CS M51A and EE M16 Spring 2015 Section 1

Logic Design of Digital Systems

Dr. Yutao He

Verilog Lab #2 - Design of Combinational Systems

Due: May 13rd, 2015

(1) Name: Wu, Ying Bin
Student ID: 104485521

Signature: Ying Bin Wu

(2) Name: Yuan, Qing
Student ID: 104009336

Signature: Qing Yuan

Date: 5/13/15

Verilog Lab #2 Project Requirement

Dr. Yutao He

Due: 5/13/2015

Abstract
The purpose of this project is to convert a BCD into a seven-segment display decoder. The seven-segment display decoder is used to display a number by turning respective LEDs on or off depending on the BCD code.

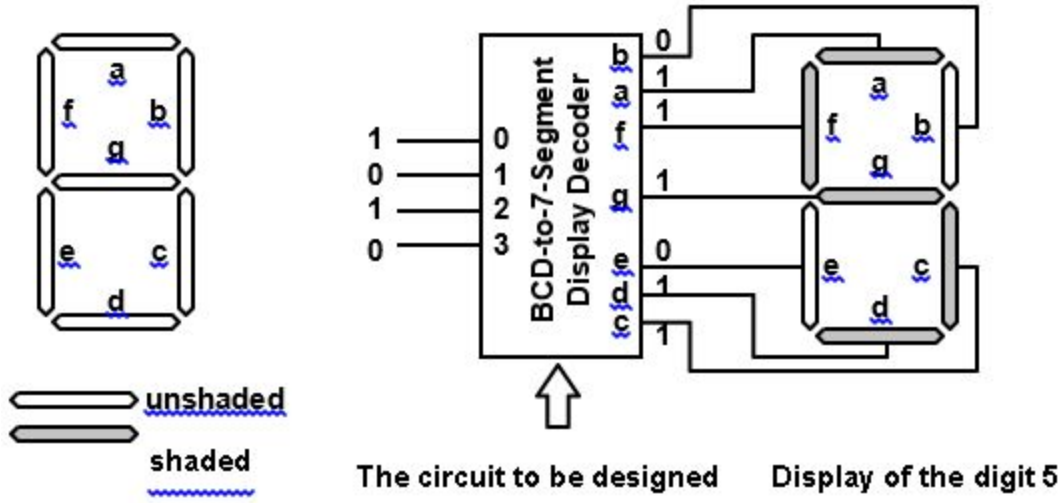
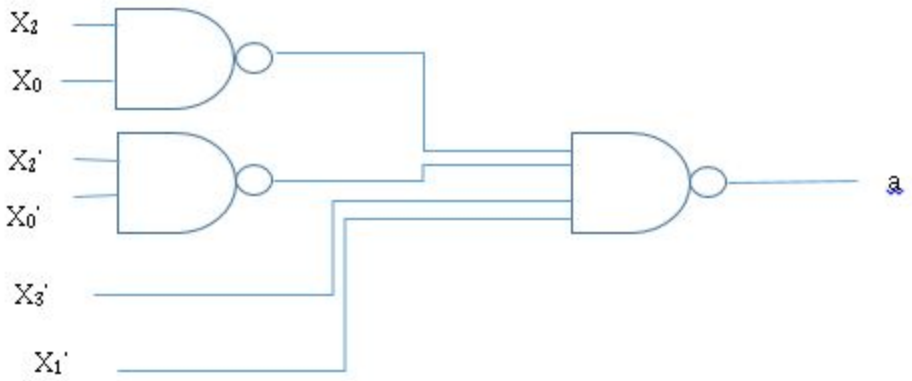
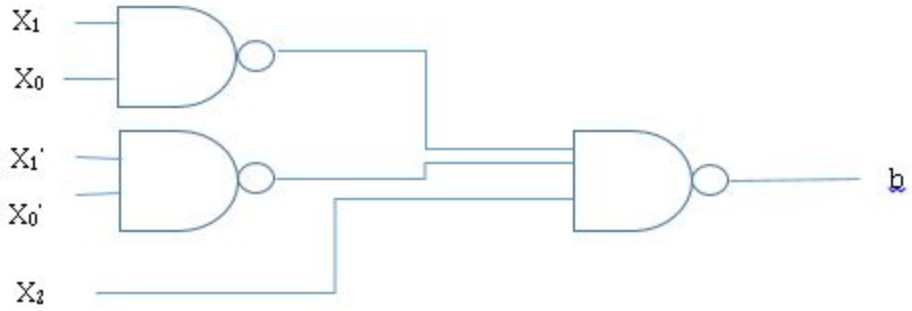


Figure 1: The Structure of the Seven-Segment Display Device

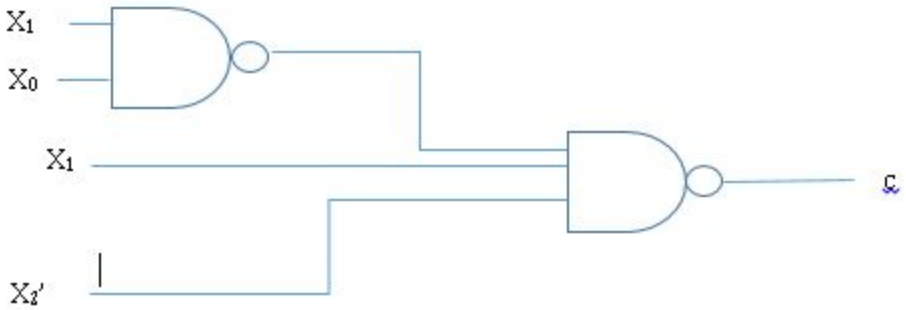
Switching Functions of the Circuit



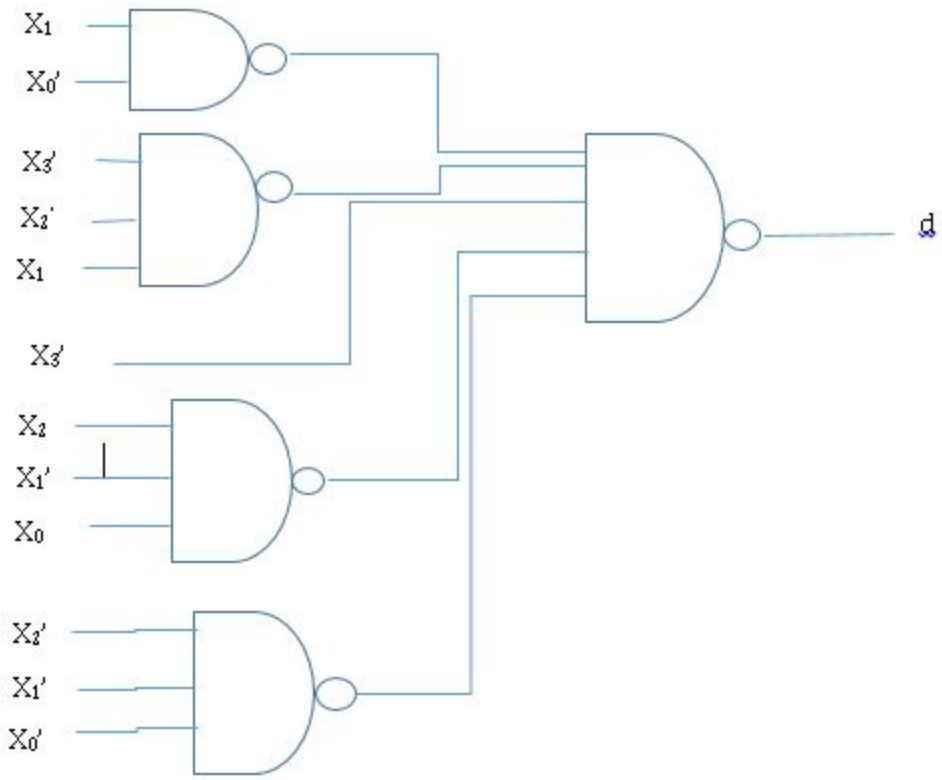
$a=((X_2X_0)'(X_2'X_0')'X_3'X_1')'$



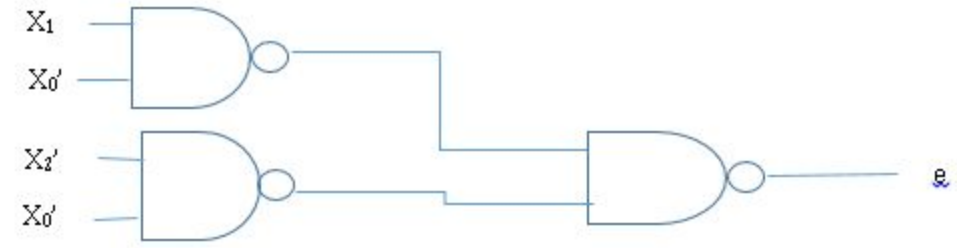
$b=((X_1X_0)'(X_1'X_0')'X_2)'$



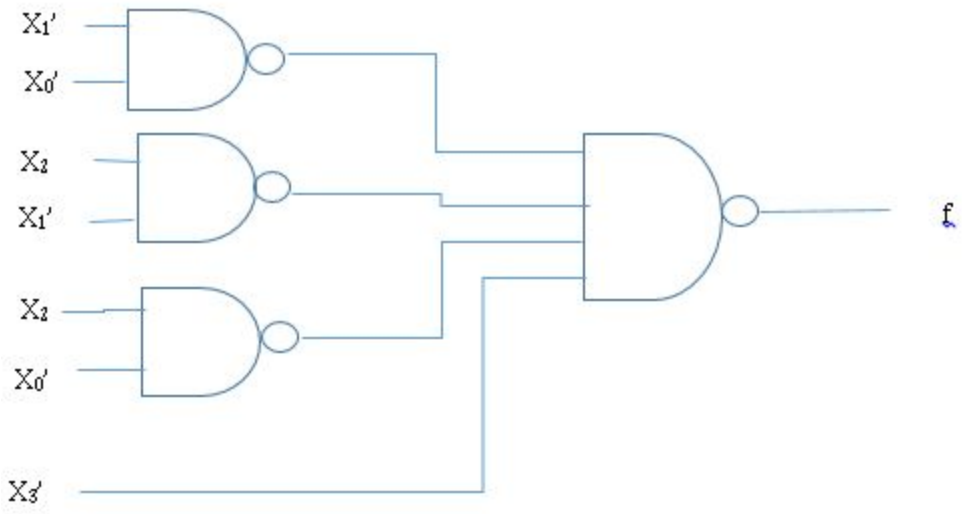
$c=((X_1X_0)'X_1X_2')'$



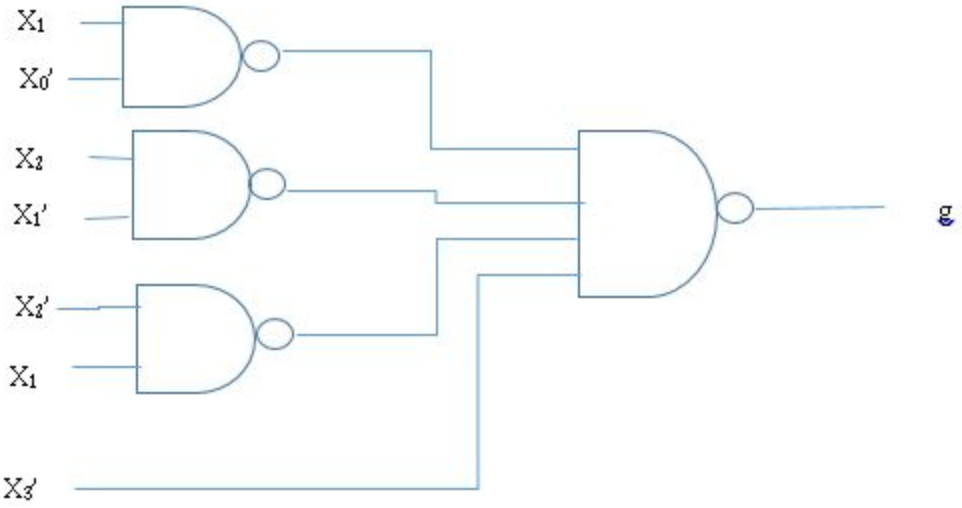
$d=((X_1X_0')'(X_3'X_2'X_1)'(X_2X_1'X_0)'(X_2'X_1'X_0')X_3')'$



$e=((X_1X_0')'(X_2'X_0'))'$



$f=((X_1'X_0')'(X_2X_1')'(X_2X_0')'X_3')'$



$g=((X_1X_0')'(X_2X_1')'(X_2'X_1)'X_3')'$

Verilog Code

module csm51a_proj2(X3, X2, X1, X0, A, B, C, D, E, F, G);

input X3;
input X2;
input X1;

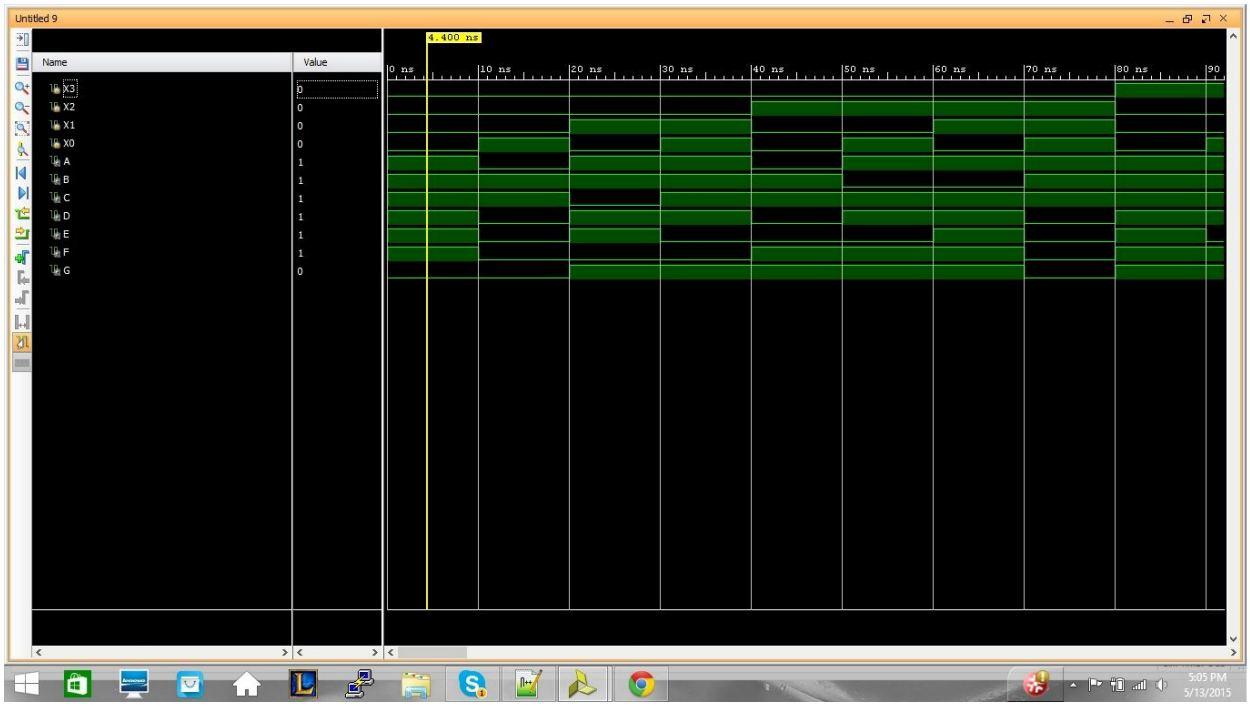
```
input X0;
output A;
output B;
output C;
output D;
output E;
output F;
output G;

assign NAND1 = ~(~X2 & ~X0);
assign NAND2 = ~(X1 & X0);
assign NAND3 = ~(X2 & X0);
assign NAND4 = ~(~X1 & ~X0);
assign NAND5 = ~(X1 & ~X0);
assign NAND6 = ~(~X3 & ~X2 & X1);
assign NAND7 = ~(X2 & ~X1 & X0);
assign NAND8 = ~(~X2 & ~X1 & ~X0);
assign NAND9 = ~(X2 & ~X1);
assign NAND10 = ~(X2 & ~X0);
assign NAND11 = ~(~X2 & X1);


assign A = ~((NAND1) & (NAND3) & (~X3) & (~X1));
assign B = ~((NAND2) & (NAND4) & (X2));
assign C = ~((NAND2) & (~X2) & (X1));
assign D = ~((NAND5) & (NAND6) & (NAND7) & (NAND8) & (~X3));
assign E = ~((NAND1) & (NAND5));
assign F = ~((NAND4) & (NAND9) & (NAND10) & (~X3));
assign G = ~((NAND9) & (NAND5) & (NAND11) & (~X3));

endmodule
```

Simulation Result

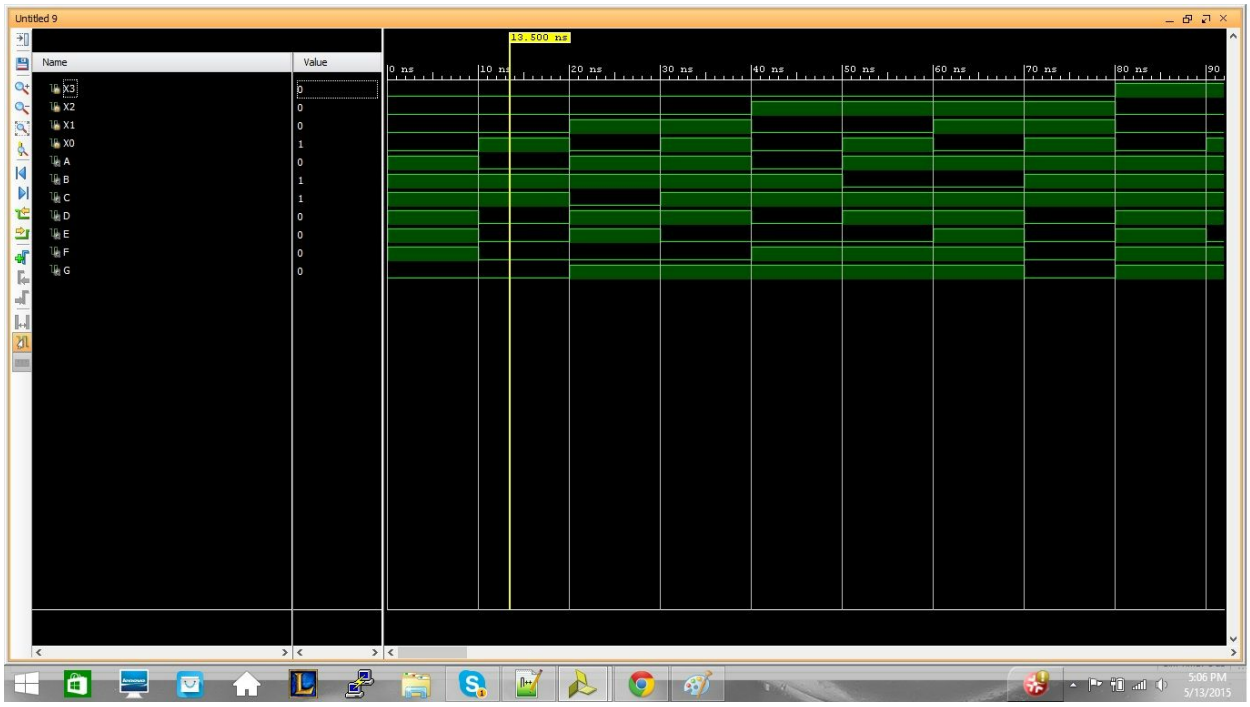
The simulation demonstrates the output values of a, b, c, d, e, f, and g based on the 4 input values x3, x2, x1, x0. The output values is responsible for lighting up the seven-segment display in a way to represent decimal numbers 0 through 9. Below are the simulation results.

*Note**

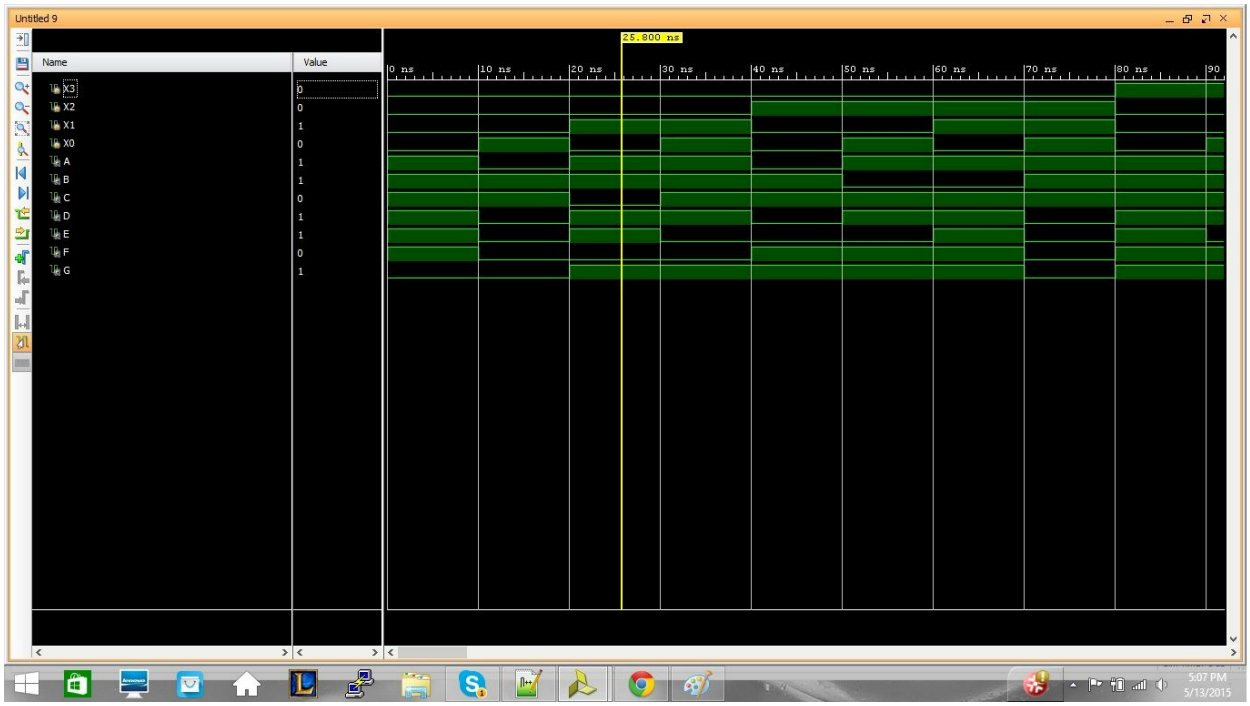
Input = (x3, x2, x1, x0), output = (a, b, c, d, e, f, g)



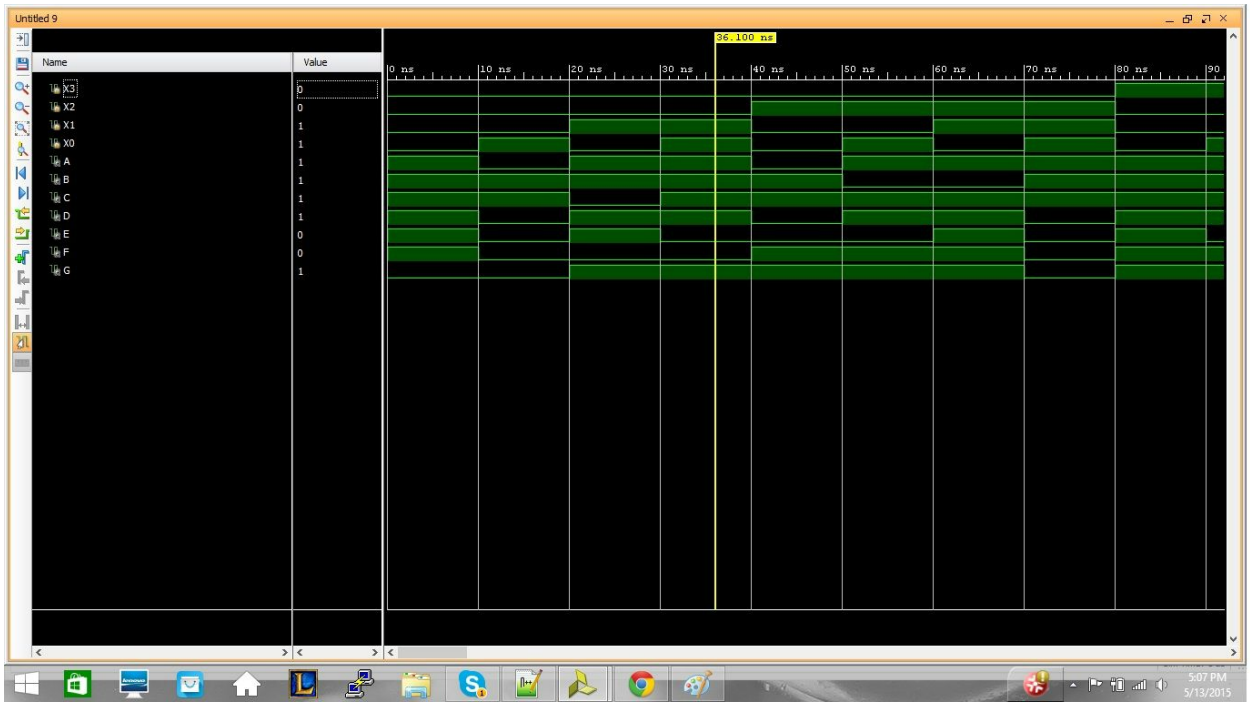
Input = (0, 0, 0, 0), Output = (1, 1, 1, 1, 1, 1, 0)



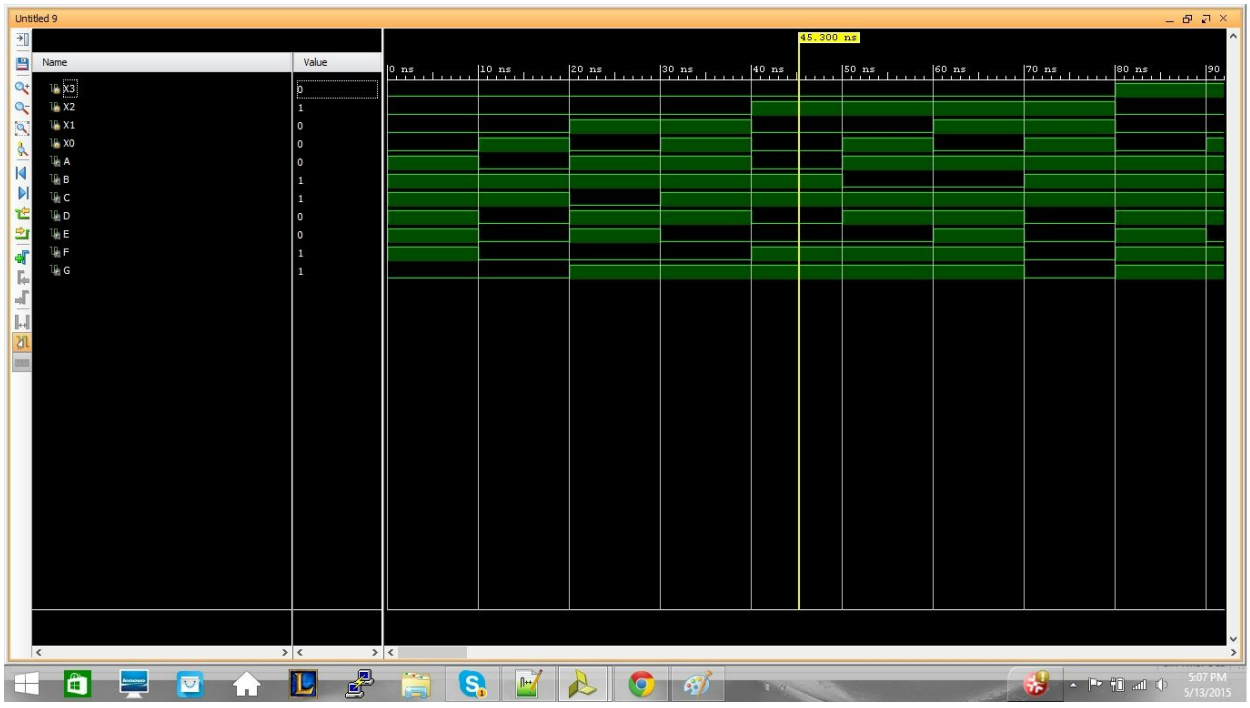
Input = (0, 0, 0, 1), Output = (0, 1, 1, 0, 0, 0, 0)



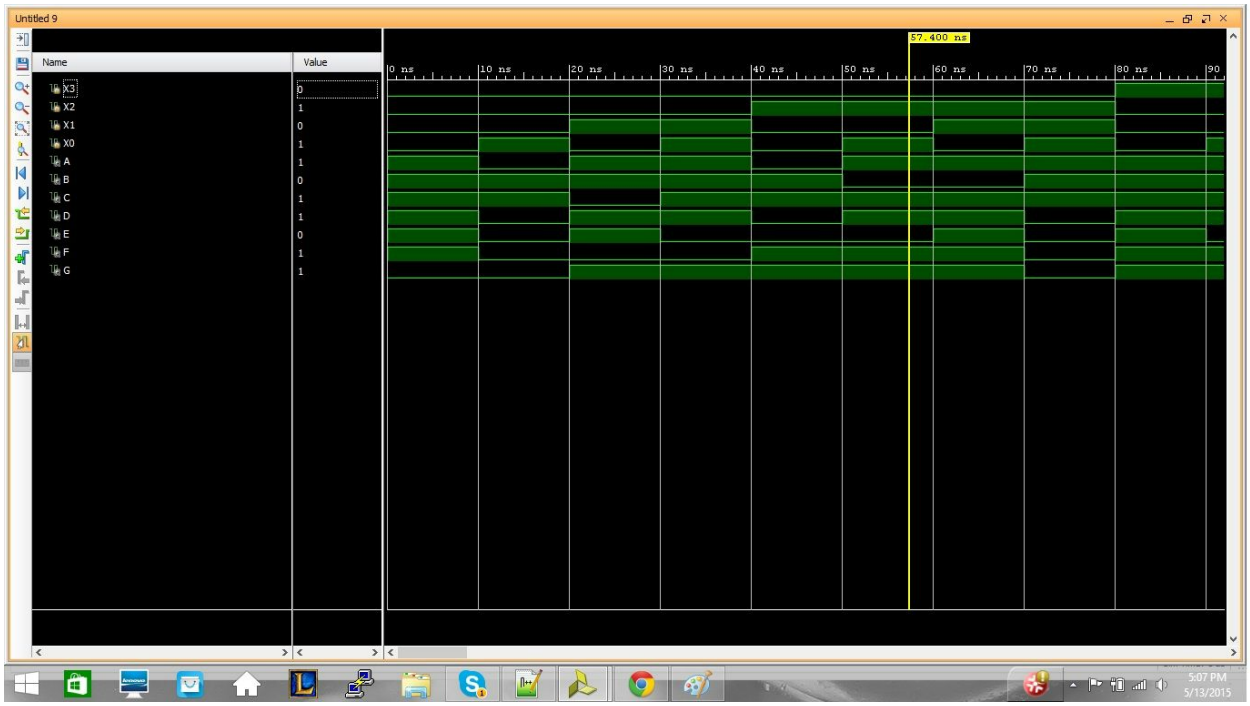
Input = (0, 0, 1, 0), Output = (1, 1, 0, 1, 1, 0, 1)



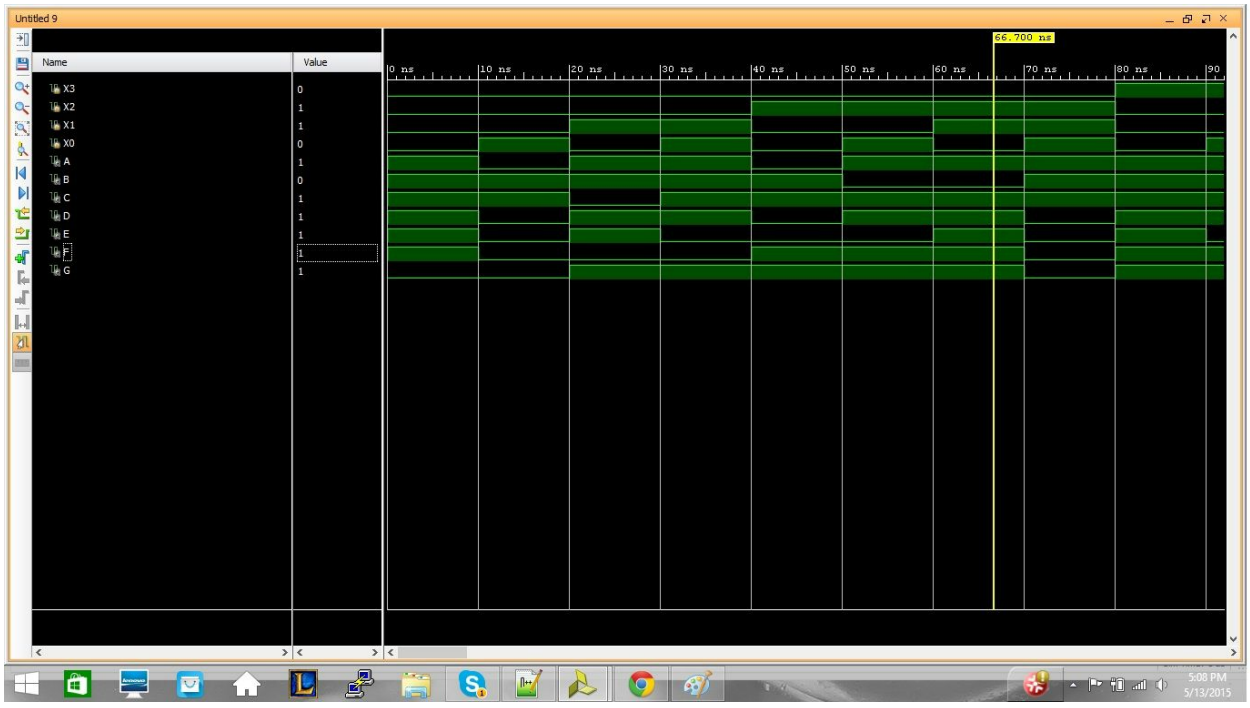
Input = (0, 0, 1, 1), Output = (1, 1, 1, 1, 0, 0, 1)



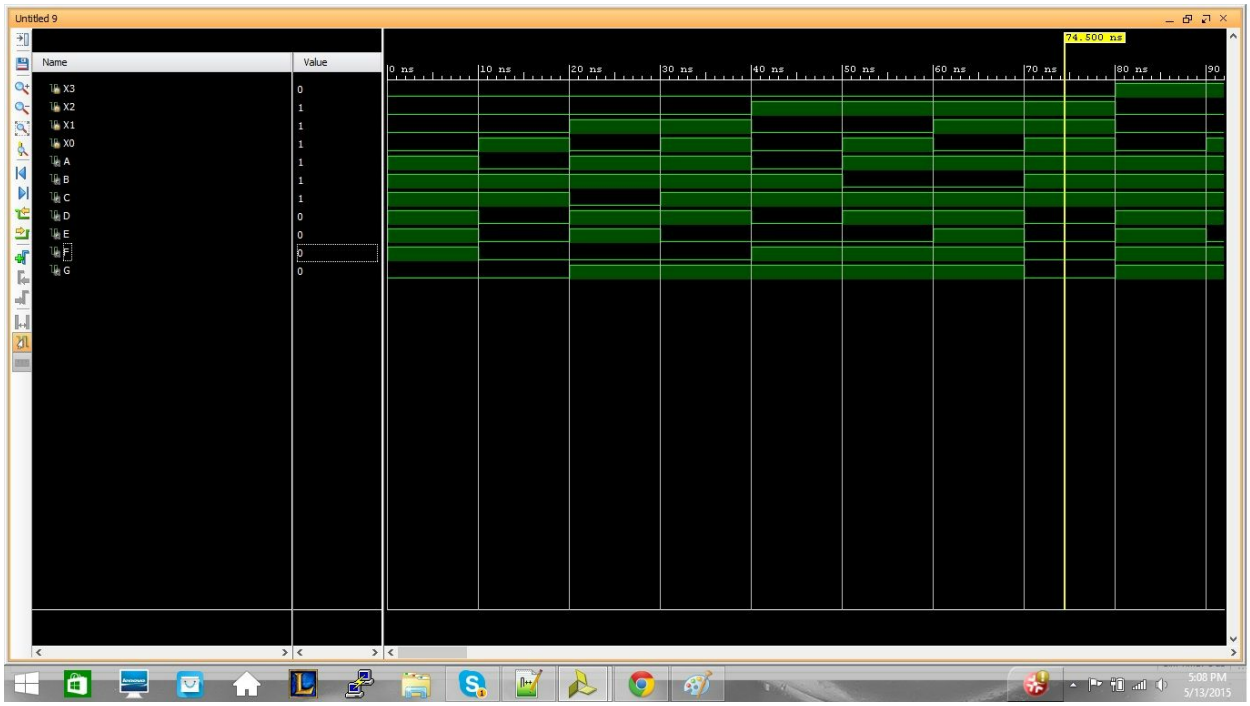
Input = (0, 1, 0, 0), Output = (0, 1, 1, 0, 0, 1, 1)



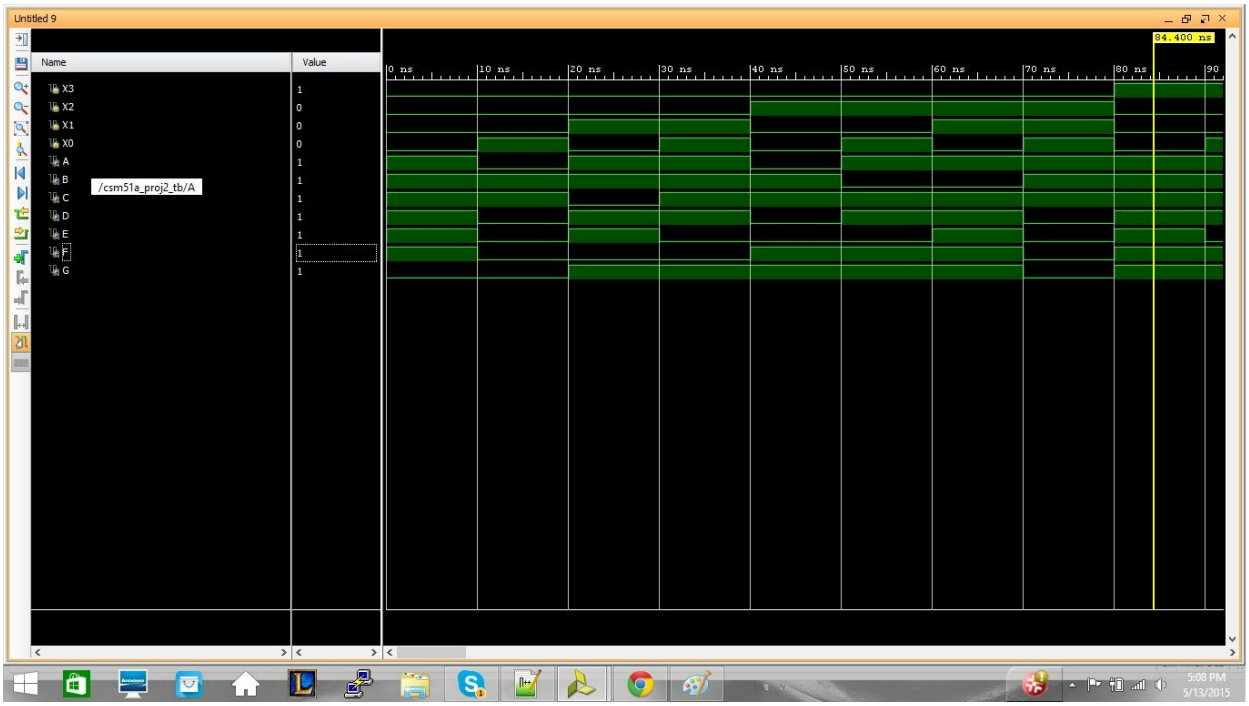
Input = (0, 1, 0, 1), Output = (1, 0, 1, 1, 0, 1, 1)



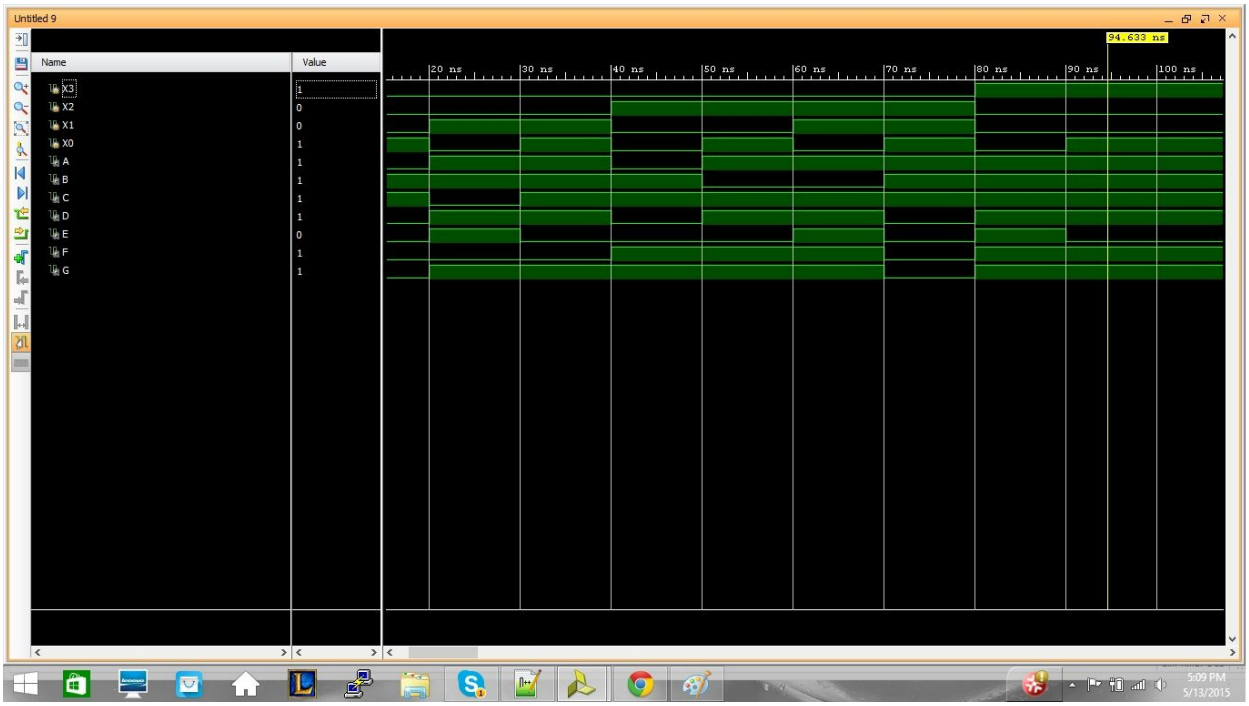
Input = (0, 1, 1, 0), Output = (1, 0, 1, 1, 1, 1, 1)



Input = (0, 1, 1, 1), Output = (1, 1, 1, 0, 0, 0, 0)



Input = (1, 0, 0, 0), Output = (1, 1, 1, 1, 1, 1, 1)



Input = (1, 0, 0, 1), Output = (1, 1, 1, 1, 0, 1, 1)

Design Review

This project was fairly simple but time consuming. The most challenging part was generating the switching functions and making sure everything was correct before actually coding in verilog. After simplifying the switching with K-maps, the coding part of the project was fairly straightforward. One problem that we encountered was actually writing the test bench function. We had a syntax error that was generating Z's for initial input values instead of the input values we assigned to x3, x2, x1, and x0. Other than that, the rest of the project went smoothly.

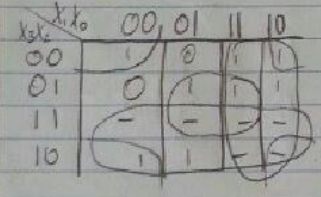
Team Member Contributions

The project as split evenly between us. We worked together on the design part and also the coding part. Ying Bin was responsible for writing out all the switching functions on paper while Qing was responsible for typing out all the code. The report is split into sections. Ying Bin was responsible for writing the abstract, switching functions, and team member contributions while Qing added the simulation results, design review, and finished the appendix.

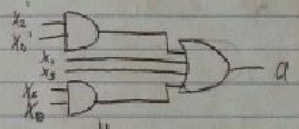
Appendix

	$x_3x_2x_1x_0$	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	1	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	1	0	1	1

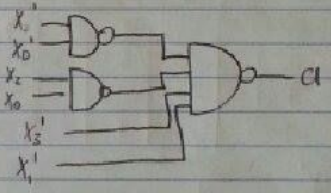
part a_0:



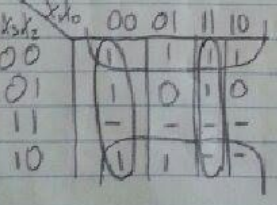
$x_3 + x_1 + x_2'x_0' + x_2x_0$



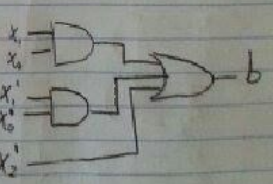
$((x_2'x_0')'(x_2x_0)' x_3' x_1')'$



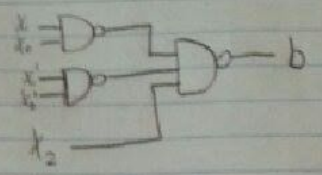
part a_0:



$x_2' + x_1'x_0' + x_1x_0$

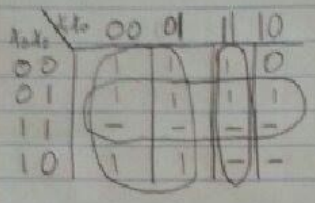


NAND-NAND next page

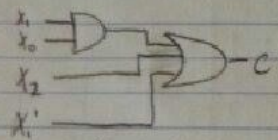


$((x_1x_0)'(x_1'x_0')'x_2)'$

part c:

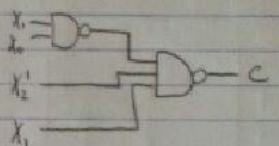


$x_1' + x_2 + x_1x_0$

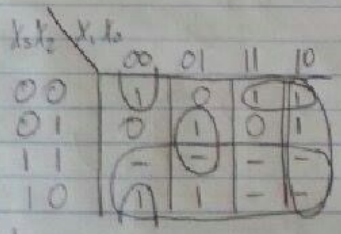


$\Downarrow$

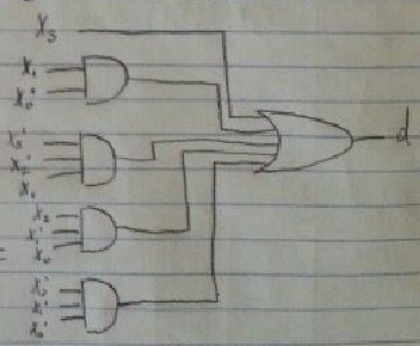
$((x_1x_0)'x_2'x_1)'$



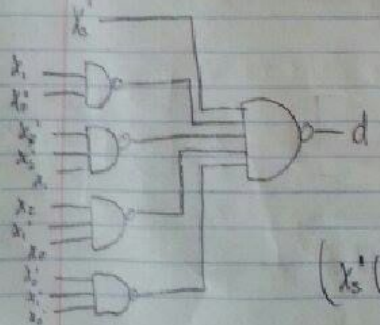
part d:



$x_3 + x_2x_0' + x_3'x_2'x_1 + x_2x_1'x_0 + x_2'x_1'x_0'$



$\Leftarrow$

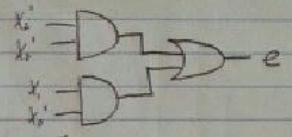


$(x_3'(x_1x_0')'(x_3'x_2'x_1)'(x_2x_1'x_0)'(x_2x_1'x_0'))'$

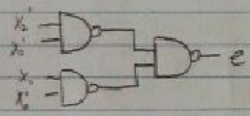
part e':

x_2x_3 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	0	0	0	1
11	-	-	-	-
10	1	0	-	-

$X_2'X_0' + X_1X_0'$



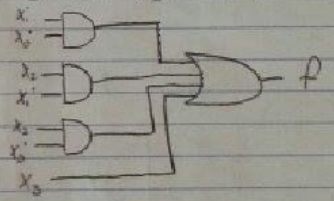
$((X_2'X_0')'(X_1X_0')')'$

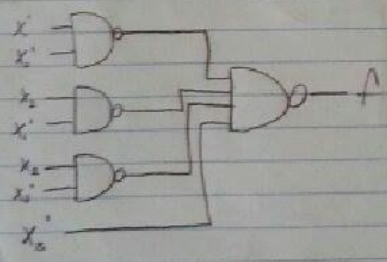


part f':

x_2x_3 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	-	-	-	-
10	1	1	-	-

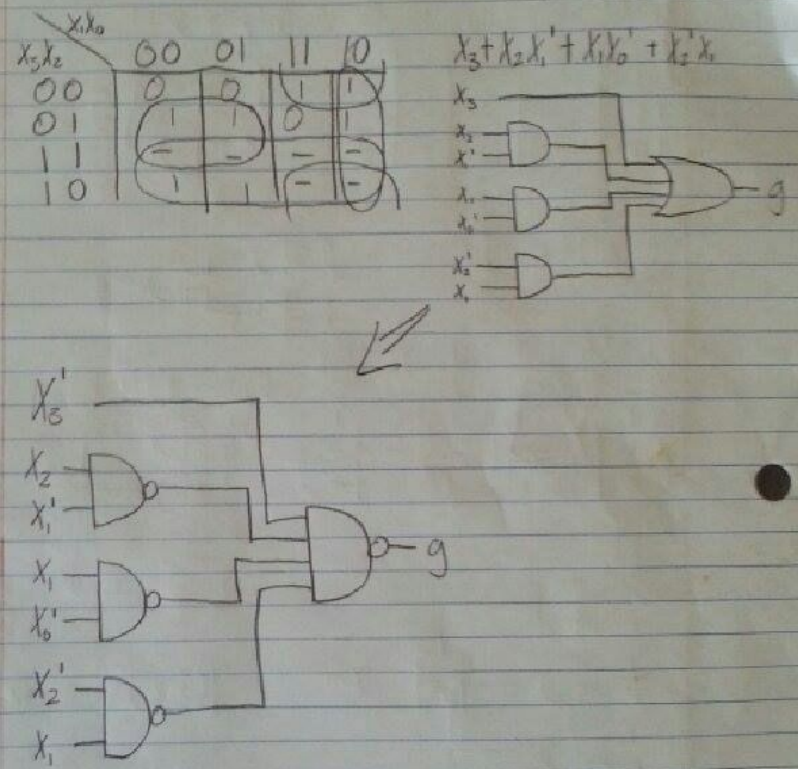
$X_3 + X_1'X_0' + X_2X_1' + X_2X_0'$





$((X_1'X_0')'(X_2X_1')'(X_2X_0')'X_3')'$

part g:


x_1x_0
x_3x_2
00 01 11 10
00
01
11
10
$x_3 + x_2x_1' + x_1x_0' + x_2'x_0$
x_3
x_2 x_1'
x_1 x_0'
x_2' x_0
g

x_3'
x_2 x_1'
x_1 x_0'
x_2' x_1
g