

# MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION (MUMBAI)

**Academic Year 2024 - 2025**

**MICRO-PROJECT PROPOSAL**

“Programs On All Addressing Modes of 8086 MIC”

***Program Name: Computer Engineering Semester: 4K* Course Title: Microprocessor Programming(MIC) CourseCode:314329**

**Submitted by,**

**Name of Member:Agarkhed.S.H EnrollNo:23212620142**

**Name of Member:Kurhade.S.H EnrollNo:23212620132**

**Name of Member:Nakil.S.S EnrollNo:23212620153**

**Name of Member:Sangle.V.V EnrollNo:23212620133**

**Under the guidance of,**

# (Prof Mr. Malge M.)



### Annexure-I A

**PART A – PLAN**

**“Micro-Project Title:** **Programs On All Addressing Modes of 8086 MIC”**

**1.0 Brief Introduction:**

1. **Immediate Addressing Mode**:
   * The operand is a constant value that is specified directly in the instruction.
   * Example: MOV AX, 5 (Moves the constant value 5 into the AX register).
2. **Register Addressing Mode**:
   * The operand is stored in a register. The instruction uses a register as the source or destination.
   * Example: MOV AX, BX (Moves the value from register BX to AX).
3. **Direct Addressing Mode**:
   * The operand is stored at a specific memory address. The address is specified directly in the instruction.
   * Example: MOV AX, [1234H] (Moves the value stored at memory address 1234H into AX).
4. **Indirect Addressing Mode**:
   * The operand's address is specified by a register. The contents of the register hold the effective memory address.
   * Example: MOV AX, [BX] (Moves the value stored at the memory address pointed to by BX into AX).
5. **Indexed Addressing Mode**:
   * The effective address of the operand is computed by adding an index register (SI or DI) to a constant displacement (offset).
   * Example: MOV AX, [SI+10H] (Moves the value stored at the address SI + 10H into AX).
6. **Based Addressing Mode**:
   * The effective address is calculated by adding a base register (BX or BP) to a displacement.
   * Example: MOV AX, [BX+5] (Moves the value stored at the address BX + 5 into AX).
7. **Based-Indexed Addressing Mode**:
   * This mode combines a base register (BX or BP) and an index register (SI or DI) with an optional displacement.
   * Example: MOV AX, [BX+SI+10H] (Moves the value from the address BX + SI + 10H intoAX.
8. **Relative Addressing Mode**:
   * This mode is used in branch instructions, where the address is a displacement relative to the current instruction pointer (IP).
   * Example: JMP LABEL (Jumps to the address of LABEL, which is calculated relative to the current instruction pointer).

**2.0 Aim of the Micro-Project:**

## This Micro-Project aims at:

* The main objective of the micro-project is “**Programs On All Addressing Modes of 8086 MIC**”.

1. **Flexibility in Data Access**:
   * Addressing modes allow the 8086 processor to access operands in various ways, making it versatile in how data is retrieved or stored. This flexibility is essential for performing different types of operations on data located in registers, memory, or both.
2. **Efficient Memory Utilization**:
   * By supporting modes like indirect, indexed, and based addressing, the 8086 can access large amounts of data efficiently, minimizing the need for hardcoding specific memory addresses. This allows the processor to access data dynamically and makes better use of available memory.
3. **Code Compactness**:
   * Addressing modes help in reducing the size of instructions by enabling the use of registers or pointers to refer to memory locations, rather than explicitly specifying full memory addresses in every instruction. This results in shorter, more efficient code.
4. **Support for Complex Operations**:
   * Some addressing modes, like based-indexed or indirect, support complex operations like accessing arrays, tables, and data structures. This is particularly useful in tasks like looping through arrays or manipulating data that requires dynamic address computation.
5. **Improved Program Modularity and Portability**:
   * Using flexible addressing modes, programs can be more modular. For example, programs can access different parts of memory or registers without having to change the code each time. This aids in writing reusable code and enhances portability across different memory layouts or configurations.
6. **Efficient Execution of Control Flow**:
   * The **relative addressing mode** allows for more efficient implementation of branch operations, such as jumps or loops, by specifying offsets relative to the current instruction pointer. This reduces the need for absolute memory addresses in branch instructions.

Overall, the addressing modes in the 8086 microprocessor provide the means for efficient, flexible, and powerful data handling, which is crucial for complex programs and efficient execution.

1. **Action Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.  No. | Details of activity | Planned Start date | Planned Finish date | Name of Responsibl e Team Members. |
| 1 | Searching the topic for the micro-project |  |  | Individual |
| 2 | Discuss about the micro-project |  |  | Individual |
| 3 | Collect the information about the micro- project |  |  | Individual |
| 4 | Information search in internet |  |  | Individual |
| 5 | Making a rough copy of micro-project |  |  | Individual |
| 6 | Arrange all information in MS word |  |  | Individual |
| 7 | Prepare a report on it using MS word |  |  | Individual |
| 8 | Submit the micro-project |  |  | Individual |

1. **Resources Required:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Name of Resource / Material | Specifications | Qty. | Remarks |
| 1 | Operating system | Windows 11 | - | - |
| 2 | Reference Book | 1.Microprocessor 8086: Architecture, Programming and Interfacing" by Sunil Mathur,  2.8086 Microprocessor Bharat Acharya Education: Architecture and Interfacing" by Bharat Acharya | - | - |
| 3 | Website | [Udemy's free Microprocessors Tutorial](https://www.udemy.com/course/8086-microprocessor/)  [NPTEL's Microprocessors and Interfacing course](https://onlinecourses.nptel.ac.in/noc20_ee11/preview) | - | - |
| 4 | Software | EMU8086: [Softonic's EMU8086 emulator](https://emu8086-microprocessor-emulator.en.softonic.com/) | - | - |

1. **Any other:**

### Conclusion:

In conclusion, the addressing modes of the 8086 microprocessor play a crucial role in providing flexibility, efficiency, and versatility in data access. They allow the processor to work with data in various ways, such as directly, indirectly, or through indexed and based methods. These modes enable the 8086 to:

* **Access data dynamically** from registers, memory, or computed addresses, improving the efficiency of code execution.
* **Simplify complex tasks** like handling arrays, loops, and data structures through indexed and based addressing.
* **Enhance program portability** and modularity by enabling indirect references and memory-relative addressing.
* **Minimize code size**, since many addressing modes allow for more compact instructions, especially when working with memory addresses indirectly or through registers.

By offering a variety of addressing modes, the 8086 processor can efficiently handle a wide range of applications, making it a powerful microprocessor for both simple and complex computing tasks. The diverse addressing modes not only optimize memory usage but also facilitate easier development of flexible and efficient software.

### Reference:

* <https://www.grafiati.com/en/literature-selections/intel-8086-microprocessor/>
* <https://www.geeksforgeeks.org/architecture-of-8086/>

***We request you to please accept our Micro-Project Proposal and we assure you of our sincere performance. Hoping for your co-operation & guidance with us.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Name of Student | Roll Number | Course Code | Student Sign. |
| 1  2  3  4 | Agarkhed.S.H  Kurhade S.H  Nakil S.S  SangleV.V | 242232  242222  242239  242223 | 314329  314329  314329  314329 |  |

***……………………. ……………………***

**Micro-Project Guide H.O.D**