

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

EE 360N, Spring 2003
Yale Patt, Instructor
Hyesoon Kim, Onur Mutlu, Moinuddin Qureshi, Santhosh Srinath, TAs
Final Exam, May 9, 2003

Name: _____

Problem 1 (30 points): _____

Problem 2 (15 points): _____

Problem 3 (10 points): _____

Problem 4 (10 points): _____

Problem 5 (15 points): _____

Problem 6 (20 points): _____

Problem 7 (10 points): _____

Problem 8 (10 points): _____

Total (120 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 (30 points):

Part a (5 points): What is the highest priority interrupt?

When does it occur?

Part b (5 points): An Omega network made up of 8×8 crossbar switches connects 512 processors to 512 memory units. How many switches does this Omega network contain?

Part c (5 points): A six-dimensional hypercube contains nodes labeled 101101 and 010111. What is the latency (in hops) to send information from one to the other?

Name: _____

Problem 1 continued:

Part d (5 points): A floating point representation in the format of the IEEE standard has 32 exactly representable numbers in each binade, and it has 14 binades that contain representable normalized positive numbers. How many bits are required to represent each floating point number?

Part e (5 points): The pipelined design of the LC-3b that you worked with in Lab 4: There was a load enable signal associated with the DE, AGEX, and MEM stages. This signal was not used to load the SR latches. Why? (20 words maximum.)

Part f (5 points): I want a perfect cache replacement policy (i.e., one that maximizes the cache hit ratio) for a two-way set associative cache. Will LRU (least recently used) replacement satisfy this requirement? Why/why not? (20 words maximum. Moin did it in four. I took five, but I stuck a “the” in there.)

Name: _____

Problem 2 (15 points):

A processor supports a one-level, byte-addressable virtual memory system with an 8-bit virtual address space and a page size of 16 bytes. The physical memory of this processor is 128 bytes. The virtual memory system supports 2 levels of privilege (kernel and user) and 3 levels of access (write, read, none).

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

Part b. How many frames are there in physical memory?

Part c. How many reasonable privilege level/access level combinations are there? Show these reasonable combinations.

Part d. A page table entry for this system contains a modified bit, a reference bit along with the bits required for access protection and translation. The structure of a page table entry is shown below:

V	M	R	Prot	PFN
---	---	---	------	-----

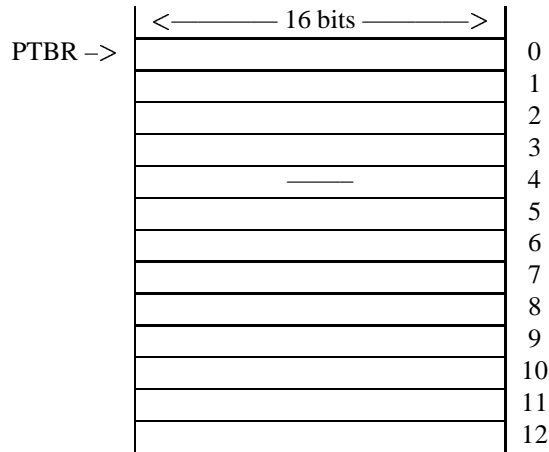
Assuming that the number of protection bits (Prot) is made as small as possible, what is the minimal size of each page table entry in bits?

Name: _____

Problem 2 continued:

Part e. What is the minimal size of the page table in bits?

Part f. Shown below are the first thirteen entries of the Page Table. We have allocated 16 bits for each entry. Your job is to fill in these entries. Since part d states that you don't need 16 bits for an entry, fill in each entry with as many leading 0's as you need to. If a PTE is valid, assume that M, R, and Prot are all 0. If a PTE is invalid, fill in all 0's for that PTE. If you do not know anything about a PTE, draw a line through the entry (see, for example, the entry for page 4).



To help you, the following table shows what would happen if the processor next attempted to access each of the following virtual addresses.

If the next access was to	Result would be
VA = xAC	Page fault
VA = x1B	PA = x3B
VA = x05	Page fault
VA = x33	Page fault
VA = x29	PA = x09

Part g. If the page table starts at the beginning of a frame, what are all the possible addresses that can be in the PTBR? Assume the page table you completed in Part f.

Name: _____

Problem 3 (10 points):

Say we have a computation which is partially parallelizable. The time it takes to perform this computation on a single processor is T_1 and the time it takes to perform this computation on a multiprocessor system with p processors is T_p . Suppose we perform the computation on a multiprocessor system with 6 processors and measure that 20% of the execution time (T_6) is spent on the parallelizable portion of the computation.

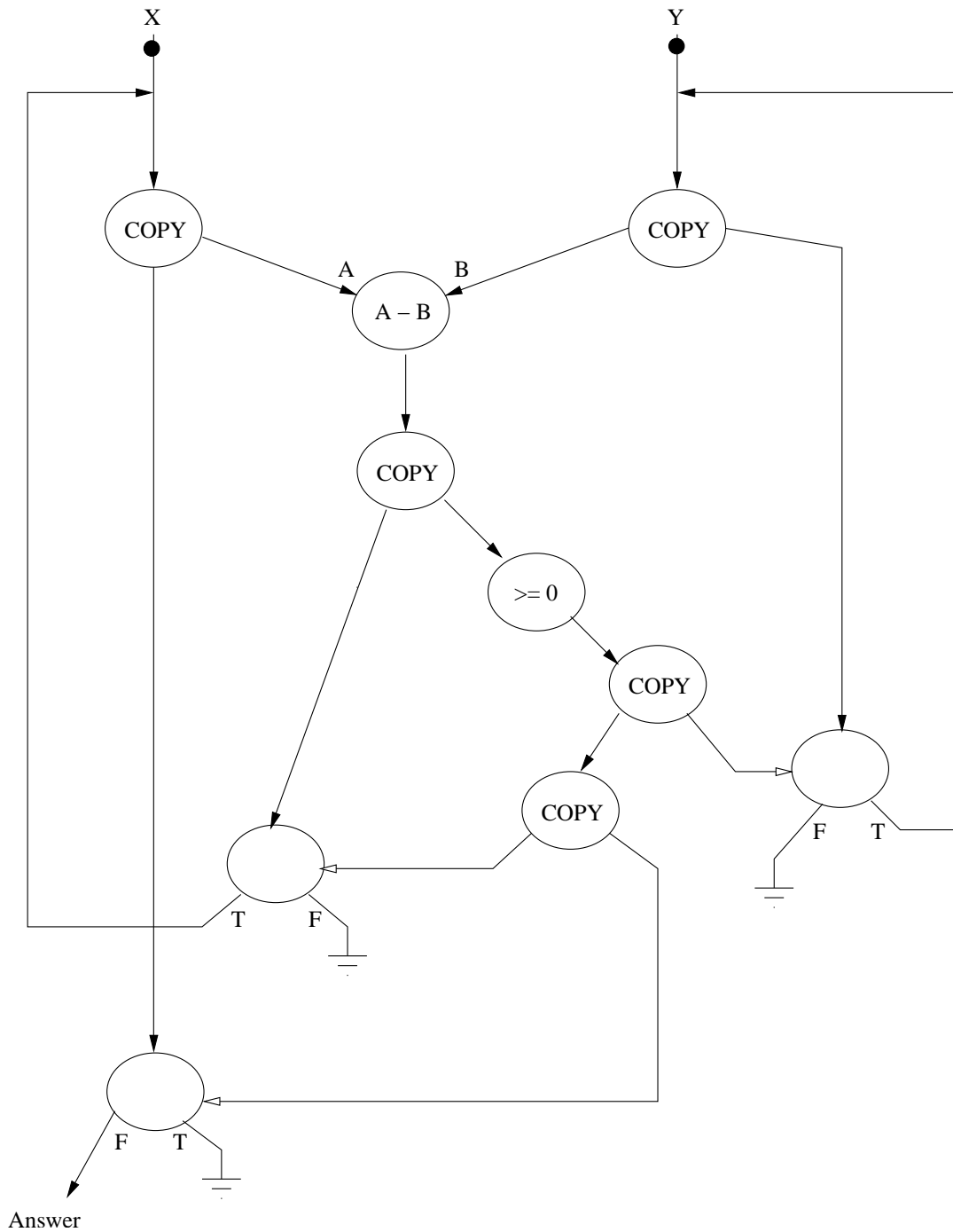
Part a. What fraction of the computation is parallelizable?

Part b. What is the speedup obtained by performing the computation on 6 processors compared to the case where we perform the computation on a single processor?

Part c. What would be the maximum speedup we would obtain if we had an infinite number of processors?

Name: _____

Problem 4 (10 points):



If X and Y are positive integers, what does the dataflow graph compute?

Answer =

Name: _____

Problem 5 (15 points):

The following numbers are represented exactly with a 9-bit floating point representation, in the format of the IEEE Floating Point standard:

$-\infty$, -1 , 0 , $5/16$, 19.5 , 48 .

Part a. How many bits are needed for the fraction?

Part b. What is the bias?

Part c. Write each number in the 9-bit floating point representation, below:

Value	Representation
48	
19.5	
5/16	
0	
-1	
$-\infty$	

Name: _____

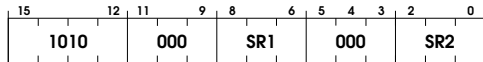
Problem 6 (20 points):

We would like to add a SWAP instruction to the LC-3b ISA that swaps the words in two memory locations. The instruction is specified as follows:

Assembler format

SWAP SR1, SR2

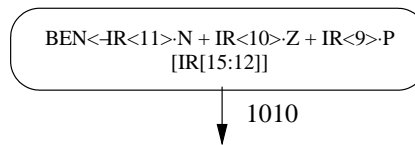
Encoding



Operation

swap (MEM[SR1], MEM[SR2]);

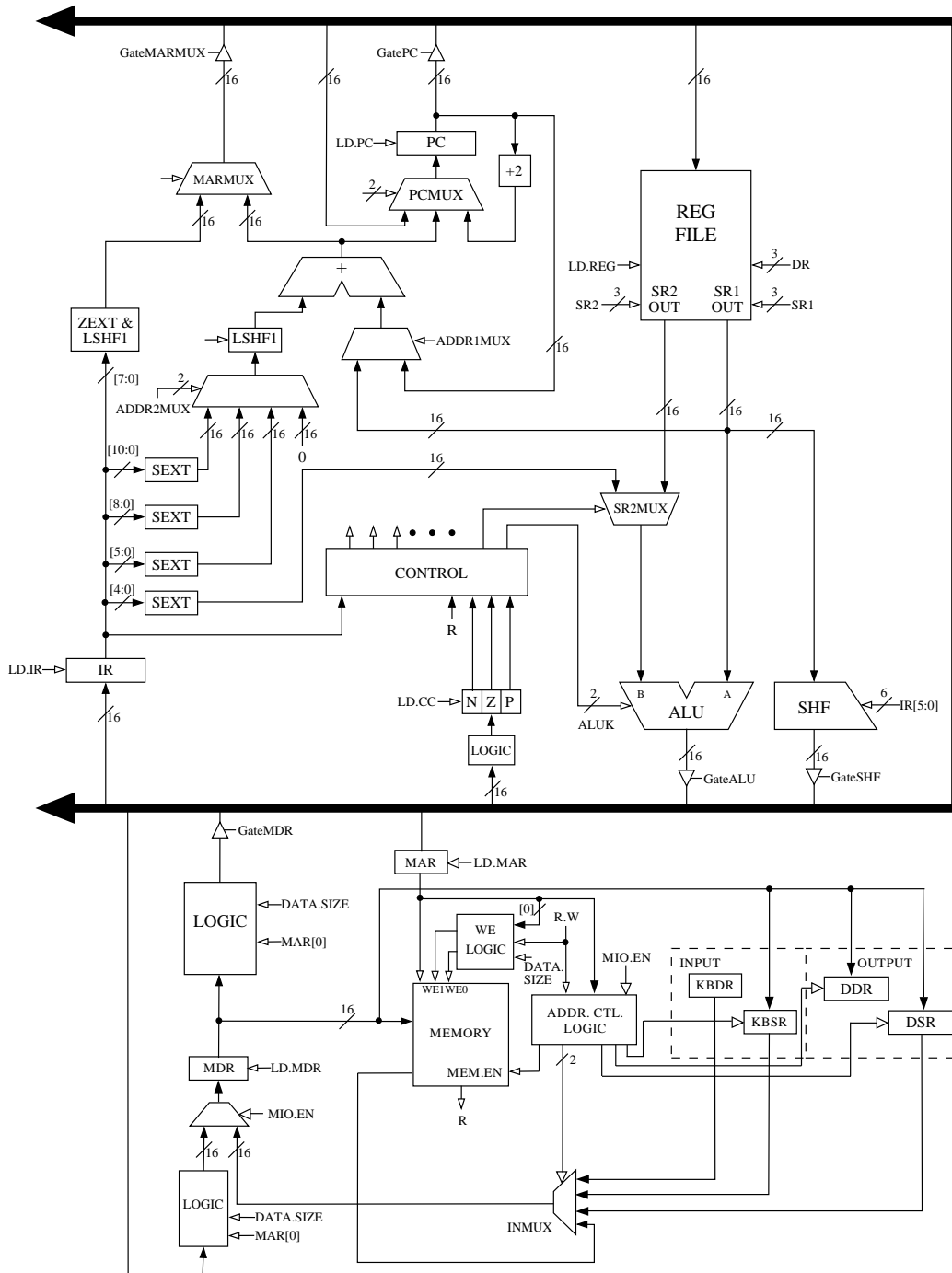
Part a. What changes do we need to make to the state diagram to accommodate this instruction, assuming this instruction does not cause any exceptions? What new states do we need to add? Show these states below. Mark the states with letters A, B, C, ... Use the notation of the LC-3b state diagram.



Name: _____

Problem 6 continued:

Part b. What changes do we need to make to the datapath? Show these changes on the datapath figure along with the control signals used to control the added structures.



Name: _____

Problem 6 continued:

Part c. Show the microinstructions for the states you added in Part A. If a control signal is 0, you can leave the entry for that control signal blank.

	A	B	C	
LD.MAR				
LD.MDR				
LD.IR				
LD.BEN				
LD.REG				
LD.CC				
LD.PC				
GatePC				
GateMDR				
GateALU				
GateMARMUX				
GateSHF				
PCMUX				(PC+2, BUS, ADDR 00, 01, 10
DRMUX				IR[11:9](0), R7(1)
SR1MUX				IR[11:9](0), IR[8:6](1)
ADDR1MUX				PC(0), BaseR(1)
ADDR2MUX				(ZERO, offset6, PCoffset9, PCoffset11 00, 01, 10, 11
MARMUX				LSHF(ZEXT[IR[7:0],1](0), adder(1)
ALUK				(ADD, AND, XOR, PASSA 00, 01, 10, 11
MIO.EN				
R.W				RD(0), WR(1)
DATA.SIZE				BYTE(0), WORD(1)
LSHF1				
NEW CONTROL SIGNAL 1				<input type="text"/>
NEW CONTROL SIGNAL 2				<input type="text"/>
NEW CONTROL SIGNAL 3				<input type="text"/>
NEW CONTROL SIGNAL 4				<input type="text"/>
NEW CONTROL SIGNAL 5				<input type="text"/>
NEW CONTROL SIGNAL 6				<input type="text"/>

Part d. What exception(s) can this instruction cause, if any? In which state(s) would the exception(s) be detected? Assume the baseline LC-3b (i.e., no virtual memory).

Name: _____

Problem 7 (10 points):

We have a multiprocessor system in which three processors are connected to a shared memory module and each processor has its own cache. Processors use Goodman's snoopy cache scheme. Processors A, B, and C read and write variable X in the following order:

- Processor B: Write X
- Processor A: Read X
- Processor C: Read X
- Processor C: Write X
- Processor C: Read X
- Processor C: Write X
- Processor B: Write X
- Processor C: Read X

Part a. Which state is the cache block for X in for each processor's cache after all accesses are complete? Assume that, in the beginning, cache block for X is invalid in all caches.

Processor A's cache:	
Processor B's cache:	
Processor C's cache:	

Part b. Mark the access with a circle if the access results in a BW broadcast.

- Processor B: Write X
- Processor A: Read X
- Processor C: Read X
- Processor C: Write X
- Processor C: Read X
- Processor C: Write X
- Processor B: Write X
- Processor C: Read X

Part c. If Processor C performs a Write to X after all the accesses, which state would the cache block for X be in for each processor's cache after this last Write access is complete?

Processor A's cache:	
Processor B's cache:	
Processor C's cache:	

Name: _____

Problem 8 (10 points):

In the pipeline of Lab 4 (see figure below), it would take $N+4$ cycles to complete N instructions, if there are no dependencies and if we have perfect caches. For example, it would take 5 cycles to complete the instruction `ADD R0, R0, R0` and it would take 7 cycles to complete the following code segment

```

ADD R1, R1, R1
AND R2, R2, R2
JMP R3
    
```

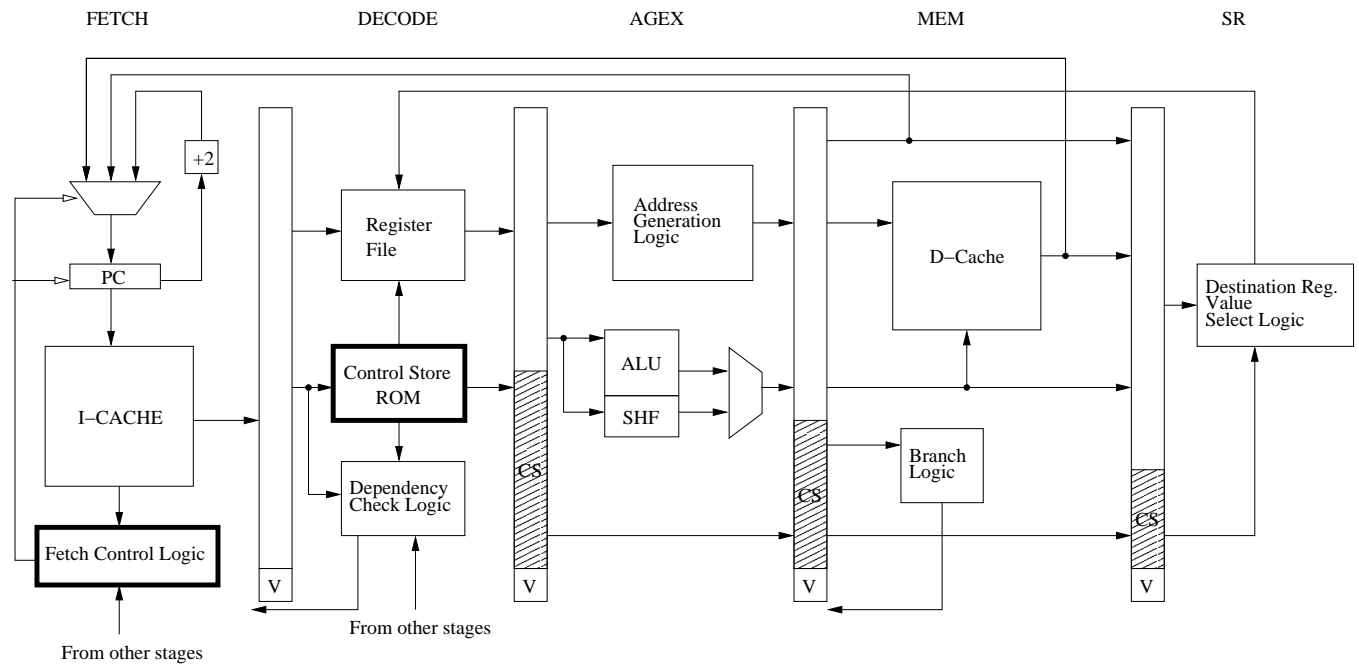


Fig.1 LC-3b pipeline diagram

Part a. How many cycles would it take to execute the following code segments? If it takes more than $N+4$ cycles explain why.

1. `ADD R0, R1, R2`
`XOR R2, R1, R0`

2. `ADD R0, R1, R2`
`AND R0, R3, R4`

Name: _____

Problem 8 continued:

3. AND R2, R1, x0
 ADD R1, R6, R1

4. ADD R7, R1, R2
 BRz X

5. ADD R7, R1, R2
 BRz X
X XOR R2, R1, R2

6. AND R3, R2, R1
 ADD R5, R4, R3
 JSR Y
Y ADD R7, R7, R7

Part b. Now assume that the I-cache is perfect but the D-cache access takes three cycles (for the first two cycles DCACHE.R is 0 and in the third cycle DCACHE.R is 1). How many cycles would it take to complete the following code segments? If it takes more than N+4 cycles explain why.

1. ADD R0, R1, R2
 LDW R0, R0, x0
 AND R0, R1, R2

2. LDW R1, R2, #3
 XOR R7, R7, R7
 BRz X
 ADD R1, R2, R3
X JSR Y

A state machine for the LC-3b (from Appendix C)

