Abdalrahman Ali Eltaher

Group 2

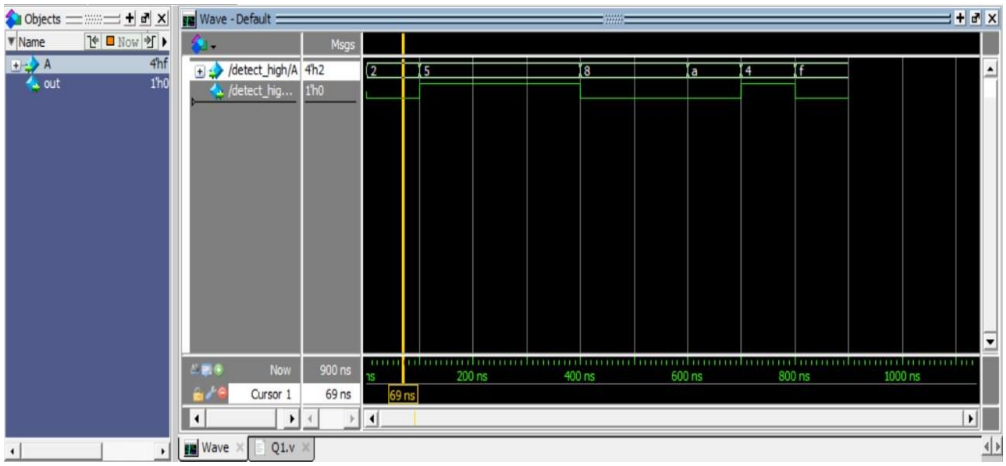
Assignment 1

•Q [1]

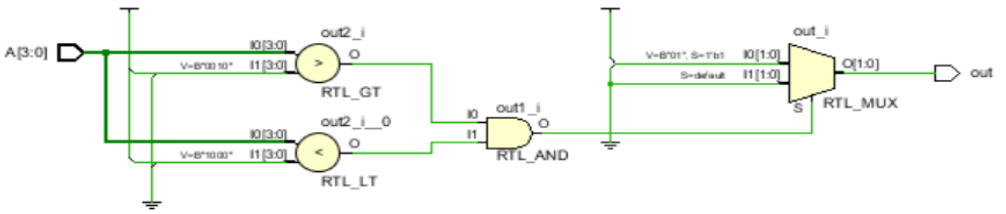
- The Code :

```
Q1.v
E: > Deploma karem > Assignment_1 > Q1.v
1  /*this program is detected if A greater than '0010'
2   and less than '1000' the out is High else is Low*/
3
4  module detect_high(A,out);
5  input [3:0] A;
6  output  out;
7  assign out = (A >4'b0010 && A<4'b1000)? 1:0;
8  endmodule
```

- The Wave :



- The Schematic :

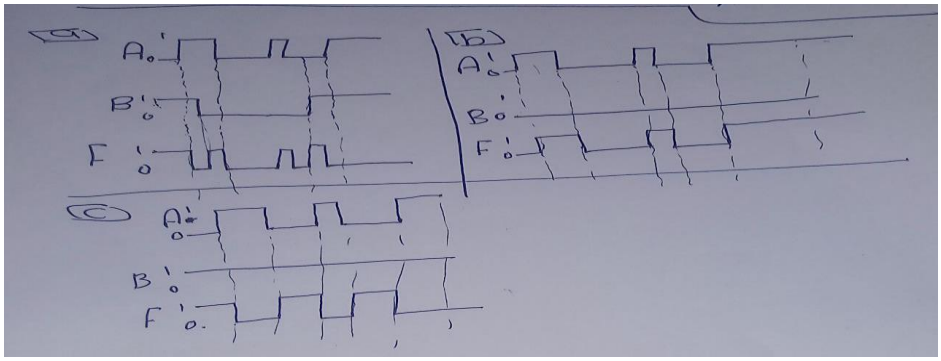


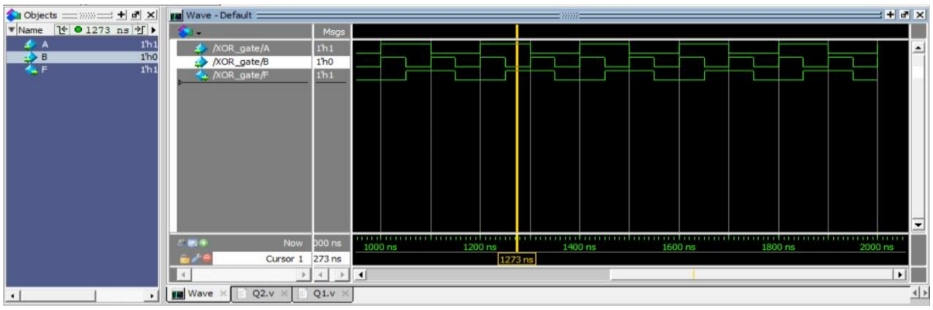
•Q[2]

- Code :

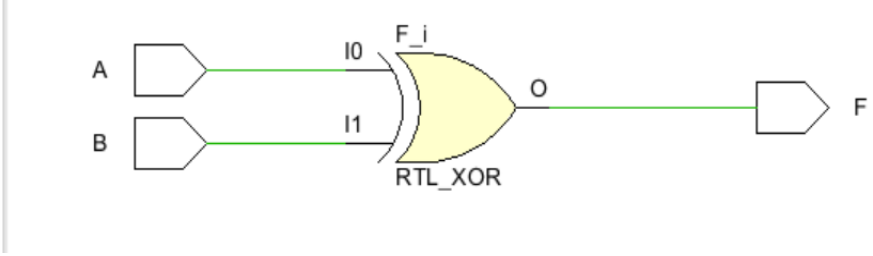
```
Q1.v    Q2.v
E: > Deploma karem > Assignment_1 > Q2.v
1  /*this program creates XOR gate When two inputs have a same value
2  the output is Low and when they have A different value
3   the output is High.  */
4
5  module XOR_gate(A,B,F);
6  input A,B;
7  output F;
8  assign F = A ^ B;
9
10 endmodule
```

- Wave:





- Schematic :



- **Q[3]**
  - Code :

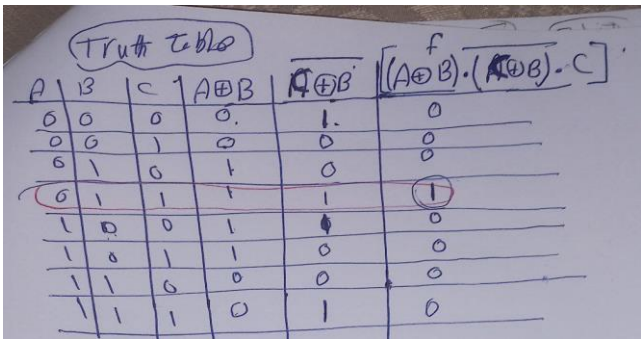
Q1.v ● Q2.v ● Q3.v ●

E: > Deploma karem > Assignment_1 > Q3.v

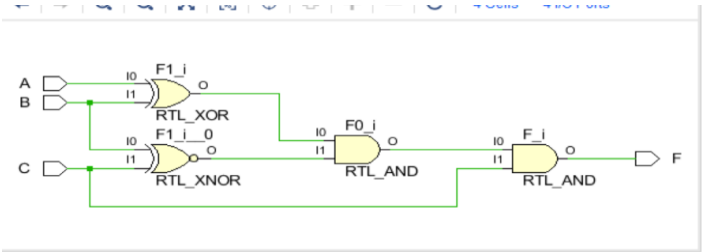
```
1  /*This program is a set of logic gates
2   to implement a specific function
3   and the output is equal to 1 when A=0 B=1, C=1*/
4   //-------------------------------------
5  module logic_gate(A,B,C,F);
6  input A,B,C;
7  output F;
8  assign F = ((A ^ B) & (B ~^ C) & (C) );
9  endmodule
```

  - Wave :

  - Truth Table :



  - Schematic :

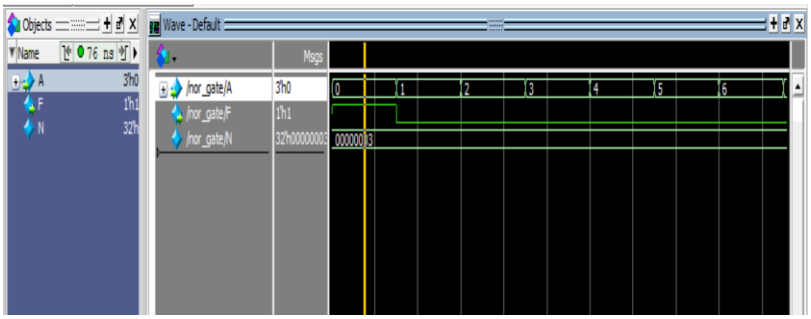


- **Q [4]**
  - Yes, NOR gate can be replaced by the comparator .

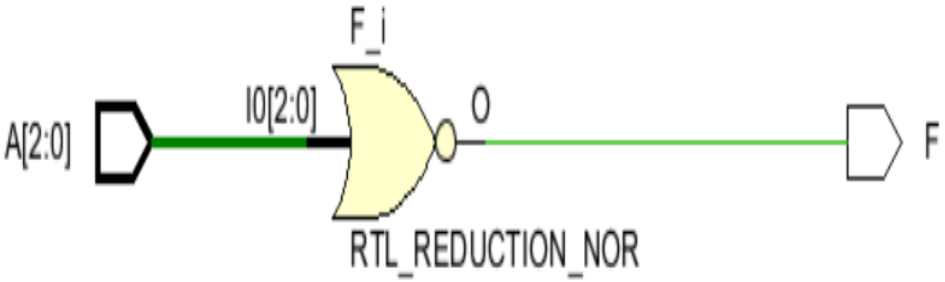
➢ Code:

```
1  // NOR gate .
2  module nor_gate(A,F);
3  parameter N = 3 ;
4  input [N-1:0] A;
5  output F ;
6  assign F = ~| A ;
7
8  endmodule
```

➢ Wave:



➢ Schematic:

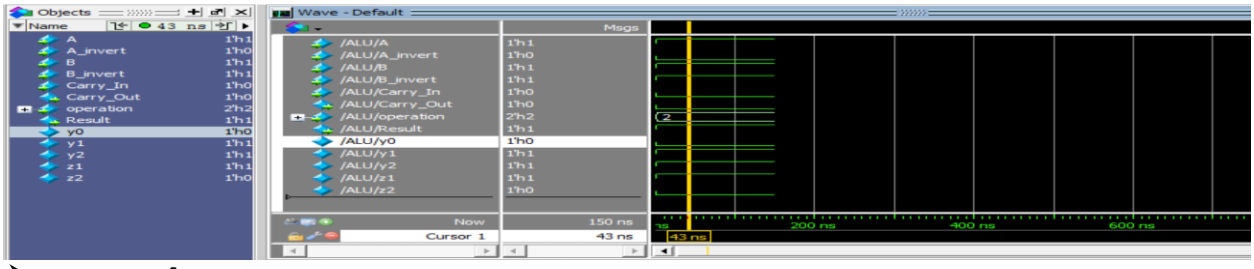


- **Q[5]**
  - Code:

```
Deploma karem > Assignment_1 > Q5.v
 1  module ALU(A,B,A_invert,B_invert,Carry_In,operation,Carry_Out,Result);
 2  input A,B,A_invert,B_invert,Carry_In;
 3  input [1:0]operation;
 4  output Carry_Out,Result ;
 5  wire z1,z2,y0,y1,y2;
 6  assign z1 = (A_invert == 1)? ~A:A;
 7  assign z2 = (B_invert == 1)? ~B:B;
 8  assign y0 = z1 & z2;
 9  assign y1 = z1 | z2 ;
10  assign {Carry_Out,y2} = Carry_In+z1+z2;
11  assign Result = (operation == 0)? y0 :(operation == 1)? y1 :y2 ;
12
13  endmodule
```

  - Wave:

    ∵ A =1 , Ainvert=0 ∴Z1=1 and ∵B=1 , Binvert=1 ∴z2=0 .
    ∴ y0=0 , y1 =1 and ∵ Carryin =0 ∴y2=1and Carryout=0.
    ∵ operation = 2'h2 ∴Result = y2=1.



  - Schematic :