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

EE 460N Fall 2016
Y. N. Patt, Instructor
Siavash Zangeneh, Ali Fakhrzadehgan, Steven Flolid, Matthew Normyle TAs
Exam 2
November 21, 2016

Name: Solution

Problem 1 (25 points):_____

Problem 2 (15 points):_____

Problem 3 (15 points):_____

Problem 4 (20 points):_____

Problem 5 (25 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.

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

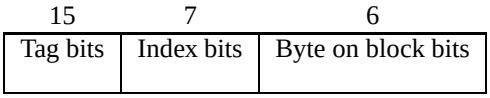
Signature:_____

GOOD LUCK!

Name:_____

Problem 1 (25 points)

Part a (5 points): A 2-way set-associative physically indexed, physically tagged write through cache implements LRU replacement. Memory space is 2^{28}, memory is byte-addressable, and the physical address bits are assigned as follows:



What are the total number of bits of storage required to implement a perfect LRU replacement policy for this cache.

2^7

Part b (5 points): If a cache is virtual, we avoid confusion as to what each address in the tag store means by one of two ways. Name one of them.

- flush cache on context switches
- keep process ID of each cache block in its tag store entry

Part c (5 points): Interrupt priority is determined by the urgency of handling the event. Is any event more urgent than "loss of power." Explain.

Machine check: a processor that does not work correctly is worse than a processor that does not work.

Part d (5 points): IEEE Floating Point uses radix 2 for dealing with exponents. The old IBM 360, currently called IBM Z-series, uses radix 16. Two advantages of radix 2 over radix 16 are:

Smaller wobble

The most significant bit does not have to be stored

Part e (5 points): Seymour Cray, in many ways the "father" of the vector computer, did not like the notion of cache memory, so the Cray I did not have caches. What is it about cache memory that he did not like?

non-determinism

Name:_____

Problem 2 (15 points)

Consider a floating point data type with all the IEEE characteristics except it contains 9 bits, rather than 32 or 64 bits. Your job: How many bits of exponent, how many bits of fraction? The bias is 4.

We can represent $2\frac{1}{4}$ and $\frac{1}{16}$ exactly.

If the rounding mode is round-toward-zero, computing $((2\frac{1}{4} + \frac{1}{16}) + \frac{1}{16})$ produces the result $2\frac{1}{4}$, and computing $((\frac{1}{16} + \frac{1}{16}) + 2\frac{1}{4})$ produces the result $2\frac{3}{8}$.

Part a (3 points): Given that computing $((2\frac{1}{4} + \frac{1}{16}) + \frac{1}{16})$ should produce the result $2\frac{3}{8}$, why was the result computed as $2\frac{1}{4}$?

$(2\frac{1}{4} + \frac{1}{16}) = 2\frac{5}{16}$ cannot be represented exactly and is rounding to $2\frac{1}{4}$. Or, there are not enough fraction bits for intermediate results of ADDs, which results in rounding errors.

Part b (7 points):

Number of bits of fraction [4] Number of bits of exponent [4]

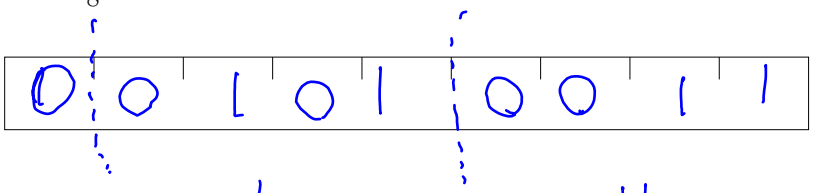
$2\frac{1}{4} = 10.01 = 1.001 \times 2^1$

$\frac{1}{16} = 1.000 \times 2^{-4} = 0.0001 \times 2^1$

Part c (5 points):

$2\frac{1}{4} + \frac{1}{16} = 2\frac{5}{16} = 1.00101 \times 2^1$ fraction bits should be fewer than 5

How would you represent $2\frac{3}{8}$ in our 9 bit floating point format?



$2\frac{3}{8} = 1.0011 \times 2^1$

exponent = 4 + 1 = 5 = 0101

However, $\frac{1}{8} + 2\frac{1}{4}$

$= 1.001 \times 2^1$
$+ 0.0001 \times 2^1$
1.0011×2^1

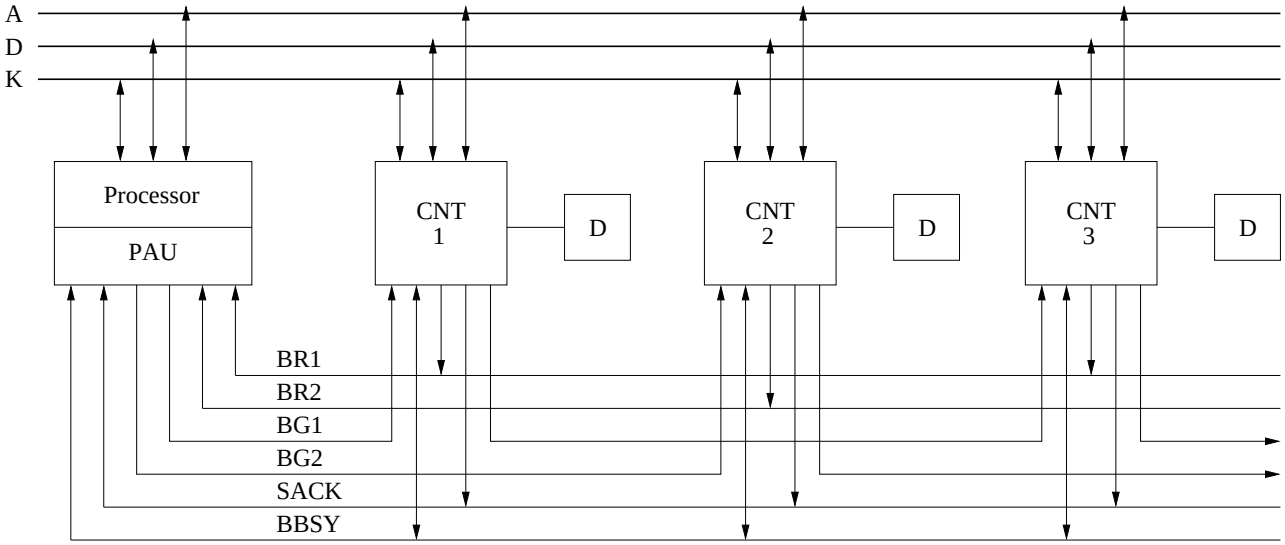
is represented exactly

So there are exactly 4 fraction bits

Name:_____

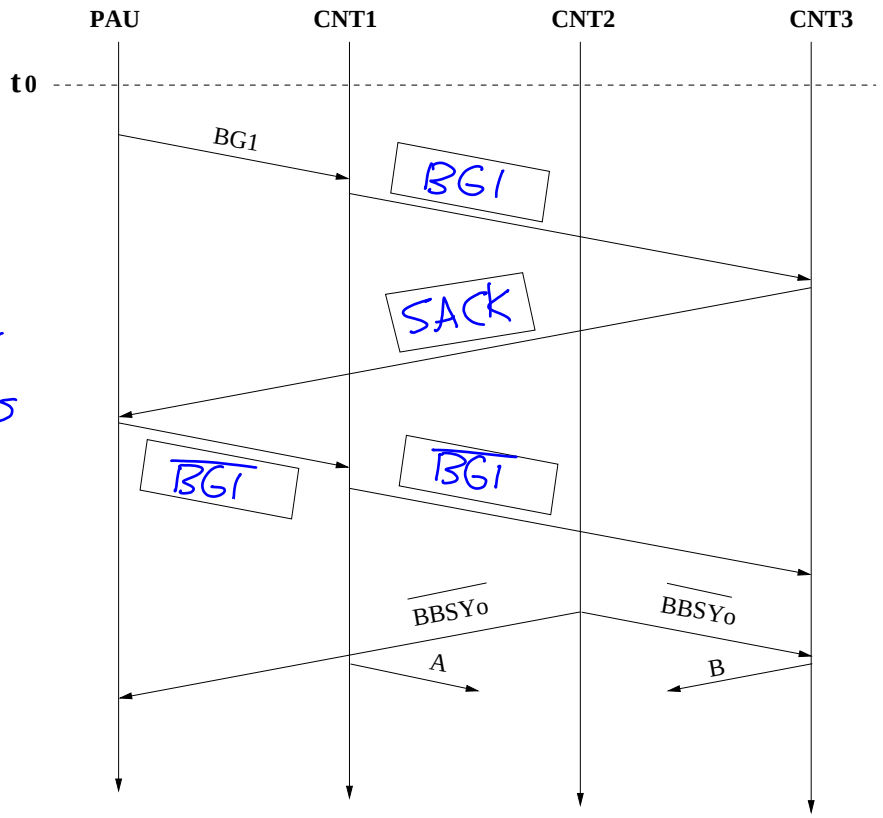
Problem 3 (15 points)

A processor/PAU and three device controllers are connected to an asynchronous bus as shown. Bus requests are of the form BR1 and BR2. BR2 is of higher priority.



In class, we showed timing diagrams for **transactions** on the bus. In this problem, we will examine the timing diagram for **arbitration**, as shown on the next page.

Name:_____



Each box 1.5 points

At t0, a bus master asserts BBSYo, allowing arbitration for the next bus cycle to begin. If you examine the entire timing diagram, can you tell which controller acquires the bus at t0? CNT2 (1 point)

The PAU asserts BG1 as shown on the diagram. The next four signals are unlabeled. Your job: provide the labels.

At some point, the current bus master completes the bus cycle and releases the bus. This causes controller 1 to provide signals A or controller 3 to provide signals B.

Does controller 1 want the bus for the next bus cycle? Yes / No (Circle one). Explain.

2 points No, since it daisy chains BG1 to CNT3

Does controller 3 want the bus for the next bus cycle? Yes / No (Circle one). Explain.

2 points Yes, since it accepts BG1 by asserting SACK

1 point What signal(s), if any, are represented by A. None

3 points What signal(s), if any, are represented by B. BBSYo, MSYN, Addr

Name:_____

Problem 4. (20 points)

Suppose we have a byte-addressable machine with a virtually indexed, physically tagged cache. The cache is two way set associative and is initially empty. The cache uses perfect LRU replacement.

The machine has a 16 bit virtual address space, with a 128 byte page size. The physical address space is 11 bits. Assume the virtual address has the form:

Leftover bits	Y Index bits	Z Byte on line bits

Part a: How many cache lines are there in the 16 bit virtual address space? Solve in terms of Z.

3 points

2^{16-Z}

Part b: Suppose you know one set of the cache is full, and nothing about the other sets. How many cache lines in the virtual address space are **definitely not** in the cache? Solve in terms of Y and Z.

3 points

$2^{16-Y-Z} - 2$

The following are consecutive memory accesses.

if cache line size > 8 bytes, ② must be hit, if < 8 byte ③ can't be hit

	Virtual Address	Physical Address	Cache Hit/Miss
1	0000 0001 1000 0000	0111 0000000	Miss
2	0000 0001 1000 1000	0111 0001000	Miss
3	0000 0001 1000 1100	0111 0001100	Hit
4	1111 0001 1000 0000	0101 0000000	Miss
5	0001 0011 1000 0000	0001 0000000	Miss
6	0000 0001 1000 0100	0111 0000100	Miss
7	0000 0000 1000 0001	1110 0000001	Miss
8	**A**	0001 xxxxxxx	Hit
9	0000 0001 1100 0100	0111 1000100	**B**

If more than 6 bits are used for indexing, ⑥ should've been a hit

If there are fewer than 6 bits used for indexing ⑧ cannot be a hit since ⑥ & ② would replace it

Part c: What range of virtual addresses can A be? Write your answer in binary.

6 points

0001 0011 1000 0XXX

6 points **Part d:** Which bits of the virtual address are used for indexing?

VA[8:3]

2 points **Part e:** Is B a hit or a miss?

miss

Name:_____

Problem 5 (25 points)

Consider an Out-of-Order exeuction/In-Order retirement processor. The ISA specifies 8 registers, R0 to R7. The microarchitecture contains one non-pipelined adder and one non-pipelined multiplier. Fetch and Decode take one cycle each. ADD execution takes 2 cycles, MUL execution takes 4 cycles, and in both cases one cycle is needed to write the result to a destination register. There is no data bypassing. The adder and multiplier each has a 3-entry reservation station. They are initially empty, and are filled from top to bottom. Each instruction remains in the reservation station until it retires.

A program consisting of five instructions takes 17 clock cycles to execute on this processor. The state of the register file is shown before execution starts (i.e., before cycle 1), at the end of cycle 7, and at the end of cycle 17.

	V	Tag	Value
R0	1	-	4
R1	1	-	5
R2	1	-	0
R3	1	-	20
R4	1	-	6
R5	1	-	3
R6	1	-	2
R7	1	-	7

Before Cycle 1

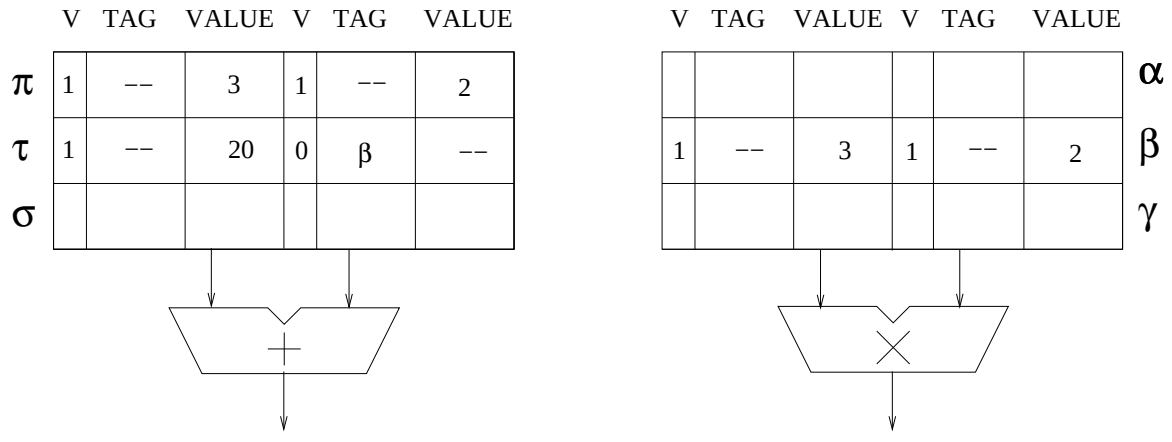
	V	Tag	Value
R0	1	-	4
R1	0	π	-
R2	1	-	20
R3	0	τ	-
R4	1	-	6
R5	1	-	3
R6	1	-	2
R7	1	-	7

End of Cycle 7

	V	Tag	Value
R0	1	-	4
R1	1	-	5
R2	1	-	31
R3	1	-	26
R4	1	-	6
R5	1	-	3
R6	1	-	2
R7	1	-	7

End of cycle 17

The state of the reservation stations at the end of cycle 7 are shown below.



Reservation Stations at the end of cycle 7

Name:_____

The following chart shows the cycles when the adder and multiplier are in use.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Multiplier			E	E	E	E	E	E	E	E							
Adder				E	E							E	E		E	E	

Part a (20 points): Specify the five-instruction program.

instruction

1	MUL R2, R0, R1
2	ADD R1, R5, R6
3	MUL R3, R5, R6
4	ADD R3, R2, R3
5	ADD R2, R1, R3

R2 ← 20
R1 ← 5
R3 ← 6
R3 ← 26

Part b (5 points): : How big is the Reorder Buffer? **Explain**. Hint: Instructions are allocated to the reorder at the same time they are placed in a reservation station.

4: Since the 5th instruction does not exist in the ADD reservation station in the end of cycle 7, that means 5th instruction could not have been decoded due to Reorder Buffer being full