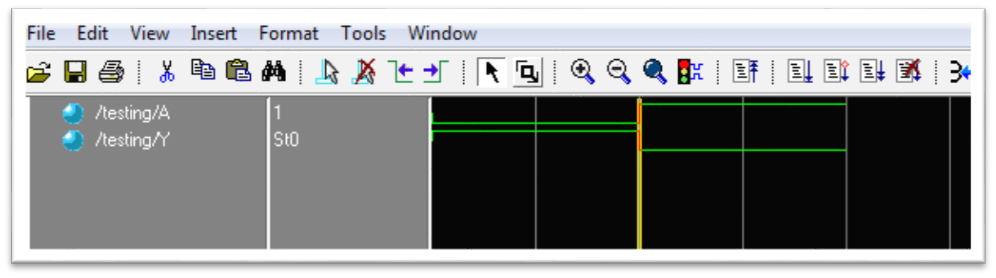
Verilog programs



Snapshot: NOT gate simulation

# Code

//start of program

//<<<<<<<<<<<<<creating a module (just like class)

module notgate(Y,A);//named notgate

//specifying input and output….

input A;

output Y;

//installing a not gate in module by calling not

not mynot(Y,A);

endmodule //>>>>>>>>>>>>>

//specifying timescale which instructs about the delay and resolution

`timescale 1ns/1ps

//creating another module (class) to test our circuit

module testing;

reg A; //register for input

wire Y; //wire for output

notgate testing\_not(Y,A); //creating an instance of out module

initial

begin

//now we specify all the possible inputs that we want out circuit to test on…

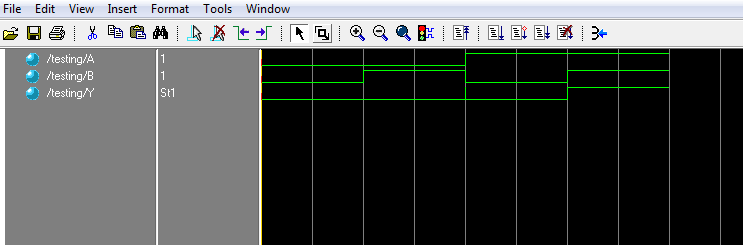
A =1'b0;

#10 A=1'b1;

#10; //this representes time delay

end

endmodule



Snapshot: AND Gate simulation

# Code

//start of program

//<<<<<<<<<<<<<creating a module (just like class)

module my\_and(Y,A,B); //named my\_and

input A,B; //specifiying input

output Y; //specifies output

and a1(Y,A,B); //arrangement is necessary….(calling and from lib)

endmodule

//telling time delay and resolution we want

`timescale 1ns/1ps

module testing;

reg A,B; //reg for input

wire Y; //wire for output

my\_and test\_and(Y,A,B); //instantiating the module we built

initially

begin

//now we tell all the possible inputs to test out circuit on

A = 1'b0; B= 1'b0;

#10 A = 1'b0; B= 1'b1;

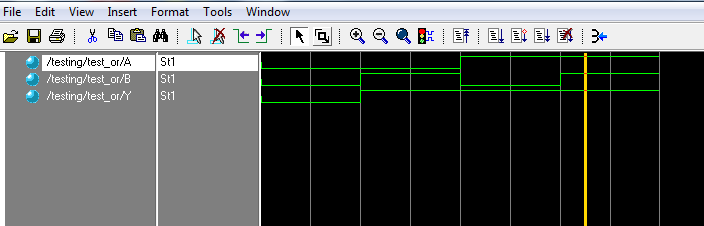
#10 A = 1'b1; B= 1'b0;

#10 A = 1'b1; B= 1'b1;

#10; // time delay

end

endmodule



Snapshot: OR Gate

# Code

//creating module for or gate

module my\_or(Y,A,B); //named my\_or

input A,B; //telling inputs

output Y; //telling the output

or a1(Y,A,B); //calling from library

endmodule

//instructing about the time delay and resolution we want

`timescale 1ns/1ps

module testing;

reg A,B; //reg for input

wire Y; //wire for output

my\_or test\_or(Y,A,B); //instantiating the module we made….. the arrangement of input and output are //necessary in these type of functions or modules

initial

begin

//specifiying all the simulation that we want out circuit to test on….it includes all the possible values

A = 1'b0; B= 1'b0;

#10 A = 1'b0; B= 1'b1;

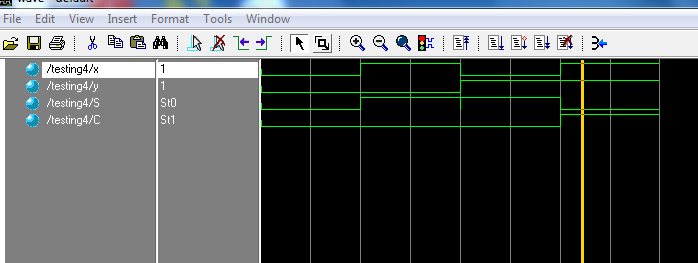
#10 A = 1'b1; B= 1'b0;

#10 A = 1'b1; B= 1'b1;

#10;

end

endmodule



Snapshot: Task 3

# Code

//making out module that we would use to sum two bits

module get(S,C,x,y); //named get

input x,y; //these are absolute inputs

output S,C; //final outputs

wire k1,k2,k3,k4; //wires for temporary

not c1(k1,x); //**not**ing x :D

not c2(k2,y); //**not**ing y :D

and a1(k3,y,k1); //anding y and x’

and a2(k4,x,k2); //anding x and y’

or c3(S,k3,k4); //oring x’y and xy’-----this gives sum

and c4(C,x,y); //anding x and y ----- this gives carry

endmodule

//specifying resolution and time delay

`timescale 1ns/1ps

//new class in a way

module testing4;

reg x,y;//inputas

wire S,C;//outputs

get t7(S,C,x,y); //instantiating out object

initial

begin

//specifying the track

x=1'b0; y=1'b0;

#10 x=1'b1; y=1'b0;

#10 x=1'b0; y=1'b1;

#10 x=1'b1; y=1'b1;

#10; //delay

end

endmodule

//MUJHAY ANDAY WALA BURGER