Project 2

Problem statement: We will have to design a BCD to Express-3 code generator. Where it will have 4 bit input (a,b,c and d) and 4 bit output (w,x,y and z).

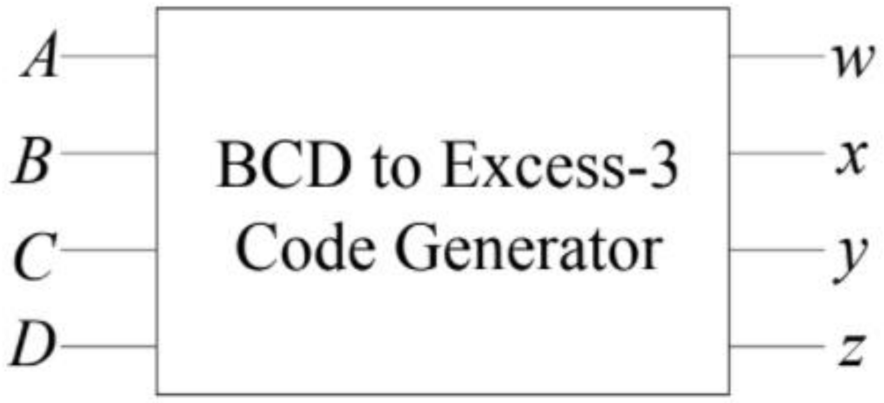


Fig. 2

The access-3 code generator will take 4bit input by its 4 input pins and the output pin will give the output in binary value. As it is a BCD access-3 code generator, it will take input between 0 and 9; for input 0(0000),1(0001),2(0010) output will be 3(0011),4(0100),5(0101) respectively.

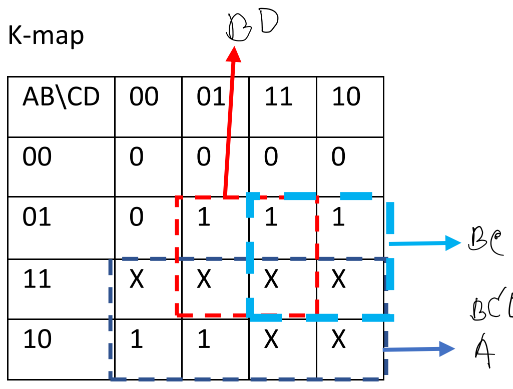
At first, we will design the input-output table

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	🟦	🟦	🟦	🟦
1	1	1	0	🟦	🟦	🟦	🟦
1	1	1	1	🟦	🟦	🟦	🟦

Valid decimal input range

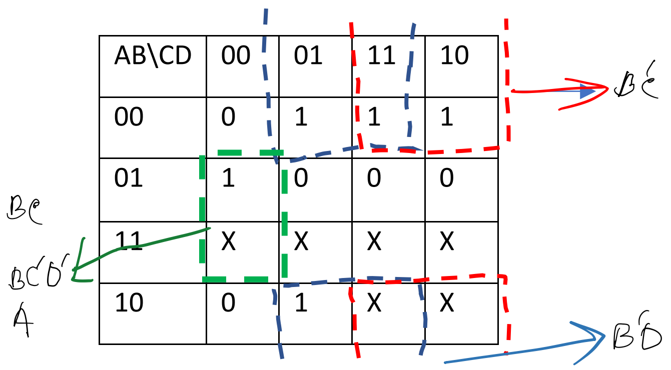
After that, for every output (w,x,y and z) we will have to find out k-map expression with respect to input (A,B,C and D)

K-map



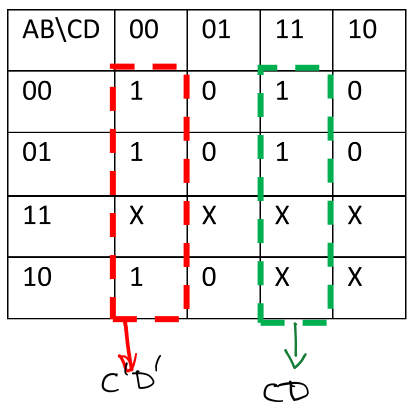
AB\CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

w=A+BC+BD



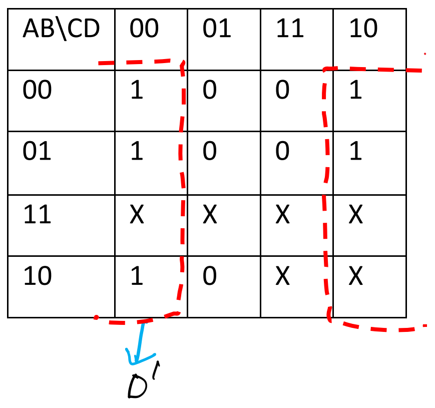
AB\CD	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

x=B’C+B’D+BC’D’



AB\CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

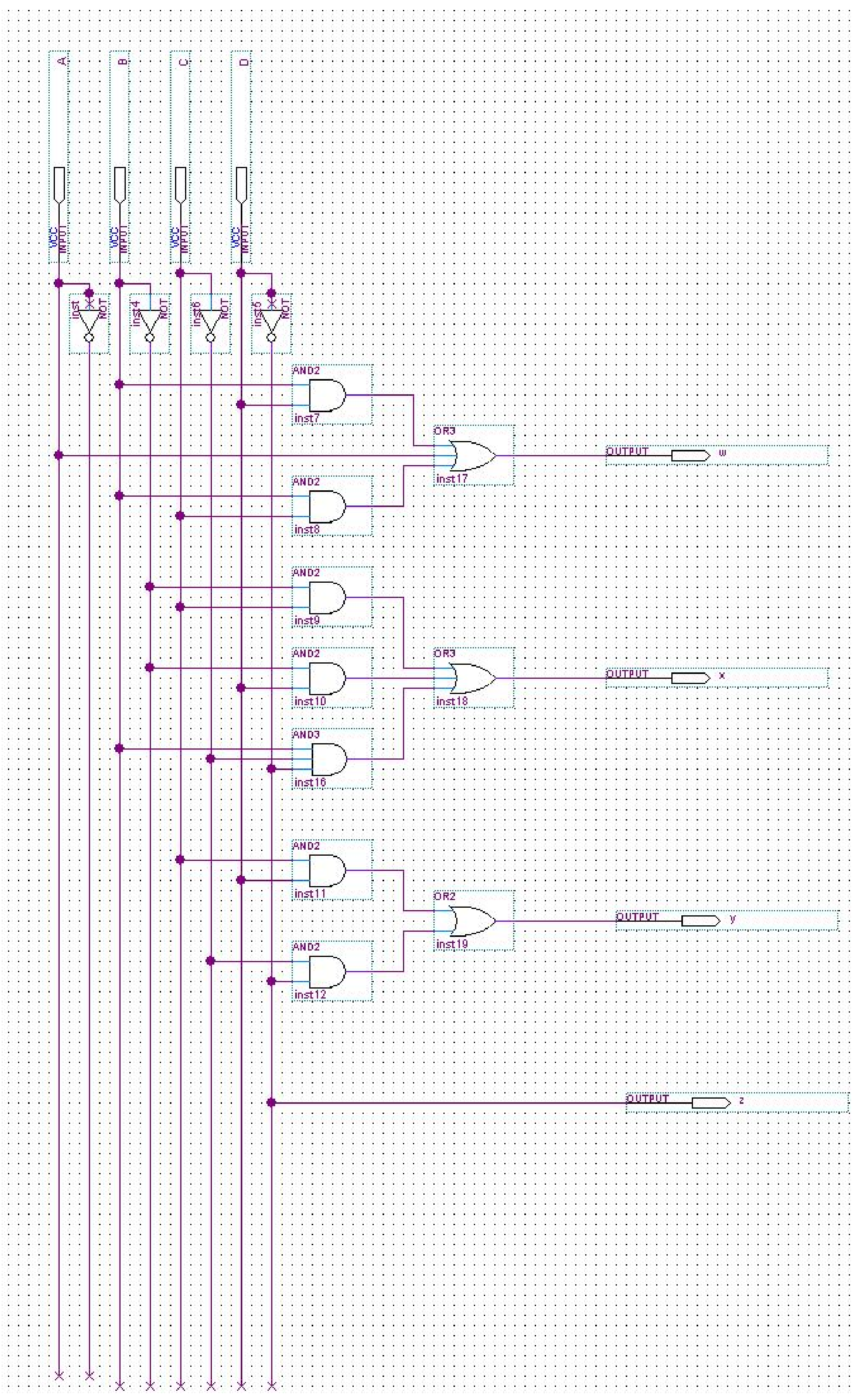
Y=CD+ C’D’



AB\CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

Z=D’

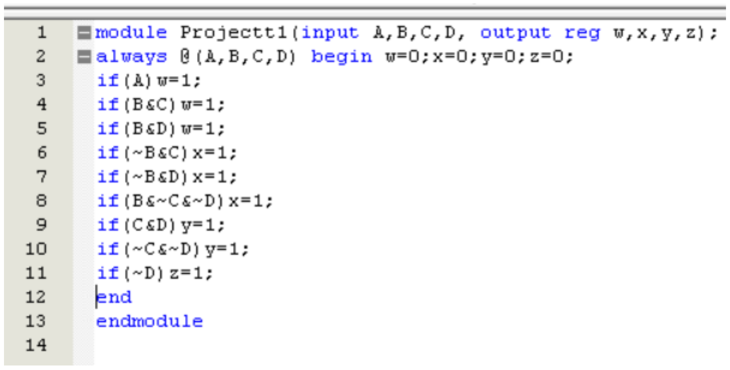
Circuit Design: After finding out the K-Map expression, we will have to make the circuit based on it-expression.


A
B
C
D
VCC INPUT
VCC INPUT
VCC INPUT
VCC INPUT
inst NOT
inst4 NOT
inst6 NOT
inst5 NOT
AND2 inst7
OR3 inst17
OUTPUT w
AND2 inst8
AND2 inst9
OR3 inst18
OUTPUT x
AND2 inst10
AND3 inst16
AND2 inst11
OR2 inst19
OUTPUT y
AND2 inst12
OUTPUT z

After designing the circuit, we will have to make the Behavioral Verilog code

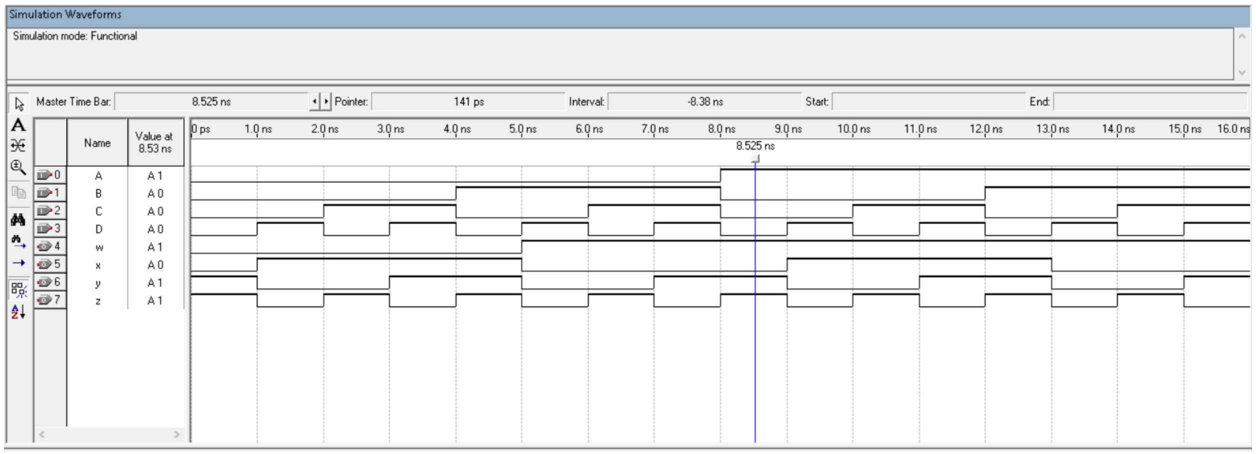
Behavioral Verilog code:

Procedural Verilog Code:



```
module Projectt1(input A,B,C,D, output reg w,x,y,z);
always @(A,B,C,D) begin w=0;x=0;y=0;z=0;
 if(A) w=1;
 if(B&C) w=1;
 if(B&D) w=1;
 if(~B&C) x=1;
 if(~B&D) x=1;
 if(B&~C&~D) x=1;
 if(C&D) y=1;
 if(~C&~D) y=1;
 if(~D) z=1;
end
endmodule
```

Vector Wave form:



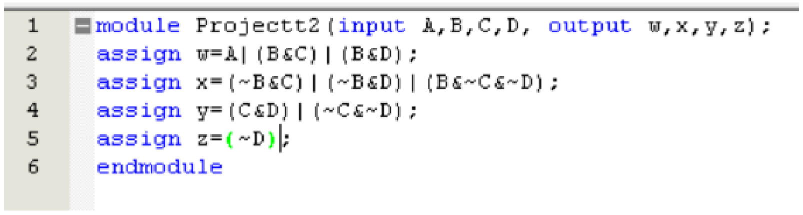
Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 8.525 ns Pointer: 141 ps Interval: -8.38 ns Start: End:

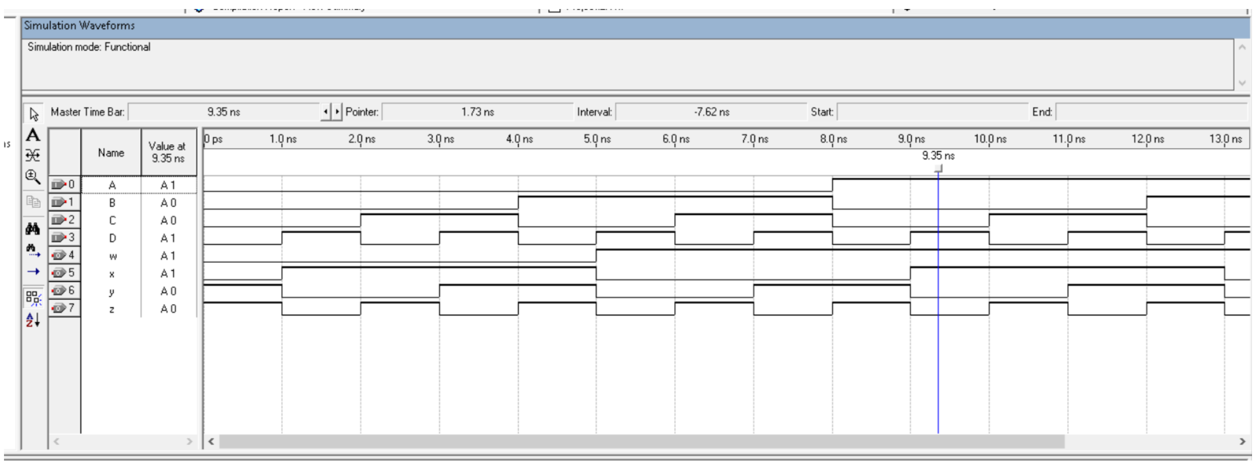
	Name	Value at 8.53 ns
0	A	A 1
1	B	A 0
2	C	A 0
3	D	A 0
4	w	A 1
5	x	A 0
6	y	A 1
7	z	A 1

Continuous Assign statement



```
module Projectt2(input A,B,C,D, output w,x,y,z);
 assign w=A|(B&C)|(B&D);
 assign x=(~B&C)|(~B&D)|(B&~C&~D);
 assign y=(C&D)|(~C&~D);
 assign z=(~D);
endmodule
```

Vector Wave form:



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 9.35 ns Pointer: 1.73 ns Interval: -7.62 ns Start: End:

	Name	Value at 9.35 ns
0	A	A 1
1	B	A 0
2	C	A 0
3	D	A 1
4	w	A 1
5	x	A 1
6	y	A 0
7	z	A 0