EECS 150 Fall 2000 2nd Midterm

Professor: R. H. Katz

INSTRUCTIONS – READ THEM NOW! All work is to be done on these pages. Partial credit is given only if we can evaluate your approach: indicate your assumptions and write as neatly as possible. Points are assigned to problems based on our estimate of how long they should take – 1 point equals 1 minute. PACE YOURSELF ACCORDINGLY: it is better to get partial credit on all of the problems than to complete a handful of them. This is a closed book examination. You will need a calculator or other information appliance. You may use a *single* 8.5" by 11" piece of paper (both sides) with prepared notes. Write your name and student ID Number at the top of each examination page.

Question	Points Assigned	Points Obtained
1	10	
2	10	
3	20	
4	20	
5	20	
Total	80	

Question 1. *Flip-flop Behavior* (10 points)
Consider a D-type storage element implemented in five different ways:

(a) D-latch (i.e., D wired to the S-input and D' wired to the R-input of an R-S latch);
(b) Clock enabled D-latch;
(c) Master-Slave Clock Enabled D-Flip-flop;
(d) Positive Edge-triggered Flip-flop;
(e) Negative Edge-triggered Flip-flop;

Complete the following timing charts indicating the behavior of these alternate storage elements. You can ignore set-up and hold time limitations (assume all constraints are meant):
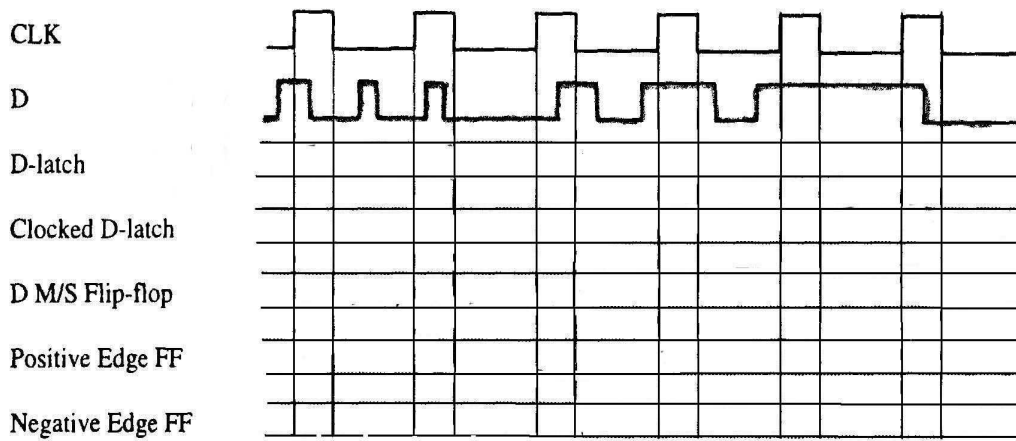
Question 2. *Ultimate Shift Register Design* (10 points)

Design the Utlimate N-bit Shift Register, with high order bit Q_{n-1} and low order bit Q_0, to the following specification. The register has a synchronous LD input that when asserted causes an external L_i bit to be loaded into the ith storage element. A synchronous Set input and a synchronous Reset input allow all storage elements to be set to a 1 or 0 respectively. The Shift operators are defined as follows.

Shift Right (SR): Q_{i+1} ---> Q_i; 0 ---> Q_{n-1}

Shift Left (SL): Q_i ---> Q_{i+1} ; 0 ---> Q_0

Rotate Right (RR): Q_{i+1} ---> Q_i ; Q_0 ---> Q_{n-1}

Rotate Left(RL): Q_i ---> Q_{i+1}; Q_{n-1} ---> Q_0

When none of the control signals are asserted, the register holds its current stat. Internally, the register is made up of conventional edge-triggered D flip-flops with nothing but a D input, a clock, and the Q output. Complete the logic schematic below to implement the shift register. Please draw neatly! HINT: Design general building block that you can personalize and resuse for each register bit.
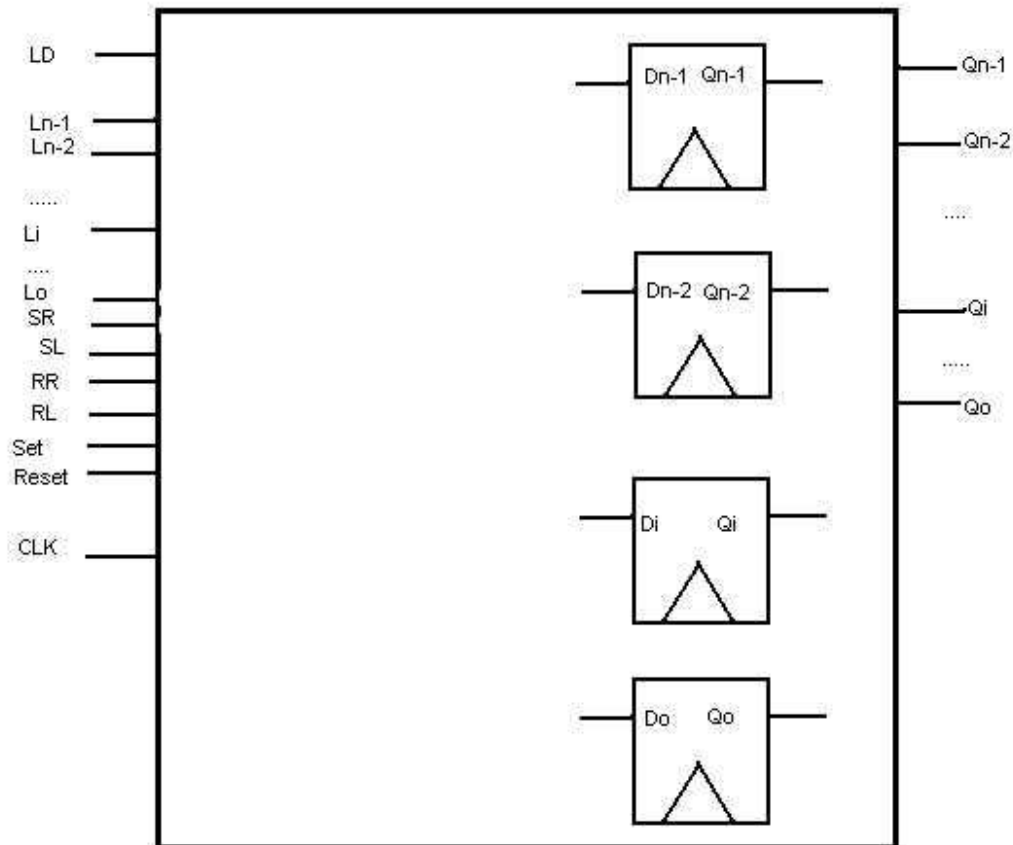
Question 3. *State Machine Design* (20 Points)

Consider the following state machine specification. The machine is binary stream sequence detector. It continuously outputs a 1 as long as there has been a (non-zero) even number of ones in the input stream since the last reset. It stops outputting one when the machine has detected the sequence 0101 in the input stream. The following illustrates the input/output behavior of the machine:

Input: 0 0 1 1 0 1 1 0 1 0 1 1 1 1 1

Output: 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0

Design a MEALY MACHINE for this specification in the boxes below.

Question 4. *State Machine Minimization* (20 Points)

Minimize the following six state Moore Machine using the Implication Chart Method.

Present State	Next State		Output
	In=0	In=1	
S0	S5	S1	0
S1	S0	S3	0
S2	S5	S1	1
S3	S0	S4	1
S4	S5	S2	0
S5	S0	S4	0
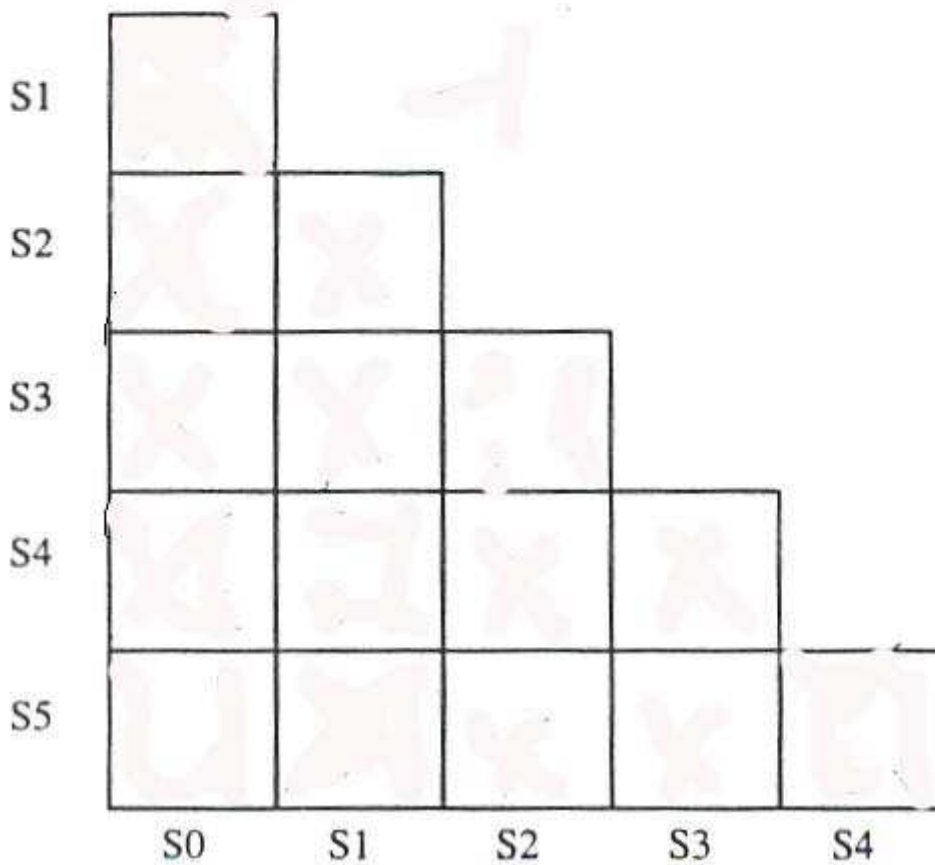
Draw a properly labeled and minimized State Diagram below:

Question 5. *State Machine Timing* (20 Points)

You are to implement a SYNCHRONOUS MEALY MACHINE *state diagram frament* the does the followin. It reads two values from a register file, adds them together, and writes the results back into the register file. The register file can read one value or write one value during any clock period, but not both (this is a so-called single port read/single port write register file). REG_A and REG_B are multi-bit read register addresses and REG_C a is the write register address. RD and WR control synchronous reading from and writing to the register file respectively. The register file operates in a single cycle. There are two buffer register with LD control inputs outside the register file. These drive an adder/subtractor circuit, the output of which writes back to the register file.

The datapath fragment is the following:

Draw the state diagram fragment below, annotating the transitions with the asserted control signals to implement the read/operate/write sequence: