ELEX 2117 : Digital Techniques 2
2021 Fall Term

Solutions to Quiz 1

Question 1

Write the module definition for a module named

(a) **counter** with two single-bit **logic** inputs named **reset** and **enable**, and a 16-bit **logic** output named **count**.

(b) **detect** with a 12-bit **logic** input named **level**, a one-bit input named **enable** and a one-bit **logic** output named **alarm**.

Multi-bit inputs should be declared with the bit indices in decreasing order.

Question 2

Write an **assign** statement that sets the signal

(a) **counter_next** to the signal **counter** plus the signal **incr**.

(b) **value_next** to the signal **value** minus the signal **decr**.

Question 3

Write the Verilog for a

(a) 12-bit numeric literal whose value is 16 (in decimal notation). Use a hexadecimal or binary base for the value.

(b) 10-bit numeric literal whose value is 8 (in decimal notation). Use a hexadecimal or binary base for the value.

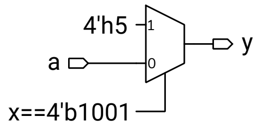
Answers for Questions 1 – 3

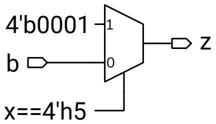
The **endmodule** in Question 1, the module declaration for Question 2 and the **module** and signal declarations and the **display** statements for Questions 2 and 3 are included so the solutions can be checked, they did not need to be included in your answer.

```
// Question 1

module counter
  (
    input logic reset, enable,
    output logic [15:0] count
    ) ;
endmodule

module detect
  (
    input logic [11:0] level,
    input logic enable,
    output logic alarm
    ) ;
endmodule

// Question 2

module q2 ;
   logic [7:0] counter=1, counter_next, incr=1 ;
   logic [7:0] value=1, value_next, decr=1 ;

   assign counter_next = counter + incr ;

   assign value_next = value - decr ;

   initial #0 $display(counter_next, value_next) ;

endmodule

// Question 3

module q3 ;

   logic [11:0] a = 12'h10, b = 12'b10000 ;

   logic [ 9:0] x = 10'h8, y = 10'b1000 ;

   initial $display(a,b,x,y) ;

endmodule
```

Question 4

Write a Verilog statement corresponding to the following block diagram. Assume all signals have been declared.

1. 

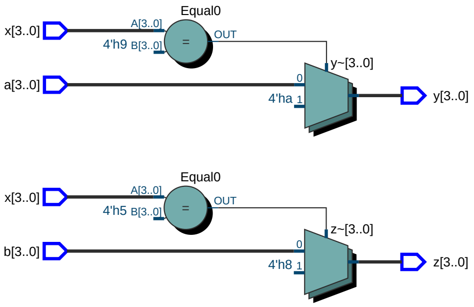
2. 

Answer

The module declarations are included so the solutions can be synthesized, they did not need to be included in your answer.

```
module q4a
  ( output logic [3:0] y,
    input  logic [3:0] a, x ) ;

   assign y = x == 4'b1001 ? 4'h5 : a ;

endmodule

module q4b
  ( output logic [3:0] z,
    input  logic [3:0] b, x ) ;

   assign z = x == 4'h5 ? 4'b0001 : b ;

endmodule
```

Synthesis Results


Equal0
x[3..0]
A[3..0]
4'h9
B[3..0]
=
OUT
y~[3..0]
a[3..0]
0
4'ha
1
y[3..0]
Equal0
x[3..0]
A[3..0]
4'h5
B[3..0]
=
OUT
z~[3..0]
b[3..0]
0
4'h8
1
z[3..0]