

CS M51A and EE M16 Spring 2014 Section 1

Logic Design of Digital Systems

Dr. Yutao He

ISE/Verilog Lab #3 - Design of Sequential  
Systems

Due: June 3, 2014

(1) Name: Liu Guojin  
Last First

Student ID: 804182525

(2) Name: Zarifian Anais  
Last First

Student ID: 704-163-927

Date: 6/2/14

Result	
Correctness	
Creativity	
Report	
Total Score	

## **(2) Abstract**

The purpose of this project is to create a digital design circuit, which would replicate the actions of a vending machine like object. A person would insert a certain number of coins until they reach a certain money limit and at that point, it would dispense gum. This vending machine like circuit can be implemented using JK flip flops to aid in telling the circuit what should happen next upon reading in a certain value of a coin.

The gum is worth 20 cents and the customer can put in dimes and nickels in order to get gum. If the customer does not have exactly 20 cents and inserts 25 cents, then the machine would know to dispense gum and an extra nickel. This circuit design also includes a probability that the person will eject all the coins they put in and at that point, there is a switch that would reset the current state of the machine to as if the customer never put in any coins.

## **(3) The Functions of the Circuit**

$$\begin{aligned}J_1 &= s_0x_0 + x_1 \\K_1 &= s_0x_0 + x_1\end{aligned}$$

$$\begin{aligned}J_0 &= x'_1x_0 \\K_0 &= x'_1x_0 + s_1x_0\end{aligned}$$

$$\begin{aligned}z_1 &= s_1x_1 + s_1x_0x_0 \\z_0 &= s_1s_0x_1\end{aligned}$$

The schematic of the circuit with the JK flip-flops can be seen in the appendix.

## **(4) The Verilog Code**

```

20 ///////////////////////////////////////////////////////////////////
21 module JKFF(
22     input J,
23     input K,
24     input clk,
25     input rst,
26     output reg Q
27 );
28
29     always @(posedge clk or posedge rst) //asynch reset
30 begin
31     if(rst == 1)
32         begin
33             Q <= 0;
34         end
35     else begin
36         case({J, K})
37             2'b00: Q <= Q; //no change
38             2'b01: Q <= 1'b0; //Clear
39             2'b10: Q <= 1'b1; //Set
40             2'b11: Q <= ~Q; //Complement
41         endcase
42     end
43 end
44 endmodule
45

```

```

1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company: T12
4 // Engineer: Anais Zarifian & Calvin Liu
5 //
6 // Create Date: 15:36:26 06/02/2014
7 // Design Name:
8 // Module Name: csm51a_proj3
9 // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module iKon (
22     input x0,
23     input x1,
24     input rst,
25     input clk,
26     output s1,
27     output s0,
28     output reg z1,
29     output reg z0
30 );

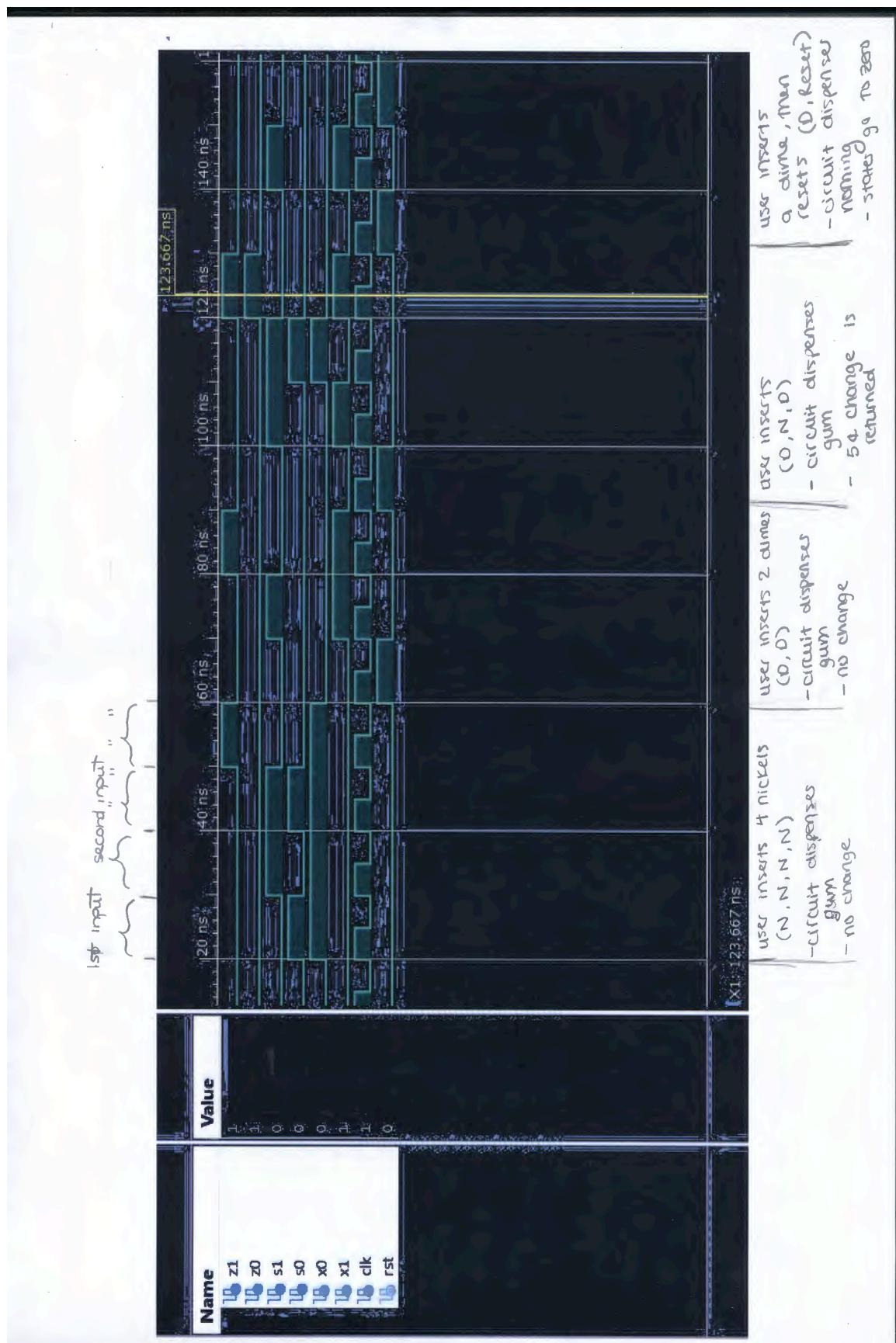
```

```

31   wire J1K1;
32   wire J0;
33   wire K0;
34   //connections to JK flip flops
35   JKFF JKFF1(J0, K0, clk, rst,s0);
36   JKFF JKFF2(J1K1, J1K1, clk, rst, s1);
37
38   always @ (posedge clk, posedge rst) begin
39     if (!rst) begin
40       z1 <= (s1&x1) | (s1&s0&x0);
41       z0 <= (s1&s0&x1);
42     end
43     else begin
44       z1 <= 0;
45       z0 <= 0;
46     end
47   end
48
49 //Assignment of J1K1
50 wire w1;
51 and (w1, s0, x0);
52 or (J1K1, x1, w1);
53
54 //Assignment of J0
55 and (J0, !x1, x0);
56
57 //Assignment of K0
58 wire w2;
59 and (w2, s1, x0);
60 or (K0, J0, w2);
61 endmodule
62

```

## (5) The Simulation Result



## **(6) The Design Review**

We started the project by working on anything we could by hand. We drew the state table, converted the state table into K-maps for minimization and then drew out the circuit based on the minimization done with the K-maps. The topics that we learned here is the implementation of a sequential systems and how to construct them by applying JK flip-flops. We learned most of these topics through homework and class so doing it on paper was simple.

A problem we came across was implementing the JK flip-flops in Verilog since we never coded in Verilog before did not really get a good overview on how to code the system and we had to learn how to implement not just a circuit, but learn how to implement a JK flip-flop and use it. We implemented the JK flip-flop correctly based on the introduction given to us by the TA, but we had trouble trying to use the JK flip flop in our circuit design. We also had trouble in how to construct the test bench file for a sequential system. By asking our peers, we learned that the sequential system can be tested as though it was a state diagram where certain inputs were declared and the output was given to us. We had trouble with the test bench file because we did not know if how it was supposed to be set up, if the output was supposed to be set for us, etc. Some tests were made based off the state diagram like if we inserted a dime in first, and then a nickel.

Most of our time was allocated to figuring out how to implement the system and researching how to create the code for it since this project is much different compared to the other projects and much more difficult. We had to use YouTube videos, TA notes, and research on how to implement the code. Many other people also had problems with this project and we had to use Piazza a lot. The project took a long time because we had many bugs as well and do not know how to debug in the new environment so tracing whether the problem was in the .v file or the test bench file was really time consuming and problematic.

## **(7) Team Member Contributions**

Anais – 50%

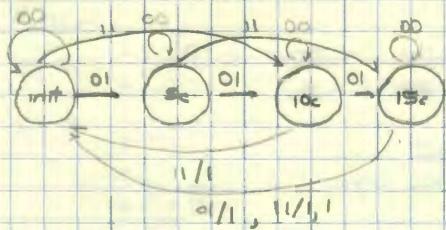
- State Diagram
- K-maps
- State-Transition Table
- Check Verilog code
- Worked on Parts 5-8 of the project report

Calvin – 50%

- Checked state diagram
- Checked state transition table
- Implemented the JK flip flop code
- Typed up Verilog code
- Worked on parts 1-4 of the project report

A majority of the project was worked on together. Anais drew the state diagram, while Calvin worked on the state transition table. The inputs for the JK flip flops were implemented and minimized by Anais, and both members checked each other work. A lot of research was done by both members of the team for implementing the Verilog code because not much was known about implementing a sequential system in Verilog. Overall, the work was evenly divided.

## (8) Appendix – The detailed design worksheet



$Q(t) \rightarrow Q(t+1)$	J	K
0 → 0	0	-
0 → 1	1	-
1 → 0	-	-
1 → 1	-	0

Input $x_0, x_1, x_2$		PS	00	01	11	reset	reset	reset	
00	init	int, 0, 0	5c, 0, 0	10c, 0, 0	init, 0, 0	int, 0, 0	int, 0, 0	int, 0, 0	
01	5c	5c, 0, 0	10c, 0, 0	15c, 0, 0					
10	10c	10c, 0, 0	15c, 0, 0	int, 1, 0					
11	15c	15c, 0, 0	int, 1, 0	int, 1, 1					

NS, Z<sub>1</sub>, Z<sub>0</sub>

PS	Q, Q'	00	01	11	reset	reset	reset	
00	00	00	01	10	00	00	00	
01	01	10	11	00				
10	10	11	00	00	↓	↓	↓	
11	11	00	00					

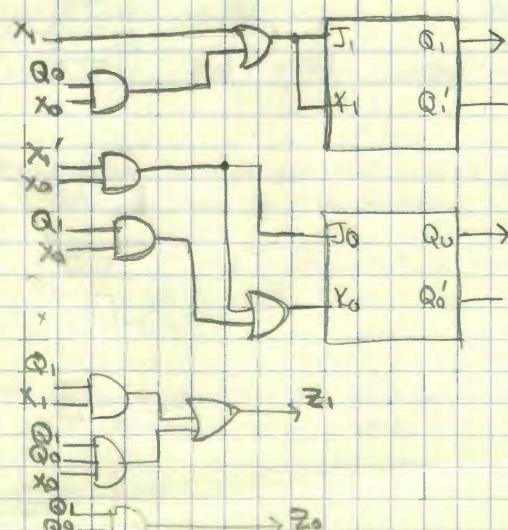
Input		Q, Q'	00	01	11
00	00	00	0-0-	0-1-	1-0-
01	01	01	0-0-	1-1-	1-0-
10	10	10	-00-	-01-	-10-
11	11	11	-0-0	-1-1	-1-1

J<sub>1</sub>, K<sub>1</sub>, J<sub>0</sub>, K<sub>0</sub>

$$J_1 = Q_0 x_0 + x_1 \\ K_1 = Q_0 x_0 + x_1$$

$$Z_1 = Q_1 x_1 + Q_0 x_0 \\ Z_0 = Q_1 Q_0 x_1$$

$$J_0 = x_1' x_0 \\ K_0 = x_1' x_0 + Q_1 x_0$$



\* JK flip flop Q outputs go around to the beginning of the circuit

