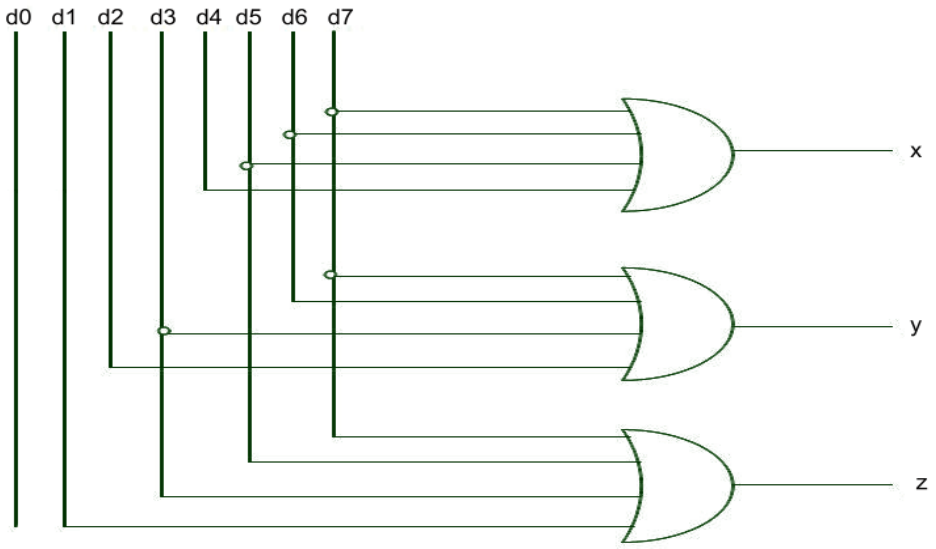
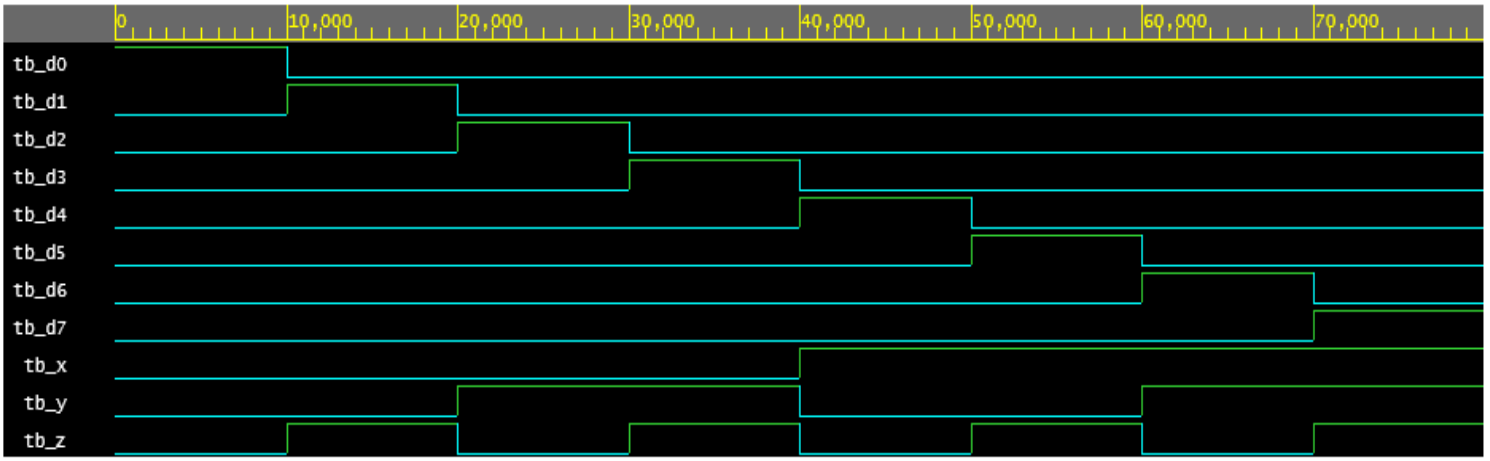
8 to 3 Encoder

d7	d6	d5	d4	d3	d2	d1	d0	x	y	z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



Based on the circuit we need 3 OR gates of four inputs where: x = d7+d6+d5+d4, y = d7+d6+d3+d2 and z = d7+d5+d3+d1. When one of the input gates is equal to 1 the outputs x,y,z (depending on the logic expressions) are forced to become equal to 1 as well. In all 3 circuits the simulation results are as follows:

Run Time: 80ns



> For instance:
10 ns: d0 = 0, **d1 = 1** , d2 = 0 , d3 = 0 , d4 = 0 , d5 = 0 , d6 = 0, d7 = 0 → **x = 0, y = 0, z = 1**
30 ns: d0 = 0, d1 = 0 , d2 = 0 , **d3 = 1** , d4 = 0 , d5 = 0 , d6 = 0, d7 = 0 → **x = 0, y = 1, z = 1**
40 ns: d0 = 0, d1 = 0 , d2 = 0 , d3 = 0 , **d4 = 1** , d5 = 0 , d6 = 0, d7 = 0 → **x = 1, y = 0, z = 0**
70 ns: d0 = 0, d1 = 0 , d2 = 0 , d3 = 0 , d4 = 0 , d5 = 0 , d6 = 0, **d7 = 1** → **x = 1, y = 1, z = 1**

*The testbench remains the same for all three models.