

Report

by Bruno Forkuoh

FILE	7838_BRUNO_FORKUOH_REPORT_155126_1555920703.DOCX (9.76M)		
TIME SUBMITTED	15-FEB-2018 02:11PM (UTC+0000)	WORD COUNT	5845
SUBMISSION ID	83218210	CHARACTER COUNT	40374

Bachelor of Science in Electrical & Electronic Engineering



Electronic Production & Test

PSDE 6018

Pseudo Random Pattern Generators

Bruno Forkuoh

TABLE OF CONTENTS

Table of Figures.....	4
1 Introduction.....	7
2 Research.....	8
2.1 PCB technology	8
2.2 Design process.....	10
2.3 Manufacturing process.....	11
2.4 Mounting components	13
2.4.1 PTH and SMT technology.....	13
2.5 Soldering process	14
2.5.1 Hand soldering.....	14
2.5.2 Wave soldering.....	15
2.5.3 Reflow soldering	16
2.6 Inspection/testing process	17
2.6.1 Automatic optical inspection (AOI)	17
2.6.2 Automatic X-Ray inspection (AXI).....	18
2.6.3 In-circuit test (ICT).....	19
3 Design of assignment requirements using Multisim	20
3.1 4-bit CAR 90.....	21
3.2 4-bit CAR 150.....	23
3.3 4-bit hybrid CAR 90 / 150.....	25
3.4 6-bit Hybrid CAR 90 / 150.....	27
4 Analysis of NI MultiSim results.....	30
4.1 Test circuit.....	30
4.2 Fault 1	31

4.3	Fault 2	32
4.4	Fault 3	33
4.5	Summary of Multisim test results	34
5	DESIGN OF ASSIGNMENT REQUIREMENT USING VHDL & IP DESIGN	35
5.1	4-bit CAR 90	35
5.2	4-bit CAR 150	36
5.3	4-bit hybrid CAR 90 / 150.....	36
5.4	6-bit hybrid CAR 90 / 150.....	37
6	Analysis of results using hardware descriptive language FPGA technology	41
6.1	Fault 1	42
6.2	Fault 2	44
6.3	Fault 3	45
6.4	Summary of FPGA test results.....	45
7	PCB design	46
8	Practical PCB build and testing of prototype on prototyping board.....	49
9	Conclusion.....	51
10	Reference List.....	52
11	Appendix	55
11.1	Clock	55
11.2	6-bit hybrid.....	56
11.3	Inverter gate.....	57
11.4	And gate	57
11.5	Or gate	58
11.6	X-or gate	58
11.7	6-bit hybrid CAR 90 / 150.....	59

TABLE OF FIGURES

Figure 2.1 - Single sided PCB (Anon., 2013).....	8
Figure 2.2 - Double sided PCB (robotroom, 2017).....	9
Figure 2.3 - Multi layered PCB (SAE Circuits, 2015).....	9
Figure 2.4 - PCB design software (Linkedin, 2015)	10
Figure 2.5 - PCB drilling machine (Kush , 2014).....	11
Figure 2.6 - Copper plate (Advameg Inc., 2014).....	12
Figure 2.7 - Tin-lead plate (Advameg Inc., 2014)	12
Figure 2.8 - Etched board (Advameg Inc., 2014)	12
Figure 2.9 - Solder mask (Advameg Inc., 2014)	12
Figure 2.10 - SMT technology (Streamline Electronics Manufacturing, 2017)	13
Figure 2.11 - Hand soldering (Sparkfun Electronics, n.d.).....	14
Figure 2.12 - Lead free wave solder thermal profile (Projects Unlimited, 2014).....	15
Figure 2.13 - Lead free reflow soldering thermal profile (Projects Unlimited, 2014) .	16
Figure 2.14 - Automatic optical inspection (Blundell Production Equipment, n.d.)...	17
Figure 2.15 - Automatic X-Ray inspection (nordsonmatrix, n.d.).....	18
Figure 2.16 - In circuit tester (PARPRO Technologies , 2016).....	19
Figure 3.1 - Typical CAR structure	20
Figure 3.2 - CAR 90 rule	21
Figure 3.3 - CAR 90 multisim circuit	21
Figure 3.4 - CAR 90 multisim simulation results	22
Figure 3.5 - CAR 150 rule	23
Figure 3.6 - CAR 150 multisim circuit	23
Figure 3.7 - CAR 150 simulation results	24
Figure 3.8 - 4-bit hybrid CAR 90 / 150 multisim circuit.....	25

Figure 3.9 - 4-bit hybrid CAR 90 / 150 simulation results	26
Figure 3.10 - 6-bit hybrid CAR 90 / 150 multisim circuit.....	27
Figure 3.11 - 6-bit hybrid CAR 90 / 150 simulation results	28
Figure 3.12 - 6-bit hybrid CAR 90 / 150 sequence	29
Figure 4.1 - Multisim test circuit	30
Figure 4.2 - Multisim test circuit simulation results	30
Figure 4.3 - Fault 1 - multisim circuit	31
Figure 4.4 - Fault 1 - multisim simulation results	31
Figure 4.5 - Fault 2 - multisim circuit	32
Figure 4.6 - Fault 2 - multisim simulation results.....	32
Figure 4.7 - Fault 3 - multisim circuit.....	33
Figure 4.8 - Fault 3 - multisim simulation results.....	33
Figure 4.9 - Multisim simulation - summary of test results.....	34
Figure 5.1 - 4-bit CAR 90 Vivado simulation results.....	35
Figure 5.2 - 4-bit CAR 150 Vivado simulation results	36
Figure 5.3 - 4-bit hybrid CAR 90 / 150 Vivado simulation results	36
Figure 5.4 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 1)	37
Figure 5.5 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 2)	37
Figure 5.6 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 3)	38
Figure 5.7 - IP integrator design	39
Figure 5.8 - CAR 90 / 150 hybrid equation	40
Figure 6.1 - Good circuit test - no fault.....	41
Figure 6.2 - Good circuit test - fault detected.....	42
Figure 6.3 - Testing of prototype circuit	43
Figure 6.4 - Testing of prototype circuit - no fault.....	43

Figure 6.5 - Prototype testing - fault 1.....	44
Figure 6.6 - Prototype testing - fault 2.....	44
Figure 6.7 - Prototype testing - fault 3.....	45
Figure 6.8 - FPGA testing - summary of results	45
Figure 7.1 - EAGLE PCB logic circuit.....	46
Figure 7.2 - PCB top layer	46
Figure 7.3 - PCB bottom layer	47
Figure 7.4 - Top and bottom layer	47
Figure 7.5 - Gerber viewer	48
Figure 8.1 - Top layer.....	49
Figure 8.2 - Bottom layer	49
Figure 8.3 - PCB testing - fault detected	50
Figure 8.4 - PCB testing - no fault detected	50

1 INTRODUCTION

The aim of this assignment is to analyse and evaluate a process for developing a prototype design into a fully manufactured and tested product that meets the user or customer needs.

This will involve investigating and understanding the use of PCB technology, manufacturing and soldering processes, and demonstrating an understanding of automatic Test Equipment for IC and PCB systems.

Intro is a little weak. The focus should be on the virtual ATE integration.

2 RESEARCH

2.1 PCB technology

Printed circuit boards, also known as PCBs, are thin boards made of insulated material, known as substrate, which support and connect electronic components via conductive tracks and pads, known as circuits (Advameg Inc., 2014).

They are used to make permanent, soldered circuits (Hewes, 2017) and can be found in many electronic appliances, such as computers, printers, stereos, digital clocks, microwave ovens, mobile phones and many others (Keen, 2017).

There are three main types of PCBs: single-sided, double-sided and multi-layered.

Single-side PCBs have components on one side of the board.



Figure 2.1 - Single sided PCB (Anon., 2013)

Double side PCBs have components on both sides due to the number of components being too much for one side.

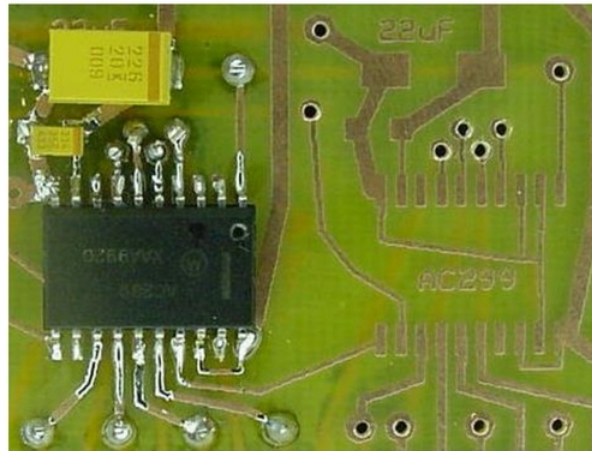


Figure 2.2 - Double sided PCB (robotroom, 2017)

Multi-layered PCBs are made of multiple layers of printed circuits separated by layers of insulated material. The components are often located on the surface and connected through plated holes drilled down to the appropriate circuit layer (Advameg Inc., 2014).

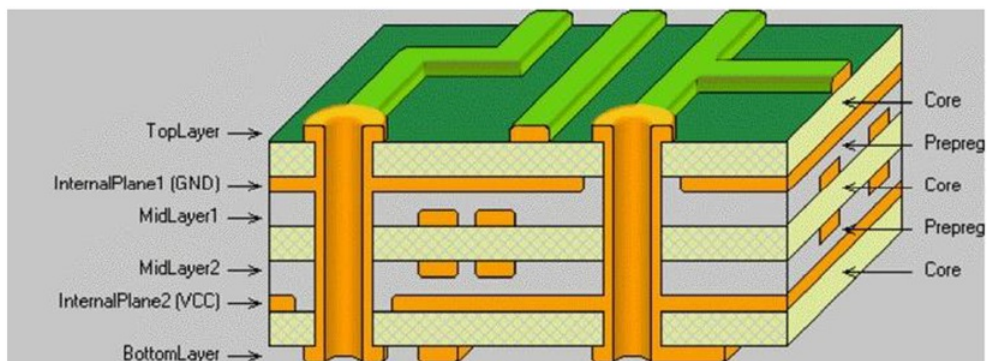


Figure 2.3 - Multi layered PCB (SAE Circuits, 2015)

2.2 Design process

PCB have unique functions for a particular product and for this reason, their design is often unique for each application.

Designing a PCB requires the use of a computer-aided design (CAD) system for the layout of the circuit pattern and the location of the holes on the board as shown in Figure 2.4.

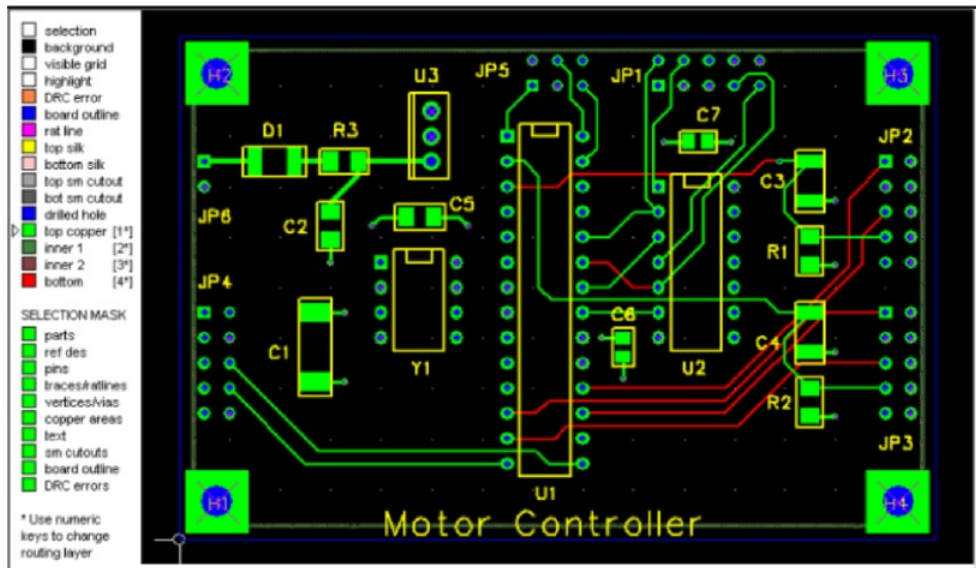


Figure 2.4 - PCB design software (Linkedin, 2015)

The information of the CAD system is then translated in such a form that it can be interpreted by the computer controlled drilling machine or automatic solder pasting machine used in the manufacturing process.

The layout produced by CAD system is also used to print out the circuit pattern in real size on a clear plastic sheet. On this image are shown the areas that are not part of the circuit pattern, in black, and the circuit pattern, in clear (Advameg Inc., 2014).

2.3 Manufacturing process

There are several stages required for the manufacture of a simple PCB. Following is a description of the manufacturing process for a double-sided PCB.

The first stage in the manufacturing process of a PCB involves pressing the copper foils and substrate, made of fiberglass epoxy resin, in order to obtain the PCB board. This sandwich is placed in an oven at about 170°C so that the copper bonds to the resin (Advameg Inc., 2014).

Once the copper foil and substrate are bonded together, a drilling machine drills the holes according to the predetermined pattern.



Figure 2.5 - PCB drilling machine (Kush , 2014)

The inside surfaces of the holes are then plated with copper in order to provide a conductive circuit from one side of the board to the other (Advameg Inc., 2014).

Once the PCB has been pressed, the printed circuit pattern must be created. There are two main processes for creating the printed circuit pattern.

The first one involves etching away the copper foil from the areas that are not part of the desired circuit pattern, while the second one involves adding copper onto the surface of the substrate. To be noted is the fact that when adopting this process, copper must not be bonded to the substrate until this stage.

After the printed circuit pattern has been created, the contact fingers, which connect the electrical components to the copper tracks, are attached to the board (Advameg Inc., 2014).

The next step involved applying a dry-film resist coating on the entire surface of the board, and tin plating the exposed circuit pattern.

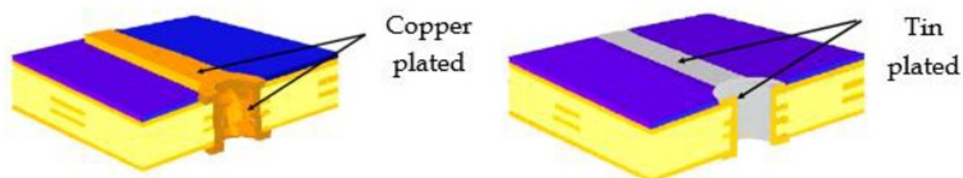


Figure 2.6 - Copper plate (Advameg Inc., 2014)

Figure 2.7 - Tin-lead plate (Advameg Inc., 2014)

The resistive dry film is then removed and the copper is etched from all parts of the board that are not tin plated as shown in Figure 2.8.

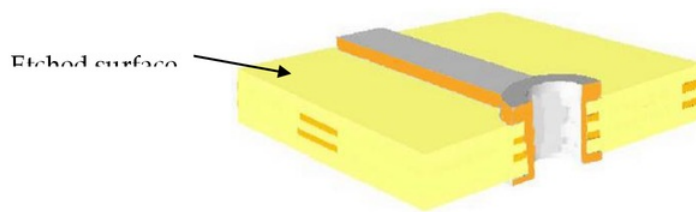


Figure 2.8 - Etched board (Advameg Inc., 2014)

The tin-lead plate is then removed before a photosensitive epoxy based ink is applied on the board as shown in Figure 2.9.

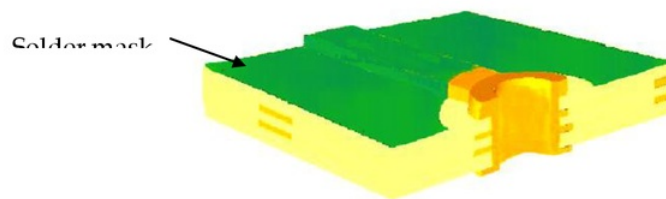


Figure 2.9 - Solder mask (Advameg Inc., 2014)

2.4 Mounting components

Once the board has been coated, plated and etched, it is ready for the components to be mounted. There are two technologies commonly used for mounting components on PCB boards: surface mount technology (SMT) and plated through holes (PTH).

2.4.1 PTH and SMT technology

PTH involves inserting the leads, or ends, of the components through drilled holes and soldering them to the pads on the other side of the board (Trizo, 2014).

SMT technology involves using automated pick and place machines and SMD (Surface Mount Devices), which are smaller and lighter.

SMT and PTH can be combined in a hybrid technology. This involves using machines specifically designed for this purpose. This hybrid technology does not only allow flexibility in the assembly process but it also gives the ability of hand soldering and repair (Trizo, 2014).

Figure 2.10 shows a PCB with both surface mount devices and components that require drilled holes.



Figure 2.10 - SMT technology (Streamline Electronics Manufacturing, 2017)

2.5 Soldering process

The soldering process allows securing the components to the board.

There are three types of soldering techniques commonly used: hand soldering, wave soldering and reflow soldering.

2.5.1 Hand soldering

Hand soldering is required when components are mounted onto the PCB using PTH.

This soldering process requires a skilled technician that hand-solders each component (Trizo, 2014).

Figure 2.11 shows the hand soldering technique previously described.

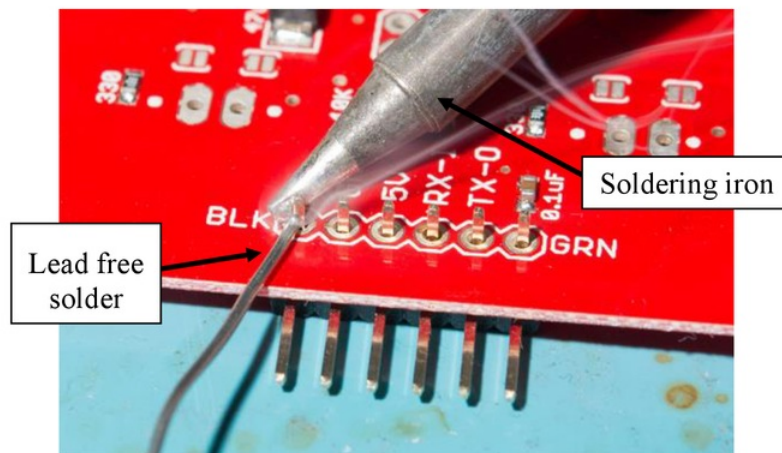


Figure 2.11 - Hand soldering (Sparkfun Electronics, n.d.)

2.5.2 Wave soldering

Wave soldering involves passing the PCB over a wave solder after the components have been glued. It is used for both SMD and PTH.

There are different stages within the wave soldering process. Figure 2.12 shows both the soldering stages and the thermal profile of this process.

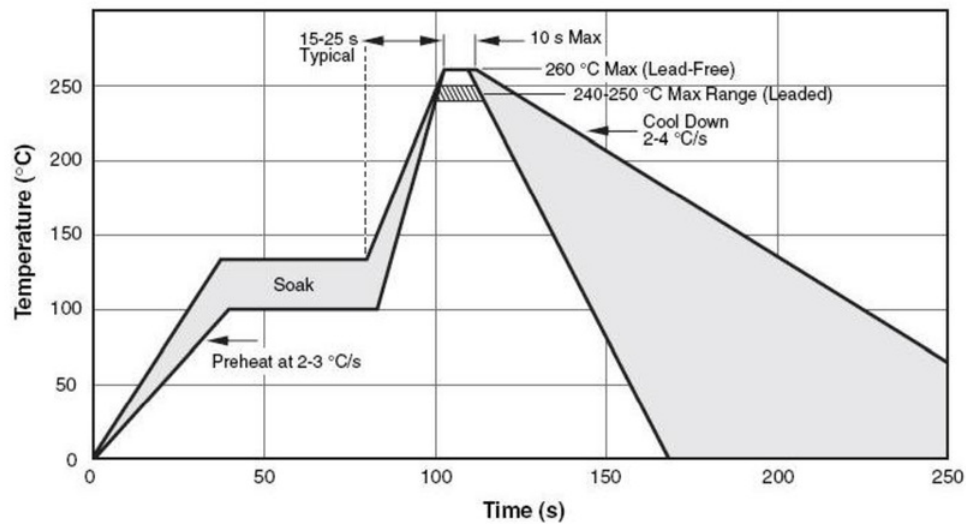


Figure 2.12 – Lead free wave solder thermal profile (Projects Unlimited, 2014)

2.5.3 Reflow soldering

Reflow soldering is the most common soldering process for SMD. It involves applying solder paste using a pick and place machine and subsequently placing the PCB in a reflow oven.

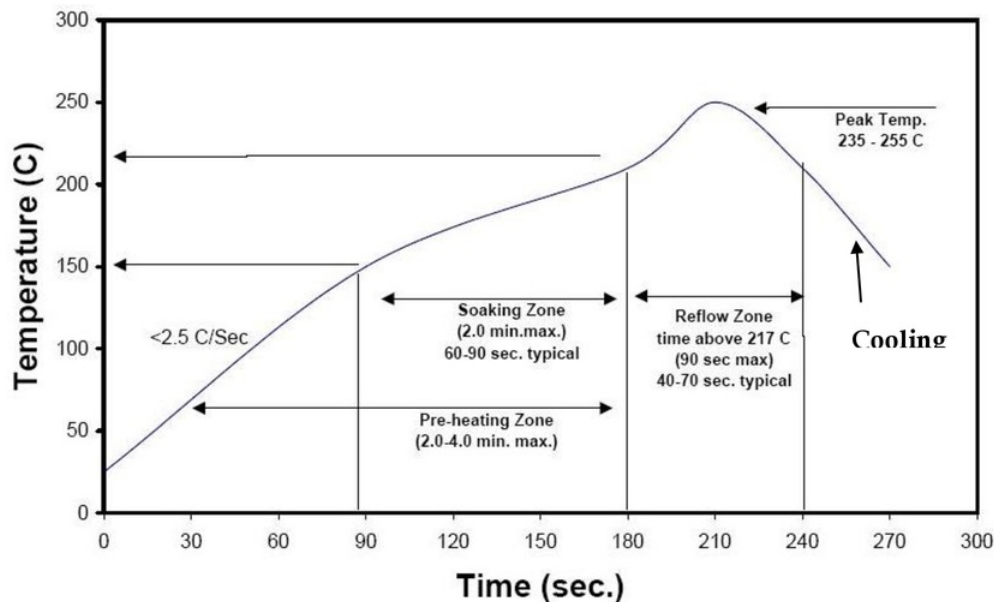


Figure 2.13 – Lead free reflow soldering thermal profile (Projects Unlimited, 2014)

As it can be seen in reflow soldering thermal profile shown in Figure 2.13, there are four stages in the reflow soldering process.

1. **Preheat** – Involves a rapid increase to the temperature set point.
2. **Soaking** – Allows the solder paste to adapt to the heat and the removal of solder paste volatiles.
3. **Reflow** – Occurs when the solder reaches the temperature where it turns liquid.
4. **Cooling** – Allows the solder paste to cool in controlled stages for solid fixing.

Failure to follow this profile can cause damage to components.

2.6 Inspection/testing process

There are three main inspection methods currently used for testing PCB and detecting defects. These are:

- Automatic optical inspection (AOI)
- Automatic X-ray inspection (AXI)
- In-circuit test (ICT)

2.6.1 Automatic optical inspection (AOI)

AOI is a fast, accurate and reliable inspection process for PCBs, commonly used after the soldering process. It allows detecting faults due to incorrect construction and therefore ensuring that the products meet the quality standards (electronicnotes, n.d.).



Figure 2.14 - Automatic optical inspection (Blundell Production Equipment, n.d.)

This inspection method involves scanning the surface of the board in order to detect defects such as scratches, stains, open circuits, shorts, thinning of solder, incorrect or missing components, and incorrectly placed components.

There are three main AOI techniques used for detecting defects:

1. **Template matching process** involves comparing the image to a proven good product.
2. **Pattern matching process** involves storing the information of a good and bad image and then comparing the product to these sets of information.
3. **Statistical pattern matching process**, which is similar to pattern matching, but involves storing images instead of information. (electronicnotes, n.d.)

2.6.2 Automatic X-Ray inspection (AXI)

AXI is mainly used for the inspection of solder joints, missing components and solder under components, as well as open circuits and shorts. It allows detecting faults where solder connections are not visible. It is also able to provide an internal view of the solder joints making possible the detection of voids (electronicsnotes, n.d.).

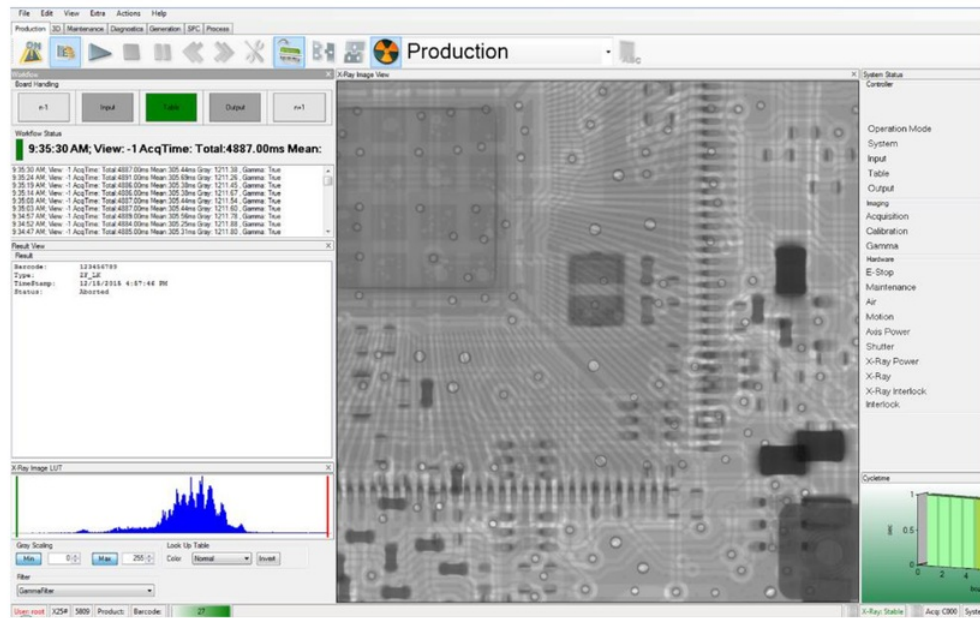


Figure 2.15 - Automatic X-Ray inspection (nordsonmatrix, n.d.)

2.6.3 In-circuit test (ICT)

ICT is a PCB testing method that ensures that the board and its components are performing to its specification. It involves accessing the circuit nodes and measuring parameters such as resistance and capacitance.

An ICT equipment normally use fixtures, known as bed of nails, designed specifically for a particular board. These acts as an interface between the board and the circuit tester.

Software is written for each board type and it contains information relating to the tests to perform, the points and the pass/fail criteria (electronicsnotes, n.d.).



Figure 2.16 - In circuit tester (PARPRO Technologies , 2016)

This test method is used because it allows easy detection of manufacturing defects, the software is easy to create and the test results easy to interpret. However, the fixtures can be expensive and difficult to modify and update (electronicsnotes, n.d.).

3 DESIGN OF ASSIGNMENT REQUIREMENTS USING MULTISIM

In order to achieve the objective of the assignment a pseudo random pattern generator was developed using CAR (Cellular Automata Register). CAR systems involve flip-flops whose next state is determined by the previous state and the value of the neighbouring flip-flop.

The outputs are fed into XOR gates and the output of the XOR gate is fed back into the input of the flip-flop.

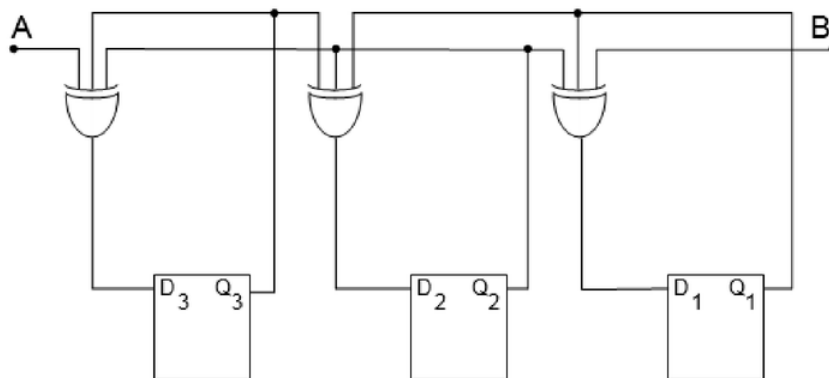


Figure 3.1 - Typical CAR structure

In order to meet the assignment requirements a 6-bit hybrid CAR was developed by combining CAR 90 and CAR 150 rules using the LFSR (Linear Feedback Shift Register) principle and exclusive or gate. This CAR system allowed achieving a maximal length and unrepeatable sequence of random numbers, as required.

During the development of the hybrid CAR system three different systems were developed and analysed:

- 4 bit CAR 90
- 4 bit CAR 150
- 4 bit hybrid CAR 90 / CAR 150

3.1 4-bit CAR 90

The 4 bit CAR 90 was achieved with an arrangement of four flip-flops where the current state of each flip-flop is determined by taking the XOR of its two immediate neighbours.

Figure 3.2 shows equation for the current state of each flip-flop.

$$s_i(t+1) = s_{i+1}(t) \oplus s_{i-1}(t)$$

Figure 3.2 - CAR 90 rule

Figure 3.3 shows the Multisim circuit for a 4-bit CAR 90 and its pseudorandom sequence. As it can be seen below this arrangement was not maximal count as the sequence was 7 numbers without repeats, and not 16 as should be achieved with 4 bits.

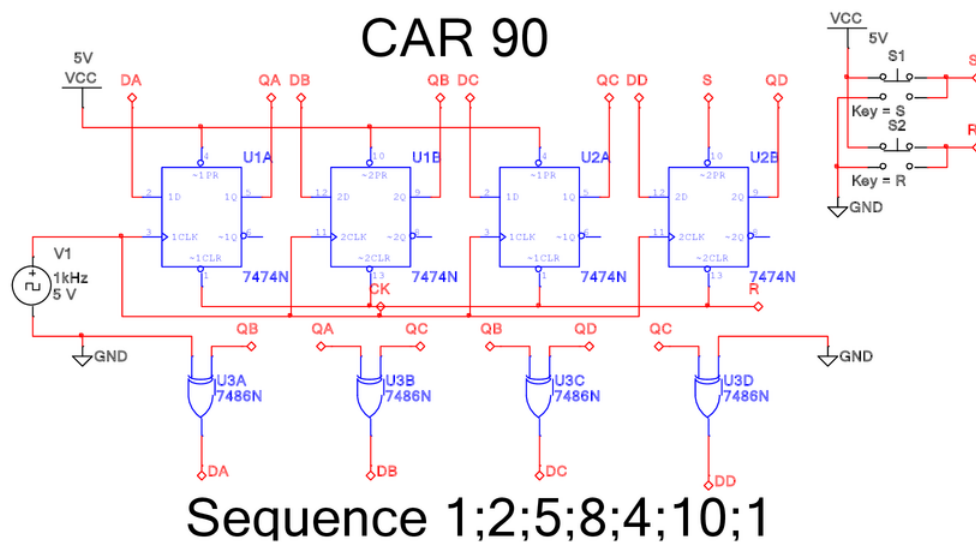


Figure 3.3 - CAR 90 multisim circuit

As it can be seen in the above circuit all '2PR' preset pins of the flip-flops were connected to 5V a part from that of the flip-flop QF, which was connected to the switch 'S1'. The purpose of this configuration is to allow the pseudorandom generator to start at the seed value, which in this case corresponds to a logic high for flip-flops QA, QB, QC and logic low for flip-flop QD.

Figure 3.4 shows the results obtained when simulating the 4-bit CAR 90 arrangement.

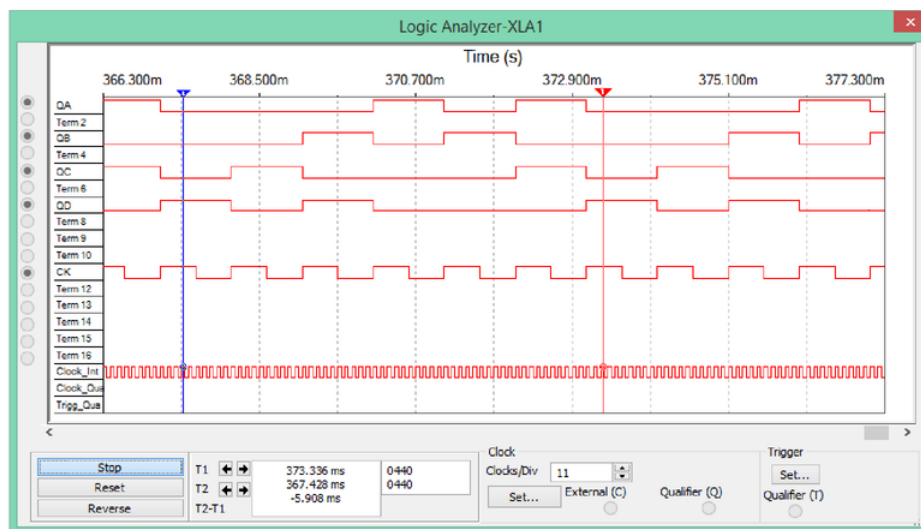


Figure 3.4 - CAR 90 multsim simulation results

3.2 4-bit CAR 150

Since maximal length was not achieved using the 4-bit CAR 90 arrangement, a 4-bit CAR 150 was developed and analysed.

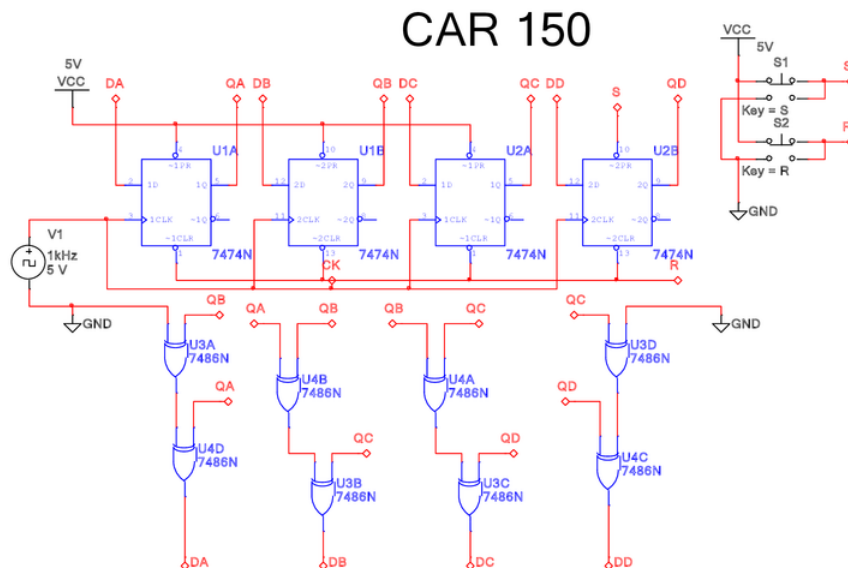
CAR 150 consists of an arrangement of flip-flops where the current state of each flip-flop is determined by taking the XOR of the current state of the flip-flop as well as that of its two immediate neighbours.

Figure 3.5 shows the equation for the CAR 150 rule as previously explained.

$$s_i(t+1) = s_{i+1}(t) \oplus s_i(t) \oplus s_{i-1}(t)$$

Figure 3.5 - CAR 150 rule

Figure 3.6 shows the simulation results of the 4-bit CAR 150 arrangement. As it can be seen below this arrangement did not produce a pseudorandom sequence of maximal length.



Sequence 1;3;4;14;5;13;1

Figure 3.6 - CAR 150 multistage circuit

Figure 3.7 shows the multisim results of the 4-bit CAR 150 arrangement.

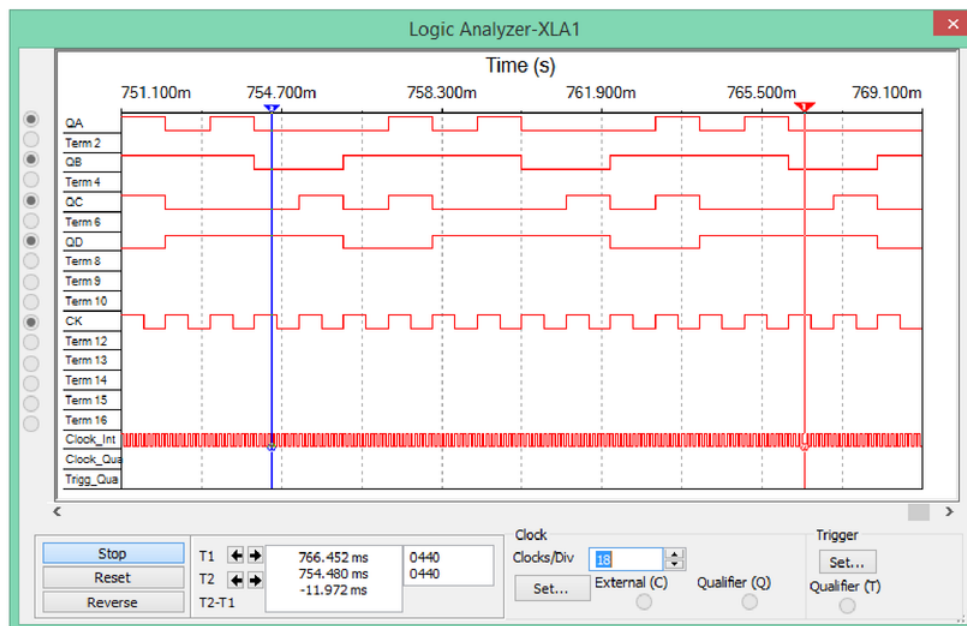


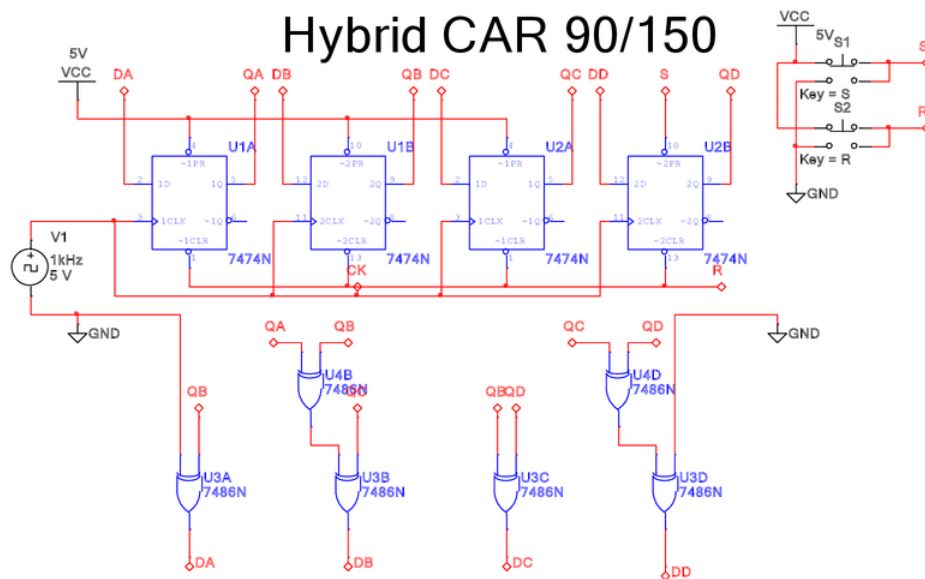
Figure 3.7 - CAR 150 simulation results

3.3 4-bit hybrid CAR 90 / 150

After failing to obtain a pseudorandom sequence of maximal length with both CAR 90 and CAR 150, a 4-bit hybrid CAR 90 / 150 was developed and analysed.

The hybrid arrangement of CAR 90 / 150 consists in combining both rules.

Figure 3.8 shows the 4-bit hybrid CAR 90 / 150 multisim circuit developed. As it can be seen below CAR 90 was used for determining the next state of flip-flops 'A' and 'C', while CAR 150 was used for determining the next state of flip-flops 'B' and 'D'.



Sequence 1;3;6;11;2;5;13;9;7;8;4;14;15;12;10

Figure 3.8 - 4-bit hybrid CAR 90 / 150 multisim circuit

As it can be seen in the sequence shown in Figure 3.8 and in the simulation results shown in Figure 3.9 by combining rule 90 and rule 150 in a 4-bit hybrid CAR maximum length (15 counts) can be achieved.

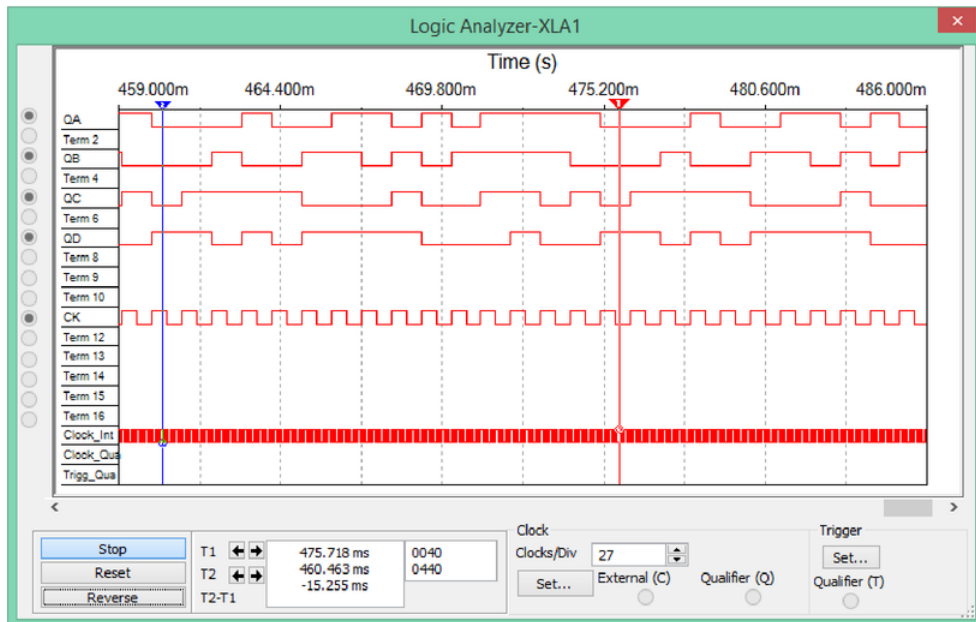


Figure 3.9 - 4-bit hybrid CAR 90/150 simulation results

The prototypes are tied in with theory in this section and the point is clear the max count is only achieved using the hybrid CAR system. Tested here using 4 bit CAR.

3.4 6-bit Hybrid CAR 90/150

In order to obtain a high quality pseudorandom sequence the number of bit of hybrid arrangement was increased to six.

Figure 3.10 shows the hybrid 6-bit hybrid CAR 90 / 150 developed using multisim.

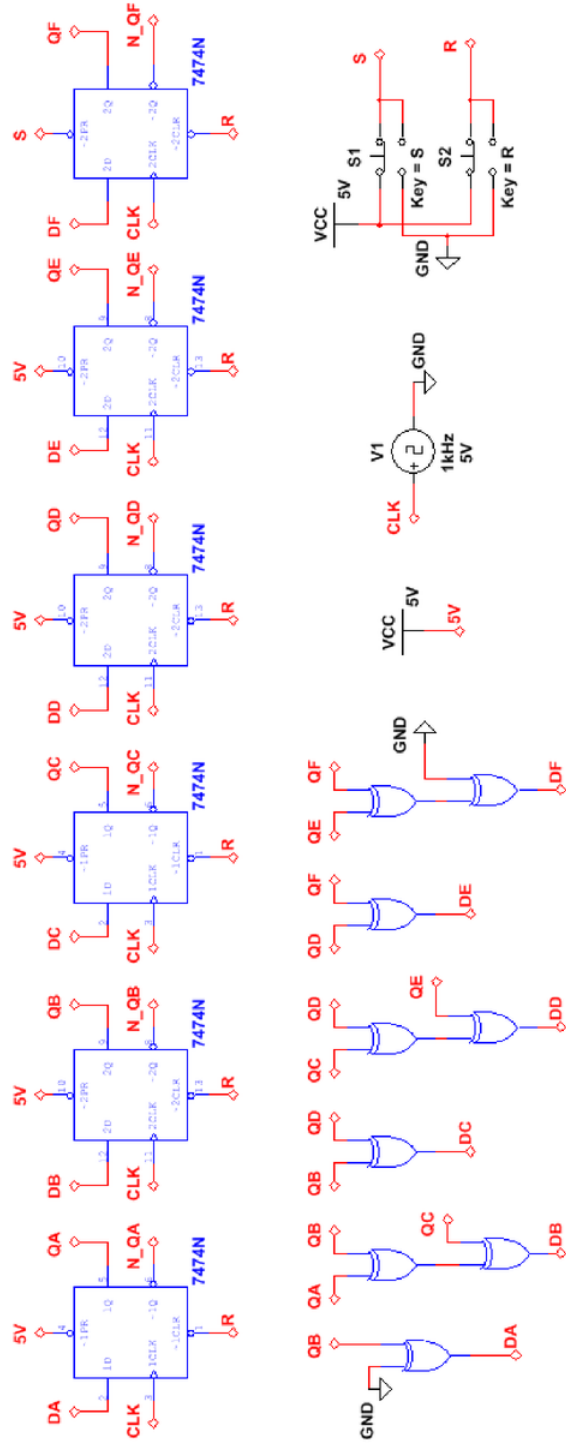


Figure 3.10 - 6-bit hybrid CAR 90 / 150 multisim circuit

Figure 3.11 shows the multisim simulation results for the 6-bit hybrid CAR 90 / 150.

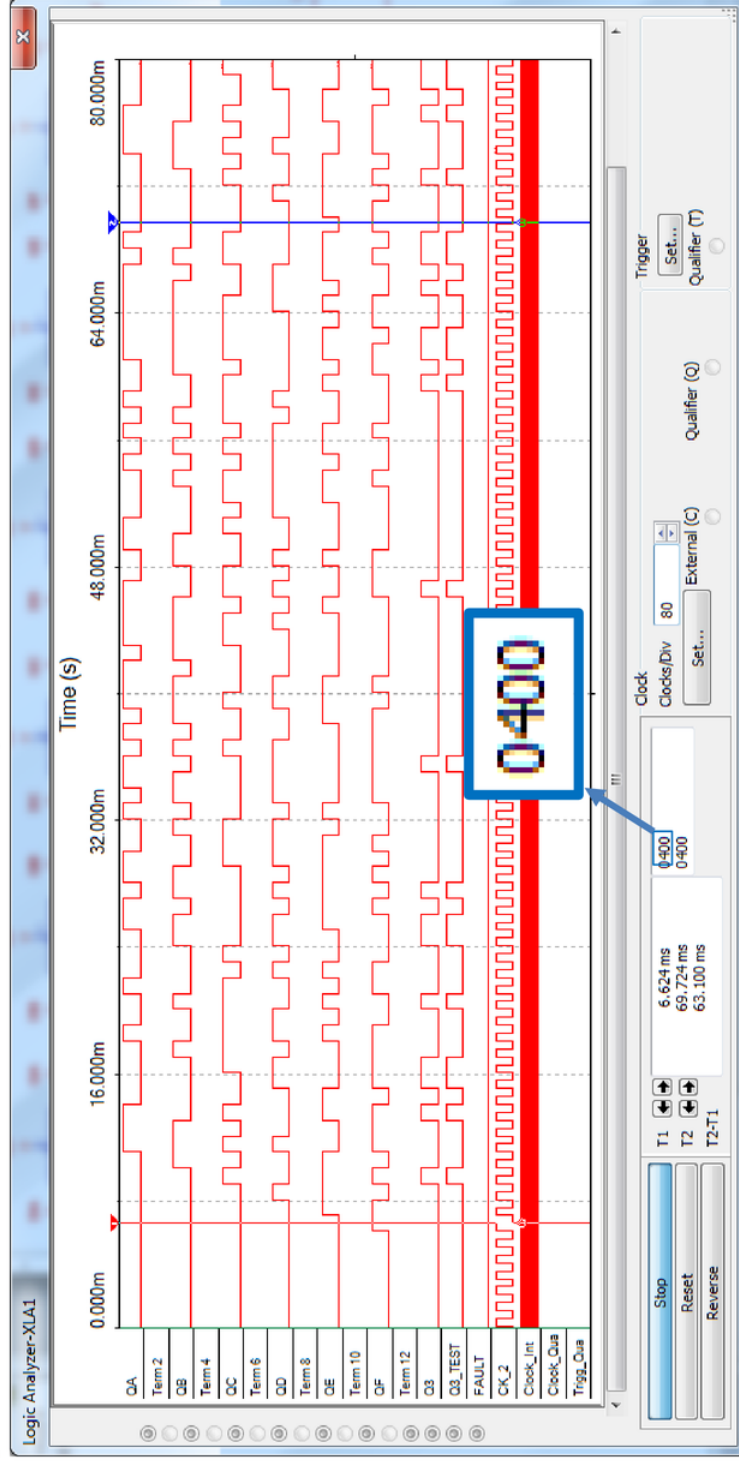


Figure 3.11 - 6-bit hybrid CAR 90 / 150 simulation results

In order to determine the pseudorandom sequence produce by the CAR system the cursor of the logic analyser was moved and the three less significant bits of the logic analysed, as highlighted above, were used to determine the count.

6 bit
CAR
shows
max

Figure 3.12 shows the pseudorandom sequence obtained from the 6-bit hybrid arrangement from the data recorded using the logic analyser on multisim. As it can be seen below, 'Count' represents the order of the sequence, while 'Result' represent the sequence of numbers in the sequence.

Count	A	B	C	Qf+Qe	Qd+Qc	Qb+Qa	Result
1	4	0	0	1	0	0	1
2	5	0	0	3	0	0	3
3	1	4	0	2	4	0	6
4	5	1	0	3	8	0	11
5	1	0	4	2	0	16	18
6	4	5	5	1	12	48	61
7	4	0	5	1	0	48	49
8	5	1	1	3	8	32	43
9	1	0	0	2	0	0	2
10	4	4	0	1	4	0	5
11	4	5	0	1	12	0	13
12	4	1	4	1	8	16	25
13	5	5	1	3	12	32	47
14	0	5	0	0	12	0	12
15	1	1	4	2	8	16	26
16	4	1	1	1	8	32	41
17	5	4	0	3	4	0	7
18	0	1	0	0	8	0	8
19	0	4	4	0	4	16	20
20	1	4	5	2	4	48	54
21	5	0	1	3	0	32	35
22	1	4	4	2	4	16	22
23	5	0	5	3	0	48	51
24	1	5	1	2	12	32	46
25	5	5	0	3	12	0	15
26	0	5	4	0	12	16	28
27	1	0	1	2	0	32	34
28	4	4	4	1	4	16	21
29	4	4	5	1	4	48	53
30	4	4	1	1	4	32	37
31	4	5	4	1	12	16	29
32	4	0	1	1	0	32	33
33	5	0	4	3	0	16	19
34	1	5	5	2	12	48	62
35	5	4	5	3	4	48	55
36	0	0	1	0	0	32	32
37	0	0	4	0	0	16	16
38	0	1	5	0	8	48	56
39	0	5	5	0	12	48	60
40	1	0	5	2	0	48	50

Count	A	B	C	Qf+Qe	Qd+Qc	Qb+Qa	Result
41	4	5	1	1	12	32	45
42	4	1	0	1	8	0	9
43	5	4	4	3	4	16	23
44	0	0	5	0	0	48	48
45	0	1	1	0	8	32	40
46	0	4	0	0	4	0	4
47	1	5	0	2	12	0	14
48	5	5	4	3	12	16	31
49	0	4	1	0	4	32	36
50	1	5	4	2	12	16	30
51	5	4	1	3	4	32	39
52	0	1	4	0	8	16	24
53	0	5	1	0	12	32	44
54	1	1	0	2	8	0	10
55	4	0	4	1	0	16	17
56	5	1	5	3	8	48	59
57	1	1	5	2	8	48	58
58	4	1	5	1	8	48	57
59	5	5	5	3	12	48	63
60	0	4	5	0	4	48	52
61	1	4	1	2	4	32	38
62	5	1	4	3	8	16	27
63	1	1	1	2	8	32	42
64	4	0	0	1	0	0	1

LEGEND

A = 1 --> NotQf.Qe --> 0 + 2 = 2
A = 4 --> Qf.NotQe --> 1 + 0 = 1
A = 5 --> Qf.Qe --> 1 + 2 = 3
B = 1 --> NotQd.Qc --> 0 + 8 = 8
B = 4 --> Qf.NotQc --> 4 + 0 = 4
B = 5 --> Qd.Qc --> 4 + 8 = 12
C = 1 --> NotQb.Qa --> 0 + 32 = 32
C = 4 --> Qb.NotQa --> 16 + 0 = 16
C = 5 --> Qb.Qa --> 16 + 32 = 48

Result = 6 bit count

Figure 3.12 - 6-bit hybrid CAR 90/150 sequence

As it can be seen from the above table, the arrangement produced the required results of a maximal count with no repeats. This was represented by the pseudorandom 63 counts without any numbers being repeated.

4 ANALYSIS OF NI MULTISIM RESULTS

4.1 Test circuit

The pseudorandom generator was then used for testing external circuits and detecting faults. A Boolean expression was developed to represent the manufactured circuit to be tested. This expression shown below represents the circuits requiring testing other circuits by comparing the results.

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

Figure 4.1 shows the multisim circuit designed that represents the good circuit.

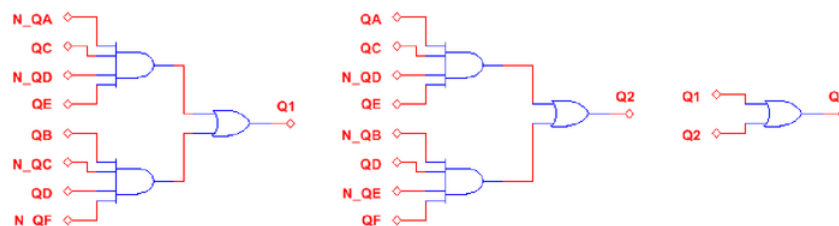


Figure 4.1 - Multisim test circuit

Figure 4.2 shows the multisim simulation results for the test circuit shown above. As it can be seen below, the 6-bit pseudorandom sequence produces another pseudorandom output at the output of the tested circuit.

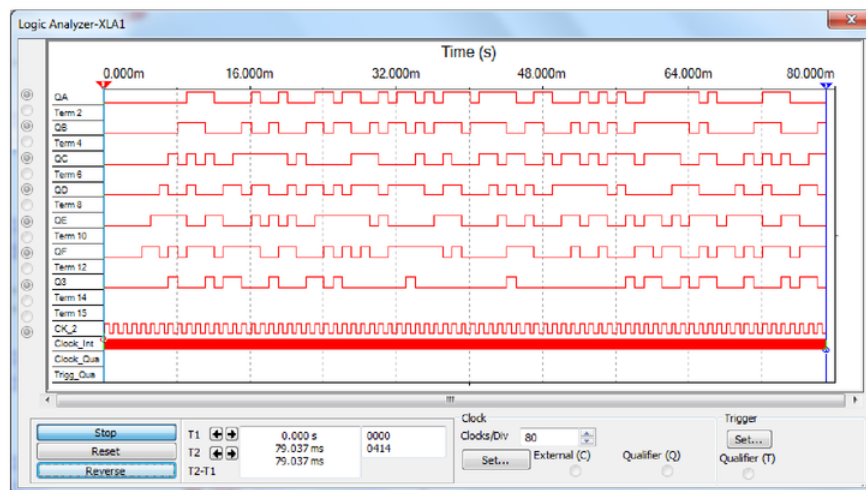


Figure 4.2 - Multisim test circuit simulation results

After the test circuit was tested, three faults were applied to a second circuit.

4.2 Fault 1

In order to simulate the first fault the fourth input of the fourth and gate, representing QF, was connected to ground. Following is the expression representing the fault.

$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.'0' \rightarrow F$ is to ground i.e. 0V

Figure 4.3 shows the multisim circuit designed in order to simulate the first fault.

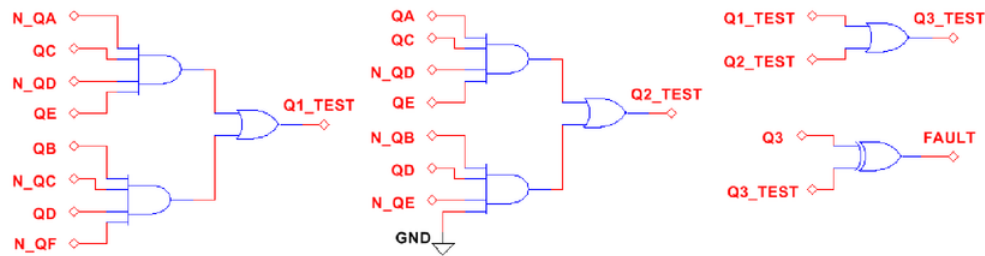


Figure 4.3 – Fault 1 - multisim circuit

In order to detect faults, the outputs of the two tested circuits, Q3 and Q3_TEST, were connected into an X_OR gate so that the output logic would go high when a fault was detected. Figure 4.4 shows the simulations results of the multisim simulation of the circuit with the first fault. As it can be seen below, the output 'FAULT' was high when faults were detected.

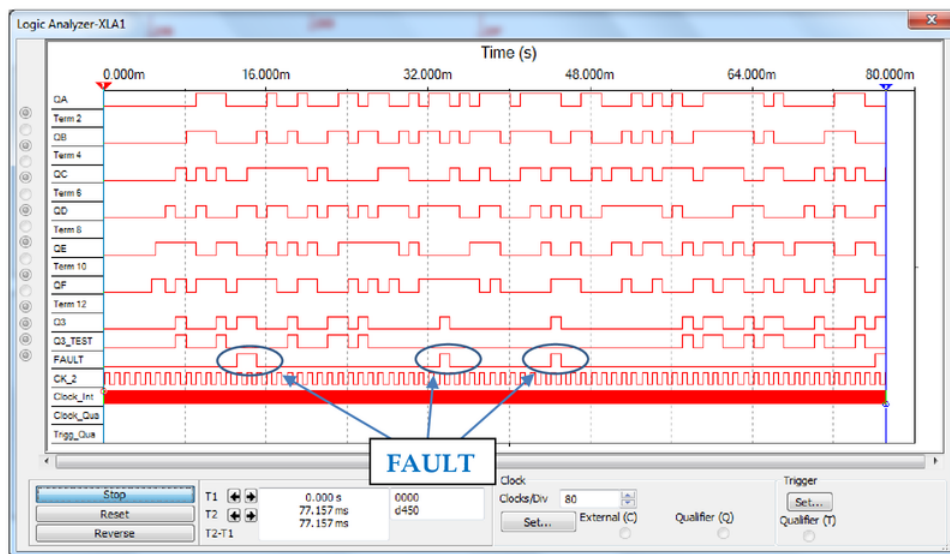


Figure 4.4 - Fault 1 - multisim simulation results

4.3 Fault 2

For the second fault the first input of the third and gate, QA, was connected to ground, Following is the Boolean expression representing the fault.

$$NA.C.ND.E + B.NC.D.NF + '0'.C.ND.E + NB.D.NE.F \rightarrow A \text{ is to ground i.e. } 0V$$

Figure 4.5 shows the multisim circuit designed in order to simulate the second fault.

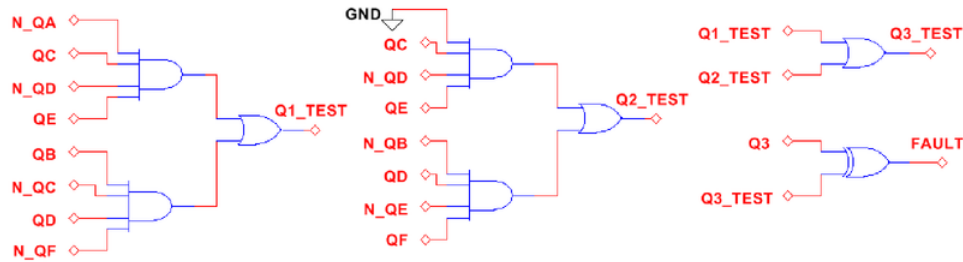


Figure 4.5 - Fault 2 - multisim circuit

Figure 4.6 shows the simulations results of the multisim simulation of the circuit with the second fault. As it can be seen below, the output 'FAULT' was high when faults were detected.

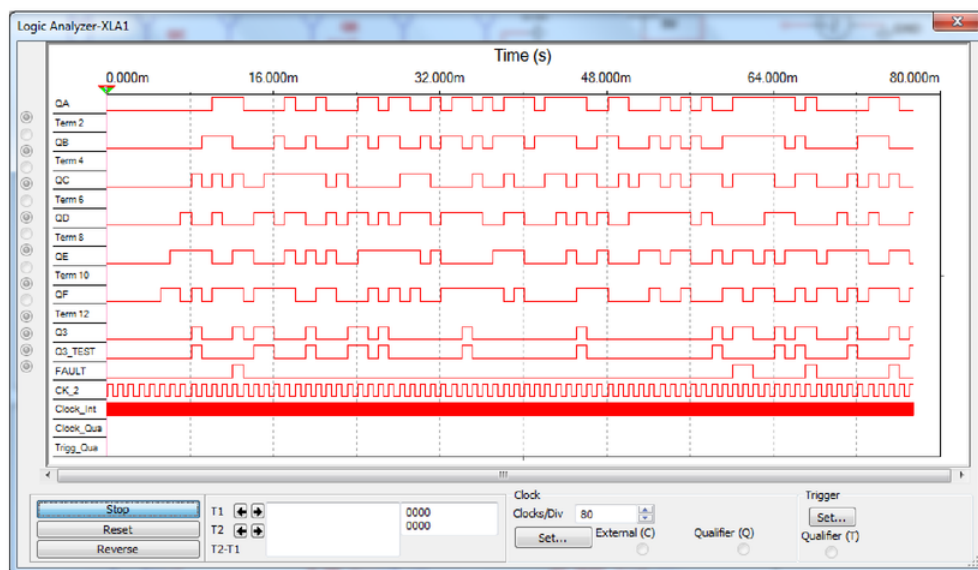


Figure 4.6 - Fault 2 - multisim simulation results

4.4 Fault 3

For the third fault the fourth input of the first and gate, QE, was connected to 5V.

Following is the Boolean expression representing the fault.

$NA.C.ND.'1' + B.NC.D.NF + A.C.ND.E + NB.D.NE.F \rightarrow E$ is to VCC i.e. 5V

Figure 4.7Figure 4.6 shows the multisim circuit designed in order to simulate the third fault.

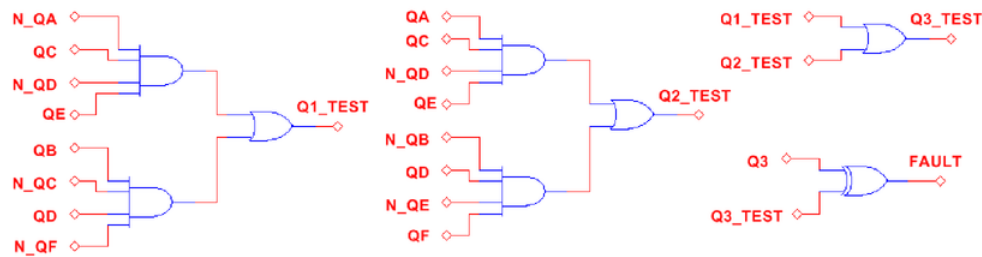


Figure 4.7 - Fault 3 - multisim circuit

Figure 4.8 shows the simulations results of the multisim simulation of the circuit with the third fault. As it can be seen below, the output 'FAULT' was high when faults were detected.

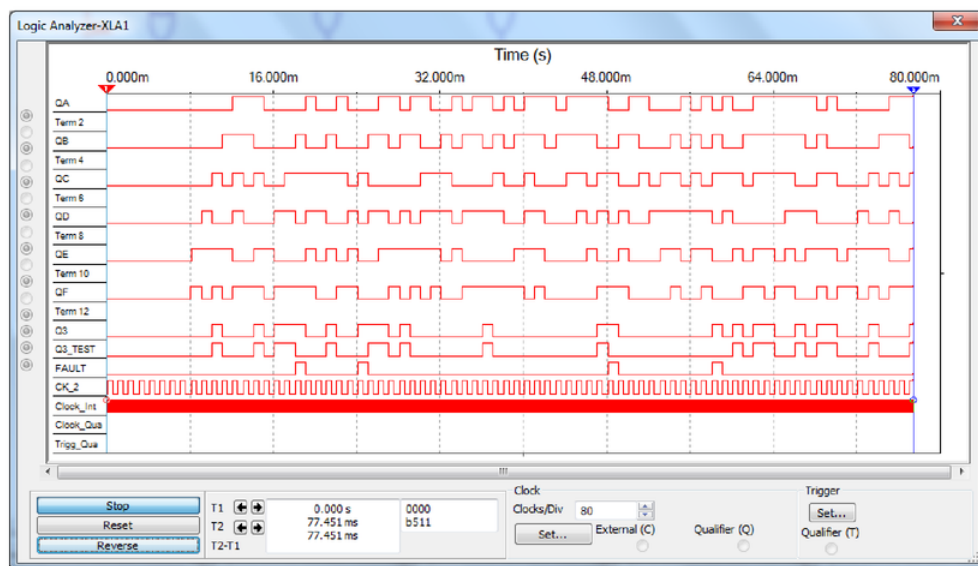


Figure 4.8 - Fault 3 - multisim simulation results

4.5 Summary of Multisim test results

Figure 4.9 shows a summary of the test results obtained from the multisim simulation when the three faults were applied to the circuit.

As it can be seen below, faults were detected during each test, proving that the circuit was functioning as expected.

6-BIT CAR BOOLEAN OUTPUT	GOOD CIRCUIT BOOLEAN EXPRESSION	FAULTY CIRCUIT BOOLEAN EXPRESSION	FAULT DETECTED Y / N
NA.NB.NC.D.NE.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.'0'$	Y
A.B.C.ND.E.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.E + B.NC.D.NF + '0'.C.ND.E + NB.D.NE.F$	Y
NA.NB.C.ND.NE.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.'1' + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	Y

Figure 4.9 - Multisim simulation - summary of test results

Excellent fault analysis with good use of tables.

5 DESIGN OF ASSIGNMENT REQUIREMENT USING VHDL & IP DESIGN

After the 6-bit hybrid CAR 90 / 150 circuit was developed and analysed using multisim, the development process was repeated using Vivado.

5.1 4-bit CAR 90

The 4-bit CAR 90 system was designed using Vivado and it was simulated using a test bench. Figure 5.1 shows the simulation obtains from the Vivado simulation of the 4-bit CAR 90 system.

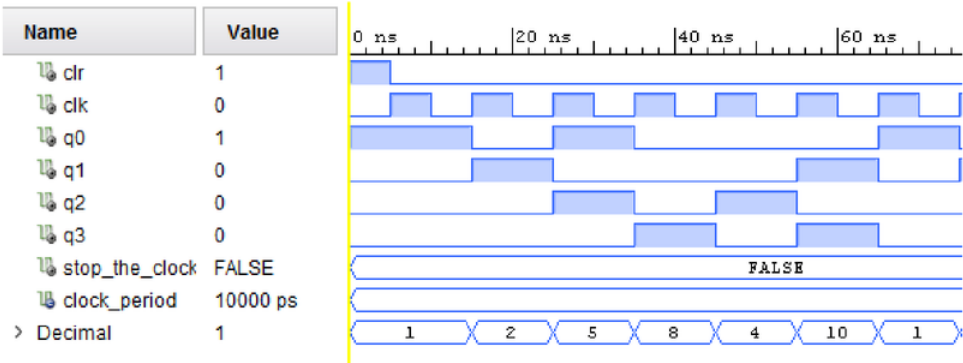


Figure 5.1 – 4-bit CAR 90 Vivado simulation results

As it can be seen above, and as obtained in the multisim simulation, the 4-bit CAR 90, did not allow achieving maximal count.

4 bit rule 90
VHDL CAR
design clearly
shows no max
count.

5.2 4-bit CAR 150

The 4-bit CAR 90 system was designed using Vivado and it was simulated using a test bench. Figure 5.2 shows the simulation obtains from the Vivado simulation of the 4-bit CAR 90 system.

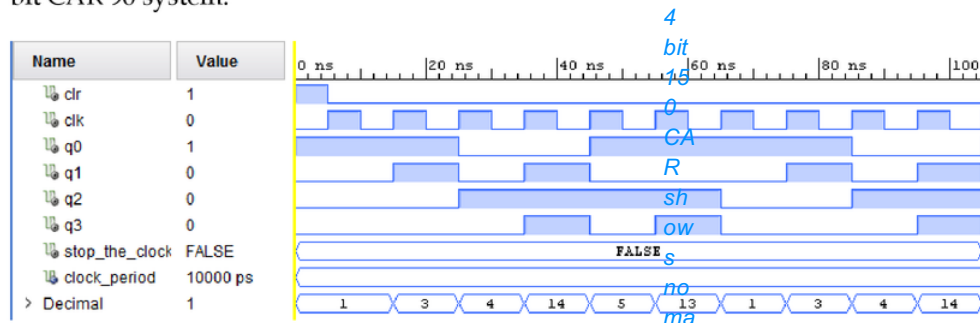


Figure 5.2 - 4-bit CAR 150 Vivado simulation results

As it can be seen above, and as obtained in the multisim simulation, the 4-bit CAR 150 did not produce a pseudorandom sequence with maximal count.

5.3 4-bit hybrid CAR 90 / 150

The 4-bit hybrid CAR 90 / 150 system was also designed using Vivado and it was simulated using a test bench. Figure 5.3 shows the simulation obtains from the Vivado simulation of the 4-bit CAR 90 / 150 system.

As it can be seen below this arrangement produced a pseudorandom sequence with maximal count (1 to 15) as expected.

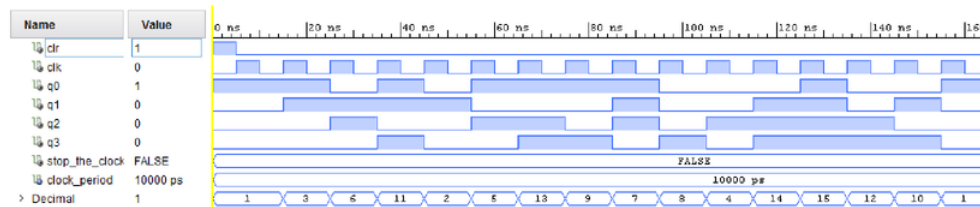


Figure 5.3 - 4-bit hybrid CAR 90/150 Vivado simulation results

Hybrid CAR here shows the max count has been achieved as proven earlier in the CAD design stage.

5.4 6-bit hybrid CAR 90 / 150

The first step towards the development of the 6-bit hybrid CAR 90 / 150 using the hardware descriptive language involved writing the code the system. The code was then simulated using a test bench.

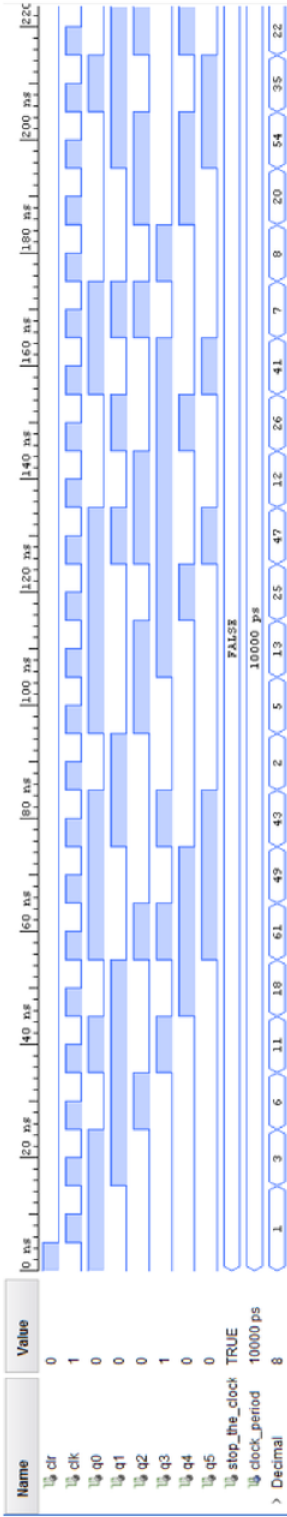


Figure 5.4 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 1)

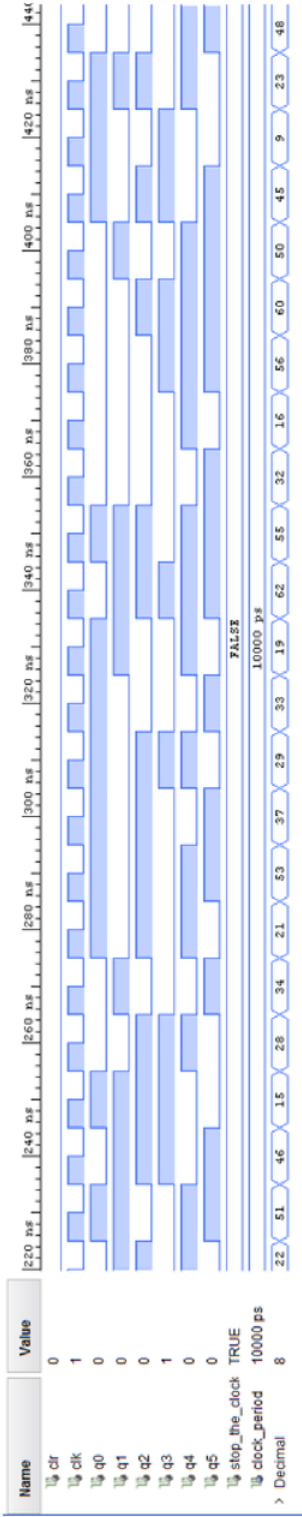


Figure 5.5 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 2)

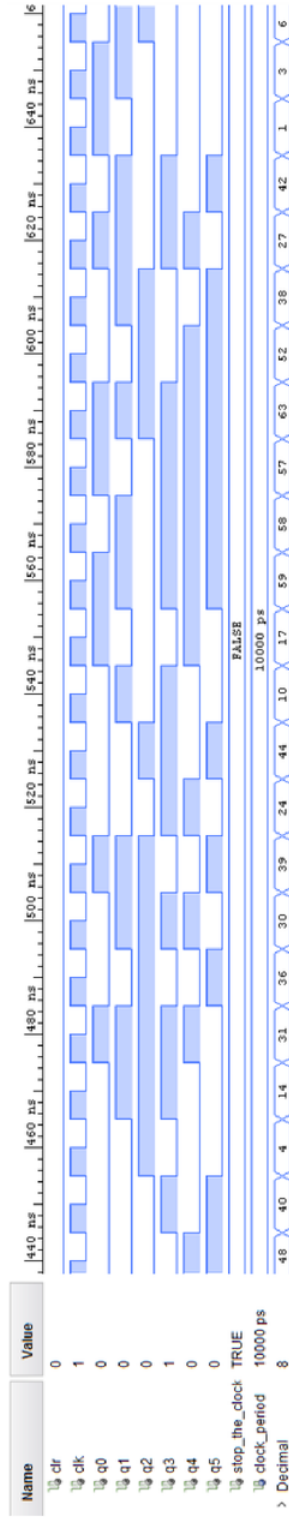


Figure 5.6 - 6-bit hybrid CAR 90 / 150 Vivado simulation results (part 3)

As it can be seen in the previous three images showing the simulation results of the 6-bit hybrid arrangement, the pseudorandom sequence obtained was the same as that obtained during the multisim simulation of the same system.

Following the simulation, the 6-bit hybrid system was designed in Vivado using the IP Integrator tool. This involved adding IP blocks as shown in the design flow in Figure 5.7. As it can be seen below, the X_OR gate was used for comparing the output of the good circuit and that of the circuit to be tested in order to detect a fault when the two outputs were different.

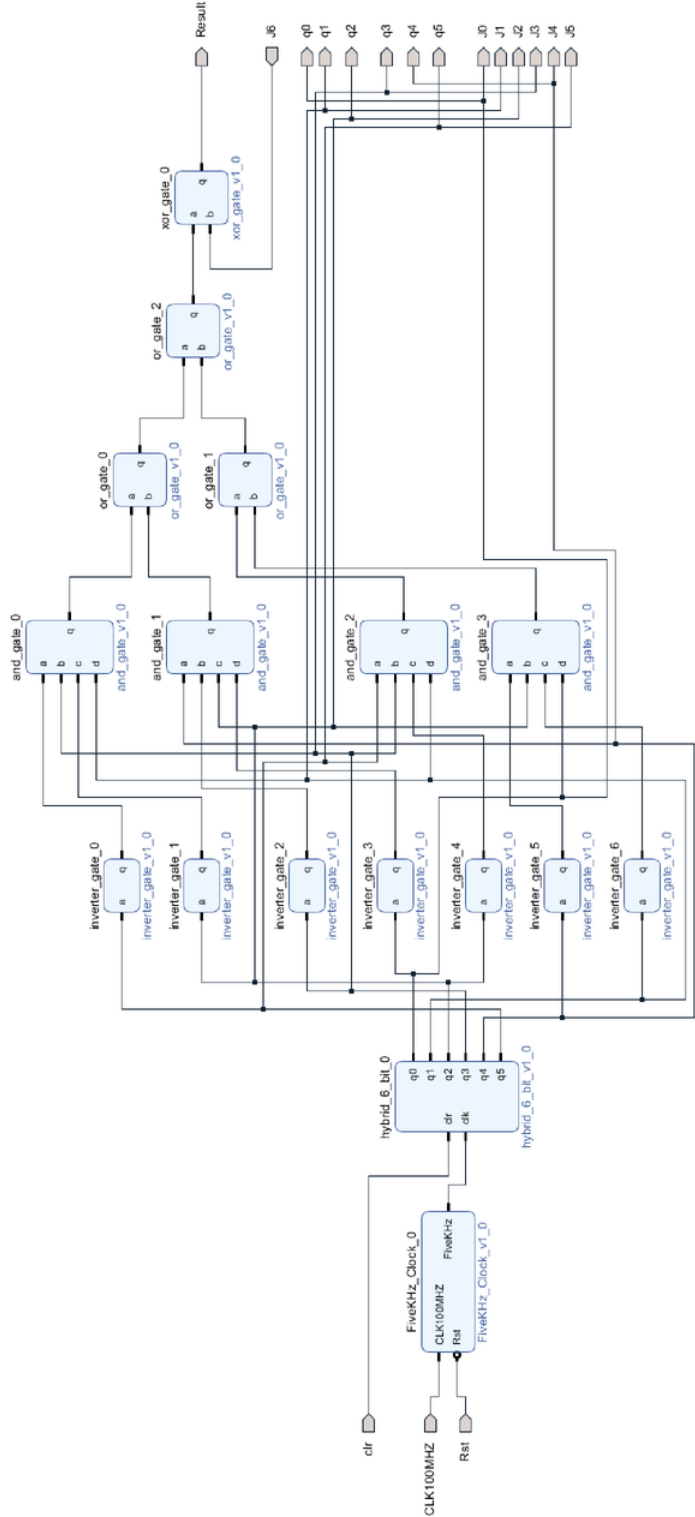


Figure 5.7 – IP integrator design

6 bit
CAR
design
is

The code used for creating the 6-bit hybrid system is shown in Figure 5.8.

As it can be seen below, D registers were used to control the behaviour of the system.

Each register corresponds to one bit.

The equations for 'D(1)', 'D(3)' and 'D(5)' were based on the CAR 90 arrangement, which means that the current state is determined by taking the XOR of its two immediate neighbours.

The equations for 'D(0)', 'D(2)' and 'D(4)', which represent the other three bits of the 6-bit count used the CAR 150 arrangement where the current state of each flip-flop is determined by taking the XOR of the current state as well as that of its two immediate neighbours.

```
D(5) <= '0' xor qs(4);  
D(4) <= qs(5) xor qs(4) xor qs(3);  
D(3) <= qs(4) xor qs(2);  
D(2) <= qs(3) xor qs(2) xor qs(1);  
D(1) <= qs(2) xor qs(0);  
D(0) <= qs(1) xor qs(0) xor '0';
```

*6 bit CAR is identified in the
VHDL [here](#).*

Figure 5.8 - CAR 90 / 150 hybrid equation

The IP entity and the rest of the code for the 6-bit hybrid system is shown in section 0 of the appendix.

All other IP entities used in the 6-bit hybrid system, and their code, is shown in the appendix section of the report.

6 ANALYSIS OF RESULTS USING HARDWARE DESCRIPTIVE LANGUAGE FPGA TECHNOLOGY

Following the design of the IP blocks using Vivado, the program was transferred onto the FPGA development board NEXYS 4 DDR.

The first test carried out using the development board involved transferring the IP block design without connecting the 'J6' pin, which should be the output of the tested circuit. This test allowed ensuring that the fault that the designed system was able to detect faults via the LED.

In Figure 6.1 is shown the FPGA development board when the fault LED was not on.

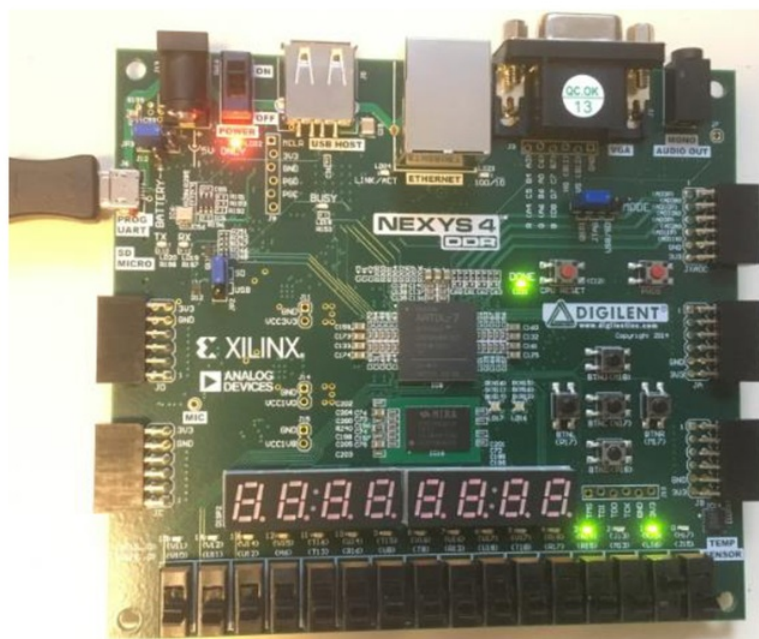


Figure 6.1 - Good circuit test – no fault

6.1 Fault 1

In Figure 6.2 is shown the FPGA board when a fault was detected. This meant that the output from the board, connected to the X_OR gate was at logic high, and therefore different from the other output that was not connected and therefore at logic '0'.

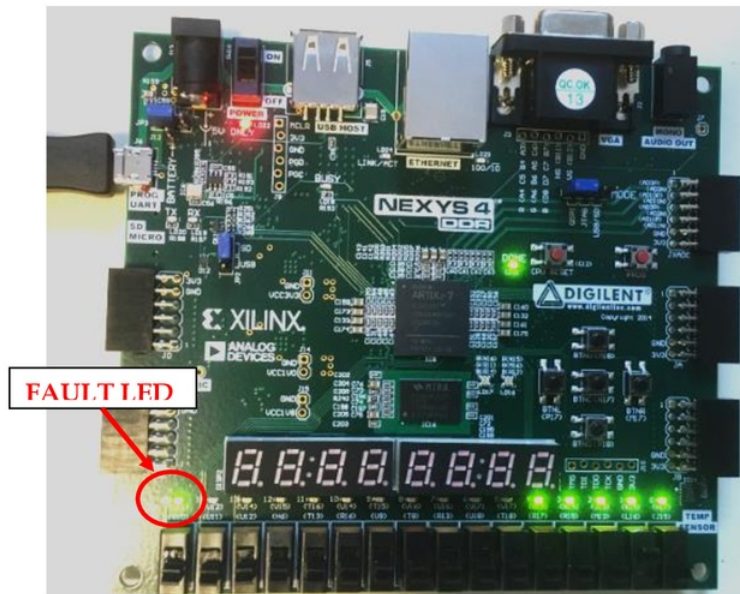


Figure 6.2 - Good circuit test - fault detected

After proving that the developed circuit was able to detect faults, a circuit was built on breadboard with the same logic as that designed using Vivado.

Figure 6.3 shows the prototype circuit tested against the good circuit.

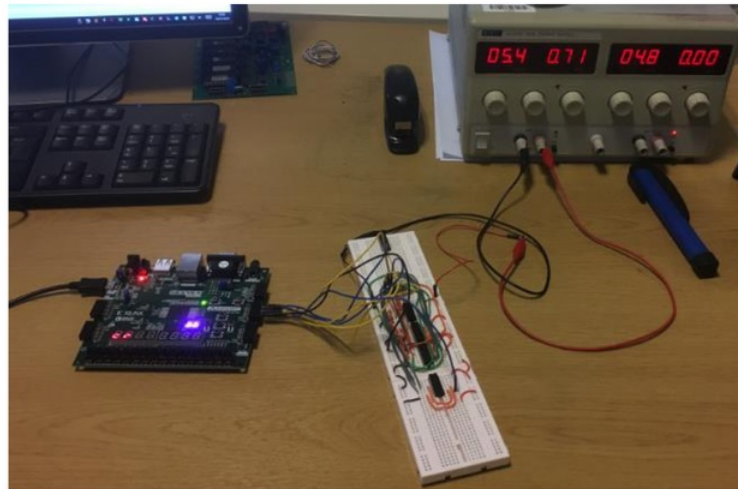


Figure 6.3 - Testing of prototype circuit

Once the prototype circuit was connected to the FPGA board, a test was carried out to ensure that no faults were present in the prototype circuit.

As it can be seen in Figure 6.4 no faults occurred, which confirmed that the prototype circuit was functioning as expected.

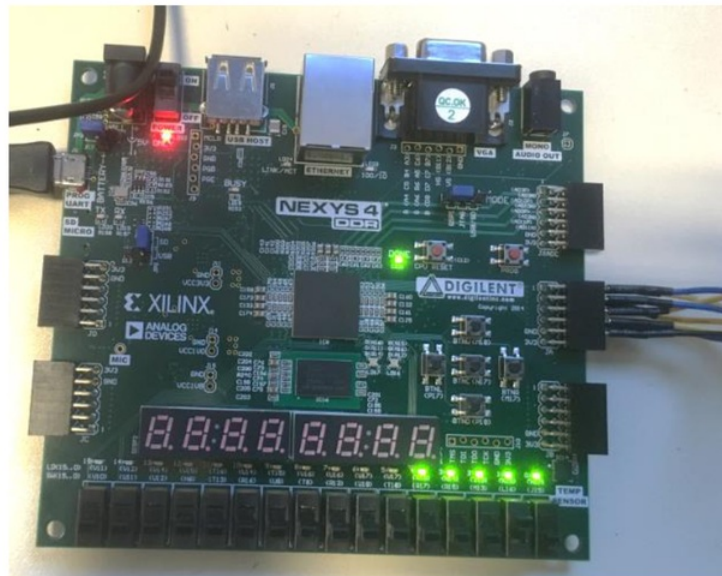


Figure 6.4 - Testing of prototype circuit - no fault

In order to simulate the first fault the fourth input of the fourth and gate on the prototype board, representing QF, was connected to ground.

Figure 6.5 shows the fault detected on the FPGA board.

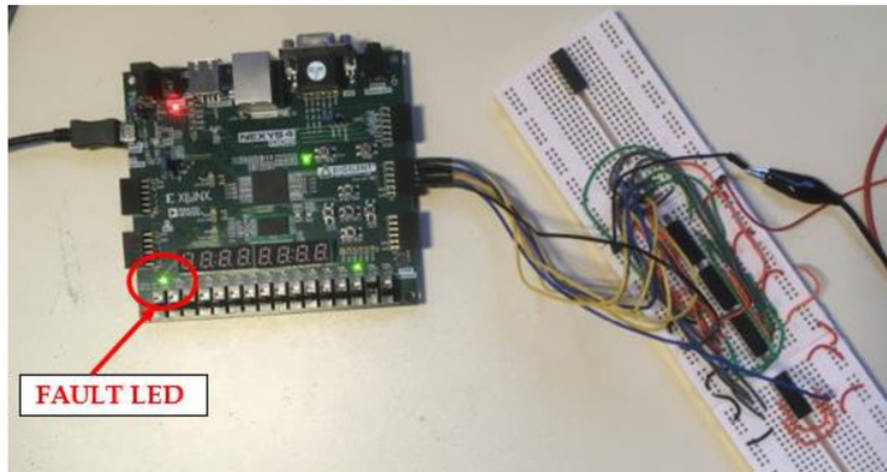


Figure 6.5 - Prototype testing - fault 1

6.2 Fault 2

For the second fault the first input of the third and gate, QA, was connected to ground.

Figure 6.6 shows the fault detected on the FPGA board.

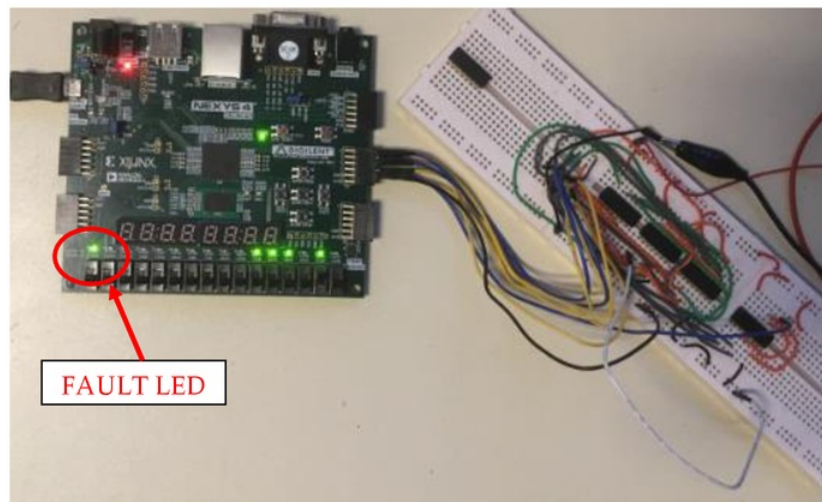


Figure 6.6 - Prototype testing - fault 2

6.3 Fault 3

For the third fault the fourth input of the first and gate, QE, was connected to 5V.

Figure 6.7 shows, once again, the fault detected on the FPGA board.

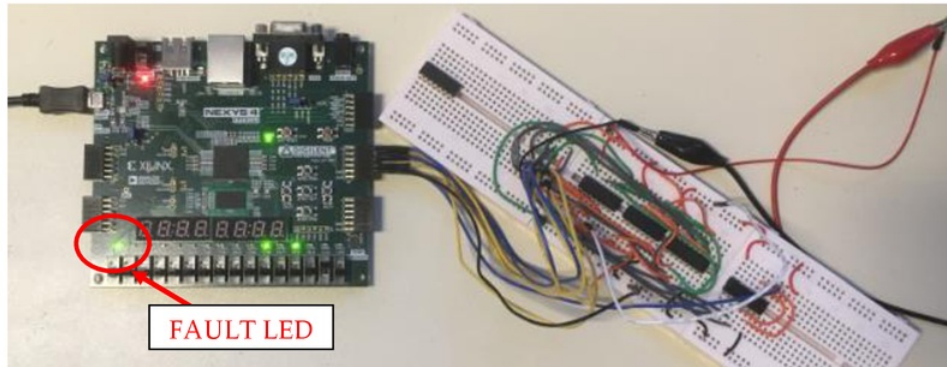


Figure 6.7 - Prototype testing - fault 3

6.4 Summary of FPGA test results

Figure 6.8 shows a summary of the test results obtained from the tests carried out on the NEXYS DDR4 development board.

As it can be seen below, faults were detected during each test, proving that the circuit was functioning as expected.

6-BIT CAR BOOLEAN OUTPUT	GOOD CIRCUIT BOOLEAN EXPRESSION	FAULTY CIRCUIT BOOLEAN EXPRESSION	FAULT DETECTED Y / N
NA.NB.NC.D.NE.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.'0'$	Y
A.B.C.ND.E.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.E + B.NC.D.NF + '0'.C.ND.E + NB.D.NE.F$	Y
NA.NB.C.ND.NE.F	$NA.C.ND.E + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	$NA.C.ND.'1' + B.NC.D.NF + A.C.ND.E + NB.D.NE.F$	Y

Figure 6.8 - FPGA testing - summary of results

Practical CUT tests are explained with good images.

7 PCB DESIGN

The PCB circuit was designed using the EAGLE software.

This involved designing the logic circuit shown in Figure 7.1.

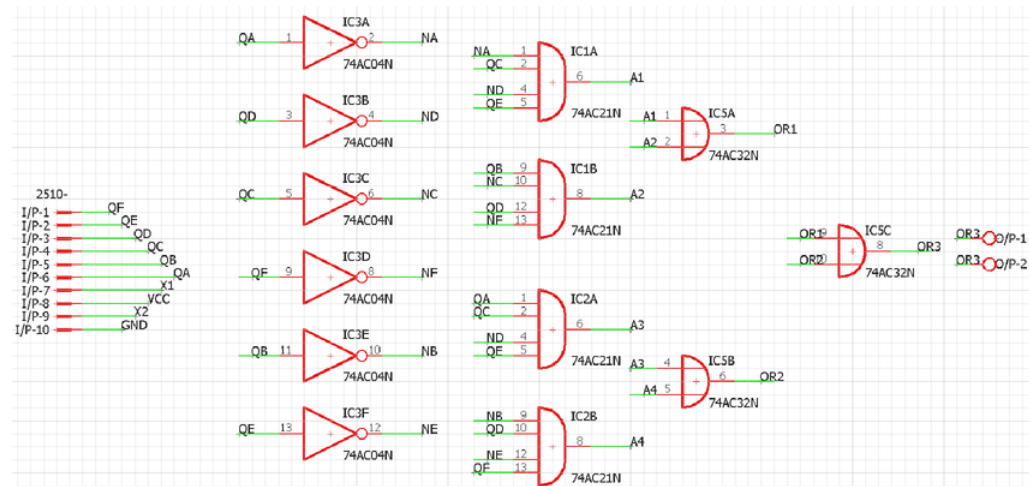


Figure 7.1 - EAGLE PCB logic circuit

From this circuit the PCB board design was created. The PCB created was a double layer as required.

Figure 7.2 shows the top layer.

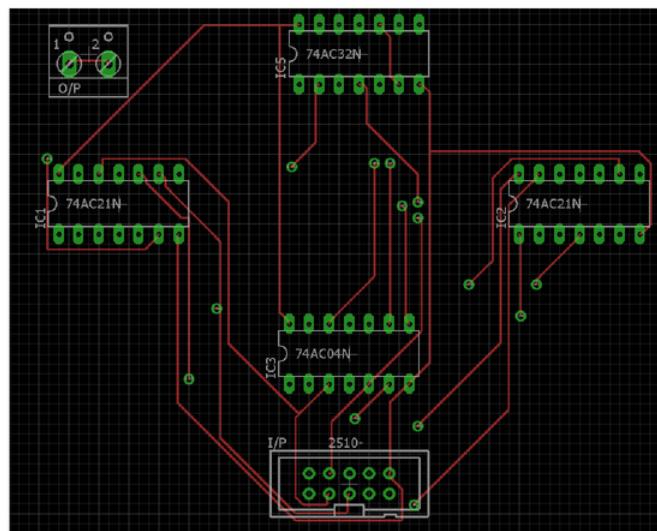


Figure 7.2 - PCB top layer

Figure 7.3 shows the bottom layer.

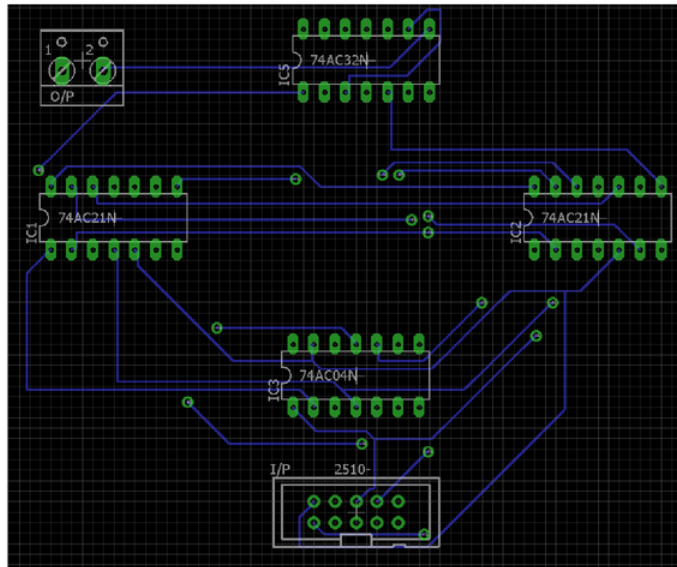


Figure 7.3 - PCB bottom layer

Figure 7.4 shows the top and bottom layer, the vias that allow connecting the two layers and the components to be fitted onto the PCB.

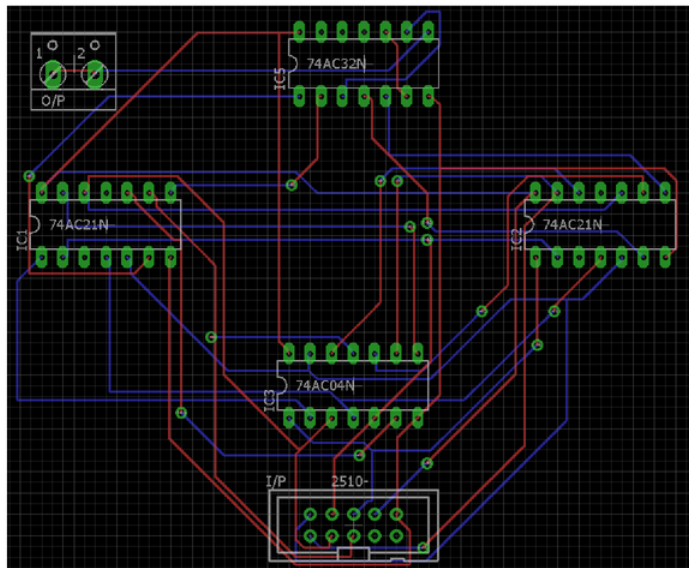


Figure 7.4 - Top and bottom layer

Following the design of the PCB using the EAGLE software, the files created in order to be able to manufacture the PCB were load onto Gerber viewer, which allowed

checking that all information required for the manufacture of the PCB was present in the project folder.

Figure 7.5 shows the online Gerber viewer screen. As it can be seen below, in red and blue, are shown the copper tracks on both top and bottom layer. In pink is shown the copper around holes to be drilled, which are shown in white.

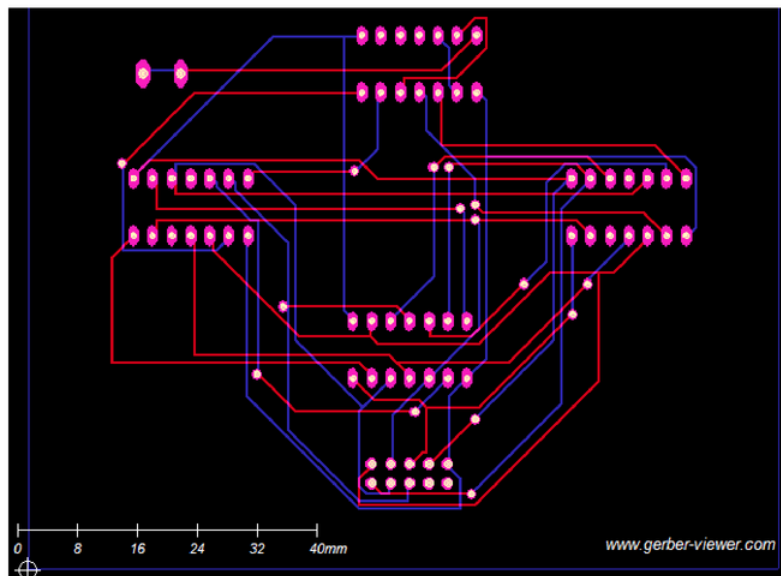


Figure 7.5 - Gerber viewer

*PCB design and
manufacture process is
explained well.*

8 PRACTICAL PCB BUILD AND TESTING OF PROTOTYPE ON PROTOTYPING BOARD

The PCB was manufactured using the files created from the Eagle software.

Figure 8.1 and Figure 8.2 shows the top and bottom layers of the manufactured PCB.

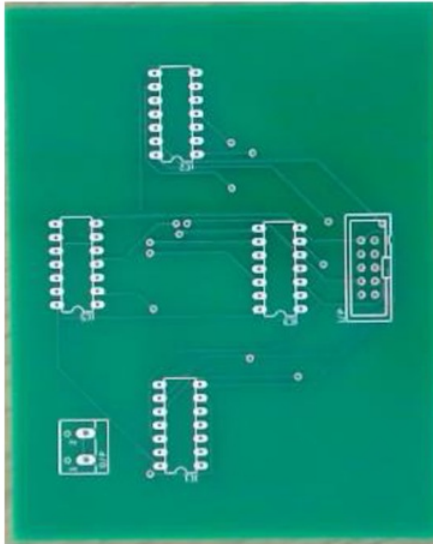


Figure 8.1 - Top layer

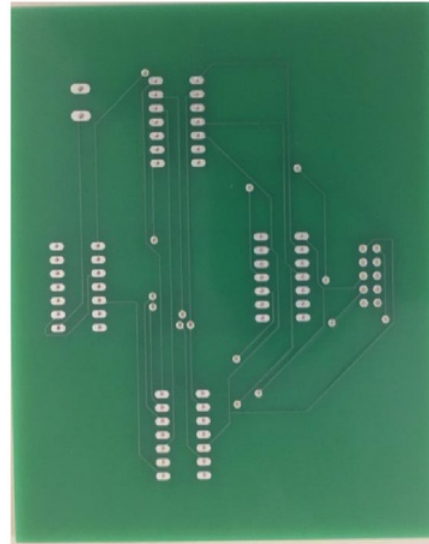


Figure 8.2 - Bottom layer

Following the manufacture of the two layer PCB, two chips with AND gates, one with OR gates and another one with XOR gates were soldered in the appropriate slots. A two pin terminal block was also soldered in order to be able to power the logic gates as well as the pins for the 6-bit pseudorandom sequence.

Figure 8.3 shows the testing board and the manufacture PCB prior to initiating the testing.

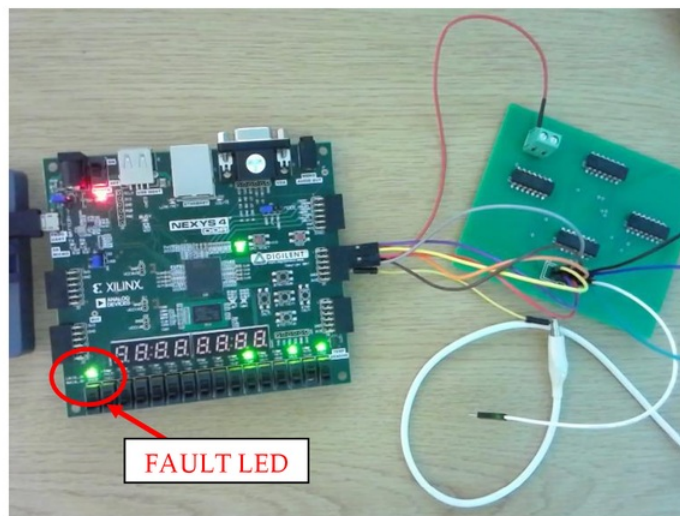


Figure 8.3 - PCB testing - fault detected

Figure 8.4 shows the testing the manufactured PCB. As it can be seen below, the fault LED did not come on, meaning that the manufactured PCB was functioning as expected.

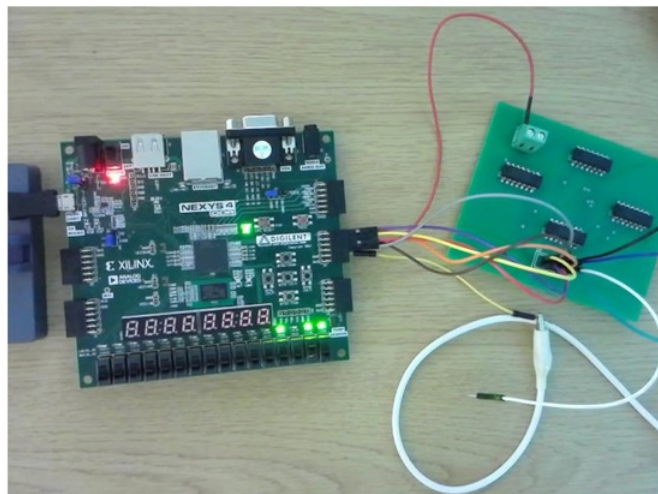


Figure 8.4 - PCB testing - no fault detected

9 CONCLUSION

The aim and objectives of the assignment stated in the introduction were fully met as explained further below.

Research was carried out into PCB technology and, design, manufacturing and soldering processes. Inspection and automatic testing processes were also investigated, in particular, the use of AOI and AXI in the PCB production process for ATE.

Following the research mentioned above, a set of pseudorandom pattern generators were designed and analysed using a CAD / CAM system, Multisim, and hardware descriptive language system, Vivado.

The hardware descriptive language produced using Vivado was then extended in order to produce a testing circuit able to test a prototype circuit and obtain pass / fail results.

A PCB was then manufacture and tested using the testing circuit mentioned above.

*The report
has a good
structure and
is very well
presented.*

*The quality of
the
explanations
is excellent.*

10 REFERENCE LIST

Advameg Inc., 2014. *Printed Circuit Board*. [Online]

Available at: <<http://www.madehow.com/Volume-2/Printed-Circuit-Board.html>>

[Accessed 28th November 2017].

Anon., 2013. *Software to design a 2-sided PCB with discrete components*. [Online]

Available at: <<https://electronics.stackexchange.com/questions/64008/software-to-design-a-2-sided-pcb-with-discrete-components>>

[Accessed 7th December 2017].

Blundell Production Equipment, n.d. *7-Inspex-Omni Digital Inspection & Measurement System*. [Online]

Available at: <<http://blundell.co.uk/product/inspex-omni-digital-inspection-measurement-system/>>

[Accessed 7th December 2017].

electronicsnotes, n.d. *ICT, In Circuit Test Tutorial*. [Online]

Available at: <<https://www.electronics-notes.com/articles/test-methods/automatic-automated-test-ate/ict-in-circuit-test-what-is-primer.php>>

[Accessed 8th December 2017].

electronicnotes, n.d. *Automatic optical inspection, AOI systems*. [Online]

Available at: <<https://www.electronics-notes.com/articles/test-methods/automatic-automated-test-ate/aoi-optical-inspection.php>>

[Accessed 7th December 2017].

electronicsnotes, n.d. *Automated X-Ray Inspection AXI for PCB & BGA*. [Online]

Available at: <<https://www.electronics-notes.com/articles/test-methods/automatic-automated-test-ate/axi-x-ray-inspection.php>>

[Accessed 7th December 2017].

Hewes, J., 2017. *Printed Circuit Boards (PCBs)*. [Online]

Available at: <<https://electronicsclub.info/pcb.htm>>

[Accessed 28th November 2017].

Keen, D., 2017. *What Are Printed Circuit Boards Used for?*. [Online]
Available at: <<https://sciencing.com/printed-circuit-boards-used-5031475.html>>
[Accessed 28th November 2017].

Kush , D., 2014. *High-Speed PCB drilling spindle*. [Online]
Available at: <http://www.buildyouridea.com/cnc/pcb_drill/pcb_drill.html>
[Accessed 7th December 2017].

Linkedin, 2015. *What PCB Design Software Do You Use?*. [Online]
Available at: <<https://www.linkedin.com/pulse/what-pcb-design-software-do-you-use-lisa-liu-jingwen>>
[Accessed 2017 December 2017].

nordsonmatrix, n.d. *Inspection engine*. [Online]
Available at: <<http://www.nordsonmatrix.com/product/MIPS-AXI>>
[Accessed 7th December 2017].

PARPRO Technologies , 2016. *Testing*. [Online]
Available at: <<http://www.calquality.com/services/testing>>
[Accessed 7th December 2017].

Projects Unlimited, 2014. *Reflow profile*. [Online]
Available at: <http://www.puiaudio.com/images/reflow_profile.jpg>
[Accessed 7th December 2017].

Projects Unlimited, 2014. *Wave solder profile*. [Online]
Available at: <<http://www.puiaudio.com/resources-faqs.aspx#q1-2>>
[Accessed 7th December 2017].

robotroom, 2017. *PCBs with SMT Components*. [Online]
Available at: <<http://www.robotroom.com/PCB3.html>>
[Accessed 7th December 2017].

SAE Circuits, 2015. *M/L Up to 24 Layers*. [Online]
Available at: <<http://www.saecircuits.com/pcb-capabilities/ml-24-layers/>>
[Accessed 7th December 2017].

Sparkfun Electronics, n.d. *How to Solder: Through-Hole Soldering*. [Online]

Available at: <<https://learn.sparkfun.com/tutorials/how-to-solder-through-hole-soldering>>

[Accessed 7th December 2017].

Streamline Electronics Manufacturing, 2017. [Online]

Available at: <<http://sem-inc.com/>>

[Accessed 6th December 2017].

Trizo, 2014. *PTH Assembly: An Old Technology with New Uses*. [Online]

Available at: <<http://www.trizo.com/news/17/pth-assembly--an-old-technology-with-new-uses>>

[Accessed 6th December 2017].

Trizo, 2014. *The Advantages of PTH Assembly*. [Online]

Available at: <<http://www.trizo.com/news/21/the-advantages-of-ptth-assembly>>

[Accessed 6th December 2017].

Good reference list used for research.

11 APPENDIX

11.1 Clock

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity clk_hybrid_wrapper is
    port (
        CLK100MHZ : in STD_LOGIC;
        Rst : in STD_LOGIC;
        clr : in STD_LOGIC;
        q0 : out STD_LOGIC;
        q1 : out STD_LOGIC;
        q2 : out STD_LOGIC;
        q3 : out STD_LOGIC );
end clk_hybrid_wrapper;
architecture STRUCTURE of clk_hybrid_wrapper is
    component clk_hybrid is
        port (
            q0 : out STD_LOGIC;
            q1 : out STD_LOGIC;
            q2 : out STD_LOGIC;
            q3 : out STD_LOGIC;
            CLK100MHZ : in STD_LOGIC;
            Rst : in STD_LOGIC;
            clr : in STD_LOGIC );
    end component clk_hybrid;
begin
    clk_hybrid_i: component clk_hybrid
        port map (
            CLK100MHZ => CLK100MHZ,
            Rst => Rst,
            clr => clr,
            q0 => q0,
            q1 => q1,
            q2 => q2,
            q3 => q3
        );
end STRUCTURE;
```



11.2 6-bit hybrid

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

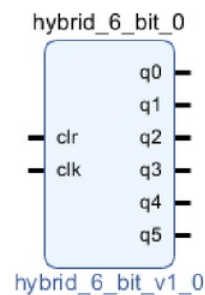
entity hybrid_6_bit is
    port(
        clr : in STD_LOGIC;
        clk : in STD_LOGIC;
        q0,q1,q2,q3,q4,q5 : out std_logic
    );
end hybrid_6_bit;

architecture Behavioral of hybrid_6_bit is
    signal D, qs: STD_LOGIC_VECTOR(5 downto 0);
begin

    D(5) <= '0' xor qs(4);
    D(4) <= qs(5) xor qs(4) xor qs(3);
    D(3) <= qs(4) xor qs(2);
    D(2) <= qs(3) xor qs(2) xor qs(1);
    D(1) <= qs(2) xor qs(0);
    D(0) <= qs(1) xor qs(0) xor '0';

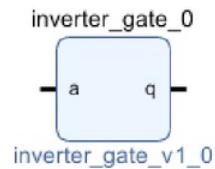
    process(clk, clr)
    begin
        if clr = '1' then
            qs <= "000001";
        elsif clk'event and clk = '1' then
            qs <= D;
        end if;
    end process;

    q0 <= qs(0);
    q1 <= qs(1);
    q2 <= qs(2);
    q3 <= qs(3);
    q4 <= qs(4);
    q5 <= qs(5);
end Behavioral;
```



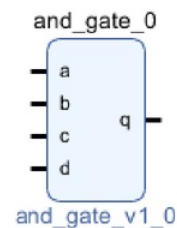
11.3 Inverter gate

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity inverter_gate is
5  Port ( a : in std_logic;
6        q : out std_logic);
7  end inverter_gate;
8  architecture Behavioral of inverter_gate is
9
10 begin
11   q <= not a;
12
13 end Behavioral;
```



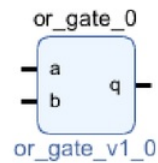
11.4 And gate

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity and_gate is
5  -- Port ( );
6  Port ( a : in std_logic;
7        b : in std_logic;
8        c : in std_logic;
9        d : in std_logic;
10       q : out std_logic);
11 end and_gate;
12
13 architecture Behavioral of and_gate is
14
15 begin
16   q <= a and b and c and d;
17
18 end Behavioral;
```



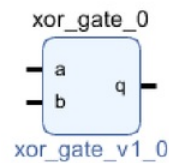
11.5 Or gate

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity or_gate is
5
6  Port ( a : in std_logic;
7         b : in std_logic;
8         q : out std_logic);
9  end or_gate;
10 architecture Behavioral of or_gate is
11
12  begin
13    q <= a or b;
14
15  end Behavioral;
```



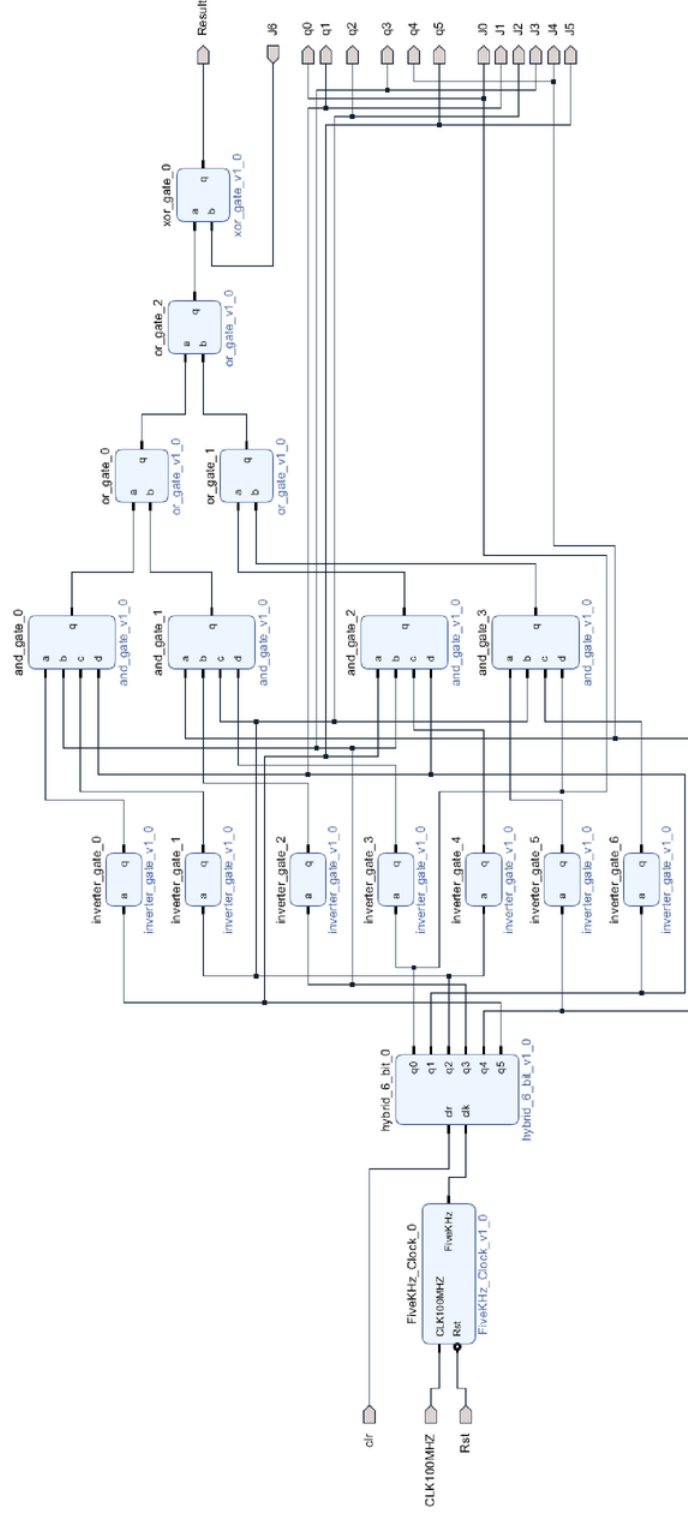
11.6 X-or gate

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity xor_gate is
5  -- Port ( );
6  Port ( a : in std_logic;
7         b : in std_logic;
8         q : out std_logic);
9  end xor_gate;
10
11 architecture Behavioral of xor_gate is
12
13  begin
14    q <= a xor b;
15
16  end Behavioral;
```



11.7 6-bit hybrid CAR 90/150

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3
4  entity hybrid_6_bit is
5      port(
6          clr : in STD_LOGIC;
7          clk : in STD_LOGIC;
8          q0,q1,q2,q3,q4,q5 : out std_logic
9      );
10 end hybrid_6_bit;
11
12 architecture Behavioral of hybrid_6_bit is
13     signal D, qs: STD_LOGIC_VECTOR(5 downto 0);
14     begin
15         D(5) <= '0' xor qs(4);
16         D(4) <= qs(5) xor qs(4) xor qs(3);
17         D(3) <= qs(4) xor qs(2);
18         D(2) <= qs(3) xor qs(2) xor qs(1);
19         D(1) <= qs(2) xor qs(0);
20         D(0) <= qs(1) xor qs(0) xor '0';
21
22         process(clk, clr)
23             begin
24                 if clr = '1' then
25                     qs <= "000001";
26                 elsif clk'event and clk = '1' then
27                     qs <= D;
28                 end if;
29             end process;
30
31         q0 <= qs(0);
32         q1 <= qs(1);
33         q2 <= qs(2);
34         q3 <= qs(3);
35         q4 <= qs(4);
36         q5 <= qs(5);
37     end Behavioral;
```



Appendix containing the VHDL components. This could have been referenced from the main report.

V.Good report. Everything seems to be covered well.

Report

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

82/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

Text Comment. Intro is a little weak. The focus should be on the virtual ATE integration.

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

Text Comment. Research is v.good and includes a variety of ATE used in production industry.

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

Text Comment. The prototypes are tied in with theory in this section and the point is clear the max count is only achieved using the hybrid CAR system. Tested here using 4 bit CAR.

PAGE 27

PAGE 28

Text Comment. 6 bit CAR shows max count and repeat after count has finished.

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

Text Comment. Excellent fault analysis wit good use of tables.

PAGE 35

Text Comment. 4 bit rule 90 VHDL CAR design clearly shows no max count.

PAGE 36

Text Comment. 4 bit 150 CAR shows no max count.

Text Comment. Hybrid CAR here shows the max count has been achieved as proven earlier in the CAD design stage.

Text Comment. 6 bit CAR design is correct and includes the x-or gate.

Text Comment. 6 bit CAR is identified in the VHDL here.

Text Comment. Practical CUT tests are explained with good images.

Text Comment. PCB design and manufacture process is explained well.

Text Comment. The report has a good structure and is very well presented. The quality of the explanations is excellent.

Text Comment. Good reference list used for research.

Text Comment. V.Good report. Everything seems to be covered well.

Text Comment. Appendix containing the VHDL components. This could have been referenced from the main report.

GRADING FORM: CSG-RUBIC

BRUNO FORKUOH

0

RESEARCH-INFORMED LITERATURE



SMT technology is explained well with appropriate diagrams. 4/5 SMT and PTH comparison is illustrated. 4/5 Automatic testing is explained with optical and ICT with diagrams. 4.5/5 X-RAY testing is covered appropriately. 3.5/5 Research is very good. The only point here is the intro is a little weak. This should have some focus on virtual ATE integration.

16/20

KNOWLEDGE AND UNDERSTANDING



The design of hybrid CAR using CAD and VHDL is very good. ALL stages are explained. Very good demonstration of understanding. 6/7 The PCB design process is explained and includes the

17/20

appropriate diagrams. 5.5/7 Design of CAR using FPGA has gone through the necessary stages/ prototypes with good explanations. 5.5/6

ANALYSIS



Tests results have proved the design and this is explained very well. The comparison to theory is clear and demonstrates a thorough understanding of this process. This is good. 9/10 The analysis of the results of the FPGA and PCB testing is explained with good attention to detail. 8/10

17/20

PRACTICAL APPLICATION AND DEP.



The practical application and planning here is excellent. This shows initiative and innovation because the tests go beyond what was asked for. Excellent in this area. 13.5/15 the FPGA prototyping and testing is excellent. Again good initiative shown here. 8.5/10 Prototyping procedures for development was carried out extremely well.

22/25

SKILLS FOR PROFESSIONAL PRAC.



A very professional look about the use of software in this report. 3/5 Conclusion should have focus on virtual ATE integration techniques. 3/5 There is mention of ATE which helps here. The report as a whole is very good. 4/5

10/15