

1102216

by Sean Parry

MARK AGREED AT 60%



Submission date: 13-Feb-2023 11:12PM (UTC+0000)

Submission ID: 197213255

File name: 3005_Sean_Parry_1102216_181626_1555283462.docx (8.83M)

Word count: 5236

Character count: 23197

Course: BSc (Hons) Electrical and Electronic Engineering

Module: SGEE6002 Electronic Production and Test

Assignment Title: Fault Modelling

Name: Sean Parry

University number: 1102216

Date Submitted: 13/02/2023

Word Count: 3006 words

Table of Contents

1. Executive Summary	5
2. Research.....	6
3. Design of assignment requirements	9
4. Analysis of NI Multi-Sim results	10
5. Practical build and test using LabVIEW	12
6. CAM design and build of prototype.....	13
7. Conclusions and Recommendations	14
8. Citation List	15
9. Appendix	17

Contents table should be more detailed for the number of tasks required.

Table of figures

Figure 1 – Stuck at 0 fault model	17
Figure 2.....	17
Figure 3 - Fault equivalent	18
Figure 4 - ESD example sources and values.....	18
Figure 5 - PCB damaged by ESD	18
Figure 6 - Preventative measures against ESD	18
Figure 7 - CUT (Circuit Under Test)	19
Figure 8 - Reduced tests - 3 input NAND gate	20
Figure 9 - Reduced tests - 2 input NOR gate.....	20
Figure 10 - Reduced tests - 2 input AND gate	20
Figure 11 - Reduced tests - 2 input OR gate	20
Figure 12 - Derived table	21
Figure 13 - Reduced table	22
Figure 14 - Further reduced table after applying wild card rules.....	22
Figure 15 - Multisim test using Word Generator	23
Figure 16 - Multisim result for no fault condition	23
Figure 17 - A SA1 Fault condition	24
Figure 18 - Test result for input A SA1	24
Figure 19 - B SA1 fault condition	25
Figure 20 - Test result for input B SA1	25
Figure 21 - C SA1 fault condition	26
Figure 22 - Test result for input C SA1	26
Figure 23 - Next state table.....	27
Figure 24 - Output QA.....	27
Figure 25 - Output QB.....	27
Figure 26 - Output QC	28
Figure 27 - Output QD	28
Figure 28 - Output QE.....	28
Figure 29 - Output QF.....	29
Figure 30 - 6-Bit Register with no fault condition	29
Figure 31 - Test result with no fault condition	30
Figure 32 - CUT with input A SA1 fault condition.....	30
Figure 33 - Input A SA1 fault test result.....	30
Figure 34 - CUT with pulse B SA1 fault condition	31
Figure 35 - Input B SA1 fault test result.....	31

Figure 36 - CUT with input F SA1 fault condition	32
Figure 37 - Input F SA1 fault test result	32
Figure 38 - NI LabVIEW	33
Figure 39 - NI LabVIEW	33
Figure 40 - NI LabVIEW	34
Figure 41 - NI LabVIEW ASA1	34
Figure 42 - NI LabVIEW BSA1	35
Figure 43 - NI LabVIEW CSA1	35
Figure 44 - NI ELVIS ASA1	36
Figure 45 NI ELVIS BSA1	37
Figure 46 - NI ELVIS CSA1	38
Figure 47 - NI FPGA ASA1	39
Figure 48 - NI FPGA BSA1	40
Figure 49 - NI FPGA CSA1	41
Figure 50 - PCB ASA1	42
Figure 51 - PCB BSA1	43
Figure 52 - PCB CSA1	44
Figure 53 - Circuit design on Eagle	44
Figure 54 - Gerber view bottom layer	45
Figure 55 - Gerber view top layer	45
Figure 56 - Mechanical drill holes	46
Figure 57 - Board complete	46
Figure 58 - Completed board - bottom layer	47
Figure 59 - Completed board - top layer	47
Figure 60 - Completed board on eagle	48

Good list of figures

1. Executive Summary

To demonstrate a systematic understanding of electronic production techniques. The assignment was focused on ATE (Automatic Test Equipment) for testing a prototype design where research, simulation and practical investigations were undertaken. Research was carried out as a foundation to build upon. Research was carried out to evaluate a range of processes, techniques and strategies such as IC fabrication faults and SSF modelling to successfully develop a prototype design. An evaluation of the methods and techniques required to manufacture a sustainable product that meets the wider engineering requirements. Research was undertaken to determine the effects of Electro Static Discharge (ESD) and prevention methods used within the electronic design. Then the prototype design and fault model. The circuit prototype was designed and built of a combinational circuit consisting of four gates, having 6 inputs and an output. Implementation of fault modelling techniques was carried out using a stuck at fault table to detect SA0 and SA1 faults on the prototype. Then the testing of the prototype to meet customer needs using ATE designs. This involved the developing of a b bit pattern generator ATE system which utilises the derived minimum number of test vectors required to test the circuit. An analysis of the test results obtained and applying various different input faults. The prototype circuit was built on the NI ELVS unit and using NI LabVIEW and NI Multi-sim circuit 6-bit pattern generator to generate the minimum number of tests vectors to test the prototype. An analysis of the practical results of the ATE systems was carried out by applying identical input fault conditions to the prototype. Then, finally, a CAM design of the prototype was designed and built. An industry approved CAM software known as Eagle and Gerber files were used to develop a double-sided prototype PCB of the CUT.

There are good points summarised here, however, there are parts here that are unclear!

2. Research

Due to the processes used to manufacture integrated circuit, there are small variations that can result in poorly constructed circuits and built in faults. As stated in VLSI Design Verification and Test, 'Failure mode is used in reference to the manifestation of a "defect" at the electrical level.' (Plusquellic, 2004). These defects can present themselves as shorts or open circuits in the metal layers of the circuit. There are several causes of this, including, unwanted material or unwanted etching in the material. Chemical deposits left from processing. Dust particles, causing problems during photolithography. Optical lithography is a process that uses a light sensitive polymer to print complex circuit wiring patterns onto a thin wafer material. Semiconductor Lithography (Photolithography) – The Basic Process explains how thin layers of conductors (e.g. copper) and insulators (e.g. Silicon dioxide) connect many electronic components such as transistors together and 'dope' regions of silicon to alter the conductivity when voltage is applied, connect many electronic components such as transistors together and 'dope' regions of silicon to alter the conductivity when voltage is applied, (Mack, 2020).

This is a good point

The complexity of modern IC's and the high precision processes used to manufacture them can be the cause of fabrication faults which occur when producing these devices. An IEEE journal declares that the severity of imperfections in the fabrication process grows proportionally with size and density of the chip, (Koren, 2021). IC fabrication faults cover a range of fault modelling techniques which enable test and analysis of integrated circuits. The different categories of IC fabrication faults are as follows; firstly, gate level stuck at fault which occurs when a logic gate has an input stuck at 1 or 0 which effects the logical output of the device. Secondly, transistor level stuck at fault describes a transistor as a switch that is wither stuck open or closed. Thirdly, bridging faults which describe various open or short circuit faults between connections within a circuit. A short between two wires is known as a bridge fault. Fourthly, delay faults which describe the fault being with resistive open or short circuits which have the potential to cause a signal to propagate so slow that the time specification is not achieved. Fifthly, resistive open faults represent common manufacturing defects in integrated circuits and result in delay faults that cause timing failures and reliability issues. The delay faults need to be detected by speed testing the silicon at the speed faster than the requires speed, (Mohammadat, 2021). Sixthly, IC fabrication faults include fault modelling techniques for common faults as seen above, (Fall, 2021). There are three properties which define a single stuck at fault. Firstly, only one line or wire is faulty. Secondly, the faulty line is permanently set to zero or one. Thirdly, the fault can be at the input or the output of a logic gate.

Good point here with supporting ref source

Fault types are identified with supporting ref sources

Ref in appendix

Figure 1 shows an example of determining a test vector displaying a stuck at fault of NAND gate with input 'a' stuck at 0. The fault can be determined by applying the test vector '11' to the gate. This will result in the output of the gate to transition from 0 to logic 1.

The main stuck at faults that can be modelled are: firstly, SA1 stuck at logic 1. Secondly, SA0 – stuck at logic 0. Thirdly, bridging fault – short circuit between two gates. Fourthly, feedback bridging fault – short circuit between an output and feedback of a gate.

Ref in appendix

The diagram in figure 2 demonstrates the stuck at fault categories. Delay faults can also occur which are caused by adding resistance in the line. This results in added propagation delay. Here we can see how the logic gate is forced either to logic 1 (Stuck at 1) or logic 0 (Stuck at 0) by a defect linking the output to the voltage high or ground, however these aren't the only causes of a SA1 or SA0. Transistors can develop faults where they are stuck open or closed. There are also other faults known as bridging faults between two gates, or between an output and feedback. Delay faults can also occur, caused by added resistance in the line that makes a voltage signal take longer than desired to reach the next stage, altering the timing. Resistive Open Faults Detectability Analysis can Implications for Testing Low Power Nanometric IC's document describes, "Resistive open faults (ROFs) represent common manufacturing defects in integrated circuit (IC) interconnects and result in delay faults that cause timing failures and reliability risks." (Mohammadat, 2015). The document also further explains how these delay faults need to be detected by speed testing the silicon at the required or faster than required speed.

Good research with this topic

Figure 3 shows the fault equivalent method applied to a two input AND gate and a two input OR gate. When one input is stuck at logic 0 in the AND gate, the output will be logic 0. Therefore, two of the three faults are joined with the arrows can be eliminated. The same method can be applied to the OR gate and in both causes, the fault equivalent technique reduces the number of fault vectors. Fault equivalence is when two faults are equally dominant and only one test needs to be performed to detect either, (Uber, 2021). The single stuck at fault model can be produced to detect faults using a known I/O table to compare with the expected output results. Statistically single fault tests cover a large number of multiple faults, this means that most of the fault tests can be removed by the fault collapse method, (Mazumder, 2002).

This area SSF is covered very well with the supporting ref sources and images in appendix.

This should have a heading!

Electronic discharge is the release or transfer of static electricity from a positive charged object to a negative charged object. There is a large potential difference which causes the damage. The discharge will always take the path of least resistance to equalise the charge. Any part of the path that is unable to withstand the energy associated with the discharge sustains damage. The small discharges can damage the components on IC's easily when a high charge dissipates through the components at a lower charge, (Toms, 2015). The damage caused by ESD is more significant as modern IC's decrease as the surface area of the discharge affects a larger proportion of the circuit. Some sources of ESD and relative voltages that can occur during everyday life and give large magnitudes are shown in figure 4. When ESD occurs both current and heat are generated and the heat is released onto an electronic device. The small components can melt or even vaporise causing device failure. If the damage is not catastrophic then the device will survive initially onto to cause delayed failures and reduce the life of the device. This is known as the latent defect, (Rouse, 2011). Figure 5 shows an example of ESD damage to a PCB. Electronics manufacturers therefore, establish electrostatic protective areas free of static known as clean rooms which use measures to prevent charging, such as avoiding highly changing materials and measures to remove static such as grounding human workers, providing antistatic devices, and controlling humidity. ESD simulators may be used to test electronic devices, for example with a human body model or a charged device model. Some of these preventative measures include wearing a wrist strap or ESD control footwear during the manufacturing process and using conductive, static dissipative and anti-static material as demonstrated in figure 6, (Storr, 1998). ESD devices should be contained in a static protective bag or container at all times during storage or transportation, (Mini-circuits, n.d.).

Ref image in appendix

Ref image in appendix

Good coverage of this topic and supported with relevant images and ref sources

There should be an investigation on wider modern methods for manufacturing a sustainable product.

3. Design of assignment requirements

The circuit design in figure 7 was created using ECAD design and will be used as the CUT for *Image ref in appendix* the test and analysis. The SA fault model technique was implemented in order to detect SA faults at zero (SA0) SA faults at logic 1 (SA1). The number of test vectors required for each individual gate was reduced using the formula number of tests where $m + 1$ where m is the number of inputs. Figure 8 demonstrates the 3 input NAND gate SA faults model and shows that there are 3 inputs and using the formula there will be 4 tests required. Figure 9 shows the table with the reduced number of tests for the 2 input NOR gate. Figure 10 shows the table with the reduced number of tests for the 2 input AND gate. Figure 11 shows the table with the reduced number of tests for the 2 input OR gate. Figure 12 shows the derived table for all possible stuck at fault tests and the possible fault scenario required to test each stage. The highlighted tests in figure 12 indicate the duplicate tests which cause redundancies. Test 1 and 9 are duplicated and tests 5, 8 and 13 are duplicates. Only one of these are required and therefore this table can be reduced. Figure 13 shows the resultant table after the redundant tests have been terminated. Test 9, 8 and 13 have been removed. Test 5 can be altered using the wildcards so that it becomes identical to test 4. This created a redundant test vector and was removed according to the equivalence rule. The same rule was applied to tests 6 and 10 and redundant test was removed. The result was a further reduced table consisting of 8 test vectors instead of the original 13 as shown in figure 14. It can be noticed that the fault condition of the redundant test vector requirements is added to the existing test vector.

SSF model used to create test vectors for each gate
Initial table is created
Redundancies are eliminated using SSF model rules

Figure 15 demonstrates the CUT ATE using a Multisim Word Generator to generate the test sequence required at the results. The test shown in figure 15 demonstrates the no fault condition for the CUT. The resultant waveform was analysed using the logic analyser. Figure 16 demonstrates the simulation results from the word generator test sequence and indicates the no faults condition in the resultant waveform between the two cursors. Figure 16 is the no fault condition for the CUT and no positive pulses on the resultant output shows the correct operation of the test.

Ref in appendix

NFF condition achieved. This is critical before tests can be carried out

4. Analysis of NI Multi-Sim results

The design circuit using Multisim was used to demonstrate the fault conditions from the test vectors derived from tables. The tests confirm the fault condition and test result by using the simulation design to generate the data sequence for the CUT inputs. Figure 17 demonstrates the CUT modified with a fault condition according to the table of test vectors of input A SA1, B SA1 and C SA1. The test result in figure 18 shows the resultant output from the logic analyser at logic 1 when input A is held at logic 1 throughout the test sequence. The fault condition is indicated between the two cursors. The test result in figure 19 and 20 shows the test circuit and resultant output from the logic analyser at logic 1 when input B is held at logic 1 throughout the test sequence. The fault condition is indicated between the two cursors. The test result in figure 21 and 22 shows the test circuit and resultant output from the logic analyser at logic 1 when input C is held at logic 1 throughout the test sequence. The fault condition is indicated between the cursors.

Ref in appendix

Ref in appendix

Ref in appendix

There is an error in one of the tests here. See figure 20, B- SA1.

A 6 – bit register was designed and implemented as part of the ATE design methods for the purpose of testing the CUT. The same fault scenario will be utilised for fault detection and an analysis will be carried out of the results from this method. Figure 23 shows the next stage table required for the design of the finite state machine that produces the test vectors for inputs A to F. Karnaugh maps were used to generate the logic stated required for the feedback combinational logic for the input sequence. Figure 24 shows the K-map for output QA of the register. See figure 24 for the Boolean expression from the K map. Figure 25 shows the K-map for output QB of the register. See figure 25 for Boolean expression. Figure 26 shows the K-map for output QC of the register. Figure 27 shows the K-map for output QD of the register. Figure 28 shows the K-map for output QE of the register. Figure 29 shows the K-map for output QF of the register.

Ref in appendix

Ref in appendix

Expressions have been determined from K-maps

The 6-bit register shown in figure 30 was created from the results of the Next State logic design process. The feedback circuits are the results of the Boolean expressions taken from the K map procedure to minimise the combinational circuit. The CUT (Circuit Under Test) is shown in the image and the output of the CUT is compared with the output of a good known working identical circuit. The XOR gate is used to differentiate the outputs of the CUT and the known working circuit. The result of the test with no fault conditions is shown in figure 31. Both outputs of the CUT and known good test circuits are the same. The blue line is the output of the XOR gate and no pulses are present which indicates no differences at the output.

The good cct and CUT are both missing from figure 30. The X-OR gate is also missing

Figure 32 shows the fault condition with input A SA1 on CUT where the input A is permanently attached to a logic 1 condition for the entire test run. Figure 33 shows the test result from the logic analyser where the fault is detected by a difference in the outputs from the CUT and the known good test circuit. The same test run was carried out for faults set for input B SA1 and input C SA1 and the resultant output of each is seen in figure 34. See figure 34 for CUT with input B SA1 fault condition. See figure 35 for input B SA1 fault test result. See figure 36 for CUT with input F SA1 fault condition. See figure 37 for input F SA1 fault test result.

Tests for all three fault conditions have proven successful

5. Practical build and test using LabVIEW

The following LabVIEW program allows the interface with the DAQmx development board. This enables the physical circuit to interface with the LabVIEW program. This is the virtual reality power of LabVIEW. The physical circuit can be controlled and tested using the virtual instrumentation to LabVIEW. The inputs run from Q to A this is where A is the Msb and Q the Lsb. The fault detection is from the output of the EX-OR gate on the CUT. The binary data are the results from the fault model reduction logic table. The test number is the iteration of the 'for loop' in the program. Figure 39, this first block is for driving the inputs into the DAQmx dev board via the DAQmx instrument channel I/O controller. The 'while' loops is for the program to run forever until stopped manually. The 'for loop' runs 8 iterations for the 8 test vectors and the time delay is for the sample rate. As seen in figure 40, the second block takes the output from the 'EX-OR' gate from the DAQmx dev board and displays its logical condition. To expand on the following an NI ELVIS board was used to generate the minimum number of test vectors to test the prototype. Figures 38, 39 and 40 display the prototype on NI LabVIEW. Figures 38 to 40 are the LabVIEW program for the test vectors and the feedback from X-OR gate. Figure 39 displays the output on 7 lines 0 – 6 the decimal values from the table of test vectors. Figure 40 displays the line 23 output from X-OR gate which is read and displayed on the front panel. The program is made up of array functions for the sequence of test vectors and time delays. The communications and control is via the DAQMX design tools which enable the read and write data from the prototype board. Three faults were then generated, ASA1, BSA1 and CSA1, see figures 41, 42 and 43. The fault results were then displayed on the NI ELVIS board as seen in figures 44, 45 and 46.

Good explanation

This is correct

This could have been written using an improved method i.e. using smaller paragraphs and separating the test procedures.

All the tests are present with evidence. There are minor errors with the heading containing FPGA. There are no FPGA boards used!

6. CAM design and build of prototype

An industry approved CAM (Computer Aided Manufacture) software system to develop a double-sided sustainable prototype PCB of the CUT that meets the customers, satisfies a wider engineering context and relevant to legal and aesthetic requirements. Firstly, the FPGA board was then implemented to the NI ELVIS board and again all three faults were displayed again as seen in figures 47, 48 and 49. Three faults were then generated, ASA1, BSA1 and CSA1 and the results were displayed on LabVIEW, see figures 50, 51 and 52. Secondly a software known as eagle was used as an industry approved CAM software. See figure 53 for the circuit design created on eagle. Gerber view was then used analyse each layer of the PCB circuit board. Gerber files are used by PCB manufacturers to translate the details of your design into the physical properties of the PCB. The Gerber files are typically generated by the PCB design software, although the process of doing so will vary with each CAD tool. The bottom and top layers are seen in figures 54 and 55. The mechanical board with the drill holes are displayed in figure 56. Figure 57 shows the board complete. Figures 58 and 59 show the bottom and top layers of the complete board. Figure 60 shows the completed board on the eagle software.

Gerber files are created and are ref in appendix

This explaantion is very confusing and the testing of the PCB and the production of the PCB using Eagle should be separate entities. Smaller pagraphs should be used to keep the content more organised.

7. Conclusions and Recommendations

Multi-sim word generator was used to drive CUT inputs and test with no fault conditions. Further tests for 3 SSF fault conditions taken from the test table, ASA1, BSA1 and CSA1. The output was then compared with a generated reference code by means of an XOR gate. The 6-bit pattern generator with no fault conditions and 3 fault condition (same faults). The tests should show the same results and as seen by comparing the two they are fairly similar. Discrepancies are down to accuracy of software and equipment. The results were as expected. LabVIEW with ELVIS II DAQ board – G code program was used to drive the 6 bit input patterns in an auto sequence which repeated after the end input condition. This provided a real time process with a visual interface. The no fault conditions and 3 fault conditions were recorded and the results were as expected.

Conclusion is too basic, however, there are no real unexpected outcomes, or reasons for testing inputs at specific points mentioned.

No recommendations are included

8. Citation List

Fall, 2021. *Fault models, fault collapsing and logic simulation*. [Online]

Available at: www.web.eecs.umich.edu/~mazum/F02/lectures/lec mazum3.pdf

[Accessed 16th January 2021].

Koren, I., 2021. *Defect tolerance in VLSI circuits*. [Online]

Available at: www.ieeeexplore.ieee.org/abstract/document/705525

[Accessed 16 January 2021].

Mack, A., 2020. *Semiconductor Lithography (Photolithography) - The Basic Process*.

[Online]

Available at: <http://www.lithoguru.com/scientist/lithobasics.html>

Mazumder, P., 2002. *Digital Testing*. [Online]

Available at: www.web.eecs.umich.edu/~mazum/F02/lectures/lec mazum3.pdf

[Accessed 17 January 2021].

Mini-circuits, A.-4.-0., n.d. *Prevention and Control of Electrostatic Discharge (ESD)*. [Online]

Available at: [https://www.pdfFiller.com/jsfiller-](https://www.pdfFiller.com/jsfiller-desk18/?requestHash=7cd75a7ef74f94fb093bb3a9f34a3f7878aa74769fd9c9e7866c4e169e1640fb&projectId=621272445&utm_source=PDFfiller_Edit_Fill_Sign_any_PDF_document&utm_medium=extension&utm_campaign=chrome#80a2ce175c63a7cd8cdaf)

[desk18/?requestHash=7cd75a7ef74f94fb093bb3a9f34a3f7878aa74769fd9c9e7866c4e169e1640fb&projectId=621272445&utm_source=PDFfiller_Edit_Fill_Sign_any_PDF_document&utm_medium=extension&utm_campaign=chrome#80a2ce175c63a7cd8cdaf](https://www.pdfFiller.com/jsfiller-desk18/?requestHash=7cd75a7ef74f94fb093bb3a9f34a3f7878aa74769fd9c9e7866c4e169e1640fb&projectId=621272445&utm_source=PDFfiller_Edit_Fill_Sign_any_PDF_document&utm_medium=extension&utm_campaign=chrome#80a2ce175c63a7cd8cdaf)

[Accessed 17 January 2021].

Mohammadat, M., 2015. *Resistive Open Faults Detectability Analysis*. [Online]

Available at: <https://eprints.soton.ac.uk/364712/1/manuscript.pdf>

Mohammadat, M., 2021. *Resistive Open Faults Detectability Analysis*. [Online]

Available at: www.eprints.soton.ac.uk/364712/1/manuscript.pdf

[Accessed 17 January 2021].

Plusquellic, J., 2004. *VLSI Design Verification and Test - Defects*. [Online]

Available at: http://ece-research.unm.edu/jimp/vlsi_test/slides/defects.pdf

Rouse, M., 2011. *Electronic discharge (ESD)*. [Online]

Available at: www.whatis.techtarget.com/definition/electrostatic-discharge-ESD

[Accessed 17 January 2021].

Storr, P., 1998. *Electro Static Discharge (ESD) Precautions*. [Online]

Available at: www.philipstorr.id.au/pcbook/book5/esd.htm

[Accessed 17 January 2021].

Toms, T. R., 2015. United States of America, Patent No. US20100091475A1.

Uber, R. J., 2021. *Fault Collapsing in Digital Circuits Using Fast Fault*. [Online]

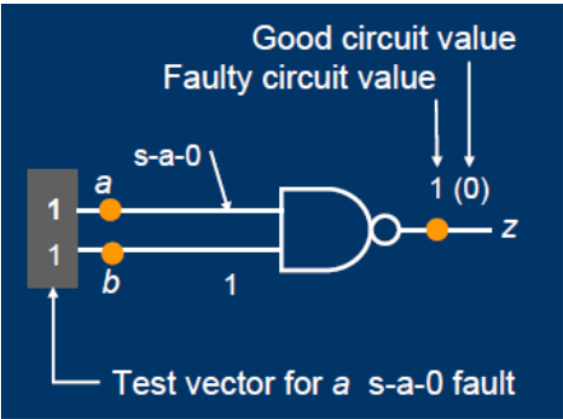
Available at: www.hal.inria.fr/hal-01578613/document

[Accessed 16 January 2021].

Good number of reference sources

9. Appendix

Figure 1 – Stuck at 0 fault model



Ref from report

This image should be referenced

Figure 2

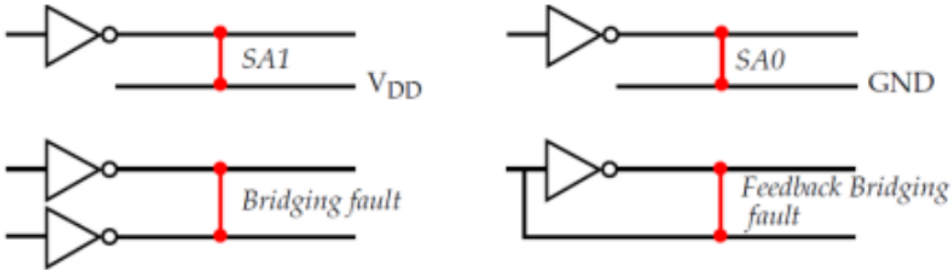
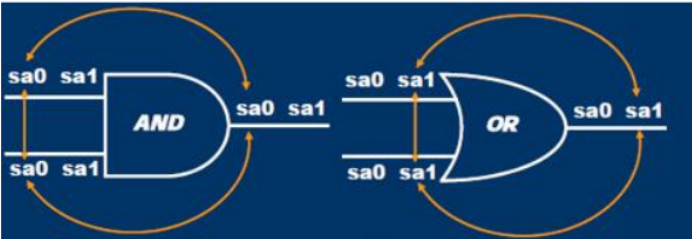


Image should be referenced

Figure 3 - Fault equivalent



Ref from report

Image should have a ref source

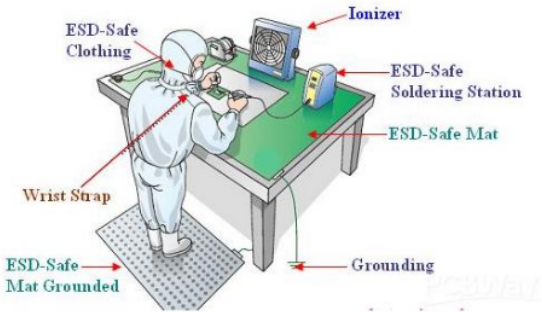
Figure 4 - ESD example sources and values

Source of ESD voltage	Typical Electro Static Voltage in KV at a relative humidity of 20%
Walking across Vinyl floor	12 KV
Walking across synthetic carpet	35 KV
Opening a plastic bag	20 KV
Arising from a foam cushion	18 KV
Sliding plastic box on a carpet	18 KV
Removing plastic tape for a PC board	12 KV
Removing shrink film from PC board	12 KV

Figure 5 - PCB damaged by ESD



Figure 6 - Preventative measures against ESD



The images should have ref sources

Figure 7 - CUT (Circuit Under Test)

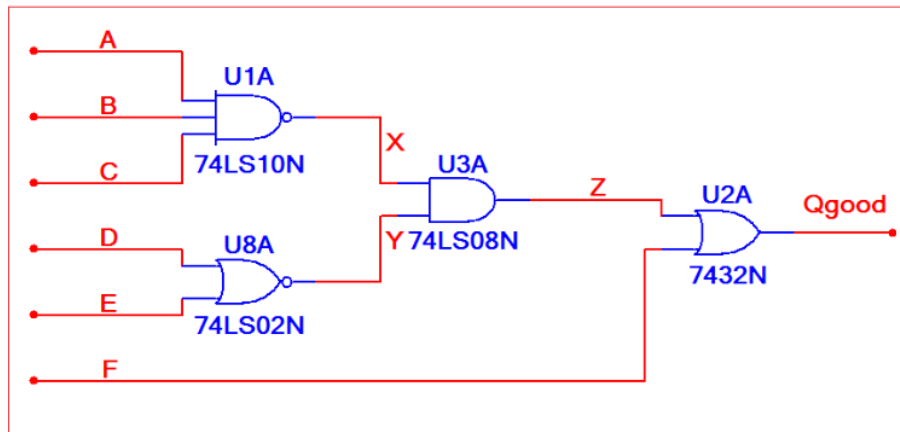


Figure 8 - Reduced tests - 3 input NAND gate

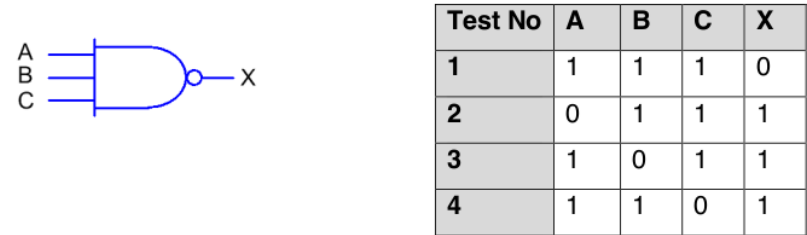


Figure 9 - Reduced tests - 2 input NOR gate

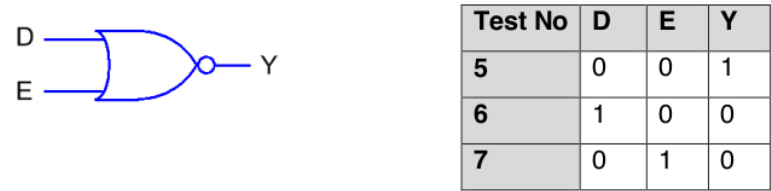


Figure 10 - Reduced tests - 2 input AND gate

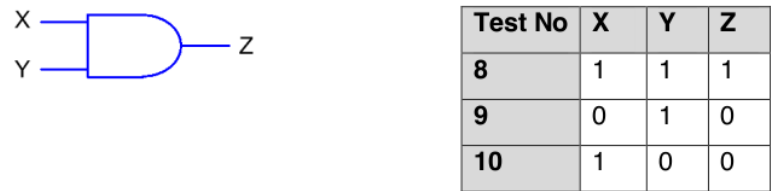
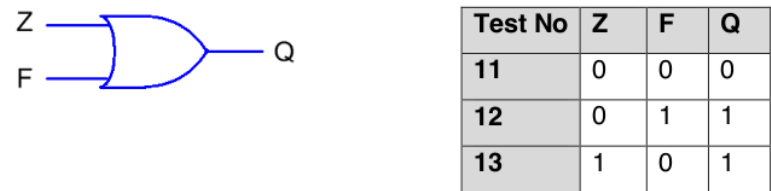


Figure 11 - Reduced tests - 2 input OR gate



Ref from report

SSF test vectors are created using theory

Figure 12 - Derived table

Test No	A	B	C	D	E	F	Q good	HEX	X	Y	Z	Fault
1	1	1	1	0	0	0	0	70	0	1	0	A+B+C SA0
2	0	1	1	0	0	0	1	31	1	1	1	A SA1
3	1	0	1	0	0	0	1	51	1	1	1	B SA1
4	1	1	0	0	0	0	1	61	1	1	1	C SA1
5	0*	0*	0*	0	0	0	1	01	1	1	1	D+E SA1
6	0*	0*	0*	1	0	0	0	08	1	0	0	D SA0
7	0*	0*	0*	0	1	0	0	04	1	0	0	E SA0
8	0*	0*	0*	0	0	0	1	01	1	1	1	X+Y SA0
9	1	1	1	0	0	0	0	70	0	1	0	X SA1
10	0*	0*	0*	1**	1**	0	0	0C	1	0	0	Y SA1
11	1	1	1	1**	1**	0	0	7C	0*	0*	0	F+Z SA1
12	1	1	1	1**	1**	1	1	7F	0*	0*	0	F SA0
13	0*	0*	0*	0	0	0	1	01	1	1	1	Z SA0

Key:

Initial table of test vectors is determined

* At least one condition must be at logic 0

Wildcard rules are identified and applied

** At least one condition must be at logic 1

Figure 13 - Reduced table

Test No	A	B	C	D	E	F	Qgood	HEX	X	Y	Z	Fault
1	1	1	1	0	0	0	0	70	0	1	0	A+B+C SA0, X SA1
2	0	1	1	0	0	0	1	31	1	1	1	A SA1
3	1	0	1	0	0	0	1	51	1	1	1	B SA1
4	1	1	0	0	0	0	1	61	1	1	1	C SA1
5	0*	0*	0*	0	0	0	1	01	1	1	1	D+E SA1, X+Y SA0, Z SA0
6	0*	0*	0*	1	0	0	0	08	1	0	0	D SA0
7	0*	0*	0*	0	1	0	0	04	1	0	0	E SA0
10	0*	0*	0*	1**	1**	0	0	0C	1	0	0	Y SA1
11	1	1	1	1**	1**	0	0	7C	0*	0*	0	F+Z SA1
12	1	1	1	1**	1**	1	1	7F	0*	0*	0	F SA0

Redundancies eliminated

Figure 14 - Further reduced table after applying wild card rules

Test No	A	B	C	D	E	F	Qgood	HEX	X	Y	Z	Fault
1	1	1	1	0	0	0	0	70	0	1	0	A+B+C SA0, X SA1
2	0	1	1	0	0	0	1	31	1	1	1	A SA1
3	1	0	1	0	0	0	1	51	1	1	1	B SA1
4	1	1	0	0	0	0	1	61	1	1	1	C SA1, D+E SA1, X+Y SA0, Z
6	0*	0*	0	1	0	0	0	08	1	0	0	D SA0, Y SA1
7	0*	0*	0	0	1	0	0	04	1	0	0	E SA0
11	1	1	1	1*	1*	0	0	7C	0*	0*	0	F+Z SA1
12	1	1	1	1*	1*	1	1	7F	0*	0*	0	F SA0

Further redundancies eliminated by applying wildcard rule

Figure 15 - Multisim test using Word Generator

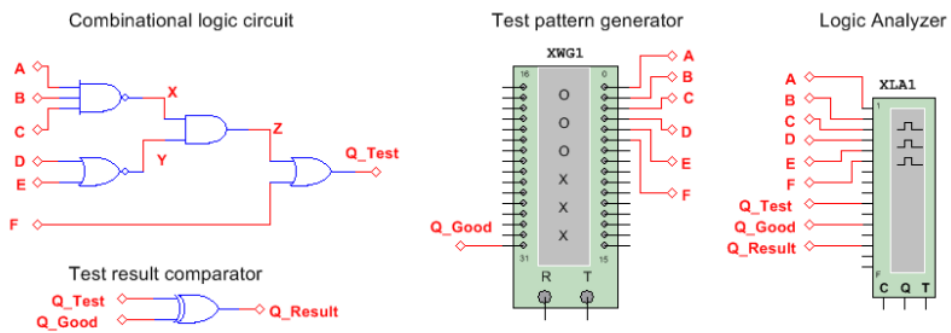
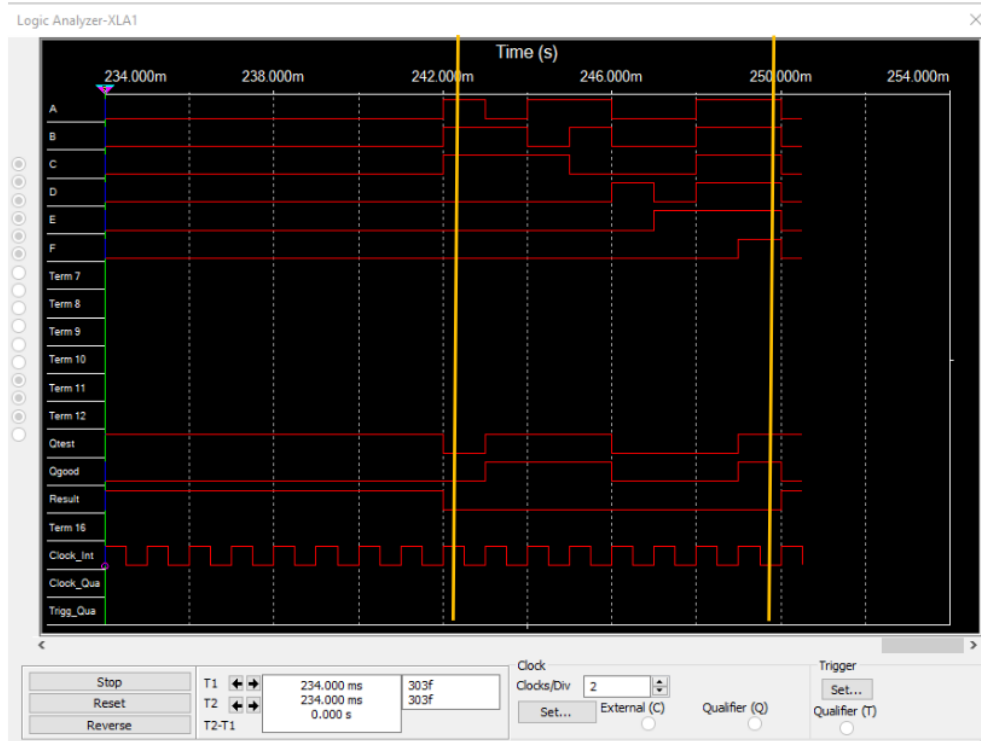


Figure 16 - Multisim result for no fault condition



NFF condition is achieved

Figure 17 - A SA1 Fault condition

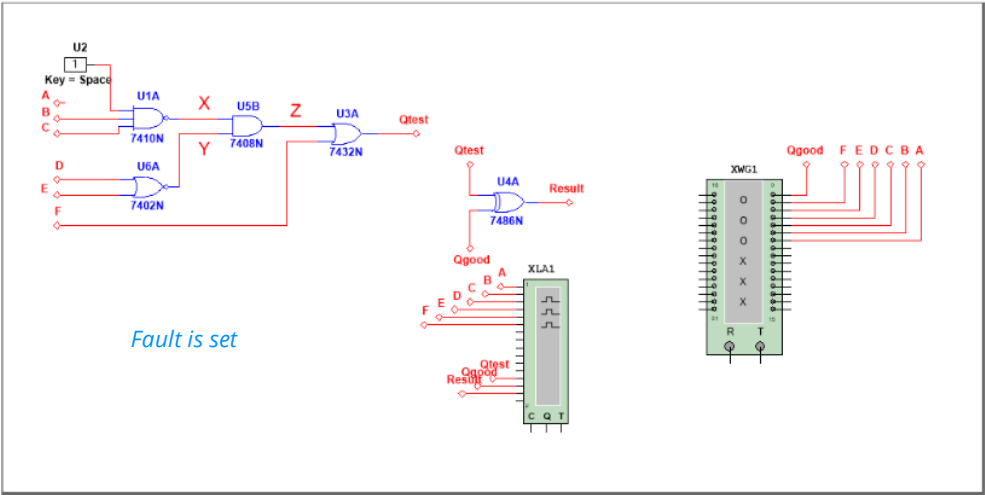


Figure 18 - Test result for input A SA1

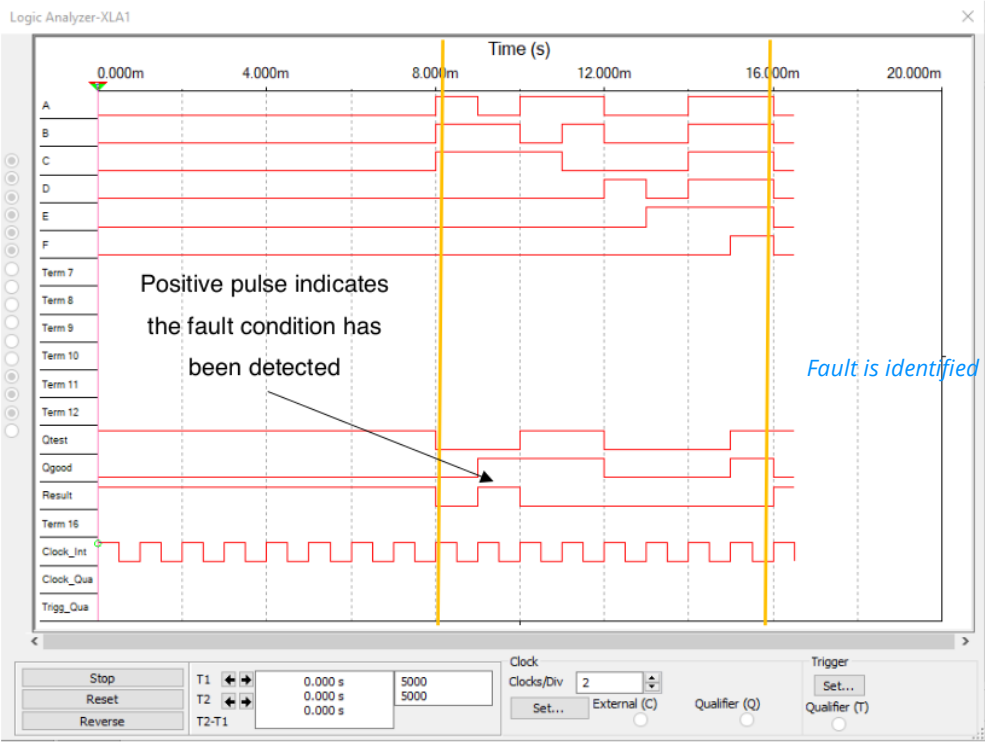


Figure 19 - B SA1 fault condition

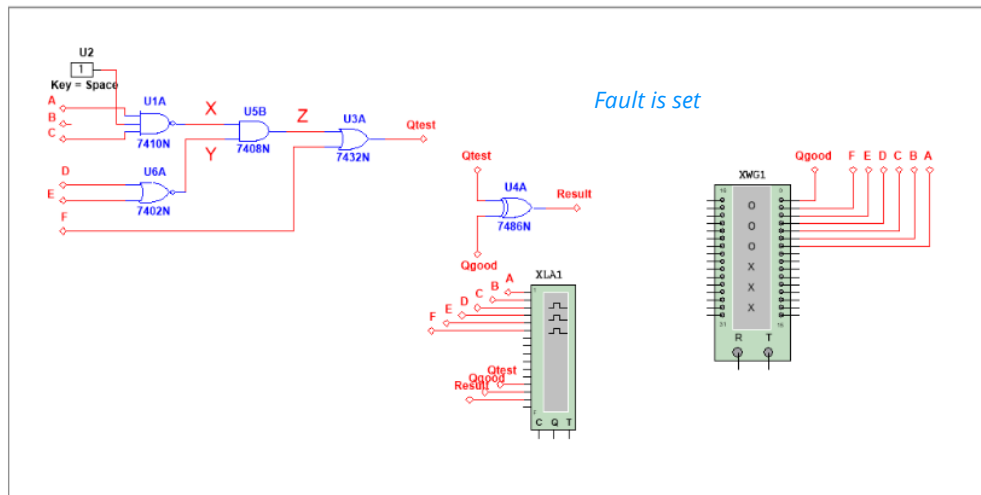
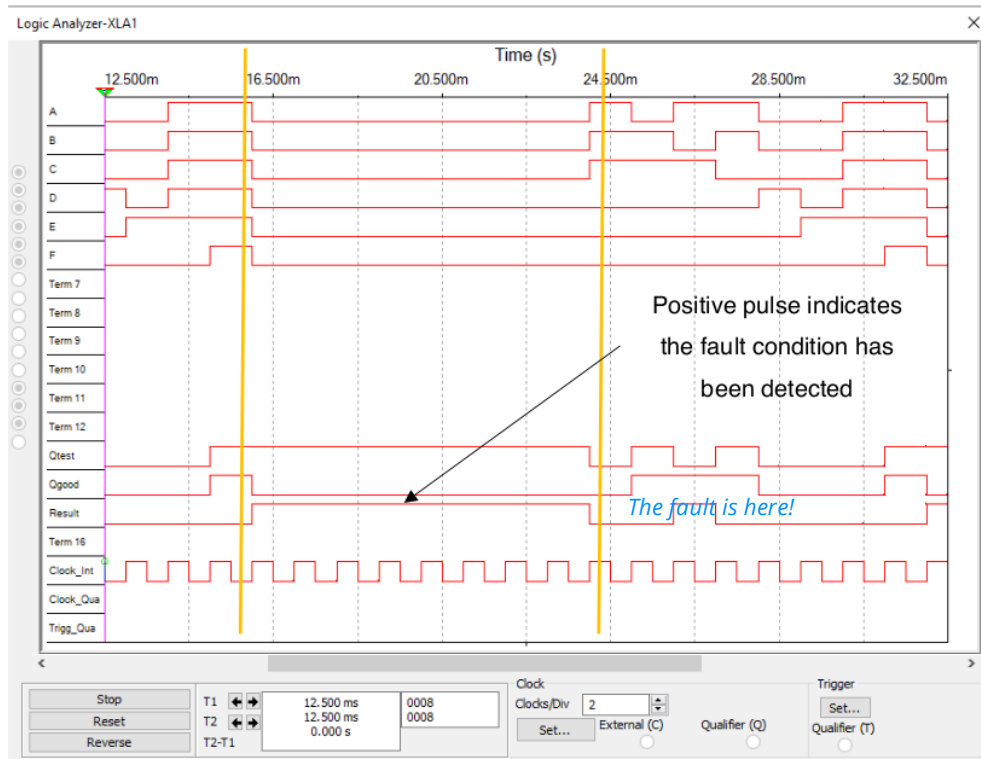


Figure 20 - Test result for input B SA1



Fault is identified at output, however the label is incorrect

Figure 21 - C SA1 fault condition

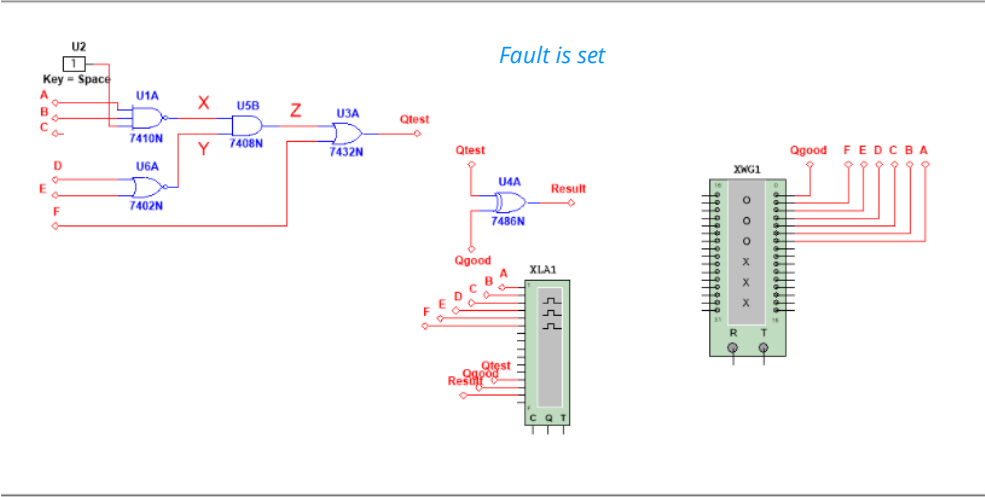


Figure 22 - Test result for input C SA1

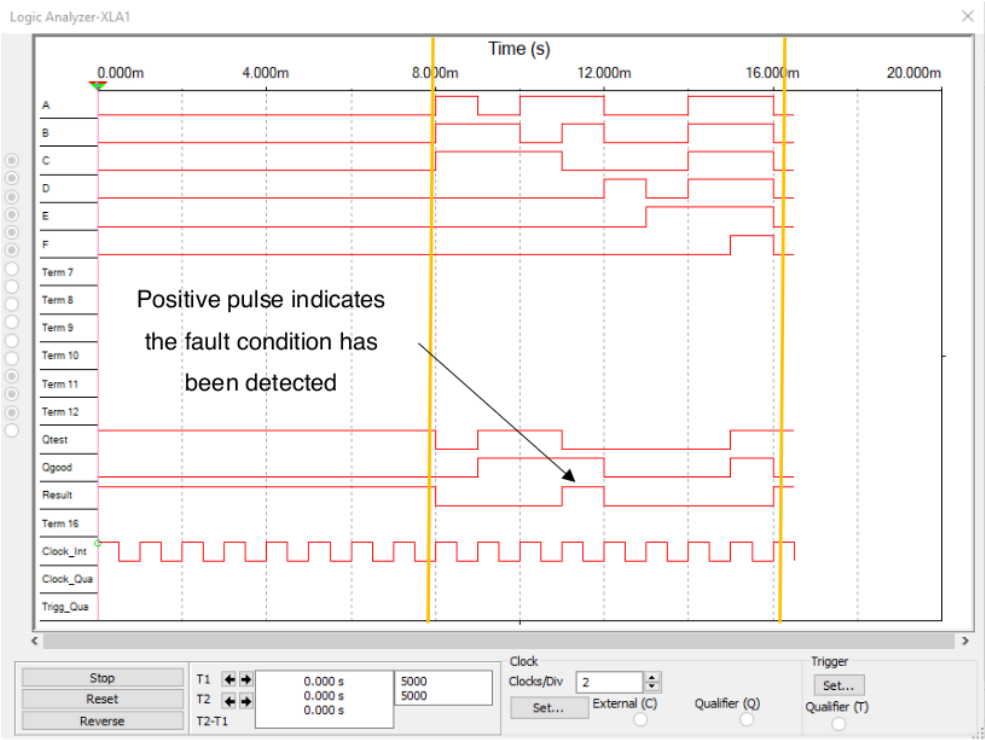


Figure 23 - Next state table

PRESENT STATE						NEXT STATE					
A	B	C	D	E	F	A+	B+	C+	D+	E+	F+
1	1	1	0	0	0	0	1	1	0	0	0
0	1	1	0	0	0	1	0	1	0	0	0
1	0	1	0	0	0	1	1	0	0	0	0
1	1	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	1	1	1	1	0
1	1	1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	0

Figure 24 - Output QA

Next state table is correct

ABC DEF	000	001	011	010	100	101	111	110
000	0	0	1	0	0	1	0	0
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	1	0
110	0	0	0	0	0	0	1	0

$$A.B.C.D.E + A.NB.C.ND.NE.NF + NA.B.C.ND.NE.NF + NA.NB.NC.ND.E.NF$$

Figure 25 - Output QB

ABC DEF	000	001	011	010	100	101	111	110
000	0	0	0	0	0	1	1	0
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	1	0
110	0	0	0	0	0	0	1	0

$$A.C.ND.NE.NF + A.B.C.D.E + NA.NB.NC.ND.E.NF$$

Figure 26 - Output QC

ABC DEF	000	001	011	010	100	101	111	110
000	0	0	1	0	0	0	1	0
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	1	0
110	0	0	0	0	0	0	1	0

$$NA.NB.NC.ND.E.NF + NA.B.C.ND.NE.NF + A.B.C.ND.NE.NF + A.B.C.D.E$$

Figure 27 - Output QD

ABC DEF	000	001	011	010	100	101	111	110
000	0	0	0	0	0	0	0	1
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	1	0

$$NA.NB.NC.ND.E.NF + A.B.NC.ND.NE.NF + A.B.C.D.E.NF$$

Figure 28 - Output QE

ABC DEF	000	001	011	010	100	101	111	110
000	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	1	0

$$NA.NB.NC.ND.E.NF + NA.NB.NC.D.NE.NF + A.B.C.D.E.NF$$

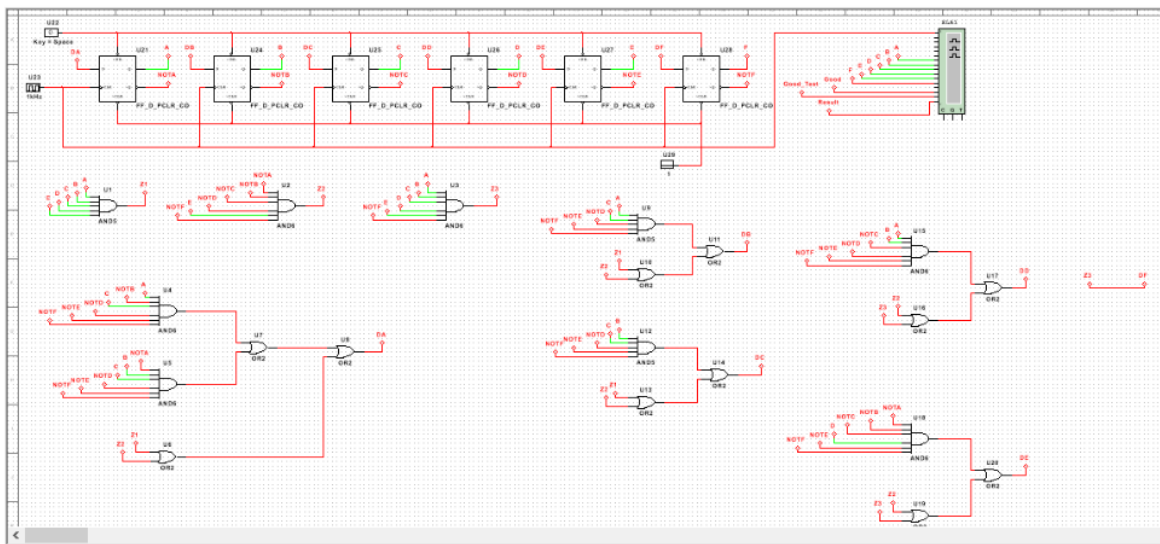
Figure 29 - Output QF

A B C	000	001	011	010	100	101	111	110
D E F	000	0	0	0	0	0	0	0
	001	0	0	0	0	0	0	0
	011	0	0	0	0	0	0	0
	010	0	0	0	0	0	0	0
	100	0	0	0	0	0	0	0
	101	0	0	0	0	0	0	0
	111	0	0	0	0	0	0	0
	110	0	0	0	0	0	1	0

K-maps and derived expressions are correct

A.B.C.D.E.NF

Figure 30 - 6-Bit Register with no fault condition



The two CUT's are missing here! The first circuit for the test cct, and the second for the good circuit used for comparison.

Figure 31 - Test result with no fault condition

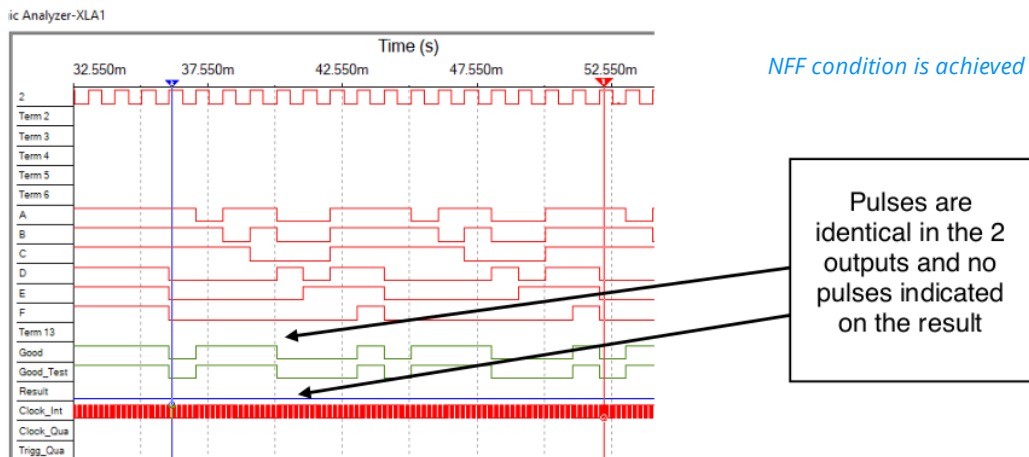


Figure 32 - CUT with input A SA1 fault condition

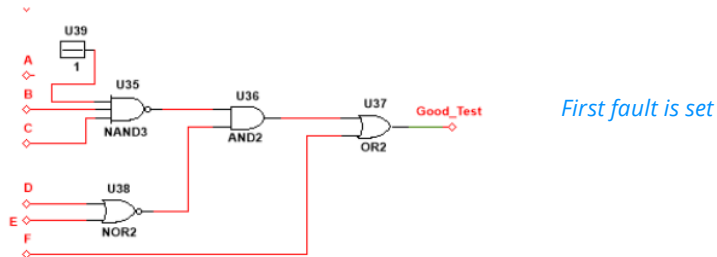


Figure 33 - Input A SA1 fault test result

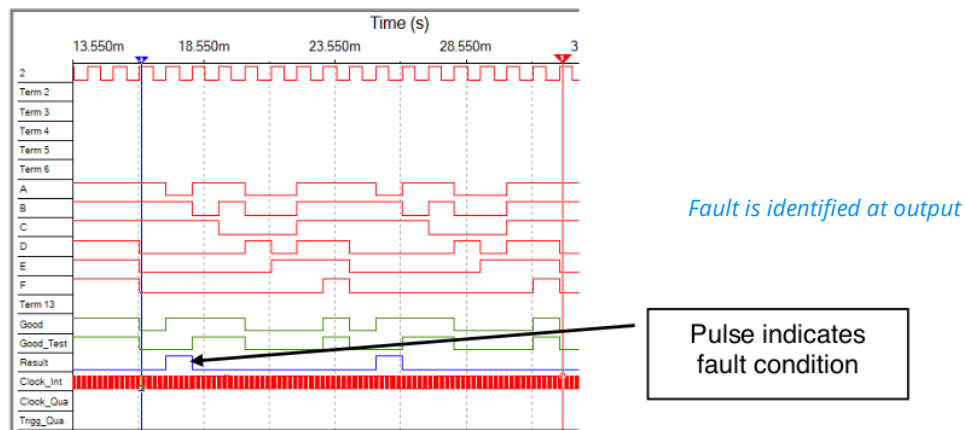


Figure 34 - CUT with pulse B SA1 fault condition

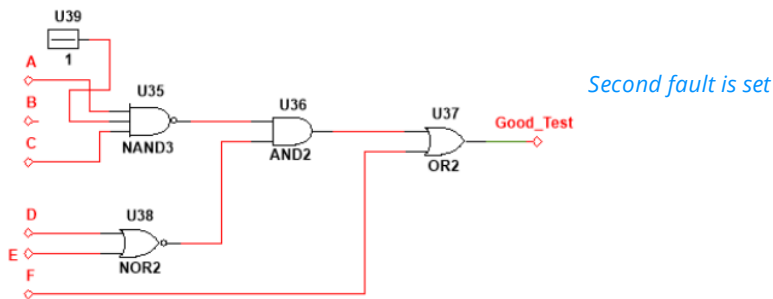


Figure 35 - Input B SA1 fault test result

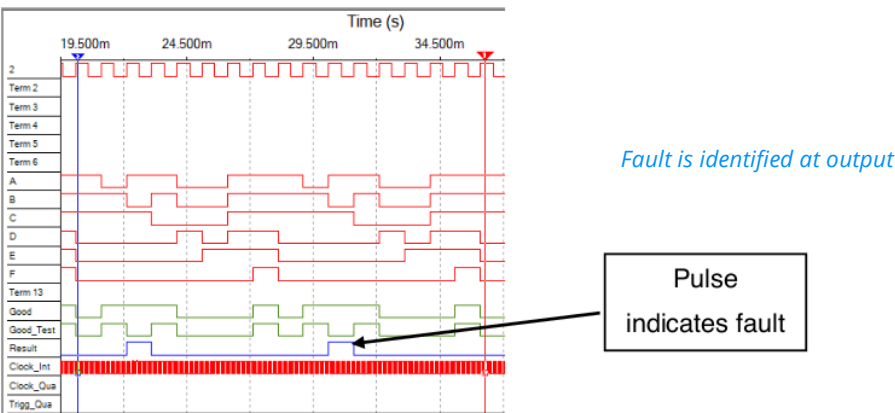


Figure 36 - CUT with input F SA1 fault condition

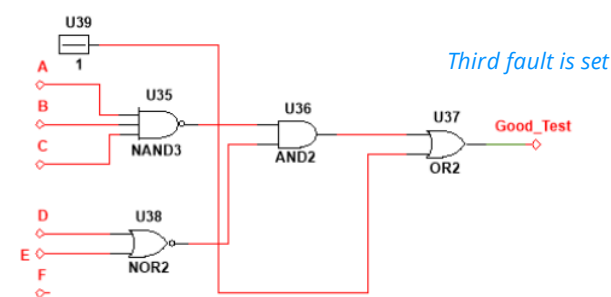
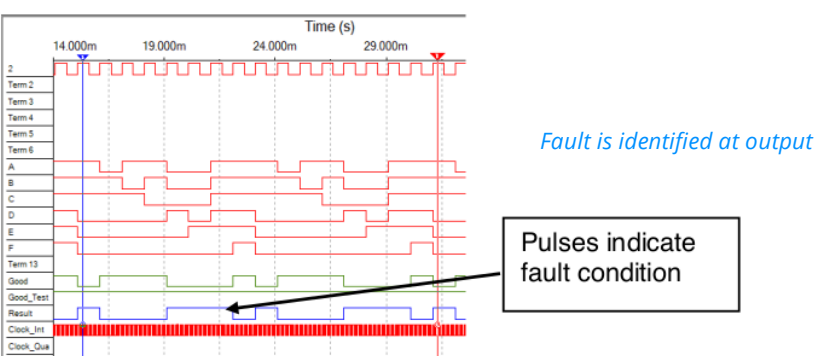


Figure 37 - Input F SA1 fault test result



Binary Data

Test Number

0

0

Digital conditions for inputs and outputs

0

Q F E D C B A

Fault Detected

Stop FOR Loop

Stop While Loop

STOP

STOP

Font panel has been ref
from report

The screenshot shows a LabVIEW block diagram titled "Digital conditions for inputs and outputs". The diagram is enclosed in a "While Loop" (indicated by a red stop sign icon in the bottom right corner). The main components and their connections are as follows:

- Inputs and Initialization:**
 - A constant "one channel for all lines" (blue arrow) is connected to the "Dev2/port0/line0" input of the "Digital U32 to 1Chan 1Samp" block.
 - A "Test Number" array (blue box) with values [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] is connected to the "Test Number" indicator (blue box) and the "Digital U32 to 1Chan 1Samp" block.
- Data Processing and Output:**
 - The "Digital U32 to 1Chan 1Samp" block outputs "Binary Data" (blue box) to a "Digital Output" indicator (blue box).
 - The "Binary Data" is also connected to a "Digital conditions for inputs and outputs" block (green box) which outputs "error out" (blue box) to an "error out" indicator (blue box).
- Control Flow:**
 - A "Stop FOR Loop" block (green box) is connected to the "error out" indicator.
 - A "Stop While Loop" block (green box) is connected to the "error out" indicator.

Block program is correct
and main function
explained

Figure 40 - NI LabVIEW

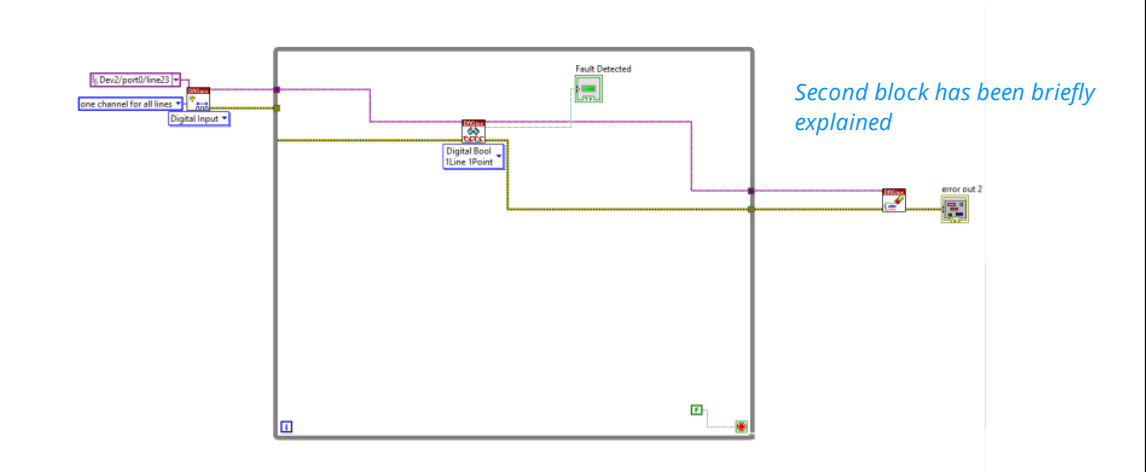


Figure 41 – NI LabVIEW ASA1

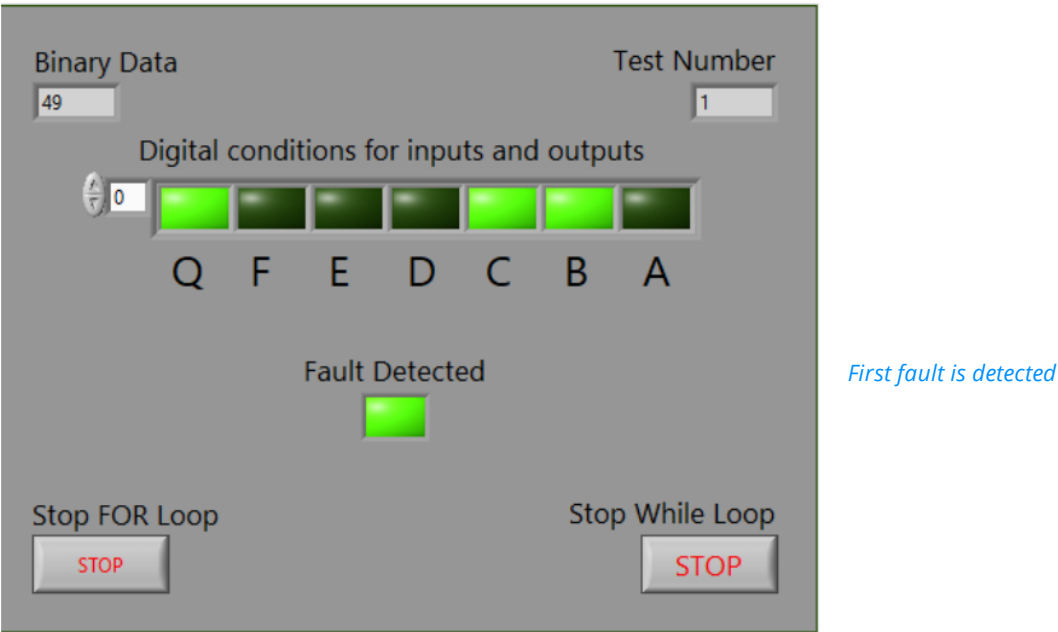
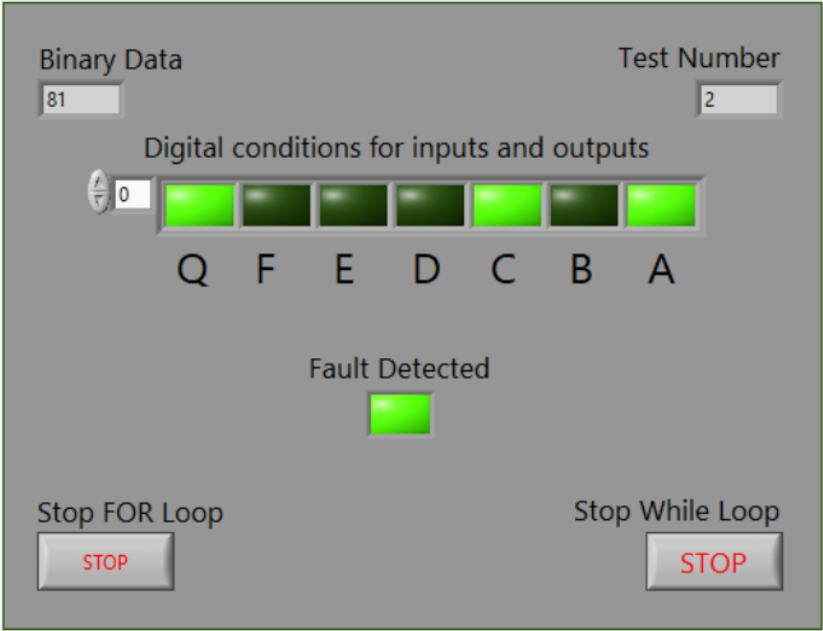
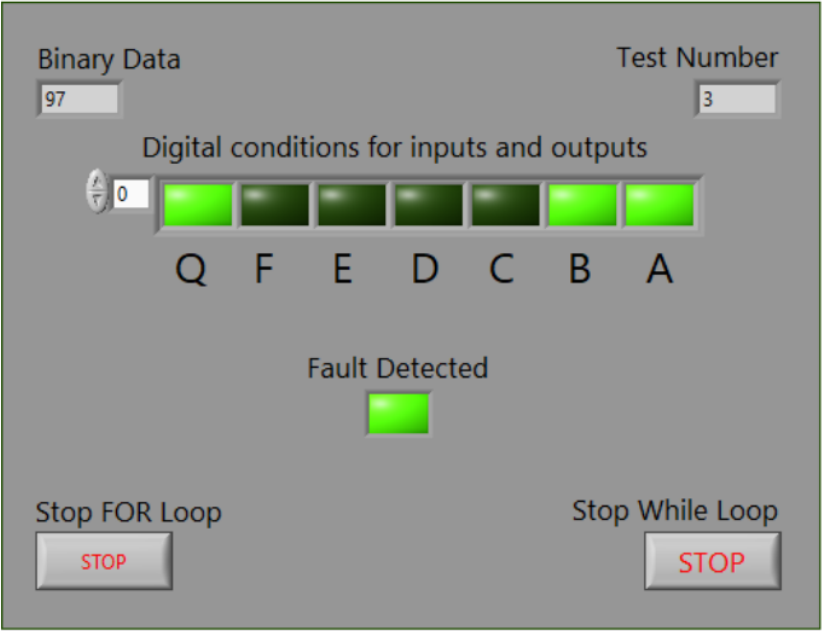


Figure 42 - NI LabVIEW BSA1



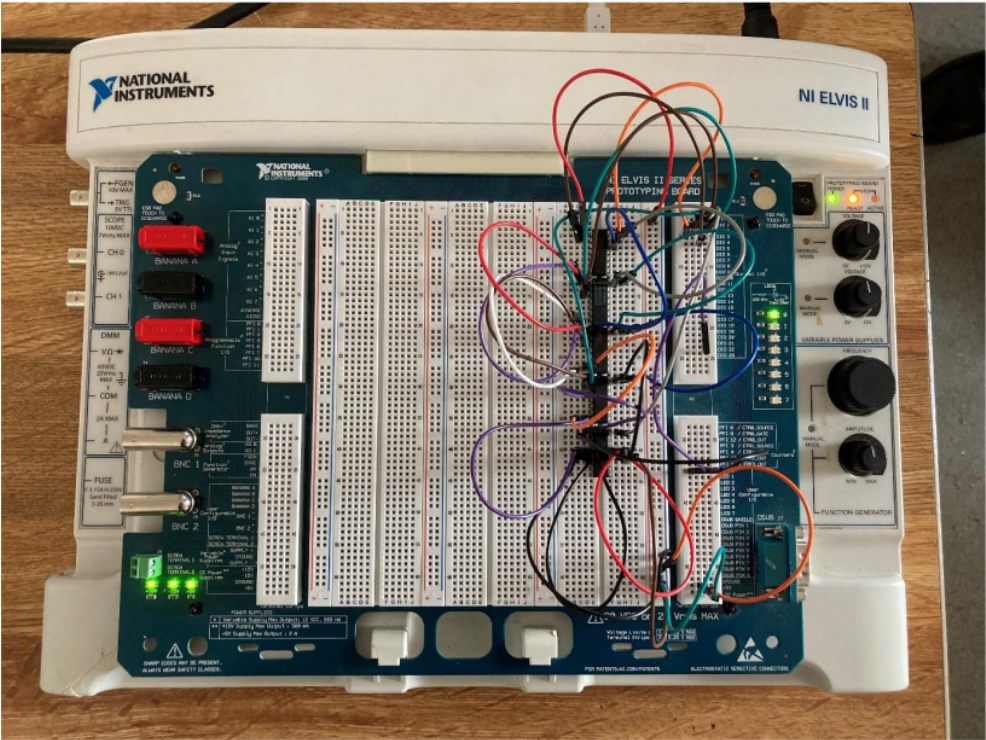
Second fault is detected

Figure 43 - NI LabVIEW CSA1



Third fault detected

Figure 44 - NI ELVIS ASA1



Fault
evidence
image

Figure 45 NI ELVIS BSA1

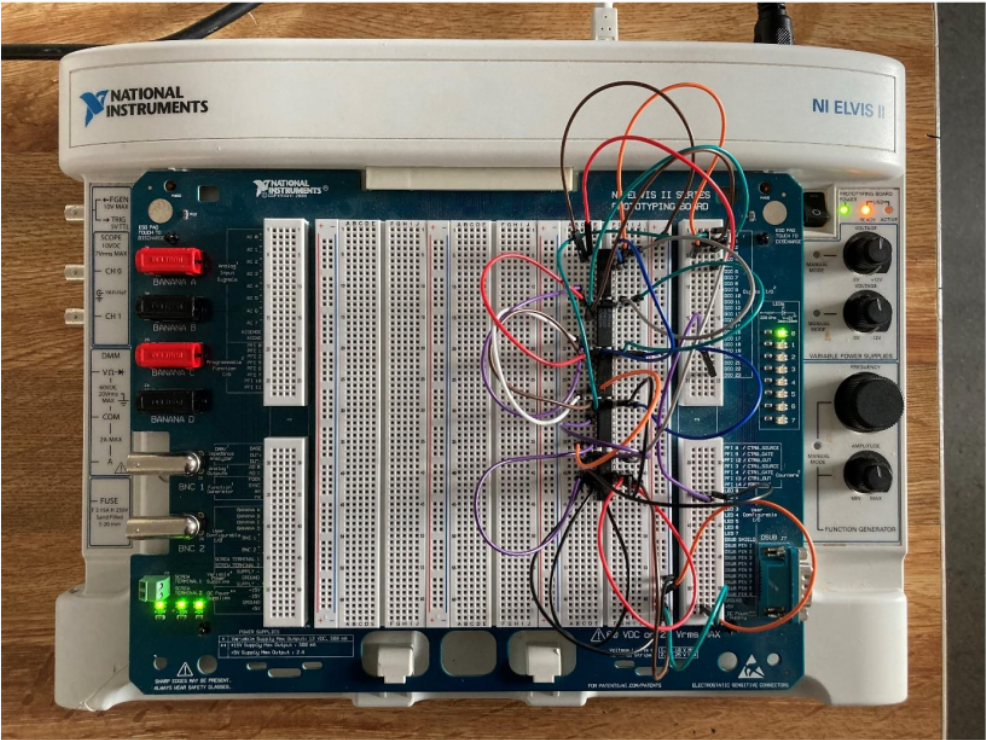
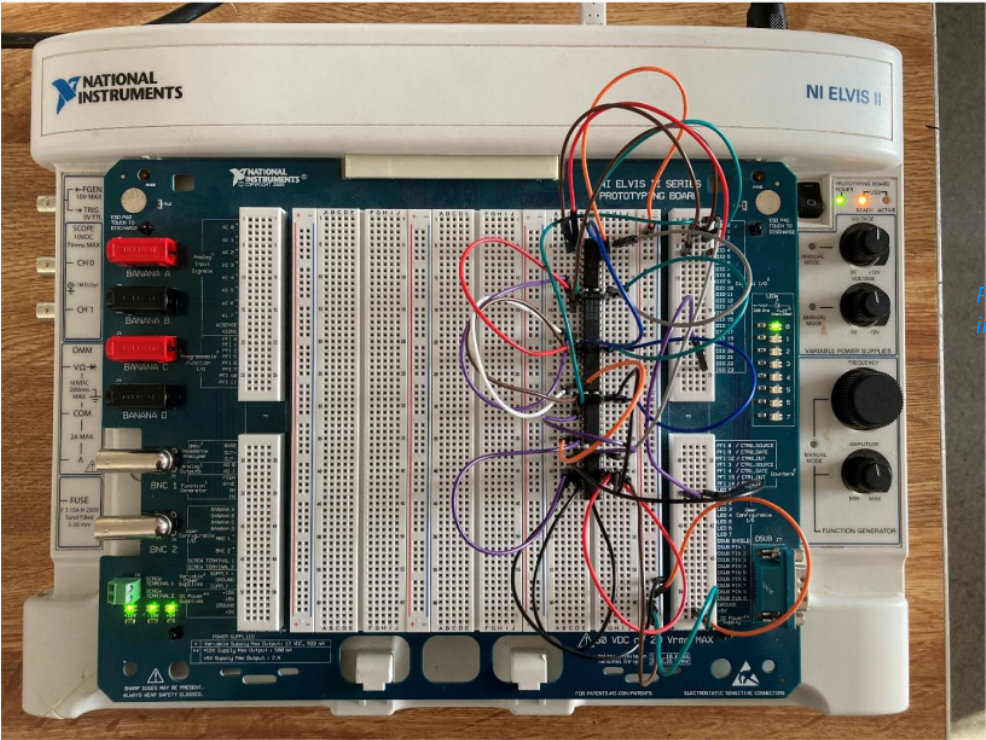
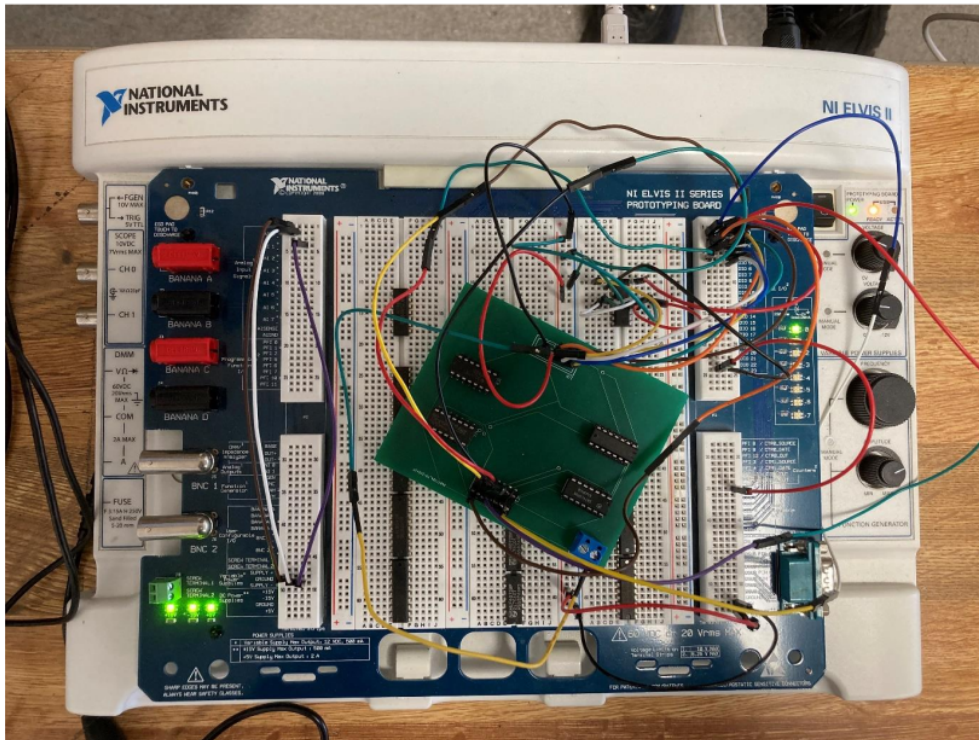


Figure 46 - NI ELVIS CSA1



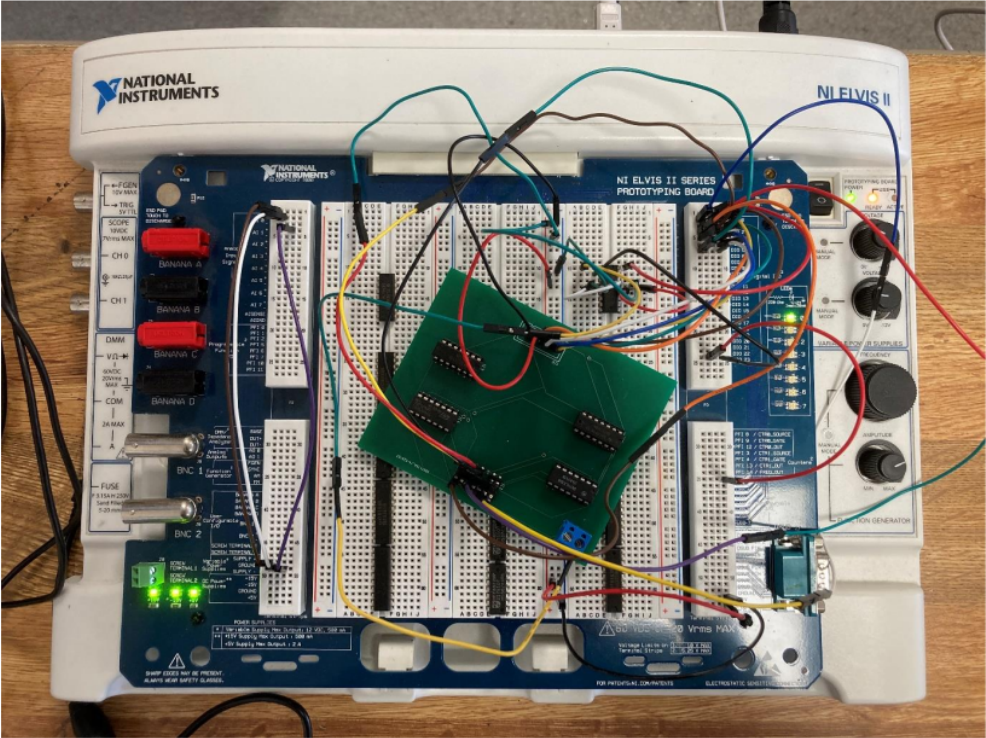
Fault evidence
image

Figure 47 - NI FPGA ASA1 *This is not a FPGA!*



*PCB test
evidence*

Figure 48 - NI FPGA BSA1



PCB test
evidence

Figure 49 - NI FPGA CSA1

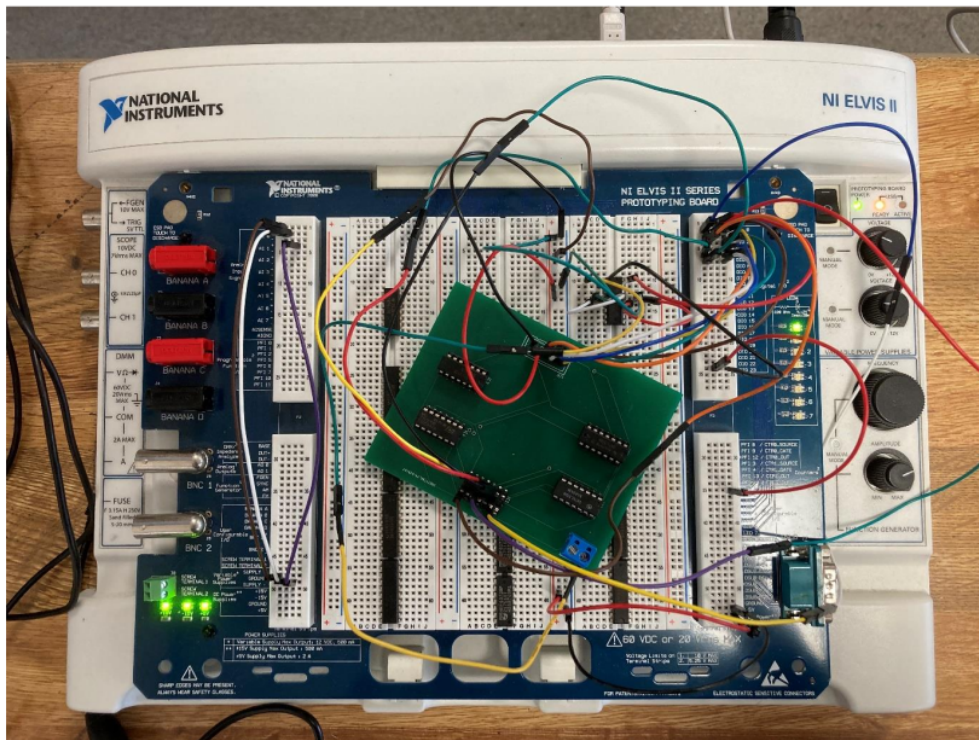


Figure 50 - PCB ASA1

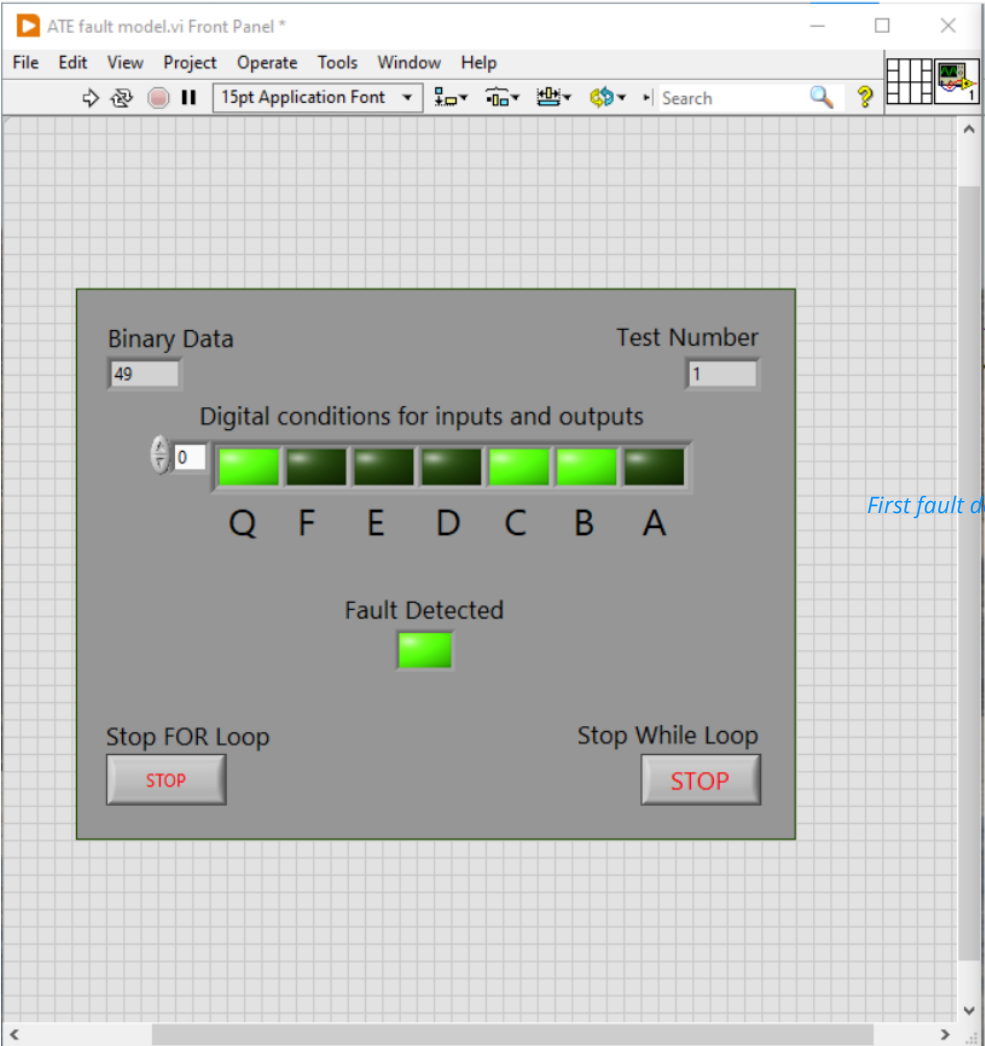


Figure 51 - PCB BSA1

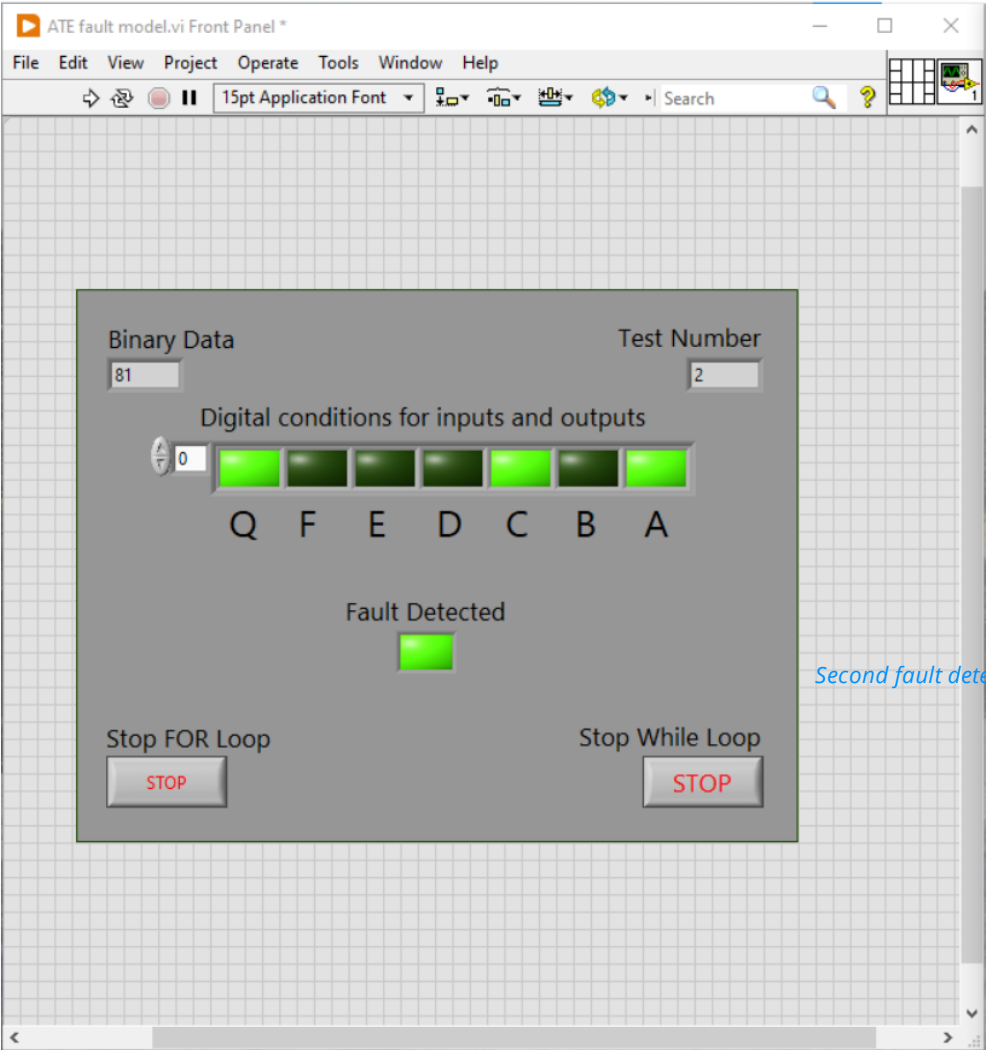


Figure 52 - PCB CSA1

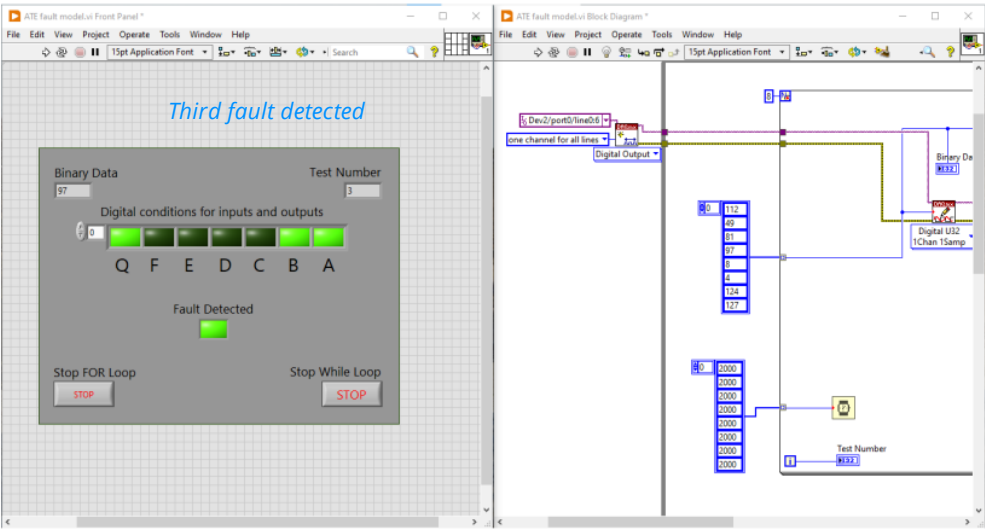
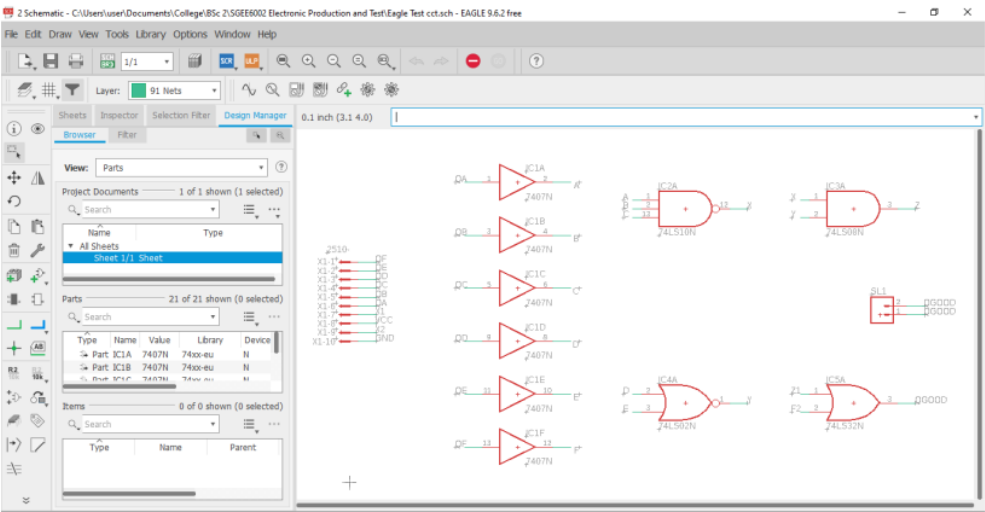


Figure 53 - Circuit design on Eagle



Schematic is correct

Figure 54 - Gerber view bottom layer

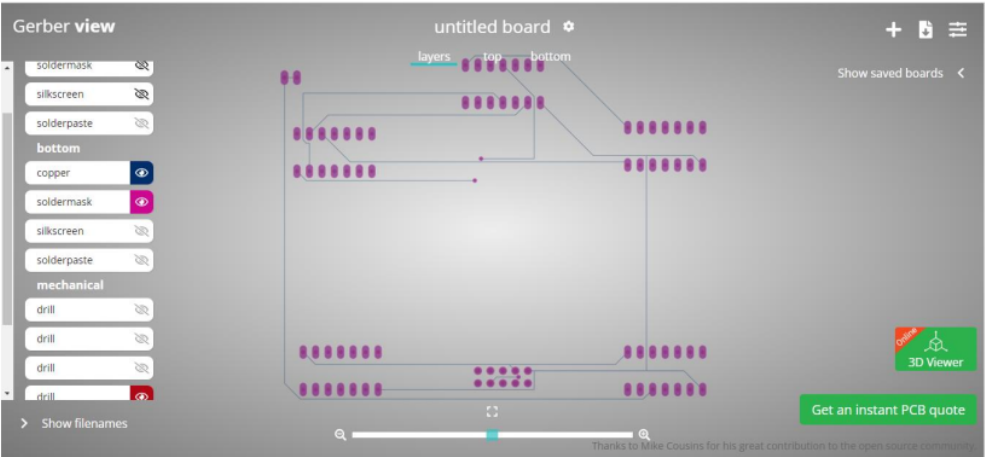


Figure 55 - Gerber view top layer

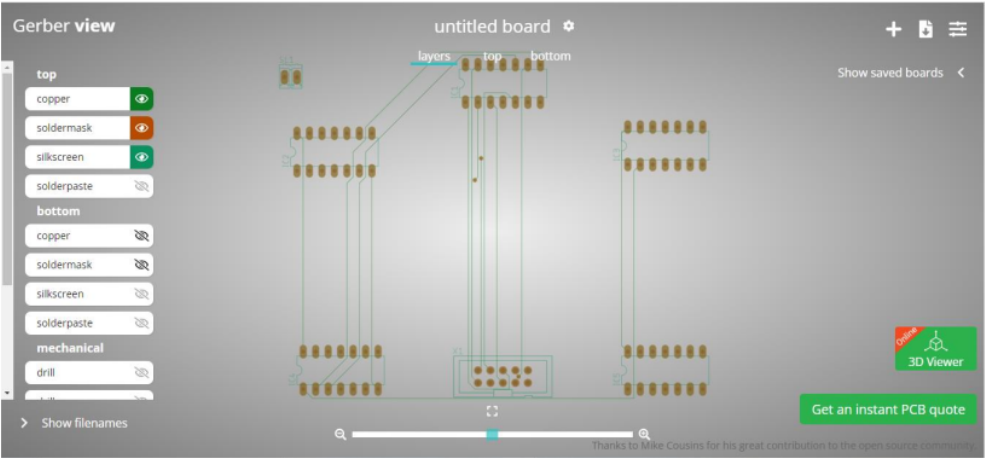


Figure 56 - Mechanical drill holes

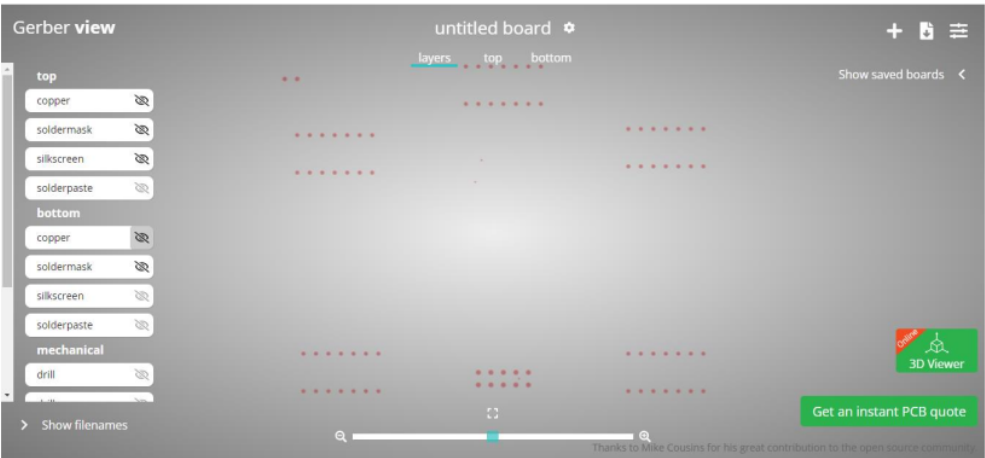


Figure 57 - Board complete

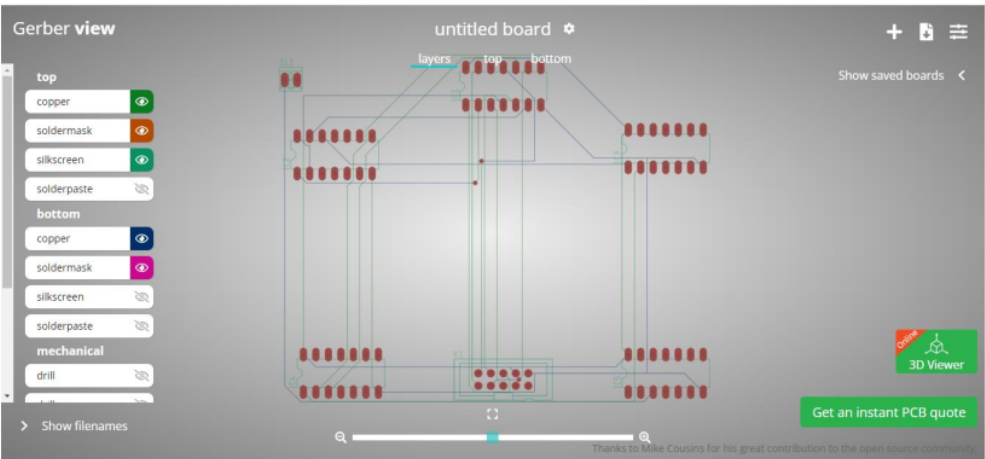


Figure 58 - Completed board - bottom layer

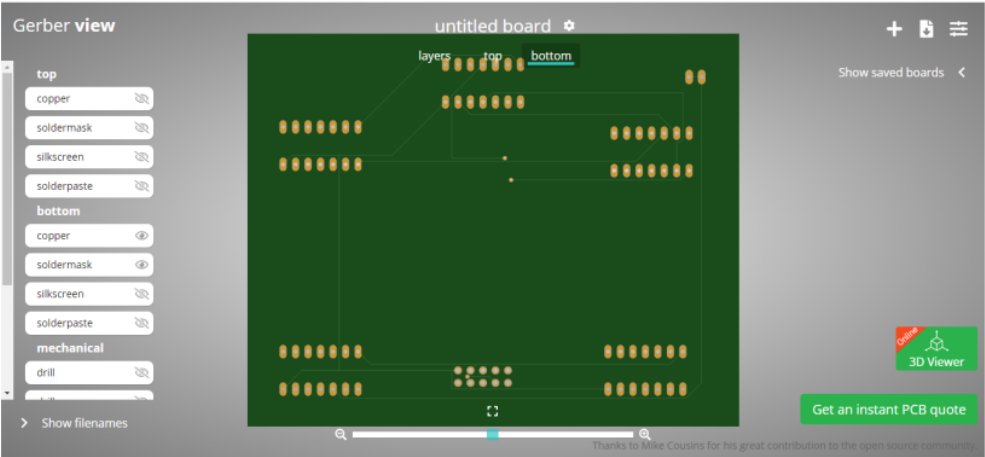
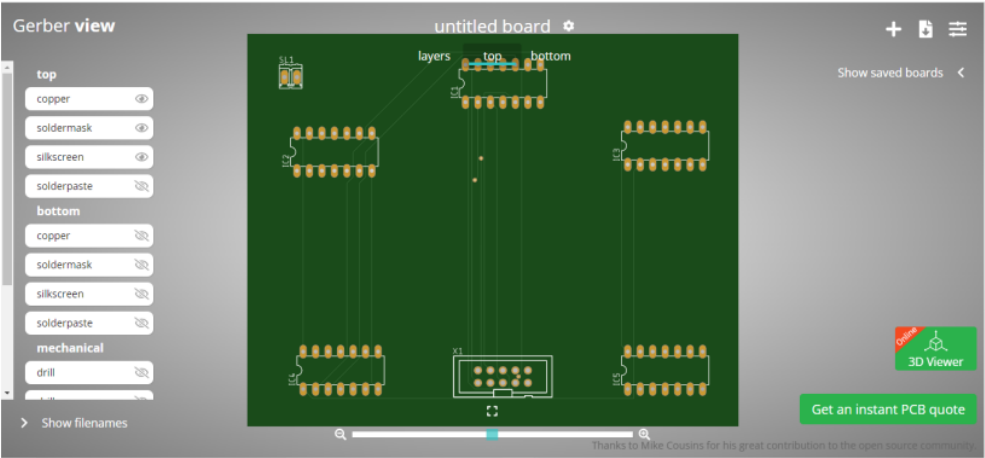
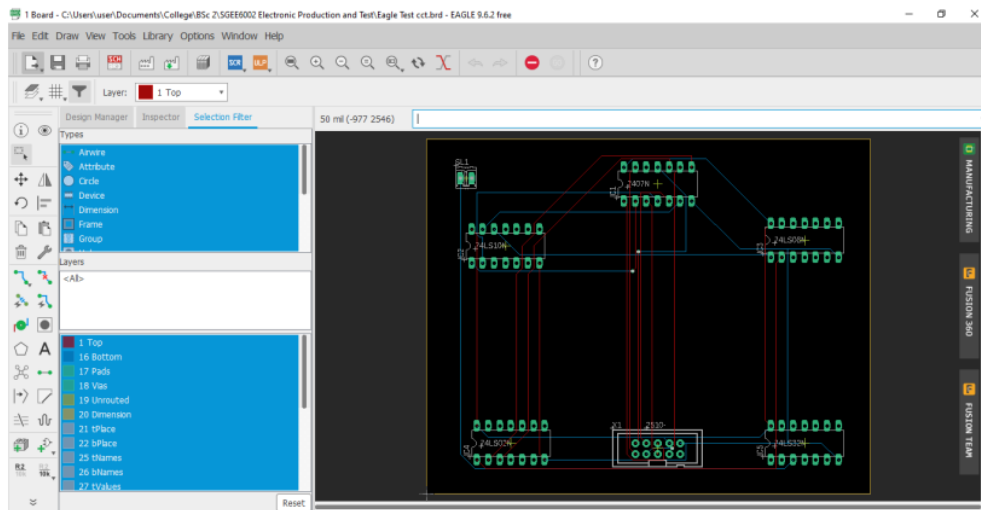


Figure 59 - Completed board - top layer



Gerber files are identified

Figure 60 - Completed board on eagle



PCB design is correct with low via count

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

Text Comment. Contents table should be more detailed for the number of tasks required.

PAGE 3

PAGE 4

Text Comment. Good list of figures

PAGE 5

Text Comment. There are good points summarised here, however, there are parts here that are unclear!

PAGE 6

Text Comment. This is a good point

Text Comment. Good point here with supporting ref source

Text Comment. Fault types are identified with supporting ref sources

PAGE 7

Text Comment. Ref in appendix

Text Comment. Ref in appendix

Text Comment. Good research with this topic

Text Comment. This area SSF is covered very well with the supporting ref sources and images in appendix.

PAGE 8

Text Comment. This should have a heading!

Text Comment. Ref image in appendix

Text Comment. Ref image in appendix

Text Comment. Good coverage of this topic and supported with relevant images and ref sources

Text Comment. There should be an investigation on wider modern methods for manufacturing a sustainable product.

PAGE 9

Text Comment. Image ref in appendix

Text Comment. SSF model used to create test vectors for each gate

Text Comment. Initial table is created

Text Comment. Redundancies are eliminated using SSF model rules

Text Comment. Ref in appendix

Text Comment. NFF condition achieved. This is critical before tests can be carried out

PAGE 10

Text Comment. Ref in appendix

Text Comment. Ref in appendix

Text Comment. Ref in appendix

Text Comment. There is an error in one of the tests here. See figure 20, B- SA1.

Text Comment. Ref in appendix

Text Comment. Ref in appendix

Text Comment. Expressions have been determined from K-maps

Text Comment. The good cct and CUT are both missing from figure 30. The X-OR gate is also missing

PAGE 11

Text Comment. Tests for all three fault conditions have proven successful

PAGE 12

Text Comment. Good explanation

Text Comment. This is correct

Text Comment. This could have been written using an improved method i.e. using smaller paragraphs and separating the test procedures.

Text Comment. All the tests are present with evidence. There are minor errors with the heading containing FPGA. There are no FPGA boards used!

PAGE 13

Text Comment. Error here! There is no FPGA

Text Comment. PCB test ends here

Text Comment. This is the start of the PCB design using Eagle software

Text Comment. Gerber files are created and are ref in appendix

Text Comment. This explaantion is very confusing and the testing of the PCB and the production of the PCB using Eagle should be separate entities. Smaller pagraphs should be used to keep the content more organised.

PAGE 14

Text Comment. Conclusion is too basic, however, there are no real unexpected outcomes, or reasons for testing inputs at specific points mentioned.

Text Comment. No recommendations are included

PAGE 15

PAGE 16

Text Comment. Good number of reference sources

PAGE 17

Text Comment. Ref from report

Text Comment. This image should be referenced

Text Comment. Image should be referenced

PAGE 18

Text Comment. Ref from report

Text Comment. Image should have a ref source

Text Comment. The images should have ref sources

PAGE 19

PAGE 20

Text Comment. Ref from report

Text Comment. SSF test vectors are created using theory

PAGE 21

Text Comment. Initial table of test vectors is determined

Text Comment. Wildcard rules are identified and applied

PAGE 22

Text Comment. Redundancies eliminated

Text Comment. Further redundancies eliminated by applying wildcard rule

PAGE 23

Text Comment. NFF condition is achieved

PAGE 24

Text Comment. Fault is set

Text Comment. Fault is identified at output

PAGE 25

Text Comment. Fault is set

Text Comment. The fault is here!

Text Comment. Fault is identified at output, however the label is incorrect

PAGE 26

Text Comment. Fault is set

Text Comment. Fault is identified at output

PAGE 27

Text Comment. Next state table is correct

PAGE 28

PAGE 29

Text Comment. K-maps and derived expressions are correct

Text Comment. The two CUT's are missing here! The first circuit for the test cct, and the second for the good circuit used for comparison.

PAGE 30

Text Comment. NFF condition is achieved

Text Comment. First fault is set

Text Comment. Fault is identified at output

PAGE 31

Text Comment. Second fault is set

Text Comment. Fault is identified at output

PAGE 32

Text Comment. Third fault is set

Text Comment. Fault is identified at output

PAGE 33

Text Comment. Font panel has been ref from report

Text Comment. Block program is correct and main function explained

PAGE 34

Text Comment. Second block has been briefly explained

Text Comment. First fault is detected

PAGE 35

Text Comment. Second fault is detected

Text Comment. Third fault detected

PAGE 36

Text Comment. Fault evidence image

PAGE 37

PAGE 38

Text Comment. Fault evidence image

PAGE 39

Text Comment. This is not a FPGA!

Text Comment. PCB test evidence

PAGE 40

Text Comment. PCB test evidence

PAGE 41

Text Comment. PCB test evidence

PAGE 42

Text Comment. First fault detected

PAGE 43

Text Comment. Second fault detected

PAGE 44

Text Comment. Third fault detected

Text Comment. Schematic is correct

PAGE 45

PAGE 46

PAGE 47

Text Comment. Gerber files are identified

PAGE 48

Text Comment. PCB design is correct with low via count

SEAN PARRY

0

RESEARCH-INFORMED LITERATURE

 Good research carried out with SSF and ESD, however the wider modern methods are not covered. A good number of relevant sources have been investigated on the topics covered.

12/20

KNOWLEDGE AND UNDERSTANDING

 A good understanding of the test vectors has been demonstrated and the use of SSF methods. 3.5/5 Labview design is correct and includes all test vectors. The explanation includes good significant details, however, the content is merged with PCB design and therefore confusing in parts. Minor errors with confusing FPGA in explanations. 3/5 ECAD alternatives have been implemented and explained. 3.5/5 CAM design is complete and explained. Gerber files are present. Some minor errors with FPGA in explanation. 3/5

13/20

ANALYSIS

 ECAD analysis includes good detail of test results. There is an error with one test. The explanations could have been written more clearly. 7/10 Test results confirm the theory. Three faults have been

12/20

set to check consistency. 3.5/5 Conclusion is too basic with no recommendations. 1.5/5

PRACTICAL APPLICATION AND DEP.

 PCD prototype is covered including Gerber files. The explanation is not very clear and confusion with FPGA is present. 5.5/10 Final prototype tests have been carried out successfully. This should have been explained separate to the PCB design, as two different types of software are used, and are completely separate tasks. 5.5/10

11/20

SKILLS FOR PROFESSIONAL PRAC.

 Extensive use of software is evident in the report. 6/10 The report layout could have been more professional using improved methods of writing. The sequence of tasks was confusing in parts with some minor errors. 2.5/5 word count is within limits. 3.5/5

12/20