Model Sim

LAB # 13

Fall 2020

CSE-304L Computer Organization and Architecture Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: ______________

Submitted to:

Engr. Amaad Khalil

March 19, 2021

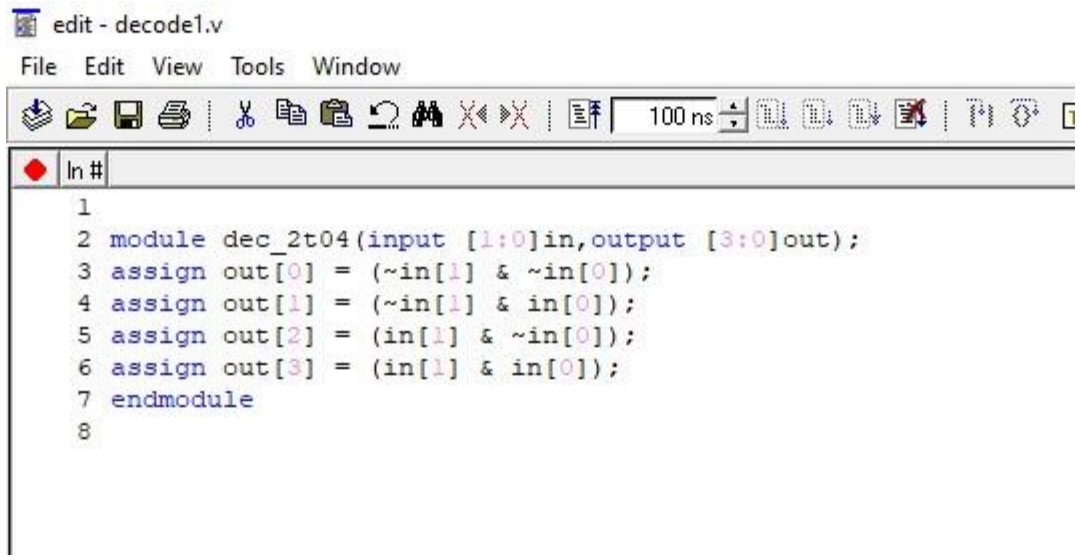
Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

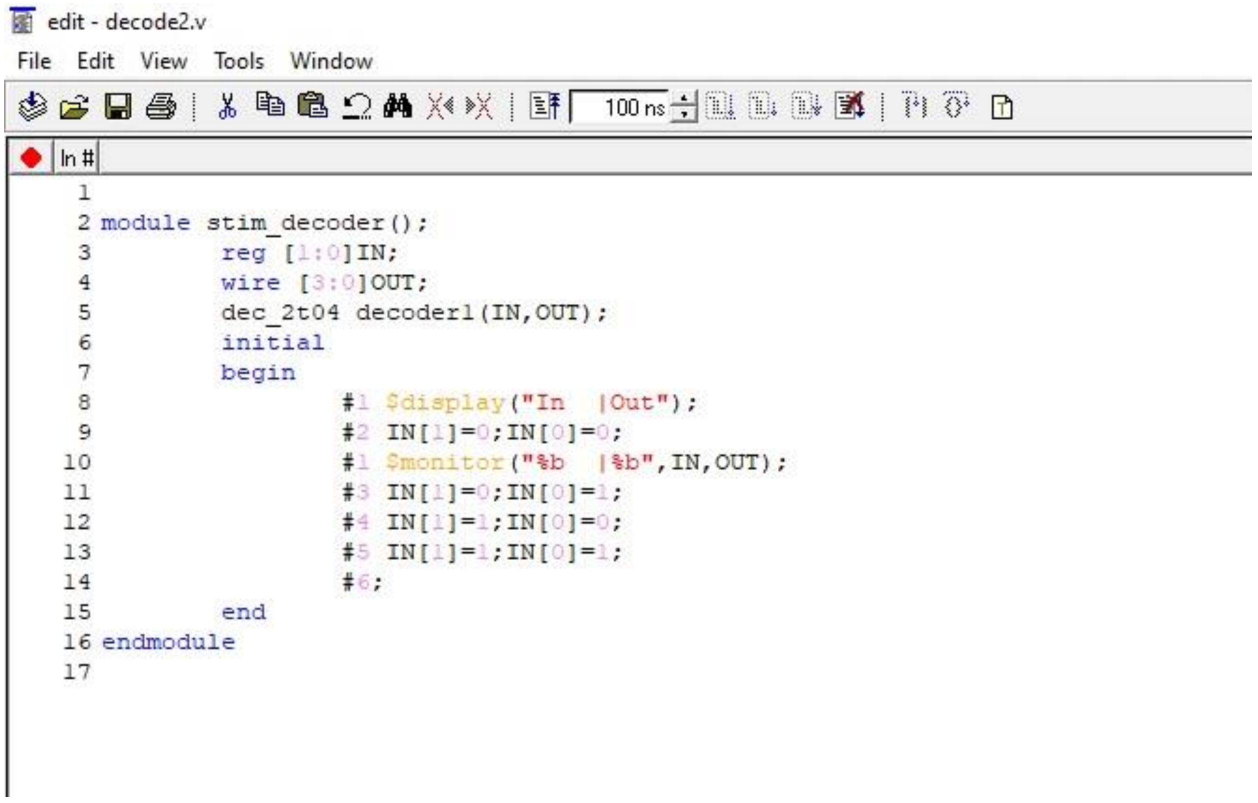
TASK01:

Write a Verilog code for 2x4 Decoder using Dataflow Level modeling.

Code:

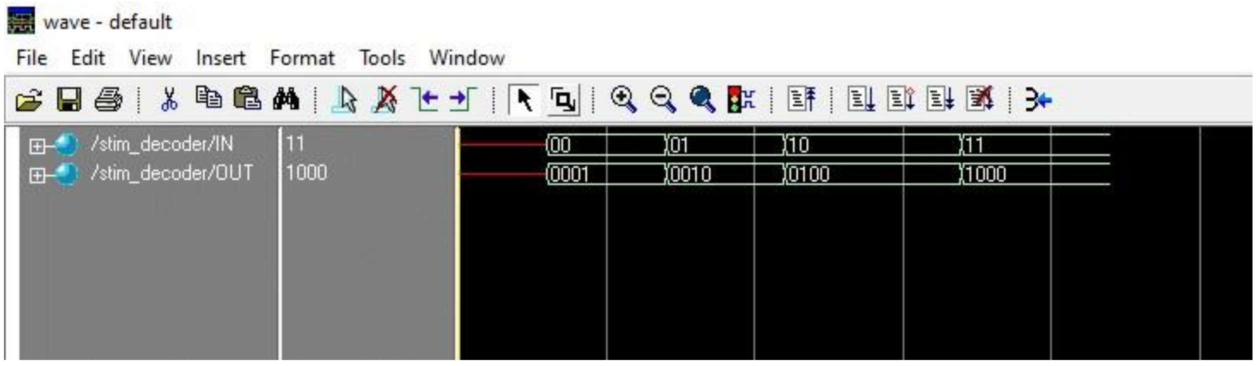


```
module dec_2t04(input [1:0]in,output [3:0]out);
assign out[0] = (~in[1] & ~in[0]);
assign out[1] = (~in[1] & in[0]);
assign out[2] = (in[1] & ~in[0]);
assign out[3] = (in[1] & in[0]);
endmodule
```



```
module stim_decoder();
        reg [1:0]IN;
        wire [3:0]OUT;
        dec_2t04 decoder1(IN,OUT);
        initial
        begin
                #1 $display("In  |Out");
                #2 IN[1]=0;IN[0]=0;
                #1 $monitor("%b  |%b",IN,OUT);
                #3 IN[1]=0;IN[0]=1;
                #4 IN[1]=1;IN[0]=0;
                #5 IN[1]=1;IN[0]=1;
                #6;
        end
endmodule
```

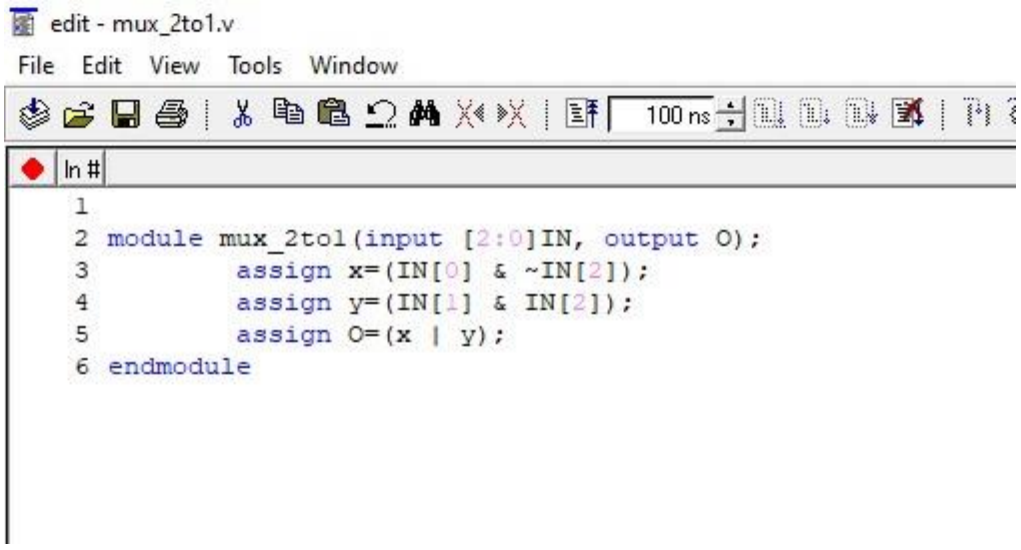
Output Wave:



TASK02:

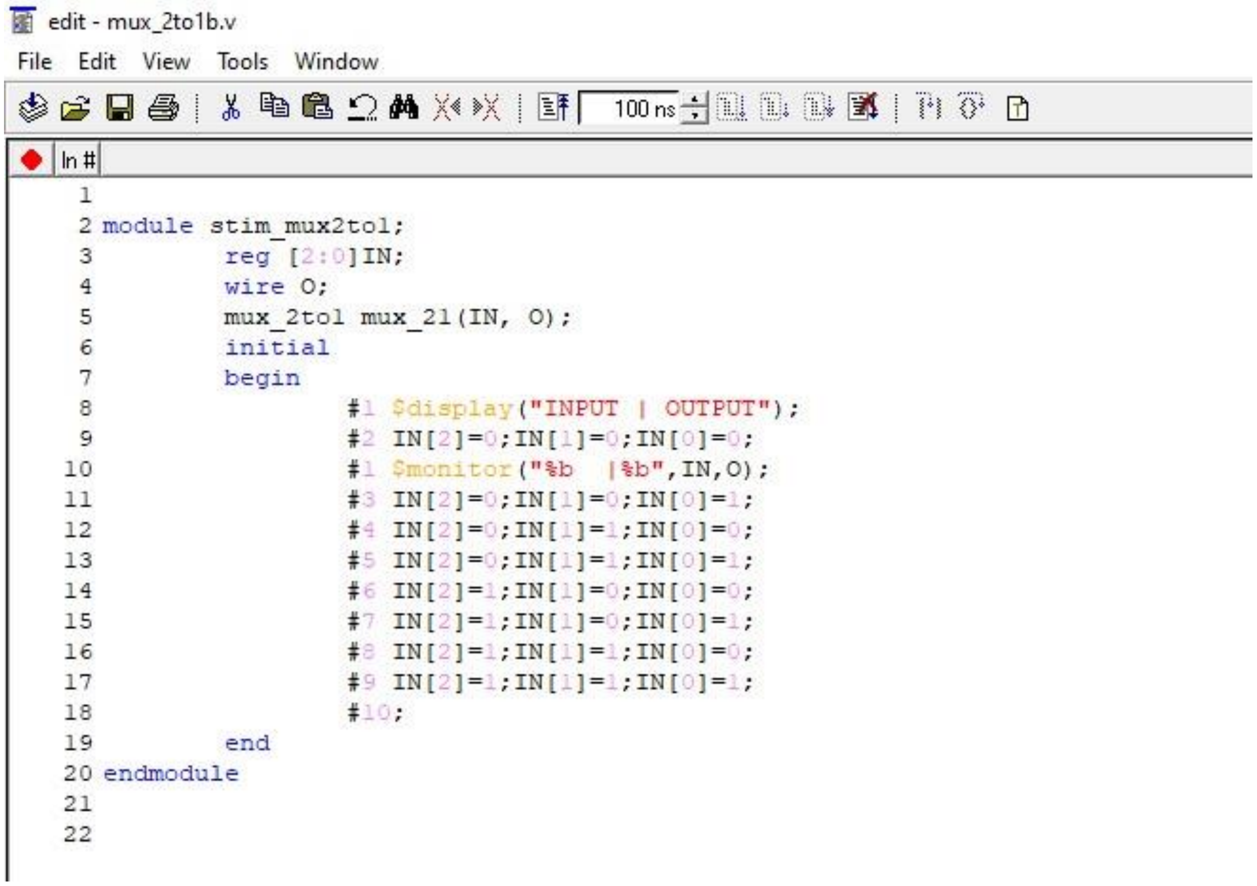
Write a Verilog code for 2x1 MUX using Dataflow Level modeling.

Code:



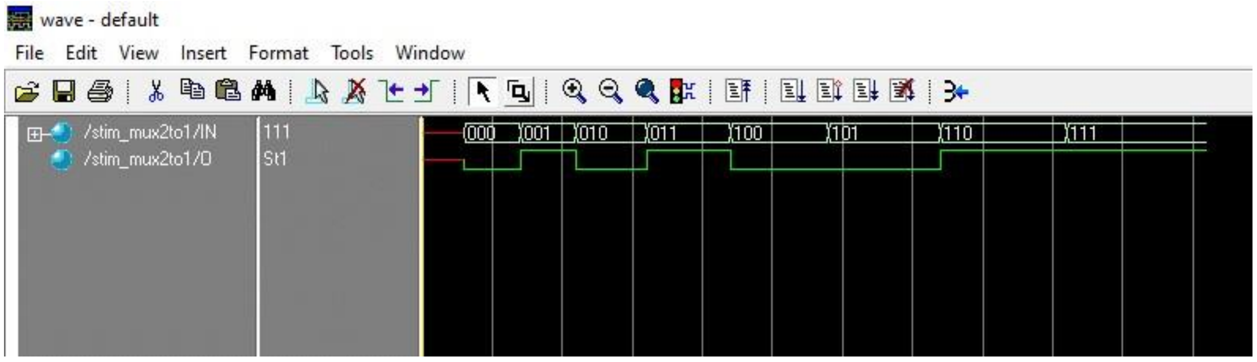
```

module mux_2tol(input [2:0]IN, output O);
        assign x=(IN[0] & ~IN[2]);
        assign y=(IN[1] & IN[2]);
        assign O=(x | y);
endmodule
```



```
module stim_mux2to1;
        reg [2:0]IN;
        wire O;
        mux_2to1 mux_21(IN, O);
        initial
        begin
                #1 $display("INPUT | OUTPUT");
                #2 IN[2]=0;IN[1]=0;IN[0]=0;
                #1 $monitor("%b   |%b",IN,O);
                #3 IN[2]=0;IN[1]=0;IN[0]=1;
                #4 IN[2]=0;IN[1]=1;IN[0]=0;
                #5 IN[2]=0;IN[1]=1;IN[0]=1;
                #6 IN[2]=1;IN[1]=0;IN[0]=0;
                #7 IN[2]=1;IN[1]=0;IN[0]=1;
                #8 IN[2]=1;IN[1]=1;IN[0]=0;
                #9 IN[2]=1;IN[1]=1;IN[0]=1;
                #10;
        end
endmodule
```

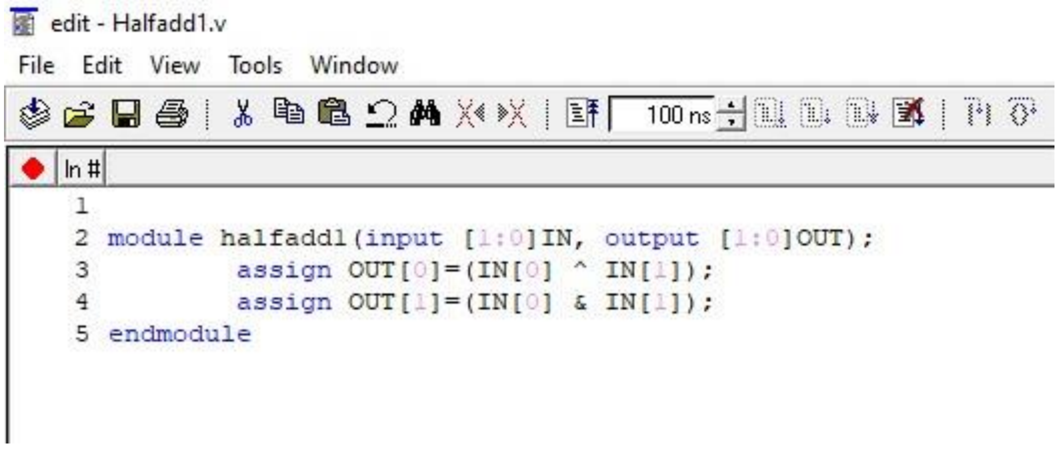
Output Wave:



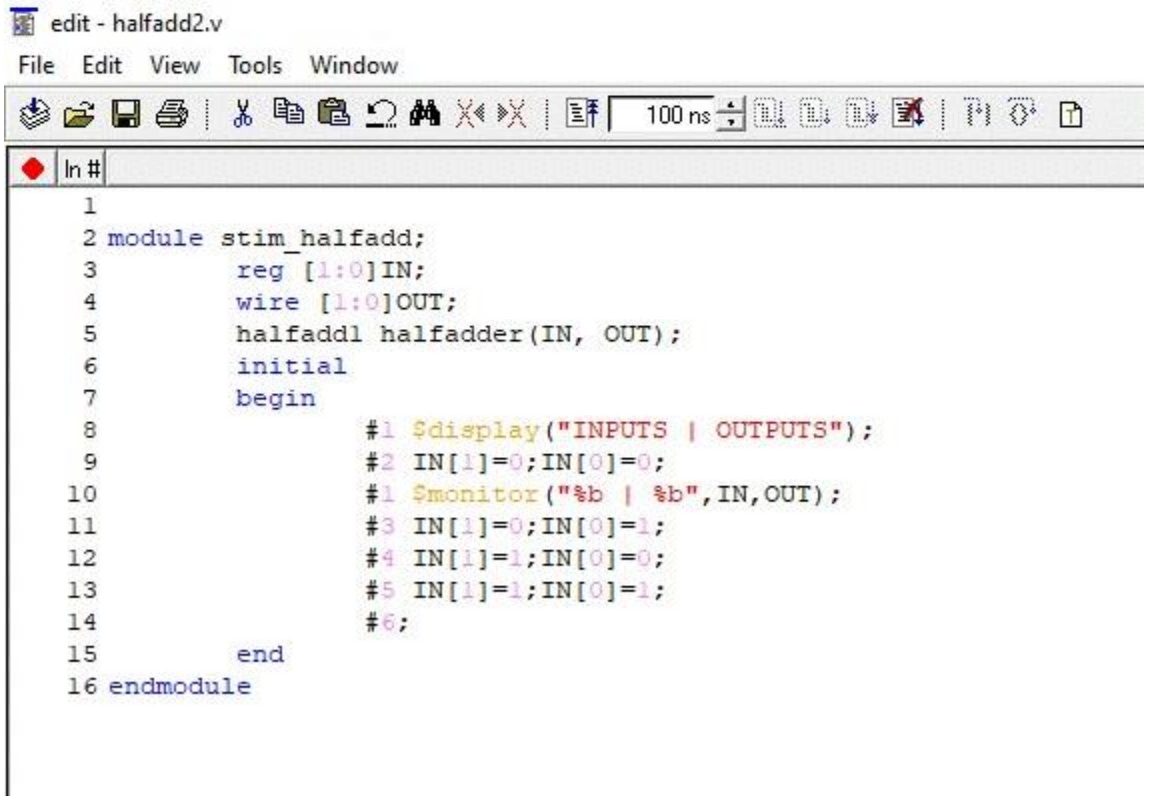
TASK03:

Write a Verilog code for Half Adder using Dataflow Level modeling.

Code:

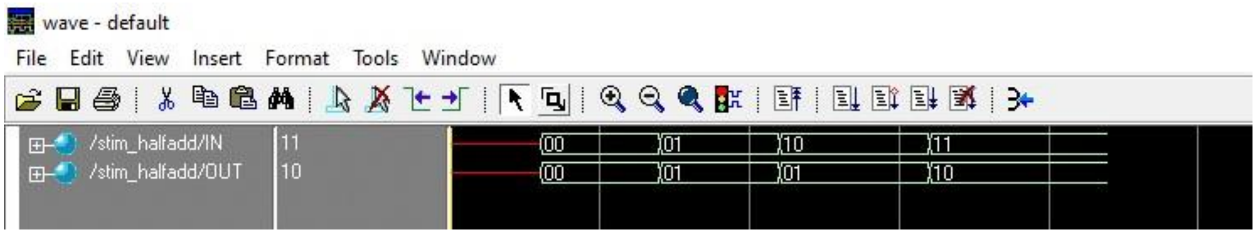


```
module halfadd1(input [1:0]IN, output [1:0]OUT);
        assign OUT[0]=(IN[0] ^ IN[1]);
        assign OUT[1]=(IN[0] & IN[1]);
endmodule
```



```
module stim_halfadd;
        reg [1:0]IN;
        wire [1:0]OUT;
        halfadd1 halfadder(IN, OUT);
        initial
        begin
                #1 $display("INPUTS | OUTPUTS");
                #2 IN[1]=0;IN[0]=0;
                #1 $monitor("%b | %b",IN,OUT);
                #3 IN[1]=0;IN[0]=1;
                #4 IN[1]=1;IN[0]=0;
                #5 IN[1]=1;IN[0]=1;
                #6;
        end
endmodule
```

Output Wave:

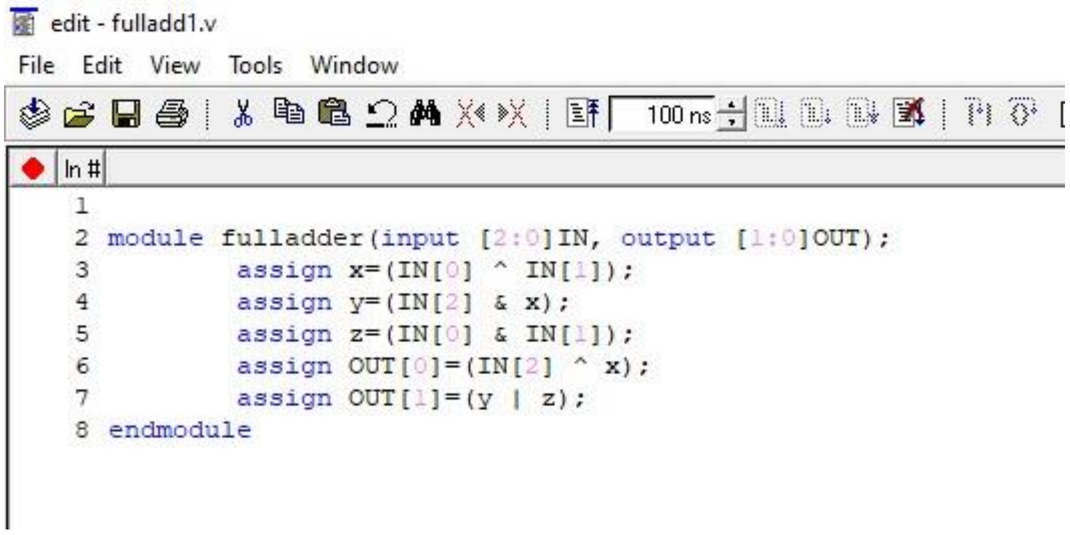


```
/stim_halfadd/IN   11   00  01  10  11
/stim_halfadd/OUT  10   00  01  01  10
```

TASK04:

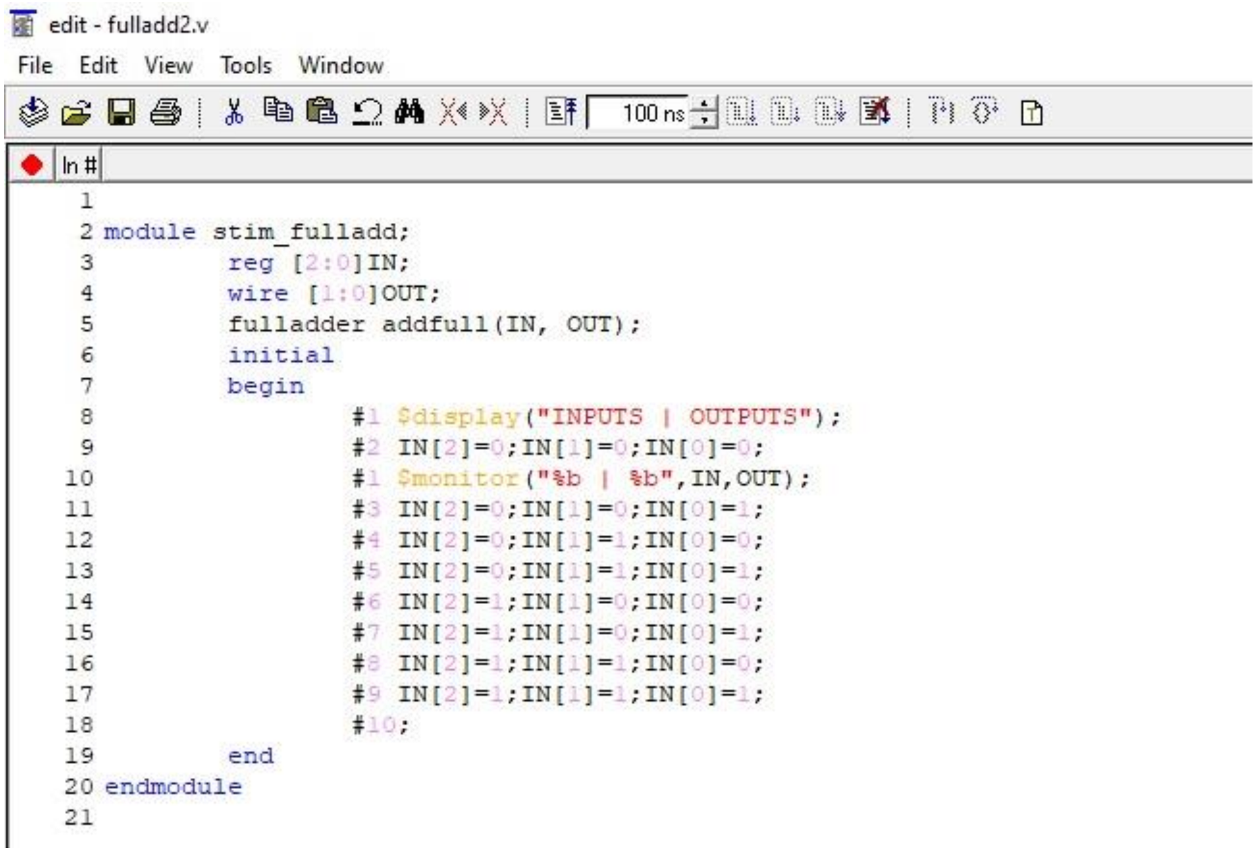
Write a Verilog code for Full Adder using Dataflow Level modeling.

Code:



```verilog

module fulladder(input [2:0]IN, output [1:0]OUT);
        assign x=(IN[0] ^ IN[1]);
        assign y=(IN[2] & x);
        assign z=(IN[0] & IN[1]);
        assign OUT[0]=(IN[2] ^ x);
        assign OUT[1]=(y | z);
endmodule
```

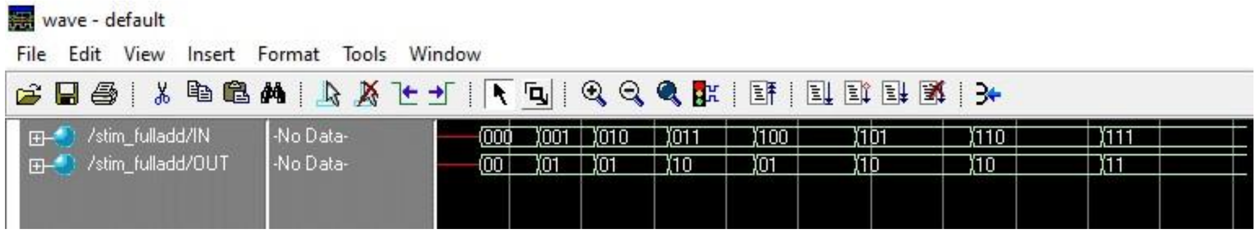


```verilog

module stim_fulladd;
        reg [2:0]IN;
        wire [1:0]OUT;
        fulladder addfull(IN, OUT);
        initial
        begin
                #1 $display("INPUTS | OUTPUTS");
                #2 IN[2]=0;IN[1]=0;IN[0]=0;
                #1 $monitor("%b | %b",IN,OUT);
                #3 IN[2]=0;IN[1]=0;IN[0]=1;
                #4 IN[2]=0;IN[1]=1;IN[0]=0;
                #5 IN[2]=0;IN[1]=1;IN[0]=1;
                #6 IN[2]=1;IN[1]=0;IN[0]=0;
                #7 IN[2]=1;IN[1]=0;IN[0]=1;
                #8 IN[2]=1;IN[1]=1;IN[0]=0;
                #9 IN[2]=1;IN[1]=1;IN[0]=1;
                #10;
        end
endmodule

```

Output Wave:

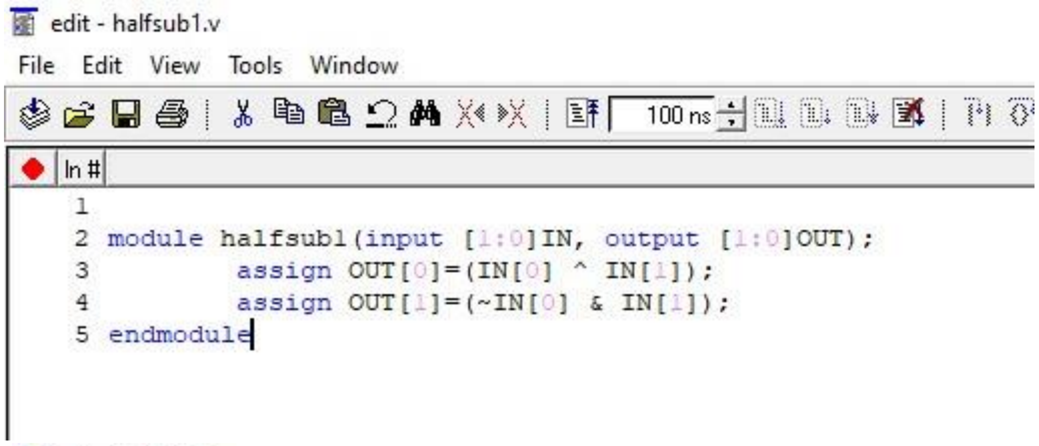


```
/stim_fulladd/IN    -No Data-   000  001  010  011  100  101  110  111
/stim_fulladd/OUT   -No Data-   00   01   01   10   01   10   10   11
```

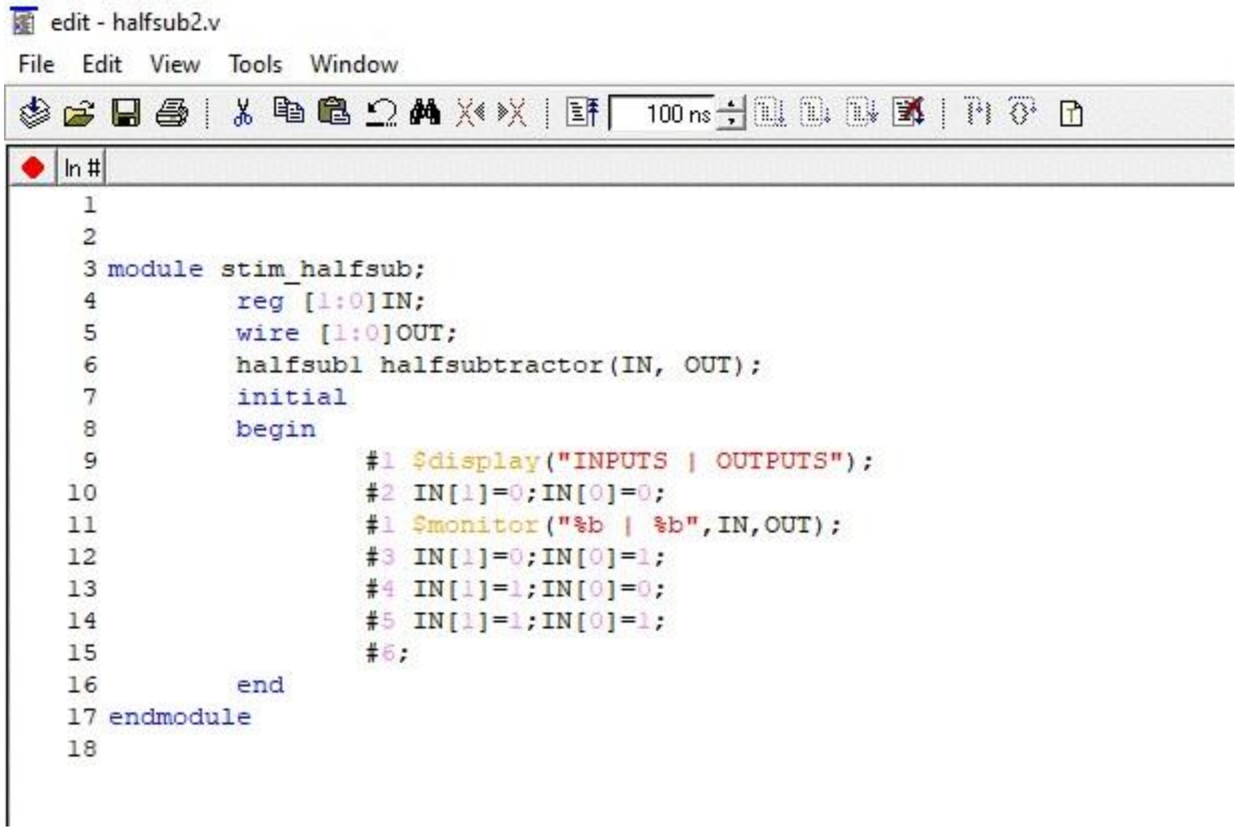
TASK05:

Write a Verilog code for Half Subtractor using Dataflow Level modeling.

Code:



```
module halfsub1(input [1:0]IN, output [1:0]OUT);
        assign OUT[0]=(IN[0] ^ IN[1]);
        assign OUT[1]=(~IN[0] & IN[1]);
endmodule
```

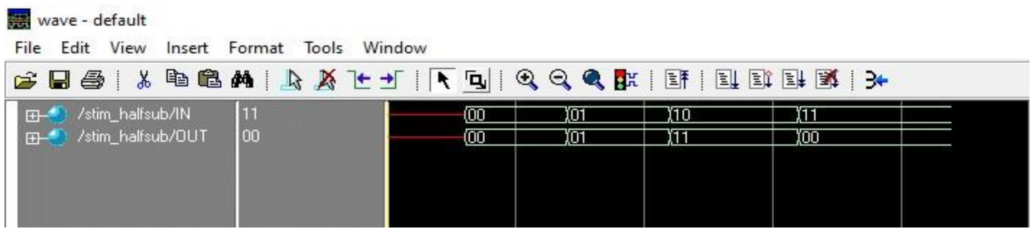


```


module stim_halfsub;
        reg [1:0]IN;
        wire [1:0]OUT;
        halfsub1 halfsubtractor(IN, OUT);
        initial
        begin
                #1 $display("INPUTS | OUTPUTS");
                #2 IN[1]=0;IN[0]=0;
                #1 $monitor("%b | %b",IN,OUT);
                #3 IN[1]=0;IN[0]=1;
                #4 IN[1]=1;IN[0]=0;
                #5 IN[1]=1;IN[0]=1;
                #6;
        end
endmodule

```

Output Wave:


wave - default
File Edit View Insert Format Tools Window
/stim_halfsub/IN
/stim_halfsub/OUT