ECE LAB ASSIGNMENT

AND GATE

Design:

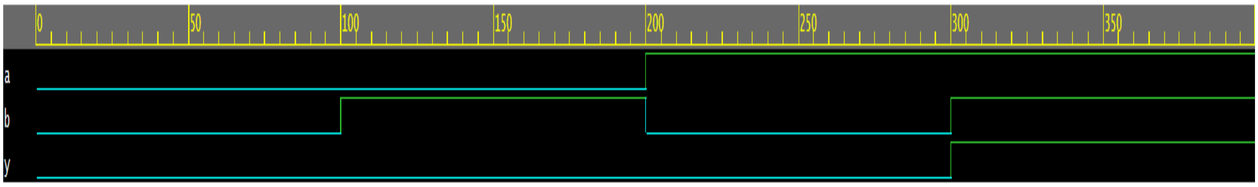
```
module andgate(input a,b, output y);
  and(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  andgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("andgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
```

```
   a=0; b=0; #100;
   a=0; b=1; #100;
   a=1; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

Output



BEHAVIOURAL MODEL

Design:

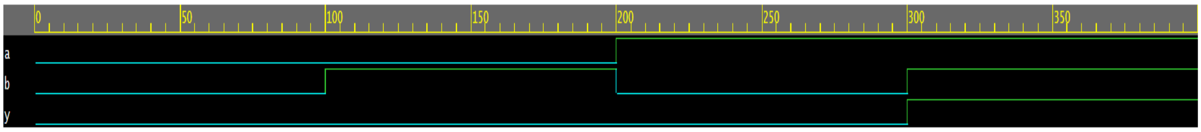
```
module andgate(a,b,y);
input a,b;
 output reg y;
 always@(a or b)
  begin
   if(a==0 & b==0)
    y=0;
   else if(a==0&b==1)
    y=0;
   else if(a==1&b==0);
   y=0;
   else y=1;
```

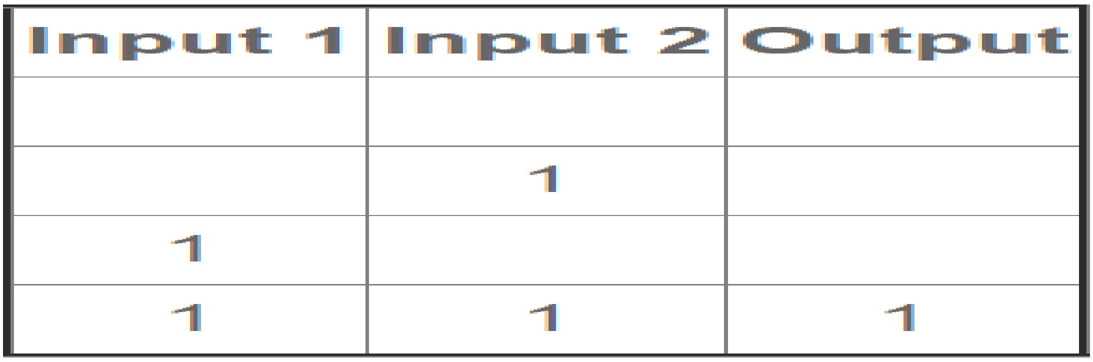
```
    end

endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  andgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("andgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output:





Input 1	Input 2	Output
	1	
1		
1	1	1

OR GATE

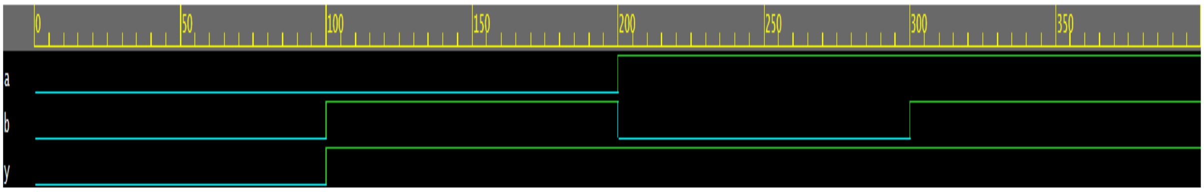
Design:

```
module orgate(a,b,y);
  input a,b;
  output y;
  or o1(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  orgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("orgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output:



BEHAVIOURAL MODEL

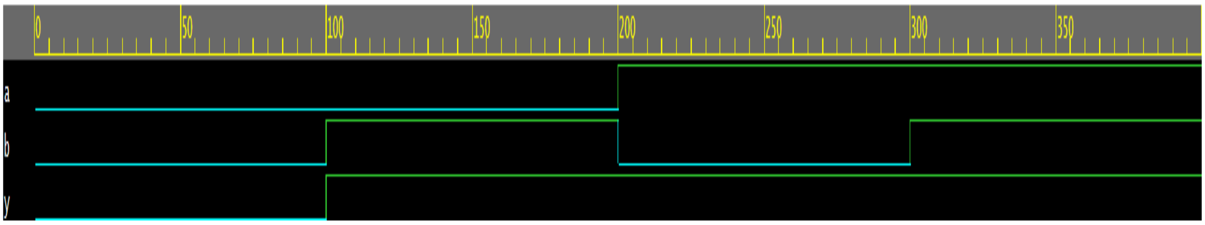
Design:

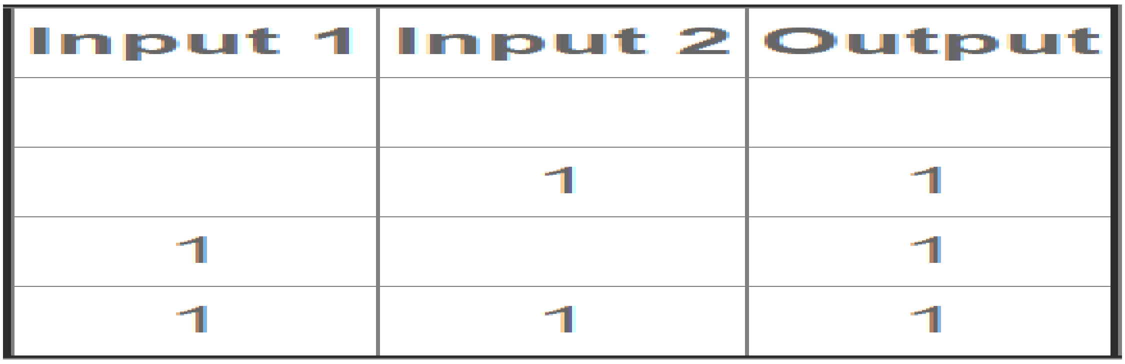
```
module orgate(a,b,y);
  input a,b;
  output reg y;
  always @(a orb)
    begin
      if(a==0 & b==0)
        y=0;
      else
        y=1;
    end
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  orgate uut(.a(a),.b(b),.y(y));

  initial
  begin
   $dumpfile("orgate.vcd");
   $dumpvars(1);
  end

 initial
  begin
   a=0; b=0; #100;
   a=0; b=1; #100;
   a=1; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

Output:





Input 1	Input 2	Output
	1	1
1		1
1	1	1

NOT GATE

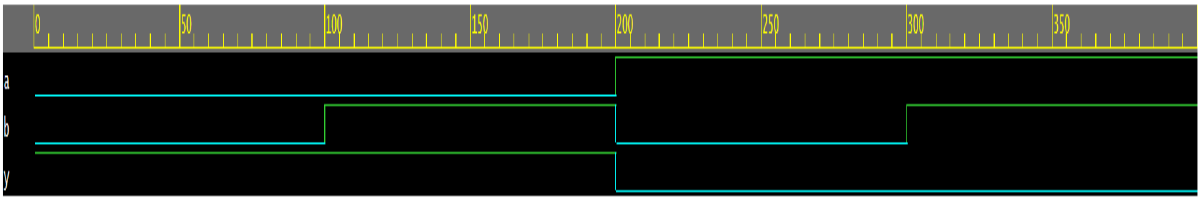
Design:

```
module notgate(a,b,y);
  input a,b;
  output y;
  not n1(y,a);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  notgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("notgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output:



BEHAVIOURAL MODEL

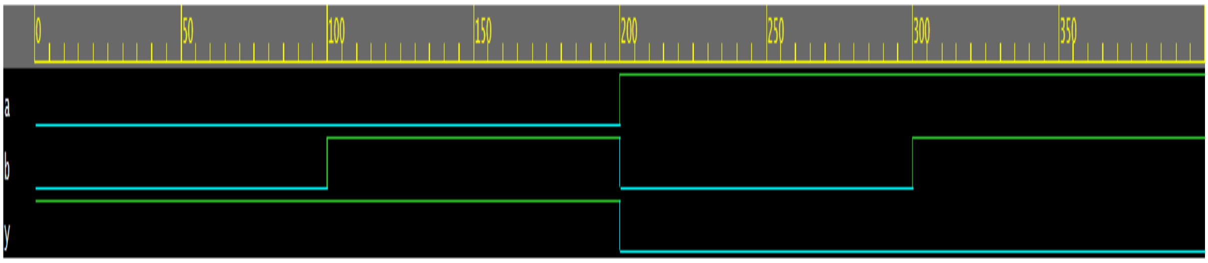
Design:

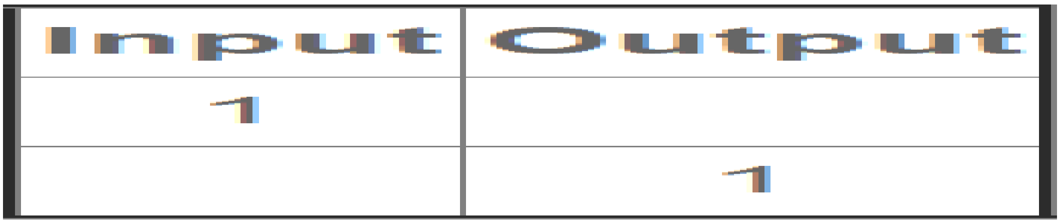
```
module notgate(a,b,y);
 input a,b;
 output reg y;
 always @(a)
   begin
     if(a==0)
       y=1;
     else
       y=0;
   end
endmodule
```

Testbench:

```
module test();
 reg a,b;
 wire y;
 notgate uut(.a(a),.b(b),.y(y));


 initial
   begin
     $dumpfile("notgate.vcd");
     $dumpvars(1);
   end


 initial
   begin
     a=0; b=0; #100;
     a=0; b=1; #100;
     a=1; b=0; #100;
     a=1; b=1; #100;
   end
endmodule
```

Output:





Input	Output
1	
	1

NOR GATE

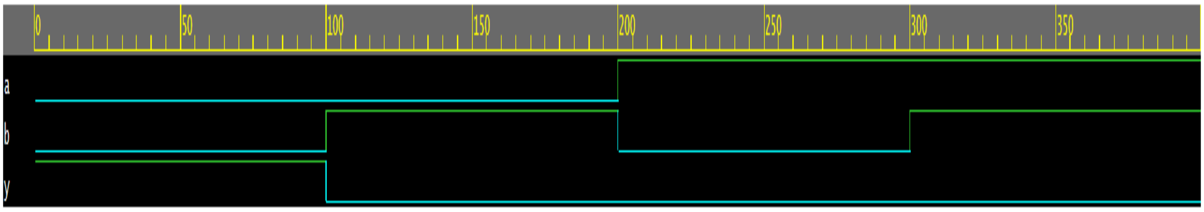
Design:

```
module norgate(a,b,y);
  input a,b;
  output y;
  nor n1(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  norgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("norgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output:



BEHAVIOURAL MODEL

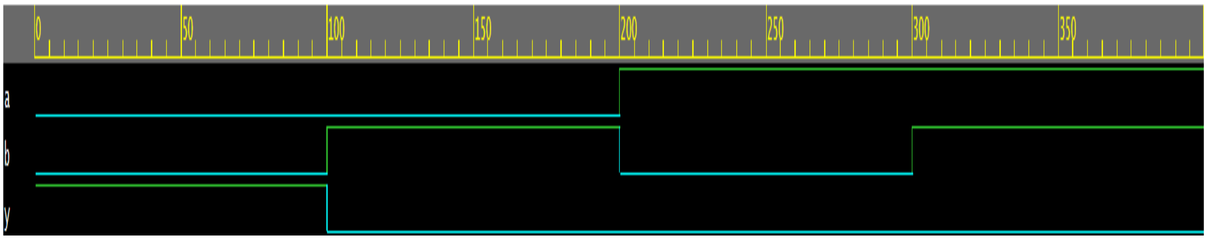
Design:

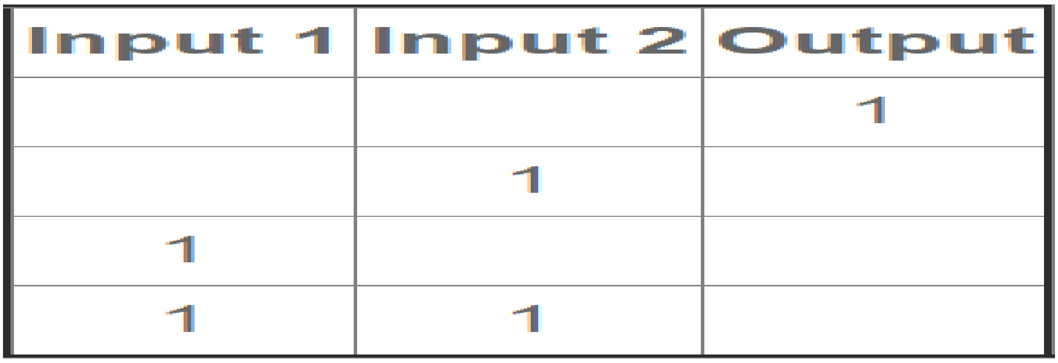
```
module norgate(a,b,y);
  input a,b;
  output reg y;
  always @(a or b)
    begin
      if(a==0 & b==0)
        y=1;
      else
        y=0;
    end
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  norgate uut(.a(a),.b(b),.y(y));

  initial
    begin
    $dumpfile("norgate.vcd");
    $dumpvars(1);
  end

 initial
  begin
    a=0; b=0; #100;
    a=0; b=1; #100;
    a=1; b=0; #100;
    a=1; b=1; #100;
  end
endmodule
```

Output:





Input 1	Input 2	Output
		1
	1	
1		
1	1	

NAND GATE

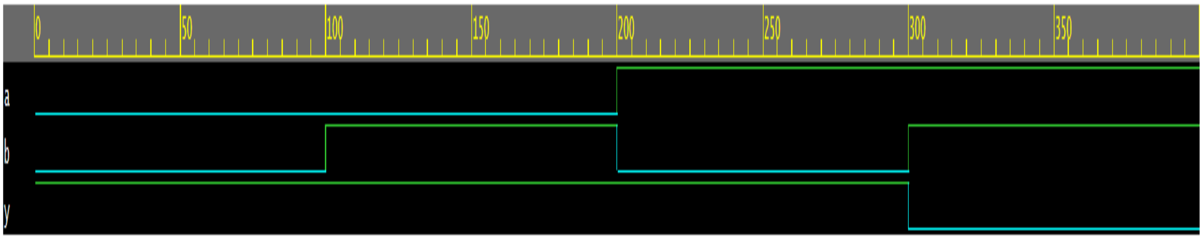
Design:

```
module nandgate(a,b,y);
  input a,b;
  output y;
  nand n1(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  nandgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("nandgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output



BEHAVIOURAL MODEL

Design:

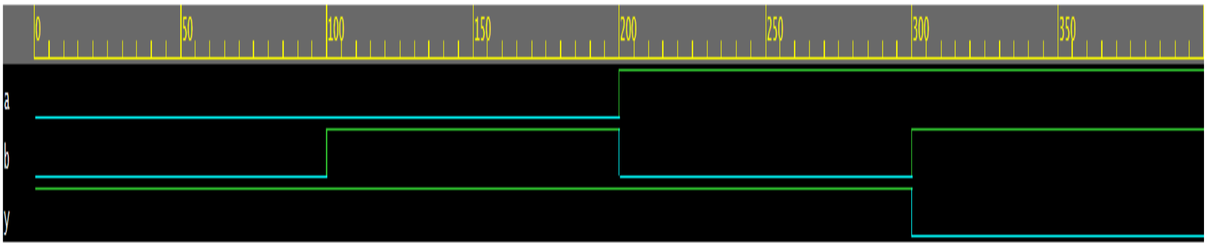
```
module nandgate(a,b,y);
  input a,b;
  output reg y;
  always @(a or b)
    begin
      if(a==1 & b==1)
        y=0;
      else
        y=1;
    end
endmodule
```

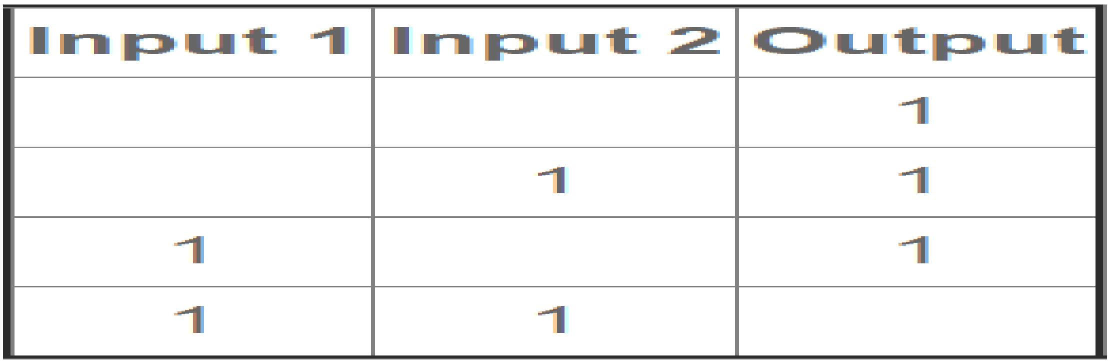
Testbench:

```
module test();
```

```
 reg a,b;
 wire y;

 nandgate uut(.a(a),.b(b),.y(y));

 initial
  begin
   $dumpfile("nandgate.vcd");
   $dumpvars(1);
  end

 initial
  begin
   a=0; b=0; #100;
   a=0; b=1; #100;
   a=1; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

Output





Input 1	Input 2	Output
		1
	1	1
1		1
1	1	

XOR GATE

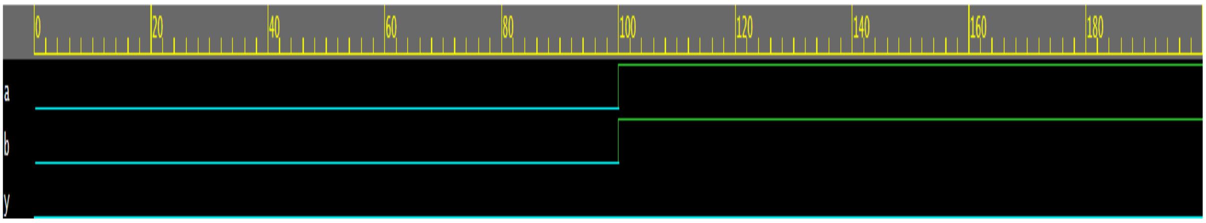
Design:

```
module xorgate(a,b,y);
  input a,b;
  output y;
  xor x1(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  xorgate uut(.a(a),.b(b),.y(y));

  initial
  begin
   $dumpfile("xorgate.vcd");
   $dumpvars(1);
  end

 initial
  begin
   a=0; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

Output



BEHAVIOURAL MODEL

Design

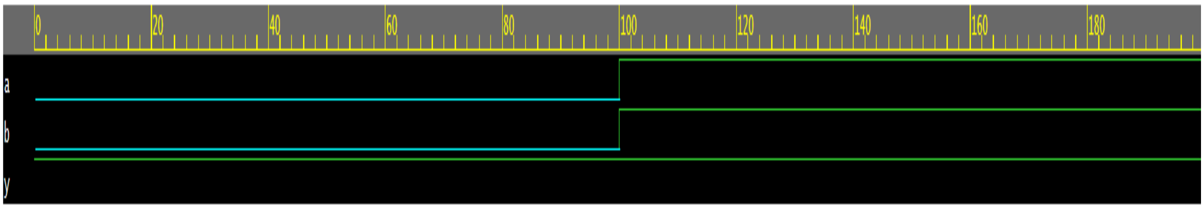
```
module xorgate(a,b,y);
 input a,b;
 output reg y;
 always @(a or b)
  begin
   if(a==0 & b==0)
    y=0;
   else if(a==1 & b==1)
    y=0;
```

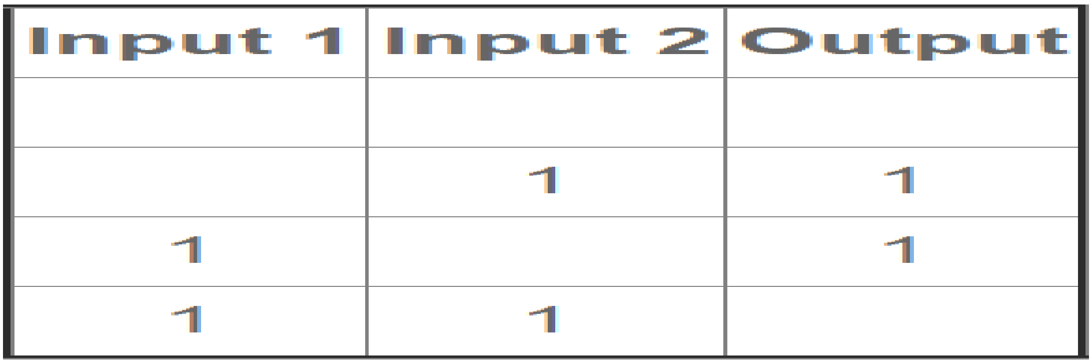
```
      else
        y=1;
    end
endmodule
```

Testbench

```
module test();
  reg a,b;
  wire y;

  xorgate uut(.a(a),.y(y));

  initial
    begin
      $dumpfile("xorgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output





Input 1	Input 2	Output
	1	1
1		1
1	1	

XNOR GATE

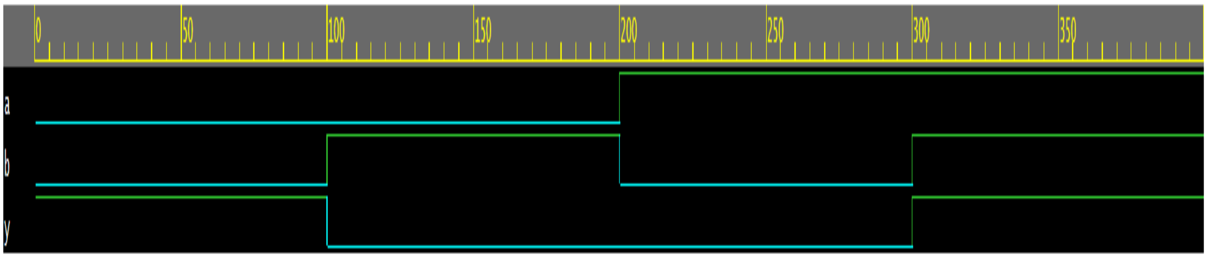
Design:

```
module xnorgate(a,b,y);
  input a,b;
  output y;
  xnor x1(y,a,b);
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  xnorgate uut(.a(a),.b(b),.y(y));

  initial
  begin
   $dumpfile("xnorgate.vcd");
   $dumpvars(1);
  end

 initial
  begin
   a=0; b=0; #100;
   a=0; b=1; #100;
   a=1; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

Output:



BEHAVIOURAL MODEL

Design:

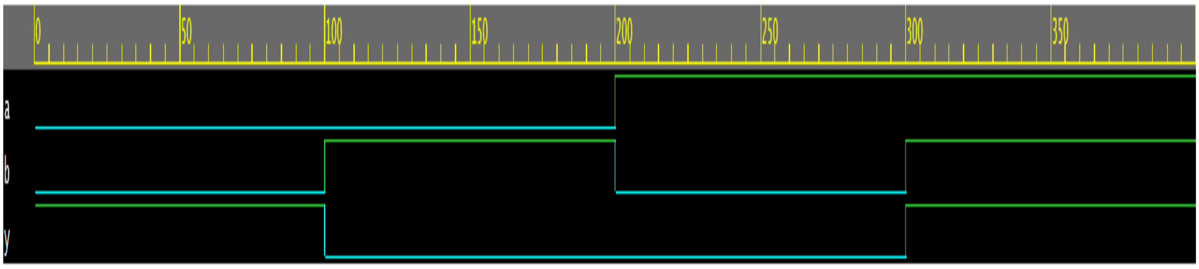
```
module xnorgate(a,b,y);
 input a,b;
 output reg y;
 always @(a or b)
  begin
   if(a==0 & b==0)
    y=1;
```

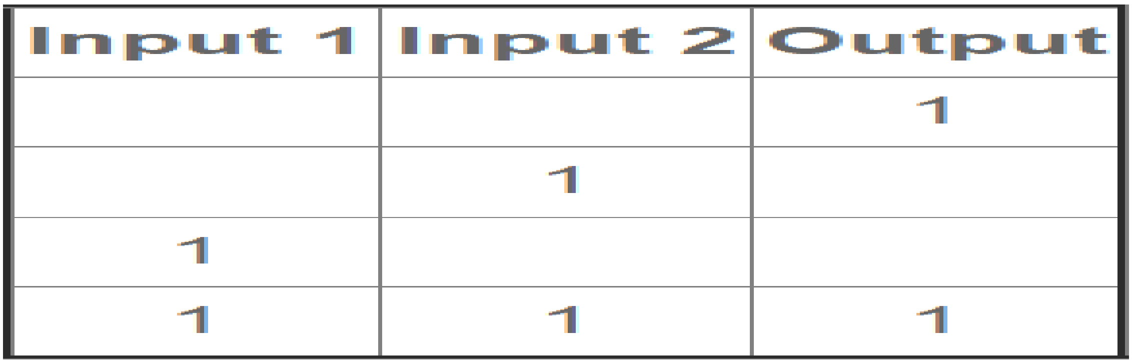
```
      else if(a==1&b==1)
        y=1;
      else
        y=0;
    end
endmodule
```

Testbench:

```
module test();
  reg a,b;
  wire y;

  xnorgate uut(.a(a),.b(b),.y(y));

  initial
    begin
      $dumpfile("xnorgate.vcd");
      $dumpvars(1);
    end

  initial
    begin
      a=0; b=0; #100;
      a=0; b=1; #100;
      a=1; b=0; #100;
      a=1; b=1; #100;
    end
endmodule
```

Output:





Input 1	Input 2	Output
		1
	1	
1		
1	1	1