

COA - Lab 2

🕒 Created	@Jan 21, 2021 5:43 PM
🏷️ Tags	

Basic Gates:

- AND
- OR
- NOT
- NOR
- NAND
- XOR
- XNOR

AND gate:

Module:

```
module anddemo(x,y,z);  
input x,y;  
output z;  
assign z = x&y;  
endmodule
```

Testbench:

```
module andgate_demo_tb;  
reg t_a, t_b;  
wire t_y;
```

```
anddemo my_gate(t_a, t_b, t_y);
```

```
initial
begin
    $monitor(t_a, t_b, t_y);
    t_a=0; t_b=0; #10;
    t_a=0; t_b=1; #10;
    t_a=1; t_b=0; #10;
    t_a=1; t_b=1; #10;
    $finish;
end
endmodule
```

```
nirmalk@nimmyOS-arch ~$ master + ./verilogtrial
000
010
100
111
```

OR Gate:

Module:

```
module orgate(a, b, y);
input a, b;
output y;
assign y = a | b;
endmodule
```

Testbench:

```
module orgate_test;
wire t_y;
reg t_a, t_b;

orgate test_gate(t_a, t_b, t_y);

initial
begin
```

```

$monitor(t_a, t_b, t_y);
t_a = 1'b0;
t_b = 1'b0;

#5

t_a = 1'b0;
t_b = 1'b1;

#5

t_a = 1'b1;
t_b = 1'b0;

#5

t_a = 1'b1;
t_b = 1'b1;

end
endmodule

```

```

nirmalk@nimmyOS-arch:~/Sem4Code/iVerilog/lab_2$ ./verilogtrial
000
011
101
111
nirmalk@nimmyOS-arch:~/Sem4Code/iVerilog/lab_2$ _

```

NOT Gate:

Module:

```

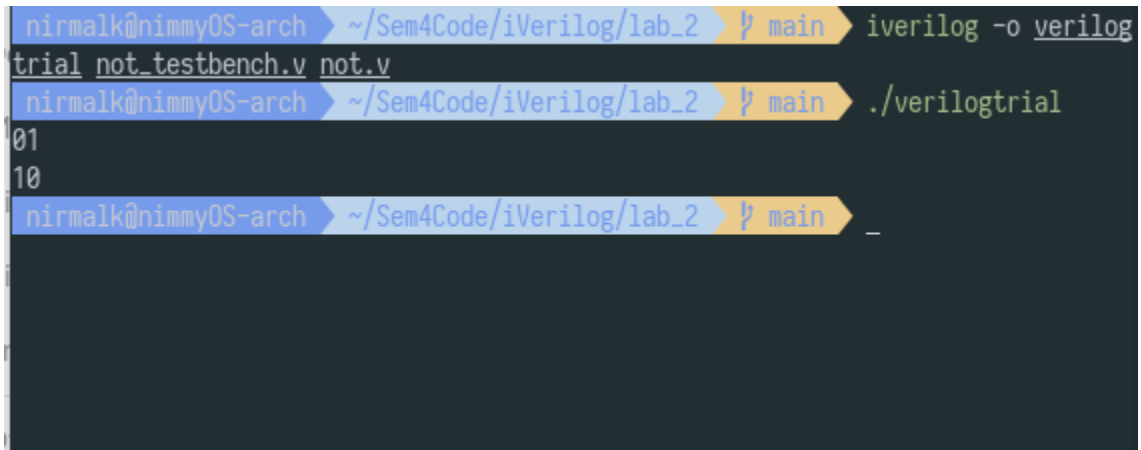
module notgate(a,y);
input a;

```

```
output y;  
assign y = !a;  
endmodule
```

Testbench:

```
module notgate_test;  
wire t_y;  
reg t_a;  
  
notgate test_gate(t_a,t_y);  
  
initial  
begin  
$monitor(t_a,t_y);  
t_a = 1'b0;  
  
#5  
  
t_a = 1'b1;  
  
end  
endmodule
```



A terminal window showing the execution of Verilog commands. The prompt is 'nirmalk@nimmyOS-arch'. The first command is 'iverilog -o verilogtrial not_testbench.v not.v', which compiles the testbench and the module. The second command is './verilogtrial', which runs the simulation. The output shows the values of 't_a' and 't_y' at two time steps: 01 and 10. At time 01, 't_a' is 0 and 't_y' is 1. At time 10, 't_a' is 1 and 't_y' is 0. The prompt ends with a dash '_'.

```
nirmalk@nimmyOS-arch ~ /Sem4Code/iVerilog/lab_2 % main iverilog -o verilog  
trial not_testbench.v not.v  
nirmalk@nimmyOS-arch ~ /Sem4Code/iVerilog/lab_2 % main ./verilogtrial  
01  
10  
nirmalk@nimmyOS-arch ~ /Sem4Code/iVerilog/lab_2 % main _
```

NOR Gate

Module:

```

module norgate(a,b,y);
input a,b;
output y;
assign y = ~(a | b);
endmodule

```

Testbench:

```

module norgate_test;
wire t_y;
reg t_a,t_b;

norgate test_gate(t_a,t_b,t_y);

initial
begin

    t_y = 0;
    $monitor(t_a,t_b,t_y);
    t_a = 1'b0;
    t_b = 1'b0;

    #5

    t_a = 1'b0;
    t_b = 1'b1;

    #5

    t_a = 1'b1;
    t_b = 1'b0;

    #5

    t_a = 1'b1;
    t_b = 1'b1;

end
endmodule

```

```

X nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 1/ main iverilog -o veril
ogtrial nor_testbench.v nor.v
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 1/ main ± ./verilogtrial
001
010
100
110
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 1/ main ± _

```

NAND gate:

Module:

```

module nandgate(a,b,y);
input a,b;
output y;
assign y = ~(a & b);
endmodule

```

Testbench:

```

module nandgate_test;
wire t_y;
reg t_a,t_b;

nandgate test_gate(t_a,t_b,t_y);

initial
begin
$monitor(t_a,t_b,t_y);
t_a = 1'b0;
t_b = 1'b0;

#5

t_a = 1'b0;
t_b = 1'b1;

#5

t_a = 1'b1;
t_b = 1'b0;

```

#5

```
t_a = 1'b1;  
t_b = 1'b1;  
  
end  
endmodule
```

```
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 % iverilog -o veril  
ogtrial nand_testbench.v nand.v  
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 % ./verilogtrial  
001  
011  
101  
110  
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 %
```

XOR Gate:

Module:

```
module xorgate(a,b,y);  
input a,b;  
output y;  
assign y = ((a&!b) | (b&!a));  
endmodule
```

Testbench:

```
module xorgate_test;  
wire t_y;  
reg t_a,t_b;  
  
xorgate test_gate(t_a,t_b,t_y);  
  
initial  
begin  
$monitor(t_a,t_b,t_y);  
t_a = 1'b0;
```

```

t_b = 1'b0;

#5

t_a = 1'b0;
t_b = 1'b1;

#5

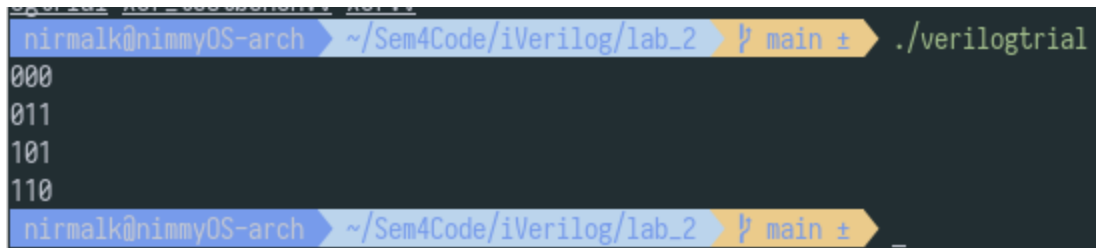
t_a = 1'b1;
t_b = 1'b0;

#5

t_a = 1'b1;
t_b = 1'b1;

end
endmodule

```



```

nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 $ ./verilogtrial
000
011
101
110
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 $

```

XNOR Gate:

Module:

```

module xnorgate(a,b,y);
input a,b;
output y;
assign y = !((a&!b) | (b&!a));
endmodule

```

Testbench:


```

module xnorgate_test;
wire t_y;
reg t_a,t_b;

xnorgate test_gate(t_a,t_b,t_y);

initial
begin
$monitor(t_a,t_b,t_y);
t_a = 1'b0;
t_b = 1'b0;

#5

t_a = 1'b0;
t_b = 1'b1;

#5

t_a = 1'b1;
t_b = 1'b0;

#5

t_a = 1'b1;
t_b = 1'b1;

end
endmodule

```

```

nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 iverilog -o veril
ogtrial xnor_testbench.v xnor.v
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2 ./verilogtrial
001
010
100
111
nirmalk@nimmyOS-arch ~/Sem4Code/iVerilog/lab_2

```