Verilog

1) Implementation of basic gates using gate level modeling

DESIGN:

```
module basic_gates_G(
input a,
input b,
output [6:0] y);

        and a1(y[6],a,b);
        or o1(y[5],a,b);

        nand n1(y[4],a,b);
        nor n2(y[3],a,b);

        xor x1(y[2],a,b);
        xnor x2(y[1],a,b);

        not n3(y[0],a);

endmodule
```

TestBench:

```
module tb;
        reg in1,in2;
        wire [6:0] out;

        basic_gates_G b1 ( .y(out), .b(in2),.a(in1));

        initial begin
                $monitor (" %b  %b  %b",in1,in2,out);
                {in1,in2}=2'b00; #10;
                {in1,in2}=2'b01; #10;
                {in1,in2}=2'b10; #10;
                {in1,in2}=2'b11; #10;
                End
endmodule
```

run.do:

```
#cd
{C:\Users\alisy\Documents\verilog\class1\basicgates}

#environment creation
vlib work

#compilation
vlog my_not_gate.v
vlog my_not_gate_tb.v

#simulation
vsim work.my_not_tb
run -all
```

LOGIC FUNCTION	LOGIC SYMBOL	BOOLEAN EXPRESSION	TRUTH TABLE		
			INPUTS		OUTPUTS
			B	A	Y
AND		A+B=Y	0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		A+B=Y	0	0	0
			0	1	1
			1	0	1
			1	1	1
inverter		$A=\overline{A}$		0	1
				1	0
NAND		$\overline{A+B}=Y$	0	1	1
			0	0	1
			1	1	0
NOR		$\overline{A+B}=Y$	1	0	1
			0	1	0
			0	0	0
			1	1	0
XOR		$A \oplus B=Y$	1	0	0
			0	1	0
			0	0	1
			1	1	1
XNOR		$\overline{A \oplus B}=Y$	1	0	0
			0	0	1
			0	1	0
			1	0	0
			1	1	1

Fig 2.34 Summary of basic logic gates