DATA LEVEL MODELLING

1.

```
module a1_d(output b,input a);
 assign b=~a;
endmodule
module a1_d_tb;
 reg a;
 wire b;
 a1_d 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_d(output c,input a,b);
 assign c=(a & b);
endmodule
module a2_d_tb;
 wire c;
 reg a,b;
 a2_d 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_d(output c,input a,b);
 assign c=a|b;
endmodule
module a3_d_tb;
 wire c;
 reg a,b;
 a3_d 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_d(output c,input a,b);
```

```
 assign c=~(a & b);
endmodule
module a4_d_tb;
 wire c;
 reg a,b;
 a4_d 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_d(output c,input a,b);
 assign c=~(a|b);
endmodule
module a5_d_tb;
 wire c;
 reg a,b;
 a5_d 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_d(output c,input a,b);
 assign c=(a&~b)|(~a&b);
endmodule
module a6_d_tb;
 wire c;
 reg a,b;
 a6_d 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_d(output c,input a,b);
 assign c=(~a&~b)|(a&b);
```

```
endmodule
module a7_d_tb;
 wire c;
 reg a,b;
 a7_d 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_d(output i,input a,b,c,d,e,f,g,h);
  assign i=~(a|b|c|d|e|f|g|h);
endmodule
module a8_d_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a8_d 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_d(output i,input a,b,c,d,e,f,g,h);
  assign i=~( a & b & c & d & e & f & g & h );
endmodule
module a9_d_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a9_d 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);
assign 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);
assign 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);
assign 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);
assign 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);
assign 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_d(output f,input a,b,c);
  wire f1,f2,f3;
  assign f1=~(~a&b);
  assign f2=(b&c);
  assign f3=(~a&~b);
  assign f=(f1|f2|f3);
endmodule
module a15_d_tb;
 reg a,b,c;
 wire f;
 a15_d 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 d_not_nand(output b,input a);
  assign b=~(a&a);
endmodule
module d_not_nand_tb;
 wire b;
 reg a;
 d_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 d_and_nand(output c,input a,b);
  wire f1;
  assign f1=~(a&b);
  assign c=~(f1&f1);
endmodule
module d_and_nand_tb;
 wire c;
 reg a,b;
 d_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 d_or_nand(output c,input a,b);
  wire f1,f2;
  assign f1=~(a&a);
  assign f2=~(b&b);
  assign c=~(f1&f2);
endmodule
module d_or_nand_tb;
 wire c;
 reg a,b;
 d_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 d_xor_nand(output c,input a,b);
  wire f1,f2,f3;
  assign f1=~(a&b);
  assign f2=~(f1&a);
  assign f3=~(f1&b);
  assign c=~(f2&f3);
endmodule
module d_xor_nand_tb;
 wire c;
 reg a,b;
 d_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 d_xnor_nand(output c,input a,b);
 wire f1,f2,f3,f4;
 assign f1=~(a&b);
 assign f2=~(f1&a);
 assign f3=~(f1&b);
 assign f4=~(f2&f3);
 assign c=~(f4&f4);
endmodule
module d_xnor_nand_tb;
 wire c;
 reg a,b;
 d_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 d_not_nor(output b,input a);
  assign b=~(a|a);
endmodule
module d_not_nor_tb;
 wire b;
 reg a;
 d_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
module d_and_nor(output c,input a,b);
  wire f1,f2;
  assign f1=~(a|a);
  assign f2=~(b|b);
  assign c=~(f1|f2);
endmodule
module d_and_nor_tb;
 wire c;
 reg a,b;
 d_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 d_or_nor(output c,input a,b);
  wire f1;
  assign f1=~(a|b);
  assign c=~(f1|f1);
endmodule
module d_or_nor_tb;
 wire c;
 reg a,b;
 d_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 d_xor_nor(output c,input a,b);
  wire f1,f2,f3,f4;
  assign f1=~(a|a);
  assign f2=~(b|b);
  assign f3=~(a|b);
  assign f4=~(f1|f2);
  assign c=~(f3|f4);
endmodule
module d_xor_nor_tb;
 wire c;
 reg a,b;
 d_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 d_xnor_nor(output c,input a,b);
 wire f1,f2,f3;
 assign f1=~(a|b);
 assign f2=~(f1|a);
 assign f3=~(f1|b);
 assign c=~(f2|f3);
endmodule
module d_xnor_nor_tb;
 wire c;
 reg a,b;
 d_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
```

17.

```
module a17_d(output c,s,input a,b);
  assign s=(a^b);
  assign c=(a&b);
endmodule
module a17_d_tb;
 wire c,s;
 reg a,b;
 a17_d 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 a18_d(output c_out,output [3:0] sum,input [3:0] a,input [3:0] b,input c_in);
  assign {c_out, sum} = a + b + c_in;
endmodule
module a18_d_tb;
   reg [3:0] a;
   reg [3:0] b;
   reg c_in;
wire c_out;
   wire [3:0] sum;
   integer i;
   a18_d 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
```