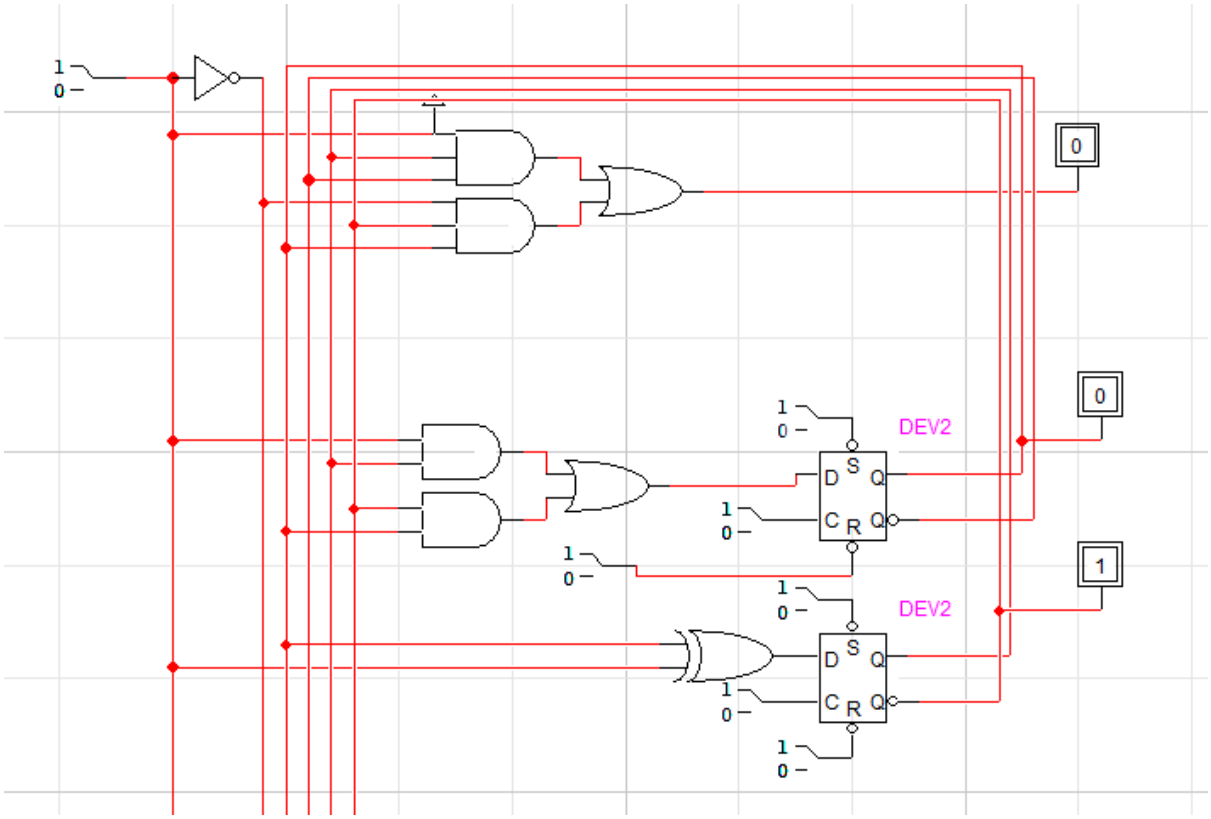
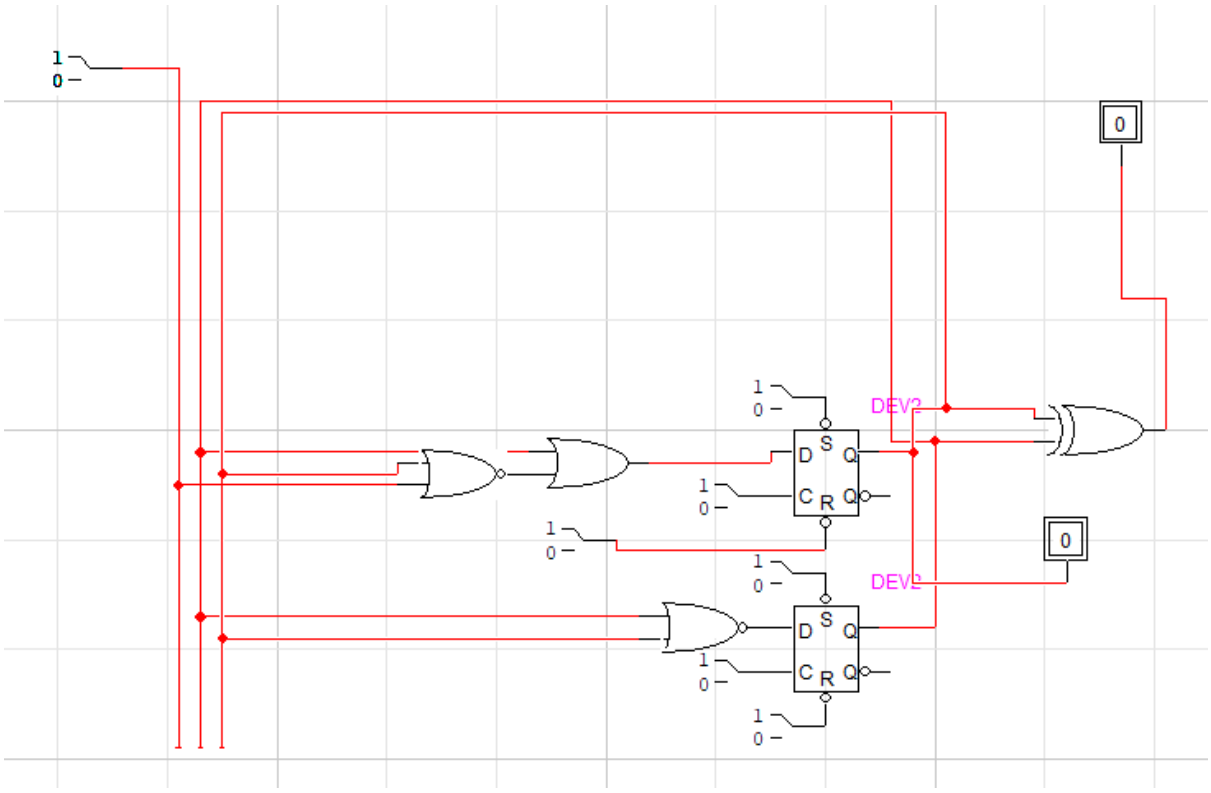
printOut 1/2



printout 2/3



```
`timescale 1ns / 1ps
module lllll(
input d0,d1,
output q0,q1,q2,q3,q4,q5,q6 );
reg[1:0]tem = 2'b00;
reg[6:0]En ;
always@(tem)begin
if(tem == 11)
tem = tem + 1'b1;
else if(tem == 10)
tem = tem - 1'b1;
else
tem = tem;
end
always@tem begin
case(tem)
4'b0000 : En <= 7'b1111110;
4'b0001 : En <= 7'b0110000;
4'b0010 : En <= 7'b1101101;
4'b0011 : En <= 7'b1111001;
4'b0100 : En <= 7'b0110011;
4'b0101 : En <= 7'b1011011;
4'b0110 : En <= 7'b1011111;
4'b0111 : En <= 7'b1110000;
4'b1000 : En <= 7'b1111111;
4'b1001 : En <= 7'b1111011;
endcase
end
assign q6 = En[6];
assign q5 = En[5];
assign q4 = En[4];
assign q3 = En[3];
assign q2 = En[2];
assign q1 = En[1];
assign q0 = En[0];
endmodule
```