Digital Logic Design
CSE345(4)

Mini Project

“ 3 bit Binary Square Calculator Circuit ”

Prepared for

Touhid Ahmed
Lecturer
Department of Computer Science & Engineering
East West University

Prepared by

Maksudul Haque Chowdhury	**2019-1-60-182**
Md Jakir Hossain	**2019-1-60-163**
Plabon Banik	**2019-1-60-167**

Department of Computer Science and Engineering
East West University

Square calculator

P7: 3 bit Binary Square Calculator Circuit: Design a combinational logic circuit, that takes a 3 bit

binary number as input and output will be square of the input number. For example, if the input number

is (5)10=(101)2, then the output will be (25)10 = (11001)2

Objective

Here we will have design a combinational logic circuit which takes input of 3bit binary number and gives an output of the square of that number. So the highest value in 3bit can be 111 which is 7 in decimal and we should have 49 as the output which is a 6bit binary number. So the output will be of 6bits.

Working Procedure

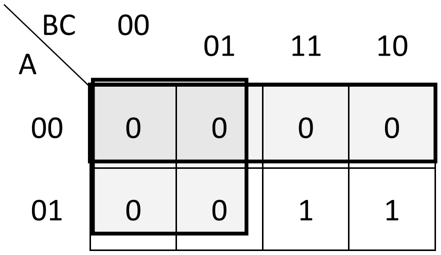
Now if we draw the truth table based on the objective we get,

SL	A	B	C	F5	F4	F3	F2	F1	F0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	1	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	1	0	0
7	1	1	1	1	1	0	0	0	1

For F5 we get the Kmap ,

A \ BC	00	01	11	10
00	0	0	0	0
01	0	0	1	1

SOP, F(A,B,C)=AB, and cost =1+2=3

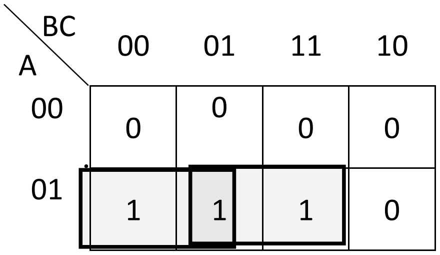


A \ BC	00	01	11	10
00	0	0	0	0
01	0	0	1	1

POS, F(A,B,C)=(A)(B) where cost=1+2=3

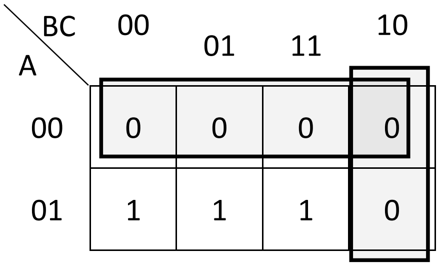
Now the most simplified is F(A,B,C)=AB.

For F4 we get the Kmap ,



A \ BC	00	01	11	10
00	0	0	0	0
01	1	1	1	0

SOP, F(A,B,C)=AB’ + AC, and cost =3+2+2+2=9

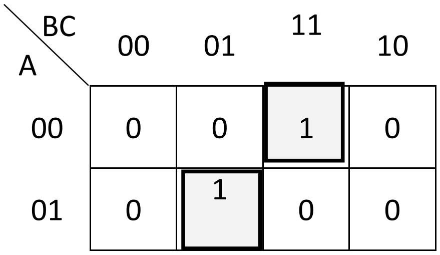


A \ BC	00	01	11	10
00	0	0	0	0
01	1	1	1	0

POS, F(A,B,C)=(A)(B’+C) where cost=2+2+2=6

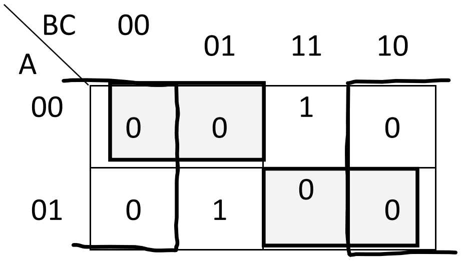
Now the most simplified is F(A,B,C)=(A)(B’+C)

For F3 we get the Kmap ,



BC / A	00	01	11	10
00	0	0	1	0
01	0	1	0	0

SOP, F(A,B,C)=A’BC + AB’C, and cost =3+3+3+2=11

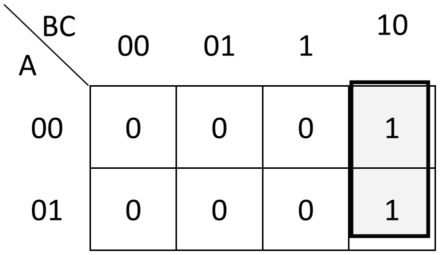


BC / A	00	01	11	10
00	0	0	1	0
01	0	1	0	0

POS, F(A, B, C) = (A + B)(C)(A' + B') where cost=3+2+2+3=10

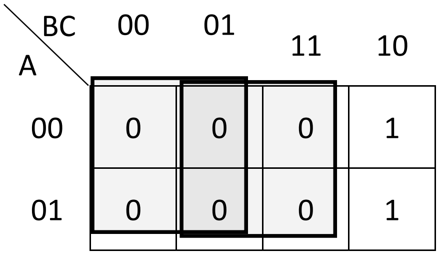
Now the most simplified is F(A, B, C) = (A + B)(C)(A' + B')

For F2 we get the Kmap ,



BC / A	00	01	1	10
00	0	0	0	1
01	0	0	0	1

SOP, F(A,B,C)=BC’, and cost =1+2=3



BC / A	00	01	11	10
00	0	0	0	1
01	0	0	0	1

POS, F(A, B, C) =(B)(C’) where cost=1+2=3

Now the most simplified is F(A,B,C)=BC’

For F1 we get the Kmap ,

BC A	00	01	1	10
00	0	0	0	0
01	0	0	0	0

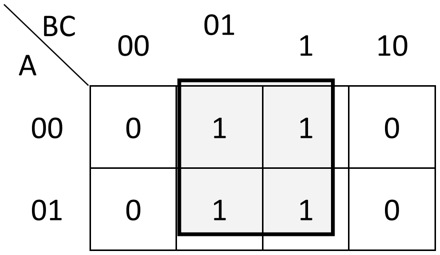
SOP, F(A,B,C)=0, and cost =0

BC A	00	01	11	10
00	0	0	0	0
01	0	0	0	0

POS, F(A, B, C) = 0 where cost= 0

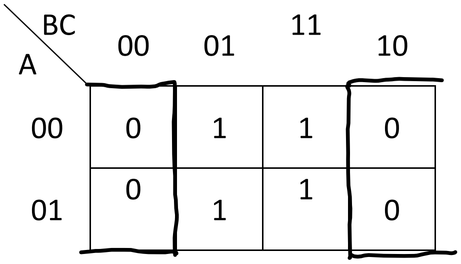
Now the most simplified is F(A, B, C) = 0

For F0 we get the Kmap ,



BC A	00	01	1	10
00	0	1	1	0
01	0	1	1	0

SOP, F(A,B,C)=C, and cost = 0



BC A	00	01	11	10
00	0	1	1	0
01	0	1	1	0

POS, F(A, B, C) = C where cost=0

Now the most simplified is F(A, B, C) = C

Expression Analysis

Now for all the outputs we get,

F5(A,B,C)=AB.

F4(A,B,C)=(A)(B’+C)

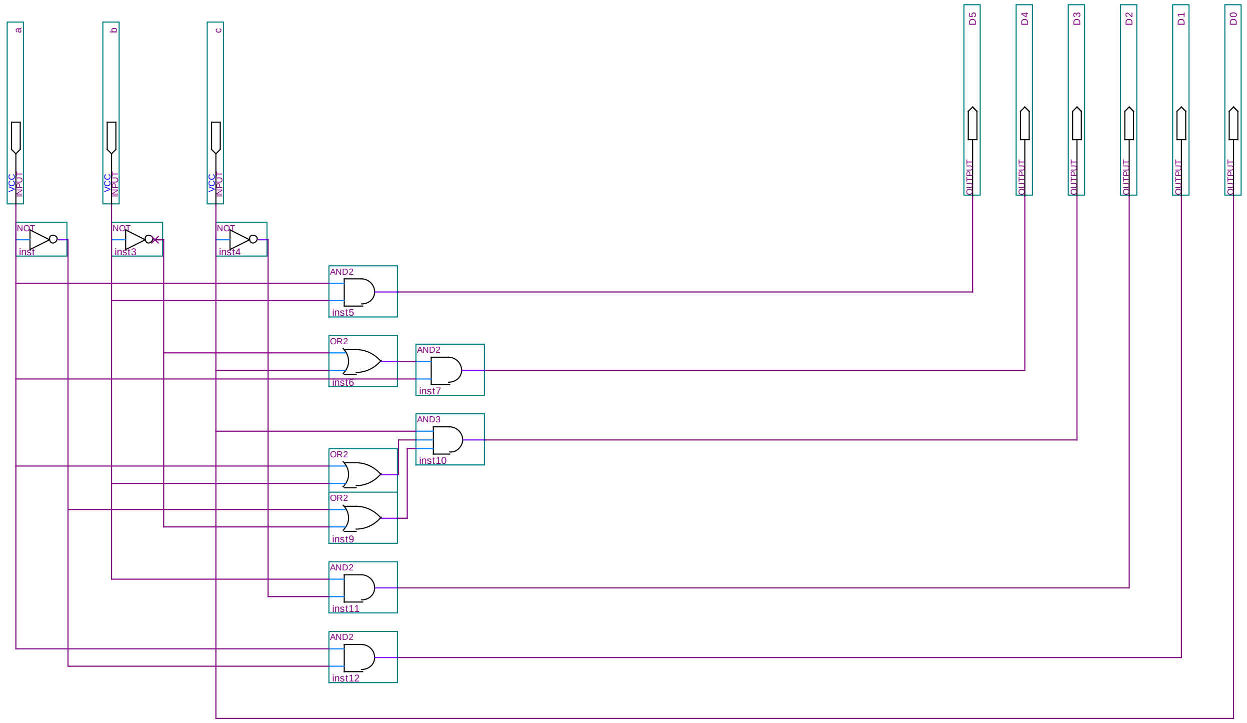
F3(A, B, C) = (A + B)(C)(A' + B')

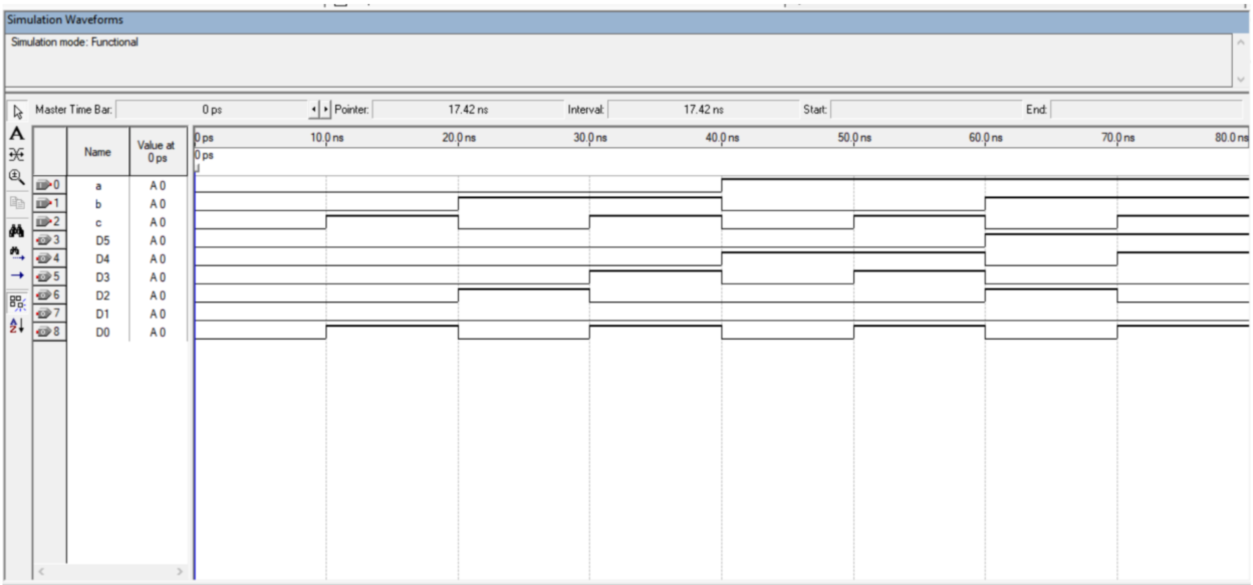
F2(A,B,C)=BC’

F1(A, B, C) = 0

F0(A, B, C) = C

Now if we draw the schematics from this expressions we get




Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 0 ps Pointer: 17.42 ns Interval: 17.42 ns Start: End:

	Name	Value at 0 ps
0	a	A 0
1	b	A 0
2	c	A 0
3	D5	A 0
4	D4	A 0
5	D3	A 0
6	D2	A 0
7	D1	A 0
8	D0	A 0

Now if we analyze the output we see that,

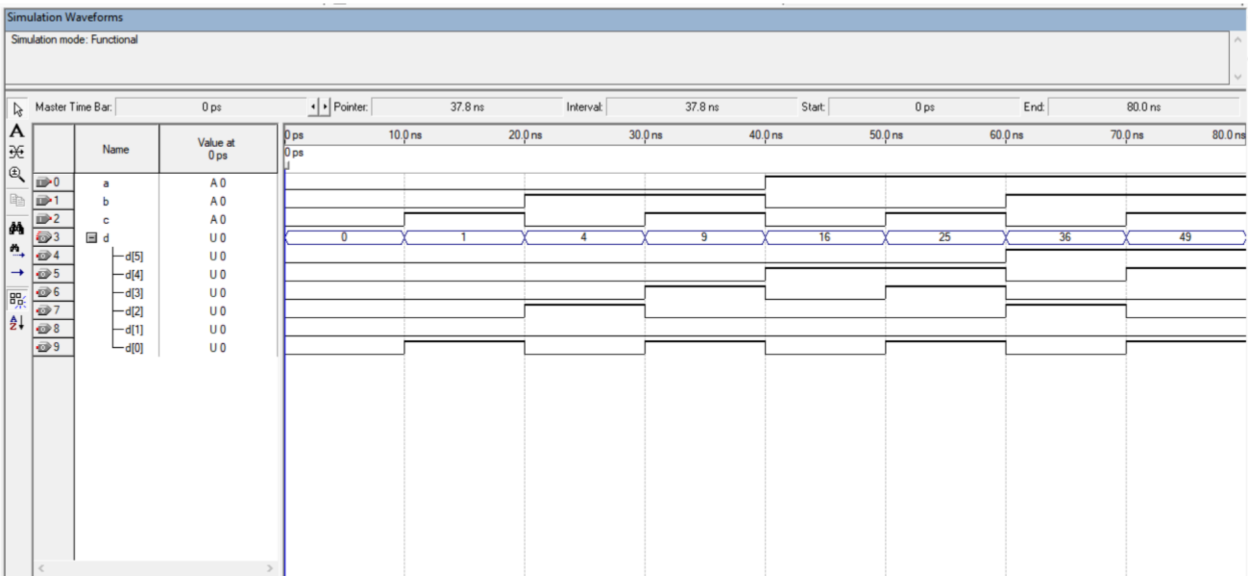
Schematic Simulation Analysis

Input (Binary)	Expected output (Binary)	Simulated output (Binary, $D_5D_4D_3D_2D_1D_0$)	Output Value in Decimal
000	000000	000000	0
001	000001	000001	1
010	000100	000100	4
011	001001	001001	9
100	010000	010000	16
101	011001	011001	25
110	100100	100100	36
111	110001	110001	49

From the truth table of we can write the behavioral Verilog code like this,

```
module nsqr(a,b,c,d);
      input a,b,c;
      output reg [5:0]d;
      always @ (a,b,c)
            begin
                  case({a,b,c})
                        3'b000:d=6'b000000;
                        3'b001:d=6'b000001;
                        3'b010:d=6'b000100;
                        3'b011:d=6'b001001;
                        3'b100:d=6'b010000;
                        3'b101:d=6'b011001;
                        3'b110:d=6'b100100;
                        3'b111:d=6'b110001;
                  endcase
            end
endmodule
```

After simulation the result is like this,



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 0 ps   Pointer: 37.8 ns   Interval: 37.8 ns   Start: 0 ps   End: 80.0 ns

	Name	Value at 0 ps
0	a	A 0
1	b	A 0
2	c	A 0
3	d	U 0
4	d[5]	U 0
5	d[4]	U 0
6	d[3]	U 0
7	d[2]	U 0
8	d[1]	U 0
9	d[0]	U 0

Verilog Simulation Analysis

Input	Expected Output	Simulated Result
0	0	0
1	1	1
2	4	4
3	9	9
4	16	16
5	25	25
6	36	36
7	49	49

We can see that the simulated and expected outcomes are exactly similar.

Decision:

We see that the outcomes we are having are exactly the way we expected. So we can say that our circuit is behaving as expected and there are no major or minor error which could be marked as crucial breaking point for the design.

Discussion:

The circuit we designed here is a 3bit binary input and 6bit binary output circuit. Which takes a 3bit number (0-7) and returns the square value of that number(0-49). This kind of circuits are used in many cases like multiplication processes, financial mathematical processes and in so many other cases.

But here we see that the prime limitation is, it cannot take number larger than 7 or $(111)_2$ as input. Also its only taking unsigned numbers as input signed numbers are not being calculated. So these are the two prime limitation of our circuit otherwise this design is efficient and error free which is evident from the analysis shown above