Department of Electrical and Computer Engineering
The University of Texas at Austin

EE 360N, Spring, 2001
Yale Patt, Instructor
Onur Mutlu, Kameswar Subramaniam, TAs
Exam 1, February 28, 2001

Name:________________________________

Problem 1 (25 points):_________

Problem 2 (15 points):_________

Problem 3 (20 points):_________

Problem 4 (20 points):_________

Problem 5 (20 points):_________

Total (100 points):__________________

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is recorded on each sheet of the exam.

GOOD LUCK!

Name:______________________________

Problem 1 (25 points):

Part a (5 points): Consider the following assembly language program. As part of the assembly process, a symbol table is constructed. Fill in the symbol table.

```
        .ORIG   x3000
        JSR     A
        OUT
        BRnzp   DONE
A       AND     R0,R0,#0
        ADD     R0,R0,#5
        JSR     B
        RET
DONE    HALT
B       LD      R1,ASCII
        ADD     R0,R0,R1
        RET
ASCII   .FILL   x0030
        .END
```

Symbol	Address

Part b (5 points): The program is supposed to print the number 5 on the screen and then halt. It does not work. Why? Answer in no more than 15 words.

Part c (5 points): Which of the following are part of the LC-2 ISA, and which are part of the microarchitecture. For each of the four structures below circle either ISA or Microarch.

IR ISA Microarch

Z flag ISA Microarch

R5 ISA Microarch

BEN ISA Microarch

Part d (5 points): What characteristic of the cache exploits temporal locality? (two words are enough to answer this question – if you use more than ten, your answer will be considered wrong on account of verbosity!)

What characteristic of the cache exploits spatial locality? (two words are enough to answer this question – if you use more than ten, your answer will be considered wrong on account of verbosity!)

Name:________________________________

Problem 1 continued:

Part e (5 points): We would like to add the opcode XOR to the LC-2 ISA.

	15–12	11–9	8–6	5	4–3	2–0
XOR DR, SR1, SR2	XOR	DR	SR1	0	00	SR2
XOR DR, SR1, imm5	XOR	DR	SR1	1	imm5	

Can we change any existing opcode to XOR without losing backward compatibility? If so, which one, and why? NOTE: maintaining backward compatibility means programs written for the LC-2 before the change will still execute correctly.

	15–12	11–9	8–6	5–0
ADD+	0001	DR	SR1	0 | 00 | SR2
ADD+	0001	DR	SR1	1 | imm5
AND+	0101	DR	SR1	0 | 00 | SR2
AND+	0101	DR	SR1	1 | imm5
BR	0000	n | z | p	pgoffset9	
JSR	0100	L | 00	pgoffset9	
JSRR	1100	L | 00	BaseR	index6
LD+	0010	DR	pgoffset9	
LDI+	1010	DR	pgoffset9	
LDR+	0110	DR	BaseR	index6
LEA+	1110	DR	pgoffset9	
NOT+	1001	DR	SR	111111
RET	1101	000000000000		
RTI*	1000	000000000000		
ST	0011	SR	pgoffset9	
STI	1011	SR	pgoffset9	
STR	0111	SR	BaseR	index6
TRAP	1111	0000 (bits 11–8)	trapvect8 (bits 7–0)	

Name:________________________________

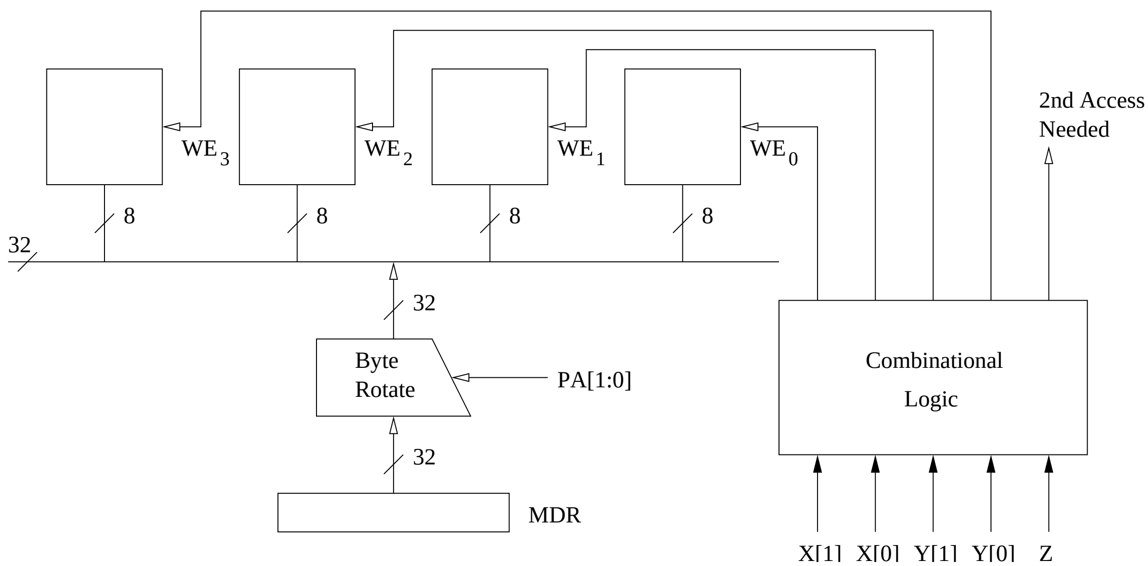
Problem 2 (15 points):

A 64KB write-through cache has been designed to work with a computer having an 16 MB physical address space. The cache is 8-way set associative, has a block size of 8 bytes, and uses a victim/non-victim pair of bits to implement a pseudo-LRU replacement policy. The cache is not sectored. How many bits of storage are required to implement the tag store? How many bits of storage are required to implement the data store?

Name:________________________________

Problem 3 (20 points):

The physical memory system below allows unaligned accesses. We wish to support STB (store byte), STH (store half-word) and STW (store word) instructions. A half-word is two bytes; a word is four bytes.



MDR contains the data to be stored into memory. If two memory accesses are required to complete the store, the signal "2nd Access Needed" should be asserted during the first access.

Your task is to complete the specification of the control logic required to generate the write enable signals. Other logic will provide the proper address bits for the memory chips.

Part I. The WE signals are determined by the values of two 2-bit signals (X and Y) and one 1-bit signal (Z). Identify them

X[1]:

X[0]:

Y[1]:

Y[0]:

Z:

Name:________________________________

Problem 3 continued:

Part II. Complete the 5-input, 5-output truth table to specify the combinational logic required for the four Write Enable signals and the "2nd Access Needed" signal

X[1]	X[0]	Y[1]	Y[0]	Z	WE_3	WE_2	WE_1	WE_0	2nd
0	0	0	0	0					
0	0	0	0	1					
0	0	0	1	0					
0	0	0	1	1					
0	0	1	0	0					
0	0	1	0	1					
0	0	1	1	0					
0	0	1	1	1					
0	1	0	0	0					
0	1	0	0	1					
0	1	0	1	0					
0	1	0	1	1					
0	1	1	0	0					
0	1	1	0	1					
0	1	1	1	0					
0	1	1	1	1					
1	0	0	0	0					
1	0	0	0	1					
1	0	0	1	0					
1	0	0	1	1					
1	0	1	0	0					
1	0	1	0	1					
1	0	1	1	0					
1	0	1	1	1					
1	1	0	0	0					
1	1	0	0	1					
1	1	0	1	0					
1	1	0	1	1					
1	1	1	0	0					
1	1	1	0	1					
1	1	1	1	0					
1	1	1	1	1					

Name:______________________________

Problem 4 (20 points):

We have an ISA that supports an 8-bit, byte-addressable virtual address space. The corresponding physical memory has only 128 bytes. For this system, each page contains 16 bytes and, in the page table, each entry is one byte in length. A simple, one-level translation scheme is used, and the page table resides in physical memory.

i. How many bits of each virtual address is the virtual page number?

ii. What is the size of the physical frame number field in each page table entry?

The initial contents of the frames of physical memory are shown below:

Frame 0	empty
Frame 1	Page 9
Frame 2	Page 5
Frame 3	Page 10
Frame 4	empty
Frame 5	Page 11
Frame 6	Page table
Frame 7	Page 2

iii. Give the physical address corresponding to the virtual address 00101010.

iv. What are the contents of the PTBR (provide a physical address) ?

v. We show below enough space to draw a page table. We have started it for you by showing where the first three entries should go.

The contents of a page table entry are as follows (V refers to the valid bit and PFN is the physical frame number):

<—— 8 bits ——>

V		PFN

Using this PTE format, complete the entries of the page table that correspond to the contents of the physical memory shown above.

Name:________________________________

Problem 5 (20 points):

We wish to implement a new instruction ADDM, which (similar to an IA-32 instruction) adds the contents of a memory location to the contents of a register or an immediate value. The specification of this instruction is as follows:

Assembler Formats

ADDM DR, SR1, SR2
ADDM DR, SR1, imm5

Encodings

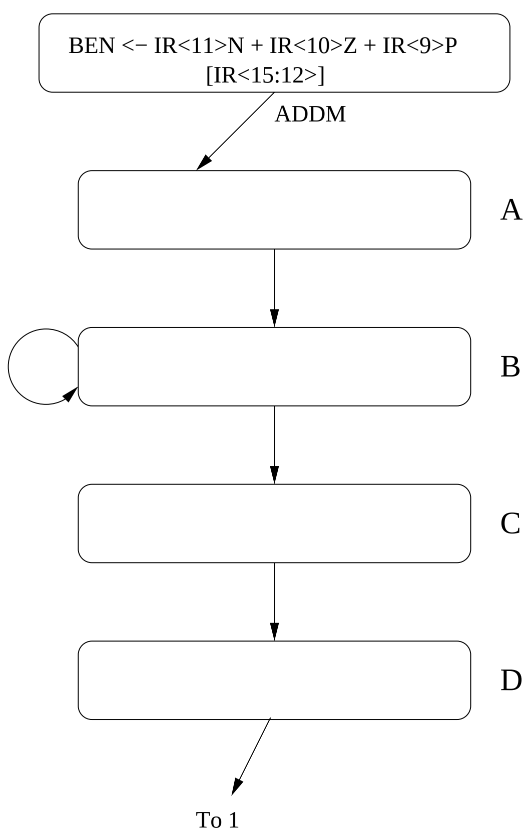
15 12	11 9	8 6	5	4 3	2 0
ADDM	DR	SR1	0	00	SR2

15 12	11 9	8 6	5	4 0
ADDM	DR	SR1	1	imm5

Operation

if (bit[5] == 0)
    DR = Memory[SR1] + SR2;
else
    DR = Memory[SR1] + SEXT(imm5);
setcc(DR);

a. We show below an addition to the state diagram necessary to implement ADDM. Using the notation of the LC-2 State Diagram (which we have included for your reference), describe inside each "bubble" what happens in each state.



b. Add to the Data Path (which we have supplied) any additional structures and any additional control signals needed to implement ADDM. Label the additional control signals "Extra signal 1", "Extra signal 2", etc.

c. The processing in each state A,B,C,D is controlled by asserting or negating each control signal. Enter a 1 or a 0 as appropriate for the microinstructions corresponding to states A,B,C,D

	LD.MAR	LD.MDR	LD.IR	LD.BEN	LD.REG	LD.CC	LD.PC	GatePC	GateMDR	GateALU	GateMARMX	PCMX { PC+1, BUS, REG, CONCAT / 00 , 01 , 10 , 11 }	MEM.EN	RW RD(0), WR(1)	DRMX IR[11:9] (0) , R7 (1)	SR1MX { IR[11:9], IR[8:6], R7 / 00 , 01 , 10 }	MARMX { IR[7:0], R+ZEXT, CONCAT / 00 , 01 , 10 }	ALUK { ADD, AND, NOT, PASSA / 00 , 01 , 10 , 11 }	Extra control signal 1 (if needed)	Extra control signal 2 (if needed)	Extra control signal 3 (if needed)	Extra control signal 4 (if needed)
A																						
B																						
C																						
D																						