## Topic for today:

Assembly language

### Machine Language

Machine language for a particular ISA is the programming language that consists of the instructions written in binary.

It is the only programming language that the CPU can execute directly.

#### Notice

- Programming directly in machine language seems difficult (remembering the codes, getting the addresses correct in binary, etc.)
- Reading and debugging a program in machine language is especially painful

### Assembly Language

Assembly language is a human-readable representation of machine language. The basic principle is that one line of assembly language corresponds to one line of machine language. Both operations and addresses are expressed symbolically rather than in binary.

# Fields in an assembly language instruction

An instruction in a typical assembly language (including the Marie assembly language) has four fields: label, operation, operand (or address), and comment.

### Assembler

The software that converts assembly language to machine language is called an <u>assembler</u>.

An assembler takes a <u>source file</u> and creates an <u>object file</u>.

### Assembler Directive

An assembler *directive* is a command to the assembler in an assembly language program. It is not translated into machine code.

#### Two-pass assembler

A two-pass assembler (which is the typical design for an assembler) translates assembly language to machine language by going through a program twice:

- 1. The first pass assigns line numbers and creates a symbol table to record the line numbers referenced by labels.
- 2. The second pass "fills in the blanks" by replacing labels with actual line numbers.



Assemblers – 2 <sup>nd</sup> Pass			
<ul> <li>After the second pass, the assembly is complete.</li> </ul>		Address	Instruction
		Address	monuction
1104		100	Load X
1104		101	Add Y
3105		102	Store Z
2106	X 104	103	Halt
	Y 105	104 X,	DEC 35
7000	1 105	105 Y,	DEC -23
0023	Z 106	106 Z,	HEX 0000
FFE 9			
0000			
			10

# Programming in assembly language vs. a high-level language

Programming in assembly language is a slower process than programming in a highlevel language ( $C^{++}$ , Java, etc.). However, the code it produces is typically more efficient (i.e., runs faster and/or takes up less memory).

So, there is a tradeoff between programmer time and code efficiency.