

[illegible]

## 16 to 4 Encoder:

I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	D3	D2	D1	D0
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1
0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	0
0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	0	1
0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	1	1	0	0
0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	1	0	1	1
0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	1	0	1	0
0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	1	0	0	1
0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	1	0	0	0
0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	0	1	1	1
0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	0	1	1	0
0	0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	1	X	X	X	X	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	X	X	X	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	X	X	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	X	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

## Codes:

**Decoder 4to16:**

```
22 module deco_4to16(i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15, z0, z1, z2, z3);
23 input z0,z1,z2,z3;
24 output reg i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15;
25 always@(z0,z1,z2,z3)
26 begin
27 if(z0 == 1'b0 && z1 == 1'b0 && z2 == 1'b0 && z3 == 1'b0)
28 begin
29 i0 = 1'b0;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
30 i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
31 i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
32 i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
33 end
34 else if(z0 == 1'b0 && z1 == 1'b0 && z2 == 1'b0 && z3 == 1'b1)
35 begin
36 i0 = 1'b1;i1 = 1'b0;i2 = 1'b1;i3 = 1'b1;
37 i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
38 i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
39 i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
40 end
41 else if(z0 == 1'b0 && z1 == 1'b0 && z2 == 1'b1 && z3 == 1'b0)
42 begin
43 i0 = 1'b1;i1 = 1'b1;i2 = 1'b0;i3 = 1'b1;
44 i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
45 i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
46 i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
47 end
48 else if(z0 == 1'b0 && z1 == 1'b0 && z2 == 1'b1 && z3 == 1'b1)
49 begin
50 i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b0;
51 i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
52 i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
53 i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
54 end
55 else if(z0 == 1'b0 && z1 == 1'b1 && z2 == 1'b0 && z3 == 1'b0)
56 begin
57 i0 = 1'b1;i1 = 1'b1;i2 = 1'b0;i3 = 1'b1;
58 i4 = 1'b0;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
59 i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
60 i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
61 end
62 else if(z0 == 1'b0 && z1 == 1'b1 && z2 == 1'b0 && z3 == 1'b1)
63 begin
```

```
64    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
65    i4 = 1'b1;i5 = 1'b0;i6 = 1'b1;i7 = 1'b1;
66    i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
67    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
68 end
69 else if(z0 == 1'b0 && z1 == 1'b1 && z2 == 1'b1 && z3 == 1'b0)
70 begin
71    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
72    i4 = 1'b1;i5 = 1'b1;i6 = 1'b0;i7 = 1'b1;
73    i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
74    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
75 end
76 else if(z0 == 1'b0 && z1 == 1'b1 && z2 == 1'b1 && z3 == 1'b1)
77 begin
78    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
79    i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b0;
80    i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
81    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
82 end
83 else if(z0 == 1'b1 && z1 == 1'b0 && z2 == 1'b0 && z3 == 1'b0)
84 begin
85    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
86    i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
87    i8 = 1'b0;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
88    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
89 end
90 else if(z0 == 1'b1 && z1 == 1'b0 && z2 == 1'b0 && z3 == 1'b1)
91 begin
92    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
93    i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
94    i8 = 1'b1;i9 = 1'b0;i10 = 1'b1;i11 = 1'b1;
95    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
96 end
97 else if(z0 == 1'b1 && z1 == 1'b0 && z2 == 1'b1 && z3 == 1'b0)
98 begin
99    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
100    i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
101    i8 = 1'b1;i9 = 1'b1;i10 = 1'b0;i11 = 1'b1;
102    i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
103 end
104 else if(z0 == 1'b1 && z1 == 1'b0 && z2 == 1'b1 && z3 == 1'b1)
105 begin
106    i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
107    i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
```

```
108     i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b0;
109     i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
110 end
111 else if(z0 == 1'b1 && z1 == 1'b1 && z2 == 1'b0 && z3 == 1'b0)
112 begin
113     i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
114     i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
115     i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
116     i12 = 1'b0;i13 = 1'b1;i14 = 1'b1;i15 = 1'b1;
117 end
118 else if(z0 == 1'b1 && z1 == 1'b1 && z2 == 1'b0 && z3 == 1'b1)
119 begin
120     i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
121     i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
122     i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
123     i12 = 1'b1;i13 = 1'b0;i14 = 1'b1;i15 = 1'b1;
124 end
125 else if(z0 == 1'b1 && z1 == 1'b1 && z2 == 1'b1 && z3 == 1'b0)
126 begin
127     i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
128     i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
129     i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
130     i12 = 1'b1;i13 = 1'b1;i14 = 1'b0;i15 = 1'b1;
131 end
132 else
133 begin
134     i0 = 1'b1;i1 = 1'b1;i2 = 1'b1;i3 = 1'b1;
135     i4 = 1'b1;i5 = 1'b1;i6 = 1'b1;i7 = 1'b1;
136     i8 = 1'b1;i9 = 1'b1;i10 = 1'b1;i11 = 1'b1;
137     i12 = 1'b1;i13 = 1'b1;i14 = 1'b1;i15 = 1'b0;
138 end
139 end
140 endmodule
```

## Test bench code:

```
module tb_deco4to16;
reg z0,z1,z2,z3;
wire i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15;
deco_4to16 m3(i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15, z0, z1, z2, z3);
initial
begin
    z0 = 1'b0; z1 = 1'b0; z2 = 1'b0; z3 = 1'b0;
    #10 z0 = 1'b0; z1 = 1'b0; z2 = 1'b0; z3 = 1'b1;
    #10 z0 = 1'b0; z1 = 1'b0; z2 = 1'b1; z3 = 1'b0;
    #10 z0 = 1'b0; z1 = 1'b0; z2 = 1'b1; z3 = 1'b1;
    #10 z0 = 1'b0; z1 = 1'b1; z2 = 1'b0; z3 = 1'b0;
    #10 z0 = 1'b0; z1 = 1'b1; z2 = 1'b0; z3 = 1'b1;
    #10 z0 = 1'b0; z1 = 1'b1; z2 = 1'b1; z3 = 1'b0;
    #10 z0 = 1'b0; z1 = 1'b1; z2 = 1'b1; z3 = 1'b0;
    #10 z0 = 1'b0; z1 = 1'b1; z2 = 1'b1; z3 = 1'b1;
    #10 z0 = 1'b1; z1 = 1'b0; z2 = 1'b0; z3 = 1'b0;
    #10 z0 = 1'b1; z1 = 1'b0; z2 = 1'b0; z3 = 1'b1;
    #10 z0 = 1'b1; z1 = 1'b0; z2 = 1'b1; z3 = 1'b0;
    #10 z0 = 1'b1; z1 = 1'b0; z2 = 1'b1; z3 = 1'b1;
    #10 z0 = 1'b1; z1 = 1'b1; z2 = 1'b0; z3 = 1'b0;
    #10 z0 = 1'b1; z1 = 1'b1; z2 = 1'b0; z3 = 1'b1;
    #10 z0 = 1'b1; z1 = 1'b1; z2 = 1'b1; z3 = 1'b0;
    #10 z0 = 1'b1; z1 = 1'b1; z2 = 1'b1; z3 = 1'b1;
end
endmodule
}
```

**16 to 4 Encoder:****Code:**

```
module Encoder16to4(i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14,i15,z0,z1,z2,z3);  
input i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14,i15;  
output z0,z1,z2,z3;  
orgate a1(i1,i3,i5,i7,i9,i11,i13,i15,z0);  
orgate a2(i2,i3,i6,i7,i10,i11,i14,i15,z1);  
orgate a3(i4,i5,i6,i7,i12,i13,i14,i15,z2);  
orgate a4(i8,i9,i10,i11,i12,i13,i14,i15,z3);  
endmodule  
  
module orgate(a,b,c,d,e,f,g,h,z);  
input a,b,c,d,e,f,g,h;  
output z;  
assign z = a | b | c | d | e | f | g | h;  
endmodule
```

## Test bench code:

```

module tb_Encoder16to4;
reg i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14,i15;
wire z0,z1,z2,z3;

Encoder16to4 m2(i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14,i15,z0,z1,z2,z3);

initial
begin

    i0 = 1'b1;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b1;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b1;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b1;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b1;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b1;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b1;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b1;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b1;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b1;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b1;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b1;i13 = 1'b0;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b1;i14 = 1'b0;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b1;i15 = 1'b0;
#10 i0 = 1'b0;i1 = 1'b0;i2 = 1'b0;i3 = 1'b0;i4 = 1'b0;i5 = 1'b0;i6 = 1'b0;i7 = 1'b0;i8 = 1'b0;i9 = 1'b0;i10 = 1'b0;i11 = 1'b0;i12 = 1'b0;i13 = 1'b0;i14 = 1'b0;i15 = 1'b1;
end
endmodule

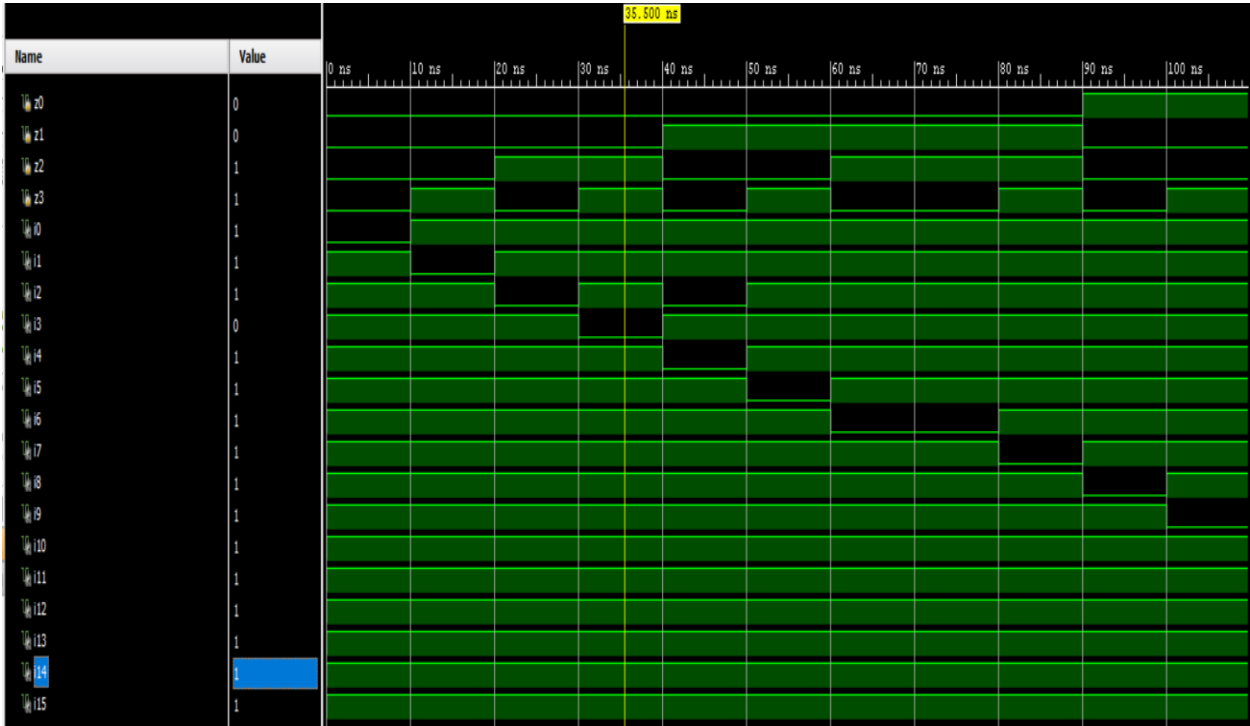
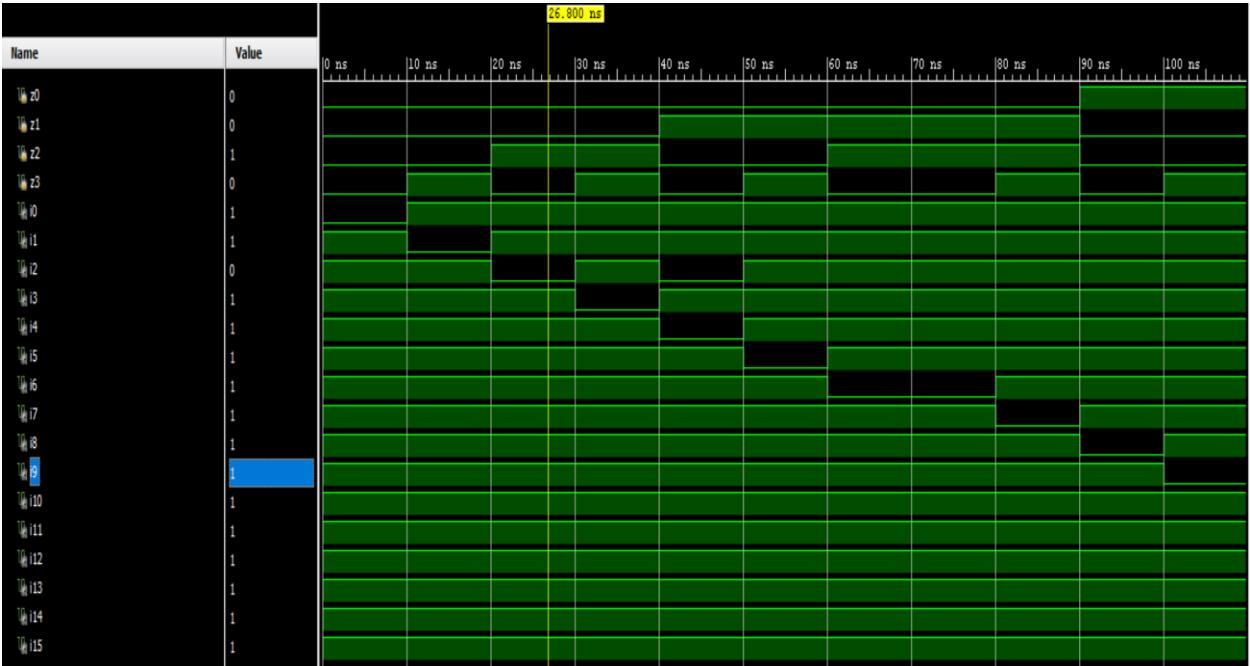
```

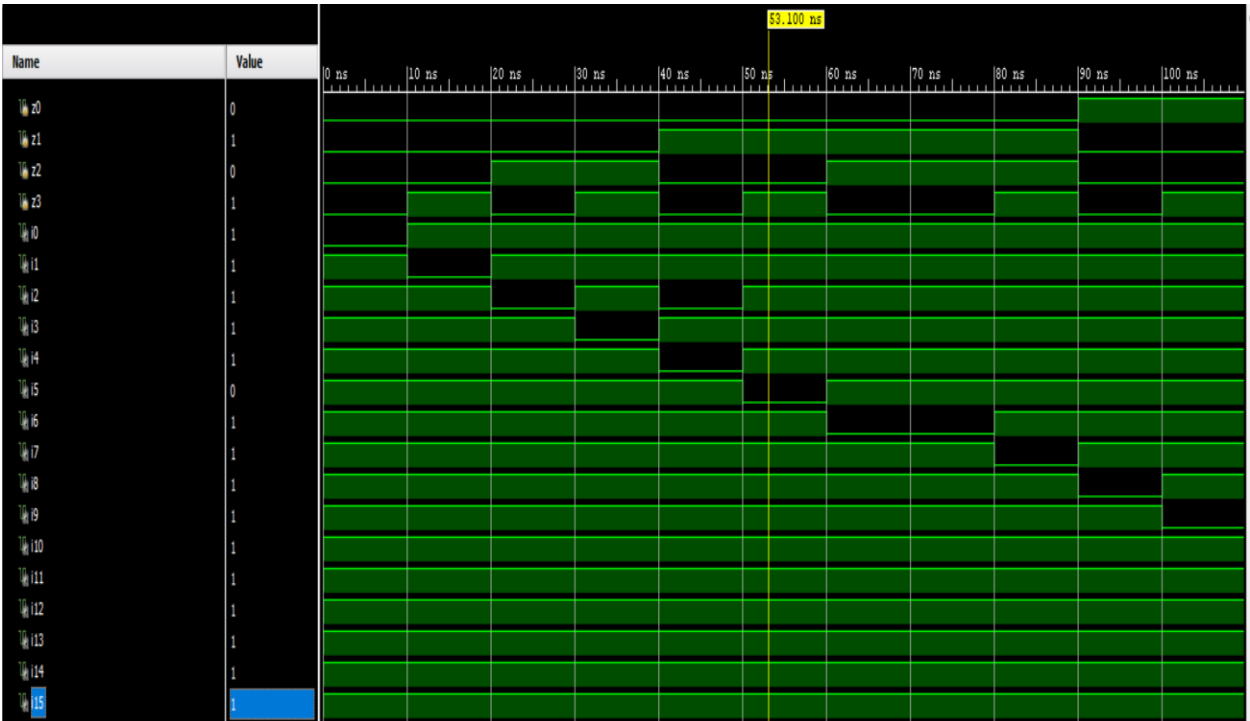


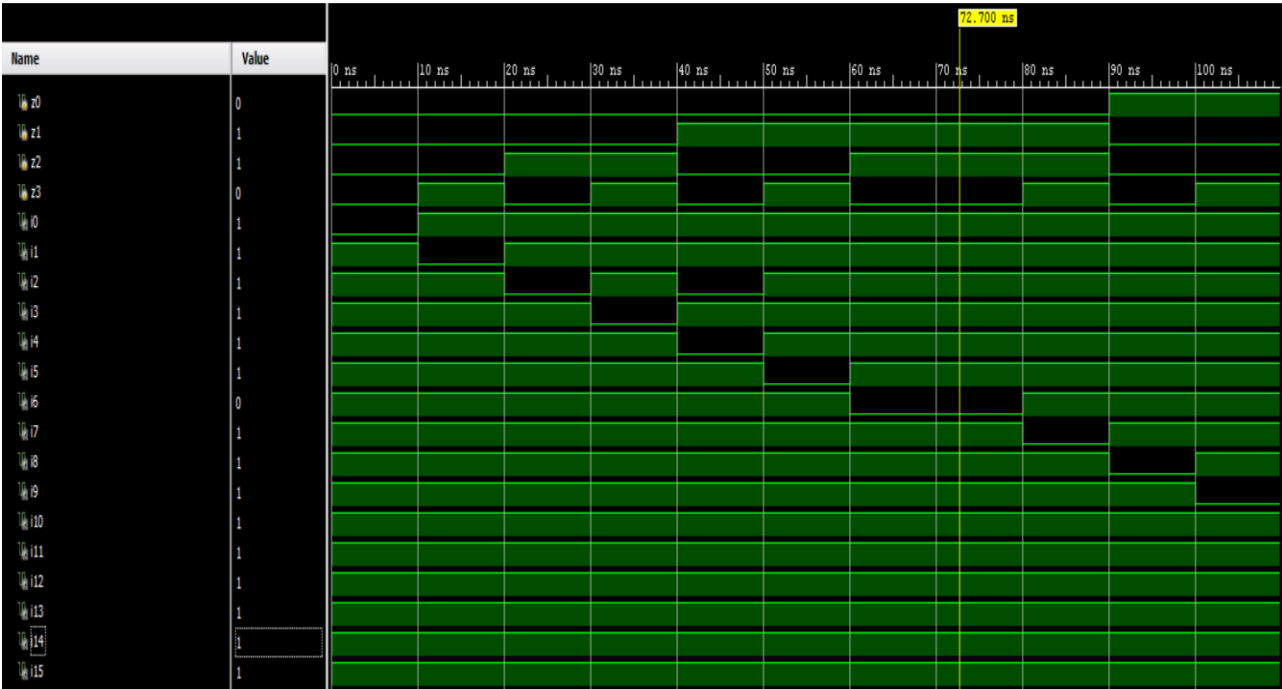
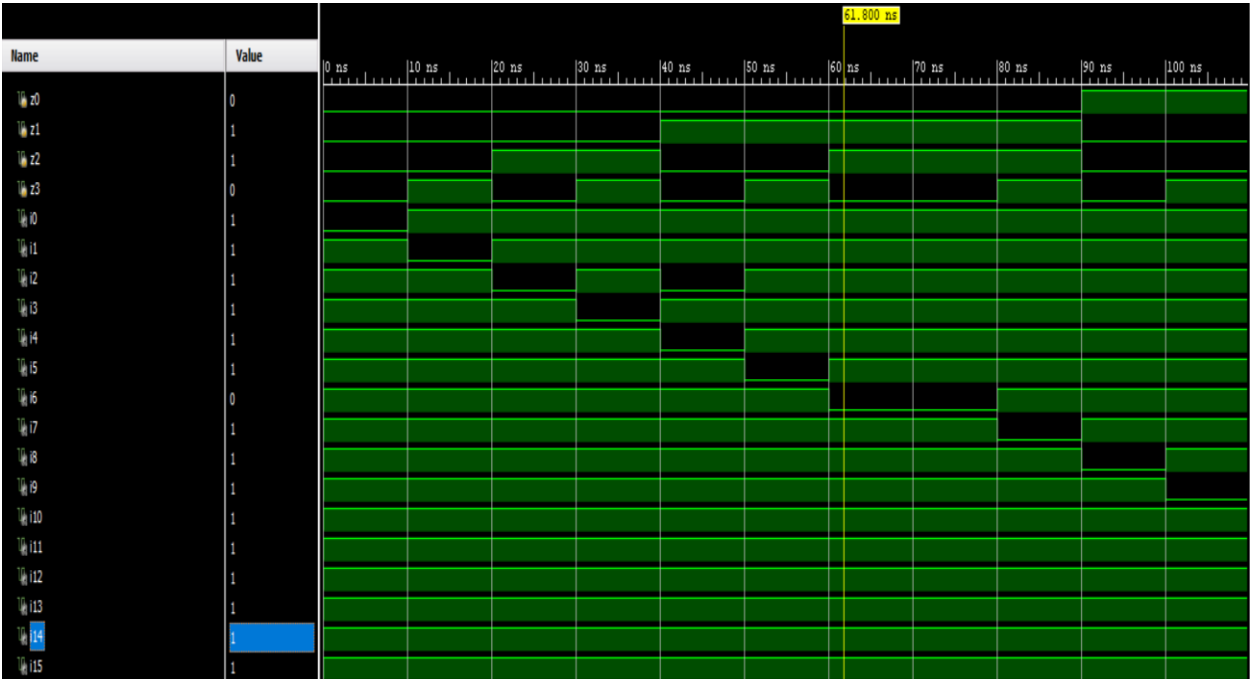
Results:

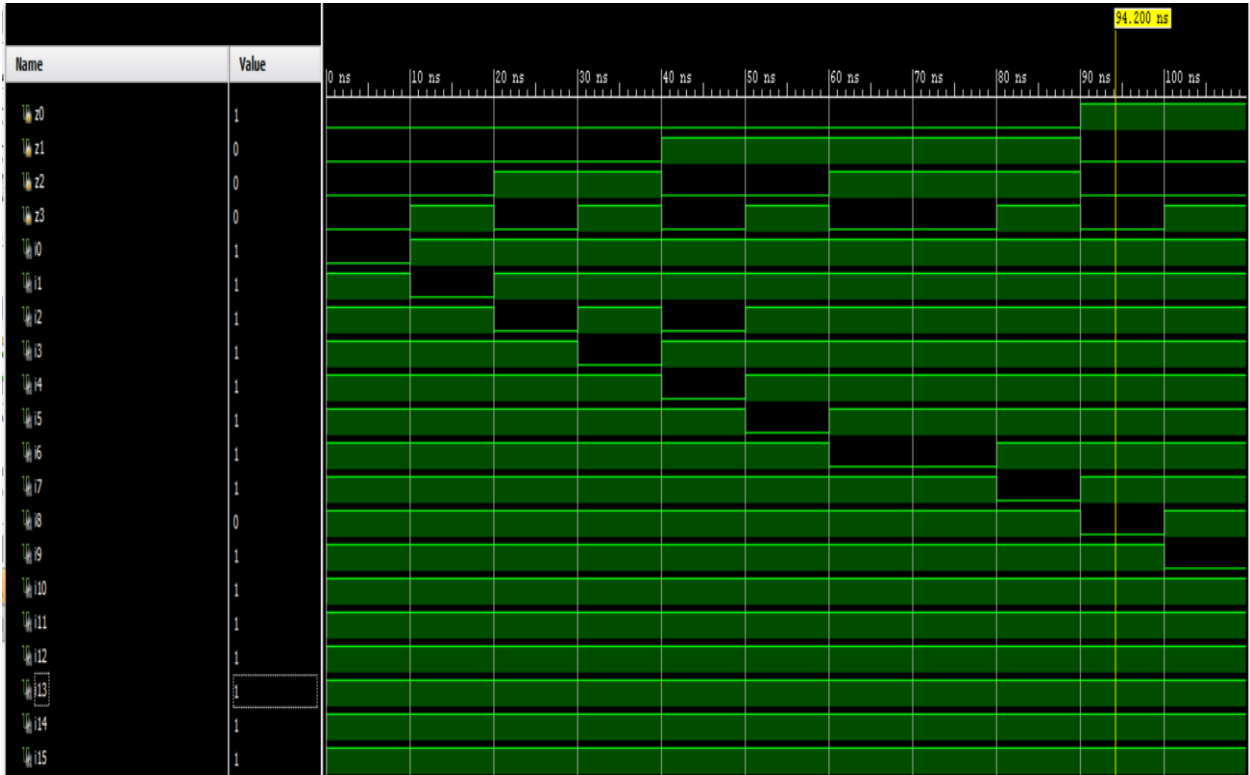
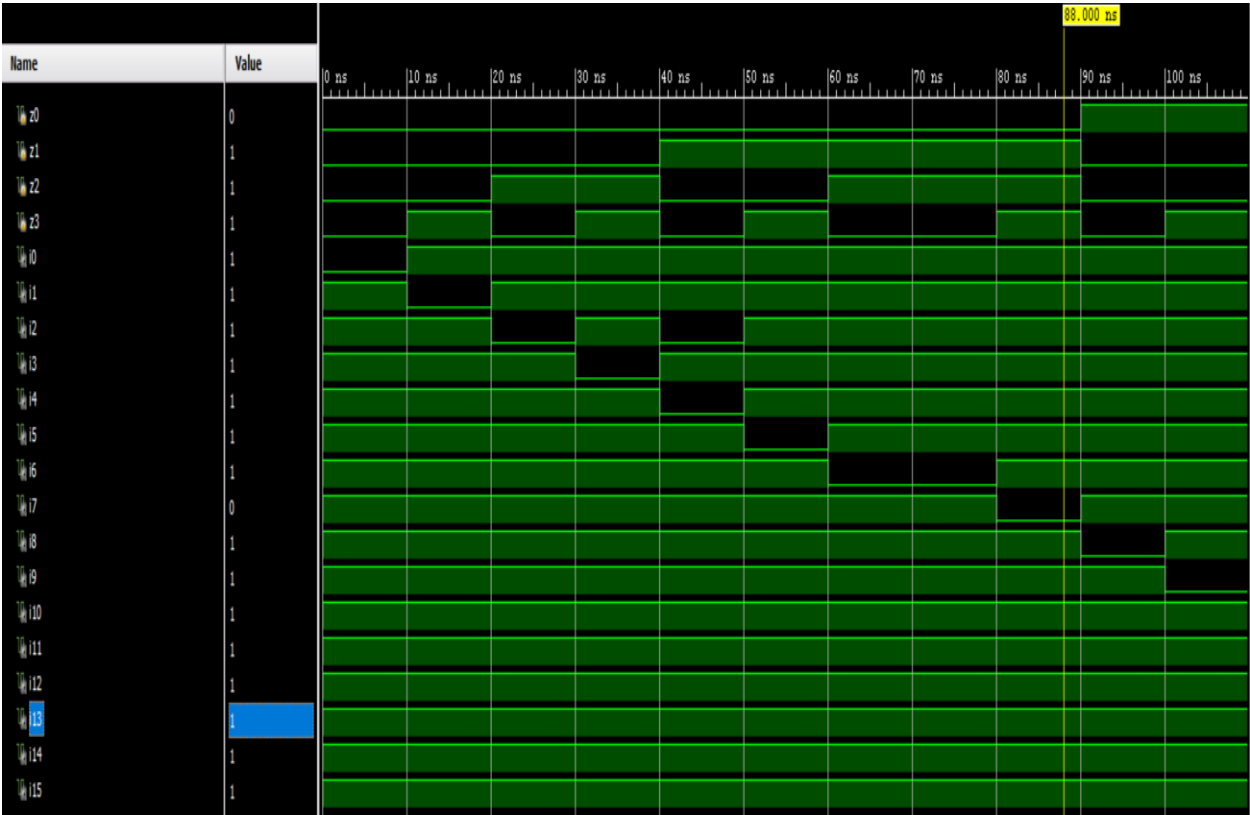
Decoder 4to16:

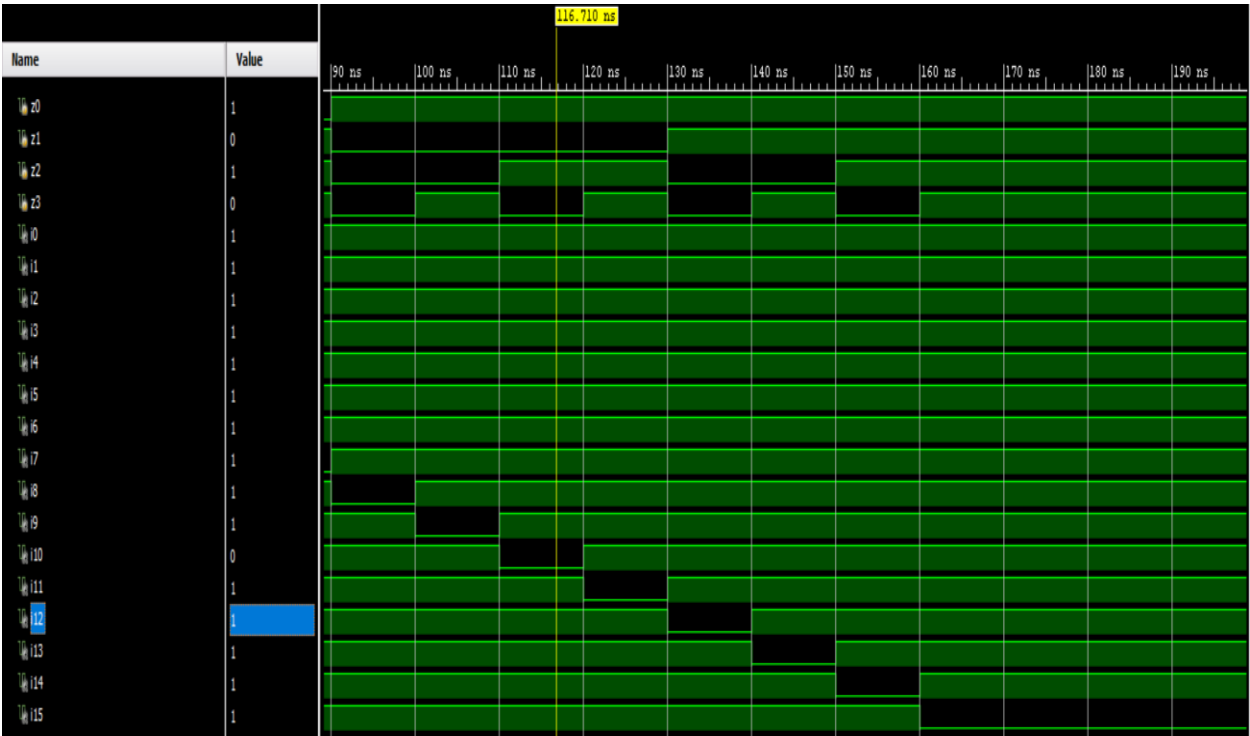
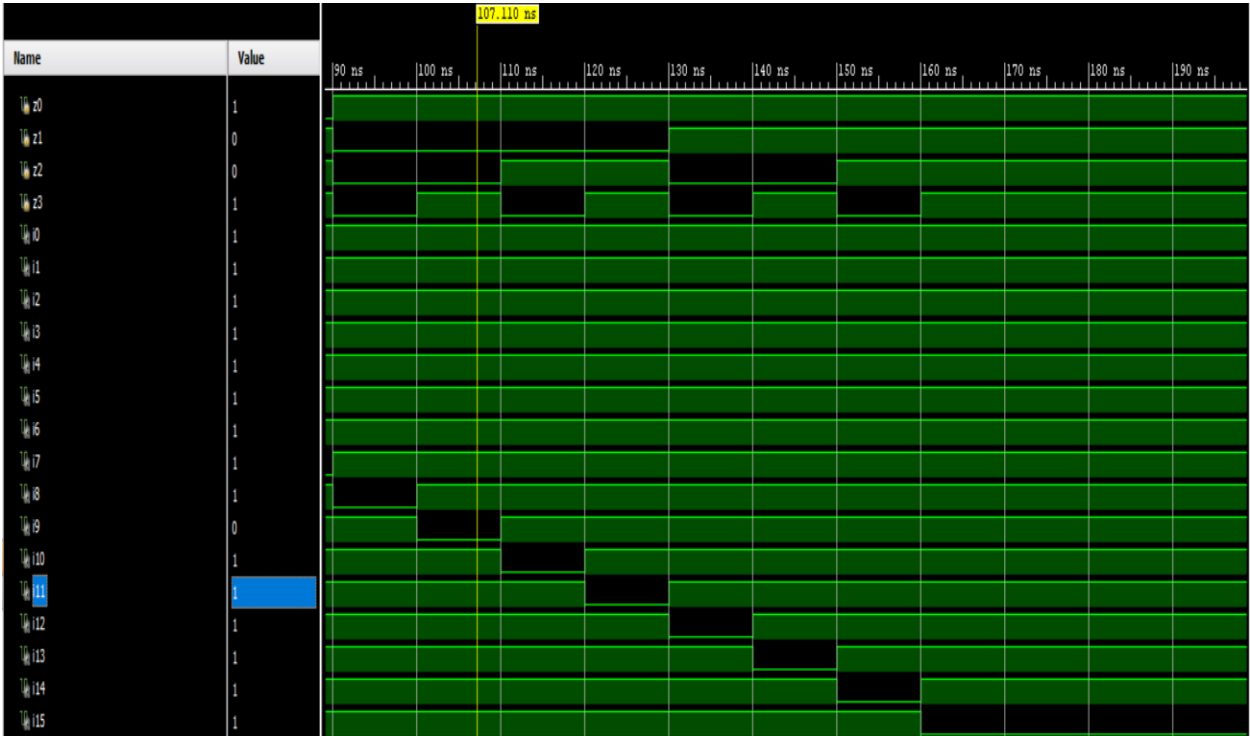


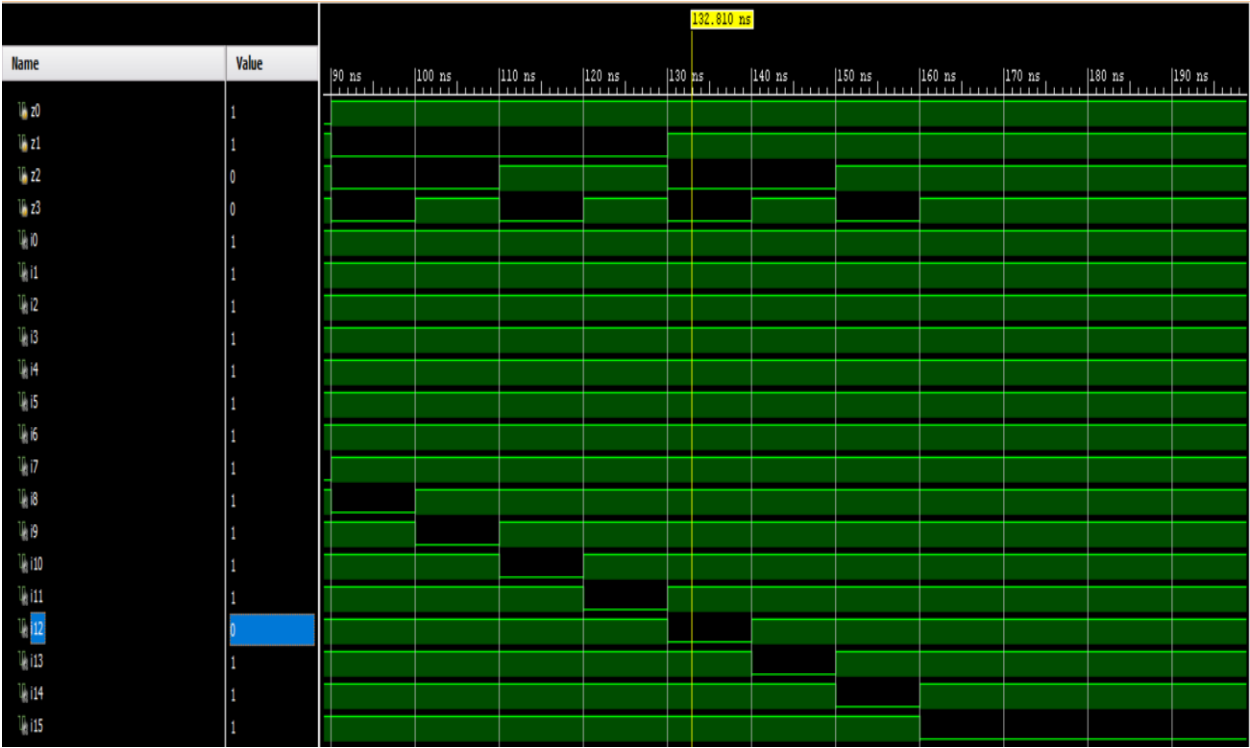
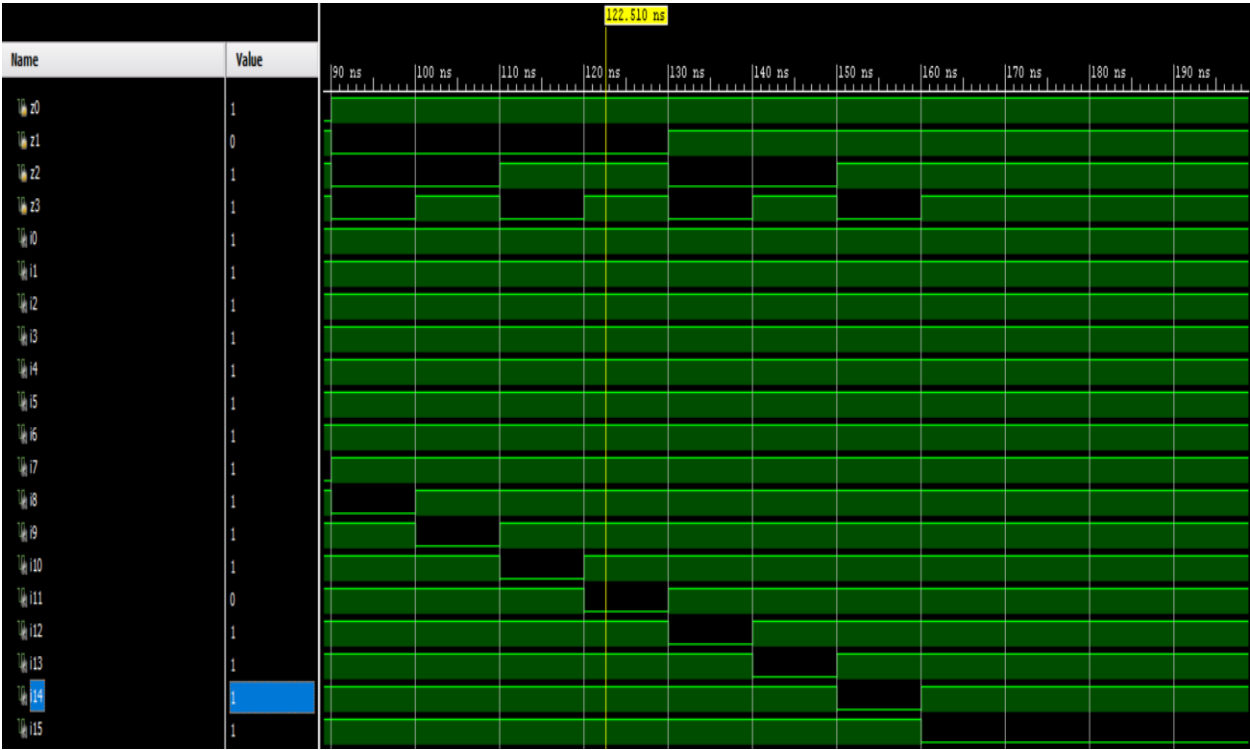


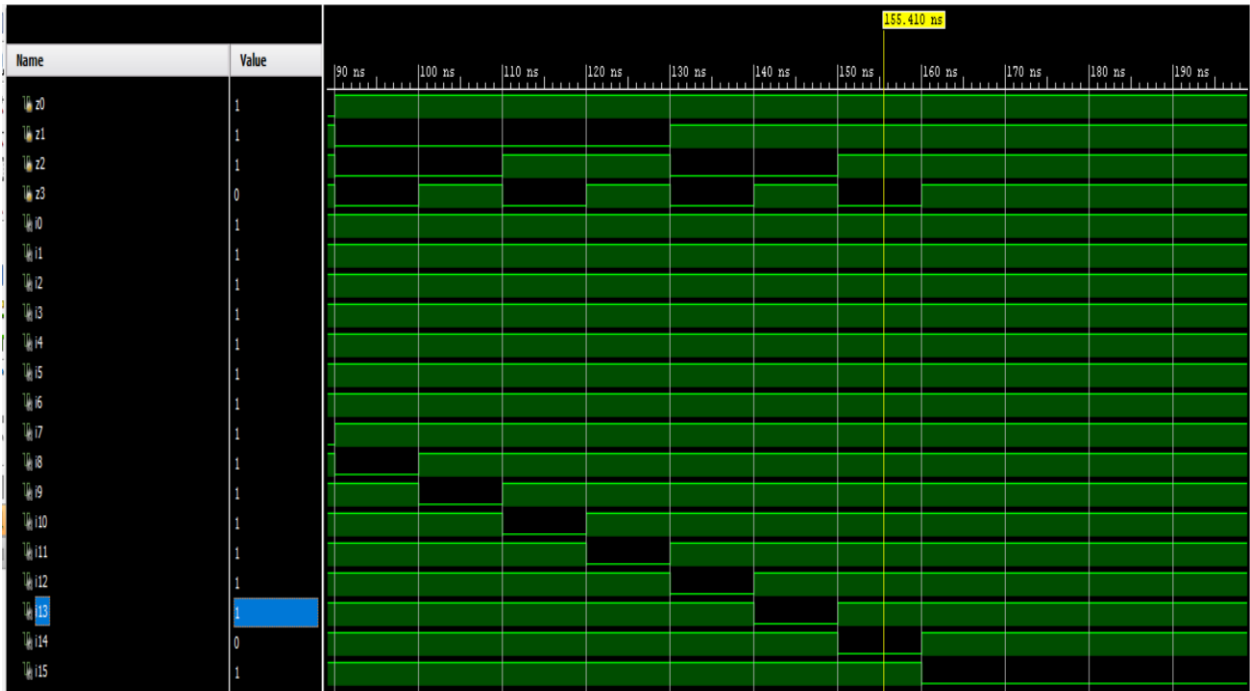
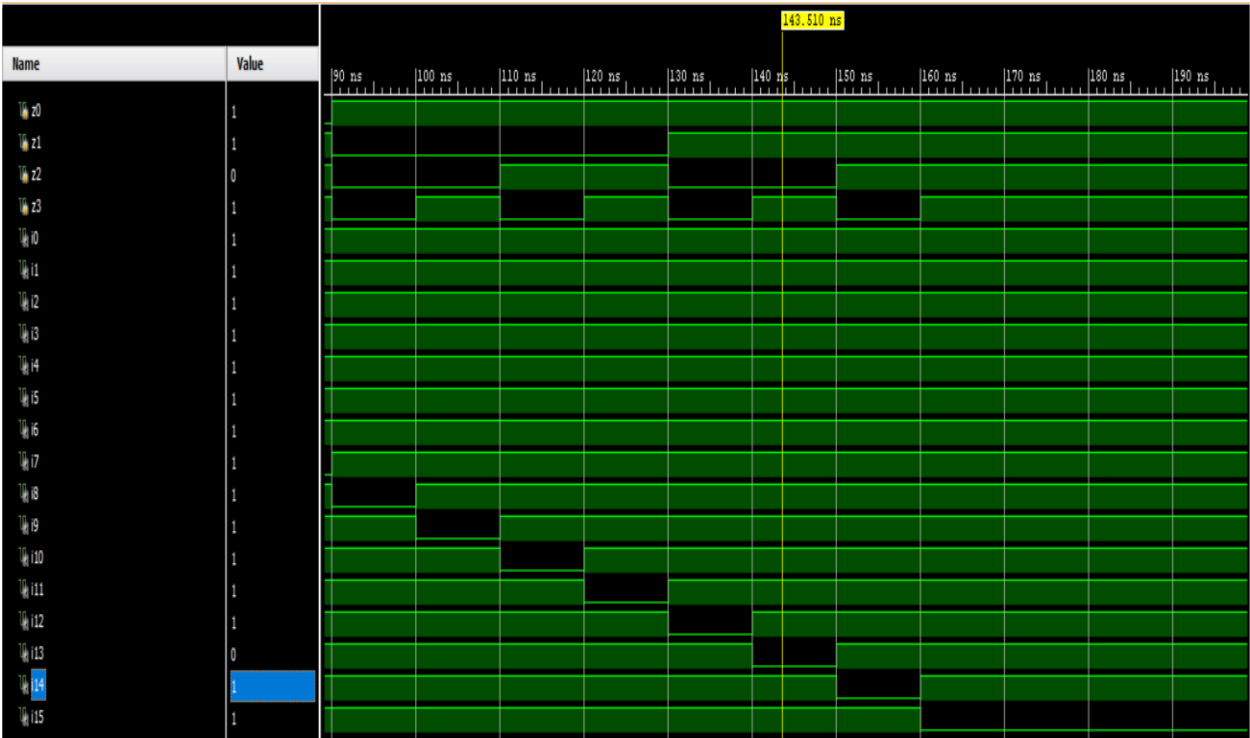




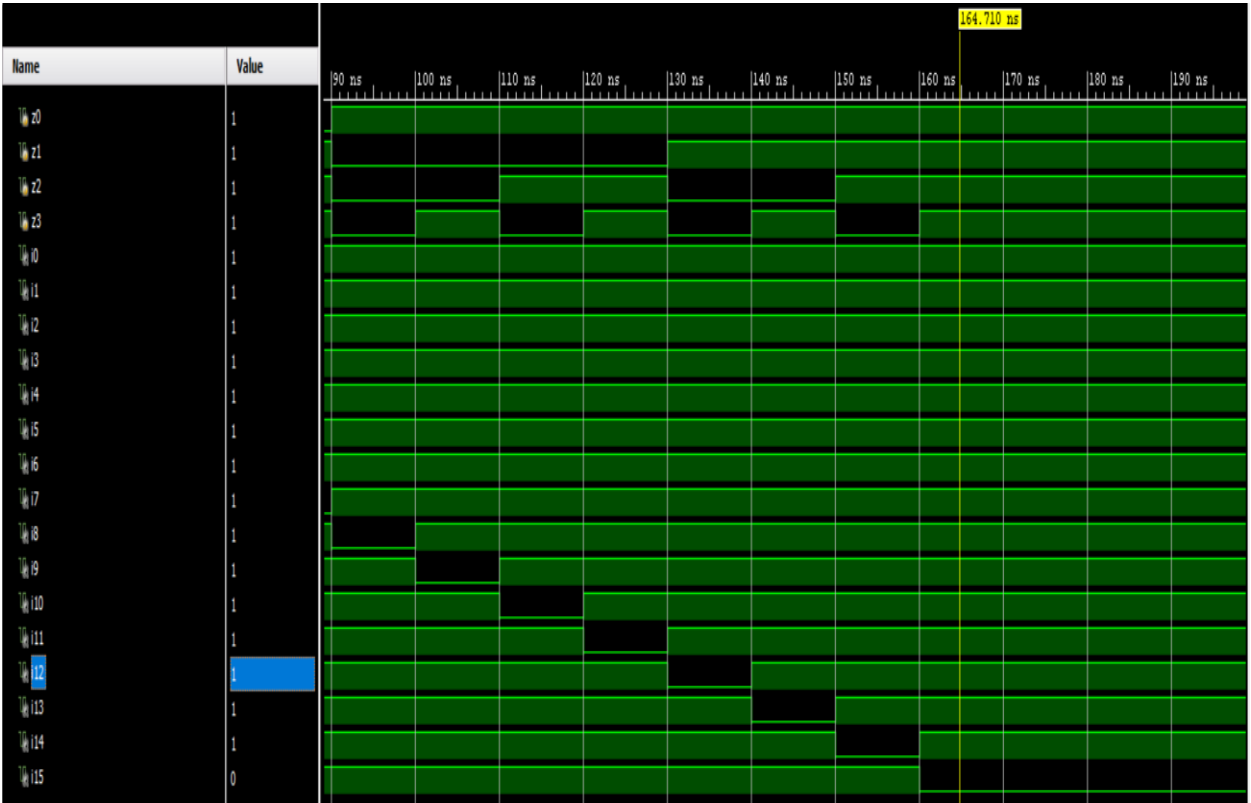




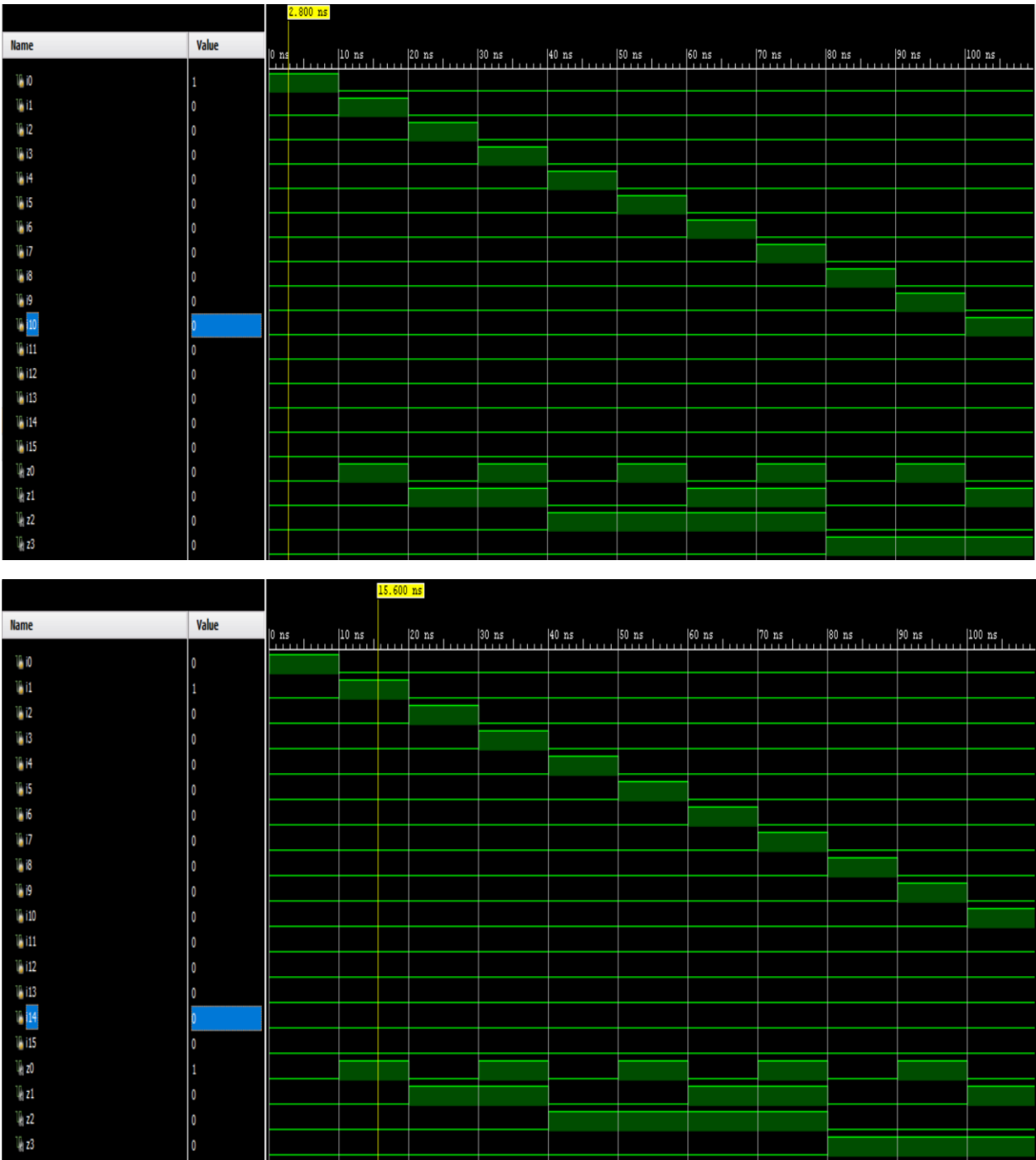


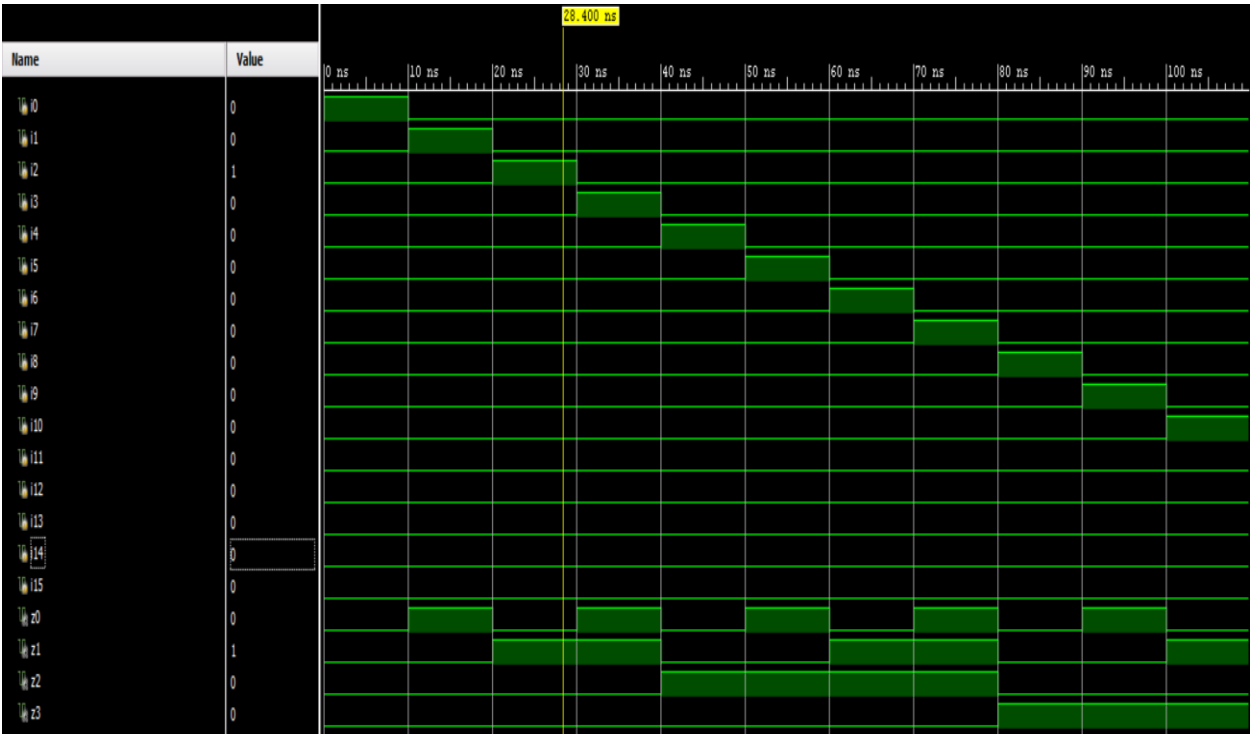






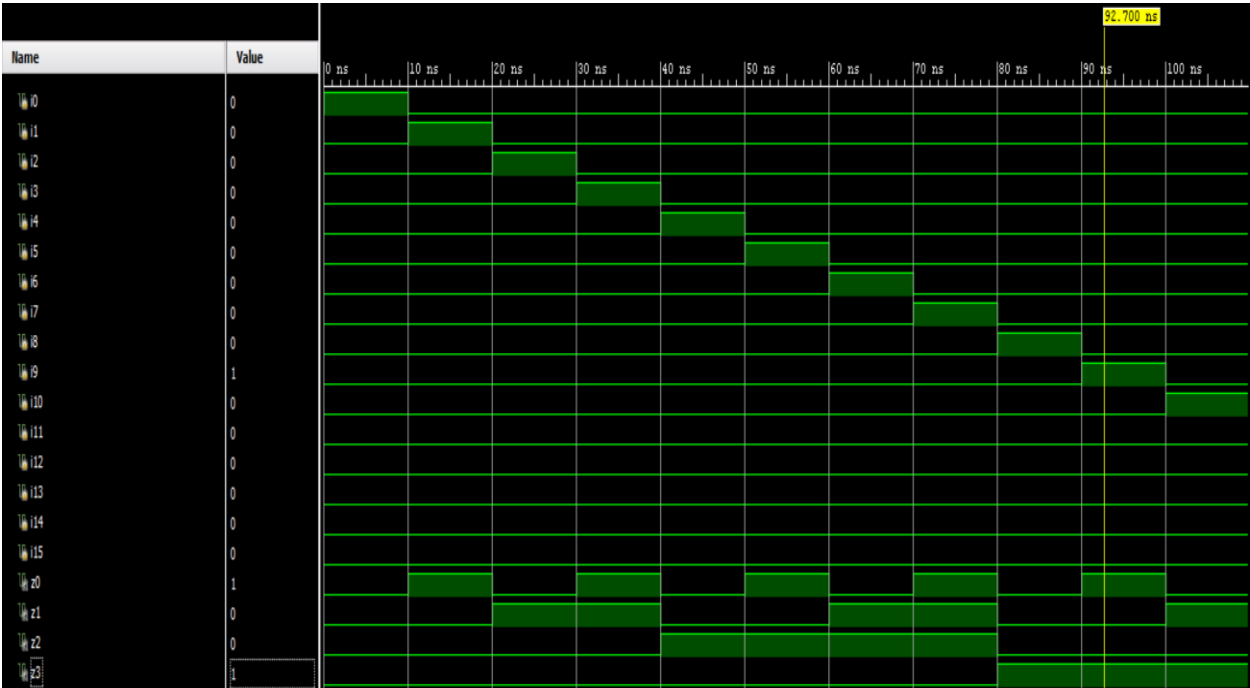
16 to 4 Encoder:

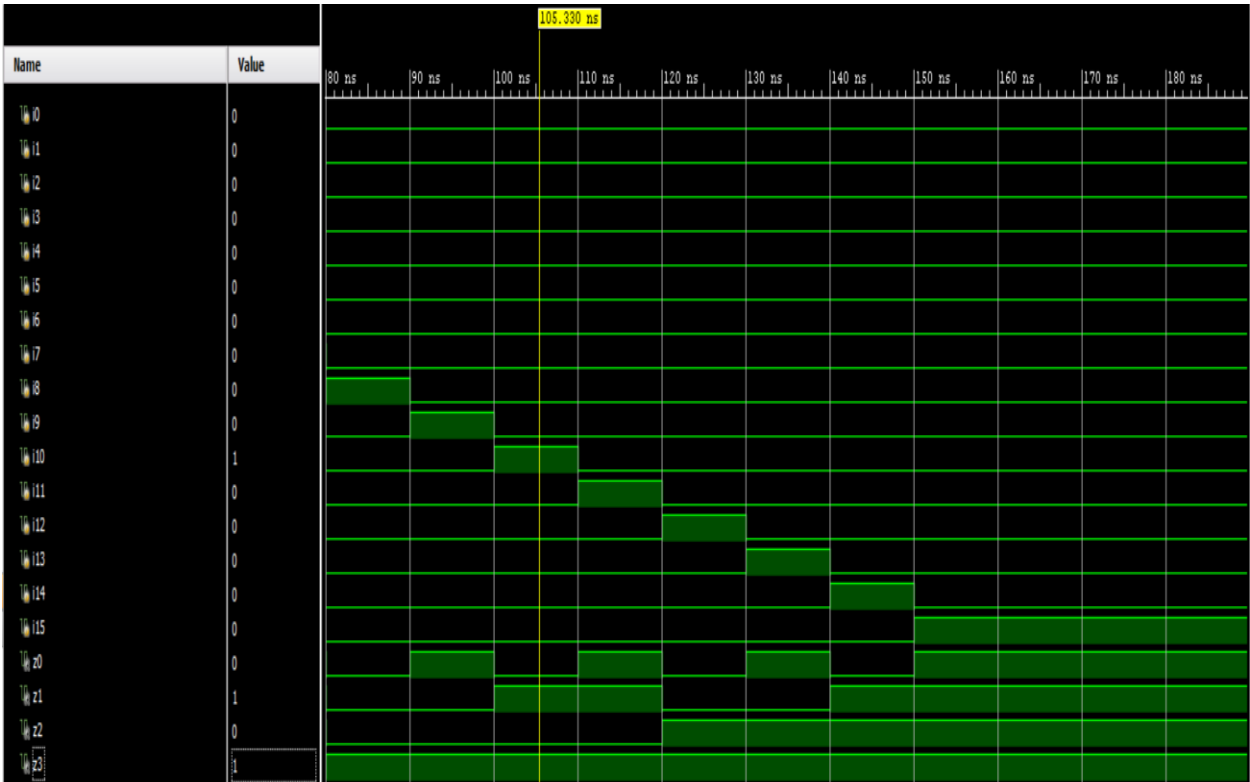


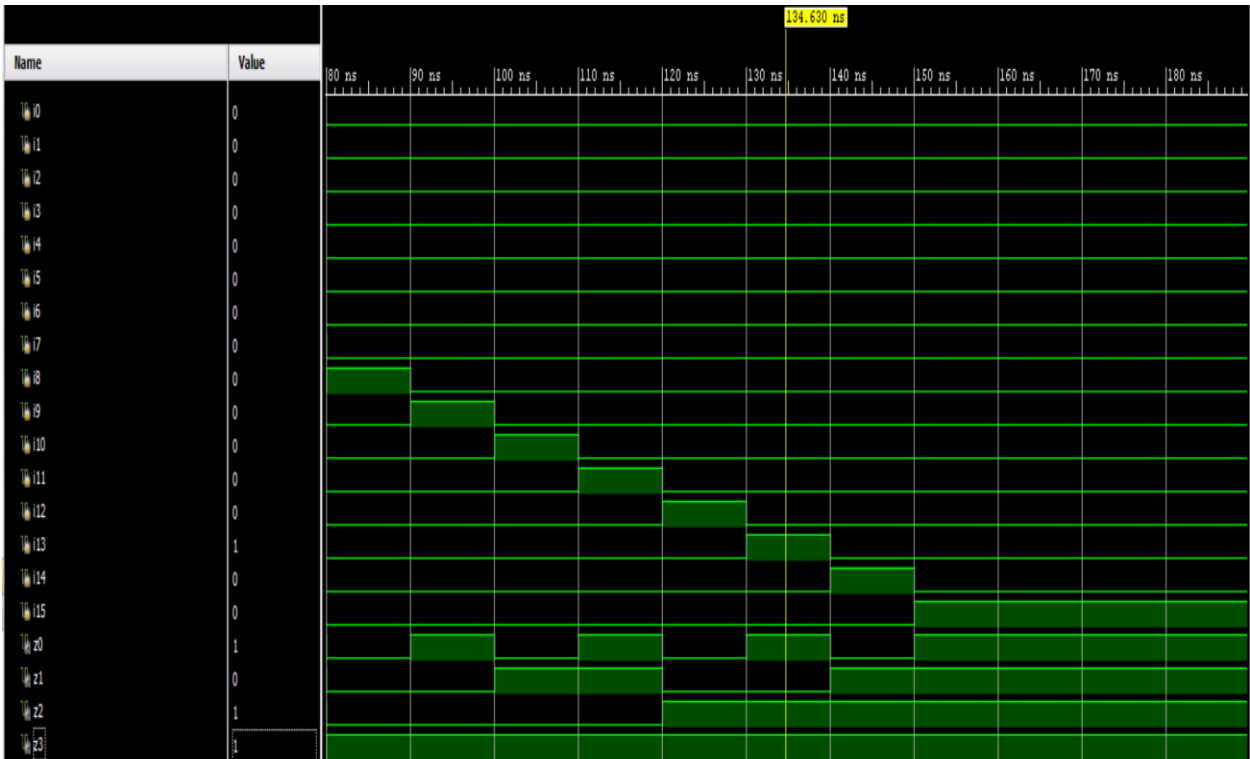
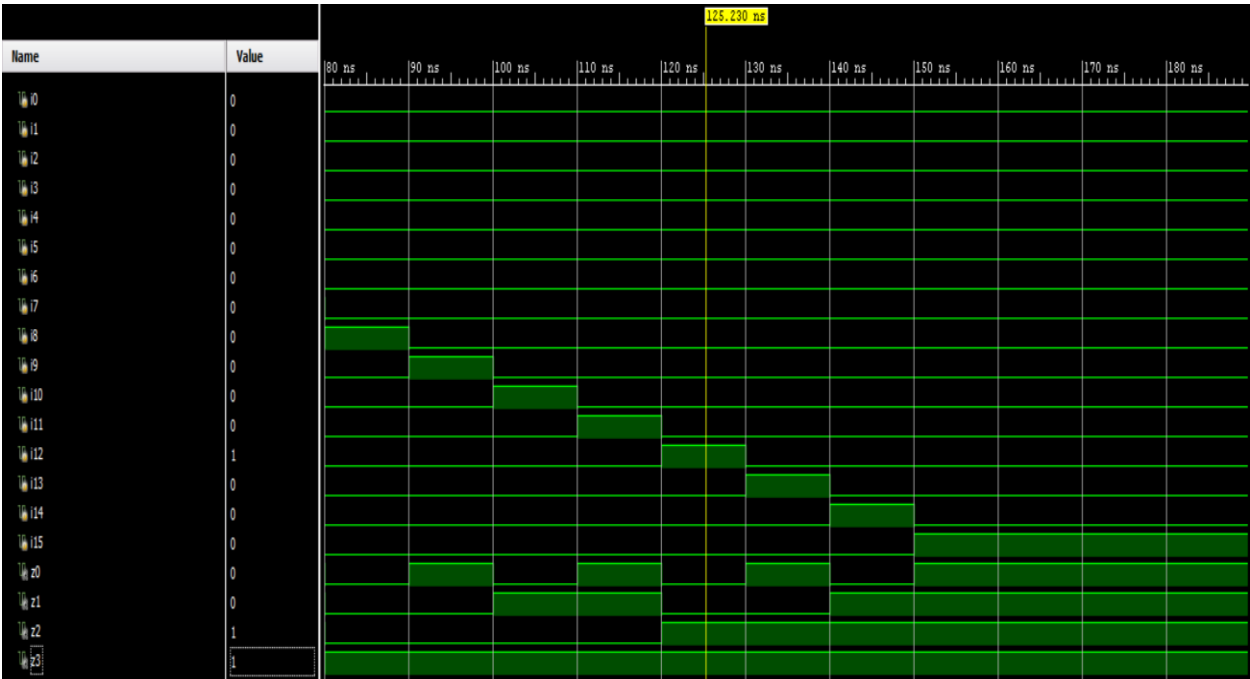




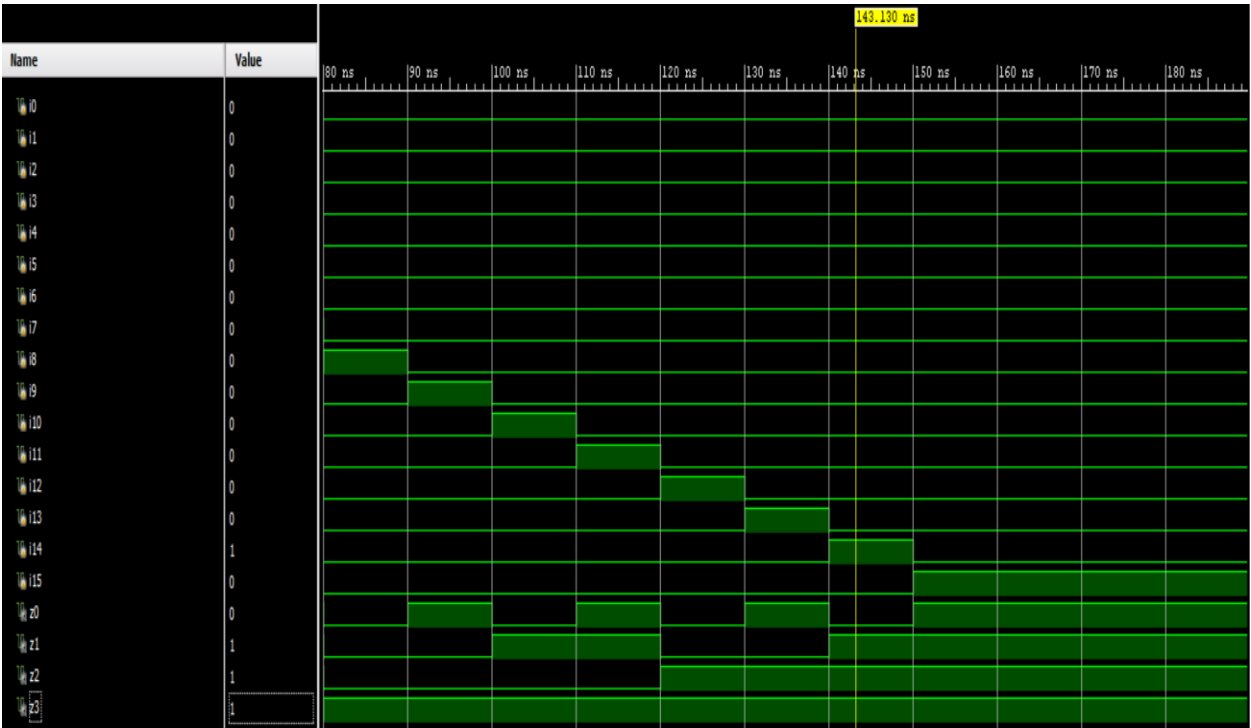












**Conclusion:**

Now I can simulate 16 to 4 Encoder and 4 to 16 Decoder codes