ECE 120 Third Midterm Exam
Spring 2016

Tuesday, April 19, 2016

Name: ____________________________________ NetID: ________________

Discussion Section:

9:00 AM	[ ] AB1	
10:00 AM	[ ] AB2	
11:00 AM	[ ] AB3	
12:00 PM	[ ] AB4	
1:00 PM	[ ] AB5	[ ] ABA
2:00 PM	[ ] AB6	
3:00 PM	[ ] AB7	[ ] ABB
4:00 PM	[ ] AB8	[ ] ABC
5:00 PM	[ ] AB9	[ ] ABD

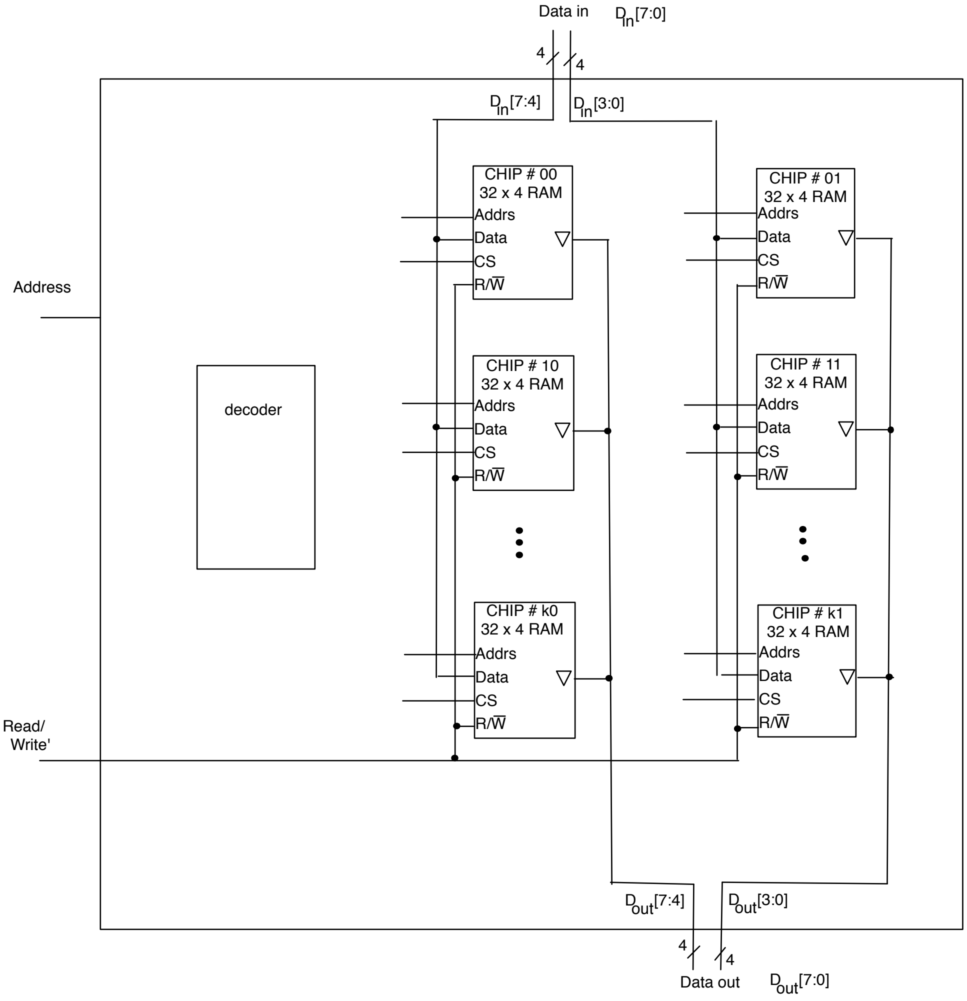
- **Be sure that your exam booklet has 8 pages.**
- **Write your name, netid and check discussion section on the title page.**
- **Do not tear the exam booklet apart.**
- **Use backs of pages for scratch work if needed.**
- **This is a closed book exam. You may not use a calculator.**
- **You are allowed one handwritten 8.5 x 11" sheet of notes (both sides).**
- **Absolutely no interaction between students is allowed.**
- **Clearly indicate any assumptions that you make.**
- **The questions are not weighted equally. Budget your time accordingly.**
- **Show your work.**

Problem 1	19 points	____________
Problem 2	18 points	____________
Problem 3	22 points	____________
Problem 4	15 points	____________
Problem 5	26 points	____________
Total	100 points	____________

Problem 1 (19 points): Memory

1. (12 points) Complete the logic diagram below for this 256 x 8 RAM constructed from 32 x 4 RAMs. **Clearly label all wires and components**. Specifically:

   a. Give the size of the decoder and label its inputs and outputs.
   b. Draw and label the address lines. *E.g. use A[3:0] notation*.
   c. Draw the CS input lines to the 32 x 4 RAMs.



2. (7 points) Consider a 32 x 32 RAM constructed using 16 x 8 RAM chips. **Note**: This part uses different RAMs than was used in part 1.

   How many 16 x 8 RAM chip(s) are needed? ________________

   Specify the signal widths (number of bits) for each of the following external signals to the 32 x 32 RAM.

   Data-In = ____________ Address = ____________ R/W' = ____________

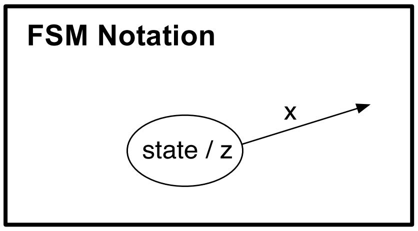
Problem 2 (18 points): FSM Design

In this problem you will implement a **0101** sequence recognizer. The circuit has one input ***x***, one output ***z***, and the output is 1 if and only if the pattern **0101** has been detected in the input stream.

Example:

Input:	***x*** =	0	0	1	0	1	0	1	0	1	0	0	1	0	1	1	0	1	. . .
Output:	***z*** =	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	. . .

1. (10 points) Draw the *Moore* state diagram, using as few states as possible. Label the states A, B, C, etc. and give the meaning of each state.

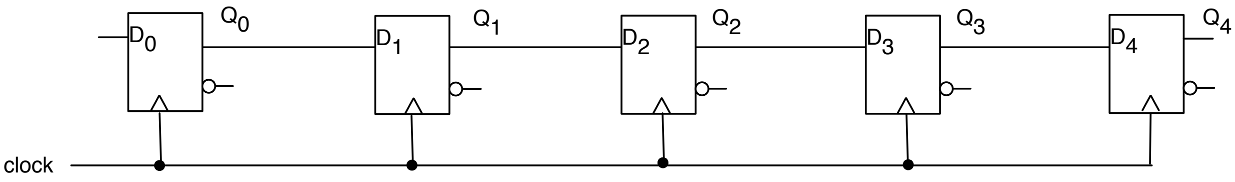

FSM Notation

State	**Meaning**
A	

2. (3 points) What is the **minimum** number of flip-flops needed to implement your circuit from part 1?

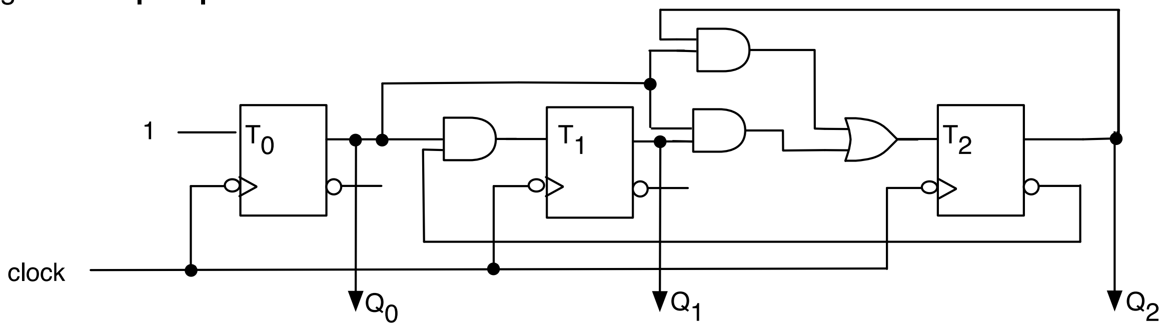
Answer ____________________

3. (5 points) Shown below is a 5-bit **shift register**, constructed with 5 positive-edge-triggered D flip-flops. Use this shift register and **only one gate** to implement a circuit which recognizes 0101 *just like the example at the top of the page*. Be sure to label input ***x*** and output ***z***.



Problem 3 (22 points): FSM Circuit Analysis

The circuit below shows a 3-bit synchronous counter constructed using 3 negative-edge-triggered **T flip-flops**.



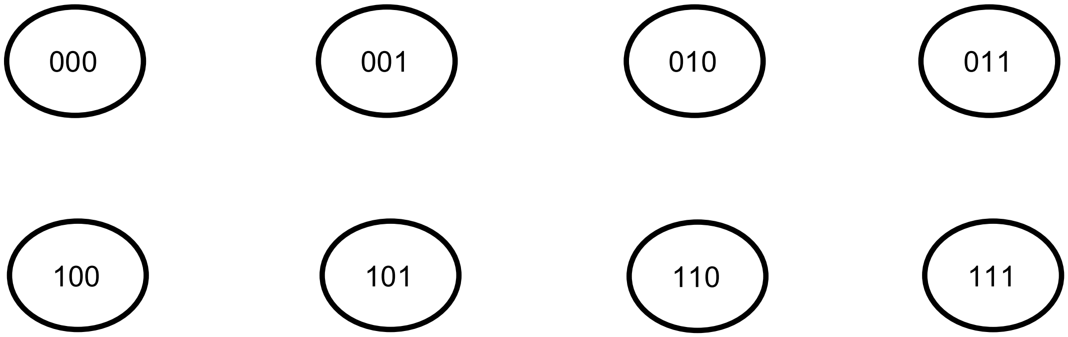
1. (5 points) Give Boolean expressions for the T_2, T_1, T_0 flip-flop inputs, each as a function of the state variables Q_2, Q_1, Q_0.

   T_2 = ____________ T_1 = ____________ T_0 = ____________

2. (9 points) Complete the following table.

Current State			Flip-Flop Inputs			Next State		
Q_2	Q_1	Q_0	T_2	T_1	T_0	Q_2^+	Q_1^+	Q_0^+
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

3. (5 points) Complete the state transition diagram.



4. (3 points) Assuming the start state is $Q_2Q_1Q_0$=000, in what sequence does this counter count?

Problem 4 (15 points): Von Neumann model

1. (5 points) Which phase(s) of the instruction cycle access memory when processing the LEA instruction? Circle **ALL correct** answers. If a phase is not used at all in processing this instruction, *don't circle it*.

FETCH DECODE EVALUATE ADDRESS

FETCH OPERANDS EXECUTE STORE RESULT

2. (5 points) Which component(s) of the Von Neumann model are involved in processing the JMP instruction? Processing includes all phases of the instruction cycle. Circle **ALL correct** answers.

MEMORY PROCESSING UNIT CONTROL UNIT INPUT OUTPUT

3. (5 points) Which component(s) of the Von Neumann model set the *GatePC* signal in the LC-3 datapath? Circle **ALL correct** answers.

MEMORY PROCESSING UNIT CONTROL UNIT INPUT OUTPUT

Problem 5 (26 points): LC-3 instructions

The following LC-3 program fragment, represented as four hexadecimal numbers, is stored in memory at the indicated locations and the following values are stored in registers:

Address	Instruction
x3FFF	xAFFE
x4000	x2001
x4001	x743F
x4002	x3002

Register	Value
R0	xF021
R1	xF023
R2	xF025
R3	xF027

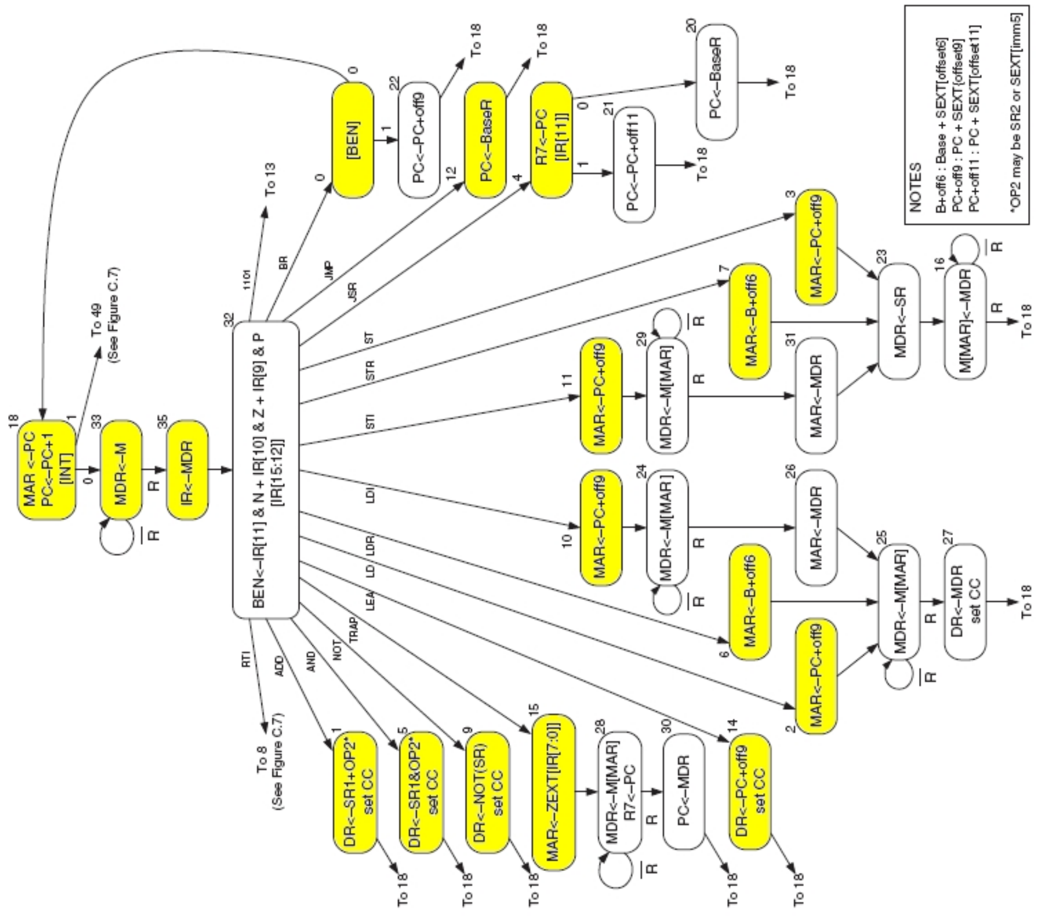
1. (12 points) Complete the following table. (*Refer to the LC-3 handout at the end of the exam.*)

Address	Instruction	Binary instruction	RTL (Be specific to this instruction)
x3FFF	xAFFE	1010 111 111111110	R7 ← M[M[PC - 2]] setcc
x4000	x2001		
x4001	x743F		
x4002	x3002		

2. (14 points) Assuming PC is initially set to x4000, trace the execution of the given program segment for **two** instruction cycles, filling in the table below. Write down the values stored in the PC, IR, MAR, MDR, R0, N, Z, and P registers **at the end of each instruction cycle**. Values for PC, IR, MAR, MDR, and R0 should be written in ***hexadecimal***. Values for N, Z, and P should be written in ***binary***.

PC	IR	MAR	MDR	R0	N	Z	P

LC-3 FSM



NOTES: RTL corresponds to execution (after fetch!); JSRR not shown

LC-3 Instructions

ADD	0001	DR	SR1	0	00	SR2	ADD DR, SR1, SR2

DR ← SR1 + SR2, Setcc

ADD	0001	DR	SR1	1	imm5	ADD DR, SR1, *imm5*

DR ← SR1 + SEXT(imm5), Setcc

AND	0101	DR	SR1	0	00	SR2	AND DR, SR1, SR2

DR ← SR1 AND SR2, Setcc

AND	0101	DR	SR1	1	imm5	AND DR, SR1, *imm5*

DR ← SR1 AND SEXT(imm5), Setcc

BR	0000	n	z	p	PCoffset9	BR{nzp} *PCoffset9*

((n AND N) OR (z AND Z) OR (p AND P)):
PC ← PC + SEXT(PCoffset9)

JMP	1100	000	BaseR	000000	JMP BaseR

PC ← BaseR

JSR	0100	1	PCoffset11	JSR *PCoffset11*

R7 ← PC, PC ← PC + SEXT(PCoffset11)

TRAP	1111	0000	trapvect8	TRAP *trapvect8*

R7 ← PC, PC ← M[ZEXT(trapvect8)]

LD	0010	DR	PCoffset9	LD DR, *PCoffset9*

DR ← M[PC + SEXT(PCoffset9)], Setcc

LDI	1010	DR	PCoffset9	LDI DR, *PCoffset9*

DR ← M[M[PC + SEXT(PCoffset9)]], Setcc

LDR	0110	DR	BaseR	offset6	LDR DR, BaseR, *offset6*

DR ← M[BaseR + SEXT(offset6)], Setcc

LEA	1110	DR	PCoffset9	LEA DR, *PCoffset9*

DR ← PC + SEXT(PCoffset9), Setcc

NOT	1001	DR	SR	111111	NOT DR, SR

DR ← NOT SR, Setcc

ST	0011	SR	PCoffset9	ST SR, *PCoffset9*

M[PC + SEXT(PCoffset9)] ← SR

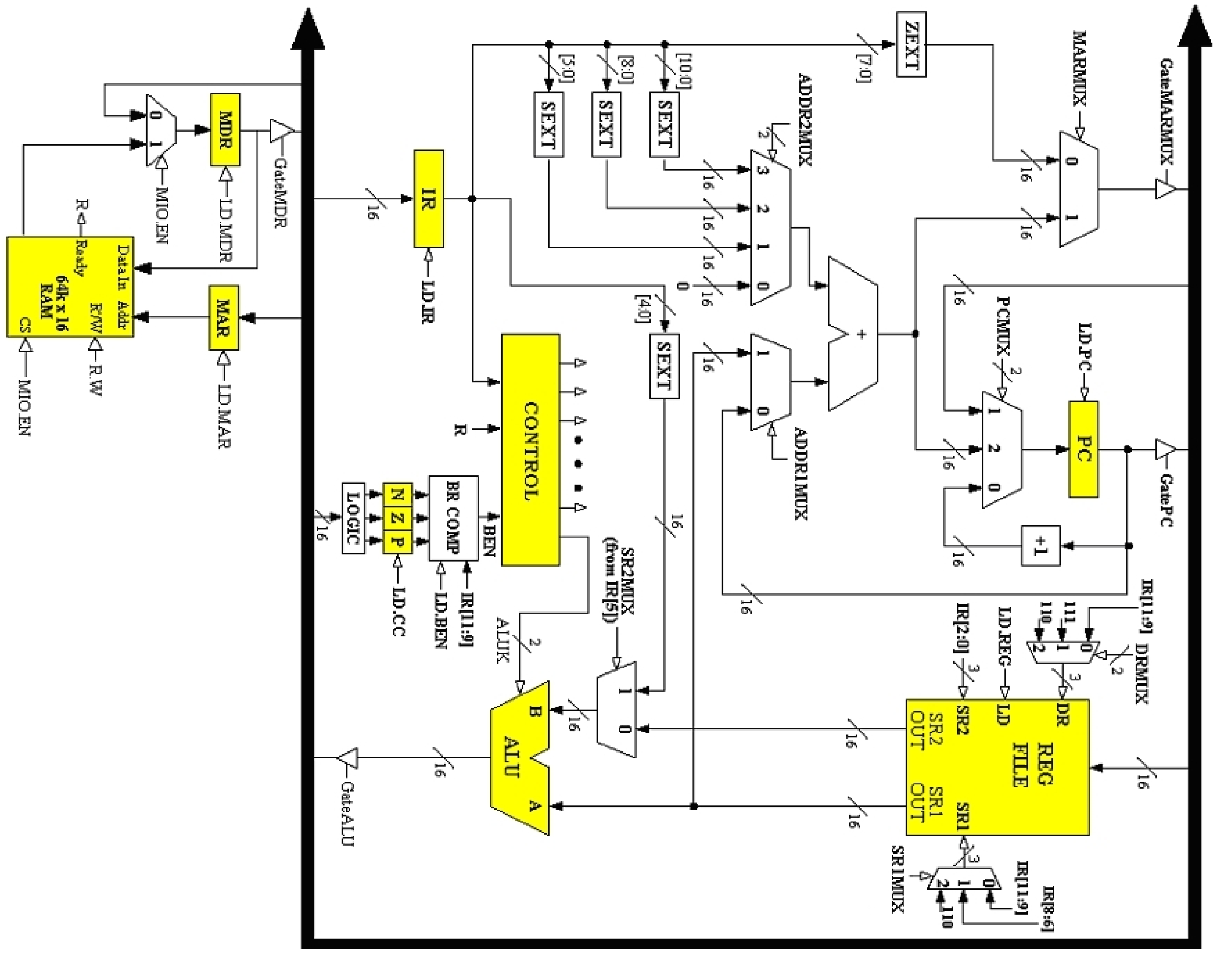
STI	1011	SR	PCoffset9	STI SR, *PCoffset9*

M[M[PC + SEXT(PCoffset9)]] ← SR

STR	0111	SR	BaseR	offset6	STR SR, BaseR, *offset6*

M[BaseR + SEXT(offset6)] ← SR

LC-3 Datapath



LC-3 Datapath Control Signals

Signal	Description
LD.MAR	= 1, MAR is loaded
LD.MDR	= 1, MDR is loaded
LD.IR	= 1, IR is loaded
LD.PC	= 1, PC is loaded
LD.REG	= 1, register file is loaded
LD.BEN	= 1, updates Branch Enable (BEN) bit
MARMUX	= 0, chooses ZEXT IR[7:0] = 1, chooses address adder output
ADDR1MUX	= 0, chooses PC = 1, chooses reg file SR1OUT
ADDR2MUX	= 00, chooses "0...00" = 01, chooses SEXT IR[5:0] = 10, chooses SEXT IR[8:0] = 11, chooses SEXT IR[10:0]
PCMUX	= 00, chooses PC + 1 = 01, chooses system bus = 10. chooses address adder output
SR1MUX	= 00, chooses IR[11:9] = 01, chooses IR[8:6] = 10, chooses "110"

Signal	Description
LD.CC	= 1, updates status bits from system bus
GateMARMUX	= 1, MARMUX output is put onto system bus
GateMDR	= 1, MDR contents are put onto system bus
GateALU	= 1, ALU output is put onto system bus
GatePC	= 1, PC contents are put onto system bus
MIO.EN	= 1, Enables memory, chooses memory output for MDR input = 0, Disables memory, chooses system bus for MDR input
R.W	= 1, M[MAR]<-MDR when MIO.EN = 1 = 0, MDR<-M[MAR] when MIO.EN = 1
ALUK	= 00, ADD = 01, AND = 10, NOT A = 11, PASS A
DRMUX	= 00, chooses IR[11:9] = 01, chooses "111" = 10, chooses "110"