

James Lynch - Assignment 2

by James Lynch

FILE	11008_JAMES_LYNCH_JAMES_LYNCH_-_ASSIGNMENT_2_152847_729165938.DOCX		
TIME SUBMITTED	30-APR-2018 04:24PM (UTC+0100)	WORD COUNT	2118
SUBMISSION ID	87923685	CHARACTER COUNT	13945

HNC Electrical & Electronic Engineering
Module PSDE4004; Electronics 1

Assignment 2
Digital Circuit Design

Name: James Lynch

Date: 30/04/2018

Word Count: 2118

Contents

Introduction	3
Task 1 – Background Research.....	3
Part 1: The Differences between Sequential and Combinational Logic	3
Part 2: Finite State Machines	4
Part 3: Benefits of FPGA's and ASIC's.....	7
Task 2 – Designing the Circuit	8
Task 3 – Circuit Design Simulation	9
Task 4 –Evaluation	12
Bibliography	18
Figure 1	3
Figure 2	3
Figure 3	4
Figure 4	5
Figure 5	5
Figure 6	6
Figure 7	6
Figure 8	9
Figure 9	10
Figure 10	11
Figure 11	13
Figure 12	14
Figure 13	15
Figure 14	16
Figure 15	17
Table 1: Differences	4
Table 2: Advantages and Disadvantages.....	7
Table 3: State Table 0-9	8
Table 4: State Table 0-5	8
Table 5: Test table.....	12

Introduction

In this assignment, I will be looking into the digital side of electronics. I have been tasked with the designing and testing of a digital stopwatch on two seven segments displays. This will be designed using a finite state machine and accomplished on a FPGA. After this, the circuit will be simulated on multisim in PLD mode. From there, the circuit will be downloaded onto a Diligent Nexys 4 test board to appropriately test if the design meets the specification.

Task 1 – Background Research

Part 1: The Differences between Sequential and Combinational Logic

Combinational and Sequential logic are the basis of most digital electronics, forming a foundation of knowledge for the subject. Combinational logic is the implementation of Boolean circuits to find an output where it is a pure function of the present input, whereas sequential logic's output depends on the sequence of past inputs as well as on the present value. (Dewansh, 2015)

Figure 1 and Figure 2 show the basic concept of each circuit with the differences between them found in Table 1.

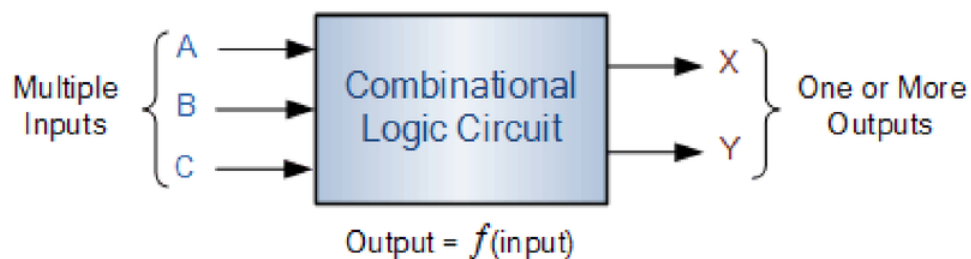


Figure 1 - Combinational Logic: https://www.electronics-tutorials.ws/combinational/comb_1.html

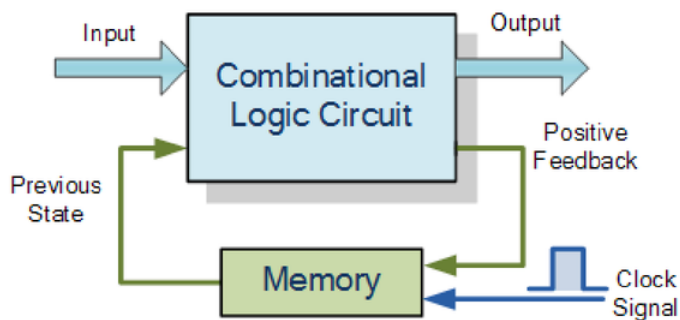


Figure 2 - Sequential Logic: https://www.electronics-tutorials.ws/sequential/seq_1.html

Table 1: Differences

Combinational	Sequential
The output is a function of the present inputs.	The output is a function of a clock, presenting inputs and the previous states of the system.
Does not have the ability to store data (State).	Has the memory to store present states that will be sent as a control input to start the next operation.
Doesn't require feedback, simply outputting the input according to the designed logic.	Involves feedback from the output to the input to be stored in the memory for the next operation.
Used in Arithmetic and Boolean operations.	Used for storing data and thus is used in (for example) RAM.
Logic gates are used for the building blocks.	Flip flops are used for the building blocks.
Independent of a clock and thus is able to operate without a trigger.	Uses a clock function, being able to operate on a triggered electronic pulse.

Part 2: Finite State Machines

(AllAboutCircuits.com, n.d.) The mathematical term "Finite State Machine" describes a diagram that shows the operation of a sequential circuit, also known as a State Diagram. A state Diagram is made up of circles and arrows to present the operation of a sequential circuit. Figure 3 shows an example of what a state diagram looks like, this being the Moore Finite State Machine.

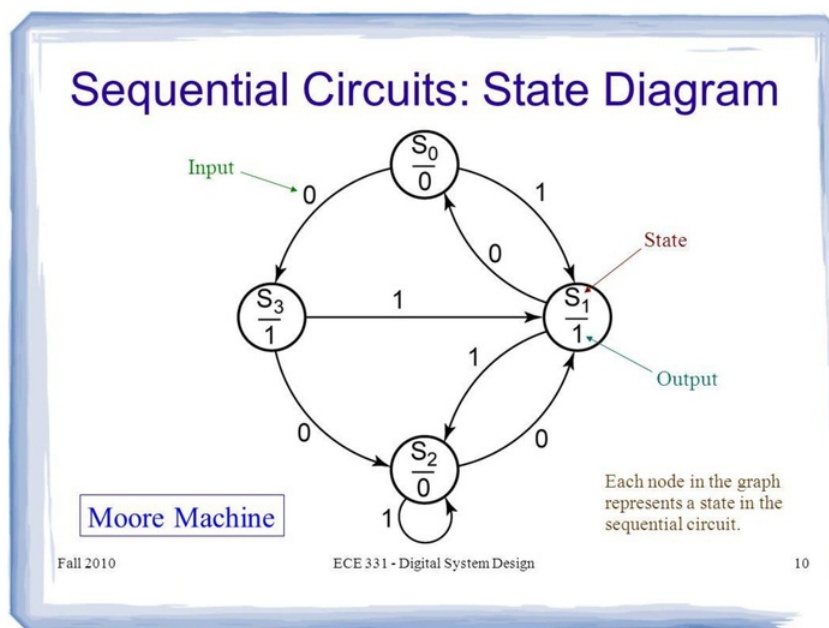


Figure 3 - State Diagram: http://images.slideplayer.com/16/4913292/slides/slide_10.jpg

Each circle represents a "state" – this being a condition that the machine can be found at – and each arrow corresponds to a "transition" from one state to the next, this happening once every clock cycle. The number in the middle of each arrow represents the current state we are in.

From here, we can now implement each state as a binary. State 0 is equal to being the binary 00, state 1 = 01, state 2 = 10 and state 3 is equal to 11. After this, we can organise a table in which each of the binary codes can be organised into – this being called a “State Table” (as seen in Figure 4).

Current State		Input I	Next State		Outputs Y
A	B		A _{next}	B _{next}	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	X	X	X
1	1	1	X	X	X

Figure 4 - State Table: <https://sub.allaboutcircuits.com/images/44003.png>

The state table is just as reliable as the state diagram for describing the behaviour of a sequential circuit.

Finally implementing the theoretical “Machine” into a circuit can be done through a number of means, however flip flops are the most often used. For example, if implemented with D-flip flops we would need the same amount of state columns. Every flip flop added another state column must be implemented. The column that correlates to each flip flop shows the input given to the flip flop in order to move from its current state to the next state – this being easy for the D-flip flop as the necessary input is equal to the next state (the rows that contain X’s count as do not care and we can use them to decide whether or not they are a binary 1 or binary 0). Looking at Figure 5, we can see what this looks like.

Current State		Input I	Next State		Outputs Y	Flip Flop Inputs	
A	B		A _{next}	B _{next}		D _A	D _B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	1	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

Figure 5 - D-flip flop state table: <https://sub.allaboutcircuits.com/images/44004.png>

These binary states can then be put into a Karnaugh map in order to find the Boolean expression. To do this, the binary 1’s are grouped together in groups of 2, 4 or 8 and then brought together for the expression. For each of the table above, see Figure 6.

		DA			
		00	01	11	10
A	BI	0	0	1	0
		0	1	X	X

		DB			
		00	01	11	10
A	BI	0	1	0	0
		0	0	X	X

Figure 6 - Karnaugh maps: <https://sub.allaboutcircuits.com/images/44006.png>

The expressions for these being: $D_A = A.I + B.I = (A + B).I$

$$D_B = \overline{A}.B.I$$

After this, take the Boolean expressions and put them into gate functions on a circuit, forming the full foundation of a clocked circuit of sequential logic. Figure 7 shows this as a complete circuit.

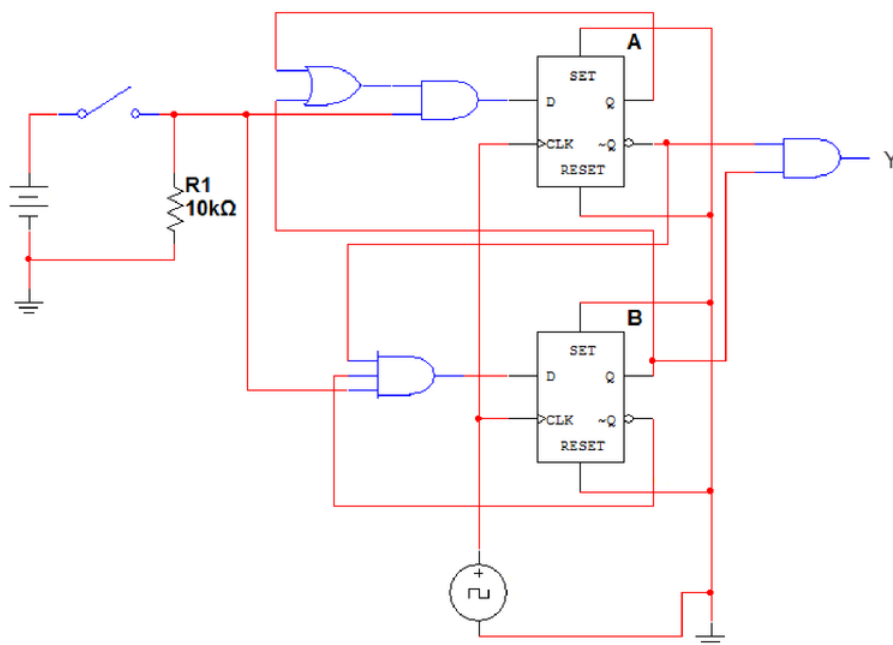


Figure 7 - Complete D-Flip Flop Sequential Circuit: <https://sub.allaboutcircuits.com/images/44010.png>

Part 3: Benefits of FPGA's and ASIC's

Looking at the PDF found from (asic-blogspot.in, 2013), we can see that the advantages and disadvantages have been well summarised for both FPGA's and ASIC's. (See Table 2).

Table 2: Advantages and Disadvantages.

FPGA Advantages	ASIC Advantages	FPGA Disadvantages	ASIC Disadvantages
Is quicker to get to market due to being a readymade design.	For high volume designs, costs are cheap due to less implementation.	Increased power consumption, no control over power.	Slower to market as it can take a year or more to design.
No NRE (Non Recurring Expenses): Due to FPGA's tools being cheap it's easier to get a hold of.	Great design flexibility and speed optimisation.	Locked with the resources of the FPGA, limited in its design size.	Some design issues – dealing with checks and signal issues if not taken care of appropriately.
Simple design cycle: Due to the system software dealing with the routing, placement and timing, manual intervention is lessened.	Can be optimised for low power usage and good battery life.	As quantity increases, cost per product increases also.	Very expensive tools required for ASIC and ASIC products.
More predictable project cycle: The design flow eliminates potential problems due to the design logic being already synthesised and verified in the FPGA.	Can be implemented into an analogue circuit with mixed signal designs.		
Field Reprogram ability: A new bit stream can be uploaded instantly and remotely from the device.	ASIC can perform a Design for Test function.		
Reusable: Its main advantage, new designs can be implemented onto the FPGA to reprogram it.			
Good for prototyping and limited production.			
Used for lower speed, complexity and volume designs.			
Special hardware's to add a better performance onto the whole deal.			
Synthesis is easy.			
No floor panel needed, tool is efficient.			

Also to note is that (Joan, 2009) summarised this nicely into 5 different parts:

1. An ASIC can be described as a unique type of IC meant for specific applications while an FPGA is reprogrammable.
2. An ASIC can no longer be altered once made.
3. It is normal to design and test on an FPGA before being applied onto an ASIC.
4. ASIC's waste very little materials when compared to an FPGA, plus the recurring costs are low.
5. FPGA's are better when building when measured against an ASIC.

Task 2 – Designing the Circuit

In this part, I will be designing a digital stopwatch, counting up from 0 to 59, displaying the numbers on two seven segment displays that will be implemented onto a FPGA using a finite state machine. To do this, I must first look at each of the states of the seven segment displays (as seen in Table 3 and Table 4).

Table 3: State Table 0-9

CurrentState	NextState	Current State				Next State			
		Q a	Q b	Q c	Q d	Q a+	Q b+	Q c+	Q d+
State0	State1	0	0	0	0	0	0	0	1
State1	State2	0	0	0	1	0	0	1	0
State2	State3	0	0	1	0	0	0	1	1
State3	State4	0	0	1	1	0	1	0	0
State4	State5	0	1	0	0	0	1	0	1
State5	State6	0	1	0	1	0	1	1	0
State6	State7	0	1	1	0	0	1	1	1
State7	State8	0	1	1	1	1	0	0	0
State8	State9	1	0	0	0	1	0	0	1
State9	State0	1	0	0	1	0	0	0	0
State10	X	X	X	X	X	X	X	X	X
State11	X	X	X	X	X	X	X	X	X
State12	X	X	X	X	X	X	X	X	X
State13	X	X	X	X	X	X	X	X	X
State14	X	X	X	X	X	X	X	X	X
State15	X	X	X	X	X	X	X	X	X

Table 4: State Table 0-5

CurrentState	NextState	Current State			Next State		
		Q a	Q b	Q c	Q a+	Q b+	Q c+
State0	State1	0	0	0	0	0	1
State1	State2	0	0	1	0	1	0
State2	State3	0	1	0	0	1	1
State3	State4	0	1	1	1	0	0
State4	State5	1	0	0	1	0	1
State5	State0	1	0	1	0	0	0
State6	X	X	X	X	X	X	X
State7	X	X	X	X	X	X	X

Following on from this, we can make the Karnaugh Maps and the Boolean expression for both. These can be seen Appendix A and B (Figure 14 and Figure 15).

Task 3 – Circuit Design Simulation

Using the knowledge gained from the previous task, we can now set up to simulate the circuits in multisim.

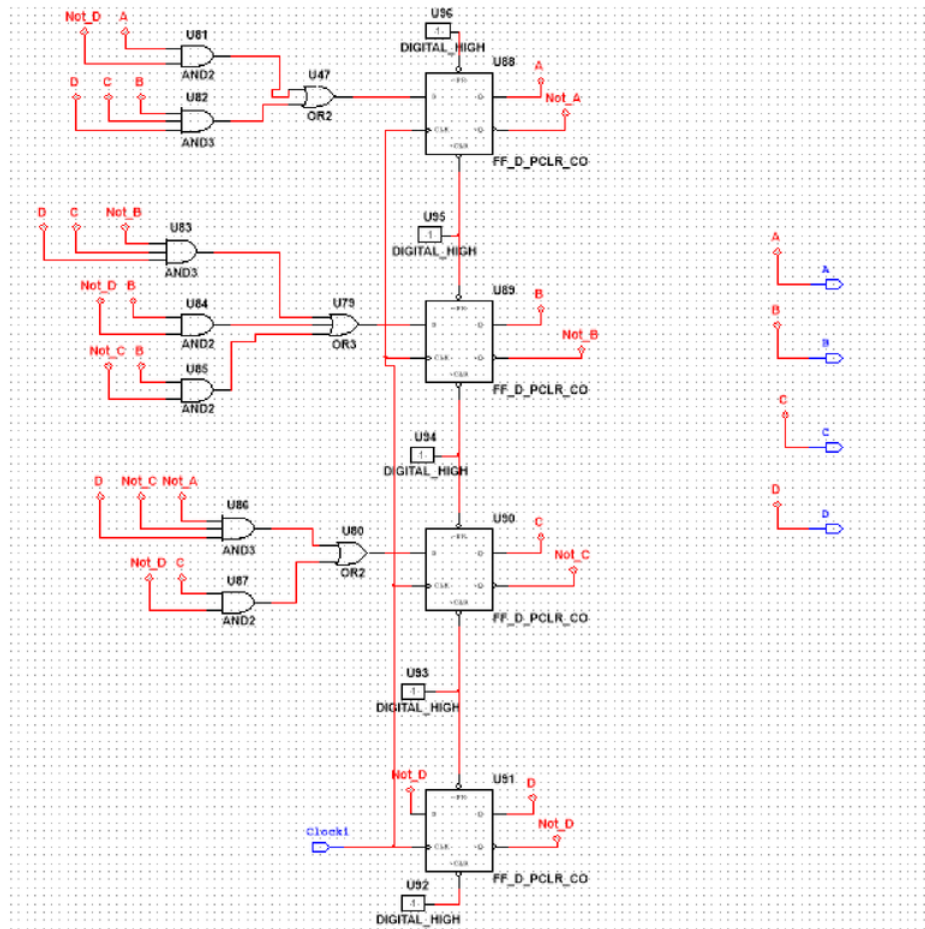


Figure 8: 0-9 counter PLD

Figure 8 shows the circuit design for the 0-9 counter. Using flip flops, I was able to create the output required from the Boolean expressions. Figure 9 also shows this, but for the 0-5 counter.

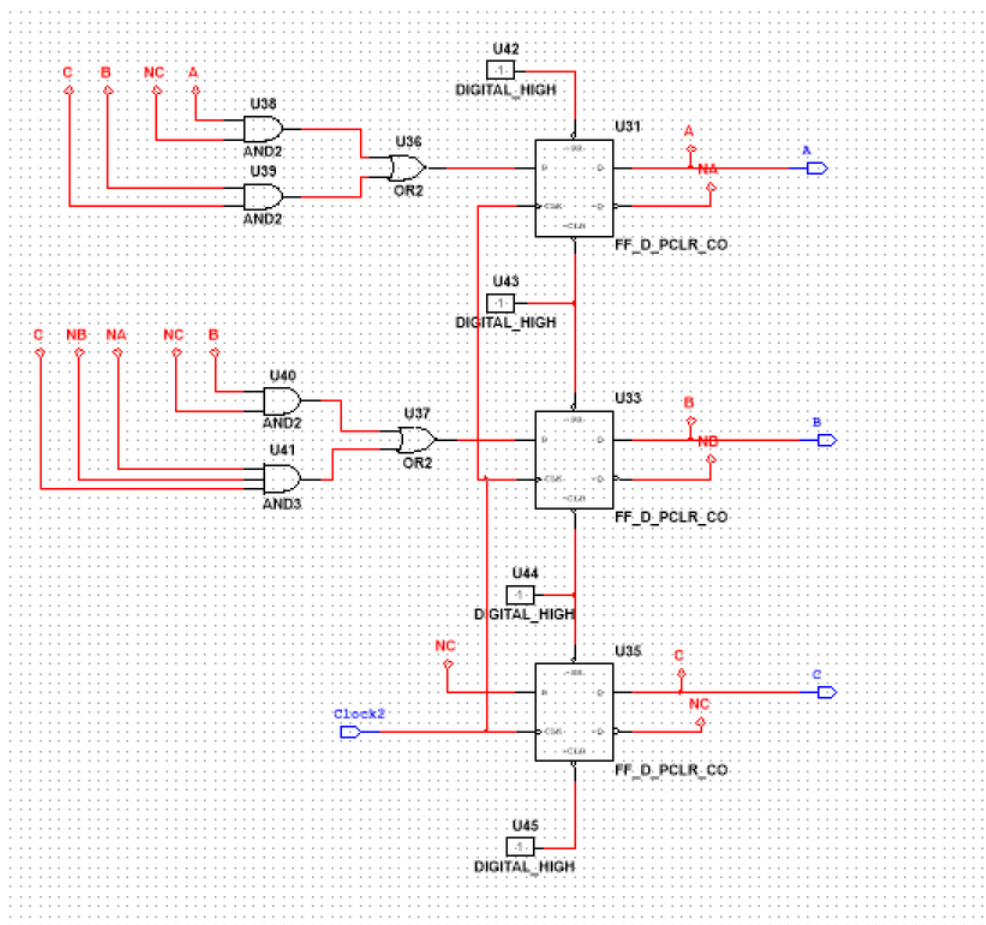


Figure 9: 0-5 counter PLD

Task 4 –Evaluation

Downloading this full design onto the Digital Nexys 4 test board, tests were conducted to see how well the simulation worked compared to specification (see Table 5)

Table 5: Test table

Number	Description	Expected Result	Actual Result
1	Power On	Count Starts	Count Started from 0-9 (see Figure 11, Figure 12 & Figure 13)
2	Stop and Start	Functions as it should, stopping and starting the count.	Unable to stop the count.
3	Implemented on FPGA.	Tested on Digital Nexys (downloaded)	Was tested on the device and functions.

Looking at these results, though there aren't many tests, we can see that the circuit worked towards the specification. This is due to the fact that it was able to produce a seven segment display that counted up from 0 to 59.

However the design failed to match the stop start function of the specification. Without this, the design will count up over and over until it is powered down.

To conclude, the design does work as a digital clock, counting from 0 to 59 and meeting specification in that regard. To improve on this, I must implement a stop start system into the design as well in order to further meet the specification.

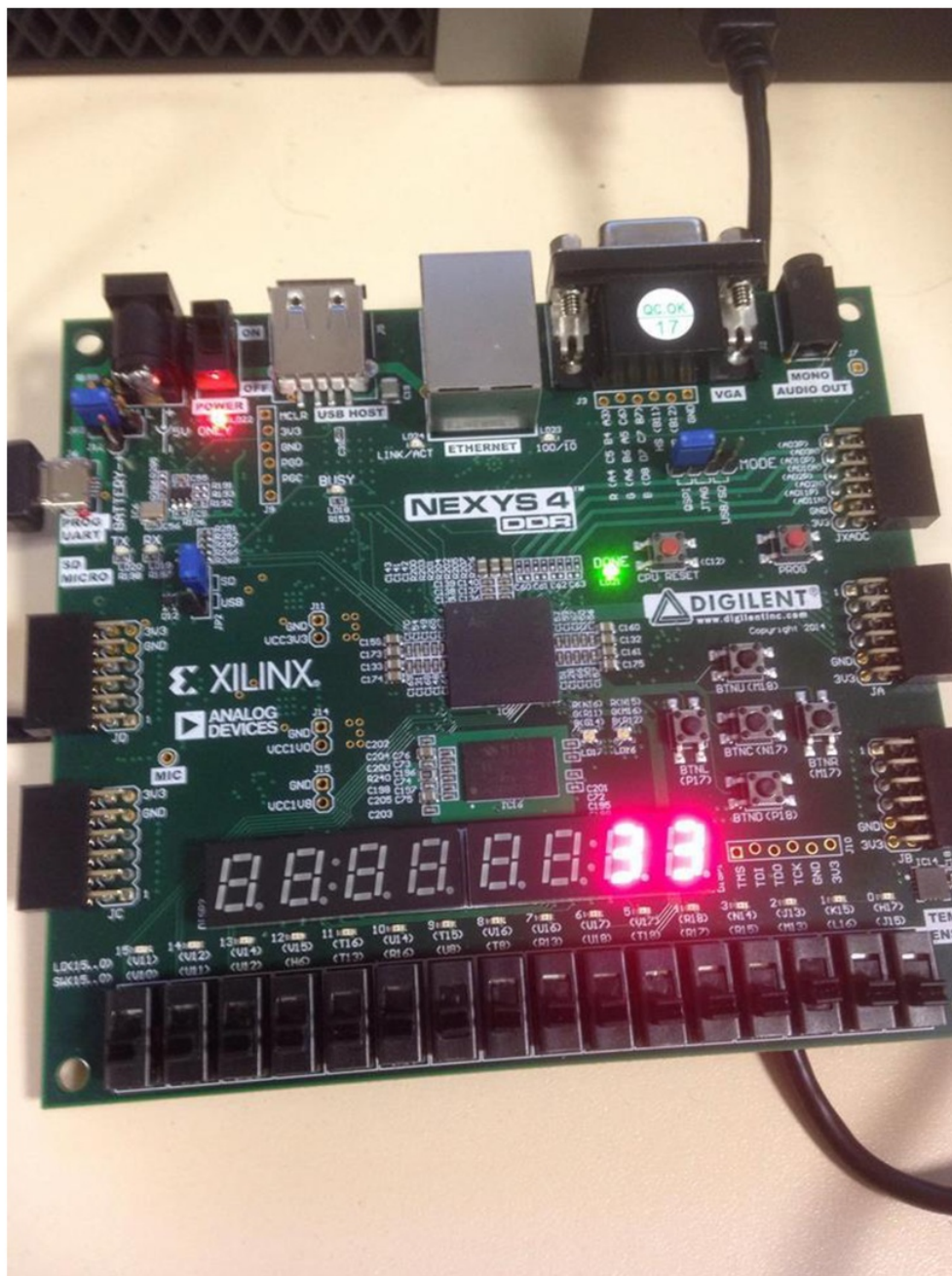


Figure 11: Segment Display Test 1

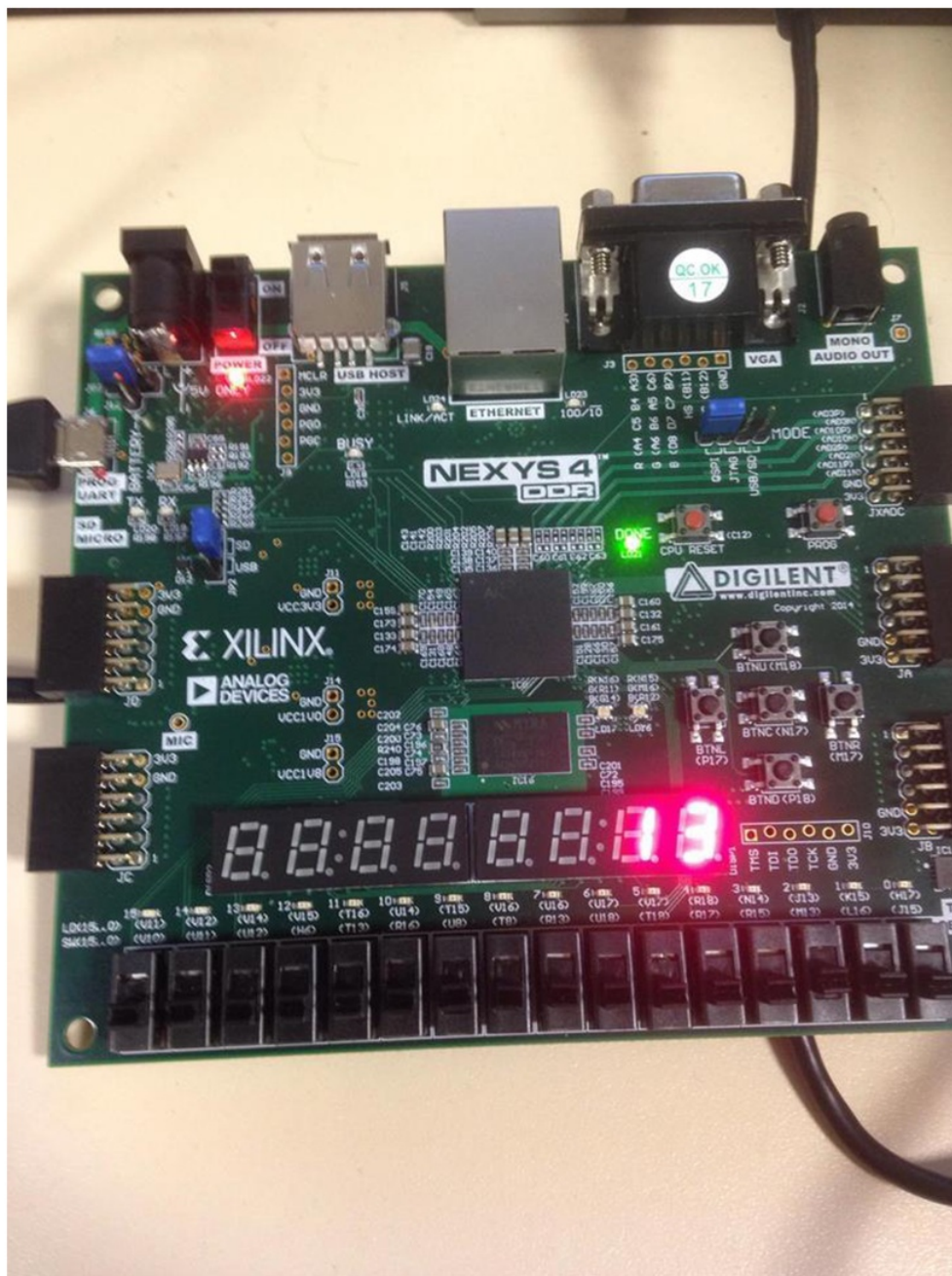


Figure 12: Segment Display Test 2

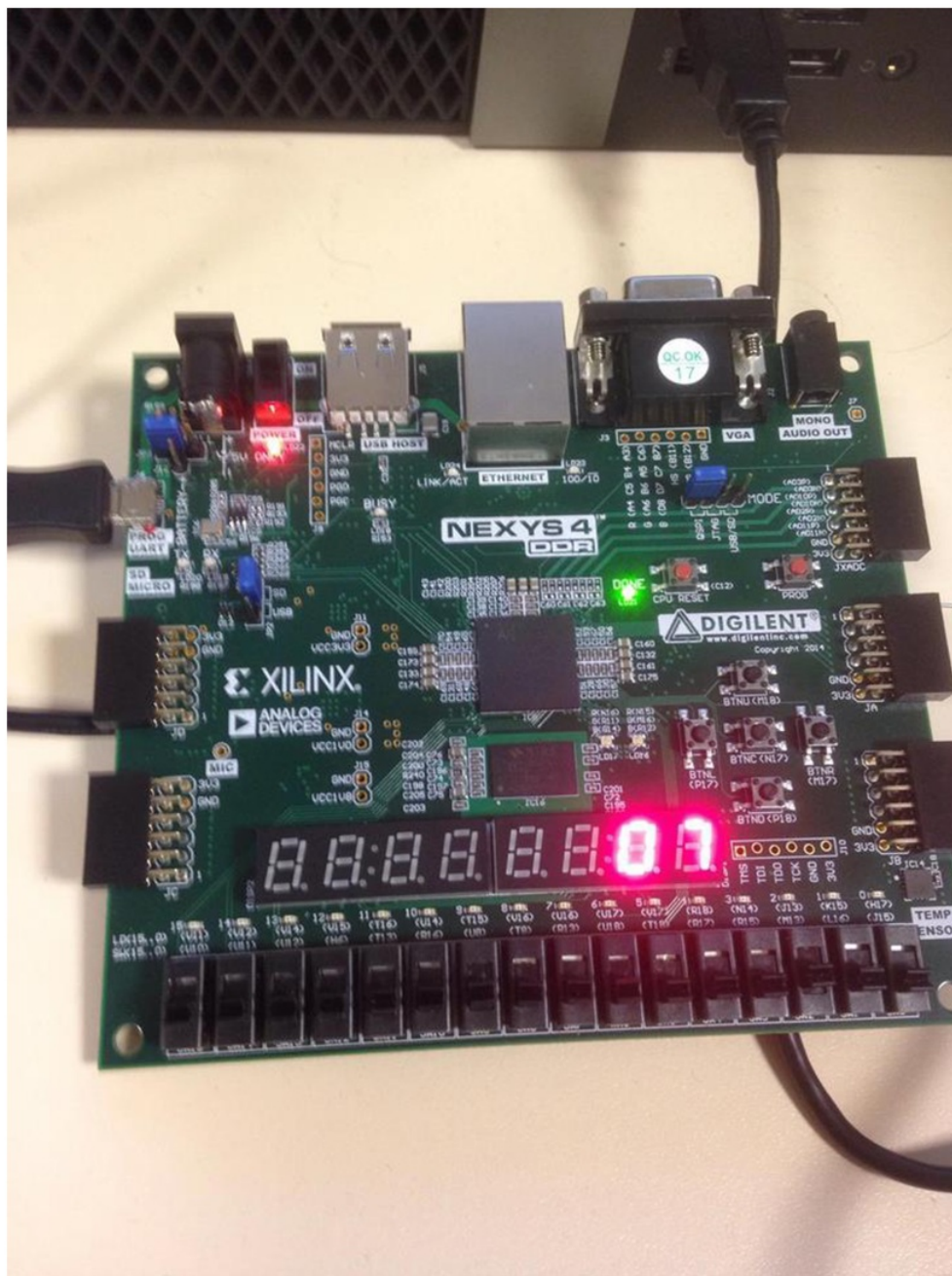


Figure 13: Segment Display Test 3

0-9 count Appendix A
 $QA' = A\bar{D} + BCD$

$$QD' = \bar{B}CD + B\bar{D} + B\bar{C}$$

$AB \backslash CD$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	X	X
10	1	0	X	X

$$QC' = A\bar{C}D + C\bar{D}$$

$AB \backslash CD$	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	1	X

$$QD' = \bar{D}$$

	00	01	11	10
00	0	1	0	1
01	0	0	0	1
11	X	X	X	X
10	0	0	X	X

	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

Figure 14: Appendix A, 0-9 Counter Karnaugh Maps

Appendix B: 0-5 Count

$$QA^+ = A\bar{C} + BC$$

A \ BC	00	01	11	10
00	0	0	1	0
01	1	0	x	x
11	x	x	x	x
10	x	x	x	x

$$QB^+ = B\bar{C} + \bar{A}\bar{B}C$$

A \ BC	00	01	11	10
00	0	1	0	1
01	0	0	x	x
11	x	x	x	x
10	x	x	x	x

$$QC^+ = \bar{C}$$

A \ BC	00	01	11	10
00	1	0	0	1
01	1	0	x	x
11	x	x	x	x
10	x	x	x	x

	00	01	11	10
00				
01				
11				
10				

Figure 15: Appendix B, 0-5 Counter Karnaugh Maps

Bibliography

- AllAboutCircuits.com. (n.d.). *Finite State Machines*. Retrieved from allaboutcircuits.com:
<https://www.allaboutcircuits.com/textbook/digital/chpt-11/finite-state-machines/>
- asic-blogspot.in. (2013). *FPGA-ASIC Design Advantages-Disadvantages*. Retrieved from
http://www.idc-online.com/technical_references/pdfs/electronic_engineering/FPGA_ASIC_Design_Advantages_Disadvantages.pdf
- Dewansh. (2015, October 27th). *Difference between Combinational and Sequential logic circuits*. Retrieved from VLSFacts.com: <http://www.vlsifacts.com/difference-combinational-sequential-logic-circuits/>
- Joan, B. (2009, December 9). *Difference Between ASIC and FPGA*. Retrieved from DifferenceBetween.net: <http://www.differencebetween.net/technology/difference-between-asic-and-fpga/>

James Lynch - Assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

40/100

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE



There was evidence of further reading and research with the inclusion of a reference list and appropriate in text referencing. Your interpretation and understanding of the additional research was good. Your references were appropriate to the task, however your reference list was a little short for this assignment.

5

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally satisfactory. However the report lacked a lot of the required detail and there were many gaps in your knowledge of the basic principles. Additionally, your understanding of the higher level concepts was not shown. Further work - There were a few omissions and misunderstandings throughout the task, which could be read up on for future reference. Explanations need to be written in much more detail and written in a more logical and clearer way.

13

ANALYSIS



Your analysis of the task was brief and lacked a lot of the required detail. You have demonstrated reasonable analytical skills and your approach to the task was logical. You carried out a reasonable evaluation of the results, but the conclusions were

5

very brief.

PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally good. You used the resources available in MultiSim appropriately when designing the circuit and you used appropriate options for testing and evaluating the circuit performance. You managed to design a basic circuit that functioned within the specification and achieved this by only asking for assistance when appropriate. Your results were good and allowed for a reasonable conclusion to be attempted.

12

SKILLS FOR PROFESSIONAL PRAC.



In this assignment you have demonstrated skills in working individually and also collaboratively. Your written report was generally good and contained most of the required detail, it was also written in a coherent and logical way. Try not to get too reliant on handwritten annexes though.

5