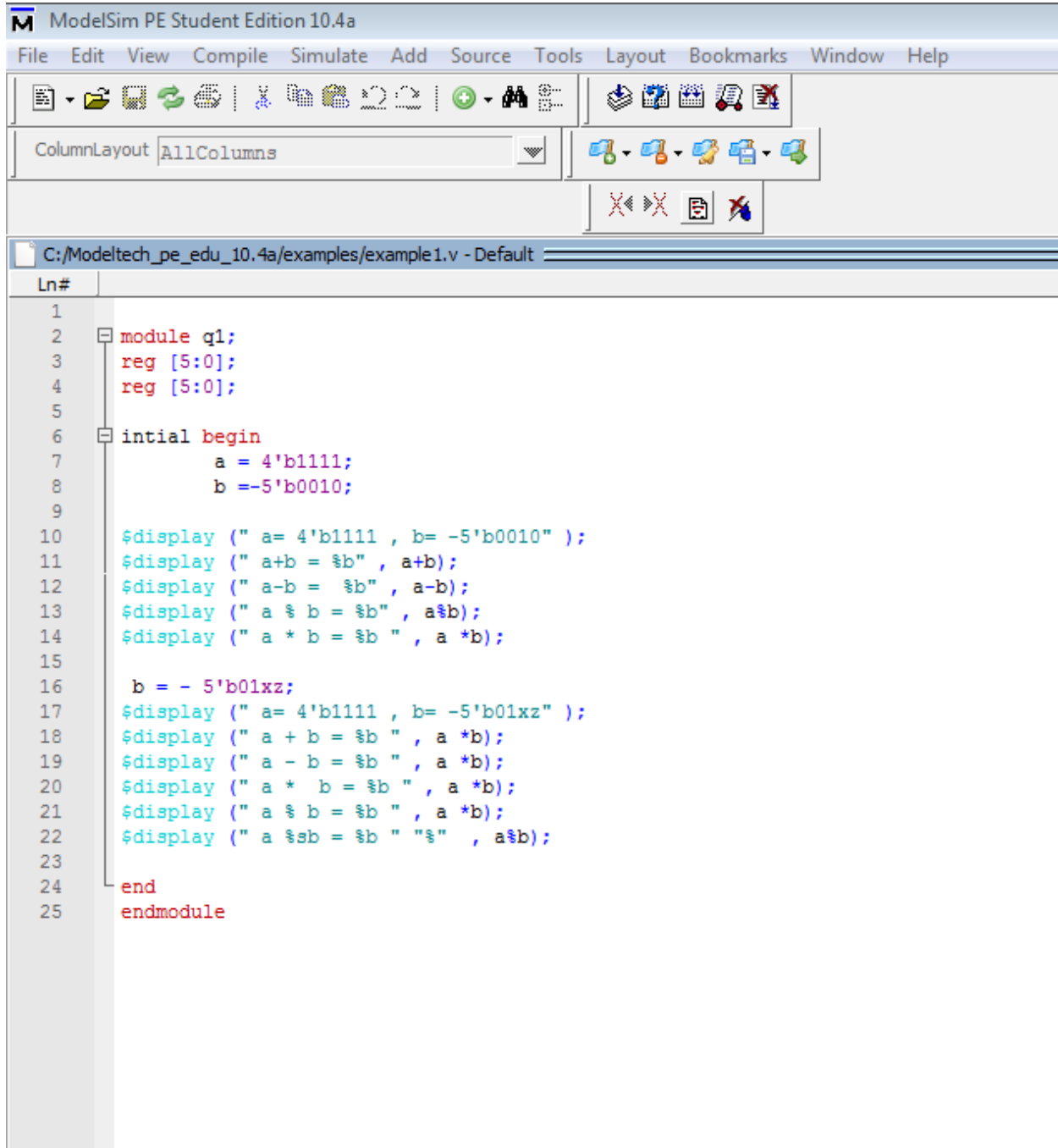


Chirag solanki

15035

Lab assignment 2

Answers 1



```
ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
ColumnLayout AllColumns
C:/Modeltech_pe_edu_10.4a/examples/example1.v - Default
Ln#
1
2 module q1;
3   reg [5:0];
4   reg [5:0];
5
6   initial begin
7       a = 4'b1111;
8       b = -5'b0010;
9
10      $display (" a= 4'b1111 , b= -5'b0010" );
11      $display (" a+b = %b" , a+b);
12      $display (" a-b = %b" , a-b);
13      $display (" a % b = %b" , a%b);
14      $display (" a * b = %b " , a *b);
15
16      b = - 5'b01xz;
17      $display (" a= 4'b1111 , b= -5'b01xz" );
18      $display (" a + b = %b " , a *b);
19      $display (" a - b = %b " , a *b);
20      $display (" a * b = %b " , a *b);
21      $display (" a % b = %b " , a *b);
22      $display (" a %sb = %b " "%b" , a%b);
23
24   end
25   endmodule
```

```

module q1;

reg [5:0];

reg [5:0];

intial begin

    a = 4'b1111;

    b = -5'b0010;


$display (" a= 4'b1111 , b= -5'b0010" );

$display (" a+b = %b" , a+b);

$display (" a-b = %b" , a-b);

$display (" a % b = %b" , a%b);

$display (" a * b = %b " , a *b);


    b = - 5'b01xz;

$display (" a= 4'b1111 , b= -5'b01xz" );

$display (" a + b = %b " , a *b);

$display (" a - b = %b " , a *b);

$display (" a * b = %b " , a *b);

$display (" a % b = %b " , a *b);

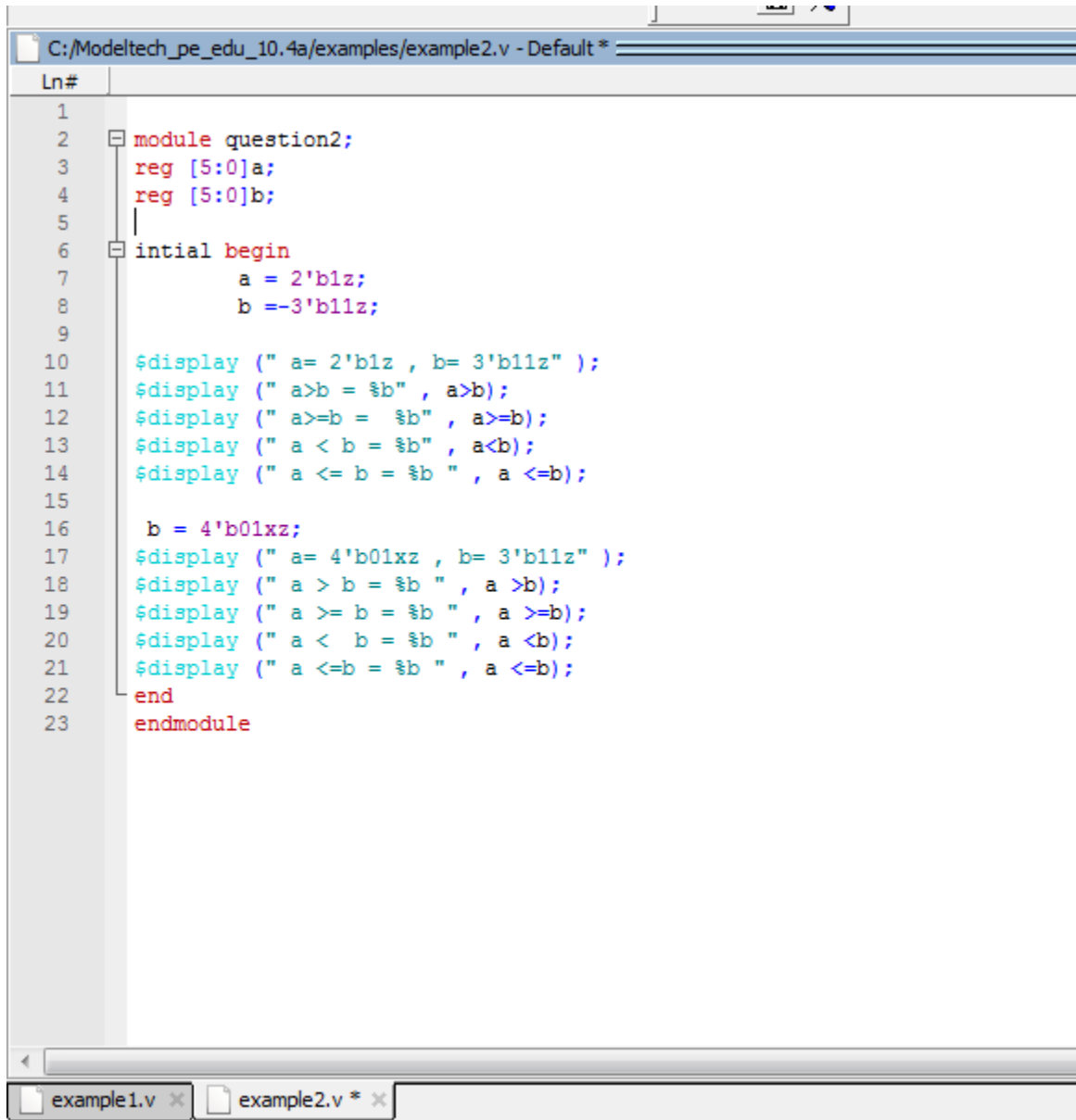
$display (" a %sb = %b " "%" , a%b);

end

endmodule

```

## Answer 2



The screenshot shows a Verilog code editor window with the title bar "C:/Modeltech\_pe\_edu\_10.4a/examples/example2.v - Default \*". The code is as follows:

```
Ln# 1
2 module question2;
3   reg [5:0]a;
4   reg [5:0]b;
5   |
6   initial begin
7       a = 2'b1z;
8       b = -3'b11z;
9
10      $display (" a= 2'b1z , b= 3'b11z" );
11      $display (" a>b = %b" , a>b);
12      $display (" a>=b = %b" , a>=b);
13      $display (" a < b = %b" , a<b);
14      $display (" a <= b = %b " , a <=b);
15
16      b = 4'b01xz;
17      $display (" a= 4'b01xz , b= 3'b11z" );
18      $display (" a > b = %b " , a >b);
19      $display (" a >= b = %b " , a >=b);
20      $display (" a < b = %b " , a <b);
21      $display (" a <= b = %b " , a <=b);
22   end
23 endmodule
```

The editor has a line number column on the left. The code is color-coded: keywords (module, initial, end, endmodule, reg) are in red, variables (a, b) are in blue, and literals and operators are in black. The initial block is indented. The code includes two sets of display statements to show the values of 'a' and 'b' and the results of various comparisons.

```
module question2;
```

```
reg [5:0]a;
```

```
reg [5:0]b;
```

```
initial begin
```

$a = 2'b1z;$

$b = 3'b11z;$

$\$display (" a = 2'b1z , b = 3'b11z" );$

$\$display (" a > b = \%b" , a > b);$

$\$display (" a >= b = \%b" , a >= b);$

$\$display (" a < b = \%b" , a < b);$

$\$display (" a <= b = \%b " , a <= b);$

$b = 4'b01xz;$

$\$display (" a = 4'b01xz , b = 3'b11z" );$

$\$display (" a > b = \%b " , a > b);$

$\$display (" a >= b = \%b " , a >= b);$

$\$display (" a < b = \%b " , a < b);$

$\$display (" a <= b = \%b " , a <= b);$

end

endmodule

### Answers 3

```
C:/Modeltech_pe_edu_10.4a/examples/example3.v - Default
Ln#
1  module qeustion3();
2
3      initial
4      begin
5          $display ("3'b11z==3'b11z=%b" , (3'b11z==3'b11z));
6          $display ("3'bx1z==3'b11z=%b" , (3'b11z===3'b11z));
7          $display ("3'b01z!==3'b01x=%b" , (3'b11z!==3'b01x));
8          $display ("3'b11z!==3'b01z=%b" , (3'b11z!===3'b01z));
9          $display ("3==5=%b" , (3==5));
10         $display ("3==3=%b" , (3==3));
11         $display ("3!=5=%b" , (3!=5));
12         $display ("3!=3=%b" , (3!=3));
13
14     end
15 endmodule
16
17
```

```
module qeustion3();
```

```
    initial
```

```
    begin
```

```
        $display ("3'b11z==3'b11z=%b" , (3'b11z==3'b11z));
```

```
        $display ("3'bx1z==3'b11z=%b" , (3'b11z===3'b11z));
```

```
$display ("3'b01z!==3'b01x=%b" , (3'b11z!==3'b01x));
```

```
$display ("3'b11z!==3'b01z=%b" , (3'b11z!==3'b01z));
```

```
$display ("3==5=%b" , (3==5));
```

```
$display ("3==3=%b" , (3==3));
```

```
$display ("3!=5=%b" , (3!=5));
```

```
$display ("3!=3=%b" , (3!=3));
```

```
end
```

```
endmodule
```

```
example 4
```

```
C:/Modeltech_pe_edu_10.4a/examples/example4.v - Default
Ln#
1
2    'timescale 10ns/1ps
3
4    module answer4;
5
6        reg [1:0]a;
7        reg [1:]b;
8        reg [2:0]c;
9
10    initial begin
11
12        a=1'bx; b=2'b1z; c=3'bxxx;
13        #1 a=1'bz; b=2'b0z; c=3'blxx;
14        #1 a=1'b1; b=2'bxz; c=3'b0xx;
15        #1 $finish;
16
17    end
18
19    initial begin
20
21        $monitor ($time , " abar =%b , bbar=%0b , cbar=%0b", !a , !b , !c );
22
23    end
24    endmodule |
```

'timescale 10ns/1ps

module answer4;

reg [1:0]a;

reg [1:]b;

reg [2:0]c;

initial begin

a=1'bx; b=2'b1z; c=3'bxxx;

#1 a=1'bz; b=2'b0z; c=3'b1xx;

#1 a=1'b1; b=2'bxz; c=3'b0xx;

#1 \$finish;

end

initial begin

\$monitor (\$time , " abar =%b , bbar=%0b ,cbar=%0b", !a , !b , !c );

end

endmodule

example 5



```
C:/Modeltech_pe_edu_10.4a/examples/example5.v - Default
Ln#
1  module answer5();
2
3  initial begin
4
5      $display ("%g 1'bx&&2'bxz =%b " , $time (1'bx&&2'bxz));
6
7      #1 $display ("%g 2'b0x||2'bxz =%b " , $time (2'b0x||1'b1z));
8
9      #1 $display ("%g 2'b001&&2'b1z=%b " , $time (2'b001&&2'b1z));
10
11     #1 $display ("%g 2'b001&&2'b1z =%b " , $time (2'b001&&2'b1z));
12
13     #1 $display ("%g 2'b0z|| 4'b01xz =%b " , $time (2'b0z|| 4'b01xz));
14
15     end
16 endmodule |
17
18
19
20
21
22
```

```
module answer5();
```

```
initial begin
```

```
$display ("%g 1'bx&&2'bxz =%b " , $time (1'bx&&2'bxz));
```

```
#1 $display ("%g 2'b0x| | 2'bxz =%b " , $time (2'b0x| | 1'b1z));
```

```
#1 $display ("%g 2'b001&&2'b1z=%b " , $time (2'b001&&2'b1z));
```

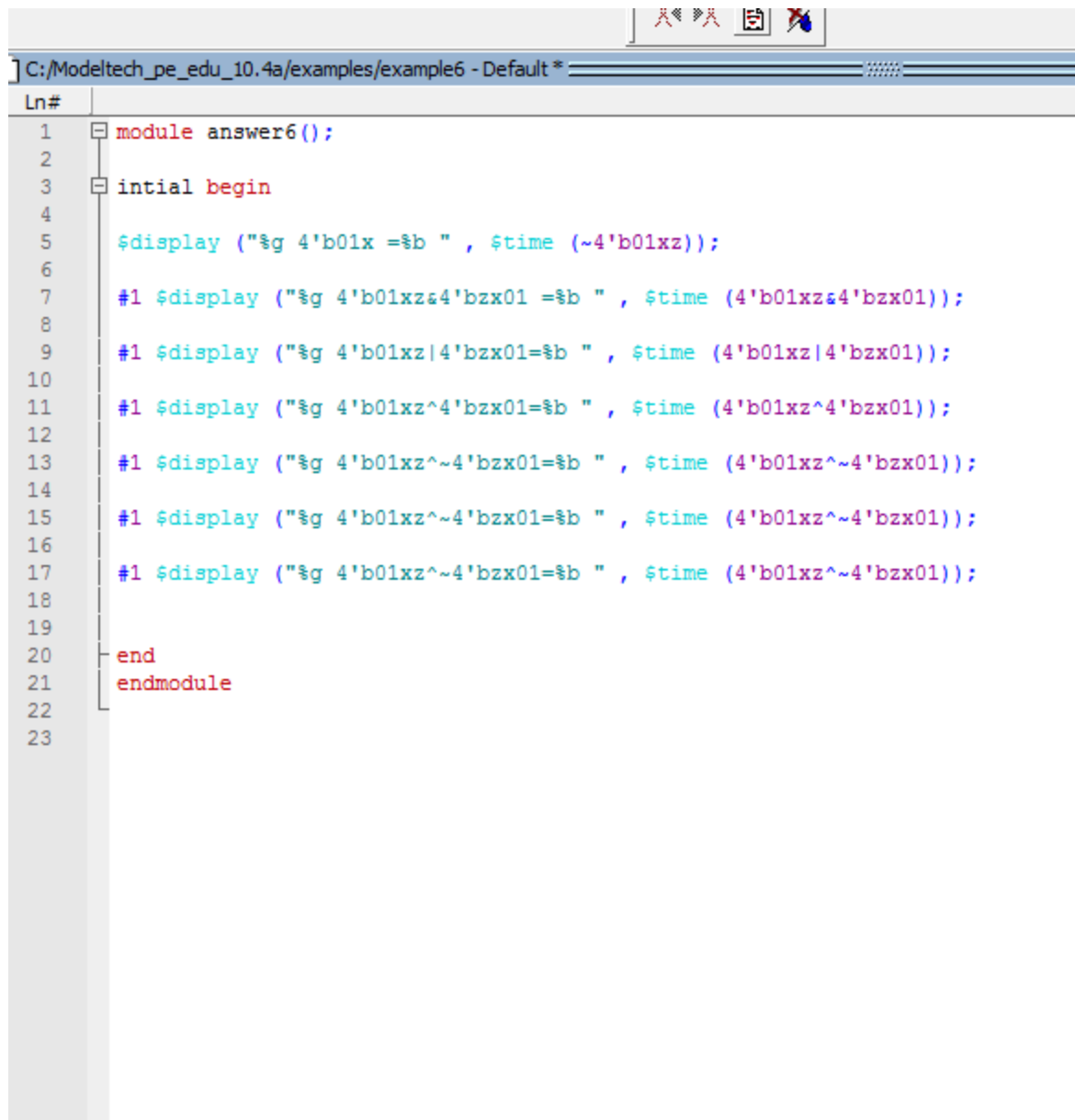
```
#1 $display ("%g 2'b001&&2'b1z =%b " , $time (2'b001&&2'b1z));
```

```
#1 $display ("%g 2'b0z| | 4'b01xz =%b " , $time (2'b0z| | 4'b01xz));
```

```
end
```

```
endmodule
```

```
answer 6
```



```
Ln# 1 module answer6();
2
3 initial begin
4
5     $display ("%g 4'b01x =%b " , $time (~4'b01xz));
6
7     #1 $display ("%g 4'b01xz&4'bzx01 =%b " , $time (4'b01xz&4'bzx01));
8
9     #1 $display ("%g 4'b01xz|4'bzx01=%b " , $time (4'b01xz|4'bzx01));
10
11    #1 $display ("%g 4'b01xz^4'bzx01=%b " , $time (4'b01xz^4'bzx01));
12
13    #1 $display ("%g 4'b01xz~4'bzx01=%b " , $time (4'b01xz~4'bzx01));
14
15    #1 $display ("%g 4'b01xz^~4'bzx01=%b " , $time (4'b01xz^~4'bzx01));
16
17    #1 $display ("%g 4'b01xz^~4'bzx01=%b " , $time (4'b01xz^~4'bzx01));
18
19
20 end
21 endmodule
22
23
```

module answer6();

initial begin

\$display ("%g 4'b01x =%b " , \$time (~4'b01xz));

```
#1 $display ("%g 4'b01xz&4'bzx01 =%b " , $time (4'b01xz&4'bzx01));
```

```
#1 $display ("%g 4'b01xz|4'bzx01=%b " , $time (4'b01xz|4'bzx01));
```

```
#1 $display ("%g 4'b01xz^4'bzx01=%b " , $time (4'b01xz^4'bzx01));
```

```
#1 $display ("%g 4'b01xz^~4'bzx01=%b " , $time (4'b01xz^~4'bzx01));
```


```
#1 $display ("%g 4'b01xz^~4'bzx01=%b " , $time (4'b01xz^~4'bzx01));
```

```
#1 $display ("%g 4'b01xz^~4'bzx01=%b " , $time (4'b01xz^~4'bzx01));
```

```
end
```

```
endmodule
```

```
answer 7
```



C:/Modeltech\_pe\_edu\_10.4a/examples/example7 - Default

Ln#	
1	<input type="checkbox"/> module answer7();
2	
3	<input type="checkbox"/> initial begin
4	
5	\$display ("%g %4'b01xz=%b", \$time (%4'b01xz));
6	
7	#1 \$display ("%g ~ 4'b01xz=%b", \$time (^ 4'b01xz));
8	#1 \$display ("%g ^4'b01xz=%b", \$time (^4'b01xz));
9	##1 \$display ("%g ~^4'b01xz=%b", \$time (~^4'b01xz));
10	
11	
12	end
13	endmodule

```
module answer7();
```

```
initial begin
```

```
$display ("%g &4'b01xz=%b", $time (&4'b01xz));
```

```
#1 $display ("%g ~|4'b01xz=%b", $time (^|4'b01xz));
```

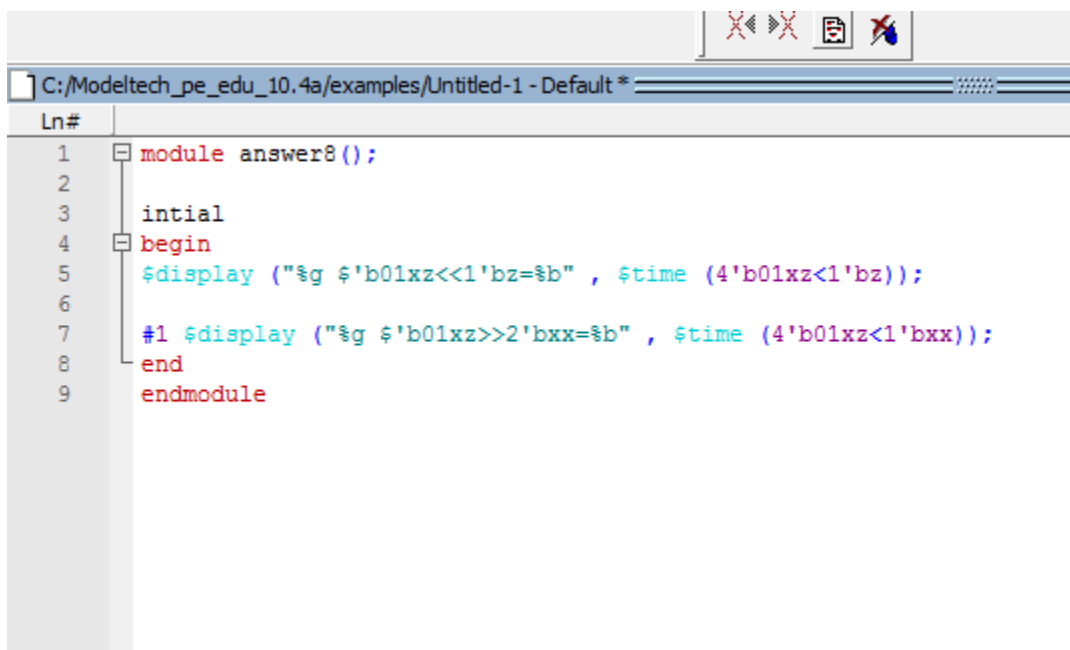
```
#1 $display ("%g ^4'b01xz=%b", $time (^4'b01xz));
```

```
##1 $display ("%g ~^4'b01xz=%b", $time (~^4'b01xz));
```

```
end
```

```
endmodule
```

```
answer 8
```



```
module answer8();  
    initial  
    begin  
        $display ("%g $'b01xz<<1'bz=%b" , $time (4'b01xz<1'bz));  
        #1 $display ("%g $'b01xz>>2'bxx=%b" , $time (4'b01xz<1'bxx));  
    end  
endmodule
```

```
module answer8();
```

```
  intial
```

```
  begin
```

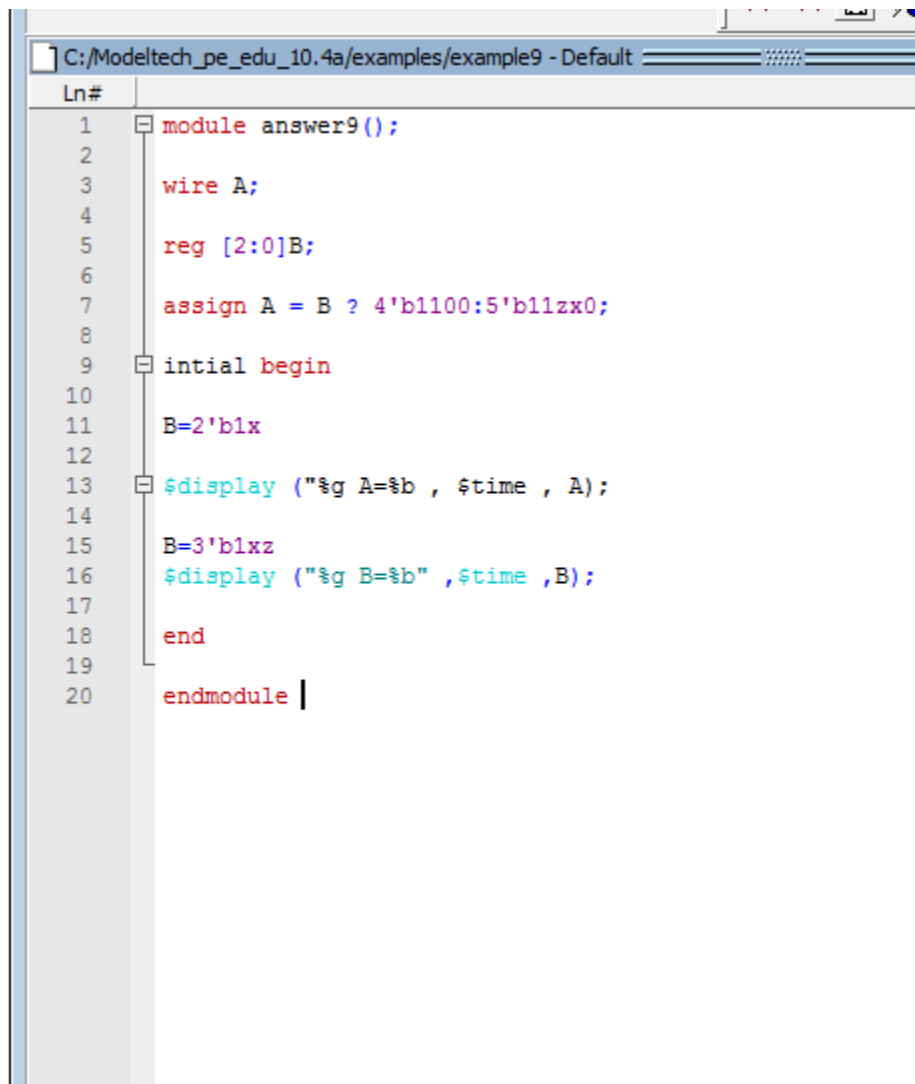
```
    $display ("%g $'b01xz<<1'bz=%b" , $time (4'b01xz<1'bz));
```

```
    #1 $display ("%g $'b01xz>>2'bxx=%b" , $time (4'b01xz<1'bxx));
```

```
  end
```

```
endmodule
```

## example 9



```
C:/Modeltech_pe_edu_10.4a/examples/example9 - Default
Ln#
1  module answer9();
2
3      wire A;
4
5      reg [2:0]B;
6
7      assign A = B ? 4'b1100:5'b11zx0;
8
9      initial begin
10
11          B=2'b1x
12
13      $display ("%g A=%b , $time , A);
14
15          B=3'b1xz
16          $display ("%g B=%b" , $time , B);
17
18      end
19
20  endmodule |
```

```
module answer9();
```

```
    wire A;
```

```
    reg [2:0]B;
```



```
assign A = B ? 4'b1100:5'b11zx0;
```

```
initial begin
```

```
B=2'b1x
```

```
$display ("%g A=%b", $time, A);
```

```
B=3'b1xz
```

```
$display ("%g B=%b", $time, B);
```

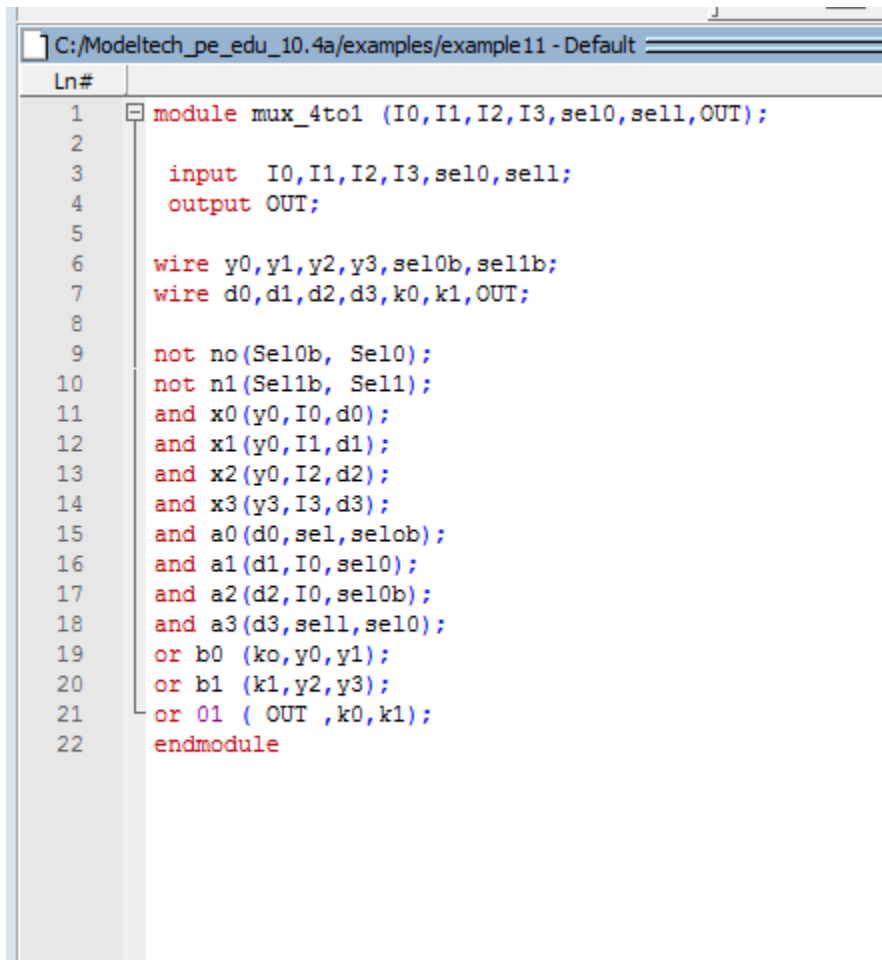
```
end
```

```
endmodule
```

example 10

```
C:/Modeltech_pe_edu_10.4a/examples/example10 - Default
Ln#
1  `timescale 10ns/100ps
2
3  module teststrength1();
4
5      reg A,B;
6      wire X;
7
8      buf (strong1,weak0) g1(X,A);
9      buf ( weak1,strong0) g2 (X,B);
10
11     initial
12     begin
13
14         A=1;
15         B=1;
16
17         $display (" X = %b ,A = %b ,B = %b " , X,A,B);
18
19     end
20 endmodule
21
22 module teststrength2();
23
24     reg i1,i2,ctrl;
25     wire X;
26
27     bufif0 (strong1,weak0) g1 (X,i1,ctrl);
28     bufif0 (strong1, weak0) g2 (X,i2,ctrl);
29
30     initial
31     begin
32         ctrl=1;
33         i1=0;
34         i1=1;
35         $display (" X =%b , i1 =%b ,ctrk =%b,ctrl=%b" , x,i1,i2,ctrl);
36
37     end
38 endmodule |
39
```

## Example 11



```
1 module mux_4to1 (I0,I1,I2,I3,sel0,sel1,OUT);
2
3     input  I0,I1,I2,I3,sel0,sel1;
4     output OUT;
5
6     wire y0,y1,y2,y3,sel0b,sel1b;
7     wire d0,d1,d2,d3,k0,k1,OUT;
8
9     not no(sel0b, sel0);
10    not n1(sel1b, sel1);
11    and x0(y0,I0,d0);
12    and x1(y0,I1,d1);
13    and x2(y0,I2,d2);
14    and x3(y3,I3,d3);
15    and a0(d0,sel,sel0b);
16    and a1(d1,I0,sel0);
17    and a2(d2,I0,sel0b);
18    and a3(d3,sel1,sel0);
19    or b0 (ko,y0,y1);
20    or b1 (k1,y2,y3);
21    or O1 ( OUT ,k0,k1);
22 endmodule
```

```
module mux_4to1 (I0,I1,I2,I3,sel0,sel1,OUT);
```

```
input I0,I1,I2,I3,sel0,sel1;
```

```
output OUT;
```

```
wire y0,y1,y2,y3,sel0b,sel1b;
```

```
wire d0,d1,d2,d3,k0,k1,OUT;
```

```

not no(sel0b, sel0);

not n1(sel1b, sel1);

and x0(y0,l0,d0);

and x1(y0,l1,d1);

and x2(y0,l2,d2);

and x3(y3,l3,d3);

and a0(d0,sel,selob);

and a1(d1,l0,sel0);

and a2(d2,l0,sel0b);

and a3(d3,sel,sel0);

or b0 (ko,y0,y1);

or b1 (k1,y2,y3);

or O1 ( OUT ,k0,k1);

endmodule

```

```

`timescale 10ns/100ps

```

```

module teststrength1();

```

```
reg A,B;
```

```
wire X;
```

```
buf (strong1,weak0) g1(X,A);
```

```
buf ( weak1,strong0) g2 (X,B);
```

```
intial
```

```
begin
```

```
A=1;
```

```
B=1;
```

```
$display (" X = %b ,A = %b ,B = %b " , X,A,B);
```

```
end
```

```
endmodule
```

```
module teststrength2();
```

```
reg i1,i2,ctrl;
```

```
wire X;
```

```
bufif0 (strong1,weak0) g1 (X,i1,ctrl);
```

```
bufifo (strong1, weak0) g2 (X,i2,ctrl);
```

```
initial
```

```
begin
```

```
    ctrl=1;
```

```
    il=0;
```

```
    il=1;
```

```
    $display (" X =%b , il =%b ,ctrk =%b,ctrl=%b" , x,i1,i2,ctrl);
```

```
end
```

```
endmodule
```

example 12

```
C:/Modeltech_pe_edu_10.4a/examples/example12 - Default
Ln#
1 module tb_mux_4to1;
2   reg I0,I1,I2,I3,sel0,sel1;
3   wire OUT;
4   mux_4to1 dut (I0,I1,I2,I3,sel0,sel1,OUT);
5
6   initial
7   begin
8
9       I0 = 0;
10      I1 = 0;
11      I3 = 0;
12      sel0=0;
13      sel1=0;
14      #5 I0 = 0;I1=0;I2=0;I3=0;
15      #5 I0 = 1;I1=0;I2=0;I3=0;
16      #5 I0 = 0;I1=1;I2=0;I3=0;
17      #5 I0 = 0;I1=0;I2=1;I3=0;
18      #5 I0 = 0;I1=0;I2=1;I3=0;
19      #5 I0 = 0;I1=1;I2=1;I3=0;
20      #5 I0 = 1;I1=1;I2=1;I3=0;
21      #10 $finish;
22
23   end
24
25   initial
26
27   $monitor ( $time , , I3,I2,I2,I0,, sel1,sel0,,OUT);
28 endmodule
```

```
module tb_mux_4to1;
```

```
reg I0,I1,I2,I3,sel0,sel1;
```

```
wire OUT;
```

```
mux_4to1 dut (I0,I1,I2,I3,sel0,sel1,OUT);
```

```
initial
```

```
begin
```

```
I0 = 0;
```

l1 = 0;

l3 = 0;

sel0=0;

sel1=0;

#5 l0 = 0;l1=0;l2=0;l3=0;

#5 l0 = 1;l1=0;l2=0;l3=0;

#5 l0 = 0;l1=1;l2=0;l3=0;

#5 l0 = 0;l1=0;l2=1;l3=0;

#5 l0 = 0;l1=0;l2=1;l3=0;

#5 l0 = 0;l1=1;l2=1;l3=0;

#5 l0 = 1;l1=1;l2=1;l3=0;

#10 \$finish;

end

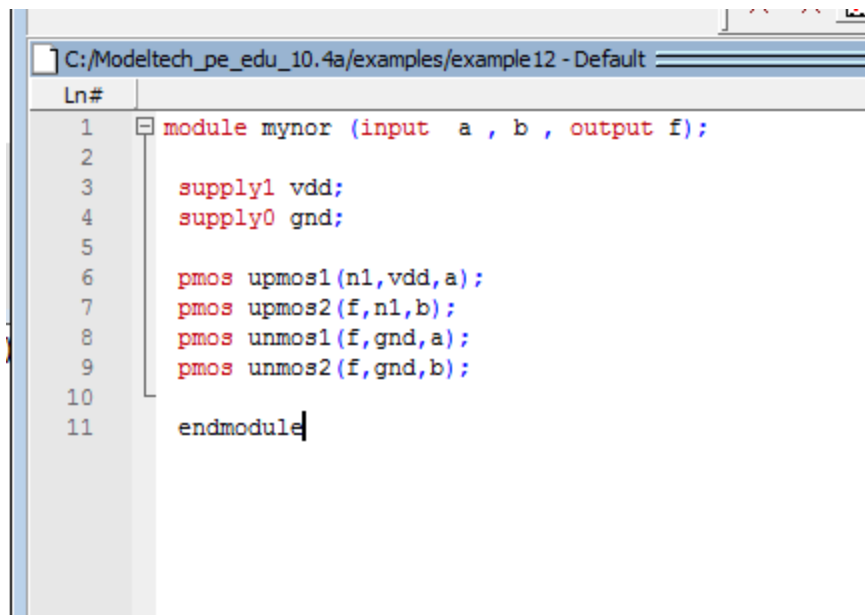
initial

\$monitor ( \$time ,, l3,l2,l2,l0,, sel,sel0,,OUT);

endmodule

example 12



A screenshot of a Verilog code editor window. The title bar shows the file path: "C:/Modeltech\_pe\_edu\_10.4a/examples/example12 - Default". The editor displays a Verilog module named "mynor" with inputs "a" and "b", and output "f". The code is as follows:

```
1 module mynor (input a , b , output f);  
2  
3     supply1 vdd;  
4     supply0 gnd;  
5  
6     pmos upmos1(n1,vdd,a);  
7     pmos upmos2(f,n1,b);  
8     pmos unmos1(f,gnd,a);  
9     pmos unmos2(f,gnd,b);  
10  
11 endmodule
```

```
module mynor (input a , b , output f);
```

```
supply1 vdd;
```

```
supply0 gnd;
```

```
pmos upmos1(n1,vdd,a);
```

```
pmos upmos2(f,n1,b);
```

```
pmos unmos1(f,gnd,a);
```

```
pmos unmos2(f,gnd,b);
```

```
endmodule
```

```
test batch :
```

```

n#
1  `include "answer12.v"
2
3  module test_12();
4
5      reg a_r, b_r;
6
7      wire f_w;
8
9      mynor dut {
10
11          .a(a_r),
12          .b(b_r),
13          .f(f_w)
14
15      };
16
17      initial begin
18
19          a_r=1'b0; b_r=1'b0;
20          #1 a_r=1'b0;b_r =1'b1;
21          #1 a_r=1'b1;b_r =1'b1;
22          #1 a_r=1'b1;b_r =1'b1;
23          #1 $finish;
24
25      end
26      initial begin
27
28          $monitor ("a=%0b,b=%0b,f_w=%0b", a_r,b_r,f_w);
29
30      end
31  endmodule \

```

`include "answer12.v"

module test\_12();

reg a\_r, b\_r;

wire f\_w;

mynor dut {

```

        .a(a_r),
        .b(b_r),
        .f(f_w)

);

initial begin

    a_r=1'b0; b_r=1'b0;
#1  a_r=1'b0;b_r =1'b1;
#1  a_r=1'b1;b_r =1'b1;
#1  a_r=1'b1;b_r =1'b1;
#1  $finish;

end

initial begin

    $monitor ("a=%0b,b=%0b,f_w=%0b", a_r,b_r,f_w);

end

endmodule \

```