CprE 381 – Computer Organization and Assembly Level Programming

HW-09

[Note from Joe: This is a continuation of the previous assignment, and focuses on control and data hazards.]

Reading: Patterson & Hennessy, Sections 4.7-4.8

1) Data Hazards and Forwarding

(a) Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a stall?

```
add $3, $4, $2
sub $5, $3, $1
lw  $6, 200($3)
add $7, $3, $6
```

(b) With regard to the program in HW-08 Exercise 1a), explain what the forwarding unit is doing during the fifth cycle of execution. If any comparisons are being made, mention them.

(c) With regard to the program in HW-08 Exercise 1a), explain what the hazard detection unit is doing during the fifth cycle of execution. If any comparisons are being made, mention them.

2) Optimizing Pipelined MIPS Code

(a) How could we modify the following code to make use of a delayed branch slot?

```
Loop: lw   $2, 100($3)
      addi $3, $3, 4
      beq  $3, $4, Loop
```

(b) The example on page 280 shows how to *maximize* performance on our pipelined datapath with forwards and stalls on a use following a load. Rewrite the following code to *minimize* performance on this datapath – that is, reorder the instructions so that this sequence takes the *most* clock cycles to execute while still obtaining the same result.

```
lw  $2, 100($2)
lw  $3, 200($7)
add $4, $2, $3
add $6, $3, $5
```

```
sub $8, $4, $6
lw  $7, 300($8)
beq $7, $8, Loop
```

3) Advanced Data Hazards

(a) Consider an instruction sequence used for a memory-to memory copy:

```
lw $2, 100($5)
sw $2, 200($6)
```

The elaboration starting on page 312 discusses this situation and states that additional forwarding hardware can improve its performance. Show the necessary additions to the datapath of Figure 4.57 to allow code like this to run without stalling. Include forwarding equations (such as the ones appearing on pages 306–311) for all of the control signals for any new or modified multiplexors in your datapath. Finally, rewrite the stall formula on page 314 so that this code sequence won't stall.

(b) Consider the following instructions as they are being executed on a five stage pipelined datapath:

```
lw $1, 40($6)
add $2, $3, $1
add $1, $6, $4
sw $2, 20($4)
and $1, $1,$4
```

i. If there is no forwarding or hazard detection, what NOPs would need to be inserted to ensure correct execution?

ii. If the processor has forwarding, but no hazard detection unit, what would happen when this code executes?