<TITLE>

BY:

Joseph Gabriel Gonzales

Justine Paul Guiao

Jovielene Lanorio

Bernieson Sahagun

CMSC 132 ST-1L

COMPUTER ARCHITECTURE

1. **Introduction and Statement of the Problem**
   1. **SYNTAX**

* ADD *param1* *param2*

Adds the two parameters *(param1* and *param2)* and saves the value in *param1.* If the value of loaded in *param1* after the computation is greater than 2 digits, overflow flag is set to 1. If not, overflow flag is set to 0.

* LOAD *param1* *param2*

Loads and sets a value for *param1* using *param2*. If the value loaded is greater than two digits, overflow flag is set to 1. If not, overflow flag is set to 0. Floating point values are not allowed.

* SUB *param1 param2*

Subtracts the two parameters *(param1* and *param2)* and saves the value in *param1.* If the value of loaded in *param1* after the computation is greater than 2 digits, overflow flag is set to 1. If not, overflow flag is set to 0.

* CMP *param1 param2*

Compares *param1* and *param2* by subracting *param2* from *param1*. Zero flag and negative flag are set to 0 by default. If the result is 0, zero flag is set to 1. If the result is less than 0, the negative flag is set to 1.

* 1. **ADDITIONAL SYNTAX ASSUMPTIONS**

<List and describe additional assumptions on instructions and parameters>

1. **Pipelined Design**

<Discuss the instruction pipelining implemented : in-order-execution or out-of-order-execution>

<Use figures and illustrations if needed>

<Discuss ADDITIONAL INSTRUCTION PIPELINING ASSUMPTIONS>

1. **Program Design**
   1. **Program Flow**

<Modular or not? Use flowchart if needed>

* 1. **Program Output**

<Sample output of your program. Use screenshot figures if needed>

<Describe each item on the screenshot.>

1. **User Manual**

* User must download the whole file system using git clone https://github.com/TheGabCode/AMRS.git
* User can change the values and instructions in “instructions.txt”
* Open terminal, compile and execute by entering the following commands
  + javac \*.java
  + java CMSC132 instructions.txt

1. **Conclusion**
2. **References**

* Teach-Sim Educational Simulators
  + The CPU Simulator, OS Simulator, Compiler
  + <http://www.teach-sim.com/>
* HASE – Computer Architecture Simulation Environment
  + http://www.icsa.inf.ed.ac.uk/research/groups/hase/