**Design verilog code NAND,XOR,XNOR**

// Code your design here

// We are writting code xor,xnor,nand gates

**module special\_gates(a,b,c,d,e);**

**input a,b;**

**output c,d,e;**

//output reg c,d,e ; we are also write in a single line

**reg c,d,e;**

//frist you started with always block

**always@(a,b)** // in side of bracket change input date to change output date

**begin**

**c=~(a&b);** // in place of ~ to change ! ,same output show you do slef cheack out output same or not.

**d=a^b;**

**e=~(a^b);**

**end**

**endmodule**

**Testbench verilog code NAND,XOR,XNOR**

// Code your testbench here

// in testbench module name is not same design module name .

**module tbspecial\_gates();**

// design input in testbench write in reg data type and output is assign wire type .

**wire c,d,e;**

**reg a,b;**

**integer i;** // i is stored the value

/\*we are writting code to connect design part in testbench using uut(unit under test)\*/

// syntex= name of design module name , uut,(write all input and output name).

**special\_gates uut(a,b,c,d,e);**

//special\_gates uut(.a(a) .b(b).c(c).d(d).e(e); this is also correct

**initial**

**begin**

**$display(" time I/N a I/N b O/P NAND O/P XOR O/P XNOR");**

**$monitor($time," a=%b b=%b c=%b d=%b e=%b",a,b,c,d,e);**

**{a,b}=i;** // this syntx means two bit stored in {a,b},

**for(i=0;i<=4;i=i+1)**

**begin**

**{a,b}=i**;// ex1 i=0 ,a=0 b=0 ex2 i=1 a=0 b=1

**#5;** //means every loop 5 unit time delay

**end**

**$finish**;// program end

**end**

**endmodule**

**Input and output value**

**time I/N a I/N b O/P NAND O/P XOR O/P XNOR**

**0 a=0 b=0 c=1 d=0 e=1  
 5 a=0 b=1 c=1 d=1 e=0  
 10 a=1 b=0 c=1 d=1 e=0  
 15 a=1 b=1 c=0 d=0 e=1  
 20 a=0 b=0 c=1 d=0 e=1**