

Experiment 01

Learning Objective: Student should be able to apply Assembly Language Programming to enter and display 8 bit & 16 bits number_

Tools: TASM/MASM

Theory:

Assembler Directives:-

This type of statements includes commands that are addressed to the assembler, such as:
Constant and variable definition.
Allocation of memory space and initialization of memory, and
Control of the assembly process

List of assembler directives

a. Data Allocation Directives

DB.....define byte

DW.....define word (2 bytes)

DD.....define double word (4 bytes)

DQ.....define quadword (8 bytes)

DT.....define tenbytes

EQU.....equate, assign numeric expression to a name

Examples:

```
db      100 dup (?)    define 100 bytes, with no initial values for bytes
db      "Hello"        define 5 bytes, ASCII equivalent of "Hello".
maxint   equ    32767
count    equ    10 * 20    ; calculate a value (200)
```

ENDS.....used to indicate the end of the segment.

END..... used to indicate the end of program.

PROC.....used to indicate the beginning of a procedure.

ENDP..... used to indicate the end of a program.

ENDM..... used to indicate the end of a program.

SEGMENT.....used to indicate the start of the segment.

TITLE.....used to indicate the title of the program.

EQUused to give a name to some value or to a symbol. Each time the assembler finds the name in the program, it will replace the name with the value or symbol you given to that name.

ASSUME..... Associates a logical segment to processor segment.

e.g. Example:

ASSUME CS:CODE ;

TASM COMMANDS:

C :>cd foldername

C:/foldername>edit filename.asm

After this command executed in command prompt an editor window will open. Program should be typed in this window and saved. The program structure is given below.

Structure of Program:

.model tiny/small/medium/large

.Stack <some number>

.data

; Initialize data

; which is used in program.

.code

.startup

; Program logic goes here.

.exit

end

To run the program, the following steps have to be followed:

C:/foldername>Tasm filename.asm

After this command is executed in command prompt if there are no errors in program regarding to syntax the assembler will generates an object module as discuss above.

C:/foldername>Tlink filename.obj

After verifying the program for correct syntax and the generated object files should be linked together. For this the above link command should be executed and it will give an EXE file if the model directive is small as discuss above.

C:/foldername>td filename.exe

After generating EXE file by the assembler it's the time to check the output. For this the above command is used and the execution of the program can be done in different ways. It is as shown below:

__ g ; complete execution of program in single step.

__ t ; Stepwise execution.

__ d ds: starting address or ending address ; To see data in memory locations

__ p; Used to execute interrupt or procedure during stepwise execution of program

__ q ; To quit the execution.

5. Procedure/ Algorithm

Program to accept 8 bit number and display 8 bit number

Explanation: Conversions from ASCII to binary usually start with keyboard data entry. If a single key is typed the conversion is accomplished by subtracting a 30H from the number. If more than one key is typed, conversion from ASCII to binary still requires that 30H be subtracted, but there is one additional step. After subtracting 30H, the prior result is first multiplied by 10, the number is added to the result. The algorithm used to convert ASCII to binary is:

1. Begin with a binary result of 0.
2. Subtract 30H from the character typed on the keyboard to convert it to BCD.
3. Multiply the binary result by 10 and add the new BCD digit.
4. Repeat steps 2 and 3 until the character typed is not an ASCII coded number of 30H-39H.

Functions and Interrupts:

1. Display message on screen.
Mov ah,09
Lea dx, msg
Int 21h
2. Enter single char from user.
Mov Ah,01
Int 21h
Return : AL= ASCII value
3. Display single char on screen.
Mov Ah,02
Int 21h

Application:

1. Conversion from ASCII to BCD
2. Conversion from BCD to ASCII

Design:

Result and Discussion:

Learning Outcomes:

The student should have the ability to

- LO1 List the features of Assembly language.
- LO2 Identify the role of translator in programming language.
- LO3 List and define the assemble directives.
- LO4 Implement a basic program using assembly language features.

Course Outcomes: Upon completion of the course students will be able to make use of instructions of 8086 to build assembly and Mixed language programs.

Conclusion:



TCET

DEPARTMENT OF COMPUTER ENGINEERING (COMP)

(Accredited by NBA for 3 years, 3rd Cycle Accreditation w.e.f. 1st July 2019)

Choice Based Credit Grading Scheme (CBCGS)

Under TCET Autonomy



Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				