The University of Alabama in Huntsville
ECE Department
Homework Assignments
CPE 431/531 01/91/92
Fall 2015
Homework 4 Solution

4.8.1(5), 4.8.3(10), 4.9.2(10), 4.9.4(10), 4.9.5(10), 4.9.6(10), 4.13.1(5), 4.13.4(20), 4.15.1(10), 4.15.2(10)

4.8 In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

Also, assume that instructions executed by the processor are broken down as follows:

alu	beq	lw	sw
45%	20%	20%	15%

4.8.1 What is the clock cycle time in a pipelined and non-pipelined processor?
Non-pipelined: cycle time = IF + ID + EX + MEM + WB = (250 + 350 + 150 + 300 + 200) ps = 1250 ps
Pipelined: cycle time = max (250, 350, 150, 300, 200) ps = 350 ps

4.8.3 If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what isthe new clock cycle time of the processor?
The stage to split is the longest stage, ID which has a latency of 350 ps.
New pipelined: cycle time = max (250, 175, 175, 150, 300, 200) ps = 300 ps

4.9 In this exercise, we examine how data dependences affect execution in the basic 5-stage pipeline described in Section 4.5. Problems in this exercise refer to the following sequence of instructions:

```
(1)     or   r1, r2, r3
(2)     or   r2, r1, r4
(3)     or   r1, r1, r2
```

Also, assuming the following cycle times for each of the options related to forwarding:

Without Forwarding	With Full Forwarding	With ALU-ALU Forwarding Only
250 ps	300 ps	290 ps

4.9.2 Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

Dependence	**Type**	**Hazard?**
(1)-(2)	**1**	**Yes**
(1)-(3)	**2**	**Yes**
(2)-(3)	**1**	**Yes**

```
or   r1, r2, r3
nop
nop
or   r2, r1, r4
nop
nop
or   r1, r1, r2
```

4.9.4 What is the total execution time of this instruction sequence without forwarding and with full forwarding? What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding?

Without Forwarding: Clock Cycles = 4 cycles pipeline fill + 7 instructions = 11 cycles
Full Forwarding: Clock Cycles = 4 cycles pipeline fill + 3 instructions = 7 cycles

$$Speedup = \frac{ET_{WithoutForwarding}}{ET_{FullForwarding}} = \frac{CC_{WtihoutForwarding} * CT_{WithoutForwarding}}{CC_{FullForwarding} * CT_{FullForwarding}}$$

$$= \frac{11 cyles * 250\, ps/cycle}{7 cycles * 300\, ps/cycle} = \frac{2750}{2100} = 1.31$$

4.9.5 Add nop instructions to this code to eliminate hazards if there is ALU-ALU forwarding only (no forwarding from the MEM to the EX stage).

Dependence	**Type**	**Hazard?**
(1)-(2)	**1**	**Yes**
(1)-(3)	**2**	**Yes**
(2)-(3)	**1**	**Yes**

```
or   r1, r2, r3
or   r2, r1, r4
nop
nop
or   r1, r1, r2
```

The dependence between (1) and (2) is handled by the ALU-ALU forwarding. The dependence between (1) and (3) cannot be forwarded from MEM/WB so there has to be a cycle delay between (2) and (3). Once that happens, the dependence from (2) to (3) does not get forwarded either.

4.9.6 What is the total execution time of this instruction sequence with only ALU-ALU forwarding? What is the speedup over a no-forwarding pipeline?

Without Forwarding: Clock Cycles = 4 cycles pipeline fill + 7 instructions = 11 cycles
ALU-ALU Forwarding: Clock Cycles = 4 cycles pipeline fill + 5 instructions = 9 cycles

$$Speedup = \frac{ET_{WithoutForwarding}}{ET_{FullForwarding}} = \frac{CC_{WtihoutForwarding} * CT_{WithoutForwarding}}{CC_{FullForwarding} * CT_{FullForwarding}}$$

$$= \frac{11 cyles * 250\, ps / cycle}{9 cycles * 290\, ps / cycle} = \frac{2750}{2610} = 1.05$$

4.13 This exercise is intended to help you understand the relationship between forwarding, hazard detection, and ISA design. Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a 5-stage pipelined datapath:

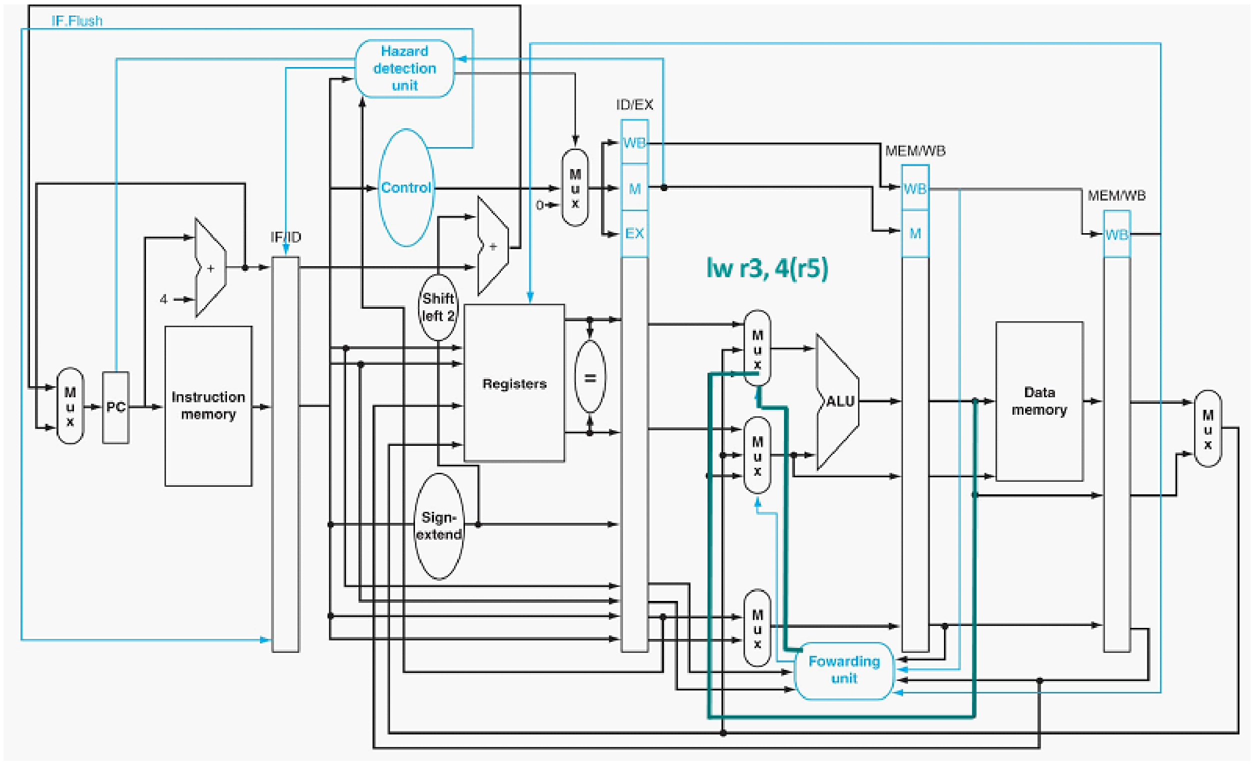
```
(1)  add    r5, r2, r1
(2)  lw     r3, 4(r5)
(3)  lw     r2, 0(r2)
(4)  or     r3, r5, r3
(5)  sw     r3, 0(r5)
```

4.13.1 If there is no forwarding or hazard detection, insert nops to ensure correct execution.
Dependences: (1) – (2), (1) - (4), (1) – (5), (4) – (5)
When there is no forwarding or hazard detection, the WB stage of the producing instruction and the ID stage of the consuming instruction must occur in the same cycle which means that there need to be two instructions in between the producer and consumer. Once the (1) – (2) hazard is resolved, that also takes care of (1) – (4) and (1) – (5).

```
add    r5, r2, r1
nop
nop
lw     r3, 4(r5)
lw     r2, 0(r2)
nop
or     r3, r5, r3
nop
nop
sw     r3, 0(r5)
```

4.13.4 If there is forwarding, for the first five cycles during the execution of this code, specify which signals are asserted in each cycle by hazard detection and forwarding units in Figure 4.60.
The (1) – (2) hazard can be solved by forwarding when the lw gets to the EX stage. That is the only dependence that comes into play in the first 5 cycles. `lw` is in the EX stage in cycle 4.

Cycle	IF	ID	EX	MEM	WB
1	add				
2	lw	add			
3	lw	lw	add		
4	or	lw	lw	add	
5	sw	or	lw	lw	add



There are no hazards so PCWrite is asserted every cycles and no flushing occurs.

4.15 The importance of having a good branch predictor depends on how often conditional branches are executed. Together with branch predictor accuracy, this will determine how much time is spent stalling du to mispredicted branches. In this exercise, assume that the breakdown of dynamic instructions into various instruction categories is as follows:

R-type	BEQ	JMP	LW	SW
40%	25%	5%	25%	5%

Also, assume the following branch predictor accuracies:

Always-Taken	Always-Not-Taken	2-Bit
40%	25%	5%

4.15.1 Stall cycles due to mispredicted branches increase the CPI. What is the extra CPI due to mispredicted branches with the always-taken predictor? Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used.
The penalty of a misprediction is 2 cycles, the instructions in IF and ID need to be flushed when the branch outcome is determined. The stall cycles is the frequency of misprediction times the misprediction penalty.
CPI_{extra} = 0.4*0.25*2 = 0.2

4.15.2 Repeat 4.15.1 for the "always-not-taken" predictor.
CPI_{extra} = 0.25*0.25*2 = 0.125