ELEX 2117 : Digital Techniques 2
2022 Winter Term

Solutions to Final Exam

-

Question 1

Write a Verilog module named **swap** with two 8-bit inputs named **b0** and **b1**, a one-bit input named **big** (or **little**), and a 16-bit output named **w**. The output **w** is set as follows:

- when **big** (or **little**) is high, **w** is set to a 16-bit value with **b1** (or **b0**) as the most significant byte and **b0** (or **b1**) as the least significant byte.
- when **big** (or **little**) is low, **w** is set to a 16-bit value with **b0** (or **b1**) as the most significant byte and **b1** (or **b0**) as the least significant byte.

Follow the mandatory course coding guidelines except that you need not include comments.

Solution

The following solution and testbench:

```
module swap
  ( input logic [7:0] b0, b1,
    input logic big,
    output logic [15:0] w ) ;

   assign w = big ? {b1,b0} : {b0,b1} ;
   // assign w = little ? {b0,b1} : {b1,b0} ;

endmodule

module swap_tb ;

   logic [7:0] b0, b1;
   logic big ;
   logic [15:0] w ;

   swap s0 (.*) ;

   initial begin
      {b0,b1,big} = {8'd1,8'd2,1'b1} ;
      #0 $display("%x",w) ;
      {b0,b1,big} = {8'd1,8'd2,1'b0} ;
      #0 $display("%x",w) ;
   end

endmodule
```

prints out:

```
# 0201
# 0102
```

Question 2

A state machine has a one-bit **step** input, a one-bit **reset** input, a **clock** input, and an 8-bit **value** output. The value of **value** changes only on the rising edge of **clock**. If **reset** is asserted, **value** is set to **8'hff** (or **8'h00**). Otherwise if **step** is asserted **value** is decremented (or incremented) by 3 (or 5). Otherwise **value** is held at the same value. You do not need to consider a possible under (or over) flow of **value**.

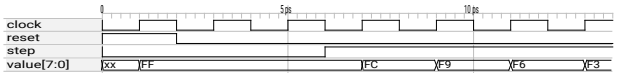
Write a Verilog module named **stepper** that implements this state machine. Follow the mandatory course coding guidelines except that you need not include comments.

Solution

The following solution and testbench:

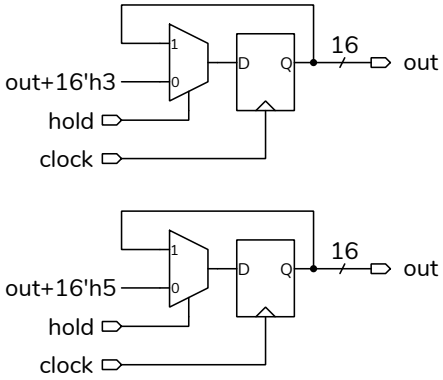
```
module stepper
  ( input logic step, reset, clock,
    output logic [7:0] value ) ;

   always_ff @(posedge clock) value
     <= reset ? 8'hff :
        step ? value - 8'd3 :
        value ;

`ifdef false
   always_ff @(posedge clock) value
     <= reset ? 8'h00 :
        step ? value + 8'd5 :
        value ;
`endif

endmodule

module stepper_tb ;

   logic step, reset, clock ;
   logic [7:0] value ;

   stepper s0 (.*) ;

   initial begin
      $dumpfile("stepper.vcd") ;
      $dumpvars() ;
      {step,reset,clock} = 3'b010 ;
      #2 reset = '0 ;
      #4 step = '1 ;
      #10 $finish ;
   end

   always #1 clock = !clock ;

endmodule
```

produces the following waveforms:



Question 3

Write a Verilog module corresponding to the diagram shown at right. **hold** and **clock** are one-bit inputs and **out** is a 16-bit output. **out** in the expression is the same signal as the output. Follow the mandatory course coding guidelines except that you need not include comments.

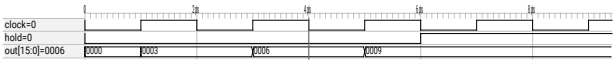


Solution

The following solution and testbench:

```
module q3
  ( input logic hold, clock,
    output logic [15:0] out ) ;

   always @(posedge clock)
     out <= hold ? out : out+16'h3 ;

endmodule

module q3_tb ;

   logic hold, clock ;
   logic [15:0] out;

   q3 q0 (.*) ;

   initial begin
      $dumpfile("q3.vcd") ;
      $dumpvars() ;
      {hold,clock} = 2'b00 ;
      #6 hold = '1 ;
      #8 $finish ;
   end

   always #1 clock = !clock ;

endmodule
```

produces the following waveforms if **out** has an initial value of 0 at the start of the simulation:



Question 4

Given the following Verilog declarations:

```
logic [15:0] x = 16'd90 ;
logic  [3:0] y = 4'b111 ;
```

or

```
logic [11:0] x = 16'd45 ;
logic  [3:0] y = 4'b101 ;
```

fill in in the table below with the value of the each expression as a hexadecimal (base 16) number (Verilog format not necessary) and the length (number of bits) as a decimal (base 10) number...

Solution

The operators in the table are: concatenation, bit-slicing (most significant nybble), bitwise AND, subtraction, and the conditional operator. The solutions are given in Tables 1 and 2.

Question 5

Fill in the blank boxes in the table below so that all values in each row are consistent (agree with each other). The first row is an example.

Solution

Tables 3 and 4 show the solution.

Question 6

The waveform at right is measured on an asynchronous serial interface. Tic marks on the time axis mark bit periods. The bit order and levels follow the standard for asynchronous serial interfaces. An 8-bit value was transferred. The diagram shows the complete transmission of one byte, including any additional bits required for synchronization.

...

```
logic [15:0] x = 16'd90 // 16'h005a or 16'b101_1010 ;
logic  [3:0] y = 4'b111 //  4'h7 or 4'd7;
```

expression	value (in hexadecimal)	length (in bits, decimal)
{x,y}	5a7	20
x[7:4]	5	4
x & y	2	16
x - y	53	16
x[0] ? y : ~y	8	4

Figure 1: Solution for Question 4.

```
logic [11:0] x = 16'd45 // 12'h02d or 12'b10_1101
logic  [3:0] y = 4'b101 //  4'h5  or 4'd5
```

expression	value (in hexadecimal)	length (in bits, decimal)
{x,y}	2d5	16 (or 20)
x[7:4]	2	4
x & y	5	12 (or 16)
x - y	28	12 (or 16)
x[0] ? y : ~y	5	4

Figure 2: Solution for Question 4.

signal name	truth value (T/F)	logic level (H/L)	Verilog value as output (0/1)	Verilog value in expression (0/1)
$\overline{\texttt{hot}}$	T	L	0	1
running	T	H	1	1
odd*	F	H	1	0
$\overline{\texttt{failed}}$	T	L	0	1

Figure 3: Solution for Question 5.

signal name	truth value (T/F)	logic level (H/L)	Verilog value as output (0/1)	Verilog value in expression (0/1)
$\overline{\texttt{hot}}$	T	L	0	1
`running`	F	L	0	0
`odd*`	T	L	0	1
$\overline{\texttt{failed}}$	F	H	1	0

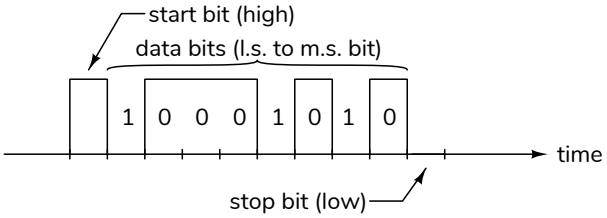
Figure 4: Solution for Question 5

What value was transmitted over the interface? Give your answer as a Verilog constant with the correct width and a hexadecimal base. Show how you obtained your answer.

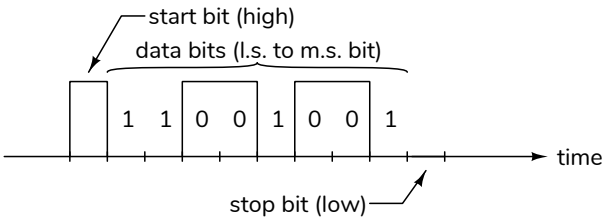
Solution

The following figures label the start and stop bits and the data bits which are transmitted least-significant-bit first and with a low value representing a '1' bit.

For the first version of the question the bits are 0101 0001 or 8'h51:



For the second version of the question the bits are 1001 0011 or 8'h93:



Question 7

A sequential logic circuit uses registers with a minimum setup time requirement of 4 (or 2) ns and a maximum clock-to-output delay of 4 (or 3) ns. It needs to run at a clock rate of 50 (or 40) MHz. What is the maximum allowable propagation delay through any combinational logic path in the circuit? Show your work and include units in your answer.

Solution

Using

$$t_{SU}\text{ (avail)} = T_{clock}\text{ - }t_{CO}\text{ (max) - }t_{PD}\text{ (max)}$$

and that at zero slack (maximum clock rate), t_{SU} (avail) = t_{SU} (required) we can solve for t_{PD}:

$$t_{PD}\text{ (max)} = T_{clock}\text{ - }t_{CO}\text{ (max) - }t_{SU}\text{ (required)}$$

$$= \frac{1}{50\times10^6} - 4 - 4 = 12\text{ ns (or } \frac{1}{40\times10^6} - 3 - 2 = 20\text{ ns)}$$

Question 8

You need to design a 256 (or 128) kByte memory for a 16-bit CPU using 64 (or 32) k × 4 memory ICs.

(a) How many address bits does each memory IC have?

(b) How many IC's will be needed for each bank?

(c) How many banks will be required?

(d) If an address decoder places the first byte of this memory at address `0x10 0000`, what is the last address in this memory?

Solution

(a) How many address bits does each memory IC have?

$n = \log_2(64\text{k}) = 16$ or $\log_2(32\text{k}) = 15$

(b) How many IC's will be needed for each bank?

$W/w = 16/4 = 4$

(c) How many banks will be required?

256k/(16/8)/64k = 2 banks or 128 k/(16/8)/32k = 2

(d) If an address decoder places the first byte of this memory at address **`0x10 0000`**, what is the last address in this memory?

0x10_0000+256×2^{10}-1 = 0x14_0000-1 =

0x13_FFFF (1,310,719)

or

0x10_0000+128×2^{10}-1 = 0x12_0000-1 =

0x11_FFFF = (1,179,647).

Question 9

You are choosing the logic levels for a new design and need to maintain a noise margin of 1.5 (or 1) V for both high and low logic levels. $V_{IL(max)}$ is 2 V and $V_{IH\ (min)}$ is 3 V. What are $V_{OL(max)}$ and $V_{OH(min)}$?

Solution

For low logic levels the noise margin is $V_{IL(max)} - V_{OL(max)} = 1.5\ V$ so $V_{OL(max)} = V_{IL\ (max)} - 1.5 = 2 - 1.5 = 0.5\ V$ (or 1 V).

For high logic levels the noise margin is $V_{OH(min)} - V_{IH(min)} = 1.5\ V$. so $V_{OH(min)} = V_{IH\ (min)} + 1.5 = 3 + 1.5 = 4.5\ V$ (or 4 V).

Question 10

A CMOS digital logic circuit consumes 100 mW at a supply voltage of 5 V and a clock frequency of 10 (or 5) MHz. The clock frequency is reduced to 1 MHz and the supply voltage to 3 V. What is the new power consumption?

Solution

$$\frac{P_2}{P_1} = \frac{f_2}{f_1} \cdot \left(\frac{V_2}{V_1}\right)^2$$

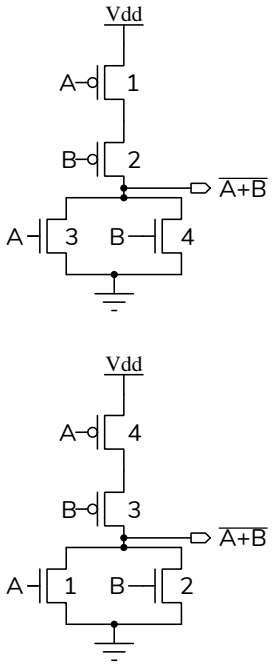
with $P_1 = 100$ mW, f_1=10 MHz, $f_2 = 1$ MHz, $V_1 = 5$ V and $V_2 = 3$ V,

$$\frac{P_2}{=} 100\left(\frac{1}{10}\right) \cdot \left(\frac{3}{5}\right)^2 = 3.6 \text{ mW}$$

(or 7.2 mW).

Question 11

The schematic at right shows a CMOS NOR gate with the four transistors numbered from 1 to 4. Both A inputs are at the same logic level. Both B inputs are at the same logic level. Which transistors are on (conducting) when the output is high?



Solution

For the output to be high, both transistors connecting the output to V_{dd} must be on and neither of the transistors connecting the output to ground may be on. In the first schematic transistors 1 and 2 are on. In the second schematic transistors 3 and 4 are on.