# INTRODUCTION TO MARIE

Group Name : T K O

Group Members: Ramadie O – 27543404

Johnson TK – 27353362

Tlape K – 27693988

## 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 (Linda Null, Julia Lobur). In addition, the publisher provides a set of simulator programs for the machine, written in Java.

MARIE.js is a JavaScript version implementation of MARIE. It aims to be as faithful to the original Java programs as it can, while improving on features to make concepts more intuitive and easier to understand.

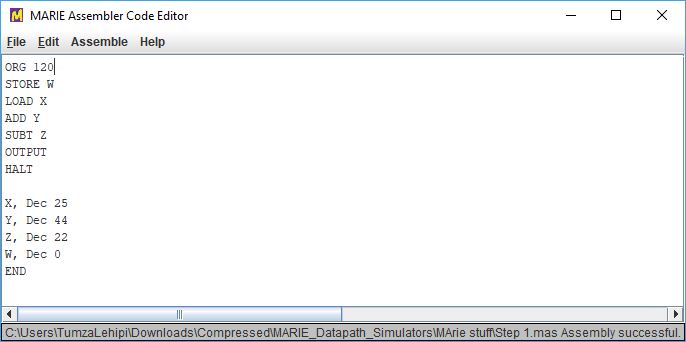
Each personal computer has a microprocessor that manages the computer's arithmetical, logical, and control activities. The aim also includes to design

Each family of processors has its own set of instructions for handling various operations such as getting input from keyboard, displaying information on screen and performing various other jobs. These set of instructions are called 'machine language instructions'.

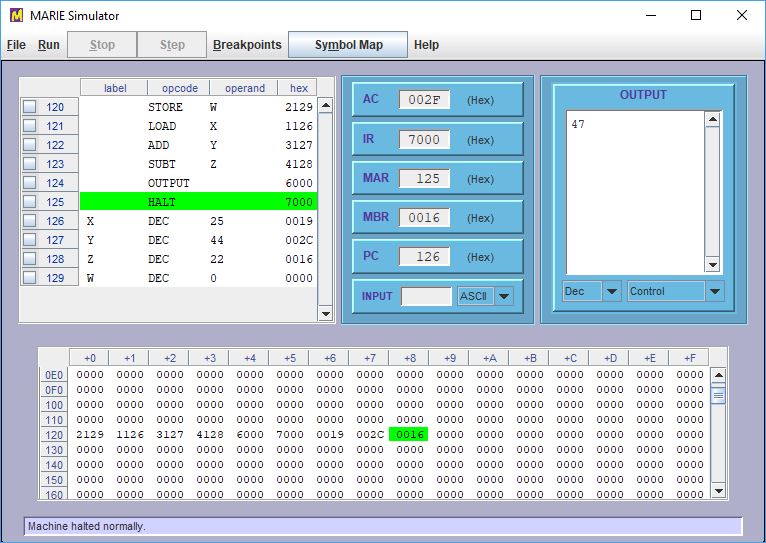
A processor understands only machine language instructions, which are strings of 1's and 0's. However, machine language is too obscure and complex for using in software development. So, the low-level assembly language is designed for a specific family of processors that represents various instructions in symbolic code and a more understandable form

The aim of this practical is to write a program in Assembly using the MARIE Simulator to find the sum of the following equation, **W = X + Y – Z,** store and display the output. W,X,Y and Z are given as 0,25,44 and 22.

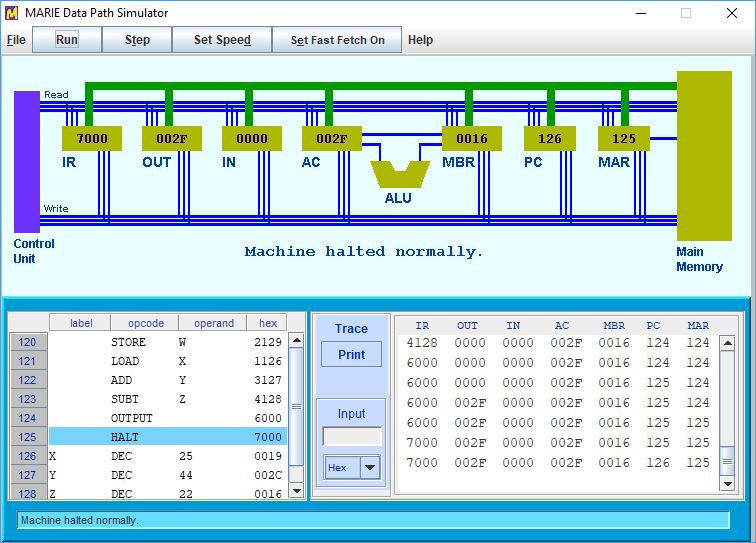
## Discussion



The **assembler code editor** was used to write pre-defined assembly code, the code was then assembled. As displayed above, memory locations X,Y,Z and Z were used to store the following decimal numbers 25,44,22 and 0 respectively. The assembly code above translates to; start at memory location 120 and store the variable W, load the decimal stored at memory location X, add the decimal number at memory location Y, subtract the decimal number stored at memory location Z and store the output.



The **MARIE simulator** was opened and the file containing the assembled code was loaded onto the program, then the program was ran which produced the expected output.



Marie Data Path Simulator was opened and the file containing the assembly code was loaded onto the simulator and was ran.

## Conclusion

The aim of this practical was to write a program in Assembly using the MARIE Simulator to find the sum of the following equation, **W = X + Y – Z,** store and display the output. W,X,Y and Z are given as 0,25,44 and 22.

The simulation was successful and gave an output value of 47.