16.482 / 16.561: Computer Architecture and Design

Fall 2013

Midterm Exam

October 21, 2013

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: Computer performance	/ 14
Q2: Evaluating instructions	/ 16
Q3: Binary multiplication	/ 18
Q4: IEEE floating-point format	/ 18
Q5: Pipelining	/ 18
Q6: Dynamic branch prediction	/ 16
TOTAL SCORE	/ 100

1. (14 points) ***Computer performance***

We use two compilers to generate machine code for the same program, on the same processor, which has a 10 ns cycle time. The instruction breakdown is shown below:

Compiler	ALU instructions	Loads	Stores	Branches
1	30	45	15	10
2	80	45	20	55

Assume that ALU instructions take 1 cycle, loads and stores each take 4 cycles, and branches take 2 cycles. **For all parts of this problem, show all work for full credit.**

a. (8 points) Calculate the average CPI for each code sequence.

b. (6 points) Which sequence runs faster? By how much?

2. (16 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).

- `$s0 = 0x00100000, $t0 = 0x00000006, $t1 = 0x00000007`
- Contents of memory (all values are in hexadecimal)

Address				
0x00100000	0C	15	27	30
0x00100004	FF	27	DD	CC

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 0x00100000, show the values at addresses 0x00100000-0x00100003.

a. (8 points)

```
lh      $t2, 6($s0)
add     $t3, $t0, $t1
addi    $t4, $t1, -6
sub     $t5, $t3, $t4
```

b. (8 points)

```
ori     $s1, $t0, 0xFFF0
sll     $s2, $s1, 16
sra     $s3, $s2, 16
sb      $t8, 2($s0)
```

3. (18 points) ***Binary multiplication***

You are given A = -7 and B = 3. Assume each operand uses four bits. Show how the binary multiplication of A * B would proceed using Booth's Algorithm.

4. (18 points) ***IEEE floating-point format***
For each part of this problem, show all work for full credit.

a. (9 points) Convert the decimal value 7.75 into single-precision floating-point format.

b. (9 points) Convert the single-precision floating-point value 0xC1280000 into decimal.

5. (18 points) ***Pipelining***

Consider the following code sequence. Assume both branches are not taken—all instructions in the loop are executed:

```
loop:     add $s3, $s0, $s2
          lbu $t0, 0($s3)
          beq $t0, $zero, end
          addi $t1, $zero, 90
          slt $t2, $t1, $t0
          bne $t2, $zero, end
          sb $t1, 0($s3)
          addi $s2, $s2, 1
          sw $s2, 0($s1)
          j loop
end:      ...
```

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

a. (10 points) If we assume we have a pipelined datapath **without forwarding**, how long will one loop iteration take?

5 (continued) Again, consider the following code sequence, and assume both branches are not taken:

```
loop:     add $s3, $s0, $s2
          lbu $t0, 0($s3)
          beq $t0, $zero, end
          addi $t1, $zero, 90
          slt $t2, $t1, $t0
          bne $t2, $zero, end
          sb $t1, 0($s3)
          addi $s2, $s2, 1
          sw $s2, 0($s1)
          j loop
end:      ...
```

b. (8 points) If we now assume a pipelined datapath **with forwarding**, how many cycles will the code take?

6. (16 points) ***Dynamic branch prediction***

a. (10 points) Your processor executes a program containing the high-level code snippet below:

```
for (i = 0; i < 2; i++) {
     for (j = 0; j < 2; j++)
     { <body of loop> }
}
```

When compiled, this code contains two branches, as shown below. The BNE controls the end of the inner loop (with index variable `j`). The BEQ controls the end of the outer loop (with index variable `i`). You are given the addresses of each branch in both decimal and hexadecimal.

Address (Decimal)	Address (Hex)		
4	0x04	`loop1`	…
			…
16	0x10	`loop2`	…
			…
28	0x1C		`BNE R4, R0, loop2`
			…
56	0x38		`BEQ R7, R8, loop1`

Your processor contains a 2-bit branch history table with 8 entries. All entries are initially set to 11. Complete the table shown below, which tracks the predictions made by this predictor for the code above. Remember that "T" stands for "taken" and "NT" for "not taken".

Outer Loop Iteration	Inner Loop Iteration	Branch	BHT Entry #	BHT Entry Value	Pred.	Actual Outcome	New BHT Entry Value
1	1	BNE		11	T	T	11
1	2	BNE		11	T	NT	
1	-	BEQ		11	T	T	11
2	1	BNE				T	
2	2	BNE				NT	
2	-	BEQ		11	T	NT	

b. (6 points) Assume you have a 4 entry (2,2) correlating predictor. For each part of this question, you are given a single line of the predictor, which is used to predict the given branch, as well as the current global history. For the given branch outcome, determine the prediction, new predictor state, and new global history.

i. **Predictor entries:**

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

Current global history:

0	**1**

Branch outcome: Not taken

Prediction?

New predictor state?

New global history?

ii. **Predictor entries:**

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

Current global history:

0	**0**

Branch outcome: Taken

Prediction?

New predictor state?

New global history?