

PSDE6018 Assignment 1 Dan Lee

by Daniel Lee

FILE	12774_DANIEL_LEE_PSDE6018_ASSIGNMENT_1_DAN_LEE_155126_205 1053620.DOCX (12.19M)		
TIME SUBMITTED	09-FEB-2018 05:27 PM (UTC+0000)	WORD COUNT	6082
SUBMISSION ID	83022766	CHARACTER COUNT	35929



BSc: Electrical and Electronic Engineering

Electronic Production and Test

PSDE6018

Pseudo Random Pattern Generators

Dan Lee

Lecturer: Rob Jenkins



Contents

Table of Figures	3
Table of Tables	6
1 – Introduction	7
2 – Research	8
2.1 – SMT Technology.....	8
2.2 – SMT and PTH Technology	9
2.3 – Automatic Optical Inspection	10
2.4 – Automatic X-Ray Inspection.....	13
2.5 – In Circuit Testing.....	15
4 – Design, Testing and Analysis of NI MultiSim	18
5 - Design, Testing and Analysis of Hardware Descriptive Language FPGA Technology.....	25
5.1 – CAR 150 4-Bit.....	25
5.2 – CAR 90 4-Bit.....	26
5.3 – CAR 90/150 4-Bit Hybrid	27
5.4 – CAR 90/150 6-Bit Hybrid	28
5.5 – Xilinx Programmable Board and Built Breadboard Test Circuit.....	30
6 – PCB Practical Design, Build, Testing and Analysis of Prototype on Prototyping Board.....	36
7 – Conclusions and Recommendations	43
8 – References	44

9 – Appendix	45
Appendices 1 – CAR 90/150 Hybrid 4-Bit Code	45
Appendices 2 – AND Gate Code and IP Circuit Diagram	46
Appendices 3 – CAR 90/150 6-Bit Hybrid Code and IP Circuit Diagram	47
Appendices 4 – Invertor Gate Code and IP Circuit Diagram.....	48
Appendices 5 – Clock Code and IP Circuit Diagram.....	49
Appendices 6 – OR Gate Code and IP Circuit Design.....	50
Appendices 7 – XOR Gate Code and IP Circuit Diagram	51
Appendices 8 – CAR 150 Code	52
Appendices 9 – CAR 90 Code	53

Table of Figures

Figure 1: SMT package (Storey, 2009).	8
Figure 2: DIP package (Storey, 2009).	9
Figure 3: Automated Optical Inspection system diagram (Axiomtek, 2018).	10
Figure 4: X-ray imaging (a) fluoroscopy (b) three dimensional reverse geotmetry imaging (Landers, et al., 1994).	13
Figure 5: GenRad 2287 ICT (SMTnet, 2018).	17
Figure 6: Comparison table of automated testing methods relating to PCB manufacturing (Poole, 2016).	17
Figure 7: Wolfram's CAR rule.	18
Figure 8: Rule 150 CAR.	18
Figure 9: CAR 90/150 6-bit hybrid circuit built on MultiSim.	19
Figure 10: Logic analyser screenshot taken from MultiSim circuit.	21
Figure 11: Fault 1 introduced to MultiSim circuit.	22
Figure 12: Fault 2 introduced to MultiSim circuit.	23
Figure 13: Fault 3 introduced to MultiSim circuit.	24
Figure 14: Vivado CAR 150 4-bit simulated timing diagram.	25
Figure 15: Vivado CAR 90 4-bit simulated timing diagram.	26
Figure 16: Vivado CAR 90/150 4-bit hybrid simulated timing diagram.	27
Figure 17: Vivado CAR 90/150 4-bit hybrid IP circuit diagram.	27
Figure 18: Vivado CAR 90/150 6-Bit Hybrid simulated timing diagram (section one).	28

Figure 19: Vivado CAR 90/150 6-Bit Hybrid simulated timing diagram (section two).	28
Figure 20: Vivado CAR 90/150 6-Bit Hybrid simulated timing diagram (section three).	28
Figure 21: Vivado CAR 90/150 6-Bit Hybrid simulated timing diagram (section four).	29
Figure 22: Vivado CAR 90/150 6-bit hybrid IP circuit diagram.	29
Figure 23: Vivado pseudo random pattern generator.	30
Figure 24: Xilinx PCB with downloaded software and built circuit on breadboard.	31
Figure 25: Explanation of Xilinx board components.	31
Figure 26: Breadboard built circuit to allow faults to be introduced.	32
Figure 27: 74ALS21 Dual 4-Input AND Gate pin-out diagram.	32
Figure 28: First fault introduced to the breadboard circuit.	33
Figure 29: Second fault introduced to the breadboard circuit.	34
Figure 30: Third fault introduced to the breadboard circuit.	35
Figure 31: Schematic of PCB components using the Eagle software.	36
Figure 32: Circuit layout schematic design using the Eagle software.	36
Figure 33: CMP file viewed using Online Gerber-Viewer.	37
Figure 34: CMP and DRD files viewed using Online Gerber-Viewer.	37
Figure 35: CMP, DRD and PLC files viewed using Online Gerber-Viewer.	38
Figure 36: CMP, DRD, PLC and SOL files viewed using Online Gerber-Viewer.	38
Figure 37: CMP, DRD, PLC, SOL and STC files viewed using Online Gerber-Viewer.	39

Figure 38: CMP, DRD, PLC, SOL, STC and STS files viewed using Online Gerber-Viewer.	39
Figure 39: Eagle generated file definitions (build-electronic-circuits, 2018).....	40
Figure 40: PCB fabricated by local manufacturers prior to soldering components. ...	40
Figure 41: Manufactured and soldered PCB with no faults present.	41
Figure 42: Power supply to PCB switched off to introduce a fault.	42
Figure 43: CAR 90/150 hybrid 4-bit Vivado coding.	45
Figure 44: Four input AND gate Vivado code.....	46
Figure 45: Four input AND gate IP circuit diagram.	46
Figure 46: CAR 90/150 6-Bit Hybrid Vivado code.	47
Figure 47: CAR 90/150 6-Bit Hybrid IP circuit diagram.	47
Figure 48: Invertor Vivado code.	48
Figure 49: Invertor IP circuit diagram.	48
Figure 50: Clock Vivado code.	49
Figure 51: Clock IP circuit diagram.....	49
Figure 52: OR gate Vivado code.	50
Figure 53: OR gate IP circuit diagram.	50
Figure 54: XOR gate Vivado code.....	51
Figure 55: XOR gate IP circuit diagram.....	51
Figure 56: CAR 90 Vivado code.....	52
Figure 57: CAR 90 Vivado code.....	53

Table of Tables

Table 1: Test results for CAR 90/150 6 bit hybrid circuit.	20
Table 2: MultiSim fault simulation test result table.	24

1 – Introduction

The overall aim of this assignment is to provide evidence that the student can evaluate and analyse a range of prototype development techniques, strategies and processes concerning a fully manufactured and tested product, using expert knowledge, which meets consumer needs.

The following objectives have been identified to enable the overall aim to be ascertained:

- Understand the use of automatic test equipment (ATE) to test PCB and IC systems.
- PCB technology and manufacturing processes must be investigated and understood, including PTH and SMT soldering processes.
- Design a product using a PCB prototype to meet user or customer requirements.

Research will be conducted in relation to the PCB technology, and the soldering process, for double sided PCBs namely SMT technology and SMT & PTH technology. The testing of these PCB systems, identified as automatic optical inspection (AOI) and automatic X-ray Inspection will also be discussed.

A 6-bit Hybrid CAR (90/150) circuit will be designed and simulated using a CAD system with the results analysed to ensure maximum length is obtained. This design will then be implemented practically, using a hardware descriptive language, with a further analysis of test results being produced.

This Hybrid CAR design will then be adapted to suit a double-sided PCB design with the use of SMT components to enable manual power and input/output connections.

A PCB will subsequently be manufactured with all components inserted accordingly. A "Reflow" oven will be utilised to conduct the soldering process as far as practically possible.

The PCB produced will be used to test the validity of a digital prototype product, with six inputs and one output, to meet customer or user requirements.

Introduction is clear.

2 – Research

An integrated circuit (IC) is typically mounted on a printed-circuit board (PCB) which is used to connect it to other ICs. Digital systems incorporate multilayer PCBs consisting of copper wiring etched onto multiple thin layers of fibreglass. These are then laminated into a single board that is normally 1/16th inch thick (Wakerly, 2006).

2.1 – SMT Technology

Components in modern PCBs use surface-mount technology (SMT). The leads of SMT IC packages are manipulated to ensure flat contact is made with the top surface of the PCB. A special “solder paste” is utilised on the PCB contact pads using a stencil before SMT components are mounted onto the surface. This paste is applied to ensure all hole patterns are aligned ready for the contact pads to be soldered. By hand or by machine, the SMT components are placed onto the pads and placed into an oven that melts the solder paste holding the components in place. This molten paste solidifies once cooled (Wakerly, 2006).

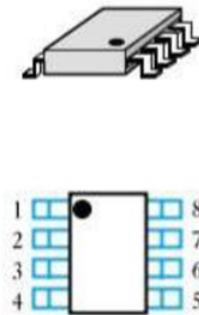


Figure 1: SMT package (Storey, 2009).

The SMT package, shown in *Figure 1*, consist of much smaller sizes which can make them difficult to be assembled manually, however, these are most commonly used for high-volume products using automatic assembly techniques (Storey, 2009).

Both packages have pins numbered anticlockwise, when viewed from the top, with pin one signified by a notch and/or dot (Storey, 2009).

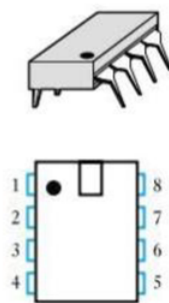
2.2 – SMT and PTH Technology

Double-sided boards have circuits on both sides that can be connected by drilling a hole through the board before depositing metal on the internal surface of the hole. This process is known as through-hole metallisation and if the internal surface is coated, as opposed to filled, it is known as plated-through hole (PTH) (Landers, et al., 1994).

PTH Technology relates to three basic profiles of components that are known as: multiple-lead, axial-lead and radial-lead:

- Multiple-lead:
 - Components consisting of a collection of components, which are housed in a single package, such as the DIP package shown in *Figure 2*.
- Axial-lead:
 - Components are usually cylindrical in shape with leads running through the centre and legs either end. These can include diodes, resistors and capacitors.
- Radial-lead:
 - These components have various shapes but all leads exit from the component from one end. For example, a transistor's leads typically exit from the underneath (Landers, et al., 1994).

Dual in-line pin (DIP) packages have long leads, which stick through the PCB ready for underside soldering. Where circuits are to be produced in small numbers the DIP package, shown in *Figure 2*, is commonly used (Storey, 2009).



Good comparison and diagrams.

Figure 2: DIP package (Storey, 2009).

2.3 – Automatic Optical Inspection

Automatic optical inspection (AOI) systems monitor PCBs for defects visually. This system can be incorporated into the production line post soldering process to ensure issues are identified early. Nodules, stains and scratches, along with a variety of other surface feature defects, are detected in addition to dimensional defects including shorts, thinning solder and open circuits. Incorrect, missing or misplaced components are also identified using this process. Prior to these systems being applied to the industry, this form of checking would be carried out manually by an operator (Poole, 2016).

AOI involves visual scanning of the surface of the PCB. The board is lit by numerous light sources and one, or numerous, high definition cameras are utilised to construct an image of the board. This captured image is processed and subsequently compared with data the machine contains of what the ideal PCB should look like. This comparison between the actual and ideal images enables the AOI system to highlight and detect any suspect or defect areas (Poole, 2016).

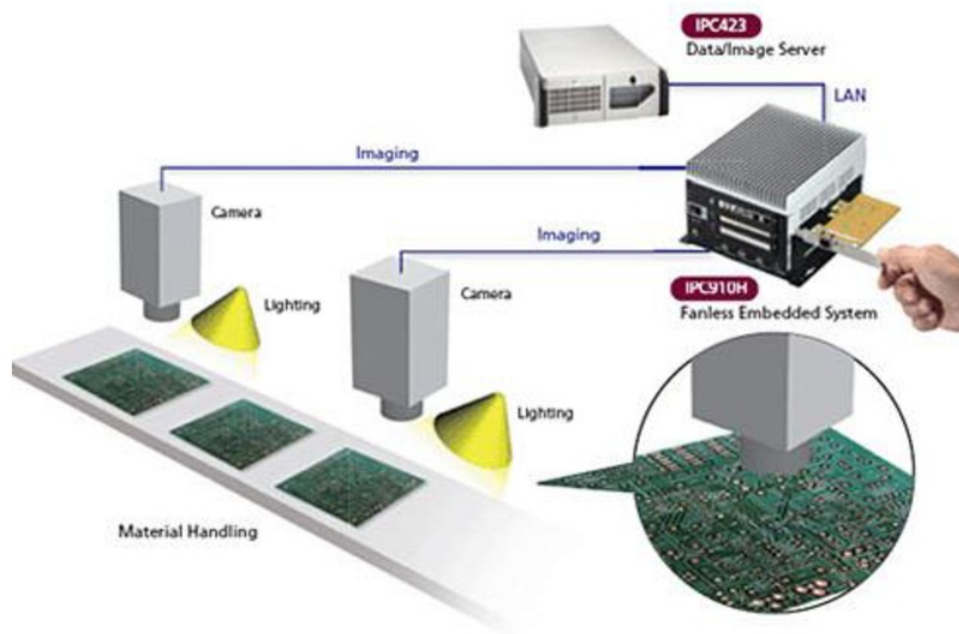


Figure 3: Automated Optical Inspection system diagram (Axiomtek, 2018).

AOI uses several techniques to produce an analysis of whether a PCB is defective or satisfactory:

- Template Matching:
 - Image from cameras is compared to an image from an ideal board, also referred to as a “golden board”.
- Pattern Matching:
 - System stored data of both satisfactory and poor PCBs. These are then matched with the generated image to confirm compliance.
- Statistical Pattern Matching:
 - Similar to previously explained techniques, however, a wide variety of failures and generated PCB images are stored in the system and a statistical method is applied to address issues. This technique allows accommodation for minor acceptable discrepancies without producing errors (Poole, 2016).

The image capturing aspect of the AOI system can be conducted through streaming video or a still image camera system. The first concept functions by frames being obtained through a streaming video with the single generated image being produced from these completed frames. This technique is not as accurate as the other highlighted form but it can be conducted at a high speed. The second concept uses a camera to take a photo of the PCB. This method requires a high quality lighting system and the camera must be closely placed to the target device (Poole, 2016).

A variety of light sources can be applied to the system including:

- Fluorescent Lighting:
 - This is the most commonly used form of lighting in AOI applications. The disadvantage with fluorescent lighting is the lamps degrade over time which can have a detrimental effect on the light quality and hence the defect identification process.

- LED Lighting:
 - This technological advance has enabled the AOI system's lighting source to reach a high level of stability. The degradation of the light source can be compensated for by increasing the current flow through the LEDs, which also enables the intensity to be controlled by the machine operator. This form of lighting is superior to earlier lighting technologies such as fluorescent and incandescent.
- Infra-red or Ultra-violet:
 - These forms can better identify certain defects within the PCB as this light source can emphasise them more effectively (Poole, 2016).

In order for the AOI system to function efficiently the machine must be programmed correctly. The machine can be configured using algorithms where PCB data is inputted and the system generates its own profile of the ideal board. This method is beneficial as fewer actual boards are required. The other technique involves a "golden board" which entails using several high quality PCBs that are passed through the machine. The AOI system learns their attributes, such as solder profiles and components, in order to compile the data required to compare the PCBs to be tested (Poole, 2016).

2.4 – Automatic X-Ray Inspection

X-ray inspection systems place an object between an X-ray source and an image-recording device, for example a halide film. X-rays are converted into visible light when striking a phosphor-coated screen when using a non-destructive evaluation technique known as fluoroscopy. The resultant image is transmitted to a monitor via a video camera (Landers, et al., 1994).

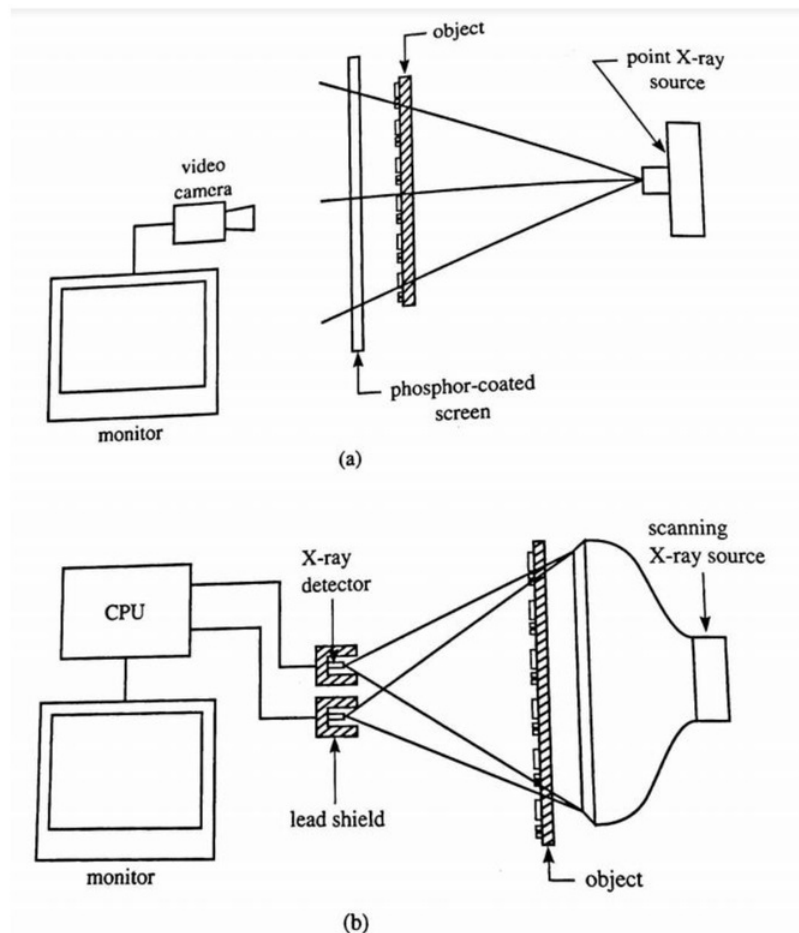


Figure 4: X-ray imaging (a) fluoroscopy (b) three dimensional reverse geometry imaging (Landers, et al., 1994).

Fluoroscopy and film images lack clarity of visible light spectrum photography, which is caused by scattered radiation. This occurs when the X-rays are impinged upon the object. This results in image degradation as the scatter radiation randomly strikes the image-capturing medium (Landers, et al., 1994).

To overcome the negative characteristics of this process, the three-dimensional reverse geometry technique was developed which reduces scatter radiation and increases pictorial clarity due to the size and placement of the detector. This technique provides an opposite approach to X-ray inspection as it uses a large source for X-ray generation and a point source detector that is illustrated in *Figure 3b*. Data from the detector is then inputted directly into the computer producing a digitised X-ray image that is displayed on a high-resolution monitor (Landers, et al., 1994).

Reverse geometry X-ray systems determine solder thickness uniformity on surface-mounted device leads without mechanical manipulation (Landers, et al., 1994).

2.5 – In Circuit Testing

In circuit testing (ICT) uses a bed of nails to connect PCB components to a machine which measures each component's performance such as resistance, capacitance and other variables. The machine will check each component in turn to evaluate whether or not it has been correctly placed and is of the correct value. The three major elements of an ICT device are as follows:

- In Circuit Tester:
 - The system consists of a matrix of sensors and drivers that conduct a variety of measurements on the target device once placed onto a large connector located on the machine.
- Fixture:
 - This system connector interfaces the actual test PCB and the in circuit tester. Each fixture will be designed specific to the type of board and connects the driver sensor points directly to the relevant points on the PCB using a "bed of nails".
- Software:
 - Board specific software is written into the machine that instructs the system what tests to perform and what measurements dictate whether the board has passed or failed (Poole, 2016).

Advantages of ICT include ease of detection of manufacturing defects, ease of programme generation and ease of test result interpretation. Disadvantages of ICT can include expensive fixtures, difficulty in updating fixtures and difficulty in test access on larger PCBs (Poole, 2016).

There are also several variations of ICT. The main types of machine available consist of:

- Standard ICT Machine:
 - These machines are capable of basic resistance / continuity measurements but can possibly also test capacitance and device functionality.

- Flying Probe Tester:
 - Due to complete access bed of nails fixtures being costly and difficult to alter to suit variations in PCB design another approach is the use of a roving or flying probe. This method holds the PCB in place with all connections made through a few probes that can move around the circuit and make contact with components at will. These probes are manoeuvred under software control consequently making this method highly versatile in alterations in PCB design being accommodated for through software amendments.
- Manufacturing Defect Analyser (MDA):
 - This form of testing provides measurements for basic variables such as insulation and resistance. It is also extremely effective at identifying short circuits across tracks and open circuit connections.
- Cableform Tester:
 - The method concerns testing cables. Similar to the MDA with one difference being that occasional high voltages are applied to check for insulation.

An example of an ICT system can be seen in *Figure 5*. The manufacturer GenRad offer a range of bench-top component testers. Their 1735 Component Test System can be used for linear, discrete and digital components. The user inputs the appropriate limits and specifications into the workstation that the PCB on test must adhere to (Riley, 1988).

Figure 6 illustrates a comparison table highlighting possible PCB defects and the possibility each of the three forms of defect detection methods discussed in this section of the report to identify the relevant defect.

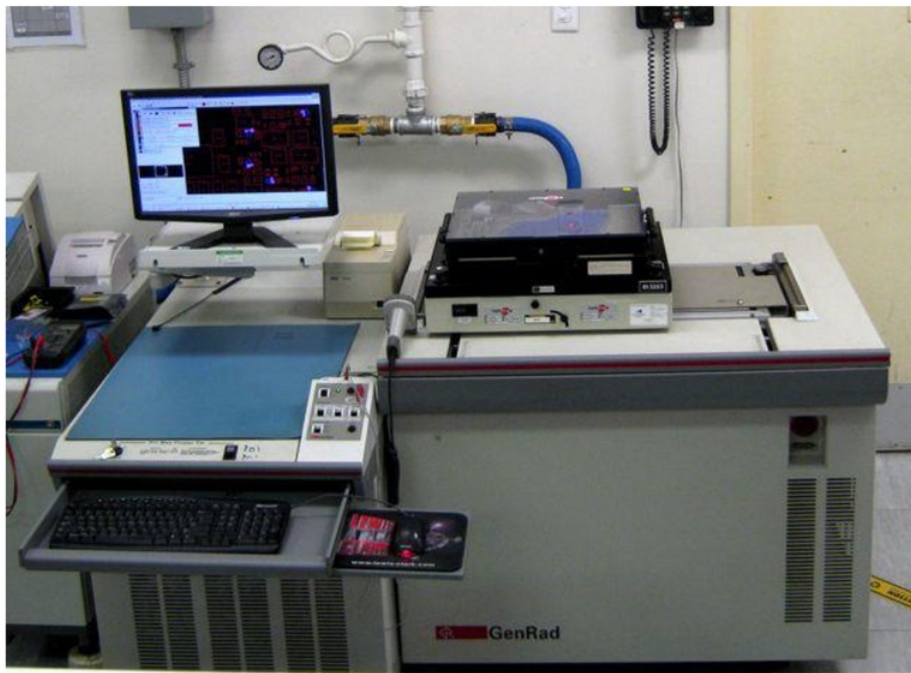


Figure 5: GenRad 2287 ICT (SMTnet, 2018).

COMPARISON OF THE MAJOR DEFECT DETECTION CAPABILITIES OF AOI, AXI AND ICT				
	DEFECT TYPE	AOI	AXI	ICT
Soldering defects				
	Open circuits	Y	Y	Y
	Solder bridges	Y	Y	Y
	Solder shorts	Y	Y	Y
	Insufficient solder	Y (not heel of joint)	Y	N
	Solder void	N	Y	N
	Excess solder	Y	Y	N
	Solder quality	N	Y	N
Component defects				
	Lifted lead	Y	Y	Y
	Missing component	Y	Y	Y
	Misaligned or misplaced component	Y	Y	Y
	Incorrect component value	N	N	Y
	Faulty component	N	N	Y
BGA and CSP defects				
	BGA shorts	N	Y	Y
	BGA open circuit connections	N	Y	Y

Figure 6: Comparison table of automated testing methods relating to PCB manufacturing (Poole, 2016).

Good explanation and coverage of ATE types used in modern production industry.

4 – Design, Testing and Analysis of NI MultiSim

A cellular automata register (CAR) is made up of flip-flops whose next state is determined by prior values and the neighbouring flip-flops values.

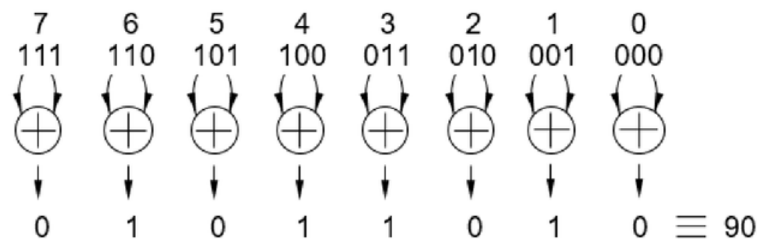


Figure 7: Wolfram's CAR rule.

Figure 7, devised by Wolfram, refers to a circuit known as a, "Rule 90 CAR" in which the next state of a flip-flop output, given as Q_t^+ , is achieved by inputting the flip-flop outputs into exclusive OR logic gates, for example: Q_{t-1} and Q_{t+1} .

- Rule 90 is: $Q_t^+ = Q_{t-1} \oplus Q_{t+1}$

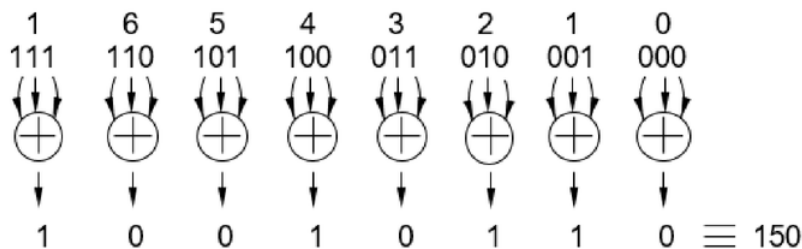


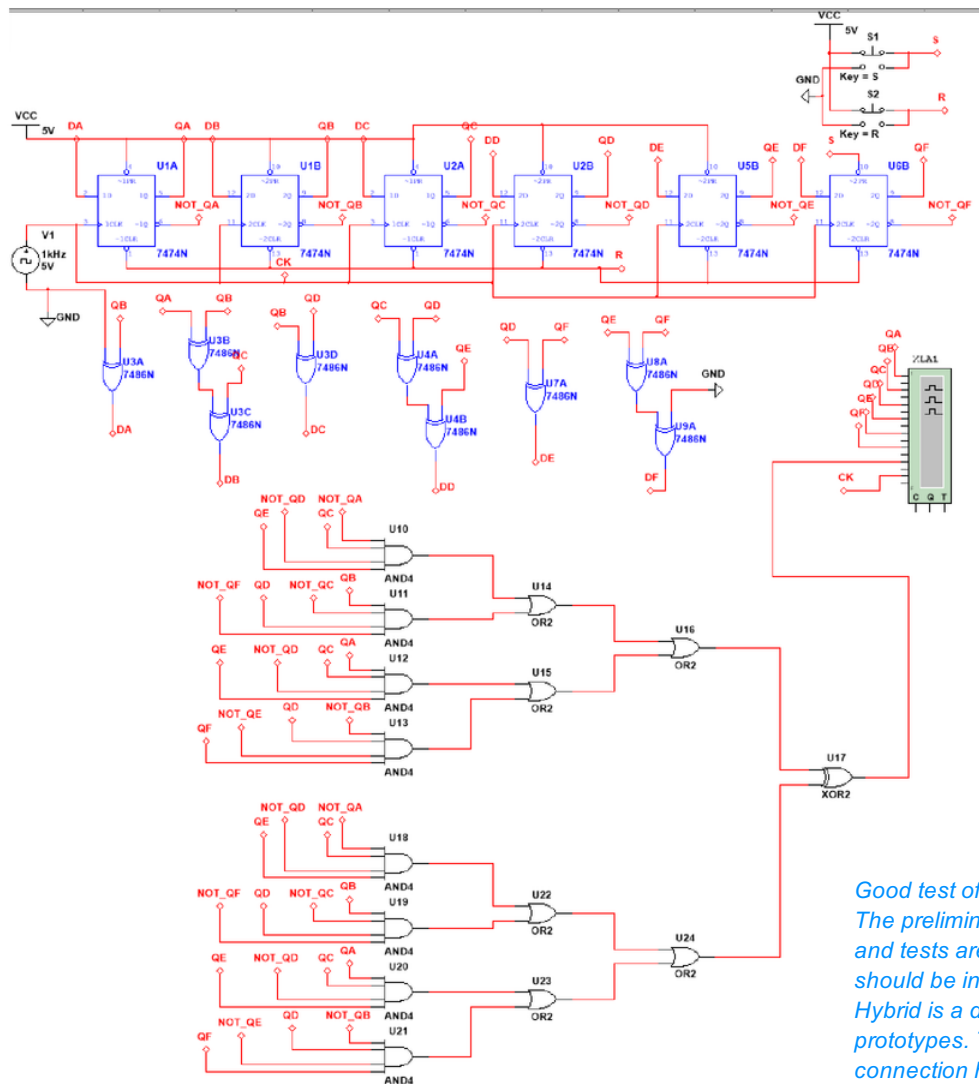
Figure 8: Rule 150 CAR.

Figure 8 illustrates the rule for a CAR 150 circuit.

- Rule 150 is: $Q_t^+ = Q_{t-1} \oplus Q_t \oplus Q_{t+1}$

These two circuits can be combined to form a hybrid circuit that forms a maximal length pseudo-random number generator. A CAR 90/150 6 bit hybrid circuit was constructed using the MultiSim software to test these theories.

*This is a logical
explanation of CAR rules
90 and 150 operation.*



Good test of hybrid CAR.
The preliminary prototypes
and tests are missing and
should be included. This
Hybrid is a derivative of the
prototypes. The CUT is
connection here is correct.

Figure 9: CAR 90/150 6-bit hybrid circuit built on MultiSim.

The software was simulated and a table of results was compiled (shown in Table 1) and a screenshot of the logic analyser has been included in Figure 10, which corresponds with the binary count identified in Table 1.

Table 1: Test results for CAR 90/150 6 bit hybrid circuit.

Car 90	Car 150	Car 90	Car 150	Car 90	Car 150	Bin count	Repeats y/n
0	0	0	0	0	1	1	Y
0	0	0	0	1	1	3	N
0	0	0	1	1	0	6	N
0	0	1	0	1	1	11	N
0	1	0	0	1	0	18	N
1	1	1	1	0	1	61	N
1	1	0	0	0	1	49	N
1	0	1	0	1	1	43	N
0	0	0	0	1	0	2	N
0	0	0	1	0	1	5	N
0	0	1	1	0	1	13	N
0	1	1	0	0	1	25	N
1	0	1	1	1	1	47	N
0	0	1	1	0	0	12	N
0	1	1	0	1	0	26	N
1	0	1	0	0	1	41	N
0	0	0	1	1	1	7	N
0	0	1	0	0	0	8	N
0	1	0	1	0	0	20	N
1	1	1	0	0	0	54	N
1	0	0	0	1	1	35	N
0	1	0	1	1	0	22	N
1	1	0	0	1	1	51	N
1	0	1	1	1	1	45	N
0	0	1	1	1	1	15	N
0	1	1	0	0	0	28	N
1	0	0	0	1	0	34	N
0	1	0	1	0	1	21	N
1	1	0	1	0	1	53	N
1	0	0	1	0	1	37	N
0	1	1	1	0	1	29	N
1	0	0	0	0	1	33	N
0	1	0	0	1	1	19	N
1	1	1	1	1	0	62	N
1	1	0	1	1	1	55	N
1	0	0	0	0	0	32	N
0	1	0	0	0	0	16	N
1	1	1	0	0	0	56	N
1	1	1	1	0	0	60	N
1	1	0	0	1	0	50	N
1	0	1	1	0	1	45	N
0	0	1	0	0	1	9	N
0	1	0	1	1	1	23	N
1	1	0	0	0	0	48	N

1	0	1	0	0	0	40	N
0	0	0	1	0	0	4	N
0	0	1	1	1	0	14	N
0	1	1	1	1	1	31	N
1	0	0	1	0	0	36	N
0	1	1	1	1	0	30	N
1	0	0	1	1	1	39	N
0	1	1	0	0	0	24	N
1	0	1	1	0	0	44	N
0	0	1	0	1	0	10	N
0	1	0	0	0	1	17	N
1	1	1	0	1	1	59	N
1	1	1	0	1	0	58	N
1	1	1	0	0	1	57	N
1	1	1	1	1	1	63	N
1	1	0	1	0	0	52	N
1	0	0	1	1	0	38	N
0	1	1	0	1	1	27	N
1	0	1	0	1	0	42	N
0	0	0	0	0	1	1	Y

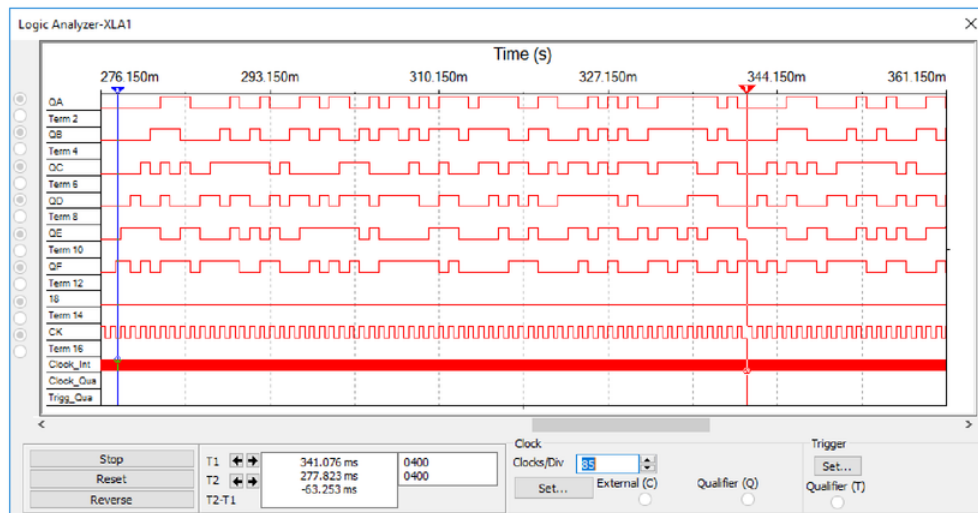


Figure 10: Logic analyser screenshot taken from MultiSim circuit.

The two y-axis cursors (T1 and T2) have been positioned from a binary count of 1 to when this count repeats itself at 1 again.

The following Boolean expression defines the above circuit:

$$\text{NOT_QA.QC.NOT_QD.QE} + \text{QB.NOT_QC.QD.NOT_QF} + \text{QA.QC.NOT_QD.QE} + \text{NOT_QB.QD.NOT_QE.QF} \rightarrow \text{Fault free circuit.}$$

The flat line labelled "18" in *Figure 10* proves that there were no faults detected in the circuit. A fault in the circuit is produced when not all of the inputs to the AND logic gates are the same as the identical second half of the logic gate circuit. When this is true, the XOR gate produces an output and a pulse can be seen on the logic analyser timing diagram.

To test the circuit three different faults were introduced within the second half of the logic circuit on the AND gate inputs.



Figure 11: Fault 1 introduced to MultiSim circuit.

During the testing of the first fault the QD input on the fourth AND gate was connected to ground to produce a "0" value. With reference to the above figure, the fault is registered where the red cursor crosses the pulse on the "18" input. This input was received from the XOR gate labelled U17 in *Figure 9*.

The following Boolean expression defines the above circuit:

$$\text{NOT_QA.QC.NOT_QD.QE} + \text{QB.NOT_QC.QD.NOT_QF} + \text{QA.QC.NOT_QD.QE} + \text{NOT_QB."0".NOT_QE.QF} \rightarrow \text{QD is to ground i.e. 0V} = \text{Fault 1.}$$

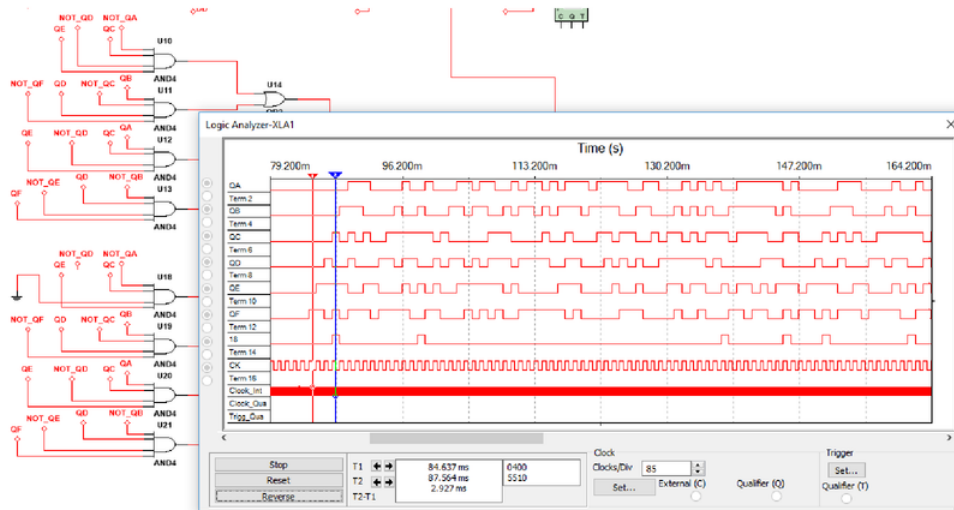


Figure 12: Fault 2 introduced to MultiSim circuit.

During the testing of the second fault the NOT_QD input on the first AND gate was connected to ground to produce a "0" value.

The following Boolean expression defines the above circuit:

$$\text{NOT_QA.QC."0".QE} + \text{QB.NOT_QC.QD.NOT_QF} + \text{QA.QC.NOT_QD.QE} + \text{NOT_QB.QD.NOT_QE.QF} \rightarrow \text{NOT_QD is to ground i.e. 0V} = \text{Fault 2}$$

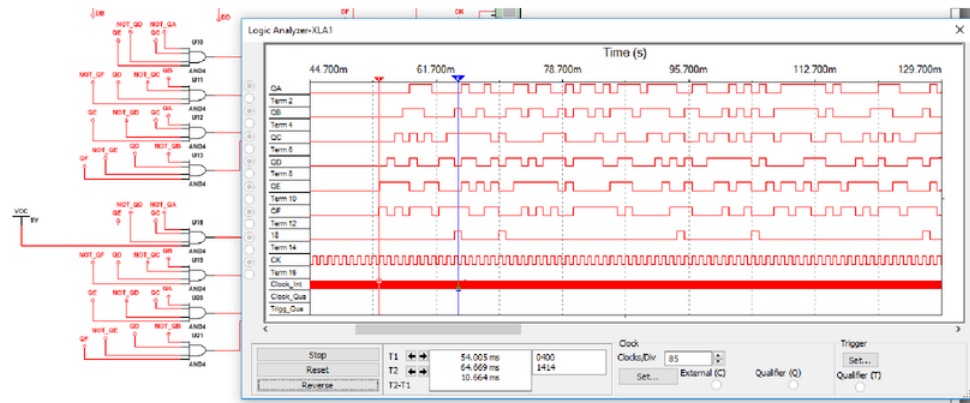


Figure 13: Fault 3 introduced to MultiSim circuit.

During the testing of the third fault the QE input on the first AND gate was connected to VCC to produce a “+5V” value.

The following Boolean expression defines the above circuit:

$$\text{NOT_QA.QC.NOT_QD. "1" + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF} \rightarrow \text{QE is to VCC i.e. 5V} = \text{Fault 3}$$

With the MultiSim simulation test results obtained, a table of results was constructed:

Table 2: MultiSim fault simulation test result table.

Boolean Output from Circuit	Boolean Output From Fault Free Circuit	Simulation Fault Boolean Outputs from Test Circuit	Fault Detected? (Y/N)
NOT_QA.NOT_QB.NOT_QC.QD.NOT_QE.QF	NOT_QA.QC.NOT_QD.QE + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF	NOT_QA.QC.NOT_QD.QE + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB."0".NOT_QE.QF	Y
NOT_QA.NOT_QB.QC.NOT_QD.QE.QF	NOT_QA.QC.NOT_QD.QE + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF	NOT_QA.QC."0".QE + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF	Y
NOT_QA.QB.QC.NOT_QD.NOT_QE.QF	NOT_QA.QC.NOT_QD.QE + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF	NOT_QA.QC.NOT_QD. "1" + QB.NOT_QC.QD.NOT_QF + QA.QC.NOT_QD.QE + NOT_QB.QD.NOT_QE.QF	Y

The test and fault analysis is good and shows the 3 different fault possibilities.

5 - Design, Testing and Analysis of Hardware Descriptive Language FPGA Technology

Using the Vivado software, various CAR programmes were constructed and simulated to prove the theory behind these pseudo random pattern generators.

5.1 – CAR 150 4-Bit

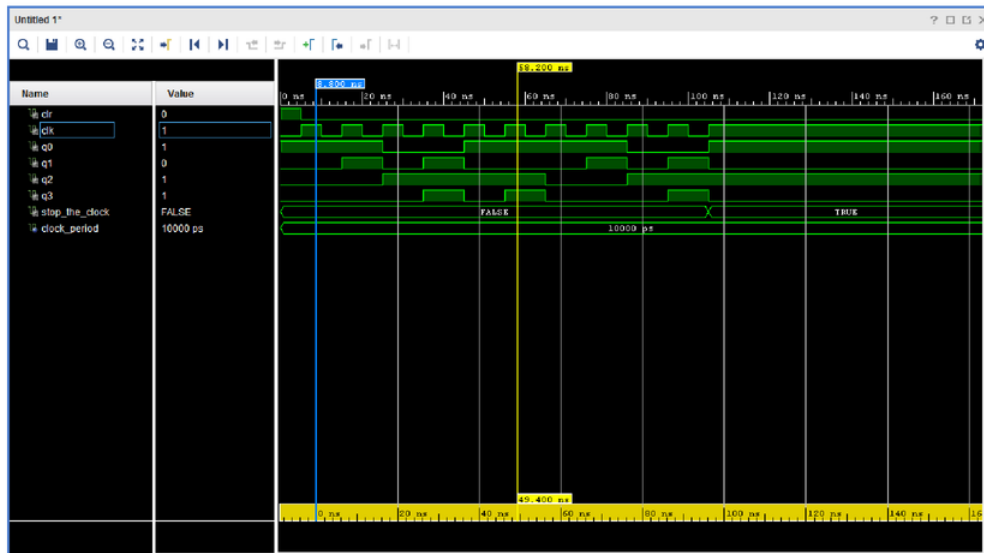


Figure 14: Vivado CAR 150 4-bit simulated timing diagram.

The above figure shows a timing diagram of a CAR 150 4-bit circuit. The blue y-axis cursor shows where the binary count starts and at 80 ns this binary count repeats. The yellow cursor signifies the previous binary count before the repeat.

5.2 – CAR 90 4-Bit

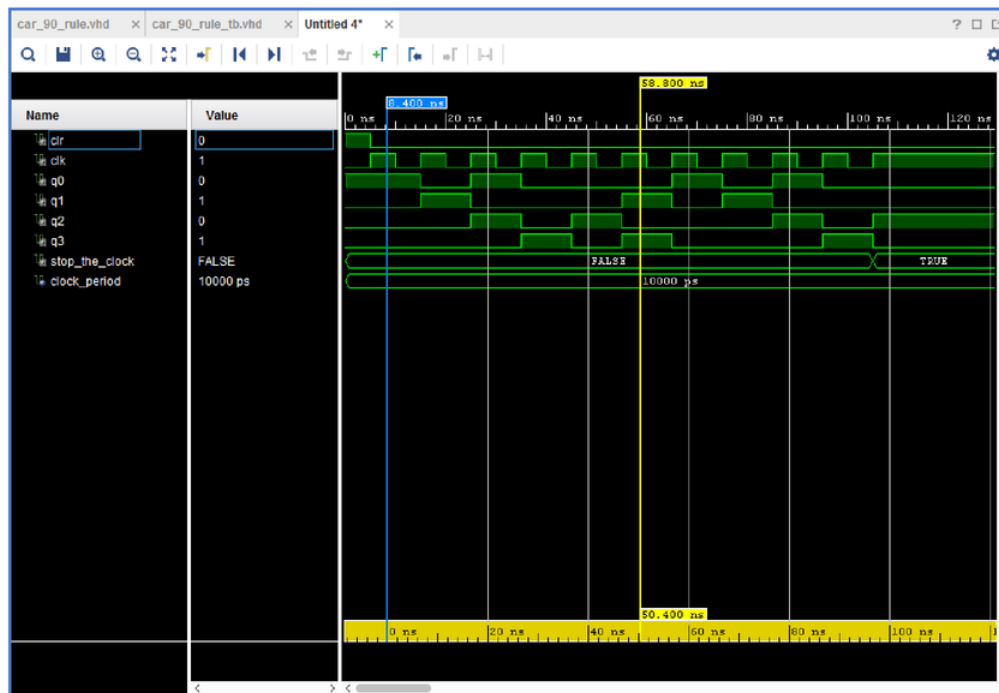


Figure 15: Vivado CAR 90 4-bit simulated timing diagram.

The above figure shows a timing diagram of a CAR 90 4-bit circuit. The blue y-axis cursor shows where the binary count starts and at 65 ns this binary count repeats. The yellow cursor signifies the previous binary count before the repeat.

5.3 – CAR 90/150 4-Bit Hybrid

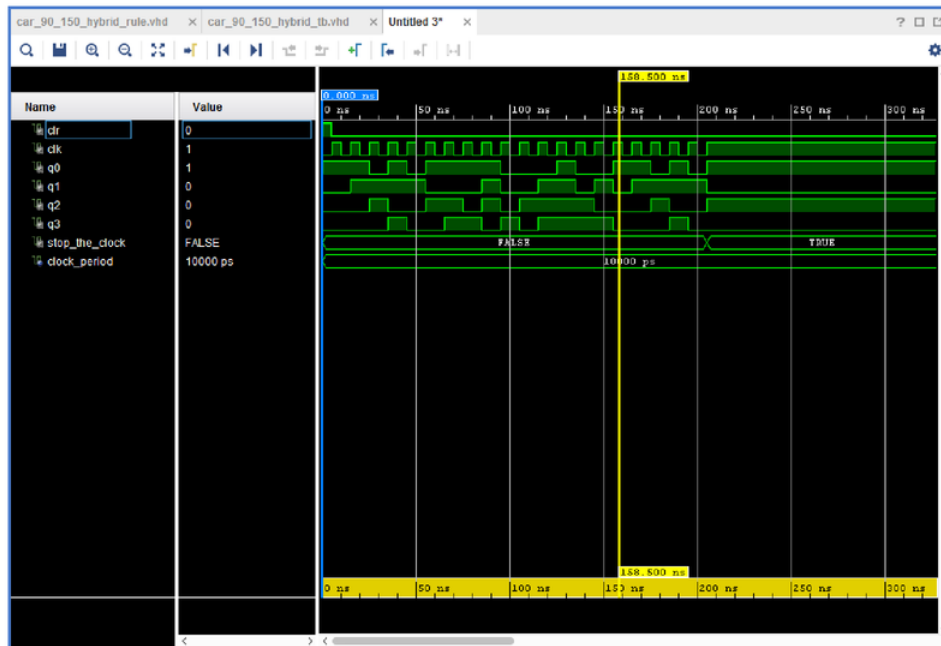


Figure 16: Vivado CAR 90/150 4-bit hybrid simulated timing diagram.

The above figure shows a timing diagram of a CAR 90/150 4-bit hybrid circuit. The blue y-axis cursor shows where the binary count starts and the yellow cursor signifies when the binary count is repeated.

Using the Vivado software the relevant IP circuit was also designed.

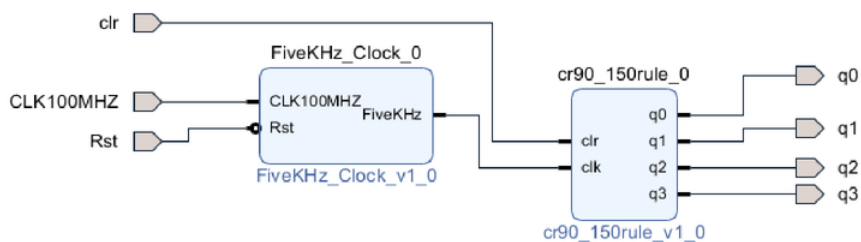


Figure 17: Vivado CAR 90/150 4-bit hybrid IP circuit diagram.

Note: See Appendices 1 for associated code.

The
prototype
design and
test looks
correct.
This
demonstrat

As seen previously in the MultiSim built circuit, this circuit counts from 1 to 63. Therefore, due to the maximum length, the following four figures all illustrate the Vivado timing diagram of the circuit from a starting bit of 1 through to when it repeats.



28

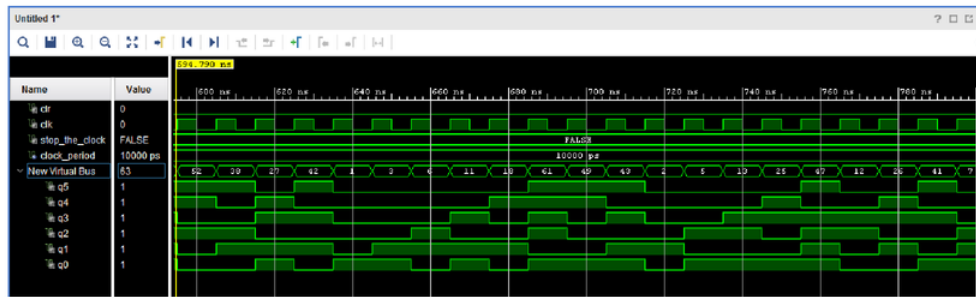


Figure 21: Vivado CAR 90/150 6-Bit Hybrid simulated timing diagram (section four).

A virtual bus was also programmed into this simulation to enable easier analysis of when the count starts to repeat.

Another IP circuit diagram was also designed using the Vivado software.

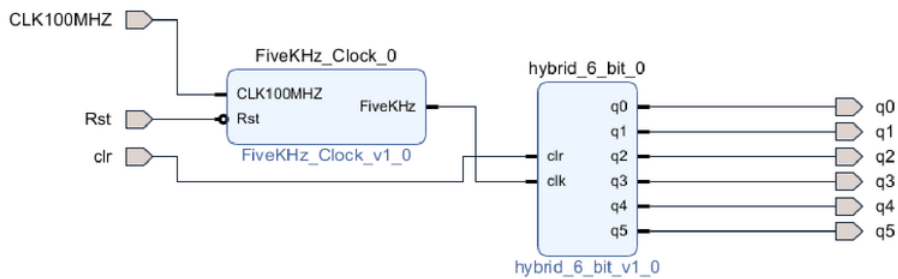


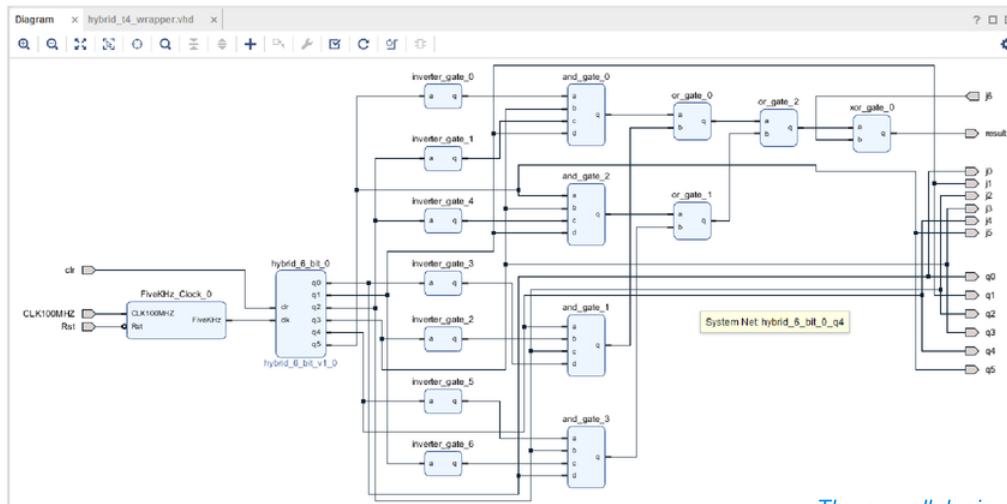
Figure 22: Vivado CAR 90/150 6-bit hybrid IP circuit diagram.

By referring back to *Table 1*, where the binary count for a CAR 90/150 6-bit hybrid circuit was identified after testing the MultiSim circuit, it can be cross-checked by observing the Vivado produced timing diagrams, *Figures 18 – 21*, to confirm the software has been programmed correctly.

Using *Appendices 3* it can be seen that the CAR 90/150 6-bit hybrid code has been constructed with reference to the CAR 90 and CAR 150 codes illustrated in *Appendices 9* and *8* respectively. Under the *architecture* section of the CAR 90/150 6-bit hybrid circuit *D(0)* to *D(5)* are adapted from *D(0)* to *D(3)* of both CAR 90 and 150 circuits.

5.5 – Xilinx Programmable Board and Built Breadboard Test Circuit

After testing the variations of the CAR circuits using the Vivado software, an identical circuit to that of the one tested using MultiSim, was produced within Vivado.



*The overall design is correct
and the test are correct.*

Figure 23: Vivado pseudo random pattern generator.

The IP circuit diagram shown above is one half of the FPGA practical test circuit.

This was used to act as the circuit with no faults that was to be compared with a physical circuit built on breadboard where actual faults were to be introduced.

The individual IP circuits, along with all associated code, are detailed in *Appendices 2 – 7*.

This Vivado built software was downloaded to a Xilinx PCB before being connected to an identical circuit built on breadboard.

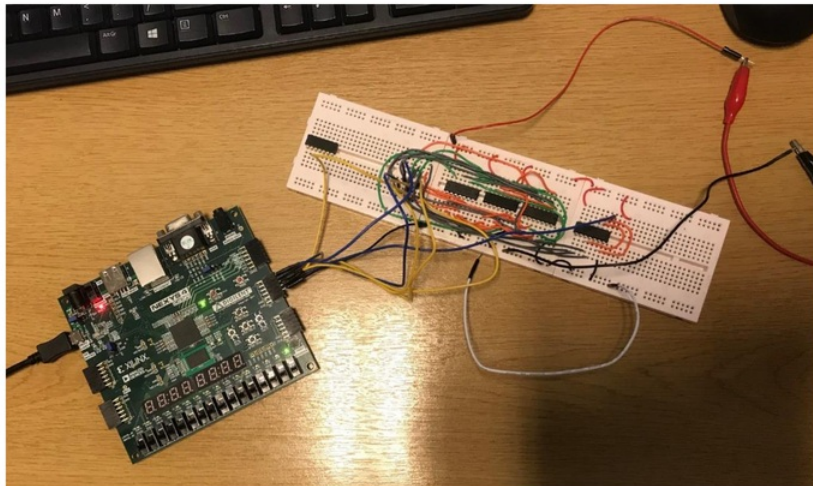


Figure 24: Xilinx PCB with downloaded software and built circuit on breadboard.

During the MultiSim tests, a logic analyser was observed to acknowledge any faults if/when present. With this practical, results were obtained by observing LEDs on the Xilinx board.

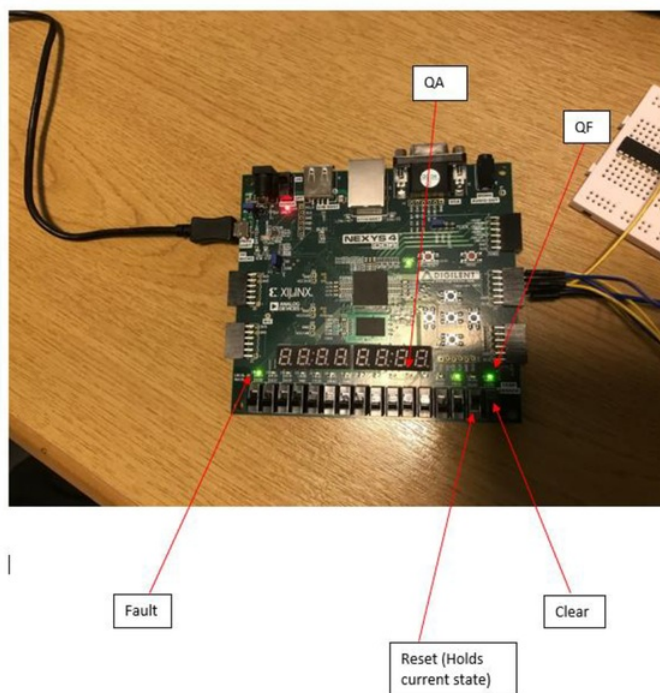


Figure 25: Explanation of Xilinx board components.

With reference to *Figure 26*, placed on the breadboard from left to right, the first and second chips are the AND gates, the third chip an inverter and the fourth chip is the OR gates. The XOR gate that is seen on the MultiSim circuit is integrated within the Xilinx chip.

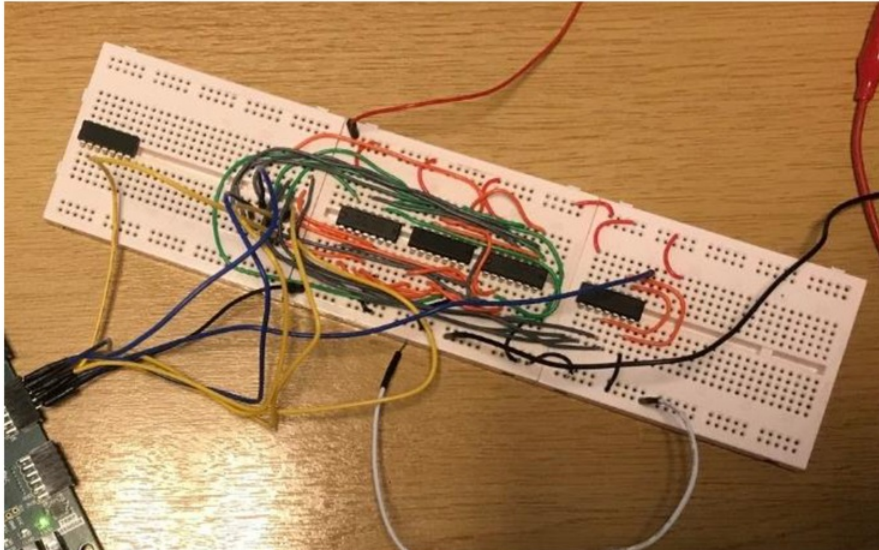


Figure 26: Breadboard built circuit to allow faults to be introduced.

In order to test this design, the same three faults used to test the MultiSim software were implemented again onto the breadboard built circuit seen above. The below figure shows the pin-out of the AND gate packages.

74ALS21 Dual 4-Input AND Gate

LOGIC SYMBOL

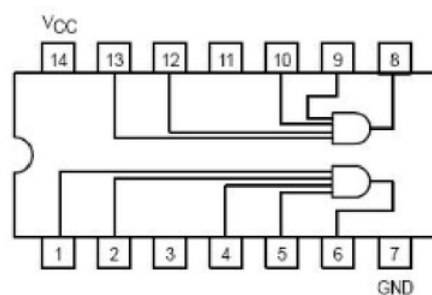


Figure 27: 74ALS21 Dual 4-Input AND Gate pin-out diagram.

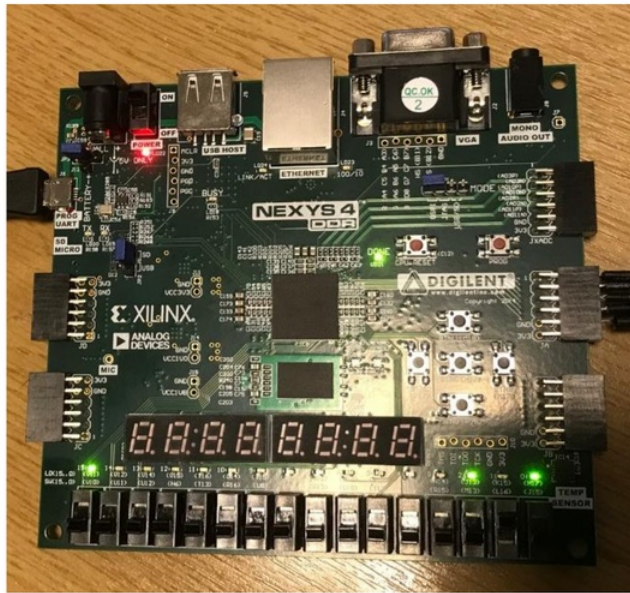


Figure 28: First fault introduced to the breadboard circuit.

The first fault was QD on the final AND gate and it was connected to ground (0 V). By looking at the chip documentation, *Figure 27*, the lead connecting pin 12 on the second chip was disconnected and connected to ground.

By observing the LEDs, LED illuminated equals logic 0, on the breadboard it was confirmed that the fault was detected and the following binary result was recorded:

Binary count: 111010

Boolean expression: NOT_QA.NOT_QB.NOT_QC.QD.NOT_QE.QF

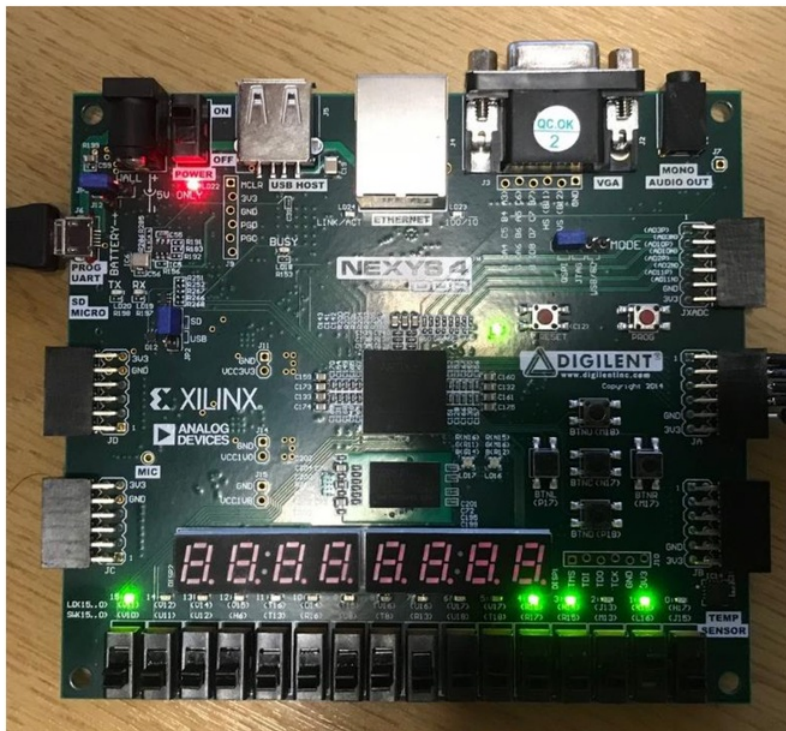


Figure 29: Second fault introduced to the breadboard circuit.

The second fault was NOT_QD on the first AND gate and it was connected to ground (0 V). The lead connecting pin 4 on the first chip was disconnected and connected to ground.

By observing the LEDs on the breadboard, it was confirmed again that the fault was detected and the following binary result was recorded:

Binary count: 100101

Boolean expression: $QA \cdot \text{NOT_QB} \cdot \text{NOT_QC} \cdot QD \cdot \text{NOT_QE} \cdot QF$.

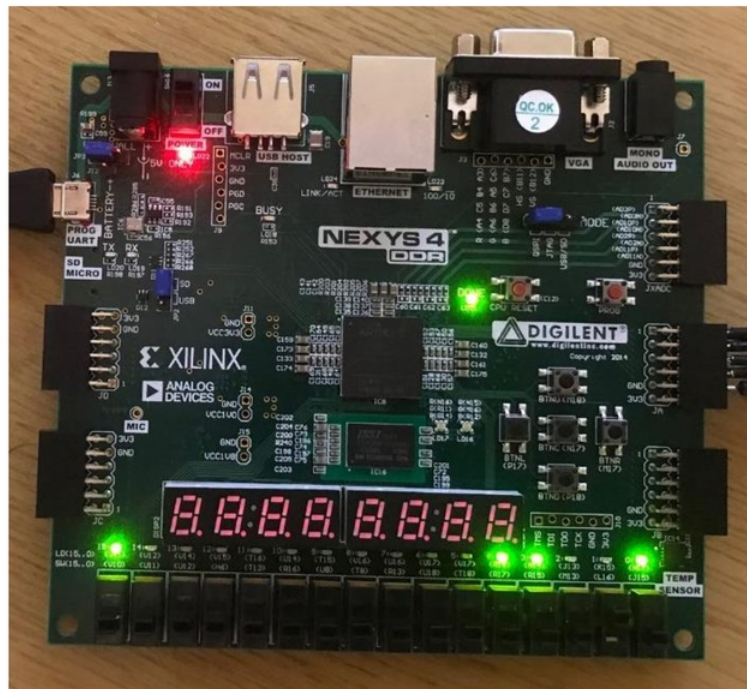


Figure 30: Third fault introduced to the breadboard circuit.

The third fault was QE on the first AND gate and it was connected to VCC (+5 V). The lead connecting pin 5 on the first chip was disconnected and connected to a supply of +5 V.

By observing the LEDs on the breadboard, it was confirmed for the final time that the fault was detected and the following binary result was recorded:

Binary count: 100110

Boolean expression: $QA.NOT_QB.NOT_QC.QD.QE.NOT_QF$

A direct comparison cannot be conducted between these Boolean expressions and the MultiSim Boolean expressions as these faults occur at various different output combinations and not exclusively to the expressions stated.

*Practical testing
with evidence is
good.*

6 – PCB Practical Design, Build, Testing and Analysis of Prototype on Prototyping Board

The next stage in the process was to design a PCB with the design subsequently being sent off to a manufacturer to be fabricated. This pre-fabricated board will include all necessary tracks to connect components with soldering being carried out at the college.

Using the Eagle software, the following schematic was designed:

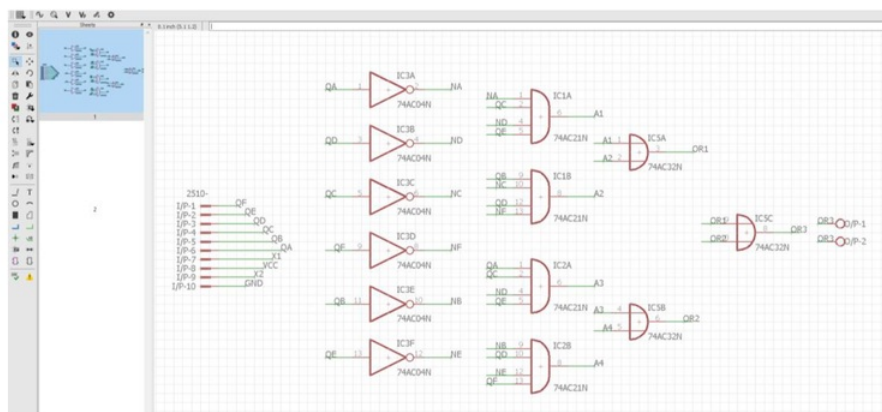


Figure 31: Schematic of PCB components using the Eagle software.

Using the Eagle software again, a circuit layout schematic was produced:

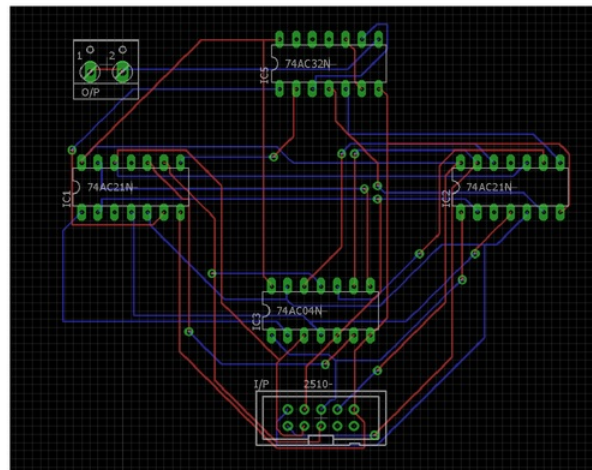


Figure 32: Circuit layout schematic design using the Eagle software.

Once this had been completed, a website known as “Online Gerber-Viewer” was used to display all the individual layers that produce the image depicted in *Figures 32 - 38*.

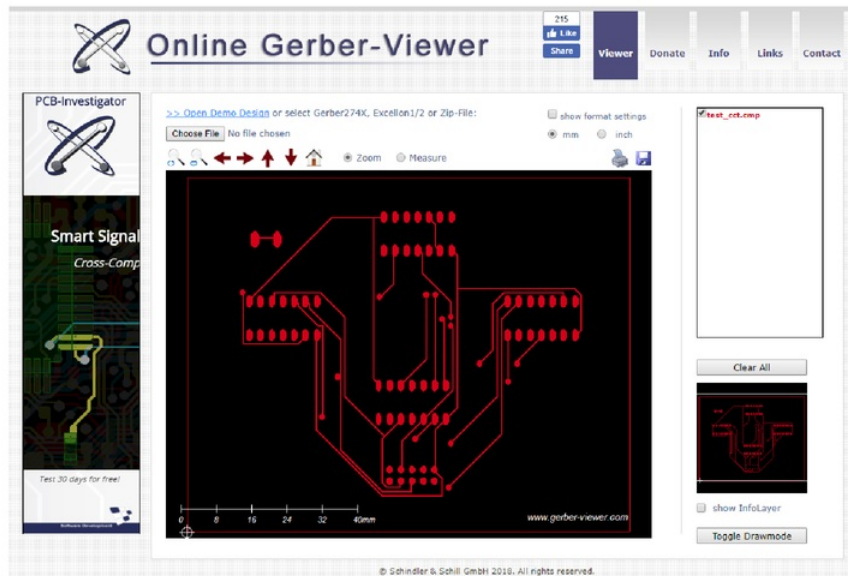


Figure 33: CMP file viewed using Online Gerber-Viewer.

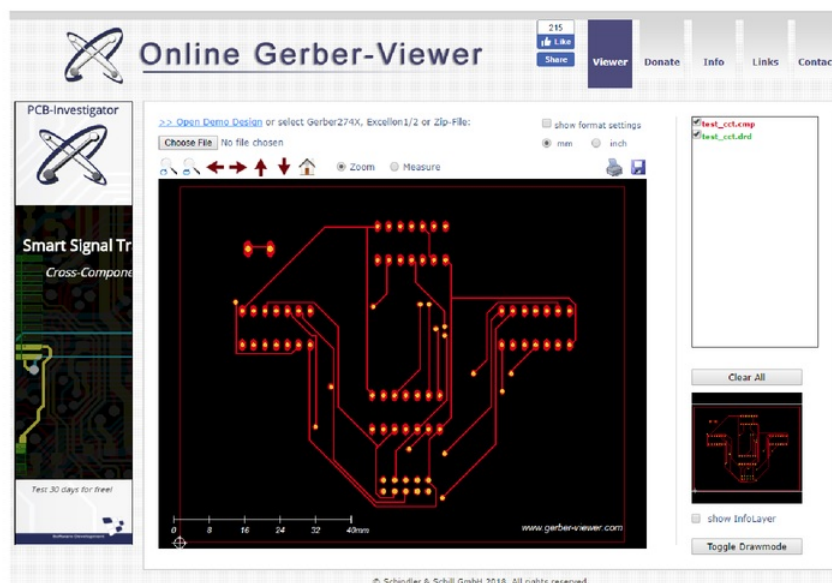


Figure 34: CMP and DRD files viewed using Online Gerber-Viewer.

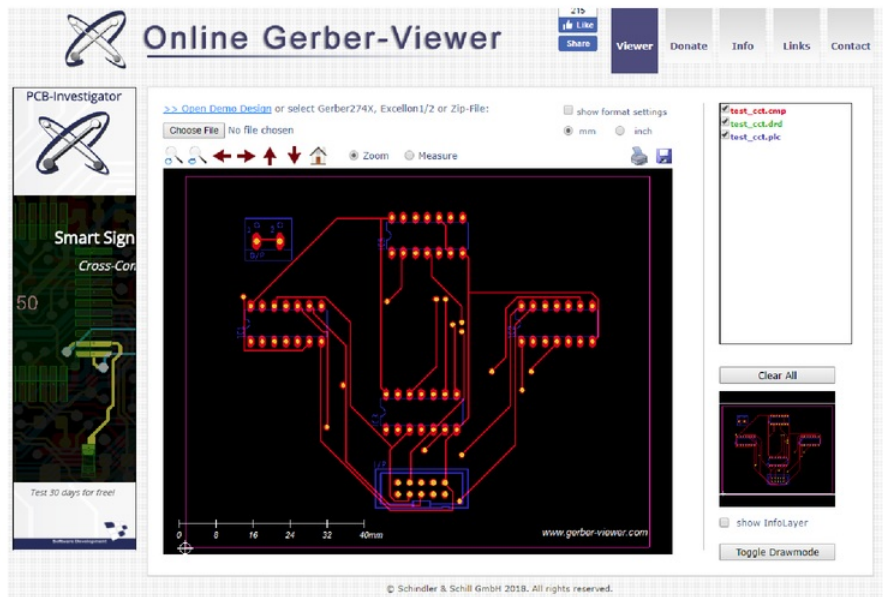


Figure 35: CMP, DRD and PLC files viewed using Online Gerber-Viewer.

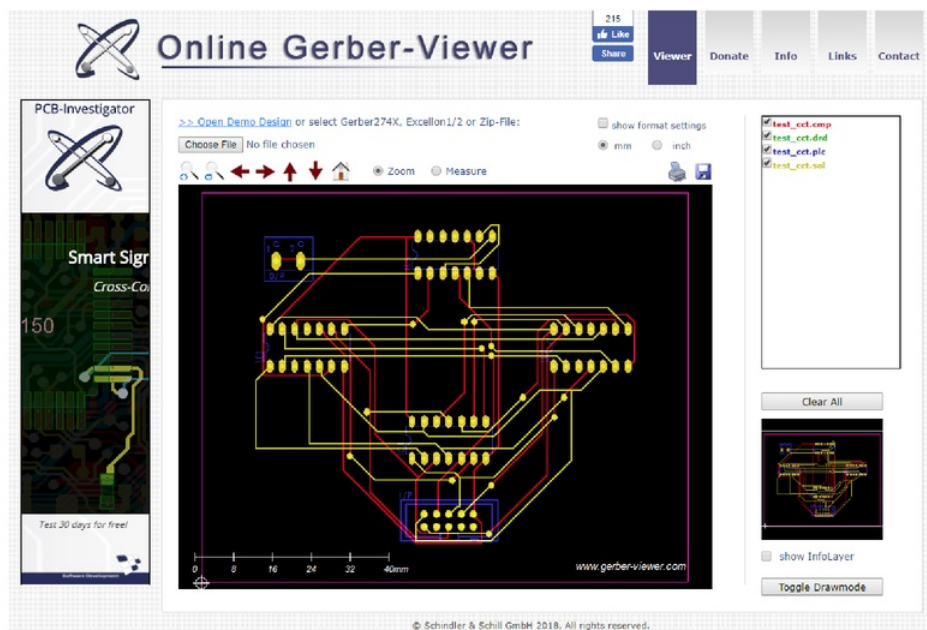


Figure 36: CMP, DRD, PLC and SOL files viewed using Online Gerber-Viewer.

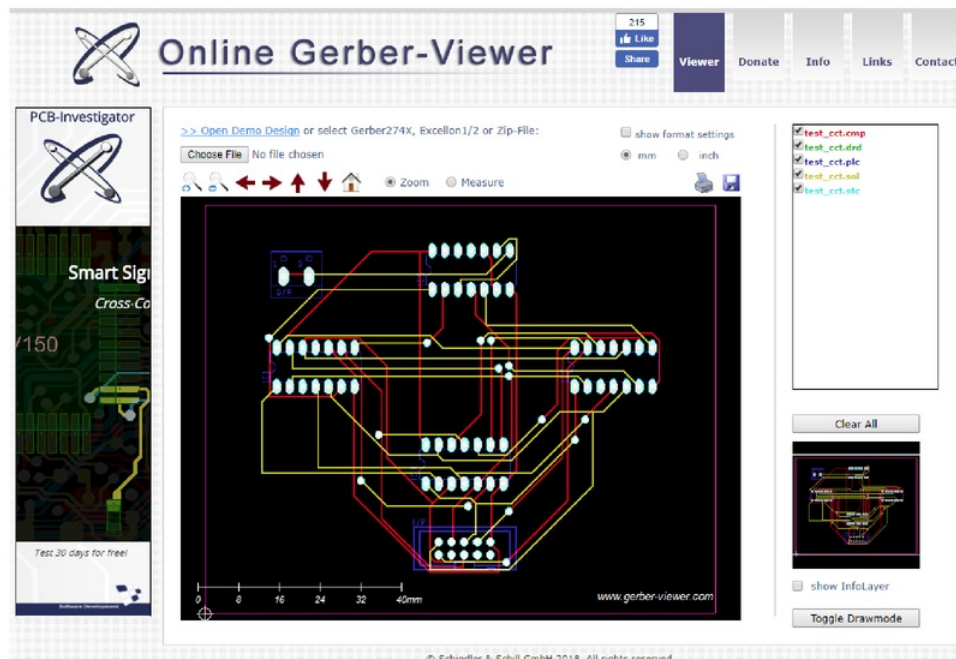


Figure 37: CMP, DRD, PLC, SOL and STC files viewed using Online Gerber-Viewer.

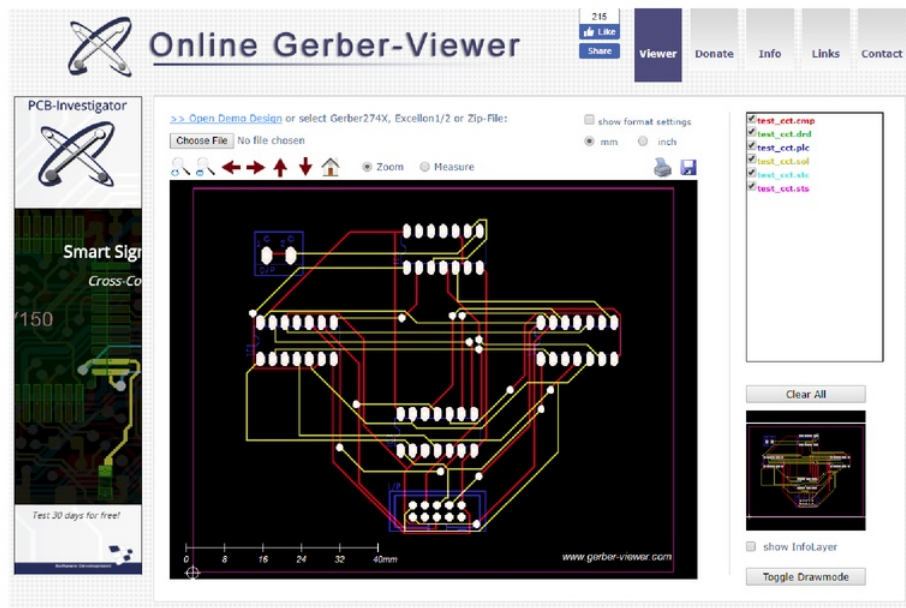


Figure 38: CMP, DRD, PLC, SOL, STC and STS files viewed using Online Gerber-Viewer.

These Eagle generated files all concern different aspects of the PCB. Each are explained below in *Figure 39*:

- *.cmp (Copper, component side)
- *.dri (Drill file)
- *.dri (Drill Station Info File) – Usually not needed
- *.gpi (Photoplotter Info File) – Usually not needed
- *.plc (Silk screen, component side)
- *.pls (Silk screen, solder side)
- *.sol (Copper, solder side)
- *.stc (Solder stop mask, component side)
- *.sts (Solder stop mask, solder side)

Figure 39: Eagle generated file definitions (build-electronic-circuits, 2018).

These layer files were sent to a local PCB manufacturing company for fabrication of the circuit board. These files are used due to their versatility. They are widely available to be interpreted on a huge range of software packages.

The unsoldered PCB was received from the manufacturer before all required components were soldered in their designated positions.

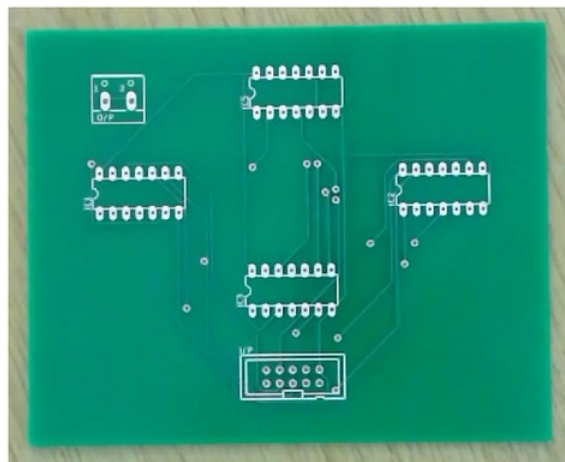


Figure 40: PCB fabricated by local manufacturers prior to soldering components.

Components were subsequently soldered to the PCB shown in *Figure 40* and connected to the Xilinx board, which had previously been programmed during earlier testing with the breadboard circuit.

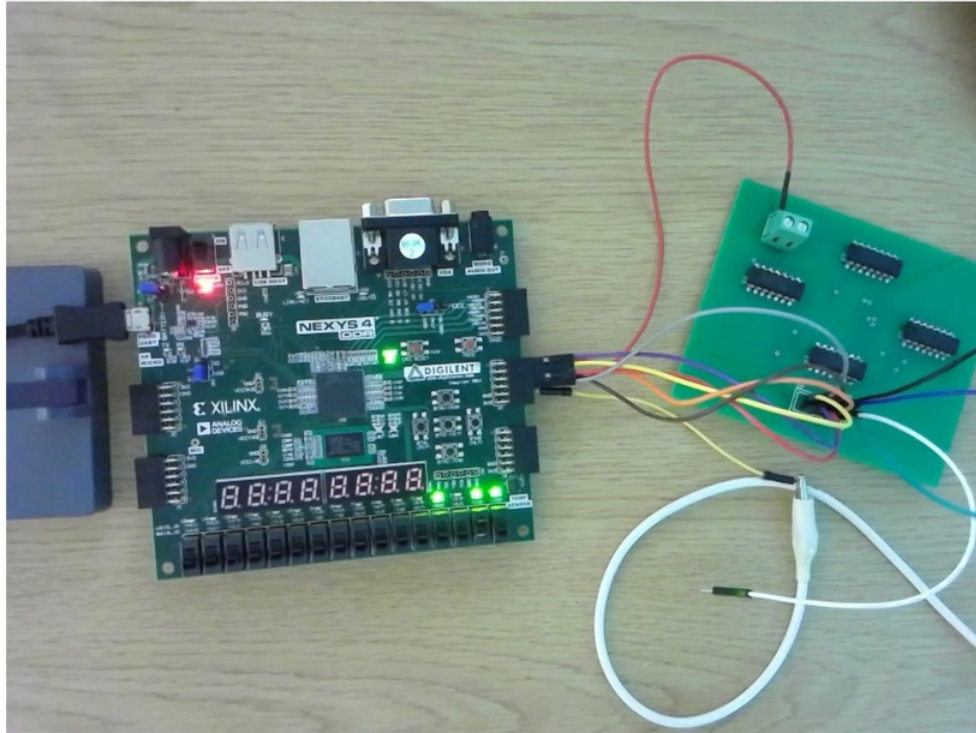


Figure 41: Manufactured and soldered PCB with no faults present.

Both the Xilinx board and PCB were powered with the LEDs being monitored for testing purposes. This sequence was observed until the maximum count had been reached and the initial binary count had repeated to ensure no faults were present in the system.

As faults could not be introduced in the PCB circuit, as in the breadboard due to the tracks being etched into the board, the power was turned off to the PCB to test the Xilinx board fault detection.

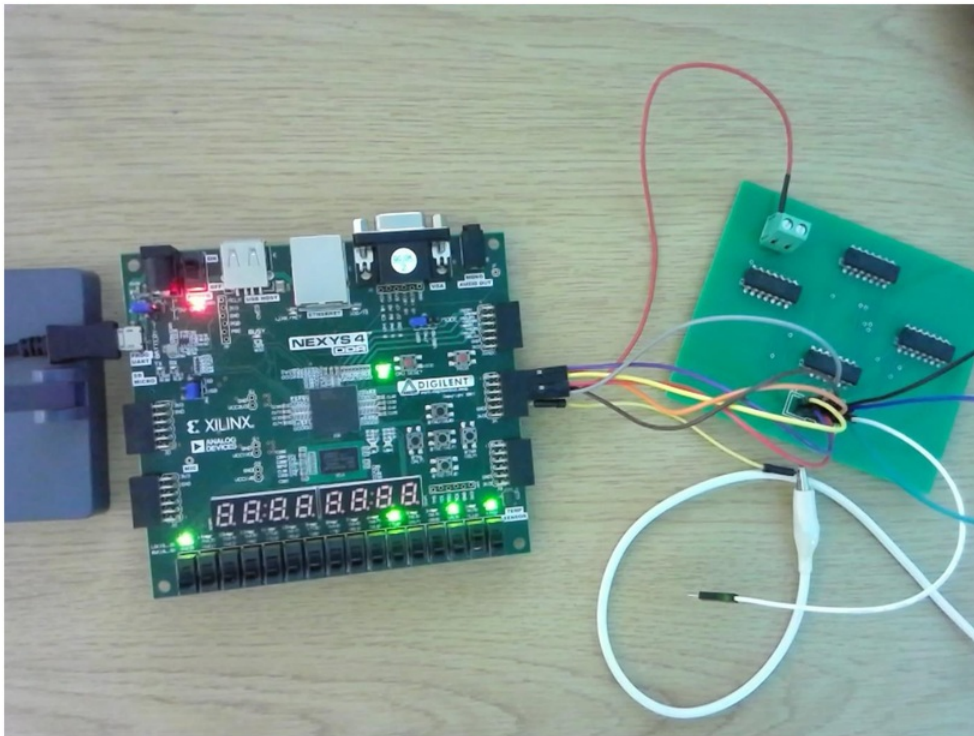


Figure 42: Power supply to PCB switched off to introduce a fault.

As portrayed in Figure 42, the fault LED has illuminated signifying a faulty circuit through a discrepancy between the two halves of the pseudo random pattern generator.

PCB has been designed and explained., The PCB cct has been tested using the ATE hybrid 6 bit CAR system.

7 – Conclusions and Recommendations

The assignment aim and objectives were identified in detail to ensure that all necessary requirements were fulfilled upon completion of this report.

Research was conducted into double-sided PCBs, soldering, SMT technology, SMT and PTH technology, automatic optical and X-ray inspection techniques along with ICT. This provided a solid foundation in which to build upon for the simulative and practical investigations.

Wolfram's principles were discussed in relation to CAR 90 and 150 individual circuits before combining the two to produce a greater maximum count in a 6-bit hybrid arrangement.

The CAR 90/150 6-Bit Hybrid concept was then designed, built and tested using MultiSim software with all results analysed to ensure the maximum count had been achieved.

Using Vivado software, the various different components that comprised the MultiSim circuit were designed, coded and brought together, using their IP circuit diagram forms, to produce a Xilinx board equivalent. This was then subsequently downloaded to the programmable board to produce one half of the pseudo random pattern generator. A virtually identical circuit was built physical on breadboard and connected to the Xilinx board to conduct further tests. Analysis was again conducted to ensure the breadboard circuit was acceptable to be applied to a manufactured to requirements PCB produced by a local company.

Using Eagle software, all the required files were generated in order for a manufacturer to fabricate the PCB ready for the breadboard components to be soldered in the college. Once returned, components were mounted and soldered before testing with the Xilinx board.

In conclusion, with all test results obtained and relevant analysis conducted, it can be confirmed that all simulated circuits and practically built circuits functioned correctly and efficiently. In every scenario, if a fault were introduced in one-half of the circuit the fault would always be registered and be visually flagged up to the user.

8 – References

Axiomtek, 2018. *Axiomtek Automated Optical Inspection*. [Online]

Available at:

[http://us.axiomtek.com/Default.aspx?MenuId=Solutions&FunctionId=SolutionView&ItemId=582&Title=AOI+\(Automatic+Optical+Inspection\)](http://us.axiomtek.com/Default.aspx?MenuId=Solutions&FunctionId=SolutionView&ItemId=582&Title=AOI+(Automatic+Optical+Inspection))

[Accessed 4th February 2018].

build-electronic-circuits, 2018. *build-electronic-circuits*. [Online]

Available at: <https://www.build-electronic-circuits.com/gerber-file/>

[Accessed 8th February 2018].

Landers, T. L. et al., 1994. *Electronics Manufacturing Processes*. 1st ed. Upper Saddle River: Prentice-Hall Inc.

Poole, I., 2016. *Electronics-notes*. [Online]

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

[Accessed 4th February 2018].

Riley, F., 1988. *The Electronics Assembly Handbook*. 1st ed. Berlin: Springer-Verlag Berlin Heidelberg.

SMTnet, 2018. *SMTnet*. [Online]

Available at:

http://www.smtnet.com/mart/index.cfm?fuseaction=view_item&item_id=80865

[Accessed 4th February 2018].

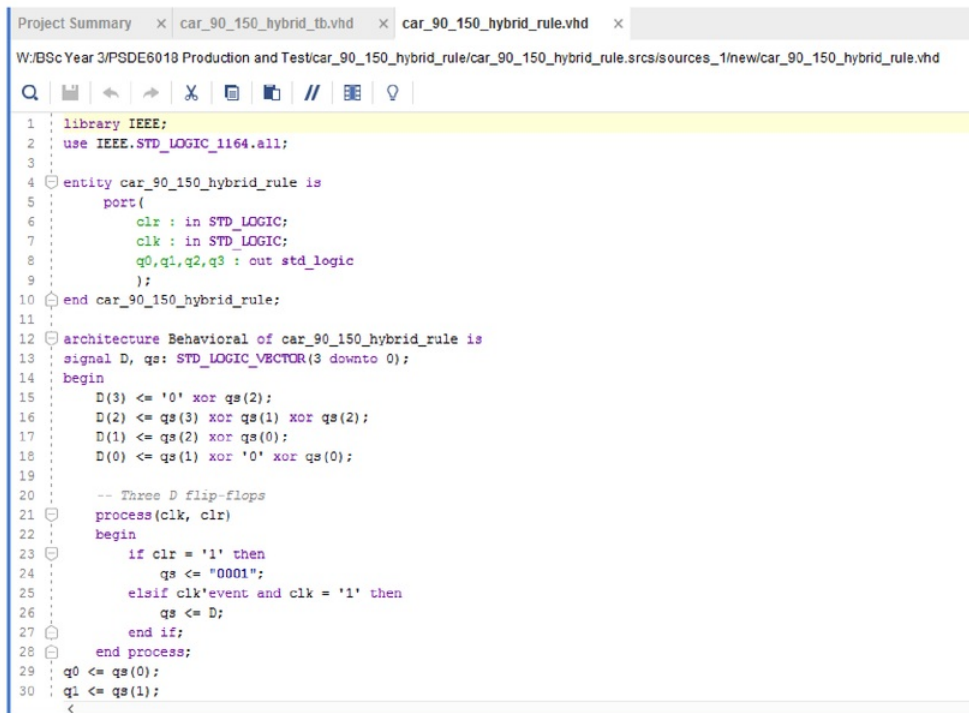
Storey, N., 2009. *Electronics: A Systems Approach*. 4th ed. Harlow: Pearson Prentice Hall.

Wakerly, J. F., 2006. *Digital Design: Principles and Practices*. 4th ed. Noida: Pearson Prentice Hall.

Good number of relevant sources.

9 – Appendix

Appendices 1 – CAR 90/150 Hybrid 4-Bit Code



```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.all;
3
4 entity car_90_150_hybrid_rule is
5     port(
6         clr : in STD_LOGIC;
7         clk : in STD_LOGIC;
8         q0,q1,q2,q3 : out std_logic
9     );
10 end car_90_150_hybrid_rule;
11
12 architecture Behavioral of car_90_150_hybrid_rule is
13     signal D, qs: STD_LOGIC_VECTOR(3 downto 0);
14     begin
15         D(3) <= '0' xor qs(2);
16         D(2) <= qs(3) xor qs(1) xor qs(2);
17         D(1) <= qs(2) xor qs(0);
18         D(0) <= qs(1) xor '0' xor qs(0);
19
20         -- Three D flip-flops
21         process(clk, clr)
22         begin
23             if clr = '1' then
24                 qs <= "0001";
25             elsif clk'event and clk = '1' then
26                 qs <= D;
27             end if;
28         end process;
29         q0 <= qs(0);
30         q1 <= qs(1);
31     <
```

Figure 43: CAR 90/150 hybrid 4-bit Vivado coding.

Appendices 2 – AND Gate Code and IP Circuit Diagram

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5
6  entity and_gate is
7  -- Port ( );
8  Port ( a : in std_logic;
9        b : in std_logic;
10       c : in std_logic;
11       d : in std_logic;
12       q : out std_logic);
13 end and_gate;
14
15 architecture Behavioral of and_gate is
16
17 begin
18   q <= a and b and c and d;
19
20 end Behavioral;
21
22
```

Figure 44: Four input AND gate Vivado code.

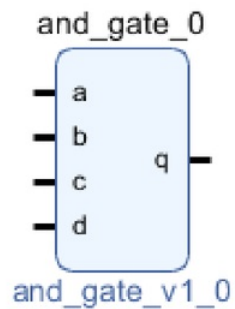


Figure 45: Four input AND gate IP circuit diagram.

Appendices 3 – CAR 90/150 6-Bit Hybrid Code and IP Circuit Diagram

```

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
16         D(5) <= '0' xor qs(4);
17         D(4) <= qs(5) xor qs(4) xor qs(3);
18         D(3) <= qs(4) xor qs(2);
19         D(2) <= qs(3) xor qs(2) xor qs(1);
20         D(1) <= qs(2) xor qs(0);
21         D(0) <= qs(1) xor qs(0) xor '0';
22
23         -- Three D flip-flops
24         process(clk, clr)
25         begin
26             if clr = '1' then
27                 qs <= "000001";
28             elsif clk'event and clk = '1' then
29                 qs <= D;
30             end if;
31         end process;
32         q0 <= qs(0);
33         q1 <= qs(1);
34         q2 <= qs(2);
35         q3 <= qs(3);
36         q4 <= qs(4);
37         q5 <= qs(5);
38     end Behavioral;
39
40
41

```

Figure 46: CAR 90/150 6-Bit Hybrid Vivado code.

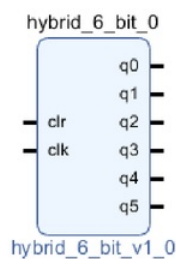


Figure 47: CAR 90/150 6-Bit Hybrid IP circuit diagram.

Appendices 4 – Inverter Gate Code and IP Circuit Diagram

```
Q [Icons]
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5
6  entity inverter_gate is
7  -- Port ( );
8  Port ( a : in std_logic;
9        q : out std_logic);
10 end inverter_gate;
11 architecture Behavioral of inverter_gate is
12
13 begin
14   q <= not a;
15
16 end Behavioral;
17
18
```

Figure 48: Inverter Vivado code.

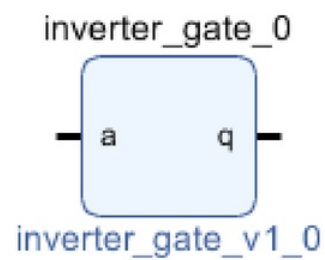


Figure 49: Inverter IP circuit diagram.

Appendices 5 – Clock Code and IP Circuit Diagram

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4  use IEEE.std_logic_unsigned.all;
5
6  entity FiveKHz_Clock is
7
8      Port ( CLK100MHZ : in    STD_LOGIC;
9            Rst       : in    std_logic;
10           FiveKHz   : out   STD_LOGIC
11
12       );
13
14 end FiveKHz_Clock;
15
16 architecture Behavioral of FiveKHz_Clock is
17
18     signal cnt1      : integer := 0;
19
20     signal temp1     : std_logic := '0';
21
22
23 begin
24
25
26     process(CLK100MHZ,Rst)
27     begin
28         if Rst = '1' then
29             temp1 <= '0';
30             cnt1 <= 0;
31         elsif rising_edge (CLK100MHZ) then
32             if(cnt1= 50000000) then
33                 temp1 <= not temp1;
34                 cnt1 <= 0;
35             else
36                 cnt1 <= cnt1 +1 ;
37             end if;
38         end if;
39
40     end process;
41     FiveKHz <= temp1;
42
43 end Behavioral;

```

Figure 50: Clock Vivado code.

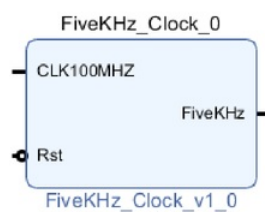


Figure 51: Clock IP circuit diagram.

Appendices 6 – OR Gate Code and IP Circuit Design

```
Q [Icons]
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5
6  entity or_gate is
7  -- Port ( );
8  Port ( a : in std_logic;
9        b : in std_logic;
10       q : out std_logic);
11 end or_gate;
12 architecture Behavioral of or_gate is
13
14 begin
15   q <= a or b;
16
17 end Behavioral;
18
19
```

Figure 52: OR gate Vivado code.

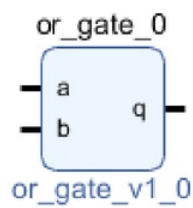


Figure 53: OR gate IP circuit diagram.

Appendices 7 – XOR Gate Code and IP Circuit Diagram

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5
6  entity xor_gate is
7  -- Port ( );
8  Port ( a : in std_logic;
9        b : in std_logic;
10       q : out std_logic);
11 end xor_gate;
12
13 architecture Behavioral of xor_gate is
14
15 begin
16   q <= a xor b;
17
18 end Behavioral;
19
```

Figure 54: XOR gate Vivado code.

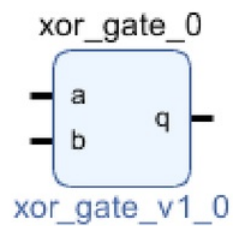
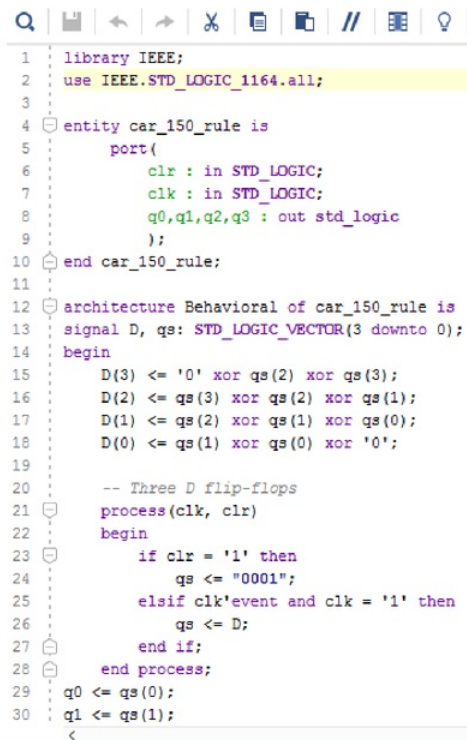


Figure 55: XOR gate IP circuit diagram.

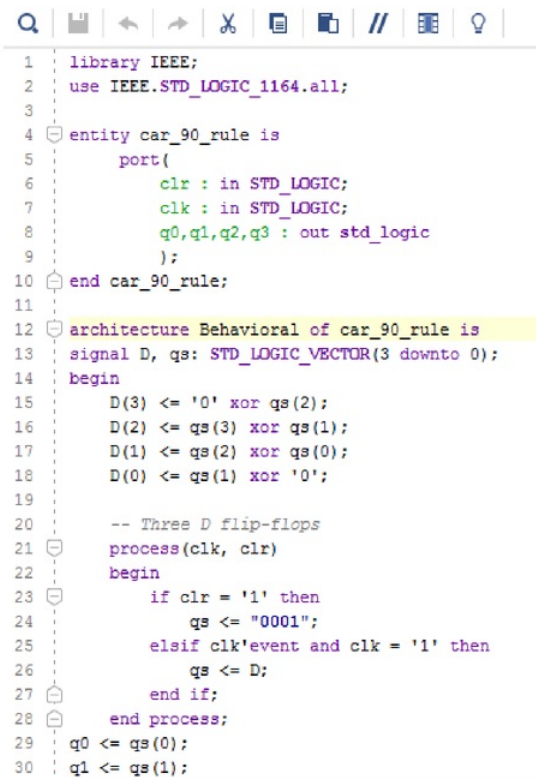
Appendices 8 – CAR 150 Code



```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.all;
3
4 entity car_150_rule is
5     port(
6         clr : in STD_LOGIC;
7         clk : in STD_LOGIC;
8         q0,q1,q2,q3 : out std_logic
9     );
10 end car_150_rule;
11
12 architecture Behavioral of car_150_rule is
13     signal D, qs: STD_LOGIC_VECTOR(3 downto 0);
14     begin
15         D(3) <= '0' xor qs(2) xor qs(3);
16         D(2) <= qs(3) xor qs(2) xor qs(1);
17         D(1) <= qs(2) xor qs(1) xor qs(0);
18         D(0) <= qs(1) xor qs(0) xor '0';
19
20         -- Three D flip-flops
21         process(clk, clr)
22         begin
23             if clr = '1' then
24                 qs <= "0001";
25             elsif clk'event and clk = '1' then
26                 qs <= D;
27             end if;
28         end process;
29         q0 <= qs(0);
30         q1 <= qs(1);
31     <
```

Figure 56: CAR 90 Vivado code.

Appendices 9 – CAR 90 Code



```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3
4  entity car_90_rule is
5      port(
6          clr : in STD_LOGIC;
7          clk : in STD_LOGIC;
8          q0,q1,q2,q3 : out std_logic
9      );
10 end car_90_rule;
11
12 architecture Behavioral of car_90_rule is
13     signal D, qs: STD_LOGIC_VECTOR(3 downto 0);
14     begin
15         D(3) <= '0' xor qs(2);
16         D(2) <= qs(3) xor qs(1);
17         D(1) <= qs(2) xor qs(0);
18         D(0) <= qs(1) xor '0';
19
20         -- Three D flip-flops
21         process(clk, clr)
22         begin
23             if clr = '1' then
24                 qs <= "0001";
25             elsif clk'event and clk = '1' then
26                 qs <= D;
27             end if;
28         end process;
29         q0 <= qs(0);
30         q1 <= qs(1);
```

Figure 57: CAR 90 Vivado code.

*Appendix does
demonstrate the prototype
design and development
carried out.*

PSDE6018 Assignment 1 Dan Lee

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

Text Comment. Introduction is clear.

PAGE 9

PAGE 10

Text Comment. Good comparison and diagrams.

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

Text Comment. Good explanation and coverage of ATE types used in modern production industry.

PAGE 19

Text Comment. This is a logical explanation of CAR rules 90 and 150 operation.

PAGE 20

Text Comment. Good test of hybrid CAR. The preliminary prototypes and tests are missing and should be included. This Hybrid is a derivative of the prototypes. The CUT is connection here is correct.

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

Text Comment. The test and fault analysis is good and shows the 3 different fault possibilities.

PAGE 26

PAGE 27

PAGE 28

Text Comment. The prototype design and test looks correct. This demonstrates the hybrid rule using a 4 bit system.

PAGE 29

Text Comment. Good full count test of 6 bit CAR

PAGE 30

PAGE 31

Text Comment. The overall design is correct and the test are correct.

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

Text Comment. Practical testing with evidence is good.

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

Text Comment. PCB has been designed and explained., The PCB cct has been tested using the ATE hybrid 6 bit CAR system.

PAGE 44

Text Comment. Conclusion has good points.

PAGE 45

Text Comment. Good number of relevant sources.

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

Text Comment. Appendix does demonstrate the prototype design and development carried out.

DANIEL LEE**0****RESEARCH-INFORMED LITERATURE**

SMT research is reasonable. There are diagrams to aid the explanations. 3/5 The technologies are covered but no solder process techniques are covered. 2.5/5 The ATE types used in production industry is covered in good detail. 8.5/10 Good. Could have pointed out virtual ATE integration.

14/20**KNOWLEDGE AND UNDERSTANDING**

Design of 6 bit CAR has some prototypes missing in the CAD design process. 3/7 Design of PCB and explanation of the manufacture is good. 5/7 The FPGA 6 bit CAR design process is explained and the prototype stages are included. 4/7 Reasonable. Some minor errors.

12/20**ANALYSIS**

The test results and analysis of the 6 bit CAR lack the preliminary prototype stages which would be more effective for comparison to theory. 4.5/10 The FPGA prototype stages could have been more logically explained and arranged more orderly. PCB analysis is much better in comparison. 6.5/10 Some explanations missing. Tables of comparison would have helped.

11/20

PRACTICAL APPLICATION AND DEP.



Manufacture process of PCB is explained well. 10/15 The FPGA prototyping could have been explained. 4/10 Some missing explanations for comparison.

14/25

SKILLS FOR PROFESSIONAL PRAC.



Use of software was logical. 3.5/5 Report could have better arranged in a more logical manner. 2.5/5 Conclusion makes some good points. 3/5 Needed more content and discussion of virtual ATE integration.

9/15