

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

EE 460N Spring 2011
Y. N. Patt, Instructor
Faruk Guvenilir, Milad Hashemi, Yuhao Zhu, TAs
Final Exam
May 13, 2011

Name: Solution

Problem 1 (25 points): 25

Problem 2 (10 points): 10

Problem 3 (10 points): 10

Problem 4 (25 points): 25

Problem 5 (25 points): 25

Problem 6 (25 points): 25

Total (120 points): 120

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.

Please sign the following. I have not given nor received any unauthorized help on this exam.

Signature: Solution

GOOD LUCK!

Name: Solution

Problem 1 (25 points)

Part a (4 points): We wish to detect errors that occur in the transmission of data from A to B by means of even parity.

Suppose we wish to transmit the byte 01010011, what would we additionally transmit to be able to detect single bit errors in a byte of data. Explain in 20 words or fewer.

1 Extra bit with value ϕ
 Even Parity \rightarrow even number of ones
 in transmission

Part b (4 points): In the x86 virtual memory structure, the base address of the Page Directory is contained in a register that is part of the TSS. What is the name of that register?

CR3

Part c (5 points): For each parameter, label the interconnection structure(s) that is (are) best and the one(s) that is (are) worst by putting a B or a W in the appropriate entries.

	Cost	Contention	Latency
Bus	B	W	B
Omega Network			W
Crossbar	W	B	B

Part d (5 points): If a two-dimensional matrix of size 10 rows by 14 columns is stored in row major order, and one wanted to perform a vector load of column 9 into vector register V2, one would have to be sure of what before executing the vector load instruction?

Vlength: 10
 Vstride: 14

Part e (7 points): A tag store entry of a physically indexed, physically tagged direct-mapped, write-through 8KB cache consists of 12 bits. Assume the cache is used in a uniprocessor environment and is not sectored.

How many LRU bits in each tag store entry: 0

How many dirty bits in each tag store entry: 0

How big is the physical address space: 2^{24} locations

NO LRU necessary
 no dirty
 Assume Byte Addressable.

8KB = 2^{13} locations
 12 byte entry size - 1 valid bit
 = 11 tag bits

Name: Solution

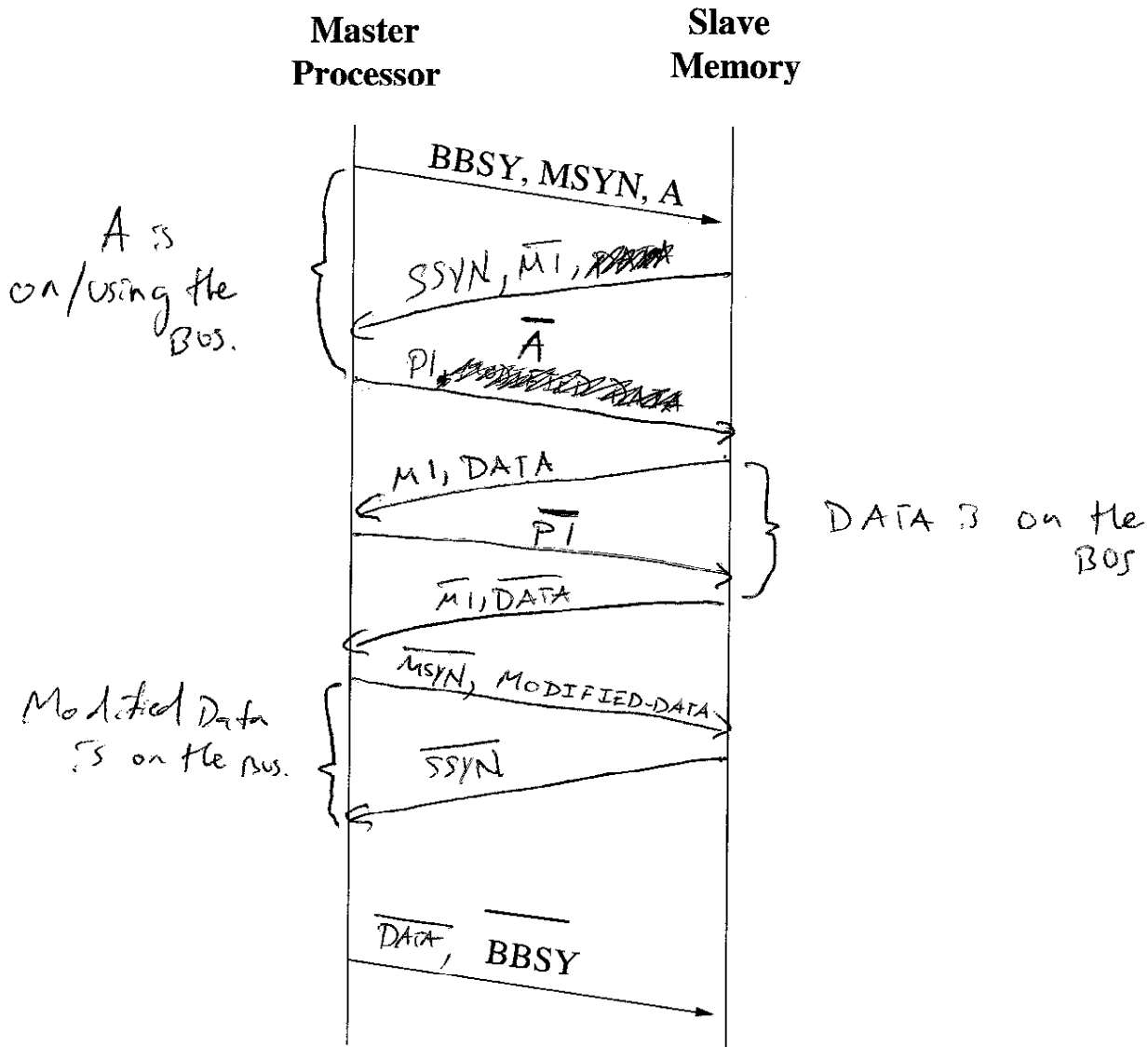
Problem 2 (10 points)

Recall the asynchronous bus we described in class. Among the devices on the bus are a processor connected via its Processor Bus Controller and memory connected via its Memory Controller. Assume the bus is pending and multi-plexed.

The processor can use the bus to perform a "read-modify-write" operation on a memory location, as follows: The processor reads a value from memory, performs a computation on the value, and writes a new value to the same location in memory, all within a single bus cycle.

To do this, two new signals are required: PI, a new control signal from the processor bus controller, and M1, a new control signal from the memory controller.

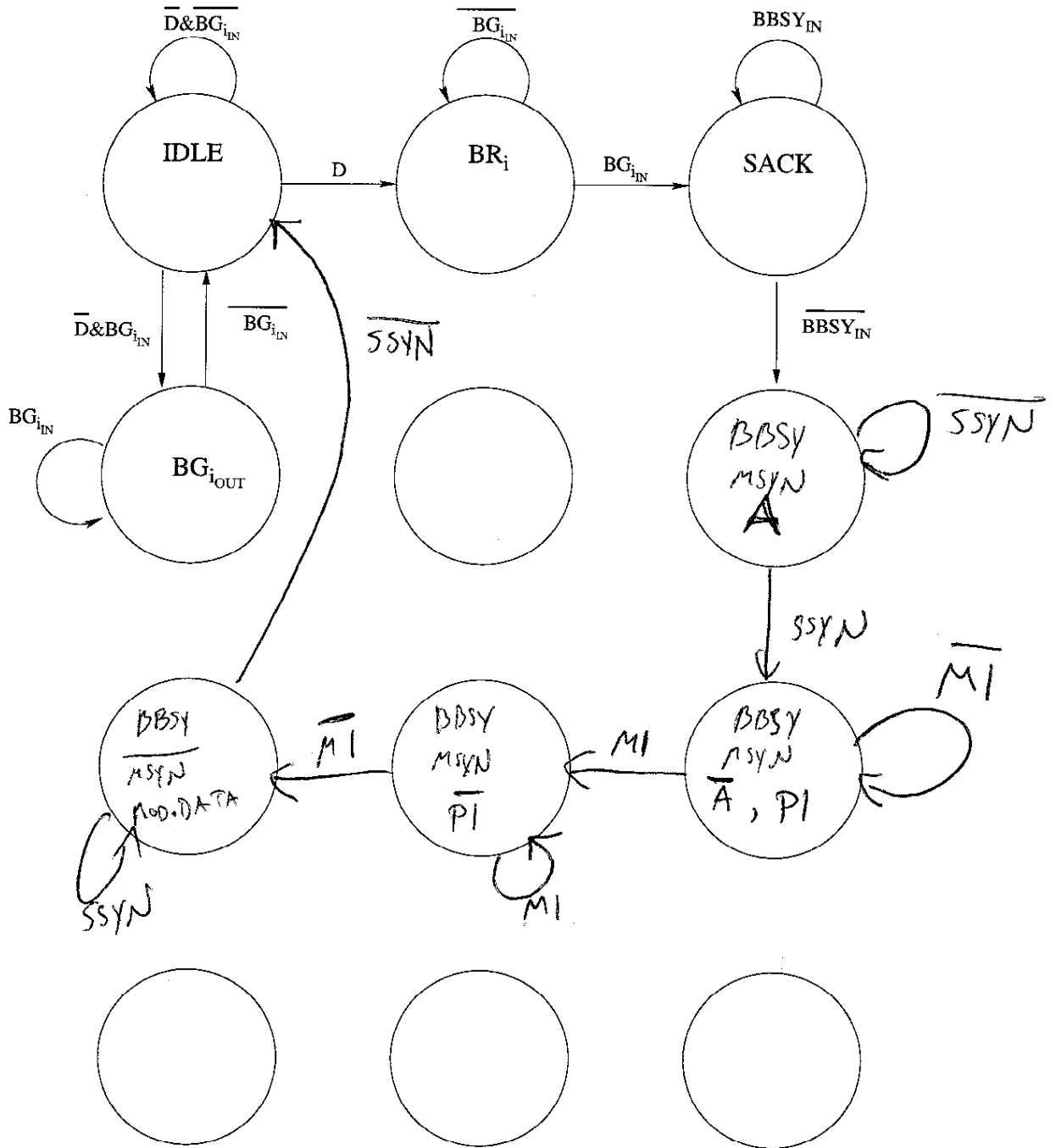
Part a: Complete the transaction timing diagram for the read-modify-write operation.



Name: Solution

Problem 2 Continued

Part b: Complete the state machine for the Processor Bus Controller to complete the read-modify-write operation. Note that the controller must handle bus arbitration as well as processing the transaction. Some of the states have been filled in for you. You may not need all of the states provided.



Name: Solution

Problem 3 (10 points)

As you know the IEEE Floating Point standard has a 32-bit representation and a 64-bit representation. In this problem we assume IEEE decided to add a 10-bit representation, with its main characteristics consistent with the other two representations.

In this 10-bit representation, the value $33/256$ is represented exactly as $00011|00001$

Part a: Determine the number of bits for

exponent: fraction: bias:

Note: Bias is another word for n in an excess-n code.

Part b: What is the value of an ULP in the smallest normalized positive binade?

$$\frac{33}{256} = \frac{1}{256} + \frac{32}{256} = \frac{1}{256} + \frac{1}{8} = 0.00100001 \times 2^0 = 1.00001 \times 2^{-3}$$

$$3 = n = -3$$

$$n = 6$$

$$0 | 0011 | 0000 \rightarrow 1.0001 \times 2^{3-6} \Rightarrow 1.0001 \times 2^{-3} \rightarrow .00100001 \times 2^0$$

\uparrow \uparrow
 $\frac{1}{8}$ $\frac{1}{256}$

$$0 | 0001 | 0000 \frac{1}{\text{ULP}} \rightarrow 1.0001 \times 2^{1-6} \rightarrow 1.0001 \times 2^{-5} \rightarrow 0.0000100001 \times 2^0$$

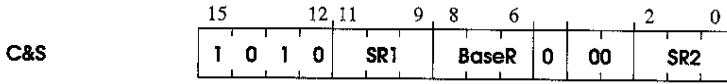
$\frac{1}{2}$ $\frac{1}{8}$ $\frac{1}{16}$ $\frac{1}{32}$ $\frac{1}{64}$ $\frac{1}{128}$ $\frac{1}{66}$ $\frac{1}{66}$
 $\frac{1}{1024}$

Name: Solution

Problem 4 (25 points):

We wish to extend the ISA of the LC-3b to include a Compare and Set instruction, C&S, which uses one of the unused opcodes 1010.

The format of the C&S instruction is:



The operation of the instruction is:

```

IF (M[BaseR] == SR2)
    Z = 1;
    M[BaseR] = SR1;
ELSE
    Z = 0;
  
```

As one word-sized.

That is, if the contents of the memory location specified by BaseR is equal to the contents of SR2, Z is set to 1 and the contents of SR1 are written to the memory location. If they are not equal, Z is set to 0. Note that when Z is set to 0, either N or P must be set to 1. When Z is set to 1, N and P are set to 0.

To implement C&S, we need 5 additional microinstructions, a few changes to the data path, and a small change to the microsequencer. We will deal with each of these in turn.

Part a: The modified datapath (changes in bold) is shown below. Complete the datapath changes by filling in the box with the simple combinational logic necessary to support C&S.

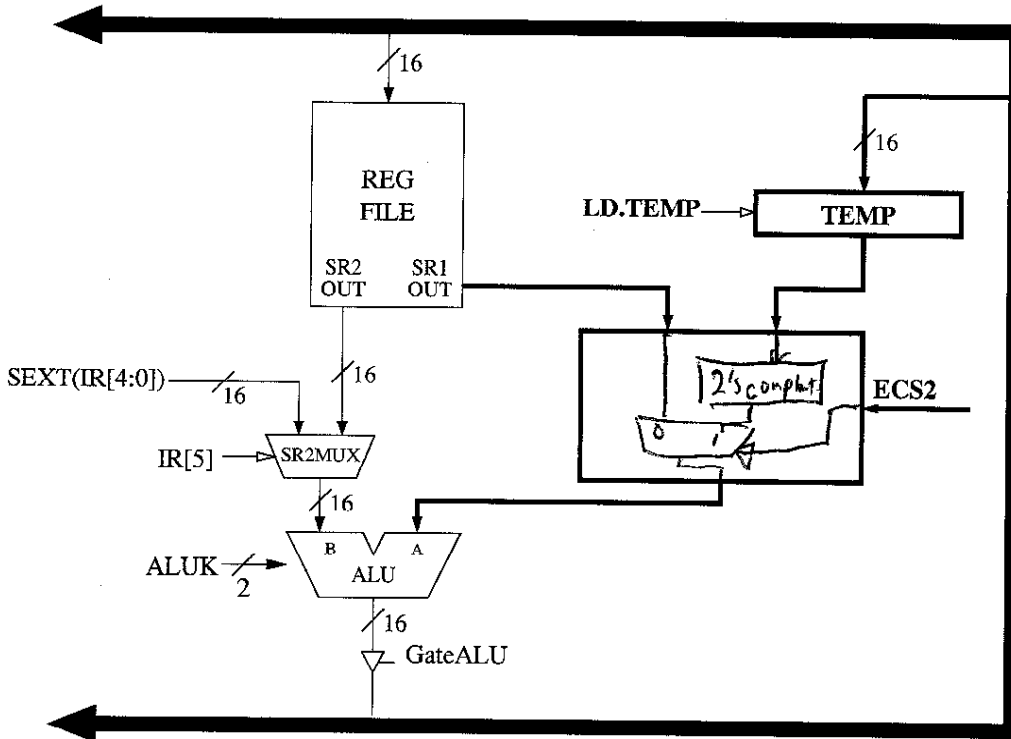
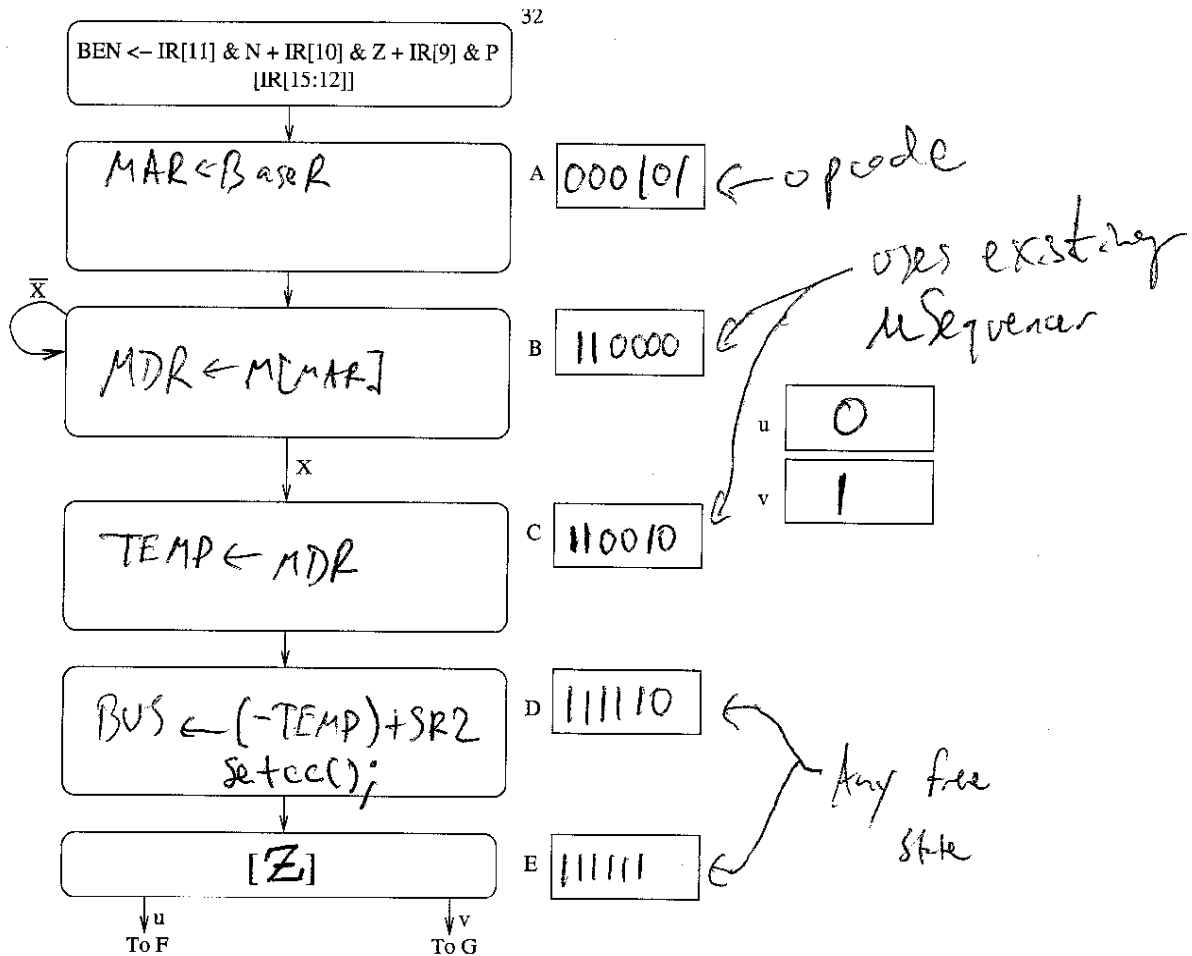


Figure 1: Modified datapath to support C&S instruction

Name: Solution

Problem 4 continued:

Part b: Five additional states (A, B, C, D, E) are needed to implement C&S. Complete the description of these states, including the binary representations for each state number.



What is the signal X? Memory Ready Bit

Name: Solution

Problem 4 continued:

Part d: Relevant parts of the microinstructions for states A, B, C, D and E are shown below. Fill in the table with the appropriate values. If a control signal is a don't care, enter a 0.

Note: Please use the encodings specified in the control signals attachment for all signals.

	LD.MAR	LD.MDR	LD.CC	LD.TEMP	GateMDR	GateALU	DRMUX	SRIMUX	ALUK[1:0]	MIO.EN	R.W	DATA.SIZE	ECS1	ECS2
A	1	0	0	0	0	1	0	1	1	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0	1	0	1	0	0
C	0	0	0	1	1	0	0	0	0	0	0	1	0	0
D	0	0	1	0	0	1	0	0	0	0	0	0	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Name: Solution

Problem 5 (25 points)

A byte-addressable, write-back cache of fixed total size and fixed line size is implemented as both a direct mapped cache and also as an N-way set-associative cache. In both cases, we will assume the cache is initially empty.

First, consider the cache organized as a direct mapped cache. The following sequence of 11 accesses generates the hits/misses shown.

No conflict in between these two accesses, allows the hit

Conflict! Causes eviction.

Address	R/W	Direct Mapped (Miss/Hit)
010000010	R	MISS
110010011	R	Miss
110101000	R	Miss
0011000101	R	MISS
011011100	R	MISS
1010110101	R	Miss
110010000	R	Miss
010001111	R	Hit
010111111	W	Miss
0110110100	R	MISS
0110100101	R	Miss

Note that high bits are identical, but still a miss

Hits because of this need at least 3 offset bits

Restricts offset bits to at most 4

Part a: What is the cache line size? 16

Explain why. Please be concise but clear

The only hit forces at least 3 offset bits.
 The last miss forces at most 4 offset bits.
 The access immediately before hit can only be a miss if third access causes conflict eviction with 3 tag bits, 3 index bits, 4 offset bits.

Part b: What are the number of index bits for the direct mapped cache? 3

Explain why. Please be concise but clear.

See above.

Name: Solution

Fixed total size!
 (see prev. page.)
 #I bits must be ~~less~~
 fewer than in part a.

Problem 5 Continued

Now consider the cache organized as a N-way set-associative cache.

The total size of the tag store for the N-way set associative cache is 112 bits. Each tag store entry contains, in addition to the tag bits, 10 more bits which include the valid bit, modified bit, LRU bits, snoop cache bits and other bits whose functions do not affect this problem.

$(10+T) \cdot 2^I \cdot N = 112, T+I=6$

We have expanded the table to show the hit/misses for the same sequence of accesses when the cache is organized as an N-way set-associative cache.

← known from part a.

Address	R/W	Direct Mapped (Miss/Hit)	N-way associative (Miss/Hit)
0100001010	R	MISS	MISS
1100001111	R	Miss	Miss
1110001000	R	Miss	MISS
0011000101	R	MISS	Miss
0110111100	R	MISS	Miss
1010110101	R	Miss	MISS
1100100000	R	Miss	HIT
0100001111	R	Hit	HIT
0101111111	W	Miss	Miss
0110110100	R	MISS	Hit
0110100101	R	Miss	MISS

Notice the identical index bits.

Hits because of this

Part c: What is N?

2

Because of fixed total size,
 if I=2, N=2
 if I=1, N=4
 if I=0, N=8

Explain why. Please be concise but clear.

$(10 + 4) \cdot 2^2 \cdot 2 = 112$
 $4 \cdot 2 = 6$
 Solved system of equations above.

$2^I \cdot N$ must be 8

Part d: What is the number of index bits for the N-Way set associative cache?

2

Explain why. Please be concise but clear

See above.

Part e: Is the cache write allocate? Explain why. Please be concise but clear.

No. Hit can only be hit if write does not cause conflict/eviction.

Part f: Please complete the table above by filling in "H" or "M" for each of the blank entries.

Name: Solution

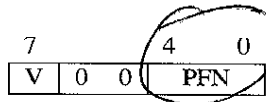
Problem 6 (25 points)

You are asked to examine the behavior of a new instruction STII R0,R1 on a two-level virtual memory system, similar to the VAX.

First, the characteristics of the memory system:

Virtual Address Space:	256 Bytes	Number of Page Frames:	32
User Space Range:	x00 to x7F	Page Size:	4 Bytes
System Space Range:	x80 to xFF	PTE Size:	1 Byte

The machine is byte addressable. The system includes a six-entry TLB. **The TLB is only used for User Space Pages.** The PTE format is as follows:



5 bits PFN $\rightarrow 2^5 = 32$ frames
 * 4 byte page size
 = 128 bytes

Part a: How many bytes of Physical Memory are there?

128 bytes

Next, the characteristics of STII R0, R1:

STII performs a double indirection on the address in R1 and stores R0 into that location. That is,

$$MEM_3[MEM_2[MEM_1[R1]]] \leftarrow R0$$

where $MEM_i[j]$ denotes the contents of the memory location whose address is j.

Part b: What is the maximum number of physical memory accesses that can take place to get the value $MEM_1[R1]$?

3

two level translation, 1 page walk. (Exclude Fetch.)

Part c: Let's consider one instance of the execution of STII R0,R1.

The state of the processor before this instruction is executed is partially shown below:

R0: xCD
 R1: x6D

The TLB:

V	PN	PTE	
		V	PFN
1	x0C	1	x10
1	x0D	1	x1F
1	x0F	1	x08
0	-	-	-
0	-	-	-
0	-	-	-

Name: Solution

Problem 6 continued

Memory:

User page table	x00	x60	x20	x66	x40	x6C	x60	x72	SPT	
	x01	x61	x21	x67	x41	x6D	x61	x91		
	x02	x62	x22	x81	x42	x6E	x62	x73		
	x03	x82	x23	x68	x43	x99	x63	x97		
	x04	x63	x24	x69	x44	x87	x64	x74		System page table
	x05	x64	x25	x98	x45	x6F	x65	x75		
	x06	x65	x26	x6A	UPT x46	x70	x66	x80		
	x07	x87	x27	x6B	x47	x71	x67	x76		

After the instruction is fetched and decoded, as many as 9 subsequent memory accesses could be required to process this instruction, if there are no page faults, interrupts, or other unhappy events.

The table below shows the sequence of physical memory accesses required to process STII R0,R1, given the machine state shown above.

Calculate
USER
SBR
to get
these
bitmasks;
These
need
to
match
TLB
H.T. PFN
Matches,

Access #	VA	PA	Data	Description
1	N/A	01100110 x66	x80	PTE for page of System Space Containing ↓
2	10101011 xAB	x03	x82	PTE for page containing M[R1] (U.S.) ↓
3	x6D	x09	x13	V. Address, M[R1]
4	N/A	x61	x91	PTE for page containing ↓ (S.S.)
5	10010100 x94	01000100 x44	x87	PTE for page containing M[R1] (U.S.) ↓
6	x13	00011111 x1F	x3D	V. Address of where to store Data
7	x3D	00100001 x21	xCD	Data Being Stored (R0) to M[M[R1]]
8				
9				

Part d: Is the first access to physical memory the result of a TLB hit?

No.

Your job: Complete the table.

PFN doesn't match TLB

Note: N/A is a potential answer for entries in the VA column. Data is the data Read or Written in the corresponding memory access.

In the Description column, state what is being read/written (e.g. PTE for page xx).

You may not need all 9 rows, so draw a line through the unused rows

Note that it is possible that you arrive at a Physical Address whose contents are not given to you. Despite this, it is possible to fill in every entry in the table.

Name: Solution

Problem 6 continued

Part e:

What is the value of the UBR (User Space Page Table Base Register)? x90

What is the value of the SBR (System Space Page Table Base Register)? x5C

$$SS.PN + SBR = PA \text{ of S.P.T entry}$$

$$U.S.PN + UBR = V.A. \text{ of U.P.T entry}$$

Access 5
x94 → 1001 0100

PN + SBR = x61 = 0110 0001

Access 2
xAB → 1010 1011

SBR = x61 - x5 = x5C

xA + x5C = x66

Access 3
x6D → 0110 1101

x1B + UBR = xAB

UBR = x90

? + x90 = x94

x4 = Access 6's Page Number
Access 6 is user. V.A, region bit = 0

0 | 00100 | 01 = x13

offset bit from PA.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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	PCoffset9									
JMP	1100			000			BaseR			000000						
JSR	0100			1	PCoffset11											
JSRR	0100			0	00		BaseR			000000						
LDB ⁺	0010			DR			BaseR			boffset6						
LDW ⁺	0110			DR			BaseR			offset6						
LEA ⁺	1110			DR			PCoffset9									
NOT ⁺	1001			DR			SR			1	11111					
RET	1100			000			111			000000						
RTI	1000			000000000000												
LSHF ⁺	1101			DR			SR			0	0	amount4				
RSHFL ⁺	1101			DR			SR			0	1	amount4				
RSHFA ⁺	1101			DR			SR			1	1	amount4				
STB	0011			SR			BaseR			boffset6						
STW	0111			SR			BaseR			offset6						
TRAP	1111			0000			trapvect8									
XOR ⁺	1001			DR			SR1			0	00		SR2			
XOR ⁺	1001			DR			SR			1	imm5					
not used	1010															
not used	1011															

Figure 2: LC-3b Instruction Encodings

Table 1: Data path control signals

Signal Name	Signal Values
LD.MAR/1:	NO(0), LOAD(1)
LD.MDR/1:	NO(0), LOAD(1)
LD.IR/1:	NO(0), LOAD(1)
LD.BEN/1:	NO(0), LOAD(1)
LD.REG/1:	NO(0), LOAD(1)
LD.CC/1:	NO(0), LOAD(1)
LD.PC/1:	NO(0), LOAD(1)
GatePC/1:	NO(0), YES(1)
GateMDR/1:	NO(0), YES(1)
GateALU/1:	NO(0), YES(1)
GateMARMUX/1:	NO(0), YES(1)
GateSHF/1:	NO(0), YES(1)
PCMUX/2:	PC+2(0) ;select pc+2 BUS(1) ;select value from bus ADDER(2) ;select output of address adder
DRMUX/1:	11.9(0) ;destination IR[11:9] R7(1) ;destination R7
SR1MUX/1:	11.9(0) ;source IR[11:9] 8.6(1) ;source IR[8:6]
ADDR1MUX/1:	PC(0), BaseR(1)
ADDR2MUX/2:	ZERO(0) ;select the value zero offset6(1) ;select SEXT[IR[5:0]] PCoffset9(2) ;select SEXT[IR[8:0]] PCoffset11(3) ;select SEXT[IR[10:0]]
MARMUX/1:	7.0(0) ;select LSHF(ZEXT[IR[7:0]],1) ADDER(1) ;select output of address adder
ALUK/2:	ADD(0), AND(1), XOR(2), PASSA(3)
MIO.EN/1:	NO(0), YES(1)
R.W/1:	RD(0), WR(1)
DATA.SIZE/1:	BYTE(0), WORD(1)
LSHF1/1:	NO(0), YES(1)

Table 2: Microsequencer control signals

Signal Name	Signal Values
J/6:	
COND/2:	COND ₀ ;Unconditional COND ₁ ;Memory Ready COND ₂ ;Branch COND ₃ ;Addressing Mode
IRD/1:	NO, YES

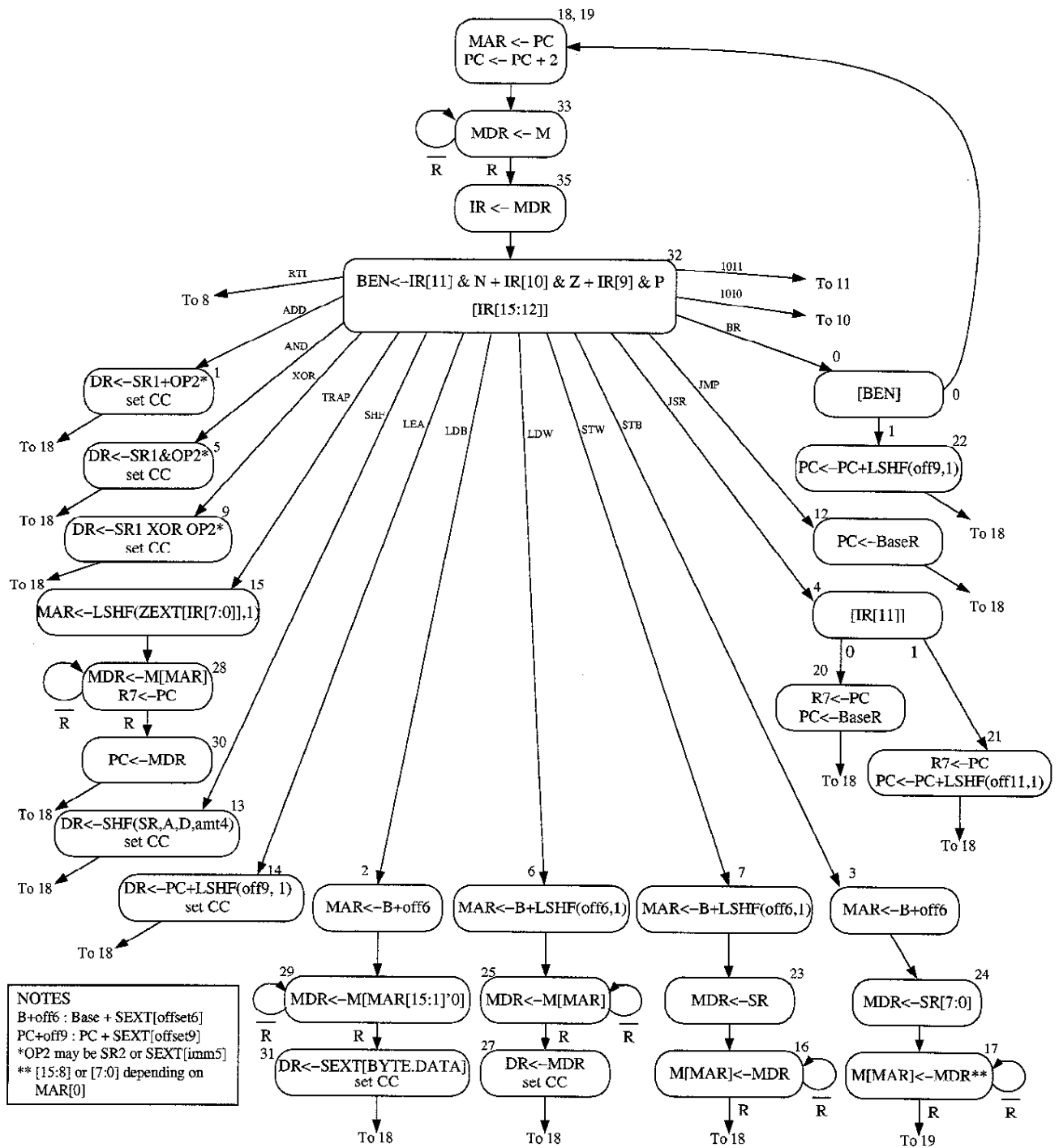


Figure 3: A state machine for the LC-3b

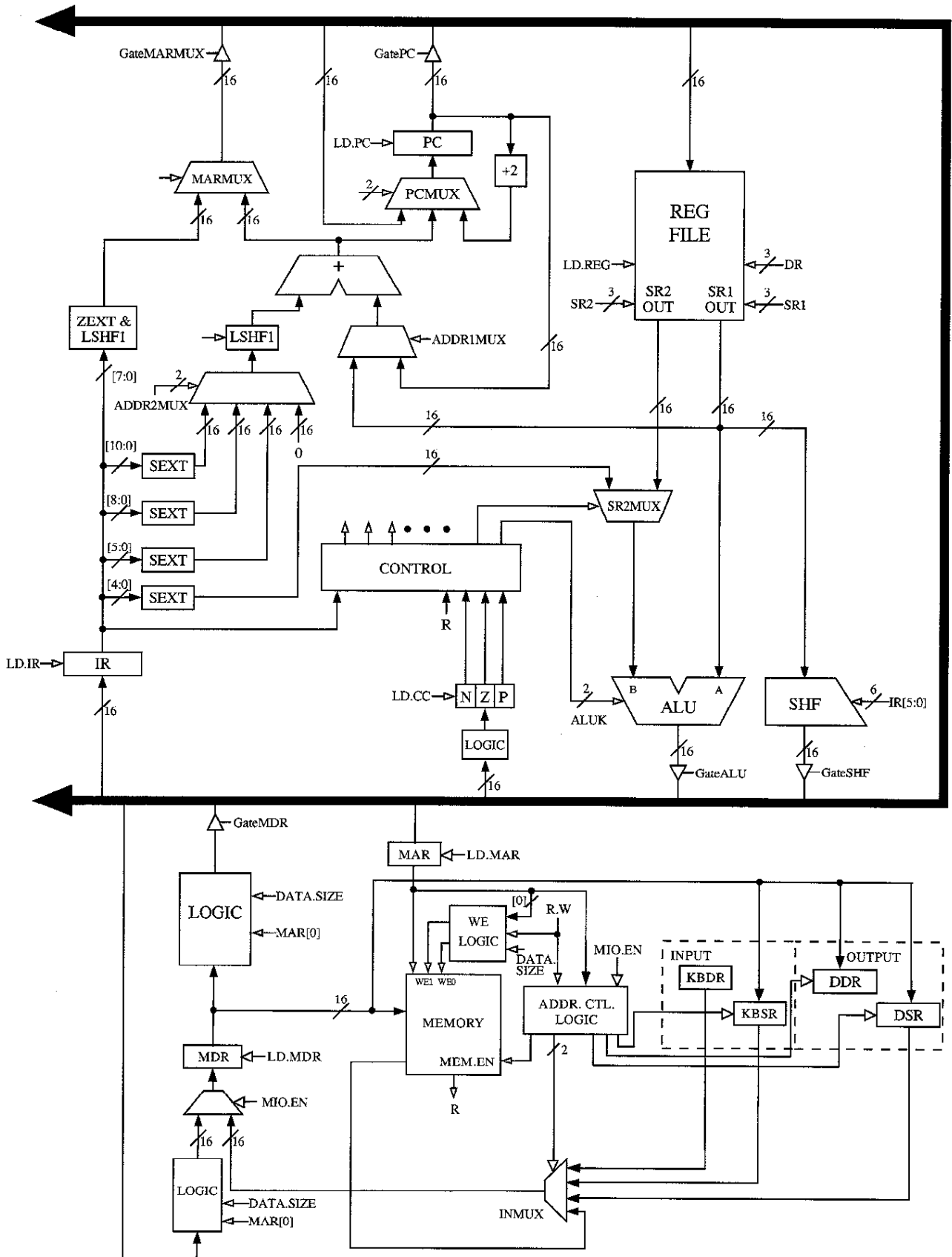


Figure 4: The LC-3b data path

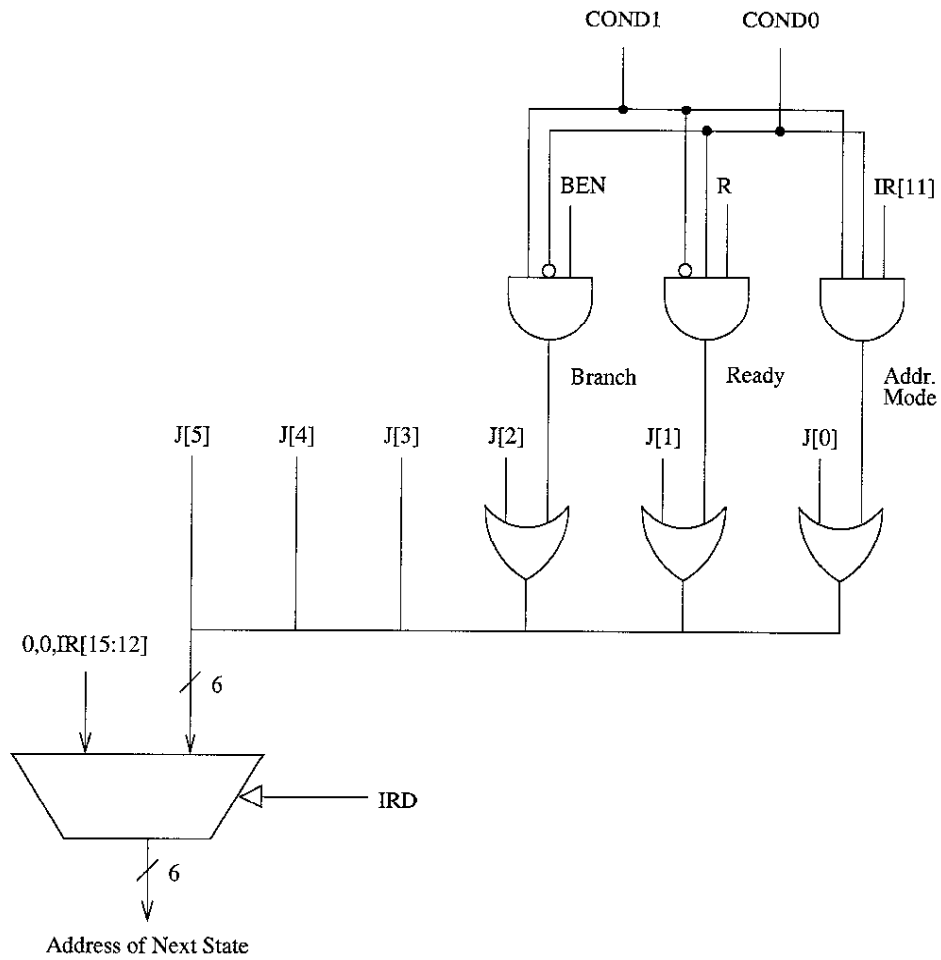


Figure 5: The microsequencer of the LC-3b base machine