and fed back to the Digital Output as a 16-bit number on lines 0 to 15. $(x_1/5)*256$

Returning the result from the 8-bit comparator via lines 21 to 23, the 3-bit binary number was displayed on the Front Panel. *Good*

The design also had clear functions added for successive loops, and error indicators added for fault finding.

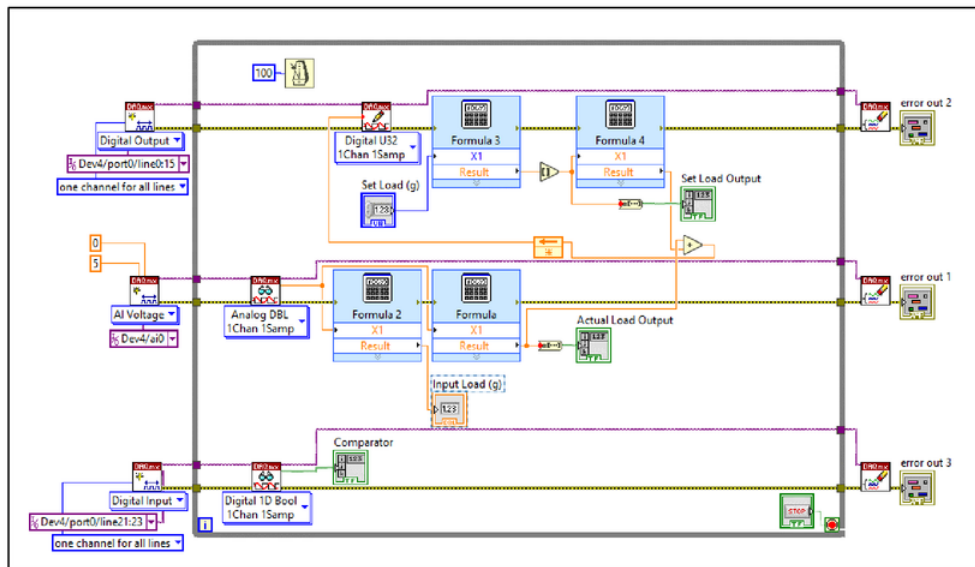
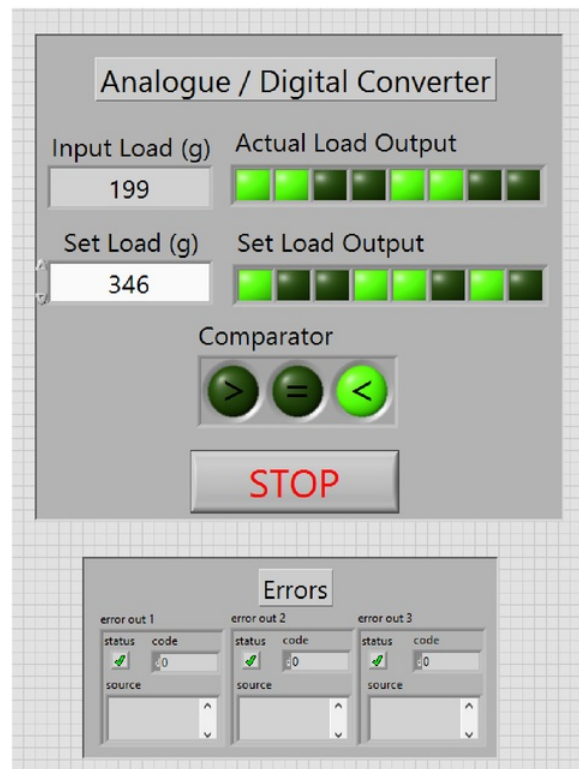


Figure 3.8: Addition of Digital Input, Comparator and Gram Conversion

Good LabVIEW design shown but formulae shown incorrect!



Good

Figure 3.9: Front Panel of vi

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

4.1 SIGNAL CONDITIONING CIRCUIT

The Load Cell used in the design is the LCKD-1KG (OMEGA Engineering inc., 2018) model, and from the specifications of the cell detailed in the table below, a simulated circuit was built as shown in Figure 4.1. To simulate a load pressing on the Load Cell, the A resistors have their resistance increased as the B resistors have their resistance values lowered.

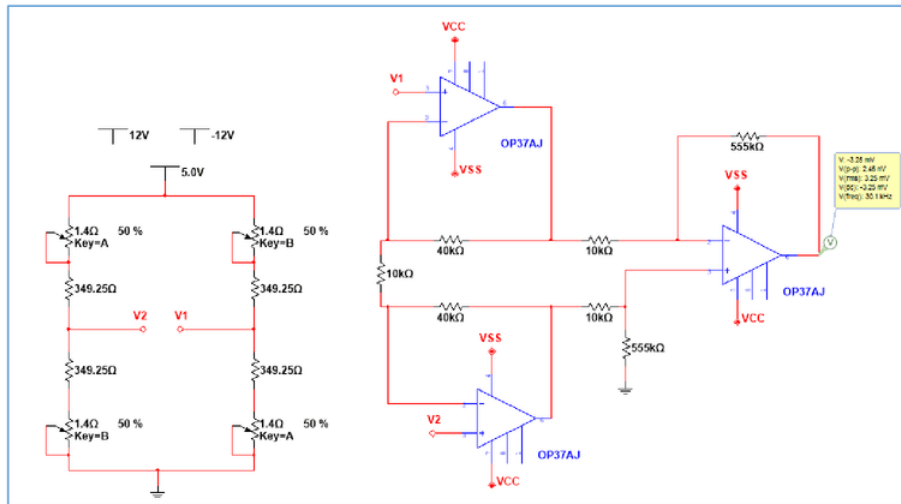


Figure 4.1: Instrumentation Amplifier

During the simulation, the resistance values were altered in 5% increments, with the Voltage level on Vout recorded in Table 4.1, and the Data was plotted in Figure 4.2.

Resistors A	Resistors B	Vout (V)	Load (g)
50	50	0	0
55	45	0.496	100
60	40	0.996	200
65	35	1.5	300
70	30	1.99	400
75	25	2.49	500
80	20	2.99	600
85	15	3.49	700
90	10	3.99	800
95	5	4.49	900
100	0	4.99	1000

Table 4.1: Vout according to Load

The plotted data showed a linearity between Output Voltage and applied Load. The linear line could be down to the 'ideal' characteristics of the components used, however the simulated results are acceptable.

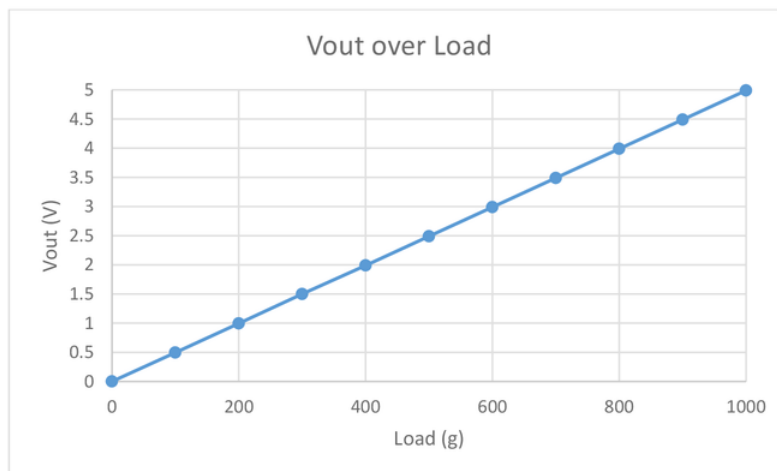


Figure 4.2: Vout over Load
Good, correct graphical analysis
produced

4.2 8-BIT COMPARATOR

In running the 8-bit Comparator, fixed values of 135 and 131 were chosen, along with their binary values. The interactive constants would be configured to the binary values to trigger the comparisons to be identified.

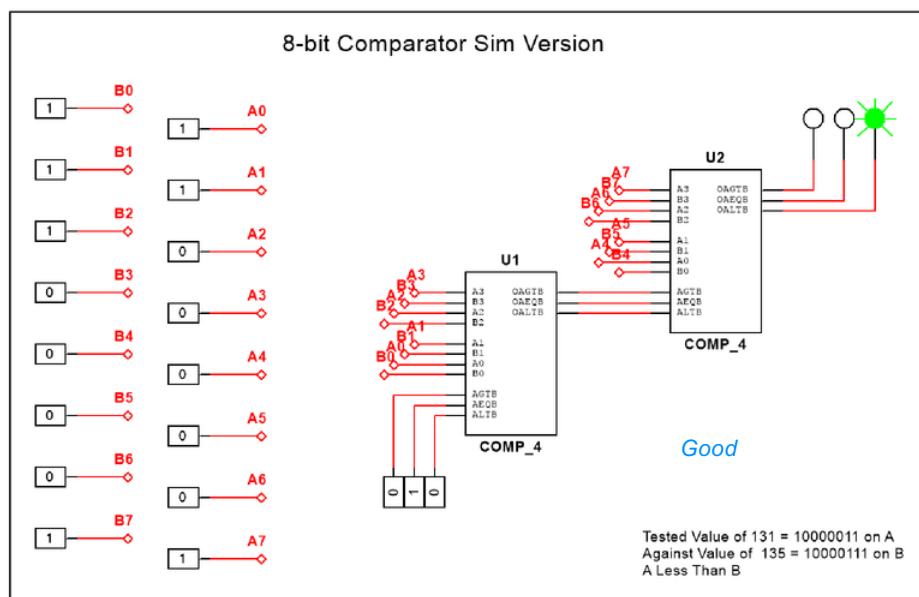


Figure 4.3: A Less Than B

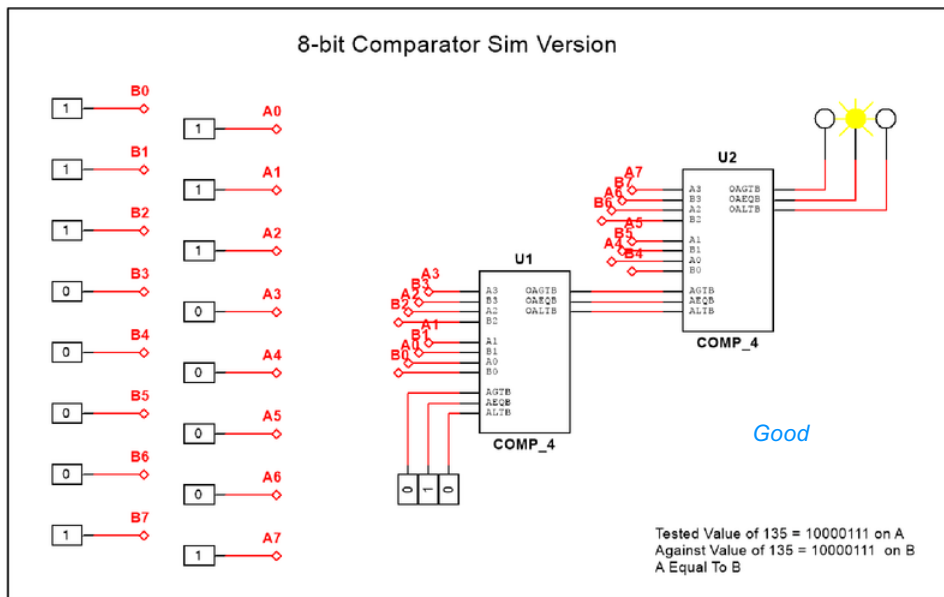


Figure 4.4: A Equal to B

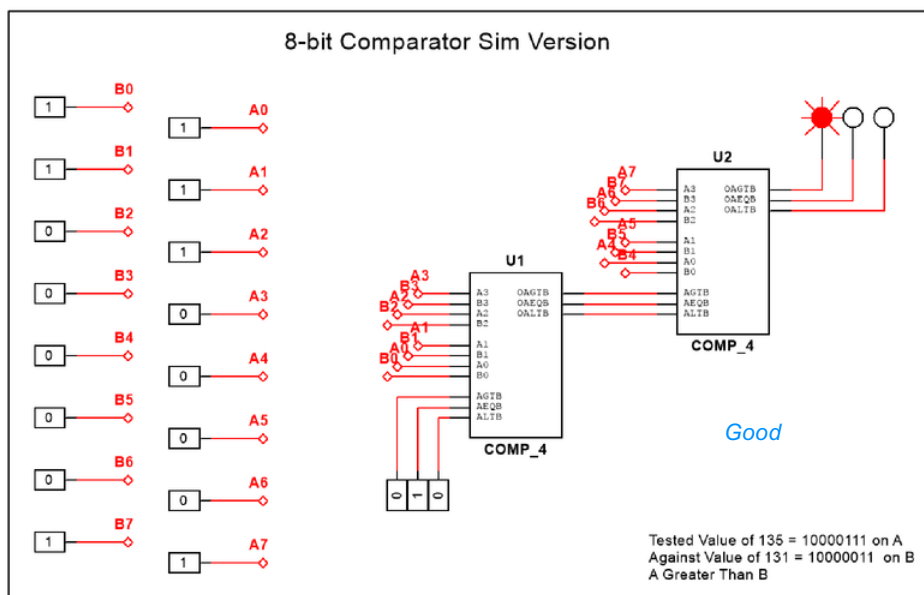


Figure 4.5: A Greater Than B

The results from the simulation proved to accurate enough to continue with running on the board.

Correct simulation analysis carried out

4.3 THE LABVIEW VI

The test involved for the ADC involved inputting a value for the Analogue Input and reading the calculated Digital Value and Binary Display.

Test 1 result is confirmed as $\frac{1.36}{5} \times 256 = 69.632$

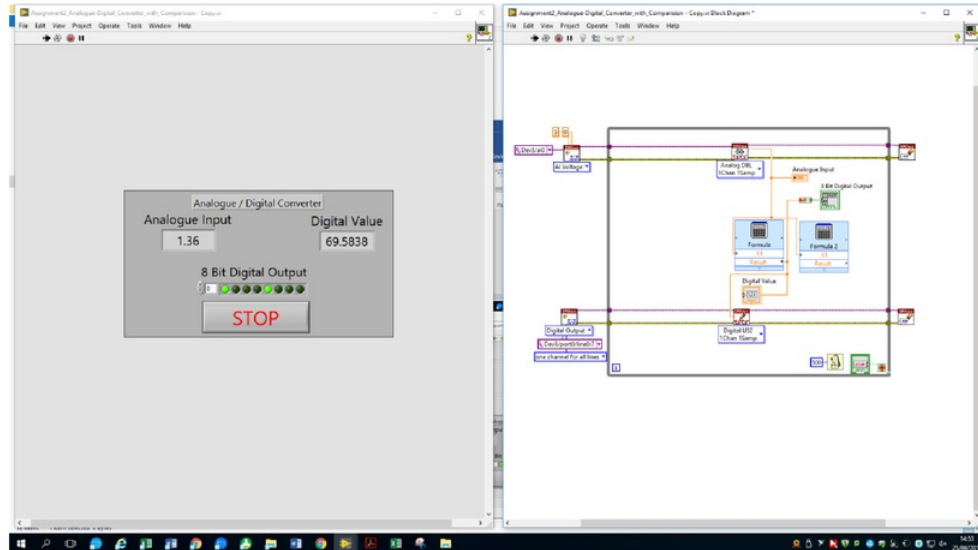


Figure 4.6: ADC Test 1

Test 2 result is confirmed as $\frac{2.81}{5} \times 256 = 143.872$

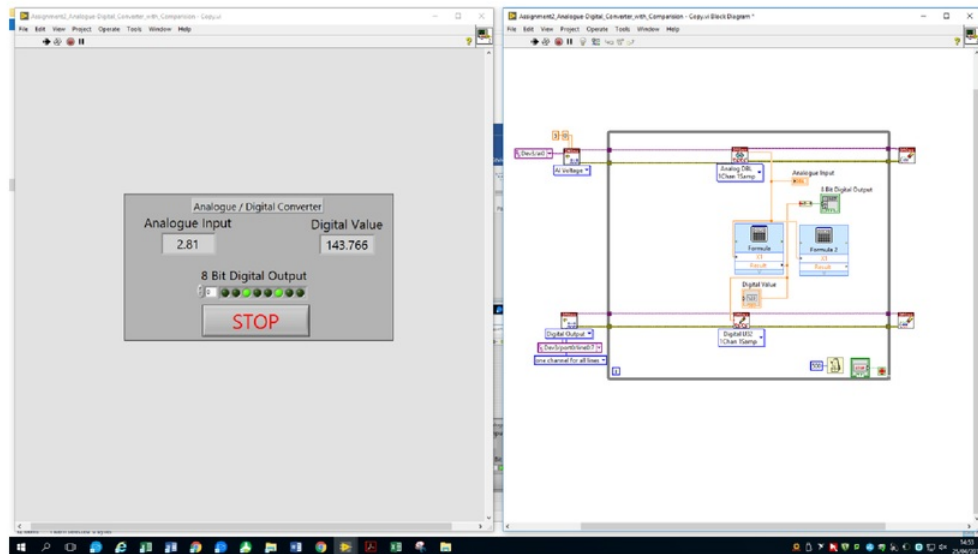


Figure 4.7: ADC Test 2

Finally, Test 3 result is confirmed as $\frac{3.73}{5} \times 256 = 190.976$

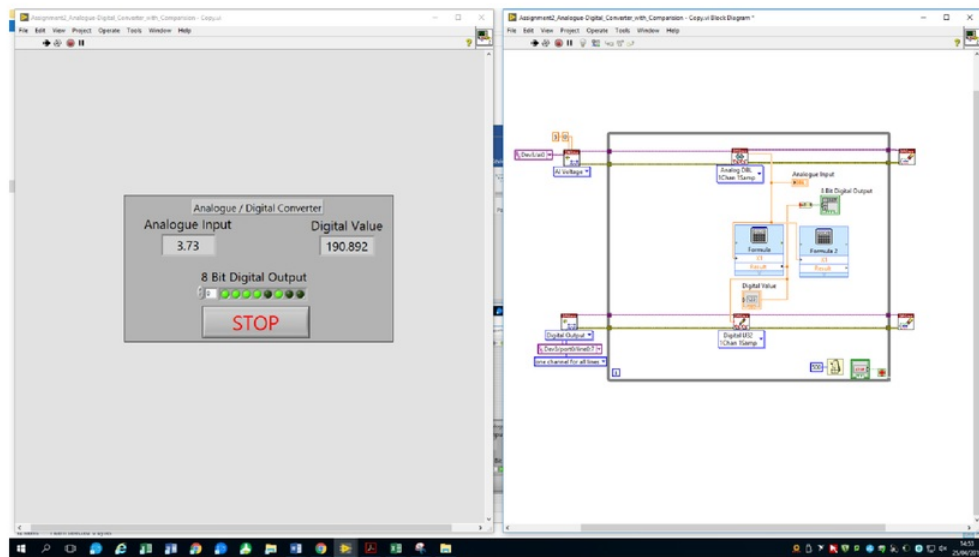


Figure 4.8: ADC Test 3

The binary result for all three tests we accurate representations of the digital value. While the manually calculated value differed slightly, the issue may have been with rounding of digits.

Very good understanding shown in 8 bit ADC conversion

5 PRACTICAL BUILD AND TEST VIA NI ELVIS BOARD

The Desktop running the LabVIEW vi was linked via USB to the NI ELVIS II developing board, which in turn had the Prototyping board slotted into it. The Nexus 4 FPGA Board was placed on top of the Prototyping Board's Breadboard component and connecting leads were connected between the Nexus 4's Ports as specified in the PLD file, and the appropriate Digital Input and Output pins as specified in the LabVIEW vi. The Nexus 4 then had the Final version of the generated .bit file downloaded for processing the 16-bit binary numbers coming from the Prototyping Board and sending back the 3-bit binary number. When running the LabVIEW vi, a fixed set-point was set and the virtual power supply within the National Instruments' Toolkit was set to values less than, equal to, and greater than and recorded.

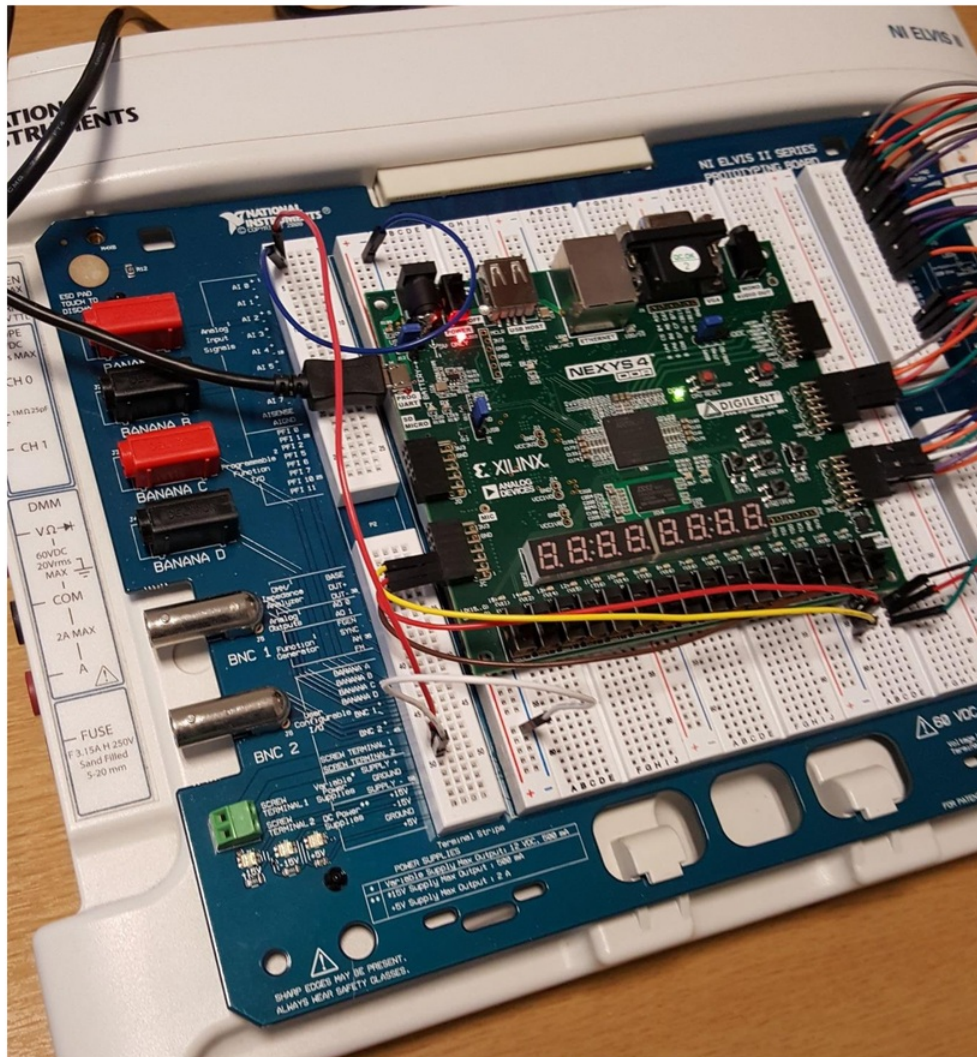


Figure 5.1: Physical Build

Good pictorial evidence shown and explained well

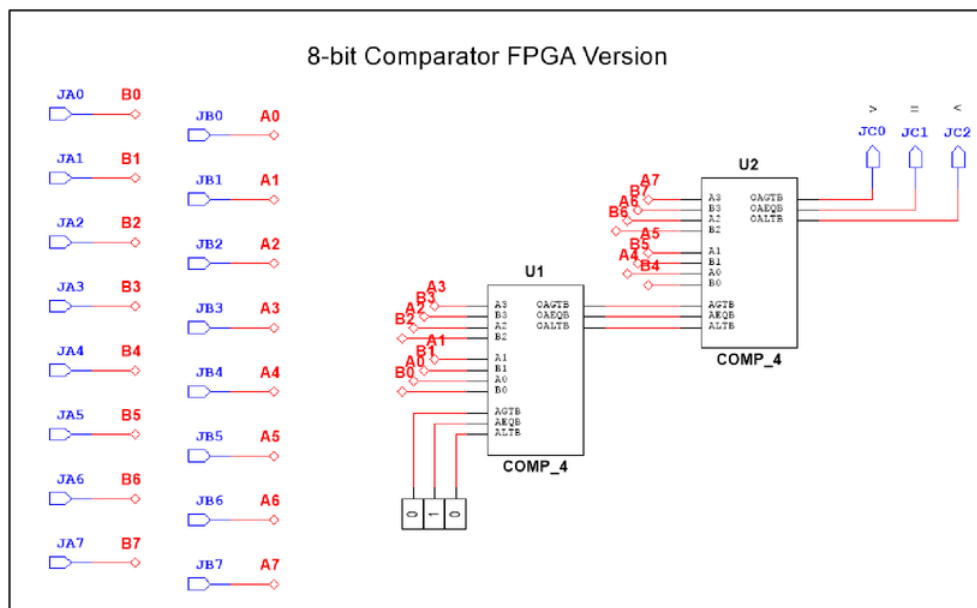


Figure 5.2: Amended 8-bit Comparator

How was this transferred to the FPGA?

6 ANALYSIS OF NI LABVIEW / FPGA RESULTS

6.1 FPGA

Using the PLD file from Figure 3.2, a standalone practical test was carried out using the same settings from Section 4.2 and monitoring the output via three LEDs on the Nexus 4 board. The same results were yielded

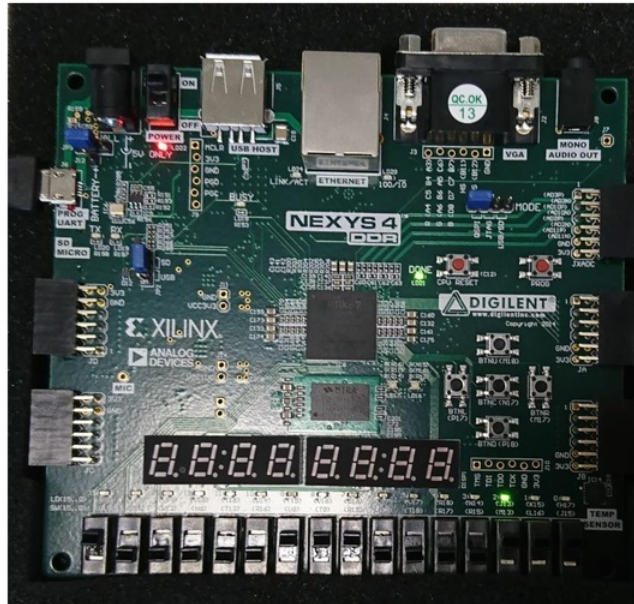


Figure 6.1: A Less Than B

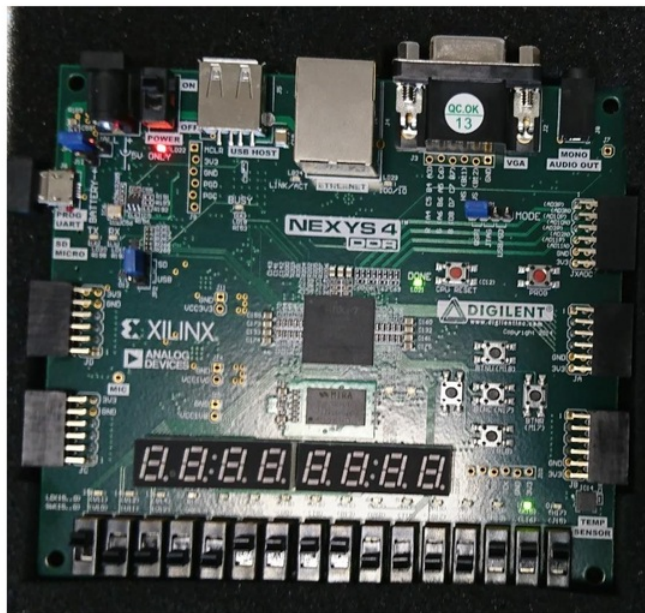


Figure 6.2: A Equal To B

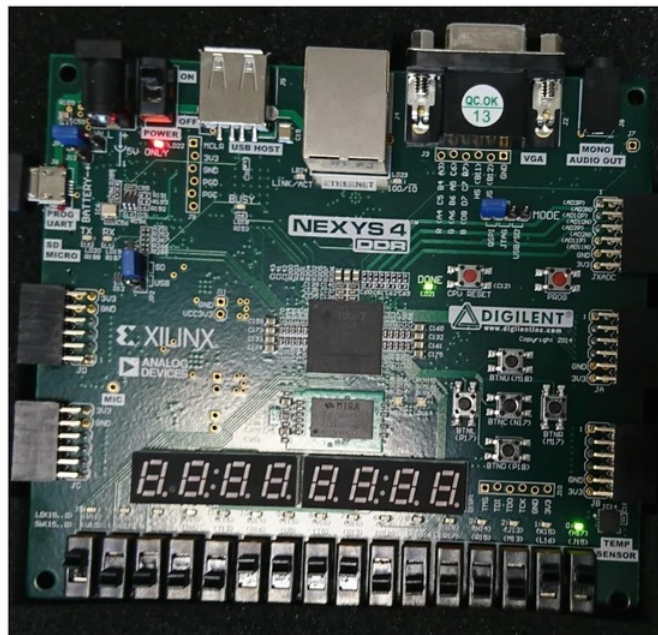


Figure 6.3: A Greater Than B

You should have explained that the switch setting were the same as the simulated switch settings in the simulation in MultiSim

6.2 LABVIEW

The first test for the Practical Build was to set a fixed set-point of 346 grams and adjusting the Load via the adjustable virtual power supply and observe the changes. To identify a 'Less Than' result, a load of 107 grams was chosen which yielded the following result.

$$\text{Input Load}(g) = \frac{533.35\text{mV}}{5V} \times 1000 = 106.67g$$



Figure 6.4: Less Than

An 'Equal To' result was yielded by setting the load as 327 grams.

$$\text{Input Load}(g) = \frac{1.74V}{5V} \times 1000 = 348g$$

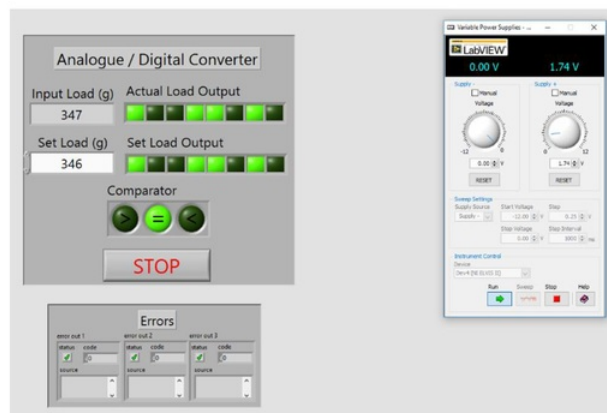


Figure 6.5: Equal To

The following Greater Than result was procured by setting the Load as 739 grams.

$$\text{Input Load}(g) = \frac{3.71V}{5V} \times 1000 = 742g$$



Figure 6.6: Greater Than

There is a slight discrepancy between the manually calculated result of the analogue input and LabVIEW's conversion of the analogue input, however the discrepancy is slight, possibly due to the supply's virtual setting (PC) differing from the physical supply of the ELVIS II Development board.

Excellent set of results and analysed very well

7 ADAPT DESIGN TO FIT NEW PURPOSES OR APPLICATIONS

It would be useful to be able to record the weight over time with a weighing system. This feature is simple enough to implement to the existing design. Individual results are recorded using a Write to Measurement File tool. The first file contains columns for time and the Measured Load. The second file contains the set-load. The final three files contain the comparative states, which are multiplied by 200 to give a discernible value as the unmultiplied values may appear effectively as a straight line on a graph.

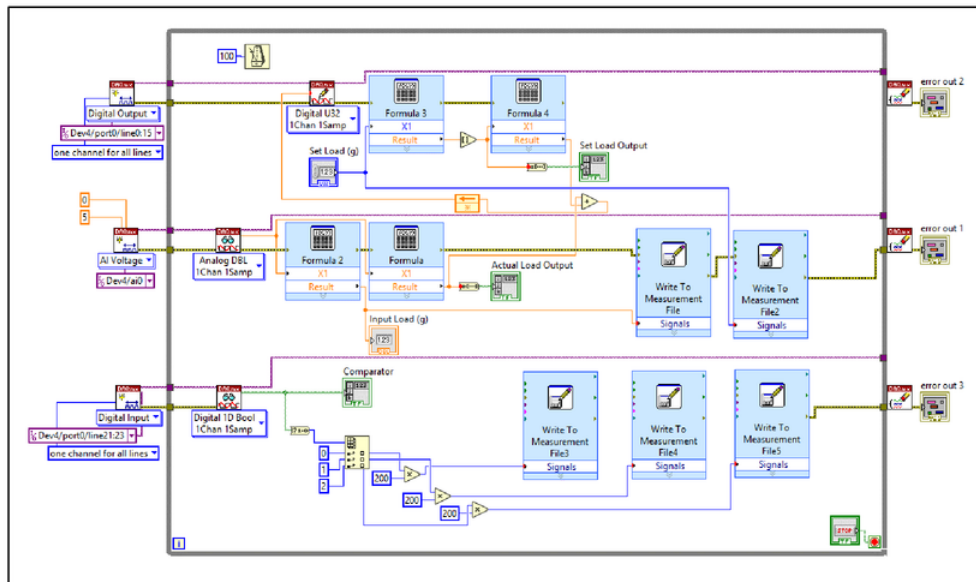


Figure 7.1: Data Logging Added

From running the vi, and manually adjusting the supply from 0V to 5V, the files would be populated with values over time. Once the process was completed, an Excel Spreadsheet was then populated with the data from the .lvm files.

Good design shown, but why the use of the 'index array' on the 3 inputs from the FPGA?

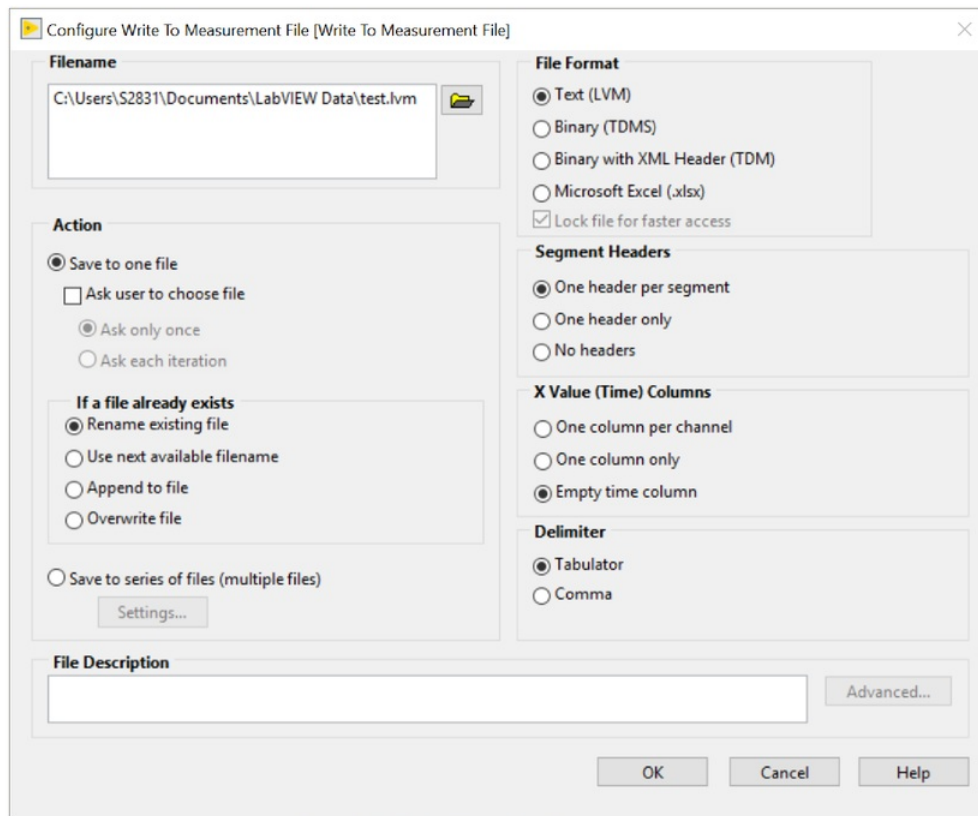


Figure 7.2: Configuration of the Data Logging tools

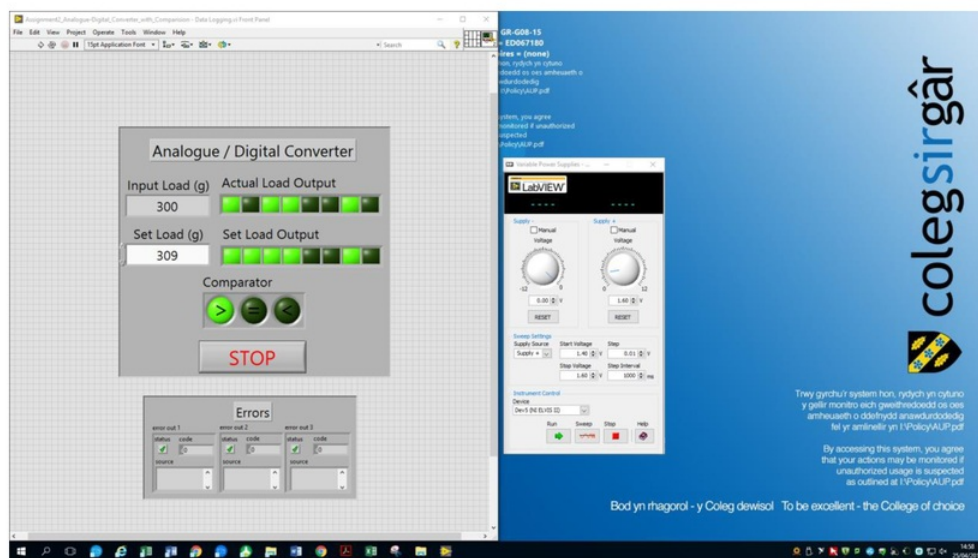


Figure 7.3: Supply adjustment

8 ANALYSIS OF DATA LOGGING EXCEL RESULTS

Using a set-point of 300, the virtual supply ramping from below, to above the set-point, the following chart was produced.

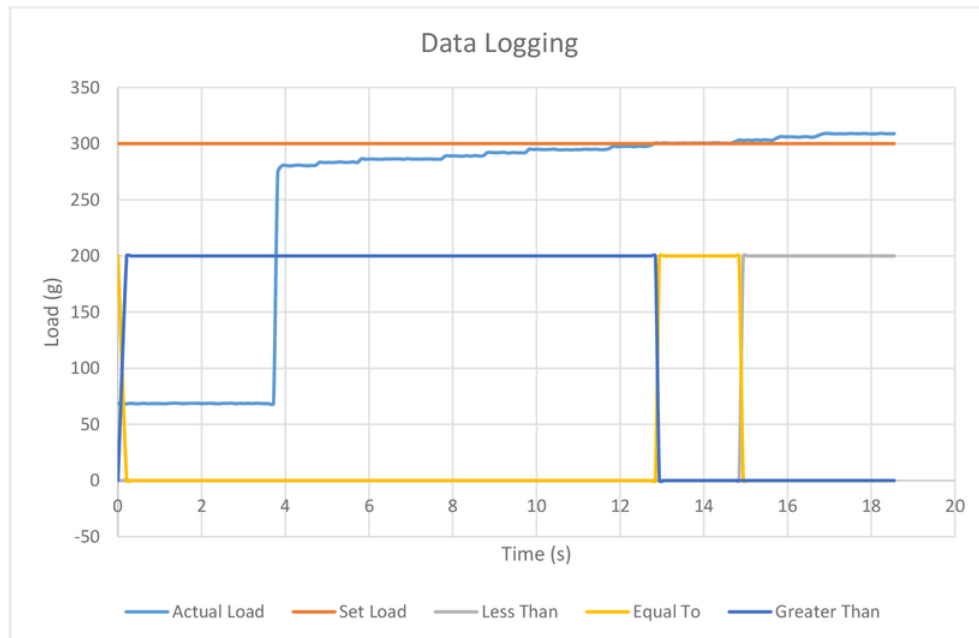


Figure 8.1: Output of LabVIEW vi

As identified in the legend at the bottom of the chart, the comparator correctly identifies when the set-point exceeded, matched or otherwise. There are particularly small transitioning gaps between states, which could be explained by cycle times of FPGA (Quick) and the PC (Very Quick) though it could be due to the wait loop in the LabVIEW vi. In any case, the Data Logging functionality is optimal.

Seems that there is confusion between the less than and the greater than values!

9 CONCLUSIONS

In a Practical application of the LabVIEW ADC and 8-bit comparator, the use of the FPGA could be considered overkill, as is the use of a PC for the purposes of monitoring a 1KG Load Cell. However, the principle of the design is solid enough to be applied to a larger scale weighing system such as a Lorry Load monitoring, or perhaps the integration of weighing into another LabVIEW vi application, where reports would need to be generated.

While an actual Load Cell was not used in the final application of the circuit, it was at least confirmed through the signal conditioning circuit that the relationship between measured weight and output voltage was linear, so that just a virtual power supply on its own was sufficient to simulate a load cell without the need for a lookup table to be added to the block diagram of the vi.

In regards to the application of the FPGA and PC for the design, an alternative method would be to use two 4-bit CMOS Comparator chips in tandem with a LattePanda Development Board, which is capable of running the Windows 10 Operating System with its 1.8GHz processor, 4GB of RAM and minimum of 32GB of flash memory, which is enough to run a LabVIEW vi, and store data on a networked drive (LattePanda, 2018).



Figure 9.1: LattePanda Board running Windows 10 with 7" screen

Good general conclusion given and an alternative solution noted.

10 CITATIONS

Applied Measurements Ltd., 2018. *What is Ratiometric Output?*. [Online]
Available at: <https://appmeas.co.uk/resources/pressure-measurement-notes/what-is-ratiometric-output/>
[Accessed 2 April 2018].

National Instruments Corporation, 2018. *NI-DAQmx Measurement Services Software - National Instruments*. [Online]
Available at: <https://www.ni.com/dataacquisition/nidaqmx.htm>
[Accessed 30 March 2018].

National Instruments Corporation, 2018. *What is NI-DAQmx? - National Instruments*. [Online]
Available at: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P8baSAC>
[Accessed 29 March 2018].

Nexperia, 1990. *4-bit magnitude comparator*. [Online]
Available at: https://assets.nexperia.com/documents/data-sheet/74HC_HCT85_CNV.pdf
[Accessed 19 April 2018].

OMEGA Engineering inc., 2018. *Subminiature Industrial Compression Load Cell*. [Online]
Available at: <https://www.omega.com/Pressure/pdf/LCKD.pdf>
[Accessed 15 March 2018].

Sparkfun, 2016. *Analog to Digital Conversion*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>
[Accessed 19 April 2018].

11 IMAGE SOURCES

Figure 2.15	https://assets.nexperia.com/documents/data-sheet/74HC_HCT85_CNV.pdf
Figure 2.16	https://assets.nexperia.com/documents/data-sheet/74HC_HCT85_CNV.pdf
Figure 2.17	https://www.omega.com/Pressure/pdf/LCKD.pdf
Figure 3.5	https://1oomzzme3s617r8yzt8qutjk-wpengine.netdna-ssl.com/wp-content/uploads/2016/10/Op-Amp_Instrumentation_Amplifier-300x217.jpg
Figure 9.1	http://www.lattepanda.com/wp-content/uploads/2016/08/533-900x600.jpg

Assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

69/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

Text Comment. Good, concise introduction....

PAGE 6

PAGE 7

PAGE 8

PAGE 9

Text Comment. You didn't add the 'write to measurement file'!

Text Comment. Good detail on DAQmx icons which is what the brief asked for...

PAGE 10

Text Comment. Good schematic diagram shown but how does it work?

PAGE 11

Text Comment. Good detail

Text Comment. Good

Text Comment. 1024 bits (0-1023)

Text Comment. Should be shown as $(V_{in} / V_{ref}) * \text{number of bits}$ In this case $(3.14 / 5) *$

PAGE 12

PAGE 13

Text Comment. Very good detailed research carried out on the type of load cell used in the assignment

PAGE 14

PAGE 15

Text Comment. Very good detailed design shown for simulation and downloading to the FPGA

PAGE 16

Text Comment. Good design shown but why is a gain of 500 required?

PAGE 17

PAGE 18

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

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

Text Comment. Good

Text Comment. Good LabVIEW design shown but formulae shown incorrect!

PAGE 19

Text Comment. Good

PAGE 20

PAGE 21

Text Comment. Good, correct graphical analysis produced

Text Comment. Good

PAGE 22

Text Comment. Good

Text Comment. Good

Text Comment. Correct simulation analysis carried out

PAGE 23

PAGE 24

Text Comment. Very good understanding shown in 8 bit ADC conversion

PAGE 25

Text Comment. Good pictorial evidence shown and explained well

PAGE 26

Text Comment. How was this transferred to the FPGA?

PAGE 27

PAGE 28

Text Comment. You should have explained that the switch setting were the same as the simulated switch settings in the simulation in MultiSim

PAGE 29

PAGE 30

Text Comment. Excellent set of results and analysed very well

PAGE 31

Text Comment. Good design shown, but why the use of the 'index array' on the 3 inputs from the FPGA?

PAGE 32

PAGE 33

Text Comment. Seems that there is confusion between the less than and the greater than values!

PAGE 34

Text Comment. Good general conclusion given and an alternative solution noted.

PAGE 35

GRADING FORM: CSG-RUBIC

EMYR DYER**69****RESEARCH-INFORMED LITERATURE**

Analogue / Digital Converters: Satisfactory research although it could have been much more detailed (2/5) Load Cells: Very good detailed research carried out on the type of load cell used in the assignment (4/5) Digital Magnitude Comparators: Good schematic diagram for 4 bit device shown but how does it work? (2/5) NI DAQmx in LabVIEW: Good detail on DAQmx icons which is what the brief asked for... (4/5)

12**KNOWLEDGE AND UNDERSTANDING**

Design of the signal conditioning circuit for LCKD-1Kg Load Cell: Good design shown but why is a gain of 500 required? (3/5) Design of LabVIEW 'vi': Good LabVIEW design shown but formulae shown incorrect! (6/10) Design of 4/8-bit Magnitude Comparator: Very good detailed design shown for simulation and downloading to the FPGA (8/10) Re-design to suit new applications: Good design shown, but why the use of the 'index array' on the 3 inputs from the FPGA? (3/5)

20

ANALYSIS

	Analysis of signal conditioning circuit: Good, correct graphical analysis produced (4/5) Analysis of MultiSim Results: Correct simulation analysis carried out and confirmed with FPGA practical results (4/5) Analysis of NI LabVIEW / FPGA results. Excellent set of results and analysed very well (8/10) Analysis of data logging files in EXCEL: Seems that there is confusion between the 'less than' and the 'greater than' values (3/5)	19
---	---	-----------

PRACTICAL APPLICATION AND DEP.

	Build interface between LabVIEW and the FPGA board via the NI ELVIS 11 board: Good pictorial evidence shown and explained well (7/10)	7
---	---	----------

SKILLS FOR PROFESSIONAL PRAC.

	Deployment of appropriate media: Very good use made of screen dumps and photographic evidence in design and analysis of assignment components (4/5) Report layout and readability: Report flowed well and covered all aspects of the assignment (4/5) Appropriate conclusion: Good general conclusion given and an alternative solution noted. (3/5)	11
---	--	-----------