**CISM 314 LAB 2: MARIE ARCHITECTURE**

**PYTHON**

**GROUP NAMES: T MPOFU**

**K KOK 26797445**

**NP MATLAPENG 27441393**

**AIM:** Write a program in Assembly using the MARIE Simulator to find the sum of the following equation and store and display the output.

**W = X + Y - Z**

The program should find the sum and store it in W. Use the following values as variables.

**Introduction**

MARIE ('Machine Architecture that is Really Intuitive and Easy') is a machine architecture and assembly language served only for educational purposes from The Essentials of Computer Organization and Architecture

The MARIE architecture has the following characteristics:

* + Binary, two's complement data representation.
  + Stored program, fixed word length data and instructions.
  + 4K words of word-addressable main memory.
  + 16-bit data words.
  + 16-bit instructions, 4 for the opcode and 12 for the address.
  + A 16-bit arithmetic logic unit (ALU).

Seven registers for control and data movement

MARIE’s seven registers are:

* + Accumulator, AC, a 16-bit register that holds a conditional operator (e.g., "less than") or one operand of a two-operand instruction.
  + Memory address register, MAR, a 12-bit register that holds the memory address of an instruction or the operand of an instruction.
  + Memory buffer register, MBR, a 16-bit register that holds the data after its retrieval from, or before its placement in memory.
  + Program counter, PC, a 12-bit register that holds the address of the next program instruction to be executed.
  + Instruction register, IR, which holds an instruction immediately preceding its execution.
  + Input register, InREG, an 8-bit register that holds data read from an input device.
  + Output register, OutREG, an 8-bit register, that holds data that is ready for the output device

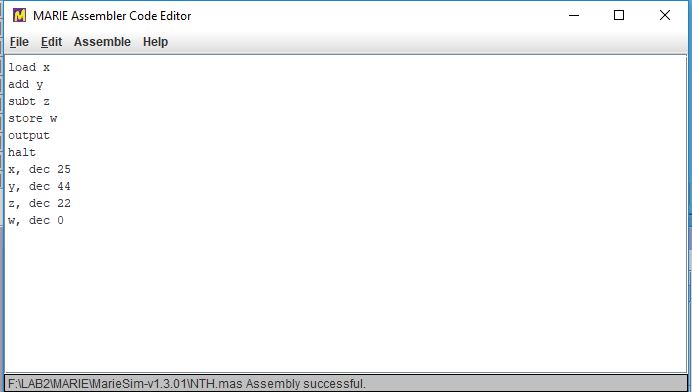


Fig 1: The Code for the program that add Y and subtract Z to get W

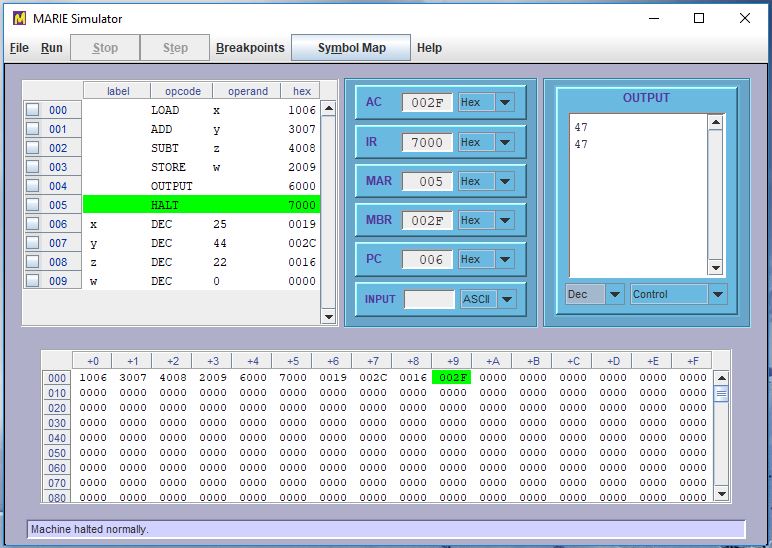


Fig 2: The Marie simulator after running the program

First you save the program, the extension should be .mas and then assemble it, and then there will be a file with an extension .mex that is the file that is loaded to the Marie simulator

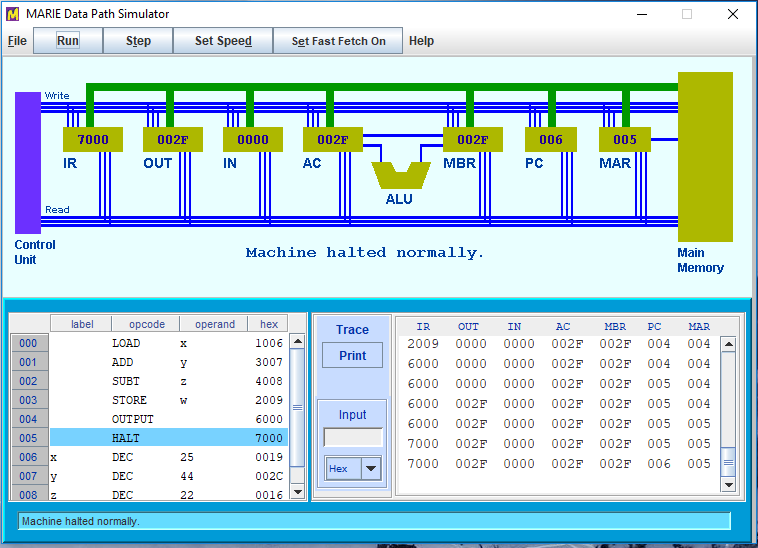


Fig 3: The Marie DPS (Data Path Simulator)

The program will now be loaded in the DPS after it has be executed, the DPS

1. Major components are connected via a 16-bit-wide bus.
2. During each clock cycle, the bus can transfer one 16-bit quantity from one specified device to another.
3. Some additional connections also exist. These can make additional transfers during the same cycle as a value is transferred on the bus.
4. Each instruction is implemented by a series of computations and transfers.