Name:_____

Problem 1 (25 points):

Part a (5 points): The main element of storage required to store a single bit of information depends on whether we are talking about DRAM cells or SRAM cells.

For DRAM cells it is: **a capacitor**

For SRAM cells it is: **a latch**

Part b (5 points):

The primary purpose of segmentation is: **protection**

The primary purpose of paging is: **virtual memory**

Part c (5 points): The reference bit in a PTE is used for what purpose?

page replacement

The similar function is performed by what bit or bits in a cache's tag store entry?

LRU/pseudo-LRU bits

Part d (5 points): We note that condition codes get set by the three load instructions and the four operates in the last cycle of the instruction cycle when they load the destination register. So, someone suggested we get rid of the LD.CC control signal and use instead the LD.REG signal to load condition codes, If we did this, without changing anything else, would the LC-3b work correctly? Why/why not?

No, it won't. TRAP, JSR/JSRR load the destination register without setting the condition codes.

Part e (5 points): A cache has the block size equal to the word length. What property of program behavior, which usually contributes to higher performance if we use a cache, does not help the performance if we use THIS cache?

spatial locality

Name:_____

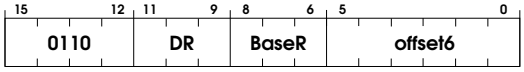
Problem 2 (20 points):

Little Computer Inc. has decided to support unaligned accesses in the LDW instruction. The specification of the LDW instruction is as follows:

Assembler Format

LDW DR, BaseR, offset6

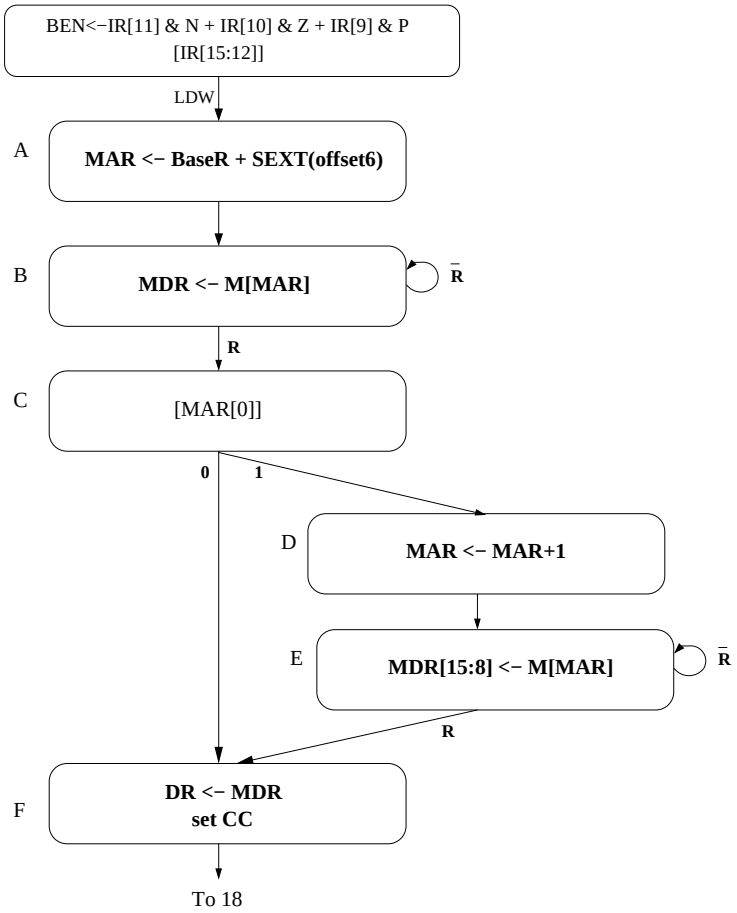
Encoding



Operation

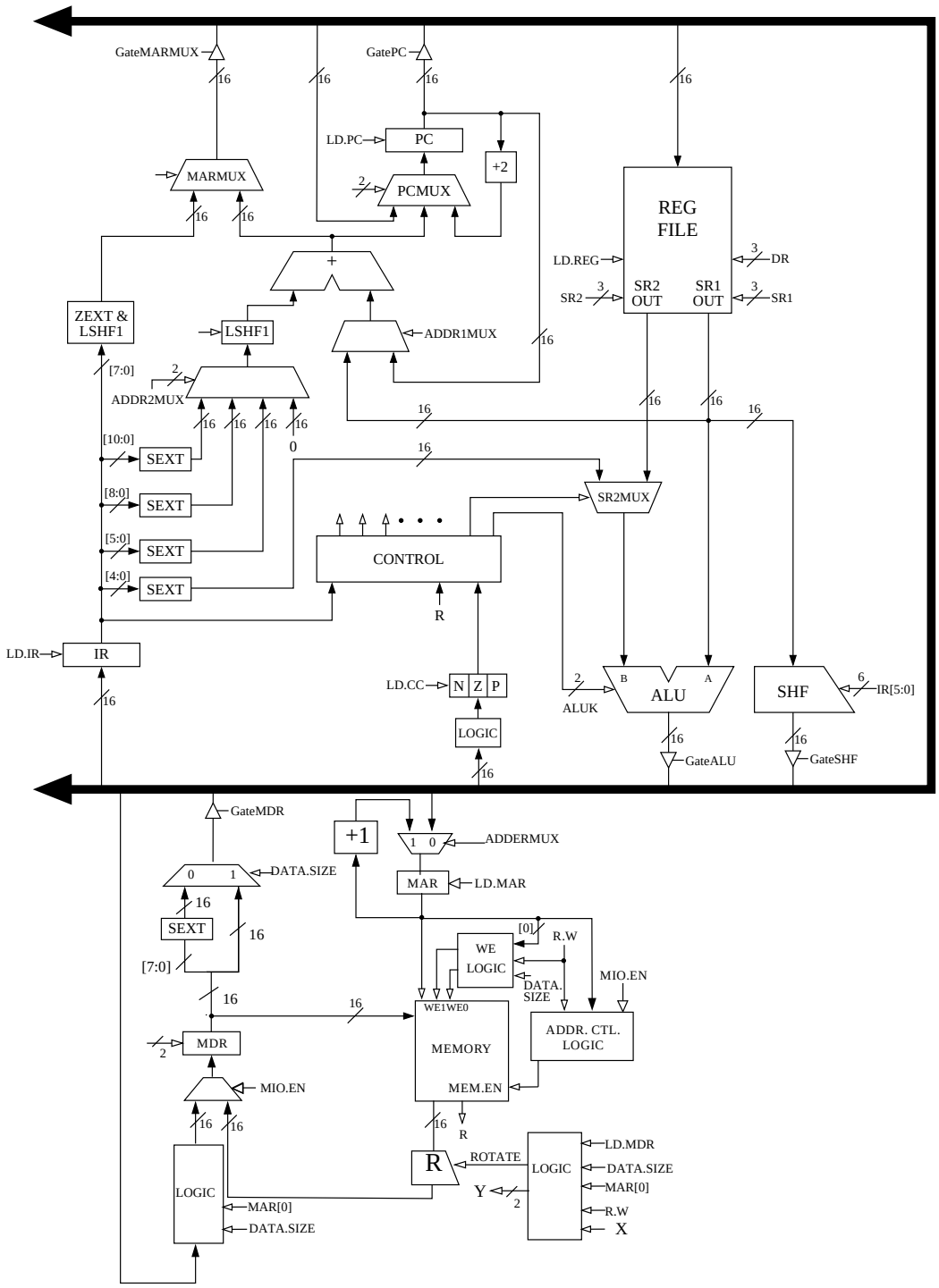
DR = MEM[BaseR+SEXT(offset6)];
setcc(DR);

Part a. (5 points) We show below the states used to implement the LDW instruction. Using the notation of the LC-3b state diagram, describe inside each "bubble" what happens in each state. We have already given you what happens in state C. In this state, MAR[0] is tested and next state is determined based on the value of MAR[0]. **The modified datapath is shown on the next page.**



Name:_____

Problem 2 continued:



Name:_____

Problem 2 continued:

Part b. (8 points) The modified datapath shown on the previous page contains a logic block whose inputs are LD.MDR, DATA.SIZE, R.W, MAR[0], and X. The outputs of this logic block are the two-bit signal Y and a 1-bit ROTATE signal. Identify precisely in the boxes below the signals X, Y[0], and Y[1]. Four or five words should be more than enough for each signal. Identify the specific value for X in each input combination of the truth table. Complete the output columns of the truth table.

Signal X: **first/second access**

Signal Y[0]: **LD.MDR low byte**

Signal Y[1]: **LD.MDR high byte**

Correct answer:

R.W	DATA.SIZE	LD.MDR	MAR[0]	X	Y[1]	Y[0]	ROTATE
READ	BYTE	NO	0	**x**	**0**	**0**	**x**
READ	BYTE	NO	1	**x**	**0**	**0**	**x**
READ	BYTE	LOAD	0	**x**	**x**	**1**	**0**
READ	BYTE	LOAD	1	**x**	**x**	**1**	**1**
READ	WORD	NO	0	**x**	**0**	**0**	**x**
READ	WORD	NO	1	**x**	**0**	**0**	**x**
READ	WORD	LOAD	0	**0** **1**	**1** **1**	**1** **0**	**0** **1**
READ	WORD	LOAD	1	**x**	**x**	**1**	**1**

Acceptable answer:

R.W	DATA.SIZE	LD.MDR	MAR[0]	X	Y[1]	Y[0]	ROTATE
READ	BYTE	NO	0	**x**	**0**	**0**	**x**
READ	BYTE	NO	1	**x**	**0**	**0**	**x**
READ	BYTE	LOAD	0	**x**	**x**	**1**	**0**
READ	BYTE	LOAD	1	**x**	**x**	**1**	**1**
READ	WORD	NO	0	**x**	**0**	**0**	**x**
READ	WORD	NO	1	**x**	**0**	**0**	**x**
READ	WORD	LOAD	0	**x**	**1**	**1**	**0**
READ	WORD	LOAD	1	**0** **1**	**x** **1**	**1** **0**	**1** **1**

Name:_____

Problem 2 continued:

Part c. (7 points) The processing in each state (A, B, C, D, E, F) 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, D, E, F. The control signals for state C are already filled in for you.

	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	ADDERMUX (BUS(0), MAR+1(1))	X
state A	1										1				1	1	0 1	1							
state B		1																		1		1			
state C	0	0	0	0	0	0	0	0	0	0	0	0	0 0	0	0	0	0 0	0	0 0	0	0	0	0	0	0
state D	1																							1	
state E		1																		1		1			1
state F					1	1			1													1			

Entries left blank are 0.

Name:_____

Problem 3 (20 points):

We hired a new circuit designer from A&M to help us implement the LC-3b, and he loaded the microinstructions into the wrong control store locations, as noted on the state machine shown in Figure 1. No problem, we can fix it with some quick fixes to the microsequencer. Figure 2 identifies the "new" microsequencer.

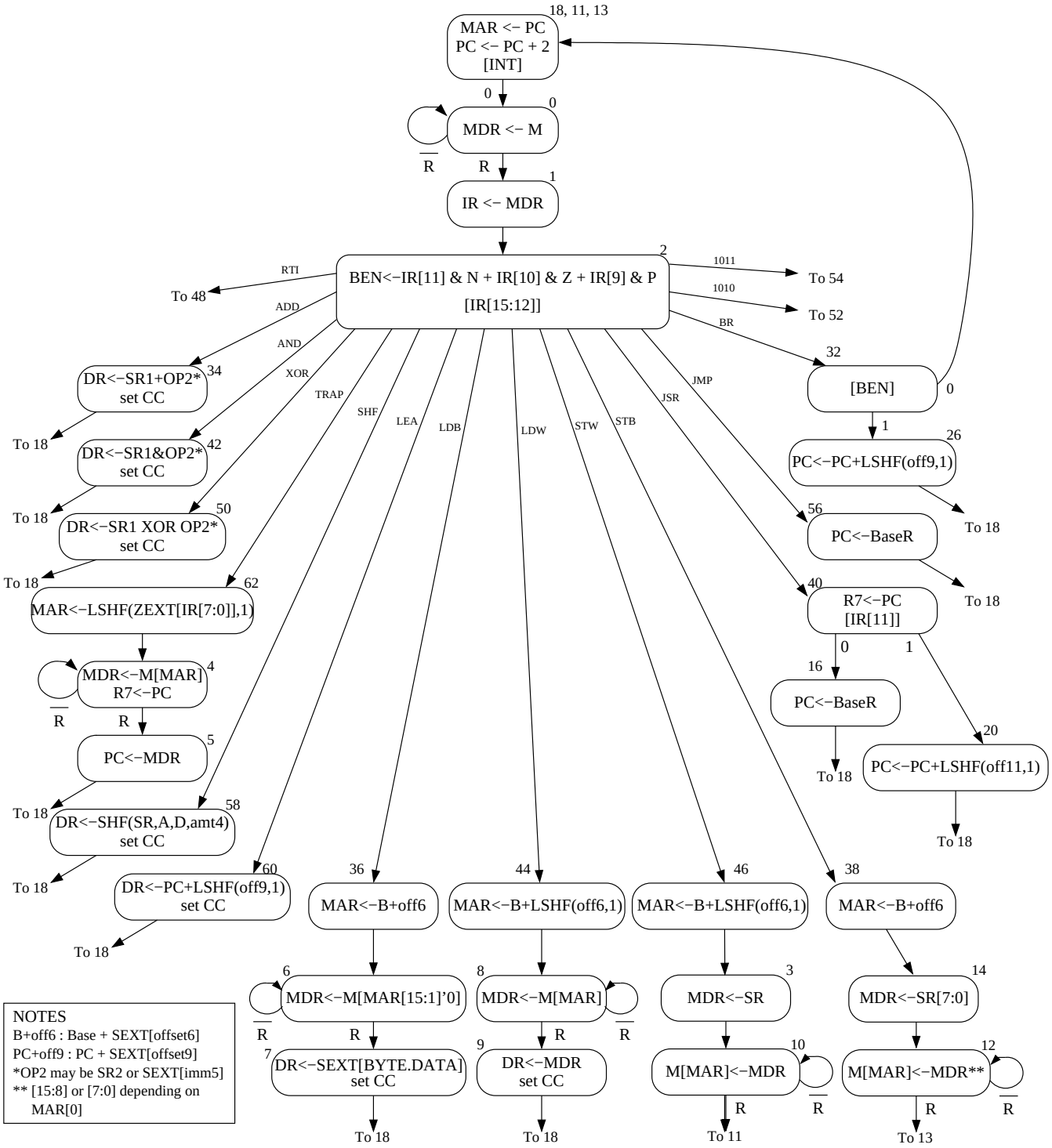


Figure 1

Name:_____

Problem 3 continued:

Cond[1:0]	Signal Values
00	Unconditional
01	Branch
10	Ready
11	Addressing Mode

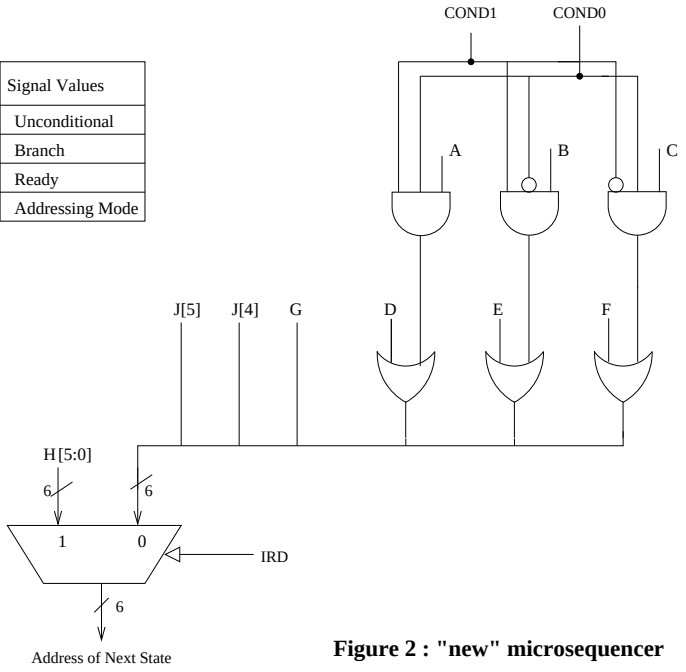


Figure 2 : "new" microsequencer

Part a. (10 points) Identify the signals A through G in the boxes provided below. A few words at most should suffice for each box.

A	**IR[11]**
B	**R**
C	**BEN**
D	**J[2]**
E	**J[0]**
F	**J[3]**
G	**J[1]**

Part b. (6 points) Identify separately each bit of H[5:0].

H[5]	H[4]	H[3]	H[2]	H[1]	H[0]
1	**IR[15]**	**IR[14]**	**IR[13]**	**IR[12]**	**0**

Part c. (4 points) In which state / states is IRD asserted? **2**

Name:_____

Problem 4 (15 points):

An LC-3b system ships with a two-way set associative, write back cache with perfect LRU replacement. The tag store requires a total of 4352 bits of storage. What is the block size of the cache? This is one problem where you really do need to show all your work on the paper.

Hint: 4352 = $2^{12} + 2^{8}$.

4 bytes

Solution:

address = 16 bits
bits for identifying byte in block = b bits
bits used for index = i bits
tag bits = 16 - b - i bits

contents of tag store entry for a set: 1 LRU bit, 1 valid bit per block, 1 dirty bit per block, 1 tag per block
size of tag store entry for a set = 1 + 2 x (2 + 16 - b - i)

tag store size = 2^{i} x (1 + 2 x (2 + 16 - b - i))

$2^{12} + 2^{8} = 2^{i}$ x (1 + 2 x (2 + 16 - b - i))
2^{8} x (1 + 16) = 2^{i} x (1 + 2 x (2 + 16 - b - i))

i = 8, b = 2
block size = 2^{b} = 4 bytes

Name:_____

Problem 5 (20 points):

A machine with 64KB, byte addressable virtual memory and 4KB physical memory has two-level virtual address translation similar to the VAX. The page size of this machine is 256 bytes. Virtual address space is partitioned into the P0 space, P1 space, system space and reserved space. The space a virtual address belongs to is specified by the most significant two bits of the virtual address, with 00 indicating P0 space, 01 indicating P1 space, and 10 indicating system space. Assume that the PTE is 32 bits and of the format 10000000..000PFN.

For a single load instruction the physical memory was accessed three times. The first access was at location x108 and the value read from that location (x108, x109, x10A, x10B) was x80000004. Hint: What does this value mean?

The second access was at location x45C and the third access was at location x942.

If SBR = x100, P0BR = x8250 and P1BR = x8350,

Part a. (7 points) What is the virtual address corresponding to physical address x45C ?

VA = x **825C**

Part b. (6 points) What is 32 bit value read from location x45C ?

Value = x | **8** | **0** | **0** | **0** | **0** | **0** | **0** | **9** |

Part c. (7 points) What is the virtual address corresponding to physical address x942 ?

VA = x **0342**