

1302736

by Jason Booth

FILE	19735_JASON_BOOTH_1302736_179157_96657460.DOCX (1.25M)		
TIME SUBMITTED	03-MAY-2021 09:37PM (UTC+0100)	WORD COUNT	2976
SUBMISSION ID	151477647	CHARACTER COUNT	15948

Course: HNC Electrical and Electronic Engineering

Module: SGEE4003 – Electronics 1

Assignment: 2. Digital Circuit Design

Name: Jason Booth

University number: 1302736

Date submitted: 03/05/2021

Word count: 2568

80-85

- ① EXCELLENT RESEARCH AND REFERENCE LIST, BUT YOUR IN TEXT CITATIONS DO NOT ALWAYS USE HARVARD.
- ② DESIGN WAS EXPLAINED IN DETAIL AND WELL PRESENTED.
- ③ TESTING WAS COMPREHENSIVE AND THE RESULTS WERE ANALYSED IN DETAIL.
- ④ REASONABLE CONCLUSIONS.

Contents

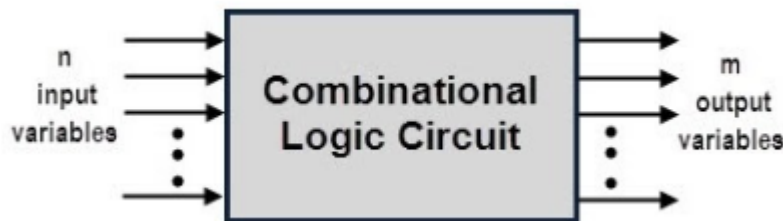
Background Research.....	1
Differences between combinational and sequential logic	1
Use of a Finite State Machine as a design tool	3
Mealy State Machine	4
Moore State Machine	5
Relative Merits of FPGA's and ASIC's.....	6
Design Method.....	8
7-Segment Display	8
Truth Table.....	8
Karnaugh Maps & Boolean Equations	9
7-Segment Display Circuits	11
0-5 Counter Design	13
State Diagram.....	13
State allocation	13
State Table	14
Karnaugh Maps	14
Circuit	15
0-9 Decade Counter Design	15
State Allocation.....	16
State Diagram.....	16
State Table	16
Karnaugh Maps	17
0-9 Counter Circuit.....	18
Simulation	19
Simulating 7-Segment display.....	19
Simulating the 0-5 Counter	20
Simulating 0-9 Decade counter.....	21
Simulating final design	22
Conclusion.....	22
References	23
Appendix	24

INTRODUCTION

Background Research

Differences between combinational and sequential logic

The major difference between combinational and sequential logic circuit is that the combinational logic circuit consists of only logic gates while the sequential logic circuits consist of logic gates and memory elements (Eletronics Coach, 2021). Combinational circuits consist of two or more logic gates to produce a required function.



USE HARVARD FOR
DIAGRAMS AS WELL

Figure 1 - <https://www.electronicshub.org/introduction-to-combinational-logic-circuits/>

The logic gates in the combinational circuit accepts the inputs and depending on the type of function of the logic gate (i.e. AND gate, OR gate, NOR gate, etc), produces an output.

(Administrator, 2015) states that, 'Combinational Logic and Sequential Logic are the building blocks of Digital System Design. Combinational Circuits include Multiplexers, Demultiplexers, Encoders, Decoders, etc. whereas Sequential Circuits are Latches, Flip-flops, Counters, Registers etc.'

Combinational circuits are widely used in a variety of applications such as computers, industrial processing, digital processing, digital communications, calculators, digital measuring techniques, automatic control of machines, etc. (2015)

Combinational logic can be put into mainly three types of classification which are named arithmetic and logic circuits, data transmission circuits and code converter circuits. (2015)

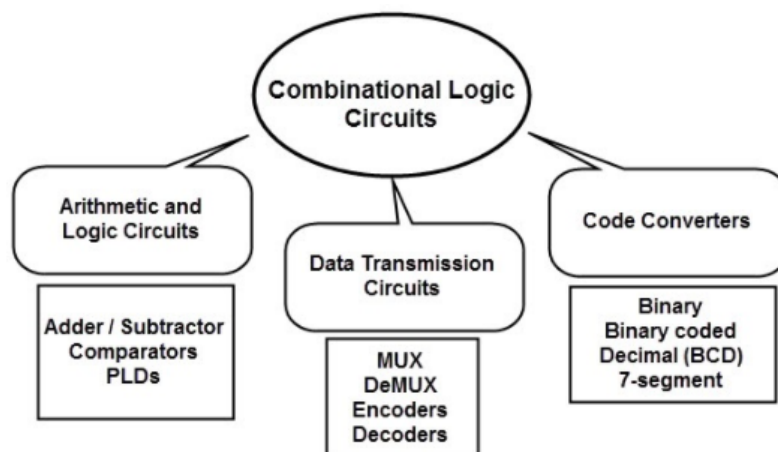


Figure 2 - <https://www.electronicshub.org/introduction-to-combinational-logic-circuits/>

According to (Eletronics Coach) a crucial factor that differentiates sequential and combination logic circuit is the state of the output elements. The state of the output elements of the combination circuit depends on only present state of the input elements. On the opposite, the output achieved

from sequential circuit is dependent on the present state of the input as well as past state of the output as can be seen in the image below.

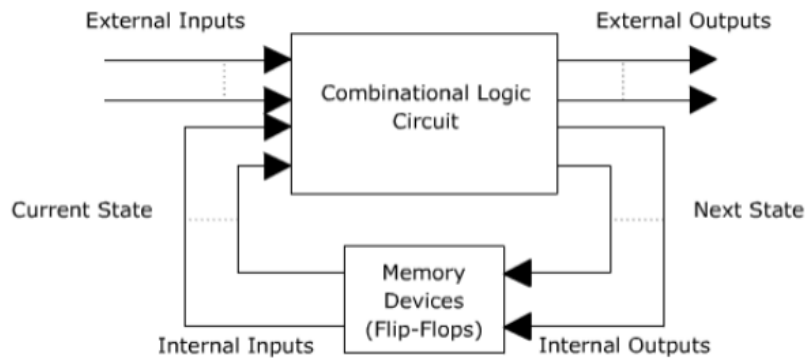


Figure 3 - <https://www.electronicshub.org/sequential-circuits-basics/>

S-R Latch

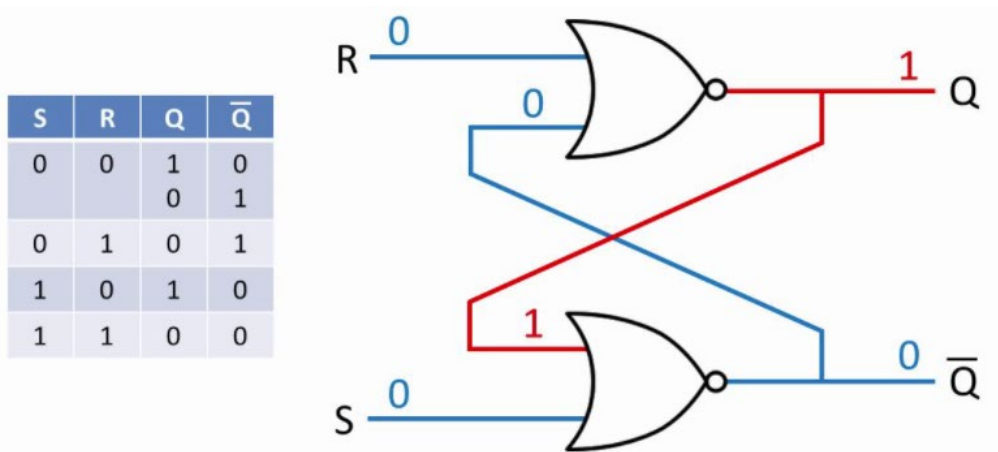


Figure 4 - <https://i.ytimg.com/vi/-aQH0ybMd3U/maxresdefault.jpg>

Figure 4 shows a logic circuit of a S-R (set-reset) latch. An S-R latch is created by connecting two NOR gates in an arrangement that the output of one feeds back to the input of another, and vice versa, as shown in figure 4 (All About Circuits, 2020). Looking at the truth table when $S=1$ and $R=0$ this 'sets' the multivibrator (latch) to $Q=1$ and $\bar{Q}=0$. When $R=1$ and $S=0$ it 'resets' the latch. If S and R are of logic 0 then the circuit latches in their prior states. Having both $S=1$ and $R=1$ would result in both outputs being the same state which need opposite states for the S-R latch to properly function. This is known as an invalid state or illegal state and a race condition will occur between the two outputs and cannot be predicted which output will go high or when. This can make the circuitry unstable and very unreliable. (2020)

J-K Flip-flop

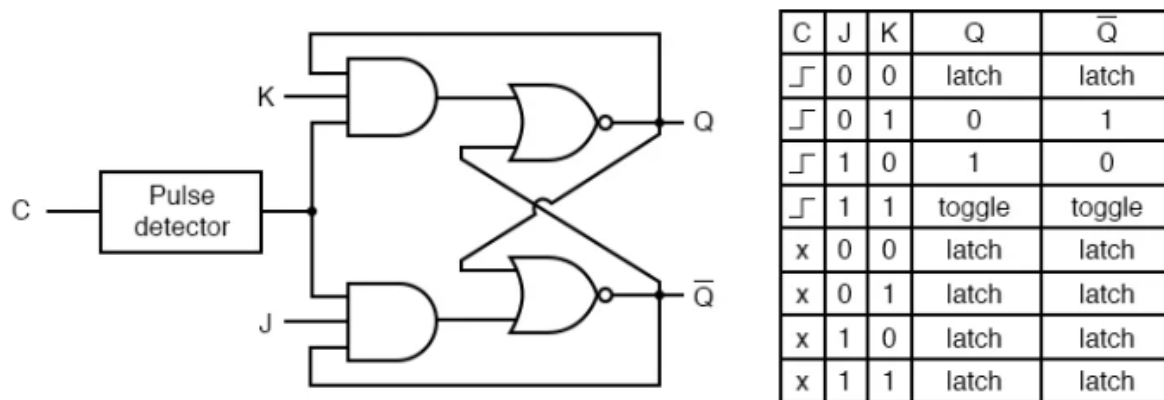


Figure 47 - <https://www.allaboutcircuits.com/textbook/digital/chpt-10/j-k-flip-flop/> J-k Flip-flop with Truth table

J-K flip-flops are a modified version of an S-R latch with no illegal states. It does this by using three input and gates with each gate receiving feedback from Q and $\bar{Q}$ and incorporating a clock signal as an input to both AND gates. This allows the J input to go high only when the circuit is reset and allow the K input to go high when the circuit is set. This creates an interlocking so that both outputs cannot activate simultaneously. In the circuits set state, the J input is constrained by the 0 output of $\bar{Q}$. In the reset state the K input is constrained by the 0 output of Q. When J and K are both 1, a set state blocks input J so the flip-flop acts as if J=0 and K=1. When the clock pulses, the outputs will toggle from set to reset. The next clock pulse will toggle the circuit from reset to set. (2020)

According to (Electrical4U, 2020), JK flip-flops can be triggered upon the leading edge of the clock or on its trailing edge, hence can be either positive or negative edge triggered.

Use of a Finite State Machine as a design tool

On a published article written by (Bors, 2018), he says that a finite state machine is a computational model which can be used to simulate sequential logic. Finite state machines are used in different areas such as mathematics, games, artificial intelligence, etc.

According to (Elprocus, 2013 - 2021), finite state machines are put into two types of classification which are Mealy state machine and Moore state machine.

According to (Bevilacqua, 2013), 'FSMs are commonly used to organize and represent an execution flow, which is useful to implement AI in games.' Fernando says that you could think of a FSM as a brain of an enemy in a game, with each state representing an action such as attack or hide, call for help, etc. The image below helps to understand an FSM implemented as AI in a game.

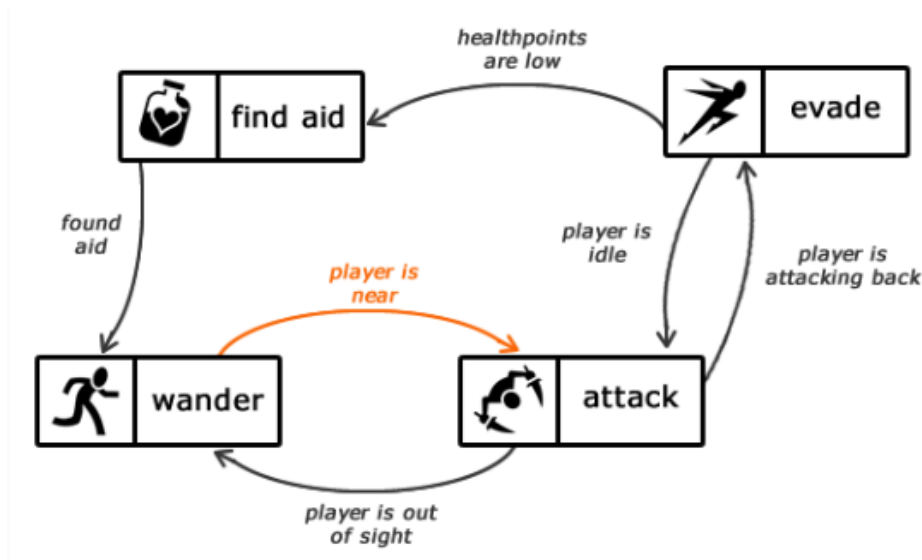


Figure 5 - <https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>

Mealy State Machine

In a logic circuit, if the outputs require the current inputs and memory states, the finite state machine is known as a mealy state machine.

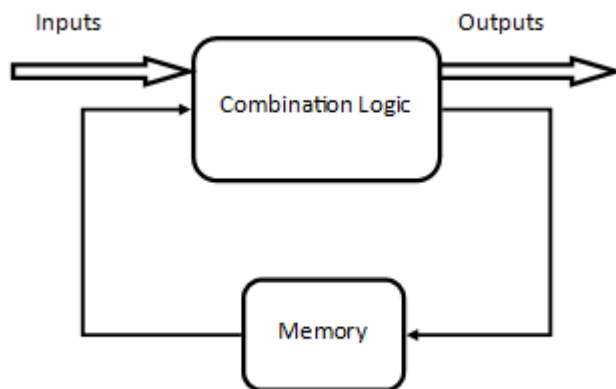


Figure 6 - Combination logic with memory

The above image is a block diagram of a mealy state machine. There are two parts to the FSM which is combination logic and memory. The memory is used to provide feedback to the combinational logic by providing the outputs as inputs. (2013 - 2021)

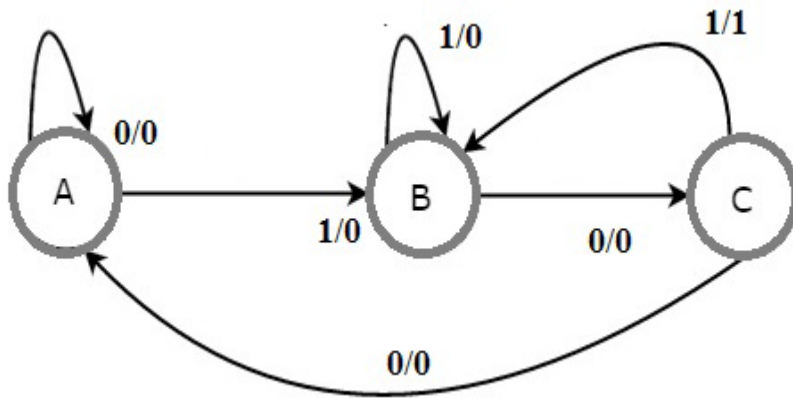


Figure 7 - <https://www.elprocus.com/finite-state-machine-mealy-state-machine-and-moore-state-machine/> State Diagram of Mealy State Machine

Above is a state diagram of a mealy state machine. As can be seen in the diagram, there are three states which are state A, B, and C. Each state has two conversions.

According to (Elprocus), the number of states required in a mealy machine is below or the same to the number of states required in a Moore state machine. There is an equal Moore state machine for every mealy state machine.

Moore State Machine

(Elprocus) says that a Moore state machine is known when the output depends on the current state. Below is a block diagram of a Moore state machine. It consists of two parts, the combinational logic and the memory.

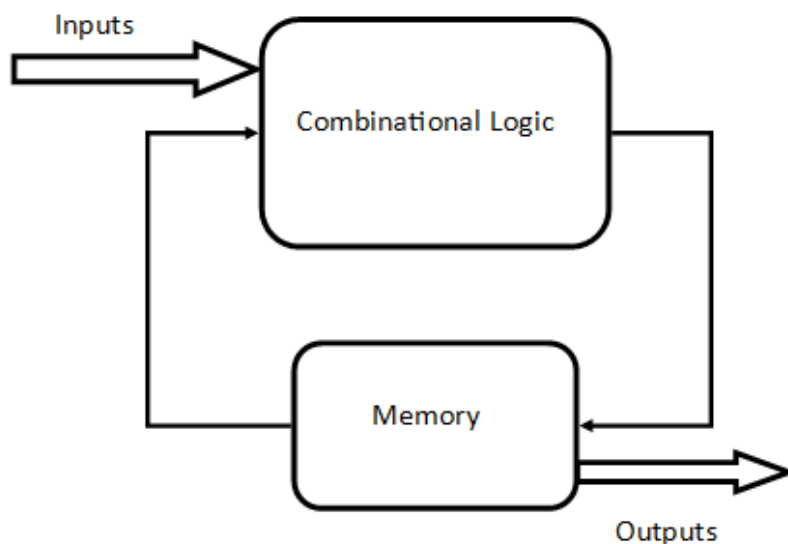


Figure 8

The next state is decided by the current inputs as well as current states. The machine will generate the outputs depending on further states.

The image below shows the state diagram of a Moore state machine. It includes four states as well as the individual outputs.

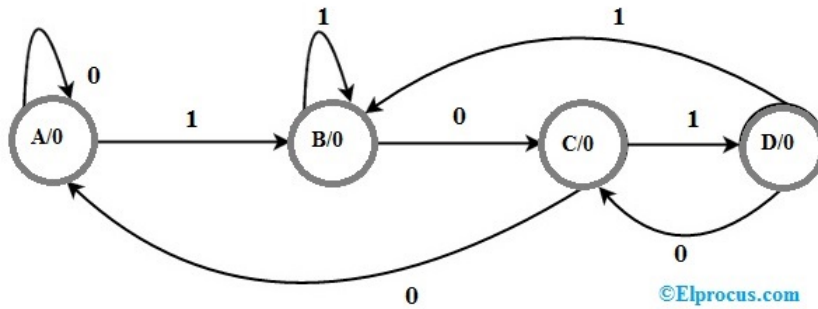


Figure 9 - <https://www.elprocus.com/finite-state-machine-mealy-state-machine-and-moore-state-machine/> State diagram of Moore state machine

In the state diagram above there are two conversions from every state which is dependent on the input value.

Relative Merits or FPGA's and ASIC's

According to (Xilinx, 2021), Field Programmable Gate Arrays (FPGA) are semi-conductor devices (chips) which are based around a cluster of configurable logic blocks connected via programmable interconnections. They can be reprogrammed to suit functionality or application specifications after manufacturing. The ability to be reprogrammed is what distinguishes FPGA's from Application Integrated Circuits (ASIC).

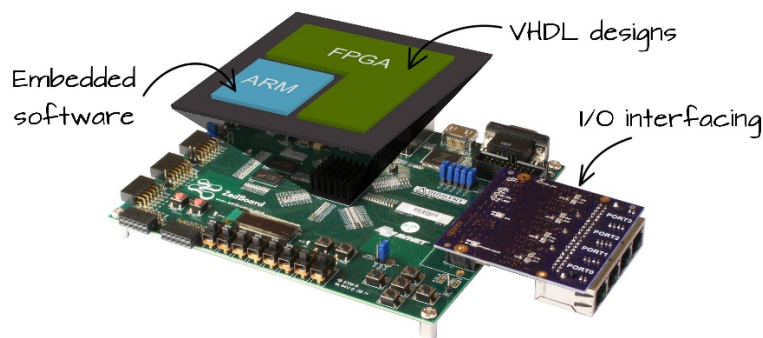


Figure 45 – [fpga-programming.jpg \(3800x1600\) \(wp.com\)](https://www.wp.com/fpga-programming.jpg) FPGA

Advantages of FPGAs

According to (Hardware Bee, 2019), there are ten main advantages to FPGAs, these are:

1. Better performance
2. Programmability
3. Cost Efficiency
4. Parallel task performance
5. Prototyping
6. Faster time to market
7. Simpler Design Cycles
8. Adaptability
9. Real time application
10. System on chip

An ASIC is a microchip that is designed for a specific application. (Sigenics, 2019) uses an example where an ASIC may be used, 'for example, a cell phone company may design an ASIC to combine the

display backlight controller with the battery charging circuit into a single IC to make the phone smaller.'



Figure 46 - [shutterstock_737150020.jpg \(1920x1200\) \(bitcoinist.com\)](#) ASIC

Advantages of ASICs

According to (Sigenics), there are 7 main advantages to ASICs, these are:

1. Price Savings
2. Noise Reduction
3. Signal Integrity
4. Size, weight and power improvement
5. IP protection
6. Device Reliability
7. Competitive Edge

VERY GOOD RESEARCH.

Design Method

A digital stopwatch is constructed by the addition of 3 circuits. These circuits are:

1. 7-Segment decoder
2. Decade counter
3. 0-5 counter

7-Segment Display

The 7-segment decoder is used for displaying the binary information in a form of decimal (or digits). The 7-segment display consists of LEDs in a certain arrangement to display a digit, as shown in the image below.

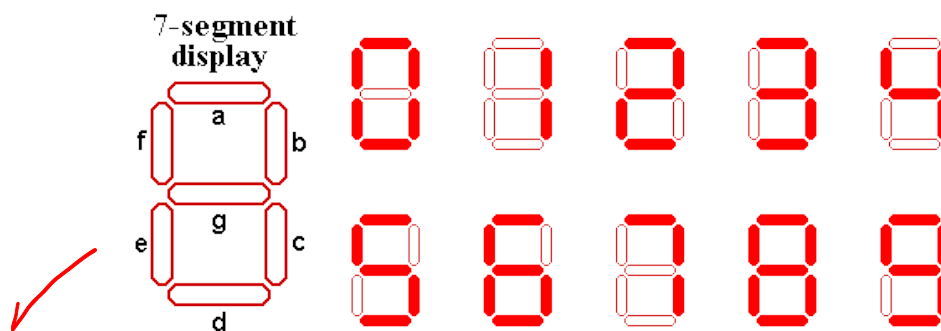


Figure 10

<http://www.physics.udel.edu/~watson/scen103/colloq2000/images/7seg1.gif>

Truth Table

For the 7-segment display to work, a truth table is formed to show the data that is to be outputted to the seven segments to display each digit.

Truth Table											
State	Inputs				Outputs (segments)						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
Don't care states	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0	x	x	x	x	x	x	x
	1	1	0	1	x	x	x	x	x	x	x
	1	1	1	0	x	x	x	x	x	x	x
	1	1	1	1	x	x	x	x	x	x	x

Figure 11

To demonstrate the understanding of this truth table state 0 should display zero (0) on the 7-segment display. When all 4 inputs (A, B, C, D) are of bit 0, the outputs a, b, c, d, e, f are all bit 1 which will light up the corresponding segments and form a zero on the display. Output g remains a bit logic of 0 so that LED segment g does not light up which would form an 8.

Karnaugh Maps & Boolean Equations

The main purpose of Karnaugh maps is to simplify the equations so that the circuits are much smaller as if you were to derive the equations from the main truth table above, they would be much bigger and requiring lots of extra logic gates.

	AB/CD	00	01	11	10
seg a	00	1	0	1	1
	01	0	1	1	1
	11	x(1)	x(1)	x(1)	x(1)
	10	1	1	x(1)	x(1)
$a = A + C + BD + \overline{BD}$					

Figure 12

The states of segment a is arranged into a Karnaugh map as in the image above, but the last two rows are flipped so states 8, 9 10(X), 11(X) are in the fourth row and the rest of the don't care states are in the third row. The inputs AB are located at the left side of the map and inputs CD are on the top of the map. The creation of the Boolean equations is formed by grouping all the output 1's. In the Karnaugh map above there are four groupings, each with a different colour to represent a different group. For the red group, input A is the same for all the outputs, therefore the equation is A. For the blue group input C is the same in columns three and four, therefore the equation is C. For the green group, inputs B and D are the same, therefore the equation is BD. In the black group, input B and D are the same but because the outputs are zero, the equation is NOTB and NOTD ($\overline{BD}$).

	AB/CD	00	01	11	10
seg b	00	1	1	1	1
	01	1	0	1	0
	11	x(1)	x	x(1)	x
	10	1	1	x(1)	x(1)
$b = \overline{B} + CD + \overline{CD}$					

Figure 13

	AB/CD	00	01	11	10
seg g	00	0	0	1	1
	01	1	1	0	1
	11	x(1)	x(1)	x(1)	x(1)
	10	1	1	x(1)	x(1)
$g = A + C\bar{D} + B\bar{C} + \bar{B}C$					

Figure 18

MAPS ARE CLEARLY GROUPED

7-Segment Display Circuits

Now that the Boolean equations are found for each segment of the 7-segment display, a circuit can be made to properly operate and display digits on the display. The circuits are built on MultiSim and are shown in the images below.

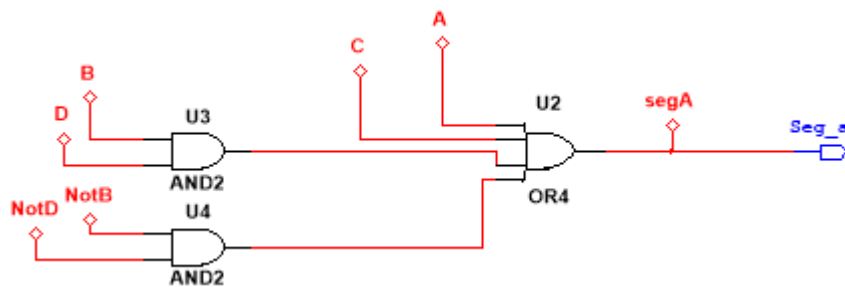


Figure 19

As can be seen in the image above, for segment_a of the display to work the Boolean equation is put into a circuit using two AND gates and 1 OR gate.

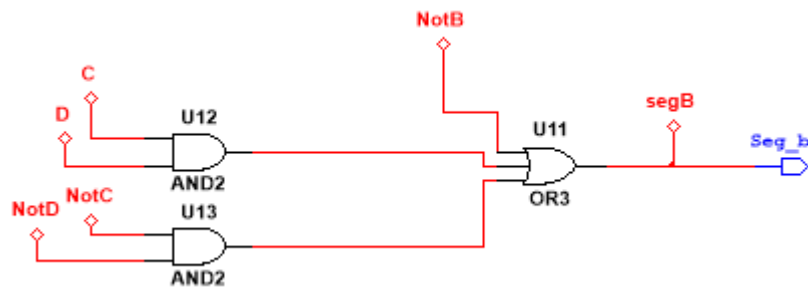


Figure 20

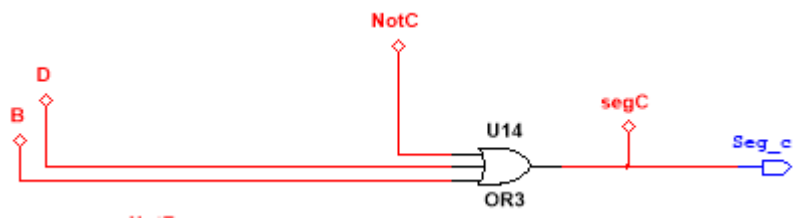


Figure 21

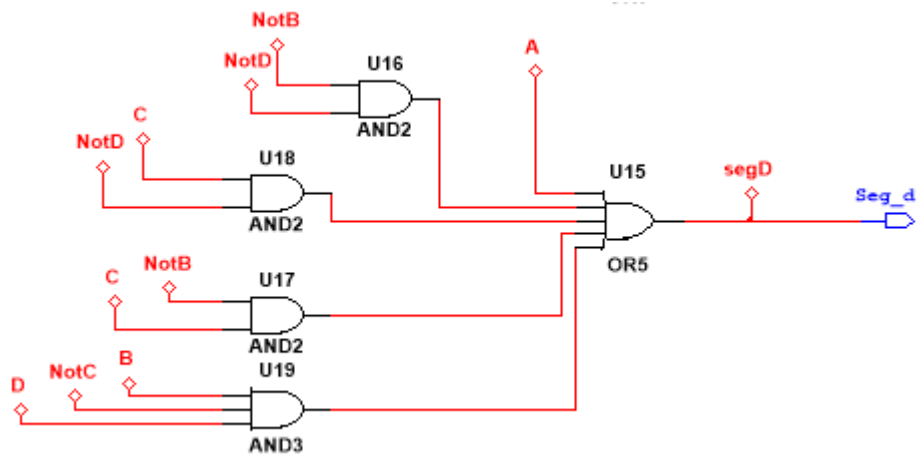


Figure 22

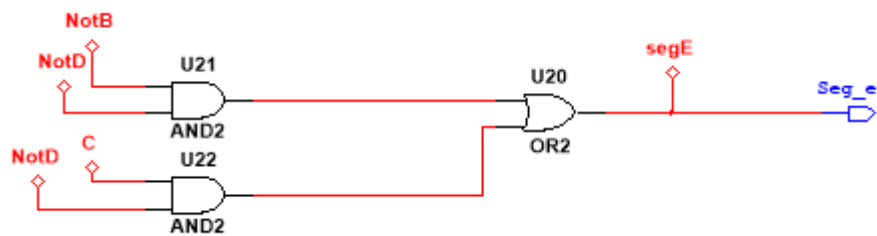


Figure 23

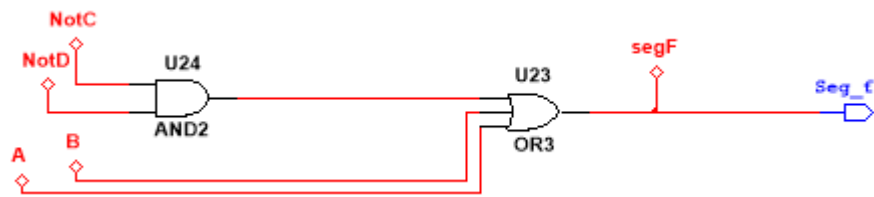


Figure 24

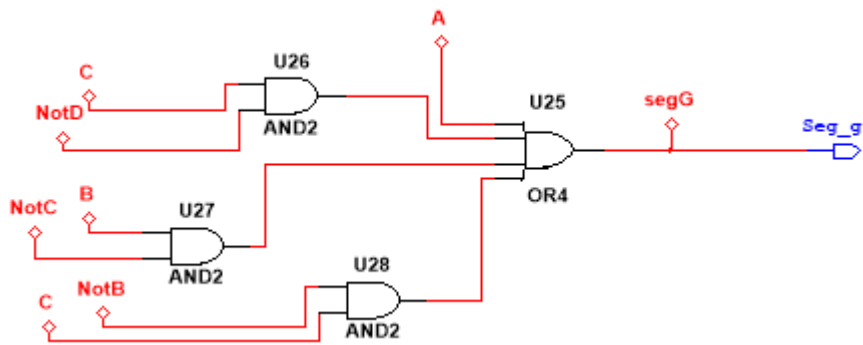


Figure 25

0-5 Counter Design

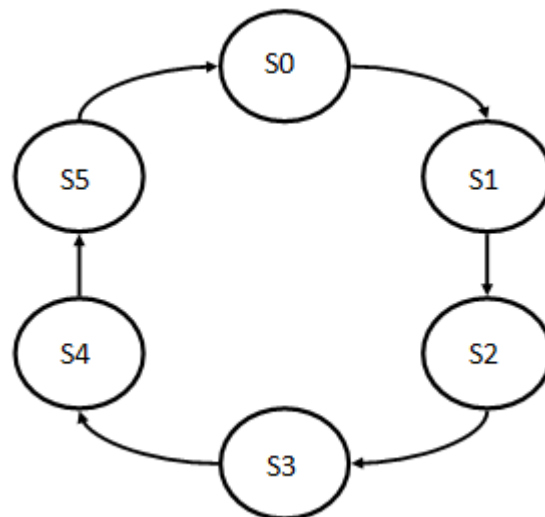
The specification for this design is to count repeatedly from 0 to 5. To achieve this a finite state machine is created in MultiSim using flip-flops to have a sequential circuit.

State allocation

States	Function
S0	Count 0
S1	Count 1
S2	Count 2
S3	Count 3
S4	Count 4
S5	Count 5

Figure 26 - State table & State diagram

State Diagram



State Table

Present State	Next State	Present			Next		
		Qa	Qb	Qc	Qa+	Qb+	Qc+
S0	S1	0	0	0	0	0	1
S1	S2	0	0	1	0	1	0
S2	S3	0	1	0	0	1	1
S3	S4	0	1	1	1	0	0
S4	S5	1	0	0	1	0	1
S5	S0	1	0	1	0	0	0
S6	X	1	1	0	X	X	X
S7	X	1	1	1	X	X	X

Figure 27 - State table

Now that the state table is produced, we can create Karnaugh maps to obtain a simplified Boolean equation.

Karnaugh Maps

	Qa/Qb/Qc				
		00	01	11	10
Qa+	0	0	0	1	0
	1	1	0	x(1)	x(1)
$Q_A + = Q_B Q_C + Q_A \overline{Q_C}$					

Figure 28

	Qa/Qb/Qc				
		00	01	11	10
Qb+	0	0	1	0	1
	1	0	0	x	x(1)
$Q_B + = Q_B \overline{Q_C} + Q_A \overline{Q_C} Q_C$					

Figure 29

	$Q_a/Q_b/Q_c$				
$Q_c +$		00	01	11	10
0		1	0	0	1
1		1	0	x	x(1)
		$Q_c + = \overline{Q_c}$			

Figure 30

Now that the Boolean equations have been achieved from the groupings in the Karnaugh maps the circuit can be built on MultiSim.

Circuit

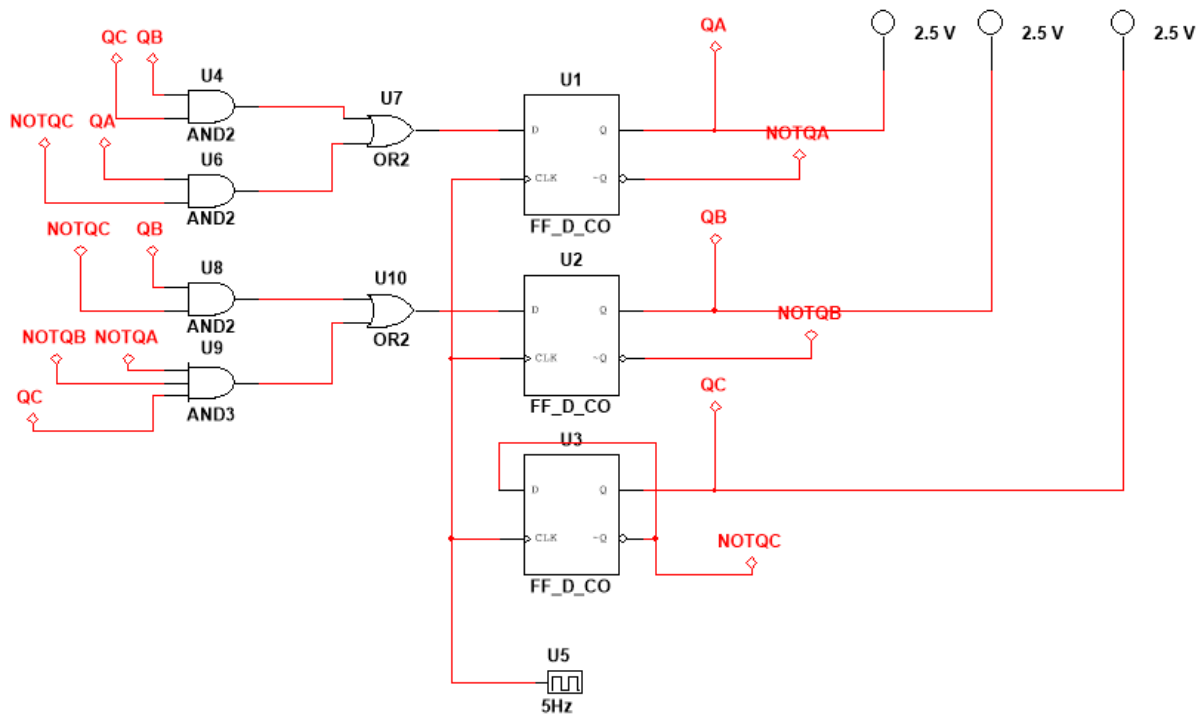


Figure 31

The circuit above has the Boolean equations connected to the input of the AND gates and each part of the equation goes to the OR gate. From there, the output of the OR gates is connected two of the flip flops, with the final flip flop taking its output to its input creating a positive feedback.

0-9 Decade Counter Design

The specification of this design is to create a counter that will repeatedly count from 0 to 9 using finite state machine method. This will require 4 flip-flops for the design.

State Allocation

States	Function
S0	Count 0
S1	Count 1
S2	Count 2
S3	Count 3
S4	Count 4
S5	Count 5
S6	Count 6
S7	Count 7
S8	Count 8
S9	Count 9

State Diagram

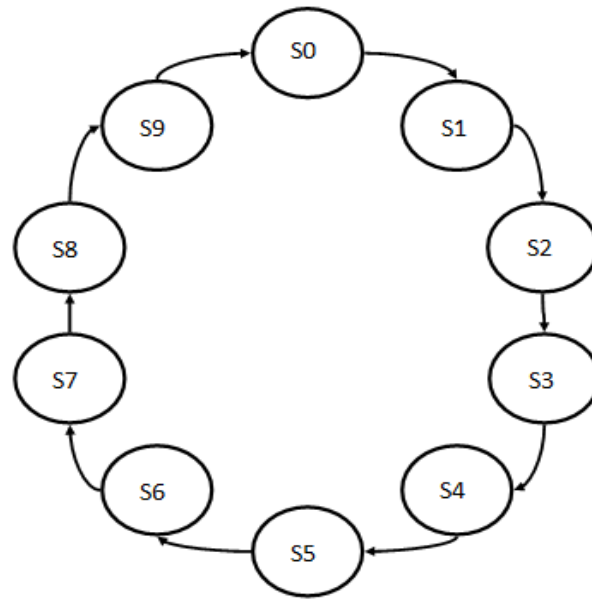


Figure 32 – State allocation & State Diagram

State Table

Present State	Next State	Present				Next			
		Q _A	Q _B	Q _C	Q _D	Q _A +	Q _B +	Q _C +	Q _D +
S0	S1	0	0	0	0	0	0	0	1
S1	S2	0	0	0	1	0	0	1	0
S2	S3	0	0	1	0	0	0	1	1
S3	S4	0	0	1	1	0	1	0	0
S4	S5	0	1	0	0	0	1	0	1
S5	S6	0	1	0	1	0	1	1	0
S6	S7	0	1	1	0	0	1	1	1
S7	S8	0	1	1	1	1	0	0	0
S8	S9	1	0	0	0	1	0	0	1
S9	S0	1	0	0	1	0	0	0	0
S10	X	1	0	1	0	X	X	X	X
S11	X	1	0	1	1	X	X	X	X
S12	X	1	1	0	0	X	X	X	X
S13	X	1	1	0	1	X	X	X	X
S14	X	1	1	1	0	X	X	X	X
S15	X	1	1	1	1	X	X	X	X

Figure 33 – State table

WELL PRESENTED

Karnaugh Maps

		QAQB/QCQD			
		00	01	11	10
$Q_A +$	00	0	0	0	0
	01	0	0	1	0
	11	1	0	x(1)	x(1)
	10	x(1)	x	x	x(1)
		$Q_{A+} = Q_A \overline{Q_D} + Q_B Q_C Q_D$			

Figure 34

		QAQB/QCQD			
		00	01	11	10
$Q_B +$	00	0	0	1	0
	01	1	1	0	1
	11	0	0	x	x(1)
	10	x	x	x(1)	x
		$Q_{B+} = Q_B \overline{Q_C} + \overline{Q_B} Q_D + \overline{Q_B} Q_C Q_D$			

Figure 35

		QAQB/QCQD			
		00	01	11	10
$Q_C +$	00	0	1	0	1
	01	0	1	0	1
	11	0	0	x	x(1)
	10	x	x	x	x(1)
		$Q_{C+} = \overline{Q_A} Q_C Q_D + Q_C \overline{Q_D}$			

Figure 36

		QAQB/QCQD			
		00	01	11	10
$Q_D +$	00	1	0	0	1
	01	1	0	0	1
	11	1	0	x	x(1)
	10	x(1)	x	x	x(1)
		$Q_{D+} = \overline{Q_C}Q_D + Q_C\overline{Q_D}$			

Figure 37

0-9 Counter Circuit

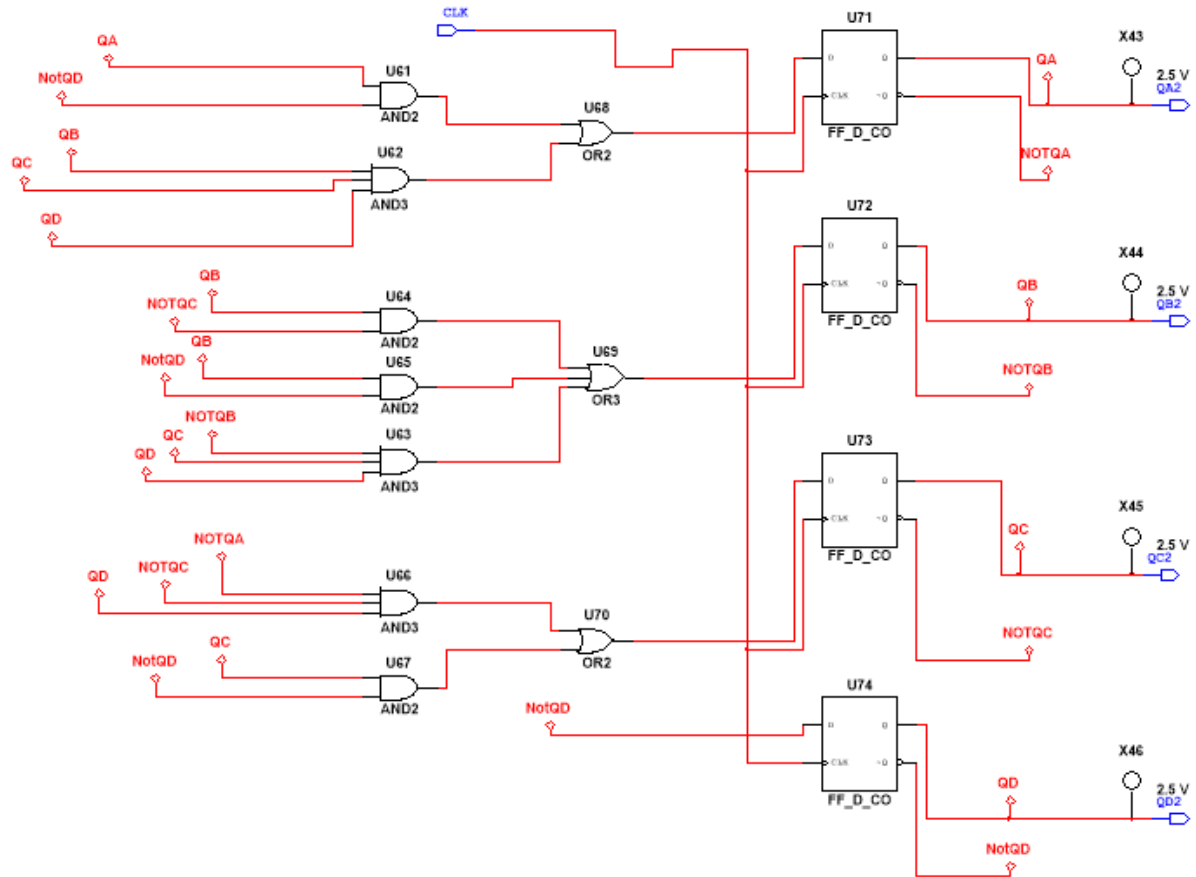


Figure 38 - 0-9 Counter circuit on MultiSim

Simulation

Simulating 7-Segment display

To test the 7-segment circuit a display was built on MultiSim using LEDs as shown in the image below. As well as using the inputs from the truth table in the design section of the report and their corresponding inverse logic. At state 0, inputs A B C D are of logic 0 but the inputs are also connected to inverter gates which give a logic of 1 as can be seen in the diagram below. Therefore it satisfies the Boolean equations/segment circuits for all segments apart from segG and produces a 0 on the display.

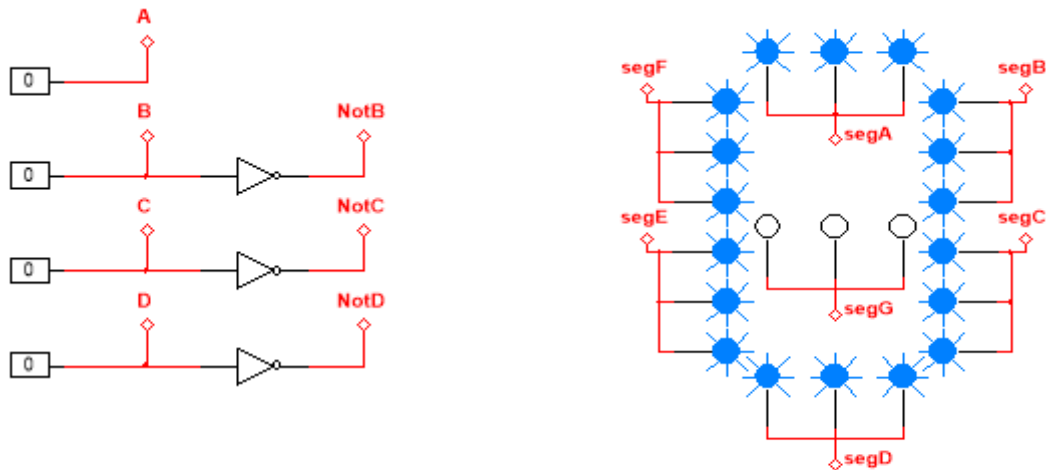


Figure 39 - Inputs to bring on 0 sign on 7-segment display

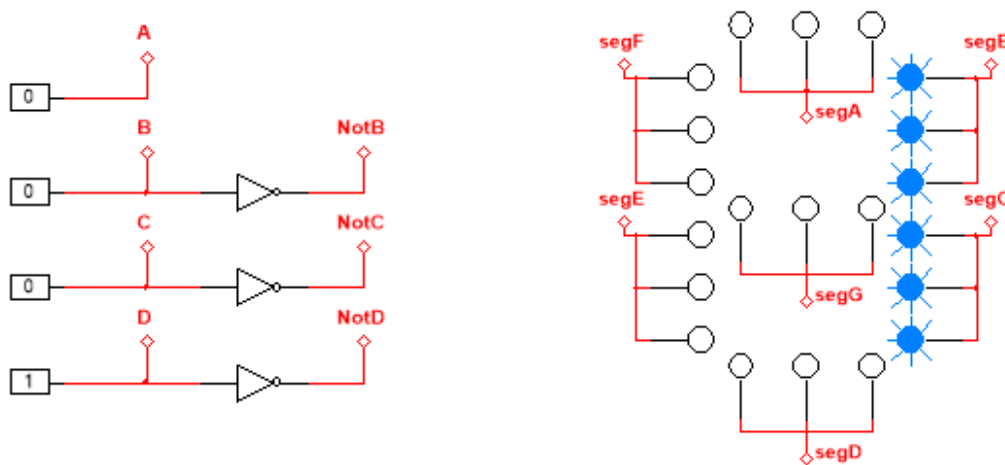


Figure 40 - Inputs to bring on 1 sign on 7-segment display

By entering the required inputs for state 1 it switches the LEDs on segB and segC to display a 1 as seen in the above image. This was repeated for state 2-9 to prove display works.

Simulating the 0-5 Counter

To simulate the 0-5 counter, a logic analyser is connected to the outputs to produce a visual representation of the outputs.

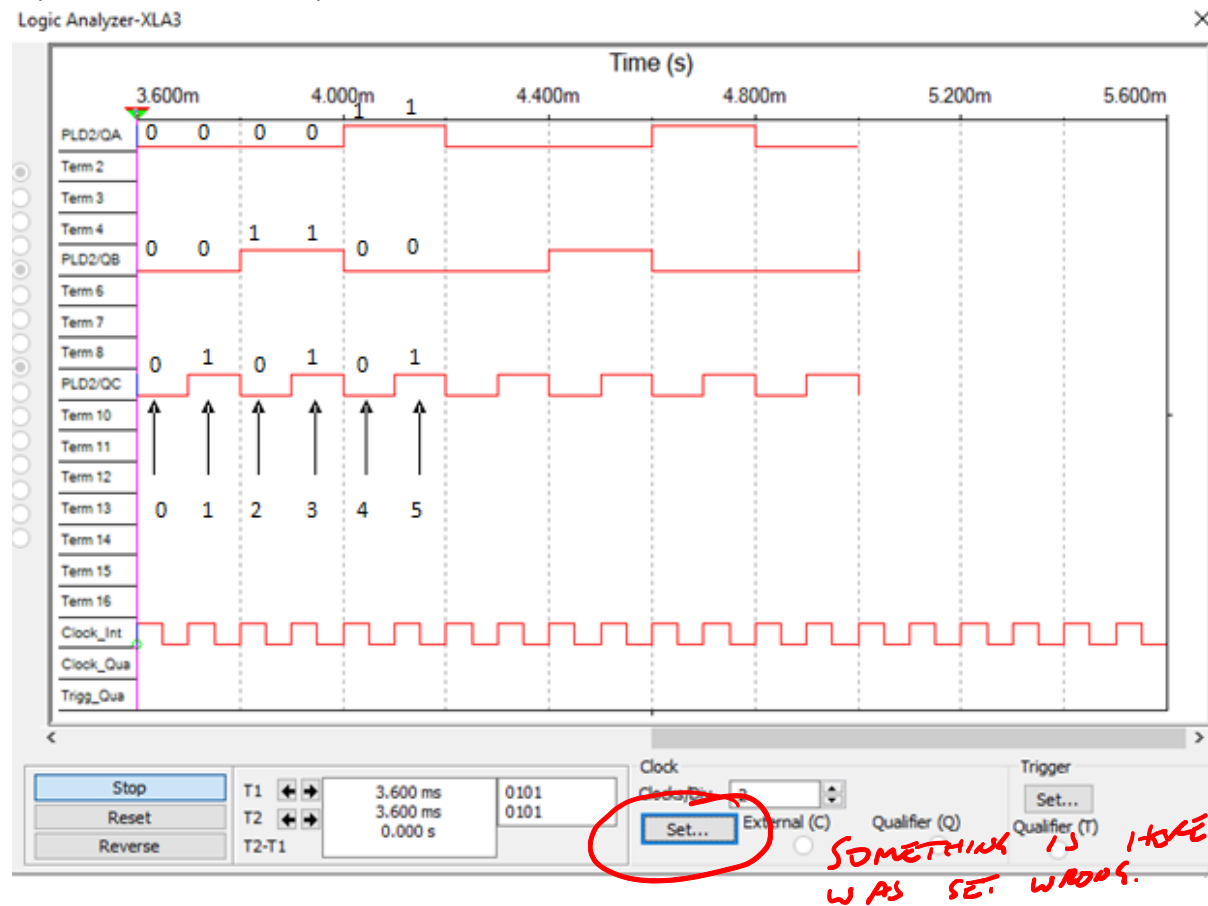


Figure 41 – Logic analyser connected to outputs of the 0-5 Counter circuit in MultiSim

By analysing the results from the logic analyser above and comparing it to the state table we can determine if the logic is correct and works. The numbers at the bottom of represent the states and when the output line for QA QB QC goes high it indicates a logic 1 and when low indicates logic 0. For state 0 the results show that QA=0, QB=0, QC=0 which matches the state table created earlier. This is repeated for all states in the truth table created in the design method to prove the counter works.

Simulating 0-9 Decade counter

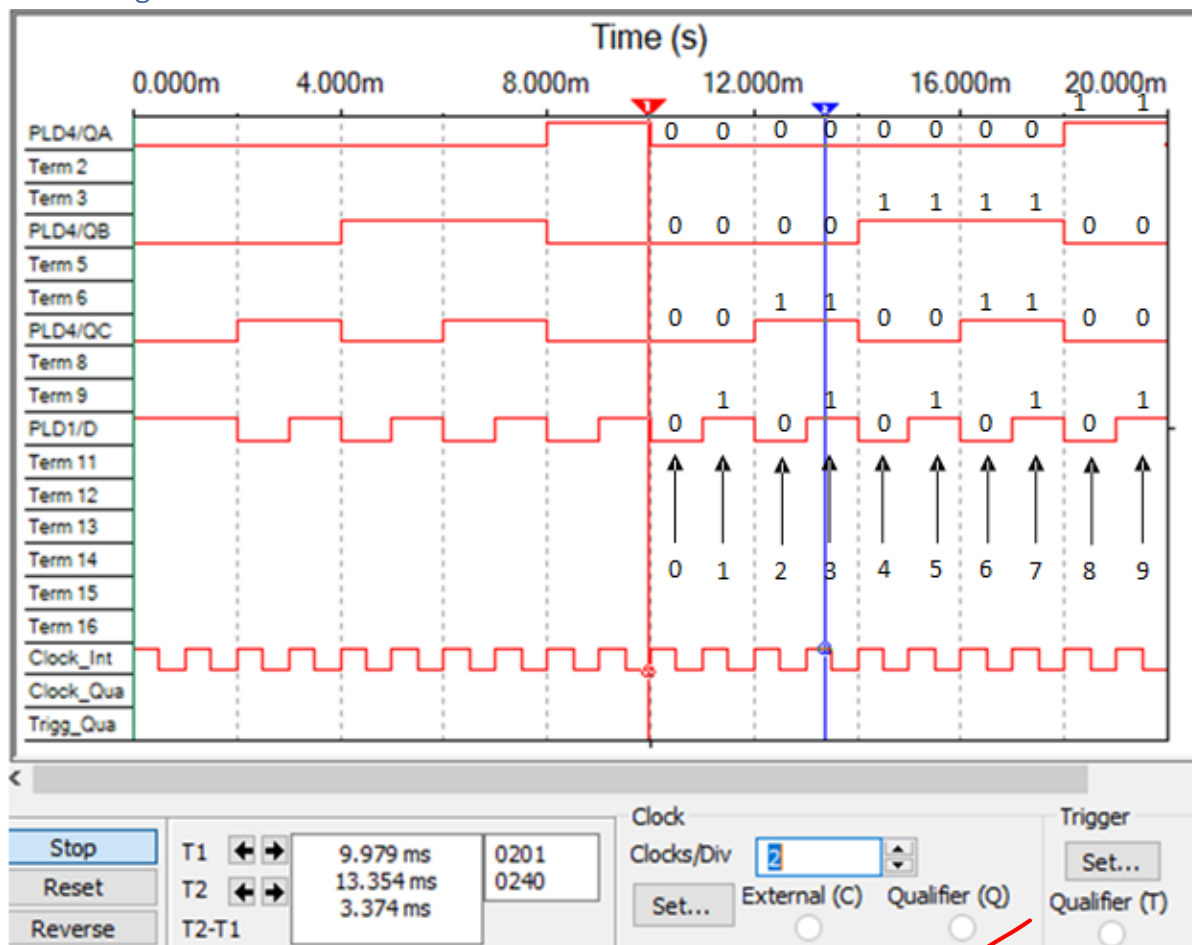


Figure 42 - Logic analyser connected to outputs of 0-9 Counter circuit in MultiSim

As can be seen in the image above the counter is repeatedly counting from 0-9. This is proved by using the same method of simulating the 0-5 counter. The logic in the image matches with the truth table created in the design of the 0-9 counter.

Simulating final design

Simulating the final design is a lot trickier and time consuming than the counters, this is because there are a lot more outputs to be monitored by the logic analyser and these outputs are segments to both displays as shown in the diagram below. The 0-5 counter is connected to the top display (i.e. seg_a2, etc) and the decade counter connected to seg_a1, etc. It is clear the bottom section is the decade counter because a lot more outputs are changing.

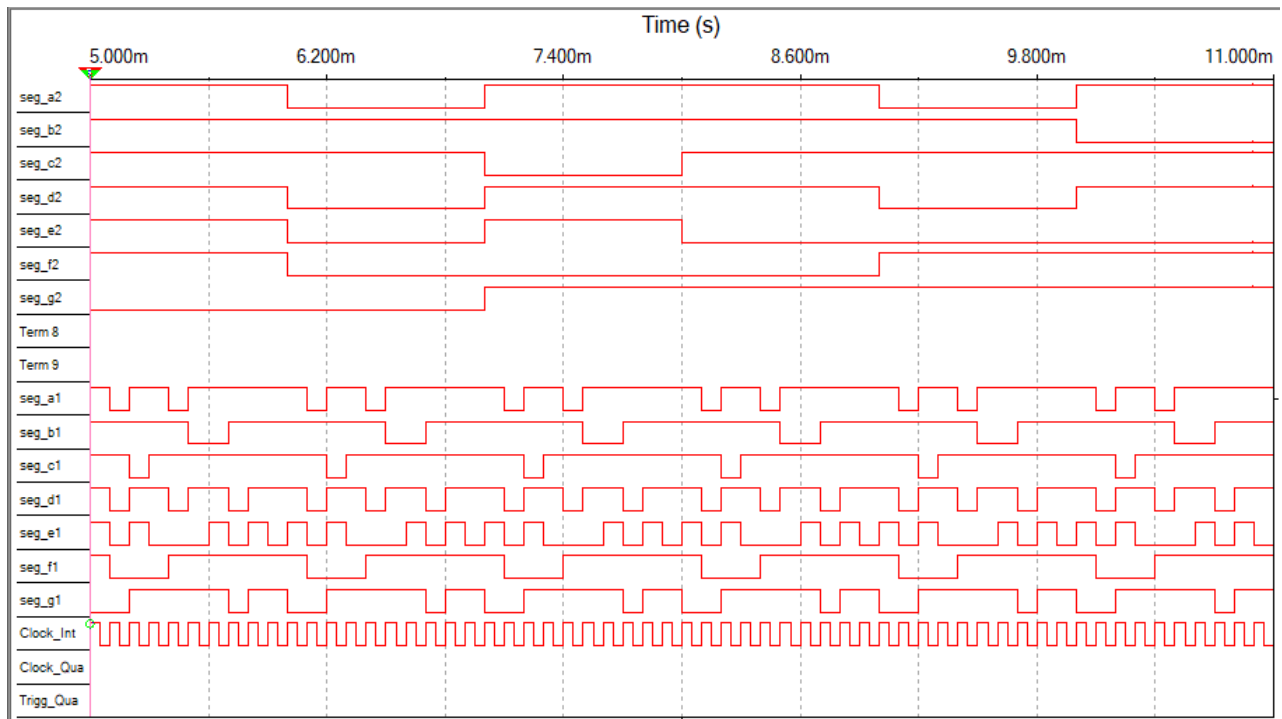


Figure 43 - Logic analyser of outputs to each segment of both displays in Multisim

Conclusion

From the findings of the testing of the circuits, it is concluded that the circuit will count from 00 to 59 repeatedly and can be used as a digital stopwatch. Because it works as intended, it proves the theory and designs of the truth tables, Karnaugh maps, Boolean equations, etc, to be correct. Creating an FSM in the counter circuit using bistable gates (D-type flip-flops) was crucial in the counting circuit to store memory and create a sequence by using a clock as an input to the gate, this also backs up the research about sequential logic. An S-R latch would not work for the counter as there is no clock input pin and has the possibility to perform illegal states which would cause a race condition and lead to an unstable/unreliable circuit. The use of combinational logic (gates) to create the 7-segment display worked as intended.

From the research, FPGA work great as a design tool as it can be reprogrammed, whereas ASICs cannot be reprogrammed, however ASICs are designed for one application only to meet customer needs, so in result less gates are used in the circuitry which can make the device smaller, less noisy, faster processing speeds, and more cost effective.

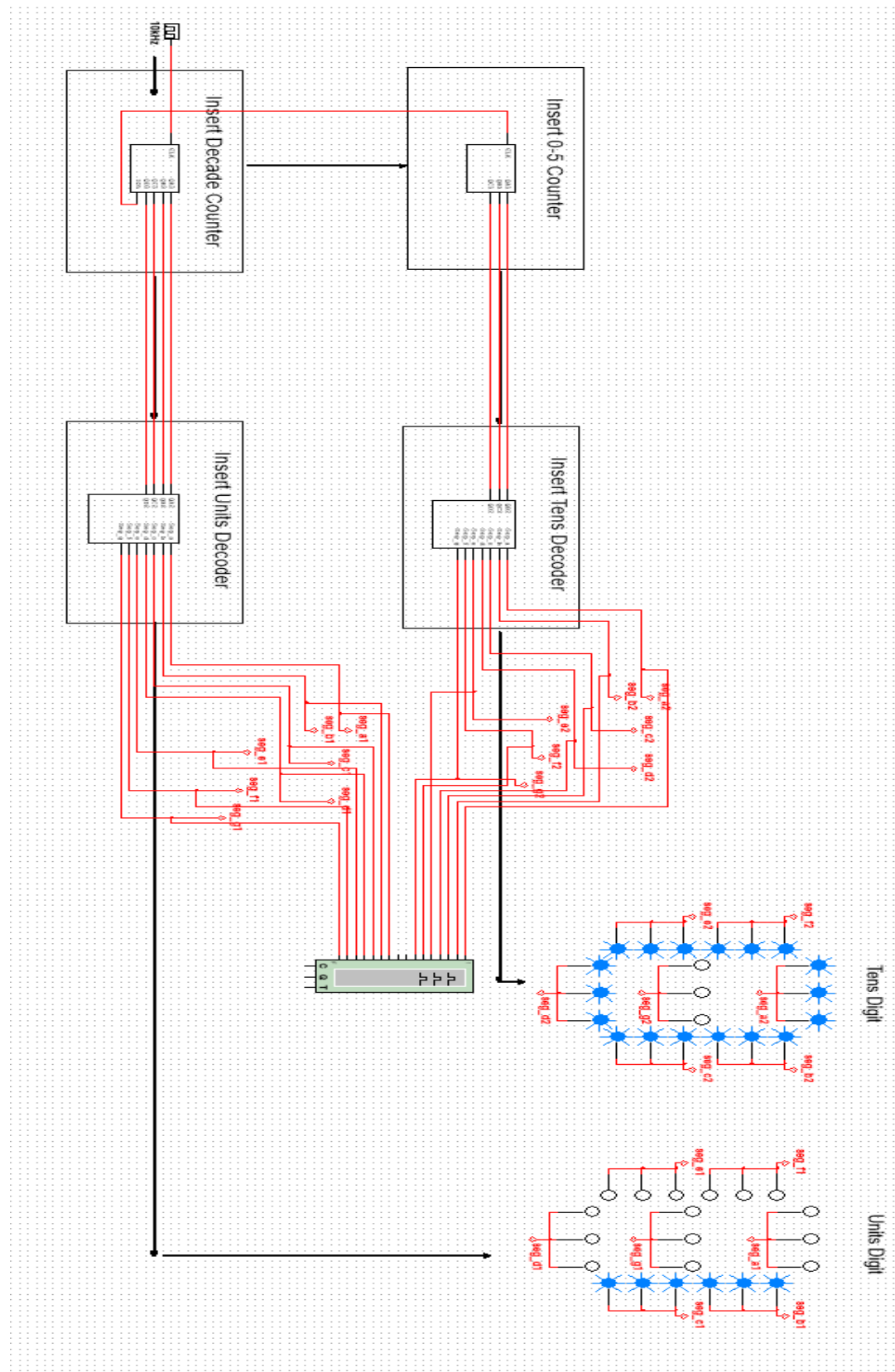
REASONABLE CONCLUSIONS

References

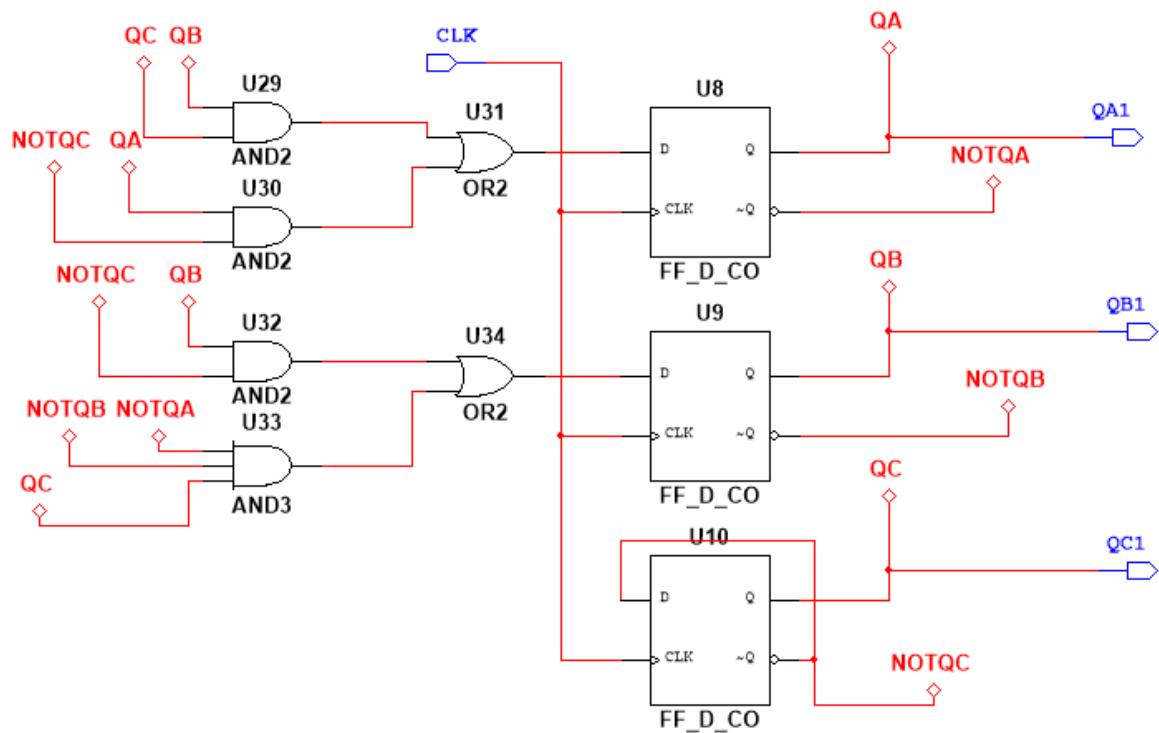
- Administrator, 2015. *Introduction to Combinational Logic Circuits*. [Online]
Available at: <https://www.electronicshub.org/introduction-to-combinational-logic-circuits/>
[Accessed 2nd May 2021].
- All About Circuits, 2020. *The S-R Latch*. [Online]
Available at: <https://www.allaboutcircuits.com/textbook/digital/chpt-10/s-r-latch/#:~:text=The%20simplest%20bistable%20device%2C%20therefore%2C%20is%20known%20as,outputs%20are%20supposed%20to%20be%20in%20opposite%20states.>
[Accessed 2nd May 2021].
- Bevilacqua, F., 2013. *Finite-State Machines: Theory and Implementation*. [Online]
Available at: <https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>
[Accessed 2nd May 2021].
- Bors, M. L., 2018. *What is a Finite State Machine?*. [Online]
Available at: <https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c>
[Accessed 2th May 2021].
- Electrical4U, 2020. *JK Flip Flop: What is it? (Truth Table & Timing Diagram)*. [Online]
Available at: <https://www.electrical4u.com/jk-flip-flop/>
[Accessed 2nd May 2021].
- Eletronics Coach, 2021. *Difference between Combinational and Sequential Logic Circuit*. [Online]
Available at: <https://electronicscoach.com/difference-between-combinational-and-sequential-logic-circuit.html>
[Accessed 15th April 2021].
- Elprocus, 2013 - 2021. *Finite State Machine: Mealy State Machine and Moore State Machine*. [Online]
Available at: <https://www.elprocus.com/finite-state-machine-mealy-state-machine-and-moore-state-machine/>
[Accessed 2nd May 2021].
- Hardware Bee, 2019. *Top 10 FPGA Advantages*. [Online]
Available at: <https://hardwarebee.com/top-10-fpga-advantages/#:~:text=Top%2010%20FPGA%20Advantages%201%20Better%20Performance.%20One,Time%20Application.%20...%2010%20System%20on%20Chip.%20>
[Accessed 2nd May 2021].
- Sigenics, 2019. *7 ASIC Integration Benefits*. [Online]
Available at: <https://www.sigenics.com/blog/7-asic-integration-benefits#:~:text=7%20ASIC%20Integration%20Benefits%201%20Price%20Savings.%20An,...%203%20Signal%20Integrity.%20...%20More%20items...%20>
[Accessed 2nd May 2021].
- Xilinx, 2021. *What is an FPGA? Field Programmable Gate Array*. [Online]
Available at: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>
[Accessed 2nd May 2021].

Appendix

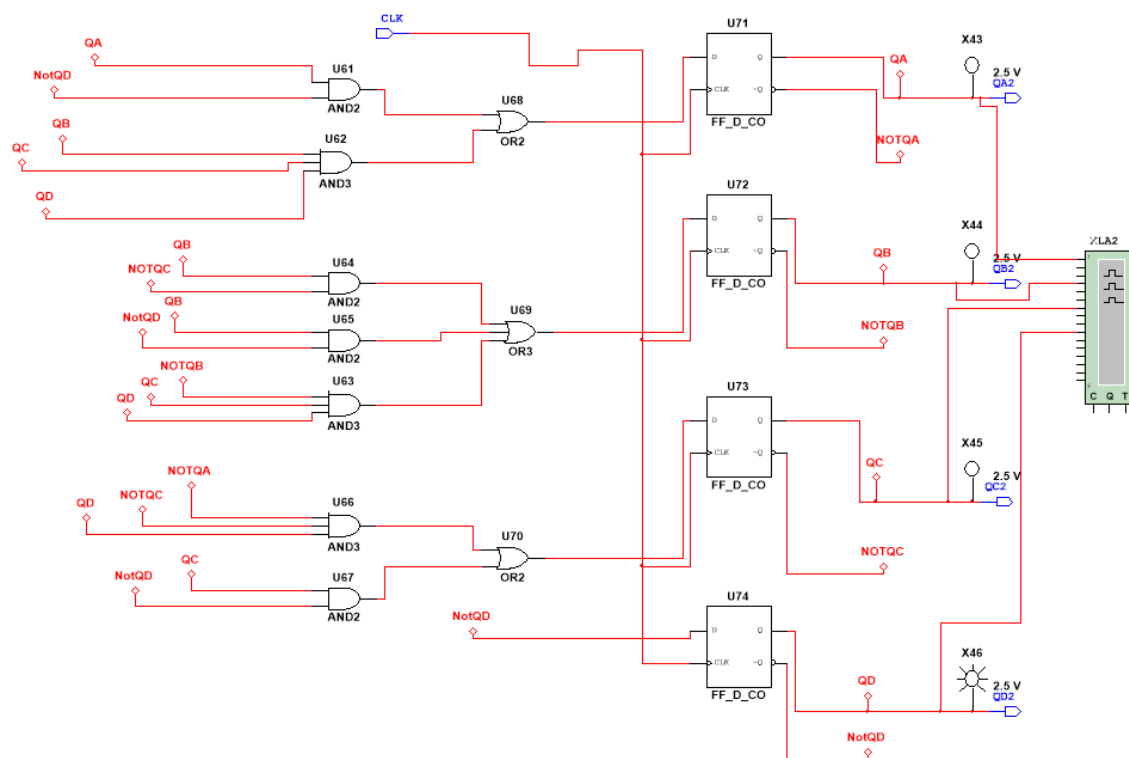
Subcircuits connected together to produce a working counter from 00-59.



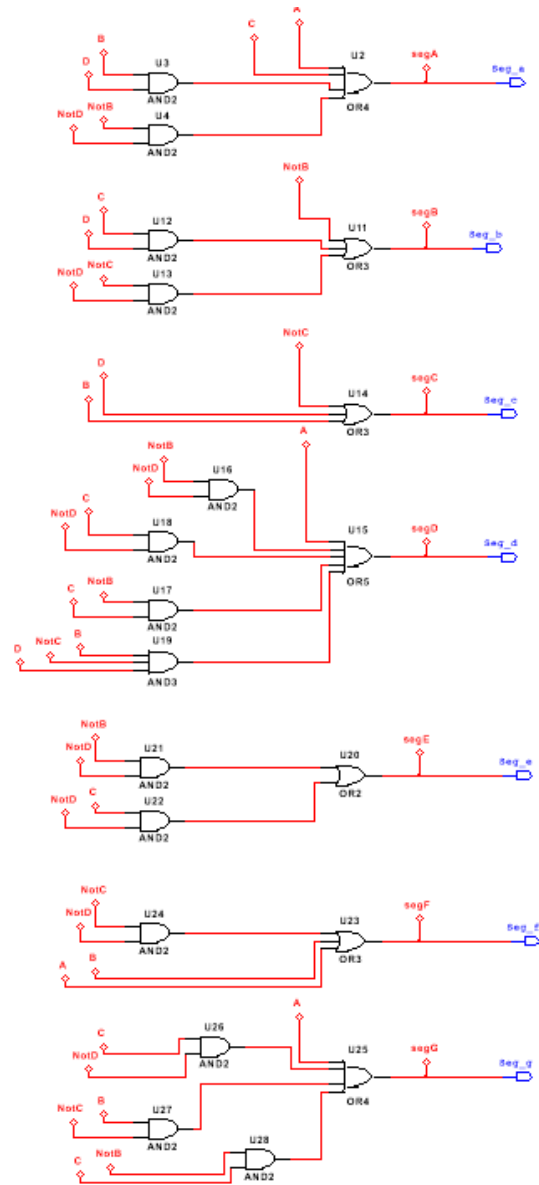
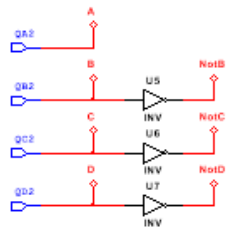
0-5 Counter circuit



0-9 Decade counter



7-Segment display circuit



FINAL GRADE

GENERAL COMMENTS

Instructor

85/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

GRADING FORM: CSG-RUBIC

JASON BOOTH

0

RESEARCH-INFORMED LITERATURE



There was evidence of extensive further reading and research from credible and reliable sources. Your interpretation and understanding of the additional research was very good and demonstrated a deep underlying appreciation of the subject. Your references were consistently applied and appropriate to the task and your reference list was of an appropriate length for this assignment.

8/10

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally excellent. There were very few gaps in your knowledge of the basic principles and your understanding of the higher level concepts was demonstrated to a high degree. You have demonstrated a highly competent critical understanding of the main concepts and expanded your knowledge well beyond that contained in the course notes.

26/30

ANALYSIS



Your analysis of the task was excellent, both in class and as shown in this assignment. You have demonstrated highly competent analytical skills and your approach to the task was highly logical. You carried out a detailed evaluation of the results and included an excellent conclusion relating your practical results to accepted theory. You have demonstrated a capacity for independent thought and your interpretation of the quantitative and qualitative data was excellent with sound judgements based on wholly valid arguments.

26/30

PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally excellent. You used both the basic and advanced features of the resources appropriately when designing the circuit and you used appropriate advanced options for testing and evaluating the circuit performance. You managed

17/20

to design an effective circuit that functioned well within the specification and achieved this by rarely asking for assistance thus demonstrating some advanced problem solving skills. Your results were comprehensive and allowed for a detailed conclusion to be written.

SKILLS FOR PROFESSIONAL PRAC.



In this assignment you have demonstrated skills in working individually and also collaboratively. Your written report was written in a professional and confident style, however you still need to be more efficient with your word count. The report was very well presented, with excellent diagrams in appropriate places, good sentence/paragraph structure and clear calculations.

8/10