Assembler Directives (Pseudo opcodes)

→ we need them to write program.

→ These are not instructions but these are used to make structure program.

→ Used to convert and Assembly language program into machine language code modules.

→ It is a statement to give direction to the assembler to ~~the~~ perform task of the assembly process.

① DB: Define Byte (8 bits value)

→ It is used to reserve byte or byte of memory location in the available memory.

example: ① RANKS DB 01H, 02H, 03H, 04H.

→ Reserve four memory locations.

② MESSAGE DB 'GOOD MORNING'

→ Reserve no. of bytes of memory location as same as no. of characters present in string.

② DW: Define Word (16 bits value)

→ It makes the assembler reserve the number of memory words instead of bytes.

example: WORDS DW 1234H, 4567H, 78ABH, D45CH

→ reserve four words space. In this,

LB → lower memory add.

UB → higher memory add.

③ DQ : Define Quadword (64 bits value).

→ It is used to direct the assembler to reserve 4 words of memory.

④ DT: Define Ten Bytes (10 bytes value).

→ Directs the assembler to define the specified variable req 10- bytes.

→

EQU: Equate to

→ Used to assign constant value to a symbol.

→ Macro-assembler replaces every occurance of symbol by its value.

Syntax : Symbol-name EQU value.

example: NUMBER EQU 100

→ In the program, whenever there is number it automatically take the value as 100.

ORG : Originate

→ It assigns the location counter with the value defined in the directive.

Syntax : ORG $+Numerical-value.

Example:

```
ORG   1000H
ORG   $
ORG   $+200
```

Program organization Directives

* ASSUME :

→ Conveys logical name of segment to assembler

→ Segment register points ~~to~~ the logical segment

eg ~~ASSUME~~ Assume CS: code, DS: Data.

* SEGMENT

→ Indicates the beginning of logical segment.

→ SEGMENT follows the name of the segment.

example Data SEGMENT
{Data declaration}.
Data ENDS

ENDS : End of the segment.

* END : End of program.