Lab 4 1 Due Tues., 30 Sept., 2:30pm

CMPSC 210
Principles of Computer Organization
Fall 2014
John Wenskovitch
http://cs.allegheny.edu/~jwenskovitch/teaching/CMPSC210

Lab 4
23 September 2013
Due (via Sakai and hard copy) Tuesday, 30 September, 2:30pm
30 points

Goals

Use branching instructions in MIPS to write a loop. Use these loops in two different programs.

General Instructions

Paste your programs and a screenshot of your program outputs in MARS into a document (PDF or open office or plain text). Upload this document to Sakai and hand in a hard copy by the due date. BE SURE YOU PUT YOUR NAME AND THE LAB NUMBER at the top of the document!

Also upload the programs themselves as a separate file. The program must be FULLY COMMENTED — header comments should provide information about author, date, program purpose (input/output and summary of processing performed); in-code comments should describe variables and summarize major steps in the algorithm. For MIPS programs, be as generous with comments as possible—they are mostly to help YOU, but they’ll help me, too.

Use white space, e.g., blank lines, to make the program more readable.

Add Two Arrays [15-points]

Write a MIPS assembly language program that adds two arrays (element-by-element addition) and prints their sum. For instance, if the two arrays are:

\[
\begin{array}{c}
a = \begin{pmatrix}
2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \\
+ & + & + & + & + & + & + & + & + & +
\end{pmatrix}
\end{array}
\]

\[
\begin{array}{c}
b = \begin{pmatrix}
-1 & 0 & 1 & -2 & 0 & 2 & -3 & 0 & 3 & -4 \\
+ & + & + & + & + & + & + & + & + & +
\end{pmatrix}
\end{array}
\]

then their sum is:
All three arrays should be of size at least 10. You may initialize \( a \) and \( b \) using the \texttt{.word} directive in your program’s \texttt{.data} section.

Print out the element of the sum in a single line, labeled and with values separated by spaces. The line should be terminated by a “\n” character. For instance, if the arrays \( a \) and \( b \) are as above, the output will look like:

\[ \text{The sum is: 1 4 7 6 10 14 11 16 21 16} \]

Run it. Take a screenshot of the output and paste it, along with the text of your program, into the report document. Also upload the MIPS program to your lab3 folder on Sakai.

\textbf{Add Two Binary Numbers [15 points]}

Write a MIPS program that adds two 16-bit binary numbers, performing the addition in decimal and writing the answer back into memory in binary.

You will store two binary numbers in string form in the \texttt{.data} section of your code, plus allocate space for the answer:

\begin{verbatim}
A: .asciiz "0111000011000100"
B: .asciiz "1100010001110000"
C: .space 17
\end{verbatim}

\( A \) and \( B \) are two strings, each of which contains an unsigned 16-bit binary number in a 16-character long string form. You have to fill up \( C \) with the binary representation of the results of summing up \( A \) and \( B \).

To earn full credit on this lab, your implementation must not convert the string representation of the numbers into a register content, add the registers, and convert the result back into binary form. You must do the addition bit-by-bit using essentially a 1-bit adder.

Hint: Hexadecimal values for ‘0’ and ‘1’ are 0x30 and 0x31 respectively. If you subtract 0x30 from the character representation, you will get the decimal value. Similarly, if you add 0.30 to decimal 0 or 1, you will get the character representation of them.