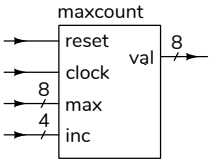
Solutions to Midterm 2

There were two versions of most questions. The questions and answers for both versions are given below.

Question 1

Write a Verilog module named **maxcount** with logic inputs named **reset** and **clock**, an 8-bit logic array input named **max**, a 4-bit logic array input named **inc**, and an 8-bit logic array output named **val**.



The output only changes on the rising edge of **clock**. If **reset** is asserted or **val** is equal to **max** then then **val** is set to zero, otherwise **val** is incremented by **inc**.

or:

The output only changes on the rising edge of **clock**. If **reset** is asserted or **val** is equal to zero then **val** is set to **max**, otherwise **val** is decremented by **inc**.

Follow the course coding guidelines but omit comments.

Answer

```
module maxcount
   ( input logic reset, clock,
     input logic [7:0] max,
     input logic [3:0] inc,
     output logic [7:0] val ) ;

`ifdef NOTDEF

   always_ff @(posedge clock) val
      <= reset ? '0 :
         val == max ? '0 :
         val + 1'b1 ;

`else

   always_ff @(posedge clock) val
      <= reset ? max :
         val == '0 ? max :
         val - 1'b1 ;

`endif

endmodule
```

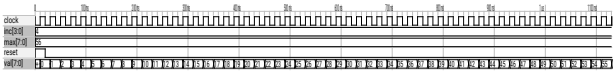
Question 2

Write a Verilog testbench module named **maxcount_tb** that:

- declares any necessary variables
- instantiates the module **maxcount** described in the previous question
- sets **max** to a value equal to the last two digits of your BCIT ID and **inc** to 2 (or 4)
- creates a continuous 50 (or 10) MHz clock, starting with a low (0) value
- asserts reset for exactly one clock cycle
- terminates (rather than pauses) the simulation when **val** is equal to **max** (or **0**)

Answer

```
// solution for ID = A00123456

module maxcount_tb ;

   logic reset, clock ;
   logic [7:0] max ;
   logic [3:0] inc ;
   logic [7:0] val ;

   maxcount m0 (.*) ;

   initial begin
      $dumpfile("maxcount_tb.vcd") ; //not needed
      $dumpvars() ;                  //not needed
      reset = '1 ;
      clock = '0 ;
      max = 56 ;
      inc = 2 ;                      // or inc = 4
      inc = 4 ;
      @(negedge clock) reset = '0 ;
      wait ( val == max ) ; // or: wait(val == 0);
      $finish ;
   end

   always #10ns clock = ~clock ; // or: always #50ns ...

endmodule
```

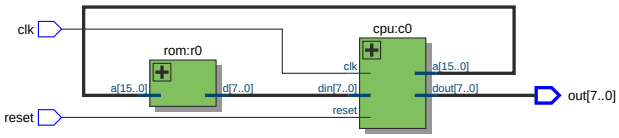


Question 3

Given the following module declarations:

```
module rom #(parameter b, w)
   ( input logic [w-1:0] a,
     output logic [b-1:0] d ) ;

module cpu #(parameter b, w)
   ( input logic clk, reset,
     input logic [b-1:0] din,
     output logic [b-1:0] dout,
     output logic [w-1:0] a ) ;
```

Write a Verilog module named **system** that implements the following diagram:



Declare any additional signals required to implement the diagram, using any valid signal names. Use a value of 8 for the parameter **b** and a value of 16 for the parameter **w** in each instantiation. Use explicit port and parameter names (e.g. **.port(signal)**, **.port**, or **#(.parameter(*n*))**).

Answer

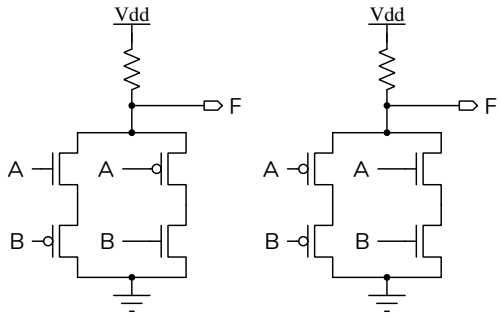
```
module rom #(parameter b, w)
   ( input logic [w-1:0] a,
     output logic [b-1:0] d ) ;

   assign d = a ? '1 : '0 ;

endmodule

module cpu #(parameter b, w)
   ( input logic clk, reset,
     input logic [b-1:0] din,
     output logic [b-1:0] dout,
     output logic [w-1:0] a ) ;

   always_ff @(posedge clk) a
      <= reset ? '0 :
         a + 1 ;

   assign dout = din ;

endmodule

module system
   ( input logic clk, reset,
     output logic [7:0] out ) ;

   logic [7:0] romd ;
   logic [15:0] a ;

   cpu #(.b(8),.w(16)) c0
      (.reset, .clk, .din(romd), .dout(out), .a ) ;

   rom #(.b(8),.w(16)) r0 (.a, .d(romd) ) ;

endmodule
```

Question 4

(a) The following table shows the voltages on inputs A and B for the circuit shown below. H corresponds to a voltage above the gate threshold voltage and L corresponds to 0 V.

Fill in the last column of the table with the voltage at point F of the circuit. Use H for a voltage of approximately V_{dd} and L for a voltage of approximately 0 V.

A	B	F
L	L	
L	H	
H	L	
H	H	



Answer

In the first circuit the output is low when A is high and B is low or when A is low and B is high.

A	B	F
L	L	H
L	H	L
H	L	L
H	H	H

In the second circuit the output is low when A and B are both high or when they are both low.

A	B	F
L	L	L
L	H	H
H	L	H
H	H	L

(b) An IC has specifications of $V_{\texttt{IL}} = 0.3\,\text{V}$ and $V_{\texttt{IH}} = 0.7\,\text{V}$. If you needed a noise margin of 0.2 V for both high and low voltages, what are the required maximum $\text{V}_{\texttt{OL}}$ and minimum $\text{V}_{\texttt{OH}}$?

Answer

Using the equations for noise margin:

$$\text{noise margin(low)} = V_{\texttt{IL(max)}} - V_{\texttt{OL(max)}}$$

$$\text{noise margin(high)} = V_{\texttt{OH(min)}} - V_{\texttt{IH(min)}}$$

and solving for $\text{V}_{\texttt{OL}}$ and $\text{V}_{\texttt{OH}}$:

$V_{\texttt{OL(max)}} = V_{\texttt{IL}}$−noise margin = 0.3−0.2 = $\boxed{0.1\,\text{V}}$

$V_{\texttt{OH(min)}} = V_{\texttt{IH}}$+noise margin = 0.7+0.2 = $\boxed{0.9\,\text{V}}$

(c) A circuit draws 50 (or 200) mA when operating with a clock frequency of 50 MHz. You would like to operate this circuit for a year when powered by a CR2016 battery that has a capacity of 100 mAh. What clock frequency should you use? *Hints: $P = IV$. V does not change. There are $365 \times 24 = 8760$ hours in a year.*

Answer

For a 100 mAh capacity battery to last 8760 hours, the current must be $^{100\,\text{mAh}}/_{8760\,\text{h}} = 11.4\,\mu\text{A}$.

Using the formula for power consumption ratio versus frequency and voltage, substituting $P_i = I_i V$, and solving for f_2 with $I_1 = 50\,\text{mA}$ (or 200 mA), $I_2 = 11.4\,\mu\text{A}$ and $f_1 = 50\,\text{MHz}$, $f_2 = {}^{I_2}/_{I_1} \cdot f_1 = \boxed{11.416\,\text{kHz}}$ (or $\boxed{2.854\,\text{kHz}}$).