Problem Set 5

Due: November 15th @ 11:59 pm

Department of Electrical and Computer Engineering
The University of Texas at Austin
EE 382N.1, Fall 2018
Instructor: Dam Sunwoo
TA: Pritesh Chhajed

Instructions

You are encouraged to work on the problem set in small groups (3 or 4 per group) and turn in one problem set for the entire group on Gradescope. Create a copy of this file to fill in your answers within the spaces provided. If you need more space than given box, make a comment there and use extra work space at end of file. There are extra pages provided for the same.

Before submitting, convert it into a PDF file. Remember to put all your names and eid in the box below. The person submitting must choose everyone in the group.

You will need to refer to the assembly language handout and the LC-3b ISA, microarchitecture, and state diagram documents on the course website.

Student names and EID

Questions

Problem 1

Consider the following piece of code:

```
for(i = 0; i < 100; i++)
  A[i] = ((B[i] * C[i]) + D[i]) / 2;
```

Part a

Translate this code into assembly language using the following instructions in the ISA (note the number of cycles each instruction takes is shown with each instruction):

Opcode	Operands	# Cycles	Description
`LEA`	`Ri, X`	1	Ri ← address of X
`LD`	`Ri, Rj, Rk`	11	Ri ← MEM[Rj + Rk]
`ST`	`Ri, Rj, Rk`	11	MEM[Rj + Rk] ← Ri
`MOVI`	`Ri, Imm`	1	Ri ← Imm
`MUL`	`Ri, Rj, Rk`	6	Ri ← Rj × Rk
`ADD`	`Ri, Rj, Rk`	4	Ri ← Rj + Rk
`ADD`	`Ri, Rj, Imm`	4	Ri ← Rj + Imm
`RSHFA`	`Ri, Rj, amount`	1	Ri ← RSHFA (Rj, amount)
`BRcc`	`X`	1	Branch to X based on condition codes

Assume it takes one memory location to store each element of the array. Also assume that there are 8 registers (R0-R7).

How many cycles does it take to execute the program?

Part b

Now write Cray-like vector/assembly code to perform this operation in the shortest time possible. Assume that there are 8 vector registers and the length of each vector register is 64. Use the following instructions in the vector ISA:

Opcode	**Operands**	**Number of Cycles**	**Description**
`LD`	`Vst, #n`	1	Vst ← n
`LD`	`Vln, #n`	1	Vln ← n
`VLD`	`Vi, X + offset`	11, pipelined	
`VST`	`Vi, X + offset`	11, pipelined	
`Vmul`	`Vi, Vj, Vk`	6, pipelined	
`Vadd`	`Vi, Vj, Vk`	4, pipelined	
`Vrshfa`	`Vi, Vj, amount`	1	

How many cycles does it take to execute the program on the following processors? Assume that memory is 16-way interleaved.

i. Vector processor without chaining, 1 port to memory (1 load or store per cycle)

ii. Vector processor with chaining, 1 port to memory

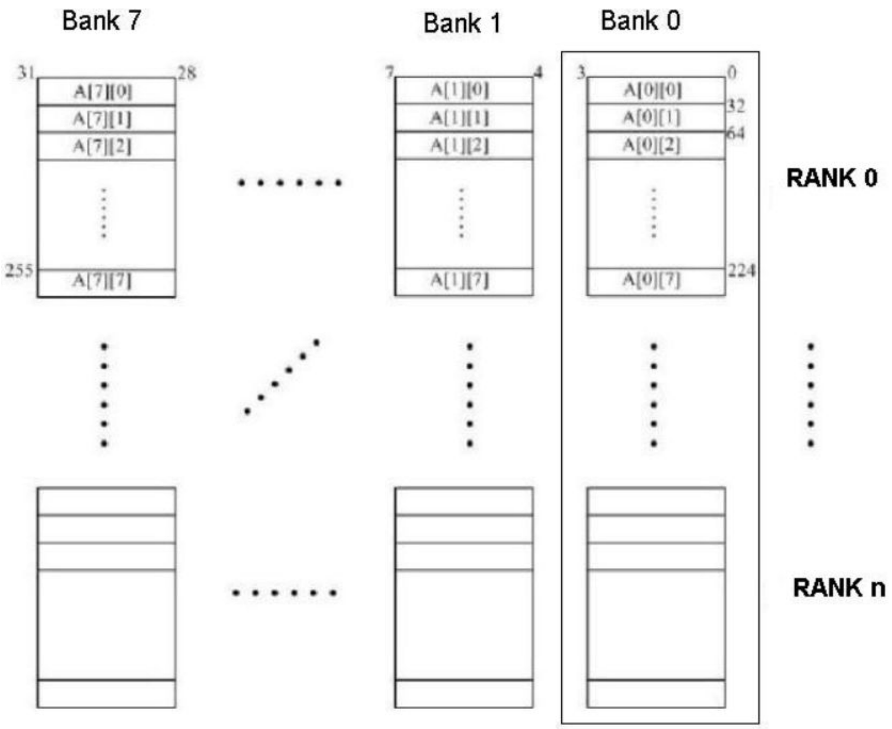
iii. Vector processor with chaining, 2 read ports and 1 write port to memory

Problem 2

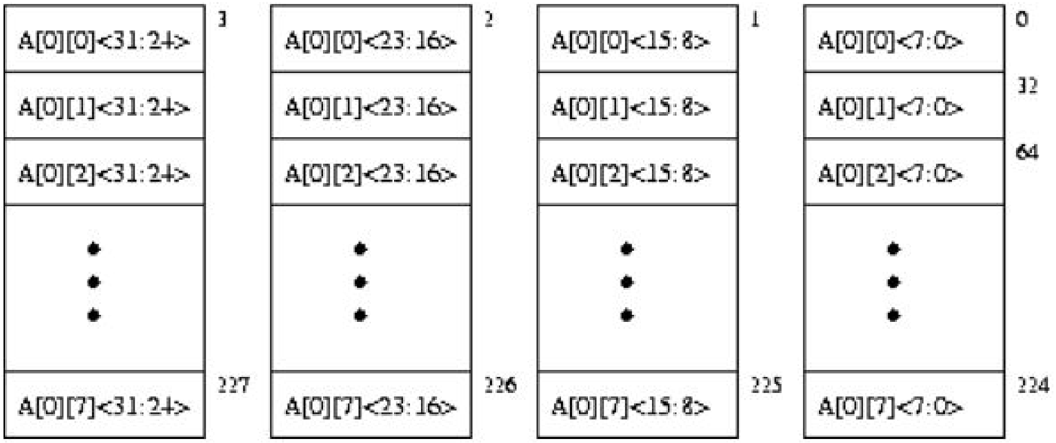
Consider the following piece of code:

```
  for(i = 0; i < 8; ++i){
    for(j = 0; j < 8; ++j){
      sum = sum + A[i][j];
    }
  }
```

The figure below shows an 8-way interleaved, byte-addressable memory. The total size of the memory is 4KB. The elements of the 2-dimensional array, A, are 4-bytes in length and are stored in the memory in column-major order (i.e., columns of A are stored in consecutive memory locations) as shown. The width of the bus is 32 bits, and each memory access takes 10 cycles.



A more detailed picture of the memory chips in Rank 0 of Bank 0 is shown below.



a. Since the address space of the memory is 4KB, 12 bits are needed to uniquely identify each memory location, i.e., Addr[11:0]. Specify which bits of the address will be used for:

- Byte on bus
- Addr[_____:_____]
- Interleave bits (bank bits)
- Addr[_____:_____]
- Chip address

- ○ Addr[_____:_____]
- ○ Rank bits
- ○ Addr[_____:_____]

b. How many cycles are spent accessing memory during the execution of the above code? Compare this with the number of memory access cycles it would take if the memory were not interleaved (i.e., a single 4-byte wide memory chip).

c. Can any change be made to the current interleaving scheme to optimize the number of cycles spent accessing memory? If yes, which bits of the address will be used to specify the byte on bus, interleaving, etc. (use the same format as in part a)? With the new interleaving scheme, how many cycles are spent accessing memory? Remember that the elements of A will still be stored in column-major order.

d. Using the original interleaving scheme, what small changes can be made to the piece of code to optimize the number of cycles spent accessing memory? How many cycles are spent accessing memory using the modified code?

Problem 3

NOTE: The cycle numbers depend on your assumptions!!!! Write them down!!!

This question is about buses and refers to the system shown in the figure below (latencies are the numbers given). The processor performs the following list of accesses, which must be issued in the order below but can complete out of order.

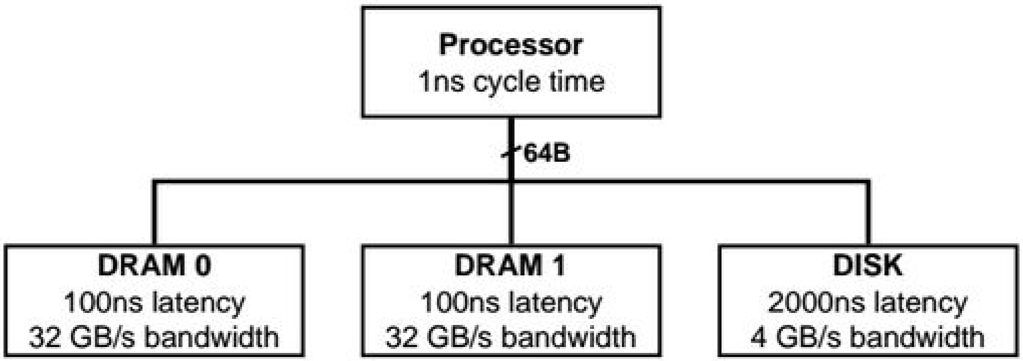
Assumptions:

- The latency of the wires is 0ns (1 bus traversal does not add latency).
- A request on the bus does not carry any data.
- The data is transferred in one long burst, and the time is determined by the bandwidth of each component. The entire transfer requires a single arbitration.
- Each component can only handle one access at a time and is busy until that access completes.
- There is no bus scheduling – all requests try to arbitrate only when they are ready and not before.
- If an arbiter is used, all transaction that lost must retry during the next arbitration cycle. Lower originating ID wins arbitration in case of a conflict and can start immediately following the arbitration delay.
- CSMA (collision sense multiple access) is a form of distributed arbitration, which requires retrying all transactions that attempt to use the bus at the same time (assume that in-progress transactions do not have to restart). The transmitter of data is considered the bus master.
- The command and data buses are arbitrated together; i.e. one device CANNOT send a command while another device sends data.

List of Accesses:

ID	Time (ns)	Component	R/W	Datasize
0	0	disk	R	4KB
1	100	DRAM0	R	32B

2	2000	DRAM1	R	32B
3	2050	DRAM1	R	32B
4	2100	disk	R	4KB
5	2110	DRAM1	R	32B



Fill in end time of each access in the table below, which describes different bus design options.
State any assumptions.

Bus Design	Time of Completion for access ID					
	0	1	2	3	4	5
Synchronous (10ns clock), pending transactions, centralized arbitration (one bus cycle (10ns) delay per arbitration).						
Asynchronous, split transactions, CSMA arbitration (retries: lower access ID waits 1ns, higher access IDs all wait 5ns).						

Problem 4

If the latency of a DRAM memory bank is 37 cycles, into how many banks would you interleave this memory in order to fully hide this latency when making sequential memory accesses?

Space for extra work