EE557--FALL 2000

MIDTERM 1 (makeup)

Closed books, closed notes
Time limit: 1 hour and 20 minutes MAX. No extension.

Q1: /15
Q2: /8
Q3: /6
Q4: /16

TOTAL: /45

Grade: /25
QUESTION 1. 15 points

Consider the 7-stage pipeline shown below.

The main difference between this pipeline and the 5-stage pipeline in H&P’s book is that execution through the ALU and data memory access each take two clocks: The two execution stages in the pipeline are called EX1 and EX2 and the two memory stages are called MEM1 and MEM2. Possible forwarding paths to the lower ALU input are also displayed. There is also forwarding inside the register file. Conditional branches are predicted not taken and, if taken, are executed in the MEM1 stage.

2.1. (3pts) Some of the forwarding paths are useless or even incorrect. Please indicate which ones by crossing them out on the figure. Do not add any forwarding paths, just remove incorrect ones.

2.2 (6pts) Consider the following program:

I1: LW $7,0($6)
I2: LW $8,0($4)
I3: ADD $7,$8,$7
I4: LW $9,0($7)
I5: BEQ $5,$9,Target
I6: SW $7,100($8)

Target:

List all data (RAW) dependencies on registers in this code fragment. For each dependency, give the two instructions and the register involved in the dependency.

----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
2.3 (6pts) Many data hazards remain in the pipeline. Since the hardware has no hazard detection to stall the pipeline, the compiler must insert NOOPs in the code to execute it correctly. Rewrite the program with the minimum number of NOOPs to avoid data hazards and execute the code correctly.
QUESTION 2 (8pts)

The following parameters apply to a system employing a 32-bit data CPU.

- Physical memory size: 16 MBytes (24 bits of physical address)
- Word size: 4 bytes. Addresses point to bytes.
- Cache size: 32KB

The cache set is first indexed with some bits of the physical address and then the cache tags are matched with some bits of the physical address.

The most significant bit of the physical address is P23 and the least significant bit is P0:

1. If the cache is direct-mapped and the block size is 64 bytes
   (2pt) Which bits of the physical address are used to index the cache:
   Answer: bits P___ to P___
   (2pt) Which bits of the physical address are used as tags in the cache?
   Answer: bits P___ to P___

2. If the cache is 8-way set-associative and the block size is 32 bytes
   (2pt) Which bits of the physical address are used to index the cache:
   Answer: bits P___ to P___
   (2pt) Which bits of the physical address are used as tags in the cache?
   Answer: bits P___ to P___
QUESTION 3 (6pts)

You are going to enhance a machine and there are two possible improvements: either make multiply instructions run four times faster than before, or memory access instructions run two times faster than before. You repeatedly run a program that takes 100 seconds to execute. Of this time, 20% is used for multiplication, 50% for memory access instructions and 30% for other tasks. What will the speedup be if you improve only multiplication? What will the speedup be if you improve only memory access? What will the speedup be if both improvements are made?

Multiplication only:

Speedup=

Memory access only:

Speedup=

Both multiplication and memory access:

Speedup=
QUESTION 4 (16pts)

Figure 2 shows a DLX floating point pipeline with multiple execution units. The address (integer) unit takes one clock. The FP adder/multiplier is fully pipelined and takes 3 clocks. The divider is pipelined in two stages and each stage takes 2 clocks.

1. (8pts) Consider the following program.

```
LOOP   LD F0,1000(R1)
       DVD F4,F0,F2
       SD F4,3000(R1)
       DVD F8,F0,F6
       SD F8,4000(R1)
       SUBI R1,8
       BNE R1,R0,LOOP
```

Forwarding is applied whenever possible. The branch is NOT delayed, is predicted not taken and is executed in the MEM stage if taken. In the following table please show the timing diagram for this loop by indicating the stage in each clock for each instruction in the table. Structural hazards on the write port of the register file are handled by issuing instructions only if they don’t conflict with previous instructions at the register file. This is done in the ID stage with a shift register.

Assuming that the loop is executed N times, what is the time in cycles needed to execute the entire loop as it is. Note that the last iteration of the loop will continue execution after the branch, and that useful instructions will be executed, so that the number of clocks associated with the last iteration is different from other iterations.

TIME = ____________ Clocks.
2. (8pts) Assume now that the branch is delayed by two cycles and that the compiler can schedule the code to improve performance. (No loop unrolling; just code scheduling within each iteration). The branch is still predicted not taken and is executed in the MEM stage if taken. Please show the new code and the schedule in Table 2.

<table>
<thead>
<tr>
<th>CODE</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
<th>T11</th>
<th>T12</th>
<th>T13</th>
<th>T14</th>
<th>T15</th>
<th>T16</th>
<th>T17</th>
<th>T18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assuming that the loop is executed N times, what is the time in cycles needed to execute the loop as it is. Note that the last iteration of the loop will continue execution after the branch, and that useful instructions will be executed, so that the number of clocks associated with the last iteration is different from other iterations.

TIME = _________________Clocks.
Figure 2. DLX pipeline with multiple execution units.