CS 341 Lab 5 - Understanding SPIM Exception Handler & Single vs. Multi-cycle

General Instructions:

1. Create a directory<roll_number>_<lab_no>. Store all the relevant files in this directory.
2. While submitting (on moodle), you have to create a tar.gz or zip of the entire<rollno>_<labno> directory in which all your relevant files reside.
3. The code you write should be well commented and easily readable. Violation of this could result in a penalty of half the points.

1. **Understanding the SPIM exception handler code [5 (2+1+2) marks]**

   The SPIM exception handler “exceptions.s” is well commented. Read and understand the code. You may have to refer to the MIPS instruction set reference for instructions you do not understand.
   Location of Exception Handler - “/usr/lib/spim/exceptions.s”

a. Explain the use of following registers/instructions in a file(readme.txt).
    1. mfc0/mtc0
    2. .set noat/ .set at
    3. Cause register
    4. EPC register

b. Make a copy of “exception.s” in your directory Somewhere and call it “exceptions-<yourname>.s”. Learn to run spim or qtspim with your own copy of the exception handler. Show this to the TA. *Hint: you may have to refer to the SPIM command line options.*

c. Give a brief description about any four exceptions along with their cause code in a file called explanation.txt. Also submit code that generates the exceptions in the same text file.

Example -

1. Arithmetic overflow code - (12) #cause code in bracket.

```
.text
main:
lui $s0, 32767
ori $s1, $s0, 65535
# Add 1 to $s1 and store in $s2. This should produce an overflow
exception
addi $s2, $s1, 1
```

2. Printing the data-address of a non-aligned store (5 (2 + 3) Marks)

A. Now add the exception handler code to “**exceptions_misalign.s file**”, to print the mis-aligned data address, corresponding to a store word instruction which causes a mis-aligned store exception.

Hint : you may have to refer to the MIPS data sheets and search in the appropriate place, this is part of the exercise.

B. Write code in a file called “misalign.s”, such that it causes a store-word mis-aligned exception. Run this code with your modified exception handler, and show that the mis-aligned data address is printed.

3. Spill over from Lab 4 - Single Vs Multi Cycle MIPS (5 (3 + 2) Marks)

We now want to use to compare the performance of a single cycle implementation with a multi-cycle implementation of the MIPS ISA. Assume that a multi-cycle implementation uses a clock which is 5 times faster than a single cycle implementation. Also assume that in a multi-cycle implementation, the number of cycles taken for **lw** is 5, for **sw** is 4, for all types of branch instructions is 3, and for all other instructions is 4.

A. Write a perl script (or in any other language), which takes the sequence of executed instructions output by the modified SPIM in an input file, and computes the number of cycles taken by the MIPS program. The script should compute both (a) the number of cycles taken in a single cycle implementation, as well as

(b) the number of cycles taken in a multi-cycle implementation. The script should print these cycle counts, and also how much faster a multi-cycle implementation is compared to a single cycle implementation.

B. Use the “gcd-noopt.s” (previous lab file) to test the working of the script. And explain the observation in a file <observation.txt>. Also give a demo to TA for this question.