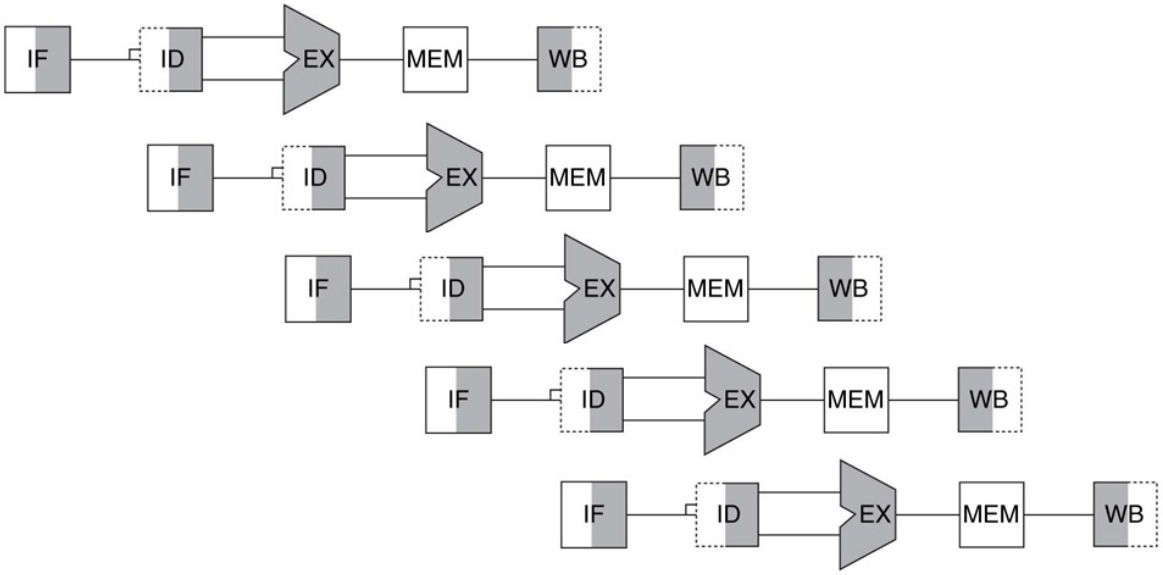
CS320 Fall 2014

Homework#6
Quiz in Lecture Thurs Nov. 13, 2014
There are no make-up quizzes!

Problem 1: Using a below figure, show the forwarding paths and stalls needed to execute the following instructions.

```
add $2, $3, $4
sw  $2, 200($3)
sub $3, $2, $8
add $8, $3, $2
add $10, $3, $8
lw  $7, 0 ($10)
or  $7, $7, $10
lw  $8, 8 ($7)
add $8, $3, $8
```



Problem 2: Consider the following code on the pipelined datapath:

```
add $1, $2, $3
lw  $4, 4 ($5)
sub $6, $7, $8
sub $9, $10, $11
add $12, $13, $14
add $15, $16, $17
add $18, $19, $20
```

(a) At the end of the seventh cycle of execution, which registers are being read and which will be written?
(b) What is the forwarding unit doing during the seventh cycle of execution. Are any comparisons being made? state them.

(c) What is the hazard detection unit doing during the seventh clock cycle of execution? Are any comparisons being made? state them.

Problem 3: Consider the execution of the following code on the pipelined datapath:

```
sub $4, $0, $5
sw  $4, 0 ($6)
add $6, $4, $5
lw  $4, 0 ($6)
```

(a) Illustrate, using the figure from Problem 1, the dependencies that need to be resolved (ie. Identify the hazards, what type are they?).
(b) Illustrate, using the figure from Problem 1, how the code is executed (with stalls and forwarding) resolving the problems.
(c) How many cycles does it take to execute this code?

Problem 4: Assume a pipeline with 4 stages: Fetch (FI), Decode & Calculate Address(DA), Fetch Operand (FO), and Execute (EX). Assume that the Execute stage also includes the MEM and WB when required.
(a) Name any potential structural hazards. Why?
(b) What are the data hazards which could occur? Why?
(c) If needed, what/where are the forwarding paths required?
(d) Do you ever need to stall? Under what cases?
(e) Draw a diagram, similar to the one used in Problem 1, for a sequence of 7 instructions. Assume the third instruction is a branch instruction. Assume no early branch detection is performed, a "branch not taken" strategy is used, and the branch is taken.
Problem 5: For the each of the following sequence of instructions, draw a diagram like Problem 1.
(a) Label the bubbles caused by stalls. (Note you should not need more than 16 clock cycles)

Code 1:

```
lw $7, 0($1)
sub $7, $3, $7
add $3, $7, $3
sw $7, 0($3)
```

Code 2:

```
lw $2, 0($3)   # Assume that this is a valid instruction in your datapath
add $3, $2, $4
sub $1, $3, $4
add $1, $5, $3
sw $5, 8($6)
```

(b) If possible, eliminate all of the bubbles caused by the instruction sequence in part (a) by reordering the instruction sequence and/or by adding bypass paths to the basic, standard pipeline diagram. Give the reordered instruction sequence.

(c) Add bypass hardware (data forwarding paths(s) into the hardware datapath) and reorder the code sequence to eliminate all the stalls for Code 2.

(d) Assume that the register writes occur in the SECOND half of the clock cycle and reads occur in the first half. Add a bypass (data forwarding path into the hardware datapath) to eliminate all stalls from the Code 2 sequence in Part (a).