

(line 207). The order in which symbols are listed in the EXTDEF and EXTREF statements is not significant.

Now we are ready to look at how external references are handled by the assembler. Figure 2.16 shows the generated object code for each statement in the program. Consider first the instruction

```
15      0003      CLOOP      +JSUB      RDREC      4B100000
```

The operand (RDREC) is named in the EXTREF statement for the control section, so this is an external reference. The assembler has no idea where the control section containing RDREC will be loaded, so it cannot assemble the address for this instruction. Instead the assembler inserts an address of zero and passes information to the loader, which will cause the proper address to be inserted at load time. The address of RDREC will have no predictable relationship to anything in this control section; therefore relative addressing is not possible. Thus an extended format instruction must be used to provide room for the actual address to be inserted. This is true of any instruction whose operand involves an external reference.

Similarly, the instruction

```
160     0017      +STCH      BUFFER,X      57900000
```

makes an external reference to BUFFER. The instruction is assembled using extended format with an address of zero. The x bit is set to 1 to indicate indexed addressing, as specified by the instruction. The statement

```
190     0028      MAXLEN      WORD      BUFEND-BUFFER      000000
```

is only slightly different. Here the value of the data word to be generated is specified by an expression involving two external references: BUFEND and BUFFER. As before, the assembler stores this value as zero. When the program is loaded, the loader will add to this data area the address of BUFEND and subtract from it the address of BUFFER, which results in the desired value.

Note the difference between the handling of the expression on line 190 and the similar expression on line 107. The symbols BUFEND and BUFFER are defined in the same control section with the EQU statement on line 107. Thus the value of the expression can be calculated immediately by the assembler. This could not be done for line 190; BUFEND and BUFFER are defined in another control section, so their values are unknown at assembly time.

As we can see from the above discussion, the assembler must remember (via entries in SYMTAB) in which control section a symbol is defined. Any attempt to refer to a symbol in another control section must be flagged as an error unless the symbol is identified (using EXTREF) as an external reference. The assembler must also allow the same symbol to be used in different control

Line	Loc	Source statement	Object code
5	0000	COPY	START 0
6		EXTDEF	BUFFER, BUFEND, LENGTH
7		EXTREF	RDREC, WRREC
10	0000	FIRST	STL RETADR 172027
15	0003	CLOOP	+JSUB RDREC 4B100000
20	0007		LDA LENGTH 032023
25	000A		COMP #0 290000
30	000D		JEQ ENDFIL 332007
35	0010		+JSUB WRREC 4B100000
40	0014		J CLOOP 3F2FEC
45	0017	ENDFIL	LDA =C'EOF' 032016
50	001A		STA BUFFER 0F2016
55	001D		LDA #3 010003
60	0020		STA LENGTH 0F200A
65	0023		+JSUB WRREC 4B100000
70	0027		J @RETADR 3E2000
95	002A	RETADR	RESW 1
100	002D	LENGTH	RESW 1
103			LTORG
	0030	*	=C'EOF' 454F46
105	0033	BUFFER	RESB 4096
106	1033	BUFEND	EQU *
107	1000	MAXLEN	EQU BUFEND-BUFFER
109	0000	RDREC	CSECT
110			.
115			SUBROUTINE TO READ RECORD INTO BUFFER
120			.
122		EXTREF	BUFFER, LENGTH, BUFEND
125	0000	CLEAR	X B410
130	0002	CLEAR	A B400
132	0004	CLEAR	S B440
133	0006	LDT	MAXLEN 77201F
135	0009	RLOOP	TD INPUT E3201B
140	000C		JEQ RLOOP 332FFA
145	000F		RD INPUT DB2015
150	0012		COMPR A,S A004
155	0014		JEQ EXIT 332009
160	0017	+STCH	BUFFER,X 57900000
165	001B		TI XR B850
170	001D		JLT RLOOP 3B2FE9
175	0020	EXIT	+STX LENGTH 13100000
180	0024		RSUB 4F0000
185	0027	INPUT	BYTE X'F1' F1
190	0028	MAXLEN	WORD BUFEND-BUFFER 000000
193			
195	0000	WRREC	CSECT
200			.
205			SUBROUTINE TO WRITE RECORD FROM BUFFER
210			.
212	0000	EXTREF	LENGTH, BUFFER
215	0002	CLEAR	X B410
220	0006	+LDT	LENGTH 77100000
225	0009	TD	=X'05' E32012
230	000C	JEQ	WLOOP 332FFA
235	0010	+LDCH	BUFFER,X 53900000
240	0013	WD	=X'05' DF2008
245	0015	TI XR	T B850
250	0018	JLT	WLOOP 3B2FEE
255		RSUB	4F0000
		END	FIRST
	001B	*	=X'05' 05

Figure 2.16 Program from Fig. 2.15 with object code.