

어셈블프로그래밍 이해

Ch6. 기본 컴퓨터 프로그래밍

- 사용자가 작성한 프로그램 : 기계에 독립적 또는 종속적
- 번역 프로그램 : 기계 종속적
 - 특정한 컴퓨터의 하드웨어에 의해 인지되어야 함
- 6장의 주요내용
 - 기본 프로그래밍의 개념(어셈블 프로그래밍)
 - 명령의 하드웨어 실행과의 관계
- 기호 프로그램을 이진 프로그램으로 번역하는 기본 동작과 구조
 - 각 명령어는 3개의 문자로 나타냄
 - 메모리 참조 명령어는 3부분으로 나타냄 : 모드비트, 연산코드, 주소 비트

간접주소 명령어

표5-2 컴퓨터 명령어

Symbol	Hexadecimal code		Description
	I = 0	I = 1	
AND	0XXX	8XXX	AND memory word to AC
ADD	1XXX	9XXX	Add memory word to AC
LDA	2XXX	AXXX	Load memory word to AC
STA	3XXX	BXXX	Store content of AC in memory
BUN	4XXX	CXXX	Branch unconditionally
BSA	5XXX	DXXX	Branch and save return address
ISZ	6XXX	EXXX	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7400		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E 0
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

메모리참조
모드비트 : 1
연산코드 : 3
주소비트 : 12

레지스터참조

입출력 명령어

기본 컴퓨터의 25개 명령어

6.2 기계어(Machine Language)

■ 프로그램

- 1. 이진코드 : 메모리상에 실제 나타나는 형태
- 2. 8진수 또는 16진수 : 2진수 코드를 알아보기 쉽게 사용
- 3. 기호코드 : 어셈블리 언어
- 4. 고급 프로그래밍 언어 : 컴퓨터 언어로 컴파일러에 의해 번역됨

표 6-2 두 수를 가산하는 이진 프로그램

Location	Instruction code
000	0010 0000 0000 0100
001	0001 0000 0000 0101
010	0011 0000 0000 0110
011	0111 0000 0000 0001
100	0000 0000 0101 0011
101	1111 1111 1110 1001
110	0000 0000 0000 0000

표 6-3 두 수를 가산하는 16진 프로그램

Location	Instruction code
000	2004
001	1005
002	3006
003	7001
004	0053
005	FFE9
006	0000

표 6-4 기호 연산 코드를 가진 프로그램

Location	Instruction code
000	LDA 004
001	ADD 005
002	STA 006
003	HLT
004	0053
005	FFE9
006	0000

AC에 004 번지의 내용을 가져온다.

AC와 005 번지의 내용을 더한다.

AC의 값을 006 번지에 저장

표 6-5 두 수를 가산하는 어셈블리 언어 프로그램

```

ORG 0
LDA A
ADD B
STA C
HLT
A, DEC 83
B, DEC -23
C, DEC 0
END

```

AC에 A 번지의 내용을 가져온다.

AC와 B 번지의 내용을 더한다.

AC의 값을 C 번지에 저장

표 6-6 두 수를 가산하는 포트란 프로그램

```
INTEGER A,B,C  
DATA A,83 B, -23  
C=A+B  
END
```

표 6-6의 포트란 프로그램은 표 6-2의 이진수로 번역된다.

6.3 어셈블리 언어

■ 언어규칙

- Label field : 기호 주소를 나타내거나 빈칸
- Instruction(opcode+operand) field : 기계 명령어나 슈도 명령 기술
- Comment field : 명령어에 대한 해설

■ 명령어 필드

- MRI : 메모리 참조(Memory Reference Instruction)
- non-MRI : 레지스터 참조 또는 입출력
- Pseudo 명령어 : 번역과정에서 필요한 정보를 알려주는 명령어
 - ORG N : 프로그램의 시작위치
 - END : 심볼릭 프로그램의 끝
 - DEC N : 10진수 N
 - HEX N : 16진수 N

두 수를 감산하는 어셈블리 언어 프로그램

표 6-8 두 수를 감산하는 어셈블리 언어 프로그램

ORG 100 LDA SUB CMA INC ADD MIN STA DIF HLT MIN, DEC 83 SUB, DEC -23 DIF, HEX 0 ↑ END	1의 보수 2의 보수
------------------------------------------------------------------------------------------------------------------------	----------------

label field

표 6-9 표 6-8의 번역된 프로그램의 리스팅

Hexadecimal code		
Location	Content	symbolic program
		ORG 100
100	2107	LDA SUB
101	7200	CMA
102	7020	INC
103	1106	ADD MIN
104	3108	STA DIF
105	7001	HLT
106	0053	MIN, DEC 83
107	FFE9	SUB, DEC -23
108	0000	DIF, HEX 0
		END

First scan : 기호-주소 테이블 작성

Second scan : 이진프로그램으로 번역

6-4 어셈블러

- 기호 언어 프로그램을 읽어 이진 프로그램으로 번역
- 기호 프로그램 : 소스 프로그램(source program)
- 이진 프로그램 : 목적 프로그램(object program)
- 소스프로그램
 - ASCII code 로 기술됨
 - CR : Carrage return(Hex : 0D)
- 목적 프로그램
 - 소스 프로그램을 두번의 번역 과정을 통해 형성됨
 - 1st pass : label의 주소를 알아냄(기호-주소 테이블 만듬)
 - 2nd pass : 명령어 번역(이진코드 생성)

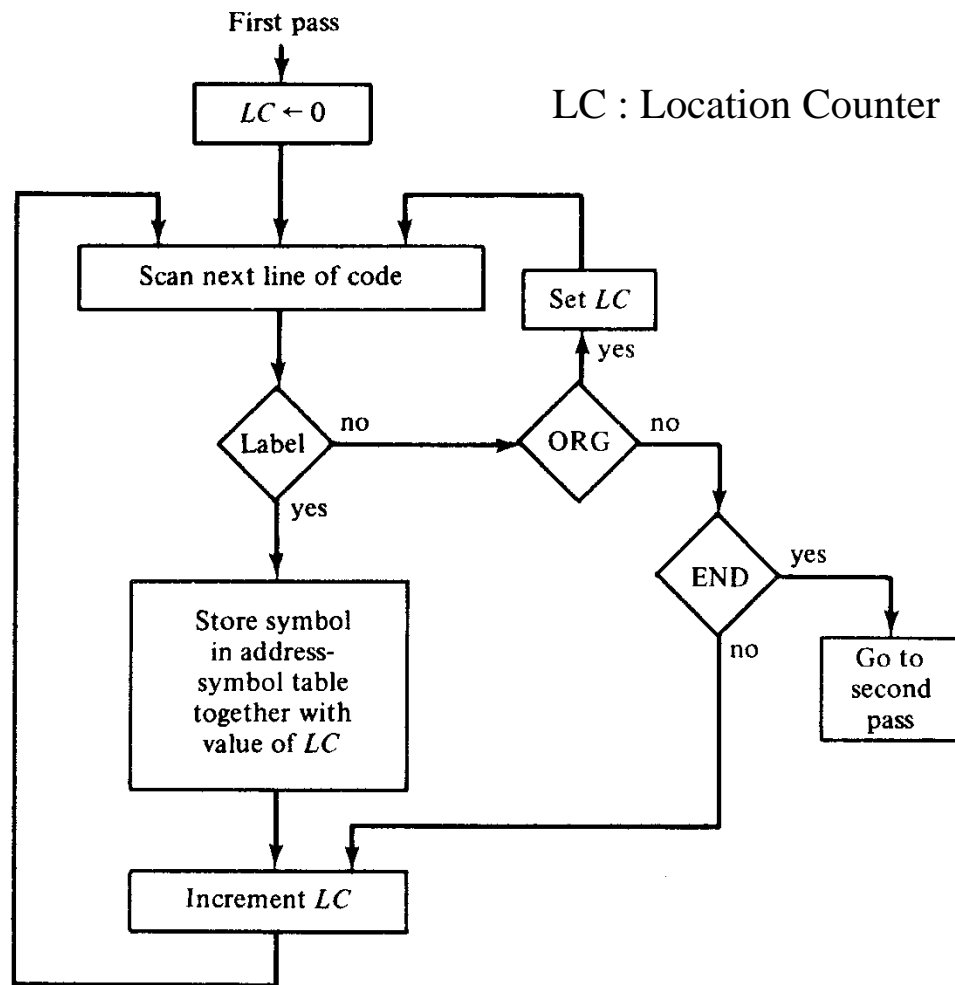


그림 6-1 어셈블러의 first pass 에 대한 흐름도

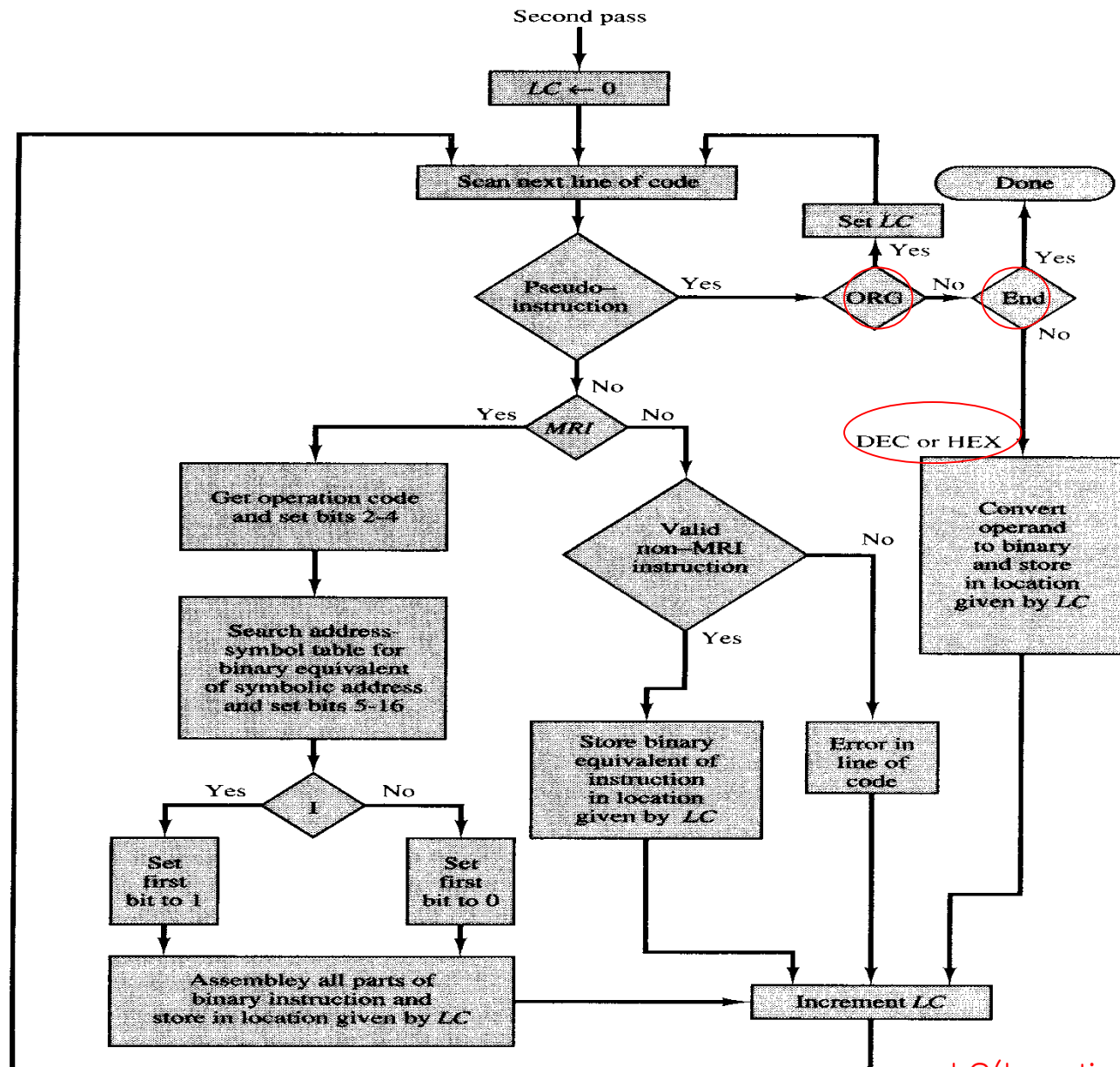


그림 6-2 어셈블러의 second pass를 위한 흐름도

LC(Location Counter)

6-5 프로그램 루프

```
DIMENSION A(100)
INTEGER SUM,A
SUM = 0
DO 3 J=1, 100
3 SUM=SUM+A(J)
```

ISZ : Increment M and skip if zero

```
ORG 100
LDA ADS
STA PTR
LDA NBR
STA CTR
CLA
LOP, ADD PTR I
ISZ PTR
ISZ CTR
BUN LOP
STA SUM
HLT
ADS, HEX 150
PTR, HEX 0
NBR, DEC -100
CTR, HEX 0
SUM, HEX 0
ORG 150
DEC 75
:
DEC 23
END
```

	ORG 100	
	LDA ADS	$AC \leftarrow 0x150$
	STA PTR	$PTR \leftarrow 0x150$
	LDA NBR	$AC \leftarrow -100$
	STA CTR	$CTR \leftarrow -100$
	CLA	$AC \leftarrow 0$
LOP,	ADD PTR I	$AC \leftarrow 0+75$ // $AC \leftarrow AC+150$ 번지 내용
	ISZ PTR	$PTR \leftarrow 0x151$
	ISZ CTR	$CTR \leftarrow -99$ (CTR이 0이면 skip next line)
	BUN LOP	과정 반복(CTR이 0될 때까지)
	STA SUM	$SUM \leftarrow AC$ (100개의 데이터 더해진 결과)
	HLT	
ADS,	HEX 150	
PTR,	HEX 0	
NBR,	DEC -100	
CTR,	HEX 0	
SUM,	HEX 0	
	ORG 150	
	DEC 75	150 번지부터 100개의 데이터가 메모리에 배치
	:	
	DEC 23	
	END	



수고하셨습니다!