CMPE 212

Spring, 2016

Project 2 Report

Sabbir Ahmed
Teaching assistant:
Ahmed Shahin

1. **Project description:**

The goal of this project is to design and implement a finite state machine through combinational and

sequential circuits. With one 1-bit input corresponding to 25 ₵ or 10 ₵, the machine is to behave

similarly to a vending machine dispensing an output after an amount of 70 ₵ or higher has been

accumulated, except when the excess is either 5 ₵ or 15 ₵, in which case a change of 5 ₵ will also be

outputted leaving the remaining as credit for the next simulation.

2. **Process:**

a. **State diagram:**

The following state diagram was derived from the description:
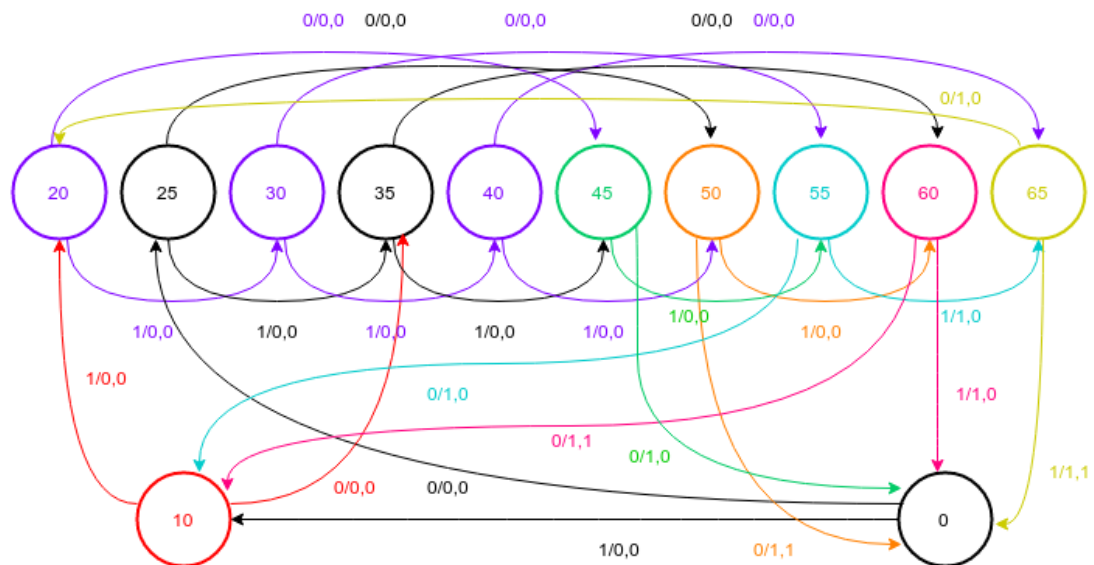
State diagram for a vending machine finite state machine with outputs y0, y1

The states were verified by an executable Python script, vending_machine_fsm.py attached to the

report in the src directory. The script is written in Python 2.7 syntax.

Usage:

Systems with a python2 compiler as default: `python vending_machine_fsm.py`

Systems with a newer Python compiler: `python2.x vending_machine_fsm.py`

The script generates the following outputs:

```
Switch = 0                      Switch = 1

A 0000 -> D 0010 (0, 0)         A 0000 -> B 0001 (0, 0)
C 0011 -> H 0110 (0, 0)         C 0011 -> E 0100 (0, 0)
B 0001 -> F 0101 (0, 0)         B 0001 -> C 0011 (0, 0)
E 0100 -> J 1101 (0, 0)         E 0100 -> G 0111 (0, 0)
D 0010 -> I 1100 (0, 0)         D 0010 -> F 0101 (0, 0)
G 0111 -> L 1110 (0, 0)         G 0111 -> I 1100 (0, 0)
F 0101 -> K 1111 (0, 0)         F 0101 -> H 0110 (0, 0)
I 1100 -> A 0000 (1, 1)         I 1100 -> K 1111 (0, 0)
H 0110 -> A 0000 (1, 0)         H 0110 -> J 1101 (0, 0)
K 1111 -> B 0001 (1, 1)         K 1111 -> A 0000 (1, 0)
J 1101 -> B 0001 (1, 0)         J 1101 -> L 1110 (0, 0)
L 1110 -> C 0011 (1, 0)         L 1110 -> A 0000 (1, 1)
```

12 states with their corresponding code assignments

b. Functions:

The script was also used to determine the minterms for each of the Boolean functions for the 4 flip flops and 2 outputs:

```
Default don't cares: [8, 9, 11, 10, 24, 25, 27, 26]
J0 [2, 4, 5, 7, 22, 23]
J0_D [12, 13, 14, 15, 28, 29, 30, 31]
J1 [1, 2, 3, 18, 19]
J1_D [4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31]
J2 [0, 5, 17, 20, 21, 28, 29]
J2_D [2, 3, 6, 7, 14, 15, 18, 19, 22, 23, 30, 31]
J3 [4, 14, 16, 18, 20, 22, 28]
J3_D [1, 3, 5, 7, 13, 15, 17, 19, 21, 23, 29, 31]
K0 [12, 13, 14, 15, 30, 31]
K0_D [0, 1, 2, 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 22, 23]
K1 [6, 12, 13, 14, 15, 30, 31]
K1_D [0, 1, 2, 3, 16, 17, 18, 19]
K2 [2, 6, 15, 18, 19, 22, 23, 30, 31]
K2_D [0, 1, 4, 5, 12, 13, 16, 17, 20, 21, 28, 29]
K3 [3, 7, 19, 21, 23, 29, 31]
K3_D [0, 2, 4, 6, 12, 14, 16, 18, 20, 22, 28, 30]
z0 [6, 12, 13, 14, 15, 30, 31]
z1 [12, 15, 30]
```

Minterms for all the functions generated by simulating the state machine
(variables with '_D' represent don't cares)

The minterms were then used to reduce functions through use of Karnaugh maps:

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	0	0	0	1
$\overline{A}.\overline{B}.C$	1	1	1	0
$\overline{A}.B.C$	x	x	x	x
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.\overline{B}.\overline{C}$	0	0	0	0
$A.\overline{B}.C$	0	0	1	1
$A.B.C$	x	x	x	x
$A.B.\overline{C}$	x	x	x	x

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	x	x	x	x
$\overline{A}.\overline{B}.C$	x	x	x	x
$\overline{A}.B.C$	1	1	1	1
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.\overline{B}.\overline{C}$	x	x	x	x
$A.\overline{B}.C$	x	x	x	x
$A.B.C$	0	0	1	1
$A.B.\overline{C}$	x	x	x	x

$$j0 = \bar{A}C\bar{D} + \bar{A}CE + ACD + \bar{A}CD\bar{E}$$

$$k0 = \bar{A} + D$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	0	1	1	1
$\overline{A}.\overline{B}.C$	x	x	x	x
$\overline{A}.B.C$	x	x	x	x
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.C$	0	0	1	1
$A.\overline{B}.C$	x	x	x	x
$A.B.C$	x	x	x	x
$A.B.\overline{C}$	x	x	x	x

$$j1 = \bar{A}E \,+\, D$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	x	x	x	x
$\overline{A}.\overline{B}.C$	0	0	0	1
$\overline{A}.B.C$	1	1	1	1
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.\overline{C}$	x	x	x	x
$A.\overline{B}.C$	0	0	0	0
$A.B.C$	0	0	1	1
$A.B.\overline{C}$	x	x	x	x

$$k1 = \bar{A}B + BD + \bar{A}D\bar{E}$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	1	0	x	x
$\overline{A}.\overline{B}.C$	0	1	x	x
$\overline{A}.B.C$	0	0	x	x
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.\overline{C}$	0	1	x	x
$A.\overline{B}.C$	1	0	x	x
$A.B.C$	1	1	x	x
$A.B.\overline{C}$	x	x	x	x

$$j2 = AE + AC + \bar{A}\bar{C}\bar{E} + \bar{B}CE$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	x	x	0	1
$\overline{A}.\overline{B}.C$	x	x	0	1
$\overline{A}.B.C$	x	x	1	0
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.C$	x	x	1	1
$A.\overline{B}.C$	x	x	1	1
$A.B.C$	x	x	1	1
$A.B.\overline{C}$	x	x	x	x

$$k2 = A + \bar{B}\bar{E} + BE$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	0	x	x	0
$\overline{A}.\overline{B}.C$	1	x	x	0
$\overline{A}.B.C$	0	x	x	1
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.C$	1	x	x	1
$A.\overline{B}.C$	1	x	x	1
$A.B.C$	1	x	x	0
$A.B.\overline{C}$	x	x	x	x

$$j3 = A\bar{B} +\, A\bar{D} + \bar{B}C\bar{D} \,+\, \bar{A}BD$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	x	0	1	x
$\overline{A}.\overline{B}.C$	x	0	1	x
$\overline{A}.B.C$	x	0	0	x
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.B.C$	x	0	1	x
$A.\overline{B}.C$	x	1	1	x
$A.B.C$	x	1	1	x
$A.B.\overline{C}$	x	x	x	x

$$k3 = \bar{B}D +\, AC$$

|	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	0	0	0	0
$\overline{A}.\overline{B}.C$	0	0	0	1
$\overline{A}.B.C$	1	1	1	1
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.\overline{B}.\overline{C}$	0	0	0	0
$A.\overline{B}.C$	0	0	0	0
$A.B.C$	0	0	1	1
$A.B.\overline{C}$	x	x	x	x

$$y0 = \overline{A}B + BD + \overline{A}\overline{C}D\overline{E}$$

	$\overline{D}.\overline{E}$	$\overline{D}.E$	$D.E$	$D.\overline{E}$
$\overline{A}.\overline{B}.\overline{C}$	0	0	0	0
$\overline{A}.\overline{B}.C$	0	0	0	0
$\overline{A}.B.C$	1	0	1	0
$\overline{A}.B.\overline{C}$	x	x	x	x
$A.\overline{B}.\overline{C}$	0	0	0	0
$A.\overline{B}.C$	0	0	0	0
$A.B.C$	0	0	0	1
$A.B.\overline{C}$	x	x	x	x

$$y1 = \overline{A}B\overline{D}\overline{E} + \overline{A}BDE + \overline{A}BD\overline{E}$$

c. Design:

After several Boolean function manipulation, a design for the circuit was realized. Since the number of flip flops can be found by computing $\lceil \log_2(states) \rceil$, 4 JK flip flops were used. A decoder was used as well to generate 4 3-bit minterms that were used multiple times. The following schematic was designed and used as the working blueprint during the implementation:
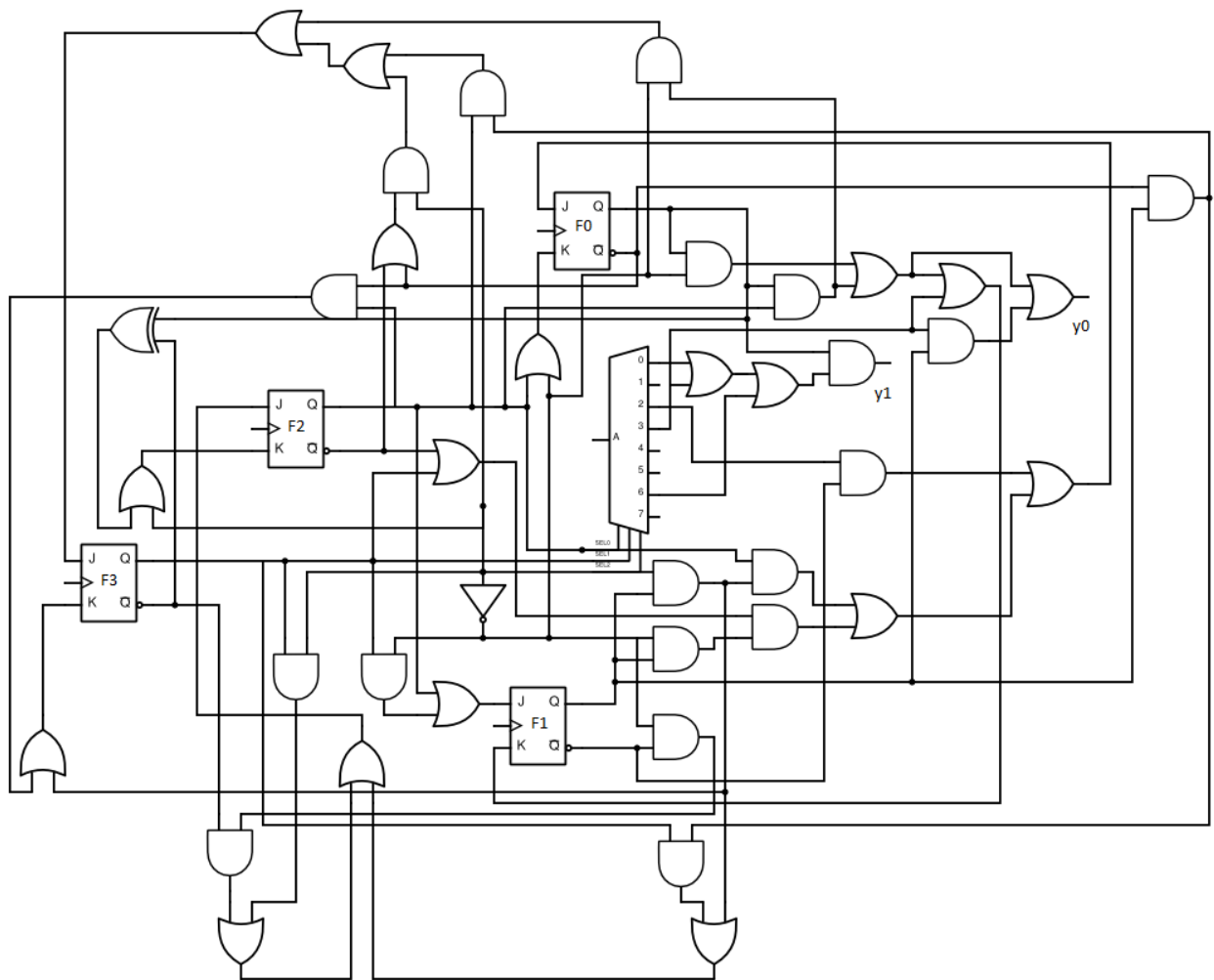
Circuit design of the vending machine finite state machine

d. Verification:

The logic of the entire circuit was verified using Verilog. Several testbenches have been included along with a makefile for convenience. The scripts were compiled using Icarus Verilog (iverilog) successfully, and ncverilog on GL, where it showed occasional runtime errors.

Usage:

Testbench for JK flip flop module:

```
make jk
```

Testbench for the machine with inputs of only 0:

```
make quarters
```

Testbench for the machine with inputs of only 1:

```
make dimes
```

Testbench for the machine with inputs of both 0 and 1:

```
make
```

3. **Conclusion:**

After the designing processes were completed, the circuit was constructed on a breadboard, and the logic above was verified again. Due to possible pin mapping errors and/or flawed logic, the state machine was not able to be implemented correctly and debugging the entire machine was not possible.