K. K. WAGH POLYTECHNIC, NASHIK.

HIRABAIHARIDASVIDYANAGARI, AMRUTDHAM, PANCHAVATI,NASHIK-422003

DEPARTMENT OF COMPUTER TECHNOLOGY

Micro-Project Report

Institute Code: 0078
Academic Year: 2022-23 Program: Computer Technology
Course: Microprocessors (MIC) Course Code: 22415 Scheme: I
Semester: 4 Class: SYCM-I Date of Report:27/03/2023

Title of Micro-Project: **Write an ALP to write a procedure for performing Basic Arithmetic operations**

1.0 Problem Definition:

The objective of this project is to create a modular procedure in 8086 assembly language to perform basic arithmetic operations like addition, subtraction, multiplication, division square,cube and factorial,without taking input from the user. The procedure will be optimized for maximum efficiency and accuracy, and the result will be stored in a register or memory location as defined in the code. The program will include appropriate comments and documentation for future reference and can be used in other programs by simply calling it. The ultimate goal is to create a versatile and efficient procedure for basic arithmetic operations.

2.0 Rationale:

Microprocessor is the main component of a computer where 8086 is the base of all upward developed processors till current processors. This course will cover the basics of 8086 and its architecture along with instruction set, assembly language programming with effective use of procedure and macros. This course also covers the architectural issues such as instruction set program and data types. On top that, the student are also introduced to the increasingly important area of parallel organisation. This subject serves as a basic to develop hardware related projects, This course will enable the students to inculcate assembly language programming concepts and methodology to solve problems.

3.0 Aim /Benefits of Micro-Project:

The Aim/Objectives of the Micro-project is:

- To develop an understanding of assembly language programming and its applications in performing basic arithmetic operations.
- To implement an ALP that can perform operations such as addition, subtraction, multiplication, and division of two numbers.also find square,cube and factorial
- To optimize the ALP to improve the performance and efficiency of arithmetic operations.
- To test and debug the ALP to ensure accurate and reliable results.
- To document the ALP code and provide instructions for its use and modification.
- To explore the potential of the ALP in solving practical problems that require arithmetic operations, such as data processing and scientific calculations.
- To evaluate the limitations and challenges of using assembly language programming for arithmetic operations and compare it with other programming languages.

4.0 Course Outcomes Achieved (COs):

CO.1: Write assembly language program for the given problem.
CO.2: Develop an assembly language program using assembler.
CO.3: Develop assembly language programs using procedures, macros and modular programming approach.

5.0 Literature Review:

The 8086 microprocessor is capable of performing arithmetic operations such as addition, subtraction, multiplication, and division using the Arithmetic and Logic Unit (ALU) and the General Purpose Registers (GPR). The procedure for performing arithmetic operations in 8086 involves loading

the operands into the GPR, selecting the appropriate instruction, and executing the operation using the ALU. The result is stored back into the GPR. Assembly language programming is commonly used to perform arithmetic operations using the 8086 microprocessor.

6.0 Actual Methodology followed:

1. Discussing on various topics and finalized one topic of ALP to write a procedure for performing Basic Arithmetic operations
2. Identify the specific arithmetic operations to be performed, such as addition, subtraction, multiplication, or division. also square,cube and factorial
3. Choose an appropriate assembly language and IDE (Integrated Development Environment) for writing and testing the ALP, such as MS-DOS,TASM,TLINK,TD.
4. Write the ALP code using the appropriate assembly language syntax and instructions to perform the chosen arithmetic operations.
5. Test the ALP code by inputting different sets of data and verifying that the output matches the expected results for each arithmetic operation.
6. Optimize the ALP code for efficiency and speed by reducing the number of instructions and minimizing memory usage.
7. Document the ALP code and its functionality, including input and output formats, data types, and any limitations or constraints.
8. Validate the ALP code by comparing it with existing solutions and verifying that it meets the requirements and specifications of the project.
9. Iterate and refine the ALP code as needed based on feedback and testing results.
10. Testing all the modules and making corrections as per guidelines suggested by Guide.
11. Finalised all modules of ALP to to perform basic operations.
12. Generated Final Output.

7.0 Actual Resources used:

S. No.	Name of Resource/material	Specifications	Qty	Remarks
1.	Laptop	HP Elitebook 830 g3-Intel(R) Core(TM) i5, 8 GB installed RAM,SSD-256 GB,HDD-1TB	01	For project work
2.	Operating System	Windows 11 Pro x64 bit	01	For running ALP Tools
3.	Editor	MS-DOS :-1.4v	01	For Editing and Making Changes of ALP
4.	Assembler	TASM :-1.4v	01	For converting ALP to MLL
5.	Linker	TLINK :-2.0v	01	For linking different object files
6.	Debugger	TD :-2.0v	01	For finding and fix errors
7.	Other software	Microsoft Office Word	01	For documentation
8.	Reference book:	The 8088 & 8086 Microprocessor	01	For studying MIC concepts

9.	Websites	www.geeksforgeeks.org www.studytonight.com www.tutorialspoint.com	03	For referring sample ALP & concept

8.0 Outputs of the Micro-Project:

❖ **CODE-**

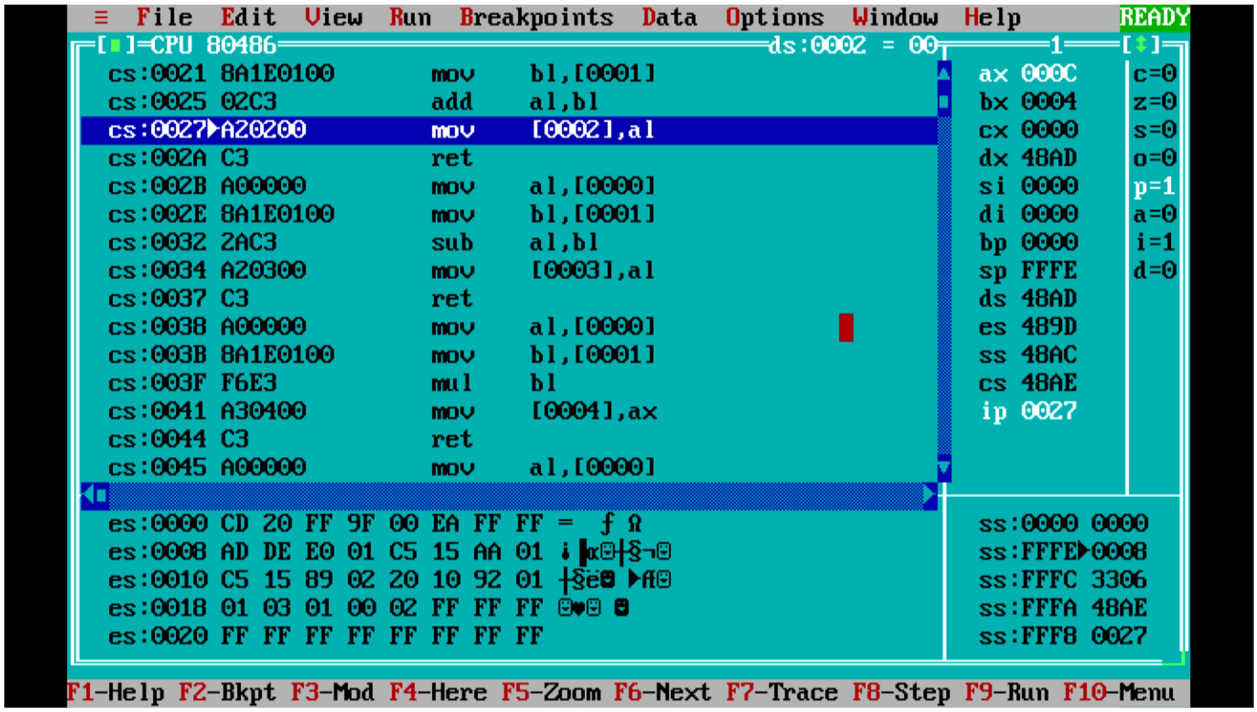
```
; Define the data segment
Data Segment
   a db 08h        ; Input variable a
   b db 04h        ; Input variable b
   res_add db 00h   ; Result of addition
   res_sub db 00h   ; Result of subtraction
   res_mul dw ?     ; Result of multiplication
   res_quo db 00h   ; Result of division (quotient)
   res_rem db 00h   ; Result of division (remainder)
   res_sqr dw ?     ; Result of squaring a
   res_cube dw ?    ; Result of cubing b
   res_fact dw ?    ; Result of factorial b
;End of Data Segment
Data ends

; Define the code segment
Code Segment
   Assume CS:Code, DS:Data

Start:
   ; Data Segment Initialization
   mov dx,Data
   mov ds,dx

   ; Perform addition of a and b
   call add1
   ; Perform subtraction of b from a
   call sub1
   ; Perform multiplication of a and b
   call mul1
   ; Perform division of a by b
   call div1
   ; Perform squaring of a
   call sqr
   ; Perform cubing of b
   call cube1
   ; Perform factorial of b
```

```
call FACTORIAL

; Exit the program
mov ah,4ch
int 21h

; Procedure to add a and b
add1 PROC NEAR
        mov al,a        ; Move the value of a to the AL register
        mov bl,b        ; Move the value of b to the BL register
        ; Add the values in AL and BL and store the result in AL
        add al,bl
        ; Move the result of the addition to the res_add variable
        mov res_add,al
        ; Return from the procedure
        RET
endp

; Procedure to subtract b from a
    sub1 PROC NEAR
        mov al,a        ; Move the value of a to the AL register
        mov bl,b        ; Move the value of b to the BL register
        ; Subtract the value of BL from AL and store the result in AL
        sub al,bl
        ; Move the result of the subtraction to the res_sub variable
        mov res_sub,al
        ; Return from the procedure
        RET
    endp

    ; Procedure to multiply a and b
    mul1 PROC NEAR
        mov al,a        ; Move the value of a to the AL register
        mov bl,b        ; Move the value of b to the BL register
        ; Multiply the values in AL and BL and store the result in AX
        mul bl
        ;Move the result of the multiplication to the res_mul variable
        mov res_mul,ax
        ; Return from the procedure
        RET
    endp

    ; Procedure to divide a by b
    div1 PROC NEAR
        mov al,a        ; Move the value of a to the AL register
```

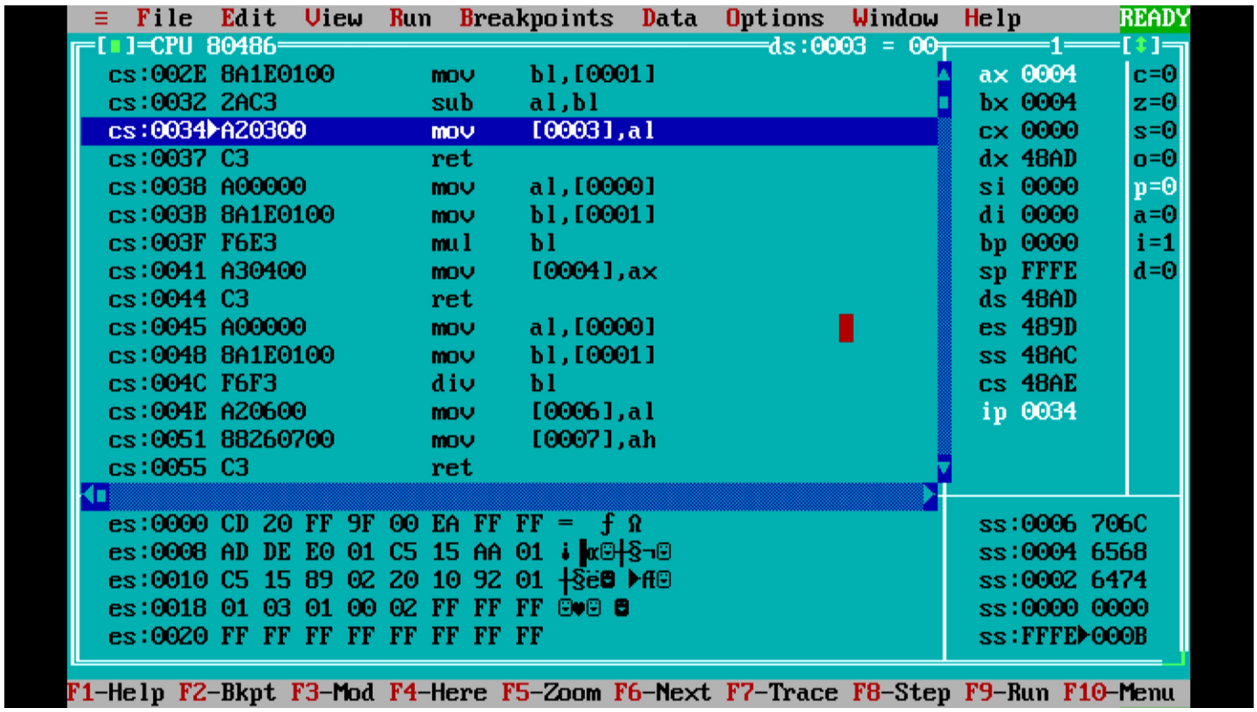
```
                mov bl,b        ; Move the value of b to the BL register
                ; Divide the value of AL by BL and store the quotient in AL and remainder in AH
                div bl
                ; Move the quotient to the res_quo variable
                mov res_quo,al
                ; Move the remainder to the res_rem variable
                mov res_rem,ah
                ; Return from the procedure
                RET
        endp

    ; Procedure to find  square a
        sqr PROC NEAR
                mov al,a       ; Move the value of a to AL
                mov bl,a       ; Move the value of a to BL
                mul bl         ; Multiply the values of AL and BL
                mov res_sqr,ax  ; Move the result to the variable res_sqr
                RET            ; Return from the procedure
        endp

    ; Procedure to find cube b
    cube1 PROC NEAR
       mov al,b  ; Move the value of b to AL
       mov bl,b  ; Move the value of b to BL
       mul bl                ; Multiply the values of AL and BL
       mul bl                ; Multiply the values of AL and BL
       mov res_cube,ax ; Move the result to the variable res_cube
       RET                           ; Return from the procedure
    endp

    ; Procedure to calculate the factorial of a number
    FACTORIAL PROC NEAR
       mov al, 1           ; Initialize result to 1
       mov bl, b           ; Start with input value
       FACT_LOOP:          ; Loop to perform multiplication
          imul bl         ; Multiply by current value of b
          dec bl          ; Decrement b
          jnz FACT_LOOP   ; Continue until b reaches 0
       mov res_fact, ax    ; Store result in res_fact
       RET
    ENDP

; End of code segment
Code ends
end Start
```

❖ **OUTPUT-**

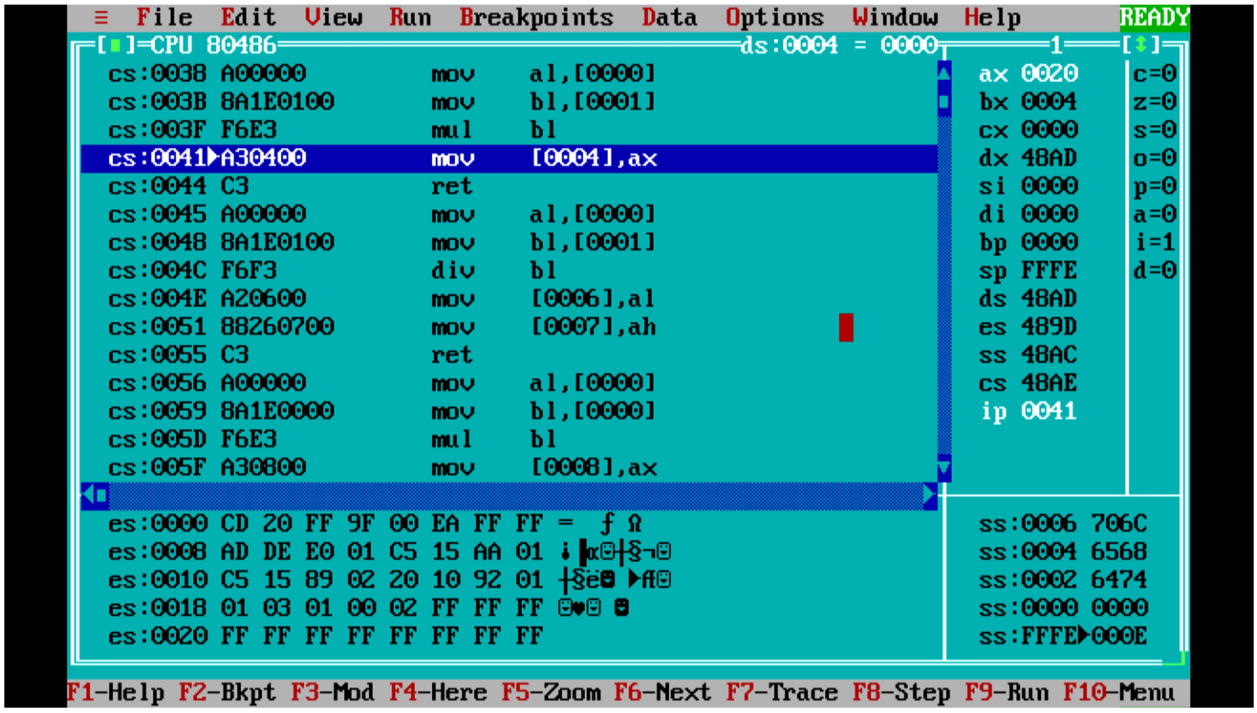
- **After Addition Result-**



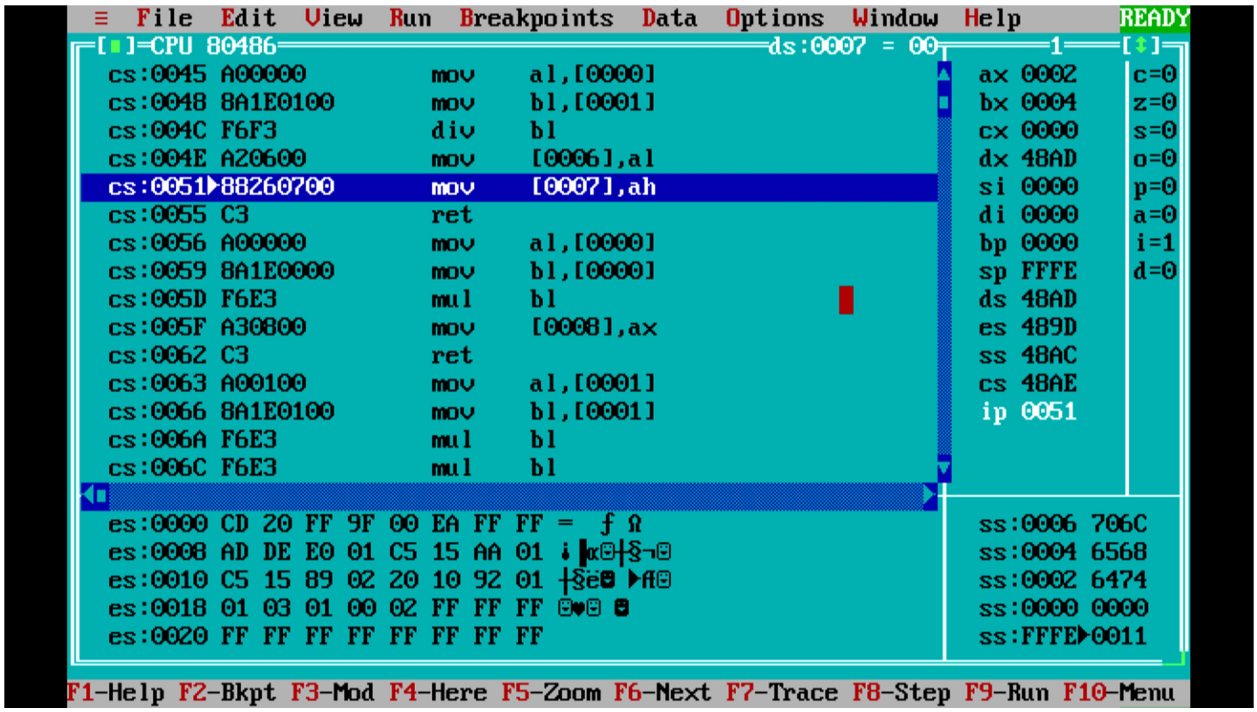
- **After Subtraction Result-**



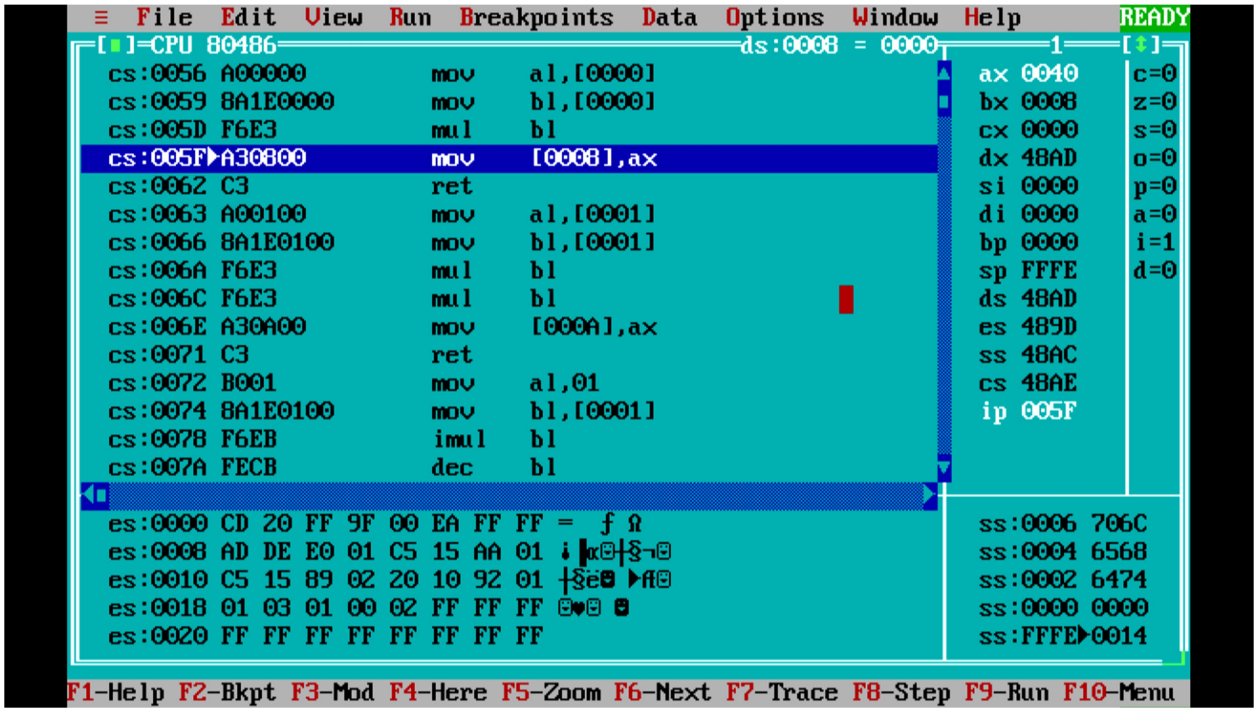
- **After Multiplication Result-**



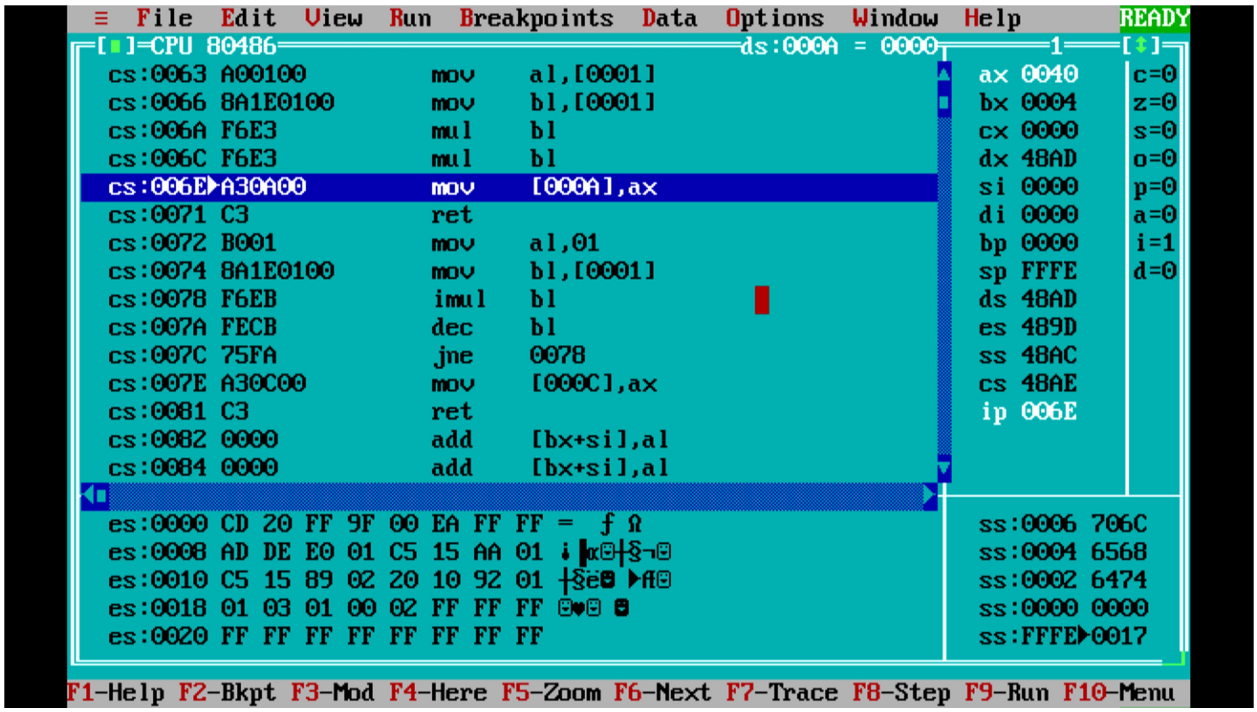
- **After Division Result-**



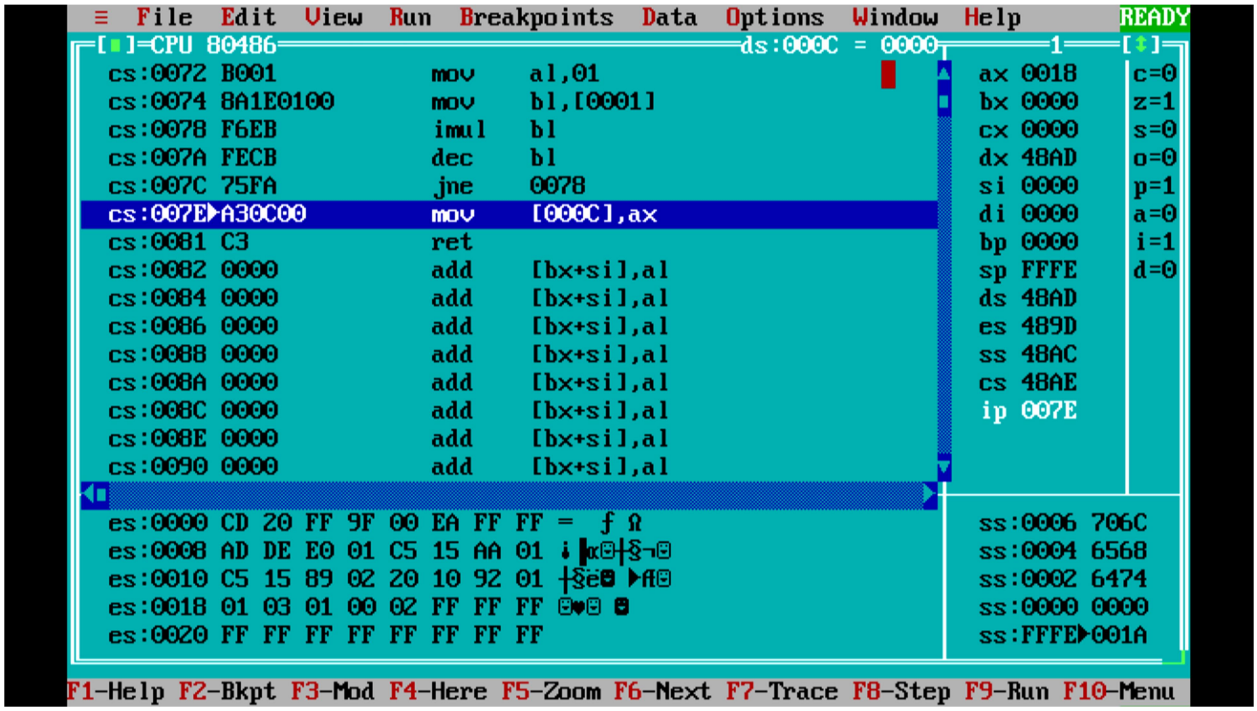
- **After Finding Square Result-**



- **After Finding Cube Result-**



- **After Finding Factorial Result-**



9.0Skill Developed / Learning outcome of this Micro-Project:

Technical Skills:

a) Understanding of 8086 microprocessor architecture
b) Knowledge of Assembly Language Programming
c) Knowledge of arithmetic operations
d) Understanding the procedures
e) Debugging skills
f) Knowledge of software tools

Soft Skills:

a) Leadership and decision making.
b) Working in team.
c) Time management.
d) Presentation and writing skills.

10.0 Applications of Micro Project:

1. Simple calculator
2. Data analysis
3. Embedded systems
4. Gaming
5. Educational purposes

11.0 Name of Group Members:

Enrolment No.	Roll No.	Seat No.	Name of Students	Student Signature
2100780077	06	385977	Baldota Kalash Sachin	
2100780081	10	385981	Bhalerao Mukunda Chandrashekhar	
2100780086	15	385986	Dhakane Aditya Arun	

Date: 27/03/2023

Evaluated by: Dated Signature of Guide: _____

Name of Guide: Ms. P.A.Agrawal