ECE 212 Electronic Design Project 2

Section 602
Anish Nagulavancha

Section 1: State Diagram

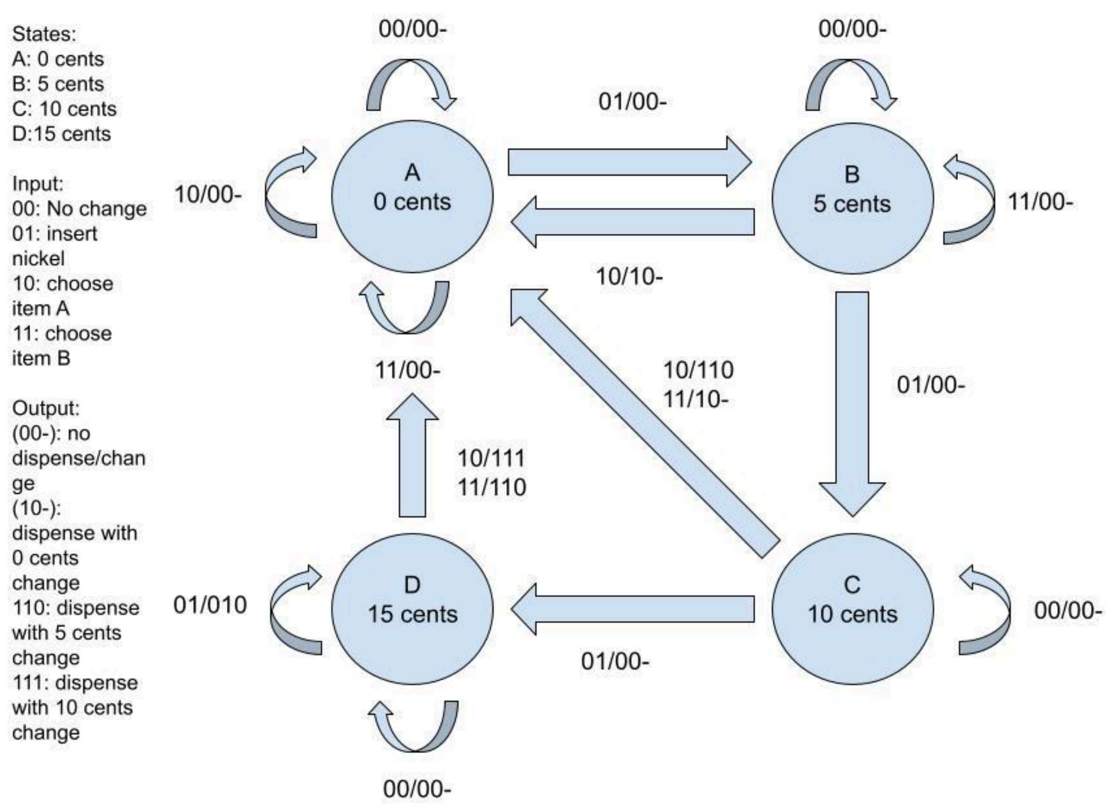


Figure 1: State diagram for vending machine controller

Section 2: Symbolic State Table

X1- input 1
X2- input 2

State	X0X1 = 00	X0X1 = 01	X0X1 = 10	X0X1 = 11
A	A/00-	B/00-	A/00-	A/00-
B	B/00-	C/00-	A/10-	B/00-
C	C/00-	D/00-	A/110	A/10-
D	D/00-	D/010	A/111	A/110

Section 3: State Transition Table

		X0X1 = 00	X0X1 = 01	X0X1 = 10	X0X1 = 11
State	Q1 Q2	Q1* Q2*	Q1* Q2*	Q1* Q2*	Q1* Q2*
A	0 0	00/00-	01/00-	00/00-	00/00-
B	0 1	01/00-	10/00-	00/10-	01/00-
C	1 0	10/00-	11/00-	00/110	00/10-
D	1 1	11/00-	11/010	00/111	00/110

Input logic table:
Q2- Enabler
Q1- 1st selector bit
Q2- 2nd selector bit

Q2	Q1	X1	X0	Q2*	Q1*	
0	0	0	0	0	0	Q2* = 0 Q1* = X0
0	0	0	1	0	1	
0	0	1	0	0	0	Q2* = 0 Q1* = 0
0	0	1	1	0	0	
0	1	0	0	0	1	Q2* = X0

0	1	0	1	1	0	Q1* = X0’
0	1	1	0	0	0	Q2* = 0 Q1* = X0
0	1	1	1	0	1	
1	0	0	0	1	0	Q2* = 1 Q1* = X0
1	0	0	1	1	1	
1	0	1	0	0	0	Q2* = 0 Q1* = 0
1	0	1	1	0	0	
1	1	0	0	1	1	Q2* = 1 Q1* = 1
1	1	0	1	1	1	
1	1	1	0	0	0	Q2* = 0 Q1* = 0
1	1	1	1	0	0	

Output logic table:
Dispense- D
Change 1- C1
Change 2- C2

Q2	Q1	X1	X0	D	C1	C2	
0	0	0	0	0	0	d	I0: D = 0 C1 = 0 C2 = d
0	0	0	1	0	0	d	
0	0	1	0	0	0	d	I1: D = 0 C1 = 0 C2 = d
0	0	1	1	0	0	d	
0	1	0	0	0	0	d	I2: D = 0 C1 = 0 C2 = d
0	1	0	1	0	0	d	
0	1	1	0	1	0	d	I3: D = X0’ C1 = 0 C2 = d
0	1	1	1	0	0	d	

1	0	0	0	0	0	d	I4: D = 0 C1 = 0 C2 = d
1	0	0	1	0	0	d	
1	0	1	0	1	1	0	I5: D = X0’ C1 = X0’ C2 = X0
1	0	1	1	0	0	d	
1	1	0	0	0	0	d	I6: D = 0 C1 = X0 C2 = X0’
1	1	0	1	0	1	0	
1	1	1	0	1	1	1	I7: D = 1 C1 = 1 C2 = X0’
1	1	1	1	1	1	0	

Section 4: K-maps

For Q1*:

Q1Q2/x1x2	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	0	0	0	0
10	0	0	0	0

Q1* = Q1*x1’ + Q2*x1’*x2

For Q2*:

Q1Q2/x1x2	00	01	11	10
00	0	1	1	0
01	1	0	1	1
11	0	1	0	0
10	0	0	0	0

Q2* = Q2*x1’*x2’ + Q1’*Q2*x1*x2 + Q2’*x1’*x2 + Q1*x1’*x2 + x1’*x2’*Q2

For Dispense(D):

Q1Q2/x1x2	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	1
10	0	1	1	1

D = Q1*x1+Q2*x1*x2’

For Change 1(C1):

Q1Q2/x1x2	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	0
10	0	0	1	1

C1 = Q1*x1+Q1*Q2*x2

For Change 2(C2):

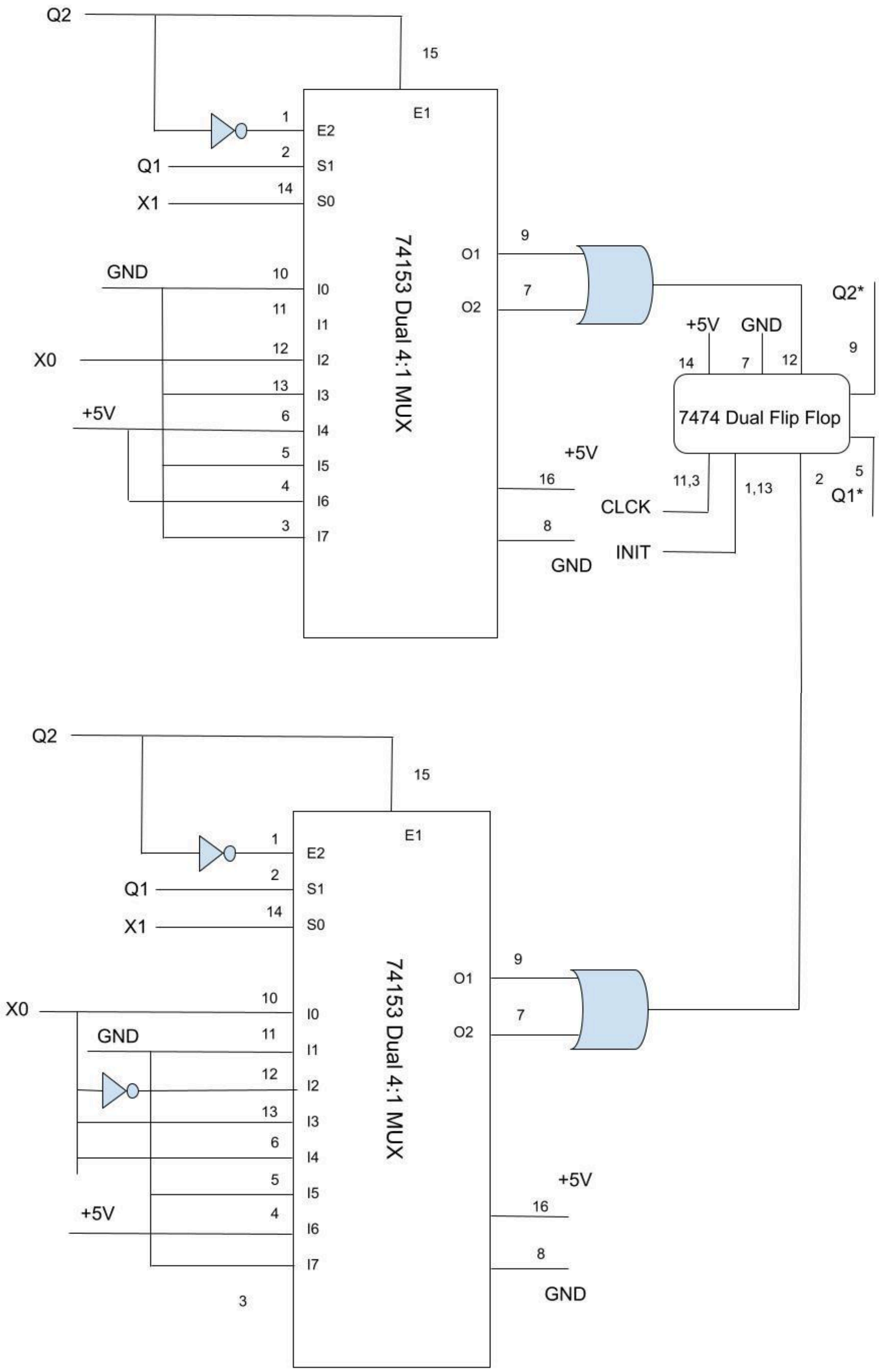
Q1Q2/x1x2	00	01	11	10
00	-	-	-	-
01	-	-	-	-
11	-	0	0	1
10	-	-	-	0

C2 = Q2’*x2*Q1’

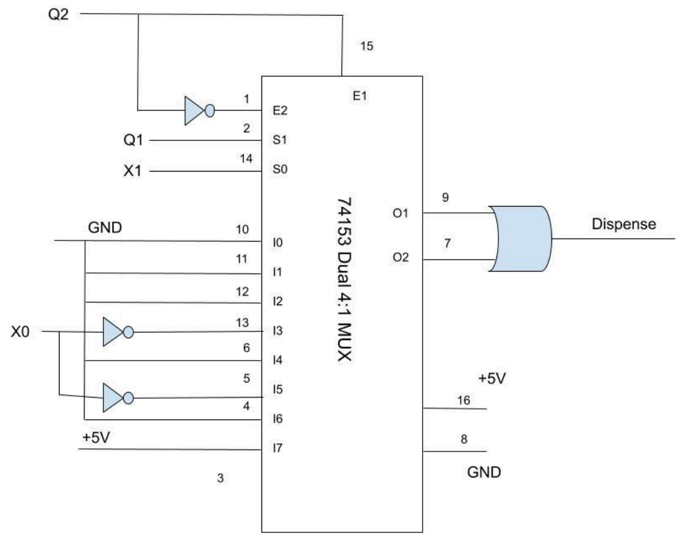
Section 5: Schematic

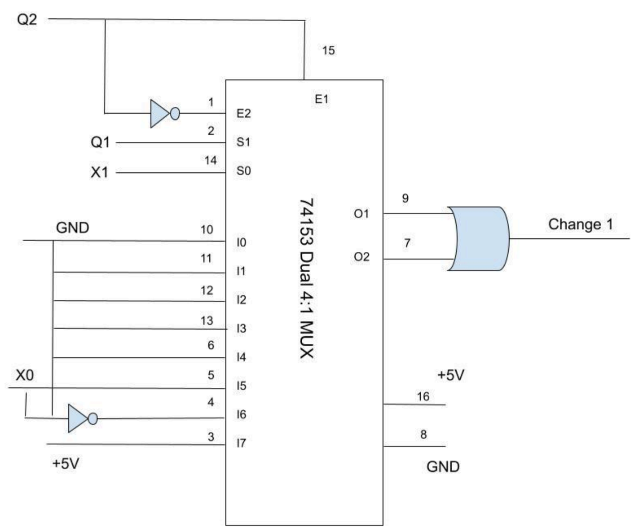
I used a 7404 inverter, 2 743 OR gates, 1 7408 AND gates, 5 74153 Muxes and a 7474 Flipflop

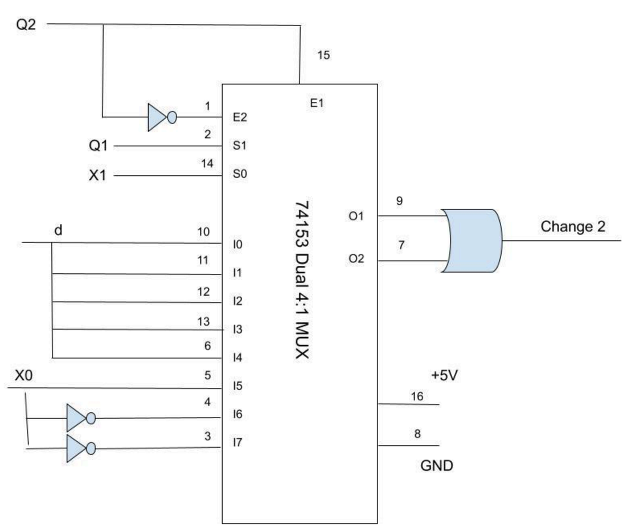
Input:



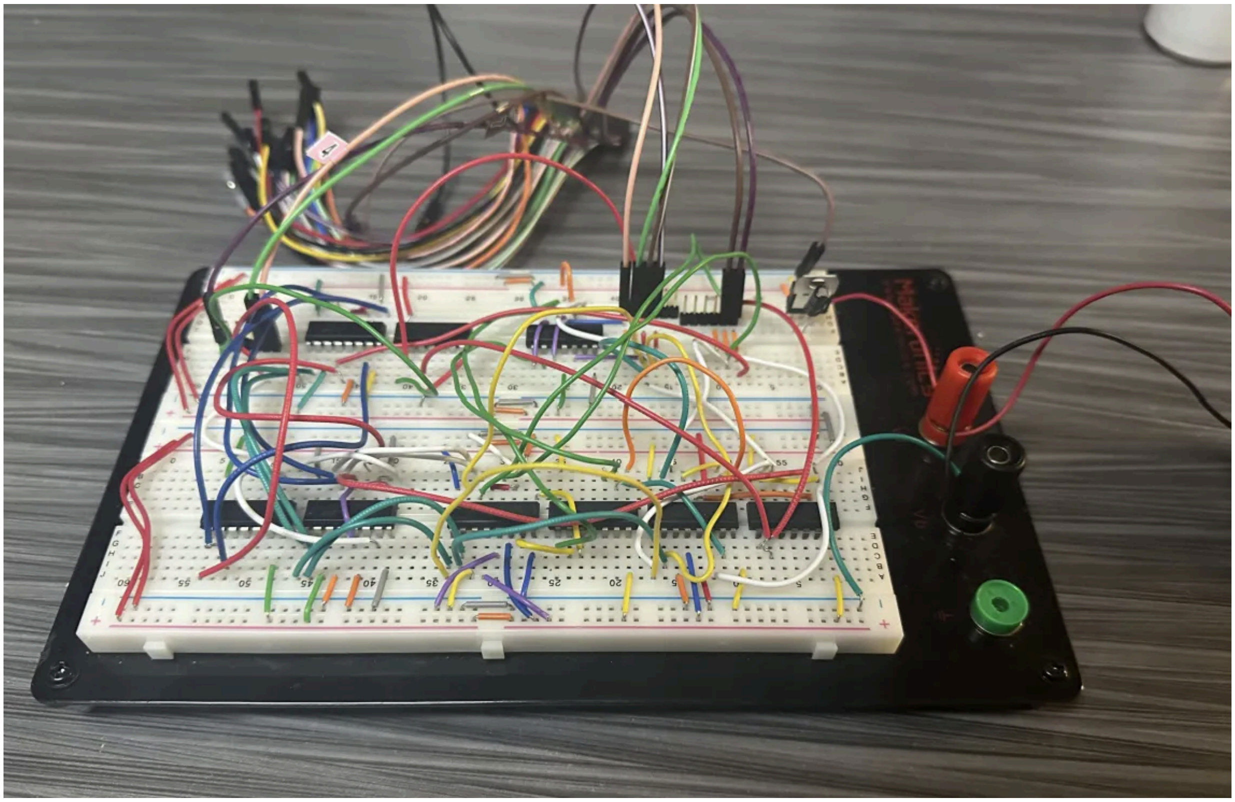
Output:


Q2
15
E1
1
E2
2
S1
Q1
14
S0
X1
74153 Dual 4:1 MUX
9
O1
7
O2
Dispense
GND
10
I0
11
I1
12
I2
X0
13
I3
6
I4
5
I5
4
I6
+5V
I7
3
+5V
16
8
GND


Q2
15
E1
1
E2
2
S1
Q1
14
S0
X1
74153 Dual 4:1 MUX
9
O1
7
O2
Change 1
GND
10
I0
11
I1
12
I2
13
I3
6
I4
X0
5
I5
4
I6
3
I7
+5V
+5V
16
8
GND


Q2
15
E1
1
E2
2
S1
Q1
14
S0
X1
74153 Dual 4:1 MUX
9
O1
7
O2
Change 2
d
10
I0
11
I1
12
I2
13
I3
6
I4
X0
5
I5
4
I6
3
I7
+5V
16
8
GND

Final Breadboard Circuit:



Section 6: Writeup

The vending machine controller implemented is a Mealy State machine. It has 4 states with 2 inputs and 3 outputs.

The four states(A,B,C,D) represent the amount deposited by the user. These are 0, 5, 10, 15 cents respectively. Furthermore, the user may choose to do nothing, attempt to select items A, B, C, or D.

The three output bits are Dispense, Change 1, Change 2. The value of dispense will be 1 if an item is dispensed from the vending machine.

Since there are 4 states(0, 5, 10, 15), we need to have 2 change bits.
The first step is to make a state diagram to represent our states and conditions including the inputs and outputs and how they relate to each other. Each state is represented by a unique binary value. A = 00, B = 01, C = 10 and D = 11. From the state diagram, we then make the symbolic state table.

From the two diagrams, we then have to make an encoded state transition table. I also have decided to use muxes to simplify the logic as without muxes, there will be too many AND/OR gates in the circuit and building it would be too complicated. I decided to use a dual 4:1 mux for each S1* and S0*. From the encoded table, I derived k-maps to give us an expression. Once the muxes were successfully implemented, I connected them to the 7474 flip flop and implemented that with a manual CLOCK and INT from the AD3.

Once completed, the same procedure was repeated for the outputs. The same 4:1 muxes were used for Dispense(D) and Change 1(C1).

Through the above diagrams and tables provided, the writeup can be implemented and the final circuit that passes all tests is also shown.