**Overview of the multi-cycle simulator**

Authors: Haoyi Shi & Haoyuan Du

**1. How does it work?**

Firstly, the simulator will read in a .mc file given from the args in console. It should be a file with machine code in it. It will print the result to the console.

Next, each line of decimal number will be saved into a large array as binary numbers. By picking the opcode segment (24:22) which are 3 bits and comparing it with the seven preserved opcodes. This simulator performance each type of instructions individually.

Then, the simulator takes other segments (regA, regB, offset\_field, etc.) depended on different types of instructions and performs the correct actions. The simulator will print the changes of current PC, Registers and Memories for each instruction until the program executes a halt. Once the each instruction has done, the instruction counter will increase by 1. Also, the cycle counter will increase by a certain number as below:

|  |  |
| --- | --- |
| Instructions | Cycles |
| add | 2 |
| nand | 2 |
| sw | 8 |
| lw | 8 |
| beq | 4 |
| jalr | 1 |
| noop | 1 |
| halt | 1 |

On the other hand, the single-cycle simulator, each instruction will take 9 cycles to finish.

The simulator will exit if any errors are catched. For instance: 1. Unused bits are not all zeros. 2. Unrecognized opcodes. 3. Store words is going to store some value in memory above the ‘halt’ instruction. Etc.

**2. Any difficulties**

1. Since most difficulties have been solved in single-cycle simulator. This multi-cycle simulator is much easier. Because all we need is to creat a new ‘cycle counter’ and keep adding it when a certain instruction has done. The hardest part is after writing a valid, complicated test case, we need to figure out the correct total instruction times and cycle numbers by ourselves first. The test case included ‘heavy’ jalr instruction and the beq one takes more efforts.

2. Writing a comprehensive and no-api-like overview is hard. Because the professor said my previous overview wasn’t thorough. So this time I add some instances and make a table. Hope he will satisfied.