CSARCH2 Mock Long Exam 3

by: Clive Jarel Ang and Enzo Arkin Panugayan

Important Reminders:

1. Read ALL instructions carefully and thoroughly before answering this mock exam.
2. The use of calculators and other computing devices are NOT allowed in the exam. However, you will be answering this in the comfort of your own home, so I literally have 0 control over the enforcement of that rule. ¯_(ツ)_/¯
3. **Cheating in ANY form during the actual exam will be considered a major offense, merit you a 0.0 in the course, and would result in both Sir Rog and I becoming very sad. (Please do NOT pass notes around!!!)**
4. This exam is GOOD FOR 4 HOURS. To be sufficiently prepared for the long exam proper, try to finish this mock exam in a shorter amount of time (while keeping a high score obv).
5. Yes, I'm sadistic and this mock exam reflects that. - Clive
6. I'm more sadistic than Clive and you will know what parts I wrote - Enzo

---

I. Cache Memory #1

A 33554432 Bit Direct Mapping Non-Load Through Policy Cache has a block size of 512 words. There are 262144 MM blocks. Each word has 32 bits. CPU fetches memory blocks (0-4000) 5 times. MAT is 10 ns.

a.) MM Address Bits	
b.) How many CM blocks	
c.) Tag Block Word	
d.) MM size in bits	
e.) CM size in bits (Including Tag and Valid Bit)	
f.) What CM block will Memory Address 111387 be stored in?	
g.) What CM block will Memory Block 111387 be stored in?	
h.) Total Access Time	
i.) Assume a 4 Way Set, CM size in bits	
j.) Assume a 8 Way Set, CM size in bits	
k.) Assume a 16 Way Set, CM size in bits	
l.) Hit and Miss in 1st Pass	
m.) Hit and Miss in 2nd Pass	
n.) Hit and Miss in 3rd Pass	

II. Microprogramming #1

1. Given these Assembly Code, identify which Assembly Code is correct and give the corresponding Microcode (If the Assembly Code is invalid, write Invalid)

   a. MOV RAX, RBX

   b. MOV RSI, [RSP * 4 + ALPHA]

   c. ADD [RBX], [RAX]

   d. CALL L3 (Assume L3 as Offset)

   e. CALL L3 (Assume L3 as Address)

   f. XOR [RBX], RAX + RBX

   g. JUMP L1

   h. SUB RSI, [RCX + RAX * 4 + ALPHA]

   i. IMUL RAX, RBX, 10

2. Theoretically we have a machine that has the ff. registers: RHP (Head Pointer) and RTP (Tail Pointer). Their purpose is similar to the RSP and RBP but for queues instead. ENQUEUE src will enqueue the src data into the RHP address and will increase the RHP. DEQUEUE dst will dequeue from the RTP address into the dst and will increase the RTP.

   a. What is the Microcode for ENQUEUE EAX

   b. What is the Microcode for DEQUEUE RAX

3. Theoretically we have a machine that does not have a SUB or NEG component in the ALU. How will the computer perform SUB RAX, RBX, represent in Microcode?

4. Theoretically we have a machine that does not have a MUL or Bitshift components in the ALU. We have a theoretical Assembly Code of MUL5 dst, src that will do the ff. RDX:dst $\leftarrow$ src * 5. 5 is not located in the IRDF.

   What will the Microcode for MUL5 RAX, RBX be?

5. Theoretically we have a machine that can accept the ff. microcode:

   a. ADD dst, src1, src2, src3, src4, src5

      where dst $\leftarrow$ src1 + src2 + src3 + src4 + src5

      What is the microcode for ADD RAX, RBX, RCX, RDX, RSI, RDI?

   b. QUADRUPLE src

      where RDX:src $\leftarrow$ src*4

      What is the microcode for QUADRUPLE RAX

c. BITWISE dst, src

where RAX $\leftarrow$ src1 $\wedge$ src2

RBX $\leftarrow$ src1 $\vee$ src2

RCX $\leftarrow$ src1 $\oplus$ src2

What is the microcode for BITWISE RDX, RAX

6. Theoretically these set of Microcode has a corresponding theoretical assembly code. Optimize the ff. Microcode while considering the order of operations to be equivalent.

a. RSPout, MARin, READ, WMFC
RSPout, SELECT 4, ADD, Zin
Zout, SELECT 4, ADD, Zin
Zout, RSPin
MDRout, RAXin, END

b. RSPout, SELECT 4, ADD, Zin
Zout, SELECT 4, ADD, Zin
Zout, RSPin
RSPout, MARin, READ
RBPout, RSPin, WMFC
RSPout, RBXin
MDRout, RAXin, END

c. RAXout, RBXin
RAXout, MARin, READ, WMFC
MDRout, SELECT 4, MUL, Zin
RBXout, Yin
Zout, SELECT Y, ADD, Zin
RAXout, RCXin
Zout, RDXin, END

7. Given the ff Microcode, what is the corresponding assembly code?
ASSUME that the default immediate address is ALPHA and immediate data is 0x1F

a. RDIout, MARin, READ, WMFC
MDRout, Yin
RSIout, SELECT Y, ADD, Zin
Zout, RSIin, END

b. ESIout, Yin

IRAFout, SELECT Y, ADD, Zin
Zout, MARin, READ, WMFC
MDRout, Yin
EAXout, SELECT Y, MUL, Zin
Zout, EDXin, EAXin, END

c. STATUSFLAGSout, AHin, END

d. RSPout, MARin, READ, SELECT 4, ADD, Zin
Zout, SELECT 4, ADD, Zin
Zout, RSPin, WMFC
MDRout, PCin, END

e. IRDFout, Yin
RBXout, SELECT Y, SUB, END

8. [Bonus] Theoretically we have a machine that has the ff. registers: RHP (Head Pointer), RTP (Tail Pointer), and RSX (Size Register). Their purpose is similar to the RSP and RBP but for lists instead while the RSX stores the size of the array by index. Note that each index only has 32 bit. The machine also has additional SELECT 2 option in the ALU-A Multiplexer.

APPEND src will append the src data into the RHP address and will increase the RHP.
REPLACE src1, src2 will replace the data of src1 into index src2
FETCH dst, src will fetch the data from index src into dst.
SIZE dst will store the size of the array into dst.
DELETE dst will store the data at the end of the list into dst and decrease the RHP.

a. What is the Microcode for APPEND RAX
b. What is the Microcode for REPLACE RAX, RBX
c. What is the Microcode for APPEND EBX
d. What is the Microcode for FETCH RAX, RCX
e. What is the Microcode for SIZE RDX
f. What is the Microcode for DELETE RAX

III. Signed Binary Multiplication #1

Multiply using pencil-and-paper method:

Operands	Signed Dec.	Bin. (least number of bits)
Multiplicand	43	
Multiplier	-22	

Intermediate	Partial Product
1	
2	
3	
4	
5	
6	
7	
8	
Sum	

IV. Signed Binary Multiplication #2

Given the following decimal numbers, determine the Booth's and Extended Booth's multipliers:

Multiplicand	102
Multiplier	-179

Multiplier in Booth's	
Multiplier in Extended Booth's	

V. Binary Multiplication #3 (Sequential Circuit Binary Multiplier)

Given the following operands:

Multiplicand	01110
Multiplier	10011

	A	Q
1st Pass		
2nd Pass		
3rd Pass		
4th Pass		
5th Pass		

VI. Restoring Division

Given the following decimal operands:

Dividend	63
Divisor	13

What is the value of A and Q at the end of each pass? Use the least number of bits when representing the numbers as unsigned integers.

	A	Q
1st Pass		
2nd Pass		
3rd Pass		
4th Pass		
5th Pass		
6th Pass		

VII. Non-Restoring Division

Given the following decimal operands:

Dividend	63
Divisor	13

What is the value of A and Q at the end of each pass? Use the least number of bits when representing the numbers as unsigned integers.

	A	Q
1st Pass		
2nd Pass		
3rd Pass		
4th Pass		
5th Pass		
6th Pass		