COMP2611: Computer Organization

The Pipelined Processor

The Pipelined Processor

Pipeline hazards

- exercises

Re-ordering instructions

- exercises

Forwarding and pipeline stalls

- exercises

Exercises

Exercises

Question 1: Suppose that the pipeline (IF, ID, EXE, MEM, WB) is used. Identify any pipeline hazards in each of the following sequences of MIPS instructions. Also, specify the type (structural, data or control) of the hazard, if any, and explain its cause.

(a)

```
1: sw $s1, 0($t0)
2: add $s2, $s0, $s1
3: add $s2, $s3, $s4
4: add $s1, $s5, $s6
```

Solution: Both the stage MEM in 1 and the stage IF in 4 need to access the memory, causing a structural hazard.

Exercises

(b)

```
1: sw $s1, 0($t0)
2: add $s2, $s2, $s1
3: add $s2, $s3, $s3
```

Solution: No hazards.

Exercises

(c)

```
1: lw $s1, 0($t0)
2: lw $s2, 4($s1)
3: add $s2, $s3, $s3
```

Solution: There is a data dependency between 1 and 2 via $s1, causing a data hazard. A load-and-use

Exercises

(d)

```
1: lw $s1, 0($t0)
2: sub $s3, $s4, $s2
3: bne $t0, $t1, target
4: add $s2, $s5, $s6
```

Solution: There is a branch instruction in 3, so there is a control hazard.

(e)

1: lw $s1, 0($t0)

2: sub $s3, $s4, $s2

3: sub $s3, $s1, $t0

Solution: There is a data dependency between 1 and 3 via $s1, causing a data hazard.

The Pipelined Processor

Pipeline hazards

- exercises

Re-ordering instructions

- exercises

Forwarding and pipeline stalls

- exercises

Exercises

Exercises

Question 1: Data dependency can be resolved by re-ordering the instructions. Do so in the following sequence of instructions in order to resolve the data dependency in $s1 between the instructions 1 and 2.

```
1: sub $s1, $t0, $t1
2: add $s2, $s0, $s1
3: add $s5, $s3, $s4
4: add $s5, $s6, $s6
```

Solution:

```
#note: two pipeline stalls are needed
1: sub $s1, $t0, $t1
2: add $s5, $s3, $s4
3: add $s5, $s6, $s6
4: add $s2, $s0, $s1
```

The Pipelined Processor

Pipeline hazards

- exercises

Re-ordering instructions

- exercises

Forwarding and pipeline stalls

- exercises

Exercises

Exercises

Question 1: Suppose that we execute the following instruction sequence in the pipeline, and it uses the forwarding of the output of the ALU to the inputs of the ALU in the next instruction. Fill in the table below with the appropriate pipeline stages (IF, ID, EXE, MEM, WB) or bubbles (BUB).

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB					
add $s2, $s0, $s1										
add $s5, $s3, $s4										
add $s5, $s6, $s2										

Solution:

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB					
add $s2, $s0, $s1		IF	ID	EXE	MEM	WB				
add $s5, $s3, $s4			IF	ID	EXE	MEM	WB			
add $s5, $s6, $s2				IF	ID	EXE	MEM	WB		

Exercises

Question 2: Suppose that we execute the following instruction sequence in the pipeline. The pipeline uses no forwarding, register writes in the first half of the cycle and register reads in the second half. Fill in the table below with the appropriate pipeline stages (IF, ID, EXE, MEM, WB) or bubbles (BUB).

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB						
lw $s2, 0($s1)											
add $s5, $s3, $s4											
add $s7, $s6, $s5											

Solution:

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB							
lw $s2, 0($s1)		BUB	BUB	IF	ID	EXE	MEM	WB				
add $s5, $s3, $s4					IF	ID	EXE	MEM	WB			
add $s7, $s6, $s5						BUB	BUB	IF	ID	EXE	MEM	WB

The Pipelined Processor

Pipeline hazards

- exercises

Re-ordering instructions

- exercises

Forwarding and pipeline stalls

- exercises

Exercises

Exercises

Question 1: Write down a sequence of 3 MIPS instructions that generates a control hazard and explain its cause(s).

Solution:

```
sw $s0, 4($t0)
beq $s0, $zero, IsZero
add $s1, $s0, $s1
```

Question 2: Write down a sequence of 3 MIPS instructions that generates 2 data hazards and explain their causes.

Solution:

```
lw $s0, 4($t0)
add $s1, $s0, $s1
sw $s1, 0($t1)
```

Exercises

Question 3: Suppose that we execute the following instruction sequence in the pipeline. The pipeline uses no forwarding, register writes in the first half of the cycle, register reads in the second half and a separate memory for data and instructions. Fill in the table below with the appropriate pipeline stages (IF, ID, EXE, MEM, WB) or bubbles (BUB).

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB						
lw $s2, 0($s1)											
add $s5, $s3, $s4											
add $s7, $s6, $s5											

sub $s1, $t0, $t1	IF	ID	EXE	MEM	WB							
lw $s2, 0($s1)		BUB	BUB	IF	ID	EXE	MEM	WB				
add $s5, $s3, $s4					IF	ID	EXE	MEM	WB			
add $s7, $s6, $s5						BUB	BUB	IF	ID	EXE	MEM	WB

Exercises

Question 4: Suppose that we execute the following instruction sequence in the pipeline. The pipeline uses forwarding (output of ALU to input of the ALU of next instruction), register writes in the first half of the cycle, register reads in the second half and a separate memory for data and instructions. Fill in the table below with the appropriate pipeline stages (IF, ID, EXE, MEM, WB) or bubbles (BUB).

lw $s1, 4($s0)	IF	ID	EXE	MEM	WB						
add $s2, $s3, $s4											
add $s5, $s1, $s2											
add $s7, $s5, $s2											

lw $s1, 4($s0)	IF	ID	EXE	MEM	WB						
add $s2, $s3, $s4		IF	ID	EXE	MEM	WB					
add $s5, $s1, $s2			BUB	IF	ID	EXE	MEM	WB			
add $s7, $s5, $s2					IF	ID	EXE	MEM	WB		