16.482 / 16.561: Computer Architecture and Design

Summer 2014

Lecture 6: Key Questions

June 10, 2014

1. What do we mean by “speculation?”

2. Why must we separate instruction completion from instruction commit in a processor that allows speculative execution?

3. What is a reorder buffer (ROB), and what is its purpose?

4. Describe the fields in each ROB entry.

5. Describe the differences in Tomasulo’s Algorithm when speculation is added.

Example: Follow the execution of the code below through all cycles, showing the appropriate state for each piece of hardware in Tomasulo's Algorithm with speculation. Fill in the tables provided. Assume 2 cycle latency (1 EX, 1 MEM) for loads/stores, 6 cycles for multiply, 2 cycles for integer addition, and 1 cycle for the branch.

Cycle	**1**	**2**	**3**																	
`L.D` `F0,0(R1)`	IF	IS	EX																	
`MUL.D` `F4,F0,F2`		IF	IS																	
`S.D` `F4,0(R1)`			IF																	
`DADDIU` `R1,R1,#-8`																				
`BNE` `R1,R2,Loop`																				
`L.D` `F0,0(R1)`																				
`MUL.D` `F4,F0,F2`																				
`S.D` `F4,0(R1)`																				
`DADDIU` `R1,R1,#-8`																				
`BNE` `R1,R2,Loop`																				

6. How does the reorder buffer help us avoid memory hazards?

7. How do we handle exceptions in a speculative machine?

8. Define fine-grained and coarse-grained multithreading.

9. Define simultaneous multithreading.

Multithreading example

Assume you are using a processor with the following characteristics:

- 4 functional units: 2 ALUs, 1 memory port (load/store), 1 branch
- In-order execution

Given the three threads below, show how these instructions would execute using:

- Fine-grained multithreading
- Coarse-grained multithreading
  - Switch threads on any stall over 2 cycles
- Simultaneous multithreading
  - Thread 1 is preferred, followed by Thread 2 and Thread 3

You should assume any two instructions without stalls between them are independent.

Threads:

Thread 1:	Thread 2:	Thread 3:
ADD.D	SUB.D	L.D
L.D	stall	stall
stall	L.D	stall
stall	S.D	stall
stall	L.D	stall
stall	stall	stall
SUB.D	ADD.D	stall
S.D	stall	ADD.D
stall	BNE	stall
BEQ		stall
		S.D
		stall
		stall
		BEQ

Extra space to work on multithreading example