CSE 410 Assignment 4

Spring 2008

Due: Midnight, Wednesday 4/30/2008

Submit all of your files via Catalyst Collect it. Please do not forget to comment your codes and put you name and UWNetID in every files you submitted.

https://catalysttools.washington.edu/collectit/dropbox/telmas/2218

HOMEWORK ASSIGNMENT

P&H 3rd edition, read Chapter 6 through the end of 6.6.
In P&H 2nd edition, read Chapter 6 through 6.7.

1. Convert the following decimal numbers to 2's complement, 16-bit binary numbers. Show your answers in both binary and hexadecimal
   a. -231
   b. -4,420
   c. 789
   d. -2^15

2. Convert the following 16-bit 2's complement hexadecimal numbers to decimal.
   a. 0xBEEF
   b. 0x4001
   c. 0xFEAA
   d. 0x4545

3. What is the difference between latency and throughput? What is the effect of pipelining on these two metrics?

4. Describe the different kinds of dependencies and their causes.

5. Identify all of the data dependencies in the following code. Which dependencies can be resolved via forwarding, and which will cause a stall?

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

6. What type of hazard exist in the following code:

```
             add $3, $3, $4
             bne $3, $12, jump_label
             add $5, $5, $6
jump_label:  sub $5, $5, $6
```

PROGRAMMING ASSIGNMENT

Implement a MIPS function `quad(a,b,c,x)` that computes the integer value ax2 + bx + c for integer arguments a, b, c, and x.

You should use the standard MIPS function calling- and register usage-conventions discussed in class and described in the book.

Then write a main program that calls your quad function several times and prints out the result of each function call on a separate line. Optionally, you can add labels and identifying text to the output, but this is not required. Use the `syscall` instruction to produce a human readable output. Appendix A of the book has more information about the .

a) Your main program should call the `quad` function three times with the following parameter values, and print out the value returned.

a=3, b=5, c=2, x=5
a=1, b=-7, c=23, x=-6
a=7, b=9, c=-1, x=15

b) In addition, you main program should also call the min/max function with the same parameter values as above and print out the result.

Your main program should look like the following:

```
int main() {

   quad(3,5,2,5);            // prints  "Results is: …" after quad returns
   ..
   ..
   ..
   min_max(3,5,2,5);         // prints  "Minimum is: …"
                                         Maximum is: …" after min_max returns
   ..
   ..
   return 0;
}
```

Extra credit 1: The value ax2 + bx + c can be calculated using 3 multiplications and 2 additions. Can you reduce this and save an instruction or two in your code?

Extra credit 2: The calling convention you saw in the class up to 4 arguments can be passed through registers. Rewrite your program such that the functions quan and min_max receive their arguments only through the stack, similar to x86 calling convention. Rewrite main so that it passes arguments to these functions through the stack.