

CSE433

# Embedded Systems

Assignment 2 Report

OĞUZHAN AGKUŞ  
161044003

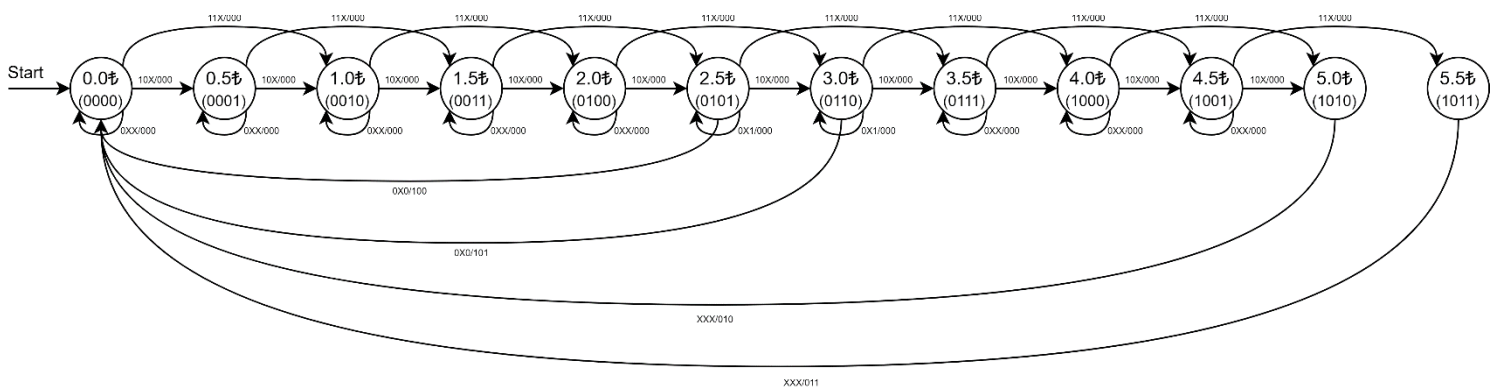
## Objective

Design an FSM controller in Verilog but without using all capabilities of Verilog. First raw state diagram and then derive state table and Boolean expressions. Then implement the Boolean expressions with Verilog.

## Assumptions

- Vending machine can take 0.5₺ and 1₺.
- It can supply either water or sandwich.
- Water requires 2.5₺.
- Sandwich requires 5₺.
- The machine can give only 0.5₺ return.
- Customer does not necessarily give any cash. If s/he wants water, does not give more than 3₺. If s/he gives more than 3₺, it means s/he wants sandwich.
- When inserted coin amount reached to 5₺ or 5.5₺, the machine directly supply sandwich without checking selection sensor.
- The machine has no expire condition. It continues where it left.
- Inputs:
  - S1 → 0: No coin, 1: Coin entered
  - S2 → 0: 0.5₺, 1: 1₺
  - S2 → 0: Water, 1: Sandwich
- Outputs:
  - M1 → 0: Stop, 1: Run motor for water
  - M2 → 0: Stop, 1: Run motor for sandwich
  - M3 → 0: Stop, 1: Run motor for return

## State Diagram



## State Table

		0	1
S1	Coin entry	No entry	Entry
S2	Coint type	0.5₺	1₺
S3	Selection	Water	Sandwich
M1	Water	Stop	Run
M2	Sandwich	Stop	Run
M3	Change	Stop	Run

Present State					Inputs			Next State					Outputs		
State Name	Q3	Q2	Q1	Q0	S1	S2	S3	State Name	D3	D2	D1	D0	M1	M2	M3
0.0¢	0	0	0	0	0	X	X	0.0¢	0	0	0	0	0	0	0
					1	0	X	0.5¢	0	0	0	<u>1</u>	0	0	0
					1	1	X	1.0¢	0	0	<u>1</u>	0	0	0	0
0.5¢	0	0	0	1	0	X	X	0.5¢	0	0	0	<u>1</u>	0	0	0
					1	0	X	1.0¢	0	0	<u>1</u>	0	0	0	0
					1	1	X	1.5¢	0	0	<u>1</u>	<u>1</u>	0	0	0
1.0¢	0	0	1	0	0	X	X	1.0¢	0	0	<u>1</u>	0	0	0	0
					1	0	X	1.5¢	0	0	<u>1</u>	<u>1</u>	0	0	0
					1	1	X	2.0¢	0	<u>1</u>	0	0	0	0	0
1.5¢	0	0	1	1	0	X	X	1.5¢	0	0	<u>1</u>	<u>1</u>	0	0	0
					1	0	X	2.0¢	0	<u>1</u>	0	0	0	0	0
					1	1	X	2.5¢	0	<u>1</u>	0	<u>1</u>	0	0	0
2.0¢	0	1	0	0	0	X	X	2.0¢	0	<u>1</u>	0	0	0	0	0
					1	0	X	2.5¢	0	<u>1</u>	0	<u>1</u>	0	0	0
					1	1	X	3.0¢	0	<u>1</u>	<u>1</u>	0	0	0	0
2.5¢	0	1	0	1	0	X	0	0.0¢	0	0	0	0	<u>1</u>	0	0
					0	X	1	2.5¢	0	<u>1</u>	0	<u>1</u>	0	0	0
					1	0	X	3.0¢	0	<u>1</u>	<u>1</u>	0	0	0	0
					1	1	X	3.5¢	0	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0
3.0¢	0	1	1	0	0	X	0	0.0¢	0	0	0	0	<u>1</u>	0	<u>1</u>
					0	X	1	3.0¢	0	<u>1</u>	<u>1</u>	0	0	0	0
					1	0	X	3.5¢	0	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0
					1	1	X	4.0¢	<u>1</u>	0	0	0	0	0	0
3.5¢	0	1	1	1	0	X	X	3.5¢	0	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0
					1	0	X	4.0¢	<u>1</u>	0	0	0	0	0	0
					1	1	X	4.5¢	<u>1</u>	0	0	<u>1</u>	0	0	0
4.0¢	1	0	0	0	0	X	X	4.0¢	<u>1</u>	0	0	0	0	0	0
					1	0	X	4.5¢	<u>1</u>	0	0	<u>1</u>	0	0	0
					1	1	X	5.0¢	<u>1</u>	0	<u>1</u>	0	0	0	0
4.5¢	1	0	0	1	0	X	X	4.5¢	<u>1</u>	0	0	<u>1</u>	0	0	0
					1	0	X	5.0¢	<u>1</u>	0	<u>1</u>	0	0	0	0
					1	1	X	5.5¢	<u>1</u>	0	<u>1</u>	<u>1</u>	0	0	0
5.0¢	1	0	1	0	X	X	X	0.0¢	0	0	0	0	0	<u>1</u>	0
5.5¢	1	0	1	1	X	X	X	0.0¢	0	0	0	0	0	<u>1</u>	<u>1</u>

## Boolean Expressions

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
110	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1
111	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1
101	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
100	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

$$D3 = Q3Q2'Q1' + S1Q3'Q2Q1Q0 + S1S2Q3'Q2Q1$$

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
001	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
011	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
010	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
110	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
111	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
101	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0
100	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0

$$D2 = Q3'Q2Q1'Q0' + S1'Q3'Q2Q1Q0 + S1'S3Q3'Q2 + S1Q3'Q2'Q1Q0 + S1Q3'Q2Q1' + S1S2'Q3'Q2Q0' + S1S2Q3'Q2'Q1$$

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
001	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
011	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
010	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
110	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1
111	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1
101	1	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0
100	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0

$$D1 = S1'Q3'Q1Q0 + S1'Q3'Q2'Q1 + S1'S3Q3'Q1 + S1Q3'Q1'Q0 + S1S2'Q3'Q1Q0' + S1Q2'Q1'Q0 + S1S2Q3'Q1' + S1S2Q2'Q1'$$

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0
001	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0
011	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0
010	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0
110	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0
111	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0
101	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1
100	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1

$$D0 = S1'Q3'Q1Q0 + S1'Q2'Q1'Q0 + S1'S3Q3'Q0 + S1S2'Q3'Q0' + S1S2'Q2'Q1'Q0' + S1S2Q3'Q0 + S2Q2'Q1'Q0$$

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$M1 = S1'S3'Q3'Q2Q1'Q0 + S1'S3'Q3'Q2Q1Q0'$$

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
111	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

$$M2 = Q3Q2'Q1$$

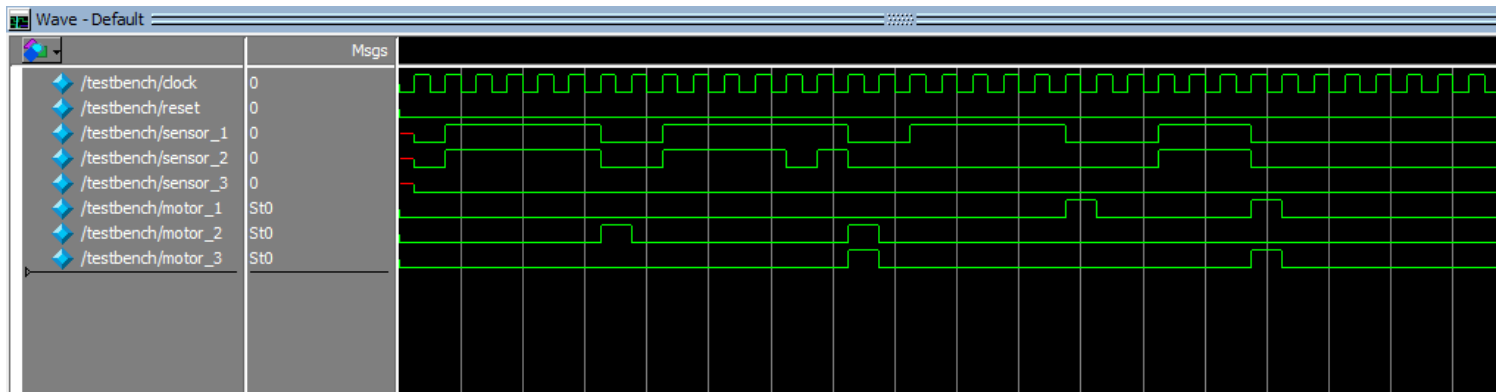
	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
000	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
001	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
011	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
010	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
111	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

$$M3 = S1'S3'Q3'Q2Q1Q0' + Q3Q2'Q1Q0$$

## Testbench Results

There are four scenarios for buying stuffs.

- First one is, inserting 5 1¢, then getting a sandwich.
- Second one is, inserting 1 0.5¢ and 5 1¢, then getting a sandwich with 0.5¢ return.
- Third one is, inserting 1 0.5¢ and 2 1¢, then getting a water.
- Fourth one is, inserting 3 1¢, then getting a water and 0.5¢ return



Follow the motor numbers in graphic.