Pr.1) “Systems Programming and Operating Systems”, by D.M. Dhamdhere, Second Edition , Page No.97 (Fig.4.8)

Problem statement : Apply PassI of a two-pass Assembler to the following assembly language code and generate Intermediate Code (IC), Symbol Table, Literal Table and POOL Table according to IC VarientI.

Line No.	Label	Opcode	Operand1	Operand2
1		START	200	
2		MOVER	AREG	=’5’
3		MOVEM	AREG	A
4	LOOP	MOVER	AREG	A
5		MOVER	CREG	B
6		ADD	CREG	=’1’
7		MOVER	AREG	A
8		MOVER	CREG	B
9		MOVER	AREG	A
10		MOVER	CREG	B
11		MOVER	AREG	A
12		BC	ANY	NEXT
13		LTORG		
14		MOVER	AREG	A
15	NEXT	SUB	AREG	=’1’
16		BC	LT	BACK
17	LAST	STOP		
18		ORIGIN	LOOP+2	
19		MULT	CREG	B
20		ORIGIN	LAST+1	
21	A	DS	1	
22	BACK	EQU	LOOP	
23	B	DS	1	
24		END		

Solution : Following tables are used as input along with the above mentioned assembly language code input to solve the problem. While implementing in the laboratory, you can hard code these tables in any suitable data structures.

OPTAB

Mnemonic Opcode	Class	Code for mnemonic
STOP	IS	00
ADD	IS	01
SUB	IS	02
MULT	IS	03
MOVER	IS	04
MOVEM	IS	05
COMP	IS	06
BC	IS	07
DIV	IS	08
READ	IS	09
PRINT	IS	10
START	AD	01
END	AD	02
ORIGIN	AD	03
EQU	AD	04
LTORG	AD	05
DC	DL	01
DS	DL	02

Registers

AREG	1
BREG	2
CREG	3
DREG	4

Condition Codes

LT	1
LE	2
EQ	3
GT	4
GE	5
ANY	6

Solution

Intermediate Code (IC)

	Source Code (Input to PassI of assembler)				Location Counter (LC)	Intermediate Code (IC) (Output of PassI of assembler)		
Line No.	Label	Opcode	Operand1	Operand2		IC for Opcode	IC for Operand1	IC for Operand2
1		START	200			(AD,01)	(C,200)	
2		MOVER	AREG	=’5’	200	(IS,04)	(1)	(L,01)
3		MOVEM	AREG	A	201	(IS,05)	(1)	(S,01)
4	LOOP	MOVER	AREG	A	202	(IS,04)	(1)	(S,01)
5		MOVER	CREG	B	203	(IS,04)	(3)	(S,03)
6		ADD	CREG	=’1’	204	(IS,01)	(3)	(L,02)
7		MOVER	AREG	A	205	(IS,04)	(1)	(S,01)
8		MOVER	CREG	B	206	(IS,04)	(3)	(S,03)
9		MOVER	AREG	A	207	(IS,04)	(1)	(S,01)
10		MOVER	CREG	B	208	(IS,04)	(3)	(S,03)
11		MOVER	AREG	A	209	(IS,04)	(1)	(S,01)
12		BC	ANY	NEXT	210	(IS,07)	(6)	(S,04)
13		LTORG			211	(DL,01)	(C,5)	
					212	(DL,01)	(C,1)	
14		MOVER	AREG	A	213	(IS,04)	(1)	(S,01)
15	NEXT	SUB	AREG	=’1’	214	(IS,02)	(1)	(L,03)
16		BC	LT	BACK	215	(IS,07)	(1)	(S,05)
17	LAST	STOP			216	(IS,00)		
18		ORIGIN	LOOP+2			(AD,03)	(S,02)+2	
19		MULT	CREG	B	204	(IS,03)	(3)	(S,03)
20		ORIGIN	LAST+1			(AD,03)	(S,06)+1	
21	A	DS	1		217	(DL,02)	(C,1)	
22	BACK	EQU	LOOP		No IC, Reflect in SYMTAB			
23	B	DS	1		218	(DL,02)	(C,1)	
24		END				(AD,02)		
25					219	(DL,01)	(C,1)	

Symbol Table (SYMTAB)

Symbol	Address
A	217
LOOP	202
B	218
NEXT	214
BACK	202
LAST	216

Literal Table (LITTAB)

Literal	Address
=’ 5’	211
=’1’	212
=’1’	219

PoolTable (POOLTAB)

Literal No.
#1
#3
