Macro Processor - Basics1

Basics -1 TOPICS

Macro Definition and Call, Macro Expansion, Macro Parameters, Intro. to Macro Pass 1 and Pass 2, MNT, MDT

Macro Processor Basics 1

TOPICS

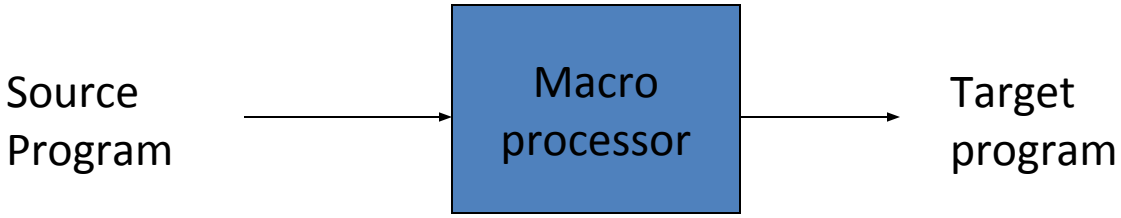
- Macro Definition and Call,
- Macro Expansion,
- Macro Parameters,
- Intro. to Macro Pass I and Pass 2, MNT, MDT

What is a Macro Processor? (general)

- Program that copies a **stream of text** from one place to another, making a **systematic set of replacements** as it does so.
- Macro processors are often embedded in (are a part of) other programs, such as assemblers and compilers

Introduction

- Macro : Abbreviation for a group of instructions
- Macro processor : Acts as a preprocessor to assembler
- Source prog : Assembly Lang. program with macros
- Target prog : Assembly Lang. program without macros



Macro Structure

MACRO <Macro Name> < Parameters> *HEADER*

<Set of Assembly Statements > *BODY*

MEND *FOOTER*

Macro Definition and Call

```
(Macro Definition)
MACRO INCR
    MOVER AREG, X
    ADD AREG,='1'
    MOVEM AREG,X
MEND
.................
(Macro Call (Invocation)
INCR
```

Sample Input: **Assembly Program** with Macro Definition and Call

```
MACRO INCR
    MOVER AREG, X
    ADD AREG,='1'
    MOVEM AREG,X
MEND
    START   100
    READ    X

    INCR

    PRINT   X
    STOP
X   DS      1

    END
```

Macro Expansion by Macro processor (Input- Output Sample)

```
MACRO INCR
    MOVER AREG, X
    ADD AREG,='1'
    MOVEM AREG,X
MEND
    START   100
    READ    X
    INCR
    PRINT   X
    STOP
X   DS      1
    END
```

```
    START   100
    READ    X
    MOVER AREG, X
    ADD AREG,='1'
    MOVEM AREG,X
    PRINT   X
    STOP
X   DS      1
    END
```

Call ***INCR*** *expanded*

Macro Definition and Call (with Parameters)

```
(Macro with parameters)
MACRO INCR  &R1, &V1
    MOVER &R1, &V1
    ADD &R1,='1'
    MOVEM &R1,&V1
MEND
.................
(Macro Call (Invocation)
INCR BREG, Y
.............................
(Formal Parameters R1, V1)
(Actual Parameters BREG, Y)
```

Sample Input: Assembly Program with Macro Definition and Call

```
MACRO INCR1 &V1, &INCR_BY, &R1
    MOVER &R1, &V1
    ADD &R1, &INCR_BY
    MOVEM &R1,&V1
MEND
    START   100
    READ    X
    READ    Y
    INCR1   X,Y,AREG
    PRINT   X
    STOP
X   DS      1
Y   DS      1
    END
```

Macro Expansion by Macro processor (Input- Output Sample)

```
MACRO
    INCR1 &V1, &INCR_BY, &R1
    MOVER &R1, &V1
    ADD &R1, &INCR_BY
    MOVEM &R1,&V1
MEND
    START   100
    READ    X
    READ    Y

    INCR1   X,Y,BREG

    PRINT   X
    STOP
X   DS      1
Y   DS      1
    END
```

```
    START   100
    READ    X
    READ    Y

    MOVER  BREG, X
    ADD BREG, Y
    MOVEM  BREG,X


    PRINT   X
    STOP
X   DS      1
Y   DS      1
    END
```

Call INCR1 X,Y,BREG expanded

Macro Parameters

- **Positional Parameters**
  - Written as &Parameter_Name
  - E.g. &V1, &INCR_BY, &R1
    - V1,INCR_BY, R1 are **positional prameters**
  - Actual values are **substituted** on the **basis of their position** in Macro-call Statement
- **Keyword Parameters**
  - Used to assign default values to parameters
  - Specified by name in macro call
  - Form: <parameter name> = <parameter value>

- **MIXED** : Macro can be defined with both positional and Keyword parameter

Sample Input: Assembly Program with Keyword parameters

```
MACRO INCR2 &V1=X, &INCR_BY=Y, &R1=AREG
    MOVER &R1, &V1
    ADD &R1, &INCR_BY
    MOVEM &R1,&V1
MEND
-----
INCR2 INCR_BY=B,R1= CREG , V1=A
```

```
MOVER   CREG,A
ADD     CREG,B
MOVEM   CREG,A
```

Macro Expansion by Macro processor (Input- Output Sample)

```
MACRO
    INCR2 &V1, &INCR_BY, &R1
    MOVER &R1, &V1
    ADD &R1, &INCR_BY
    MOVEM &R1,&V1
MEND
    START   100
    READ    X
    READ    Y

    INCR2   X,Y,BREG

    PRINT   X
    STOP
X   DS      1
Y   DS      1
    END
```

```
    START   100
    READ    X
    READ    Y

    MOVER  BREG, X
    ADD BREG, Y
    MOVEM  BREG,X


    PRINT   X
    STOP
X   DS      1
Y   DS      1
    END
```

Call INCR2 X,Y,BREG expanded

More on Macro PARAMETERS (Not restricted to Operands)

- Positional Parameters
- Keyword Parameters
- Mixed Parameters (e.g. in next slide)
- **Other uses of Parameters**
  - Formal parameters are **not restricted to OPERANDS**
    - Formal parameters can appear in **any fields** of a statement in macro
    - **Label – Opcode – Operand** (e.g. in next-to-next slide)

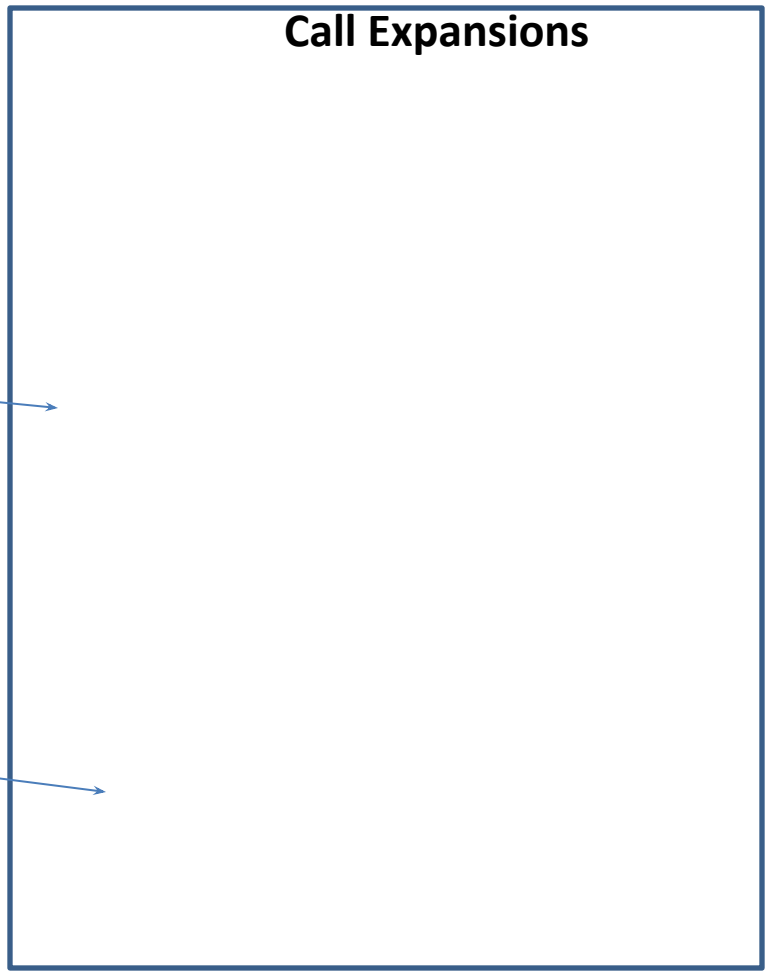
Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
    &LABEL1   MOVER       BREG, &X
          &OP1        BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N,SUB ,L2
```



Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
     &LABEL1   MOVER       BREG, &X
           &OP1         BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N,SUB ,L2
```

Call Expansions

```
L1    MOVER    BREG, A
      ADD          BREG,B
      MOVEM     BREG,A
```

```
L2    MOVER    BREG, M
      SUB          BREG,N
      MOVEM     BREG,M
```

Default value & keyword parameters in macros

Program with macro	Expanded code
MACRO **XYZ &a, &b = 10, &c** **LOAD &a** **ADD &b** **STORE &c** **MEND** **XYZ 4, ,7** **XYZ 3, 8, 4** **END**	

Default value & keyword parameters in macros

Program with macro	Expanded code
MACRO **XYZ &a, &b = 10, &c** **LOAD &a** **ADD &b** **STORE &c** **MEND** **XYZ 4, ,7** **XYZ 3, 8, 4** **END**	**LOAD 4** **ADD 10** **STORE 7** **LOAD 3** **ADD 8** **STORE 4** **END**

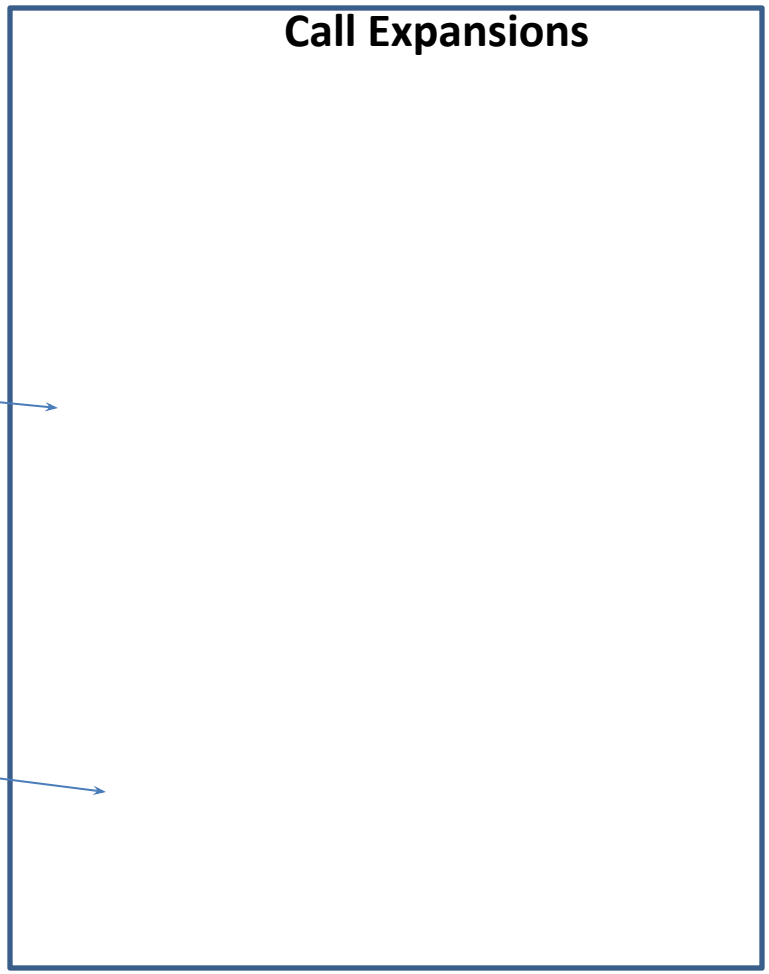
Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
    &LABEL1   MOVER       BREG, &X
          &OP1        BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N,SUB ,L2
```



Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
     &LABEL1   MOVER       BREG, &X
           &OP1         BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N,SUB ,L2
```

Call Expansions

```
L1    MOVER    BREG, A
      ADD          BREG,B
      MOVEM     BREG,A
```

```
L2    MOVER    BREG, M
      SUB          BREG,N
      MOVEM     BREG,M
```

Formal v/s positional parameter list and Actual v/s positional parameter list

Formal parameter	Positional parameter
arg1	#1
arg2	#2
arg3	#3

Actual parameter	Positional parameter
SUB	#1
X	#2
Y	#3

Formal-Actual Parameters

DS-1 MNT (Macro Name Table)

MNAME 0

…..INPUT---ALP WITH MACROS
MACRO MNAME &A,&B,&C
MOVER &A,&B
ADD &C,=‘20’
MOVEM X,&A
MEND

…..
MNAME AREG,M,BREG

MOVER AREG,M
ADD BREG,=‘20’
MOVEM X,AREG

MNAME CREG,Z,AREG

MOVER CREG,Z
ADD AREG,=‘20’
MOVEM X,CREG

Positional parameters-A,B,C
Keyword parameters

# 1	&A
# 2	&B
# 3	&C

# 1	AREG
# 2	M
# 3	BREG

MOVER #1,#2
ADD #3,=‘20’
MOVEM X,#1

# 1	CREG
# 2	Z
# 3	AREG

(DS-2) MDT (Macro Definition Table)

MOVER #1,#2
ADD #3,=‘20’
MOVEM X,#1
MEND

Formal v/s positional parameter list

Actual v/s positional parameter list

Formal parameter	Positional parameter
arg1	#1
arg2	#2
arg3	#3

Actual parameter	Positional parameter
SUB	#1
X	#2
Y	#3

Data structures for Expansion of Macros

- Macro Name Table (MNT)
- Macro Definition Table (MDT)
- ALA – Argument List Array

MNT and MDT

Name of macro	No. of parameters	Starting Index (row)	End Index (row)
SAMPLE1	0	1	2
SAMPLE2	0	3	4
SAMPLE3	0	5	6

1	**LOAD A**
2	**ADD B**
3	**LOAD X**
4	**SUB Y**
5	**LOAD P**
6	**DIV Q**

MNT and MDT

Name of macro	No. of parameters	Starting Index (row)	End Index (row)
SAMPLE1	0	1	2
SAMPLE2	0	3	4
SAMPLE3	0	5	6

1	**LOAD A**
2	**ADD B**
3	**LOAD X**
4	**SUB Y**
5	**LOAD P**
6	**DIV Q**

Modified MNT and MDT

Name of macro	No. of parameters	Starting Index
SAMPLE1	0	1
SAMPLE2	0	4
SAMPLE3	0	7

1	**LOAD A**
2	**ADD B**
3	MEND
4	**LOAD X**
5	**SUB Y**
6	MEND
7	**LOAD P**
8	**DIV Q**
9	MEND

Nested Macro Call (within definition)

```
LOAD A
STORE B
MACRO
 ADD1 &ARG
      LOAD X
      STORE &ARG
MEND
MACRO
 ADDS &A1, &A2, &A3
      LOAD C
      STORE D
      ADD1 5
      ADD1 &A1
      LOAD &A2
      LOAD &A3
MEND
ADDS  D1, D2, D3
END
```

Formal Parameter Table

#1	A1
#2	A2
#3	A3

Nested Macro Calls Actual Parameter

#1	D1
#2	D2
#3	D3

(MDT) Macro definition Table

```
0     LOAD X
      STORE #1
2     MEND
 3    LOAD C
      STORE D
      ADD1 5
      ADD1 #1
      LOAD #2
      LOAD #3
 9    MEND
```

(MNT) Macro Name table

```
ADD1 0
ADDS  3
```

Macro Processor Basics 2

Recap: Basics1
More on Macro Parameters
Nested Macro Call
Nested Macro Definitions
More on Nested Macro Definitions
Advanced Macro Facilities : AIF, AGO, Sequencing Symbols, Expansion time variables

Recap : Basics 1

- What is a Macro processor? (Input- Output)
- Macro Definition
- Macro Call
- Macro Expansion
- Macro Parameters
- 2 Pass Macro Processor (MNT, MDT)

More on Macro PARAMETERS

- Positional Parameters
- Keyword Parameters
- Mixed Parameters (e.g. in next slide)
- **Other uses of Parameters**
  - Formal parameters are **not restricted to OPERANDS**
    - Formal parameters can appear in any fields of a statement in macro
    - **Label – Opcode – Operand** (e.g. in next-to-next slide)

Default value & keyword parameters in macros

Program with macro	Expanded code
MACRO **XYZ &a, &b = 10, &c** **LOAD &a** **ADD &b** **STORE &c** **MEND** **XYZ 4, ,7** **XYZ 3, 8, 4** **END**	

Default value & keyword parameters in macros

Program with macro	Expanded code
MACRO **XYZ &a, &b = 10, &c** **LOAD &a** **ADD &b** **STORE &c** **MEND** **XYZ 4, ,7** **XYZ 3, 8, 4** **END**	**LOAD 4** **ADD 10** **STORE 7** **LOAD 3** **ADD 8** **STORE 4** **END**

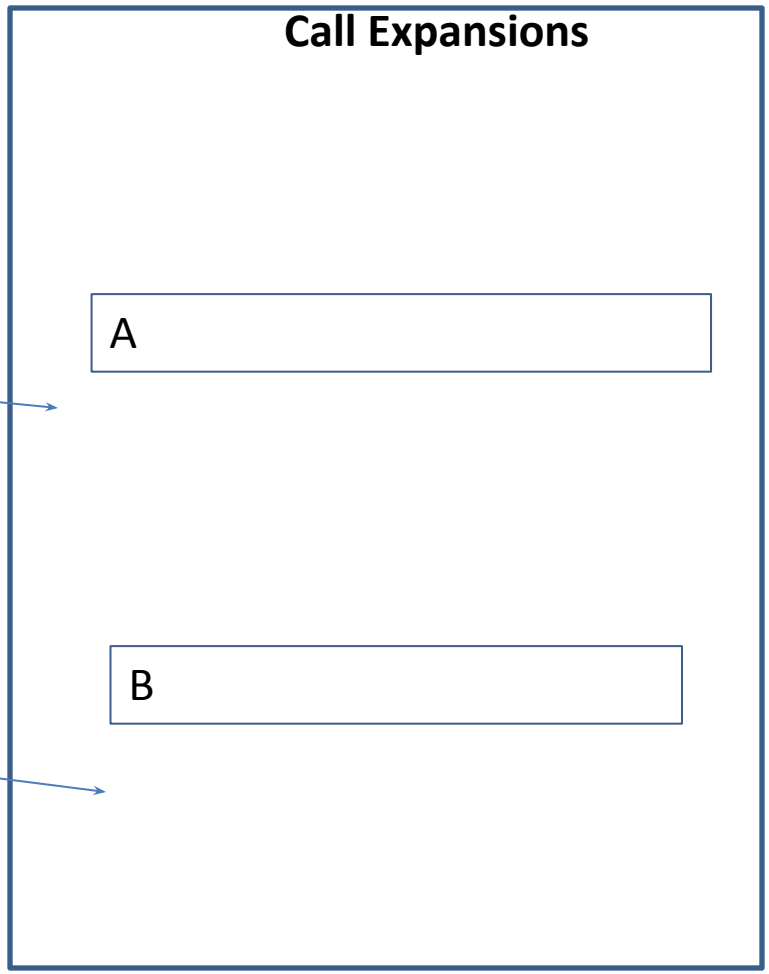
Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
    &LABEL1   MOVER       BREG, &X
          &OP1        BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N, ,L2
```



Formal parameters in LABEL And OPERAND FIELD Also

```
MACRO
      CALC      &X, &Y, &OP1=ADD, &LABEL1=L1
    &LABEL1   MOVER       BREG, &X
          &OP1        BREG,&Y
                MOVEM       BREG,&X
MEND

---------
CALC  A,B,  ,L1

----

CALC  M,N, ,L2
```

Call Expansions

```
L1    MOVER    BREG, A
      ADD        BREG,B
      MOVEM    BREG,A
```

```
L2    MOVER    BREG, M
      ADD        BREG,N
      MOVEM    BREG,M
```

Types of Macros

- Simple macro
- Macro with parameters

```
(Macro Definition)
MACRO INCR
        MOVER AREG, X
        ADD AREG,='1'
        MOVEM AREG,X
MEND
.................
(Macro Call (Invocation)
INCR
```

```
(Macro with parameters)
MACRO INCR  &R1, &V1
        MOVER &R1, &V1
        ADD &R1,='1'
        MOVEM
        &R1,&V1
MEND
.................
(Macro Call (Invocation)
INCR BREG, Y

(Formal Parameters R1,
        V1)

(Actual Parameters
        BREG, Y)
```

- Nested macro
  - Macro call within a macro definition
  - Macro definition within a macro definition
- Conditional macro
- Recursive macro

Sample: Nested Macro Call

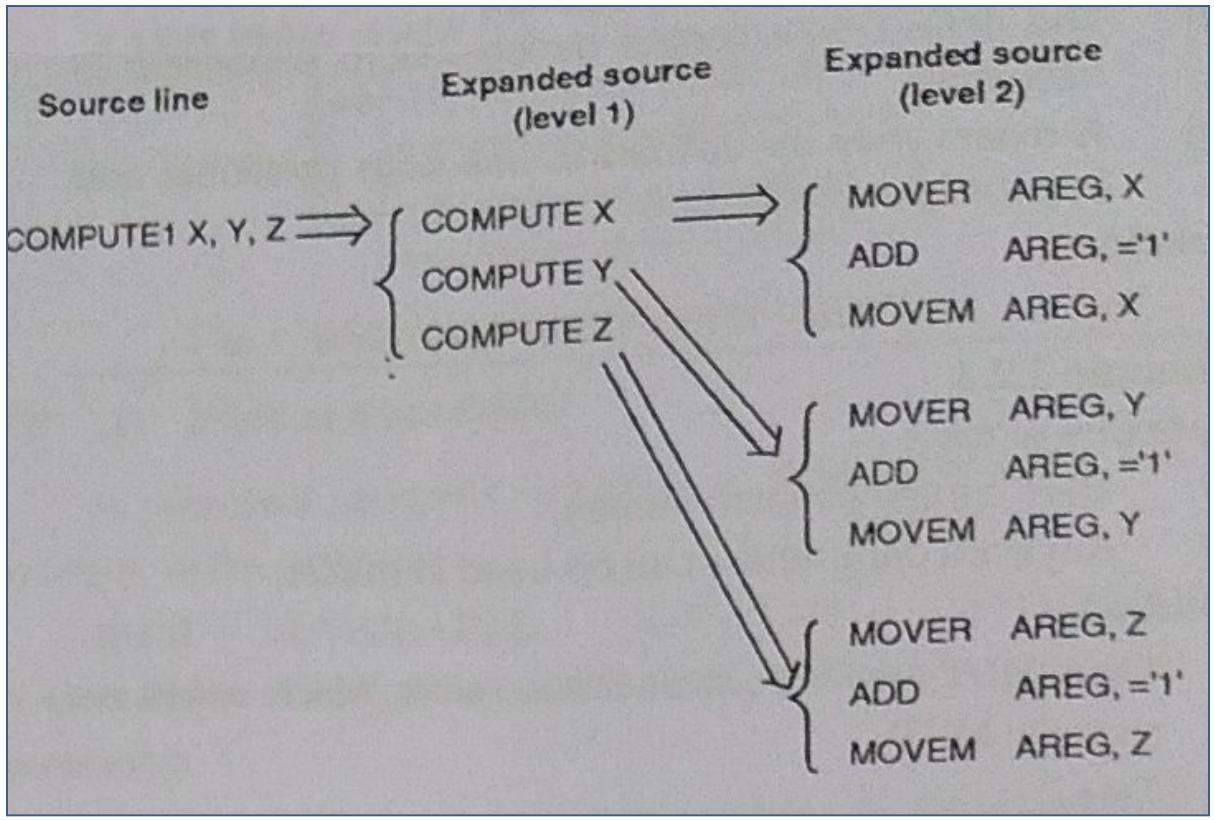
```
LOAD A
STORE B
MACRO
 ADD1 ARG
      LOAD X
      STORE ARG
MEND
MACRO
 ADD5 A1, A2, A3
      LOAD C
      STORE D
      ADD1 5
      ADD1 A1
      LOAD A2
      LOAD A3
MEND
ADD5 D1, D2, D3
END
```

Nested Macro Call

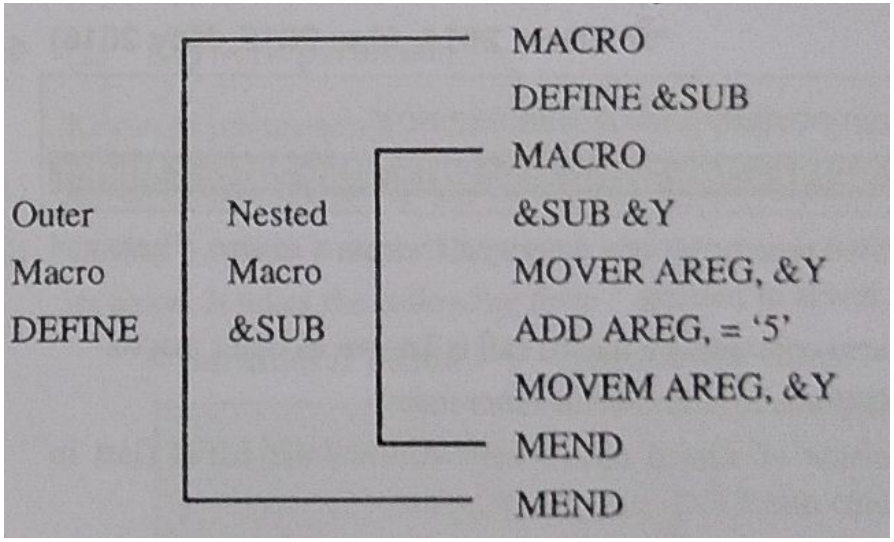
```
MACRO
        COMPUTE         &ARG
        MOVER           AREG, &ARG
        ADD             AREG, = '1'
        MOVEM           AREG, &ARG
MEND
MACRO
        COMPUTE1        &ARG1, &ARG2, &ARG3
        COMPUTE         &ARG1
        COMPUTE         &ARG2
        COMPUTE         &ARG3
MEND
```

```
MACRO
        COMPUTE         &ARG
        MOVER           AREG, &ARG
        ADD             AREG, = '1'
        MOVEM           AREG, &ARG
MEND
MACRO
        COMPUTE1        &ARG1, &ARG2, &ARG3
        COMPUTE         &ARG1
        COMPUTE         &ARG2
        COMPUTE         &ARG3
MEND
```

Nested Macro Call and its expansion



Nested Macro Definition



- DEFINE NESTED

```
MACRO
NESTED   &Y
MOVER    AREG, &Y
ADD      AREG, = '5'
MOVEM    AREG, &Y
MEND
```

Recap : Nested Macros

A. Macro call within macro definition	B. Macro definition within macro definition
LOAD A STORE B MACRO ADD1 ARG LOAD X STORE ARG MEND MACRO ADD5 A1, A2, A3 LOAD C STORE D ADD1 5 ADD1 A1 LOAD A2 LOAD A3 MEND ADD5 D1, D2, D3 END	MACRO OUTER &MACRONM, &OP MACRO &MACRONM &X, &Y, &Z LOAD &X &OP &Y STORE &Z MEND MEND

More on Nested Macro Definitions

- How to call nested macro?

CALL to Inner Macro

Wrong way of Placing call to inner macro

```
MACRO
OUTER  &INNERMNM, &OP
        MACRO
    &INNERMNM  &X, &Y, &Z
            LOAD &X
            &OP  &Y
            STORE  &Z
        MEND
MEND
……
ADDER A,B,C      Wrong CALL  (Outside the macro, if no
            Call to OUTER macro)
……
```

```
MACRO
OUTER  &INNERMNM, &OP
        MACRO
    &INNERMNM  &X, &Y, &Z
            LOAD &X
            &OP  &Y
            STORE   &Z
        MEND
MEND
….
OUTER ADDER, ADD
……
ADDER A,B,C
```

Correct CALL, with correct Call to outer

END

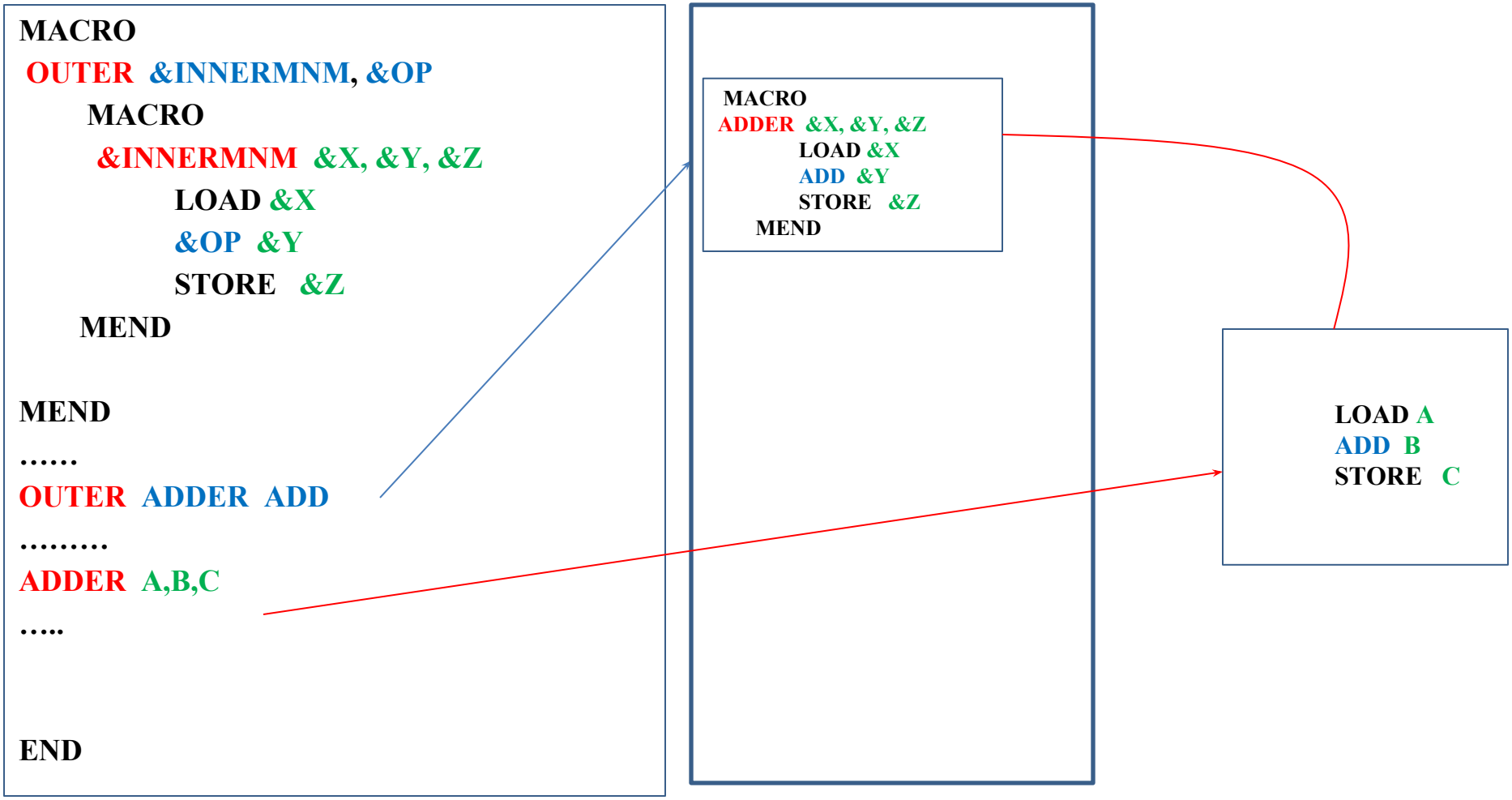
Solution: Call OUTER Correctly along with Call to ADDER

```
MACRO
ADDER  &X, &Y, &Z
        LOAD &X
        ADD  &Y
        STORE   &Z
    MEND
```

```
LOAD A
ADD  B
STORE   C
```

Placing Call to Inner Macro, outside the outer macro, but along with correct Call to Outer Macro

(Has to be Handled by the Macro-Processor)



Placing Call to Inner macro, outside the outer macro

Call inner macro outside outer macro along with Call to outer macro

```
MACRO
 OUTER  &INNERMNM, &OP
       MACRO
        &INNERMNM  &X, &Y, &Z
              LOAD &X
              &OP  &Y
              STORE  &Z
       MEND

MEND
……
OUTER  ADDER  ADD
……
……
OUTER SUBTRACTOR  SUB
……
ADDER  A,B,C
…..
SUBTRACTOR M,N,O
…..


END
```

```
MACRO
ADDER  &X, &Y, &Z
        LOAD &X
        ADD  &Y
        STORE  &Z
    MEND
```

```
MACRO
SUBTRACTOR  &X, &Y, &Z
        LOAD &X
        SUB  &Y
        STORE  &Z
    MEND
```

```
LOAD A
ADD  B
STORE  C
```

```
LOAD M
SUB  N
STORE  O
```

Q) Can a CALL to DIVIDER D,E,F be made as below ??

```
MACRO
 OUTER  &INNERMNM, &OP
       MACRO
        &INNERMNM  &X, &Y, &Z
              LOAD &X
              &OP  &Y
              STORE   &Z
       MEND

MEND
……
OUTER  ADDER  ADD
……
……
OUTER SUBTRACTOR  SUB
……
ADDER  A,B,C
…..
SUBTRACTOR M,N,O
…..
DIVIDER  D,E,F  ????  X

END
```

```
MACRO
ADDER  &X, &Y, &Z
        LOAD &X
        ADD  &Y
        STORE   &Z
    MEND
```

```
MACRO
SUBTRACTOR  &X, &Y, &Z
        LOAD &X
        SUB  &Y
        STORE   &Z
    MEND
```

```
LOAD A
ADD  B
STORE   C
```

```
LOAD M
SUB  N
STORE   O
```

Advance Macro Facility

Need

- Conditional reordering the sequence of MACRO expansion
- Conditional selection of the machine instructions appearing in expansion of macro call

How to Change Flow of control during Macro Expansion

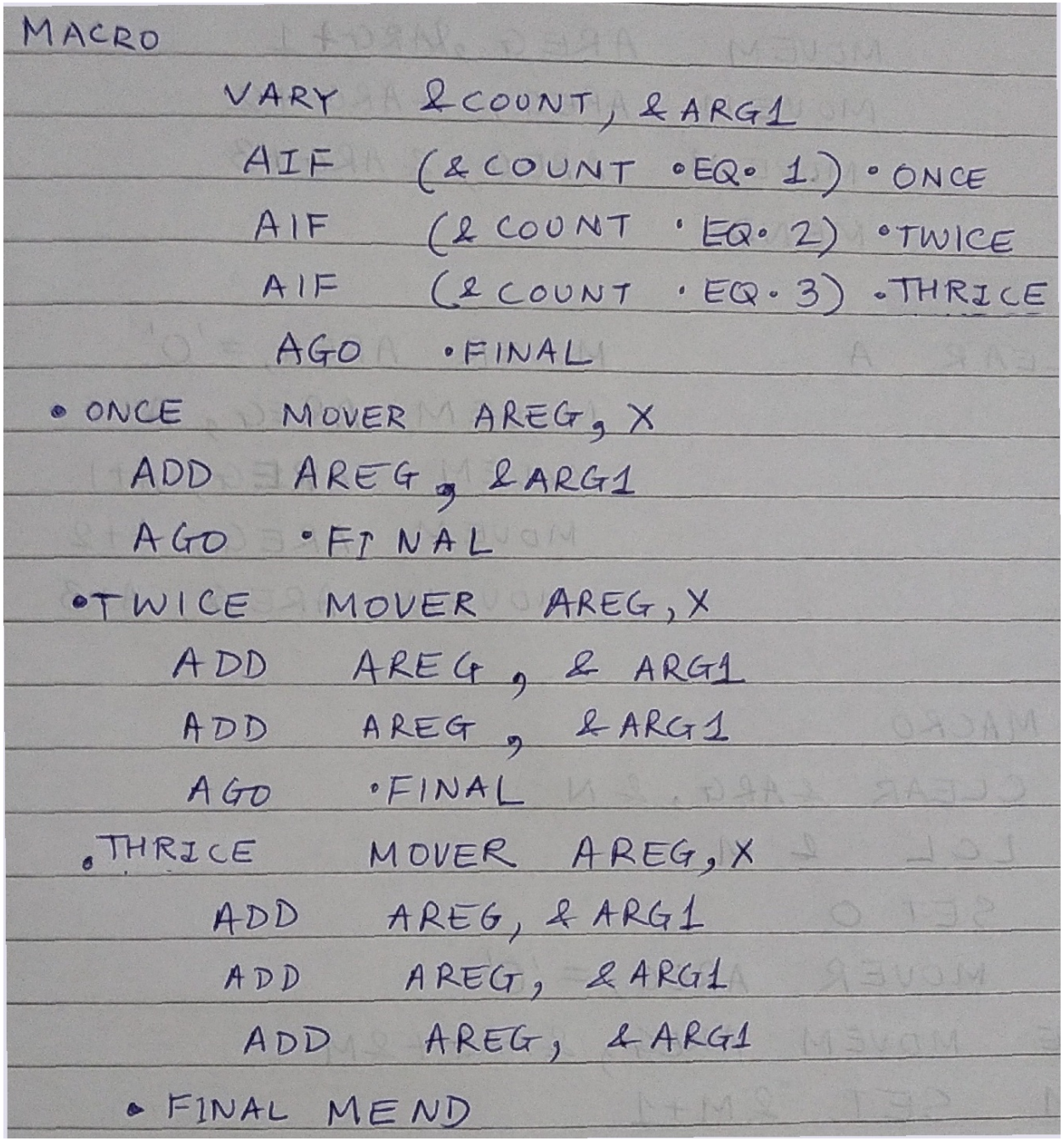
Use of the following:

- Psueop-op – **AIF (Conditional Branch)**
  - Similar to IF statement
  - Syntax: AIF (<expression>) <sequencing symbol>

- Psueop-op – **AGO (Unconditional Branch)**
  - Similar to GO TO statement
  - AGO <sequencing symbol>

- **Sequencing Symbol –** Label used for branching

MACRO
With
Advance Macro Facilities –
AIF, AGO
and
Sequencing Variable



Use of AIF, AGO and Sequencing Variable and Macro Expansion

```
MACRO
        VARY    &COUNT, &ARG1
        AIF     (&COUNT .EQ. 1) .ONCE
        AIF     (&COUNT .EQ. 2) .TWICE
        AIF     (&COUNT .EQ. 3) .THRICE
        AGO     .FINAL
.ONCE   MOVER   AREG, X
        ADD     AREG, &ARG1
        AGO     .FINAL
.TWICE  MOVER   AREG, X
        ADD     AREG, &ARG1
        ADD     AREG, &ARG1
        AGO     .FINAL
.THRICE MOVER   AREG, X
        ADD     AREG, &ARG1
        ADD     AREG, &ARG1
        ADD     AREG, &ARG1
.FINAL  MEND
```

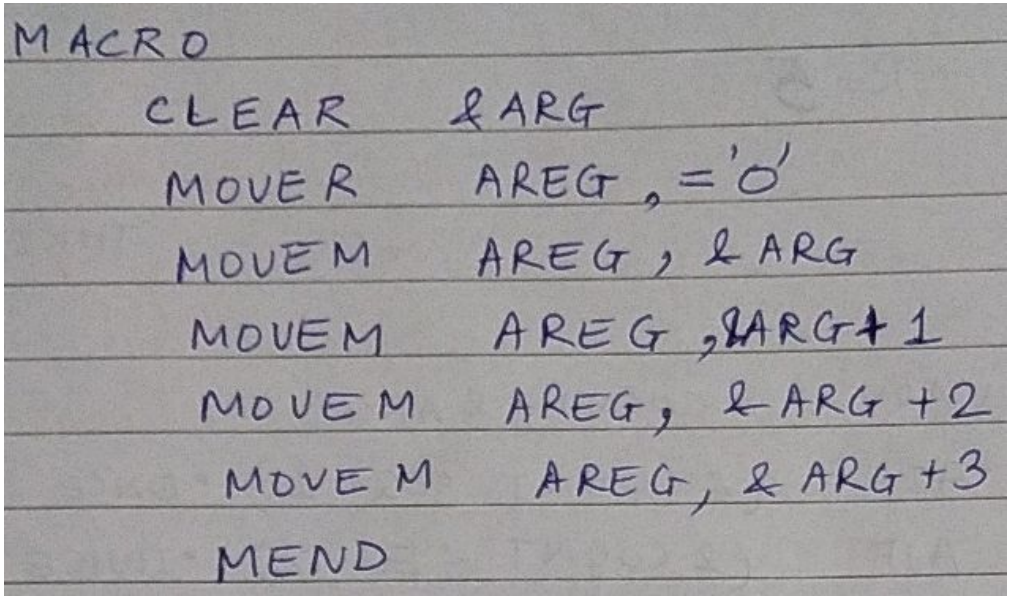
Macro Call	Expansion	
VARY 1, Y	MOVER	AREG, X
	ADD	AREG, Y
VARY 2, Y	MOVER	AREG, X
	ADD	AREG, Y
	ADD	AREG, Y
VARY 3, Y	MOVER	AREG, X
	ADD	AREG, Y
	ADD	AREG, Y
	ADD	AREG, Y

Expansion Time Variables (EVs)

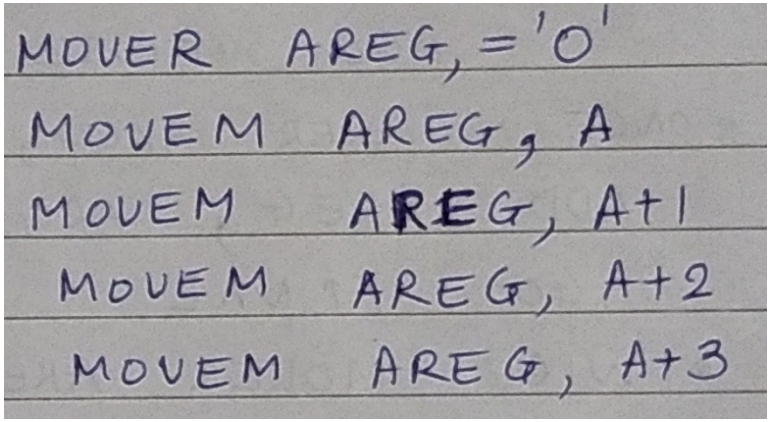
- Used during Macro Expansion
- Declared as Local Variable using **LCL**
  - **LCL <&variable name> [, <&variable name>,..]**
- **SET** is used to manipulate EVs

Need for EVs – When similar statements in MACRO

- E.g Similar statements in Macro



- EXPANSION of call say, **CLEAR A**
  - Results in the following



Use of EVs to avoid typing out similar statements (by Programmer)

- MACRO with EV

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

- Call expansion

CLEAR A,3

```
MOVER  AREG,='0'
MOVEM  AREG, A
MOVEM  AREG, A+1
MOVEM  AREG, A+2
MOVEM  AREG, A+3
```

Comment handling in macros

- Comments given in macro definition, should not reflect in target file, otherwise all other comment should reflect in target file as it is.
- ; start of comment

Thank You !!

Thank you !!