

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

EE 360N, Fall 2004
Yale Patt, Instructor
Aater Suleman, Huzefa Sanjeliwala, Dam Sunwoo, TAs
Exam 2, November 10, 2004

Name: _____

Problem 1 (20 points): _____

Problem 2 (20 points): _____

Problem 3 (15 points): _____

Problem 4 (25 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. (20 points)

Part a (5 points): The SACK signal is a message from X to Y.

X: Y:

X asserts SACK, when

When Y receives SACK, it

Part b (5 points): Stripmining would be unnecessary if (in less than 15 words):

Part c (5 points): An ISA specification contains in part: a 32-bit virtual address space, 25-bit physical address space, and page size of 4KB. Byte-addressible. We implement a 8KB direct-mapped, virtually-indexed, physically-tagged cache. We do not restrict the mapping between virtual page number and physical frame number.

How many tag bits are required in each tag-store entry to make this work.

Part d (5 points): A four-way set associative cache contains 256 sets. Suppose we make exactly one change - that is, we change it to a sector cache where the size of each sector is 1/4 the size of the original block. The number of bits of storage needed to implement the tag store increases by bits.

Name: _____

Problem 2. (20 points) We wish to design a new ISA, similar to the LC-3b, but with certain differences (mainly for the benefit of this exam question!).

The ISA will have a 16 bit virtual address space, supported by a virtual memory management structure similar to the two-level translation scheme of the VAX. Virtual memory will be partitioned into three regions, allocating 3/4 of the virtual memory to process space, 1/8 to system space, and 1/8 left unassigned. VA[15:13] will specify the region of memory. VA[15:13] = 110 is system space, VA[15:13] = 111 is unassigned. The rest is Process Space.

The ISA specifies 4KB of physical address space, partitioned into page frames of 64 bytes each. PTEs are 16 bits. The Process Page Table is in system space, starting at address xC300. The System Page Table is in physical memory, starting at address x200.

The Assembly language for this ISA contains the instruction LDB R2,X. In executing LDB R2,X, the value loaded into R2 is obtained from physical address xA6D. What is the VA of X?

(Note: To successfully do this, you need to know that the PTE of the page containing X is stored in physical addresses x59C and x59D. Also, that physical addresses x224, x225 contain the PTE of the page in system space on which the PTE of the page containing X is stored.)

Name: _____

Problem 3. (15 points) A tree (triangle) replacement scheme is used to implement the pseudo LRU policy for a 4-way set-associative cache. We need 3 LRU bits per set to implement this scheme.

LRU[2]: specifies whether the last access came from Way0,Way1 or from Way2,Way3
(a 0 means Way0,Way1 was least recently accessed)

LRU[1]: specifies which of the two Way1 or Way0 was accessed least recently.
(a 0 means Way0 was least recently accessed)

LRU[0]: specifies which of the two Way2 or Way3 was accessed least recently.
(a 0 means Way2 was least recently accessed)

Part a (5 points) The state of LRU[2:0] determines which Way to replace on a cache miss. Complete the table below.

LRU[2:0]	Replace
0 0 0	Way 0
0 0 1	Way 0
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

An access to a Way (whether due to replacement after a miss, or to a hit) results in LRU[2:0] being updated. Complete the table below which specifies the state of LRU[2:0] as a result of an access to each Way.

Access to	LRU[2:0]
Way 0	1 1 -
Way 1	
Way 2	
Way 3	

(Note: The - in 11- means LRU[0] is left unchanged.)

Name: _____

Part b (5 points) The table below is intended to show a snapshot of a single initially empty set of a cache after making 8 sequential accesses, all LD instructions to that set. The replacement policy is the tree-replacement scheme described above. Assume the block size = the size of a single data element. The state of LRU[2:0] and the addresses of data stored in each block after each access are shown. Line 1 in the table reflects the first access, LD A. The LRU bits are set to 110, A is in Way 0, and Ways 1,2,and 3 are all invalid. Your job is to complete the table. Assume each LD accesses only one of 5 distinct addresses (A,B,C,D,E). A memory address may be accessed more than once. Note: There is only one unique sequence of loads that will produce the state shown after the 8th access (LD E).

	Access	LRU[2]	LRU[1]	LRU[0]	Way0	Way1	Way2	Way3
1	LD A	1	1	0	A	-	-	-
2	LD C	0	1	1	A	-	C	-
3	LD B							
4	LD D							
5	LD _							
6	LD _							
7	LD _							
8	LD E	1	1	0	E	A	C	D

Part c (5 points). Prove that the solution of Part b is unique. Use this space ONLY and please be clear.

Name: _____

Problem 4. (25 points) Little Computer Inc. is now planning to build a new computer that is more suited for scientific applications. LC-3b can be modified for such applications by replacing the data type Byte with Vector. The new computer will be called LmmVC-3 (Little 'mickey mouse' Vector Computer 3). Your job is to help us implement the datapath for LmmVC-3. LmmVC-3 ISA will support all the scalar operations that LC-3b currently supports except the LDB and STB will be replaced with VLD and VST respectively. Our datapath will need to support the following new instructions:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOVI Vstride, amount6	1011			000			000			amount6						
MOVI Vlength, amount6	1011			001			000			amount6						
VLD VDR, BaseR, offset6	0010			VDR			BaseR			offset6						
VADD VDR, VSR1, VSR2	1010			VDR			VSR1			0	1	0	VSR2			
VADD VDR, VSR1, SR2	1010			VDR			VSR1			0	0	0	SR2			
VST VSR, BaseR, offset6	0011			VSR			BaseR			offset6						

Note: VDR = Vector Destination Register, VSR = Vector Source Register

MOVI

If IR[11:9] = 000, MOVI moves the unsigned quantity amount6 to Vector Stride Register (Vstride).
 If IR[11:9] = 001, MOVI moves the unsigned quantity amount6 to Vector Length Register (Vlength).
 This instruction has already been implemented for you.

VLD

VLD loads a vector of length Vlength from memory into VDR. VLD uses the opcode previously used by LDB. The starting address of the vector is computed by adding the LSHF1(SEXT(offset6)) to BaseR. Subsequent addresses are obtained by adding Vstride to the address of the preceding vector element.

VST

VST writes the contents of VSR into memory. VST uses the opcode previously used by STB. Address calculation is done in the same way as for VLD.

VADD

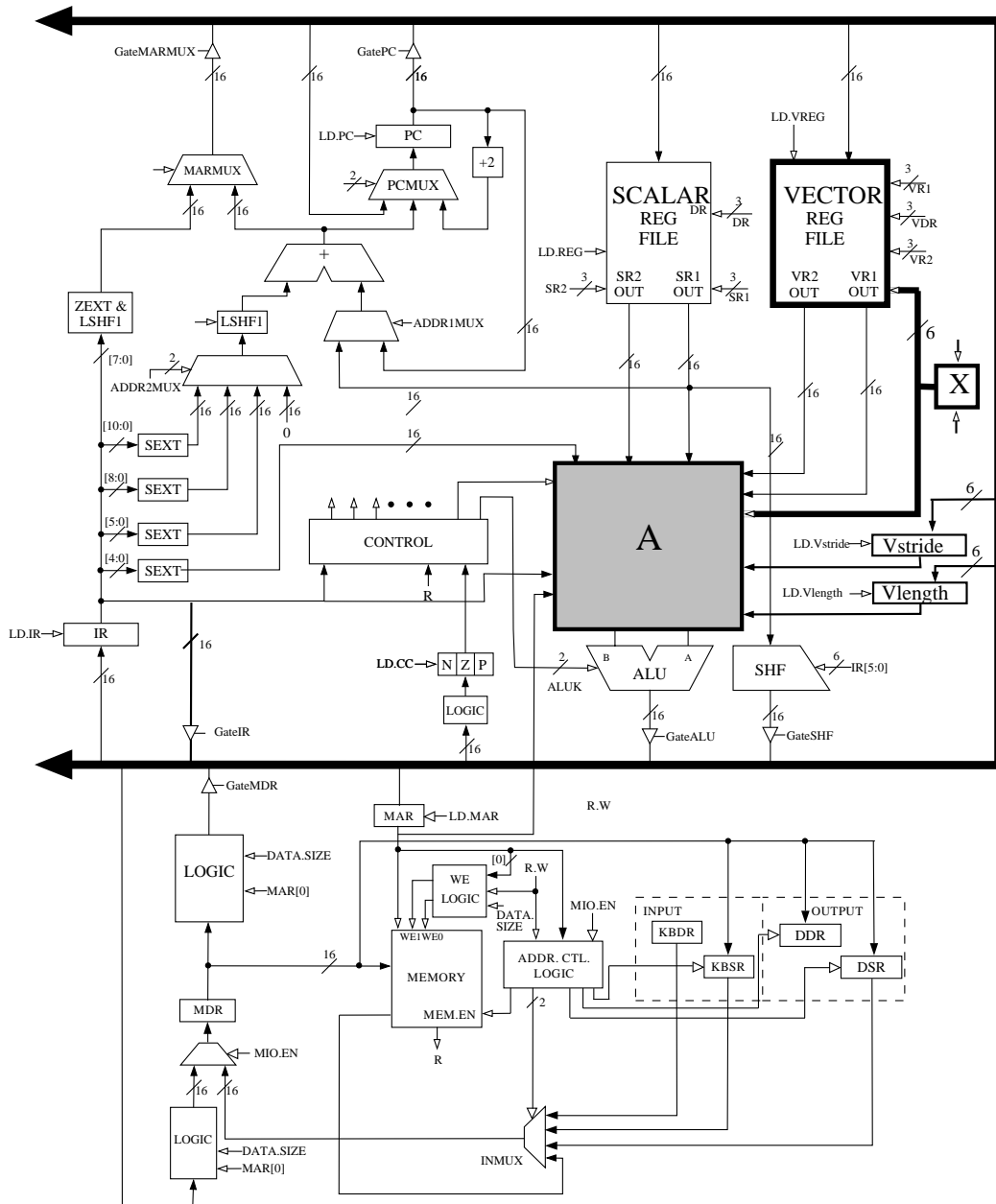
If IR[4] is a 1, VADD adds two vector registers (VSR1 and VSR2) and stores the result in VDR.
 If IR[4] is a 0, VADD adds a scalar register (SR2) to every element of VSR and stores the result in VDR.

VLD, VST, and VADD do not modify the content of Vstride and Vlength registers.

The following five hardware structures have been added to LC-3b in order to implement LmmVC-3.

- Vector Register File with eight 63-element Vector registers
- Vector Length Register
- Vector Stride Register
- Grey box A
- Box labeled X

These structures are shown in the LmmVC-3 datapath shown on next page.



Name: _____

Part a (4 points) A 6-bit input to the Vector Register file has been labeled X on the datapath diagram. What is the purpose of this input. (Answer in less than 10 words)

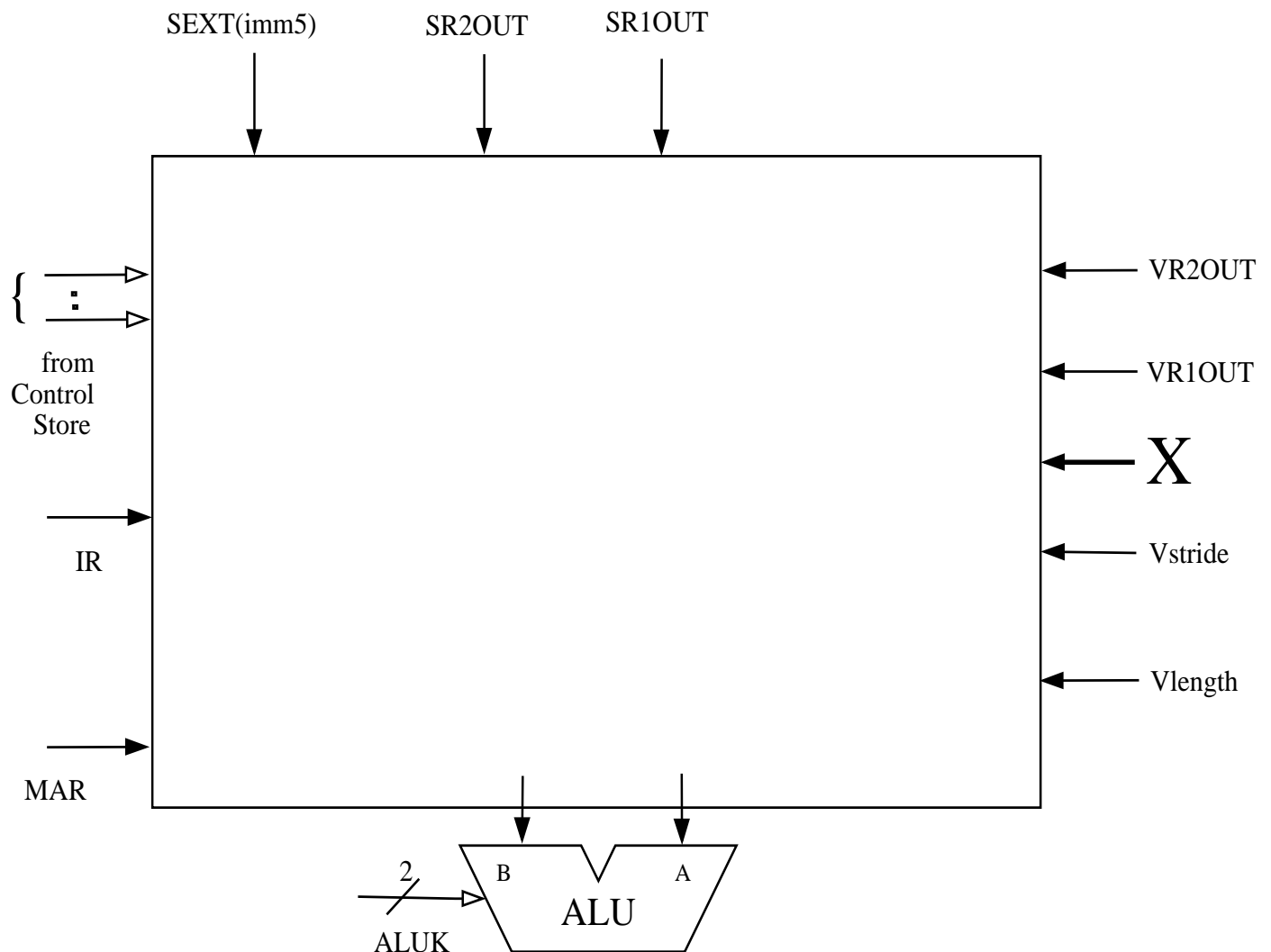
The logic structure X contains a 6-bit register and some additional logic. X has two control signals as its inputs. What are these signals used for?

Part b (12 points) Grey box A contains several additional muxes on both input lines to the ALU. Complete the logic diagram of grey box A (shown below) by showing all muxes and interconnects. You will need to add new signals to the control store; be sure to clearly label them in the logic diagram.

Hint 1: Keep in mind that we will still need to support all the existing scalar operations.

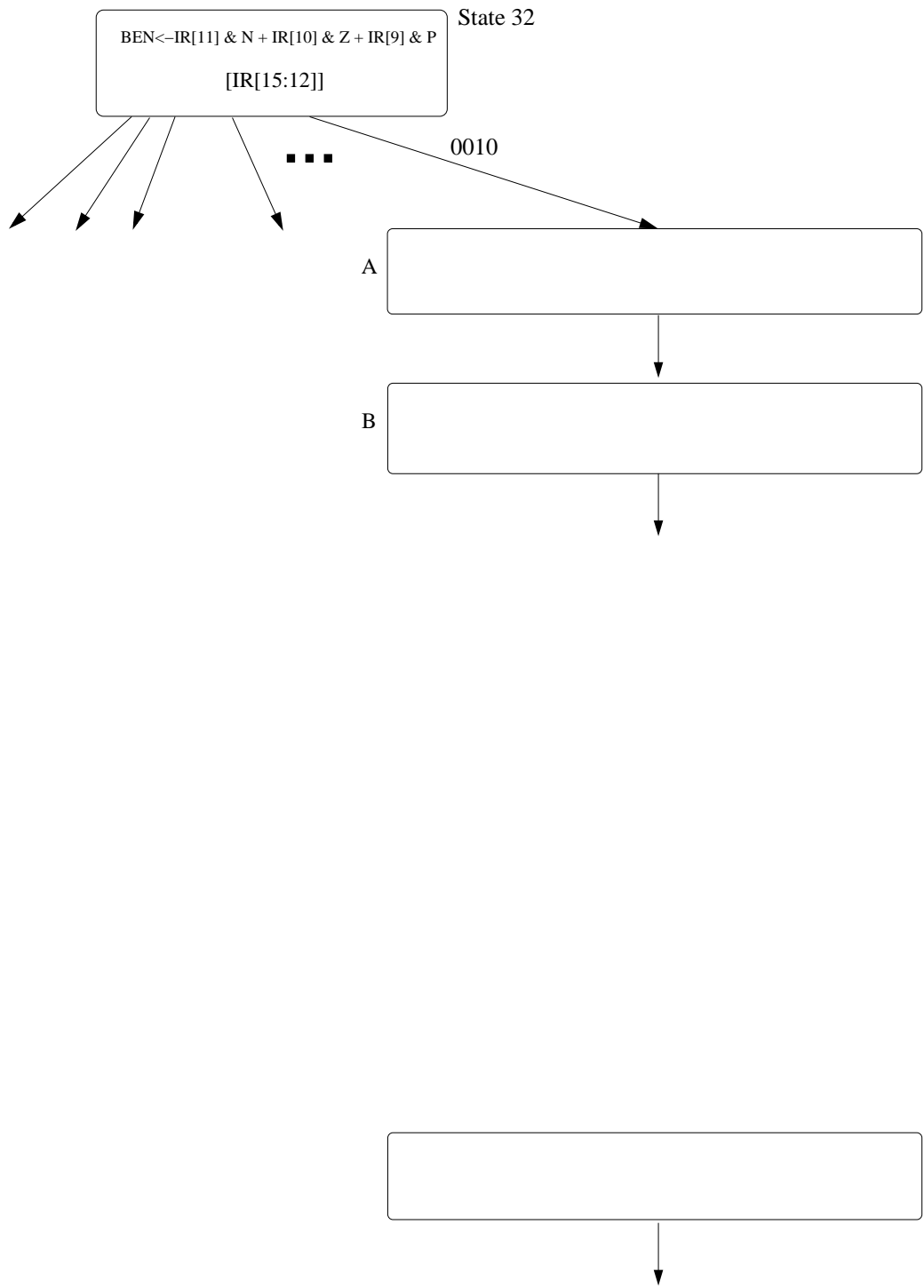
Hint 2: An XOR can be used to compare two values.

Hint 3: Our solution required 3 additional control signals and 7 2-to-1 muxes.



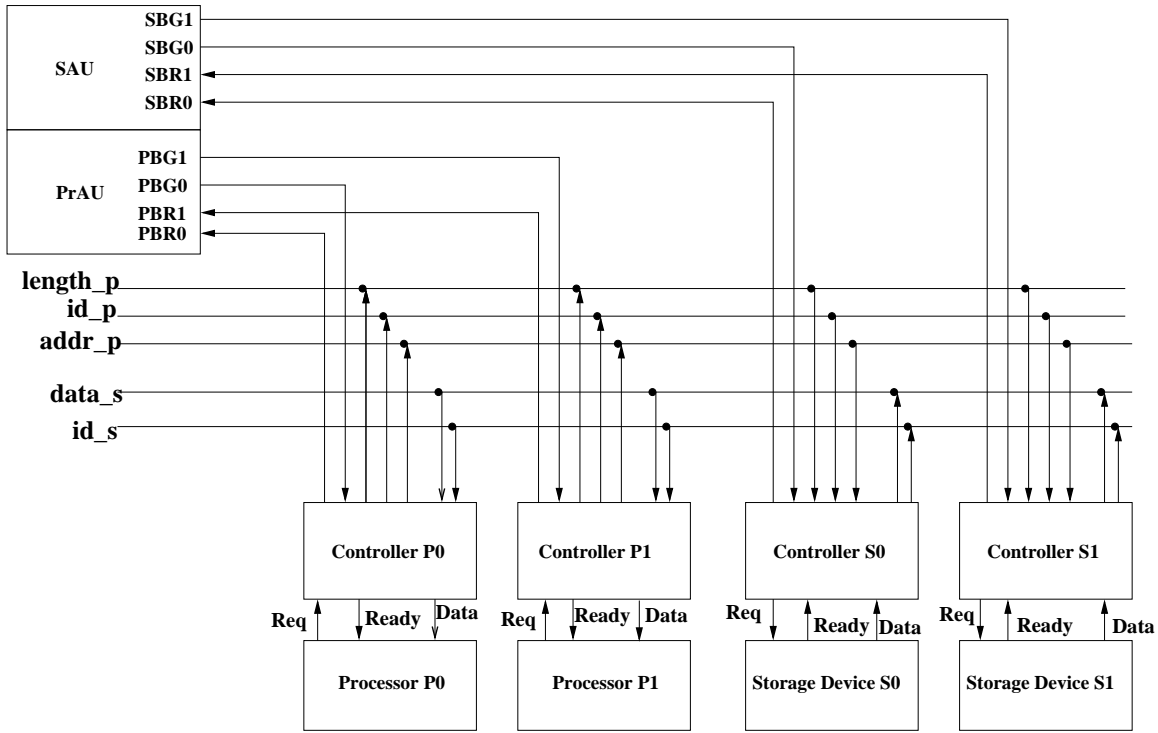
Name: _____

Part c (9 points) We show the beginning of the state diagram necessary to implement VLD. Using the notation of the LC-3b State Diagram, add the “bubbles” you need to implement VLD. Inside each “bubble” describe what happens in that state. You can assume that you are allowed to make any changes to the microsequencer that you find necessary. You do not have to make/show these changes. Make sure your design works when $Vlength = 0$. Full credit will be awarded to solutions that require not more than 6 states.



Name: _____

Problem 5. (20 points) We show below a synchronous split transaction bus system employing centralized arbitration. There are two processors (P0 and P1) and two storage devices (S0 and S1). For the sake of simplicity, we assume that processors will only *Read* data from the storage devices.



We describe below the arbitration and data transfer mechanisms:

Arbitration

There are two arbitration units, Processor Arbitration Unit(PrAU) and Storage Device Arbitration Unit(SAU). PrAU arbitrates the *addr_p* bus across P0 and P1. SAU arbitrates the *data_s* bus across S0 and S1. This allows one processor to request a transfer while the other processor is receiving data. P0 has higher priority than P1 and S0 has higher priority than S1. Once the SAU has granted the *data_s* bus to a storage device, it does not re-arbitrate until the device deasserts its bus request. This allows the storage device to carry out the entire transfer in consecutive cycles.

Data Transfer

Following is how a transfer between Processor *i* (Pi) and Storage Device *j* (Sj) proceeds:

1. When Pi needs data from Sj, it asserts the Bus Request signal (PBRi). Upon receiving the Bus Grant signal (PBGi) from the PrAU, the controller asserts the starting address on *addr_p*, the 'Vaddr' (valid address signal, not shown in the figure), number of elements to be transferred on 'length_p' (Count), and its Id (i in this case) on 'id_p'. Pi now waits for data from Sj.
2. When Sj sees an address that is in its address range with 'Vaddr' asserted, it latches the address, length, and Id and starts processing the request.
3. When Sj is ready to return the data, it asserts SBRj. Upon receiving the Grant signal (SBGj), it starts the process of transferring the data.
4. Sj asserts data on *data_s*, the Id of the processor that requested the data on *id_s* and the 'Vid' (valid id signal, where Vid = SBG0 OR SBG1, not shown in the figure). It also decrements its Count of the remaining elements.
5. When Pi sees its Id on *id_s* with 'Vid' signal asserted, it latches the data element and decrements its Count.
6. Pi and Sj cycle through steps 4 and 5 until Count reaches 0.

Name: _____

The device and controller synchronize using the Dev and Ready signals. When a transfer needs to be carried out the Dev signal is asserted. The availability of valid data is indicated by asserting the 'Ready' signal.

NOTE: For the sake of simplicity, assume each processor can have only one pending transfer. Each storage device can buffer up to two requests since each processor could have one pending transfer from a given storage.

Part a (10 points) Construct the state machine for the processor controller P_i . Show relevant inputs and outputs on all arcs. (Remember that the controller keeps track of the number of remaining data transfers in a 'Count' register.)

Part b (10 points) Construct the state machine for the storage device controller S_i . Show relevant inputs and outputs on all arcs. (Remember that the controller keeps track of the number of remaining data transfers in a 'Count' register.)