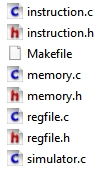
Computer Architecture Project 1 Report

104062261 葉毓浩

1. Project Description
   1. Program Flow chart:

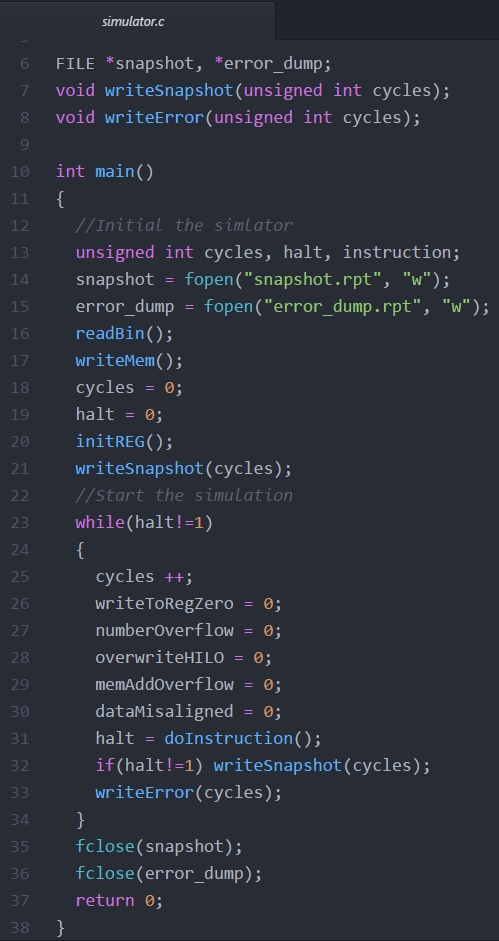


* 1. Detailed Description:



The file I created in my project is modularize almost same as the suggestion.

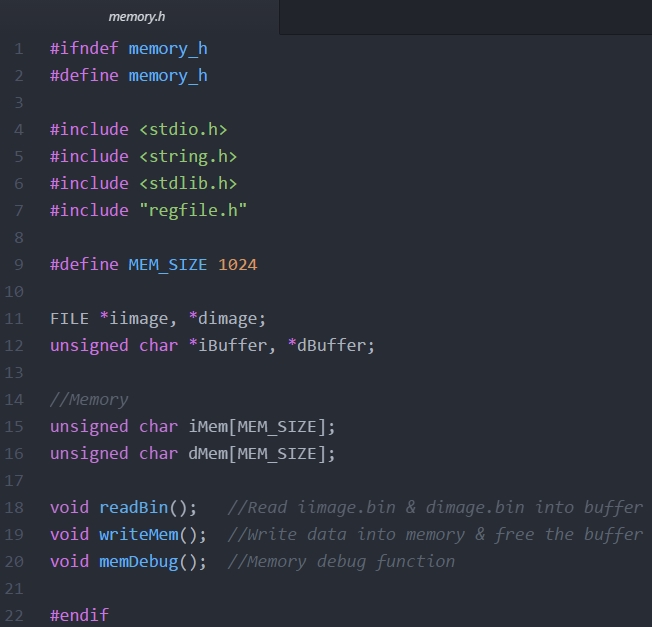
1. Simulator part:



The main function and two file write function “snapshot.rpt” and “error\_dump.rpt” was put in here. The main function is the big picture of the single-cycle processor. The main process is written in the main function.

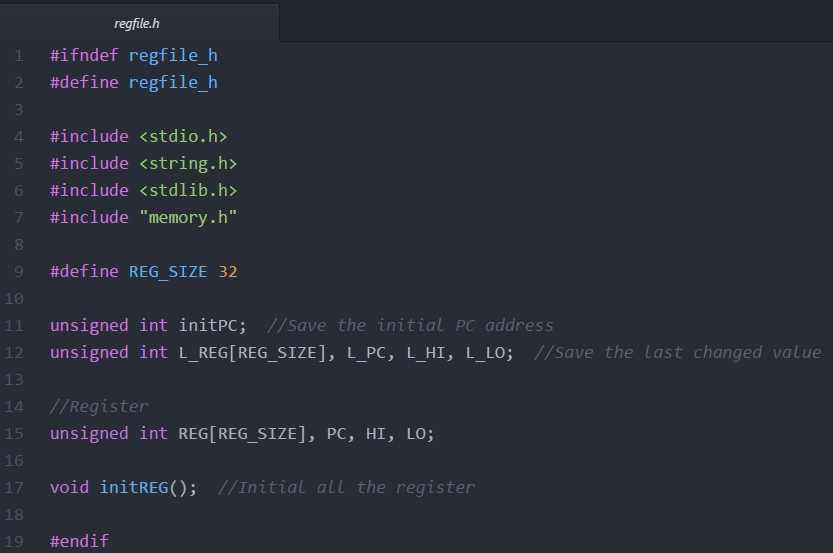
1. Memory part:

The processor memory is written in here. And the read binary file and write memory function is also written in here. Using buffer to save the data read form the binary file, first initial the “PC” and “Stack Pointer” address, then according to the number of words that mentioned in the binary file and write the words into the instruction memory and data memory.



1. Register file part:

The register value, the register last changed value and the initial PC address is save in this part, an initial register function is written in here too. That initial function is copy the register value to the last changed register before the simulator is going to work.

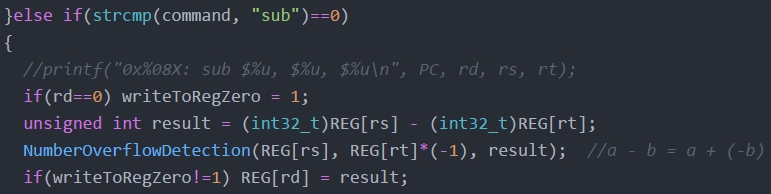


1. Instruction part:

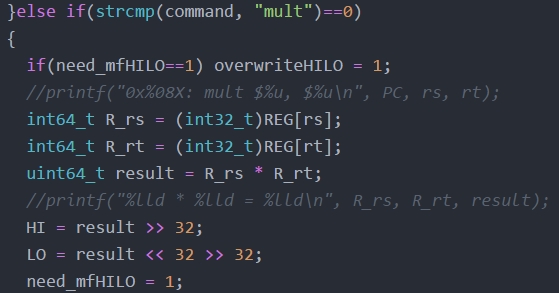


It is the heart of my simulator, many important function is written in here. The “doInstruction” function is to get the instruction form the instruction memory and call the “decode” function to handle what to do. Then depending on the instruction and call “type” function to do the next step. In the “type” function, it will do the right process and detect is there any error command. The error will be recorded and print it out in the “error\_dump.rpt”.

1. Some detail in the decoding part:



The number overflow detection in subtract is special. Turn the formula “A - B” into “A + (-B)” so that we can use the number overflow detection with the addition function together.

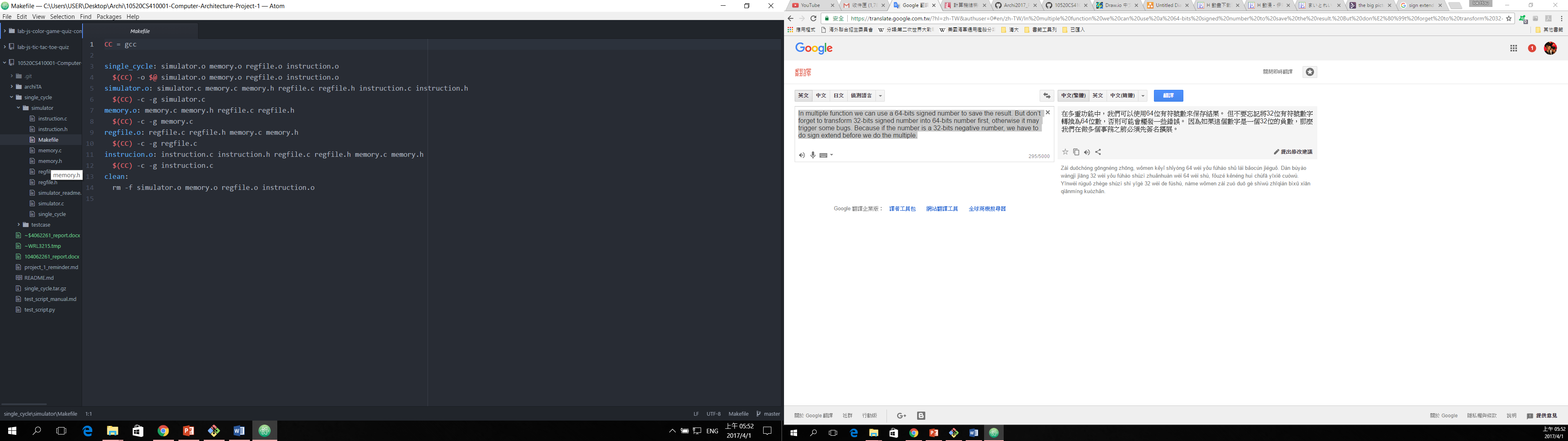


In multiple function we can use a 64-bits signed number to save the result. But don’t forget to transform 32-bits signed number into 64-bits number first, otherwise it may trigger some bugs. Because if the number is a 32-bits negative number, we have to do sign extension before we do the multiple.



Immediate is a signed number. After saving it as an unsigned number, we should remember every time we use it as a signed number we should transform it back to a signed number.

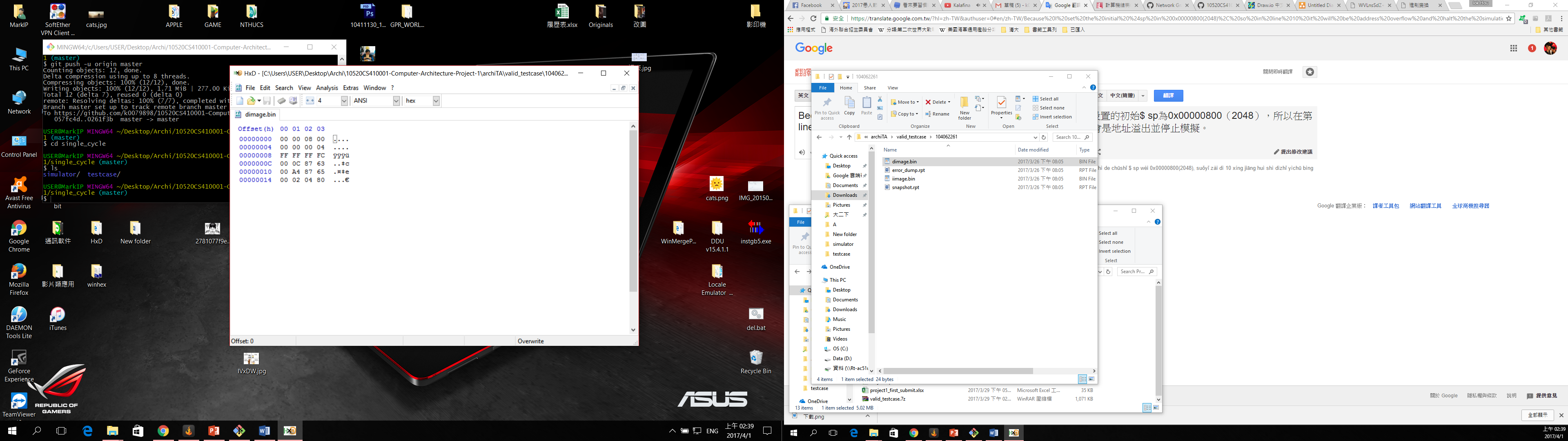
1. Makefile part:

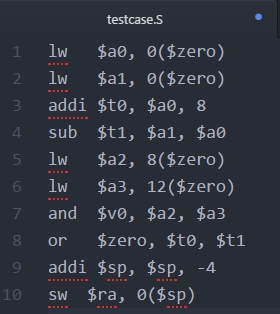


Makefile can help us easier to compile the new program after we do some changes. And we can write a clean command to help us clean up the “\*.o” file after we compile the new program.

1. Test case Design

2-1) Detail Description of Test case:





A simple test case to test some instruction processing, write $0 error and address overflow. Because I set the initial $sp in 0x00000800(2048), so in line 10 it will be address overflow and halt the simulation.