# 16.482 / 16.561: Computer Architecture and Design

Fall 2014

Midterm Exam
October 16, 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. You do not have to submit this sheet when you turn in your exam.

You will have three hours to complete this exam.

| | |
|---|---|
| Q1: Evaluating instructions | / 8 |
| Q2: Binary multiplication | / 16 |
| Q3: IEEE floating-point format | / 20 |
| Q4: Datapaths and pipelining | / 16 |
| Q5: Dynamic branch prediction | / 21 |
| Q6: Dynamic scheduling | / 19 |
| **TOTAL SCORE** | / 100 |

1. (8 points) ***Evaluating instructions***

Assume the following initial state prior to executing the instructions below. Note that the result of each instruction may depend on prior instructions.

- `$t4 = 0x0000000A, $t5 = 0x00000004, $t6 = 0x00101000`
- Contents of memory (all values are in hexadecimal)

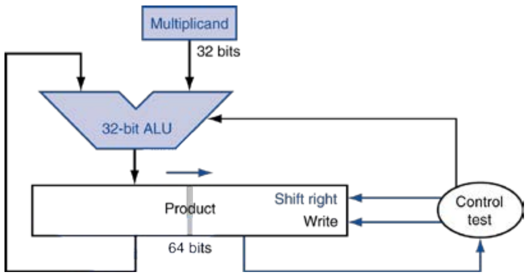
| **Address** | Lo | | | Hi |
|---|---|---|---|---|
| 0x00101600 | AA | 14 | 92 | 44 |
| 0x00101604 | 08 | 22 | 11 | 13 |

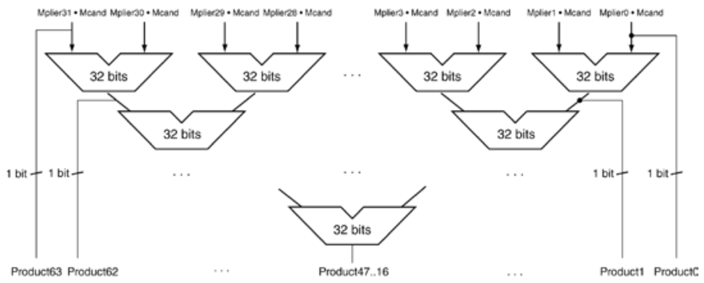
For each instruction below, list the changed register or memory location(s) and its new value.

```
   add  $s0, $t4, $t5

   ori  $s0, $s0, 0xCC11

   sh   $s0, 0x606($t6)

   lb   $s1, 0x600($t6)

   slt  $s2, $t4, $t5

   bne  $s2, $zero, L

   sra  $s1, $s1, 8

L: addi $s1, $s1, -1
```

2. (16 points) ***Binary multiplication***

a. (6 points) The two multipliers discussed in class are shown below:





Describe one major advantage of the optimized multiplier (on the left), and one major advantage of the tree multiplier (on the right).

b. (10 points) You are given A = -6 and B = 3. Assume each operand uses four bits. Show how the binary multiplication of A * B would proceed using the optimized multiplier.

3. (20 points) ***IEEE floating-point format***

Multiply the two IEEE single-precision floating-point values 0x40900000 and 0xc1480000. For full credit, you must show all work, including:

- Convert the two values into binary
- Perform the multiplication in binary, not decimal
- Re-encode the result in IEEE single-precision format

4. (16 points) ***Datapaths and pipelining***

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

a. (4 points) You are given two simple processors—one using a single-cycle design, the other a basic 5-stage pipeline—in which the instruction stages take the following amount of time:

- Fetch (IF): 20 ns
- Decode (ID): 25 ns
- Execute (EX): 30 ns
- Memory access (MEM): 40 ns
- Write back (WB): 25 ns

How much time does each processor take to execute a single instruction, from start to finish?

b. (4 points) Now, assume each processor is redesigned with a faster ALU that reduces the time for the EX stage to 20 ns. How much time does a single instruction take on the redesigned processors? Explain your answer.

4 (continued)

c. (8 points) Consider the following code sequence:

```
lw   $t0, 0($s1)
add  $t2, $t0, $s0
lw   $t1, 8($s1)
add  $t3, $t1, $s0
xor  $t4, $t2, $t3
sw   $t4, 16($s1)
and  $t5, $t2, $t3
sw   $t5, 16($s1)
```

Determine the time required to execute this sequence on the pipelined processor from part (b), assuming that processor uses a 5-stage pipeline with forwarding. Express your answer in ns, not cycles.

5. (21 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. Note that these branches are inside a loop, but neither one controls the number of loop iterations.

```
   Address
Decimal  Hex
    148  0x94    BEQ $t0, $s0, label1
                 …
    236  0xEC    BNE $t5, $t6, label2
```

Your processor contains a sixteen-entry, 2-bit branch history table. Initially, entries 0-7 (the first eight lines of the table) all have the state 10, and entries 8-15 (the last eight lines of the table) all have the state 01.

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 | BEQ | | | | T | |
| 1 | BNE | | | | NT | |
| 2 | BEQ | | | | NT | |
| 2 | BNE | | | | T | |
| 3 | BEQ | | | | NT | |
| 3 | BNE | | | | NT | |
| 4 | BEQ | | | | T | |
| 4 | BNE | | | | T | |

5 (continued)

b. (4 points) How many entries are in an (8, 2) correlating branch predictor if the predictor uses 6 bits from each branch's address to choose the appropriate row within the predictor? Show all work to justify your answer.

c. (3 points) Explain the purpose of a branch target buffer (BTB).

6. (19 points) ***Dynamic scheduling***

a. (3 points) What potential problems with dynamic scheduling are prevented using register renaming?

b. (3 points) Explain why instructions in dynamically scheduled processors do not have to execute the exact same stages for each instruction (for example, only loads and stores access memory, branches do not have a write back stage, etc.).

c. (3 points) In a dynamically scheduled processor with speculation, why must instructions commit in order, even though they are allowed to complete out of order?

a. (10 points) Complete the pipeline diagram below to show how the given code is executed on a dynamically scheduled processor without speculation. Assume the following latencies, which refer to the number of execution cycles unless otherwise noted:

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

Also assume that the processor only contains one common data bus. 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `L.D F2,0(R1)` | | | | | | | | | | | | | | | | | | | | |
| `MUL.D F6,F2,F0` | | | | | | | | | | | | | | | | | | | | |
| `L.D F4,8(R1)` | | | | | | | | | | | | | | | | | | | | |
| `SUB.D F8,F4,F0` | | | | | | | | | | | | | | | | | | | | |
| `ADD.D F10,F6,F8` | | | | | | | | | | | | | | | | | | | | |
| `S.D F10,-8(R1)` | | | | | | | | | | | | | | | | | | | | |
| `S.D F8,16(R1)` | | | | | | | | | | | | | | | | | | | | |
| `DADDUI R1,R1,32` | | | | | | | | | | | | | | | | | | | | |
| `SLTI R2,R1,640` | | | | | | | | | | | | | | | | | | | | |
| `BNE R2,R0,Loop` | | | | | | | | | | | | | | | | | | | | |