

First name \_\_\_\_\_ **LAST NAME** \_\_\_\_\_

**LAB Session (Circle one):**

Mon 2-5

Tues 2 – 5

Wed 2-5

Thurs 11 - 2

**MIDTERM 2 (Version 1):**

1. Write your name above. **Circle your lab session!**  
**Circle the wrong session or Not circle it, 1 point will be deducted from total score.**
2. **One-side A4-sized note sheet is allowed.**
3. **No Calculator and electronic device** (computer, smart/cell phone, ...) **is allowed**
4. To receive credit, show all your work clearly and readably. Write only answer gets 0
5. There are 6 pages in this exam.
6. Do not write multiple answers, otherwise, the wrong one will be graded.
7. Ask if you have any questions or confusion.

Problem	Score	Your score
1	30	
2	25	
3	45	
Total	100	

1. (30 points)

**1a) (12 pts)** Convert the given decimal numbers to **7-bit two's complement binary numbers** and add them. Then indicate whether the sum overflows or not.

Overflow? (circle one) and explain your reason below:      **Yes**      No

$$(-37) + (-31)$$

Decimal:      -37 + -31 = -68

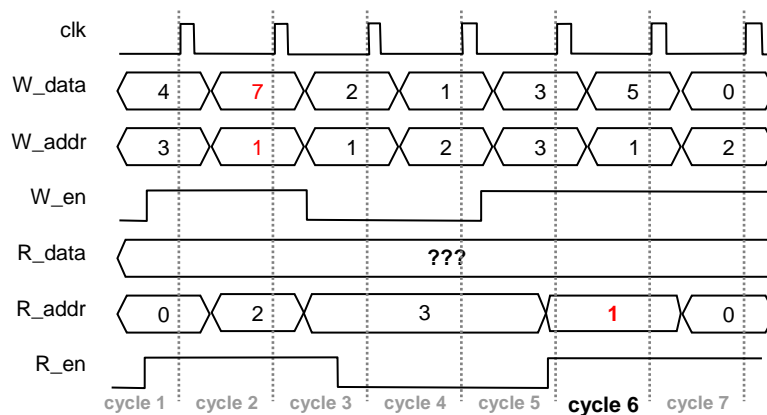
Binary      **0**  
              1011011+  
              1100001  
              **1**0111100

The **two carry-out bits** are different => overflow

Alternative explanation: 0111100 = 4 + 8 + 16 + 32 = 60 (not equal to -68)

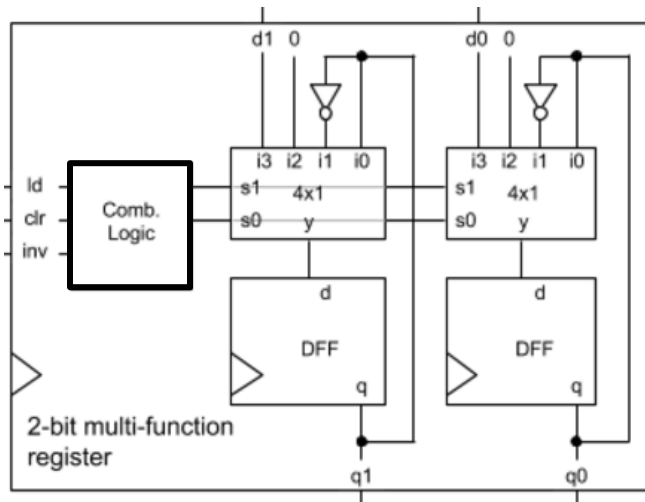
**1b) (8 pts)** Consider the register file timing diagram shown below, answer the questions.

1b.1) What is the value of **R\_data** in **cycle 6**?            **7**      



1b.2) If this is a **16x16** register file, how many bits for W\_addr and R\_addr?            **4**       bits

**1c) (10 pts)** Given the following circuit and incomplete functional table



clr	ld	inv	s1	s0	Operation
1	x	x	1	0	Clear (q1q0 = 00)
0	1	x	1	1	Parallel Load
0	0	1	0	1	Invert bits
0	0	0	0	0	Maintain

x – don't care

1c.1 (4 pts) Complete (write values of s1, s0) in the functional table above

1c.2) (6 pts) Find combinational logic circuit

**s1** ld inv

clr	00	01	11	10
0			1	1
1	1	1	1	1

$$s1 = ld + clr$$

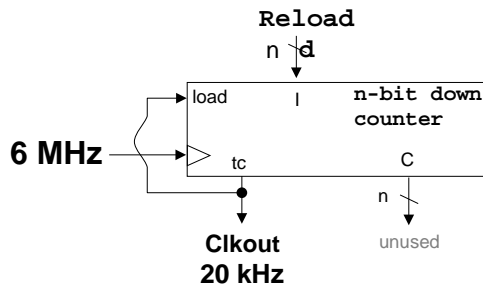
**s0** ld inv

clr	00	01	11	10
0		1	1	1
1				

$$s0 = clr' ld + clr' inv$$

## 2) (25 pts) Counter and its application

**2a) (13 pts)** Given the following circuit, if the frequency of **Clkout** is **20 kHz**,



2a.1) What **value** should be used for **Reload**?

**299**

2a.2) What is the *minimum* value of **n** (minimum of bit-width needed for the counter to work as specified)?

$$6M/20k = 6e6/2e4 = 3e2 = 300$$

$$299 = 256 + 43 = 256 + 32 + 8 + 2 + 1 \\ = 100101011$$

⇒ **9 bits**

**2b) (12 pts)** Given the following Verilog code for ClkDiv, the **Clk input is connected to a 100M Hz (100\*10<sup>6</sup> Hz)** – pin E3 on the Nexys 4 board used in the lab, answer the following questions.

Note: The period (or cycle) of 100 MHz is 0.01us = 0.01\*10<sup>-6</sup> sec

```
`timescale 1ns / 1ps
module ClkDiv(Clk, Rst, ClkOut);
  input Clk, Rst;
  output reg ClkOut;

  parameter DivVal = 250000;
  reg [18:0] DivCnt;
  reg ClkInt;

  always @(posedge Clk) begin
    if( Rst == 1 )begin
      DivCnt <= 0;
      ClkOut <= 0;
      ClkInt <= 0;
    end
    else begin
      if( DivCnt == DivVal ) begin
        ClkOut <= ~ClkInt;
        ClkInt <= ~ClkInt;
        DivCnt <= 0;
      end
      else begin
        ClkOut <= ClkInt;
        ClkInt <= ClkInt;
        DivCnt <= DivCnt + 1;
      end
    end
  end
endmodule
```

2b.1) What is the **frequency (in Hz)** of the **ClkOut** signal?

ClkOut has \_\_\_\_\_ Hz

$$100M / (2*2.5e5) = 100e6/5e5 \\ = 200$$

2b.2) If we want to create a **ClkOut signal with frequency of 20k Hz**, what is the correct value of DivVal?

Note: Clk input has 100MHz frequency.

$$100M/20k = 100e6/2e4 = 5000$$

$$DivVal = 5000/2 \Rightarrow 2500 \text{ (2499 to be exact)}$$

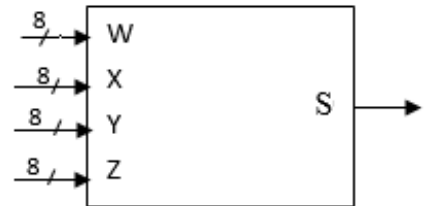
**READ ME: For problem 3, choose 1 problem to complete**(Pick one: Problem **3a** or **3b**)

10 bonus points if you can complete both 3a and 3b with 100% correct answers

3a (45 points) Design a circuit that has four 8-bit unsigned inputs and 1-bit output S that is

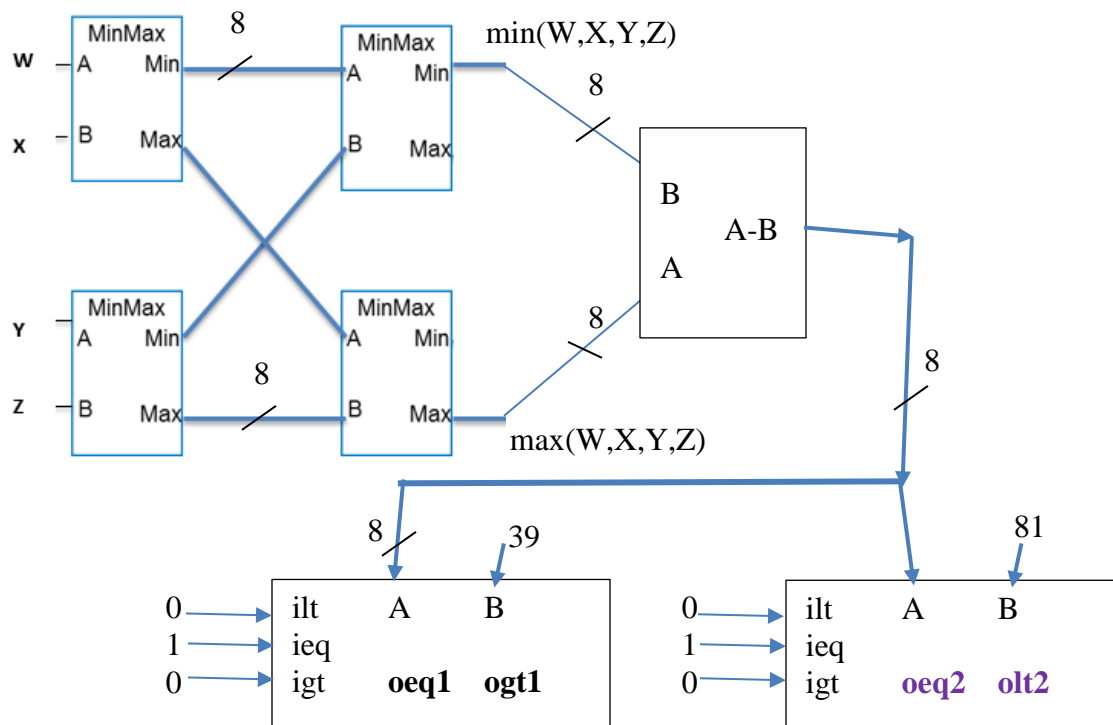
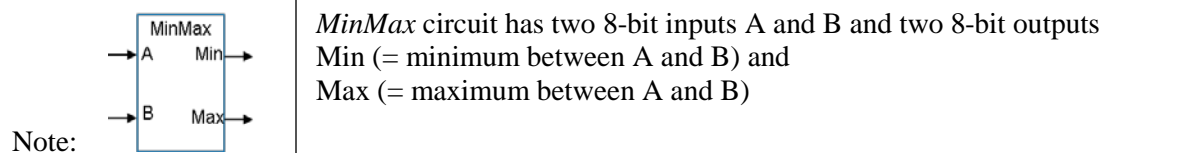
$S = 1$  if

$\max(W,X,Y,Z) - \min(W,X,Y,Z)$  is between 39 and 81 (inclusive),  
otherwise  $S = 0$ .



Clearly connect all components, indicate bit-width of each wire, write sufficient details in every component in your circuit.

You have the following datapath components available (unsigned only) – decoders, encoders, muxes (any size), 8-bit adder, 8-bit subtractor, 8-bit magnitude comparator, 8-bit MinMax shown below, and any logic gates (and, or, not) with any input numbers.



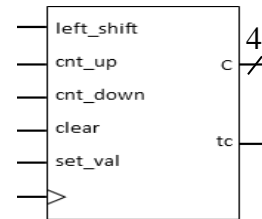
$$S = (\text{oeql} \mid \text{ogt1}) \& (\text{oeq2} \mid \text{olt2})$$

**READ ME: For problem 3, choose 1 problem to complete**(Pick one: Problem **3a** or **3b**)

10 bonus points if you can complete both 3a and 3b with 100% correct answers

3b) (45 points) Design a 4-bit up/down-counter that has the following control inputs:

- clear resets the count to 0,
- set\_val sets the counter to 7,
- left\_shift shifts a counter value to the left by up to 3 bits,
- cnt\_up enables counting up, and
- cnt\_down enables counting down



The counter has 4-bit “C” as the counter output and 1 bit terminal count (tc) output. Control inputs and counter outputs are shown in the block diagram. The **order of operation from highest to lowest priority** is clear, set\_val, left\_shift, cnt\_up, and cnt\_down. When all control inputs are 0, the circuit maintain its content.

When the operation mode is left\_shift, the amount of shift is determined by the state of the cnt\_up and cnt\_down as shown in the table. For example, if left\_shift = 1, cnt\_up cnt\_down = 10 (2 in decimal), the counter value is shifted to the left by 2 bits

left_shift	cnt_up	cnt_down	shift amount
1	0	0	0
1	0	1	1
1	1	0	2
1	1	1	3

Clearly connect all components, indicate bit-width of each wire, write sufficient details in every component in your circuit.

You have **ONLY** the following datapath components available (unsigned only):

decoders, encoders, muxes (any size), 4-bit parallel load register, 4-bit adder, 4-bit subtractor, 4-bit magnitude comparator, the 2-bit left shifter with enable (use symbol  $\ll 2$ , if enable = 1, it left-shifts the 4-bit input by two bits, else passes its input to its output without shifting), the 1-bit left shifter with enable ( $\ll 1$ ), and logic gates. Note that there is no 1-bit register available.

clear	set_val	left_shift	cnt_up	cnt_down	C
1	x	x	x	x	0
0	1	x	x	x	0111
0	0	1	0	0	Maintain
0	0	1	0	1	$C \ll 1$
0	0	1	1	0	$C \ll 2$
0	0	1	1	1	$C \ll 3$
0	0	0	1	x	$C + 1$
0	0	0	0	1	$C - 1$
0	0	0	0	0	Maintain (this is what I choose since the problem did not specify)

clear - implemented using clr control signal of a 4-bit register

