CS2102 02 Digital Logic Design Midterm 2 (1:20pm-3:10pm, May 21th)

Name: _____ **ID:** _____ **(Return this with your answer sheet)**

Truth or False (30%):

(Write down T or F. You don't need to give the reason.)

1. ___T___ An n:1 binary-select multiplexer can be implemented by using an m:n encoder and an n:1 multiplexer.
2. ___T___ Priority encoder converts the one-hot output of the arbiter to its binary value. Therefore priority encoder can be made of an arbiter and a decoder.
3. ___T___ ROM is a look-up table. It accepts an address as input, and outputs the value stored in the table at that address.
4. ___F___ The addition of a positive number and a negative number, both in 2's complement form, will neither produce an overflow, nor a carry-out.
5. ___T___ The n-bit signed 2's complement system can represent 2^{n-1} different negative numbers.
6. ___F___ The controlling value of NAND gate is logic 1; the controlling value of XOR gate is logic 0.
7. ___T___ The hold time of a flip-flop defines the time that data input must be maintained at a constant value prior to the application of the positive clock pulse.
8. ___T___ State table or state diagram provides exactly the same information of an FSM, only with different representations.
9. ___T___ In a Moore machine, the outputs are functions of the present state only; the next-state combinational logic, on the other hand, is a function of both the present state and primary inputs.
10. ___T___ To implement a ROM of 256 addresses using a 2-D array of 32 rows and 8 columns, we need a 5:32 decoder and an 8:1 MUX to select the target value.

Answer the following questions: (70%)

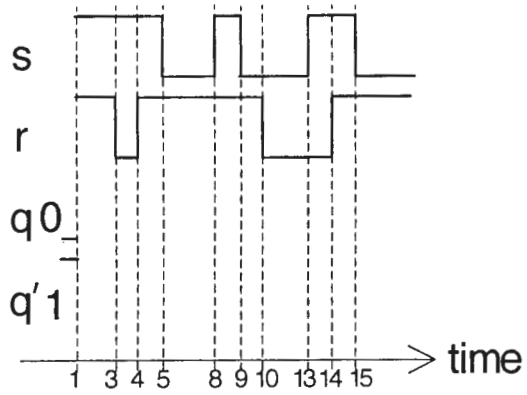
(Write down your intermediate results. Do not give the final answer only.)

1. (10%) **[Conversion]** Number conversions:
   (a) (4%) Convert the $(615.3)_8$ to hexadecimal.
   (b) (4%) Convert $(25.1875)_{10}$ to octal number.
   (c) (2%) Represent –42 in the 8-bit signed-magnitude representation.
2. (6%) **[2's Complement]** Show the detailed procedure of the following additions with 7-bit binary numbers, using 2's complement for negative numbers. You should also translate the result to decimal, and indicate how you determine if there is an overflow in hardware (instead of your brain!).
   (a) (3%) (–13) + (–49);
   (b) (3%) (+28) – (+49).
3. (8%) **[MUX]** Design a 3-input majority function $f(c, b, a)$ by using three 2:1 MUXes. The output $f = 1$ only if two or more inputs are 1.

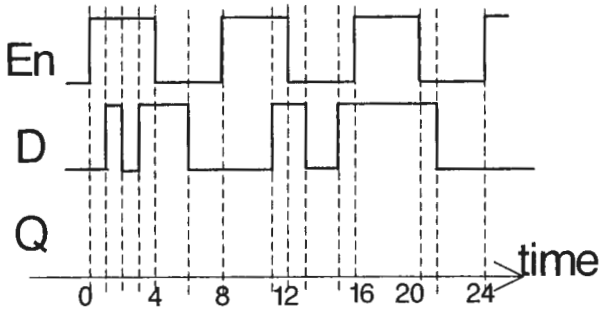
4. (6%) **[Add/Subtract]** Draw the logic diagram of a 4-bit 2's complement add/subtract unit, using full adders and two-input gates as basic blocks. The primary inputs are $\{X, A, B\}$, where $A = \{a_3, a_2, a_1, a_0\}$ and $B = \{b_3, b_2, b_1, b_0\}$. The primary outputs are $\{O, S\}$, where $S = \{s_3, s_2, s_1, s_0\}$. When $X = 0$, $S = A + B$. When $X = 1$, $S = A - B$. The output O indicates overflow. Please also label the carry signals, i.e., $\{c_4, c_3, c_2, c_1, c_0\}$ on the diagram.

5. (16%) **[Latch]**

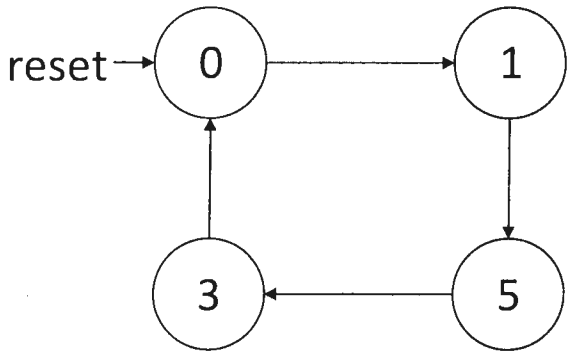
   (a) (8%) Complete the following timing diagram for an RS latch with NAND gates. Assume q begins at 0 (q' begins at 1) at $t = 1$. Ignore the delay of the RS latch (i.e., the delay of the two NAND gates is zero).



   (b) (8%) Complete the following timing diagram for a gated D latch with NAND gates. Assume Q begins at 0. Again, ignore the delay of the D latch.



6. (12%) **[FSM]** Design a 3-bit counter with D flip-flops that repeats the following sequence after reset: 0, 1, 5, 3, 0, 1, 5, 3, ... Therefore the state diagram is a loop of 4 states, as shown in the figure below. Draw the state table and design the logic diagram. After the logic optimization (using K-map) of the FSM, draw the resultant state diagram with all the 8 states specifically.



7. (14%) **[sequential design]** Design a gray-code up-down counter. The counter has one input, Dir. The outputs include the 2-bit state and a Carry_Out. The initial state is 00 after reset. When Dir=1, the circuit up-counts from 00 to 01, to 11, to 10, then back to 00, and repeats. The output Carry_Out=1 when the state transits from 10 to 00. When Dir=0, the counter down-counts from 00 to 10, to 11 to 01, back to 00, and repeats. The output Carry_Out=1 when the state transits from 00 to 10.
   (a) (2%) Is it Moore machine or Mealy machine?
   (b) (6%) Draw the state diagram and its state table.
   (c) (6%) Derive the logic to compute the next-state and output. Design the gate-level diagram of the FSM.

Good Luck and Happy Testing!!

If you have too much time left, there is always a joke for you:

One day, a mechanical engineer, an electrical engineer, and a computer engineer were driving down the street in the same car when it broke down.

The mechanical engineer said, "I think the engine broke. We have to fix it."

The electrical engineer said, "I think there was a spark and something's wrong with the electrical system. We have to fix it."

Both of them turned to the computer engineer and said, "What do you think?"

The computer engineer said, "I have no idea what happened. But why don't we close all the windows, and then open the windows again. That always works for me!!"