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

EE 460N Spring 2019
Y. N. Patt, Instructor
Aniket Deshmukh, Chester Cai, Mohammad Behnia, TAs
Final Exam
May 17, 2019

Name:____

Part A	**Part B**
Problem 1 (10 points):_______	Problem 6 (20 points):_______
Problem 2 (10 points):_______	Problem 7 (20 points):_______
Problem 3 (10 points):_______	Problem 8 (20 points):_______
Problem 4 (10 points):_______	Problem 9 (20 points):_______
Problem 5 (10 points):_______	
Part A Total (50 points):______________	Part B Total (80 points):______________

Total (130 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.

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

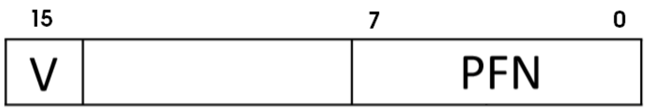
Signature:____

GOOD LUCK!

Name:____________________________________

Problem 1 (10 points): Assume virtual address space is 2^{20} bytes, physical address space is 2^{16} bytes. Page size is 256 bytes.

PTEs are 16 bits. PTE[15] is the valid bit. PTE[7:0] is the Page Frame Number. A PTE is shown below:



The TLB is as shown below:

V	Page Number	PTE[15] ... PTE[7:0]
1	x324	1 ... x32
0	x365	0 ... x33
1	x365	1 ... x34
0	x347	0 ... x35
1	x382	0 ... x36

The location corresponding to VA x36543 is resident in physical memory. What is its physical address?

Name:___________________________________

Problem 2 (10 points): An 8GB physical memory has a 4-way physically addressed, set associative cache. The line size is 4 bytes. The cache's data store has a capacity of 128 bytes.

Your job: Answer the following questions

Number bits in physical address:	
Number of offset bits:	
Number of index bits:	
Number of tag bits:	

Name:__________________________________

Problem 3 (10 points): Recall the pipelined processor that you designed (or tried to design!) for Lab 6. It has five stages: FETCH, DECODE, AGEX, MEM and SR.

Part a: A program that has no control instructions and no stalls due to dependencies is run on your processor. Assume all loads and stores are cache hits and take one cycle to access. The number of instructions in the program is large. What is the average number of cycles per instruction (CPI) for when this program runs on your pipelined processor?

Part b: Repeat Part a, except this time one out of every five instructions in the program is a control instruction. On each control instruction, the pipeline inserts bubbles for 3 cycles. What is the average CPI for this program?

Part c: Repeat Part a again, except this time not all loads and stores are cache hits. In fact, your D cache has an average hit ratio of 90% on this program. On a D cache miss, the pipeline stalls for 10 cycles. 40% of the instructions in your program are loads and stores. What is the average CPI for this program?

Name:___________________________________

Problem 4 (10 points): Assume IEEE-like floating point numbers, except each number is represented with 8 bits, 3 bits for exponent.

Represent $2\frac{7}{8}$:

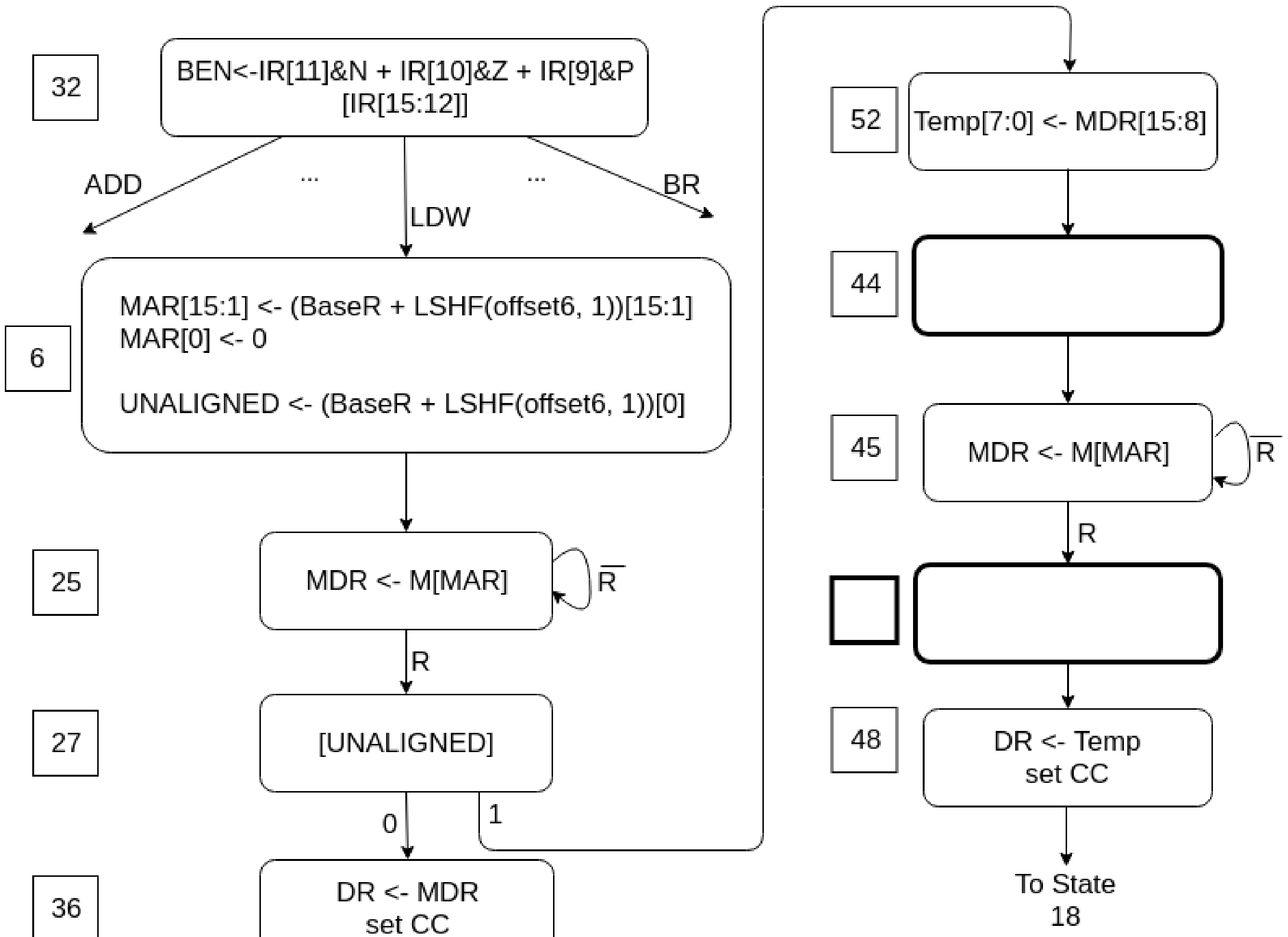
Represent $1\frac{9}{16}$: +

Represent the sum (assume rounding to 0):

Name:___________________________________

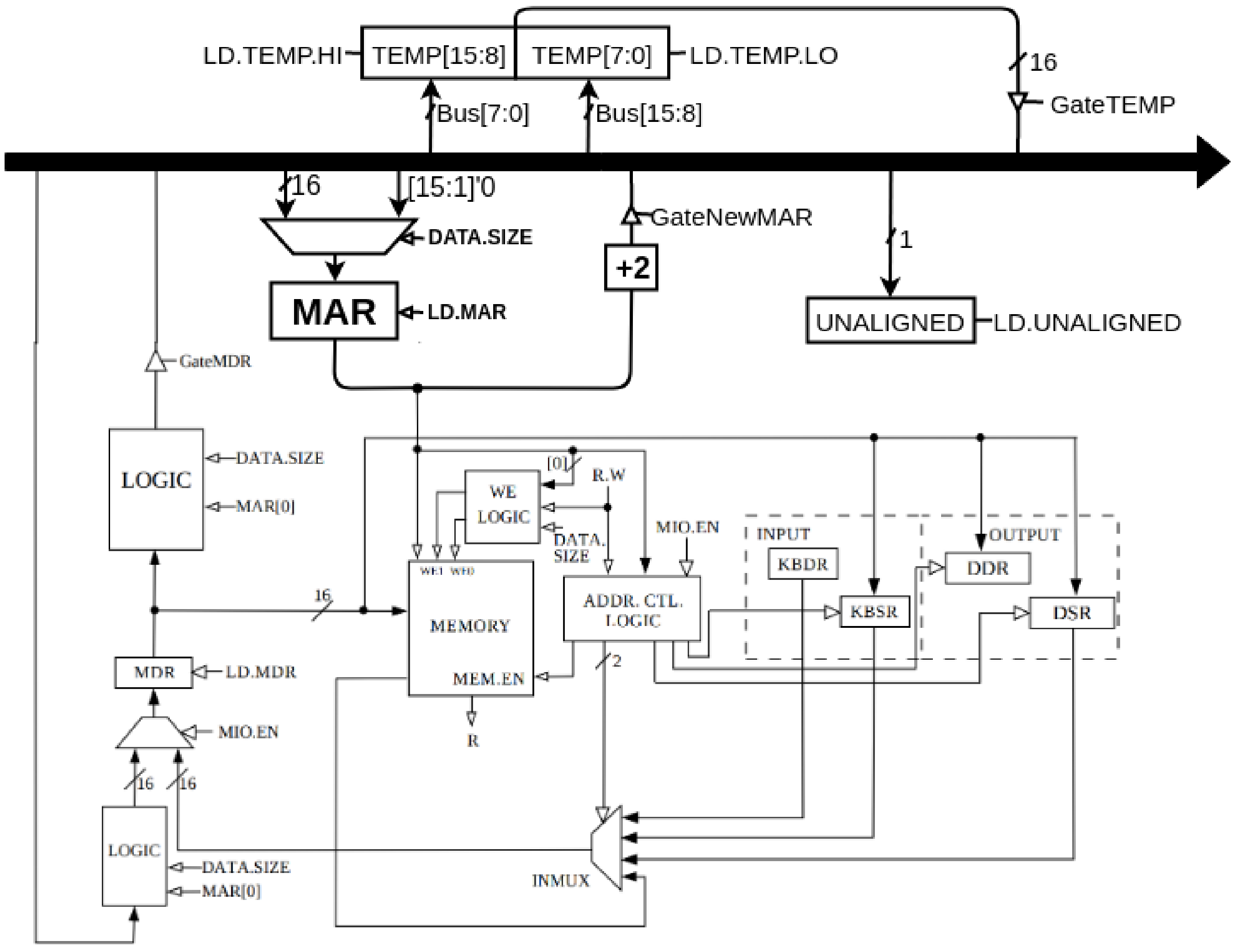
Problem 5 (10 points): We wish to augment the LC-3b with unaligned memory accesses. In this problem, we will focus on unaligned accesses for load instructions. To do this, changes to the datapath, state diagram and microsequencer are needed. You are given the datapath and the microsequencer on the next two pages.

Your job: Fill in the missing entries in the state diagram.



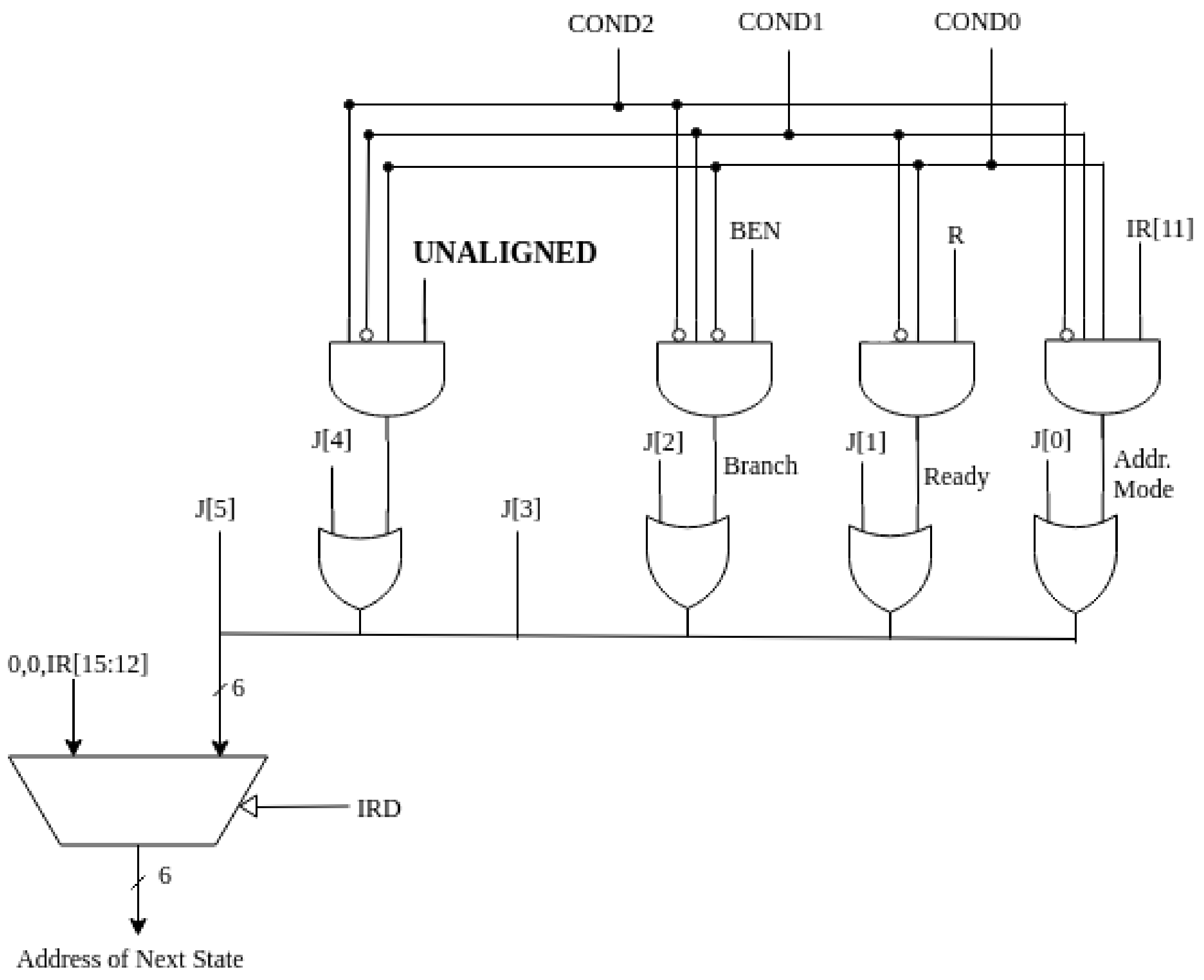
PROBLEM CONTINUES ON NEXT PAGE

Name:______________________________________



PROBLEM CONTINUES ON NEXT PAGE

Name:___________________________________


COND2
COND1
COND0
UNALIGNED
BEN
R
IR[11]
J[4]
J[2]
Branch
J[1]
Ready
J[0]
Addr.
Mode
J[5]
J[3]
0,0,IR[15:12]
6
IRD
6
Address of Next State

Name:___________________________________

Problem 6 (20 points): You are hired as the summer intern for Little Computer company. The company′s biggest competitor just launched another computer called SuperComputer3. Your assignment is to find out as many specifications as you can about SuperComputer3′s memory subsystem. From the public information about SuperComputer3, you have learned:

- SuperComputer3 has a byte addressable memory system.
- The physical address contains 12 bits.
- SuperComputer3 contains one level of physically addressed cache of size 128 bytes.
- Main memory is implemented with the DRAM technology we studied in class.
- The DRAM access does not start until a miss in the cache is confirmed. That is, in the event of a cache miss, the latency observed by the processor is the latency of the cache + the latency of the DRAM.
- Opening a row in DRAM takes a fixed amount of time regardless whether there is another row in the row buffer or not.
- The memory controller issues loads and stores in order.
- The DRAM bus is 8 bytes.
- A DRAM address consists of four fields, with the exact number of bits for each field unknown.

Row	Column	Bank	Byte on Bus

You conducted an experiment where 10 byte accesses were made. Each access was issued after the previous result was received by the processor. You also measured how long each access took. Assume that initially, the cache was empty and all the row buffers were closed.

Access Number	Read/ Write	Address	# of cycles taken
1	R	0b 0 1 0 0 0 1 1 1 0 0 0 0	104
2	R	0b 0 1 0 0 0 1 1 1 0 1 0 1	4
3	R	0b 0 1 0 0 0 1 1 1 1 0 0 0	104
4	R	0b 0 1 0 0 1 1 1 1 1 0 0 0	24
5	R	0b 1 0 0 0 0 1 1 1 0 0 0 0	104
6	R	0b 0 1 0 0 0 1 1 1 0 0 0 0	104
7	R	0b 1 0 0 0 0 1 1 1 0 0 0 0	104
8	R	0b 1 1 0 0 0 1 1 1 0 0 0 0	104
9	R	0b 0 0 0 1 0 1 1 1 0 0 0 0	104
10	R	0b 0 0 1 0 0 1 1 1 0 0 0 0	24

PROBLEM CONTINUES ON NEXT PAGE

Name:__________________________________

Part a: What is the cache line size?

Part b: What is the associativity of the cache?

Part c: How long does it take to access the cache?

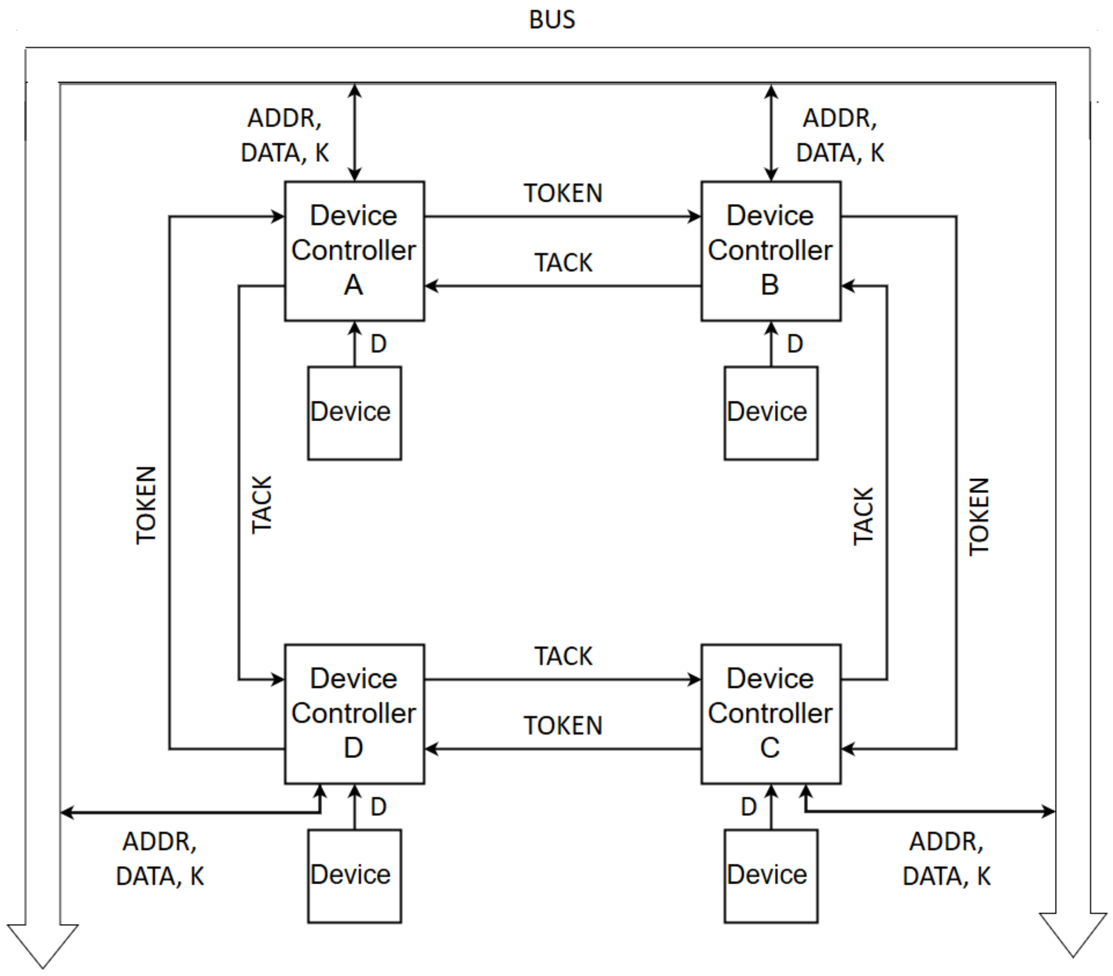
Part d: How many cycles does it take to open a row in DRAM?

Part e: How many Rows are there in a DRAM chip?

Part f: One of your friends is an intern at the Super Computer company. One night, he accidentally told you that SuperComputer3′s main memory is N-way interleaved. In the morning, you could not remember what the number N was from last night. All you can remember is that N is less than 64. Explain how you would find out how many banks the main memory has.

Name:___________________________________

Problem 7 (20 points): Consider a system of device controllers, organized in a ring as shown below. Each controller provides service to its device when its device asserts D.



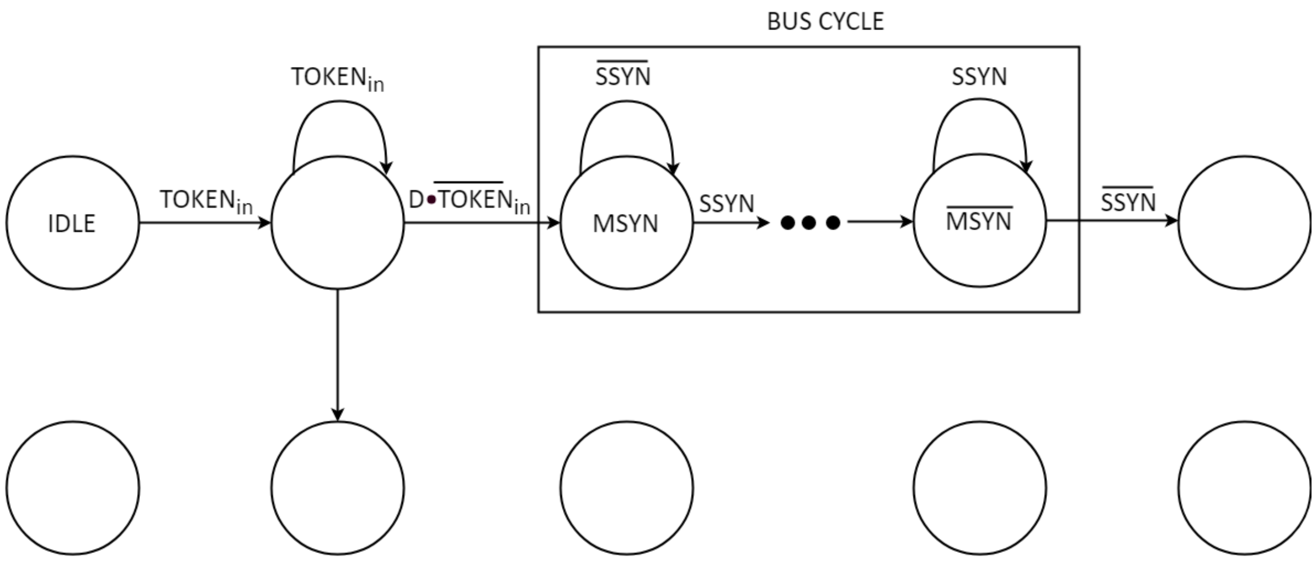
Arbitration occurs by means of a TOKEN signal. When a bus cycle completes, the device controller that is the master in that bus cycle passes the TOKEN to the next device controller in the ring in a clockwise direction. If the device associated with that device controller wants service as indicated by asserting D, that controller initiates the next bus cycle. If not, it passes the TOKEN to the next device controller in the ring.

The scheme proceeds as follows: Assume device controller A has just completed the bus cycle and sends the TOKEN to device controller B, the next device controller in the ring. When device controller B receives the TOKEN, it asserts TACK, signalling device controller A that it has the TOKEN and device controller A can negate TOKEN. When device controller B learns that TOKEN has been negated, it either initiates the bus cycle if its device wants service, or it passes TOKEN to the next device controller if its device does not want service.

PROBLEM CONTINUES ON NEXT PAGE

Name:___________________________________

Part a: To implement the scheme described on the previous page, each device controller contains the asynchronous state machine shown below. **Your job:** Complete the state machine. Use only as many states as you need to.



Part b: While trying to verify the state machine, an Aggie tried to initialize all the device controllers to the IDLE state. Does this present a problem? If so, what is the problem? If not, why not?

Part c: Suppose there is a point in time when no device wants the bus. Does this present a problem? If so, what is the problem? If not, why not?

Part d: Why does the device controller that has received the TOKEN wait until the previous device controller has negated TOKEN before either starting the bus cycle or passing the TOKEN to the next controller?

Name:___________________________________

Problem 8 (20 points): A six instruction program segment is run on an out-of-order processor that executes instructions based on the Tomasulo algorithm with a Reorder Buffer (ROB) for in-order retirement. The ISA specifies 8 registers, R0 to R7. The execute stage of the pipeline contains **one** pipelined adder and **one** pipelined multiplier. The processor can fetch and decode two instructions each cycle.

To handle dependencies between instructions that are fetched at the same time, an additional stage, rename, is added. The Register Alias Table (RAT) is read in the rename stage. Two instructions can be renamed every cycle, even if they have a flow dependency between them.

The ROB and the reservation stations are initially empty and filled from top to bottom. New instructions are added to the ROB **at the end of decode**. New instructions are added to the reservations stations **at the end of rename**. If there is not enough room in the reservation stations for both the instructions, the second instruction remains in the rename stage. The rename stage is stalled until the second instruction can be added to a reservation station.

Instructions remain in reservation stations until they complete execution. At the end of the last cycle of execution, each instruction stores its result into both its ROB entry and any reservation stations waiting for it. An instruction remains in the ROB until the end of the cycle in which it writes its result to the register file. The ROB can retire up to two instructions each cycle.

- Fetch, decode and rename stages take one cycle each.
- ADD takes 4 cycles to execute.
- MUL takes 6 cycles to execute.
- Both functional units have 2-entry reservation stations.
- Instruction write back to the registers from the ROB in a single cycle.
- Data forwarding is implemented.

The table below contains the six instruction program segment.

	Opcode	**DR**	**SR1**	**SR2**
I1				
I2			R1	R1
I3	MUL			
I4				
I5		R2		
I6				

Part a: Your job: Fill in the missing entries in the table.
(Note: The next page contains some useful information to help you solve this problem).

PROBLEM CONTINUES ON NEXT PAGE

Name:__________________________________

Two snapshots of the register alias table are provided: one before the first instruction is fetched and one after the program has finished execution. The state of the ROB at the end of cycle 7 is also provided.

	V	Tag	Value
R0	1	-	15
R1	1	-	2
R2	1	-	80
R3	1	-	1
R4	1	-	9
R5	1	-	2
R6	1	-	99
R7	1	-	100

Before Cycle 1

	V	Tag	Value
R0	1	-	15
R1	1	-	**19**
R2	1	-	**60**
R3	1	-	1
R4	1	-	**20**
R5	1	-	**95**
R6	1	-	99
R7	1	-	100

After Execution

V	Retired	Executed	DR	Value
1	0	**1**	R4	10
1	0	0	R1	30
1	0	0	R4	50
1	0	0	R5	70
0	0	0	R6	99
0	0	0	R7	100

After Cycle 7

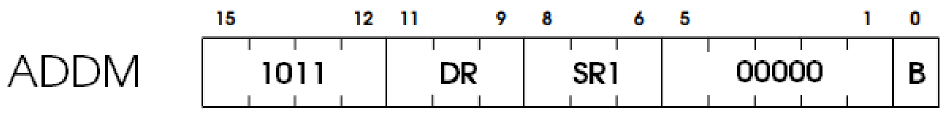
Part b: Complete the timing diagram for the execution of the six instructions. Write the stage each instruction occupies during each cycle. You have been provided 20 cycles, use as many as you need.

Instructions stalled due to dependencies in the reservation stations should be represented as "X". Instructions that finish execution and are waiting for retirement should be represented as "W". The cycle in which they retire is indicated by the register they write to. Instructions in the rename stage are indicated by an "R". If an instruction is stalled in a stage waiting for the next stage to be free, please represent that with the name of the stage and an asterisk (F*, D*, R*).

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1	F	D	R																	
I2	F	D	R																	
I3		F	D																	
I4		F	D																	
I5																				
I6																				

Name:____________________________________

Problem 9 (20 points): The LC-3b is a load/store machine. That means the ADD instruction requires its source operands to be in registers. It is often useful to be able to add a column of numbers, where the numbers are contained in successive memory locations. We can accomplish this conveniently with a new instruction ADDM, using the unused opcode 1011, as follows. The format of ADDM is shown below.



Bits[8:6] specify a register containing the address of the first source. Bits[11:9] specify a register that contains the address of the second source and also the address of the destination. Bit[0] specifies the size of each number to be added (Byte = 1, Word = 0).

The instruction does the following:

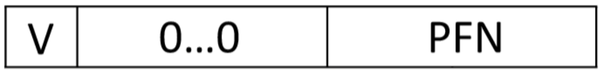
```
M[DR] <-- M[SR1] + M[DR], where M[SR1] is fetched before M[DR]
if(B == 1)
    SR1 <-- SR1 + 1
else
    SR1 <-- SR1 + 2
```

That is, after ADDM executes, SR1 contains the address of the next number in the column of numbers to be added.

Virtual Memory is implemented with a VAX-like two level translation scheme. The characteristics of the memory system are presented below.

Virtual Address Space:	64KB	Physical Address Space:	4KB
User Space:	0x0000-0x7FFF	Page Size:	256B
System Space:	0x8000-0xFFFF	PTE Size:	2B

A PTE is shown below.



A PTE includes a Valid Bit and the PFN. The low bits of the PTE are used for the PFN.

The user space page table starts at the beginning of a page. The system space page table starts at the beginning of a frame.

PROBLEM CONTINUES ON NEXT PAGE

Name:____________________________________

Part a: Consider one instance of the execution of the ADDM instruction. The microarchitecture includes a 4-entry, fully associative TLB with LRU replacement policy. Only user virtual addresses are stored for translation in the TLB. Including Instruction Fetch, what is the maximum number of memory accesses that could be needed to execute this instruction?

The state of the TLB and Register File before the execution of the instruction and the state of the Register File after execution are shown below.

Register	Value
R0	x0400
R1	x0600
R2	x3000
R3	x4000
R4	x4008
R5	x6008
R6	x0010
R7	x0100

Register File Before Execution

V	Page Number	PTE
1	x50	x8004
1	x30	x800A
0	—	—
0	—	—

TLB Before Execution

Register	Value
R0	x0400
R1	x0600
R2	x3002
R3	x4000
R4	x4008
R5	x6008
R6	x0010
R7	x0100

Register File After Execution

Part b: The table below shows an instance of execution of ADDM that requires only eight memory accesses.

Your job: Fill in the missing entries.

Access	VA	PA	Data	TLB Hit?
1				
2	xB20C		x8002	
3	x0606			
4			x333A	
5		x464	x8008	
6	xB280		x8006	
7		x608		
8			x5029	

PROBLEM CONTINUES ON NEXT PAGE

Name:___________________________________

Part c: Determine the following values.

System Page Table Base Register

User Page Table Base Register

Part d: The state of the TLB after the instruction's execution is partially given below. Fill in the remaining values.

V	Page Number	PTE
1	x50	x8004
1	x30	x800A
1		
1		