EE 142 Project 2 report **AYDIN UZUN 2015401210**

The project is about designing and simulating a coffee machine dispenser controller that takes coins as the input and gives three different types of coffee and change coins as the output. Indivdual steps of the design are given below and the state tables, design schematics,strategies and simulations are explained too.

Step 1: Obtain the specifications of the desired circuit

-The machine takes 50 kuruş and 1 lira.

-The machine can dispense

1. Latte: 50 kuruş 2. Turkish coffee: 1 lira (C) 3. Cappuccino: 2 lira

-The machine has a coin return button.

- Only one input can be active at a time

- A product can be dispensed in one clock cycle

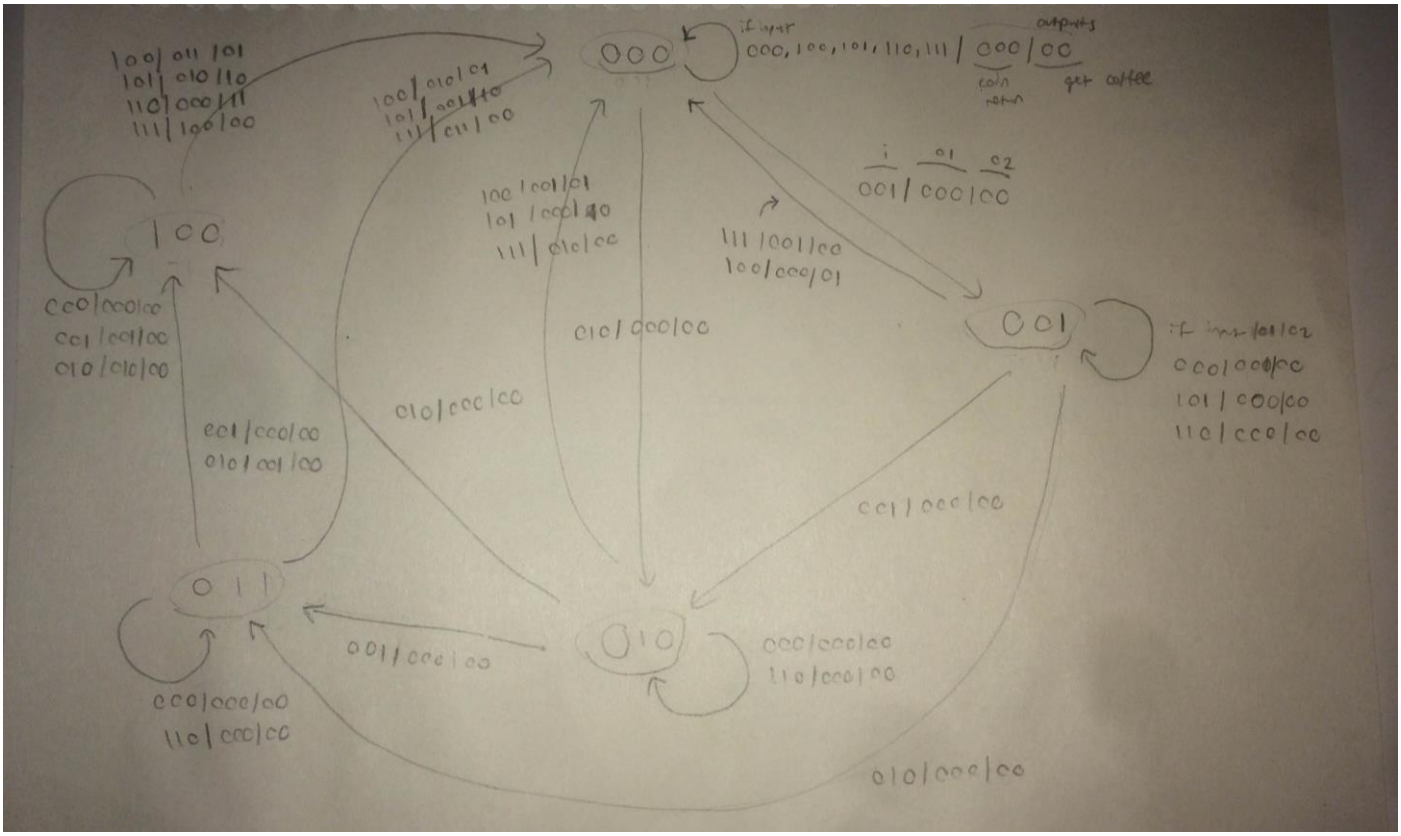
-If more than 2 lira is inserted, the money is automatically returned.

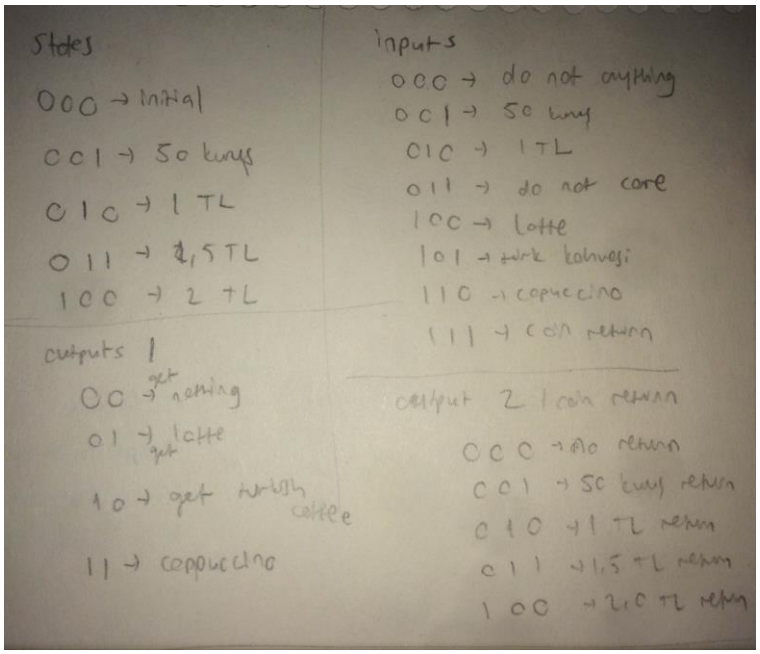
-If no inputs are active, the state machine stays in the current state.

Step 2: Develop a state diagram

A state diagram should show all possible states and the conditions for which the circuit moves from one state to the next.

My state diagram is:

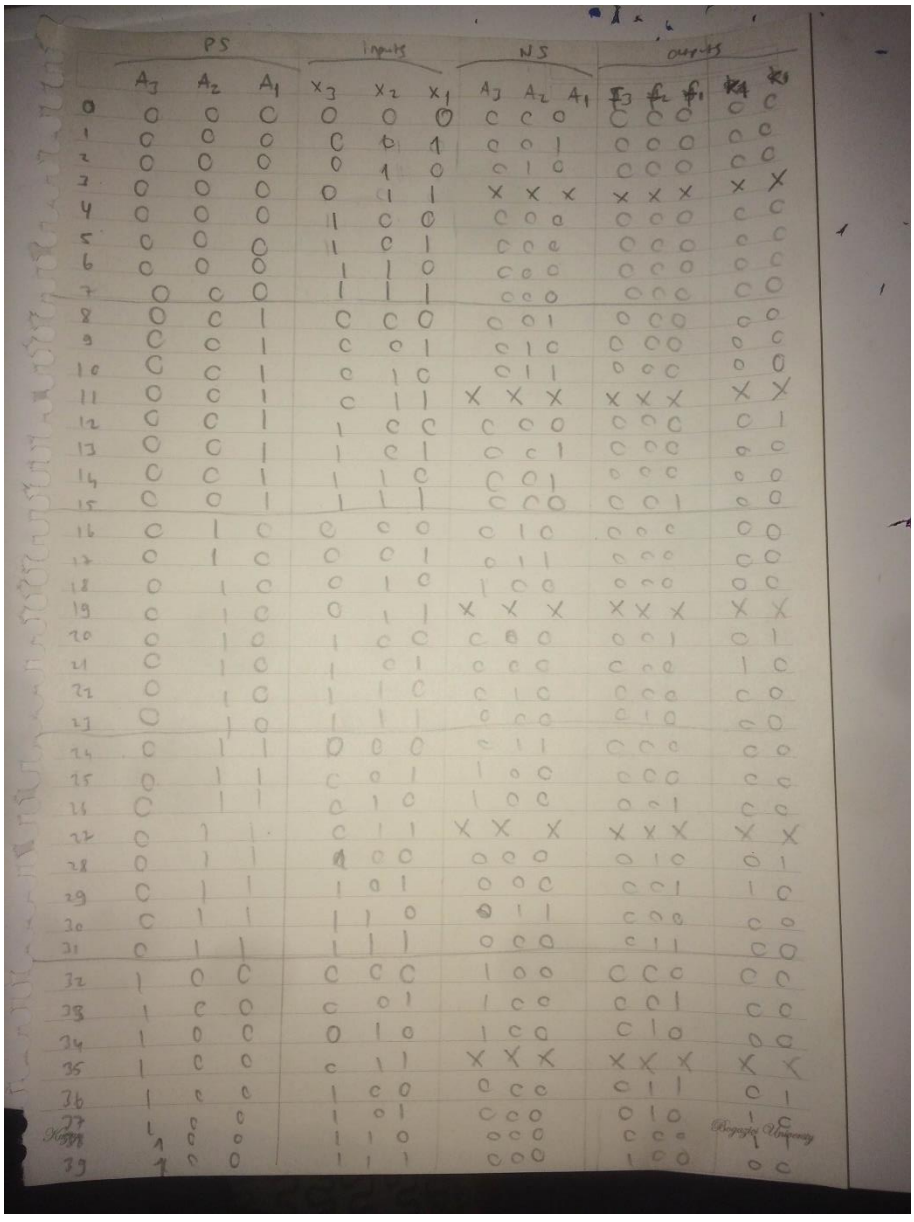




I have 5 different states: initial state, 0.50 tl, 1.0 tl, 1.5 tl, 2.0 tl. Therefore I should use at least 3 bits. And I have got 7 inputs. I use 3 bits and one input which is 011 is a don't care situation. Moreover I have two different outputs type. One is coffe output. The second one is coin output. For coffe output I have 4 states and I use 2 bits. For coin return I have 5 states and I use 3 bits. Every coin after 2 lira is returned.

For example: When we analyse the initial position 000. If I put a 50 kuruş, this input take me to the state 001. All of the other inputs are stay at initial position.

For example : Start position 001, this means I have got 0.50 tl. If my input is 100, this input takes me to initial position 000 and gives me a latte as output, but no coin return. If my input is 111, this input takes me to initial position 000 and gives me no coffee output, because input 111 is coin return input. I take 0.50 tl as a coin return output. If my input is 001, I get 0.50 kuruş more and this leads me to the station 010, in which I have 1 tl money, but I have no output. If my input is 101, I will stay at station 001, because with 0.50 tl I can not get a turkish coffee, which costs 1 tl.



Step 3 : Develop the state table from the state diagram

The input 011 is not defined therefore we take don't care input 001 as outputs and next states. The states after 100 are not defined too. (For ex: 101/110/111). The list should have 64 entries because of 6 bits, but I have 40 entries. The entries after 39 are just don't cares.

A2A1A0 represent my present state and next state, X2,X1,X0 represent my input. f2,f1,f0 represent my coin return output. K1 and K0 are my coffe output.

Step 4. Choose the type of flip-flops that will be used in the implementation (Synthesis Using *D* Flip-Flops)

-Derive the next-state logic expressions to develop the Input Logic Circuit

-Derive the logic expressions for the Output Logic Circuit [1]

I want to use D flip-flops, because the Boolean equations describing the inputs to the flip-flops can be obtained directly from the state table.[2] I choose three D flip-flops to represent the eight states, and I label their outputs A2,A1 and A0[3]. There are three inputs x3,x2,x1 and two output types (f2,f1,fo and k1,ko) . The characteristic equation of the D flip-flop is Q(t+1) = D(Q),
which means that the next-state values in the state table specify the *D* input condition for the flip-flop. The flip-flop input equations can be accessed directly from the next-state columns of A2,A1 and A0 where A2,A1 and A0 are the present-state values of flip-flops, x2,x1,x0 are the inputs, and DA2,DA1 and DA3 are the input equations for D flip-flop . The minterms for outputs from the output column in the state table and the minterms for D flip-flop inputs from the next state column in the state table: [4]

$A2(t+1) = DA2(A2,A1,A0, x3,x2,x1) = \sum (18,25,26,32,33,34)$

$A1(t+1) = DA1(A2,A1,A0, x3,x2,x1) = \sum (2,9,10,16,17,22,24,30)$

$A0(t+1) = DA0(A2,A1,A0, x3,x2,x1) = \sum (1,8,10,13,14,17,24,30)$

$f2(A2,A1,A0, x3,x2,x1) = \sum (39)$

$f1(A2,A1,A0, x3,x2,x1) = \sum (23,28,31,34,36,37)$

$f0(A2,A1,A0, x3,x2,x1) = \sum (15,20,26,29,31,33,36)$

$k1(A2,A1,A0, x3,x2,x1) = \sum (21,29,37,38)$

$k0(A2,A1,A0, x3,x2,x1) = \sum (12,20,28,36,38)$

don't cares= ∑(3,11,19,27,35,40,41,42,.....63)

But I have got 6 variables(3 inputs and 3 present states). This means I should use 6 variable K-map to implement the input condition for the flip flop and outputs. To implement from 6 variable K-map is hard therefore I take some help from internet. There was a online 6 variable K-map solver. [5]

DA2 = A2.x2' + A1.x2'.x1+A1.A0.x2'.x0

DA1= A2'.A1'.x2'.x1+A1'.A0.x2'.x0+A1.A0'.x2'.x1'+A1.x2'.x1'.x0'+A1.x2.x1.x0'

DA0=A2'.A0'.X2'.x0+A1'.A0.X2'.X0'+A0.x2'.x1'.x0'+A0.x2.x1.x0'+A1'.A0.x2.x0.x1'

f2=A2.x1.x0

f1=A1.x1.x0+A2.x2'.x1+A2.x2.x1'+A1.A0.x2.x1'.x0'

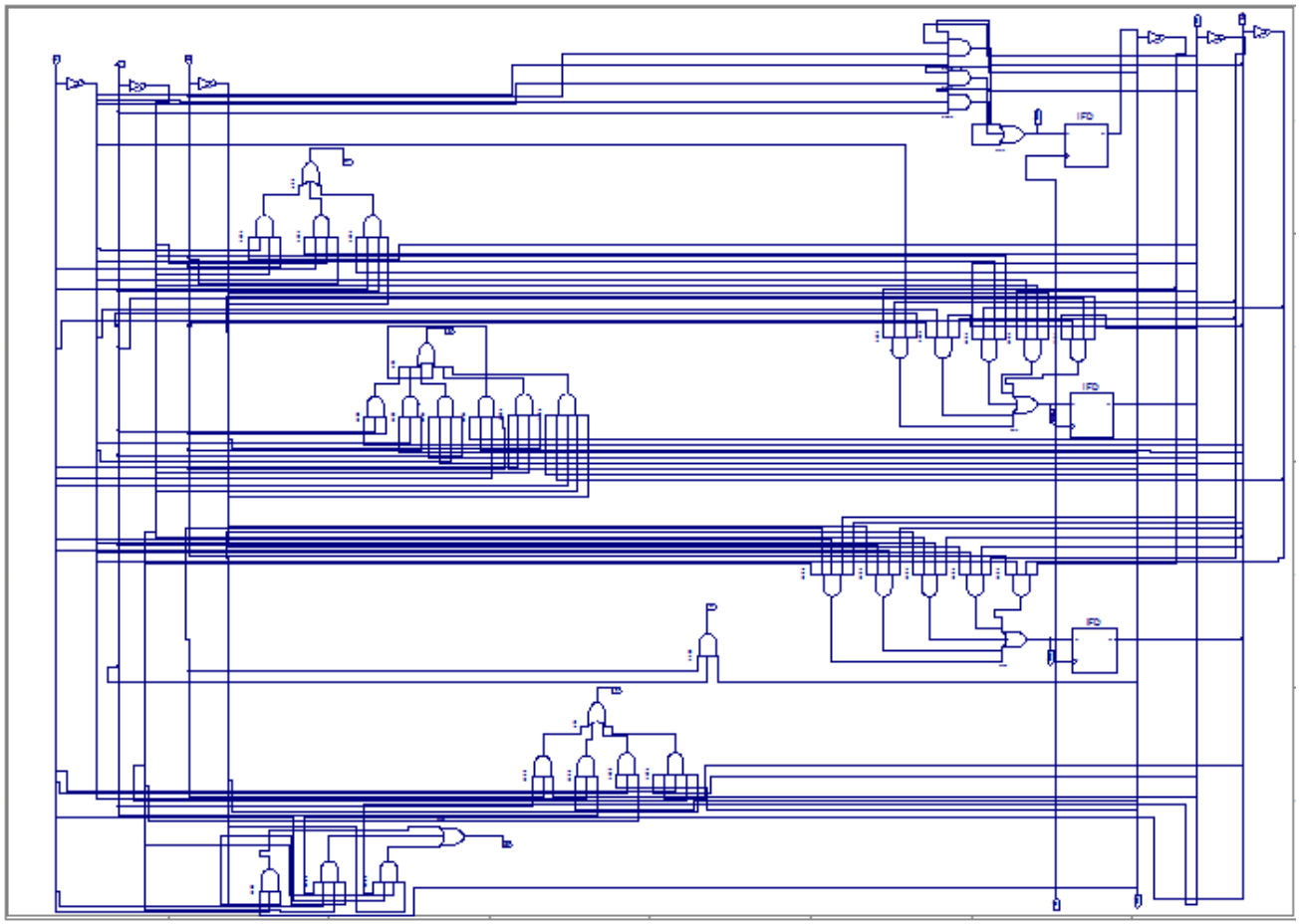
f0= A0.x1.x0+A2.x2'.x0+A1.A0.x2'.x1+A1.A0.x2.x0+A2.x2.x1'.x0'+A1.A0'.x2.x1'.x0'

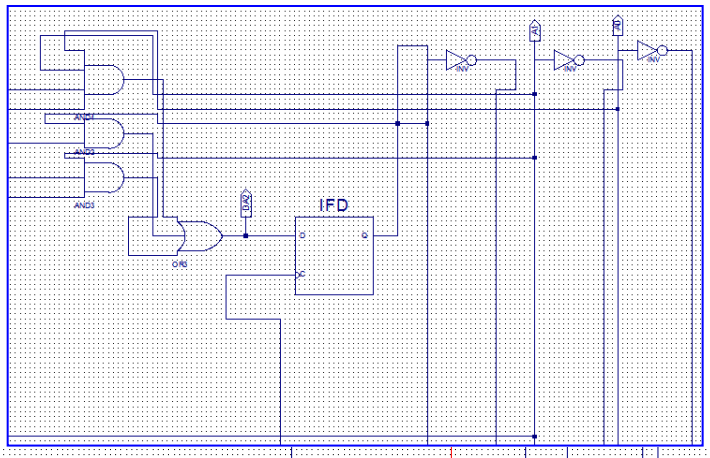
k1=A1.x2.x0.x1'+A2.x2.x0.x1'+A2.x2.x1.x0'

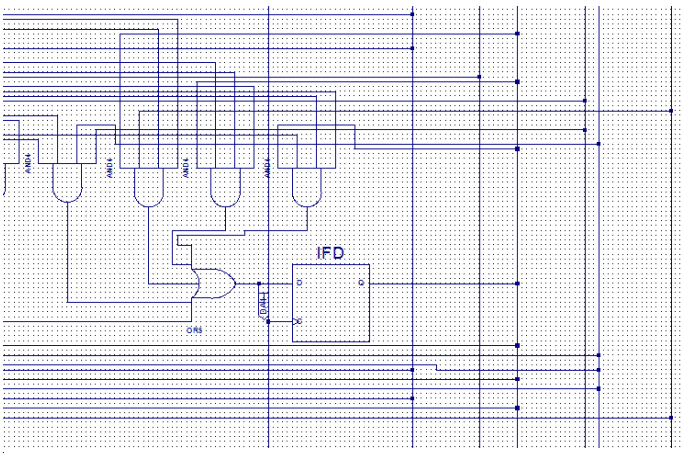
k0=A2.x2.x0'+A0.x2.x1'.x0'+A1.x2.x1'.x0'

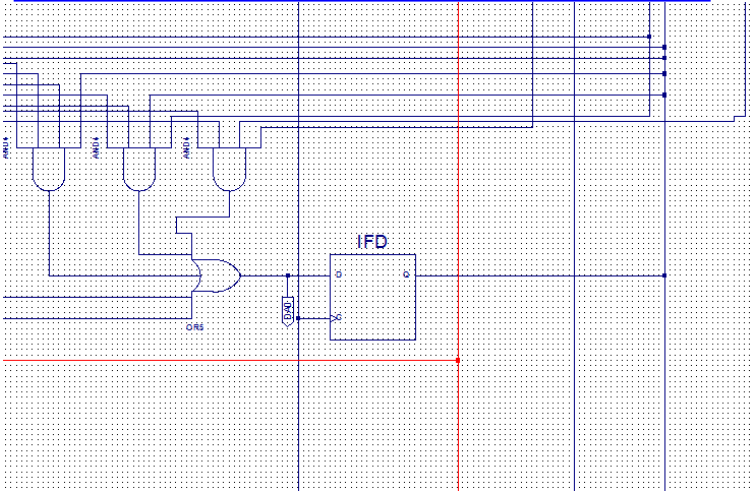
The simplified equations are above.

Step5-Implement the design with D flip-flops, AND gates,OR gates and inverters.





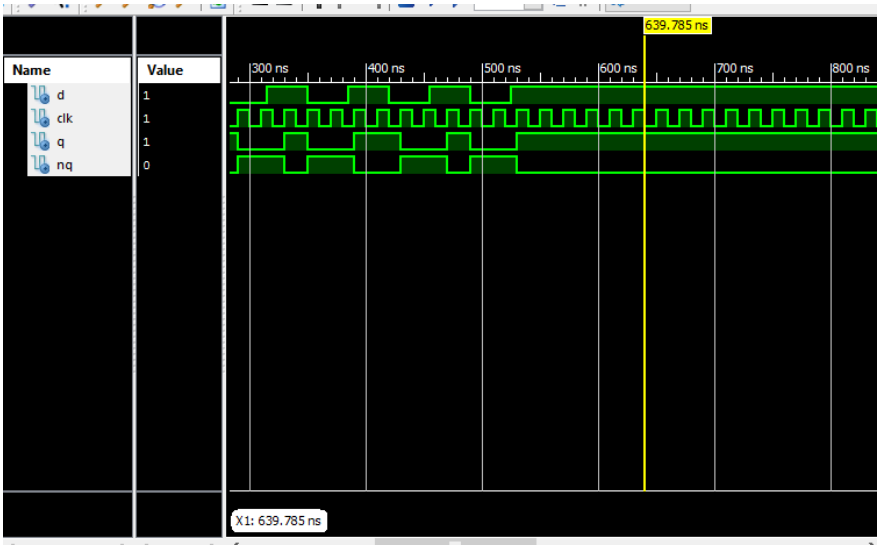




3 D Flip-Flops used in the project

Anybody implementing such complex functions should be very careful. Any little mistake can ruin all the work done.

Step5-Simulation



Simulation of the single D flip-flop.

Simulation of the total project with all inputs and outputs. It works.

References

[1] Digital Systems, Topic 14: Vending Machine Design Problem, page 15 of 26, 8/18/2010

[2] Digital Design, FIFTH EDITION, M. Morris Mano, Michael D. Ciletti, Section 5.8, Design Procedure 239

[3] Digital Design, FIFTH EDITION, M. Morris Mano, Michael D. Ciletti, Section 5.8, Design Procedure 239

[4] Digital Design, FIFTH EDITION, M. Morris Mano, Michael D. Ciletti, Section 5.8, Design Procedure 239

[5] http://www.32x8.com/var6.html