16.482 / 16.561: Computer Architecture and Design

Spring 2015

Lecture 5: Key Questions

February 19, 2015

1. What is the difference between a dependence and a hazard?

2. What are the three types of dependences and the hazards they can cause? You may want to list an example of each.

3. Dependences/Hazards: Given the code below:

```
Loop:      ADD  $1, $2, $3
I0:        ADD  $3, $1, $5
I1:        LW   $6, 0($3)
I2:        LW   $7, 4($3)
I3:        SUB  $8, $7, $6
I4:        DIV  $7, $8, $1
I5:        ADDI $4, $4, 1
I6:        SW   $8, 0($3)
I7:        SLTI $9, $4, 50
I8:        BNEZ $9, Loop
```

a. List all data dependences between instructions in the following format.

```
<register number>: <producing inst. label> → <consuming inst. label>
```

For example, the first dependence is on register `$1`, between "`Loop`" and "`I0`:"

```
$1: Loop → I0
```

b. Circle the dependences in part (a) that cause RAW hazards. Assume that the processor does not contain forwarding hardware, but that you do have the ability to write and read the register file in the same cycle.

c. List all name dependences that could cause WAR or WAW hazards, using a similar format to part (a). Since the notion of "producing" and "consuming" instructions doesn't apply, just list instructions in the order they appear in the code.

4. Describe the basic differences between a 5-stage pipeline and a more modern, realistic pipeline.

5. Describe the MIPS floating point registers and instructions.

6. Briefly describe how an instruction's latency can be used to determine the number of stalls between dependent instructions.

7. What is dynamic scheduling?

8. What are the key hardware components to implement Tomasulo’s Algorithm?

9. To understand what happens during each stage of Tomasulo's Algorithm, answer the following:

a. When does an instruction read its operands?

b. How does an instruction check to see if operands are available?

c. How does an instruction that is waiting for operands know when those operands are ready?

d. How is register renaming handled in Tomasulo's Algorithm?

e. How does an instruction check to see if it should write its result to the register file?

10. Register renaming: Given the following available reservation stations:
- Add1-Add4 (ADD.D/SUB.D)
- Mult1-Mult2 (MULT.D)
- Load1-Load2 (L.D)

Rewrite the code below with renamed registers, replacing register names with appropriate reservation stations. It may help to track the register result status for each instruction.

```
L.D     F2, 0(R1)
ADD.D   F0, F2, F6
SUB.D   F6, F0, F2
MULT.D  F2, F6, F0
DIV.D   F6, F2, F6
S.D     F6, 8(R1)
```

11. **Example:**

Follow the execution of the code below through all cycles, filling in the pipeline diagram below as you go. Use "S" to indicate a stall cycle. Assume that memory operations have a 2 cycle latency (1 EX, 1 MEM), floating point add/subtract instructions take 3 cycles, floating point multiply operations take 10 cycles, and floating point divide operations take 40 cycles.

Cycle	**1**	**2**	**3**															
`L.D F6,32(R2)`	IF	IS	EX															
`L.D F2,44(R3)`		IF	IS															
`MUL.D F0,F2,F4`			IF															
`SUB.D F8,F6,F2`																		
`DIV.D F10,F0,F6`																		
`ADD.D F6,F8,F2`																		

12. What are some advantages of Tomasulo’s algorithm?

13. What are some disadvantages of Tomasulo’s algorithm?