Compiled By Saroj Dahal

ASSEMBLER ALGORITHM AND DATA STRUCTURES

Our simple assembler uses two major internal data structure; the Operation Code table (OPTAB) and the Symbol Table (SYMTAB). OPTAB is used to look up mnemonic operation codes and translate them to their machine language equivalents. SYMTAB is used to store values(addresses)assigned to labels.

We also need a Location counter LOCCTR. This is a variable that is used to help in the assignment of addresses. LOCCTR is initialized to the beginning address specified in the START statement . After each source statement is processed, the length of the assembled instructions or date area to be generated is added to LOCCTR. Thus whenever we reach a label in the source program , the current value of LOCCTR gives the address to be associated with that label.

The Operation code Table must contain the mnemonic operation code and its machine language equivalent. In more complex assembles , this table also contains information about instruction format and length.During Pass 1 , OPTAB is used to look up and validate opearation coded in the source program. In pass 2 , it is used to translate the operation codes to machine language. Actually , in our simple SIC assembler , both of these processes could be done together in either PASS 1 or PASS 2. However , for a machine that has instructions of different lengths, we must search OPTAB in the first pass to find the instructions length for incrementing LOCCTR. Likewise , we must have the information from OPTAB is PASS 2 to tell us which instruction format to use in assembling the instruction ,and my peculiarities of the object code instruction. We have chosen to retain this structure in the current discussion because it is typical of most real assembles.

OPTAB is usually organized as a hash table , with mnemonic operation code as the key. The hash table organization is particularly appropriate , since it provides fast retrieval with a minimum of searching . In most cases , OPTAB is a static table – that is , entries are not normally added to or deleted from it. In such cases it is possible to design a special hashing function or other data structure to give optimum performance for the particular set of keys being stored. Most of the time , however , a general –purpose hashing method is used.

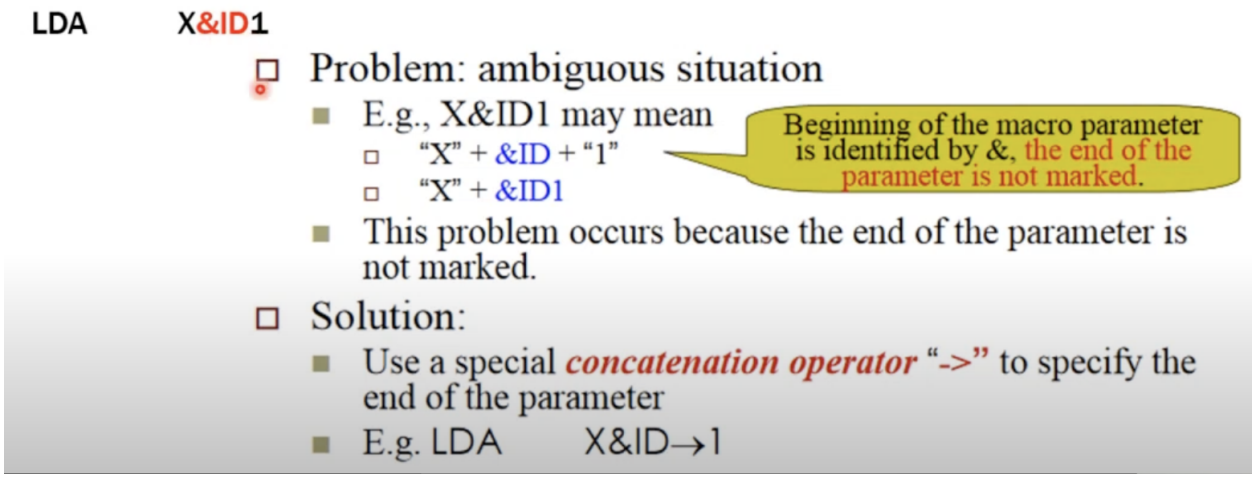
The Symbol table (SYMTAB) includes the name and value (address) for each label in the source program , together with flags to indicate error conditions (e.g,. a symbol defined in two different places). This table may also contain other information about the data area or instruction labeled – for example , its type or length. During Pass 1 of the assembler , labels are entered intoSYMTAB as they are encounterd in the source program , along with their assigned addresses (form LOCCTR). During Pass 2 , symbols used as operands are looked up in SYMTAB to obtain the addresses to be inserted in the assembled instructions.

SYMTAB is usually organized as a hash table for efficiency or insertion and retrieval. Since entries are rarely deleted from this table , efficiency of deletion is not an important consideration. Because SYMTAB is used heavily throughout the assembly , care should be taken in the selection of a hashing function. Programmers often select many labels that have similar characteristics – for example , labels that start or end with the same characters (like LOOP1 , LOOP2) or are of the same length . It is important that the hashing fuction used perform well with such no-random keys. Division of the entire key by a prime table length often gives good result.

It is possible to both passes of the assembler to read the original source program as input. However , there is certain information (such as location counter values and error flags for statements) that can or should be communicated between the two passes. For this reason , Pass usually writes an INTERMEDIATE FILE that contains each source statement together with its assigned address , error indicators , etc ,. This file is used as the input to Pass 2.This working copy of the source program can also be used to retain the results of certain operations that may be performed during Pass 1 , so these need not be performed again during Pass2. Similarly , pointers into OPTAB and SYMTAB may retained for each operation code and symbol used . This avoids the need to repeat many of the table-searching operations.We assume for simplicity that the source lines are written in a fixed format with fields LABEL , OPCODE , and OPERAND . If one of these fields contains a characted string that represents a number , we denote its numeric value with the prefix # (for example , # [OPERAND]).

4.2.1 Concatenation of Macro Parameters

- The beginning of the macro parameter is identified by the starting symbol &; however, the end of the parameter is not marked.



Problem in pre-concatenation.. And introduced a concatenation operator.

Assembly Directives in Macro Definition : MACRO, MEND.
Data Structures used by Macro-processor : DEFTAB,NAMTAB,ARGTAB
Machine Independent Macro-processor Features :

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

Disadvantages of one pass assembler:
The Forward reference problem exists in the one-pass assembler. In this problem operands are utilized before the declaration, thus assembler has no information about the operand address that is to be present in the final executable code.