BEHAVIOURAL LEVEL MODELLING

1.

```
module a1_b(output reg b,input a);
  always@(a or b)
    begin
    if(a==1'b0)
    begin
      b=1'b1;
    end
  else
    b=1'b0;
    end
endmodule
module a1_b_tb;
 reg a;
 wire b;
 a1_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
     if(a==1'b1 & b==1'b1)
     begin
     c=1'b1;
   end
       else
    c=1'b0;
     end
endmodule
module a2_b_tb;
 wire c;
 reg a,b;
 a2_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
```

```
     if(a==1'b0 & b==1'b0)
     begin
     c=1'b0;
   end
       else
    c=1'b1;
     end
endmodule
module a3_b_tb;
 wire c;
 reg a,b;
 a3_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
     if(a==1'b1 & b==1'b1)
     begin
     c=1'b0;
   end
       else
    c=1'b1;
     end
endmodule
module a4_b_tb;
 wire c;
 reg a,b;
 a4_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
     if(a==1'b0 & b==1'b0)
     begin
     c=1'b1;
```

```
   end
      else
    c=1'b0;
     end
endmodule
module a5_b_tb;
 wire c;
 reg a,b;
 a5_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
     if((a==1'b1 & b==1'b0) | (a==1'b0 & b==1'b1))
     begin
     c=1'b1;
   end
   else
     c=1'b0;
   end
endmodule
module a6_b_tb;
 wire c;
 reg a,b;
 a6_b 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_b(output reg c,input a,b);
 always@(a or b)
   begin
     if((a==1'b1 & b==1'b0) | (a==1'b0 & b==1'b1))
     begin
     c=1'b0;
   end
   else
     c=1'b1;
```

```
   end
endmodule
module a7_b_tb;
 wire c;
 reg a,b;
 a7_b 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_b(output reg i,input a,b,c,d,e,f,g,h);
  always@(a or b or c or d or e or f or g or h)
    begin
    if(a==1'b0 & b==1'b0 & c==1'b0 & d==1'b0 & e==1'b0 & f==1'b0 & g== 1'b0 & h==1'b0)
    begin
      i=1'b1;
    end
  else
    i=1'b0;
  end
endmodule
module a8_b_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a8_b 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_b(output reg i,input a,b,c,d,e,f,g,h);
  always@(a or b or c or d or e or f or g or h)
    begin
    if(a==1'b1 & b==1'b1 & c==1'b1 & d==1'b1 & e==1'b1 & f==1'b1 & g== 1'b1 & h==1'b1)
    begin
      i=1'b0;
    end
  else
    i=1'b1;
  end
endmodule
module a9_b_tb;
 wire i;
 reg a,b,c,d,e,f,g,h;
 a9_b 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_b(output reg [15:0]c, input [15:0]a);
 always @ (a)
         begin
  integer i;
  for(i = 0;i < 16;i = i + 1) begin
   if(a[i] == 1'b0) begin
    c[i] = 1'b1;
   end
   else begin
    c[i] = 1'b0;
   end
  end
 end
endmodule
module a10_b_tb;
     wire [15:0]c;
     reg [15:0]a;
     a10_b Instance (c,a);
     initial begin
   $monitor("t=%t | c=%b | a=%b",$time,c,a);
 end
endmodule
11.
module a11_b(output reg [15:0]c,input [15:0] a,b);
   always @ (a or b)
         begin
  integer i;
  for(i = 0;i < 16;i = i + 1) begin
   if(a[i] == 1'b1 & b[i] == 1'b1) begin
    c[i] = 1'b1;
   end
   else begin
    c[i] = 1'b0;
   end
  end
 end
endmodule
module a11_b_tb;
 reg [15:0]a,b;
 wire [15:0]c;
 a11_b Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
12.
module a12_b(output reg [15:0]c, input [15:0] a, b);
 always @ (a or b)
         begin
  integer i;
  for(i = 0;i < 16;i = i + 1) begin
   if(a[i] == 1'b0 & b[i] == 1'b0) begin
    c[i] = 1'b0;
```

```
   end
   else begin
    c[i] = 1'b1;
   end
  end
 end
endmodule
module a12_b_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a12_b Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

13.

```
amodule a13_b(output reg [15:0]c, input [15:0] a, b);
 always @ (a or b)
         begin
  integer i;
  for(i = 0;i < 16;i = i + 1) begin
   if(a[i] == 1'b1 & b[i] == 1'b1) begin
    c[i] = 1'b0;
   end
   else begin
    c[i] = 1'b1;
   end
  end
 end
endmodule
module a13_b_tb;
 reg [15:0] a,b;
 wire [15:0]c;
 a13_b Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

14.

```
module a14_b(output reg [15:0]c, input [15:0] a, b);
 always @ (a or b)
         begin
  integer i;
  for(i = 0;i < 16;i = i + 1) begin
   if(a[i] == 1'b0 & b[i] == 1'b0) begin
    c[i] = 1'b1;
   end
   else begin
    c[i] = 1'b0;
   end
  end
 end
endmodule
module a14_b_tb;
 reg [15:0] a,b;
```

```
 wire [15:0]c;
 a14_b Instance (c,a,b);
 initial begin
   $monitor("t=%t | c=%b | a=%b | b=%b",$time,c,a,b);
 end
endmodule
```

15.

```
module a15_b(output reg f,input a,b,c);
  reg f1,f2,f3;
  always@(a or b)
    begin
    if(~a==1'b1 & b==1'b1)
    begin
      f1=1'b0;
    end
  else
    f1=1'b1;
  end
  always@(c or b)
    begin
    if(c==1'b1 & b==1'b1)
    begin
      f2=1'b1;
    end
  else
    f2=1'b0;
  end
  always@(a or b)
    begin
    if(~a==1'b1 & ~b==1'b1)
    begin
      f3=1'b1;
    end
  else
    f3=1'b0;
  end
  always@(f1 or f2 or f3)
    begin
    if(f1==1'b0 & f2==1'b0 & f3==1'b0)
    begin
      f=1'b0;
    end
  else
    f=1'b1;
  end
endmodule
module a15_b_tb;
 reg a,b,c;
 wire f;
 a15_b 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 b_not_nand(output reg b,input a);
  always@(a)
    begin
    if(a==1'b1 & a==1'b1)
    begin
      b=1'b0;
    end
  else
    b=1'b1;
  end
endmodule
module b_not_nand_tb;
 wire b;
 reg a;
 b_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 b_and_nand(output reg c,input a,b);
  reg f1;
  always@(a or b)
    begin
    if(a==1'b1 & b==1'b1)
    begin
      f1=1'b0;
    end
  else
    f1=1'b1;
  end
  always@(f1)
    begin
    if(f1==1'b0)
    begin
      c=1'b1;
    end
  else
    c=1'b0;
  end
endmodule
module b_and_nand_tb;
 wire c;
 reg a,b;
 b_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 b_or_nand(output reg c,input a,b);
  reg f1,f2;
  always@(a)
    begin
    if(a==1'b1)
    begin
      f1=1'b0;
    end
  else
    f1=1'b1;
  end
  always@(b)
    begin
    if(b==1'b1)
    begin
      f2=1'b0;
    end
  else
    f2=1'b1;
  end
  always@(f1 or f2)
    begin
    if(f1==1'b1 & f2==1'b1)
    begin
      c=1'b0;
    end
  else
    c=1'b1;
  end
endmodule
module b_or_nand_tb;
 wire c;
 reg a,b;
 b_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 b_xor_nand(output reg c,input a,b);
  reg f1,f2,f3;
  always@(a or b)
    begin
    if(a==1'b1 & b==1'b1)
    begin
      f1=1'b0;
```

```
    end
   else
    f1=1'b1;
   end
   always@(a or f1)
    begin
    if(a==1'b1 & f1==1'b1)
    begin
      f2=1'b0;
    end
   else
    f2=1'b1;
   end
   always@(f1 or b)
    begin
    if(f1==1'b1 & b==1'b1)
    begin
      f3=1'b0;
    end
   else
    f3=1'b1;
   end
   always@(f2 or f3)
    begin
    if(f2==1'b1 & f3==1'b1)
    begin
      c=1'b0;
    end
   else
    c=1'b1;
   end
endmodule
module b_xor_nand_tb;
 wire c;
 reg a,b;
 b_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 b_xnor_nand(output reg c,input a,b);
 reg f1,f2,f3,f4;
 always@(a or b)
   begin
     if(a==1'b1 & b==1'b1)
     begin
     f1=1'b0;
   end
   else
```

```
    f1=1'b1;
   end
   always@(a or f1)
   begin
    if(a==1'b1 & f1==1'b1)
    begin
    f2=1'b0;
   end
   else
    f2=1'b1;
   end
   always@(f1 or b)
   begin
    if(f1==1'b1 & b==1'b1)
    begin
    f3=1'b0;
   end
   else
    f3=1'b1;
   end
   always@(f2 or f3)
   begin
    if(f2==1'b1 & f3==1'b1)
    begin
    f4=1'b0;
   end
   else
    f4=1'b1;
   end
   always@(f4)
   begin
    if(f4==1'b1)
    begin
    c=1'b0;
   end
   else
    c=1'b1;
   end
endmodule
module b_xnor_nand_tb;
 wire c;
 reg a,b;
 b_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 b_not_nor(output reg b,input a);
```

```
  always@(a)
    begin
    if(a==1'b1)
    begin
      b=1'b0;
    end
  else
    b=1'b1;
  end
endmodule
module b_not_nor_tb;
 wire b;
 reg a;
 b_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 b_and_nor(output reg c,input a,b);
  reg f1,f2;
  always@(a)
    begin
    if(a==1'b1)
    begin
      f1=1'b0;
    end
  else
    f1=1'b1;
  end
  always@(b)
    begin
    if(b==1'b1)
    begin
      f2=1'b0;
    end
  else
    f2=1'b1;
  end
  always@(f1 or f2)
    begin
    if(f1==1'b0 & f2==1'b0)
    begin
      c=1'b1;
    end
  else
    c=1'b0;
  end
endmodule
module b_and_nor_tb;
 wire c;
 reg a,b;
```

```
 b_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 b_or_nor(output reg c,input a,b);
  reg f1;
  always@(a or b)
    begin
    if(a==1'b0 & b==1'b0)
    begin
      f1=1'b1;
    end
  else
    f1=1'b0;
  end
  always@(f1)
    begin
    if(f1==1'b0)
    begin
      c=1'b1;
    end
  else
    c=1'b0;
  end
endmodule
module b_or_nor_tb;
 wire c;
 reg a,b;
 b_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 b_xor_nor(output reg c,input a,b);
  reg f1,f2,f3,f4;
  always@(a)
    begin
    if(a==1'b1)
    begin
      f1=1'b0;
    end
  else
```

```
    f1=1'b1;
  end
  always@(b)
    begin
    if(b==1'b1)
    begin
      f2=1'b0;
    end
  else
    f2=1'b1;
  end
  always@(a or b)
    begin
    if(a==1'b0 & b==1'b0)
    begin
      f3=1'b1;
    end
  else
    f3=1'b0;
  end
  always@(f1 or f2)
    begin
    if(f1==1'b0 & f2==1'b0)
    begin
      f4=1'b1;
    end
  else
    f4=1'b0;
  end
  always@(f3 or f4)
    begin
    if(f3==1'b0 & f4==1'b0)
    begin
      c=1'b1;
    end
  else
    c=1'b0;
  end
endmodule
module b_xor_nor_tb;
 wire c;
 reg a,b;
 b_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 b_xnor_nor(output reg c,input a,b);
 reg f1,f2,f3;
```

```
 always@(a or b)
   begin
     if(a==1'b0 & b==1'b0)
     begin
     f1=1'b1;
   end
   else
     f1=1'b0;
   end
   always@(a or f1)
   begin
     if(a==1'b0 & f1==1'b0)
     begin
     f2=1'b1;
   end
   else
     f2=1'b0;
   end
   always@(f1 or b)
   begin
     if(f1==1'b0 & b==1'b0)
     begin
     f3=1'b1;
   end
   else
     f3=1'b0;
   end
   always@(f2 or f3)
   begin
     if(f2==1'b0 & f3==1'b0)
     begin
     c=1'b1;
   end
   else
    c=1'b0;
   end
endmodule
module b_xnor_nor_tb;
 wire c;
 reg a,b;
 b_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_b(output reg c,s,input a,b);
  always@(a or b)
   begin
```

```
    if(a==1'b1 & b==1'b1)
    begin
    c=1'b1;
   end
      else
    c=1'b0;
    end
  always@(a or b)
   begin
    if((a==1'b1 & b==1'b0) | (a==1'b0 & b==1'b1))
    begin
    s=1'b1;
   end
   else
    s=1'b0;
   end
endmodule
module a17_b_tb;
 wire c,s;
 reg a,b;
 a17_b 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_b(output reg c_out,output reg [3:0] sum,input [3:0] a,input [3:0] b,input c_in);
  always @ (a or b or c_in) begin
    {c_out, sum} = a + b + c_in;
  end
endmodule
module a18_b_tb;
   reg [3:0] a;
   reg [3:0] b;
   reg c_in;
wire c_out;
   wire [3:0] sum;
   integer i;
   a18_b 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
```