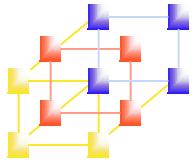


Chapter 4 Macro Processors

-- Machine-independent Macro Processor Features

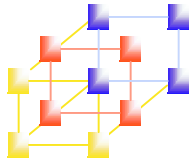


Concatenation of macro parameters

- Concatenate parameters with other character strings
 - Used when a program contains a set of series of variables
 - Ex:

TOTAL	MACRO	&ID			
	LAD	X&ID1	TOTAL	A	
	ADD	X&ID2		→	
	STA	X&ID3		{	
	MEND			LAD	XA1
				ADD	XA2
				STA	XA3

&: starting symbol of macro parameter



Concatenation of macro parameters

- Ambiguity problem

- If &ID and &ID1 are parameters

- X&ID1 may mean

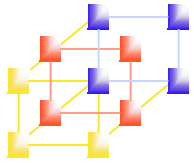
"X" + &ID + "1"

"X" + &ID1

- Solution to this ambiguity problem

- Use a special concatenation operator "->" to specify the end of the parameter

```
ID123  MACRO  &ID
        LAD   X&ID→1
        ADD  X&ID→2
        STA  X&ID→3
        MEND
```



Concatenation of macro parameters

1	SUM MACRO	&ID
2	LDA	X&ID→ 1
3	ADD	X&ID→ 2
4	ADD	X&ID→ 3
5	STA	X&ID→ S
6	MEND	

SUM

A



LDA

XA1

ADD

XA2

ADD

XA3

STA

XAS

SUM

BETA



LDA

XBEATA1

ADD

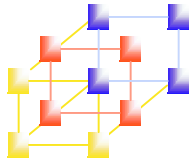
XBEATA2

ADD

XBEATA3

STA

XBEATAS



Generation of Unique Labels

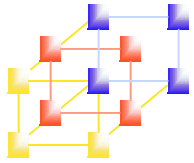
- Labels in the macro body may cause “duplicate labels” problem if the macro is invoked and expanded multiple times.

■ Ex:

CLOOP	MACRO
	:
XX
	:
	JLT XX
	:
	MEND

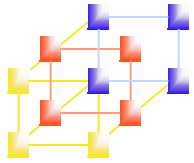
The label “XX” will be defined more than once

- Use of relative addressing at the source statement level is very inconvenient, error-prone, and difficult to read.



Generation of Unique Labels (2)

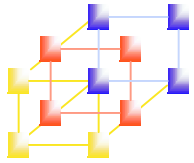
- Let the macro processor generate unique labels for each macro invocation and expansion.
 - During macro expansion, the \$ will be replaced with \$xx, where xx is a two-character alphanumeric counter of the number of macro instructions expanded.
 - xx=AA,AB,AC,.....
 - This allows 1296 macro expansions in a single program.



Generation of Unique Labels

Figure 4.7(a), pp.190

```
25      RDBUFF  MACRO  &INDEV, &BUFADR, &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   $LOOP      HAS BEEN REACHED
90      $EXIT   STX    &RECLTH    SAVE RECORD LENGTH
              MEND
```

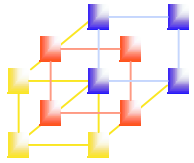


Generation of Unique Labels

Figure 4.7(b), pp.190

```
.          RDBUFF  F1, BUFFER, LENGTH

30          CLEAR  X          CLEAR LOOP COUNTER
35          CLEAR  A
40          CLEAR  S
45          +LDT   #4096      SET MAXIMUM RECORD LENGTH
50          $AALoop TD    =X'F1'  TEST INPUT DEVICE
55          JEQ    $AALoop      LOOP UNTIL READY
60          RD     =X'F1'      READ CHARACTER INTO REG A
65          COMPR A, S        TEST FOR END OF RECORD
70          JEQ    $AAEXIT      EXIT LOOP IF EOR
75          STCH  BUFFER, X    STORE CHARACTER IN BUFFER
80          TIXR  T           LOOP UNLESS MAXIMUM LENGTH
85          JLT   $AALoop     HAS BEEN REACHED
90          $AAEXIT STX      LENGTH  SAVE RECORD LENGTH
```

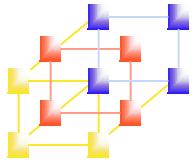



Conditional Macro Expansion

- Conditional assembly depends on parameters provides

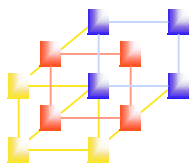
```
MACRO      &COND
.....
IF (&COND NE ' ')
    part I
ELSE
    part II
ENDIF
.....
ENDM
```

- Part I is expanded if condition part is true, otherwise part II is expanded
- Compare operator: NE, EQ, LE, GT



Macro-time variables

- Begins with “&” but is not a macro instruction parameter
- Can be used to store working values during the macro expansion
 - Store the evaluation result of Boolean expression
 - Control the macro-time conditional structures
- Be initialized to a value of 0
- Be set by a macro processor directive, SET
 - Ex: &EORCK SET 1
 &EORCTR SET &EORCTR + 1

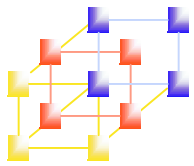


Use of macro-time conditional statements

Figure 4.8(a), pp. 191

25	RDBUFF	MACRO	&INDEV, &BUFADR, &RECLTH, <u>&EOR, &MAXLTH</u>	
26		IF	(<u>&EOR</u> NE '')	
27	<u>&EORCK</u>	SET	1	
28		ENDIF		
30		CLEAR	X	CLEAR LOOP COUNTER
35		CLEAR	A	
38		IF	(<u>&EORCK</u> EQ 1)	
40		LDCH	=X'&EOR'	SET EOR COUNTER
42		RMO	A, S	
43		ENDIF		
44		IF	(<u>&MAXLTH</u> 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	(<u>&EORCK</u> 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		JLT	\$LOOP	HAS BEEN REACHED
90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH
95		MEND		

Macro-time variable

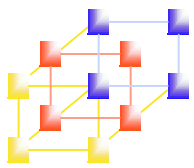


Use of macro-time conditional statements

Figure 4.8(b), pp. 191

```
.          RDBUFF  F31 BUF, RECL, 04, 2048

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

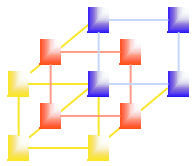


Use of macro-time conditional statements

Figure 4.8(c), pp. 192

```
.          RDBUFF  OE, BUFFER, LENGTH, , 80

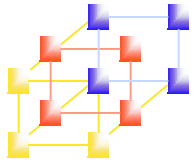
30          CLEAR  X          CLEAR LOOP COUNTER
35          CLEAR  A
47          +LDT   #80        SET MAXIMUM RECORD LENGTH
50  $ABLOOP TD    =X'0E'     TEST INPUT DEVICE
55          JEQ   $ABLOOP     LOOP UNTIL READY
60          RD    =X'0E'     READ CHARACTER IN REG A
75          STCH  BUFFER, X   STORE CHARACTER IN BUFFER
80          TIXR  T          LOOP UNLESS MAXIMUM LENGTH
87          JLT   $ABLOOP     HAS BEEN REACHED
90  $ABEXIT STX   LENGTH     SAVE RECORD LENGTH
```



Use of macro-time conditional statements

Figure 4.8(d), pp. 192

		<u>RDBUFF</u>	<u>F1. BUFF, ELENG, 04</u>	
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
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 LOOP REACHED
90	\$ACEXIT	STX	RLENG	SAVE RECORD LENGTH



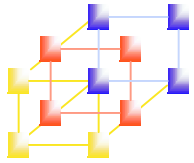
Macro-time looping statement

```
WHILE ( cond )
```

```
.....
```

```
ENDW
```

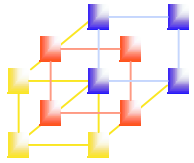
- Macro processor function
 - %NITEMS: The number of members in an argument list
- The execution of testing of IF/WHILE, SET, %NITEMS() occurs at macro expansion time



Use of macro-time looping statements

Figure 4.9(a), pp. 195

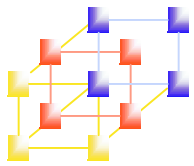
```
25      RDBUFF  MACRO   &INDEV, &BUFADR, &RECLTH, &EOR
27      &EORCT  SET     %NITEMS (&EOR) ← Macro processor function
30              CLEAR  X          CLEAR LOOP COUNTER
35              CLEAR  A
45              +LDT   #4096          SET MAX LENGTH = 4096
50      $LOOP  TD     =X'&INDEV'     TEST INPUT DEVICE
55              JEQ   $LOOP          LOOP UNTIL READY
60              RD    =X'&INDEV'     READ CHARACTER INTO REG A
63      &CTR   SET     1
64              WHILE (&CTR LE &EORCT)
65      &CTR   COMPR  =X'0000&EOR[&CTR]' ← List index
70              JEQ   $EXIT
71      &CTR   SET     &CTR+1
73              ENDW
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 RECORTD LENGTH
100             MEND
```

Use of macro-time looping statements

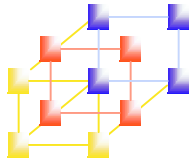
Figure 4.9(b), pp. 195

	RDBUFF	F2, BUFFER, LENGTH, (00, 03, 04)	
			List
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 A
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 BUFFER
80	TIXR	T	LOOP UNLESS MAXIMUM LENGTH
85	JLT	\$AALoop	HAS BEEN REACHED
90	\$AAEXIT STX	LENGTH	SAVE RECORD LENGTH



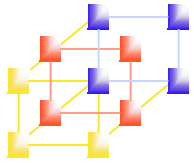
Implementation of conditional macro expansion

- **IF-ELSE-ENDIF structure**
 - The macro processor must maintain a symbol table
 - This table contains the values of all macro-time variables used.
 - Entries in this table are made or modified when SET statements are processed.
 - This table is used to look up the current value of a macro-time variable whenever it is required.
 - When an IF statement is encountered during the expansion of a macro, the specified Boolean expression is evaluated.
 - TRUE
 - The macro processor continues to process lines from DEFTAB until it encounters the next ELSE or ENDIF statement.
 - If ELSE is encountered, then skips to ENDIF
 - FALSE
 - The macro processor skips ahead in DEFTAB until it finds the next ELSE or ENDIF statement.



Implementation of conditional macro expansion

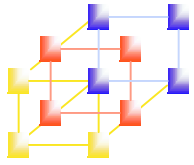
- WHILE-ENDW structure
 - When an WHILE statement is encountered during the expansion of a macro, the specified Boolean expression is evaluated.
 - TRUE
 - The macro processor continues to process lines from DEFTAB until it encounters the next ENDW statement.
 - When ENDW is encountered, the macro processor returns to the preceding WHILE, re-evaluates the Boolean expression, and takes action based on the new value.
 - FALSE
 - The macro processor skips ahead in DEFTAB until it finds the next ENDW statement and then resumes normal macro expansion.



Keyword macro parameters

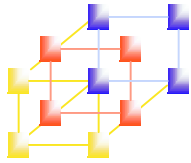
■ Positional parameters

- Parameters and arguments are associated according to their positions in the macro prototype and invocation.
- If an argument is to be omitted, a null argument should be used to maintain the proper order in macro invocation:
- Ex: `XXX MACRO &P1, &P2,, &P20,`
`XXX A1, A2,,,,,,,,,,,,,A20,.....` Null arguments
- It is not suitable if a macro has a large number of parameters, and only a few of these are given values in a typical invocation.



Keyword macro parameters

- Keyword parameters
 - Each argument value is written with a keyword that names the corresponding parameter.
 - Arguments may appear in any order.
 - Null arguments no longer need to be used.
 - Ex: `XXX P1=A1, P2=A2, P20=A20.`
 - It is easier to read and much less error-prone than the positional method.

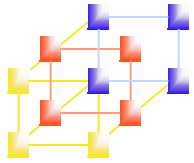


Use of keyword parameters in macro

Figure 4.10(a), pp.198

25	RDBUFF	MACRO	<u>&INDEV=F1, &BUFADR=, &RECLTH=, &EOR=04, &MAXLTH=4096</u>	
26		IF	(&EOR NE '')	
27	&EORCK	SET	1	
28		ENDIF		
30		CLEAR	X	CLEAR LOOP COUNTER
35		CLEAR	A	
38		IF	(&EORCK EQ 1)	
40		LDCH	=X'&EOR'	SET EOR CHARACTER
42		RMO	A, S	
43		ENDIF		
47		+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		JLT	\$LOOP	HAS BEEN REACHED
90	\$EXIT	STX	&RECLTH	SAVE RECORD LENGTH
95		MEND		

Parameters with default value

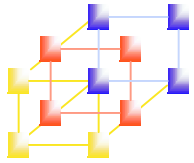


Use of keyword parameters in macro

Figure 4.10(b), pp.198

```
.          RDBUFF  BUFADR=BUFFER, RECLTH=LENGTH

30          CLEAR  X          CLEAR LOOP COUNTER
35          CLEAR  A
40          LDCH   =X'04'      SET EOR CHARACTER
42          RMO    A, S
47          +LDT   #4096      SET MAXIMUM RECORD LENGTH
50          $AALoop TD   =X'F1' TEST INPUT DEVICE
55          JEQ    $AALoop   LOOP UNTIL READY
60          RD     =X'F1'      READ CHARACTER INTO REG A
65          COMPR  A, S       TEST FOR END OF RECORD
70          JEQ    $AAEXIT    EXIT LOOP IF EOR
75          STCH   BUFFER, X   STORE CHARACTER IN BUFFER
80          TIXR   T          LOOP UNLESS MAXIMUM LENGTH
85          JLT    $AALoop   HAS BEEN REACHED
90          $AAEXUT STX      LENGTH SAVE RECORD LENGTH
```



Use of keyword parameters in macro

Figure 4.10(c), pp.199

```
1      .      RDBUFF RECLTH=LENGTH, 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
```