1) KARNAUGH MAPS

Out1

In1In0(->) In3In2 (↓)	00	01	11	10
00	x	0	0	0
01	1	0	0	0
11	1	1	1	1
10	1	1	1	1

Out2

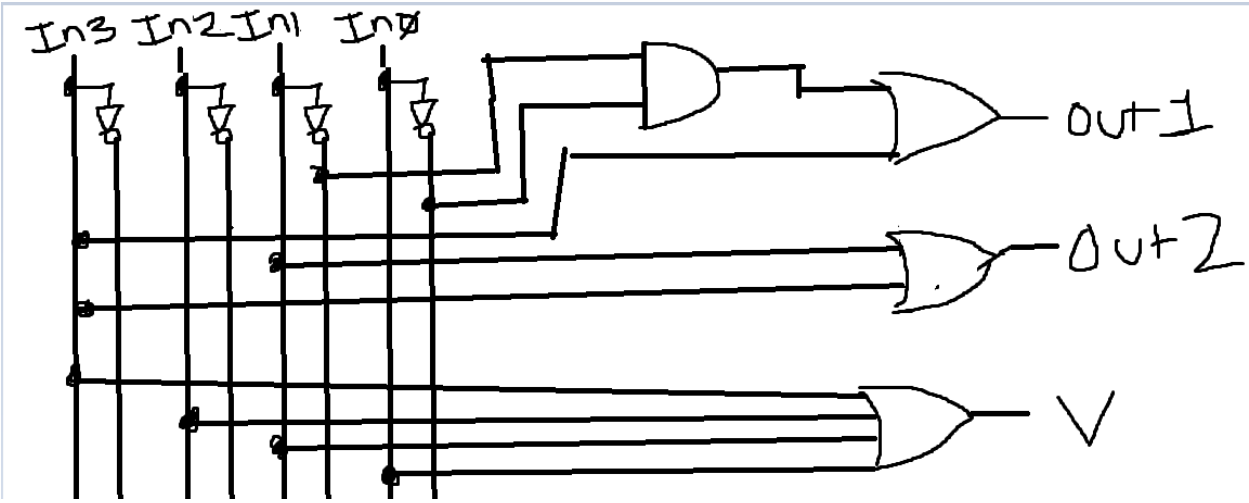
In1In0(->) In3In2 (↓)	00	01	11	10
00	x	0	1	1
01	0	0	1	1
11	1	1	1	1
10	1	1	1	1

V

In1In0(->) In3In2 (↓)	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

2)
Out1 = In1' . In0' + In3
Out2 = In1 + In3
V = In3 + In2 + In1 + In0

3)

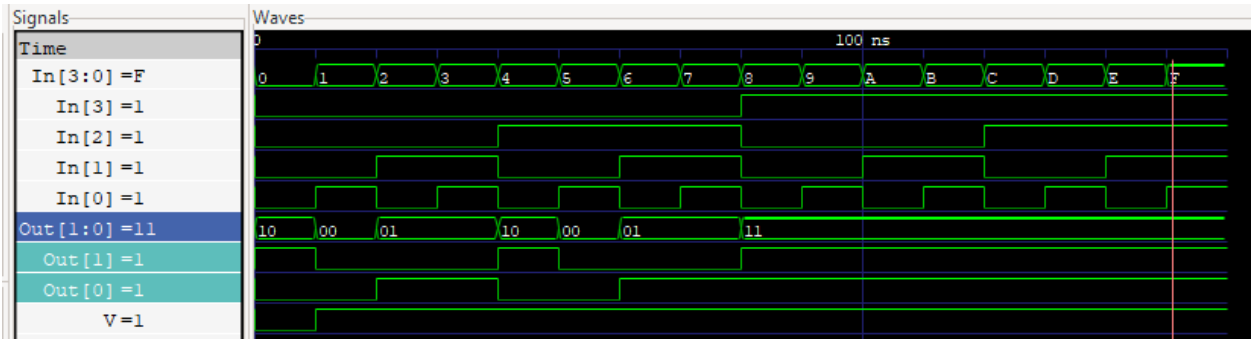


4)

```
1   module Priority_Encoder4(
2    input wire [3:0] In,
3    output wire [1:0] Out,
4    output wire V
5    );
6
7   //priority 3 -> 1 -> 0 -> 2
8   wire tempAnd;
9   and(tempAnd, !In[1], !In[0]);
10  or c1(Out[1], tempAnd, In[3]);
11  or c2(Out[0], In[3], In[1]);
12  or c3(V, In[3], In[2], In[1], In[0]);
13
14
15  endmodule
```

```
`timescale 1ns/1ns
`include "labtest.v"
module labtest_tb;
 reg [3:0] In;
 wire [1:0] Out;
 wire V;
Priority_Encoder4 uut(.In(In), .Out(Out), .V(V));
initial begin
 $dumpfile("labtest_tb.vcd");
 $dumpvars(0, labtest_tb);
 In = 4'b0000;
 #10;
 In = 4'b0001;
 #10;
  In = 4'b0010;
 #10;
  In = 4'b0011;
 #10;
  In = 4'b0100;
 #10;
  In = 4'b0101;
 #10;
  In = 4'b0110;
 #10;
  In = 4'b0111;
 #10;
  In = 4'b1000;
 #10;
  In = 4'b1001;
 #10;
  In = 4'b1010;
 #10;
  In = 4'b1011;
 #10;
  In = 4'b1100;
 #10;
  In = 4'b1101;
 #10;
  In = 4'b1110;
 #10;
  In = 4'b1111;
 #10;
 end
endmodule
```

5)



**Note , Output for Input 4’b0000 for Out[1:0] is irrelevant, only setting V to 0 is vital.