

# Chapter 7 Assembly Language

## 7.1 Assembly Language Programming - Moving Up a Level

The **purpose of assembly language** is to make the programming process more user-friendly than programming in machine language, while still providing the programmer with detailed control over the instructions that the computer can execute.

Before a program written in a high-level language can be executed, it **must be translated into a program in the ISA** of the computer on which it is expected to execute. It is often the case that each statement in the high-level language specifies **several instructions in the ISA** of the computer.

## 7.2 An Assembly Language Problem

### 7.2.1 Instructions

```
LABEL    Opcode    Operands    ; Comment
```

#### 7.2.1.1 Opcodes & Operands

These two parts are mandatory(强制的).

The **Opcode** is a symbolic name for opcode of the corresponding LC-3 instruction.

The number of **operands** depends on the operation being performed.

#### 7.2.1.2 Labels

**Labels** are symbolic names that are used to identify memory locations that are referred to explicitly in the program.

**2 reasons for using Labels:\***

- The location is the target of a branch instruction.
- The location contains a value that is loaded or stored.

#### 7.2.1.3 Comments

Comments are messages intended only for human consumption.

## 7.2.2 Pseudo-Ops

### 7.2.2.1 .ORIG

It tells the assembler where in the memory to place the LC-3 program.

### 7.2.2.2 .FILL

It tells the assembler to set aside the next location in the program and initialize it with the value of the operand.

### 7.2.2.3 .BLKW

(a **BL**o**C**k of **W**ords)

It tells the assembler to set aside some number of sequential memory locations in the program.

### 7.2.2.4 .STRINGZ

It tells the assembler to initialize a sequence of  $n + 1$  memory locations. The first  $n$  words of memory are initialized with the zero-extended ASCII codes of the corresponding characters in the string. The final word of memory is initialized to 0.

### 7.2.2.5 .END

It tells the assembler it has reached the end of the program and need not even look at anything after it.

**HINT:** .END does not stop the program & it does not even exist at the time of execution.

### \*(in 7.4.2) .EXTERNAL

It send a message to LC-3 assembler that the absence of the label corresponding to the pseudo-ops is not an error.

Used for multi-file programming.

## Machine Code

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD*	0001				DR		SR1		0		00		SR2			
ADD*	0001				DR		SR1		1		imm5					
AND*	0101				DR		SR1		0		00		SR2			
AND*	0101				DR		SR1		1		imm5					
BR	0000				n z p		PCOffset9									
JMP	1100				000		BaseR		000000							
JSR	0100				1		PCOffset11									
JSRR	0100				0 00		BaseR		000000							
LD*	0010				DR		PCOffset9									
LDI*	1010				DR		PCOffset9									
LDR*	0110				DR		BaseR		offset6							
LEA	1110				DR		PCOffset9									
NOT*	1001				DR		SR		111111							
RET	1100				000		111		000000							
RTI	1000				000000000000											
ST	0011				SR		PCOffset9									
STI	1011				SR		PCOffset9									
STR	0111				SR		BaseR		offset6							
TRAP	1111				0000		trapvect8									
reserved	1101															

## Assemble Language

ADD DR, SR1, SR2	ADD r1, r2, r3
ADD DR, SR1, imm5	ADD r1, r2, #10 <i>+进制</i>
AND DR, SR1, SR2	<i>[16,15] b1010 = 进制</i>
AND DR, SR1, imm5	<i>xA + 进制</i>
BRn LABEL	BR/BRnzp, BRz, BRp, BRnz, BRnp, BRzp
JMP BaseR	
JSR LABEL	
JSRR LABEL	
LD DR, LABEL	
LDI DR, LABEL	
LDR DR, BaseR, offset6	
LEA DR, LABEL	
NOT DR, SR	
RET	
RTI	
ST SR, LABEL	
STI SR, LABEL	
STR SR, BaseR, offset6	
TRAP trapvect8	TRAP x25

Pseudo-Ops :

- .ORIG x3000
- .FILL xFFFF
- .END

.BLKW 2

.STRINGZ "Hello World"

.EXTERNAL START of FILE

Pseudo-Ops: .ORIG x3000  
 .FILL xFFFF  
 .END

.BLKW 2  
 .STRINGZ "Hello World"  
 .EXTERNAL START of FILE \*

## 7.3 The Assembly Process

### 7.3.2 A Two-Pass Process

**The First Pass:** Creating the Symbol Table

**The Second Pass:** Generating the Machine Language Program

## 7.4 Beyond the Assembly of a Single Assembly Language Program (*Recommend Reading*)

The concept of The Executable Image

Multi-file programming