

2017194

2017294

by Hefin Williams

Submission date: 03-Mar-2023 11:36PM (UTC+0000)

Submission ID: 198466760

File name: 16384_Hefin_Williams_2017294_181628_631275596.docx (3.46M)

Word count: 4027

Character count: 19128

Course: HND / BSc (Hons) Electrical and Electronic Engineering
Module: VIRTUAL INSTRUMENTATION & CONTROL SYSTEMS
Portfolio Title: NI LabVIEW Magnitude Comparator
Name: Hefin Williams
University number: 2017194
Date submitted: 03/03/2023
Word count: 4000

Table of Contents

Element 1:	5
Introduction:.....	5
Research:	5
Digital design methods using VHDL and IP integration:	5
6 <i>This is an error</i>	
NI DAQmx data acquisition used in LabVIEW:	6
Digital Magnitude Comparators:	7
Analogue to Digital Converters:	8
Load Cells (namely LCKD-1Kg):.....	8
Instrumentation and signal conditioning:.....	9
Design Requirements:	10
4- bit adder:	10
Half adder:	10
Full adder:	11
4 Bit Adder:	11
4-bit magnitude comparator:.....	13
Combination of the 4-bit adder and 4-bit magnitude comparator.....	14
Testing and Analysis:	14
FPGA test:	15
8-bitmagnitude comparator multisim simulation test:.....	16
8-Bit magnitude comparator test on FPGA board:.....	17
Element 2:	18
Introduction:.....	18
Design requirements:	19
Signal gain and Load cells:.....	19
Load cell amplifier:	19
10 bit ADC Prototype:.....	28
12 bit ADC:	29
8 Bit ADC:.....	30
Evaluation:.....	34
Element 3:	35
Testing:.....	36
Evaluation:.....	38
Bibliography	38

Figure 1 - text example of VHDL language	5
Figure 2 - IP design example.	6
Figure 3 - DAQmx.	7
Figure 4 - 1 bit digital magnitude comparitor.....	7
Figure 5 – Unipolar single-ended SAR ADC example.	8
Figure 6 - Data table for SAR ADC. (Curran, 2005)	8
Figure 7 - Hydraulic load cell diagram.....	9
Figure 8 - Amplifier example.	9
Figure 9 - 4-bit adder.....	10
Figure 10 - code for half adder.....	10
Figure 11 - Code for Full adder.....	11
Figure 12 - Order of the 4 bit adder.....	11
Figure 13 - code for a 4-bit adder.....	12
Figure 14 - code for the IP magnitude comparator.	13
Figure 15 - IP of 4-bit magnitude comparator.	13
Figure 16 - 4-bit magnitude comparator NI multisim.	14
Figure 18 - mag 4-bit comparator testing.	14
Figure 17 - Mag 4-bit testing results.....	14
Figure 19 – Less than.	15
Figure 20 - More than.....	15
Figure 21 - equal too.	15
Figure 22 - 8-bit mag comparator AEQB test.....	16
Figure 23 - 8-bit mag comparator AGTB test.....	16
Figure 24 - 8 Bit magnitude comparator FPGA AWQB test and evidence.....	17
Figure 25 - 8-bit mag comparator ALTB test.....	17
Figure 26- Unity gain amplifier.	19
Figure 27- LCKD-1Kg Load Cell.	19
Figure 28 - Load cell with instrument amplifier.....	20
Figure 29 - results at 50% load.	21
Figure 30 - result.....	21
Figure 31 - result at 55% load.....	21
Figure 32 - results.	21
Figure 33 - result at 65% load.....	22
Figure 34 - results.	22
Figure 35 - result at 70% load.....	22
Figure 36 – results.	23
Figure 37 - results.	23
Figure 38 - results at 75% load.	23
Figure 39 - results.	23
Figure 40 - results at 80% load.	24
Figure 41 - results.	24
Figure 42 - results at 85% load.	24
Figure 43 - results	24
Figure 44 - results.	25
Figure 45 - results at 90% load.	25
Figure 46 - results.	25
Figure 47 - results at 95% load.	25
Figure 48 - results at 100% loading.	26
Figure 49 - results.	26

Figure 50 - 10 Bit ADC block diagram.....	28
Figure 51 - Panel view of 10 bit ADC	28
Figure 52 - 10 bit formula block.....	29
Figure 53 – 10 bit Boolean value at 514 grams.	29
Figure 54 - 12 bit ADC when input is set to half of the maximum.	30
Figure 55 - 12 bit ADC panel view at maximum output.	30
Figure 56 - 8 Bit ADC panel view and block diagram.....	31
Figure 57 - formula block for 8 Bit ADC.	31
Figure 58 - 8 Bit ADC set to 2.5V.....	32
Figure 59 - 8 bit ADC set to 5V.	32
Figure 60 - Greater than.....	33
Figure 61 - less than.	33
Figure 62 - equal too.	33
Figure 63 - 8 Bit magnitude comparator multisim.	34
Figure 64 - Element 3 final block diagram program.	35
Figure 65 - file configuration.....	35
Table 1 - truth table of a 1 bit magnitude comparator.	7
Table 2 - experiment graphical results.	27
Table 3 – Experiment tabulated results.....	27
Table 4 – Element 3 graphical test results.	36
Table 5 – table of results for element 3.....	37

Detailed list of figures and tables

Element 1:

Introduction:

Intro has grammatical errors, however, the detail in the content demonstrates a good understanding of the requirements.

Within Element 1 there will be research carried out, designing and implementing combinational logic systems. The programming language used is VHDL through IP integration. The programming software used is Multisim and Vivado. On these software's a 4-bit adder with a magnitude comparator is designed, built, and tested. Also, within this Element a 8-bit magnitude comparator is designed, built, and tested. The testing aspect of this element will be carried out through the instruments available to use on the software's. Each of the tests will be tabulated and evidence of the test will be proven in the appendix with a screenshot of the test carried out.

Research:

Digital design methods using VHDL and IP integration:

A VHDL's meaning is (Very High Speed Integrated Circuit Hardware Descriptive Language). This language is often used in hardware descriptive languages. When programing chips such as a FPGA or ASIC chip, the preferred programing language will be to use VHDL and is approved by IEEE. The language has amazing capabilities where it can be interfaced between LabVIEW and FPGA control boards. The language is mostly used in text based models where the logic circuit generated will look like the figure shown below. The language can monitor complex text and circuits to simulate the system. (Arar, 2017)

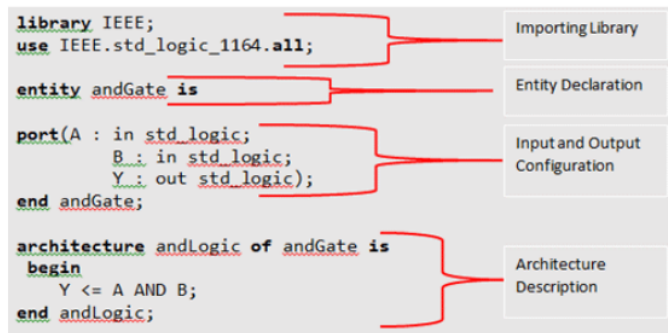


Figure 1 - text example of VHDL language

(Circuitdigest, 2022)

VHDL is a popular language as the structured text is widely used and can be applied for many various applications. A popular attraction to VHDL is its simulator facilities on Vivado, where any errors will be flagged up and can be tweaked. Intellectual Properties (IP) designing has been generated for numerous boards and one in particular would be the FPGA board. IP is a control block which contains electronic logic circuits. The block can hold circuits such as adders and comparators. IP simplifies the circuits very much as there is just a few connections necessary to use Vivado. Figure 2 below shows the simplicity of the connections where the sub circuits are within, specifically in this figure the left hand side chip is an 4 bit adder and the chip on the right is a 4 bit magnitude comparator.

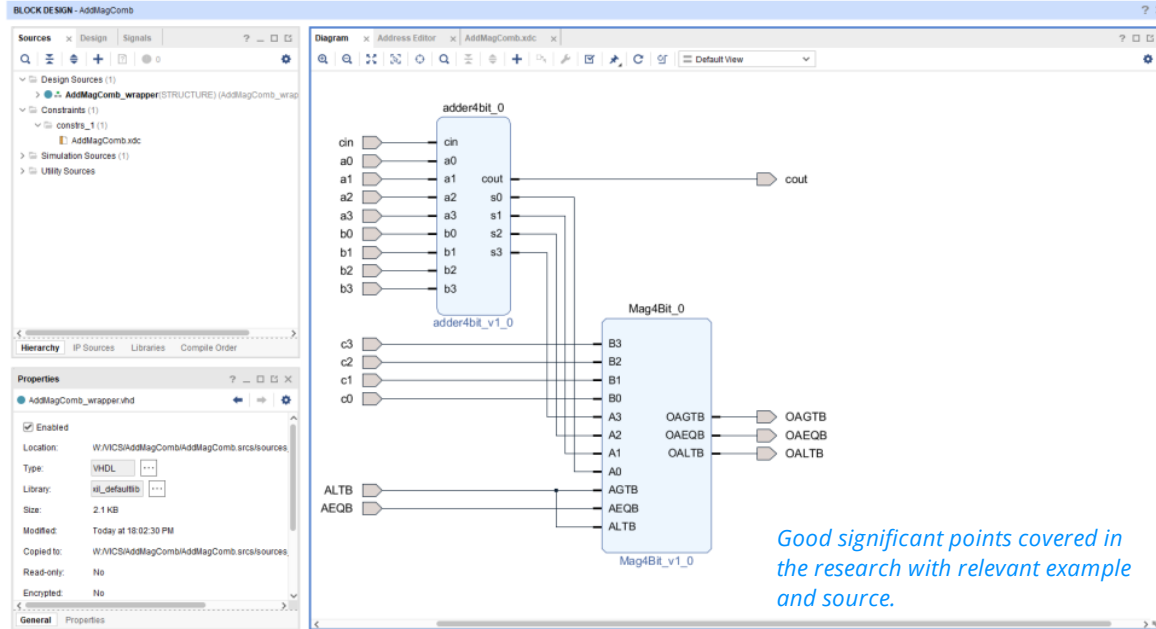


Figure 2 - IP design example.

NI DAQmx data acquisition used in LabVIEW:

NI DAQmx is a national instrument driver where it has control over the majority of the aspects within a DAQ system (Data Acquisition), it will also control the signal conditioning of either low or high level programming for a number of software's.

NI DAQmx was originally NI DAQ where it had its upgrade to who of which is NI DAQmx. This driver offers new architecture with various methods of programming interfaces. The newer version offers a more user-friendly boundary. This has resulted the driver to be easier to use and of which.

Limited research on this topic area and no supporting ref source

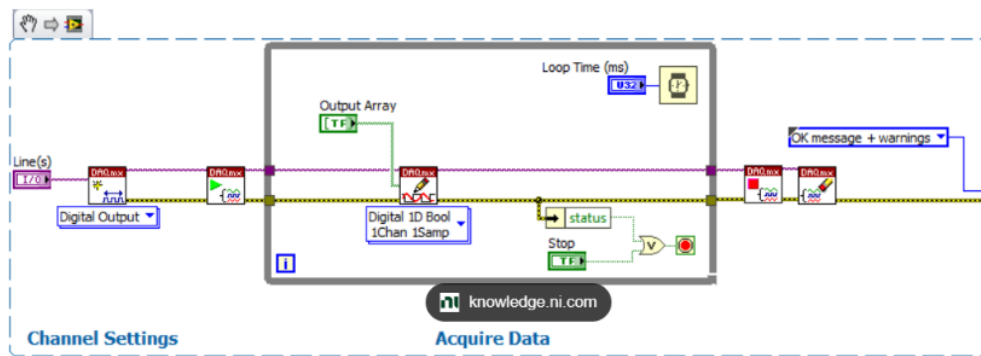


Figure 3 - DAQmx.

Diagram is not much use without an explanation of relevance. The function blocks of the DAQ should have been identified and discussed.

Digital Magnitude Comparators:

An digital magnitude comparator is built upon a number of logic circuits, the circuits included are a NOT, AND, and NOR gates. The output values are determined by the input values of the logic circuit, The image below shows one digital magnitude comparator where the results of the output values are shown on the right of the figure.

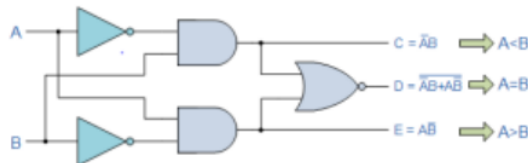


Figure 4 - 1 bit digital magnitude comparator.

The expansion of the digital mag comparator should be included. The image is relevant with supporting ref source. There are grammatical errors.

(Sasmita, 2015)

Below is the truth table of the above digital magnitude comparator. The comparator is used to monitor/read digital binary values. The circuit has three outputs, which are more than, equal too, and less than. The output will vary depending on which is true with the incoming signal.

Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Table 1 - truth table of a 1 bit magnitude comparator.

(Sasmita, 2015)

Analogue to Digital Converters:

ADC (Analogue to Digital Converters) are electronic devices which change analogue signals to digital signals. The devices can have many input elements including light, sound, temperature, and voltage. Where the signal that is generated can be inputted into controllers. ADC quality is compared by their resolution value. The resolution value varies the accuracy of the outputted value. Higher resolution results in higher quality output signals. In the ADC industry there are 5 main types of ADCs, which are Delta-Sigma, Successive Approximation, Pipeline, Flash, and Dual Slope. The most popular ADCs are the Delta-Sigma and the SAR. SAR has many good contributes where its resolution speed and its high sample rate of up to 12MHz and resolution of 20bits. (Curran, 2005)

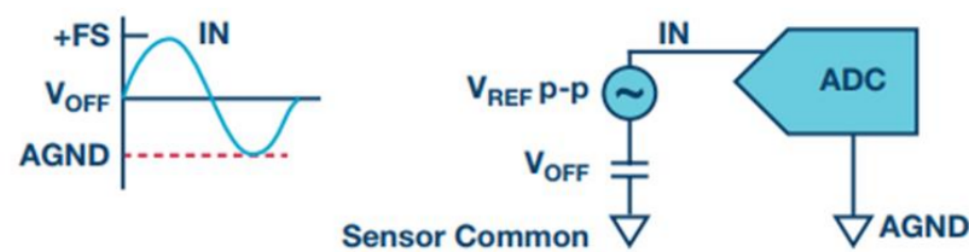


Figure 5 – Unipolar single-ended SAR ADC example.

(Curran, 2005)

Parameter	Conditions	A Grade			B Grade			Unit
		Min	Typ	Max	Min	Typ	Max	
Resolution		16			16			Bits
Analog Input								
Voltage Range	IN+ – (IN–)	0		VREF	0		VREF	V
Absolute Input Voltage	IN+, IN–	–0.1		VREF + 0.1	–0.1		VREF + 0.1	V
	IN+, IN–	–0.1		+0.1	–0.1		+0.1	V
Analog Input CMRR	F _{IN} = 250 kHz		60			60		dB

Figure 6 - Data table for SAR ADC. (Curran, 2005)

The figure above shows the different grades of which SAR ADC that are available with all relevant information if a ADC was needed to be selected these are the values u would compare to other ADCs.

Grammatical errors

Load Cells (namely LCKD-1Kg):

An load cell is a electrical device where it monitors weight through resistance, the higher the resistance the higher the load. Therefore the higher the input load the higher the output resistance. This makes the output proportional to its input. The various types of load cells are strain gauge, pneumatic, and hydraulic. The strain gauge is found to be the most popular of the three load cells

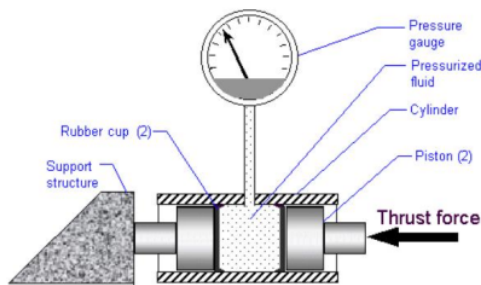


Figure 7 - Hydraulic load cell diagram.

There are significant points covered, however, there is no ref source.

Grammatical errors

Above is a hydraulic load cell showing how it internally works with the higher pressure of hydraulic oil flowing the higher the output of the load cell. This load cell would be used on a bailer to regulate the maximum applied pressure to a bale, where if the hydraulic ram has met a certain pressure which the load cell is monitoring the pressure it will indicate that the bale is ready or it can send the ram up and eject the bale if the bailer is automatic.

Instrumentation and signal conditioning:

Signal conditioning come in various forms. Where its often signals need to be scaled before they can become digital signals, to do this the signal is manipulated to prepare it for the next stage of its process of converting the signal from analogue to digital. Signal conditioning is a process where various instruments are used to provide an accurate measurement where a conditioner will condition the signal, the signal will then be usable. This instrument amplifies the signal so its thereby usable.



Figure 8 - Amplifier example.

Relevant points included in the content.

Relevant image example with explanation.

(Niazi, 2021)

The above figure shows and amplifier signal conditioner, the specific example is a siemens amplifier. This amplifier can be used to amplify a signal from a Current Transformer where the signal may need increasing before being fed into a PLC.

Design Requirements:

This task in element 1 requires designing a 4-bit adder through the use of VHDL within Vivado program. Designing the 4-bit adder is diagnosed and separated into sections. To build a 4-bit adder a full adder needs to be created alongside with half adders. Within a 4-bit adder there are 4 full adders alongside with 2 half adders.

4- bit adder:

Good points covered with the correct block diagram needed for the VHDL design.

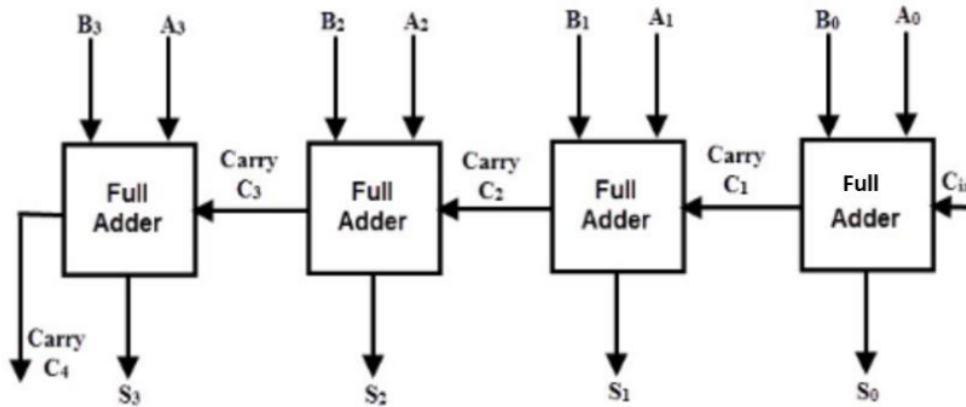


Figure 9 - 4-bit adder.

Half adder:

```
Project Summary x half_adder.vhd x
E:/Vivado Projects/half_adder/half_adder.srscs/sources_1/new/half_adder.vhd

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.all;
3 use IEEE.STD_LOGIC_unsigned.all;
4
5 entity halfadder is
6 port (a,b : in std_logic;
7       s,c : out std_logic);
8 end halfadder;
9
10 architecture Behavioral of halfadder is
11
12 begin
13   s <= a xor b;
14   c <= a and b;
15
16 end Behavioral;
```

Figure 10 - code for half adder.

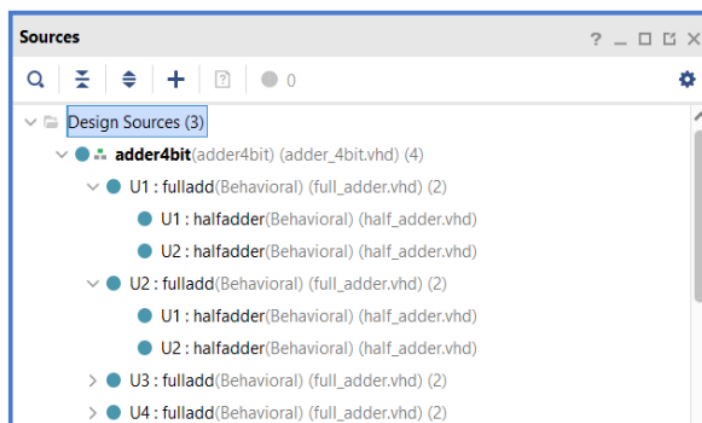
Full adder:

```
full_adder.vhd
E:/Vivado Projects/full_adder/full_adder.srscs/sources_1/new/full_adder.vhd

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity fulladd is
4  port(
5  a : in STD_LOGIC;
6  b : in STD_LOGIC;
7  cin : in STD_LOGIC;
8  cout : out STD_LOGIC;
9  s : out STD_LOGIC
10 );
11 end fulladd;
12 architecture Behavioral of fulladd is
13 component halfadder
14 port (
15 a : in STD_LOGIC;
16 b : in STD_LOGIC;
17 c : out STD_LOGIC;
18 s : out STD_LOGIC
19 );
20 end component;
21
22 signal c1 : STD_LOGIC;
23 signal c2 : STD_LOGIC;
24 signal s1 : STD_LOGIC;
25 begin
26 U1 : halfadder
27 port map(
28 a => a, b => b, c => c1, s => s1);
29 U2 : halfadder
30 port map(
31 a => s1, b => cin, c => c2, s => s);
32 cout <= c2 or c1;
33 end Behavioral;
34
```

Figure 11 - Code for Full adder.

4 Bit Adder:



Good evidence of hierarchical design requirements

Figure 12 - Order of the 4 bit adder.

```

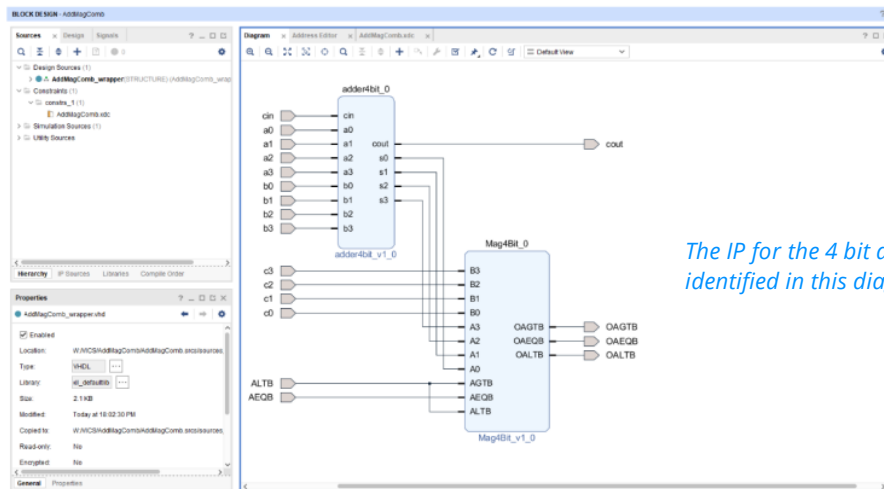
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity adder4bit is
4  port(
5  cin : in STD_LOGIC;
6  a0 : in STD_LOGIC;
7  a1 : in STD_LOGIC;
8  a2 : in STD_LOGIC;
9  a3 : in STD_LOGIC;
10
11  b0 : in STD_LOGIC;
12  b1 : in STD_LOGIC;
13  b2 : in STD_LOGIC;
14  b3 : in STD_LOGIC;
15
16  cout : out STD_LOGIC;
17  s0 : out STD_LOGIC;
18  s1 : out STD_LOGIC;
19  s2 : out STD_LOGIC;
20  s3 : out STD_LOGIC
21  );
22  end adder4bit;
23
24  architecture adder4bit of adder4bit is
25  component fulladd
26  port (
27  a : in STD_LOGIC;
28  b : in STD_LOGIC;
29  cin : in STD_LOGIC;
30  cout : out STD_LOGIC;
31  s : out STD_LOGIC
32  );
33  end component;
34  signal c1 : STD_LOGIC;
35  signal c2 : STD_LOGIC;
36  signal c3 : STD_LOGIC;
37  begin
38  U1 : fulladd
39  port map(
40  a => a0, b => b0, cin => cin, cout => c1,
41  s => s0);
42  U2 : fulladd
43  port map(
44  a => a1, b => b1, cin => c1, cout => c2,
45  s => s1);
46  U3 : fulladd
47  port map(
48  a => a2, b => b2, cin => c2, cout => c3,
49  s => s2);
50  U4 : fulladd
51  port map(
52  a => a3, b => b3, cin => c3, cout => cout,
53  s => s3);
54  end adder4bit;

```

4 bit adder code needs some explanation

Figure 13 - code for a 4-bit adder.

4-bit magnitude comparator:



The IP for the 4 bit adder should be identified in this diagram

Figure 15 - IP of 4-bit magnitude comparator.

```

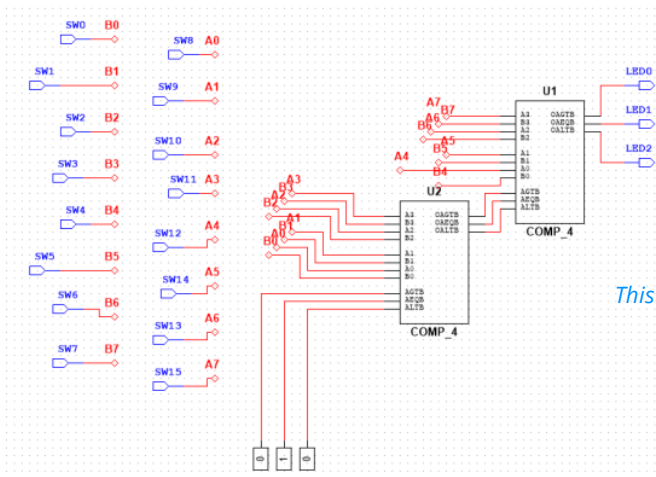
25 architecture BEHAVIORAL of COMP_4_NI is
26 begin
27 PROCESS (A0, B0, A1, A2, A3, B1, B2, B3, AGTB, AEQB, ALTB)
28 BEGIN
29   if A3>B3 then OAGTB<='1';
30   OAEQB<='0';
31   OALTB<='0';
32   elsif A3<B3 then OAGTB<='0';
33   OAEQB<='0';
34   OALTB<='1';
35   elsif A3=B3 then
36     if A2>B2 then OAGTB<='1';
37     OAEQB<='0';
38     OALTB<='0';
39     elsif A2<B2 then OAGTB<='0';
40     OAEQB<='0';
41     OALTB<='1';
42     elsif A2=B2 then
43       if A1>B1 then OAGTB<='1';
44       OAEQB<='0';
45       OALTB<='0';
46       elsif A1<B1 then OAGTB<='0';
47       OAEQB<='0';
48       OALTB<='1';
49       elsif A1=B1 then
50         if A0>B0 then OAGTB<='1';
51         OAEQB<='0';
52         OALTB<='0';
53         elsif A0<B0 then OAGTB<='0';
54         OAEQB<='0';
55         OALTB<='1';
56         elsif A0=B0 then
57           if AGTB='1' and AEQB='0' and ALTB='0' then OAGTB<='1';
58           OAEQB<='0';
59           OALTB<='0';
60           elsif ALTB='1' and AEQB='0' and AGTB='1' then OAGTB<='0';
61           OAEQB<='0';
62           OALTB<='0';
63           elsif AEQB='1' and AGTB='0' and ALTB='0' then OAGTB<='0';
64           OAEQB<='1';

```

The VHDL here is not needed, as the code is auto generated from the Msim design

Figure 14 - code for the IP magnitude comparator.

Combination of the 4-bit adder and 4-bit magnitude comparator:

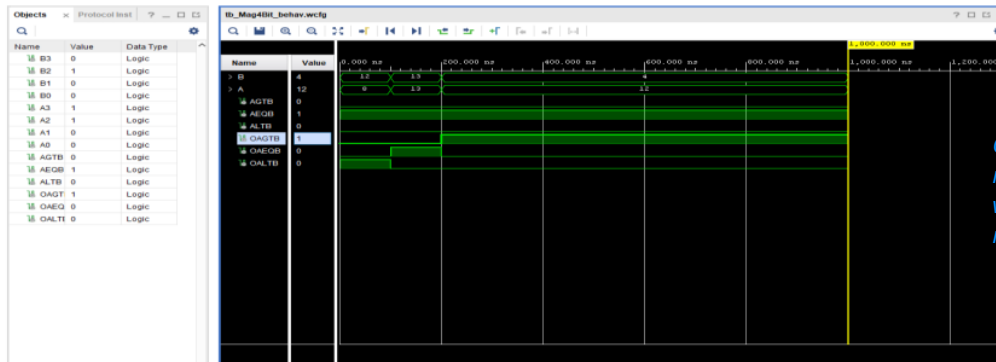


This image is incorrect

Figure 16 - 4-bit magnitude comparator NI multisim.

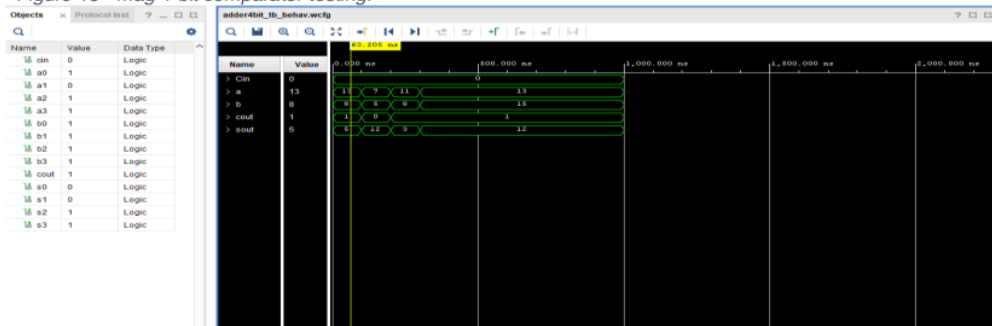
Testing and Analysis:

The figure below shows a Mag 4 bit comparator LESS THAN / GREATER THAN EQUAL TO



Good test results using virtual bus method

Figure 18 - mag 4-bit comparator testing.



Test results are good

Figure 17 - Mag 4-bit testing results.

This is the test result for the 4 it adder

The figure below shows a Adding numbers greater than 16 then c out sums to 1 and s out carries the difference more than 16. *Grammatical errors*

This needs a heading

FPGA test:

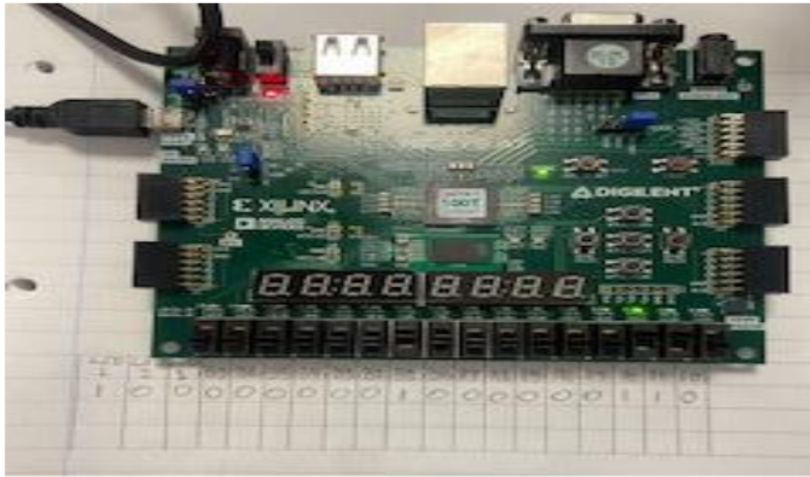


Figure 19 – Less than.



Figure 21 - equal too.



Figure 20 - More than.

These test results need to be explained .

8-bit magnitude comparator multisim simulation test:

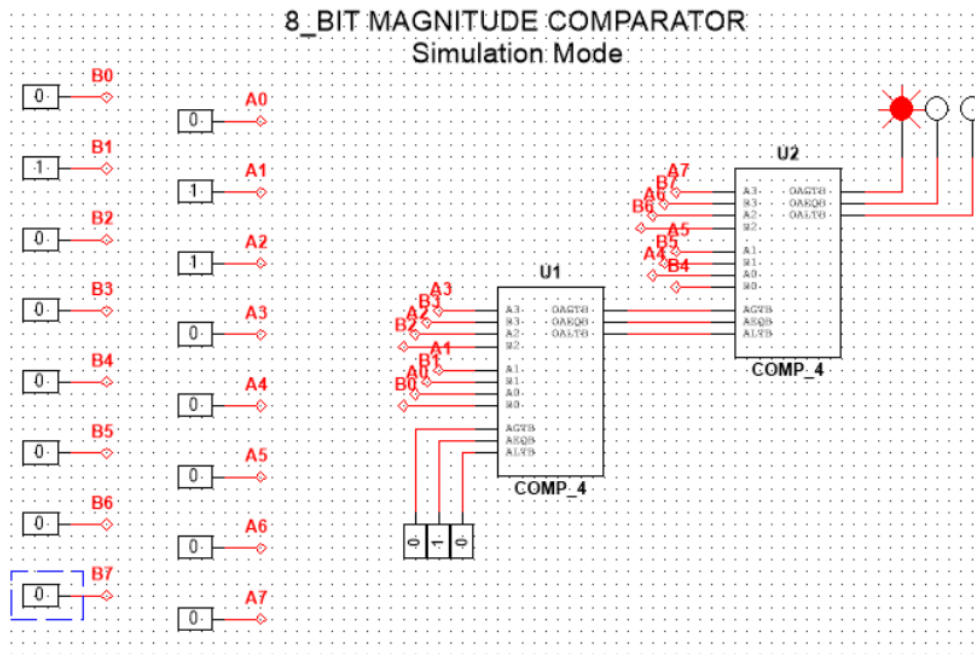


Figure 23 - 8-bit mag comparator AGTB test.

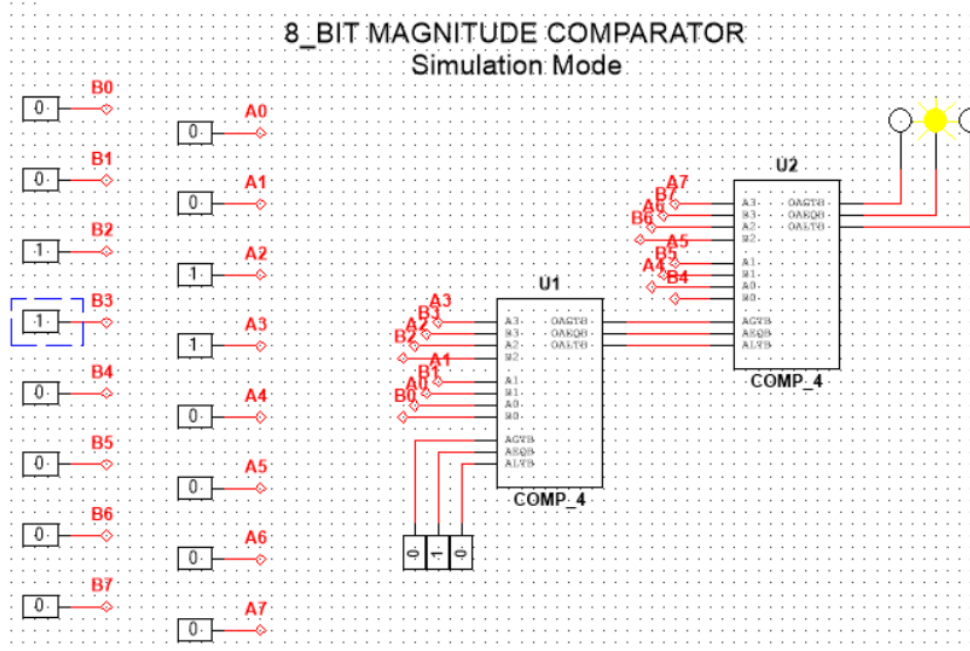


Figure 22 - 8-bit mag comparator AEQB test.

Element 2:

Introduction:

The report on element two focuses the use of LabVIEW. LabVIEW is a very popular software due to its vast capabilities. LabVIEW is an ideal software for a beginner or an experienced programmer, the software is relatively easy to use and has a wizard option if any help is required in many aspects. Element 2 is a program that will monitor weight specifically 1000gms. A load cell will give an output of 2mv/v per volt excitation over 1000gms. A testing and signalling unit will be built within multisim which will give an output of 0-5V over a range of 1000gms. Therefore, when the load cell is applied with full capacity of 1000gms the multisim unit will show a reading of 5 volts.

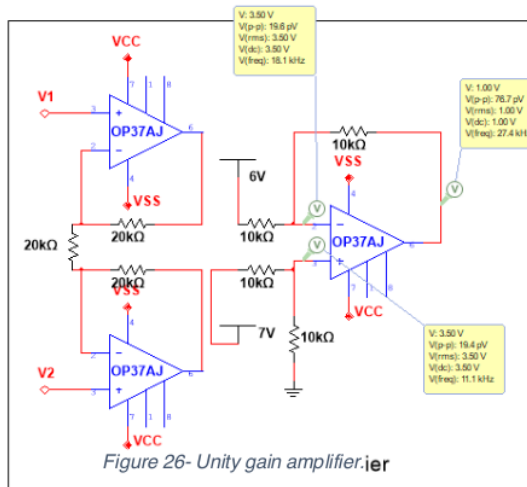
A LabVIEW program will be designed and built which will make paths with the FPGA board containing the 8-bit magnitude comparator. The NI ELVIS II board will show an output of greater than, equal to, and less than. Comparison of the set value and the actual value. Two signal coms are transferred in this report of the portfolio, one is to certify that an analogue input from LabVIEW is feeding signals into an LCKD load cell which convert the 0-5V signal into a 8 bit digital where it will then be transferred into an 8 bit digit signal and output the FPGS board. Signal two is the set value where the operator will set the value and the FPGS board will convert it into an 8 bit digital signal where it will range from 0-1000gms.

The intro indicates an understanding of requirements

Design requirements:

Signal gain and Load cells:

Figure 26 Below shows the circuit of an instrument amplifier at unity gain. Unity gain represents the input being equal to the output. For an amplifier to require unity gain the output magnitude must be matching with the input magnitude.



Good result

Load cell amplifier:

Figure 26 shows the amplifier used for the element 2 report. The amplifier is an LCKD-1Kg Load Cell. Figure 27 below shows an example of the load cell amplifier used in this portfolio. The specific load cell amplifier shown below is an example of a 1kg load cell, there are many different load cells where they can read many different weights and at many different accuracy. The load cell requires a flat and smooth surface for an accurate response. The load cell calculates the weight by using resistance, the higher the resistance the heavier the object the higher the resistance. The input of this load cell is 5V dc with an nominal output of 2mv/V. The unloaded resistance value is 350Ω.

Parameters are identified



Figure 27- LCKD-1Kg Load Cell.

Figure 28 shows the multisim generated to simulate the load cell amplifier. In order to set values and change the output values the variable resistors need changing. To have an accurate response the top left resistor is to be changed along with the bottom right resistor,

and vice versa for bottom left and top right. By doing this the output voltage ranging from 0-5V will fluctuate as the resistor values change.

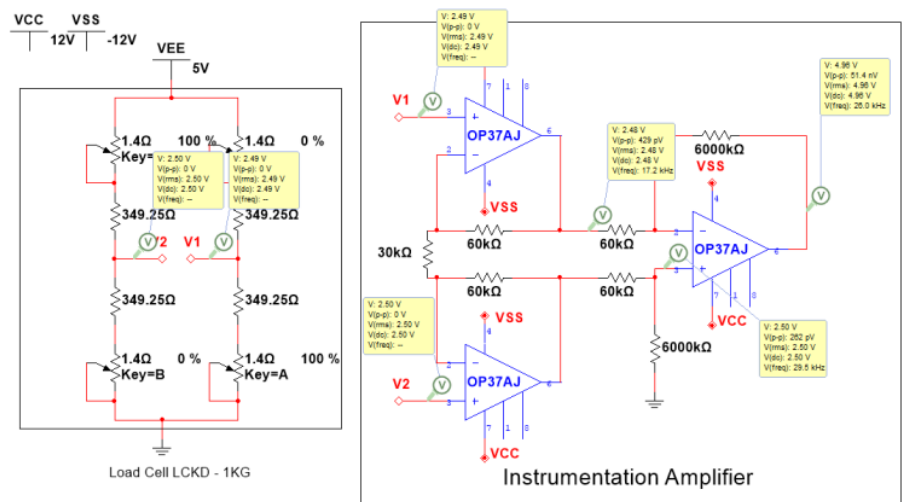


Figure 28 - Load cell with instrument amplifier.

Good choice of resistor values

There should be a heading for the test results

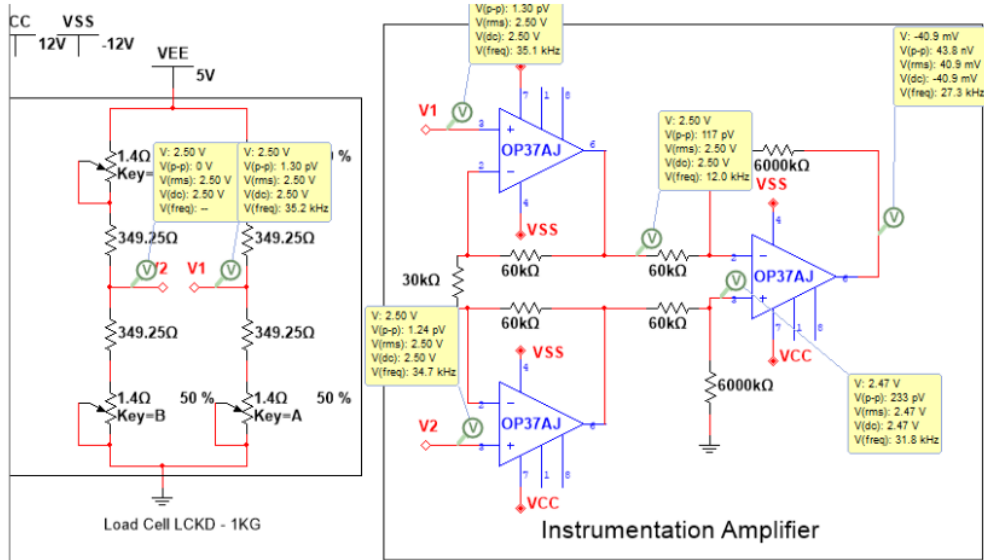


Figure 29 - results at 50% load.

V: -40.9 mV
V(p-p): 21.4 nV
V(rms): 40.9 mV
V(dc): -40.9 mV
V(freq): 27.3 kHz

Figure 30 - result

Vrms=0.041V for load of 50% to 50%.

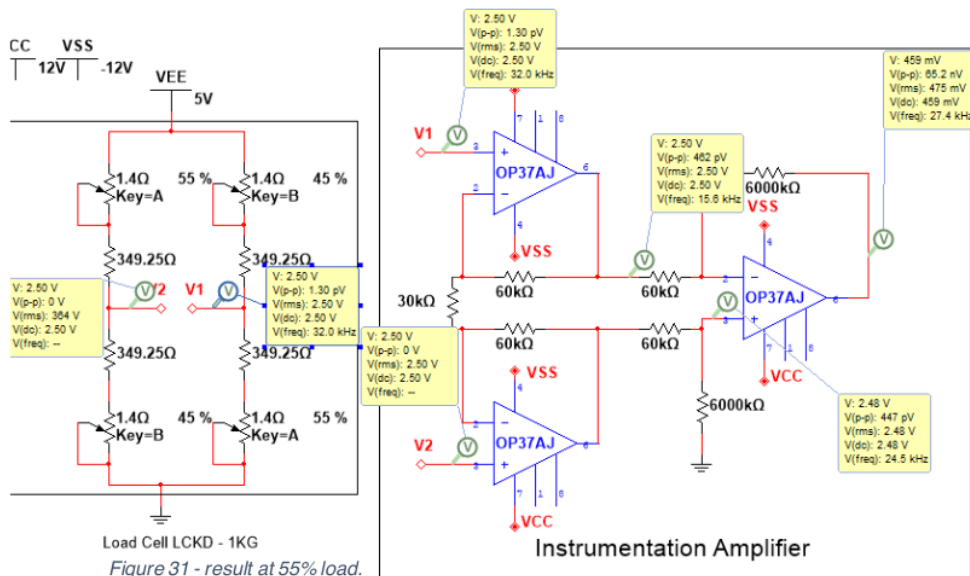


Figure 31 - result at 55% load.

V: 459 mV
V(p-p): 77.8 nV
V(rms): 483 mV
V(dc): 459 mV
V(freq): 27.0 kHz

Vrms=0.459V for load of 55% to 45%.

Figure 32 -

V: 1.46 V
V(p-p): 57.4 nV
V(rms): 1.46 V
V(dc): 1.46 V
V(freq): 25.9 kHz

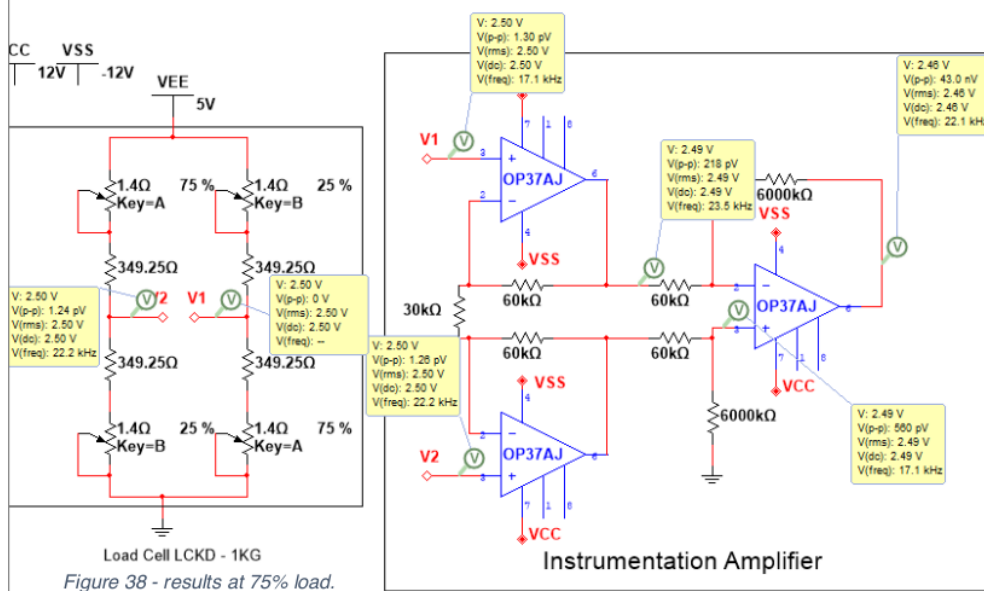
Figure 36 – results.

Vrms=1.46V for load of 65% to 35%.

V: 1.96 V
V(p-p): 82.2 nV
V(rms): 1.96 V
V(dc): 1.96 V
V(freq): 25.0 kHz

Figure 37 - results.

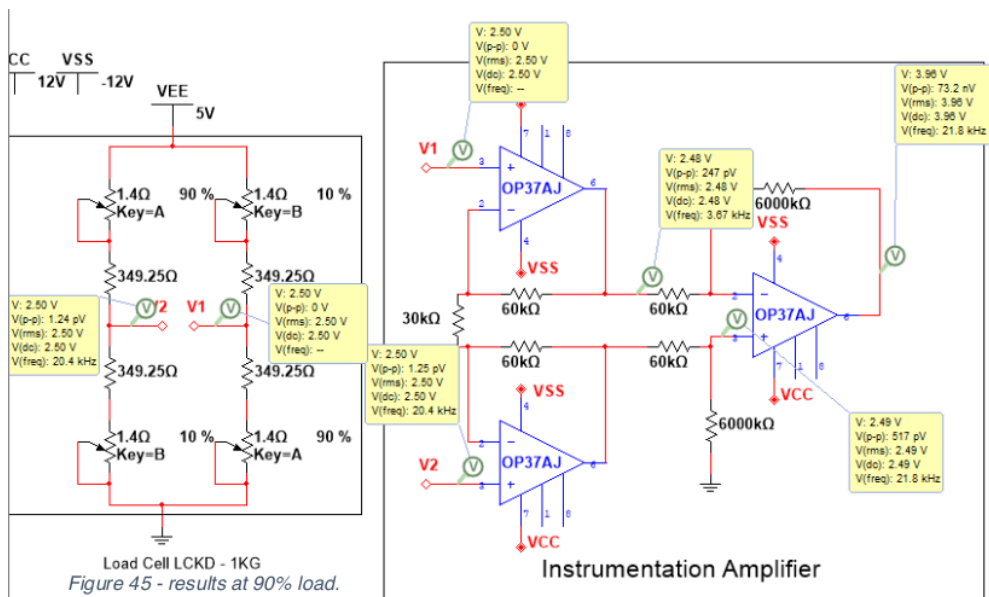
Vrms=1.96V for load of 70% to 30%.



V: 2.46 V
V(p-p): 82.8 nV
V(rms): 3.33 V
V(dc): 2.46 V
V(freq): 22.0 kHz

Vrms=2.46V for load of 75% to 25%.

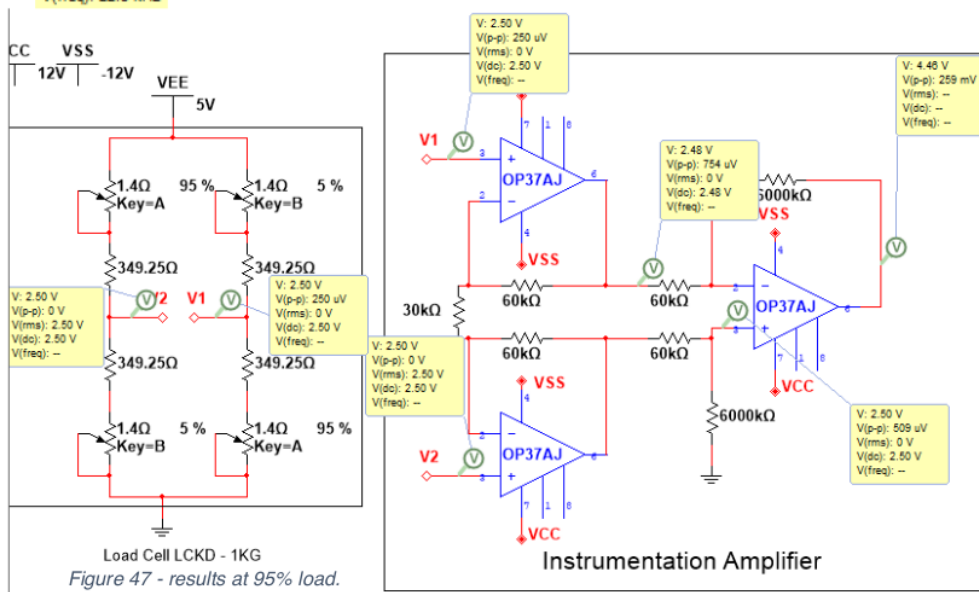
Figure 39 - results.



V: 3.96 V
V(p-p): 73.9 nV
V(rms): 3.96 V
V(dc): 3.96 V
V(freq): 22.8 kHz

Vrms=3.96V for load of 90% to 10%.

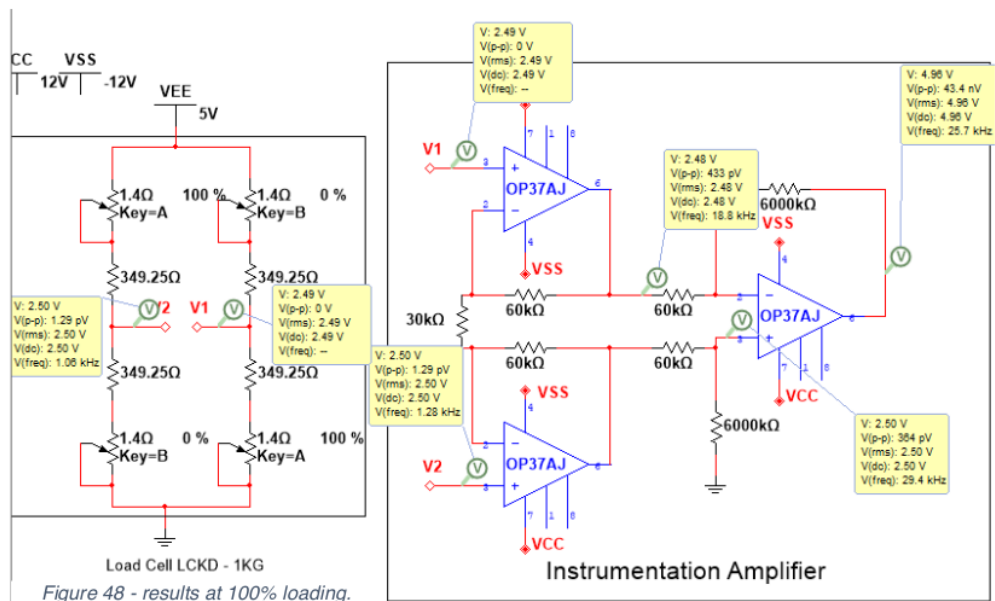
Figure 44 - results.



V: 4.46 V
V(p-p): 72.4 nV
V(rms): 4.46 V
V(dc): 4.46 V
V(freq): 22.1 kHz

Vrms=4.46V for load of 95% to 5%.

Figure 46 - results.



V: 4.96 V
V(p-p): 71.2 nV
V(rms): 4.96 V
V(dc): 4.96 V
V(freq): 28.0 kHz

Vrms=4.96V for load of 100% to 0%.

Instrument amplifier

Buffer gain = $(1 + (2 * 60k) / 30k) = 5$

Diff amp gain = $6000k / 60k = 100$

Gain required:

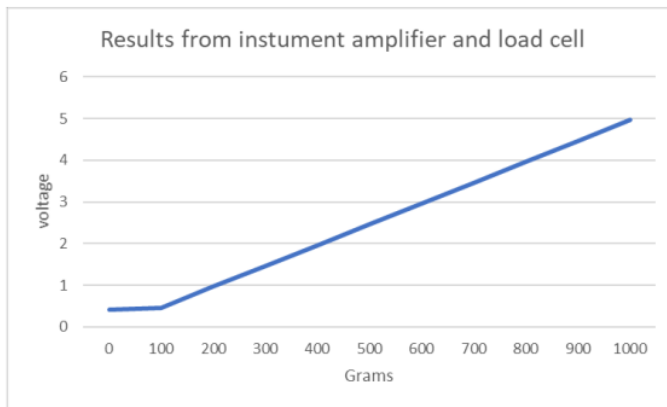
At 1KG and 5V excitation $V2 - V1 = 10mV$

At 1KG $V_{out} = 5V$

Gain $G = 5 / 0.01 = 500$

Overall gain= diff amp gain * buffer gain= $5 \times 100 = 500$

Good calculation that include the chosen resistor values



Good linear graph produced from the resulting values

Table 2 - experiment graphical results.

% Resistance	Grams	Voltage
50	0	0.41
55	100	0.46
60	200	0.96
65	300	1.46
70	400	1.96
75	500	2.46
80	600	2.96
85	700	3.46
90	800	3.96
95	900	4.46
100	1000	4.96

Good table of results

Table 3 – Experiment tabulated results.

The results in the tables above show with every 5% increase in resistance the weight in grams would increase in increments of 100. The results prove that the signal conditioning has worked and each 0.5V for every 100 grams is given out. The error recorded at 0 grams at 0.41V is the leaked voltage, although this voltage is significant, the voltage is carried through the whole experiment so therefore it can be disregarded.

The graphical table above shows the linear response of the load cell, the slight line at the front is the leaked voltage which has caused this line. Other than the voltage leakage the line linearity is what was expected.

Good points included in the explanation

10 bit ADC Prototype:

The figure below shows the block diagram of a 10 bit ADC controller drawn on LabVIEW. The program language is written in block diagram. Each of the visual blocks have various functions which work together to make the circuit function. The diagram is very simple with one input which is analogue and a 10 bit digital output.

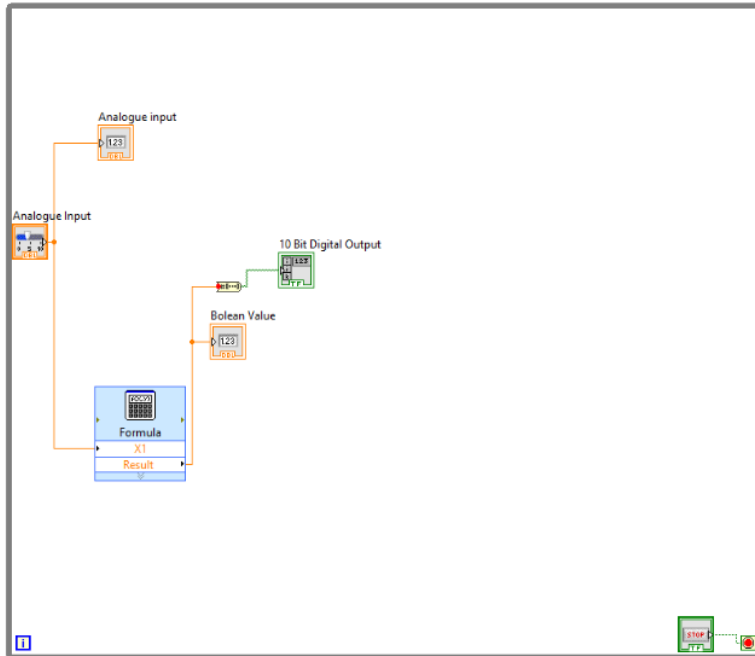
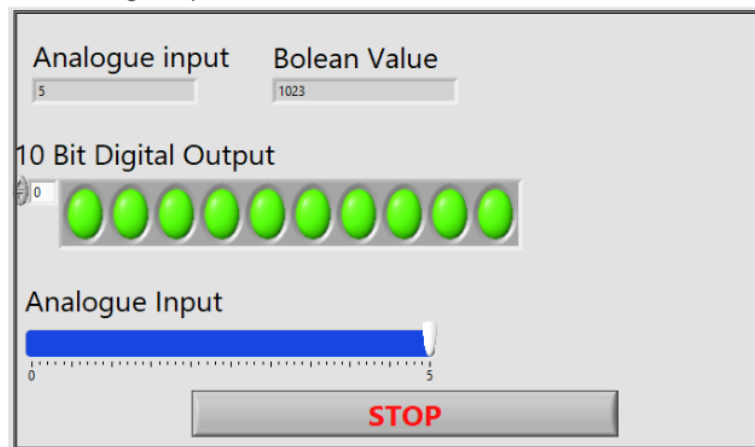


Figure 50 - 10 Bit ADC block diagram.

From the block diagram a HMI is designed to interface with the operator, the panel view in LabVIEW shows the 10 bit count with the analogue input, the Boolean value and a graphical value for the analogue input.



Good test of max value

The figure below shows the formula used with in the formula block diagram above.

figure shows

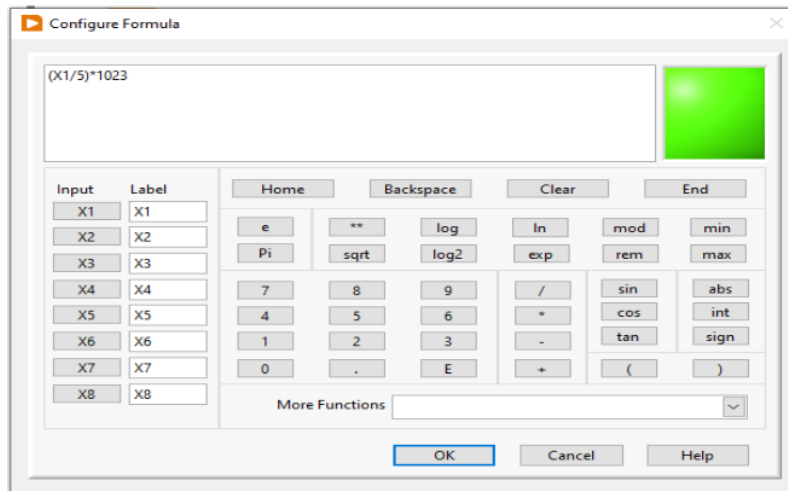
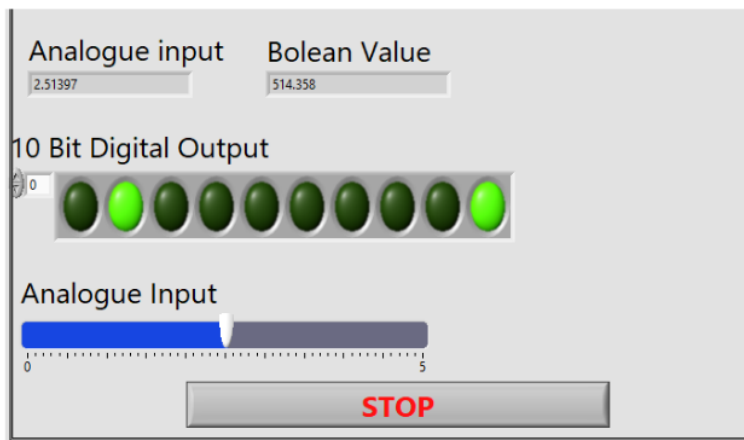


Figure 52 - 10 bit formula block.



Good test carried out

Figure 53 – 10 bit Boolean value at 514 grams.

The figure above shows the test result when the analogue input is set to half which in this case is 2.5V on 5V. As 2.5V is half of the maximum voltage the weight output should be equal to half of the maximum 1023 grams.

12 bit ADC:

From the 10 bit system the system has now been upgraded to a 12 bit prototype. As an result an additional two LED lights have been added to the panel view. With the 12 bit output there is a possible output of 4095.

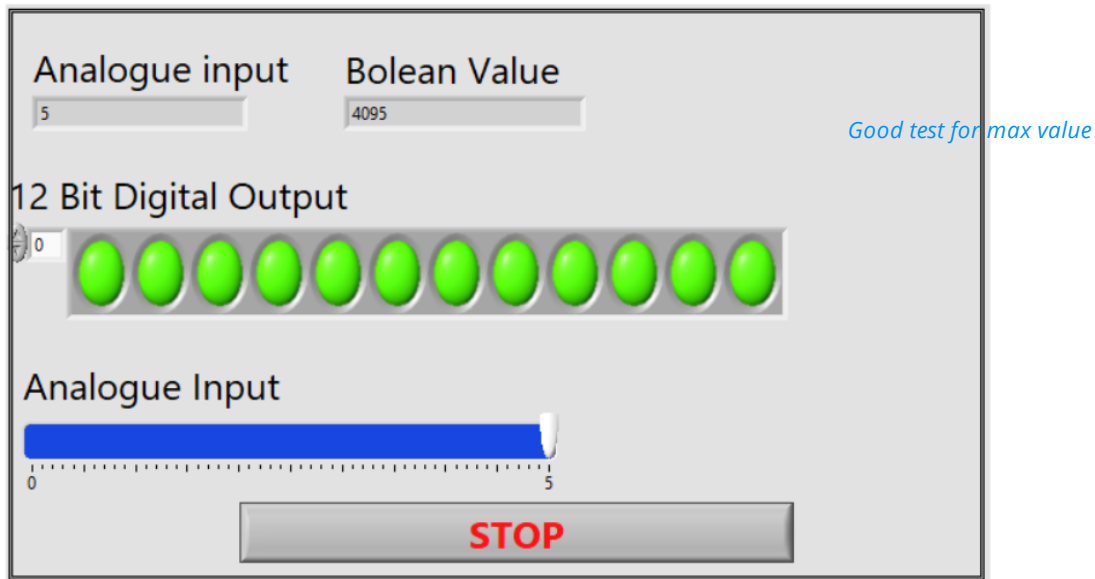


Figure 55 - 12 bit ADC panel view at maximum output.

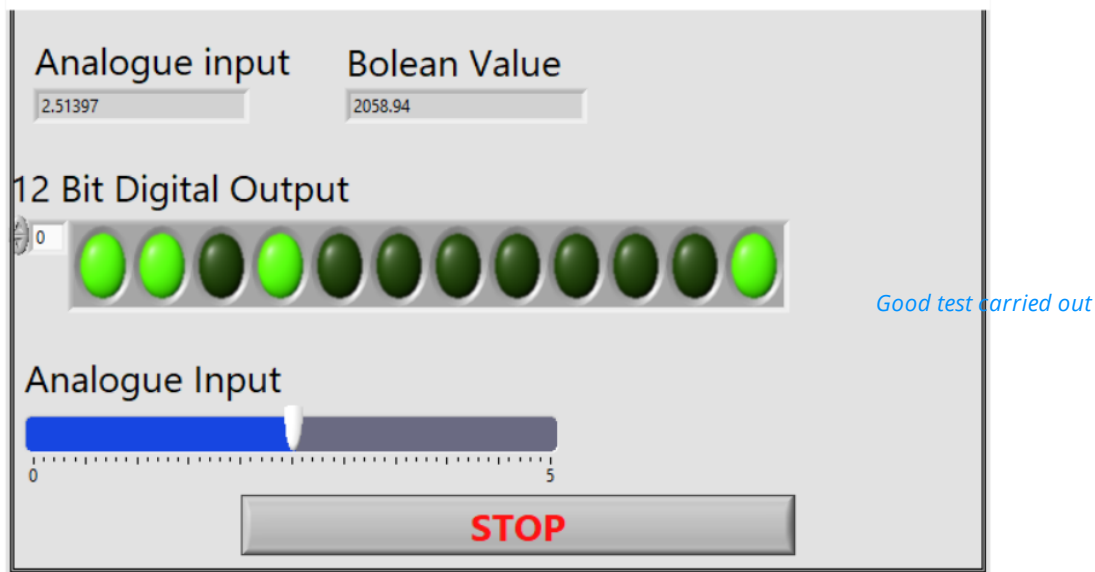


Figure 54 - 12 bit ADC when input is set to half of the maximum.

ADC resolution capabilities of LabVIEW is recognised and tested in example application prototypes

8 Bit ADC:

A 8 bit ADC block diagram and panel view was designed and implemented on LabVIEW. Within this experiment the output of the 8 bit ADC will drive outputs on a magnitude comparator, this will compare three values where the output is either more than, equal to, or less than. These signals are indicated through LEDs.

The voltage input into the system enters through Dev/1 which will have a voltage ranging from 0V to 5V. From the input the chip will have had the block diagram burned into it therefore the three led output signals will light up according to the significant number.

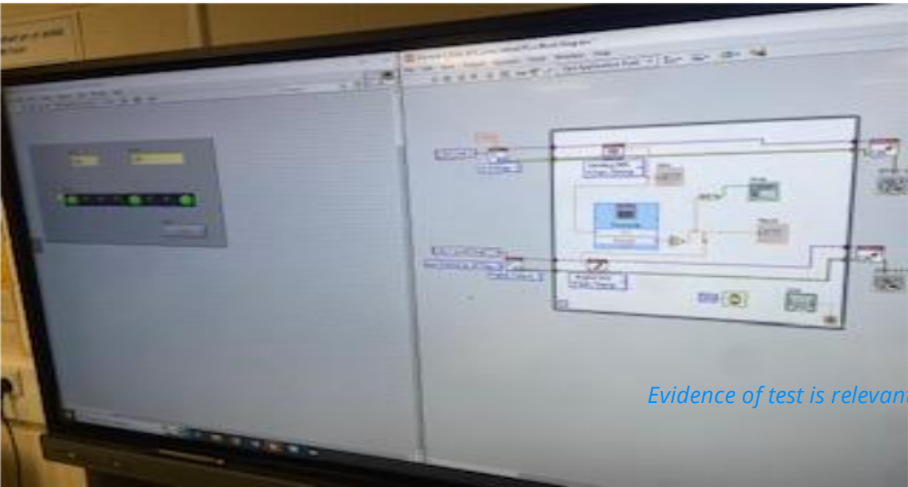


Figure 56 - 8 Bit ADC panel view and block diagram.

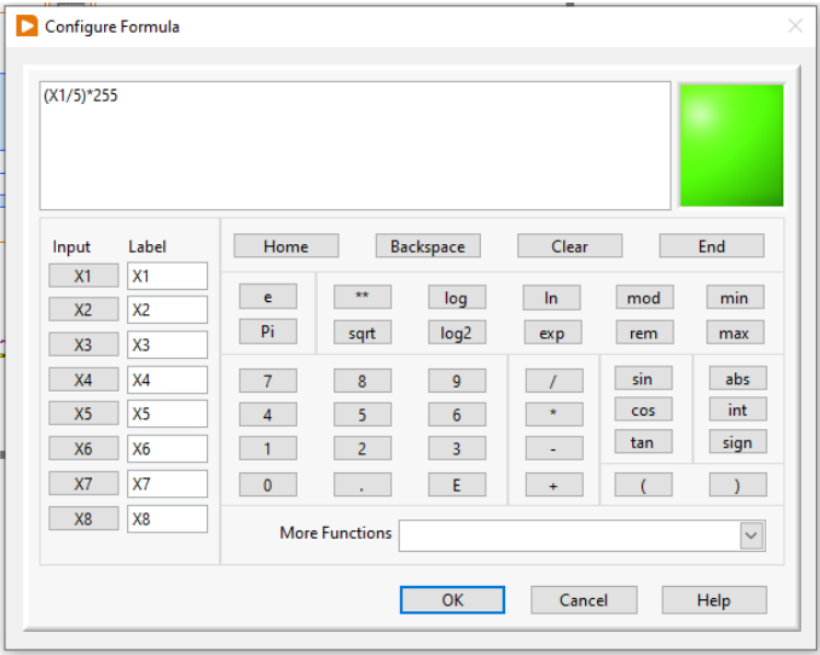


Figure 57 - formula block for 8 Bit ADC.

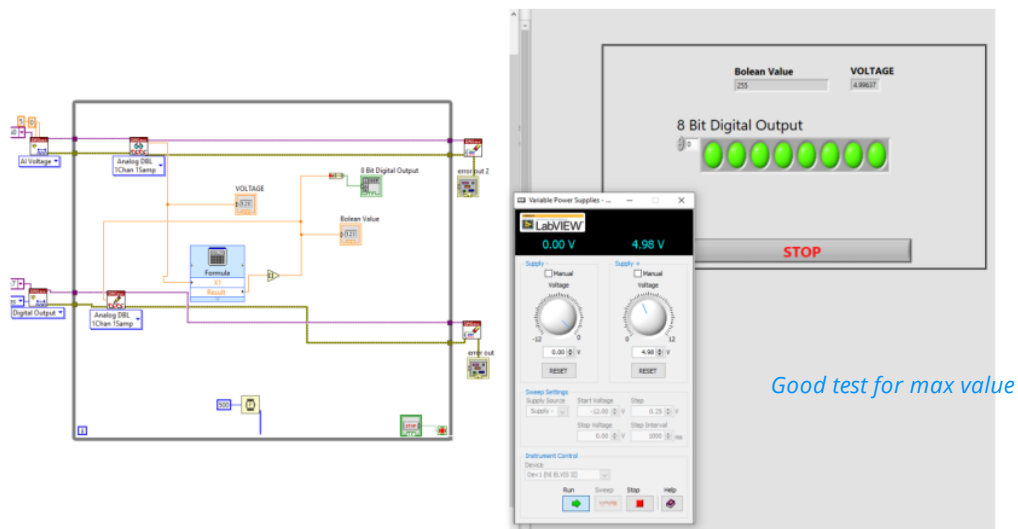


Figure 59 - 8 bit ADC set to 5V.

When the ADC is set to 5 V the results received are what was expected.

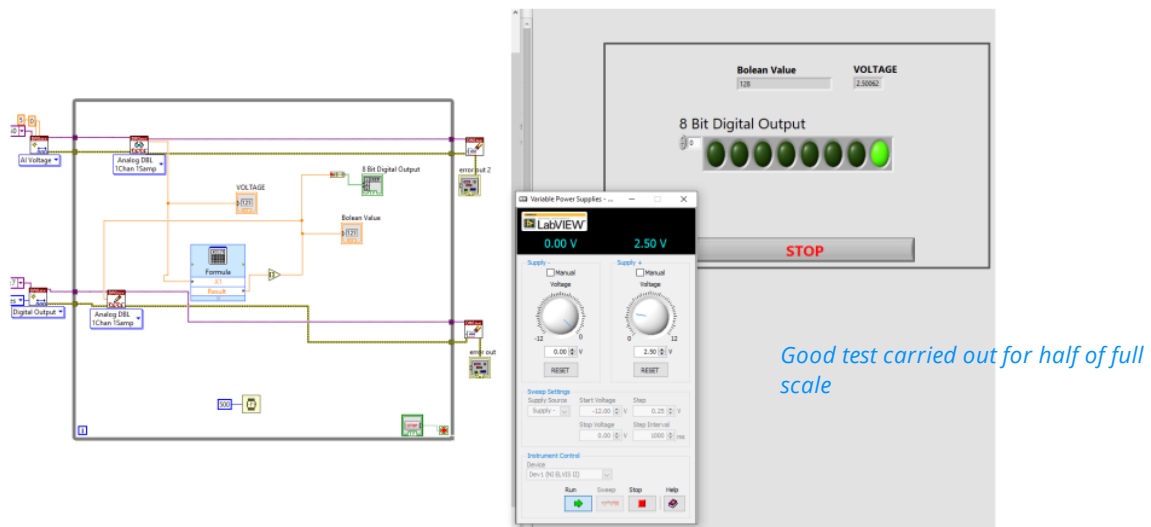


Figure 58 - 8 Bit ADC set to 2.5V.

The results gathered from setting the ADC to 2.5V is what was expected.

This should have a relevant heading



Figure 60 - Greater than.



Figure 62 - equal too.

The relevance of this should be explained



Figure 61 - less than.

From setting the Values the evidence of the greater than, equal to and the less than results are shown on the figures above.

8-BIT MAGNITUDE COMPARATOR

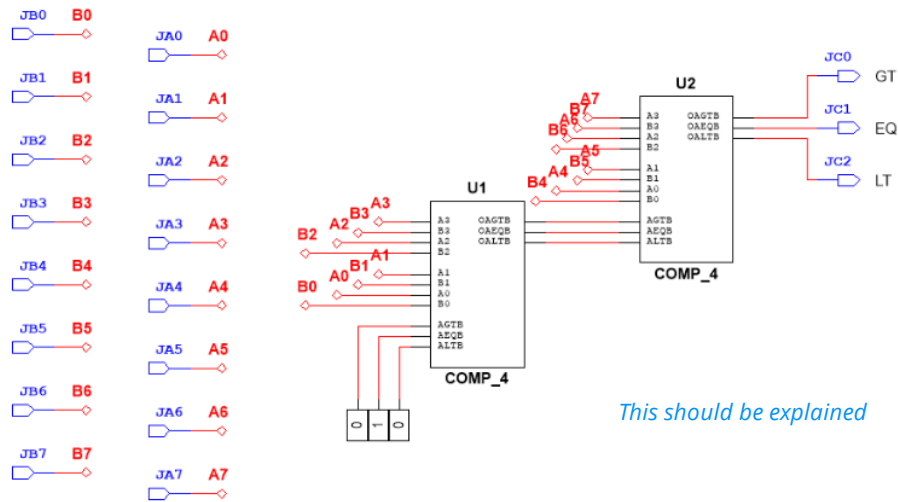


Figure 63 - 8 Bit magnitude comparator multisim.

Evaluation:

In a whole Element two was a complete success. All calculations, multisims, LabVIEW and testing went smoothly. Firstly the amplifier was very successful as it was operating according to the job spec. Where the weight would increase at expected amounts with no errors. The results gathered prove this from visually inspecting the linear graph. The linearity of the graph shows the instrument accuracy.

Computerized skills were then carried over to perform a block diagram on labview with a 12-bit ADC, 10 and a 8 bit ADC. LabVIEW proved itself to be able to use an analogue input and transferring the data into a digital output where also an analogue output was possible. From using LabVIEW lots of experience has been gained in the use of the program and familiarization with how the programming software works. This is very beneficial as many systems involve having these skills to implement on.

There are many parts missing here including the main part

Element 3:

Element 3 is an addition to the software in Element 2. The LabVIEW program was changed to be able to store and record data. Adaptations included adding recording blocks which were named files. 5 files were added to the program to record the results.

Each of the five files were recording individual aspects of the program, File number one was recording the time and the load input in seconds and grams, file number two was recording the setpoint of the program, file number three was recording the more than value of the program, file number four was recording the equal too value, where file five was recording the less than value of the program. Each of the files can be seen in the figure shown below.

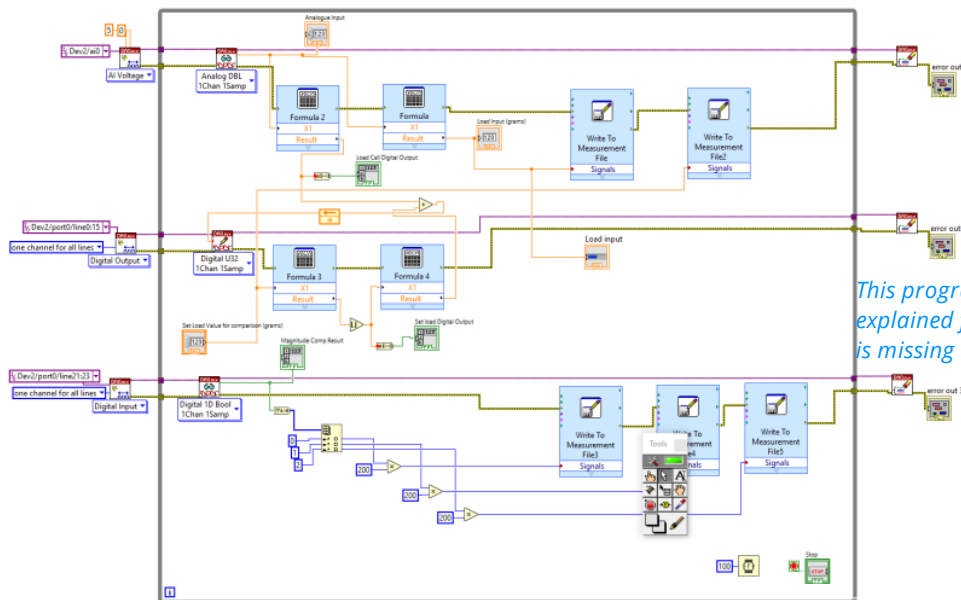


Figure 64 - Element 3 final block diagram program.

Each of the files had to be set up correctly. The figure below shows the configuration used on file one.

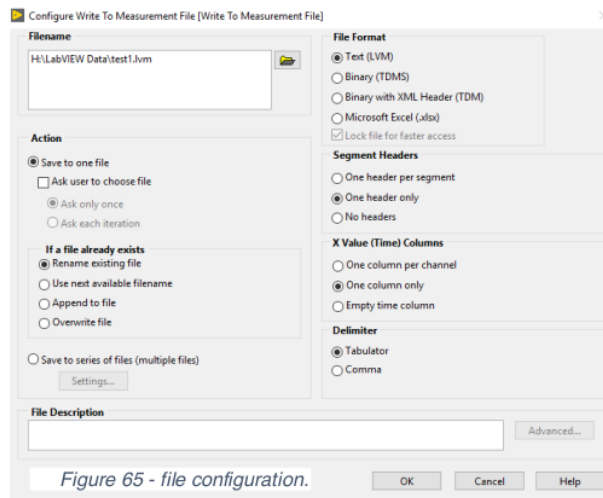


Figure 65 - file configuration.

Testing:

To test the program, it needed to be hooked up to an special board where the computer would speak to the board through a USB link, the test carried out was to test the circuit/program when its set point is set to 200 grams. The results gathered from the test is shown on the tables below.

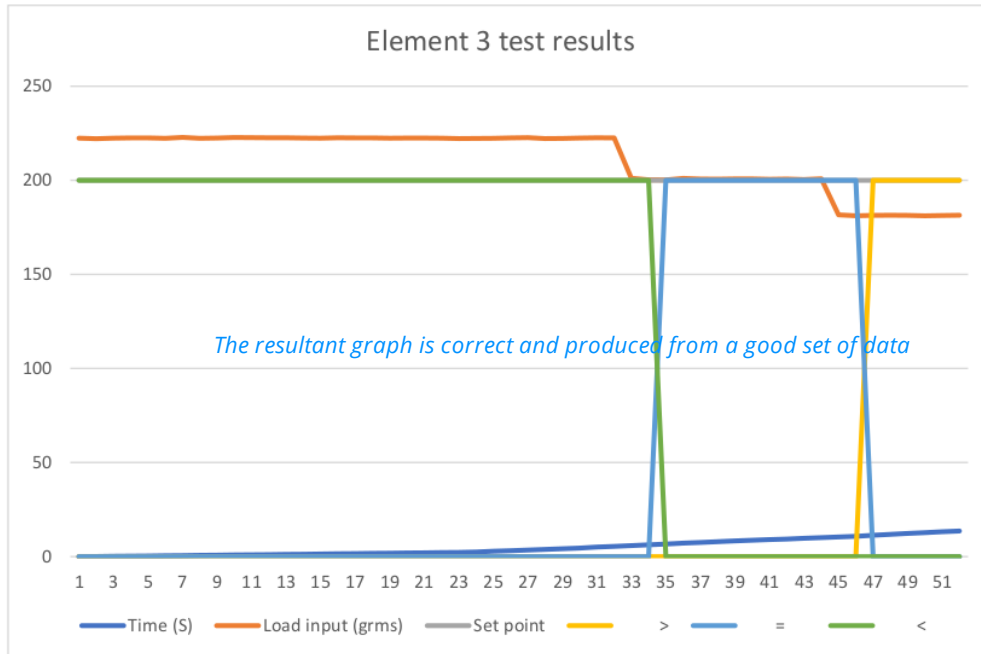


Table 4 – Element 3 graphical test results.

Time (S)	Load input (grms)	Set point	>	=	<
0	222.398554	200	0	0	200
0.100309	222.077358	200	0	0	200
0.202286	222.398554	200	0	0	200
0.30147	222.494913	200	0	0	200
0.401634	222.494913	200	0	0	200
0.50321	222.334315	200	0	0	200
0.603062	222.78399	200	0	0	200
0.703286	222.302195	200	0	0	200
0.804463	222.430674	200	0	0	200
0.902997	222.75187	200	0	0	200
1.003629	222.687631	200	0	0	200
1.104032	222.591272	200	0	0	200
1.202242	222.591272	200	0	0	200
1.304459	222.462794	200	0	0	200
1.442041	222.398554	200	0	0	200
1.540647	222.591272	200	0	0	200

1.643223	222.527033	200	0	0	200
1.742912	222.494913	200	0	0	200
1.851865	222.398554	200	0	0	200
1.951895	222.462794	200	0	0	200
2.050891	222.462794	200	0	0	200
2.151418	222.398554	200	0	0	200
2.250533	222.205836	200	0	0	200
2.480884	222.237956	200	0	0	200
2.847226	222.334315	200	0	0	200
3.169709	222.527033	200	0	0	200
3.489476	222.655511	200	0	0	200
3.859906	222.141597	200	0	0	200
4.212523	222.237956	200	0	0	200
4.561395	222.462794	200	0	0	200
5.068977	222.559153	200	0	0	200
5.476077	222.494913	200	0	0	200
5.887074	201.071123	200	0	0	200
6.29662	200.268133	200	0	0	200
6.707138	200.171774	200	0	200	0
7.169618	201.039004	200	0	200	0
7.558563	200.685688	200	0	200	0
7.951755	200.621449	200	0	200	0
8.338047	200.782047	200	0	200	0
8.687232	200.749927	200	0	200	0
9.032483	200.52509	200	0	200	0
9.37578	200.717808	200	0	200	0
9.727975	200.428731	200	0	200	0
10.07958	200.814166	200	0	200	0
10.45727	181.606652	200	0	200	0
10.82489	181.156978	200	0	200	0
11.33883	181.349695	200	200	0	0
11.8282	181.446054	200	200	0	0
12.32999	181.381815	200	200	0	0
12.76821	181.124858	200	200	0	0
13.22753	181.285456	200	200	0	0
13.6161	181.446054	200	200	0	0

Table 5 – table of results for element 3.

The table above was used to draw a graphical response of the test carried out, the results gathered was expected as the setpoint stayed consistent where the time was increasing and the equal to, less than, greater than bar was fluctuating between the load input as the load fluctuated.

Good sample of test results

Evaluation:

Element 3 was also a very successful element where every setpoint was met and satisfied, some trouble had occurred when building the program but through resolving this issue the program became more and more familiar to then be able to find and amend the fault. The successful development of the LabVIEW program resulted in a wide improvement and development within the use of LabVIEW program. The extra files added to record the results in order to graphically record them worked very well and resulted in the generation of a very well detailed graph. The whole program/system in a whole worked very well and was very reliable when everything was powered/set up correctly. Improvements on the system would be made easy as shown in Element 2, to generate a 12 bit ADC instead of the smaller 8 bit ADC.

A relevant recommendation

The evaluation is poor

Bibliography

Arar, D. S., 2017. *Getting Started with Hardware Description Language for Digital Circuit Design*. [Online]

Available at: <https://www.allaboutcircuits.com/technical-articles/hardware-description-language-getting-started-vhdl-digital-circuit-design/>

[Accessed 17 02 2023].

Circuitdigest, 2022. *Getting Started with VLSI and VHDL using ModelSim*. [Online]

Available at: <https://circuitdigest.com/microcontroller-projects/getting-started-with-vlsi-and-vhdl-using-modelsim>

[Accessed 17 02 2023].

Curran, R., 2005. *Analog devices*. [Online]

Available at: <https://www.analog.com/en/technical-articles/exploring-different-sar-adc-analog-input-architectures.html>

[Accessed 02 03 2023].

Niazi, M. A., 2021. *Control automation*. [Online]

Available at: <https://control.com/technical-articles/various-methods-of-signal-conditioning-explained/>

[Accessed 02 03 2023].

Sasmita, 2015. *electronics post*. [Online]

Available at: <https://electronicspost.com/digital-comparator/>

[Accessed 02 03 2023].

Limited reference sources used

FINAL GRADE

GENERAL COMMENTS

Instructor

40/100

PAGE 1

PAGE 2

Text Comment. This is an error

Text Comment. Comprehensive table of contents

PAGE 3

PAGE 4

Text Comment. Detailed list of figures and tables

PAGE 5

Text Comment. Intro has grammatical errors, however, the detail in the content demonstrates a good understanding of the requirements.

PAGE 6

Text Comment. Good significant points covered in the research with relevant example and source.

Text Comment. Limited research on this topic area and no supporting ref source

PAGE 7

Text Comment. Diagram is not much use without an explanation of relevance. The function blocks of the DAQ should have been identified and discussed.

Text Comment. The expansion of the digital mag comparator should be included. The image is relevant with supporting ref source. There are grammatical errors.

PAGE 8

Text Comment. There are good point included in this section, however, this could have been presented more effectively.

Text Comment. Grammatical errors

PAGE 9

Text Comment. There are significant points covered, however, there is no ref source.

Text Comment. Grammatical errors

Text Comment. Relevant points included in the content. Relevant image example with explanation.

PAGE 10

Text Comment. Good points covered with the correct block diagram needed for the VHDL design.

PAGE 11

Text Comment. Good evidence of hierarchical design requirements

PAGE 12

Text Comment. 4 bit adder code needs some explanation

PAGE 13

Text Comment. The IP for the 4 bit adder should be identified in this diagram

Text Comment. The VHDL here is not needed, as the code is auto generated from the Msim design

PAGE 14

Text Comment. This is image is incorrect

Text Comment. Good test results using virtual bus method

Text Comment. Test results are good

Text Comment. This is the test result for the 4 it adder

Text Comment. Grammatical errors

Text Comment. No explanations for test results

PAGE 15

Text Comment. This needs a heading

Text Comment. These test results need to be explained .

PAGE 16

PAGE 17

Text Comment. 8 bit mag comparator Msim tests are good

Text Comment. This is the 8 bit mag comparator and 4 bit adder combined

Text Comment. There are test results missing

Text Comment. The introduction identified the requirements, however, there is significant content missing.

PAGE 18

Text Comment. The intro indicates an understanding of requirements

PAGE 19

Text Comment. Good result

Text Comment. Parameters are identified

PAGE 20

Text Comment. Good choice of resistor values

PAGE 21

Text Comment. There should be a heading for the test results

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

Text Comment. Good calculation that include the chosen resistor values

PAGE 27

Text Comment. Good linear graph produced from the resulting values

Text Comment. Good table of results

Text Comment. Good points included in the explanation

PAGE 28

Text Comment. Good test of max value

PAGE 29

Text Comment. Good test carried out

PAGE 30

Text Comment. Good test for max value

Text Comment. Good test carried out

Text Comment. ADC resolution capabilities of LabVIEW is recognised and tested in example application prototypes

PAGE 31

Text Comment. Evidence of test is relevant

Text Comment. Block is set for 8 bit resolution

Text Comment. Good test for max value

Text Comment. Good test carried out for half of full scale

Text Comment. This should have a relevant heading

Text Comment. The relevance of this should be explained

Text Comment. This should be explained

Text Comment. There are many parts missing here including the main part

Text Comment. This program should be explained fully here, as it is missing in element 2.

Text Comment. All the files need to be included

Text Comment. The resultant graph is correct and produced from a good set of data

Text Comment. Good sample of test results

Text Comment. A relevant recommendation

Text Comment. The evaluation is poor

Text Comment. Limited reference sources used

RESEARCH-INFORMED LITERATURE

 Resraech conains many grammatical errors and content is limited with some topics. Relevant sources have been included.

4/10**KNOWLEDGE AND UNDERSTANDING**

 Element 1 includes relevant parts of the requirements . However, there are many parts missing and disjointed content. 3.5/10 Element 2 instrument amplifier has been covered with good component values chosen. ADC prototypes are included and tested. The main LabVIEW program with FPGA is not implemented or tested which is the main part of this element. Mainly a lack of evidence included in this element. 3/10 Element 3 has missing explanations and test results. There is evidence of the labVIEW program and resultsnt data from the write files . 4.5/10

11/30**ANALYSIS**

 Element 1 test results have not been fully achieved. The combination IP is not fully tested. There is evidence of the 8 bit mag comparator test with FPGA. The 4 bit mag comparator and 4 bit adder

10/25

have been tested using Vivado. There are explanations missing. 2/5 Element 2 ADC prototype test results are complete. The main LabVIEW with FPGA tests are missing. Evaluation is poor. 3.5/10 Element 3 includes evidence of the LabVIEW program with write files and a good set of test result data. The graph has been created from the resultant test data and is correct. Good scaling has been implemented in the program. There is evidence of the text .lvm files missing. 4.5/10

PRACTICAL APPLICATION AND DEP.

 Element 1 design and build has been carried out. There are missing explanations and content evidence is disjointed. The block diagram has been applied in the program and there is evidence included. The IP design has been created. 2/5 Element 2 interface mapping evidence is missing. The main LabVIEW and FPGA application is missing. There are prototypes and instrument amplifier included. 1.5/5 Element 3 includes evidence of the LabVIEW and FPGA block design, however, no front panel design is included. Data logging practical has been carried out with evidence of resultant data and graph. 4.5/10

8/20

SKILLS FOR PROFESSIONAL PRAC.

 Element 1 has many grammatical errors and content is disjointed. There is some consistency in the test section, however there are errors with identification. 2/5 Element 2 has many parts missing, there is some consistency with the

7/15

content that is present with reasonable explanations. 2.5/5 Element 3 has most of the required content in order. There are parts missing.
2.5/5