# Lecture 3

# Design of a macro preprocessor

- Accepts assembly program containing definitions and calls and translates it into assembly program which **does not contain any macro definition or calls**

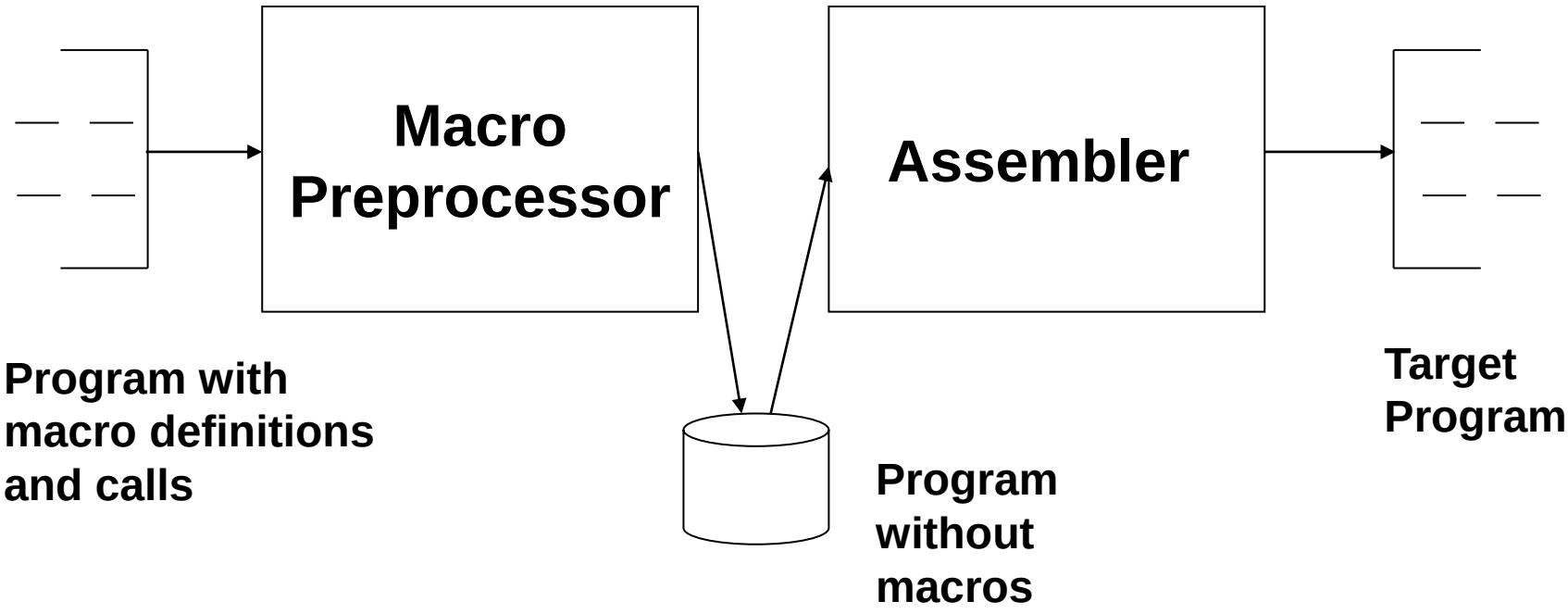
- The program form o/p by the macro preprocessor can now be handed over to assembler to obtain the target language form of program

- Macro preprocessor segregates macro expansion from the process of program assembly

- It is **economical** because it can use an existing assembler

- It is not as efficient as a **macro assembler** i.e. an assembler that performs macro expansion as well as assembly

Figure : A macro preprocessor



# Tasks in Macro expansion

1. Identify macro calls in the program
2. Determine the values of formal parameters
3. Maintain the values of expansion time variables declared in a macro
4. Organize expansion time control flow
5. Determine the values of sequencing symbols
6. Perform expansion of a model statement

# Identify macro calls

- Table called **macro name table (MNT)** is designed to hold the **names of all macros** defined in a program
- **Macro name is entered in this table when a macro definition is processed**

- While processing a statement in the source program, the preprocessor compares the string found in its mnemonic field with macro names in MNT
- Match indicates that the current statement is macro call

# Determine values of formal parameters

- **Actual parameter table (APT)** hold the **values of formal parameters during expansion** of macro call

- Each entry in table is a pair

  **(<formal parameter name>, <value>)**

- Two items needed to construct this table
  - **Names** of formal parameters
  - **Default values** of keyword parameters

- **Parameter default table (PDT)** is used for each macro. **Accessible from the MNT entry** of a macro and would contain pairs of the form

  **(<formal parameter name>, <default value>)**

- If macro call statement does not specify a value for some parameter, its **default value would be copied from PDT to APT**

# Maintain expansion time variables

- **Expansion time variables' table (EVT)** is maintained

- Contains the pairs of the form

  **(<EV name>, <value>)**

- Value field of a pair is accessed when a preprocessor statement or model statement under expansion refers to EV

# Organize expansion time control flow

- Body of a macro is stored in a table called **Macro definition table (MDT)** for use during macro expansion
- Flow of control during macro expansion determines when a model statement is to be visited for expansion
- MEC also can be used

- MEC can be initialized to first statement of macro body in MDT

- It can be updated after expanding a model statement or on processing a macro preprocessor statement

# Determine values of sequencing symbols

- **Sequencing symbols table (SST)** is maintained

  **(<sequencing symbol name>, <MDT entry#>)**

- <MDT entry#> is number of MDT entry which contains the model statement defining the SS

- This entry is **made on encountering a statement which contains the SS in its label field** or on **encountering a reference** prior to its definition

# Perform expansion of model statement

1. MEC points to MDT entry containing model statement
2. Values of formal parameters and EV's are available in APT and EVT, resp.
3. The model statement defining a SS can be identified from SST

- Expansion of model statement is achieved by **performing lexical substitution for the parameters and EV's** used in model statement