```
procedure DEFINE
   begin
       enter macro name into NAMTAB
       enter macro prototype into DEFTAB
       TEVEL := 1
       while LEVEL > 0 do
          begin
              GETT. THE
              if this is not a comment line then
                 begin
                     substitute positional notation for parameters
                     enter line into DEFTAB
                     if OPCODE = 'MACRO' then
                        LEVEL := LEVEL + 1
                     else if OPCODE = 'MEND' then
                        LEVEL := LEVEL - 1
                 end {if not comment}
          end (while)
       store in NAMTAB pointers to beginning and end of definition
   end {DEFINE}
procedure EXPAND
   begin
       EXPANDING := TRUE
       get first line of macro definition (prototype) from DEFTAB
       set up arguments from macro invocation in ARGTAB
       write macro invocation to expanded file as a comment
       while not end of macro definition do
          begin
              GETLINE
              PROCESSLINE
           end (while)
       EXPANDING := FALSE
    end (EXPAND)
procedure GETLINE
    begin
       if EXPANDING then
          begin
              get next line of macro definition from DEFTAB
              substitute arguments from ARGTAB for positional notation
       else
           read next line from input file
                                                                                               21
    end (GETLINE)
  Figure 4.5 (cont'd)
```

# Machine-Independent Macro Processor Feature

- Concatenation of Macro Parameters
- Generation of Unique Labels
- Conditional Macro Expansion
- Keyword Macro Parameters

# Concatenation of Macro Parameters

- Most macro processors allow parameters to be concatenated with other character strings
  - The need of a special catenation operator
    - LDA X&ID1
    - LDA X&ID
  - The catenation operator
    - LDA X&ID→1
- See figure 4.6

SUM MACRO &ID LDA X&ID→1 ADD X&ID→2 ADD X&ID→3 STA X&TD-S MEND (a) SUM XA1 LDA ADD XA2 ADD XA3 STA XAS SUM BETA ADD STA XBETAS 24 Figure 4.6 Concatenation of macro parameters.

#### Generation of Unique Labels

- It is in general not possible for the body of a macro instruction to contain labels of the usual kind
  - Leading to the use of relative addressing at the source statement level
    - Only be acceptable for short jumps
- Solution:
  - Allowing the creation of special types of labels within macro instructions
  - See Figure 4.7

25	RDBUFF	MACRO	&INDEV, &BUF	ADR, &RECLTH
30		CLEAR	X	CLEAR LOOP COUNTER
35		CLEAR	A	
40		CLEAR	S	
45		+LDT	#4096	SET MAXIMUM RECORD LENGTH
50	\$LOOP	TD	=X'&INDEV'	TEST INPUT DEVICE
55		JEQ	\$LOOP	LOOP UNTIL READY
60		RD	=X'&INDEV'	READ CHARACTER INTO REG A
65		COMPR	A,S	TEST FOR END OF RECORD
70		JEQ	\$EXIT	EXIT LOOP IF EOR
75		STCH	&BUFADR, X	STORE CHARACTER IN BUFFER
80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH
85		JLT	\$LOOP	HAS BEEN REACHED
90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH
95		MEND		

	n supplied by make - Text of the land of supplied of the land of	RDBUFF	F1,BUFFER,LEN	JGTH
30 35 40		CLEAR CLEAR CLEAR	X A S	CLEAR LOOP COUNTER
45 50	\$AALOOP	+LDT TD	#4096 =X'F1'	SET MAXIMUM RECORD LENGTH TEST INPUT DEVICE
55 60 65		JEQ RD COMPR	\$AALOOP =X'F1' A,S	LOOP UNTIL READY READ CHARACTER INTO REG A TEST FOR END OF RECORD
70 75		JEQ STCH	\$AAEXIT BUFFER,X	EXIT LOOP IF EOR STORE CHARACTER IN BUFFER
80 85		TIXR JLT	T \$AALOOP	LOOP UNLESS MAXIMUM LENGTH HAS BEEN REACHED
90	\$AAEXIT	STX	LENGTH (b)	SAVE RECORD LENGTH
			(b)	

Figure 4.7 Generation of unique labels within macro expansion.

27

### Generation of Unique Labels

#### • Solution:

- Allowing the creation of special types of labels within macro instructions
- See Figure 4.7
  - Labels used within he macro body begin with the special character \$
- Programmers are instructed no to use \$ in their source programs

### Conditional Macro Expansion

- Most macro processors can
  - Modify the sequence of statements generated for a macro expansion,
  - depending on the arguments supplied in the macro invocation
- See Figure 4.8

25 **RDBUFF** &INDEV, &BUFADR, &RECLTH, &EOR, &MAXLTH 26 (&EOR NE '') 27 &EORCK SET 28 ENDIF 30 CLEAR CLEAR LOOP COUNTER 35 CLEAR 38 (&EORCK EQ 1) 40 LDCH =X'&EOR' SET EOR CHARACTER 42 RMO A,S 43 ENDIF 44 (&MAXLTH EQ '') 45 +LDT #4096 SET MAX LENGTH = 4096 46 ELSE 47 +LDT #&MAXLTH SET MAXIMUM RECORD LENGTH 48 ENDIF 50 \$LOOP TD =X'&INDEV' TEST INPUT DEVICE 55 JEQ \$LOOP LOOP UNTIL READY 60 RD =X'&INDEV' READ CHARACTER INTO REG A 63 IF (&EORCK EQ 1) 65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$EXIT EXIT LOOP IF EOR 73 ENDIF 75 STCH &BUFADR, X STORE CHARACTER IN BUFFER 80 TIXR LOOP UNLESS MAXIMUM LENGTH 85 JLT \$LOOP HAS BEEN REACHED 90 \$EXIT STX &RECLTH SAVE RECORD LENGTH 95 MENID

(a)

	•	RDBUFF	F3, BUF, RECL, 0	4,2048
30 35 40 42		CLEAR CLEAR LDCH RMO	X A =X'04' A,S	CLEAR LOOP COUNTER SET EOR CHARACTER
47 50 55 60 65	\$AALOOP	+LDT TD JEQ RD COMPR	#2048 =X'F3' \$AALOOP =X'F3' A,S	SET MAXIMUM RECORD LENGTH TEST INPUT DEVICE LOOP UNTIL READY READ CHARACTER INTO REG A TEST FOR END OF RECORD
70 75 80 85 90	\$AAEXIT	JEQ STCH TIXR JLT STX	\$AAEXIT BUF,X T \$AALOOP RECL	EXIT LOOP IF EOR STORE CHARACTER IN BUFFER LOOP UNLESS MAXIMUM LENGTH HAS BEEN REACHED SAVE RECORD LENGTH
			(b)	SAVE RECORD LENGTH

Figure 4.8 Use of macro-time conditional statements.

30			RDBUFF	OE, BUFFER, I	LENGTH,,80
CLEAR   A					
#80 SET MAXIMUM RECORD LENGTH  50 \$ABLOOP TD = X'OE' TEST INPUT DEVICE  55 JEQ \$ABLOOP LOOP UNTIL READY  60 RD = X'OE' READ CHARACTER INTO REG A  75 STCH BUFFER, X STORE CHARACTER IN BUFFER  80 TIXR T LOOP UNLESS MAXIMUM LENGTH  87 JLT \$ABLOOP HAS BEEN REACHED  90 \$ABEXIT STX LENGTH SAVE RECORD LENGTH   (c)  (c)  RDBUFF F1, BUFF, RLENG, 04  30 CLEAR X CLEAR LOOP COUNTER  40 LDCH = X'O4' SET EOR CHARACTER  42 RMO A,S  44 LDT #4096 SET MAX LENGTH = 4096  55 \$ACLOOP TD = X'F1' TEST INPUT DEVICE  55 JEQ \$ACLOOP LOOP UNTIL READY  66 RD = X'F1' READ CHARACTER INTO REG A  65 COMPR A,S TEST FOR END OF RECORD  66 PD STCH BUFF, X STORE CHARACTER IN BUFFER  70 JEQ \$ACEXIT EXIT LOOP IF EOR  71 STCH BUFF, X STORE CHARACTER IN BUFFER  80 TIXR T LOOP UNLESS MAXIMUM LENGTH  81 JLT \$ACLOOP HAS BEEN REACHED  90 \$ACEXIT STX RLENG SAVE RECORD LENGTH	30		CLEAR	X	CLEAR LOOP COUNTER
\$ \$ABLOOP TD	35			A	
STCH	47		+LDT	#80	SET MAXIMUM RECORD LENGTH
60 RD =X'OE' READ CHARACTER INTO REG A 75 STCH BUFFER, X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 87 JLT \$ABLOOP HAS BEEN REACHED 90 \$ABEXIT STX LENGTH SAVE RECORD LENGTH  (c)  (c)  . RDBUFF F1, BUFF, RLENG, 04  30 CLEAR X CLEAR LOOP COUNTER 35 CLEAR A 40 LDCH =X'O4' SET EOR CHARACTER 42 RMO A, S 45 +LDT #4096 SET MAX LENGTH = 4096 50 \$ACLOOP TD =X'F1' TEST INPUT DEVICE 55 JEQ \$ACLOOP LOOP UNTIL READY 60 RD =X'F1' READ CHARACTER INTO REG A 65 COMPR A, S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF, X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 \$ACEXIT STX RLENG SAVE RECORD LENGTH	50	\$ABLOOP	TD	=X'0E'	TEST INPUT DEVICE
TIXE T LOOP UNLESS MAXIMUM LENGTH  TIXE T LOOP UNLESS MAXIMUM LENGTH  TIXE T LOOP UNLESS MAXIMUM LENGTH  TO UNLESS MAXIMUM LENGTH  TO UNLESS MAXIMUM LENGTH  SAVE RECORD LENGTH  (c)  CLEAR X CLEAR LOOP COUNTER  CLEAR A  LDCH = X'04' SET EOR CHARACTER  RMO A,S  +LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE  JEQ \$ACLOOP LOOP UNTIL READY  RD = X'F1' READ CHARACTER INTO REG A  CMPR A,S TEST FOR END OF RECORD  JEQ \$ACEXIT EXIT LOOP IF EOR  TIXE T LOOP UNLESS MAXIMUM LENGTH  SACEXIT STX RLENG SAVE RECORD LENGTH  (d)	55		JEQ	\$ABLOOP	LOOP UNTIL READY
TIXR T LOOP UNLESS MAXIMUM LENGTH  JIJT \$ABLOOP HAS BEEN REACHED  SAVE RECORD LENGTH  (c)  RDBUFF F1, BUFF, RLENG, 04  CLEAR X CLEAR LOOP COUNTER  CLEAR A  LDCH = X'04' SET EOR CHARACTER  RMO A,S  LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE  JEQ \$ACLOOP LOOP UNTIL READY  RD = X'F1' READ CHARACTER INTO REG A  COMPR A,S TEST FOR END OF RECORD  JEQ \$ACEXIT EXIT LOOP IF EOR  STCH BUFF, X STORE CHARACTER IN BUFFER  TIXR T LOOP UNLESS MAXIMUM LENGTH  AS BEEN REACHED  SAVE RECORD LENGTH	60		RD	=X'0E'	READ CHARACTER INTO REG A
87 90 \$ABEXIT STX LENGTH SAVE RECORD LENGTH  (c)  . RDBUFF F1, BUFF, RLENG, 04  30 CLEAR X CLEAR LOOP COUNTER 35 CLEAR A 40 LDCH = X'04' SET EOR CHARACTER 42 RMO A, S 45 +LDT #4096 SET MAX LENGTH = 4096 50 \$ACLOOP TD = X'F1' TEST INPUT DEVICE 55 JEQ \$ACLOOP LOOP UNTIL READY 60 RD = X'F1' READ CHARACTER INTO REG A 65 COMPR A, S TEST FOR END OF RECORD 65 JEQ \$ACEXIT EXIT LOOP IF EOR 76 STCH BUFF, X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH	75		STCH	BUFFER, X	STORE CHARACTER IN BUFFER
(c)  RDBUFF F1, BUFF, RLENG, 04  CLEAR X CLEAR LOOP COUNTER  CLEAR A  LDCH = X'04' SET EOR CHARACTER  RMO A,S  +LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE  JEQ \$ACLOOP LOOP UNTIL READY  RD = X'F1' READ CHARACTER INTO REG A  COMPR A,S TEST FOR END OF RECORD  JEQ \$ACEXIT EXIT LOOP IF EOR  STCH BUFF, X STORE CHARACTER IN BUFFER  TIXR T LOOP UNLESS MAXIMUM LENGTH  SACEXIT STX RLENG SAVE RECORD LENGTH  (d)	80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH
(c)  RDBUFF F1, BUFF, RLENG, 04  CLEAR X CLEAR LOOP COUNTER  CLEAR A LDCH = X'04' SET EOR CHARACTER RMO A,S LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE JEQ \$ACLOOP LOOP UNTIL READY RD = X'F1' READ CHARACTER INTO REG A COMPR A,S TEST FOR END OF RECORD JEQ \$ACEXIT EXIT LOOP IF EOR STCH BUFF, X STORE CHARACTER IN BUFFER TIXR T LOOP UNLESS MAXIMUM LENGTH SACEXIT STX RLENG SAVE RECORD LENGTH  (d)	87		JLT	\$ABLOOP	HAS BEEN REACHED
CLEAR X CLEAR LOOP COUNTER  CLEAR A  CLEAR A  LDCH = X'04' SET EOR CHARACTER  RMO A,S  LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE  JEQ \$ACLOOP LOOP UNTIL READY  RD = X'F1' READ CHARACTER INTO REG A  COMPR A,S TEST FOR END OF RECORD  JEQ \$ACEXIT EXIT LOOP IF EOR  STCH BUFF,X STORE CHARACTER IN BUFFER  TIXR T LOOP UNLESS MAXIMUM LENGTH  SACEXIT STX RLENG SAVE RECORD LENGTH	90	\$ABEXIT	STX		
CLEAR X  CLEAR LOOP COUNTER  CLEAR A  CLEAR LOOP COUNTER  CLEAR A  CLEAR LOOP COUNTER  CLEAR LOOP COUNTER				(c)	
CLEAR X  CLEAR LOOP COUNTER  CLEAR A  CLEAR LOOP COUNTER  CLEAR A  CLEAR LOOP COUNTER  CLEAR LOOP COUNTER			DDDIEE	P1 DITTE DI	PINO OA
35			TODOLL	II, BOII, KE	
CLEAR A  LDCH = X'04' SET EOR CHARACTER  RMO A,S  +LDT #4096 SET MAX LENGTH = 4096  SACLOOP TD = X'F1' TEST INPUT DEVICE  JEQ \$ACLOOP LOOP UNTIL READY  RD = X'F1' READ CHARACTER INTO REG A  COMPR A,S TEST FOR END OF RECORD  JEQ \$ACEXIT EXIT LOOP IF EOR  STCH BUFF,X STORE CHARACTER IN BUFFER  STORE CHARACTER IN BUFFER  LOOP UNLESS MAXIMUM LENGTH  SACLOOP HAS BEEN REACHED  SAVE RECORD LENGTH	30		CLEAR	Х	CLEAR LOOP COUNTER
40					
42 RMO A,S 45 +LDT #4096 SET MAX LENGTH = 4096 50 \$ACLOOP TD =X'F1' TEST INPUT DEVICE 55 JEQ \$ACLOOP LOOP UNTIL READY 60 RD =X'F1' READ CHARACTER INTO REG A 65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH					SET EOR CHARACTER
45 +LDT #4096 SET MAX LENGTH = 4096 50 \$ACLOOP TD =X'F1' TEST INPUT DEVICE 55 JEQ \$ACLOOP LOOP UNTIL READY 60 RD =X'F1' READ CHARACTER INTO REG A 65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH					
\$ACLOOP TD =X'F1' TEST INPUT DEVICE  55 JEQ \$ACLOOP LOOP UNTIL READY  60 RD =X'F1' READ CHARACTER INTO REG A  65 COMPR A,S TEST FOR END OF RECORD  70 JEQ \$ACEXIT EXIT LOOP IF EOR  75 STCH BUFF,X STORE CHARACTER IN BUFFER  80 TIXR T LOOP UNLESS MAXIMUM LENGTH  85 JLT \$ACLOOP HAS BEEN REACHED  90 \$ACEXIT STX RLENG SAVE RECORD LENGTH					SET MAX LENGTH = 4096
55 JEQ \$ACLOOP LOOP UNTIL READY 60 RD =X'F1' READ CHARACTER INTO REG A 65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)		\$ACLOOP			
60 RD =X'F1' READ CHARACTER INTO REG A 65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)					
65 COMPR A,S TEST FOR END OF RECORD 70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)	60				
70 JEQ \$ACEXIT EXIT LOOP IF EOR 75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)			COMPR		
75 STCH BUFF,X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)	70		JEQ		
80 TIXR T LOOP UNLESS MAXIMUM LENGTH 85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)	75				
85 JLT \$ACLOOP HAS BEEN REACHED 90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)					
90 \$ACEXIT STX RLENG SAVE RECORD LENGTH  (d)	85			\$ACLOOP	
		\$ACEXIT			
				(d)	
Figure 4.8 (cont'd)				NEW TOWN	
		Figure 4.8	(cont'd)		

### Conditional Macro Expansion

- Most macro processors can modify the sequence of statements generated for a macro expansion, depending on the arguments supplied in the macro invocation
- See Figure 4.8
  - Macro processor directive
    - IF, ELSE, ENDIF
    - SET
  - Macro-time variable (set symbol)
- WHILE-ENDW
  - See Figure 4.9

33

10 Distriction of the second s	25 F	RDBUFF	MACRO		ADR, &RECL/TH, &EOR
35	27 8	EORCT	SET	%NITEMS(&EOF	
### ### #### #########################	30		CLEAR	X	CLEAR LOOP COUNTER
SLOOP	35		CLEAR	A	
SECOND   S	45		+LDT	#4096	SET MAX LENGTH = 4096
60 RD =X'&INDEV' READ CHARACTER INTO REG 63 &CTR SET 1 64 WHILE (&CTR LE &EORCT) 65 COMP =X'0000&EOR[&CTR]' 70 JEQ \$EXIT 71 &CTR SET &CTR+1 73 ENDW 75 STCH &BUFADR, X STORE CHARACTER IN BUFFER 80 TIXR T LOOP UNLESS MAXIMUM LENG 85 JLT \$LOOP HAS BEEN REACHED 90 \$EXIT STX &RECLITH SAVE RECORD LENGTH 100 (a)  (a)  (b)  (c)  (c)  (c)  (c)  (d)  (e)  (e)  (e)  (e)  (e)  (f)  (f)  (f	50 5	\$LOOP	TD	=X'&INDEV'	TEST INPUT DEVICE
63 &CTR SET 1 64 WHILE (&CTR LE &EORCT) 65 COMP =X'0000&EOR[&CTR]' 70 JEQ \$EXIT 71 &CTR SET &CTR+1 73 ENIW 75 STCH &BUFADR,X STORE CHARACTER IN BUFFE 80 TIXR T LOOP UNLESS MAXIMUM LENG 85 JUT \$LOOP HAS BEEN REACHED 90 \$EXIT STX &RECLTH SAVE RECORD LENGTH 100 MEND  (a)  (a)  CLEAR X CLEAR LOOP COUNTER 45 +LDT #4096 SET MAX LENGTH = 4096 50 \$AALOOP TD =X'F2' TEST INPUT DEVICE 55 JEQ \$AALOOP LOOP UNTIL READY 60 RD =X'F2' READ CHARACTER INTO REG 60 RD =X'F2' READ CHARACTER INTO REG 60 COMP =X'000000' 70 JEQ \$AAEXIT 65 COMP =X'000000' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LENGTH 81 LOOP UNLESS MAXIMUM LENGTH 82 STCH BUFFER,X STORE CHARACTER IN BUFF 83 JUT \$AALOOP HAS BEEN REACHED 84 SAVE RECORD LENGTH	55		JEQ	\$LOOP	
64 WHILE (&CTR LE &EORCT) 65 COMP = X'0000&EOR[&CTR]' 70 JEQ \$EXIT 71 &CTR SET &CTR+1 73 ENDW 75 STCH &BUFADR, X STORE CHARACTER IN BUFFE 80 TIXR T LOOP UNLESS MAXIMUM LENG 85 JLT \$LOOP HAS BEEN REACHED 90 \$EXIT STX &RECLITH SAVE RECORD LENGTH 100 MEND  (a)  RDBUFF F2, BUFFER, LENGTH, (00,03,04)  (b)  (c)  (c)  (c)  (c)  (d)  (e)  (e)  (D)  (e)  (e)  (e)  (e)  (e	60		RD	=X'&INDEV'	READ CHARACTER INTO REG A
COMP	63	&CTR	SET	1	
JEQ	64		WHILE	(&CTR LE &EC	ORCT)
71 &CTR SET &CTR+1  73 ENDW  75 STCH &BUFADR, X STORE CHARACTER IN BUFFE  80 TIXR T LOOP UNLESS MAXIMUM LENG  85 JLT \$LOOP HAS BEEN REACHED  90 \$EXIT STX &RECLITH SAVE RECORD LENGTH  100 RDBUFF F2, BUFFER, LENGTH, (00,03,04)   (a)  CLEAR X CLEAR LOOP COUNTER  45 CLEAR A  45 +LDT #4096 SET MAX LENGTH = 4096  50 \$AALOOP TD =X'F2' TEST INPUT DEVICE  55 JEQ \$AALOOP LOOP UNTIL READY  60 RD =X'F2' READ CHARACTER INTO REG  60 COMP =X'000000'  70 JEQ \$AAEXIT  65 COMP =X'000003'  70 JEQ \$AAEXIT  65 COMP =X'000004'  70 JEQ \$AAEXIT  65 COMP =X'000004'  76 JEQ \$AAEXIT  77 STCH BUFFER, X STORE CHARACTER IN BUFFER, X  80 TIXR T LOOP UNLESS MAXIMUM LENGTH  85 JLT \$AALOOP HAS BEEN REACHED  90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH	65		COMP	=X'0000&EOR[8	CTR]
73 75 75 76 77 77 78 77 78 77 78 77 78 77 78 77 78 78	70		JEQ	\$EXIT	
TIXE T LOOP UNLESS MAXIMUM LENGERS SEXIT STX &RECLITH SAVE RECORD LENGTH  (a)  (a)  (b)  (c)  (c)  (c)  (c)  (d)  (e)  (e)  (a)  (e)  (e)  (e)  (e)  (e	71	&CTR	SET	&CTR+1	
## 1	73		ENDW		
## SEXIT   SIX   \$LOOP	75		STCH	&BUFADR, X	STORE CHARACTER IN BUFFER
SEXIT   STX   & RECLITH   SAVE RECORD LENGTH	80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH
(a)  RDBUFF F2,BUFFER,LENGTH,(00,03,04)  CLEAR X CLEAR LOOP COUNTER  CLEAR A  LDT #4096 SET MAX LENGTH = 4096  SAALOOP TD =X'F2' TEST INPUT DEVICE  DEQ SAALOOP LOOP UNTIL READY  RD =X'F2' READ CHARACTER INTO REG  COMP =X'000000'  JEQ SAAEXIT  COMP =X'000003'  JEQ SAAEXIT  COMP =X'000004'  JEQ SAAEXIT  STCH BUFFER,X STORE CHARACTER IN BUFFER,X  TIXR T LOOP UNLESS MAXIMUM LEN  B5 JLT SAALOOP HAS BEEN REACHED  90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH	85		JLT	\$LOOP	HAS BEEN REACHED
RDBUFF F2, BUFFER, LENGTH, (00,03,04)  CLEAR X CLEAR LOOP COUNTER  CLEAR A  LIDT #4096 SET MAX LENGTH = 4096  SAALOOP TD =X'F2' TEST INPUT DEVICE  DEQ SAALOOP LOOP UNTIL READY  RD =X'F2' READ CHARACTER INTO REG  COMP =X'000000'  JEQ SAAEXIT  COMP =X'000003'  JEQ SAAEXIT  COMP =X'000004'  JEQ SAAEXIT  STCH BUFFER, X STORE CHARACTER IN BUFFER, X  TIXR T LOOP UNLESS MAXIMUM LEN  TO SAAEXIT STX LENGTH SAVE RECORD LENGTH	90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH
. RDBUFF F2,BUFFER,LENGTH,(00,03,04)  30 CLEAR X CLEAR LOOP COUNTER 35 CLEAR A 45 +LDT #4096 SET MAX LENGTH = 4096 50 \$AALOOP TD =X'F2' TEST INPUT DEVICE 55 JEQ \$AALOOP LOOP UNTIL READY 60 RD =X'F2' READ CHARACTER INTO REG 65 COMP =X'000000' 70 JEQ \$AAEXIT 65 COMP =X'000003' 70 JEQ \$AAEXIT 65 COMP =X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JIJT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH	100		MEND		
RDBUFF F2, BUFFER, LENGTH, (00,03,04)  CLEAR X CLEAR LOOP COUNTER  CLEAR A +LDT #4096 SET MAX LENGTH = 4096  SO \$AALOOP TD =X'F2' TEST INPUT DEVICE  DEQ \$AALOOP LOOP UNTIL READY  RD =X'F2' READ CHARACTER INTO REG  COMP =X'000000'  DEQ \$AAEXIT  COMP =X'000003'  DEQ \$AAEXIT  COMP =X'000004'  DEQ \$AAEXIT  STCH BUFFER, X STORE CHARACTER IN BUFF  STCH BUFFER, X STORE CHARACTER IN BUFF  STIXR T LOOP UNLESS MAXIMUM LEN  JIJT \$AALOOP HAS BEEN REACHED  SAVE RECORD LENGTH				(9)	
30				(a)	
35		,	RDBUFF	F2, BUFFER, L	ENGTH, (00,03,04)
35	20		CI END	v	CLEAR LOOP COUNTER
#45					CHEAR LOOP COUNTER
50 \$AALOOP TD =X'F2' TEST INPUT DEVICE 55 JEQ \$AALOOP LOOP UNTIL READY 60 RD =X'F2' READ CHARACTER INTO REG 65 COMP =X'000000' 70 JEQ \$AAEXIT 65 COMP =X'000003' 70 JEQ \$AAEXIT 65 COMP =X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JILT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH					CET MAY LENGTH - 4096
55		AXXX 000			
60 RD =X'F2' READ CHARACTER INTO REG 65 COMP =X'000000' 70 JEQ \$AAEXIT 65 COMP =X'000003' 70 JEQ \$AAEXIT 65 COMP =X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JIJT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH		SAALOOP			
65 COMP =X'000000' 70 JEQ \$AAEXIT 65 COMP =X'000003' 70 JEQ \$AAEXIT 65 COMP =X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JIJT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH					
70			7.07		READ CHARACTER INTO ICES A
65 COMP = X'000003' 70 JEQ \$AAEXIT 65 COMP = X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER, X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JLT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH					
70			-		
65 COMP =X'000004' 70 JEQ \$AAEXIT 75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JLT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH	100				
70					
75 STCH BUFFER,X STORE CHARACTER IN BUFF 80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JLT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH					
80 TIXR T LOOP UNLESS MAXIMUM LEN 85 JUT \$AALOOP HAS BEEN REACHED 90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH	33770			5 TO (1) TO AND TO STATE OF THE	CHOPE CHAPACHED IN DIEDED
90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH				The second secon	
90 \$AAEXIT STX LENGTH SAVE RECORD LENGTH					
NA ATTENDED OF THE PROPERTY OF		A > > =====			
(b)	90	ŞAAEXIT	STX	LENGTH	SAVE RECORD LENGTH
				(b)	
Figure 4.9 Use of macro-time looping statements.		igure 4 9	Use of m	acro-time looping	g statements.

#### **Keyword Macro Parameters**

#### Positional parameters

- Parameters and arguments were associated with each other according to their positions in the macro prototype and the macro invocation statement
- Consecutive commas is necessary for a null argument

GENER "DIRECT,",,3

35

### Keyword Macro Parameters

#### Keyword parameters

- Each argument value is written with a keyword that names the corresponding parameter
- A macro may have a large number of parameters, and only a few of these are given values in a typical invocation

**GENER TYPE=DIRECT, CHANNEL=3** 

25	RDBUFF	MACRO	CINDENT-E1 CDIT	PADD— CDECLEUI CEOD OA CMANTEUI AOOC
26	KDBUFF			FADR=, &RECLTH=, &EOR=04, &MAXLTH=4096
27	CEODOR	IF	(&EOR NE '')	
28	&EORCK	SET	1	
30		ENDIF	77	CLEAR LOOP COLUMN
35		CLEAR	X	CLEAR LOOP COUNTER
		CLEAR	A	
38		IF	(&EORCK EQ 1)	
40		LDCH		SET EOR CHARACTER
42		RMO	A,S	
43		ENDIF	U. 0.3.5.7.1.7. (77.7.)	
47	<b>41.00</b> D	+LDT	#&MAXLTH	SET MAXIMUM RECORD LENGTH
50	\$LOOP	TD	=X'&INDEV'	TEST INPUT DEVICE
55		JEQ	\$LOOP	LOOP UNTIL READY
60		RD	=X'&INDEV'	READ CHARACTER INTO REG A
63		IF	(&EORCK EQ 1)	
65		COMPR	A,S	TEST FOR END OF RECORD
70		JEQ	\$EXIT	EXIT LOOP IF EOR
73		ENDIF		
75		STCH	&BUFADR,X	STORE CHARACTER IN BUFFER
80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH
85	oresea	JLT	\$LOOP	HAS BEEN REACHED
90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH
95		MEND		
			(a	

n si		RDBUFF	RECLTH=LENGTH	I, BUFADR=BUFFER, EOR=, INDEV=F3
30		CLEAR	X	CLEAR LOOP COUNTER
35		CLEAR	A	
47		+LDT	#4096	SET MAXIMUM RECORD LENGTH
50	\$ABLOOP	TD	=X'F3'	TEST INPUT DEVICE
55		JEQ	\$ABLOOP	LOOP UNTIL READY
60		RD	=X'F3'	READ CHARACTER INTO REG A
75		STCH	BUFFER, X	STORE CHARACTER IN BUFFER
80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH
85		JLT	\$ABLOOP	HAS BEEN REACHED
90	\$ABEXIT	STX	LENGTH	SAVE RECORD LENGTH
			(c)	

Figure 4.10 (cont'd)

### Macro Processor Design Options

- Recursive Macro Expansion
  - In Figure 4.3, we presented an example of the definition of on macro instruction by another.
    - We have not dealt with the invocation of one macro by another (nested macro invocation)
  - See Figure 4.11

39

10 RDBUFF MACRO &BUFADR, &RECLTH, &INDEV								
15								
20		MACRO 1	TO READ RECORD	INTO BUFFER				
25								
30		CLEAR	X	CLEAR LOOP COUNTER				
35		CLEAR	A					
40		CLEAR	S					
45		+LDT	#4096	SET MAXIMUM RECORD LENGTH				
50	\$LOOP	RDCHAR	&INDEV	READ CHARACTER INTO REG A				
65		COMPR	A,S	TEST FOR END OF RECORD				
70		JEQ	\$EXIT	EXIT LOOP IF EOR				
75		STCH	&BUFADR,X	STORE CHARACTER IN BUFFER				
80		TIXR	T	LOOP UNLESS MAXIMUM LENGTH				
85		JLT	\$LOOP	HAS BEEN REACHED				
90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH				
95		MEND						
			and definitions and but					
			(a)					
5	RDCHAR	MACRO	&IN					
10	-							
15		MACRO T	O READ CHARAC	TER INTO REGISTER A				
20				ATTENDANCE OF THE PARTY OF THE				
25		TD	=X'&IN'	TEST INPUT DEVICE				
30		JEQ	*-3	LOOP UNTIL READY				
35		RD	=X'&IN'	READ CHARACTER				
40		MEND						
A.								
(b)								
	RDBUFF BUFFER, LENGTH, F1							
	(c)							
	the smile service and the second service and the service service service and the service servi							
	Figure 4.11 Example of nested macro invocation.							
The state of the s								

#### Macro Processor Design Options

 Recursive Macro Expansion Applying Algorithm of Fig. 4.5

#### • Problem:

- The processing would proceed normally until line 50,
   which contains a statement invoking RDCHAR
- In addition, the argument from the original macro invocation (RDBUFF) would be lost because the values in ARGTAB were overwritten with the arguments from the invocation of RDCHAR

#### • Solution:

 These problems are not difficult to solve if the macro processor is being written in a programming language that allows recursive call

41

## General-Purpose Macro Processors

- Macro processors have been developed for some high-level programming languages
- These special-purpose macro processors are similar in general function and approach; however, the details differ from language to language

## General-Purpose Macro Processors

- The advantages of such a general-purpose approach to macro processing are obvious
  - The programmer does not need to learn about a different macro facility for each compiler or assembler language, so much of the time and expense involved in training are eliminated
  - A substantial overall saving in software development cost

43

# General-Purpose Macro Processors

- In spite of the advantages noted, there are still relatively few general-purpose macro processors.
  - 1. In a typical programming language, there are several situations in which normal macro parameter substitution should not occur
    - E.g. comments should usually be ignored by a macro processor

## General-Purpose Macro Processors

- 2. Another difference between programming languages is related to their facilities for grouping together terms, expressions, or statements
  - E.g. Some languages use keywords such as begin and end for grouping statements. Others use special characters such as { and }.

45

# General-Purpose Macro Processors

- 3. A more general problem involves the tokens of the programming language
  - E.g. identifiers, constants, operators, and keywords
  - E.g. blanks

## General-Purpose Macro Processors

4. Another potential problem with generalpurpose macro processors involves the syntax
used for macro definitions and macro
invocation statements. With most specialpurpose macro processors, macro invocations
are very similar in form to statements in the
source programming language

All these Macros are called Hygienic macro.

- Removes problem of "Accidental Capture of Identifiers" (LISP)

47

# The end.