The University of Sydney

School of Aerospace, Mechanical and Mechatronic Engineering

Mechatronics 1 - Introduction to Mechatronic Design

Assignment No. 1

Design Report

Biased Coin Toss State Machine Circuit

Author:
Jordan Lee Guyot

510293372

September 24, 2021

Contents

List of Figures

List of Tables

Introduction

Requirement Specifications and Analysis

This project requires a circuit to be created that can produce a 'biased coin toss'. The result of the coin toss is to be displayed on a 7 segment LED display (using some combination of the segments to indicate heads or tails respectively). The bias that this circuit will use, based on the designer's student number, is a 25% heads/75% tails bias.

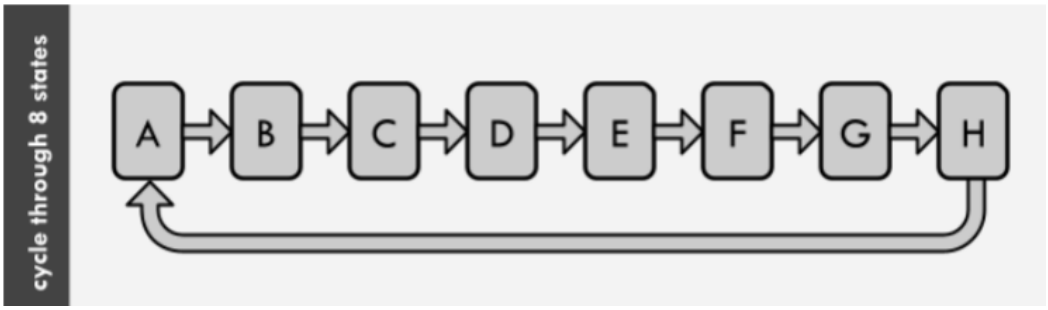

Figure 1: The state diagram of the circuit.

The first task requires a state machine to be generated with 8 states, cycling in a sequence as shown in Figure 1. The state machine should cycle at a frequency faster than the eye can see, and stop such that the current state can be displayed on a 7-segment display. Accordingly, this should result in a random output each time.

The second task requires an output logic to determine which of the states should output heads or tails to the 7-segment display to give the desired bias.

The extension task requires that a button can be used to make the output fair, i.e. a 50% heads/50% tails output.

This design report outlines the design process taken to create a circuit that meets the above specifications, with discussions about how each module can be tested and determined functional, including a brief description of how the system can be confirmed to be performing to the required specifications.

Key functionality

This circuit will be working and deemed functional if it is able to produce the specified output behaviour based on two inputs:

1. A button to stop the transition of the states (pressed = cycling, released = no cycling).
2. A button to change the output from a bias output to a fair output, as described above (pressed = fair, released = bias).

1 System/High Level Design

At a high level, a sequential logic circuit will be designed with five key modules:

1. Next State Logic
2. State Memory
3. Clock Signal
4. Output Logic
5. Output Display

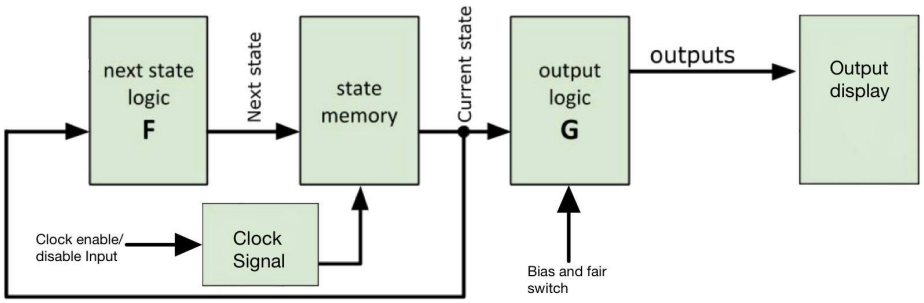

Figure 2: The high level design of the state machine outlining the key modules.

These will integrate together as seen in Figure 2. The development of each module will be outlined in turn below.

2 Detailed Design

This section will outline the design and development process of each module prior to integration. It will provide calculations used to generate each module.

2.1 Next State Logic

2.1.1 Definition of states

To represent eight different states (S) and their transitions as outlined in Figure 1, we need three bits to represent the state Q.

State	Q
A	000
B	001
C	011
D	010
E	110
F	111
G	101
H	100

Table 1: The definition of each state labelled A-H to a 3-bit number.

The states have been defined in a way such that there is only a one-bit change between the states, to achieve the simplest next state logic circuit possible as seen in Table 1.

2.1.2 State Transition Table

S	Q_1	Q_2	Q_3	S*	Q_1*	Q_2*	Q_3*
A	0	0	0	B	0	0	1
B	0	0	1	C	0	1	1
D	0	1	0	E	1	1	0
C	0	1	1	D	0	1	0
H	1	0	0	A	0	0	0
G	1	0	1	H	1	0	0
E	1	1	0	F	1	1	1
F	1	1	1	G	1	0	1

Table 2: The state transition table which defines the transition of one state to another in terms of Q.

The state-transition table is outlined in Table 2 in accordance with the MTRX1705 Style Requirements.

2.1.3 Derivation of next-state logic

Given the current state, we can define the logic for the next state. This will be done using Karnaugh maps to find a simple expression for each bit.

Q_0*:

Q_0*:		Q_1/Q_2 00	01	11	10
Q_0	0	0	0	0	1
	1	0	1	1	1

$\therefore Q_0^* = Q_0Q_2 + Q_0Q_1 + Q_1Q_2'$

Figure 3: The Karnaugh map and derived equation for Q_0.

$$\begin{aligned} Q_0* &= Q_0 \cdot Q_2 + Q_0 \cdot Q_1 + Q_1 \cdot Q_2' \\ &= Q_0(Q_1 + Q_2) + Q_1 \cdot Q_2' \\ &= Q_0(Q_1' \cdot Q_2')' + Q_1 \cdot Q_2' \\ &= ((Q_0(Q_1' \cdot Q_2')')' \cdot (Q_1Q_2')')' \end{aligned}$$

Q_1*:

Q_1:*		Q_1/Q_2 00	01	11	10
Q_0	0	0	1	1	1
	1	0	0	0	1

$\therefore Q_1^* = Q_0'Q_2 + Q_0'Q_1 + Q_1Q_2'$

Figure 4: The Karnaugh map and derived equation for Q_1.

$$\begin{aligned} Q_1* &= Q_0' \cdot Q_2 + Q_0' \cdot Q_1 + Q_1 \cdot Q_2' \\ &= Q_0'(Q_1 + Q_2) + Q_1 \cdot Q_2' \\ &= Q_0'(Q_1' \cdot Q_2')' + Q_1 \cdot Q_2' \\ &= ((Q_0'(Q_1' \cdot Q_2')')' \cdot (Q_1 Q_2')')' \end{aligned}$$

Q_2***:**

		Q_1/Q_2			
Q_2*:		00	01	11	10
Q_0	0	1	1	0	0
	1	0	0	1	1

$\therefore Q_2* = Q_0'Q_1' + Q_0Q_1$

Figure 5: The Karnaugh map and derived equation for Q_2.

$$\begin{aligned} Q_2* &= Q_0'Q_1' + Q_0Q_1 \\ &= ((Q_0'Q_1')' \cdot (Q_0'Q_1')')' \end{aligned}$$

2.1.4 Next State Logic Circuit Diagram

Given the current state, we now know what the next state should be:

$$Q_0* = ((Q_0(Q_1' \cdot Q_2')')' \cdot (Q_1Q_2')')' \quad (1)$$
$$Q_1* = ((Q_0'(Q_1' \cdot Q_2')')' \cdot (Q_1Q_2')')' \quad (2)$$
$$Q_2* = ((Q_0'Q_1')' \cdot (Q_0'Q_1')')' \quad (3)$$

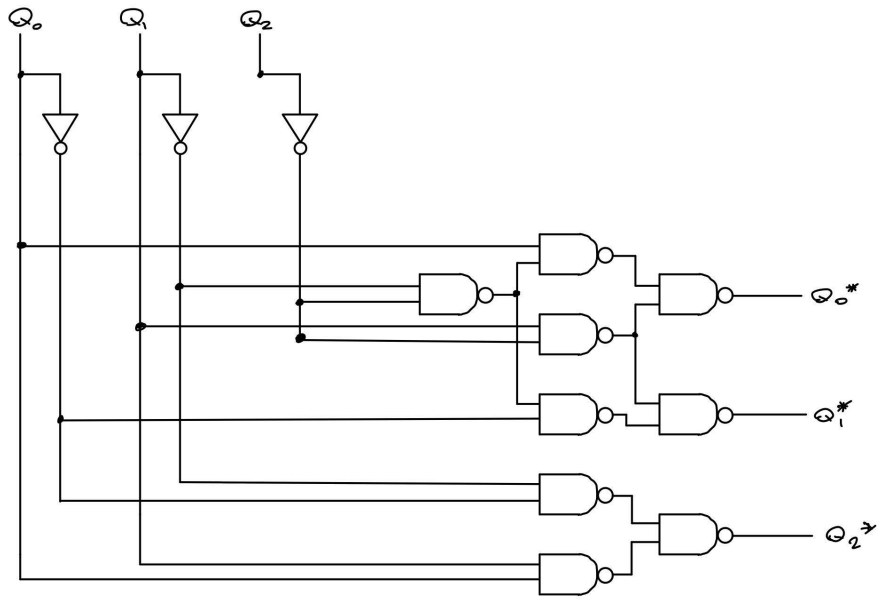

Figure 6: The next state logic diagram for the state machine.

2.1.5 Next State Logic Circuit Testing

In order to determine this module functional, it has to satisfy the following requirements:

1. Given the inputs representing the current state S (Q_0, Q_1, Q_2), the circuit outputs the logic for the next state S* (Q_0*, Q_1*, Q_2*), as defined in Table 2.

In the testing of this circuit, the inputs Q_0, Q_1 and Q_2 were represented by a pull down button, and the outputs Q_0*, Q_1*, Q_2* were represented by LED's.

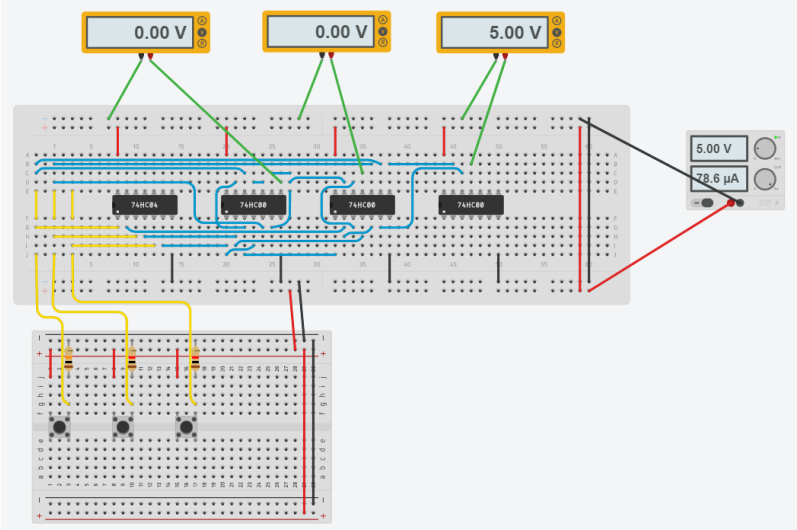

Figure 7: A tinkercad simulation of the next state logic circuit.

Firstly, the circuit layout was mapped out and simulated in tinkercad, as seen above in Figure 7. This was an efficient way of mapping out the circuit and performing an initial test of the design to validate the calculations of the next state equations, ensuring it returned the expected defined values from Table 2.

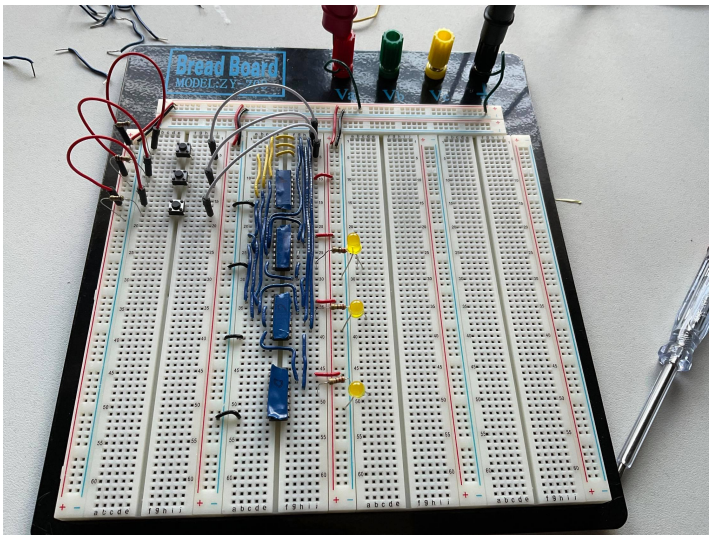

Figure 8: Photograph of the next state logic circuit physically wired up.

This circuit was then wired together physically, as seen in Figure 8, and each state was tested to validate that they output the expected voltages as defined in Table 3. This was done visually

	Input Voltages				Output Voltages		
S	Q_0	Q_1	Q_2	S*	Q_0*	Q_1*	Q_2*
A	0	0	0	B	0	0	5
B	0	0	5	C	0	5	5
D	0	5	0	E	5	5	0
C	0	5	5	D	0	5	0
H	5	0	0	A	0	0	0
G	5	0	5	H	5	0	0
E	5	5	0	F	5	5	5
F	5	5	5	G	5	0	5

Table 3: Table representing the inputs and expected output voltages of the Next State Logic Circuit.

through the use of LED's, as the use of a multi meter given the range of outputs was impractical. Upon a successful completion of testing, the requirements were satisfied, meaning that the module was deemed functional.

2.2 State Memory

2.2.1 Module Design

The memory component of the state machine will be updated when the clock triggers the change. When this occurs, the 'next state' $Q*$ becomes the current state Q.

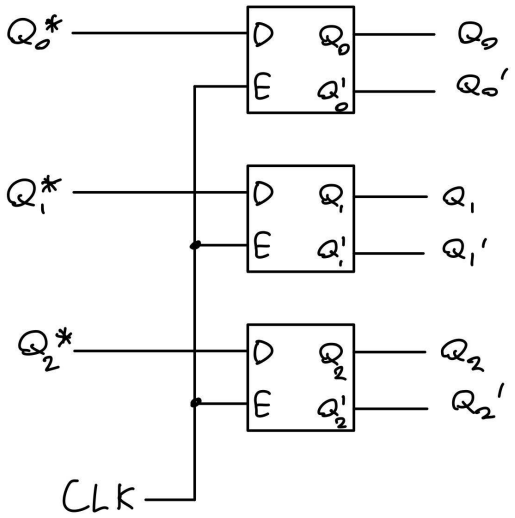

Figure 9: The circuit diagram for the State Machine's State Memory.

As seen in Figure 9, the state machine utilises an SN74LS175 Quad-D type flip-flop for the state memory, with the inputs being the next state, and the outputs being the current state.

2.2.2 State Memory Testing Circuit Testing

In order to determine this module functional, it has to satisfy the following requirements:

1. Given the inputs representing the next state S* (Q_0*, Q_1*, Q_2*), the circuit outputs the logic for the current state S (Q_0, Q_1, Q_2), as defined in Figure 9.

2. This change in output will only occur when the clock signal logic is set to HIGH.

In the testing of this circuit, the inputs Q_0*, Q_1*, Q_2* and CLK were represented by a pull down button, and the outputs Q_0, Q_1 and Q_2 were represented by LED's.

Upon testing, the circuit worked as specified, and hence, the module was deemed functional.

2.3 Clock Signal

2.3.1 Module Design

The Clock Signal modular design requires that the frequency can be altered for both high frequency (during use) and low frequency (during demonstration), and should be able to take an input that enables/disables the clock signal.

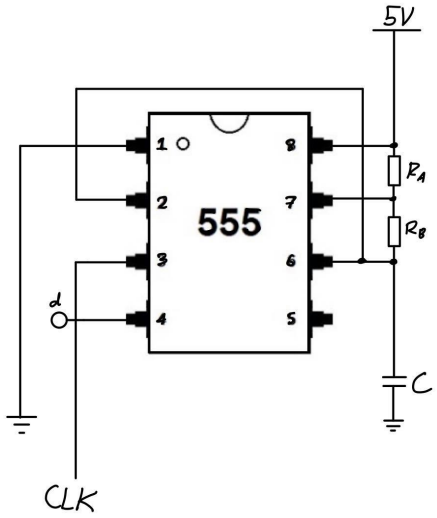

Figure 10: The circuit diagram for the clock signal.

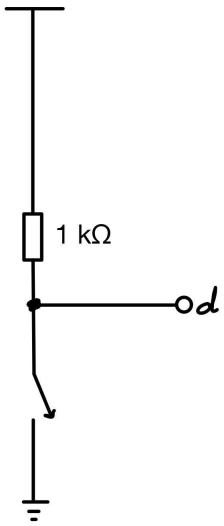

Figure 11: The circuit diagram for a pull-up resistor.

As seen in Figure 10, the clock enable/disable switch is represented by a button utilising a pull-up resistor seen in Figure 11, to allow for user input to stop the timer, which has an overall effect of stopping transitions.

Of the available capacitors, a value of

$$C = 100\mu F$$

was best suited for the circuit. Hence, for an adjustable frequency , the resistor values in R_A are

$$0 <= R_A <= 10000\Omega$$

and for a frequency of 60Hz (faster than the eye can see),

$$
\begin{aligned}
60 &<= \frac{1.44}{(R_A + 2R_b)} \\
R_B &<= \frac{1}{120}(\frac{1.44}{C} - R_A) \\
&<= \frac{1}{120}(\frac{1.44}{0.0001} - 0) \\
&<= 120
\end{aligned}
$$

So to be able to have a range of frequencies that both allow the circuit to swap faster than the eye can see, and to allow for it to change slow enough to notice the bias

$$R_B = 47\Omega$$

2.3.2 Clock Signal Circuit Testing

In order to determine this module functional, it has to satisfy the following requirements:

1. Given the inputs representing the enable/disable switch d and the adjustable resistor R_A, the circuit outputs a clock signal with an adjustable timing in Hz, as defined in Figure 10.

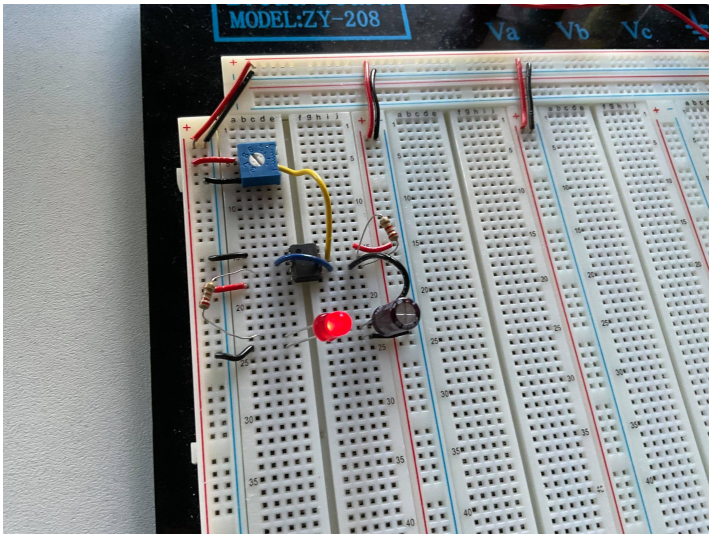

Figure 12: The physical wire up of the clock-circuit module.

In the testing of this circuit, the input d was represented by a pull-up button, R_A with an adjustable resistor and the output CLK was represented by an LED as seen in Figure 12.

The module was tested visually through the use of LED's to validate that it output the clock signal at the desired range of frequencies, and that the disable button worked as expected. Upon a successful completion of testing, the requirements were satisfied, meaning that the module was deemed functional.

2.4 Output Logic

2.4.1 Definition of states

To represent eight different states (S), the bias enable/disable input (D) and the coin flip output, we need one bit to represent the coin face, (Coin Face).

State	D	Q	Coin Face
A	0	000	H
B	0	001	H
C	0	011	T
D	0	010	T
E	0	110	T
F	0	111	T
G	0	101	T
H	0	100	T
A	1	000	H
B	1	001	H
C	1	011	T
D	1	010	T
E	1	110	H
F	1	111	H
G	1	101	T
H	1	100	T

Table 4: The table defining the states and their desired output based on the given bias.

The output logic has been defined in a way such that there is minimal bit change between the coins, to achieve the simplest next state logic circuit possible as seen in Table 4.

Inputs				7 Segment Display Outputs							
D	Q_0	Q_1	Q_2	a	b	c	d	e	f	g	DP
0	0	0	0	0	0	1	0	1	1	1	0
0	0	0	1	0	0	1	0	1	1	1	0
0	0	1	0	0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	1	1	1	1	0
0	1	0	0	0	0	0	1	1	1	1	0
0	1	0	1	0	0	0	1	1	1	1	0
0	1	1	0	0	0	0	1	1	1	1	0
0	1	1	1	0	0	0	1	1	1	1	0
1	0	0	0	0	0	1	0	1	1	1	0
1	0	0	1	0	0	1	0	1	1	1	0
1	0	1	0	0	0	0	1	1	1	1	0
1	0	1	1	0	0	0	1	1	1	1	0
1	1	0	0	0	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	1	1	0
1	1	1	0	0	0	0	1	1	1	1	0
1	1	1	1	0	0	0	1	1	1	1	0

Table 5: The table defining the desired output based on the given inputs.

The output logic table is outlined in Table 5, in accordance with the MTRX1705 Style Requirements. It refers to Figure 17 which defined the pins to their corresponding LED location.

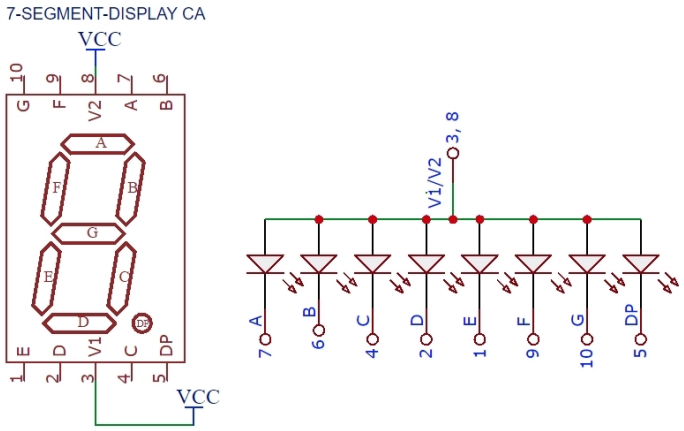

Figure 13: The documentation outlining the 7-segment LED's and their corresponding pins.

2.4.2 Derivation of output logic

Given the current state, we can define the logic for the next state. It is observed in Table 5 that a number are constants, and that c and d require logic to function. Accordingly:

a :

$$a = 0$$

b :

$$b = 0$$

c :

c:		Q_1/Q_2				
		00	01	11	10	
D/Q_0	00	1	1	0	0	
	01	0	0	0	0	
	11	1	1	0	0	
	10	1	1	0	0	$\therefore c = DQ_1' + Q_0'Q_1'$

Figure 14: The Karnaugh Map corresponding to the output c and its derived equation.

$$\begin{aligned} c &= DQ_1' + Q_0'Q_1' \\ &= Q_1'(D + Q_0) \\ &= Q_1'(D' \cdot Q_0)' \\ &= ((Q_1'(D' \cdot Q_0)')')' \\ &= d' \end{aligned}$$

d :

d:		Q_1/Q_2			
		00	01	11	10
D/Q_0	00	0	0	1	1
	01	1	1	1	1
	11	0	0	1	1
	10	0	0	1	1

$\therefore d = Q_1 + d'Q_0$

Figure 15: The Karnaugh Map corresponding to the output d and its derived equation.

$$\begin{aligned} d &= Q_1 + D'Q_0 \\ &= (Q_1'(D' \cdot Q_0)')' \end{aligned}$$

e :

$$e = 1$$

f :

$$f = 1$$

g :

$$g = 1$$

DP :

$$DP = 0$$

2.4.3 Output Logic Circuit Diagram

Given the current state, we know what the output should be:

$$a = 0 \tag{4}$$
$$b = 0 \tag{5}$$
$$c = d' \tag{6}$$
$$d = (Q_1'(D' \cdot Q_0)')' \tag{7}$$
$$e = 1 \tag{8}$$
$$f = 1 \tag{9}$$
$$g = 1 \tag{10}$$
$$DP = 0 \tag{11}$$
$$\tag{12}$$

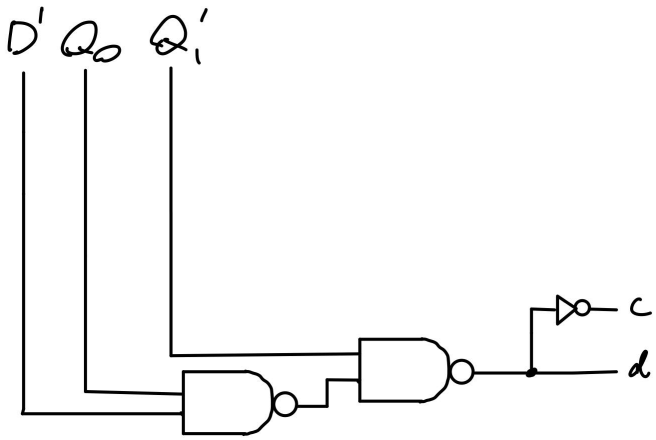

Figure 16: The output logic diagram for the state machine.

2.4.4 Output Logic Circuit Testing

In order to determine this module functional, it has to satisfy the following requirements:

1. Given the inputs representing the bias enable/disable switch (D) and current state S (Q_0, Q_1, Q_2), the circuit outputs the output logic (a, b, c, d, e, f, g, DP), as defined in Figure 5.

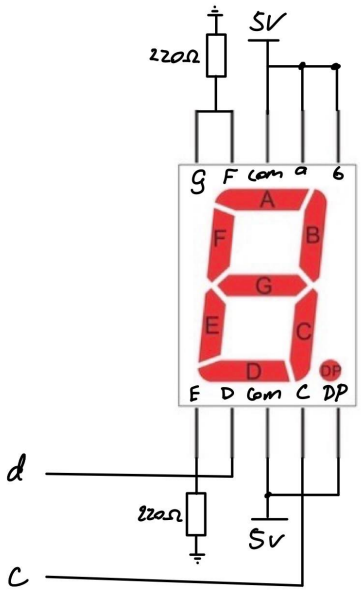

Figure 17: The output circuit for the 7-segment display.

In the testing of this circuit, the inputs Q_0, Q_1 and Q_2 were represented by a pull down button, and the outputs were represented by a 7-segment display as seen in Figure 17.

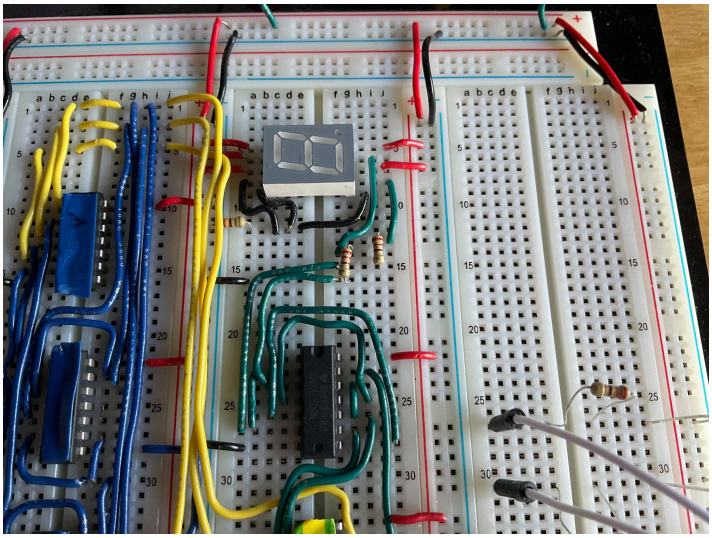

Figure 18: The physical circuit for the Next State Machine.

This circuit was wired together physically, as seen in Figure 18, and each state was tested visually using the 7-segment display to validate that they have the correct outputs as defined in

Table 5. Upon a successful completion of testing, the requirements were satisfied, meaning that the module was deemed functional.

3 Integrated Circuit

3.1 Integrated Circuit Design

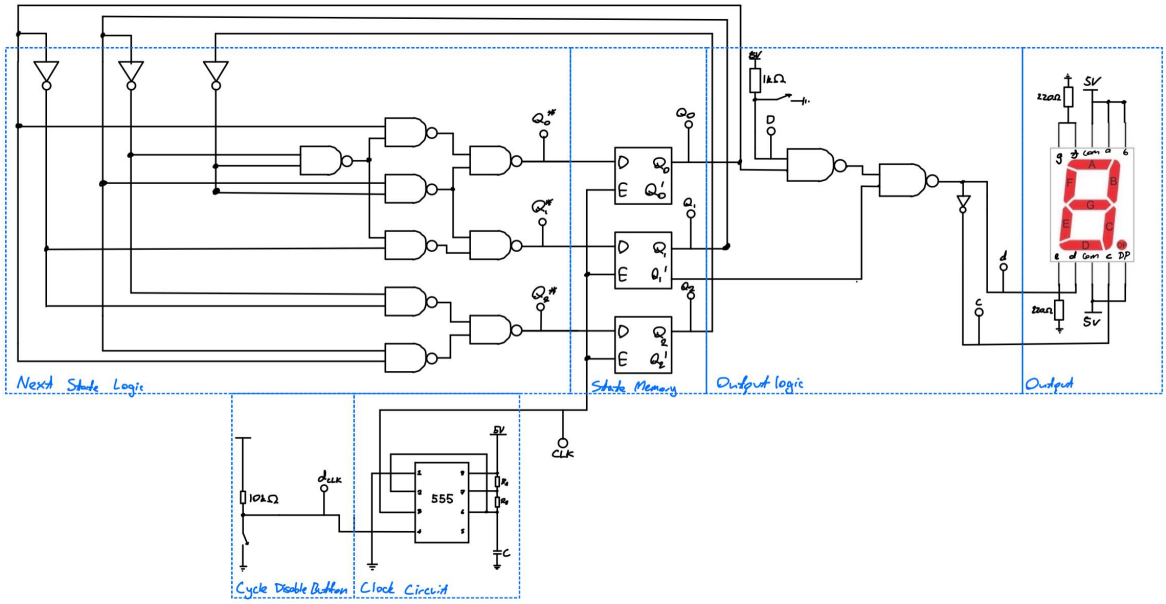

Figure 19: The integrated circuit design for the State Machine, with defined voltage testing locations for each variable.

Integrating all of our tested and functional modules from section 2, an integrated circuit design can be observed in Figure 19. This design has defined voltage testing locations for each variable to allow for tests to occur.

3.1.1 Integrated Circuit Build

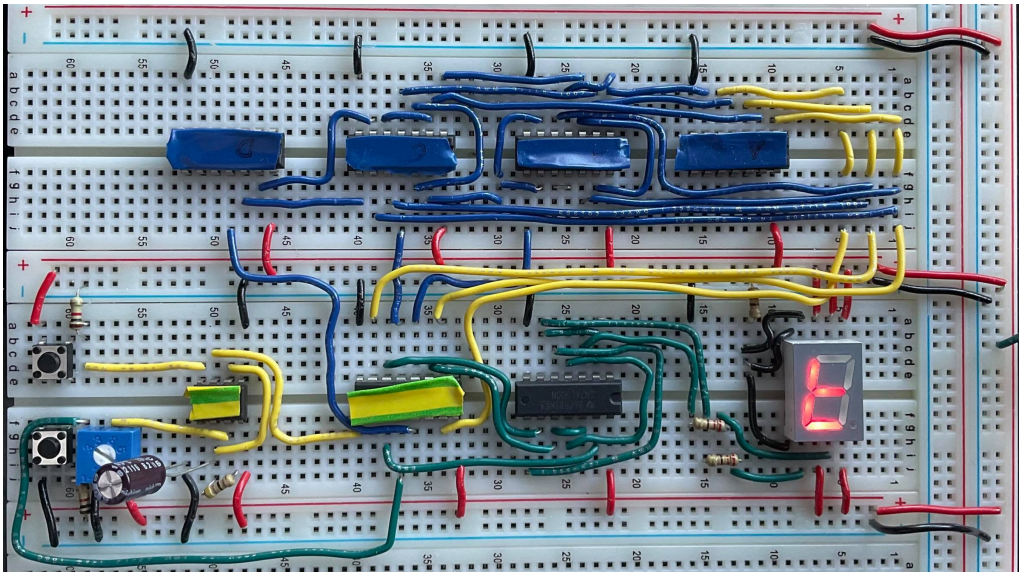

Figure 20: The integrated circuit for the State Machine.

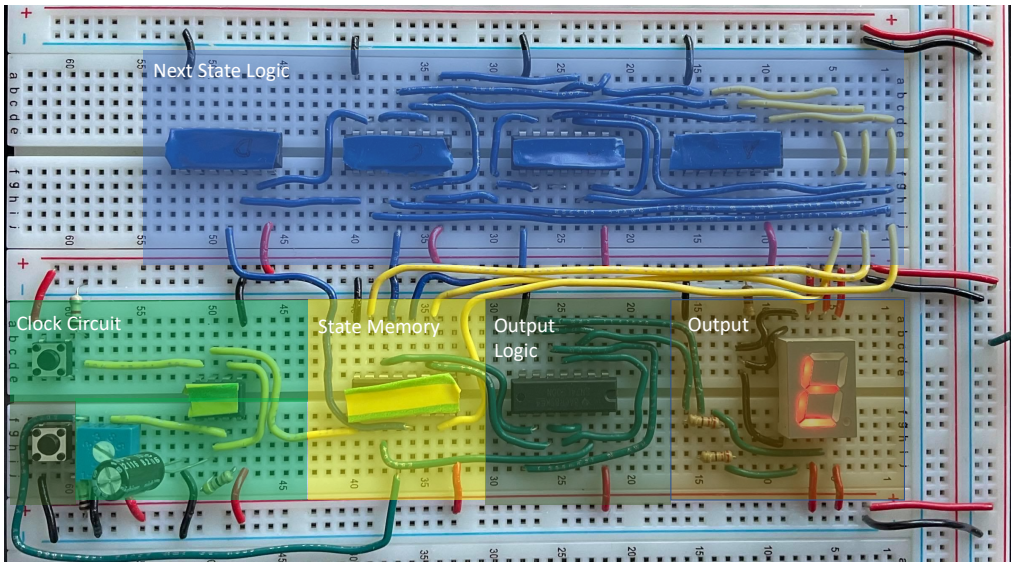

Figure 21: The integrated circuit for the State Machine, with the position of each module labelled.

As seen in Figure 20 and 21, the physical build was laid out in a way such that it could be easily accessible to the user, and that it would be easy to debug.

3.1.2 Integrated Circuit Testing

In order to test that each module has integrated successfully, the voltages were determined using the defined voltage testing locations in Figure 19. Upon testing each location using both a multi-meter and LED's to ensure they output the expected value as defined in each module, the integration test was deemed successful and the circuit was deemed functional.

4 System Testing

As defined in section 1, this State Machine is deemed functional if it is able to produce the specified output behaviour as seen in Table 5 based on two user inputs:

1. A button to stop the transition of the states (pressed = cycling, released = no cycling).
2. A button to change the output from a bias output to a fair output (pressed = fair, released = bias).

Thus, upon testing of the system solely based on the use of the defined inputs (buttons) and outputs (7-segment LED's), it was found that it satisfied these two key requirements according to Table 5, and thus, is a functional circuit that has achieved it's specifications.

Conclusions

The state machine is comprised of five key modules:

1. Next State Logic
2. State Memory
3. Clock Signal
4. Output Logic
5. Output Display

After integration and testing, it was found that it satisfied its defined key requirements, and was deemed functional.