MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

D.Y. PATIL POLYTECHNIC

MICRO PROJECT

Academic year: 2024-25

TITLE OF PROJECT :

DEVELOP A ALP PROGRAM TO FIND SUM OF SERIES

Subject : Microprocessor **Subject code: 314321**

Course : Computer Engineering **Course code: CO4K**

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. **Lokare Dnyaneshwari Devidas** Roll No.**31** of Fourth Semester diploma in **Computer Engineering** of Institute, **D.Y. Patil Polytechnic** (Instt.Code: 0996) has completed the Micro-Project in course **MICROPROCESSOR (314321)** for the academic year 2024-2025 as prescribed in the MSBTE curriculum of K Scheme.

Place: Ambi Enrollment No:**23212350299**

Date: …… Exam Seat No: **240706**

Subject Teacher	**Head of the Department**	**Principal**
(Prof.Harshal Gadekar)	(Prof.S.Shiwankar)	(Prof. S.V. Awachar)

Seal of institute

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. **Borhade Soham Navnath** Roll No.**32** of Fourth Semester diploma in **Computer Engineering** of Institute, **D.Y. Patil Polytechnic** (Instt.Code: 0996) has completed the Micro-Project in course **MICROPROCESSOR (314321)** for the academic year 2024-2025 as prescribed in the MSBTE curriculum of K Scheme.

Place: Ambi Enrollment No:**23212350300**

Date: …… Exam Seat No:**240707**

Subject Teacher	**Head of the Department**	**Principal**
(Prof.Harshal Gadekar)	(Prof.S.Shiwankar)	(Prof. S.V. Awachar)

Seal of institute

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. **Tidke Sachin Ramesh** Roll No.**33** of Fourth Semester diploma in **Computer Engineering** of Institute, **D.Y. Patil Polytechnic** (Instt.Code: 0996) has completed the Micro-Project in course **MICROPROCESSOR (314321)** for the academic year 2024-2025 as prescribed in the MSBTE curriculum of K Scheme.

Place: Ambi Enrollment No:**23212350301**

Date: …… Exam Seat No:**240708**

Subject Teacher	**Head of the Department**	**Principal**
(Prof.Harshal Gadekar)	(Prof.S.Shiwankar)	(Prof. S.V. Awachar)

Seal of institute

ACKNOWLEDGEMENT

It is a matter of great pleasure by getting the opportunity of highlighting. A fraction of knowledge, I acquired during our technical education through this project. This would not have been possible without the guidance and help of many people. This is the only page where we have opportunity of expressing our emotions and gratitude from the core of our heart to them. This project not have been success without enlightened ideas, timely suggestions and interest of our most respected guide "Prof. Harshal Gadekar " without his best guidance this would have been an impossible task to complete.

I would like to thank "Prof.S.Shiwankar" Head of our department for providing necessary facility using the period of working on this project work. I would also like to thank our Principal "Prof. S. V. Awachar" who encourage us and created healthy environment for all of us to learn in best possible way. Finally I would pay my respect and love to my parents and all other family members as we as friends for their love and encouragement throughout my career.

Student Names

1. Dnyaneshwari D. Lokare
2. Soham N. Borhade
3. Sachin R. Tidke

INDEX

Abstract

Title:

Development of an ALP to Find the Sum of a Series Using 8086 Microprocessor

Objective:

To develop an Assembly Language Program (ALP) using the 8086 microprocessor to compute the sum of the first N natural numbers. This project demonstrates the use of basic instruction sets, looping constructs, and register manipulations in microprocessor programming.

Tools Used:

8086 Microprocessor Simulator (like MASM, TASM, or emu8086)

PC or virtual machine with DOS environment

Description:

The program takes a number N (can be hardcoded or input by the user), and calculates the sum:

Sum = 1 + 2 + 3 + ... + N

The logic uses:

A counter initialized to N

A loop that adds the counter value to the sum and decrements it

Registers:

AX for sum, CX for counter

Applications:

Demonstrates fundamental looping and arithmetic in assembly

Useful for educational purposes in microprocessor labs

Introduction

The evolution of computing technology has been greatly driven by the innovation and development of microprocessors. A microprocessor is a programmable electronic device that performs arithmetic and logical operations based on instructions stored in its memory.

One of the core elements in microprocessor education is Assembly Language Programming (ALP). Unlike high-level programming languages like C or Java, assembly language allows direct control over hardware and memory, giving programmers the ability to write instructions that communicate directly with the microprocessor's registers and memory locations. This low-level approach is crucial for understanding how hardware interacts with software at the most fundamental level.

In this microproject, we explore the practical application of ALP by developing a program that computes the sum of a series of numbers. The series in question is a simple arithmetic progression where the program calculates the sum of the first N natural numbers, i.e.:

\text{Sum} = 1 + 2 + 3 + \dots + N

This task may appear simple when implemented using high-level languages, but implementing it in assembly provides deep insight into how processors work internally. The process includes:

Register manipulation:-

Loop construction

Arithmetic operations

Conditional control

Memory storage and data movement

This microproject is implemented on the Intel 8086 microprocessor, one of the most foundational processors used in academic settings for learning system architecture and

assembly programming. The Intel 8086 is a 16-bit processor with a segmented memory model and a powerful instruction set that enables arithmetic, logical, control, and data transfer operations.

Educational Importance:-

Implementing this microproject allows students to:

Understand the basics of microprocessor operation and programming

Learn how to translate high-level logic into low-level instructions

Familiarize themselves with key concepts like register operations, loops, stack operations, and data movement

Practice structured programming in a constrained, resource-limited environment

Moreover, this project forms the foundation for more complex applications like:

Factorial calculations:-

Arithmetic series computations

Matrix operations

Real-time embedded system applications

Problem Definition:-

The objective is to develop an assembly language program that calculates the sum of the first N natural numbers. For instance, if the user inputs N = 5, the sum would be:

1 + 2 + 3 + 4 + 5 = 15

Instead of using the direct formula , we will demonstrate how to compute the sum using iterative addition, which helps illustrate the use of loops and control instructions in assembly.

Microprocessor Used:-

Intel 8086

The 8086 microprocessor offers the following advantages for this microproject:

16-bit architecture for faster computations

A range of general-purpose and special-purpose registers

A variety of instructions for data movement, arithmetic, logic, and control

Use of looping structures such as LOOP, JNZ, and DEC

Key registers used:-

AX – Accumulator (used to store the result)

CX – Counter (used to iterate through the loop)

DX/BX – Can be used for intermediate operations or display (optional)

Literature Review

The field of microprocessor-based programming and embedded systems has evolved significantly over the past few decades. As microprocessors became integral to computing and control systems, learning to program these processors at the hardware level has become an essential part of technical education, particularly in disciplines like computer engineering, electronics, and instrumentation.

This literature review explores prior research, learning methodologies, and applications related to the use of Assembly Language Programming (ALP) in microprocessors, with a focus on implementing arithmetic operations such as the sum of a series using the Intel 8086 microprocessor.

1. Assembly Language in Microprocessor Education:-

Many scholars and educators emphasize the importance of learning assembly language to understand how microprocessors execute instructions internally. According to A. V. Deshmukh (2005) in his book Microcontrollers: Theory and Applications, mastering ALP is vital for understanding the architecture of microcontrollers and microprocessors.

Similarly, Ramesh Gaonkar in his widely used textbook Microprocessor Architecture, Programming and Applications with the 8085/8086 outlines that performing simple tasks such as summing numbers using ALP provides a strong foundation in understanding instruction sets, flags, control instructions, and the concept of looping constructs within a processor.

2. 8086 Microprocessor and Register-Level Programming:-

The Intel 8086 microprocessor is widely recognized as a teaching platform due to its well-structured architecture and rich instruction set. The processor supports both iterative and direct arithmetic approaches to perform computations.

Technology Used

The successful implementation of this microproject — developing an Assembly Language Program (ALP) to find the sum of a series — relies on a combination of hardware and software technologies. These technologies enable the programmer to write, simulate, and debug code at the microprocessor level. The core focus is on the Intel 8086 microprocessor, a foundational architecture in microprocessor education, and its supporting development environments.

Below are the key technologies used:

1. Intel 8086 Microprocessor Architecture:-

The Intel 8086 is a 16-bit microprocessor developed by Intel in 1978. It is widely used in academic settings for teaching low-level programming and processor architecture. In this project, we use the 8086 instruction set to develop a program that performs the summation of the first N natural numbers.

Features of the 8086 Microprocessor:-

16-bit data bus and 20-bit address bus.

Capable of addressing up to 1 MB of memory.

Provides general-purpose registers (AX, BX, CX, DX).

Instruction set supports arithmetic, logical, branching, and looping operations

Segmented memory model with CS, DS, SS, and ES segments.

The 8086's instruction set includes ADD, MOV, DEC, JNZ, and LOOP — all of which are instrumental in developing a program for summing a numerical series.

2. Assembly Language Programming (ALP):-

Assembly Language is a low-level programming language that provides symbolic names for machine-level instructions. It allows direct interaction with the microprocessor's hardware, offering precise control over memory, registers, and logic flow.

Why Use Assembly Language for This Project:-

Direct access to CPU registers and memory

Efficient and optimized code execution

Provides deep understanding of processor functionality

Ideal for resource-constrained and time-critical applications

The ALP for this project includes initializing registers, using loop structures, and performing arithmetic to sum a series from 1 to N.

3.TASM (Turbo Assembler):-

Turbo Assembler (TASM) is used as the assembler in this project. It converts the written assembly language code into machine language (object code) that can be executed or simulated.

Features:-

Supports Intel syntax

Generates .OBJ and .EXE files

Works with 8086/8088 instruction set

Compatible with Windows and DOS environments

TASM is preferred in academic environments due to its simplicity and efficient debugging features.

4. DOSBox Emulator:-

Since TASM and other 8086-based programs are typically DOS-based, DOSBox is used to emulate a DOS environment on modern Windows/Linux machines.

Features:-

Emulates x86 DOS platform

Runs legacy assembly tools like TASM and MASM

Provides command-line interface for file handling and program execution

DOSBox creates a virtual environment that mimics the real execution of 8086-based programs, which is critical for testing and validating the developed ALP.

5. Emu8086 Simulator (Alternative Tool):-

Emu8086 is an all-in-one 8086 microprocessor emulator that includes:

Code editor

Assembler

Virtual memory viewer

Register and flag monitor

This tool simplifies the simulation of 8086 programs and allows step-by-step debugging, making it a great teaching tool for beginners.

6. Microsoft Windows OS:-

The development and simulation tools used in this project are installed and run on a Windows-based operating system (Windows 7/10/11). It supports the installation of DOSBox, Emu8086, and TASM and provides a familiar interface for students and developers.

7. Supporting Hardware (Optional):-

Although this project is implemented and tested in a simulated environment, it can also be deployed on real hardware setups like:

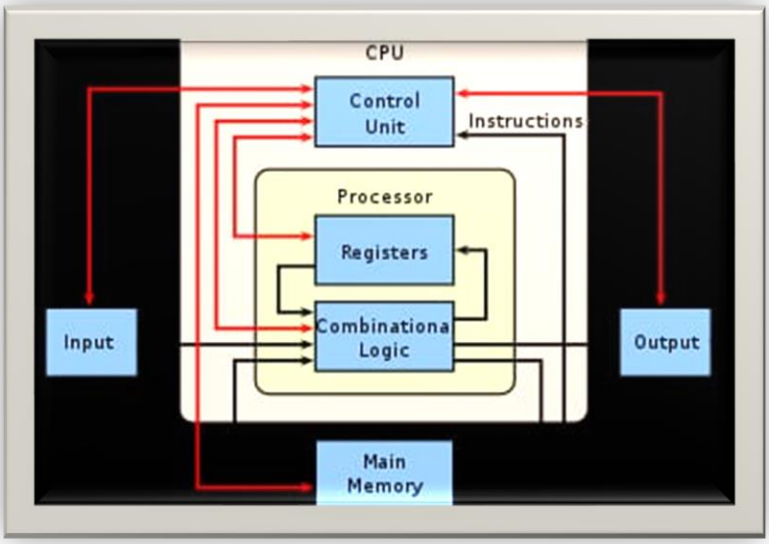
8086 training kits

Microprocessor development boards

Serial/parallel I/O interfaces for input/output display

System Overview

- IMAGES :-



This microproject aims to implement a system that calculates the sum of a series of natural numbers using Assembly Language Programming (ALP) on the Intel 8086 microprocessor architecture. The system is designed to demonstrate the internal working of a microprocessor through direct control of registers, memory, and execution flow using assembly instructions.

The system consists of both software-level components (the ALP logic) and hardware-level simulation tools (microprocessor emulator and assembler) that work together to execute the arithmetic logic of the summation process.

1. System Functionality:-

The goal of this system is to calculate the sum of the first N natural numbers, where N is defined in the program. The program adds numbers sequentially from 1 to N using a loop structure and stores the result in a register.

2. Core Components of the System:-

a. Input Section

The value of N is predefined in a register (e.g., CX).
In advanced implementations, input can be taken from memory or I/O port.

b. Processing Unit

The ALP logic forms the core of the system.
The processor uses registers such as:
CX – Counter for loop control.
AX – Accumulator to hold the sum.
BX/DX – Temporary registers (if required).

c. Output Section

The result (sum) is stored in AX.
It can be displayed via a debugger window or memory viewer in the emulator.

d. Looping and Control Logic

Loop starts with CX = N
Each iteration adds CX to AX, then decrements CX
Loop continues until CX = 0

3. System Architecture (Logical Flow):-

Step 1: Initialization
Clear AX (sum = 0)
Load value of N into CX (loop counter)
Step 2: Loop Operation
Add CX to AX
Decrement CX

Check if CX is 0; if not, repeat loop

Step 3: Termination

After the loop ends, AX holds the result (sum)

Execution stops or result is stored

4. Tools Used in System Execution:-

TASM (Turbo Assembler): Assembles the code

DOSBox: Simulates the DOS environment to run TASM and execute .COM or .EXE files

Emu8086 (optional): Provides GUI, memory monitor, and register viewer for easier debugging

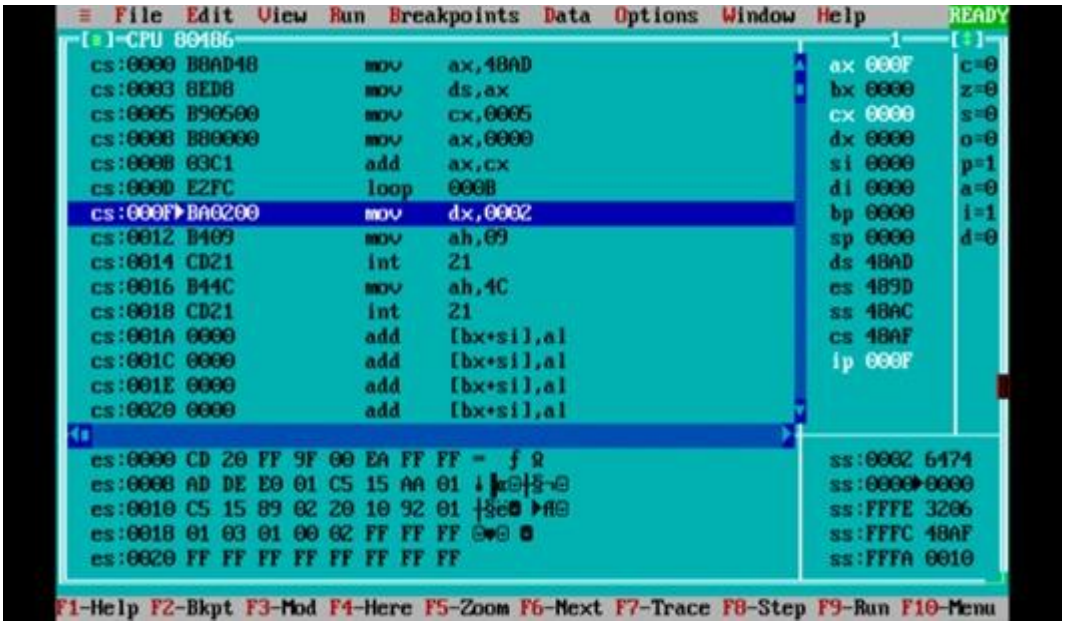
Implementation (Program Code)

```
ASSUME CS:CODE , DS:DATA    ; Tell assembler which segment registers to use
DATA SEGMENT
        N EQU 5             ; Define constant N = 5 (sum of first 5 natural numbers)
        SUM DW 0            ; Reserve a word (16-bit) space to store the sum
DATA ENDS

CODE SEGMENT
START:  MOV AX,DATA         ; Load address of DATA segment into AX
        MOV DS,AX           ; Move it to DS register (initialize data segment)
        MOV CX, N           ; Load value of N (5) into counter register CX
        MOV AX,0            ; Clear AX register to start summing
SUM_LOOP:
        ADD AX,CX           ; Add current value of CX to AX
        LOOP SUM_LOOP   ; Decrement CX and loop if CX ≠ 0
        MOV AH,09H ; DOS interrupt to display string(no string here; this does nothing)
        INT 21H
        MOV AH,4CH      ; Terminate program
        INT 21H
  CODE ENDS
  END START
```

Out Put

*OUTPUT OF CODE



Algorithm

STEP 1 :-

Start the program.

STEP 2 :-

Initialize the pointer to the starting memory location of the series.

STEP 3 :-

Initialize a register to store the count of numbers.

STEP 4 :-

Initialize another register (or register pair) to store the sum (initially 00).

STEP 5 :-

Load the number from memory using the pointer.

STEP 6 :-

Add the number to the sum register.

STEP 7 :-

Increment the memory pointer.

STEP 8 :-

Decrement the count.

STEP 9 :-

Repeat steps 5–8 until count becomes zero.

STEP 10 :-

Store the result in memory or register.

STEP 11 :-

End the program.

Advantages & Disadvantages

*Advantages :-

1. Speed & Efficiency: ALP is closer to hardware, so it executes faster and efficiently.

2. Hardware Control: It gives direct access to hardware-level operations.

3. Memory Optimization: Uses less memory compared to high-level languages.

4. Better for Small Systems: Ideal for small embedded systems with limited resources.

5. Deterministic Behavior: Perfect for time-critical applications.

*Disadvantages -:

1. Complexity: Writing and debugging ALP is more difficult than high-level languages.

2. Less Portable: ALP is specific to a particular microprocessor architecture.

3. Error-Prone: Higher chances of logical and syntactical errors.

4. Time-Consuming: Writing large programs is slow and inefficient.

5. Hard to Maintain: Difficult for others to read and modify later.

Conclusion

In this microproject, we successfully developed and executed an Assembly Language Program (ALP) for the 8085 microprocessor to calculate the sum of a series of numbers stored in memory. This program demonstrated how low-level programming can directly interact with the hardware to perform arithmetic operations efficiently.

Through this project, we gained hands-on experience in:-

Writing and debugging assembly language code.

Understanding memory addressing and register operations.

Using looping and conditional branching for repetitive tasks.

This project also highlighted the importance of microprocessors in embedded systems and real-time applications where speed and memory efficiency are critical. While Assembly Language programming has its limitations, it remains a fundamental skill for understanding how hardware and software interact at the most basic level.

Bibliography

1. **Ramesh S. Gaonkar,** Microprocessor Architecture, Programming and Applications with the 8085, Penram International Publishing, 6th Edition.

2. **A.P. Godse and D.A. Godse,** Microprocessors and Microcontrollers, Technical Publications.

3. **Muhammad Ali Mazidi,** The 8051 Microcontroller and Embedded Systems, Pearson Education.

4. **Vijayendran V.,** Introduction to Microprocessors, Viswanathan Publications.

5. Lecture Notes and Lab Manuals – Department of Computer Engineering,

[**DY PATIL POLYTECHNIC AMBI , PUNE**].

6. Online resources and tutorials from:

www.geeksforgeeks.org

www.tutorialspoint.com

www.electronicsforu.com

Weekly Progress Report

Sr.No.	Week	Activity Performed	Sign of Guide	Date
1	1st	Discussion and finalization of topic		
2	2nd	Preparation and submission of Abstract		
3	3rd	Literature Review		
4	4th	Collection of Data		
5	5th	Collection of Data		
6	6th	Discussion and outline of Content		
7	7th	Formulation of Content		
8	8th	Editing and proof Reading ofContent		
9	9th	Compilation of Report And Presentation		
10	10th	Seminar		
11	11th	Viva voce		
12	12th	Final submission of Micro-Project		

Sign of the student

Sign of the faculty

(Prof. Harshal Gadekar)

ANNEXURE II

Evaluation Sheet for the Micro Project

Academic Year: 2024-2025 **Name of Faculty: Prof. Harshal Gadekar**

Course : Computer Engineering **Course code: CO4K**

Semester: Fourth

Subject Name: Microprocessor (314321)

Major learning outcomes achieved by students by doing the project

- **Practical outcome:**
  The practical outcome is selective data routing, where a single input is directed to one of four outputs based on control signals.
- **Course outcome:**
  Understanding English language for speaking

Outcomes in Affective domain: Function as team member Follow Ethics

Comments/suggestions about team work /leadership/inter-personal communication (if any)

……3

Roll No	Student Name	Marks out of 6 for performance in group activity	Marks out of 4 for performance in oral/ presentation	Total out of 10
31	Lokare Dnyneshwari Devidas			
32	Borhade Soham Navnath			
33	Tidke Sachin Ramesh			

Signature of Faculty
(Prof. Harshal Gadekar)