Variables and Constants

- The various characters used to construct assembler variables, constants and directives are the following
- Uppercase English Alphabets : A to Z
- Lower case English Alphabets : a- z
- Numbers : 0 – 9
- Special Characters : @, $, ?, _

Variables

- Variables are symbols used in ALP statements in order to represent the variables data and address
- The advantage of using variables is that the value of the variable can be dynamically varied while running the program

Rules

- The variable name can be any of the character set
- First character should be an alphabet or underscore
- The length of a variable name depends on assembler, normally 32 characters
- Are case sensitive

Constants

- The decimal, binary or hexadecimal numbers used to represent the data or address in an ALP statement are called constants or numerical constants
- Their values are fixed and cannot be changed while running a program
- The binary (ends with B), hexadecimal (Ends with H) and decimal constants (ends with D) can be differentiated by placing a specific alphabet at the end of the constant
- A zero should be placed at the beginning of hexadecimal number, else it will be treated as a variable
- Ex. 1101B, 1060D, 92ACH

Assembly Directives

- These are instructions to the assembler regarding the program being assembled
- They are also called pseudo-instructions
- They are used to
  - specify start and end of a program,
  - attach value to variables,
  - allocate storage locations to input/output data,
  - define start and end of segments, procedures and macros etc.

- They control the generation of machine code and organization of the program
- But no machine codes are generated for assembly directives

DB (Define Byte)

- Used to define a byte type variable
- It reserves specific memory to variables and stores the values specified in the statement as initial values in the allotted memory locations
- The range of values that can be stored in a byte type variable is 0-255 for unsigned value and -128 to 127 for signed value
- Ex:- AREA DB 45 -> variable AREA is initialised with 45

DW (Define Word)

- Used to define a word type variable
- It reserves two consecutive memory locations to each variable and store the 16-bit value specified in the statement as initial value in the allotted memory locations.
- Range of values 0-65535 for unsigned value and -32768 – 32767 for singed value
- Ex:- BCODE DW ‘8E’ -> two consecutive locations are reserved for variable BCODE and initialized with ASCII value 8 and E

SEGMENT AND ENDS (END of SEGMENT)

- The directive SEGMENT is used to indicate the beginning of a code/data/stack segment.
- The directive ENDS is used to indicate the end
- General form

```
Segnam SEGMENT
.
.
Segnam ENDS
```

Where 'segnam' is the user defined name of the segment

ASSUME

- ASSUME informs the assembler, the name of the program/data segment, that should be used for a specified segment.
- The general form of a statement using ASSUME directive is given below
- ASSUME segreg : segnam, … segreg : segnam
- Where segreg is the segment register
- ASSUME CS: _CODE
  - Informs the assembler that the instruction of the program are stored in the user-defined logical segment _CODE

ORG, END, EVEN and EQU

- ORG is used to assign the starting address (effective address) for a program/data segment
- The END directive is used to terminate a program
  - The statement after the END directive will be ignored by the assembler
- The directive EVEN will inform the assembler to store the program/data segment starting from an even address
  - The 8086 requires one bus cycle to access a word at even address and two bus cycles to access a word at odd address.
  - So it helps in accessing a series of consecutive memory words quickly
- EQU is used to attach a value to a variable

PROC, FAR, NEAR and ENDP

- These directives are used to define a procedure/subroutine
- The directive PROC indicates the beginning of a procedure and ENDP indicates the end of a procedure
- The FAR or NEAR are type specifier which is used to differentiate intra-segment call (call within a segment/near call) and inter-segment call (call from another segment/far call)

- The general form of writing a procedure is

  procname PROC [NEAR/FAR]

  .

  RET

  procname ENDP

- **SHORT**
- Used to reserve one memory location for 8-bit singed displacement in jump instructions
- Ex. JMP SHORT AHEAD
  - Will reserve one memory location for an 8-bit displacement named AHEAD