Lecture 2

Nested macro calls

- Model statement in a macro may constitute a call on another macro – **Nested macro calls**
- **Macro containing the nested call is the outer macro, and the called macro is inner macro**
- Expansion of nested macro calls follows **LIFO** rule
- In a structure of nested macro calls, **expansion of the latest macro call is completed first**

Advanced macro facilities

- Aimed at supporting semantic expansion

- Facilities are grouped into
  - Facilities for alteration of flow control during expansion
  - Expansion time variables
  - Attributes of parameters

Alteration of flow control during expansion – 1/5

- Two features
  - Expansion time sequencing symbols
  - Expansion time statements AIF, AGO and ANOP

Alteration of flow control during expansion – 2/5

. <ordinary string>

- **Sequencing symbol (SS)** is defined by putting it in the **label field** of a statement in the macro body
- **Used as an operand in an AIF or AGO** statement designating the **destination** of an expansion time control transfer
- It **never appears in the expanded form** of a model statement

Alteration of flow control during expansion – 3/5

AIF (<expression>) <sequencing symbol>

- <expression> is a *relational expression* involving ordinary strings, formal parameters and their attributes, and expansion time variables

- If relational expression evaluates to true, expansion time control is transferred to the statement containing <sequencing symbol> in its label field

Alteration of flow control during expansion – 4/5

AGO <sequencing symbol>

- It **unconditionally transfers expansion time control** to the statement containing <sequencing symbol>

Alteration of flow control during expansion – 5/5

<sequencing symbol> ANOP

- It simply has the effect of **defining the sequencing symbol**

Usage of SS and AIF statements

```
MACRO
DCL_CONST        &A=12
AIF   (L'&A EQ 1)          .NEXT
.NEXT          ....
          ...
MEND
```

Macro call:

```
          DCL_CONST
```

Usage of SS, AGO and AIF statements

```
        MACRO
        EVAL  &X,&Y,&Z
        AIF    (&Y EQ &X) .ONLY
        MOVER       AREG, &X
        SUB    AREG, &Y
.ONLY       ADD   AREG, &Z
        AGO   .OVER
        MOVER       AREG, &Z
.OVER       MEND
```

- Macro call

EVAL A, B, 5

Expansion time variables – 1/3

- **Expansion time variables** (EV's) are variables which can **only be used during the expansion of macro calls**

- **Local EV** is created for use only during a particular macro call

- **Global EV** exists across all macro calls situated in a program and can be used in any macro which has a declaration for it

Expansion time variables – 2/3

- Local and global EV are created through declaration statements with the following syntax

  LCL <EV specification>],<EV specification> ..]

  GBL <EV specification>],<EV specification> ..]

- <EV specification> has the syntax

  &<EV name>

- <EV name> is ordinary string
- Values of EV's can be manipulated through SET statement

Expansion time variables – 3/3

- **SET** statement syntax is

  <EV specification> SET <SET-expression>

- <EV specification> appears in label field and SET in mnemonic field
- SET statement assigns the value of <SET-expression> to the EV specified in <EV specification>
- Value of EV can be used in any field of a model statement, and in expression of AIF statement

Attributes of formal parameters

- Attribute is written using the syntax

  <attribute name>'<formal parameter spec>

- It represents info about the value of formal parameter, i.e. about the corresponding actual parameter
- Attributes and their names:

  Type – T **Length – L** **Size – S**

Expansion time loops

```
            MACRO
            CLEAR       &X, &N
            LCL   &M
&M          SET   0
            MOVER       AREG, ='0'
.MORE       MOVEM       AREG, &X+&M
&M          SET   &M+1
            AIF   (&M NE &N) .MORE
            MEND
```

Similar to

```
MACRO
CLEAR
MOVER    AREG, ='0'
MOVEM    AREG, &A
MOVEM    AREG, &A+1
MOVEM    AREG, &A+2
MEND
```

Macro call :

CLEAR B,3

Leads to expansion

```
+       MOVER     AREG, ='0'
+       MOVEM     AREG, B
+       MOVEM     AREG, B+1
+       MOVEM     AREG, B+2
```

Conditional expansion

- **Model statement is visited only under specific conditions** during the expansion of a macro

- The **AIF and AGO** statements are used for this purpose

Expansion time loops

- **Generate many similar statements** during the expansion of a macro
- Achieved by **writing similar model statements** in the macro
- The same effect can be achieved by **writing an expansion time loop**

- Expansion time loops can be written using expansion time variables (EV's) and expansion time control transfer statements AIF and AGO

Other facilities for expansion time loops

- The **REPT statement**

  REPT <expression>

  - <expression> should **evaluate to a numerical value during macro expansion**
  - Statements between **REPT** and an **ENDM** statement would be processed for expansion <expression> number of times

```
        MACRO
        CONST10
        LCL   &M
&M      SET   1
        REPT  10
        DC    '&M'
&M      SET   &M+1
        ENDM
        MEND
```

This declare 10 constants:

1,2,..10

Other facilities for expansion time loops (cont..)

- The IRP stmt

  IRP <formal parameter>, <argument list>

  - Formal parameter in the statement takes successive values from the argument list

  - For each value, statements between **IRP** and **ENDM** statements are expanded once

```
MACRO
CONSTS    &M,&N,&Z
IRP   &Z,&M,7,&N
DC    '&Z'
ENDM
MEND
```

Macro Call:

CONSTS 4, 10

Gives The Values 4,7,10

Semantic expansion

- It is the generation of instructions tailored to requirements of a specific usage

- Can be achieved by a combination of advanced macro facilities like AIF, AGO statements and expansion time variables