

16.482 / 16.561: Computer Architecture and Design

Spring 2014

Midterm Exam
March 13, 2014

Name: _____ ID #: _____

For this exam, you may use a calculator and two 8.5" x 11" double-sided page of notes. All other electronic devices (e.g., cellular phones, laptops) are prohibited. If you have a cellular phone, please turn it off prior to the start of the exam to avoid distracting other students.

The exam contains 6 questions for a total of 100 points. Please answer the questions in the spaces provided. If you need additional space, use the back of the page on which the question is written and clearly indicate that you have done so.

You will be provided with two pages (1 double-sided sheet) that contain a list of the MIPS instructions we have covered thus far. You do not have to submit these pages when you turn in your exam.

You will have three hours to complete this exam.

Q1: Evaluating instructions	/ 12
Q2: Binary multiplication	/ 14
Q3: IEEE floating-point format	/ 18
Q4: Pipelining	/ 14
Q5: Dynamic branch prediction	/ 20
Q6: Dependences and dynamic scheduling	/ 22
TOTAL SCORE	/ 100

1. (12 points) ***Evaluating instructions***

For each part of the following question, assume the following initial state. Note that your answers to each part should use the values below—your answer to part (a), for example, should not affect your answer to part (b).

- $\$t0 = 0x00000009$, $\$t1 = 0x00000004$, $\$t2 = 0x00200000$
- Contents of memory (all values are in hexadecimal)

Address				
0x00200000	02	25	20	10
0x00200004	9A	A9	BC	CB

For each instruction sequence below, list **all** changed registers and/or memory locations and their new values. When listing memory values, list the entire word—for example, if a byte is written to 0x00200000, show the values at addresses 0x00200000-0x00200003.

a. `addi $t4, $t0, -3`
`sub $t4, $t4, $t1`
`xor $t5, $t4, $t0`
`slt $t6, $t1, $t5`

b. `lbu $s0, 4($t2)`
`ori $s1, $s0, 0x8888`
`sll $s2, $s1, 4`
`sb $s2, 2($t2)`

2. (14 points) **Binary multiplication**

You are given $A = -2$ and $B = -6$. Assume each operand uses four bits. Show how the binary multiplication of $A * B$ would proceed using Booth's Algorithm.

3. (18 points) **IEEE floating-point format**

Multiply the two IEEE single-precision floating-point values 0xC0000000 and 0xC0400000. For full credit, you must show all work, including:

- Convert the two values into binary
- Perform the multiplication in binary (as done in Homework 2)
- Re-encode the result in IEEE single-precision format

4. (14 points) ***Pipelining***

Consider the following code sequence for both parts of this question.

```
loop:    add   $t0, $t1, $t2
         lw   $t3, 10($t0)
         lw   $t4, 14($t0)
         sub  $t5, $t4, $t3
         sw   $t5, 18($t0)
         addi $t2, $t2, 4
         slti $t6, $t2, 200
         bne $t6, $zero, loop
```

For both parts of this problem, show all work for full credit.

- a. (8 points) If we assume we have a pipelined datapath **without forwarding**, how many cycles will one loop iteration take?

4 (continued)

Again, consider the following code sequence:

```
loop:    add  $t0, $t1, $t2
         lw  $t3, 10($t0)
         lw  $t4, 14($t0)
         sub $t5, $t4, $t3
         sw  $t5, 18($t0)
         addi $t2, $t2, 4
         slti $t6, $t2, 200
         bne $t6, $zero, loop
```

- b. (6 points) If we now assume a pipelined datapath **with forwarding**, how many cycles will one loop iteration take?

5. (20 points) ***Dynamic branch prediction***

- a. (14 points) Say you are executing a program that contains two branches, as shown below. You are given the addresses of each branch in both decimal and hexadecimal.

<u>Address</u>			
<u>Decimal</u>	<u>Hex</u>		
8	0x08	loop	...
			...
20	0x14	BNE R4, R0,	else
			...
44	0x2C	BEQ R7, R8,	loop

Your processor contains a four-entry, 2-bit branch history table. Initially, entry 0 (the first line of the table) has the state 01, entry 1 is 11, entry 2 is 10, and entry 3 is 00.

Complete the table below to show which BHT entry is used to predict each branch, what predictions are made based on that entry, and how the state of each BHT entry changes throughout the program. You are given the actual outcome for each branch.

Loop Iteration	Branch	BHT Entry #	BHT Entry State	Pred.	Actual Outcome	New BHT Entry State
1	BNE				NT	
1	BEQ				T	
2	BNE				T	
2	BEQ				T	
3	BNE				NT	
3	BEQ				T	
4	BNE				T	
4	BEQ				NT	

5 (continued)

b. (6 points) Assume you have a (2,2) correlating predictor in the state shown below:

00	01	10	11
10	01	01	10
11	10	01	00
01	01	11	10

1	0
---	---

If we have a branch at address 40 (0x28 in hex), what entry of the predictor will we access, and what will the prediction be? As part of your answer, circle the appropriate entry above, and briefly explain how we determine which entry to access.

6. (22 points) **Dependences and dynamic scheduling**

Answer the following questions about the code sequence below:

```
I0:      ADD.D      F4, F2, F6
I1:      L.D       F8, 0(R1)
I2:      L.D       F2, 8(R1)
I3:      MUL.D     F6, F8, F4
I4:      ADD.D     F4, F6, F2
I5:      DADDUI    R1, R1, #16
I6:      S.D      F4, 0(R1)
I7:      SLTI     R2, R1, #160
I8:      BNEZ     R2, I0
```

a. (8 points) List all true data dependences in this code. Assume the branch at the end of the loop is taken at least once. List your dependences in the form:

<register number>:<producing inst.> → <consuming inst.>

For example, a dependence involving R1 between “I2” and “I3” would be listed as:

R1: I2→I3

Your list should only contain true dependences—do not list any name dependences.

6 (continued)

b. (9 points) Complete the pipeline diagram below to show how one iteration of the loop shown in part (a) is executed on a dynamically scheduled processor without speculation. Assume the following latencies, which refer to the number of execution cycles unless otherwise noted:

- 3 cycles (1 EX, 2 MEM) for L.D and S.D
- 2 cycles for ADD.D and SUB.D
- 6 cycles for MUL.D
- 2 cycles for DADDUI
- 1 cycle for all other instructions

Note that your solution may not use all 20 cycles shown below, but it should not use more than 20 cycles.

Inst.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ADD.D F4, F2, F6																				
L.D F8, 0(R1)																				
L.D F2, 8(R1)																				
MUL.D F6, F8, F4																				
ADD.D F4, F6, F2																				
DADDUI R1, R1, #16																				
S.D F4, 0(R1)																				
SLTI R2, R1, #160																				
BNEZ R2, Loop																				

6 (continued)

- c. (5 points) Explain the difference between instruction *completion* and instruction *commit*. Why must instructions go through these two different cycles in a processor using speculative execution?