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

EE 382N.1 Fall 2018
Dam Sunwoo, Instructor
Pritesh Chhajed, TA
Exam 2
November 19, 2018

Name: SOLUTION

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: The exam is open book, open notes, bring anything type of exam, but there are constraints: Do not ask anyone or anything for help. You may only search for text in PDFs or other material that you have downloaded. NO web searches. Do not share any material related to the exam with anyone (or upload it anywhere). The exam will be made public soon.

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

Please read the following sentence, and if you agree, sign where requested: I have not given nor received any unauthorized help on this exam.

Signature:_____

GOOD LUCK!

Name:_____

Problem 1 (25 points): Answer the following questions.

Part a (5 points):

```
for (i=1; i<200; i++)
    A[i] = (A[i-1] + B[i]) * 2;
```

Is this loop vectorizable using Cray-1-style vector processing covered in the lecture? Circle one: **YES** / **(NO)**
Explain why / why not (10 words should be enough).

Each iteration is dependent on the previous one.

Part b (5 points): A 15 × 10 (rows × columns) matrix is stored in column-major order in memory. We would like to read out rows of this matrix and store them in vector registers. What should Vector Length Register (VLR) and Vector Stride Register (VST) be set to?

VLR= 10

VST= 15

Part c (5 points): The diameter of an interconnection network is defined as the maximum hops between any two nodes. The bisection bandwidth is the minimum bandwidth of any two halves of the network. What is the diameter of a 4x4 2D Torus? What is the bisection bandwidth of a 4x4 2D Torus (in terms of number of links)?

Diameter= 4

Bisection BW= 8

Part d (10 points): We are designing a 48KB virtually-indexed physically-tagged (VIPT) L1 data cache with write-back, 32-byte cache lines, and tree-like Pseudo LRU policy for replacement. This cache will be used in a system with 4KB page sizes, a 32-bit virtual address space, and a 30-bit physical address space. All addresses are byte-addressable. We want to make sure we have no synonyms (aliasing) using this cache. We also want to minimize the set-associativity of this cache.

How many **tag bits** do we need for the entire L1 data cache? Show your work.

To avoid synonyms in VIPT:
Cache size ≤ page size × associativity
48KB ≤ 4KB × assoc
min assoc = 12
There are 2^7 sets, tag bits per block: 30 − 7 − 5 = 18 bits.
Tree-like pLRU: 12 − 1 = 11 bits per set.
Total tagstore bits: $2^7 \times (12 \times (18 + 1 + 1) + 11) = 128 \times 251$
= 32128 bits

(Valid, WB)

Name:_____

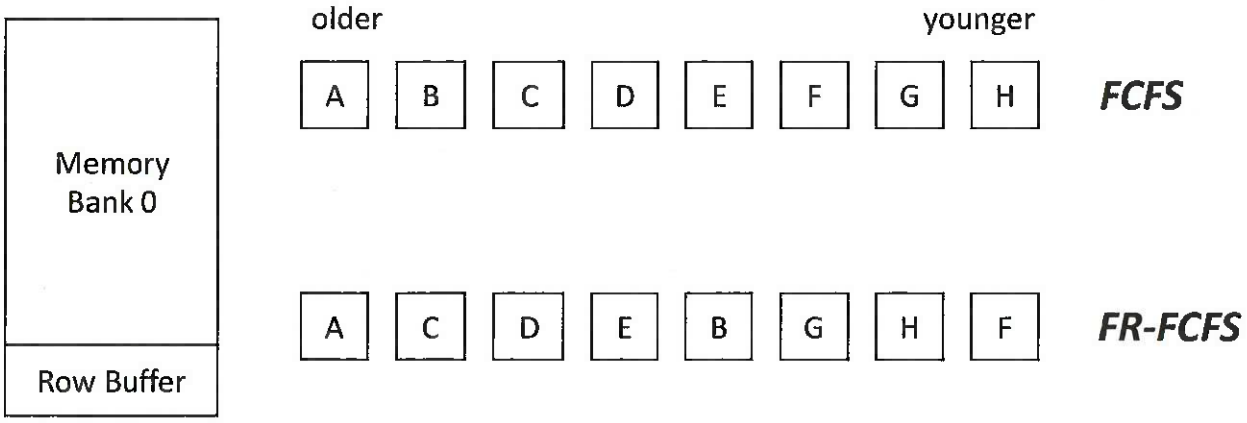
Problem 2 (15 points): The memory controller issues DRAM commands to access data from a memory bank. Consider the following commands and latencies:

- **ACTIVATE**: Loads the row that needs to be accessed into the row buffer. Also called *opening a row*. **(Latency: 10ns)**
- **PRECHARGE**: Prepares the bank for the next access and empties the row-buffer. Also called *closing a row*. **(Latency: 10ns)**
- **READ / WRITE**: Reads/writes data from/to row. **(Latency: 10ns)**

The memory controller can schedule the memory requests to increase performance. Consider the following scheduling policies:

- **FCFS (First-Come First-Serve)**: Service memory requests in the order they arrive at the memory controller.
- **FR-FCFS (First-Ready, First-Come-First-Serve)**: Service the oldest request that hits in the current open row, if any, first (i.e., First-Ready). Otherwise, use FCFS.

There are eight memory requests (labeled A through H, where A is the oldest and H is the youngest request) queued up at the memory controller at time 0. The order they are processed under FCFS and FR-FCFS, respectively, are depicted in the figure below.



Assume the following:

- The memory system has one DRAM channel and DRAM bank.
- The row-buffer is closed (empty) at time 0.
- No additional requests arrive at the memory controller.

Part a (5 points): List the memory accesses that resulted in row-buffer hits under FCFS?

D, E, H

Part b (5 points): List the memory accesses that resulted in row-buffer hits under FR-FCFS?

C, D, E, G, H

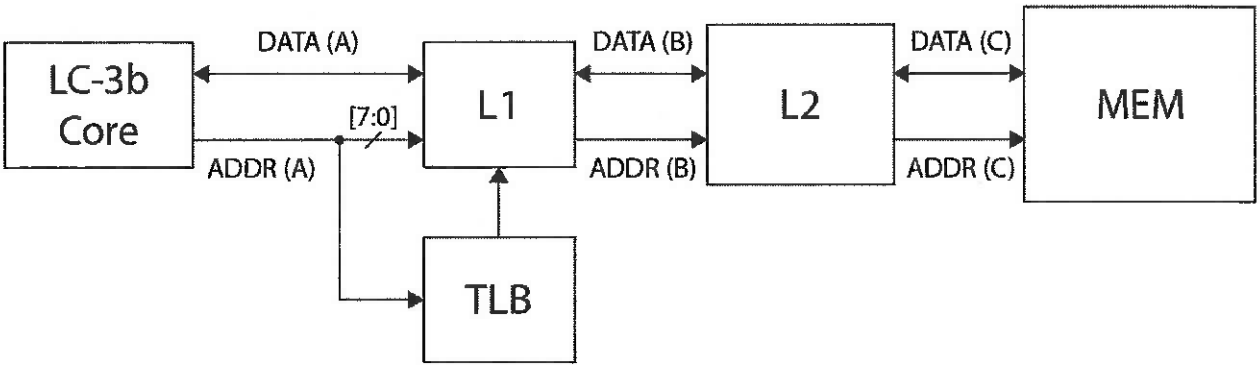
Part c (5 points): What is the total time it took to process all requests under each scheduling policy?

FCFS: 170 ns

FR-FCFS: 130 ns

Name:_____

Problem 3 (20 points): We have augmented the LC-3b with the memory hierarchy shown below.



Addr(A), Addr(B), and Addr(C) represent addresses that access respectively L1, L2, and Memory. Data(B) and Data(C) each transfer a full cache line.

The table below shows a sequence of five memory accesses from the LC-3b core. If ADDR(A) misses in L1, an access is required to L2. If ADDR(B) misses in L2, an access is required to Memory. Each of the five requests from the LC-3b core must complete before the next access from the LC-3b core is initiated.

ADDR(A)	ADDR(B)	ADDR(C)	Read/Write
0x3000			Read
	0x100		Read
		0x100	Read
0x3003			Write
0x3004			Read
	0x104		Read
0x3008			Read
	0x108		Read
		0x108	Read
0x8000			Read
	0x100		Write
	0x200		Read
		0x200	Read

You may make the following assumptions:

- Virtual addresses are 16 bits and the page size is 256B.
- The TLB has 2 entries and is fully associative.
- All accesses to the TLB are hits.
- The L1 and L2 are both physically-indexed, physically-tagged.
- The L1 contains 64 sets.
- If a cache line is present in the L1, it will also be present in the L2 (although the contents of L2 may not be correct).
- The caches are initially empty.

Name:_____

Part a (2 points): How many bytes are in an L1 cache line? 4 Bytes

Part b (2 points): How many bytes are in an L2 cache line? 8 Bytes

Part c (2 points): Fill in the VPNs and the PFNs of the two TLB entries.

VPN	PFN
0x30	0x1
0x80	0x2

Part d (3 points): Is the L1 cache write through or write back? (Circle one)

Write Through / (Write Back)

Explain

The write to 0x3003 is not written through to L2 or memory.

Part e (3 points): Is the L2 cache write through or write back? (Circle one)

Write Through / (Write Back)

Explain

The write to 0x100 at L2 in not written through to memory.

Part f (4 points): What is the associativity of the L1 cache?

1 Way(s)

Explain

Reading 0x8000 caused an eviction of 0x3000 (only other entry in set)

Part g (4 points): What is the minimum possible associativity of the L2 cache?

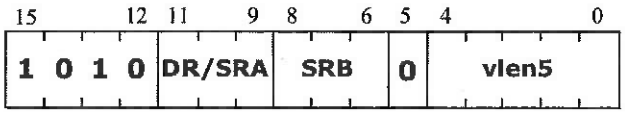
1 Way(s)

Explain

Without knowing more details of the L2, all accesses to L2 can be to different sets. This is a possible pattern with a direct mapped cache.

Name:_____

Problem 4 (20 points): Let us use one of the unused opcodes to add an instruction DOTPRODUCT (i.e., dot product) to the LC-3b ISA. Its format will be



The DOTPRODUCT of two vectors is computed as shown below:

$$\sum_{i=0}^{n-1} A[i] \times B[i]$$

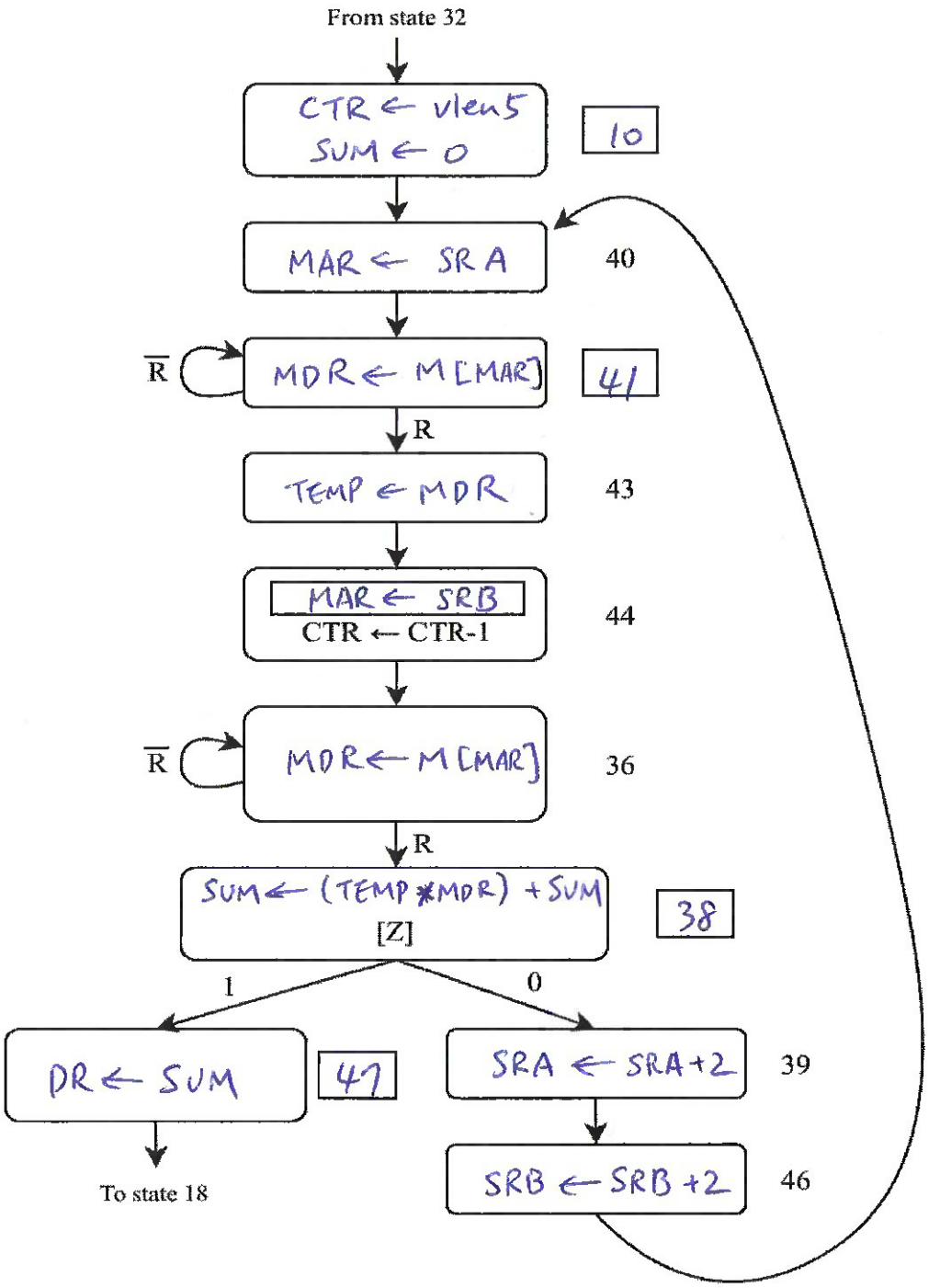
The two vectors are stored in memory. Their starting addresses are contained in SRA and SRB, and their length is specified as an immediate 5-bit value (vlen5). The instruction stores the result of the dot product in the register specified by DR. Assume vlen5 is not zero.

For this problem, you can assume no overflow will occur. Note: execution of this instruction will destroy the initial contents of SRA and SRB.

Your job: augment the LC-3b state machine, data path and microsequencer shown on the next three pages to add DOTPRODUCT to the LC-3b ISA.

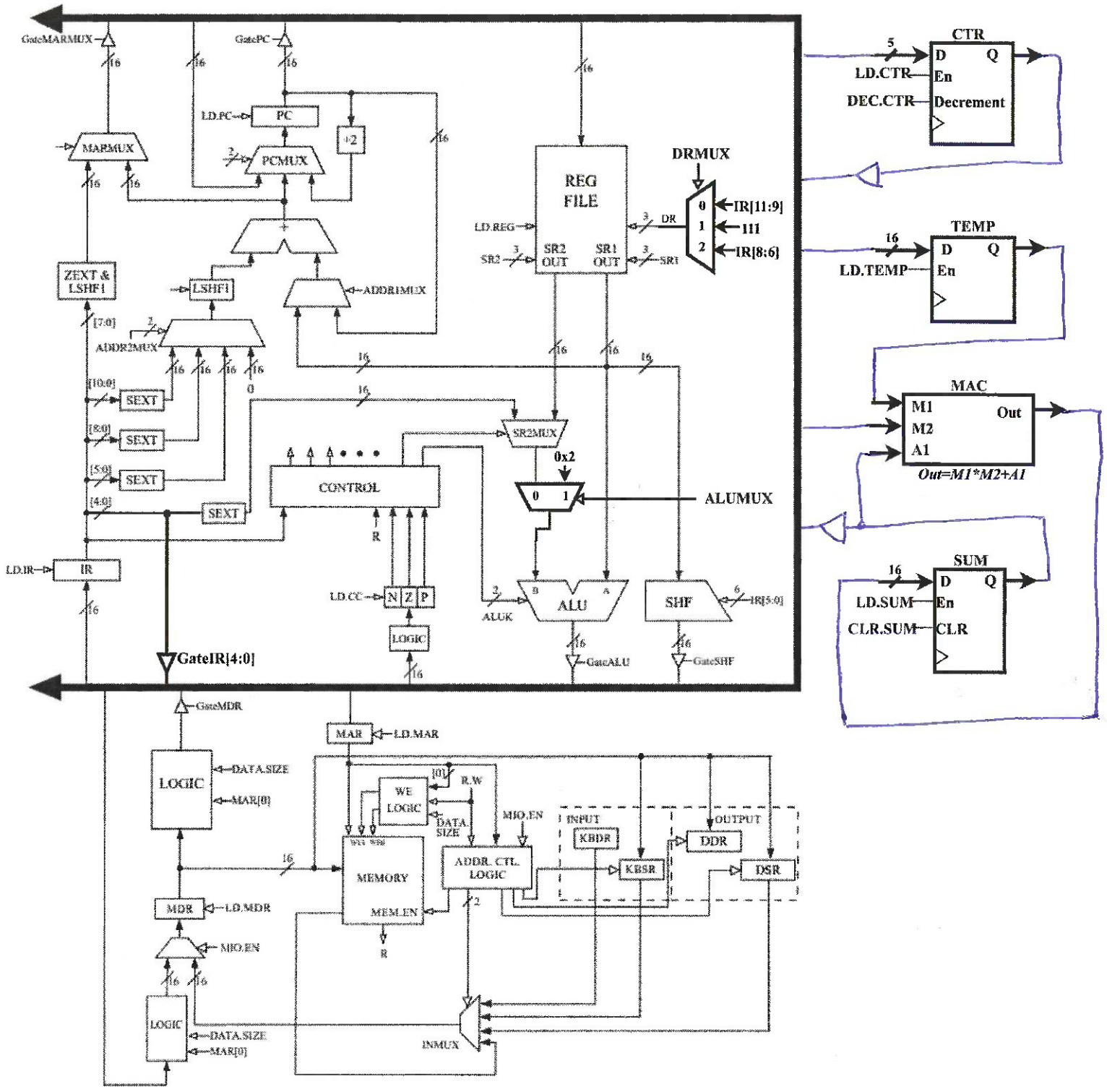
Name:_____

Part a, The state machine (10 Points): From decode (state 32), ten states are needed to complete the execution of DOTPRODUCT. One of the states (state 44) has been partially specified. Your job is to complete the specifications of all the states and add the missing state numbers.



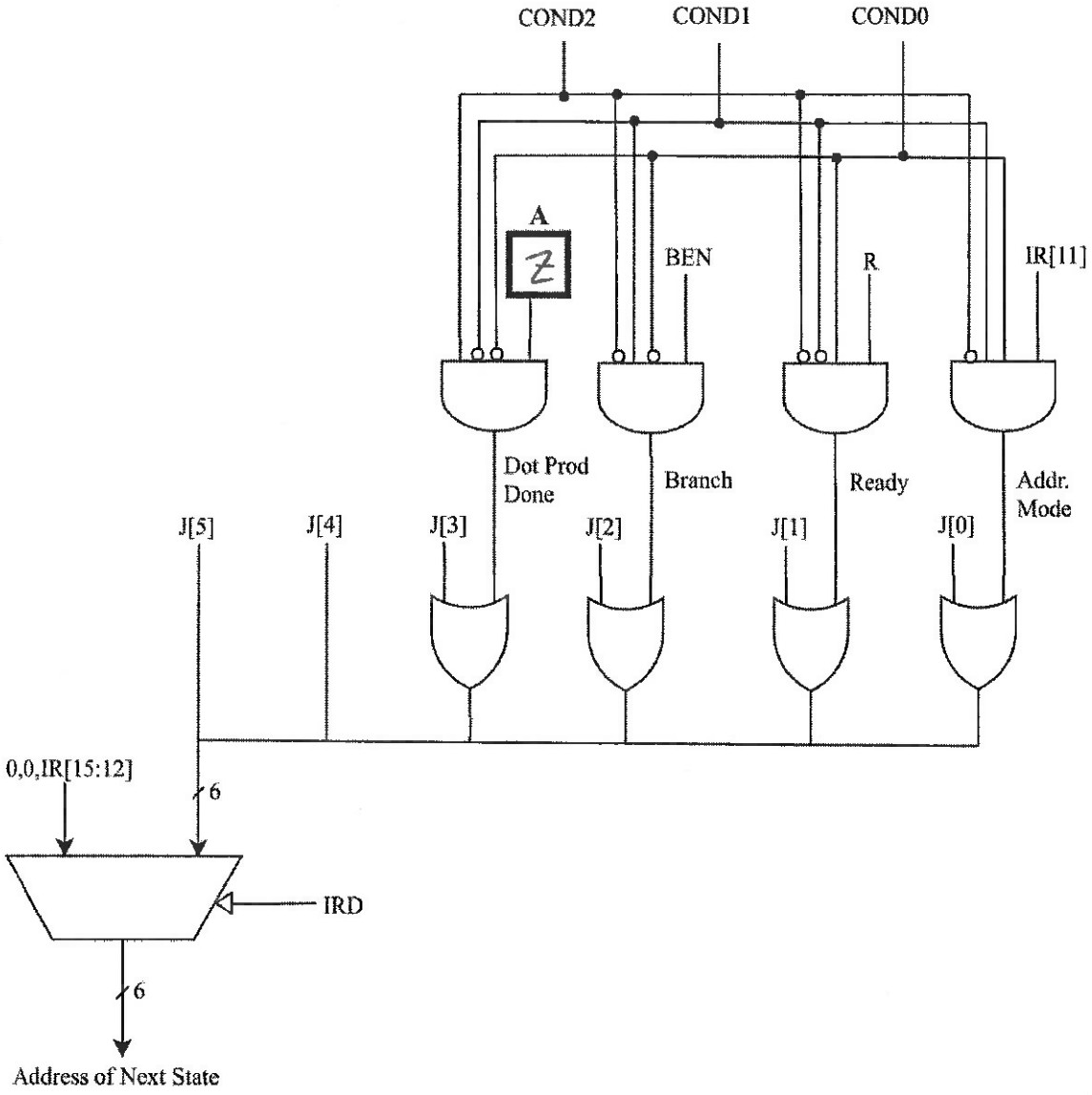
Name:_____

Part b, The data path (7 Points): We have added GateIR[4:0], ALUMUX, made changes to DRMUX, provided registers for CTR (with built-in decrement functionality), TEMP, and SUM as well as a multiply-and-accumulate (MAC) unit. The MAC computes $M1 \times M2 + A1$. Your job is to implement the changes you made in Part a by connecting the necessary structures to the LC-3b datapath. You are free to add control signals and tri-state buffers as needed.



Name:_____

Part c, The microsequencer (3 Points): To make this work, we need to add a COND2 control signal to the micorsequencer. The only thing missing to complete the change to the microsequencer is the box labeled A. Your job: fill in the box labeled A.



Name:_____

Problem 5 (20 points): Suppose the LC-3b ISA had a 12-bit, byte-addressable, virtual address space with two levels of virtual to physical translation, similar to the VAX.

A PTE is shown below:

It includes a Valid bit, a Modify bit, a 2 bit Access Control field, some number of unused bits (i.e., 0..0), and the PFN. The low bits of the PTE are used for the PFN.

The access control bits are defined as follows:

00: none
01: read-only
10: read-write
11: —

The virtual address space is divided evenly into two regions. The first half is user space, the second half is system space.

The user space page table starts at the beginning of a page. The system space page table starts at the beginning of a frame. We require 1/4 of physical memory to store the entire system page table.

A user program fetches and executes one LC-3b instruction, resulting in six accesses to physical memory, as shown by the following table:

VA	PA	Data
—	x00EA	x9...4
x08B0	x090	x9...2
x0100	x040	x2862
—	x00EE	x9...5
x08F6	x0B6	x9...1
x0572	x032	x10A0

Note: Since the size of the PTE has not been given, entire PTEs are not shown in the above table.

Part a (10 points): Fill in the entries for the following (UBR=User Base Register):

Physical Address Space: 9bit (512B)

PTE Size: 2 B

Page Size: 32 B

SBR: 0xE0

UBR: 0x8A0

Name:_____

Part b (5 points): After the instruction is executed, the register file is as shown:

Register	Value
R0	x0550
R1	x0590
R2	x00A0
R3	x0200
R4	xFFA0
R5	x000C
R6	x0010
R7	x0100

What LC-3b instruction was executed?

LDB R4, R1, 0x1E
(#-30)

Part c (5 points): Complete the entries in the memory access table shown on the previous page.

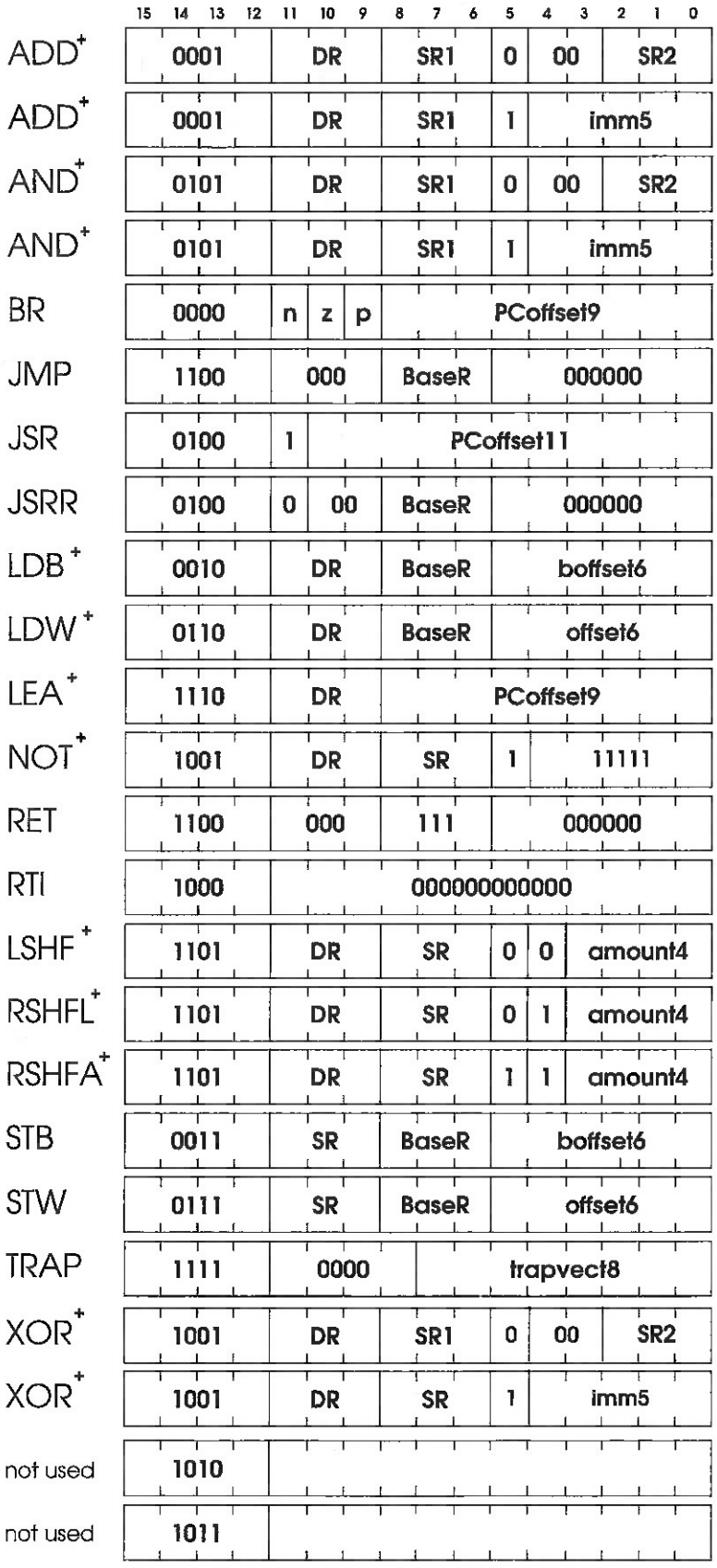


Figure 1: LC-3b Instruction Encodings