Ryerson University
Department of Electrical and Computer Engineering
COE 818–Advanced Computer Architecture

Midterm Test

Name: ___________________ Student Number: ___________________

Time limit: 1 hour 50 min

Notes:

a) Closed book.
b) No calculators.
c) Answer all questions in the space provided.

March 1, 2013
Examiners: N. Mekhiel

Q1- Assume the following C code :

```c
for(i=0; i<=1000; i++) {
    A[i] = B[i]*X + 1;
}
```

- A- Convert the C code to MIPS Assembly code assume all variables are FP. (5 MARKS)

```
loop: 
    ld $f1, 0(R1)   # B[i]
    mul $f2, $f1, $f3  # B[i] * X
    add $f4, $f1, $f2  # A[i]
    # Add instructions for A[i]
    sub $r3, $r1, $r2  # R3 = R1 - R2
    bnez $r3, loop
```

Each mistake - 1
B- Find the memory bandwidth of the above MIPS code assuming A[i], B[i] and X are 64 bit FP (using MIPS I, R, J instructions). (5 marks)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lw $f0, o$(r1)</code></td>
<td>32 = 4B</td>
</tr>
<tr>
<td><code>mult</code></td>
<td>4B</td>
</tr>
<tr>
<td><code>addi</code></td>
<td>4B</td>
</tr>
<tr>
<td><code>sd</code></td>
<td>8B</td>
</tr>
<tr>
<td><code>addi</code></td>
<td>4B</td>
</tr>
<tr>
<td><code>addi</code></td>
<td>4B</td>
</tr>
<tr>
<td><code>sub</code></td>
<td>4B</td>
</tr>
<tr>
<td><code>bnz</code></td>
<td>4B</td>
</tr>
</tbody>
</table>

\[
= \frac{8 \times 4 + 8 \times 2}{8 \text{ cycles}} = 6 \text{ B/cycles}
\]

C- Find if it is a good idea to implement a new instruction that could replace two instructions: Multiply and Increment by one instruction as:

MULTI F1, F2, F3; means F1=F2*F3 + 1

Use the above code and assume that the processor speed will be reduced to 80% of original speed when the new instruction is implemented. Assume each instruction takes 1 clock cycle. (5 marks)

without new instruction = \[8 \times 1000 \times 1 = 8000 \text{ cycles}\]

with new instruction = \[7 \times 1000 \times \frac{10}{8} = 8700 \text{ slower}\]
Q2- (total = 10 marks)
A- Find if the following accesses are aligned or not assuming R1=1000 :- (4 MARKS)

- LB R2, 11(R1); Load a byte
  \[ 1011 \mod 4 = 3 \ \text{aligned} \]

- LH R2, 17(R1); Load half word
  \[ 1017 \mod 2 = 1 \ \text{not aligned} \]

- LW R2, 18(R1); Load a word
  \[ 1018 \mod 4 = 2 \ \text{not aligned} \]

- LD F2, 100(R1); Load double word
  \[ 1103 \mod 8 = 6 \ \text{not aligned} \]

B- How does the architecture deal with nonaligned accesses (2 MARKS)

- Two memory accesses
- Need alignment network inside processor

C- How can the compiler optimize performance of general purpose register LD/SD ISA. (2 marks)

- Assign variables to registers using graph coloring
- Rescheduling to get rid of hazards
- Minimize code size
- Use fast instruction
D-List advantages and disadvantages of using the following options in ISA:- (2 marks)

- Using 16 bits rather than 32 bits for immediate
  16 bits cost less, small size instruction
  32 bits more use, simplifies compiler

- Using callee save versus caller save in subroutines
  callee save more optimized for some
  application
  caller: more conservative and reliable

Q3-PIPELINING:-

A-List all types of hazards in MIPS 5 Stages pipeline and give an example with code that could cause each type. (6 marks)

1. Structural hazard between IM, DM
   lw R1, 0(CR2)  
   add R2, R1/R3

2. Data hazard
   lw R1, 0(CR2)
   add R3, R1/R3

3. Control
   beq Z R1, loop
   instruction
   loop
C- Explain how MIPS pipeline handle FP operations and list all types of issues associated with this solution. (4 marks)

Using multi-cycle operation in execute issues:
1. Complicates exception
2. Creates structural hazard
3. Creates WAW hazard

Q4-Show the timing of the following code sequence in MIPS pipeline assume forwarding, scheduling and branch uses delayed branch and all memory references are in cache (FP Add takes 4 cycles). (10 marks)

```
LOOP: LW R3, 0(R1)
       LW R4, 0(R2)
       ADD R5, R4, R3
       ADDI R6, R5, #1
       SW R6, 0(R1)
       ADDI R1, R1, #4
       ADDI R2, R2, #4
       SUBI R8, R7, R2
       BNZ R8, LOOP
```

| INSTRUCTIONS | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| LW R3, 0(R1)|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LW R4, 0(R2)|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ADD R5, R4, R3|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ADDI R6, R5, #1|   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SW R6, 0(R1)|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ADDI R1, R1, #4|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ADDI R2, R2, #4|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SUBI R8, R7, R2|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BNZ R8, LOOP|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SW R6, 0(R1)|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

9 cycles