EE 108B Review Session #4

Michelle Hewlett

Friday November 4, 2005

5:15 PM – 6:05 PM

Room Skilling 193

1

EE 108B Review Session Agenda

- Announcements
- HW 3 Topics Overview
- HW 3 Hints
- Sample Problems

2

Announcements

- HW 3 is due Tuesday November 8th
- Midterm Regrade Requests
  - Be sure to look at the solutions.
  - Attach a sheet of paper specifying the questions with explanations.
  - Submit with the original midterm to Darlene Hadding's office in Gates 408 by **11/9/2005, 5 PM**. SCPD students have 2 weeks, so we must receive them by **11/16/2005**.
  - We reserve the right to regrade the whole exam, including deducting points for something that was not originally caught.
  - For clerical mistakes (i.e mistake in totaling your score), take the regrade to the TA's OH and we'll update your score. SCPD students can fax us the exam.

3

HW 3 Topics

- Problems 1-4: Pipelining and Hazards
- Problem 5: Modifying the Pipeline
- Problems 6-7: Simple Cache Problems

4

HW 3 Hints

- Problem 1: Like sample problems # 2, 3, 4
- Problem 2: Sample problem # 1a will give some hints
- Problem 3: Work backwards from the 'sub' instruction and make a table with the following contents: Stage, Field, Width, Value (hex), Value (dec). Stages are: IF/ID, ID/EX, EX/MEM, MEM/WB
- Problem 4: Lecture notes and book should be a good guide
- Problem 5:
  - Part b(ii): seriously consider the hint!!
  - Part c: Consider how instruction count is influenced and CPI changes with new instruction…
- Problems 6 & 7: Similar to sample problems # 5

5

Sample Problems

1. Short Questions

   a. To increase throughput, the Intel Pentium 4 has a 20 stage pipeline while the AMD Athlon XP has only 9 stages. The Intel processor is therefore able to achieve a much higher throughput. In practice however, this is not necessarily the case. Give the **most significant** reason why the Intel Pentium 4 does not complete operations at twice the rate of the Athlon.

6

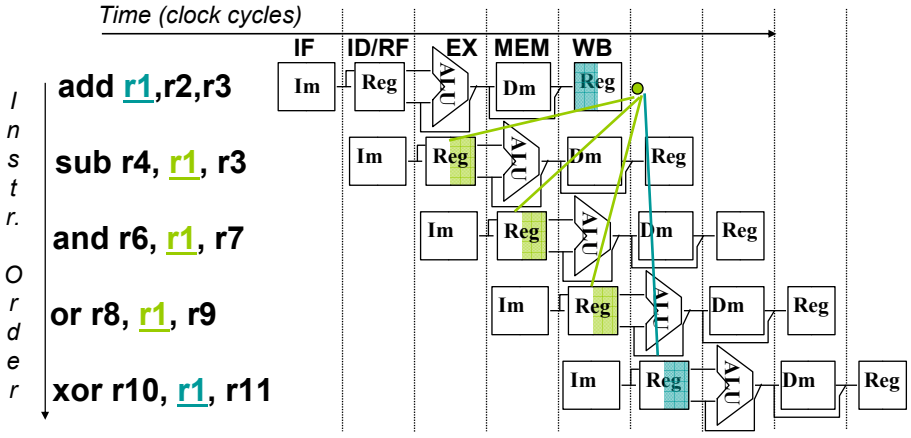
Sample Problems

b. You need to make a choice between a pipelined MIPS implementation without any bypassing or interlocking and one with bypassing and interlocking. You know exactly which programs you are going to execute. Why might you get better performance from the implementation without any bypassing or interlocking?

7

Dependencies vs. Hazards

- **Dependencies backwards in time are hazards**



8

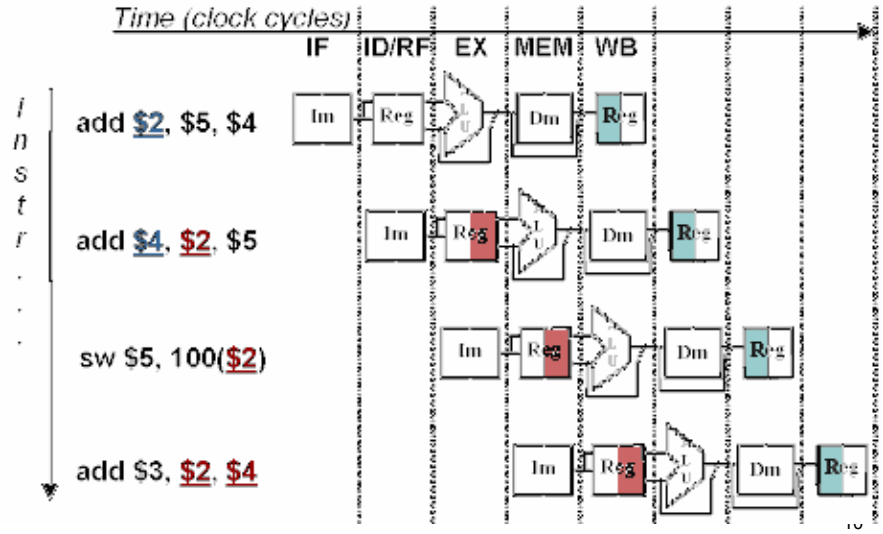
Sample Problem 2

Identify all of the **data dependencies** in the following code. Which dependencies are **data hazards** that will be resolved via forwarding?

```
add   $2, $5, $4
add   $4, $2, $5
sw    $5, 100($2)
add   $3, $2, $4
```

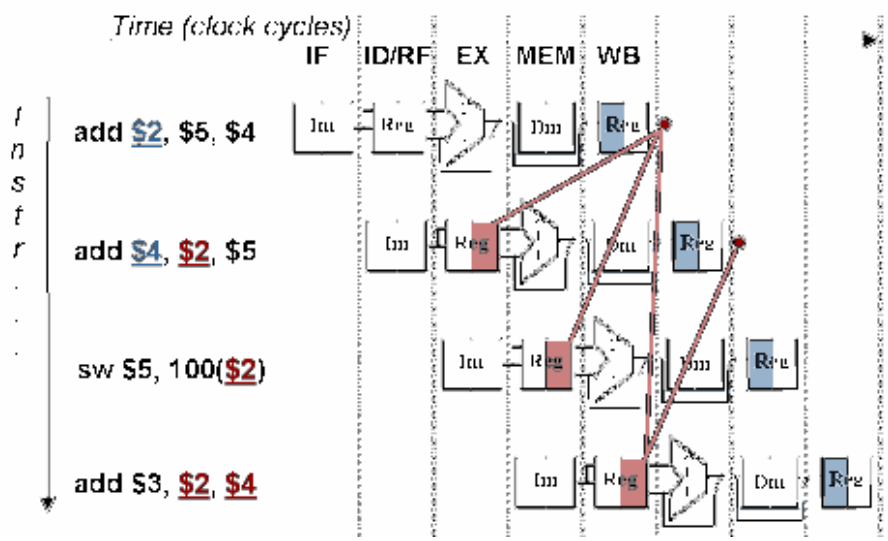
9

Sample Problem 2

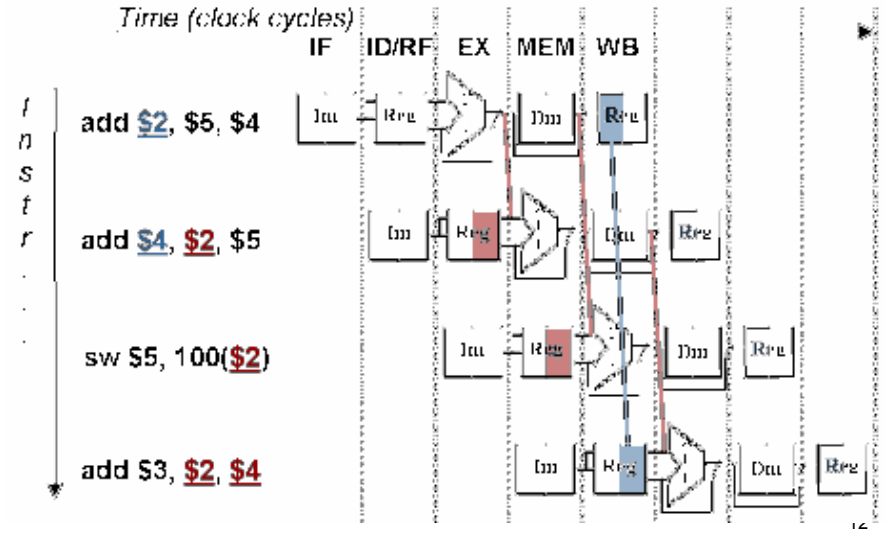


10

Problem 2: Data Dependencies



Problem 2: Forwarding



Sample Problem 3

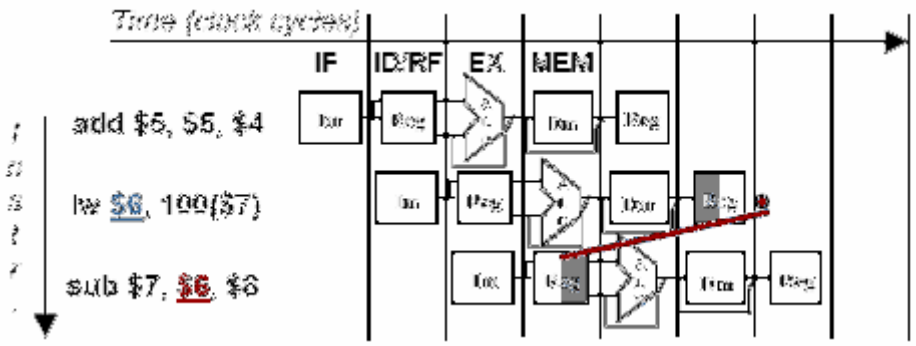
Consider executing the following code on the pipelined datapath

```
add $5, $6, $7
lw $6, 100($7)
sub $7, $6, $8
```

- How many cycles will it take to execute this code?
- What dependencies need to be resolved?
- Illustrate how the code will actually be executed (incorporating any stalls or forwarding) so as to resolve the identified problems.

13

Data Hazard: Back in time Dependency

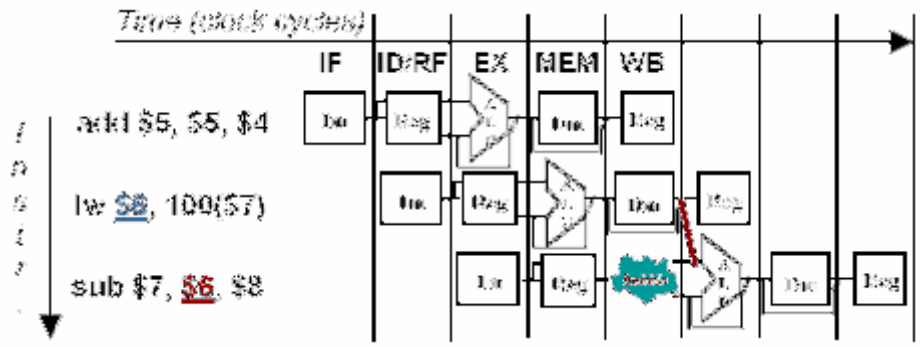


Do we really need $6 so early?

Do we actually have $6 so late?

14

1 stall is enough!



15

Sample Problem 4

Suppose the MIPS instruction set had one additional instruction: addm r1, (r2), r3 that added the contents of the memory location specified by r2 to the contents of r3 and placed the result in r1. With this instruction, a pipelined implementation might use the following 6-stage pipeline:

- IF: fetch the instruction
- RF: decode the instruction and read the registers
- EA: calculate the effective address using a special address adder
- MEM: access data memory if needed
- ALU: all ALU operations are performed here
- WB: write the result into the registers (assume that both the read and write of the registers take the entire clock cycle and the value is available only after the completion of this cycle).

16

Sample Problem 4

a. For this part assume that the machine has no bypassing or forwarding. Show that stalls would be encountered in executing the following code sequence, by annotating the following code sequence, if a stall is longer than one clock cycle, make sure you say how many cycles it is.

```
add     r1, r2,r3
addm    r4,(r5),r1
addm    r6,(r4),r1
lw      r7,0(r4)
```

17

Sample Problem 4a

	Cycle																
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
add r1,r2,r3																	
addm r4,(r5),r1																	
addm r6,(r4),r1																	
lw r7,0(r4)																	

18

Sample Problem 4

b. For this part assume that the machine has full bypassing or forwarding, and that results can be used as soon as they are available. Show what stalls would be encountered in executing the following code sequence, if a stall is longer than one clock cycle, make sure you say how many cycles it is.

```
add     r1, r2,r3
addm    r4,(r5),r1
addm    r6,(r4),r1
lw      r7,0(r4)
```

19

Sample Problem 4b

	Cycle															
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
add r1,r2,r3																	
addm r4,(r5),r1																	
addm r6,(r4),r1																	
lw r7,0(r4)																	

20

Sample Problem 5

a. Here is a set of address references given as word addresses: 3, 9, 15, 16, 1, 14, 3, 25. Assuming direct mapped cache with 8 two-word blocks that are initially empty. Label each reference as a hit or a miss and show the final contents of the cache.

21

Sample Problem 5a

3, 9, 15, 16, 1, 14, 3, 25

Index	Block[0]	Block[1]
0		
1		
2		
3		
4		
5		
6		
7		

Address	Hit or Miss?
3	
9	
15	
16	
1	
14	
3	
25	

22

Sample Problem 5

b. Now perform the same task again for two-way set associative with one-word blocks. Label each reference as a hit or a miss and show the final contents of the cache.

Sample Problem 5b

3, 9, 15, 16, 1, 14, 3, 25

Index	Way[1]	Way[0]
0		
1		
2		
3		
4		
5		
6		
7		

Address	Hit or Miss?
3	
9	
15	
16	
1	
14	
3	
25	