Wednesday, April 7, 2004

SOLUTIONS!!!

Name (print): ____________________________________________________________

Signed: __________________________________________________________________

I pledge my honor that I have neither given nor received aid on this work.

GENERAL INSTRUCTIONS

1. This is an open book, closed notes exam. You are allowed one (8.5-inch × 11-inch) sheet of paper for formulas. Use a pencil to complete your work. Follow the instructions as given in the problem. Calculators are allowed for the exam.

2. Please do all of your work on the exam itself. You may use the backs of the pages, if necessary. Clearly indicate the continuation of your work on other pages.

3. Show work for maximum partial credit.

4. Please be as neat and as well-organized as possible.

5. Clearly indicate your answers.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>MAXIMUM</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
PROBLEM 1: (30 pts)

Consider a processor that is using the dynamic branch prediction scheme described in Figure 3.7 on page 198. A portion of the branch target buffer (BTB) and a portion of the branch history table (BHT) are shown below. For each part of this problem, assume the BTB and BHT begin in this state (i.e. no changes that occur to these structures in any part of the problem carry over to other parts of the problem). Also assume that the BTB and the BHT are very large so there are no capacity replacements.

<table>
<thead>
<tr>
<th>BRANCH HISTORY TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch Instruction Address (BIA)</td>
</tr>
<tr>
<td>4810</td>
</tr>
<tr>
<td>4820</td>
</tr>
<tr>
<td>5628</td>
</tr>
<tr>
<td>6432</td>
</tr>
<tr>
<td>8148</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BRANCH TARGET BUFFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch Instruction Address (BIA)</td>
</tr>
<tr>
<td>6432</td>
</tr>
<tr>
<td>8156</td>
</tr>
<tr>
<td>4820</td>
</tr>
<tr>
<td>5628</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

The label “foo” used in the instructions in Part A through Part E refers to memory address 5628. Register R4 contains the value 0. Register R5 contains the value of 5.

Complete Part A through Part E as directed.
Part A (8 pts)
Determine whether the branch is predicted and what (if any) changes are made to the BTB and the BHT. (Each instruction is preceded by its memory address.)

4820 BEQ R4, R0, foo

Is this branch predicted taken? YES / NO (circle one)

Changes to BTB (fill in one of the three choices as required):
- Insert Entry:
  BIA: ____________ BTA: ____________
- Delete Entry:
  BIA: ____________ BTA: ____________
- No change to BTB? X

Changes to BHT (fill in one of the two choices as required):
- Updated BHT Entry:
  BIA:  ____________ Prediction State: WNT
- No change to BHT? X

Part B (8 pts)
Determine whether the branch is predicted and what (if any) changes are made to the BTB and the BHT. (Each instruction is preceded by its memory address.)

8148 BNE R4, R5, foo

Is this branch predicted taken? YES / NO (circle one)

Changes to BTB (fill in one of the three choices as required):
- Insert Entry:
  BIA: ____________ BTA: ____________
- Delete Entry:
  BIA: ____________ BTA: ____________
- No change to BTB? X

Changes to BHT (fill in one of the two choices as required):
- Updated BHT Entry:
  BIA: 8148 Prediction State: WNT
- No change to BHT? ________
Part C (8 pts)
Determine whether the branch is predicted and what (if any) changes are made to the BTB and the BHT. (Each instruction is preceded by its memory address.)

6432 BEQ R5, R0, foo

Is this branch predicted taken? YES / NO (circle one)

Changes to BTB (fill in one of the three choices as required):
- Insert Entry:
  | BIA: | BTA: |
- Delete Entry:
  | BIA: 6432 | BTA: 5628 |
- No change to BTB? _____

Changes to BHT (fill in one of the two choices as required):
- Updated BHT Entry:
  | BIA: 6432 | Prediction State: SNT |
- No change to BHT? _____

Part D (3 pts)
Is the instruction at label “foo” a branch instruction? Please explain your answer.

Yes “foo” is a branch instruction. It represents address 5628 which appears in the BHT. A branch may not necessarily appear in the BTB, since not all branches are predicted taken.

Part E (3 pts)
Which men’s basketball team did you cheer for during the NCAA Championship game?

The Georgia Institute of Technology (Georgia Tech)
PROBLEM 2: (35 pts)

Part A (16 pts)
What is the effective access time of a cache memory system in which there is a direct-mapped Level 1 (L1) cache and an 8-way set-associative Level 2 (L2) cache? Use the following parameters for your calculations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 number of sets</td>
<td>128 sets</td>
</tr>
<tr>
<td>L1 block size</td>
<td>4 words</td>
</tr>
<tr>
<td>L1 cache access time</td>
<td>2 ns per block</td>
</tr>
<tr>
<td>L2 number of sets</td>
<td>128 sets</td>
</tr>
<tr>
<td>L2 block size</td>
<td>8 words</td>
</tr>
<tr>
<td>L2 cache access time</td>
<td>15 ns per block</td>
</tr>
<tr>
<td>Main memory access time</td>
<td>60 ns per word</td>
</tr>
<tr>
<td>Main memory size</td>
<td>512M words</td>
</tr>
<tr>
<td>L1 cache hit rate</td>
<td>96%</td>
</tr>
<tr>
<td>L2 cache hit rate</td>
<td>88%</td>
</tr>
</tbody>
</table>

\[
AMAT = \text{HitTime}_{L1} + \text{MissRate}_{L1} \times (\text{HitTime}_{L2} + \text{MissRate}_{L2} \times \text{MissPenalty}_{L2})
\]

\[
AMAT = 2\text{ns} + 0.04 \times [15\text{ns} + 0.12 \left(60 \frac{\text{ns}}{\text{word}} \times 8 \frac{\text{words}}{\text{block}}\right)]
\]

\[
AMAT = 4.904\text{ns}
\]

Part B (9 pts)
Label the fields of the memory address below used to access the L1 cache in Part A and indicate the size of each field in number of bits. Assume the memory is word-addressed.

<table>
<thead>
<tr>
<th>TAG</th>
<th>INDEX</th>
<th>OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 bits</td>
<td>7 bits</td>
<td>2 bits</td>
</tr>
</tbody>
</table>
Part C (5 pts)
Find the Average Memory Access Time (AMAT) of a 750 MHz machine with a miss penalty of 20 cycles, a hit time of 2 cycles, and a miss rate of 5%.

\[
AMAT = \text{HitTime} + \text{MissRate} \times \text{MissPenalty}
\]

\[
AMAT = (2\text{cycles} + 0.05 \times 20\text{cycles}) \times \frac{1}{750MHz}
\]

\[
AMAT = 4\text{ns}
\]

Part D (5 pts)
Suppose doubling the size of the cache in Part C decreases the miss rate to 3%, but causes the hit time to increase to 3 cycles and the miss penalty to increase to 26 cycles. What clock rate would be required to maintain the same AMAT?

\[
4\text{ns} = (3\text{cycles} + 0.03 \times 26\text{cycles}) \times \frac{1}{\text{ClockRate}}
\]

\[
\text{ClockRate} = 945MHz
\]
PROBLEM 3: (35 pts)

For the following problem, assume an in-order MIPS pipelined architecture that has functional units that take the following number of execution cycles:

- FP Multiplier: 4 cycles
- FP Adder: 2 cycles
- Integer operations: 1 cycle

Also assume: (1) there is one branch delay slot, (2) there is no delay between integer operations and dependent branch instructions, and (3) the load-use latency (number of cycles between the load and an instruction that uses the returned value) is 2 cycles. Assume that all functional units are fully pipelined and bypassed.

The following code computes a dot product. Assume that R1 and R2 contain addresses of arrays of floating point numbers, and R3 contains the length of the arrays in elements. Assume that F4 is initialized to zero.

```assembly
foo  L.D   F6, 0(R1) ;load X[i]
      L.D   F8, 0(R2) ;load Y[i]
      MUL.D  F10, F6, F8 ;multiply X[i] * Y[i]
      ADD.D  F4, F4, F10 ;add sum = sum + X[i] * Y[i]
      DADDI  R1, R1, #-8 ;decrement X index
      DADDI  R2, R2, #-8 ;decrement Y index
      DADDI  R3, R3, #-1 ;decrement element count
      BNEZ  R3, foo ;loop if not done
      NOP     ;branch delay slot
```

**Part A (5 pts)**

How many cycles does each iteration take, without arranging the code? 

14 cycles
**Part B (17 pts)**
Software pipeline the given loop so that it has **three iterations** overlapped simultaneously, and so that it has **no stalls**. Give the code for your solution below. You do not need to include the start-up or clean-up code.

```assembly
foo  ADD.D  F4, F4, F10  ;add sum = sum + X[i] * Y[i]
MUL.D  F10, F6, F8 ;multiply X[i] * Y[i]
L.D   F6, -16(R1) ;load X[i]
L.D   F8, -16(R2) ;load Y[i]
DADDI  R1, R1, #-8  ;decrement X index
DADDI  R3, R3, #-1 ;decrement element count
BNEZ  R3, foo   ;loop if not done
DADDI  R2, R2, #-8 ;decrement Y index
```

**Part C (5 pts)**
How many cycles does each iteration take for your software pipelined loop?

**8 cycles**

**Part D (8 pts)**
For the software-pipelined version of the loop, what are:
- the **maximum number of execution cycles for MUL.D**
- the **maximum number of execution cycles for ADD.D**
such that the loop runs without stalls? Please explain your answer.

**Maximum number of execution cycles for MUL.D = 7**
**Maximum number of execution cycles for ADD.D = 8**

The maximum number of execution cycles is based upon the number of instructions until that value is needed by another instruction. In the case for MUL.D, the value produced is used by the ADD.D instruction from the next iteration. The latency (number of cycles between the instruction and another instruction that uses the produced value) is 6 cycles, but the execution cycle count includes the cycle when the instruction is issued. In the case for ADD.D, the value produced is used by the ADD.D instruction from the next iteration as well. The latency is 7 cycles.