GATE LEVEL MODELLING

1.

```
module a1_g(output b,input a);
 not(b,a);
endmodule
module a1_g_tb;
 reg a;
 wire b;
 a1_g Instance (b,a);
 initial begin
   a=0;
   #5 a=1;
 end
 initial begin
   $monitor("t=%t | b=%b | a=%b",$time,b,a);
 end
endmodule
```

2.

```
module a2_g(output c,input a,b);
 and(c,a,b);
endmodule
module a2_g_tb;
 wire c;
 reg a,b;
 a2_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

3.

```
module a3_g(output c,input a,b);
 or(c,a,b);
endmodule
module a3_g_tb;
 wire c;
 reg a,b;
 a3_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

4.

```
module a4_g(output c,input a,b);
```

```
 nand(c,a,b);
endmodule
module a4_g_tb;
 wire c;
 reg a,b;
 a4_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

5.

```
module a5_g(output c,input a,b);
 nor(c,a,b);
endmodule
module a5_g_tb;
 wire c;
 reg a,b;
 a5_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

6.

```
module a6_g(output c,input a,b);
 xor(c,a,b);
endmodule
module a6_g_tb;
 wire c;
 reg a,b;
 a6_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

7.

```
module a7_g(output c,input a,b);
 xnor(c,a,b);
```

```
endmodule
module a7_g_tb;
 wire c;
 reg a,b;
 a7_g Instance (c,a,b);
 initial begin
   a=0;b=0;
   #5 a=0;b=1;
   #5 a=1;b=0;
   #5 a=1;b=1;
 end
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

8.

```
module a8_g(output i,input a,b,c,d,e,f,g,h);
  nor(i,a,b,c,d,e,f,g,h);
endmodule
module a8_g_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a8_g Instance (i,a,b,c,d,e,f,g,h);
 initial begin
   $monitor("time=%t | i=%b | a=%b | b=%b | c=%b | d=%b | e=%b | f=%b | g=%b | h=%b",$time,a,b,c,d,e,f,g,h);
 end
endmodule
```

9.

```
module a9_g(output i,input a,b,c,d,e,f,g,h);
  nand(i,a,b,c,d,e,f,g,h);
endmodule
module a9_g_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a9_g Instance (i,a,b,c,d,e,f,g,h);
 initial begin
   $monitor("time=%t | i=%b | a=%b | b=%b | c=%b | d=%b | e=%b | f=%b | g=%b | h=%b",$time,a,b,c,d,e,f,g,h);
 end
endmodule
```

10.

```
module a10_d(output [15:0] c, input [15:0] a);
not g1[15:0](c,a);
endmodule
module a10_d_tb;
 reg [15:0] a;
 wire [15:0] c;
 a10_d Instance (c,a);
 initial begin
   $monitor("t=%t | c=%b | a=%b",$time,c,a);
 end
endmodule
```

11.

```
module a11_d(output [15:0]c, input [15:0] a, b);
and g1[15:0](c,a,b);
endmodule
```

```
module a11_d_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a11_d Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```
12.
```
module a12_d(output [15:0]c, input [15:0] a, b);
or g1[15:0](c,a,b);
endmodule
module a12_d_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a12_d Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```
13.
```
module a13_d(output [15:0]c, input [15:0] a, b);
nand g1[15:0](c,a,b);
endmodule
module a13_d_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a13_d Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```
14.
```
module a14_d(output [15:0]c, input [15:0] a, b);
nor g1[15:0](c,a,b);
endmodule
module a14_d_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a14_d Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```
15.
```
module a15_g(output f,input a,b,c);
  wire f1,f2,f3;
  nand(f1,~a,b);
  and(f2,b,c);
  and(f3,~a,~b);
  or(f,f1,f2,f3);
endmodule
module a15_g_tb;
 reg a,b,c;
 wire f;
```

```
 a15_g Instance (f,a,b,c);
 initial begin
   $monitor("time=%t| f=%b | a=%b | b=%b | c=%b",$time,f,a,b,c);
 end
endmodule
```

16.

USING NAND GATES

```
module g_not_nand(output b,input a);
  nand(b,a,a);
endmodule
module g_not_nand_tb;
 wire b;
 reg a;
 g_not_nand Instance (b,a);
 initial begin
   a=0;
   #10 a=1;
 end
 initial begin
   $monitor("time=%t | b=%b | a=%b",$time,b,a);
 end
endmodule
module g_and_nand(output c,input a,b);
  wire f1;
  nand(f1,a,b);
  nand(c,f1,f1);
endmodule
module g_and_nand_tb;
 wire c;
 reg a,b;
 g_and_nand Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_or_nand(output c,input a,b);
  wire f1,f2;
  nand(f1,a,a);
  nand(f2,b,b);
  nand(c,f1,f2);
endmodule
module g_or_nand_tb;
 wire c;
 reg a,b;
 g_or_nand Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
```

```
  #10 a=1;b=1;
 end
 initial begin
  $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_xor_nand(output c,input a,b);
 wire f1,f2,f3;
 nand(f1,a,b);
 nand(f2,f1,a);
 nand(f3,f1,b);
 nand(c,f2,f3);
endmodule
module g_xor_nand_tb;
 wire c;
 reg a,b;
 g_xor_nand Instance (c,a,b);
 initial begin
  a=0;b=0;
  #10 a=0;b=1;
  #10 a=1;b=0;
  #10 a=1;b=1;
 end
 initial begin
  $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_xnor_nand(output c,input a,b);
 wire f1,f2,f3,f4;
 nand(f1,a,b);
 nand(f2,f1,a);
 nand(f3,f1,b);
 nand(f4,f2,f3);
 nand(c,f4,f4);
endmodule
module g_xnor_nand_tb;
 wire c;
 reg a,b;
 g_xnor_nand Instance (c,a,b);
 initial begin
  a=0;b=0;
  #10 a=0;b=1;
  #10 a=1;b=0;
  #10 a=1;b=1;
 end
 initial begin
  $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
USING NOR GATES
module g_xnor_nor(output c,input a,b);
 wire f1,f2,f3;
 nor(f1,a,b);
 nor(f2,f1,a);
 nor(f3,f1,b);
```

```
 nor(c,f2,f3);
endmodule
module g_xnor_nor_tb;
 wire c;
 reg a,b;
 g_xnor_nor Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_xor_nor(output c,input a,b);
  wire f1,f2,f3,f4;
  nor(f1,a,a);
  nor(f2,b,b);
  nor(f3,a,b);
  nor(f4,f1,f2);
  nor(c,f3,f4);
endmodule
module g_xor_nor_tb;
 wire c;
 reg a,b;
 g_xor_nor Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_or_nor(output c,input a,b);
  wire f1;
  nor(f1,a,b);
  nor(c,f1,f1);
endmodule
module g_or_nor_tb;
 wire c;
 reg a,b;
 g_or_nor Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
```

```
 end
endmodule
module g_and_nor(output c,input a,b);
  wire f1,f2;
  nor(f1,a,a);
  nor(f2,b,b);
  nor(c,f1,f2);
endmodule
module g_and_nor_tb;
 wire c;
 reg a,b;
 g_and_nor Instance (c,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
module g_not_nor(output b,input a);
  nor(b,a,a);
endmodule
module g_not_nor_tb;
 wire b;
 reg a;
 g_not_nor Instance (b,a);
 initial begin
   a=0;
   #10 a=1;
 end
 initial begin
   $monitor("time=%t | b=%b | a=%b",$time,b,a);
 end
endmodule
```

17.

```
module a17_g(output c,s,input a,b);
  xor(s,a,b);
  and(c,a,b);
endmodule
module a17_g_tb;
 wire c,s;
 reg a,b;
 a17_g Instance(c,s,a,b);
 initial begin
   a=0;b=0;
   #10 a=0;b=1;
   #10 a=1;b=0;
   #10 a=1;b=1;
 end
 initial begin
   $monitor("time=%t | c=%b | s=%b | a=%b | b=%b",$time,c,s,a,b);
 end
```

```
endmodule
18.
module full_adder(input a,input b,input c_in,output s,output c_out);
      wire f1,f2;
  xor(s,a,b,c_in);
      and(f1,a,b);
  xor(f2,c_in,f1);
      or(c_out,f1,f2);
endmodule
module a18_g(output c_out,output [3:0] S,input [3:0] A,input [3:0] B,input c_in);
   wire C1,C2,C3;
   full_adder fa0 (A[0],B[0],c_in,S[0],C1);
   full_adder fa1 (A[1],B[1],C1,S[1],C2);
   full_adder fa2 (A[2],B[2],C2,S[2],C3);
   full_adder fa3 (A[3],B[3],C3,S[3],c_out);
endmodule
module a18_g_tb;
  reg [3:0] a;
  reg [3:0] b;
  reg c_in;
  wire c_out;
  wire [3:0] sum;
  integer i;
  a18_g Instance (c_out,sum,a,b,c_in);
  initial begin
    a <= 0;
    b <= 0;
    c_in <= 0;
    $monitor ("a=0x%0h b=0x%0h c_in=0x%0h c_out=0x%0h sum=0x%0h", a, b, c_in, c_out, sum);
    for (i = 0; i < 5; i = i+1) begin
      #10 a <= $random;
          b <= $random;
              c_in <= $random;
    end
  end
endmodule
```