Line **Source statement**

5	COPY	START	0	COPY FILE FROM INPUT OUTPUT
10	FIRST	STL	RETADR	SAVE RETURN ADDRESS
12		LDB	#LENGTH	ESTABLISH BASE REGISTER
13		BASE	LENGTH	
15	CLOOP	+JSUB	RDREC	READ INPUT RECORD
20		LDA	LENGTH	TEST FOR EOF (LENGTH = 0)
25		COMP	#0	
30		JEQ	ENDFIL	EXIT IF EOF FOUND
35		+JSUB	WRREC	WRITE OUTPUT RECORD
40		J	CLOOP	LOOP
45	ENDFIL	LDA	EOF	INSTER END OF FILE MARKER
50		STA	BUFFER	
55		LDA	#3	SET LENGTH = 3

Line	Source statement			
60		STA	LENGTH	
65		+JSUB	WRREC	WRITE EOF
70		J	@RETADR	RETURN TO CALLER
80	EOF	BYTE	C'EOF'	
95	RETADR	RESW	1	
100	LENGTH	RESW	1	LENGTH OF RECORD
105	BUFFER	RESB	4096	4096-BYTE BUFFER AREA
110	.			
115	.	SUBROUTINE TO READ RECORD INTO BUFFER		
120	.			
125	RDREC	CLEAR	X	CLEAR LOOP COUNTER
130		CLEAR	A	CLEAR A TO ZERO

Line	Source statement			
132		CLEAR	S	CLEAR S TO ZERO
133		+LDT	#4096	
135	RLOOP	TD	INPUT	TEST INPUT DEVICE
140		JEQ	RLOOP	LOOP UNTIL READY
145		RD	INPUT	READ CHARACTER INTO REGISTER A
150		COMPR	A,S	TEST FOR END OF RECORD (X'00')
155		JEQ	EXIT	EXIT LOOP IF EOF
160		STCH	BUFFER,X	STORE CHARACTER IN BUFFER
165		TIXR	T	LOOP UNLESS MAX LENGTH
170		JLT	RLOOP	HAS BEEN REACHED
175	EXIT	STX	LENGTH	SAVE RECORD LENGTH

Line	Source statement			
180		RSUB		RETURN TO CALLER
185	INPUT	BYTE	X'F1'	CODE FOR INPUT DRVICE
195	.			
200	.	SUBROUTINE TO WRITE RECORD FROM BUFFER		
205	.			
210	WRREC	CLEAR	X	CLEAR LOOP COUNTER
212		LDT	LENGTH	
215	WLOOP	TD	OUTPUT	TEST OUTPUT DEVICE
220		JEQ	WLOOP	LOOP UNTIL READY

Line	Source statement			
225		LDCH	BUFFER,X	GET CHARACTER FROM BUFFER
230		WD	OUTPUT	WRITE CHARACTER
235		TIXR	T	LOOP UNTIL ALL CHARATERS
240		JLT	WLOOP	HAVE BEEN WRITTEN
245		RSUB		RETURN TO CALLER
250	OUTPUT	BYTE	X'05'	CODE FOR OUTPUT DEVICE
255		END	FIRST	

Figure 2.5 Example of a SIC/XE program.

Line	Loc	Source statement			Object code
5	0000	COPY	START	0	
10	0000	FIRST	STL	RETADR	17202D
12	0003		LDB	#LENGTH	69202D
13			BASE	LENGTH	
15	0006	CLOOP	+JSUB	RDREC	4B101036
20	000A		LDA	LENGTH	032026
25	000D		COMP	#0	290000
30	0010		JEQ	ENDFIL	332007
35	0013		+JSUB	WRREC	4B10105D
40	0017		J	CLOOP	3F2FEC
45	001A	ENDFIL	LDA	EOF	032010
50	001D		STA	BUFFER	0F2016
55	0020		LDA	#3	010003
60	0023		STA	LENGTH	0F200D

Line	Loc	Source statement			Object code
65	0026		+JSUB	WRREC	4B10105D
70	002A		J	@RETADR	3E2003
80	002D	EOF	BYTE	C'EOF'	454F46
95	0030	RETADR	RESW	1	
100	0033	LENGTH	RESW	1	
105	0036	BUFFER	RESB	4096	
110		.			
115		.	SUBROUTINE TO READ RECORD INTO BUFFER		
120		.			
125	1036	RDREC	CLEAR	X	B410
130	1038		CLEAR	A	B400
132	103A		CLEAR	S	B440

Line	Loc	Source statement			Object code
133	103C		+LDT	#4096	75101000
135	1040	RLOOP	TD	INPUT	E32019
140	1043		JEQ	RLOOP	332FFA
145	1046		RD	INPUT	DB2013
150	1049		COMPR	A,S	A004
155	104B		JEQ	EXIT	332008
160	104E		STCH	BUFFER,X	57C003
165	1051		TIXR	T	B850
170	1053		JLT	RLOOP	3B2FEA
175	1056	EXIT	STX	LENGTH	134000
180	1059		RSUB		4F0000
185	105C	INPUT	BYTE	X'F1'	F1
195		.			

Line	Loc	Source statement			Object code
200		.	SUBROUTINE TO WRITE RECORD FROM BUFFER		
205		.			
210	105D	WRREC	CLEAR	X	B410
212	105F		LDT	LENGTH	774000
215	1062	WLOOP	TD	OUTPUT	E32011
220	1065		JEQ	WLOOP	332FFA
225	1068		LDCH	BUFFER,X	53C003
230	106B		WD	OUTPUT	DF2008
235	106E		TIXR	T	B850
240	1070		JLT	WLOOP	3B2FEF
245	1073		RSUB		4F0000
250	1076	OUTPUT	BYTE	X’05’	05
255			END	FIRST	

Figure 3.4 Example of a SIC/XE program.

Instruction Formats

Format 1 (1 byte):

8
op

Format 2 (2 bytes):

8	4	4
op	r1	r2

Format 3 (3 bytes):

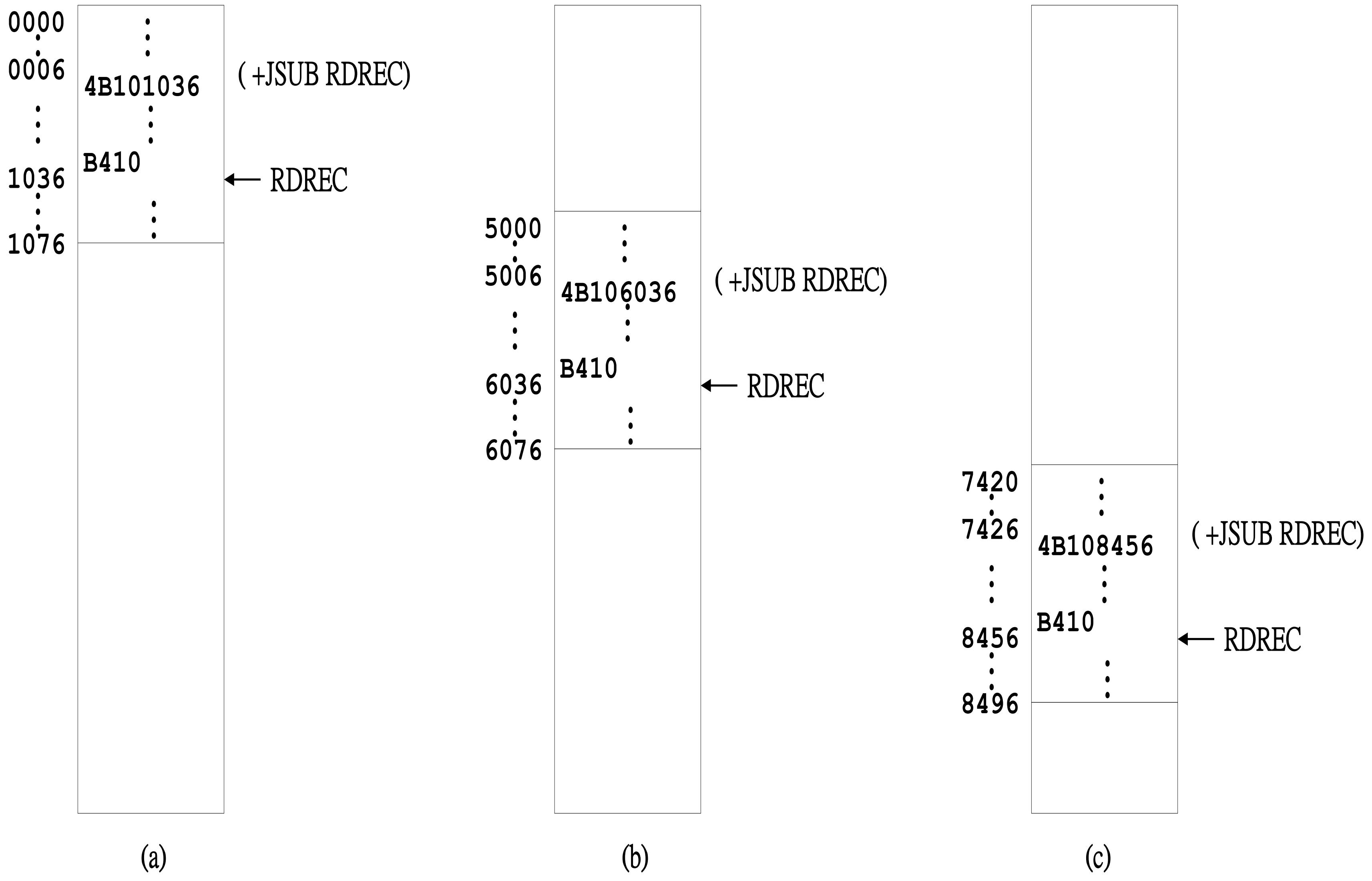
6	1	1	1	1	1	1	12
op	n	i	x	b	p	e	disp

Format 4 (4 bytes):

6	1	1	1	1	1	1	20
op	n	i	x	b	p	e	address

Addressing type	Flag bits nixbpe	Assembler Language notation	Calculation Of target Address TA	Operand	Notes
Simple	110000	op c	disp	(TA)	D
	110001	+op m	addr	(TA)	4D
	110010	op m	(PC)+disp	(TA)	A
	110100	op m	(B)+disp	(TA)	A
	111000	op c,X	disp+(X)	(TA)	D
	111001	+op m,X	addr+(X)	(TA)	4D
	111010	op m,X	(PC)+disp+(X)	(TA)	A
	111100	op m,X	(B)+disp+(X)	(TA)	A
	000---	op m	b/p/e/disp	(TA)	D S
	001---	op m,X	b/p/e/disp+(X)	(TA)	D S

Addressing type	Flag bits nixbpe	Assembler Language notation	Calculation Of target Address TA	Operand	Notes
Indirect	100000	op @c	disp	((TA))	D
	100001	+op @m	addr	((TA))	4D
	100010	op @m	(PC)+disp	((TA))	A
	100100	op @m	(B)+disp	((TA))	A
Immediate	010000	op #c	disp	TA	D
	010001	+op #m	addr	TA	4D
	010010	op #m	(PC)+disp	TA	A
	010100	op #m	(B)+disp	TA	A



Examples of program relocation.

H：Header (+programs name + 起始位置 + 長度)

T：Text

M：Modification

} Record

E：End

T(+ 起始位置 + size + 內容)

M(位置 + 自第n bit +(+/-)住址)

E(起始pc)

```
H^COPY  ^000000^001077
T^000000^1D^17202D^69202D^4B101036^032026^290000^332007^4B10105D^3F2FEC^032010
T^00001D^13^0F2016^010003^0F200D^4B10105D^3E2003^454F46
T^001036^1D^B410^B400^B440^75101000^E32019^332FFA^DB2013^A004^332008^57C003^B85
0
T^001053^1D^3B2FEA^134000^4F0000^F1^B410^774000^E32011^332FFA^53C003^DF2008^B85
0
T^001070^07^3B2FEF^4F0000^05
M^000007^05+COPY
M^000014^05+COPY
M^000027^05+COPY
E^000000
```

Figure 3.5 Object program with relocation by Modification records.

Line	Loc	Source Statement			Object code
5	0000	COPY	START	0	
10	0000	FIRST	STL	RETADR	140033
15	0003	CLOOP	JSUB	RDREC	481039
20	0006		LDA	LENGTH	000036
25	0009		COMP	ZERO	280030
30	000C		JEQ	ENDFIL	300015
35	000F		JSUB	WRREC	481061
40	0012		J	CLOOP	3C0003
45	0015	ENDFIL	LDA	EOF	00002A
50	0018		STA	BUFFER	0C0039
55	001B		LDA	THREE	00002D
60	001E		STA	LENGTH	0C0036
65	0021		JSUB	WRREC	481061
70	0024		LDL	RETADR	080033

Line	Loc	Source Statement			Object code
75	0027		RSUB		4C0000
80	002A	EOF	BYTE	C’EOF’	454F46
85	002D	THREE	WORD	3	000003
90	0030	ZERO	WORD	0	000000
95	0033	RETADR	RESW	1	
100	0036	LENGTH	RESW	1	
105	0039	BUFFER	RESB	4096	
110		.			
115		.	SUBROUTINE TO READ RECORD INTO BUFFER		
120		.			
125	1039	RDREC	LDX	ZERO	040030
130	103C		LDA	ZERO	000030
135	103F	RLOOP	TD	INPUT	E0105D

Line	Loc	Source Statement			Object code
140	1042		JEQ	RLOOP	30103F
145	1045		RD	INPUT	D8105D
150	1048		COMP	ZERO	280030
155	104B		JEQ	EXIT	301057
160	104E		STCH	BUFFER,X	548039
165	1051		TIX	MAXLEN	2C105E
170	1054		JLT	RLOOP	38103F
175	1057	EXIT	STX	LENGTH	100036
180	105A		RSUB		4C0000
185	105D	INPUT	BYTE	X'F1'	F1
190	105E	MAXLEN	WORD	4096	001000
195 200 205		.	SUBROUTINE TO WRITE RECORD FROM BUFFER		

Line	Loc	Source Statement			Object code
210	1061	WRREC	LDX	ZERO	040030
215	1064	WLOOP	TD	OUTPUT	E01079
220	1067		JEQ	WLOOP	301064
225	106A		LDCH	BUFFER,X	508039
230	106D		WD	OUTPUT	DC1079
235	1070		TIX	LENGTH	2C0036
240	1073		JLT	LOOP	381064
245	1076		RSUB		4C0000
250	1079	OUTPUT	BYTE	X'05'	05
255			END	FIRST	

Figure 3.6 Relocatable program for a standard SIC machine.

```
H^COPY  ^000000^00107A
T^000000^1E^FEC^140033^481039^000036^280030^300015^481061^3C0003^00002A^0C0039^00002D
T^00001E^15^E00^0C0036^481061^080033^4C0000^454F46^000003^000000
T^001039^1E^FFC^040030^000030^E0105D^30103F^D8105D^280030^301057^548039^2C105E^38103F
T^001057^0A^800^100036^4C0000^F1^001000
T^001061^19^FE0^040030^E01079^301064^508039^DC1079^2C0036^381064^4C0000^05
E^000000
```

Figure 3.7 Object program with relocation by bit mask.

Loc	Source statement			Object code
0000	PROGA	START	0	
		EXTDEF	LISTA,ENDA	
		EXTREF	LISTB,ENDB,LISTC,ENDC	
		.		
		.		
		.		
0020	REF1	LDA	LISTA	03201D
0023	REF2	+LDT	LISTB+4	77100004
0027	REF3	LDX	#ENDA−LISTA	050014
		.		
		.		
		.		

Loc	Source statement			Object code
0040	LISTA	EQU	*	
		.		
		.		
0054	ENDA	EQU	*	
0054	REF4	WORD	ENDA−LISTA+LISTC	000014
0057	REF5	WORD	ENDC−LISTC−10	FFFFF6
005A	REF6	WORD	ENDC−LISTC+LISTA−1	00003F
005D	REF7	WORD	ENDA−LISTA−(ENDB−LISTB)	000014
0060	REF8	WORD	LISTB−LISTA	FFFFC0
		END	REF1	

Loc	Source statement			Object code
0000	PROG B	START	0	
		EXTDEF	LISTB,ENDB	
		EXTREF	LISTA,ENDA,LISTC,ENDC	
		.		
		.		
		.		
0036	REF1	+LDA	LISTA	03100000
003A	REF2	LDT	LISTB+4	772027
003D	REF3	+LDX	#ENDA−LISTA	05100000
		.		
		.		
		.		
0060	LISTB	EQU	*	

Loc	Source statement			Object code
		.		
		.		
0070	ENDB	EQU	*	
0070	REF4	WORD	ENDA−LISTA+LISTC	000000
0073	REF5	WORD	ENDC−LISTC−10	FFFFF6
0076	REF6	WORD	ENDC−LISTC+LISTA−1	FFFFFF
0079	REF7	WORD	ENDA−LISTA−(ENDB−LISTB)	FFFFF0
007C	REF8	WORD	LISTB−LISTA	000060
		END		

Figure 3.8 Sample program illustrating linking and relocation.

Loc	Source statement			Object code
0000	PROGC	START	0	
		EXTDEF	LISTC,ENDC	
		EXTREF	LISTA,ENDA,LISTB,ENDB	
		.		
		.		
		.		
0018	REF1	+LDA	LISTA	03100000
001C	REF2	+LDT	LISTB+4	77100004
0020	REF3	+LDX	#ENDA−LISTA	05100000
		.		
		.		
		.		

Loc	Source statement			Object code
0030	LISTC	EQU	*	
		.		
		.		
0042	ENDC	EQU	*	
0042	REF4	WORD	ENDA−LISTA+LISTC	000030
0045	REF5	WORD	ENDC−LISTC−10	000008
0048	REF6	WORD	ENDC−LISTC+LISTA−1	000011
004B	REF7	WORD	ENDA−LISTA−(ENDB−LISTB)	000000
004E	REF8	WORD	LISTB−LISTA	000000
		END		

Figure 3.8 (cont'd)

```
H^PROGA ^000000^000063
D^LISTA ^000040^ENDA  ^000054
R^LISTB ^ENDB  ^LISTC ^ENDC
 .
 .
T^000020^0A^03201D^77100004^050014
 .
 .
T^000054^0F^000014^FFFFF6^00003F^000014^FFFFC0
```

Figure 3.9 Object programs corresponding to Fig. 3.8.

```
T^000054^0F^000014^FFFFF6^00003F^000014^FFFFC0
M^000024^05^+LISTB
M^000054^06^+LISTC
M^000057^06^+ENDC
M^000057^06^-LISTC
M^00005A^06^+ENDC
M^00005A^06^-LISTC
M^00005A^06^+PROGA
M^00005D^06^-ENDB
M^00005D^06^+LISTB
M^000060^06^+LISTB
M^000060^06^-PROGA
E^000020
```

Figure 3.9 (cont'd)

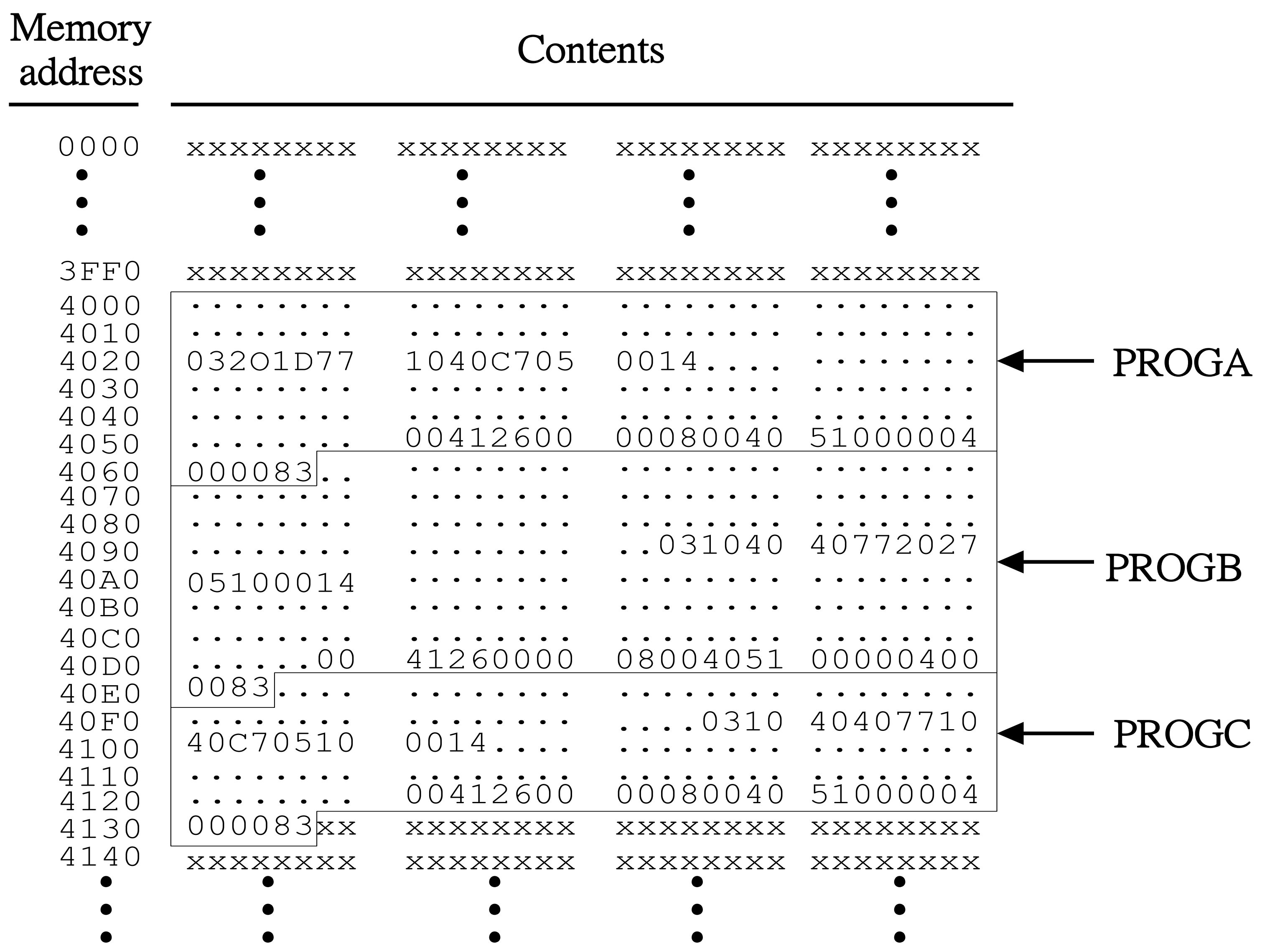
```
H^PROGB ^000000^00007F
D^LISTB ^000060^ENDB  ^000070
R^LISTA ^ENDA  ^LISTC ^ENDC
 .
 .
T^000036^0B^03100000^772027^05100000
 .
 .
T^000070^0F^000000^FFFFF6^FFFFFF^FFFFF0^000060


M^000037^05^+LISTA

M^00003E^05^+ENDA

M^00003E^05^-LISTA
```

```
M^000070^06^+ENDA
M^000070^06^-LISTA
M^000070^06^+LISTC
M^000073^06^+ENDC
M^000073^06^-LISTC
M^000076^06^+ENDC
M^000076^06^-LISTC
M^000076^06^+LISTA
M^000079^06^+ENDA
M^000079^06^-LISTA
M^00007C^06^+PROGB
M^00007C^06^-LISTA
E
```



```
Memory
address     Contents

0000   xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx
 •        •         •         •         •
 •        •         •         •         •
 •        •         •         •         •
3FF0   xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx
4000   ........  ........  ........  ........
4010   ........  ........  ........  ........
4020   03201D77  1040C705  0014....  ........
4030   ........  ........  ........  ........
4040   ........  ........  ........  ........
4050   ........  00412600  00080040  51000004
4060   000083..  ........  ........  ........
4070   ........  ........  ........  ........
4080   ........  ........  ........  ........
4090   ........  ........  ..031040  40772027
40A0   05100014  ........  ........  ........
40B0   ........  ........  ........  ........
40C0   ........  ........  ........  ........
40D0   ......00  41260000  08004051  00000400
40E0   0083....  ........  ........  ........
40F0   ........  ........  ....0310  40407710
4100   40C70510  0014....  ........  ........
4110   ........  ........  ........  ........
4120   ........  00412600  00080040  51000004
4130   000083xx  xxxxxxxx  xxxxxxxx  xxxxxxxx
4140   xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx
 •        •         •         •         •
 •        •         •         •         •
 •        •         •         •         •
```

Figure 3.10(a) Programs from Fig. 3.8 after linking and loading.

```
H^PROGC ^000000^000051
D^LISTC ^000030^ENDC  ^000042
R^LISTA ^ENDA  ^LISTB ^ENDB
 .
 .
T^000018^0C^03100000^77100004^05100000
 .
 .
T^000042^0F^000030^000008^000011^000000^000000
M^000019^05^+LISTA
M^00001D^05^+LISTB
M^000021^05^+ENDA
```

```
M^000021^05^-LISTA
M^000042^06^+ENDA
M^000042^06^-LISTA
M^000042^06^+PROGC
M^000048^06^+LISTA
M^00004B^06^+ENDA
M^00004B^06^-LISTA
M^00004B^06^-ENDB
M^00004B^06^+LISTB
M^00004E^06^+LISTB
M^00004E^06^-LISTA
E
```

Figure 3.9 (cont'd)

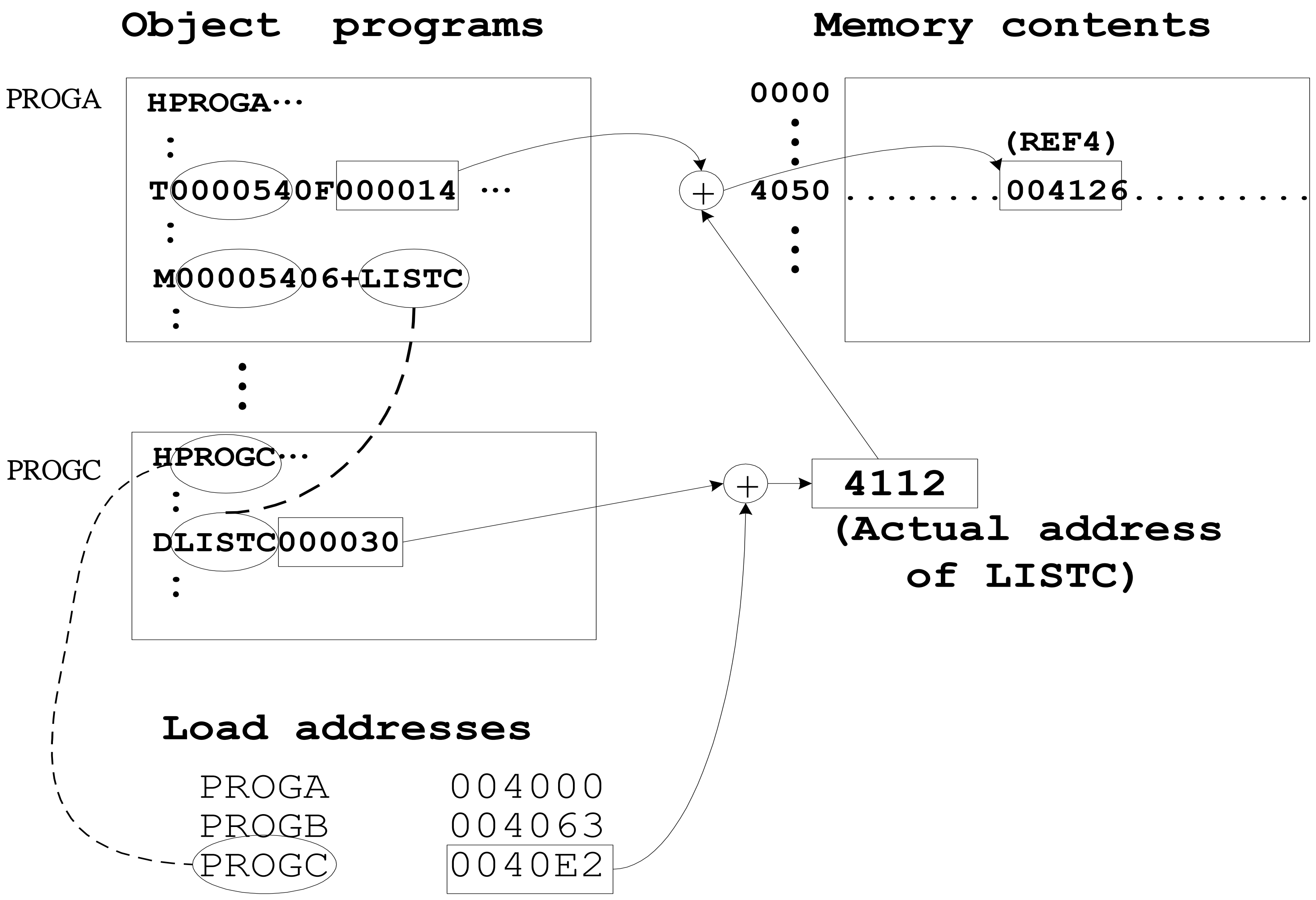


Figure 3.10(b) Relocation and linking operations performed on REF4 from PROGA.

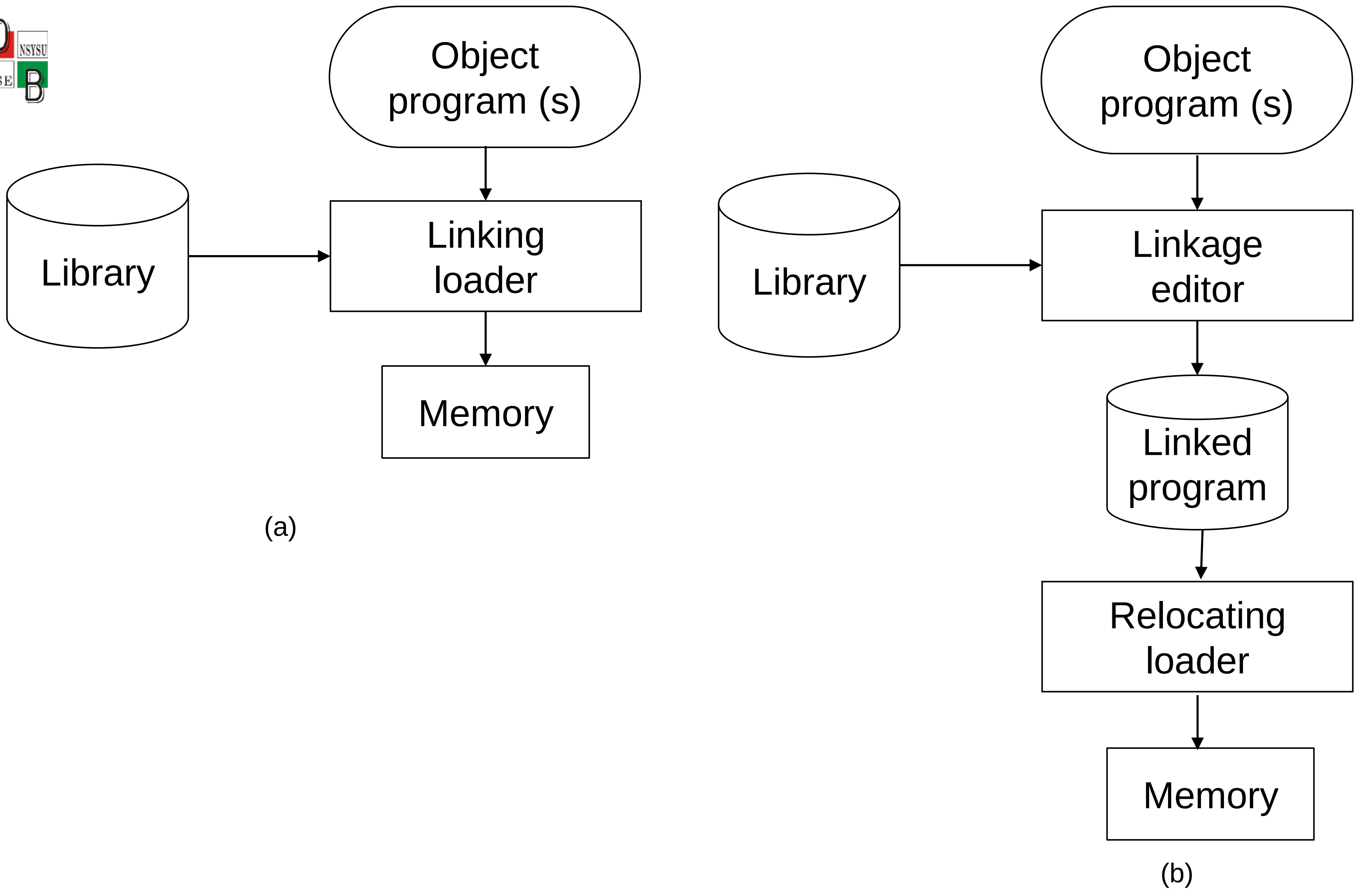


Figure 3.13 Processing of an object program using (a) linking loader and (b) linkage editor.

Line Symbolic Representation of Generated Code

1	STATS	START	0	{program header}
		EXTREF	XREAD,XWRITE	
		STL	RETADR	{save return address}
		J	{EXADDR}	
	RETADR	RESW	1	
3	SUM	RESW	1	{variable declarations}
	SUMSQ	RESW	1	
	I	RESW	1	
	VALUE	RESW	1	
	MEAN	RESW	1	
	VARIANCE	RESW	1	
5	{EXADDR}	LDA	#0	{SUM ：= 0}
		STA	SUM	

Line Symbolic Representation of Generated Code

6		LDA	#0	{SUMSQ ：= 0}
		STA	SUMSQ	
7		LDA	#1	{FOR I：= 1 TO 100 }
	{L1}	STA	I	
		COMP	#100	
		JGT	{L2}	
9		+JSUB	XREAD	{READ(VALUE)}
		WORD	1	
		WORD	VALUE	
10		LDA	SUM	{SUM ：= SUM +VALUE}
		ADD	VALUE	
		STA	SUM	

Line Symbolic Representation of Generated Code

11		LDA	VALUE	{SUMSQ ：= SUMSQ +VALUE *VALUE}
		MUL	VALUE	
		ADD	SUMSQ	
		STA	SUMSQ	
		LDA	I	{end of FOR loop}
		ADD	#1	
		J	{L1}	
13	{L2}	LDA	SUM	{MEAN ：= SUM DIV 100}
		DIV	#100	
		STA	MEAN	

Line Symbolic Representation of Generated Code

14		LDA	SUMSQ	{VARIANCE：= SUMSQ DIV 100-MEAN * MEAN}
		DIV	#100	
		STA	T1	
		LDA	MEAN	
		MUL	MEAN	
		STA	T2	
		LDA	T1	
		SUB	T2	
		STA	VARIANCE	

Line Symbolic Representation of Generated Code

15		+JSUB	XWRITE	{WRITE(MEAN, VARIANCE)}
		WORD	2	
		WORD	MEAN	
		WORD	VARIANCE	
		LDL	RETADR	{return}
		RSUB		
	T1	RESW	1	{working variables used}
	T2	RESW	1	
		END		

Figure 5.21 Symbolic representation of object code generated for the program from Fig. 5.1

```
PROGRAM  STATS
VAR
   SUM , SUMSQ , I , VALUE , MEAN , VARIANCE : INTEGER
BEGIN
   SUM := 0;
   SUMSQ := 0;
   FOR I := 1 TO 100 DO
      BEGIN
      READ(VALUE);
      SUM := SUM + VALUE;
      SUMSQ := SUMSQ + VALUE * VALUE
      END;
   MEAN := SUM DIV 100;
   VARIANCE := SUMSQ DIV 100 – MEAN * MEAN;
   WRITE(MEAN , VARIANCE)
END.
```

FIGURE 5.1 Example of a Pascal program.