Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

POA EXPERIMENT 9

Jigar Siddhpura
60004210155

POA Experiment 9 : Factorial

Aim : Implement assembly program to find factorial with or without macros.

Theory : 1. In assembly program, concept of macros offer a powerful tool for code abstraction & simplification.
2. First implementation utilizes, traditional language program initializes a data segment with single variable 'A', the no. of which factorial is to be calculated.
3. The program iteratively ~~dec~~ decrements 'A' by 1 & multiplies it with the value in 'A' ~~unit~~ until 'A' equals 1.
4. Later storing it in 'FACT' variable. This approach relies on explicit instr & repetitive code

Macros : 1. Macros in ALP acts as a preprocessor directive that allows programmer to define reusable code segments.
2. They maintain code reusability, readability & reduce redundancy by encapsulating repitive instr. into single named entity.

→ Second program utilizes MACRO instruction to find factorial using FACT label.
It takes a single argument 'F' representing the no. for which factorial is to be calculated.
It encapsulates the iterative factorial calculation process.
This implementation initializes a data segment with number 'num' & utilizes 'FACT' macro to calculate factorial storing result in RESULT variable.

1. By using MACRO, it reduces the no. of explicit instructions enhancing code readability, making it easier to understant & maintain.
2. The without macro implementation, while achieving same result involves more manual instr.; leading to code duplication & chaces of errors.

Conclusions :

Hence, we implemented both programs, with & without macros to find factorial of a no. & understood the use of macros & how it is beneficial for code org., readability & maintainibity.

Code :

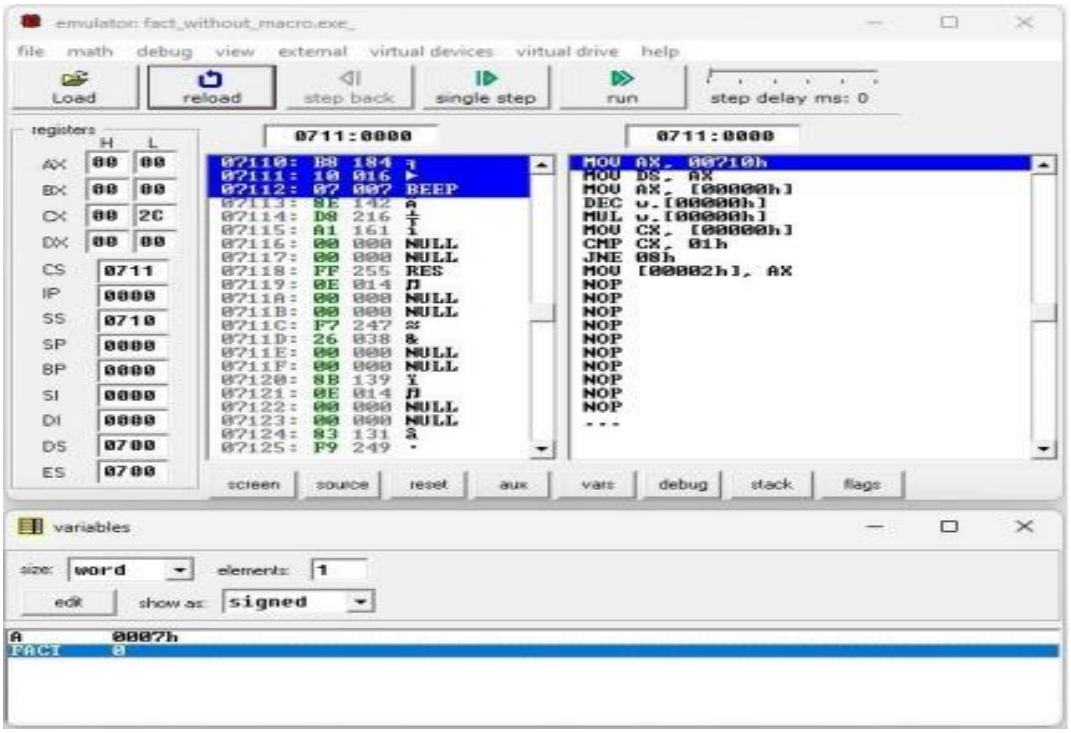
Factorial of a number without using macro instruction

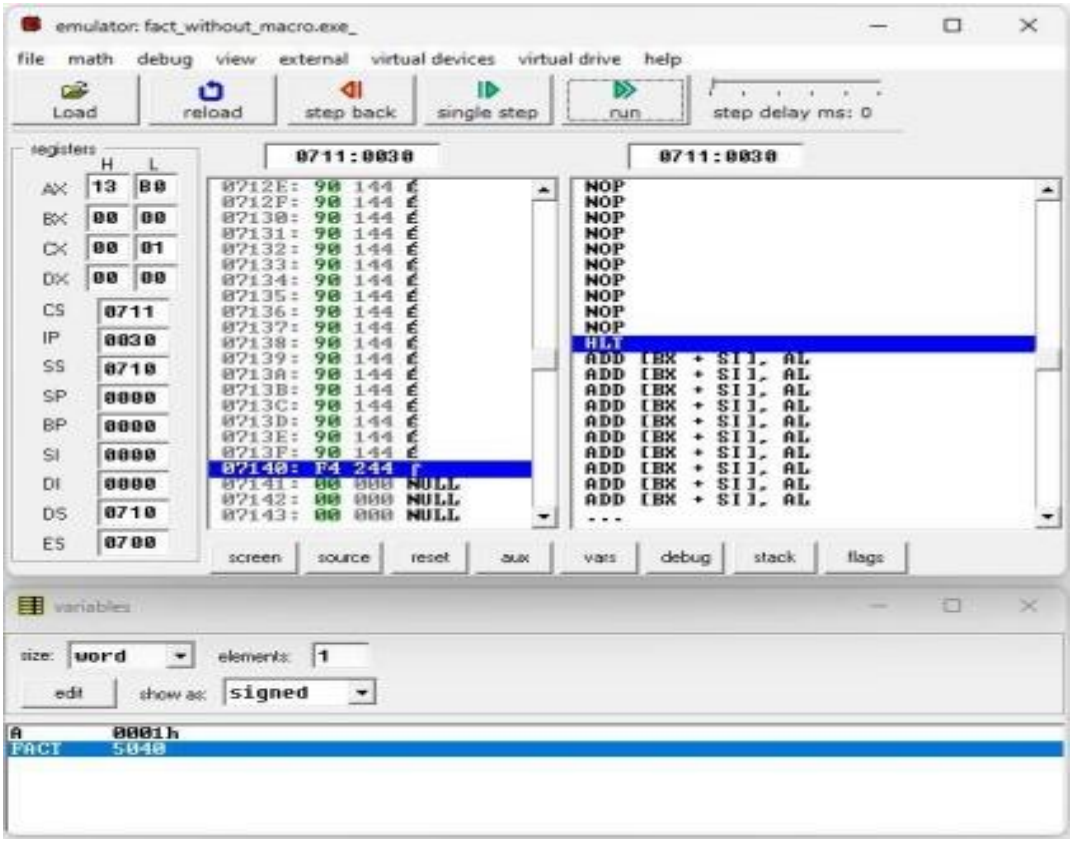
```
org 100h

DATA SEGMENT
   A DW 7
   FACT DW ?
   DATA ENDS
CODE SEGMENT
   START:
   MOV AX,DATA
   MOV DS,AX
   MOV AX,A
   L1:
   DEC A
   MUL A
   MOV CX,A
   CMP CX,01
   JNZ L1
   MOV FACT,AX
   CODE ENDS
END START

Ret
```

Factorial of a number using macro instruction :

```
FACT MACRO F
   UP:
   MUL F
   DEC F
   JNZ UP
ENDM
   DATA SEGMENT
   NUM DW 06H
   RESULT DW ?
ENDS
   CODE SEGMENT
   START:
   MOV AX, DATA
   MOV DS,AX
   MOV CX,NUM
   MOV AX,0001H
   FACT NUM
   MOV RESULT,AX

ENDS
END START
```

Output (without macro):





Output (with Macro) :