MTRX 1705

Introduction to Mechatronic Design

Assignment 2

Train Controller State Machine

Report

Submission Date: 21/05/2017

Authors

Vishant Prasad (SID: 470416309)

Manan Vora (SID: 470318966)

Beau Hobba (SID: 470213225)

TABLE OF CONTENTS

Executive Summary

This report covers the design and implementation of a state machine to control a train. This state machine takes inputs from a train controller, and sends outputs back to the train controller.

Firstly, a state machine was designed to determine the location of the train, dependant on three sections. After this, the outputs of the state machine were used to design an output logic for speed, and changing the direction of travel from outer to inner loops.

Over the course of this design, various tests were used to ensure the capability of the machine, such as encoders.

All circuits and design must be based only on 74LS series logic ICs due to their robustness and reliability.

Introduction

Subject Matter

The following report tackles the issue of a small city plagues with human-related errors. That is, the necessity of the introduction of efficient train networks. This report will cover the design and implementation of an efficient state machine to control a train network. This automation concept will be carried out using a model representing the rail system.

Overview

The assumptions made over this report are:

- The train can travel from track B to track C and vice versa.
- The circuit to determine the anticlockwise and clockwise direction need not be implemented.
- The circuit to determine the speed of the train is based on the weather and train type.
- There are two train types; express and freight.
- The weather affects track C and requires the train to travel through the track B or inner loop if the weather is bad.
- When the weather is bad, the speed on track C is set to low and the tracks are adjusted for the inner loop.
- The inner loop has a medium speed and track A has a fast speed.
- If the train is detected in the inner loop, by default it should be directed to the inner loop again. The same statement is true for the outer loop.
- Track C has a fast speed if weather is fine.
- Track C is always preferred.

Initial Notes

1. **Build State Machine to find the Location of the Train**
    a. Inputs
        i. T0 T1 T2
    b. Outputs
        i. Binary Outputs 0 and 1
            1. 00 = Unknown
            2. 01 = A
            3. 10 = B
            4. 11 = C

2. **Build State Machine to set the Speed of the Train**
    a. Inputs
        i. Binary Outputs from Previous State Machine
        ii. Temperature - [if T<0 (Use track B)]
        iii. Express or Freight
    b. Outputs
        i. Binary Output Speed
            1. 00 - Stopped
            2. 01 - Slow
            3. 10 - Medium
            4. 11 - Fast
3. **Direction Design**

Circuit Design Plan:

Vcc - Red

Ground - Black

Location State Machine:

Colours of Inputs

T0 - Sensor 1

T1 - Sensor 2

T2 - Sensor 3

Colours of Outputs

Q0 - Sensor Output 1

Q1 - Sensor Output 2

Speed Output

Inputs

Q0 - Sensor Output 1

Q1 - Sensor Output 2

T - Temperature

E - Train Type

Outputs

sp0 - Speed Output 1

sp1 - Speed Output 2

Design of State Machine

The state machine generates two binary variables that determines the location of the train. The location of the train can be defined in four states:

- Unallocated = U = {Q0 Q1} = {00}
- Section A = A = {Q0 Q1} = {01}
- Section B = B = {Q0 Q1} = {10}
- Section C = C = {Q0 Q1) = {11}

As there are two outputs, two D flip flops are required.

The current state is defined as Q0 and Q1, whereas the next state is defined as Q0* and Q1*.

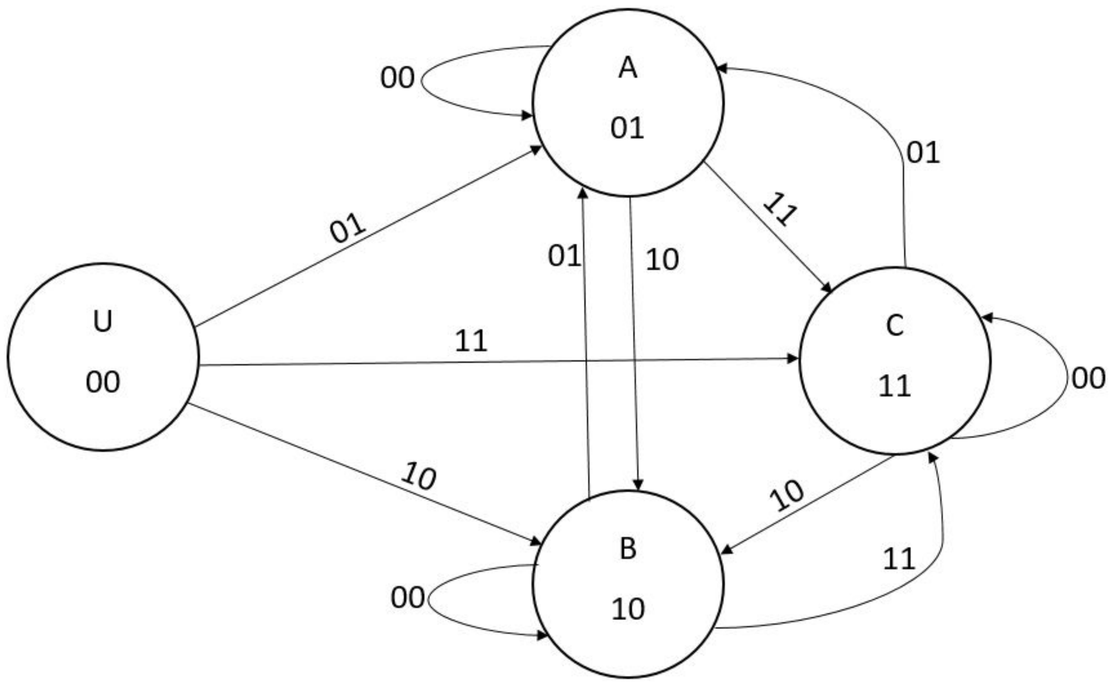
The state machine requires only two sensor statements to determine the location of the train. Table 1 below shows the truth table for these sensors.

T0	**T1**	**T2**	**Train Detector State**	**Train Detector Location**
0	0	0	No Sensor	No Sensor
0	0	1	No Sensor	
0	1	0	Sensor a0	Sensor A
0	1	1	Sensor a1	
1	0	0	Sensor b0	Sensor B
1	0	1	Sensor b1	
1	1	0	Sensor c0	Sensor C
1	1	1	Sensor C1	

Thus, only the sensors at the beginning of each track (i.e a0, b0 and c0) will be used to determine the next state logic.

A state diagram has been generated and shown below:

State Diagram



From this state diagram, the following state table was produced:

State Table

Input				Output	
Q0	**Q1**	**T0**	**T1**	**Q0***	**Q1***
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1

From this, the following truth tables were produced for outputs Q0* and Q1*

Truth Table for Output Q0*

Input				Output
Q0	Q1	T0	T1	Q0*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Truth Table for Output Q1*

Input				Output
Q0	Q1	T0	T1	Q1*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Karnaugh Map and Expression for Q0*

For Q_0^*

T_0T_1 \ Q_0Q_1	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	0	1	1
10	1	0	1	1

$$Q_0^* = \underbrace{T_0}_{x} + \underbrace{T_1' \cdot Q_0}_{y}$$

Using: $(x' \cdot y')'$ to convert to NAND

Hence: $(T_0' \cdot (T_1' \cdot Q_0)')'$

Karnaugh Map and Expression for Q1*

For Q_1^*

$Q_0Q_1 \backslash T_0T_1$	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$$Q_1^* = T_1 + (T_0' \cdot Q_1)$$

Subbing into NAND Gates:

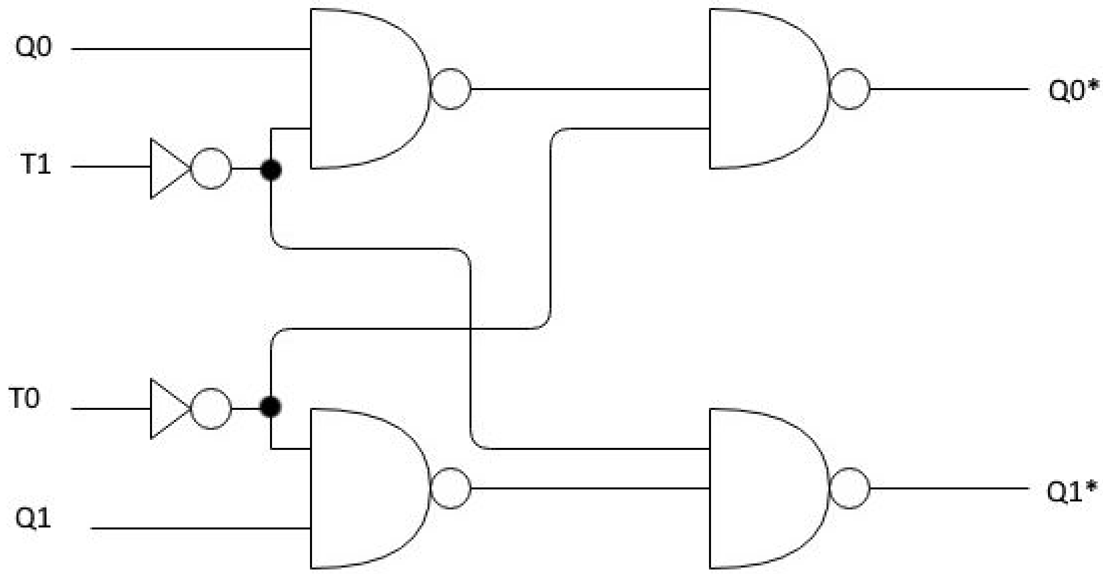
$$= (T_1' \cdot (T_0' \cdot Q_1)')'$$

Next State Logic

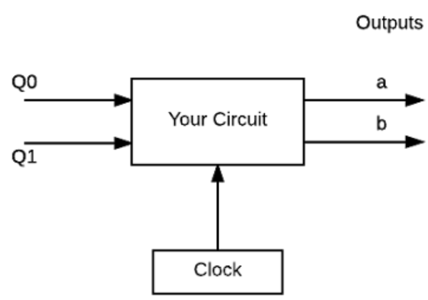
Q0* = (T0' · (T1' · Q0)')'

Q1* = (T1' · (T0' · Q1)')'

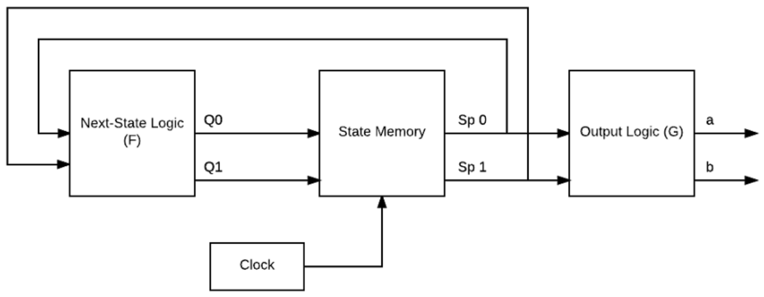
Using the expressions, a next state logic was made.



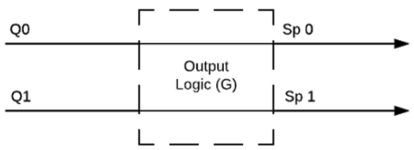
Black Box Circuit



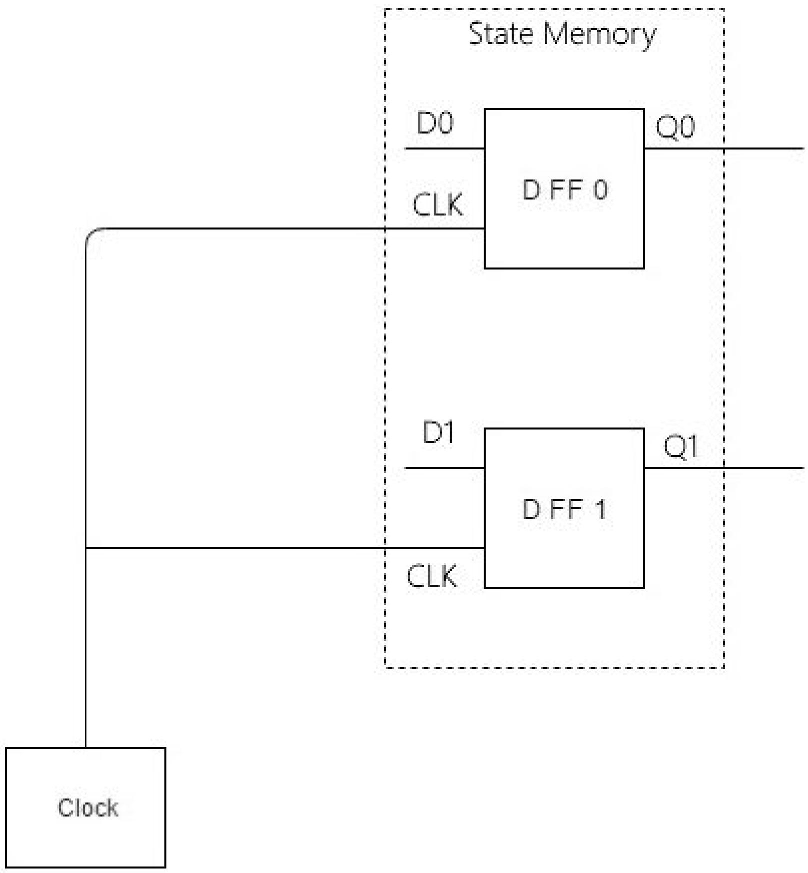
Circuit Plan Diagram



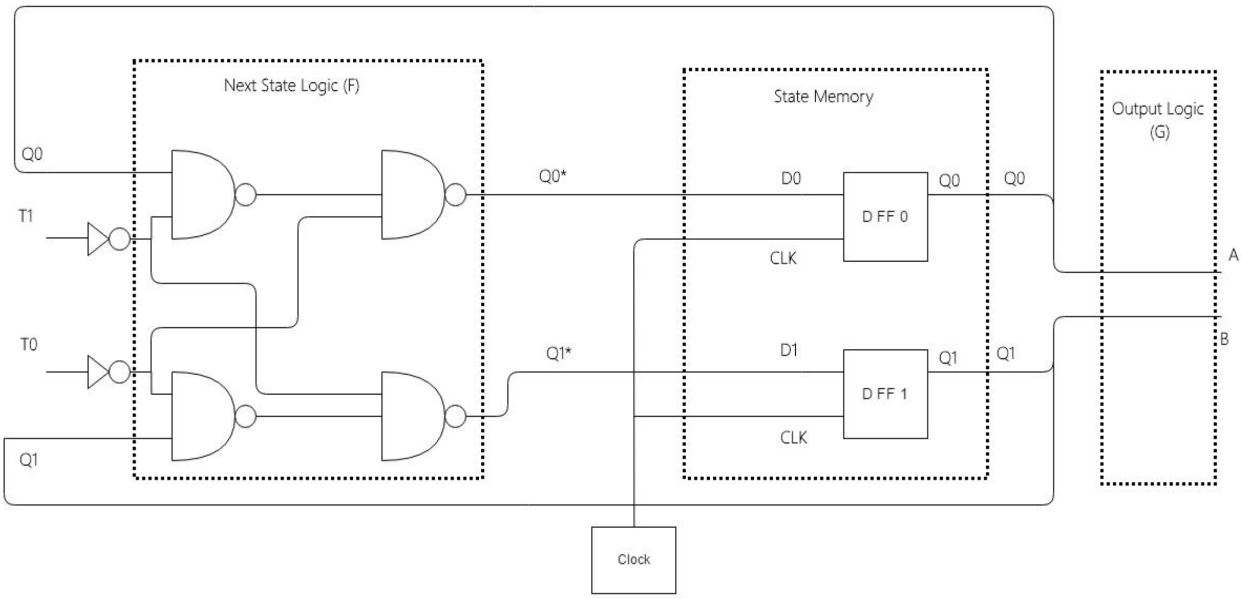
Output Logic



State Memory



System Diagram



Combinational logic for Dir output

This direction logic corresponds to the outer / inner loop direction change depending on the weather and train type.

Truth Table

Input				Output
Q0	**Q1**	**E**	**Z**	**Dir**
0	0	0	0	X
0	0	0	1	X
0	0	1	0	X
0	0	1	1	X
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Karnaugh Map and Expression for Output Dir

Direction:

Q_0Q_1 \ EZ	00	01	11	10
00	X	X	X	X
01	1	1	1	0
11	0	0	0	0
10	1	1	1	1

As X is undefined, 1 or 0. Assume X is a 1,

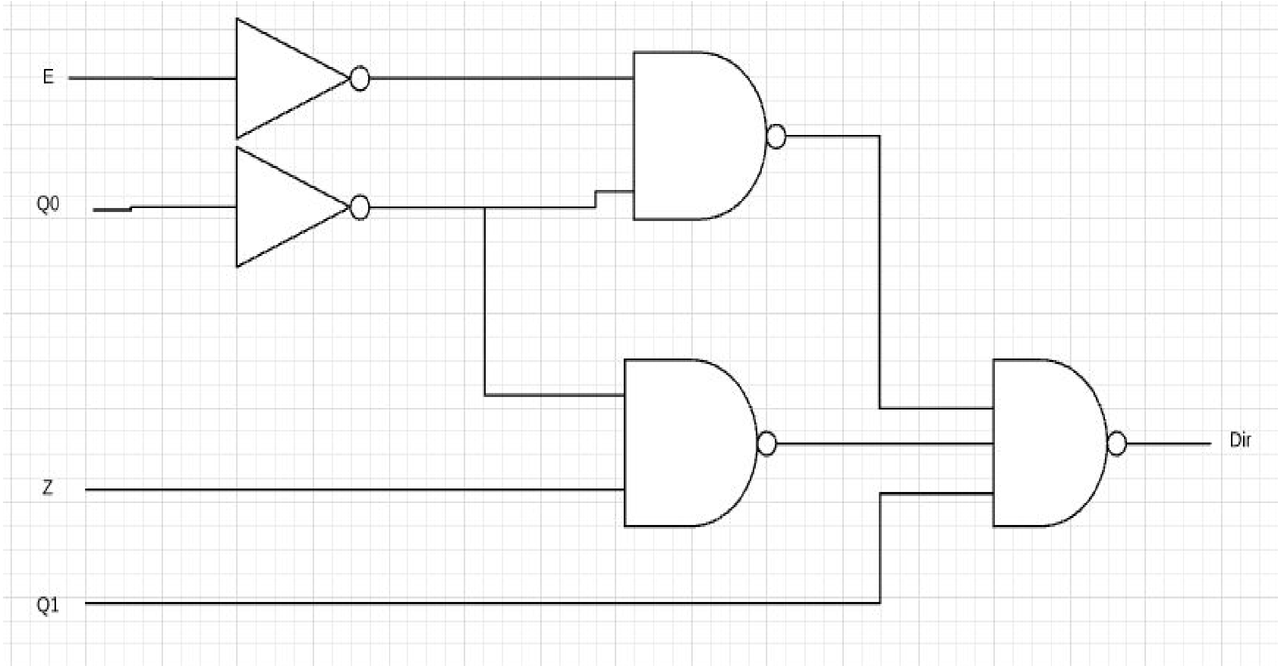
hence: $Q_1 + (E' \cdot Q_0') + (Z \cdot Q_0')$

Subbing into NAND Gates

$((Q_1) \cdot (E' \cdot Q_0')' \cdot (Z \cdot Q_0')')'$

<u>Output Logic for Output Dir</u>
The logic expression is:

Dir = (Q1 · (E' · Q0')' · (Z · Q0')')'



Combinational Logic for Speed Output

The following section describes the design and implementation of combinational logic to determine the speed of the train.

The output from this logic is fed back into the train network controller.

- Inputs
    - Binary Outputs from State Machine
    - Temperature (Z)
    - Train Type (E)
- Outputs
    - Binary Output Speed
        - 00 – Stopped
        - 01 – Slow
        - 10 – Medium
        - 11 – Fast

Truth Table for Temperature

State	Meaning
0	Temperature is above 0
1	Temperature is below 0

Track B should be used if temperature is below 0, otherwise track C has preference.

Truth Table for Speed Output

Input				Output	
Q0	**Q1**	**E**	**Z**	**Sp0**	**Sp1**
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	0	1

Karnaugh Map and Expression for Sp0

Sp0

$E2$ / Q_0Q_1	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	0	0	1
10	1	1	1	1

Hence: $(Q_0' \cdot Q_1) + (Q_0 \cdot Q_1') + 2' \cdot Q_0$

(subbing into NANDs)

$Sp0 = ((2 \cdot Q_0)' \cdot (Q_0 \cdot Q_1')' \cdot (Q_0' \cdot Q_1)')'$

Karnaugh Map and Expression for Sp1

Sp1

E2 / Q_0Q_1	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	0	1	1	1
10	0	0	0	0

$$Sp_1 = (E \cdot Q_1) + (Q_1' \cdot Q_0') + (Z Q_1 \cdot Q_0)$$

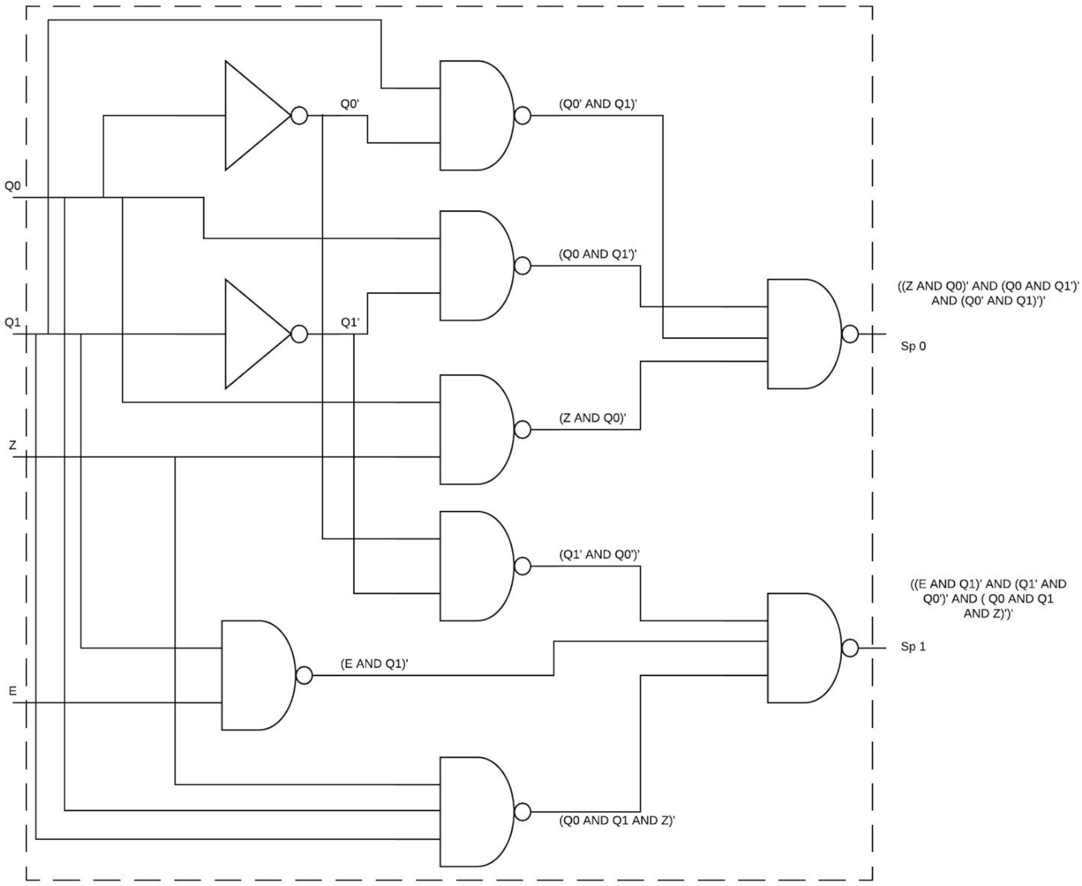
Subbing into NAND Gates:

$$((E \cdot Q_1)' \cdot (Q_1' \cdot Q_0') \cdot (Q_0 \cdot Q_1 \cdot Z)')'$$

Logic Diagram for Speed Output

$$sp0 = ((z.Q0)'.(Q0.Q1')'.(Q0'.Q1)')'$$

$$sp1 = ((E.Q1)'.(Q1'.Q0')'.(Q0.Q1.Z)')'$$



Design of Rotational Direction

The rotational direction was asked to be designed but not implemented.

This section covers the design of a state machine for determining whether the train is going clockwise or anticlockwise.

Hence, this is dependent upon the three sensors. T0 and T1 determine location and T2 determines Direction. Hence:
If the train is in the Green Zone, T2 will either read as 0 or 1. If it is zero that means the train is moving clockwise. If a one is read the train in moving anticlockwise.

State Table (Rotation):

T0 and T1 are used to find Location, T2 is used to find rotation (clockwise or anti clockwise):

T0	T1	T2	Train Detector State
0	0	0	No Sensor
0	0	1	No Sensor
0	1	0	Sensor a0
0	1	1	Sensor a1
1	0	0	Sensor b0
1	0	1	Sensor b1
1	1	0	Sensor c0
1	1	1	Sensor C1

Hence it can be confirmed, that if sensor a0 is detected first it is obviously clockwise direction.
If a1 is detected first it can be noticed the direction will be anti clockwise.

Using the fact that:
Outer Loop --> A
Inner Loop --> A
And that it is an error state (undefined state) if:
Outer Loop --> Inner Loop
Inner Loop --> Outer Loop

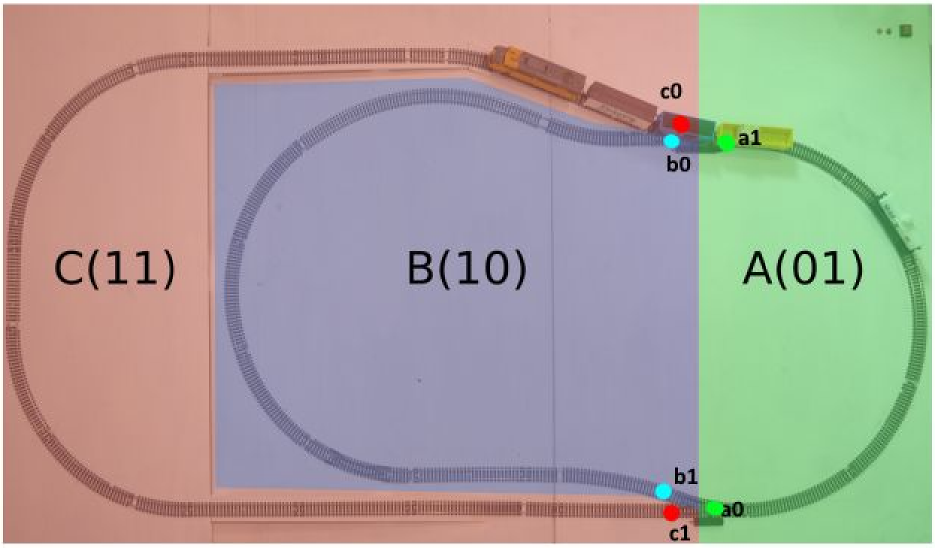
Q0	Q1	D	Q0*	Q1*
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Always enter the loop of A(01). This can be overlaid onto the whole train station and stated that rotation does not affect the path the train takes. Hence T2 just states which rotation the train takes and does not directly influence the direction.
Undefined is the value which results if the train is in A(01), as it can either enter the inner or outer loop. However due to the trains definite state out of the inner/outer loop it can concluded that rotation plays no major part.
Hence to find the direction, the variables of train type and weather must be taken into account.
Certain stipulations are made:
Freight Train cannot go fast, hence it will always prefer the inner loop (As 209m of mid speed is much better than 304m). Thus, as no conditions surround the inner loop it can be concluded that the Freight train will usually never change its direction of the inner loop.
Express train prefers the outer loop when the weather is icy. Prefers the inner loop when weather is not icy. Hence the express train will change its direction when the whether is active.

Using the values of Q0, Q1, E, Z
Using the idea that:
A – B
A – C
B – A
C – A

And any other direction is undefined, and not considered a usual direction choice.

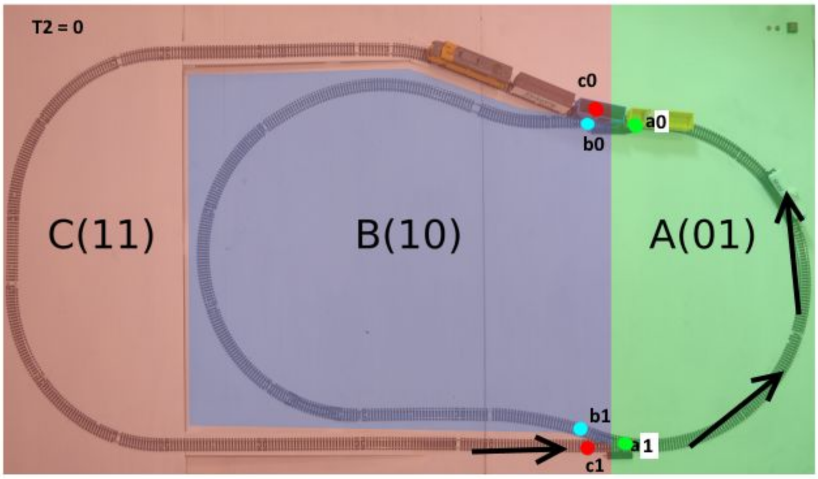
Overall, it can be simply defined through two simple equations.
T2’ = Clockwise
T2 = Anti Clockwise



Direction, also requires rotation.

When the train passes over the sensors, 3 outputs are provided. T0 and T1 determine location, however T2 is crucial in determining rotation. T2 is the input which either determines where sensor(0) or sensor(1) is active. T2 is the binary digit which is determined whether sensor(0) is activated before sensor(1). Hence if this is true = 1, and false = 0. This is showcased below:

T0	T1	T2	Train Detector State	Direction
0	1	0	Sensor a0	Clockwise
0	1	1	Sensor a1	Anticlockwise



It is important, that T2 is only determined after both sensors have gathered measurements. T2 is basically which sensor was activated first. If no second sensor picks up data, then the trains rotational can not be established. Although only one sensor is activate at a time, T0, T1 and T2 depend upon the measurements of 2 sensors. Hence this is showcased in the following table: * = previously active

Sensor(0) Activity	Sensor(1) Activity	T2	Binary
None	None	-	-
A0 active B0 active C0 active	No Previous activity	Requires a measurement between 2 adjacent sensors.	-
No Previous activity	A1 active B1 active C1 active	Requires a measurement between 2 adjacent sensors.	-
A0 active	A1 *	Clockwise	0
A0 *	A1 active	Anticlockwise	1
B0 active	B1 *	Clockwise	0
B0 *	B1 active	Anticlockwise	1
C0 active	C1 *	Clockwise	0
C0 *	C1 active	Anticlockwise	1

Hence it can be observed that, T2 is clockwise when sensor(0) is currently active [Hence the last sensor the train passed over], and sensor(0) has already been active. These two successive measurements are used to determine rotation, as the sensors are located at different areas of the track (As demonstrated in the diagram above). This can be showcased through

Train passes Sensor(0) -> Train passes Sensor(1); Sensor(0)* - Sensor(1) (Sensor 0 is previously inactive and is being measured with the now active sensor 1); Clockwise ; T2 = 1.

Overall, rotation is based on whichever sensor was activated first/last. Rotation does not have any influence on the trains next location. This is due to the track having the same conditions and locations regardless, of which way it is travelling. For example:

The freight train always prefers the inner loop. If it is section A(01) and:

IF T2 = 0 (Clockwise). The train will simply still travel into the B(10), instead of taking the C Path.

Likewise, if T2 = 1 (Anticlockwise), the train will simply travel into B(10) again, due to the track similar paths regardless of rotation.

Testing

Plan

- How do you synthesize the inputs?
  - The inputs are synthesized by connecting the switches as a user input or signal (in this case T1 & T0 & E & Z which which are the train track sensors, temperature & train type inputs).
- How do you visualise the state/outputs?
  - Created a circuit layout and breadboard layout.
  - The state machine is visualised by using, firstly, using diagrams from the logic expressions & truth tables which is implemented onto the breadboard.
  - LED's are used at the end and the clock to encode the signals assisting in visualisation.
  - Outputs (Sp0, Sp1, Dir, Sensors; T0, T1) are visualised by implementation and using logic combination.
- How do you generate a reliable clock signal?
  - Clock signal was generated using a 555 timer with a built in flip-flop.
  - Signal was determined to be operational or tested using a LED encoding to test the output at a moderate frequency (not stable).
  - LED encoder constantly turns off and on displaying the clock to be operational.
  - Clock operates with capacitance hence, temperature change causes an unstable frequency.
- How do you describe to others where the individual signals are in your breadboard?
  - Position and breadboard design.
  - Sourced Green LED (Speed), encoded Red, Blue and encoded clock LED's are sinked.
  - Displayed through colour coded wires.
    - Red = Positive Power
    - Black = Ground
    - White = Output from IC's Logic
    - Blue, Green and Orange are Sensor Outputs.
- How do you confirm the compliance with the specifications?
  - Use LED's to test outputs for all inputs & combinations.
  - Planned all outputs given each input combination.
  - Use switches to simulate each input.
- What is the initial state?
  - Initially the train Dir is outer loop (State 0)
  - Initial state of the machine is undefined (00)
  - Train is directed to the A-C outer loop A(01)-C(11)
  - For safety the initial speed is slow as the sensor maybe undefined states due to errors, damages etc.

Encoders

The machine required a simple way to show its current state, speed and direction.

Location

Using 2 small LEDS, and the logic of the state machine, the current system worked accordingly:

*A black dot represents the OFF LED

Direction:

Using the direction logic and and a blue LED, direction was shown simply by turning the LED on when it is in the INNER LOOP, and off when in the OUTER LOOP.

Speed:

Using 3 LEDS, speed logic and some simple encoding equations, speed was demonstrated accordingly.

S0	S1	Slow	Medium	Fast
0	0	0	0	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Hence,

Fast =

$S0.S1$

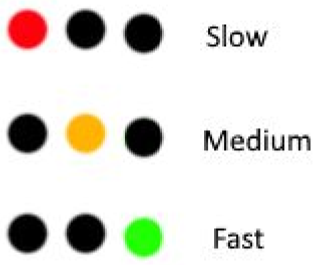
For simplification purpose, instead of using S0 or S1 for medium or low, we use the inverse.

Medium =

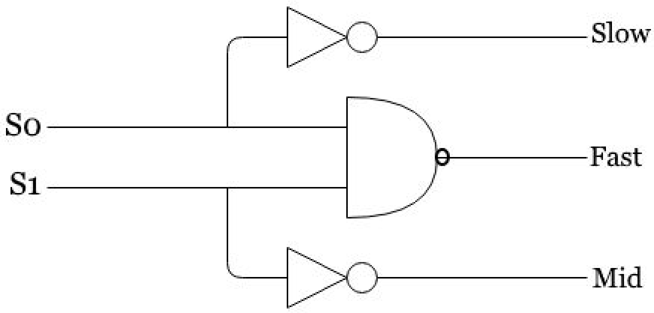
$S1'$

Slow =

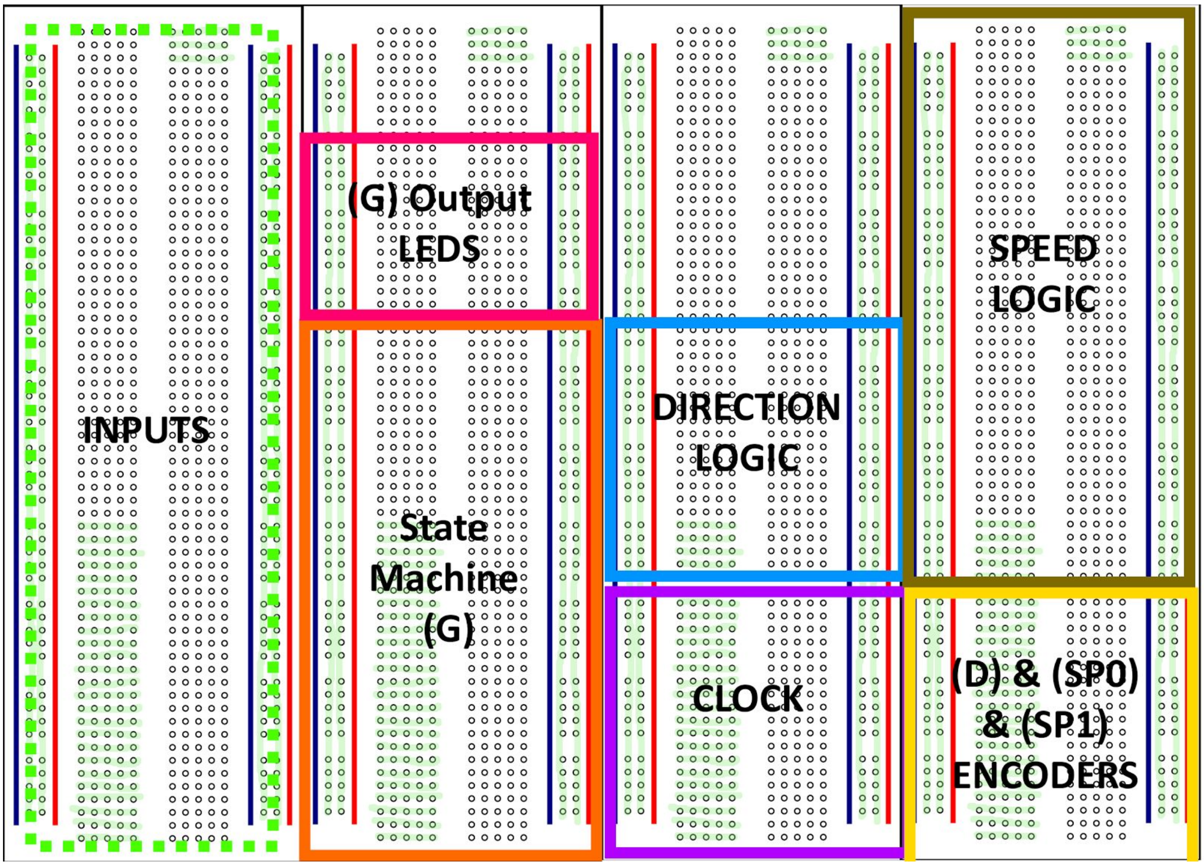
$S0'$



We sink the LED For Fast, so we can use a NAND Gate.



Board Layout



Requirement Specification

The system is required to operate a model train set. Specifically, the speed must be altered based on inputs that simulate bad weather also simulate the type of train is on the tracks. Each of these inputs also must be considered by the system outputting signals to the tracks changing the path the train must take. For example, the inner loop or the outer loop (A-B or A-C respectively). The state machine must be able to determine the location of the train in the network, set the appropriate maximum speed given this location and select the most appropriate track or loop and given the constraints of the weather must be able to select the optimal route based also based on travel time. Another theoretical circuit must be created to determine the direction of the train.

The system is determined to be operational using the state logic connected to LED outputs testing the output state (0 or 1) based on given inputs from buttons. These buttons include the temperature, train type, and sensor inputs. The outputs will simulate and display the loop, speed and direction of the train.

System Design

The tasks of the system includes:

1. Design and construction of the state machine output logic:
    a. Logic design - define the input/outputs and the logic table.
    b. Planning and construction of the state machine output logic.
    c. Demonstrate the functionality of the output logic.
    d. Visualise the logic outputs.
2. Develop a logic circuit that can determine the direction of the train:
    a. System design - define the input/outputs and the mechanism you would use.
3. Design and construction of the state machine:
    a. Next state logic and state memory.
    b. State machine design - defining the states, transition diagram, next state logic)
    c. Planning and Construction of the state machine.
    d. Demonstrate the transitions using a signal to act as the clock.
    e. Visualise the current state

The tasks fit together as the state machine defines the train's position, connecting to the sensors. Each sensor output (Q1 and Q0) defines the train's position which is then used by the logic circuit that determines the speed of the train and the circuit that determines the direction of the train and the loop required to be taken. Visualising the logic outputs and current states allows for greater understanding of the function of the system and testing and both are related in different circuits.

Detailed Design

Each part is built using all defined gates and configurations within the above sections of the report and using three different boards; one for the inputs, one for state logic and the clock (555 timer) and the last for the outputs and any other required logic such as speed. Each circuit is designed separately and all is implemented linking each output to the next circuit as inputs. The inputs of the track sensors is used to generate the outputs of the sensors and these outputs are utilized by both the speed and direction logic circuits.

Detailed Test

The detailed test was carried out by making a planned system diagram for each logic circuit as well as the state machine as a whole and match the configurations. If there were any errors (outlined some in the debugging and issues section of the report) the detailed design would have to be re-evaluated and re-designed. The entire state machine and all its outputs along with the direction output is tested using a multimeter and/or LED's to determine the output states and make sure they match the requirements.

Integration Test

The integration test is accomplished by adding all the logic circuits and testing each on using LED’s or a multimeter to measure the signal outputs and determine the output states. Once all outputs of each logic circuit is operating as expected the entire state machine is connected and a further detailed test is performed.

Acceptance Test

The acceptance test involved connecting the system to a model train system which used the tracks that sent the signals from the machine as voltage that is interpreted by the train setting its speed. Furthermore, computer software was used to test all the states of the machine and all the inputs hence, testing if the correct outputs or desired required outputs would occur guiding the train through the correct loop at the correct speed. A remote control was also used to manually change the states and inputs in the case of an error or an output failing.

Debugging & Issues

The following is an account of things that need to be re-worked or modified before the demonstration:

LED’s enncoded:

- Blue - Direction - Sinked.
- Red - Slow Speed - Sinked.
- Yellow - Medium Speed - Sinked.
- Green - Fast Speed - Sourced.
- Clock LED’s - Sinked.

Reset Button:

- Slow Speed - Outer Loop - Undefined Sensors.
- Working Reset Button connected as an input.

Circuit Design:

- Speed (Sp0) required to be re-designed.
- Direction required to be re-designed.

The following is an account of things that went wrong during the demonstration and how the problems were solved:

Circuit Design:

- LED’s need resistors; installed resistors to the output speed and direction LED’s.

Planning:

- System transition diagram re-designed; failed to define the states on the diagram.

Implementation

All expressions were converted to the same family of IC's. In this case all expressions and logic circuits were converted to only contain NAND gates. This was performed in problem solving as it was the only gates that were available. Furthermore, the triple NAND gate evident on Sp0 was actually implemented using a NAND and an inverter.

Findings

An model train can be automated and able to carry out commands from a digital state machine and corresponding/related logic circuits given particular inputs and coded outputs using logic gates, digital signals and clocks which can be implemented on a larger real world scale given real world factors are considered.

Recommendations

The above design is a successful approach to satisfy such desired requirements therefore, the following system would be recommended to be used for the automation of such train systems along with the minor adjustments and issues debugged as mentioned. Since this system was successful in the acceptance test and carried out all the requirements, it is recommended to be used in a real scenario on a bigger scale, including modifications to size of power and voltage and signal strength however, interference and noise would have to considered when implemented to the real world. Furthermore, other real world factors would have to be considered such as public influence, danger to the environment and other social and ethical issues.

Conclusion

In conclusion, the system operates as planned and satisfies the requirements. This is evident as all state outputs operated as planned and the train was directed to the correct loops at the correct speeds given the type of train and weather. This was further proven in the acceptance test which was a success. Furthermore, the detailed test and integration test revealed minor issues such as resistors on output LED's was required and the transition diagram needed to be reviewed and re-written. Aside from these minor issues, the detailed test was a success as the logic circuits and state machines worked with one another successfully and the correct output states was displayed by the LED's, given differing inputs based on the scenario.