

Assignment 2

by Bruno Forkuoh

FILE	7838_BRUNO_FORKUOH_ASSIGNMENT_2_158487_1425955822.DOCX		
TIME SUBMITTED	26-APR-2018 03:32PM (UTC+0100)	WORD COUNT	4422
SUBMISSION ID	87560542	CHARACTER COUNT	24000

Bachelor of Science in Electrical & Electronic Engineering



Electronic Production & Test

PSDE 6018

Fault Modelling

Bruno Forkuoh

TABLE OF CONTENTS

Table of Figures.....	3
1 Introduction.....	5
2 Research.....	6
2.1 IC fabrication faults	6
2.2 Electro Static Discharge (ESD)	9
3 Design of assignment requirements	12
3.1 Stuck at fault modelling	12
3.2 Test vectors / test pattern.....	14
3.3 Multisim design and testing.....	16
4 Analysis of NI MultiSim results.....	17
5 6-bit register design and test.....	20
6 Practical build and test using LabView	26
6.1 Design.....	26
6.2 Testing	29
7 Conclusion.....	32
8 Reference List.....	33

TABLE OF FIGURES

Figure 2.1 - Three input AND gate (SlidePlayer, n.d.)	6
Figure 2.2 - Stuck at fault modeling (Lima Kastensmidt , 2008)	7
Figure 2.3 - Stuck at faults 'AND' and 'OR' gates (Anon., n.d.).....	7
Figure 2.4 - Fault equivalence (Lima Kastensmidt , 2008)	8
Figure 2.5 - Fault dominance (Lima Kastensmidt , 2008).....	8
Figure 2.6 - Electrostatic discharge (ESD) (Desco, 2013)	9
Figure 2.7 - Everyday life ESD causes (Storr, 1998).....	9
Figure 2.8 - Susceptibility figures (Storr, 1998).....	10
Figure 2.9 - ESD damage on PCB (SlidePlayer, n.d.)	10
Figure 2.10 - ESD prevention measures (Stack Exchange Inc,, 2018)	11
Figure 3.1 - Combinational logic circuit.....	12
Figure 3.2 - Reduced tests - three input NAND gate.....	12
Figure 3.3 - Reduced tests - two input NOR gate	13
Figure 3.4 - Reduced tests - two input AND gate	13
Figure 3.5 - Reduced tests - two input OR gate	13
Figure 3.6 - Derived table.....	14
Figure 3.7 - Reduced table	15
Figure 3.8 - Further reduced table	15
Figure 3.9 - Multisim circuit design	16
Figure 3.10 - Multisim test results	16
Figure 4.1 - Multisim circuit design for stuck at fault testing.....	17
Figure 4.2 - A stuck at '1' Multisim test results.....	18
Figure 4.3 - D stuck at '0' Multisim test results	18
Figure 4.4 - F stuck at '1' Multisim test results.....	19

Figure 5.1 – State table.....	20
Figure 5.2 – Karnaugh map - Output QA.....	21
Figure 5.3 - Karnaugh map - Output QB.....	21
Figure 5.4 - Karnaugh map - Output QC.....	22
Figure 5.5 - Karnaugh map - Output QD	22
Figure 5.6 - Karnaugh map - Output QE.....	23
Figure 5.7 - Karnaugh map - Output QF	23
Figure 5.8 - Multisim 6-bit pattern generator.....	24
Figure 5.9 – Test results -6-bit pattern generator.....	25
Figure 6.1 - LabVIEW block diagram (Digital Outputs)	26
Figure 6.2 - LabVIEW logic diagram (Digital Inputs).....	27
Figure 6.3 - LabVIEW front panel.....	27
Figure 6.4 - NI ELVIS II development board	28
Figure 6.5 - LabVIEW test results	29
Figure 6.6 - A stuck at '1' LabVIEW test result.....	30
Figure 6.7 - D stuck at '0' LabVIEW test result.....	30
Figure 6.8 - F stuck at '1' LabVIEW test results.....	31

Good list of contents

1 INTRODUCTION

The aim of this assignment is to analyse and evaluate a range of processes, techniques and strategies required for developing a prototype design into a product that can be manufactured and tested in order to meet customer needs.

This will be achieved by:

- Demonstrating an understanding of the use of Automatic Test Equipment (ATE) in testing IC such as single stuck at fault model (SSF)
- Researching the dangers of electrostatic discharge including how it can be prevented during the design and manufacturing process.
- Building a combinational logic circuit using CAD software, such as Multisim.
- Performing vector reduction techniques using the SSF model.
- Deriving test results from the modelled circuit using both Multisim and NI LabVIEW.
- Analysing and comparing the derived and simulated test results.
- Designing a 6-bit pattern generator to show how the circuit can be tested using this method.
- Implementing the 6-bit pattern into LabVIEW for testing a practical circuit built using the Elvis development board.

2 RESEARCH

2.1 IC fabrication faults

Several defects can occur during and after the manufacture of integrated circuits (ICs). These faults can be summarised into four categories: processing defect, material defects, time-dependent failures and packaging failures (Fall, 2002).

IC fabrication faults covers a range of fault modelling techniques that make possible analysing integrated circuits after their fabrication and determining if any of the above mentioned faults are present.

IC fabrication faults include fault-modelling techniques for common faults such as transistor open and short faults, memory faults, functional faults, delay faults and analogue faults. These common faults are beyond the scope of this assignment and therefore will not be investigated further (Fall, 2002).

Single stuck at faults, object of investigation in this assignment, is a fault-modelling technique for modelling physical faults in a circuit.

Three properties 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 an input or output of a gate. (Fall, 2002)

The total number of possible stuck at faults is twice the number of line in the circuit. Figure 2.1 shows a three input and gate. Since a three input and gate has four lines, three inputs and one output, the number of possible stuck at faults is eight.



Figure 2.1 - Three input AND gate (SlidePlayer, n.d.)

Test vectors are test patterns applied to a circuit in order to detect particular fault within a circuit. The test pattern must be such that if a fault is present the output must be different from that of the good circuit. This means that if the output of a good circuit is logic one, the output must be zero if a fault is present and vice versa.

There are four types of stuck at faults:

- Stuck at '1' (SA1) – input or output is stuck to logic 1.
- Stuck at '0' (SA0) – input or output is stuck to logic 0.
- Bridging fault – Two outputs of gates are shorted together.
- Feedback bridging fault – the input and output of a gate are shorted.

Figure 2.2 shows an example of how to determine the test vector for a stuck at fault. An input 'a' SA0 fault can be detected by applying the test vector '11' to the NAND gate. This will cause the output of the gate to change from zero to one if the input 'a' SA0 fault is present.

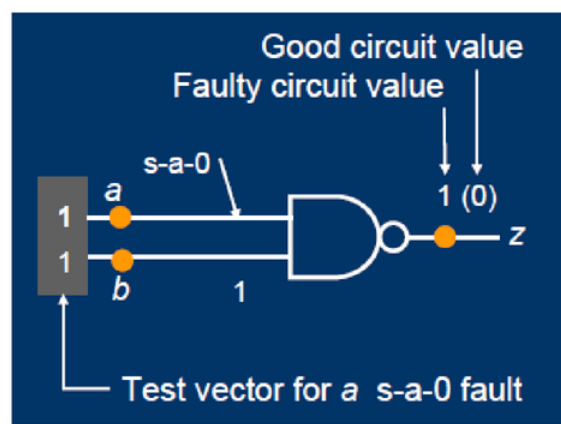


Figure 2.2 - Stuck at fault modelling (Lima Kastensmidt, 2008)

Figure 2.3 shows further examples of stuck at one and stuck at zero fault modelling applied to a three input 'AND' and a three input 'OR' gate. As it can be seen below, in all four case the stuck at fault produced a change of output as previously explained.

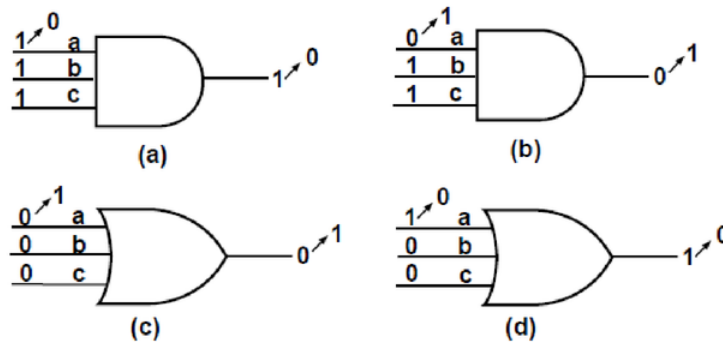


Figure 2.3 - Stuck at faults 'AND' and 'OR' gates (Anon., n.d.)

The number of single stuck at faults to be considered can be reduced using techniques such as fault equivalence and fault dominance.

Fault equivalence involves identifying and grouping into subsets two or more faults, which effect on the circuit are identical. The faults within the subsets can then be reduced by implementing the collapsed fault method, which involve reducing the faults in the subset to one fault and therefore one test vector or test pattern.

Figure 2.4 shows the fault equivalence method applied to a two input and gate and a two input or gate. In the case of the 'AND' gate when one input is stuck at zero the output will be zero and therefore two of the three faults joined by the orange line can be removed. The same principle can be applied to the 'OR' gate. In both case the fault equivalence technique reduces the number of faults that need to be considered from six to four.

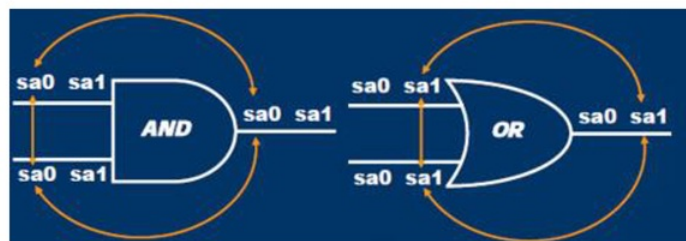


Figure 2.4 - Fault equivalence (Lima Kastensmidt , 2008)

The fault dominance technique involves two faults where one dominates the other. In the example shown in Figure 2.5 the test of F1 detect F2 and therefore F2 is said to dominate F1. This means that the test F1, highlighted in orange can be used for detecting both faults and therefore all other tests of F2 can be removed.

In this case, fault dominance allowed reducing the number of test vectors from seven to one.

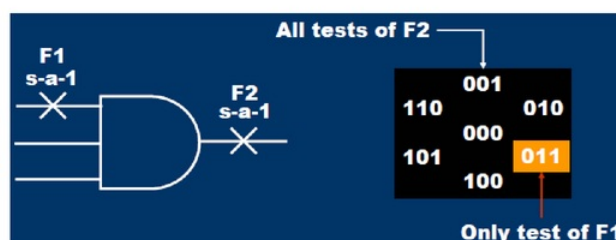


Figure 2.5 - Fault dominance (Lima Kastensmidt , 2008)

2.2 Electro Static Discharge (ESD)

Electrostatic discharge (ESD), shown in Figure 2.6, is the release or transfer of static electricity from a positively charged object being exposed to a negatively charged object.

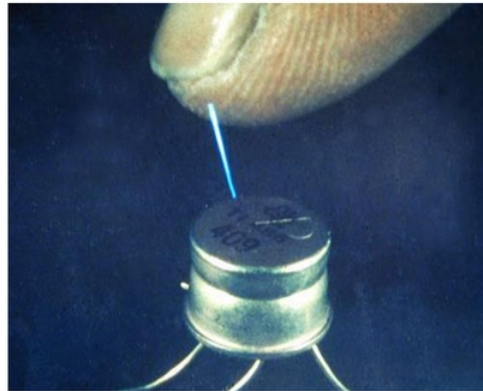


Figure 2.6 - Electrostatic discharge (ESD) (Desco, 2013)

Figure 2.7 shows some example of sources of ESD voltages. As it can be seen below, these events can occur during everyday life and can have large magnitudes.

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
Triggering a vacuum solder remover	8 KV
Spraying aerosol freezer spray	15 KV

Figure 2.7 – Everyday life ESD causes (Storr, 1998)

Figure 2.8 shows ESD susceptibility figures for some electronic devices. As it can be seen below, the voltage that can cause damages to electronic devices is very low. This is due to the reduction of components in the modern era, which means that large number of components can be fitted in very small devices.

Device type	ESD susceptibility
Bipolar transistors	380 to 7000 Volt
CMOS logic devices	250 to 3000 Volt
EPROM devices	100 volt
Film resistors	300 to 3000 Volt
TTL logic devices	1000 to 2500 Volt
Microprocessor chips	as low as 10 Volt

Figure 2.8 - Susceptibility figures (Storr, 1998)

When ESD occurs, both current and heat is generated. When the heat is released onto an electronic device, the small components can melt or vaporize causing the device to fail. In some cases, the device still functions but this event significantly reduces the life of the device. This is known as latent defect (Rouse, 2011).

Figure 2.9 shows an example of PCB damage caused by electrostatic discharge.



Figure 2.9 - ESD damage on PCB (SlidePlayer, n.d.)

Appropriate measures can be adopted during the manufacturing, testing, shipping and handling process of electronic devices in order to prevent ESD events. Some of these preventive measures involve wearing a wrist strap or ESD control footwear during the manufacturing process and using conductive, static dissipative and anti-static materials (Storr, 1998) that shield the product from a charge during packaging of electronic devices (Rouse, 2011).

Figure 2.10 shows several ESD precaution measures than can be adopted in the manufacturing area of electronic devices.

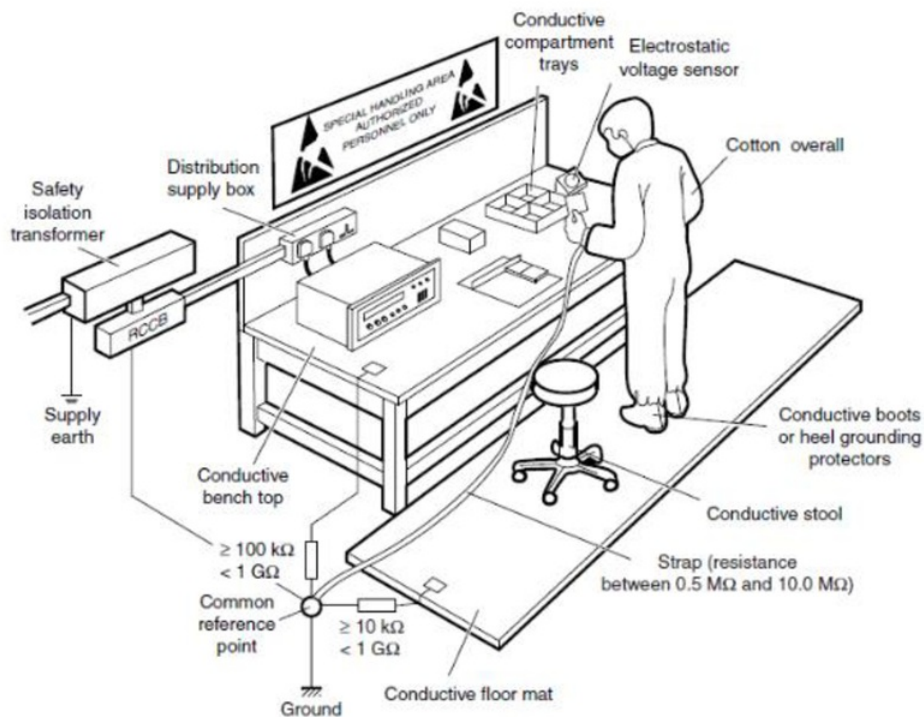


Figure 2.10 - ESD prevention measures (Stack Exchange Inc;, 2018)

The use of ESD safe containers such as Static safe bags, Conductive tote boxes, Conductive foam, Metal foil and Device storage tubes are other precaution measures that can be adopted in order to prevent ESD (Storr, 1998).

3 DESIGN OF ASSIGNMENT REQUIREMENTS

In order to achieve the objective of the assignment a combination logic circuit shown in Figure 3.1 was built using multisim.

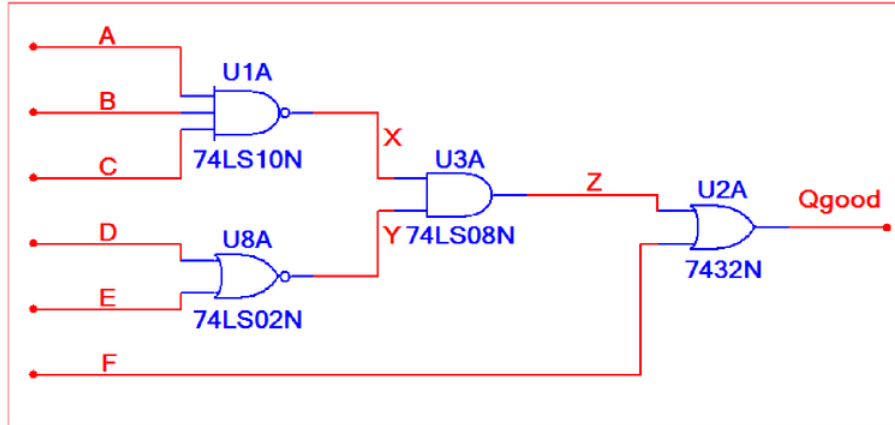


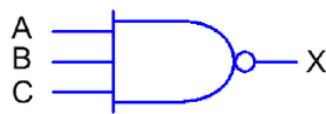
Figure 3.1 - Combinational logic circuit

3.1 Stuck at fault modelling

Following the design of the circuit the 'stuck at' fault modelling technique was implemented in order to detect stuck at zero (SA0) and stuck at one (SA1) faults.

The number of test required for each individual gate was reduced using the formula $m+1$, where 'm' is the number of inputs.

Figure 3.2 shows a NAND a gate and the table with the reduced number of tests. As it can be seen below, there are three inputs and therefore four tests are 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

Figure 3.2 - Reduced tests - three input NAND gate

Figure 3.3 shows a NOR gate and the table with the reduced number of tests

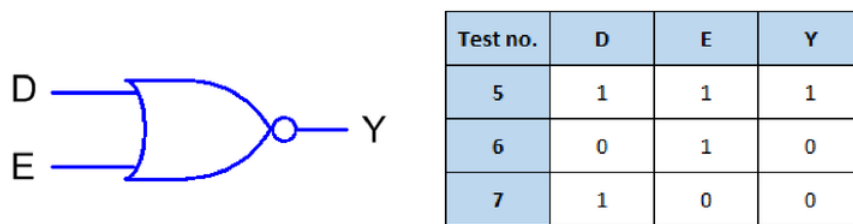
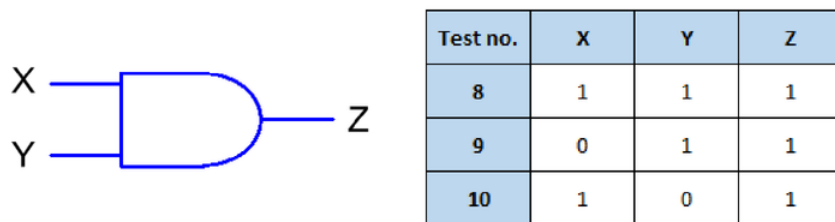


Figure 3.3 - Reduced tests - two input NOR gate

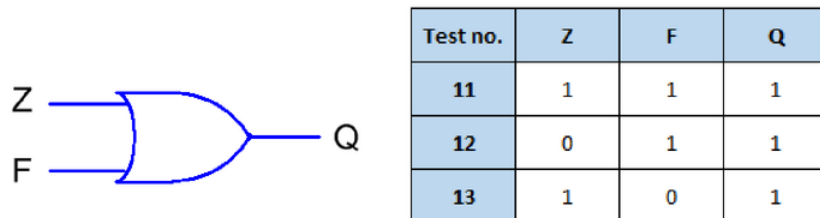
Figure 3.4 shows an AND gate and the table with the reduced number of tests.



AND o/p
should be: Z=
1 0 0

Figure 3.4 - Reduced tests - two input AND gate

Figure 3.5 shows an OR gate and the table with the reduced number of tests.



OR gate output
should be: Q =
0 1 1

Figure 3.5 - Reduced tests - two input OR gate

3.2 Test vectors / test pattern

Figure 3.6 shows the derived table of all possible 'stuck at' fault tests, also known as test vectors.

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

Figure 3.6 – Derived table

* means that at least one must be at logic 0, while ** At least one must be at logic 1

As it can be seen highlighted in the above table the test pattern for test 1 and 9, and that of test 5, 8 and 13 were the same. The equivalency rule was applied in order to reduce the number of test required.

Test model
outputs are
correct in table.

Figure 3.7 shows the reduced table obtained after applying the equivalency rule. Test 9 was removed and the fault was added to that of test 1. The faults for test 8 and 13 were added to that of test 5.

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

Figure 3.7 - Reduced table

Since At least one of input A, B and C of test 5 must be at logic '0', inputs A and B were changed to logic '1' so that test 5 and test 4 were equal. This allowed removing one of the two tests according to the equivalency rule.

The same principle was applied to test 10 where at least one of input D and E must be at logic '1', so input E was changed to logic '0' so that test 10 and test 6 were equal.

Figure 3.8 shows the further reduced table. As it can be seen below, highlighted in green, inputs A and B of test 4 were changed to logic '1', while input of F of test 6 was changed to logic '0'. This allowed removing test 5 and 10, which were added respectively to fault 4 and 6.

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

Figure 3.8 - Further reduced table

Table outputs are good. The reduction method has been implemented.

3.3 Multisim design and testing

Figure 3.9 shows the combinational logic circuit designed using Multisim in order to perform tests using the stuck at fault modelling technique. The test pattern for stuck at faults and the predicted output pattern of the combinational logic circuit were generated using a word generator, while a logic analyser was used to analyse the test results.

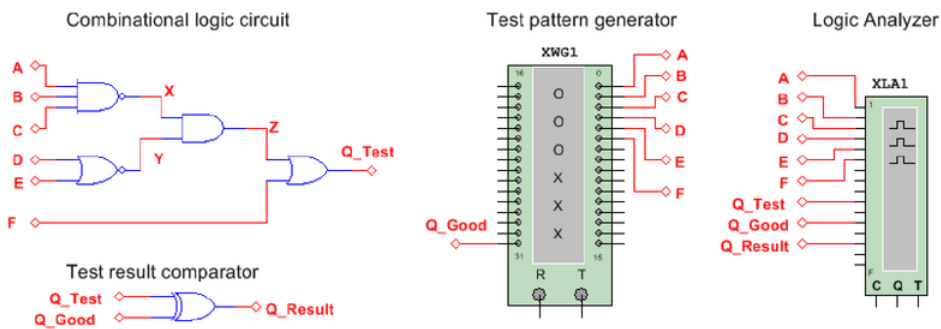


Figure 3.9 - Multisim circuit design

Figure 3.10 shows the simulations results obtained when testing the above circuit. As it can be seen below, the test pattern and the output of the combination logic circuit (Q_Results) were as predicted in Figure 3.8.

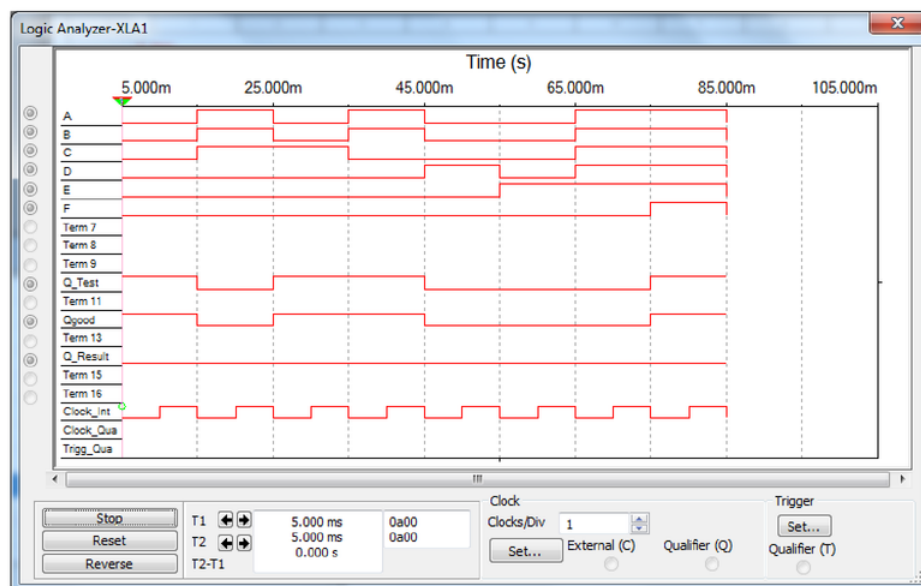


Figure 3.10 - Multisim test results

*Innovation demonstrated using
the word generator.*

4 ANALYSIS OF NI MULTISIM RESULTS

Following the design and simulation of the combination logic circuit using multisim, faults were added to the circuit to confirm that the stuck at fault modelling technique would allow detecting the faults as predicted.

Figure 4.1 shows the circuit modified in order to be able to apply faults to the circuit. As it can be seen below, three faults were added to the circuit, F SA1, D SA0 and A SA1.

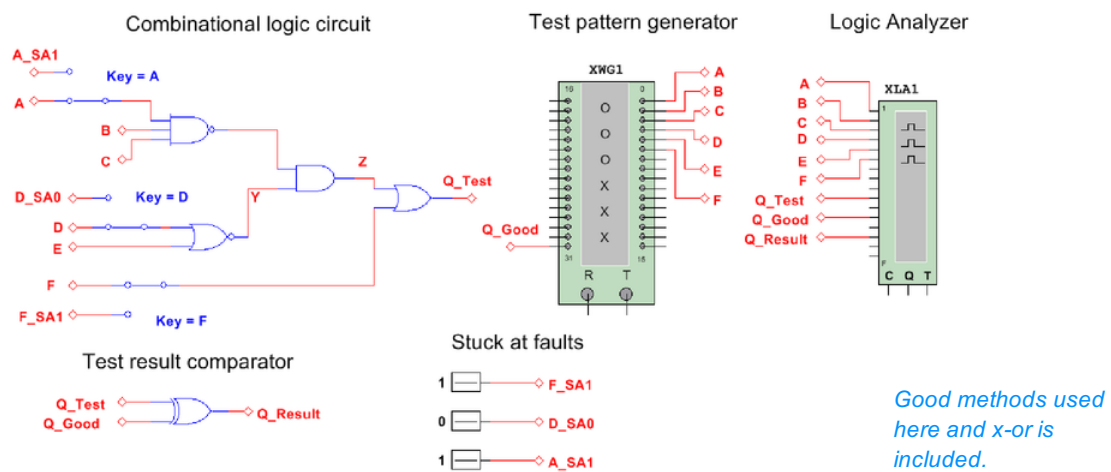


Figure 4.1 - Multisim circuit design for stuck at fault testing

The circuit was simulated after applying each of the three stuck at faults in order to ensure that the fault was detected.

Figure 4.2 shows the simulations results of the fault A stuck at '1'. As it can be seen below, a fault was detected during test number 2 as expected.

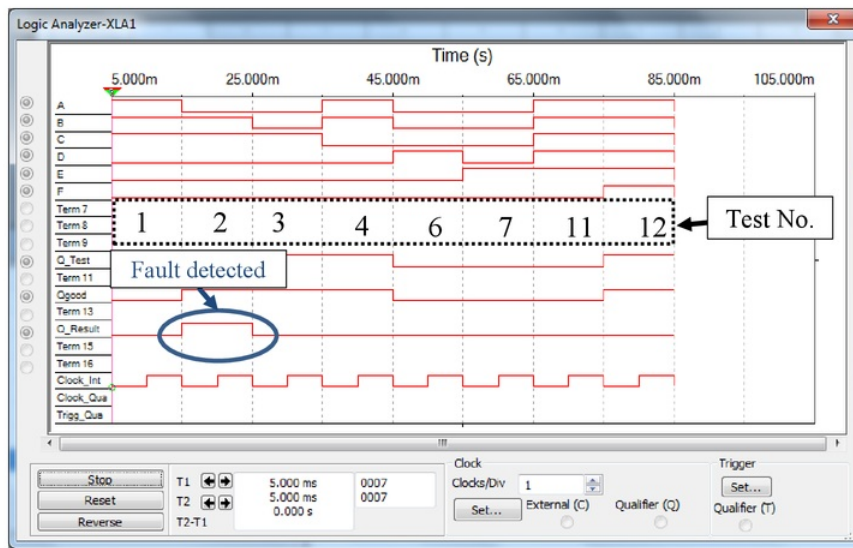


Figure 4.2 - A stuck at '1' Multisim test results

Figure 4.3 shows the simulations results of the fault D stuck at '0'. As it can be seen below, a fault was detected during test number 6 as expected.

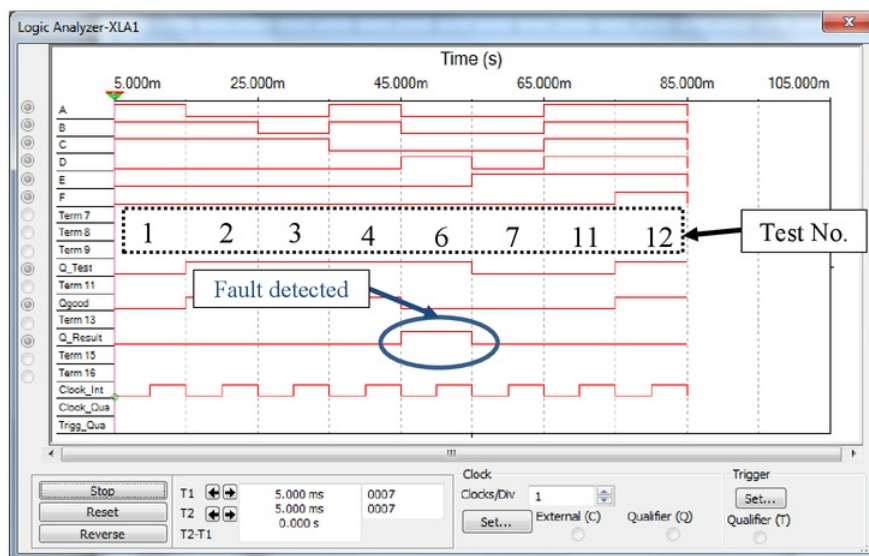


Figure 4.3 - D stuck at '0' Multisim test results

Figure 4.4 shows the simulations results of the fault F stuck at '1'. Although a fault was detected during test number 11 as expected. To be noted is the fact that faults were also detected during test number 1, 6 and 7.

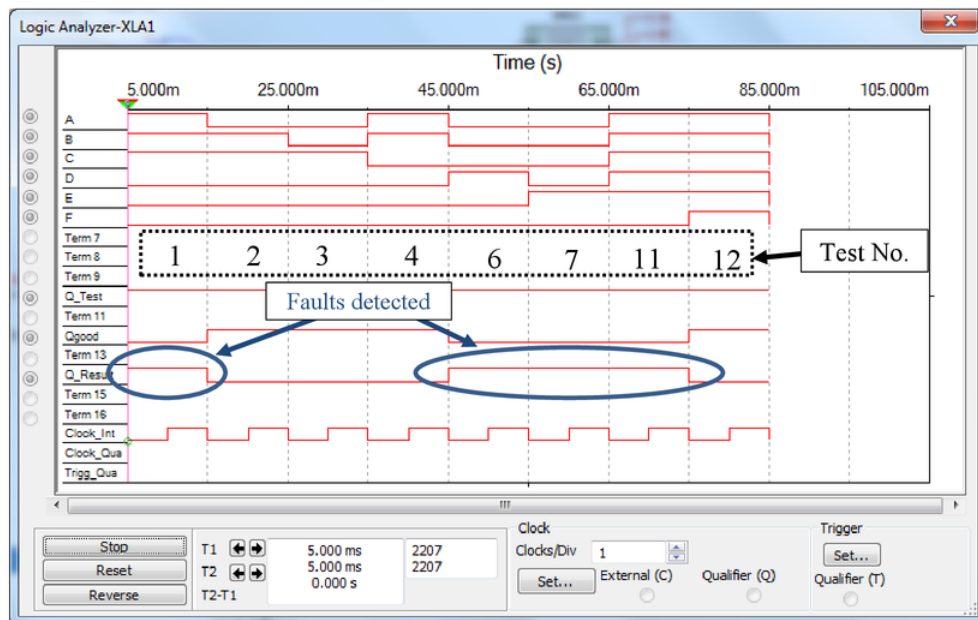


Figure 4.4 - F stuck at '1' Multisim test results

Good analysis demonstrated.

5 6-BIT REGISTER DESIGN AND TEST

Following the design and test of the combinational logic circuit using the word generator for the test pattern, a 6-bit register was designed and tested as an alternative method for automatic test equipment (ATE).

Figure 5.1 shows the state table, or the transition table, required for the design of the finite state machine that produces the test vectors, A to F, that will be used as inputs and outputs to the D type flip-flops.

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 5.1 – State table

Table contains the next state inputs to the D type registers, the outputs will then drive the circuit under test inputs.

In order to determine the Boolean expression for each input Karnaugh maps were developed for each flip-flop as follows.

Figure 5.2 shows the Karnaugh map developed for the Output 'QA' of the first flip-flop.

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

Figure 5.2 - Karnaugh map - Output QA

Below is the simplified Boolean expression obtained from the above Karnaugh map.

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

Figure 5.3 shows the Karnaugh map developed for the Output 'QB' of the first flip-flop.

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

Figure 5.3 - Karnaugh map - Output QB

Below is the simplified Boolean expression obtained from the above Karnaugh map.

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

Figure 5.4 shows the Karnaugh map developed for the Output 'QC' of the first flip-flop.

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

Figure 5.4 - Karnaugh map - Output QC

Below is the simplified Boolean expression obtained from the above Karnaugh map.

$$\overline{N}A.\overline{N}B.\overline{N}C.\overline{N}D.E.NF + \overline{N}A.B.C.\overline{N}D.NE.NF + A.B.C.\overline{N}D.NE.NF + A.B.C.D.E$$

Figure 5.5 shows the Karnaugh map developed for the Output 'QD' of the first flip-flop.

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

Figure 5.5 - Karnaugh map - Output QD

Below is the simplified Boolean expression obtained from the above Karnaugh map.

$$\overline{N}A.\overline{N}B.\overline{N}C.\overline{N}D.E.NF + A.B.\overline{N}C.\overline{N}D.NE.NF + A.B.C.D.E.NF$$

Figure 5.6 shows the Karnaugh map developed for the Output 'QE' of the first flip-flop.

A B C D E F	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

Figure 5.6 - Karnaugh map - Output QE

Below is the simplified Boolean expression obtained from the above Karnaugh map.

$$\overline{N}A.NB.NC.ND.E.NF + \overline{N}A.NB.NC.D.NE.NF + A.B.C.D.E.NF$$

Figure 5.7 shows the Karnaugh map developed for the Output 'QE' of the first flip-flop.

A B C D E F	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

Figure 5.7 - Karnaugh map - Output QF

Below is the simplified Boolean expression obtained from the above Karnaugh map.

$$A.B.C.D.E.NF$$

Good work using the k-map method for cct reduction and clear use of the mirror method used in 6 variable models.

Figure 5.8 shows the 6-bit pattern generator circuit designed using Multisim.

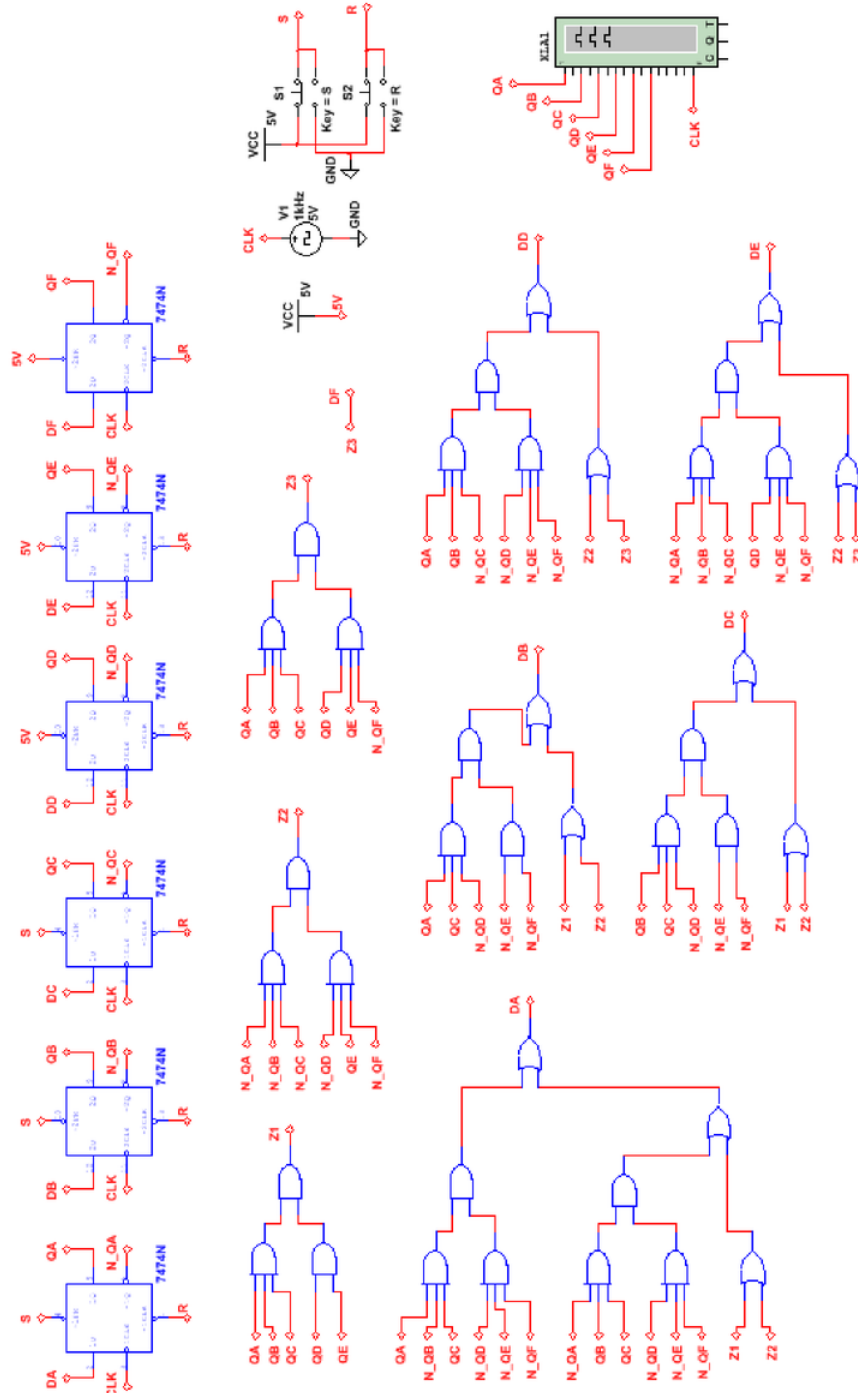


Figure 5.8 - Multisim 6-bit pattern generator

Figure 5.9 shows the test results obtained from the simulation of the 6-bit pattern generator designed using Multisim. As it can be seen below, the test pattern generated contains the eight test vectors required for testing of the circuit. The pattern obtained was the same as that shown in the state table of Figure 5.1 and therefore as expected.

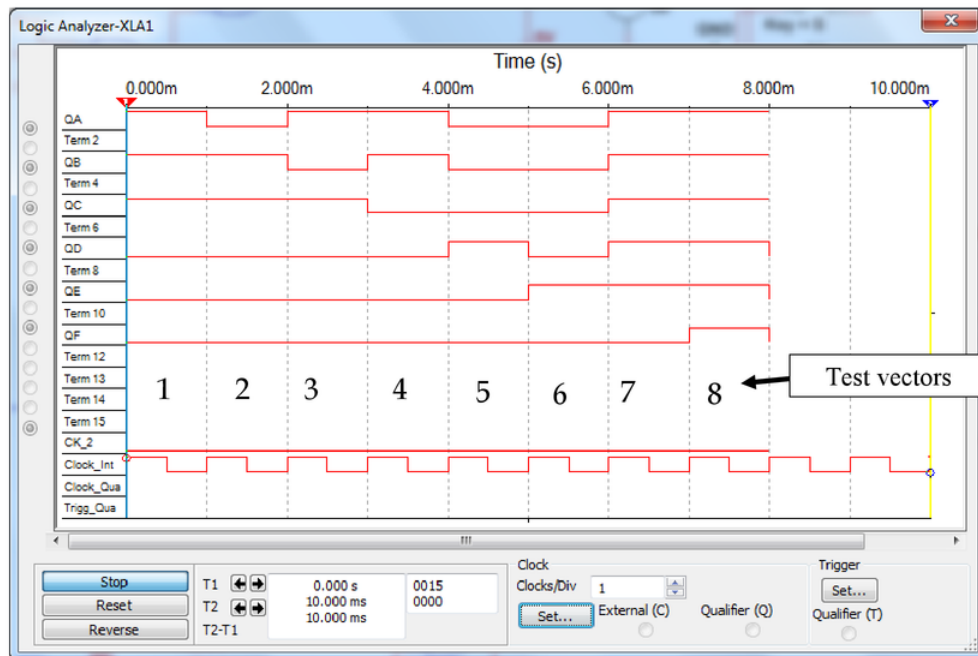


Figure 5.9 – Test results -6-bit pattern generator

Good test result showing the derived 8 test vectors.

6 PRACTICAL BUILD AND TEST USING LABVIEW

Both the combinational logic circuit and stuck at fault modelling technique using Multisim, it was practically built and tested using LabVIEW and the NI ELVIS II development board.

6.1 Design

Figure 6.1 shows the block diagram of the Digital Outputs designed using LabVIEW. As it can be seen below, a 'while loop' and a 'for loop' were designed. The 'while loop' contains two arrays of inputs, one of the values of the test vectors and the other of the timing between each test vector.

The 'for loop' allows the test vectors to be transferred to the NI ELVIS II development board, and the conversion of the digital inputs into decimal values that can be displayed to the user.

The reset button within the 'for loop', when pressed, allows the sequence to be reset and therefore return to zero, while the stop button inside the 'while loop', allows the sequence to be executed once and then return to zero.

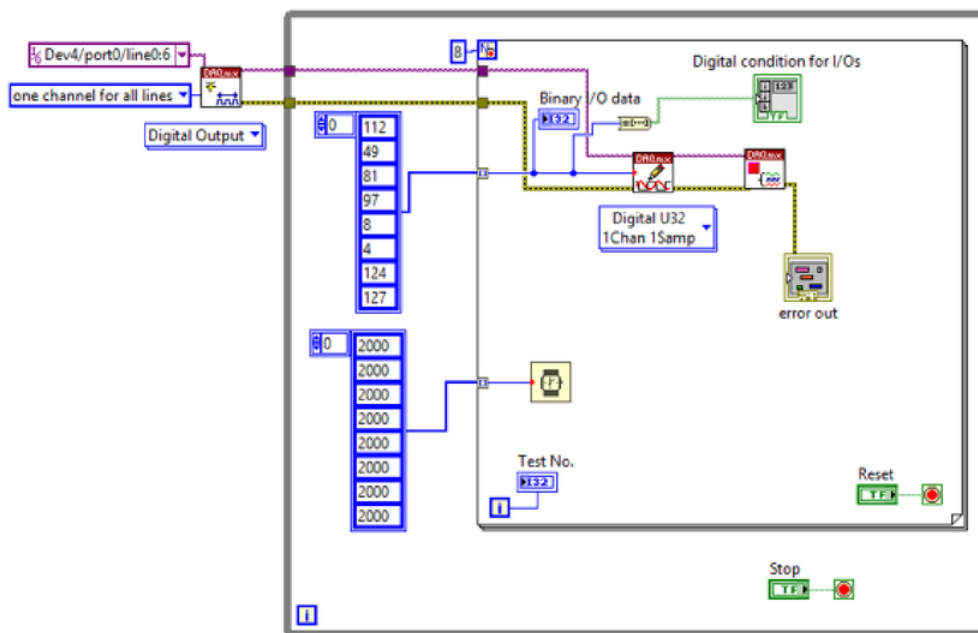


Figure 6.1 - LabVIEW block diagram (Digital Outputs)

Figure 6.2 shows the block diagram of the Digital Inputs designed using LabVIEW. As it can be seen below, the while loop allow receiving the line 23 output from X-OR gate, which is read and displayed on the front panel. This digital signal allow displaying to the user if a fault has been detected via the fault LED.

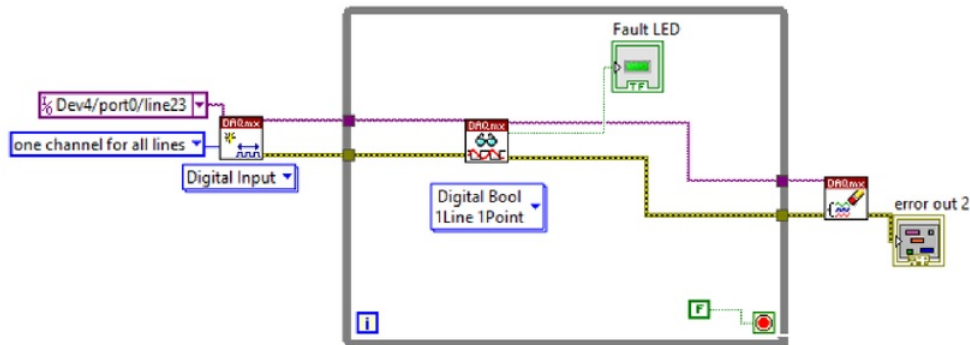


Figure 6.2 - LabVIEW logic diagram (Digital Inputs)

Figure 6.3 below shows the front panel designed using LabVIEW in order to control the ATE and display the test result to the user via the fault LED. As it can be seen below, the front panel also shows the test number, the status of the six inputs, A to F, the status of the output Q from the X-OR gate, and the decimal value corresponding to the binary I/O data. From the front panel the test sequence can also be stopped or resetted.

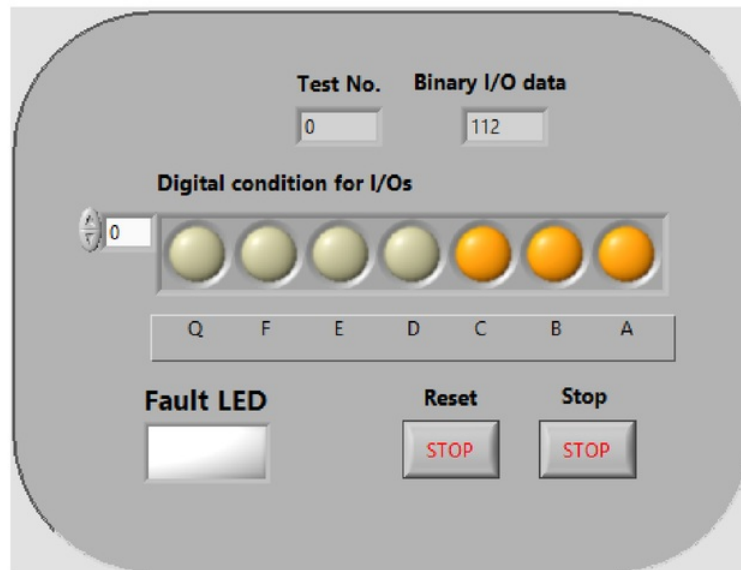


Figure 6.3 - LabVIEW front panel

Figure 6.4 shows the NI ELVIS development board used for building the combinational logic circuit to be tested using LabVIEW.

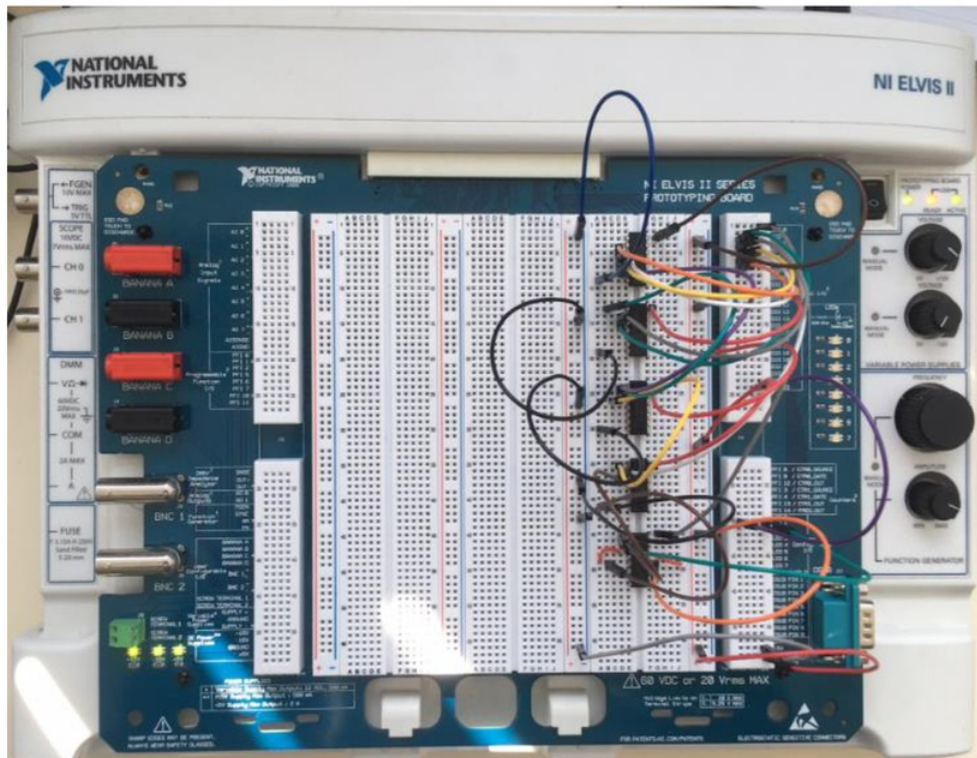


Figure 6.4 - NI ELVIS II development board

Clear
illustration of
the ELVIS
development
circuit.

6.2 Testing

In order to confirm the correct functioning of the built circuit a test of the good circuit was carried out. Figure 6.5 shows the results obtained from the test.

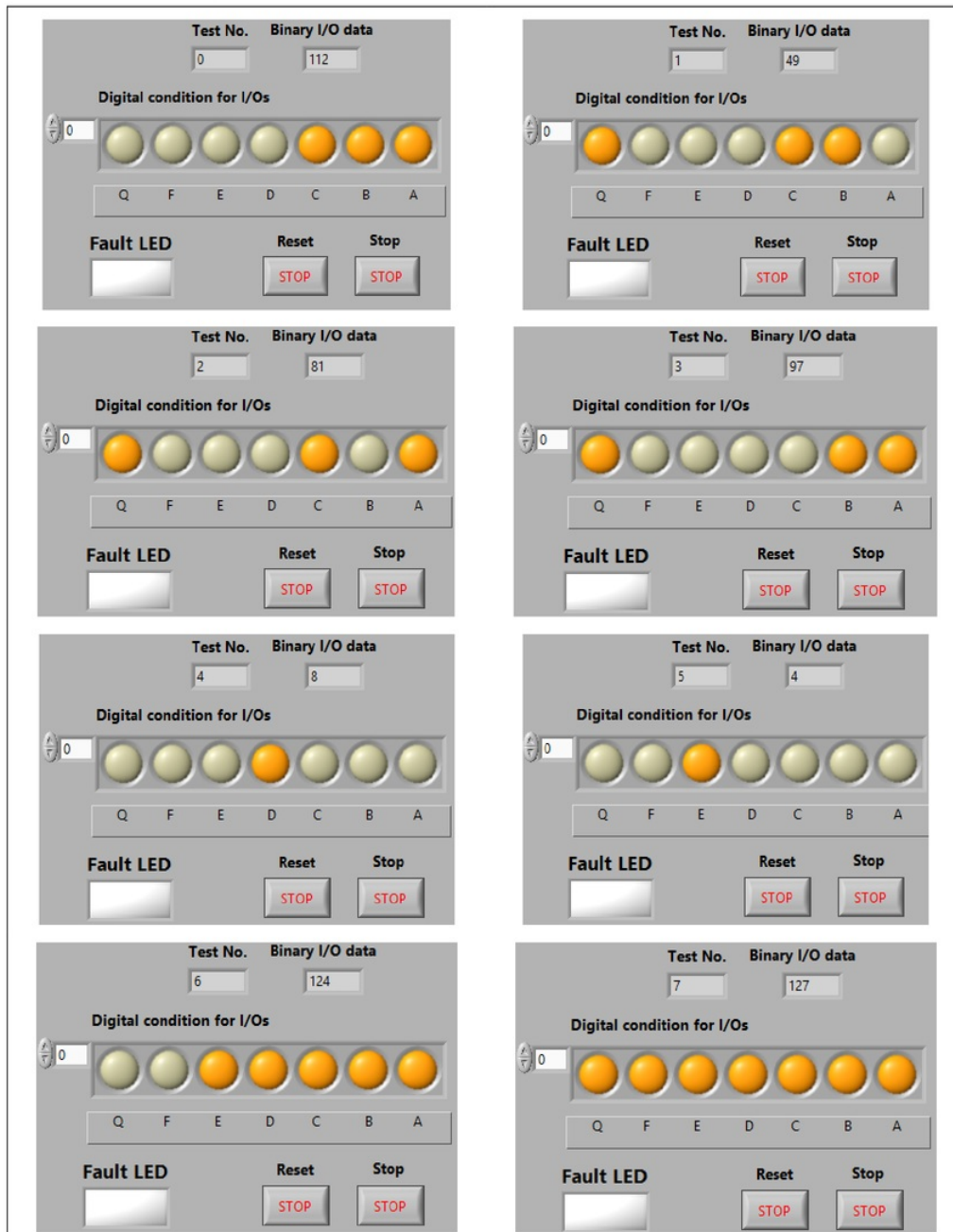


Figure 6.5 - LabVIEW test results

After testing the good circuit, the three faults applied to the Multisim circuit were applied to the practically built circuit.

Figure 6.6 shows the fault obtained during the A stuck at '1' test. As it can be seen below, the fault obtained was the same as that obtained in the multisim simulation shown in Figure 4.2.

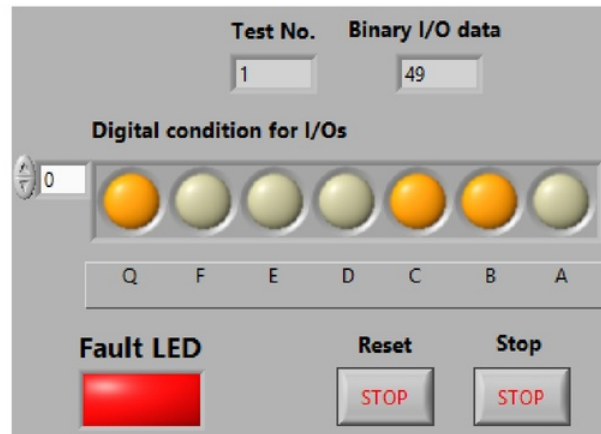


Figure 6.6 - A stuck at '1' LabVIEW test result

Figure 6.7 shows the fault obtained during the D stuck at '0' test. As it can be seen below, the fault obtained was the same as that obtained in the multisim simulation shown in Figure 4.3.

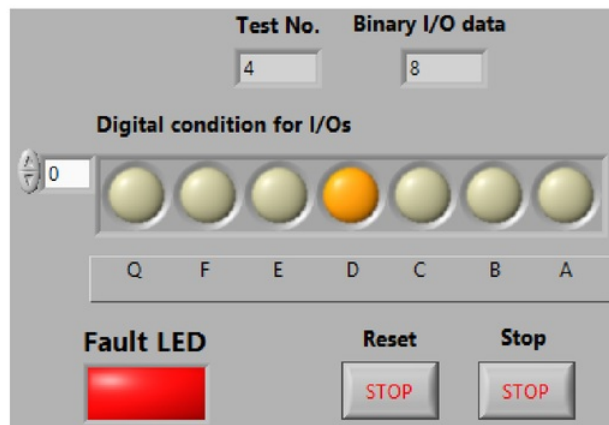


Figure 6.7 - D stuck at '0' LabVIEW test result

Figure 6.8 shows the faults obtained during the F stuck at '1' test. As it can be seen below, the faults obtained was the same as that obtained in the multisim simulation shown in Figure 4.4.

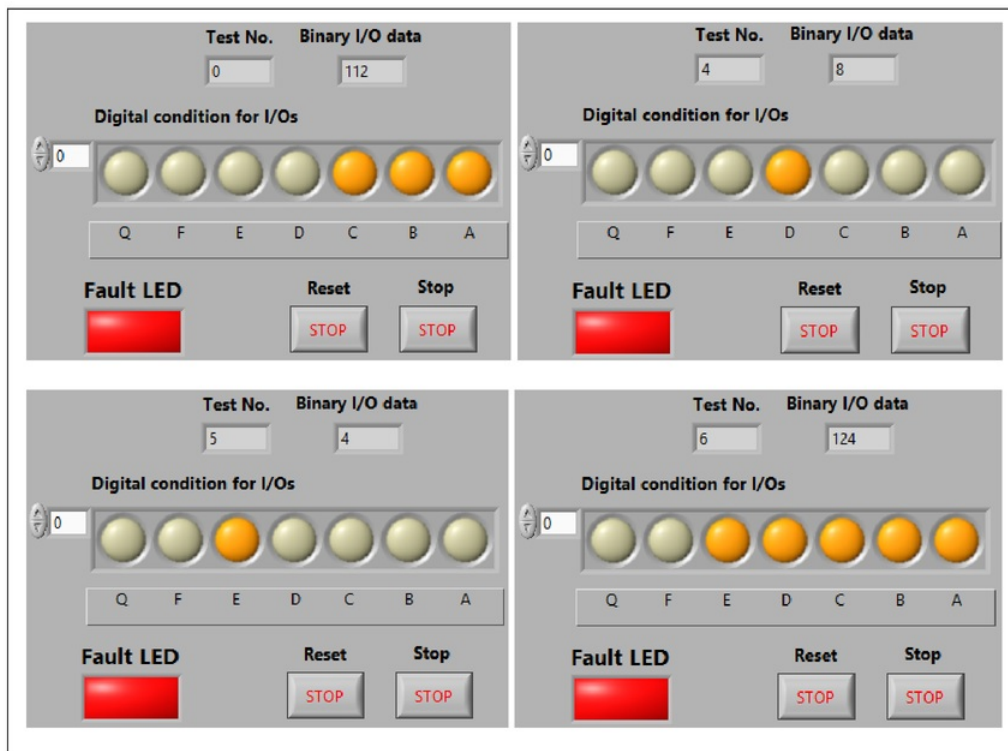


Figure 6.8 - F stuck at '1' LabVIEW test results

Very good analysis here and a good demonstration using LabView with DAQMX systems.

7 CONCLUSION

In this assignment it has been analysed and evaluated the stuck at fault modelling technique required for testing integrated circuits.

Research was carried out into IC fabrication faults, fault-modelling techniques, and the dangers of electrostatic discharge including how it can be prevented during the design and manufacturing process of electronics devices.

Following the research, a combinational logic circuit was designed and tested using Multisim. After applying vector reduction techniques using the SSF model simulation tests were performed by applying stuck at faults.

As an alternative of ATE a 6-bit pattern generator was designed in order to produce the test vectors required for the automatic testing of the designed circuit.

The circuit was then build and tested using NI LabVIEW and the Elvis development board.

8 REFERENCE LIST

Stack Exchange Inc., 2018. *To avoid ESD when you are working on computers, what should you attach your wristband to?*. [Online]

Available at: <<https://electronics.stackexchange.com/questions/22474/to-avoid-esd-when-you-are-working-on-computers-what-should-you-attach-your-wristband-to>> [Accessed 7th April 2018].

Anon., n.d. [Online]

Available at: <http://users.ece.utexas.edu/~roth/book/CH10_Slides.pdf> [Accessed 7th April 2018].

Desco, 2013. *Images of ESD Damage*. [Online]

Available at: <<https://desco.blog/2013/07/09/images-of-esd-damage-2/>> [Accessed 7th April 2018].

Fall, 2002. *Fault models, fault collapsing and logic simulation*. [Online]

Available at: <<http://web.eecs.umich.edu/~mazum/F02/lectures/lec mazum3.pdf>> [Accessed 5th April 2018].

Lima Kastensmidt, F., 2008. *Lecture 2 - Fault Modeling*. [Online]

Available at: <<http://web.eecs.umich.edu/~mazum/F02/lectures/lec mazum3.pdf>> [Accessed 6th April 2018].

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

Available at: <<https://whatis.techtarget.com/definition/electrostatic-discharge-ESD>> [Accessed 7th April 2018].

SlidePlayer, n.d. *Electrostatic Discharge*. [Online]

Available at: <<http://slideplayer.com/slide/3538727/>> [Accessed 7th April 2018].

SlidePlayer, n.d. *Fault Modeling*. [Online]

Available at: <<http://slideplayer.com/slide/10218816/>> [Accessed 7th April 2018].

Storr, P., 1998. *ElectroStatic Discharge (ESD) precautions*. [Online]
Available at: <<http://philipstorr.id.au/pcbook/book5/esd.htm>>
[Accessed 7th April 2018].

Good
referen
ces
used
and a
good
numbe
r of
source
s.

Assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

73/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

Text Comment. Good list of contents

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

Text Comment. Good use of tables for examples.

PAGE 10

PAGE 11

Text Comment. Very good research carried out with good explanations. Good use of relevant sources.

PAGE 12

PAGE 13

Text Comment. AND o/p should be: $Z = 1\ 0\ 0$

Text Comment. OR gate output should be: $Q = 0\ 1\ 1$

PAGE 14

Text Comment. Test model outputs are correct in table.

PAGE 15

Text Comment. Table outputs are good. The reduction method has been implemented.

PAGE 16

Text Comment. Innovation demonstrated using the word generator.

PAGE 17

Text Comment. Good methods used here and x-or is included.

PAGE 18

PAGE 19

Text Comment. Good analysis demonstrated.

PAGE 20

Text Comment. Table contains the next state inputs to the D type registers, the outputs will then drive the circuit under test inputs.

PAGE 21

PAGE 22

PAGE 23

Text Comment. Good work using the k-map method for cct reduction and clear use of the mirror method used in 6 variable models.

PAGE 24

Text Comment. This looks correct. The x-or should be included.

PAGE 25

Text Comment. Good test result showing the derived 8 test vectors.

PAGE 26

PAGE 27

PAGE 28

Text Comment. Clear illustration of the ELVIS development circuit.

Text Comment. Very good analysis here and a good demonstration using LabView with DAQMX systems.

Text Comment. Reasonable conclusion, but could have given your view on the reliability of methods used.

Text Comment. Good references used and a good number of sources.

GRADING FORM: CSG-RUBIC

BRUNO FORKUOH

0

RESEARCH-INFORMED LITERATURE



Good effort shown here and a good range of relevant sources used. IC fabrication - 3.5/5 ESD - 3.5/5 SAF - 4/5

11/15

KNOWLEDGE AND UNDERSTANDING



A good understanding demonstrated, but with some minor errors which did not carry through to effect the tables. Derive minimum test vectors - good despite some minor errors with gate identification models- 6/10 LabVIEW test vectors - good

21/30

explanations of design - 8/10 Multisim PLD

alternative - good - 7/10

ANALYSIS



Analysis is good and everything ties up nicely with good explanations. EDAD - test results are good - 7/10 Comparison to theory is good - 3/5

10/15

PRACTICAL APPLICATION AND DEP.



Application is good with some ingenuity demonstrated with the word generator. Prototyping good - 8/10 LabVIEW design is good with good explanation - 13/15

21/25

SKILLS FOR PROFESSIONAL PRAC.



Report is good, conclusion could have been more comparative for effectiveness. Deployment of software is good - 3.5/5 Report layout is very good - 3.5/5 Conclusion is good - 3/5

10/15