1708165

by Adam Bedner

GOOD RESEARCH, REFERENCE LIST WAS APPROPRIATE AND CORRECT CITATIONS USING HARVARD. PRACTICAL WORK WAS WELL EXPLAINED, BUT RESULTS COULD HAVE BEEN ANALYSED MORE CRITICALLY.

CONCLUSIONS WERE SHORT AND THERE WERE NO RECOMMENDATIONS, SO SUGGEST 2 MARKS LESS.

MARK SUGGESTED AT 73%.

FILE	13694_ADAM_BEDNER_1708165_172472_289263740.DOCX (5.76M)		
TIME SUBMITTED	06-MAR-2021 11:35PM (UTC+0000)	WORD COUNT	4900
SUBMISSION ID	146202918	CHARACTER COUNT	21409

Electronic Production & Test

Assignment 1 - Fault Modelling

Student name: Adam Bedner

Date: 27th February 2021

Word Count: 2700

Contents

1. Introduction 3
2. Research 3
   2.1 Stuck at faults. 3
   2.2 Electrostatic discharge (ESD) 6
   2.3 Preventive measure against ESD 7
3. Design Requirements 8
   3.1 Stuck at Fault Modelling 8
   3.2 Test vectors test pattern 10
   3.3 Multisim design tests using Word Generator. 12
4. Test Analysis Using ECAD 13
5. 6-Bit Register Design and Test 16
   5.1 Analysis of 6-Bit Register Fault Conditions. 20
   5.2 Implementation and practical testing on NI ELVIS board. 22
6. PCB Design 27
Conclusions 30
Table of figures 31
References 33

NO EXECUTIVE SUMMARY.

1. Introduction

The aim of the assignment is to carry out an analysis and evaluation of a range of processes and techniques required for developing a prototype design of a product. This can then be manufactured and tested in order to meet customer needs.

This will be achieved by the following tasks:

- Demonstrate the use of ATE in testing IC prototype designs using the SSF model
- Researching the dangers of ESD and how to implement preventive measures during design, manufacture and test processes
- Create logic circuits for test purposes using ECAD
- Perform vector reduction using SSF model
- Derive test results from the model using both ECAD simulation and NI LabVIEW
- Analysing and comparing the derived and simulated test results
- Design of 6 – bit pattern generator ATE to test circuit
- Design and implement the pattern generator into LabVIEW for testing of practical circuit
- Create a manufactured PCB practical design using ECAD technology
- Test the practical manufactured PCB using LabVIEW pattern generator program

2. Research

The following research has been carried out to understand and complete the tasks presented by this assignment.

2.1 Stuck at faults.

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 said devices. An IEEE journal states that the severity of imperfections in the fabrication process grows proportionally with the size and density of the chip. *(Koren et al, 1998)*

IC fabrication faults cover a range of fault modelling techniques which make possible test and analysis of integrated circuits.

Categories of IC fabrication faults are:

Gate level stuck at fault - occurs when a logic gate has an input stuck at 1 or 0 which effects the logical output of the device.

Transistor level stuck at fault – describes a transistor as a switch either stuck open or closed.

Bridging fault – describes various open or short circuit faults between connections within a circuit. A short between two wires is commonly referred to as a bridge fault.

Delay fault – describes the fault being with resistive open / short circuit which can cause a signal to propagate so slow that the time specification is not reached.

Resistive open faults represent common manufacturing defects in integrated circuits and result in delay faults that cause timing failures and reliability issues as that can cause the chips to become unstable or to crash if there is software involved. The delay faults need to be detected by speed testing the silicon chip by overclocking them (speed faster than the required speed), sometimes also referred to as stress test. *(Mohammadat, 2015)*

IC fabrication faults include fault modelling techniques for common faults such as those mentioned above. *(Fall 2002)*

There are three properties which define a single stuck at fault:

1. Only one line or wire is faulty

2. The faulty line is permanently set to zero or one

3. The fault can be at the input or the output of a logic gate *(Fall, 2002)*

Fig. 1 shows an example of determining a test vector for a stuck at fault of a NAND gate with input 'a' stuck at 0. The fault can be detected by applying the test vector '11' to the gate. This will cause the output of the gate to change from 0 to logic 1.

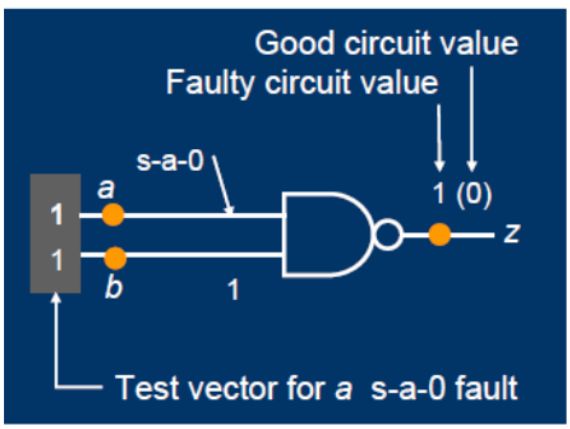


Fig. 1 Stuck at 0 fault model (Kastensmidt, 2008)

The main stuck at faults that can be modelled are:

SA1 – Stuck at logic 1

SA0 – Stuck at logic 0

Bridging fault – short circuit between two gates or conductors

Feedback bridging fault – short circuit between an output and feedback of a gate

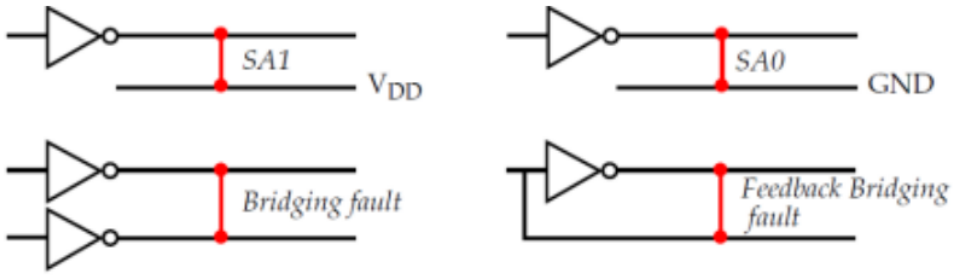


Fig. 2 examples of faults (Plusquellic, 2004)

The diagrams in *Fig. 2* demonstrate the stuck at fault categories. Delay faults can also occur which are caused by added resistance in the line. As mentioned earlier this causes added propagation delay.

Fig. 3 shows the fault equivalent method applied to a two input AND gate and a two input OR gate. When one input is in SA0 condition; the AND gate's output will be 0. Therefore, two of the three faults joined with the arrows can be removed.

Similar method is applied to the OR gate; in both cases the fault equivalent technique reduces the number of fault vectors.

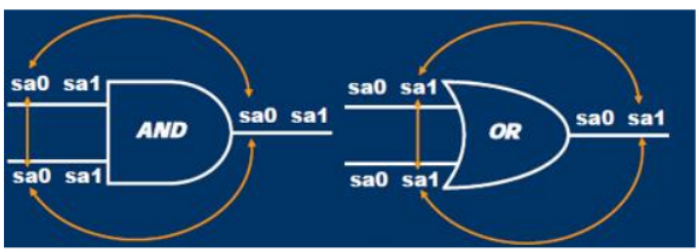


Fig. 3 Fault equivalent diagrams (Kastensmidt, 2002)

Fault equivalence is when two faults are equally dominant and only one test needs to be performed to detect either. *(Ubar, et al, 2017)*

The single stuck at fault model can be produced to detect faults using a known i/o table to compare with expected output results. Statistically single fault tests cover a very large number of multiple faults which means a large majority of the fault tests can be removed using the fault collapse method. *(Mazumder, 2002)*

2.2 Electrostatic discharge (ESD)

Electrostatic 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 can cause 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 will sustains damage. Said discharges are powerful enough to damage any delicate silicon based components. *(Toms, et al, 2015)*

The damage cause by ESD is more significant as modern IC's become smaller 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 *Table 1*:

Source of ESD Charge	**Typical Electrostatic Voltage at 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

Table 1 example of static charge sources and average voltage generated

When ESD occurs current is pushed through the dissipation path and causes the components within that path to disspate that energy in form of heat. The small components can melt or even vaporise. If the damage is not catastrophic then the device can survive initially only to cause delayed failures and reduce the stability, reliability and the life of the device. This is called the latent defect. *Fig. 4* shows a Catastrophic failure caused by a high energy ESD. *(Rouse, 2011)*

Fig. 4 PCB damaged by ESD

2.3 Preventive measure against ESD

Electronics manufacturers have areas free of static, sometimes called clean rooms. Those ESD protective areas avoid highly charging materials and uses measures to remove static such as providing antistatic devices and controlling humidity dust particles.
More generic anti-ESD techniques involves ESD-safe mats and grounding points for ground wrist straps. ESD-safe mats are usually highly conductive and grounded. The high conductivity shorts the IC's pins to avoid the internal circuity to become a path and the connection do ground provides a straight, low impedance dissipation path to ground. An example of a clean room setup is shown in *Fig. 5.*

ESD simulators may be used to test electronic devices, for example with a human body model or a charged device model. *(Storr, 1998)*

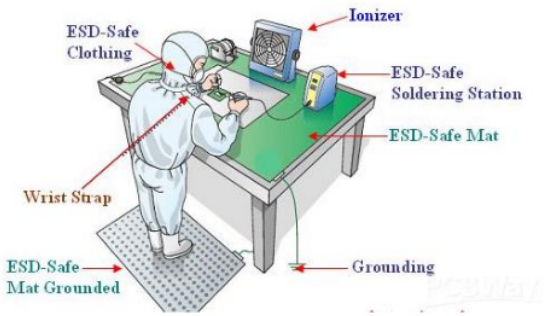


Fig. 5 preventive measures against ESD

ESD sensitive devices should be always contained in a anti-ESD bag or container during storage or transportation. *(Mini-circuits, nd)*

3. Design Requirements

The circuit design in *Fig. 6* was created using ECAD design and will be used as the CUT for the test and analysis for the assignment.

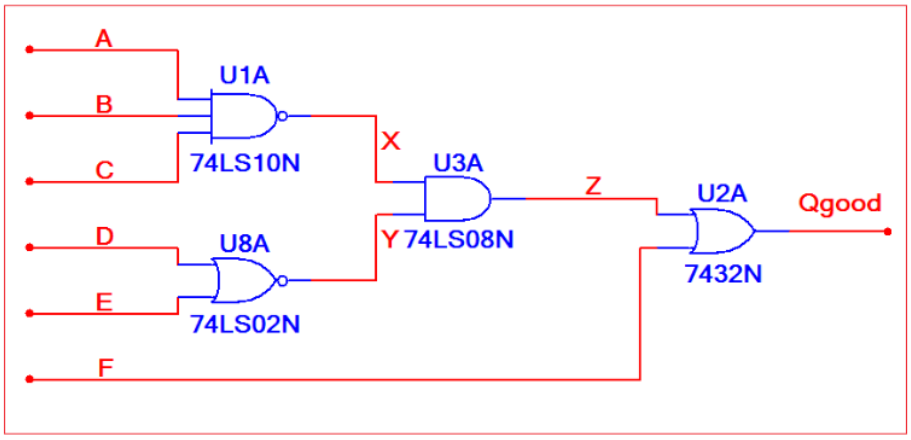


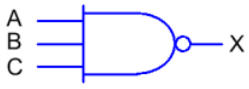
Fig. 6 CUT (Circuit Under Test)

3.1 Stuck at Fault Modelling

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 No of tests = m + 1 where m is the No of inputs.

Fig. 7 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.



Test No	A	B	C	X
1	1	1	1	0
2	0	1	1	1
3	1	0	1	1
4	1	1	0	1

Fig. 7 Reduced tests – 3 input NAND gate

Fig. 8 shows the table with the reduced No of tests for the 2 input NOR gate

Test No	D	E	Y
5	0	0	1
6	1	0	0
7	0	1	0

Fig. 8 Reduced tests - 2 input NOR gate

Fig. 9 shows the table with the reduced No of tests for the 2 input AND gate

Test No	X	Y	Z
8	1	1	1
9	0	1	0
10	1	0	0

Fig. 9 Reduced tests – 2 input AND gate

Fig. 10 shows the table with the reduced No of tests for the 2 input OR gate

Test No	Z	F	Q
11	0	0	0
12	0	1	1
13	1	0	1

Fig. 10 Reduced tests – 2 input OR gate

3.2 Test vectors test pattern

Table 2 shows the derived table for all possible stuck at fault tests and the possible fault scenario required to test each stage.

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
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

Table 2 Derived table.

Key:

* At least one condition must be at logic 0

** At least one condition must be at logic 1

The highlighted tests above indicate the duplicate tests which cause redundancies. Test 1 and 9 are duplicates and tests 5, 8 and 13 are duplicates. Only one of these are required and therefore this table can be reduced.

Table 3 shows the resultant table after the redundant tests have been eliminated. Test 9, 8 and 13 have been removed.

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

Table 3 Reduced test table

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 *Table 4* below. It can be noticed that the fault condition of the redundant test vector requirements are added to the existing test vector.

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 SA0
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

Table 4 Further reduced table after applying wild card rules

3.3 Multisim design tests using Word Generator.

Fig. 11 demonstrates the CUT ATE using a Multisim Word Generator to generate the test sequence require at the inputs. The test shown in *Fig. 11* demonstrates the no fault condition for the CUT. The resultant waveform was analysed using the logic analyser.

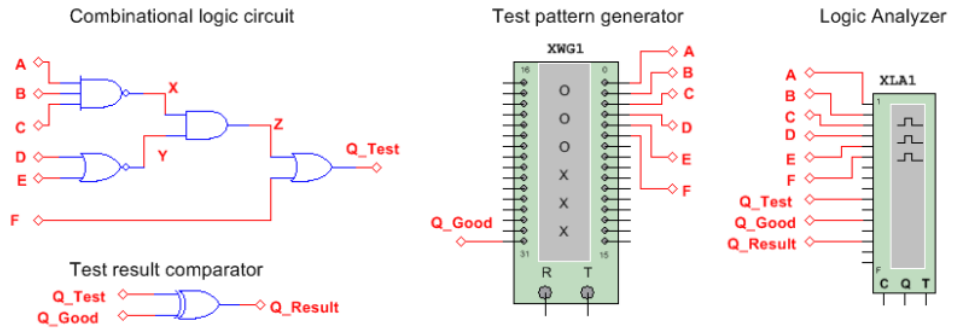


Fig. 11 Multisim test using Word Generator

Fig. 12 demonstrates the simulation results from the word generator test sequence and indicates the no faults condition in the resultant waveform between the two cursors.

This is the no fault condition for the CUT and no positive pulses on the resultant output shows the correct operation of the test.

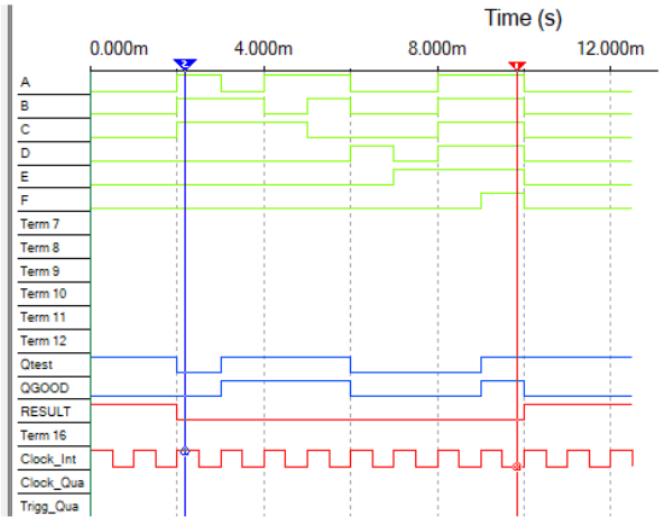


Fig. 12 Multisim Result for no fault condition

4. Test Analysis Using ECAD

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.

Fig. 13 demonstrates the CUT modified with a fault conditions according to the table of test vectors of input A SA1, B SA1 and F SA1.

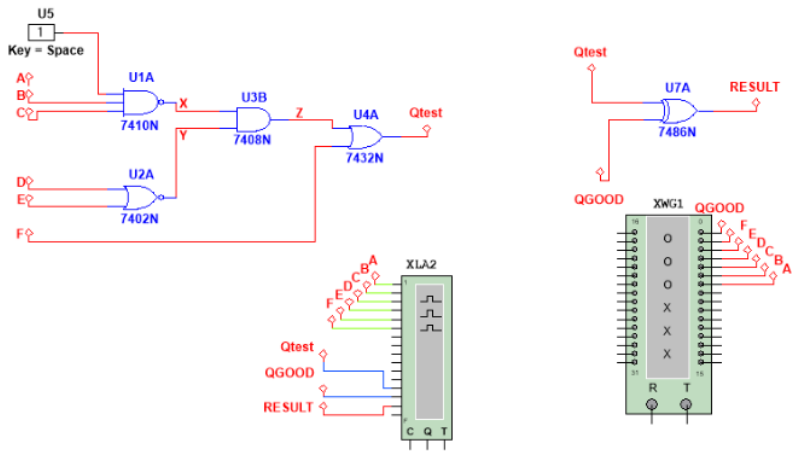


Fig. 13 A SA 1 fault condition

The test result in *Fig. 14* 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 in the image.

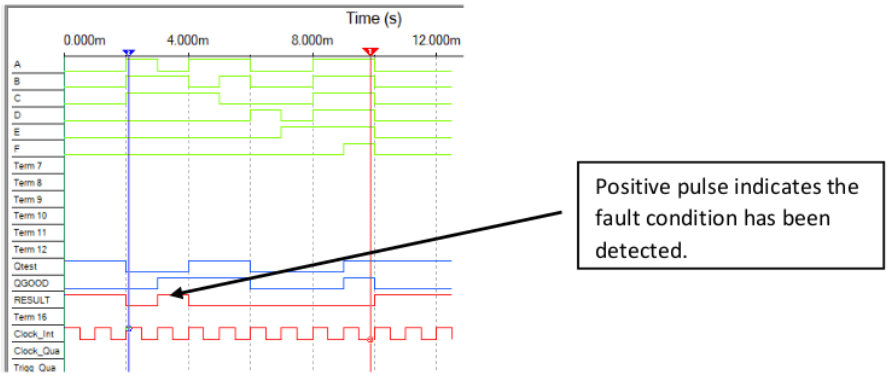


Fig. 14 Test result for input A SA1

The test result in *Fig. 15* and Fig. 16 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 in the image.

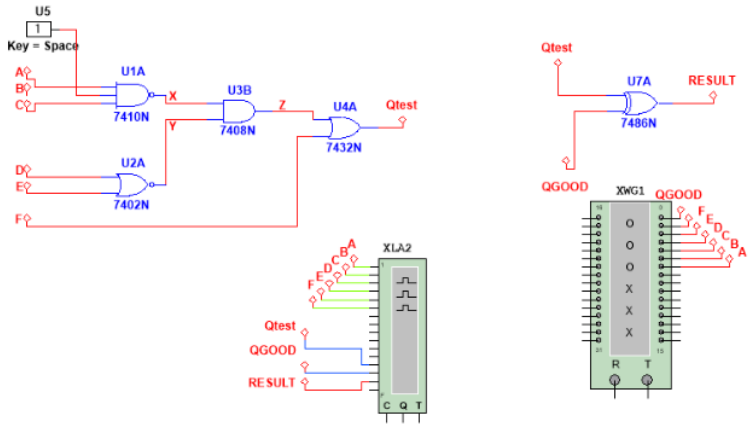


Fig. 15 B SA1 fault condition

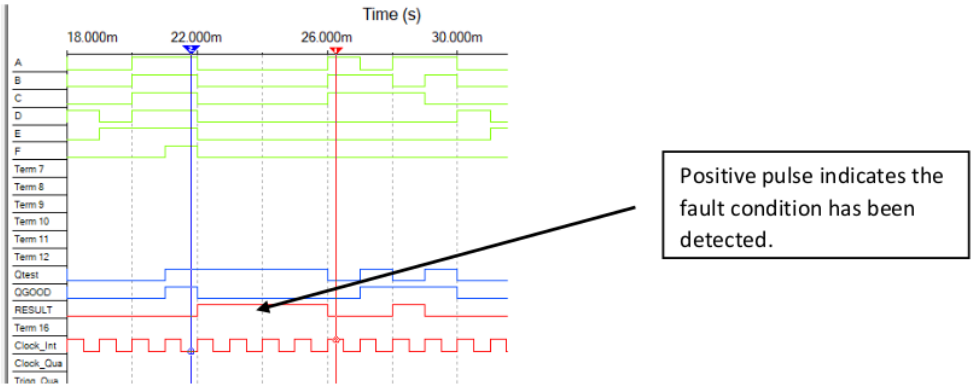


Fig. 16 Test result for input B SA1

The test result in Fig. 17 and *Fig. 18* shows the test circuit and resultant output from the logic analyser at logic 1 when input F is held at logic 1 throughout the test sequence. The fault condition is indicated between the two cursors in the image.

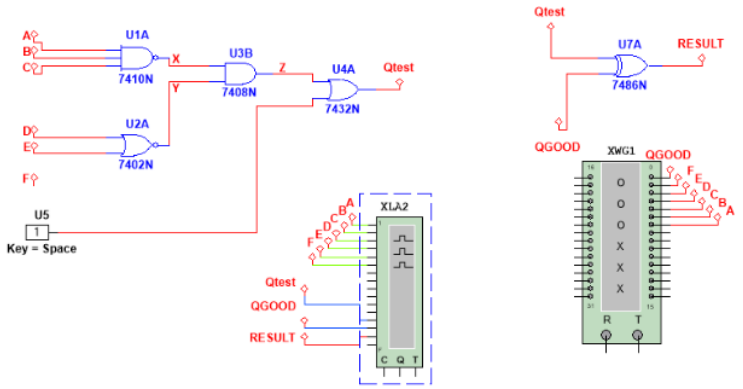


Fig. 17 F SA1 fault condition

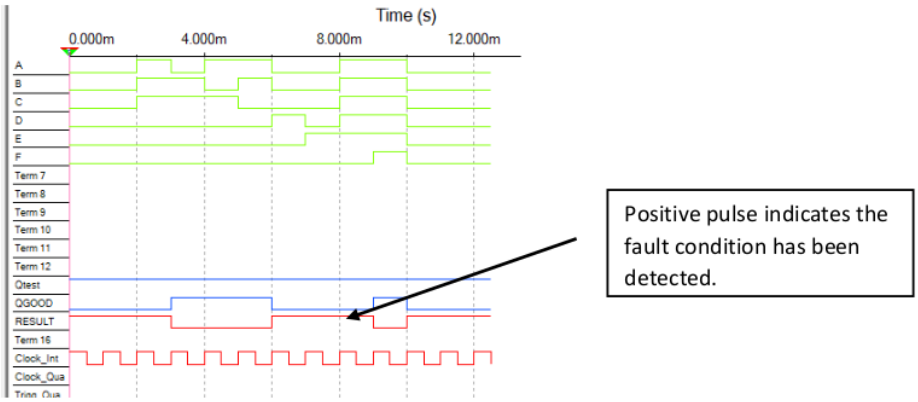


Fig. 18 Test result for input B SA1

5. 6-Bit Register Design and Test

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 used for fault detection and an analysis will be carried out of the results using this method.

Table 5 shows the next state table required for the design of the finite state machine that produces the test vectors for inputs A to F.

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

Table 5 Next state table

Karnaugh maps were used to generate the logic states required for the feedback combinational logic for the input sequence.

Table 6 shows the K-map for output QA of the register.

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

Table 6 Output QA

The Boolean expression from the K map is:

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

Table 7 shows the K-map for output QB of the register.

A B C / D E F	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

Table 7 Output QB

The Boolean expression from the K map is:

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

Table 8 shows the K-map for output QC of the register.

A B C / D E F	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

Table 8 Output QC

The Boolean expression from the K map is:

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

Table 9 shows the K-map for output QD of the register.

A B C / D E F	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

Table 9 Output QD

The Boolean expression from the K map is:

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

Table 10 shows the K-map for output QE of the register.

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

Table 10 Output QE

The Boolean expression from the K map is:

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

Table 11 shows the K-map for output QF of the register.

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	0	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

Table 11 Output QF

The Boolean expression from the K map is:

A.B.C.D.E.NF

The 6 bit regitser shown in *Fig. 19* was created from the results of the Next State logic design process. The feedback circuits are the result of the Boolean expressions taken from the K map procedur to minimise the combinational circuit.

The CUT 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 diffetentiate the outputs of the CUT and the known working cuitcuit.

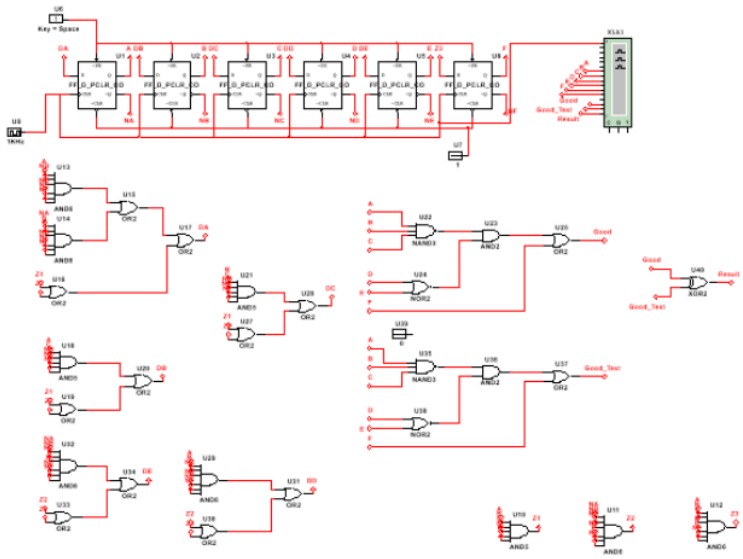


Fig. 19 6 Bit Register with no fault condition.

The result of the test with no fault condition is shown in *Fig. 20* below and both outputs of the CUT and known good test circuit are the same. The blue line is the output of the XOR gate and no pulses are present which indicates no differnces at the output.

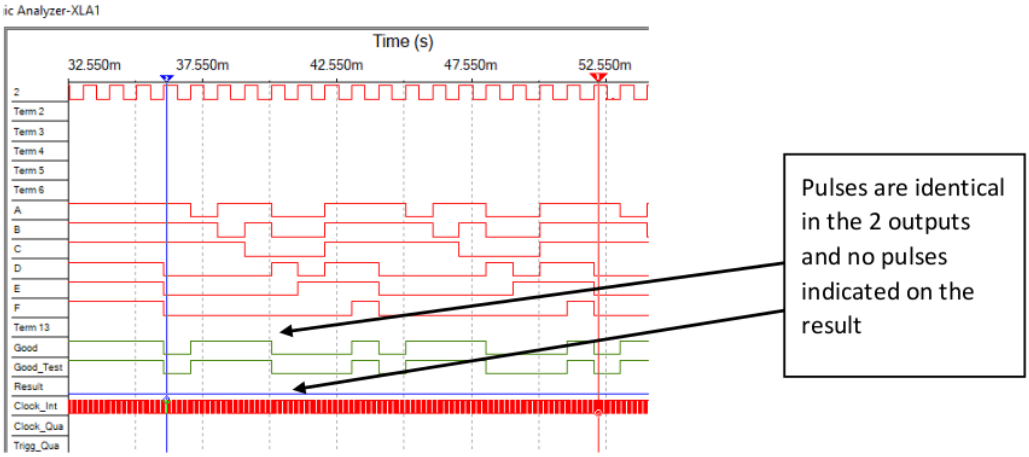


Fig. 20 Test result with no fault condition

5.1 Analysis of 6-Bit Register Fault Conditions.

Fig. 21 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.

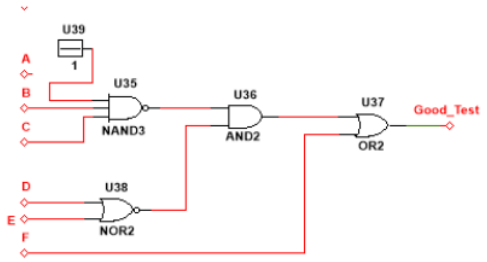


Fig. 21 CUT with input A SA1 fault condition.

Fig. 22 shows the tests result form the logic analyser where the fault is dtetced by a difference in the outputs from the CUT and the known good test circuit.

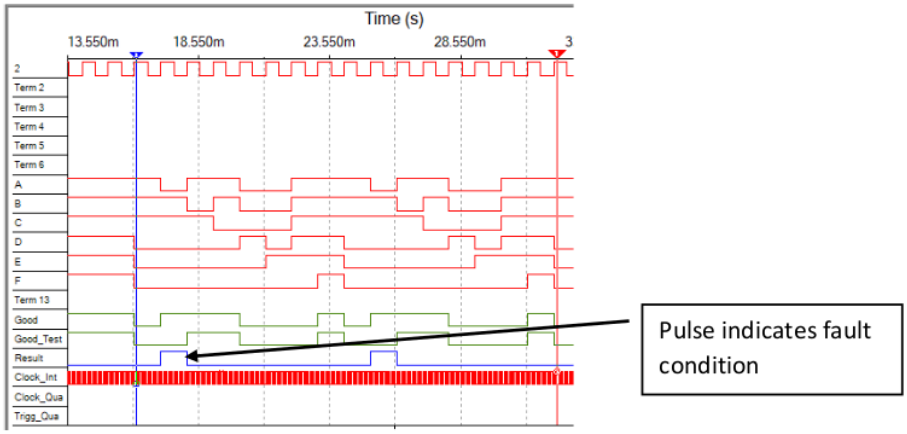


Fig. 22 Input A SA1 fault test result.

The same test run was carried out for faults set for input B SA1 and input F SA1 and the resultant output of each is shown below (*Fig. 23*-Fig. 26).

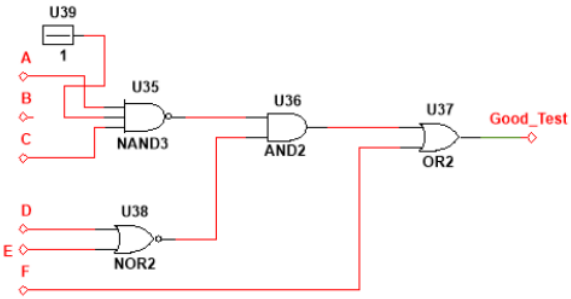


Fig. 23 CUT with input B SA1 fault condition.

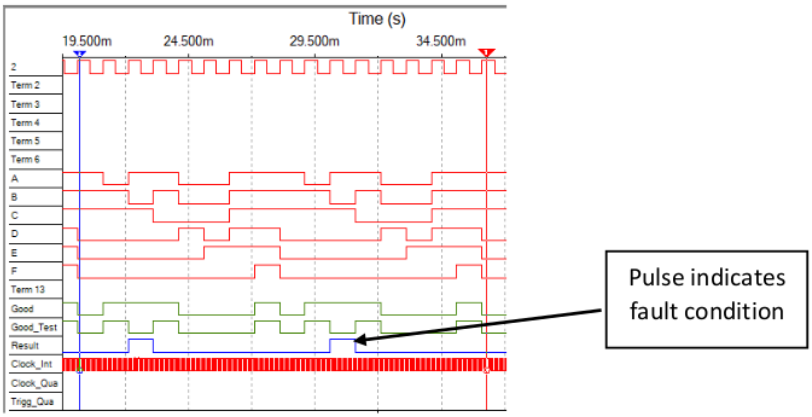


Fig. 24 Input B SA1 fault test result

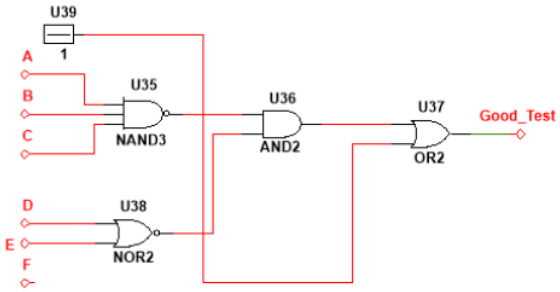


Fig. 25 CUT with input F SA1 fault condition

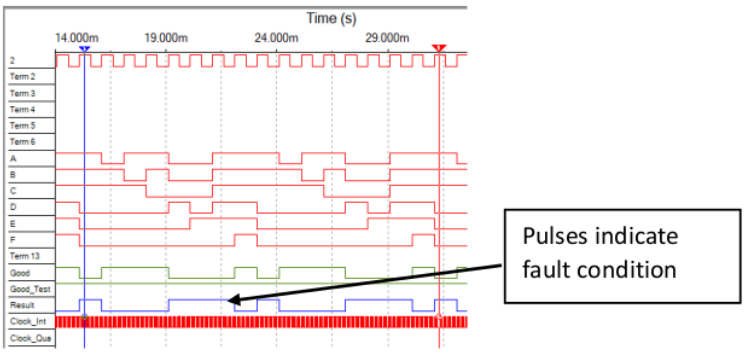


Fig. 26 Input F SA1 fault test result

5.2 Implementation and practical testing on NI ELVIS board.

A testing program was designed using NI LabVIEW and interfaced using NI Elvis II DAQ development board. The required library for interfacing the IOs was added to LabVIEW as a plugin.

The G-code is shown in *Fig. 27*; it was designed to generate the data stream used to drive CUT's inputs trough the NI Elvis's interface.

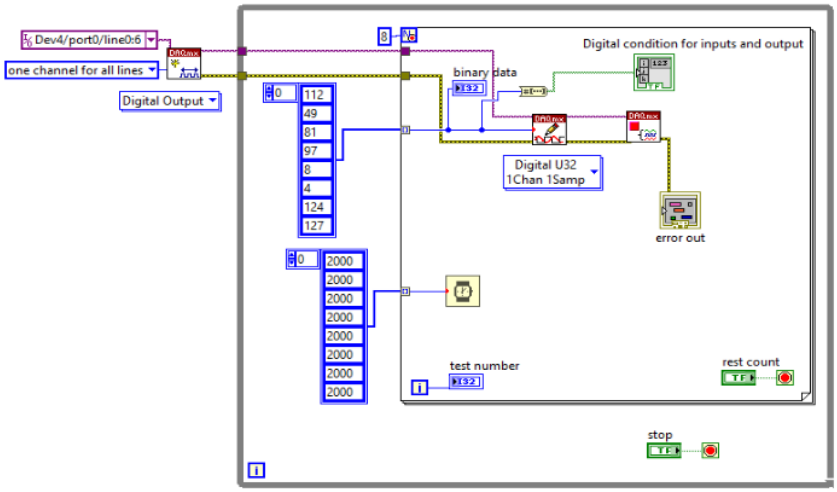


Fig. 27 LabVIEW Data Pattern Generator ATE

The CUT's output and a generated reference data are fed into a XOR gate for comparison. The XOR gate's output defines if the circuit passes or fails the test. *Fig. 28* shows the use of the DAQ input to monitor the status of the XOR gate's output.

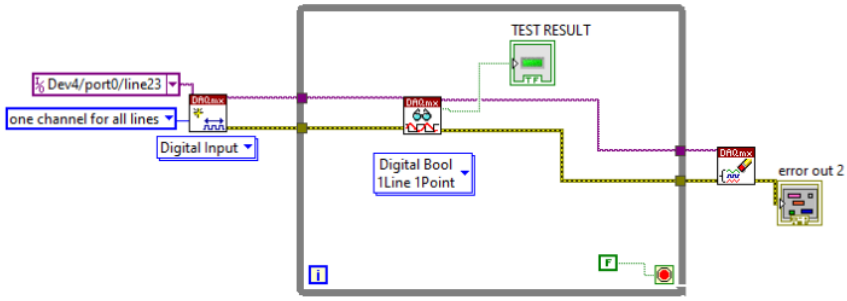


Fig. 28 LabVIEW Result output from X OR gate via DAQ input

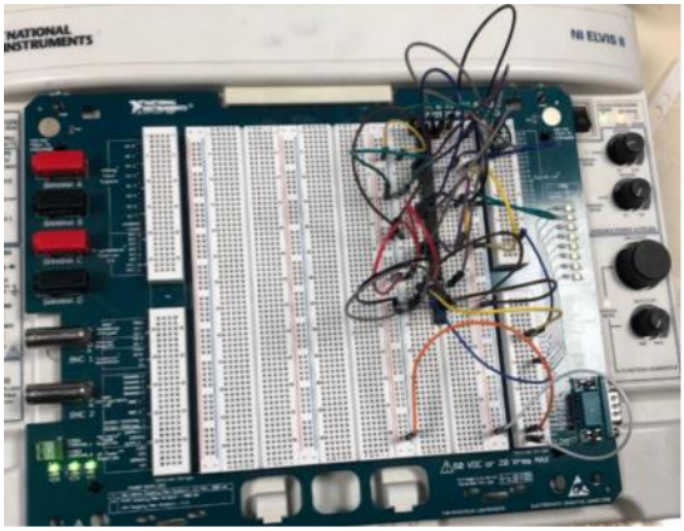


Fig. 29 Practical CUT setup using DAQ interface.

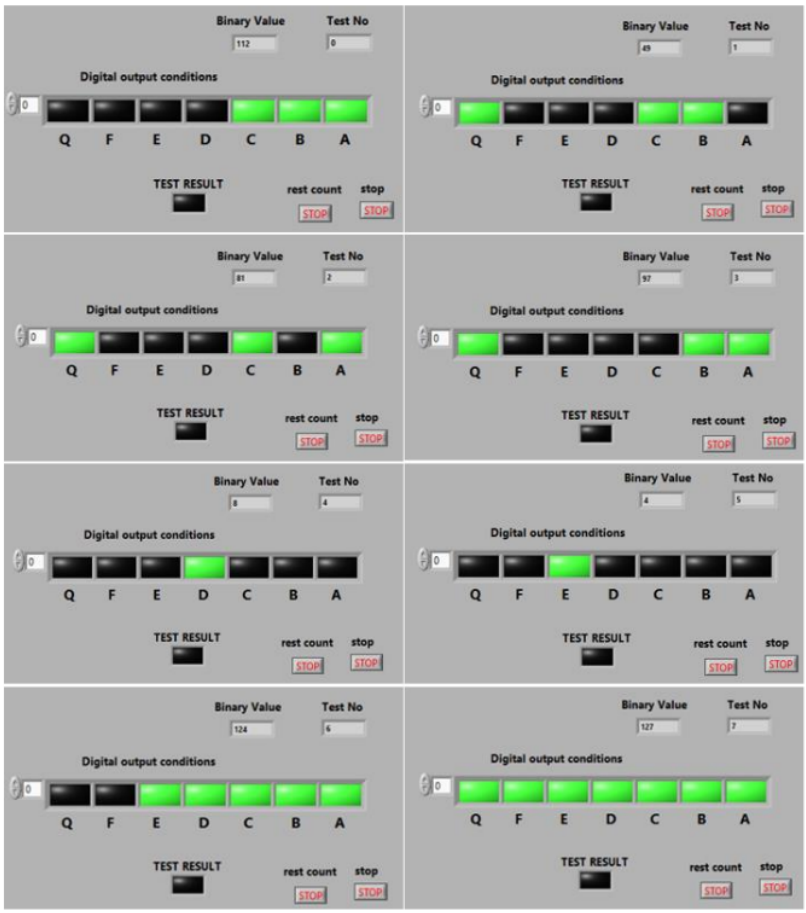


Fig. 30 Full set of tests carried out without a fault condition.

The practical setup of the CUT using the NI Elvis board is shown on *Fig. 29*. The practical setup was then run without simulating any faults to ensure proper operation; the results of the run are presented in *Fig. 30*.

The fault were then simulated same as they were simulated during ECAD test and the failed test results are shown in *Fig. 31Fig*. 33.

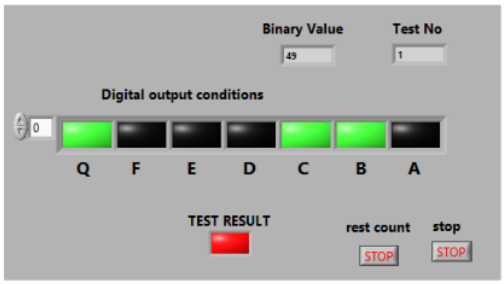


Fig. 31 Fault indicated for A SA1 condition

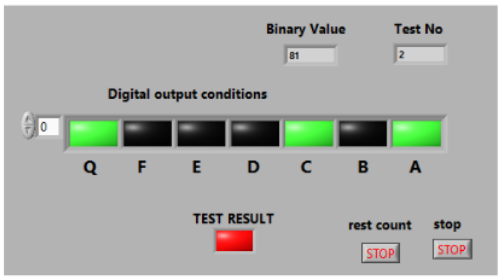


Fig. 32 Fault indicated for B SA1 condition

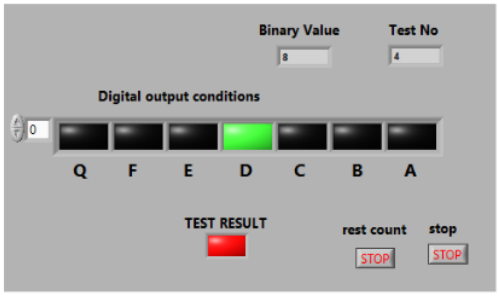


Fig. 33 Fault indicated for F SA1 condition

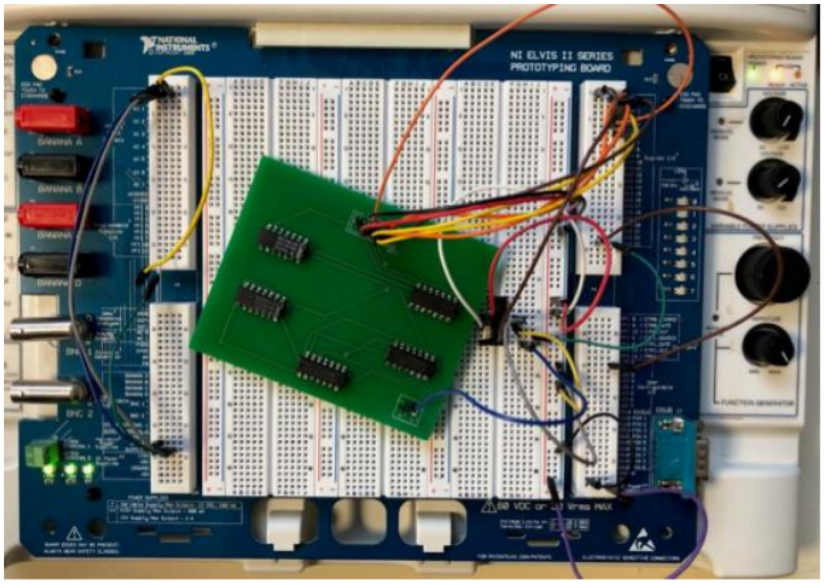


Fig. 34 Practical CUT substituted by PCB for testing.

The set up, where the pracal CUT setup was subtitued by the PCB (design described in next section) is shown in *Fig. 34.* That setup was first proven by testing it without simulated faults; then tested simulating the same faults as during the practical CUT setup. *Fig. 35Fig. 36* show examples of the carried out tests on the PCB.

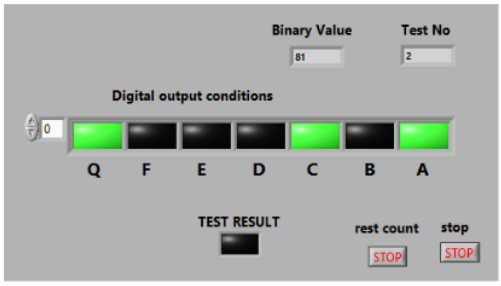


Fig. 35 No fault indicated for no fault condition set during PCB testing.

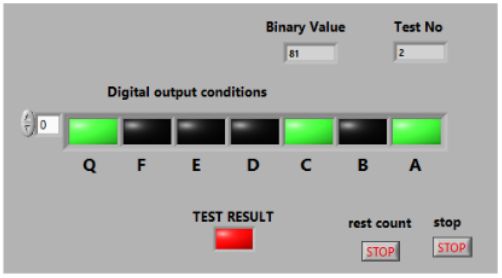


Fig. 36 Fault indicated for B SA1 condition during PCB testing.

Test no.	Practical CUD setup test results				PCB setup test results			
	No fault	**A SA1**	**B SA1**	**F SA1**	**No fault**	**A SA1**	**B SA1**	**F SA1**
1	pass	pass	pass	fail	pass	pass	pass	fail
2	pass	fail	pass	fail	pass	fail	pass	fail
3	pass	pass	fail	fail	pass	pass	fail	fail
4	pass	pass	pass	fail	pass	pass	pass	fail
5	pass	fail	fail	fail	pass	fail	fail	fail
6	pass	fail	fail	fail	pass	fail	fail	fail
7	pass	pass	pass	fail	pass	pass	pass	fail
8	pass	pass	pass	pass	pass	pass	pass	pass

Table 12 Table of practical tests results

The results for both setups are presented in *Table 1.* This proves that both setups provided an identical set of results.

6. PCB Design

To start the design, schematic diagram was drawn using Eagle PCB. The schematics (*Fig. 36*) were drawn using the standard wire connections instead of tags. Although tags could make things neater in a small project, in a more complex design that can become extremely hard to read if only tags are used. All unused pins in the chips were pulled to ground as a good practice and to improve ICs stability.

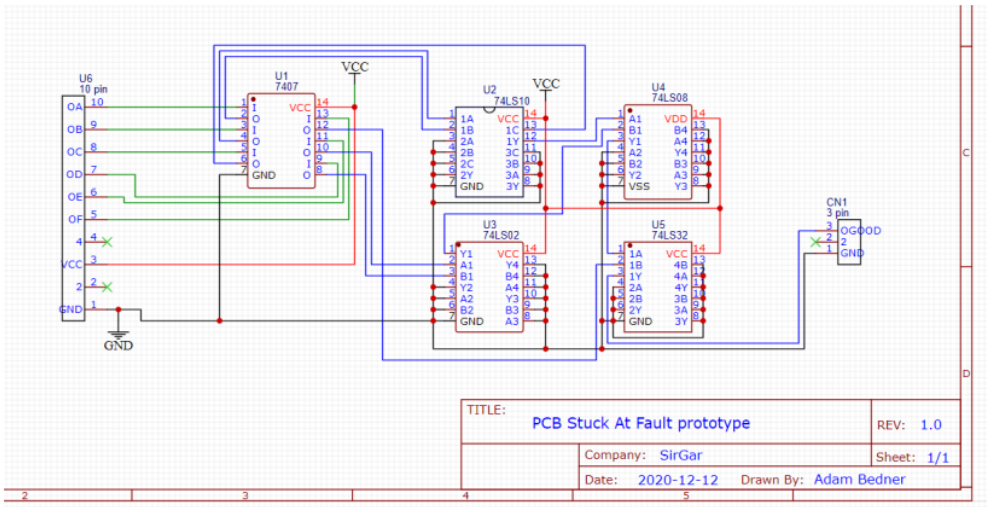


Fig. 37 Schematics of the PCB

The schematic was then transferred onto a PCB design. PCBs size, boundaries and mechanical objects (mounting holes, etc) were first defined. Trough hole components were used as those are easier to handle, solder, and probe from the larger pins. Also, the ICs were placed a good distance from each other in order to ease future probing of the circuit; otherwise, the components would have been laid out in such way that they occupy the minimum space possible. Also, any 90° bends were avoided as much as possible to improve circuit's performance and reduce impedance. The ground plane was not added as the test circuit will work only on low frequency DC signal. The finished product should include the ground plane, ideally on the bottom layer, to make the circuit more robust against interferences and improve the circuit's stability and reliability. *Fig. 38* shows the completed PCB layout, *Fig. 39* & *Fig. 40 show the copper layers and Fig. 41* presents the generated file output required for the manufacturing process.

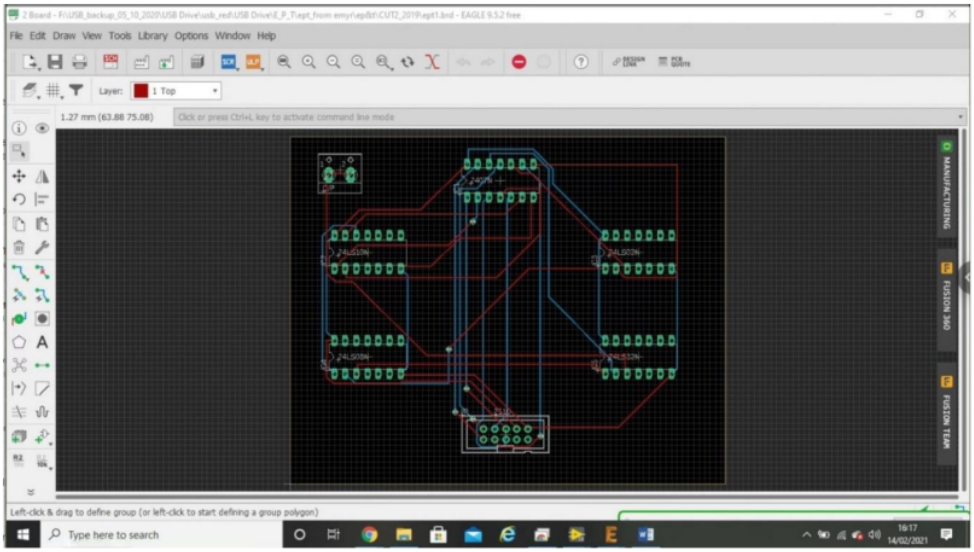


Fig. 38 PCB Layout

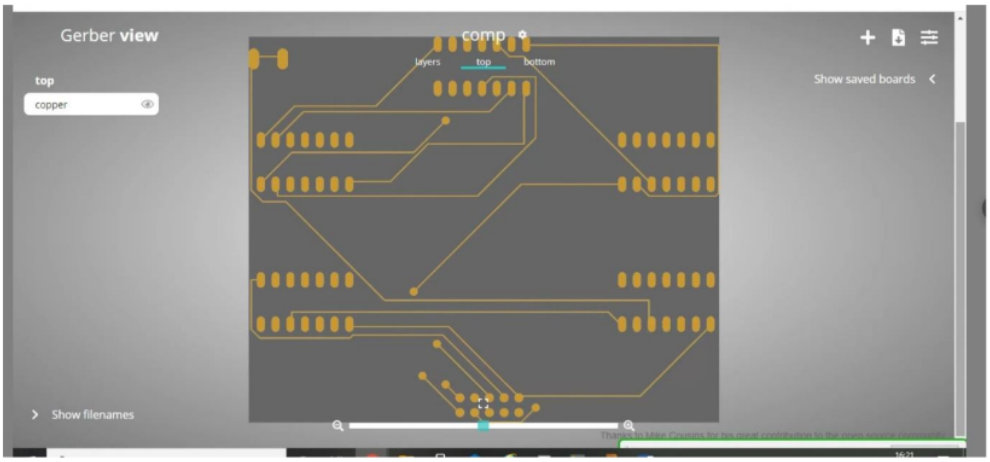


Fig. 39 Top Copper layer

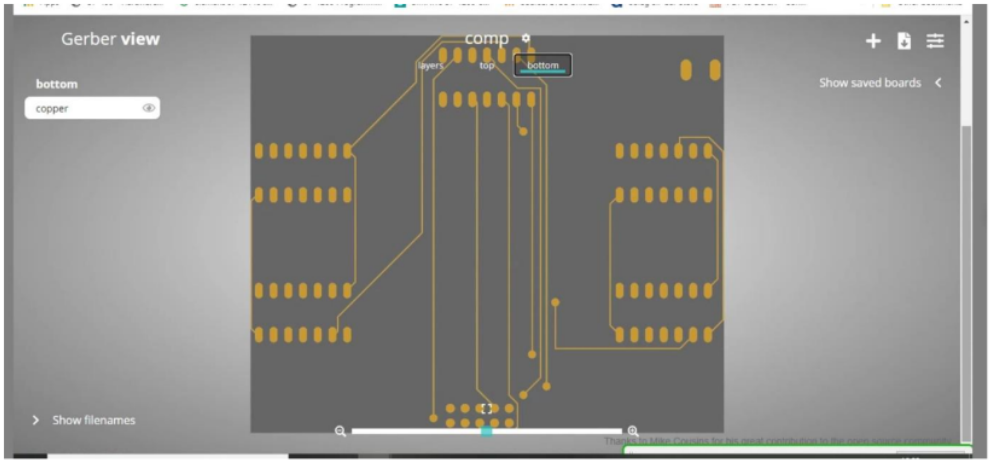


Fig. 40 bottom copper layer

comp.cmp	03/02/2019 21:27	CMP File	7 KB
comp.drd	03/02/2019 21:27	DRD File	2 KB
comp.dri	03/02/2019 21:27	DRI File	1 KB
comp.gpi	03/02/2019 21:27	GPI File	2 KB
comp	03/02/2019 21:27	S7wsvapx.ViewDo...	38 KB
comp.pls	03/02/2019 21:27	PLS File	1 KB
comp.sol	03/02/2019 21:27	SOL File	6 KB
comp.stc	03/02/2019 21:27	STC File	4 KB
CUT2_2019	24/01/2020 12:25	File	7 KB
CUT2_2019.gpi	24/01/2020 12:25	GPI File	2 KB
CUT2_2019	15/02/2019 15:02	WinRAR ZIP archive	28 KB
ept1.b#1	23/01/2019 12:59	B#1 File	36 KB
ept1	24/01/2020 11:59	EAGLE board	36 KB
ept1.pro	24/01/2020 12:00	PRO File	1 KB
ept1	14/01/2019 01:07	EAGLE schematic	59 KB

Fig. 41 Generated output files

Conclusions

The design required a CUT consisted of 6bit input combinational circuit.
The design was used to determine the test vectors to test for stuck at faults. To make the testing quicker and more efficient the process of test vector reduction was applied which reduced the complete number of tests vectors to 8.

MultiSim's word generator was used to drive CUT's inputs. The CUT was then tested with no faults simulated to ensure the proper operation of the circuit; three sets of tests followed with a simulated stuck at fault each. The output then was compared with a generated reference code by means of an XOR gate.
Ensuring that all other conditions remained the same, the circuit now was tested with a 6-bit pattern generator instead of a word generator and the results of the tests were the same.
Same condition testing was again repeated on a practical CUT setup which used LabVIEW and NI Elvis DAQ development board as interface. LabVIEW was chosen as it provides GUI of the real time processes. The results were identical for breadboard build and the PCB.

Table of figures

Fig. 1 Stuck at 0 fault model (Kastensmidt, 2008) 4
Fig. 2 examples of faults (Plusquellic, 2004) 5
Fig. 3 Fault equivalent diagrams (Kastensmidt, 2002) 5
Fig. 4 PCB damaged by ESD 6
Fig. 5 preventive measures against ESD 7
Fig. 6 CUT (Circuit Under Test) 8
Fig. 7 Reduced tests – 3 input NAND gate 8
Fig. 8 Reduced tests - 2 input NOR gate 8
Fig. 9 Reduced tests – 2 input AND gate 9
Fig. 10 Reduced tests – 2 input OR gate 9
Fig. 11 Multisim test using Word Generator 12
Fig. 12 Multisim Result for no fault condition 12
Fig. 13 A SA 1 fault condition 13
Fig. 14 Test result for input A SA1 13
Fig. 15 B SA1 fault condition 14
Fig. 16 Test result for input B SA1 14
Fig. 17 F SA1 fault condition 15
Fig. 18 Test result for input B SA1 15
Fig. 19 6 Bit Register with no fault condition. 19
Fig. 20 Test result with no fault condition 19
Fig. 21 CUT with input A SA1 fault condition. 20
Fig. 22 Input A SA1 fault test result. 20
Fig. 23 CUT with input B SA1 fault condition. 20
Fig. 24 Input B SA1 fault test result 21
Fig. 25 CUT with input F SA1 fault condition 21
Fig. 26 Input F SA1 fault test result 21
Fig. 27 LabVIEW Data Pattern Generator ATE 22
Fig. 28 LabVIEW Result output from X OR gate via DAQ input 22
Fig. 29 Practical CUT setup using DAQ interface. 23
Fig. 30 Full set of tests carried out without a fault condition. 23
Fig. 31 Fault indicated for A SA1 condition 24
Fig. 32 Fault indicated for B SA1 condition 24
Fig. 33 Fault indicated for F SA1 condition 24
Fig. 34 Practical CUT substituted by PCB for testing. 25
Fig. 35 No fault indicated for no fault condition set during PCB testing. 25
Fig. 36 Fault indicated for B SA1 condition during PCB testing. 25
Fig. 37 Schematics of the PCB 27
Fig. 38 PCB Layout 28
Fig. 39 Top Copper layer 28
Fig. 40 bottom copper layer 29
Fig. 41 Generated output files 29
Table 1 example of static charge sources and average voltage generated 6
Table 2 Derived table 10
Table 3 Reduced test table 11
Table 4 Further reduced table after applying wild card rules 11
Table 5 Next state table 16

Table 6 Output QA 16
Table 7 Output QB 17
Table 8 Output QC 17
Table 9 Output QD 17
Table 10 Output QE 18
Table 11 Output QF 18
Table 12 Table of practical tests results 25

References

I Koren. Z Koren, (1998). Defect tolerance in VLSI circuits: techniques and yield analysis
Available at: https://ieeexplore.ieee.org/abstract/document/705525
[Accessed 16th Jan 2021]

Fall, 2002. Fault models, fault collapsing and logic simulation.
Available at: http://web.eecs.umich.edu/~mazum/F02/lectures/lecmazum3.pdf
[Accessed 16th Jan 2021].

Kastensmidt, L. 2008. *Lecture 2 - Fault Modeling.*
Available at: http://web.eecs.umich.edu/~mazum/F02/lectures/lecmazum3.pdf
[Accessed 6th April 2018].

Plusquellic, J. 2004. VLSI Design Verification and Test - Defects.
Available at: http://ece-research.unm.edu/jimp/vlsi_test/slides/defects.pdf
[Accessed: 16th Jan 2021]

Ubar, R. Jürimägi, L. Orasson, E. & Raik, J. 2017. Fault Collapsing in Digital Circuits Using Fast Fault.
Available at: https://hal.inria.fr/hal-01578613/document
[Accessed: 16th Jan 2021]

Mohammadat, M. 2015. Resistive Open Faults Detectability analysis.
Available at: https://eprints.soton.ac.uk/364712/1/manuscript.pdf
[Accessed on: 17th Jan 2021]

Mazumder, P., 2002. Digital Testing. [Online]
Available at: http://web.eecs.umich.edu/~mazum/F02/lectures/lecmazum3.pdf
[Accessed on: 17th Jan 2021]

Toms, R. T., Jalilizeinali, R. & Shiqun, G., 2015. United States of America, Patent No. US20100091475A1.

Rouse, M., 2011. *Electrostatic discharge (ESD).* [Online]
Available at: https://whatis.techtarget.com/definition/electrostatic-discharge-ESD
[Accessed on: 17th Jan 2021].

Storr, P., 1998. ElectoStatic Discharge (ESD) precautions. [Online]
Available at: http://philipstorr.id.au/pcbook/book5/esd.htm
[Accessed on: 17th Jan 2021].

AN-40-005, nd. Mini-circuits. Prevention and Control of Electrostatic Discharge (ESD)
Available at: 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#80a2ce175c63a7cd8cdafd76945e9dbb
[Accessed on: 17th Jan 2021]

1708165

GRADEMARK REPORT

FINAL GRADE

75/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

GRADING FORM: CSG-RUBIC

ADAM BEDNER	**0**

RESEARCH-INFORMED LITERATURE

Research covers significant detail of the requirements and has numerous relevant sources. Research carried out shows extensive reading

11/15

around the subject area. Number of relevant sources are good.

KNOWLEDGE AND UNDERSTANDING

A sound knowledge has been demonstrated deriving the required minimum set of test vectors . 8.5/10 The design of the LabVIEW test vectors shows a great deal of study has ben carried out to extract the test sequence. 8.5/10 The PLD design alternative underpins the knowledge gained from your study and the design alternatives demonstrate a good knowledge of the subject area with regard to data handling. 4/5 The CAM design of the CUT has been carried out successfully using an approved industrial software and the Gerber design files have been created. 3/5

24/30

ANALYSIS

Extensive ECAD simulation testing has been carried out using very good designs and have proved to be very successful with good explanations and supporting images. Good reference points are made to the test vector table form previous task. 8/10 A good comparison to theory has been carried out with the test results and explained in the analysis and in the conclusion. 3/5

11/15

PRACTICAL APPLICATION AND DEP.

Prototype of CUT was carried out successfully and manufacturing files created for PCB manufacture. The resulting PCB design was error free and fit for ATE testing. 7/10 The LabVIEW automatic testing carried out has shown to be successful with a good range of test scenarios used derived from test vector table. 12/15

19/25

SKILLS FOR PROFESSIONAL PRAC.

A good overall use of a range of software packages and good programming skills have been demonstrated. 3.5/5 Report layout is very good and has the correct sequence of tasks required. 3.5/5 The conclusion covers all the good points but could have included some critical analysis. 3/5

10/15