Amrita Vishwa Vidyapeetham
Amrita School of Engineering, Bangalore
Department of Computer Science and Engineering

19CSE211 Computer Organization and Architecture
Lab Handout - 2
Basic Gates

Exercise Problems

1. Write a verilog code to implement AND gate in gate level modeling

```
module and_gl(a, b, c);
input a, b;
output c;
and a1(c,a,b);
endmodule
```

```
module and1_tb;

reg a, b;
wire c;

and_gl i(a, b, c);

initial
begin
    a=1'b0;
    b=1'b0;
    $monitor("Time:%0t a=%b
b=%b c=%b",$time, a, b, c);
    #5 a=1'b0; b=1'b0;
    #5 a=1'b0; b=1'b1;
    #5 a=1'b1; b=1'b0;
    #5 a=1'b1; b=1'b1;
end
endmodule
```

2. Write a verilog code to implement AND gate in data flow modeling

```
module and_dl(a, b, c);
input a, b;
output c;
assign c = a & b;
endmodule
```

```
module and1_tb;

reg a, b;
wire c;

and_dl i(a, b, c);

initial
begin
    a=1'b0;
    b=1'b0;
    $monitor("Time:%0t a=%b
b=%b c=%b",$time, a, b, c);
    #5 a=1'b0; b=1'b0;
    #5 a=1'b0; b=1'b1;
    #5 a=1'b1; b=1'b0;
    #5 a=1'b1; b=1'b1;
end
endmodule
```

3. Write a verilog code to implement AND gate in behavioral modeling

```
module and_bl(a, b, c);
input a, b;
output c;
reg c;

always@(a or b or c)
begin
    if(a==1'b1 && b==1'b1)
        c = 1'b1;
    else
        c = 1'b0;
end
endmodule
```

```
module and1_tb;

reg a, b;
wire c;

and_bl i(a, b, c);

initial
begin
    a=1'b0;
    b=1'b0;
    $monitor("Time:%0t a=%b
b=%b c=%b",$time, a, b, c);
    #5 a=1'b0; b=1'b0;
    #5 a=1'b0; b=1'b1;
    #5 a=1'b1; b=1'b0;
    #5 a=1'b1; b=1'b1;
end
endmodule
```

4. Write a verilog code to implement 5 inputs AND gate in gate level modeling

```
module and_5in(in1, in2, in3,
in4, in5, out);
input in1, in2, in3, in4, in5;
output out;
and a1(out, in1, in2, in3, in4,
in5);
endmodule
```

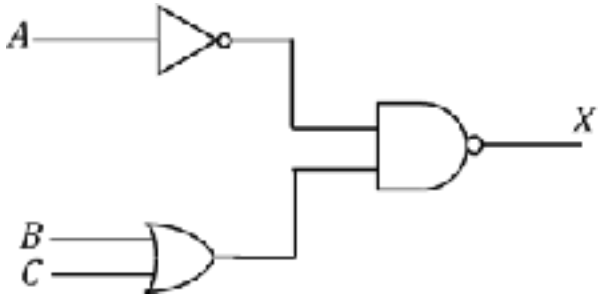
```
module and1_tb;

reg in1, in2, in3, in4, in5;
wire out;

and_5in i(in1, in2, in3, in4,
in5, out);

initial
begin
    in1=1'b0; in2=1'b0;
in3=1'b0; in4=1'b0; in5=1'b0;
    $monitor("Time:%0t
in=%b%b%b%b%b out=%b",$time,
in1, in2, in3, in4, in5, out);
    #5  in1=1'b0; in2=1'b0;
in3=1'b0; in4=1'b0; in5=1'b0;
    #5  in1=1'b0; in2=1'b0;
in3=1'b1; in4=1'b0; in5=1'b1;
    #5  in1=1'b1; in2=1'b1;
in3=1'b1; in4=1'b1; in5=1'b1;
end
endmodule
```

5. Write a verilog code to implement below logic circuit using gate level modeling




```
module exp5(a,b,c,x);
input a, b, c;
output x;
wire w1, w2;

not n1(w1, a);
or o1(w2, b,c);
nand n2(x, w1, w2);

endmodule
```

```
module exp5_tb;

reg a, b, c;
wire x;

exp5 i(a, b, c, x);

initial
begin
    a = 1'b0;
    b = 1'b0;
    c = 1'b0;
    $monitor("Time:%0t a=%b
b=%b, c=%b, x=%b",$time, a, b,
c, x);
    #5 a= 1'b0; b=1'b0; c=1'b0;
    #5 a= 1'b0; b=1'b0; c=1'b1;
    #5 a= 1'b0; b=1'b1; c=1'b0;
    #5 a= 1'b0; b=1'b1; c=1'b1;
    #5 a= 1'b1; b=1'b0; c=1'b0;
    #5 a= 1'b1; b=1'b0; c=1'b1;
    #5 a= 1'b1; b=1'b1; c=1'b0;
    #5 a= 1'b1; b=1'b1; c=1'b1;
end
endmodule
```

Assignment

1. Write a verilog code to implement following gate in data flow level modeling

a. OR

b. NAND

c. EXOR (XOR)

2. Write a verilog code to implement the below logic circuit using gate level and data flow modeling