

## Problem Set #7

**Date Assigned:** Monday, May 2, 2016

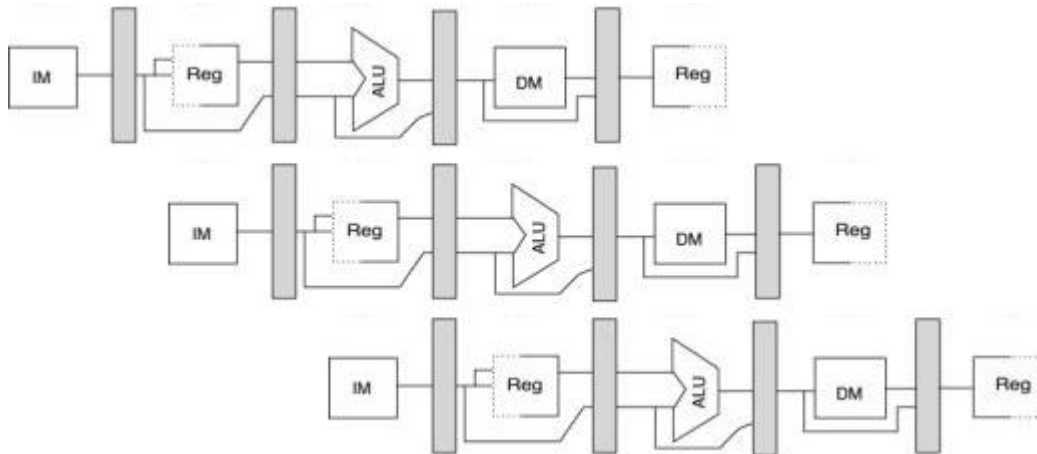
**Date Due:** Monday, May 9, 2016 @ 9:15am

**Points:** 50 pts

1. (20 pts) Consider the following MIPS program:

```
dadd r1,r2,r3  
ld r4,0(r1)  
sd r4,12(r1)  
halt
```

- Identify all of the data dependencies in the above program drawing arrows as we've done in class.
- Given the 5 stage MIPS pipeline, show the instructions as they move through the pipeline using an S for stall when necessary. Assume no forwarding.
- Rework b) assuming the maximum amount of forwarding possible.
- Complete the following diagram showing where each forwarded piece of data comes from.



2. (8 pts) Consider the following MIPS program:

```
ld r1,0(r2)  
dsub r4,r1,r5  
and r6,r1,r7  
or r8,r1,r9  
halt
```

I make the claim that using the maximum amount of forwarding, no stalls are necessary. Either accept or reject my claim. Which ever stance you take, justify your answer as you did in 1) d) above.

3. (22 pts) You are to write an assembly language program for the winMIPS64 machine that sorts a list of N signed numbers in increasing order where N is specified as the first word in a list of numbers followed by the numbers to be sorted. Your program is to be completely documented with the file documentation consisting of:

```
*****  
; Filename:  
; Author:  
; Date:  
; Class:  
; Assignment:  
; Purpose:  
; Algorithm:  
*****
```

Further, each line of assembly is to be documented. By documented, I'm not talking about just mimicking what the statement does, but provide meaningful documentation. As for the numbers to be sorted, the list will look like the following, which can be easily expanded.

```
.data
```

```
values:      .word 10,5,20,100,-5,30,45,-15,-10,25,20
```

Note: The first word specifies the number of words to follow.

On the day this assignment is due, each of you will demo your solutions. I will provide several sets of values for each of you to use in your program. We will see how many cycles each solution takes.

I want you to run your solution in the default WinMIPS64 mode (i.e. don't change any of the settings). Also, do not do any extra calculations in the data segment. The data segment is for declaring variables only.

### How to turn in your solution:

- Create a folder called 07punet, and place inside it: (1) your word document 07punet.docx and (2) your MIPS assembly program (sortPUNet.cpp). Submit the folder by placing it into the CS 430 drop folder on Grace by the deadline.
- Please make sure your problem sets are typed, answered in order, and stapled together. A hard copy of your Problem Set Solution is due on the instructor's desk by the deadline.