UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE12/L: COMPUTER SYSTEMS AND ASSEMBLY LANGUAGE

Lab 6: Decimal to Binary Converter Redux
Worth 70 points (60 lab + 10 report)

Lab Objectives:

With some time spent in MIPS and specifically to do with hardware, we are now going to code something more complex, but very familiar. We are going to make another decimal to binary converter. This one will not accept input from the keyboard but will instead read a string from memory. Serial communications are always tricky to get right in embedded applications and this will make the program much easier to work with.

Part 1:

Refer back to your flowchart for Lab 3 and think about how you would do it with the extra operations now available to you. Revise your flowchart to leverage these new features being sure to use the shift operators instead of loading masks.
The loops should be easier and being able to directly compare registers will make life easier.

Part 2:

We have given a starter file that handles the setup for you. If you run this file without any added code it will print out the incoming number and the string where you will be storing the binary string. You should only need to add code between the two comments and should not need to add any output but feel free to customize the welcome message if you desire.
Implement your new flowchart ensuring that it works with a wide range of numbers.

- Remember that the Uno32 is 32-bit, not 16.
- The input string is of unknown length but is properly terminated.
- The output string is 32 bits long.
- *inNumericString* holds the input string and *outBinaryString* holds the binary string.
    - The binary string is initialized to all ‘z’’s.
- Some Useful Assembly Commands:
    - MUL: This will multiply two registers together. As the end number needs to fit in 32 bits we don’t need to use the full MULT.
    - SB/LB: Allows loading and storing of bytes instead of words
    - SLR: Shifts all the bits of a register to the right. This allows us to instead of making static masks to shift the bit over instead to generate our new mask.
    - The different branch commands to make this process simpler.
    - SLTU is also very useful, read the documentation on what it does.

Lab Requirements

You must do the following for this lab:

- Convert decimal strings to binary leveraging the advanced features of MIPS.
- Demonstrate this ability for any number given by the Teaching staff.

Files to Submit on Canvas

- Lab6.asm
- Lab6_report.txt
- Flowchart to TA/Tutor when demonstrating

Check-off

You should demonstrate your lab to the TA/Tutor when finished to get it checked off and show them your flowchart.

Point Breakdown

- 20 pts: Converts character decimal string into a 2SC number
- 20 pts: Store as a binary string
- 10 pts: Work with negative input
- 5 pts: Good flow chart
- 5 pts: Have good comments in the code
- 10 pts: Write up