ELEX 7660 : Digital System Design
Term 201710

FINAL EXAMINATION
10:30 AM – 1:20 PM
April 21, 2017

This exam has five (5) questions on nine (9) pages. The marks for each question are as indicated. There are a total of 32 marks. Answer all questions. Write your answers and all rough work in this paper and nowhere else. Show your work. Books and notes are allowed. No electronic devices other than calculators are allowed.

This exam paper is for:

Exam 1 A00123456

Each exam is equally difficult.
Answer your own exam.

Do not start until you are told to do so.

Name: ________________________

BCIT ID: ________________________

Signature: ________________________

Question	Mark	Max.
1		7
2		4
3		6
4		7
5		8
Total		28

Question 1 (7 marks)

The following declarations and initializations appear in a System Verilog module:

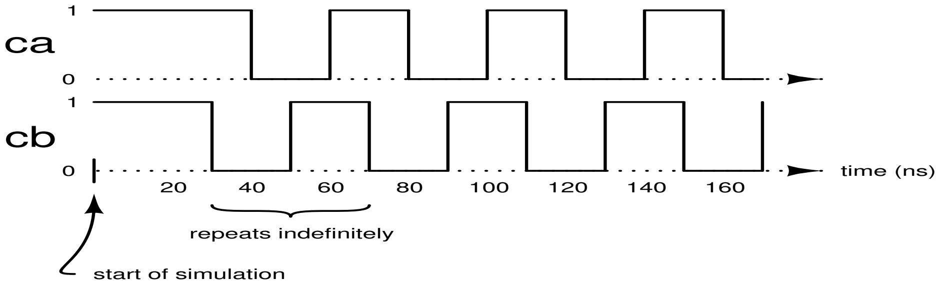
```
logic a = '1 ;
logic [7:0] b  = 8'ha6 ;
logic [1:0][3:0] c  = '{ 4'b0000 , 4'b1001 } ;
```

For each row in the following table, write in the size (in bits) and the value (in binary) of the expression in the first column. If the size is more than 8 bits, you need only show the least-significant 8 bits.

expression	size (bits)	value (binary, 8 or fewer l.s. bits)
`b ==? 8'bxxxx0110`		
`signed'(b) >>> 1`		
`{b[7:1],a}`		
`b[7:4]`		
`{2{c}}`		
`b+1'b1`		
`a ? b : c`		

Question 2 (4 marks)

The diagram below shows two waveforms that need to be generated to test a design. The waveforms are constant for the first 30 or 40 ns and periodic after that.

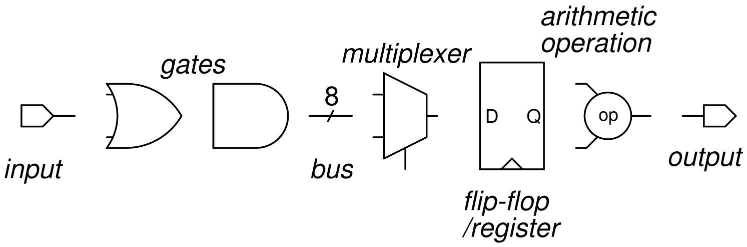


The following code includes signal declarations and an initial block. Write the System Verilog code that, if placed within the initial block, would generate these waveforms indefinitely. You may use an additional initial block.

```
module examtb ;
  logic ca, cb ;

  initial begin
  // your answer is the code that would go here.
  end

endmodule
```

Question 3 (6 marks)

Using the following symbols:



draw a schematic for the following Verilog module on the facing page. Label the inputs and outputs. Indicate the bus width if it's more than 1 bit.

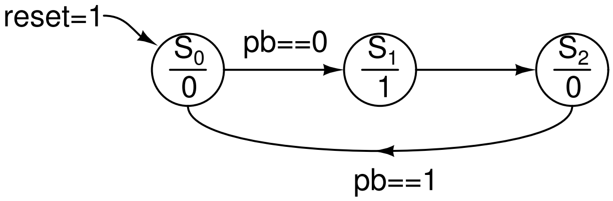
```
module mksum
  ( output logic [15:0] c,
    input logic reset, clk ) ;

   logic [15:0] a, a_next, b, b_next ;

   always@(posedge clk) begin
      a <= a_next ;
      b <= b_next ;
      c <= a_next + b_next ;
   end


   always_comb begin
      if ( reset ) begin
         a_next = '0 ;
         b_next = '0 ;
      end else begin
         a_next = a + 16'd113 ;
         b_next = b + 16'd127 ;
      end
   end

endmodule
```

Question 4 (7 marks)

A System Verilog module is defined as follows:

```
module pulsegen (
       input logic pb,
       output logic pulse,
       input logic clk, reset ) ;

       typedef enum logic [1:0] { S0, S1, S2 } state_t ;
       state_t state, next_state ;

       logic pulse_next ;

       // your code goes here

endmodule ;
```

The state transition diagram below describes the operation of this module:



The circles show the states and the value of the pb output in each state. Input conditions not shown result in no change of state. The transition from state S2 to state S3 is unconditional.

The diagram shows that when the reset input is high (1) the pulse output should go to low (0) regardless of the current state. In state S0 if the pb input goes low (0) the output should go high (1) for one clock cycle and then go low (0). The output should then stay low until the next falling edge of pb.

This is a Moore state machine with registered outputs. State transitions only happen on the rising edge of the clock.

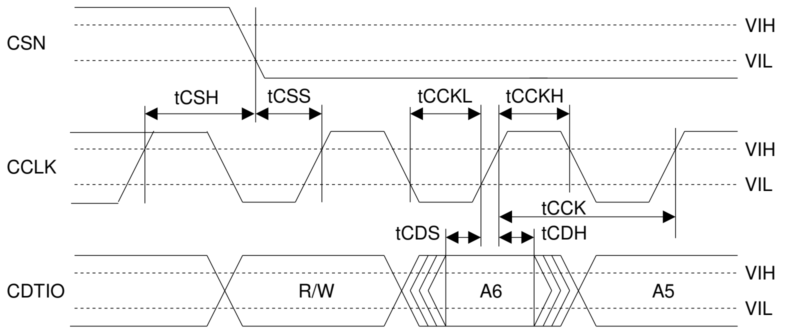
On the facing page write the missing System Verilog code that would be required to implement this module. Use the signal names defined above.

Question 5 (8 marks)

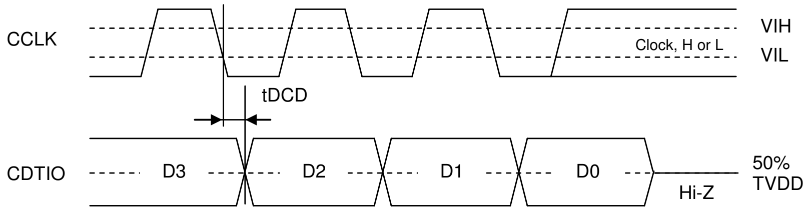
The following timing specifications are found in the datasheet for an audio codec IC:

Control Interface Timing (3-wire Serial: AK4951EG)					
CCLK Period	tCCK	200	-	-	ns
CCLK Pulse Width Low	tCCKL	80	-	-	ns
Pulse Width High	tCCKH	80	-	-	ns
CDTIO Setup Time	tCDS	40	-	-	ns
CDTIO Hold Time	tCDH	40	-	-	ns
CSN "H" Time	tCSW	150	-	-	ns
CSN Edge to CCLK "↑"	tCSS	50	-	-	ns
CCLK "↑" to CSN Edge	tCSH	50	-	-	ns
CCLK "↓" to CDTIO (at Read Command)	tDCD	-	-	70	ns
CSN "↑" to CDTIO (Hi-Z) (at Read Command)	tCCZ	-	-	70	ns

The timing diagram for a read cycle is:



The timing diagram for a write cycle is:



The signal directions are:

Signal	Direction
CSN	input
CCLK	input
CDTIO	input (during write cycles)
CDTIO	output (during read cycles)

(a) Determine if each of the following specifications is a requirement or a "guaranteed response":

specification	Requirement(R) or Guaranteed(G)
tCCK	
tCDS	
tCDH	
tDCD	

(b) What is the maximum interface clock (CCLK) *frequency*?

(c) Note that during a write cycle the rising edge of CCLK is used to clock the data on CDTIO into the IC[1]. Assuming a CCLK frequency of 1 MHz (1 μs period) what is the *maximum* allowed delay (time) between the rising edge of CCLK and CDTIO being valid? (Show your work).

(d) What is the minimum delay between the rising edge of CCLK and CDTIO being invalid?

For (c) and (d) you may assume VIH is equal to VIL (so that CCLK crosses VIH and VIL at the same time).

[1] The "chevrons" on the CDTIO waveform indicate invalid data.

– This page intentionally left blank. –