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

EE 360N, Spring 2009
Yale Patt, Instructor
Ramapriyan Chakravarthy, Khubaib, Vivekanand Venugopal, TAs
Exam 2, April 29, 2009

Name: KEY

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.

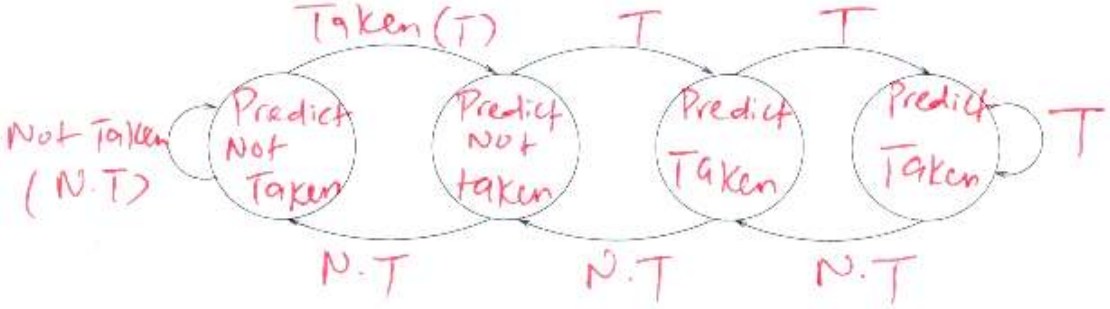
GOOD LUCK!

Name: _____

Problem 1 (25 points)

Part a (5 points): A dynamic branch predictor introduced by Jim Smith at ISCA in 1981 involved the following state diagram. Depending on which state it is in at any particular time, a branch will be predicted as taken or not taken.

Complete the state diagram by (a) labeling each arc, thereby indicating what causes the branch predictor to transfer from one state to another, and (b) writing "Predict taken" or "Predict not taken" inside each state, thereby indicating the branch prediction that will occur when a branch is fetched and decoded and the branch predictor is in each of the four states.



Part b (5 points): The Cray I did not have a data cache because its designer Seymour Cray did not like a particular property of data caches. Name the property.

Non-determinism

Instead the Cray I provided a structure that did not have that property and served to some extent the same purpose as a data cache. Name the structure.

T-registers or Backup registers

Name:_____

Problem 1 continued

Part c (7 points): We wish to design a 64 KB physical cache for an ISA having a 4KB page size. We want to keep the TLB out of the critical path, but refuse to add the complexity of a virtually indexed, physically tagged cache. What is the **minimum** associativity of our design that satisfies the above constraints?

4KB page ⇒ 12 bits for (index + offset)

Answer: 16-ways

Part d (8 points): A CMP processor consists of 8 simple cores on a single chip. A program consists of code that is 80% parallelizable. What is the speed-up obtained by executing this program on the 8-core chip versus running it on a chip that only has a single simple core.

$$Speedup = \frac{1}{\frac{0.8 \times 1}{8} + 0.2 \times 1} = \frac{1}{0.3}$$

Answer: 10/3

Suppose instead we execute this program on a chip where 4 of the 8 simple cores have been replaced by one heavy-weight core that processes 2.5 times faster than a simple core. What is the speed-up of running on this chip versus running on a chip that has eight simple cores. Assume that when the chip is processing the parallelizable code, the heavyweight core is idle.

$$speedup = \frac{\frac{0.8 \times 1}{8} + 0.2 \times 1}{0.8 \times \frac{1}{4} + \frac{0.2}{2.5}} = \frac{0.3}{0.28}$$

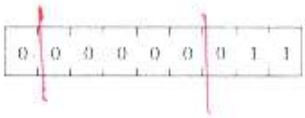
Answer: 30/28

Name:_____

Problem 2 (15 points)

In class, we discussed an IEEE-type, 6-bit floating point data type. In this problem, we grow to 9 bits: 5 bits for exponent, 3 bits for fraction. Exponent is represented with an excess-15 code.

Part a (5 points): What is the value of the number represented below?



Answer: 0.011×2^{-14}

Part b (5 points): The value 51/32 can not be represented exactly with this format. Round 51/32 to nearest, and show the representation of the rounded value below.

0 0 1 1 1 1 0 1

What is the value of the rounded result?

Answer: 1.101×2^{0} 13/8

Part c (5 points): Can we represent 51/32 exactly with a 9-bit, IEEE-type floating point data type if we reallocate the number of bits to exponent and fraction.

If **Yes**, specify a format:

Exponent: 3

Fraction: 5

Excess-code: 011 (3)

and show the exact representation of 51/32 in that format:

0 0 1 1 1 0 0 1 1

If **No**, explain why not?

Name: _____

Problem 3 (15 points)

A processor makes seven successive READ accesses to its physical memory system (PA is 12 bits). The addresses, in random order, are shown below:

x428, x364, x126, x102, x46C, x12B, x265.

Your job is to identify the sequence of the accesses, and fill in the table below with the addresses of each of the seven reads in their correct sequence.

1st	2nd	3rd	4th	5th	6th	7th
x102	x126	x428	x12B	x265	x46C	x364

Interchangeable

The soln. here is if initial value of LRU bit is 1 in set 6

Some information that may be useful:

1. Only the 2nd, 3rd, 5th, 6th, and 7th accesses were seen on the memory bus.

2. The processor has a 512 byte, 2-way set associative, write-through cache with perfect LRU replacement. The line size is 16 bytes.

3. A snapshot of the tag store before the 1st of the seven reads and a snapshot of the tag store after the 7th read are shown below. The only part shown in a tag store entry is the tag. You can assume it is valid. Many of the tag store entries unnecessary to solving this problem are not shown in order to reduce the clutter.

BEFORE

	Way 0	Way 1	LRU
Set 0	1	4	1
Set 1			0
Set 2	6	5	0
Set 3			0
Set 4			0
Set 5			0
Set 6			0 / 1
Set 7			0
Set 8			0
Set 9			0
Set 10	9	8	0
Set 11			0
Set 12			0
Set 13			0
Set 14			0
Set 15			0

problem with the LRU bit of set 6. Not solveable this way.

AFTER

	Way 0	Way 1	LRU
Set 0	1	4	1
Set 1			0
Set 2	1	4	1
Set 3			0
Set 4			0
Set 5			0
Set 6	3	4	0
Set 7			0
Set 8			0
Set 9			0
Set 10	9	8	0
Set 11			0
Set 12			0
Set 13			0
Set 14			0
Set 15			0

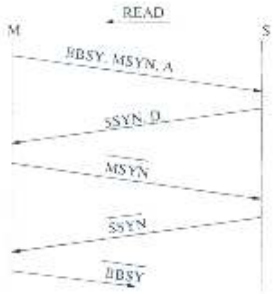
Name: _____

Problem 4 (20 points)

Two processors (P0 and P1) and one memory unit share an asynchronous bus. Either processor can request data at an address. If the other processor has the data, it supplies it. If not, the memory supplies it.

Your job in this problem will be to construct (a) the transaction timing diagram to show data supplied by the other processor, (b) the transaction timing diagram to show data supplied by the memory, (c) the asynchronous state diagram for each identical processor, and (d) the asynchronous state diagram for the memory controller.

Recall the transaction timing diagram discussed in class:

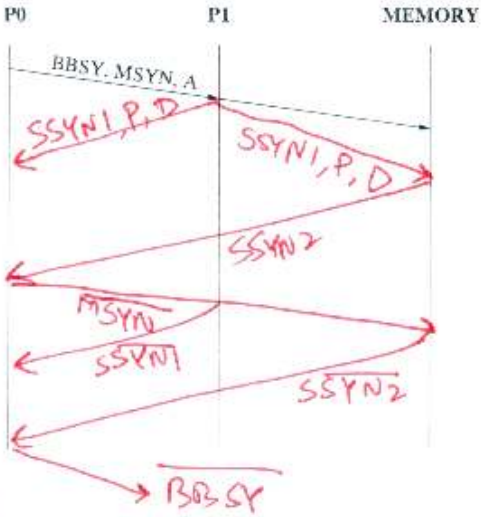


In our case, we have replaced the SSYN signal with a SSYN1 signal that each processor has and an SSYN2 signal that the memory controller has. We have also added a P signal denoting whether or not the "other" processor can supply the data requested.

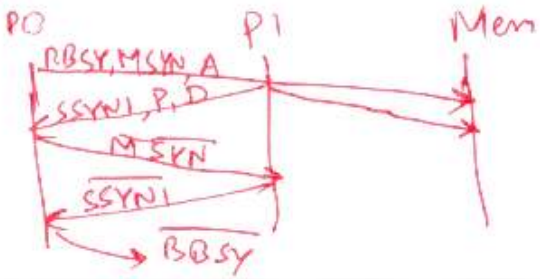
Complete the transation timing diagrams below, assuming P0 is the master:

This has 2 solns.

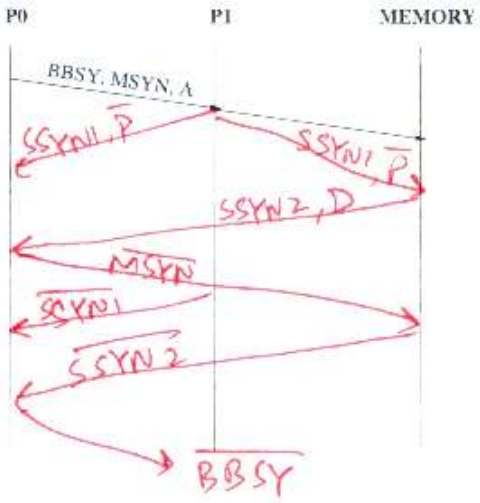
Part a (5 points): Show the Transaction Timing Diagram when data is supplied by the Processor P1.



2nd soln.:



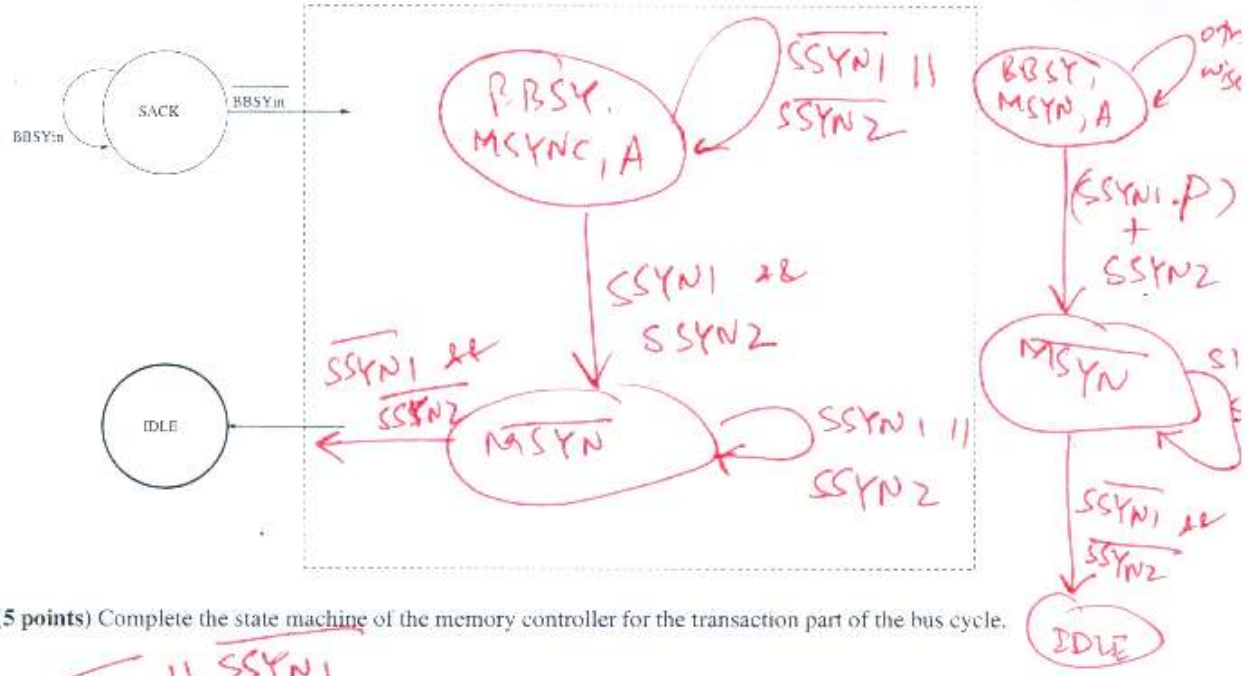
Part b (5 points): Show the Transaction Timing Diagram when data is supplied by the Memory.



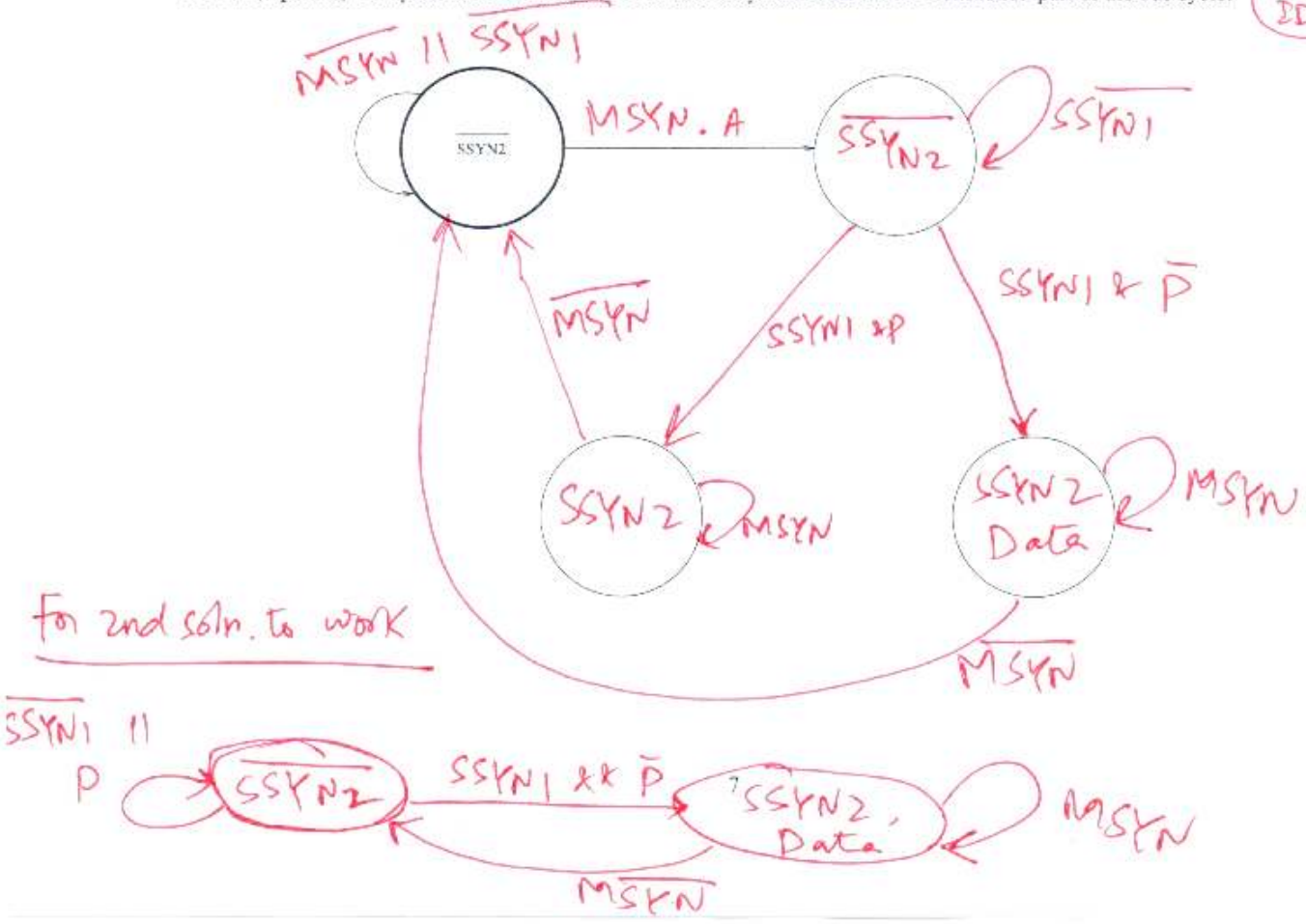
Name:_____

Problem 4 continued

Part c (5 points) Complete the state machine of the master for the transaction part of the bus cycle, once the master has secured the bus:



Part d (5 points) Complete the state machine of the memory controller for the transaction part of the bus cycle.



Name:_____

Problem 5 (25 points)

The diagram below shows a snapshot at a particular point in time of various parts of the microarchitecture for an implementation supporting out-of-order execution in the spirit of Tomasulo.

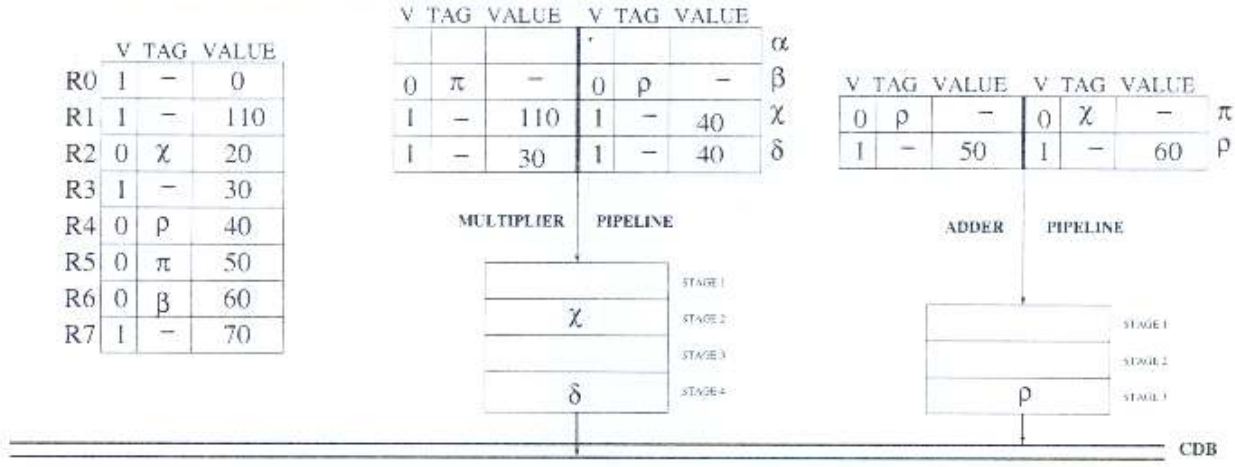
Note that both the adder and multiplier are pipelined, the adder has three stages and the multiplier has four stages. At the particular point in time when the snapshot was taken, the adder had only one instruction in the pipeline (i.e., in the final stage). The multiplier had two instructions in the pipeline, one in the final stage, one in stage 2. Note that the multiplier and adder have separate output data buses, which allow both an add result and a multiply result to update in the same cycle.

The adder has two reservation stations, the multiplier has four reservation stations. An instruction continues to occupy a reservation station slot until it completes execution and its result is written to its destination register. Assume no data forwarding.

The processor has been supplied with a six instruction program segment. Instructions are fetched, decoded and executed as discussed in class.

Your job: Complete the table identifying the six instructions, their opcodes (ADD, MUL), and their register operands (Rd, Rs1, Rs2). Also, show which stage of processing each instruction is in (F,D,E,WB) during each cycle. Finally identify the cycle when the snapshot of the microarchitecture was taken.

You might find it useful to start by attempting to draw a dataflow graph of the 6 instructions.



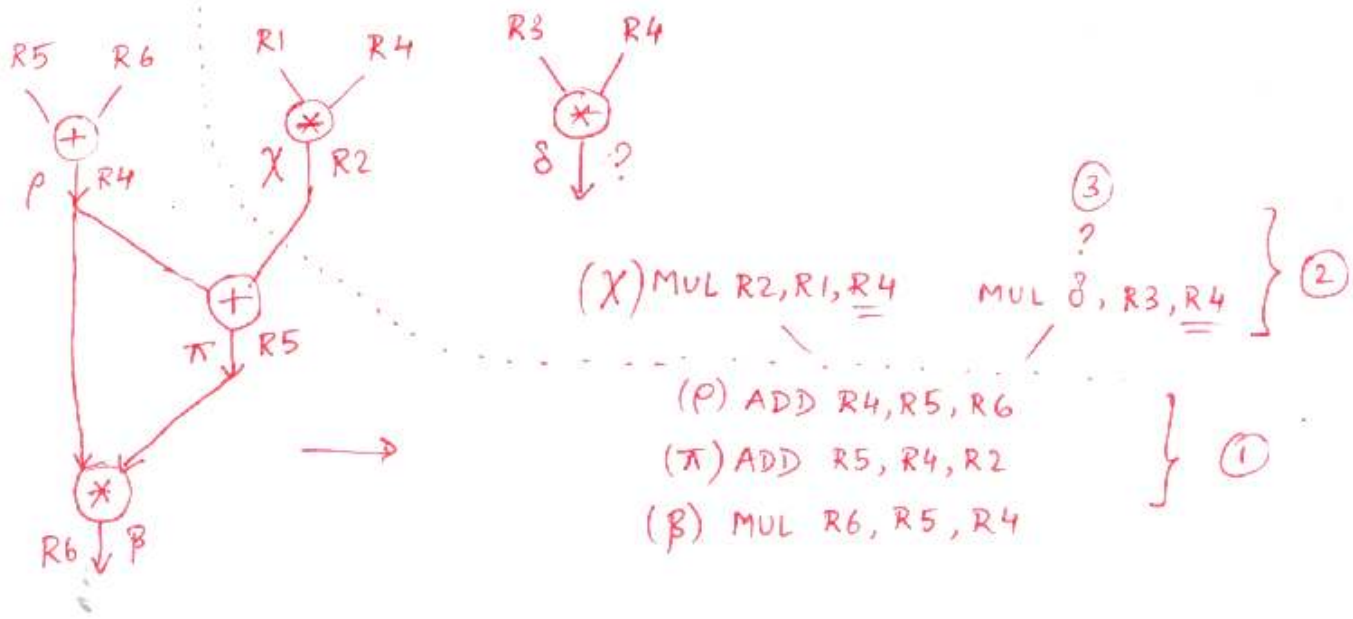
	cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	ADD R1, R5, R6	F	D	A	A	A	W																
χ	MUL R2, R1, R4		F	D	•	•	•	M	M	M	M	W											
δ	MUL R4, R3, R4			F	D	M	M	M	M	W													
ρ	ADD R4, R5, R6				F	D	A	A	A	W													
π	ADD R5, R4, R2					F	D	•	•	•	•	•	A	A	A	W							
β	MUL R6, R4, R5						F	D	•	•	•	•	•	•	•	•	M	M	M	M	W		

In which cycle was the snapshot taken?

8

Name:_____

Intentionally left blank



(1) From the dataflow graph

(2) ~~Tags δ and β were~~ These 2 MUL instructions came before ADD R4, R5, R6 because they source R4 = 40, not Tag ρ.

(3) Tag δ was over-written. So, δ = R2/R4/R5/R6. But the ADD R4, R5, R6 came after δ was used. So δ ≠ R5, δ ≠ R6.

δ = R2/R4.

If δ = R2, order must be

ADD R1, R5, R6

MUL R2, R3, R4 } so that R2 is

MUL R2, R1, R4 } overwritten

⋮

If δ = R4, order must be

ADD R1, R5, R6

MUL R2, R1, R4

MUL R4, R3, R4

⋮

Only δ = R4 satisfies the Timing ~~from~~ of the 2 MULs (δ and χ) shown in the MUL pipeline, in cycle 8.