DEPARTMENT OF ELECTRICAL ENGINEERING

HARIKISHORE SINGH

GOVERNMENT POLYTECHNIC SHEOHAR

(Department Of Science,Technology & Technical Education)

LAB MANUAL

Microprocessor & microcontroller

SUBJECT CODE :- P2420402

HARIKISHORE SINGH

GOVERNMENT POLYTECHNIC

SHEOHAR (843329)

Department of Electrical Engineering

Vision

Elevating electrical engineers by imparting innovative technical knowledge and skills for ever-changing needs of industries for sustainable development.

Mission

Ml: To provide students with a supportive environment that facilitates knowledge and excellence

in learning through innovative teaching and industrial projects.

M2: Foster leadership, communication, and teamwork abilities to address engineering challenges effectively.

M3: Promote lifelong learning and adaptability to emerging technologies for continuous professional growth and innovation.

Practical/Lab Session Outcomes (LSOs)

S.NO	Laboratory Experiment	Lab Session Outcomes
	Test and verify the features of 8085 Trainer Kit.	1. Examine the 8085 Trainer Kit. 2. Identify the vanous components in 8085 Trainer Kit
2	Write and execute an ALP for 8085 to add two 8- bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	1. Write an assembly language program based on data transfer instructions and arithmetic instructions. 2. Test the results by assembly language program.
3	Write and execute an ALP for 8085 to subtract two 8-bit nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	1. Write an assembly language program based on data transfer instructions and arithmetic instructions. 2. Test the results by executing assembly language program.
4	Write an assembly language program for finding largest / smallest number	1. Write an assembly language program based on arithmetic instructions to find smallest and largest numbers. 2. Test the results by executing assembly language program.

5	Write an assembly language program for arranging numbers in ascending/descending order	1. Write an assembly language program for arranging numbers in ascending/descending order. 2. Test the results by executing assembly language program.
6	Interface 8255 IC with the 8085 microprocessors to perform given input/output operations.	1. Develop an assembly language program to interface 8255 IC with 8051 Microcontroller. 2. Test the results using an assembly language program.
7	Test and verify the features of 8051 Trainer Kit.	1. Examine the 8051 Trainer kit. 2. Identify the vanous components in 8051 Trainer Kit.
8	Write and execute an ALP for 8051 to add two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations	1. Write an assembly language program based on Data transfer Instructions & Arithmetic Instructions. 2. Test the results by executing assembly language program
9	Write and execute an ALP for 8051 to Subtract two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations	1. Write an assembly language program based on Data transfer Instructions & Arithmetic Instructions. 2. Test the results by executing assembly language program

10	Write an ALP to generate delay, using Timer	1. Develop an assembly language program to generate delay using timer in 8051 Microcontroller. 2. Test the results by executing assembly language program
11	Write an ALP to generate delay, using Timer.	1. Develop an assembly language program to interface 7 segment display with 8051 Microcontroller. 2. Test the results using an assembly language program
12	Interface a DC Motor with 8051.	I.Test the result using an assembly language program
13	Develop a program to interface a Stepper Motor with 8051.	I .Test the results by executing an assembly language program to interface Stepper Motor with 8051.

LIST OF EXPERIMENTS

1	Test and verify the features of 8085 Trainer Kit.
2	Write and execute an ALP for 8085 to add two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations
3	Write and execute an ALP for 8085 to Subtract two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.
4	Write an assembly language program for finding largest/ smallest number.
5	Write an assembly language program for arrangmg numbers m ascending/descending order
6	Interface 8255 IC with the 8085 microprocessors to perform given input/output operations.
7	Test and verify the features of 8051 Trainer Kit.
8	Write and execute an ALP for 8051 to add two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.
9	Write and execute an ALP for 8051 to Subtract two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations
10	Write an ALP to generate delay, using Timer.
11	Develop a program to interface 7 segment display with 8051.

12	Develop a program to interface a DC Motor with 8051.
13	Develop a program to interface a Stepper Motor with 8051.

PRACTICAL OUTCOMES

CO	Statement
1	Analyze the architecture of Microprocessor IC 8085.
2	Develop the assembly language programs for various operations using instruction set of 8085 Microprocessor.
3	Interface the memory and I/O devices to 8085 Microprocessor
4	Analyze the architecture of Microcontroller IC 8051.
5	Develop the assembly language programs for various operations using instruction set of 8051 Microcontroller.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO 1 - To contribute the industry and society by applying the skills and knowledge acquired during the program period

PEO 2 - To inculcate a sense of ethics, professionalism and effective communication skills with the attitude of continuous learning.

PEO 3 - Become entrepreneurs, pursue higher education, and actively contribute to the socio-economic growth and sustainable development of the nation.

PROGRAM SPECIFIC OUTCOMES

PSO1: Demonstrate proficiency in use of modern electrical drives, Embedded systems, IOT and automation of systems to practice electrical engineering profession.

PSO2: Simulate interdisciplinary projects with emphasis on testing, performance evaluation, energy efficiency and sustainable solutions for industrial applications.

PROGRAM OUTCOMES (POS)

PO1: Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

P02: Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

P03: Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

P04: Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

P05: Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.

P06: Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

P07: Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes.

EXPERIMENT NO: 01

Aim

To study and verify the basic features and operations of the 8085 trainer kit. To understand the working of memory, keypad, display, and execution of programs.

Theory

The 8085 trainer kit is a microprocessor-based development system used for learning and testing programs of the 8085 microprocessor. It consists of the 8085 CPU, memory units (RAM and ROM), input/output ports, a hexadecimal keypad, and a display unit (usually LED or LCD).

The trainer kit allows users to enter machine-level programs using the keypad and observe the output on the display. It operates on the principle of stored program concept, where instructions and data are stored in memory and executed sequentially by the microprocessor.

Important features of the 8085 trainer kit include:

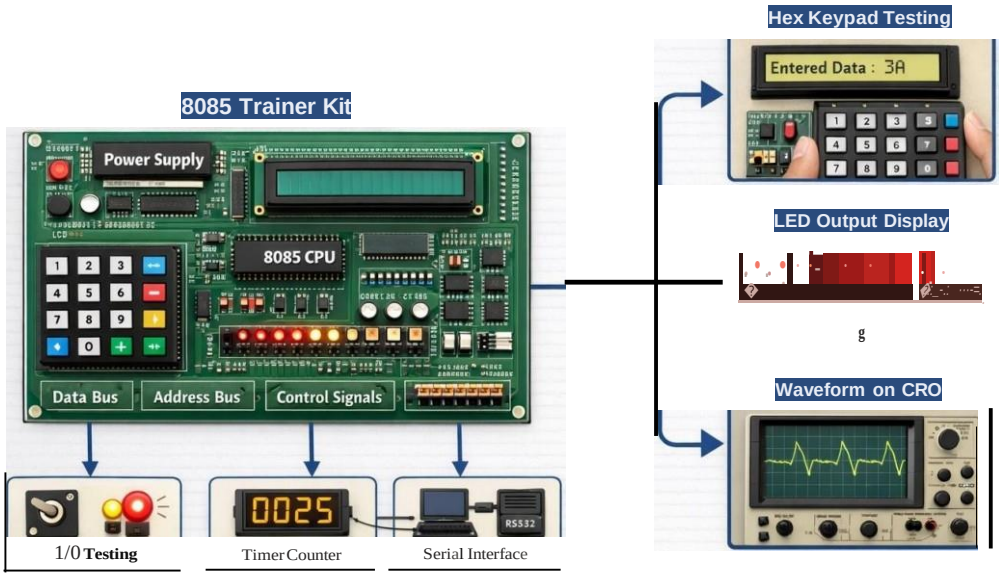
- Memory Read/Write: Allows storing and retrieving data from memory locations.
- Program Execution: Executes instructions stored in memory.
- Register Operations: Displays and modifies contents of registers.
- Single Step Execution: Executes one instruction at a time for debugging.
- Reset Function: Initializes the system.

The microprocessor uses an address bus (16-bit) to access memory locations and a data bus (8-bit) to transfer data. Control signals manage operations like read, write, and execute.

Practical Set-up/ Circuit Diagram

The 8085 trainer kit consists of a built-in microprocessor system mounted on a board. It includes a power supply, keypad for input, display for output, and memory chips connected internally. The user interacts with the system through the keypad to enter addresses and data, while the display shows memory contents and execution results. No external wiring is required as all components are pre-connected on the kit.

Test and Verify the Features of 8085 Trainer Kit



Apparatus Required

8085 Trainer Kit, Power supply cable, User manual

Precautions

1. Ensure proper power supply before switching on the kit
2. Do not press multiple keys simultaneously
3. Enter correct memory addresses and data carefully
4. Avoid switching off power during program execution
5. Handle the trainer kit gently to avoid damage
6. Always reset the system before starting a new program

Procedure

1. Switch on the 8085 trainer kit

2. Press the RESET key to initialize the system
3. Enter the starting memory address using the keypad
4. Use the WRITE or MEMORY key to store data at the specified address
5. Enter machine code instructions step by step
6. Verify the entered data using the READ or DISPLAY function
7. Execute the program using the RUN or EXECUTE key
8. Observe the output on the display
9. Use SINGLE STEP mode to execute instructions one by one (if required)
10. Repeat the process to test different features like register display and memory operations

Observations and Calculations

Observation Table:

IStepNo.I	Memory Address	Data Entered	Data Observed	IRemarksl

Result

The features of the 8085 trainer kit such as memory operation, program execution, and display functions were successfully tested and verified. The trainer kit was found to be working properly as per expected operations.

VIVA BOOK

1. Explain the 8085 trainer kit?
2. Explain the function of memory in the 8085 trainer kit.
3. What is the purpose of the RESET operation?
4. What is single step execution in a microprocessor?
5. Differentiate between RAM and ROM in a microprocessor system.

EXPERIMENT NO: 02

Aim

To write and execute an 8085 assembly language program to add two 8-bit numbers. To store the sum and carry at different memory locations.

Theory

In the 8085 microprocessor, arithmetic operations like addition are performed usmg instructions such as ADD and ADC. When two 8-bit numbers are added, the result may exceed 8 bits. In such cases, the carry is generated and stored in the carry flag (CY) of the flag register.

The ADD instruction adds the content of a register or memory to the accumulator. If the result exceeds 255 (FFH), a carry is generated. The carry flag must be checked to store it separately.

Steps involved:

- Load first number into accumulator
- Add second number to accumulator
- Store result in memory
- Check carry flag and store it separately

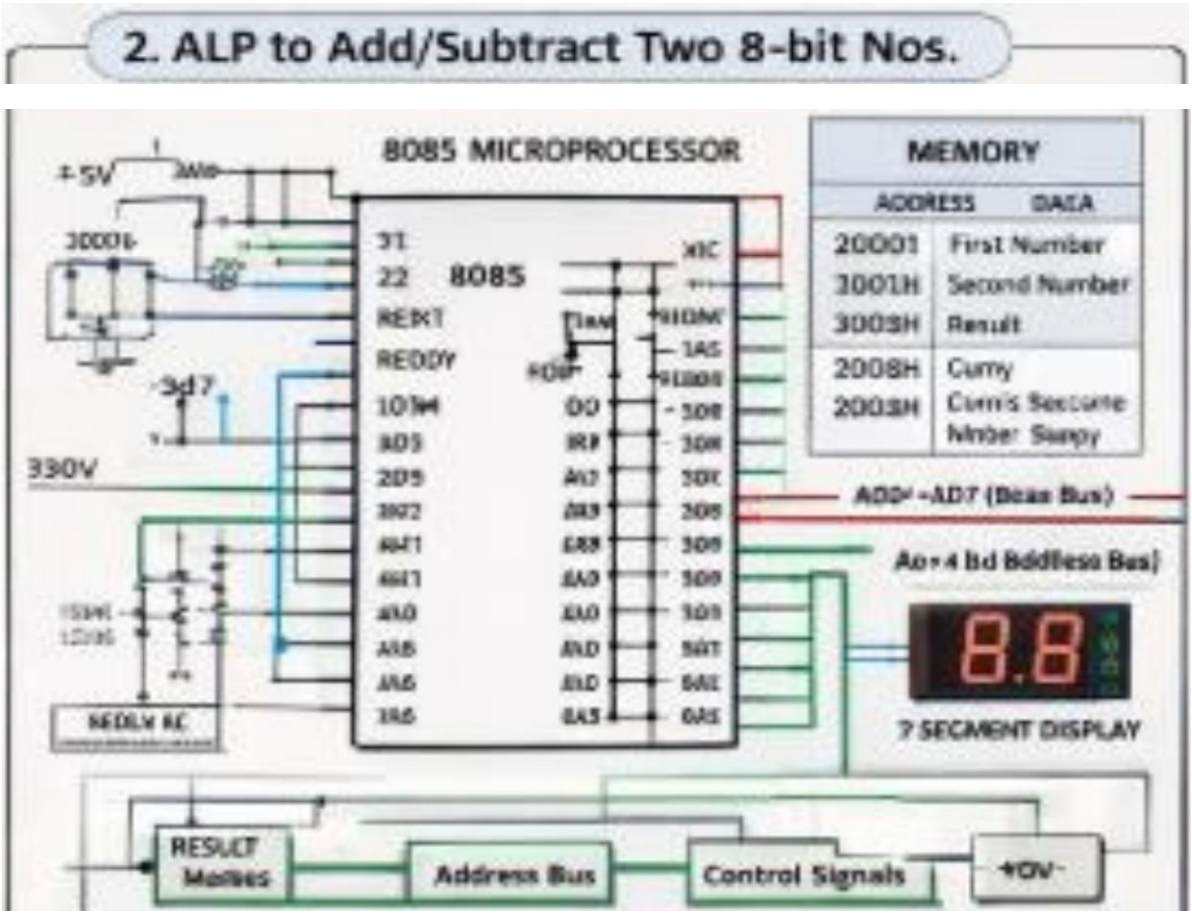
No special formula is required, but:

Sum= First Number+ Second Number

If Sum> FFH, then Carry= 1, else Carry= 0

Practical Set-up/ Circuit Diagram

Use the 8085 trainer kit with built-in memory, keypad, and display. Enter the program using the keypad. Store the two 8-bit numbers in predefined memory locations. Execute the program and observe the result stored in memory along with the carry value.



Apparatus Required

8085 Trainer Kit, Power supply, Connecting leads

Precautions

1. Check power supply before starting
2. Enter correct opcodes and addresses carefully
3. Reset the kit before execution
4. Avoid pressing wrong keys during entry
5. Verify program before running
6. Do not interrupt execution

Procedure

1. Switch on the trainer kit and press RESET
2. Enter the first number at memory location 2000H
3. Enter the second number at memory location 2001H
4. Enter the following program starting from memory location 3000H

Program:

```
LDA2000H
MOVB,A
LDA2001H
ADDB
STA2002H
JNC SKIP
MVIA, OlH
STA2003H
HLT
SKIP: MVI A, OOH
STA2003H
HLT
```

5. Execute the program
6. Check memory location 2002H for sum
7. Check memory location 2003H for carry (OOH or O1H)
8. Repeat with different inputs

Observations and Calculations

Observation Table:

First Number	Second Number	Sum (2002H)	Carry (2003H)

Example:

If 8FH + 92H = 121H---+ Sum= 21H, Carry= 0lH

Result

The program to add two 8-bit numbers was successfully executed. The result and carry were correctly stored in the specified memory locations for both carry and no-carry cases.

VIVA BOOK

1. What is an 8-bit number in microprocessor?
2. Explain the function of ADD instruction in 8085.
3. What is the carry flag and when is it set?
4. What is the difference between ADD and ADC instruction?
5. Why is accumulator used in arithmetic operations in 8085?

EXPERIMENT NO:03

Aim

To write and execute an assembly language program in 8085 microprocessor to subtract two 8-bit numbers stored in memory and store the result along with carry/borrow.

Theory

The 8085 microprocessor is an 8-bit processor capable of performing arithmetic and logical operations. In subtraction, the instruction used is SUB (for register) or SBB (subtract with borrow).

When subtracting two numbers, the result may be positive or negative. In 8085, subtraction is performed using 2's complement method internally. If the result is positive, the Carry flag remains 0. If the result is negative (borrow occurs), the Carry flag becomes 1, indicating a borrow.

The steps involved in subtraction are:

1. Load the first number (minuend) into accumulator.
2. Subtract the second number (subtrahend) using SUB instruction.
3. Store the result in a memory location.
4. Check Carry flag:
    - If CY= 0---+ No borrow (without carry case)
    - If CY= 1 ---+ Borrow occurred (with carry case)

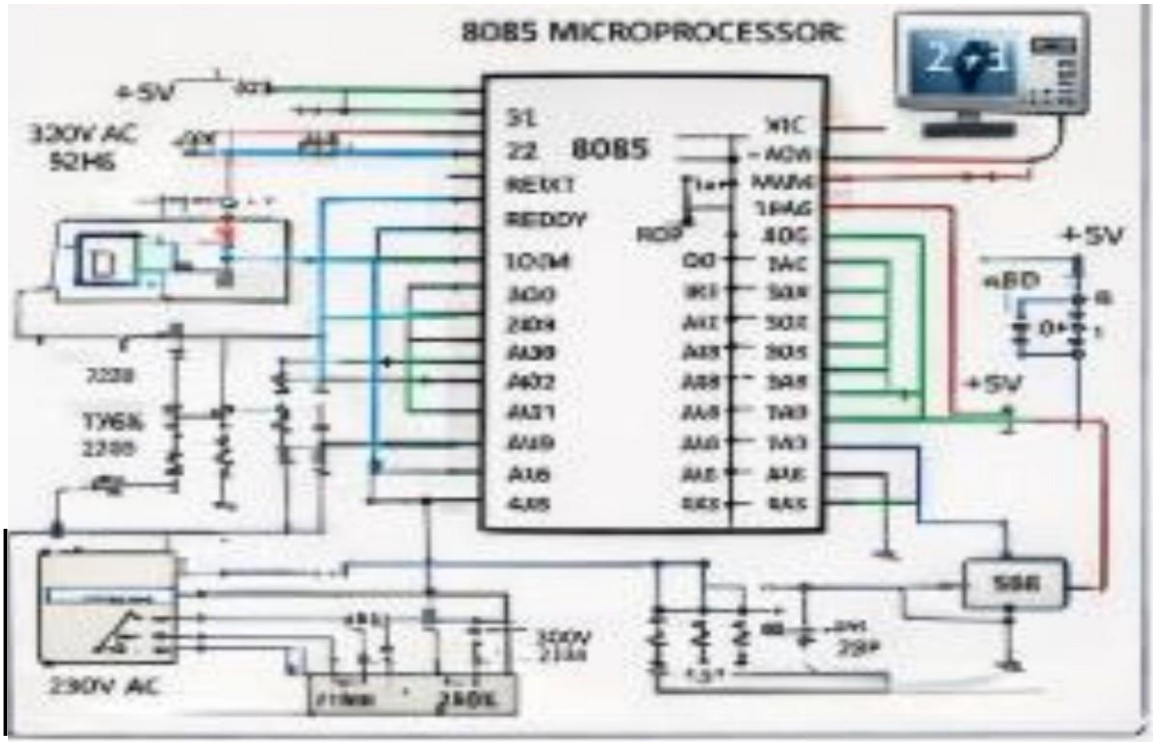
Important flags affected:

- Carry Flag (CY): Indicates borrow
- Zero Flag (Z): Set if result is zero
- Sign Flag (S): Indicates sign of result

Practical Set-up / Circuit Diagram

The experiment is performed using an 8085 microprocessor trainer kit or simulator. The data (two 8-bit numbers) are stored in predefined memory locations. The program is written using hexadecimal opcodes or assembly mnemonics. After execution, the result and carry status are stored in specified memory locations and verified using memory display.

C 3. ALP to Fand Largest/Smallest Number)



Apparatus Required

8085 Microprocessor Trainer Kit/ Simulator, Power Supply, Connecting Wires, Instruction Manual

Precautions

1. Ensure correct power supply is given to the kit.
2. Enter program carefully without errors.
3. Verify memory addresses before execution.
4. Reset the kit before running a new program.

5. Check flag conditions properly after execution.
6. Avoid loose connections in hardware setup.

Procedure

1. Switch ON the 8085 trainer kit.
2. Enter the program in memory using hexadecimal codes or mnemonics.
3. Store first 8-bit number at a memory location (e.g., 2000H).
4. Store second 8-bit number at another memory location (e.g., 2001H).
5. Load the first number into accumulator using LDA instruction.
6. Move the second number into a register (e.g., B register).
7. Perform subtraction using SUB instruction.
8. Store the result in another memory location (e.g., 2002H).
9. Check Carry flag:
    - If CY= 1, store carry/borrow at another location (e.g., 2003H).
    - If CY= 0, store OOH at carry location.
10. Execute the program.
11. Observe and verify the result and carry in memory locations.

Observations and Calculations

Observation Table:

Memory Location Data (Hex) Description

12000H	ID	First number (Minuend)
!:===== :<? .12 00 1 H	.ID	Second number (Subtrahend)

Memory Location Data (Hex) Description

12002H  Result of subtraction

Calculation:

Result= First Number - Second Number

If First Number≥Second Number - No Borrow (CY= 0)

If First Number< Second Number - Borrow (CY= **l)**

Result

The subtraction of two 8-bit numbers usmg 8085 microprocessor was successfully performed. The result was stored correctly in memory, and carry/borrow condition was verified using the Carry flag.

VIVA BOOK

1. What is an 8-bit microprocessor?
2. What is the function of the Carry flag in subtraction?
3. Explain the SUB instruction in 8085.
4. What is the difference between SUB and SBB instructions?
5. What happens when the result of subtraction is negative in 8085?

EXPERIMENT NO:04

Aim:-

To write and execute an 8085 assembly language program to find the largest and smallest number from a given set of numbers stored in memory.

Theory

The 8085 microprocessor is an 8-bit processor that performs arithmetic and logical operations. To find the largest or smallest number from a list, comparison operations are used.

The CMP instruction compares the contents of a register or memory with the accumulator. It internally performs subtraction but does not store the result; it only affects the flags.

Working principle:

1. Load the first number into the accumulator and assume it as largest or smallest.
2. Compare the accumulator with the next number using CMP instruction.
3. Based on the flag status:
    - For largest number:

    If Carry = 0--- ▸accumulator is greater or equal

    If Carry = 1 -- ▸new number is larger, update accumulator

    - For smallest number:

    If Carry = 1 -- ▸accumulator is smaller

    If Carry = 0--- ▸new number is smaller, update accumulator
4. Repeat the process for all numbers in the list.
5. Store the final result in memory.

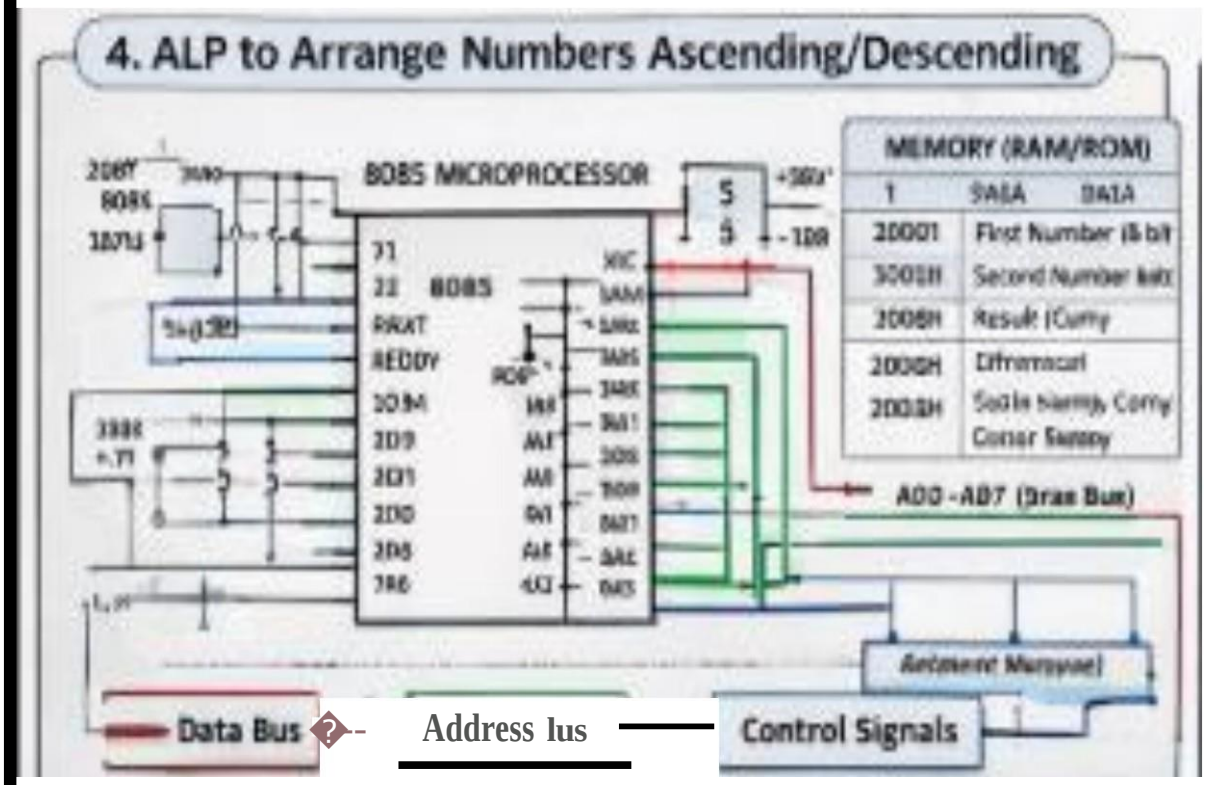
Important flags:

- Carry Flag (CY): Indicates comparison result

- Zero Flag (Z): Indicates equal values

Practical Set-up / Circuit Diagram

The experiment is performed using an 8085 microprocessor trainer kit or simulator. The input data (count and numbers) are stored in memory locations. The program is entered using keypad or software. After execution, the largest and smallest numbers are stored in specified memory locations and verified using memory display.



Apparatus Required

8085 Microprocessor Trainer Kit/ Simulator, Power Supply, Connecting Wires, Instruction Manual

Precautions

1. Ensure proper power supply before starting.

2. Enter the program carefully without mistakes.
3. Check memory addresses correctly.
4. Reset the kit before execution.
5. Verify flag conditions after comparison.
6. Avoid loose connections in hardware setup.

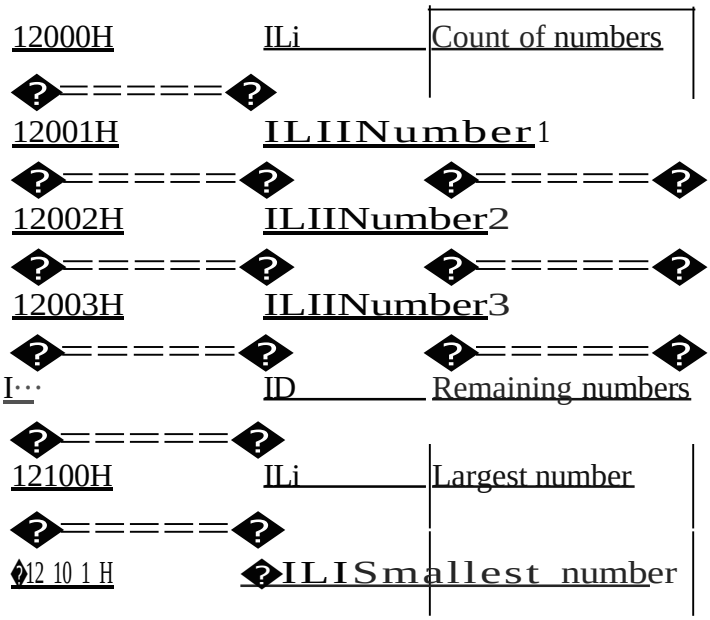
Procedure

1. Switch ON the 8085 trainer kit.
2. Enter the program into memory.
3. Store the count of numbers at memory location (e.g., 2000H).
4. Store the list of numbers starting from next location (e.g., 2001H onwards).
5. Load the first number into accumulator.
6. Initialize register for loop counter.
7. Compare accumulator with next number using CMP instruction.
8. Update accumulator if required based on carry condition.
9. Repeat the comparison for all numbers using loop.
10. Store the largest number in memory location (e.g., 2100H).
11. Repeat similar steps to find smallest number and store at another location (e.g., 2101H).
12. Execute the program.
13. Observe and verify the results in memory.

Observations and Calculations

Observation Table:

Memory Location Data (Hex) Description



Calculation:

Largest Number= Maximum value among given numbers

Smallest Number= Minimum value among given numbers

Result

The assembly language program was successfully executed to find the largest and smallest numbers from the given data. The results were stored correctly in the specified memory locations.

VIVA BOOK

1. Explain the function of CMP instruction in 8085?
2. What is the role of Carry flag in comparison?
3. Explain how largest number is determined using 8085.
4. Differentiate between finding largest and smallest number logic?
5. What happens to flags after CMP instruction?

EXPERIMENT N0:5

Aim:-

To write and execute an 8085 assembly language program to arrange a given set of numbers in ascending and descending order.

Theory

Sorting is the process of arranging data in a specific order such as ascending (smallest to largest) or descending (largest to smallest). In 8085 microprocessor, sorting can be performed using comparison and data transfer instructions.

The Bubble Sort method is commonly used for sorting in assembly language. In this method, adjacent elements are compared and swapped if they are not in the required order. This process is repeated multiple times until the entire list is sorted.

Working principle:

1. Load the count of numbers.
2. Compare two adjacent numbers using CMP instruction.
3. For ascending order:
    - If first number is greater than second, swap them.
4. For descending order:
    - If first number is smaller than second, swap them.
5. Repeat the process for all elements using loops until sorting is complete.

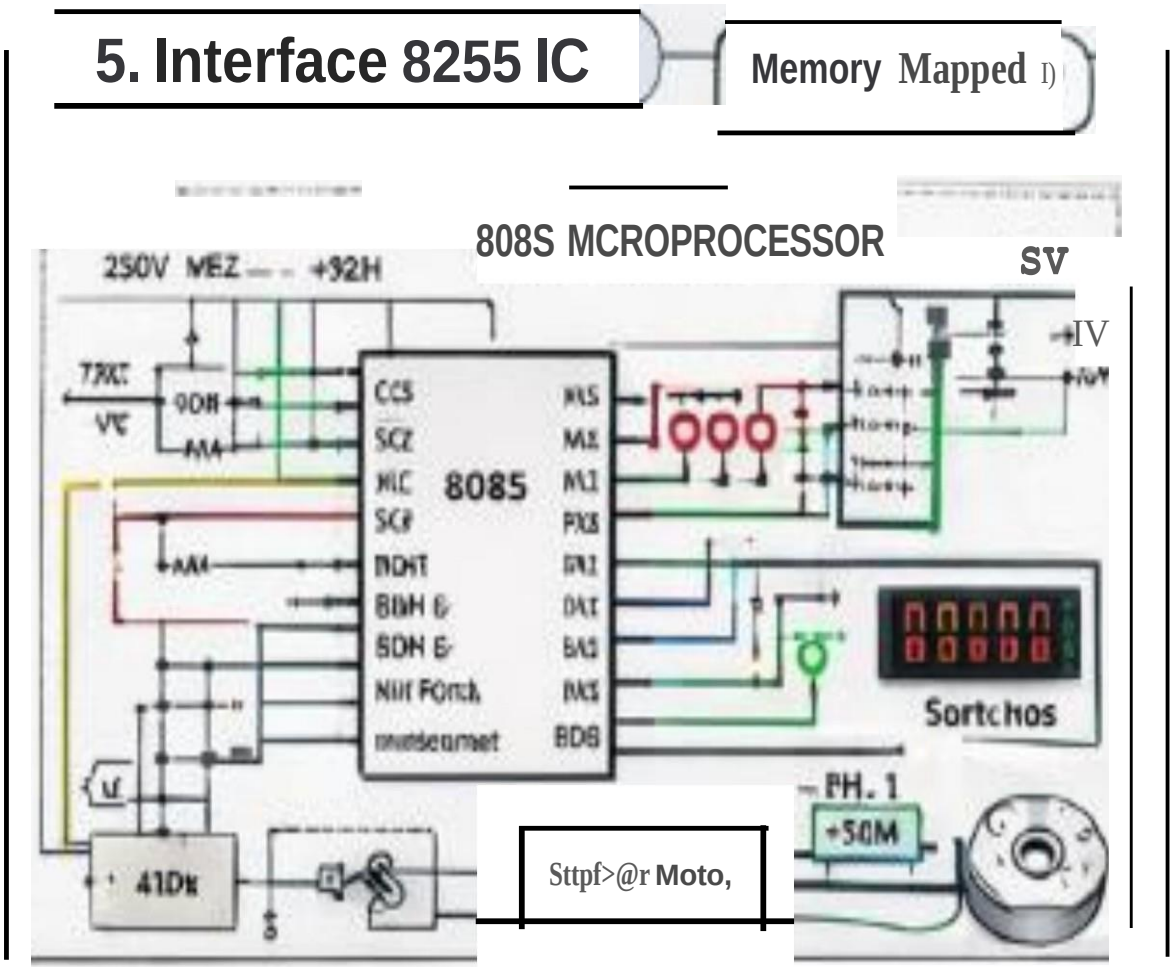
Important instructions used:

- CMP: Compare two numbers
- JC: Jump if carry (used in ascending logic)
- JNC: Jump if no carry (used in descending logic)
- MOY: Transfer data

- TNX/DCX: Increment/Decrement register pair

Practical Set-up / Circuit Diagram

The experiment is performed using an 8085 microprocessor trainer kit or simulator. The count and data values are stored in memory locations. The program is entered into memory using keypad or software. After execution, the sorted data can be observed in the same memory locations using the monitor program.



Apparatus Required

8085 Microprocessor Trainer Kit/ Simulator, Power Supply, Connecting Wires, Instruction Manual

Precautions

1. Ensure correct power supply before starting the experiment.
2. Enter the program carefully without errors.
3. Verify memory addresses before execution.
4. Reset the kit before loading a new program.
5. Avoid loose connections in hardware setup.
6. Check the data properly after execution.

Procedure

1. Switch ON the 8085 trainer kit.
2. Enter the assembly language program into memory.
3. Store the count of numbers at memory location (e.g., 2000H).
4. Store the numbers starting from memory location (e.g., 2001H onwards).
5. Initialize loop counters for outer and inner loops.
6. Compare adjacent numbers using CMP instruction.
7. Swap the numbers if they are not in required order.
8. Repeat the inner loop for all elements.
9. Repeat the outer loop until sorting is completed.
10. Execute the program.
11. Observe the sorted data in memory locations.

Observations and Calculations

Observation Table:

Memory Location Data (Hex) Description

12000H	ILIICount of numbers
12001H	ILIINumber1
12002H	ILIINumber2
12003H	ILIINumber3
I··.	ICIIRemaining numbers
After execution	Ascending/Descending data

Calculation:

Ascending Order - Arrange numbers from smallest to largest

Descending Order - Arrange numbers from largest to smallest

Result

The assembly language program was successfully executed to arrange the given numbers in ascending and descending order. The sorted values were verified in the memory locations.

VIVA BOOK

1. What is sorting in microprocessor?
2. What is Bubble Sort method?
3. What is the role of CMP instruction in sorting?
4. What is the difference between ascending and descending order?
5. Why are loops required in sorting programs?

EXPERIMENT N0:6

Aim:-

Aim

To interface 8255 Programmable Peripheral Interface with 8085 Microprocessor and perform input/output operations.

Theory

The 8255 Programmable Peripheral Interface (PPI) is used to connect input/output devices with a microprocessor. It provides three 8-bit ports: Port A, Port B and Port C, which can be programmed as input or output.

The 8255 works in different modes:

- Mode 0: Simple input/output
- Mode 1: Strobed input/output
- Mode 2: Bidirectional data transfer

In this experiment, Mode O is generally used. A control word is sent to the control register to configure ports as input or output.

Basic working principle:

1. The 8085 sends a control word to 8255 to configure ports.
2. Data is transferred through ports depending on configuration.
3. Input devices send data to microprocessor via 8255.
4. Output devices receive data from microprocessor via 8255.

Control Word Format (Mode 0):

- D7 = 1 - 1/0 mode
- D6-D5 - Port A mode
- D4 - Port A (input/output)

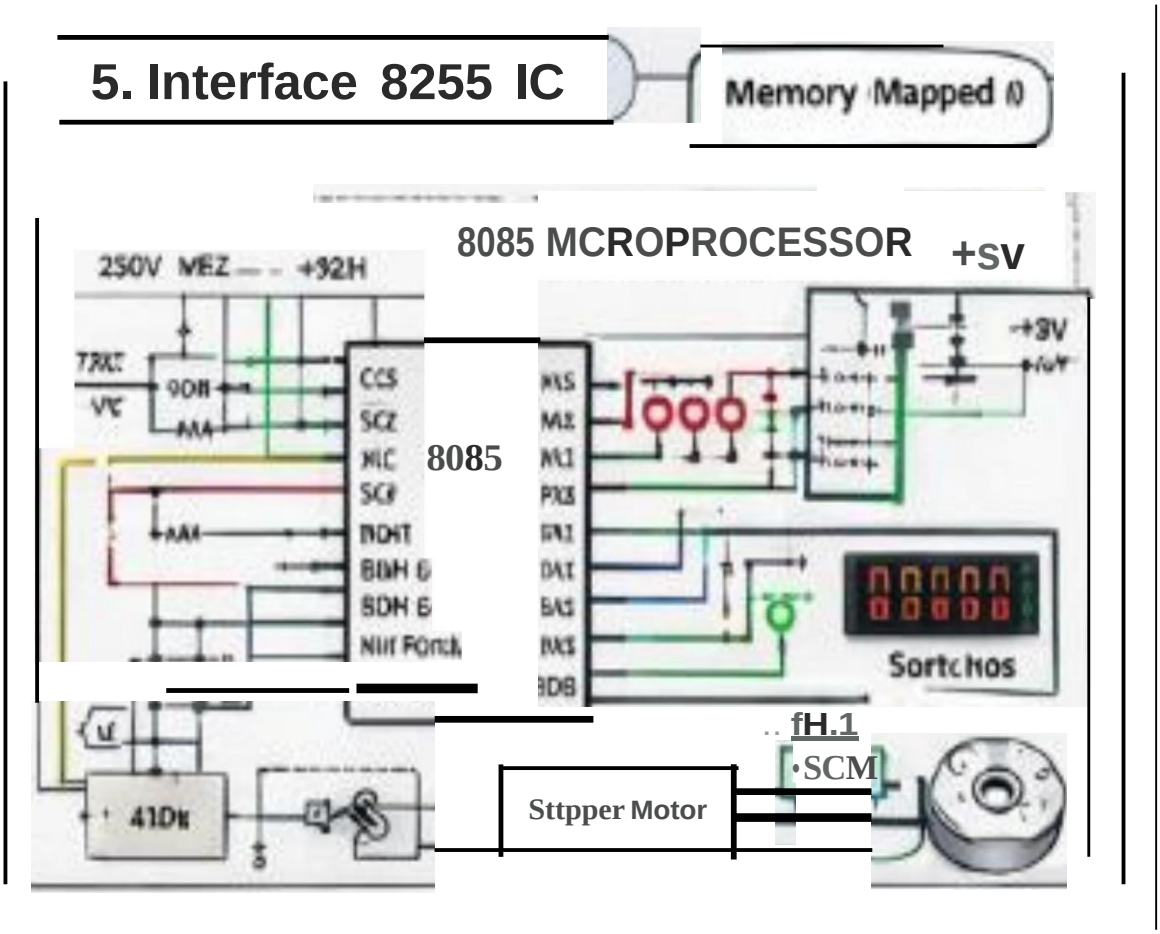
- D3 ---+ Port C upper
- D2 ---+ Port B mode
- DI ---+ Port B (input/output)
- DO---+ Port C lower

Example:

Control word = 80H ---+ All ports as output

Practical Set-up/ Circuit Diagram

The 8255 IC is interfaced with the 8085 microprocessor using address bus, data bus, and control signals. The ports of 8255 are connected to input/output devices such as switches and LEDs. The control word is loaded into the control register to configure the ports. When the program runs, data is transferred between devices and microprocessor through 8255.



Apparatus Required

8085 Microprocessor Trainer Kit, 8255 PPI IC, Power Supply, LEDs, Switches, Connecting Wires

Precautions

1. Ensure proper power supply to the circuit.
2. Check all connections before switching ON.
3. Enter correct control word for desired operation.
4. Avoid loose connections in wiring.
5. Reset the system before execution.
6. Handle ICs carefully to avoid damage.

Procedure

1. Connect the 8255 IC with the 8085 microprocessor kit.
2. Connect LEDs to output ports and switches to input ports.
3. Switch ON the power supply.
4. Write the assembly program to configure 8255 ports.
5. Load the control word into control register.
6. Write data to output port or read data from input port.
7. Execute the program.
8. Observe the output on LEDs or input from switches.
9. Verify correct operation of input/output transfer.

Observations and Calculations

Observation Table:

IPort	IIData (Hex)	IIOperation	IIResult
IPortA	IIXX	IIOu€ut	IILEDON/OFF
IPortB	IIXX	IIInput	IISwitch Status
IPortC	IIXX	I Input/Output	Control Signals
Control Reg	180H/XX	I Configuration	IMode Set

Calculation:

No mathematical calculation is required. Operation depends on control word configuration.

Result

The interfacing of 8255 with 8085 microprocessor was successfully performed. Input and output operations were carried out correctly using programmed control word.

VIVA BOOK

1. What is the function of 8255 PPI?
2. What are the different modes of 8255?
3. What is a control word in 8255?
4. What is the difference between input and output ports?
5. Why is interfacing required in microprocessor systems?

EXPERIMENT N0:7

Aim

To study, test and verify the features of 8051 Microcontroller trainer kit.

Theory

The 8051 microcontroller is an 8-bit microcontroller widely used in embedded systems. It consists of CPU, RAM, ROM, 1/0 ports, timers, serial communication, and interrupt system on a single chip.

An 8051 trainer kit is designed to learn and test the working of the microcontroller. It provides built-in hardware such as LEDs, switches, seven segment display, keypad, serial interface, and memory.

Important features of 8051:

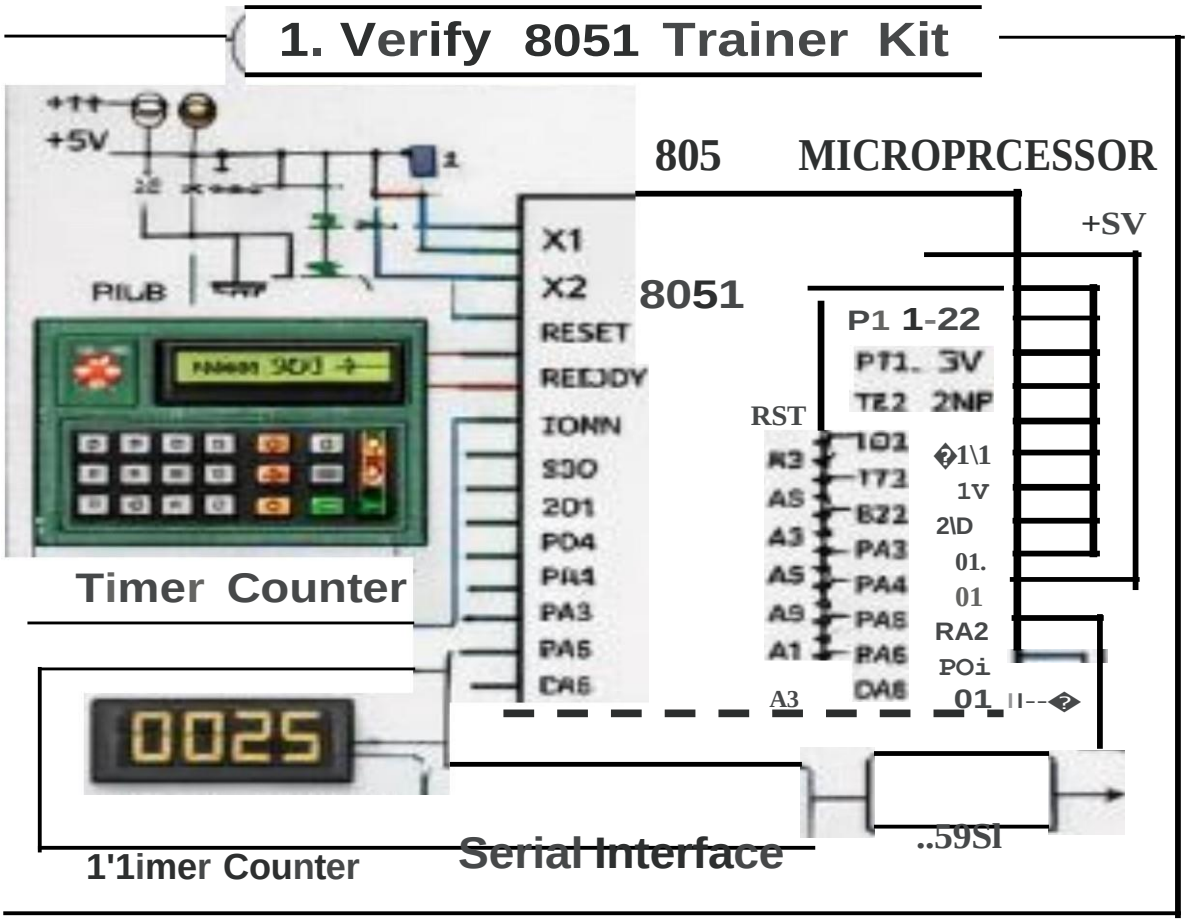
1. 8-bit CPU with Harvard architecture
2. 4 1/0 ports (Port 0, Port 1, Port 2, Port 3)
3. Two 16-bit timers/counters
4. Serial communication (UART)
5. Interrupt system
6. On-chip RAM and ROM

Working principle:

1. Program is written and stored in memory.
2. Microcontroller executes instructions sequentially.
3. Input is taken through ports or keypad.
4. Output is displayed using LEDs or displays.
5. Timers and serial ports perform special functions.

Practical Set-up / Circuit Diagram

The 8051 trainer kit consists of the microcontroller, power supply, crystal oscillator, reset circuit, input switches, LEDs, and display units. The kit is powered ON and programs are entered using keypad or computer interface. The execution of programs can be observed through output devices like LEDs and displays.



Apparatus Required

8051 Trainer Kit, Power Supply, Connecting Wires, Computer (optional)

Precautions

1. Ensure proper power supply before switching ON.

2. Do not touch components with wet hands.
3. Enter program carefully without errors.
4. Reset the kit before running a new program.
5. Avoid short circuits in connections.
6. Handle the trainer kit carefully.

Procedure

1. Switch ON the 8051 trainer kit.
2. Observe the display and ensure proper initialization.
3. Check all 1/0 ports by connecting LEDs and switches.
4. Write and load a simple program (e.g., LED blinking).
5. Execute the program and observe output on LEDs.
6. Test keypad by entering data.
7. Verify timer operation using delay program.
8. Test serial communication if available.
9. Check interrupt feature by triggering interrupt input.
10. Note down all observations.

Observations and Calculations

Observation Table:

!Feature Tested	IIInput Given	IIOutput Observed	IEJ
ILED Interface	I Program executed	ILEDs ON/OFF	IE::J
!Switch Input	IIPress switch	IIData received	IE::J

!Feature Tested	lllnput Given	IIOutput Observed	IEJ
ITimer	IIDelay program	I Time delay observed	IWorkingl
!Keypad	IIKey pressed	IIData displayed	IIWorkingl
Serial Communication	Data transmitted	IData received	IIWorkingl

Calculation:

No mathematical calculation is required. Verification is based on observation of outputs.

Result

The features of the 8051 trainer kit were successfully tested and verified. All components such as I/O ports, timers, keypad, and display were found to be working properly.

VIVA BOOK

1. Explain 8051 microcontroller?
2. Explain the main features of 8051?
3. Explain the function ofl/O ports in 8051?
4. What is the difference between microprocessor and microcontroller?
5. Explain the role of timer in 8051?

EXPERIMENT N0:8

Aim:-

To write and execute an assembly language program for 8051 Microcontroller to add two 8-bit numbers and store the result along with carry.

Theory

The 8051 microcontroller is an 8-bit controller that performs arithmetic operations using its accumulator (A register). For addition of two 8-bit numbers, the ADD instruction is used.

When two 8-bit numbers are added, the result may exceed 8 bits. In such a case, the extra bit is stored in the Carry flag (CY).

Working principle:

1. Load the first number into the accumulator.
2. Add the second number using ADD instruction.
3. Store the result in memory.
4. Check Carry flag:
    - If CY= 0 - No carry (result within 8 bits)
    - If CY= 1 - Carry generated (result exceeds 8 bits)
5. Store carry separately in another memory location.

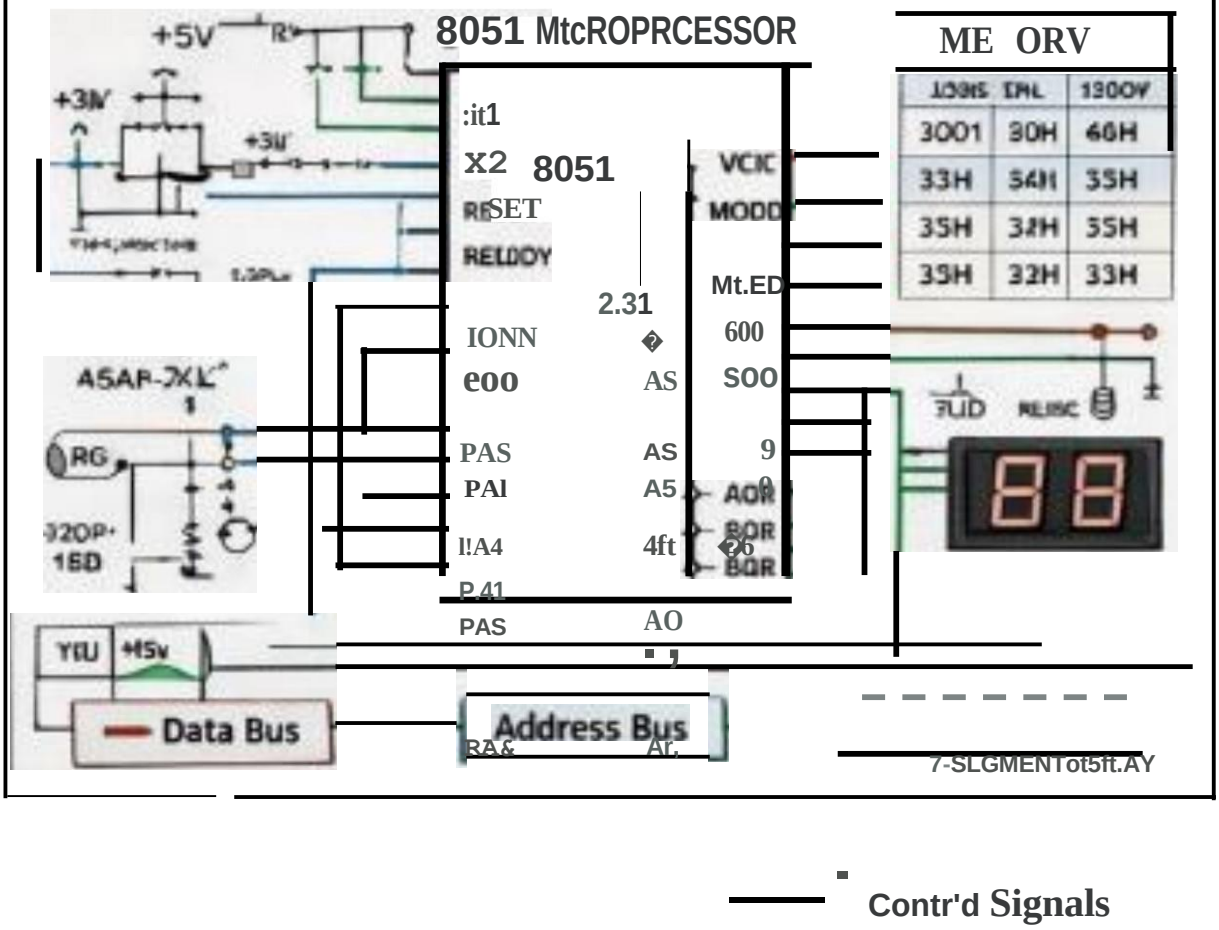
Important instructions:

- MOY: Data transfer
- ADD: Add register or memory to accumulator
- JNC: Jump if no carry
- JC: Jump if carry

Practical Set-up / Circuit Diagram

The experiment is performed using an 8051 trainer kit. The input data is stored in internal RAM locations. The program is written using assembly mnemonics and executed through the trainer kit or simulator. The result and carry are stored in specified memory locations and verified using memory display.

2. ALPs to Add/Subtract Two 8-bit Nos.



Apparatus Required

8051 Trainer Kit, Power Supply, Connecting Wires, Computer (optional)

Precautions

1. Ensure proper power supply before starting.
2. Enter program correctly without syntax errors.

3. Check memory addresses before execution.

4. Reset the kit before running program.

5. Avoid loose connections.
6. Verify carry flag carefully.

Procedure

1. Switch ON the 8051 trainer kit.
2. Enter the assembly language program into memory.
3. Store first 8-bit number in memory location (e.g., 30H).
4. Store second 8-bit number in memory location (e.g., 31H).
5. Load first number into accumulator.
6. Add second number using ADD instruction.
7. Store the result in memory location (e.g., 32H).
8. Check Carry flag using conditional jump.
9. Store carry (OOH or 0lH) in another location (e.g., 33H).
10. Execute the program.
11. Observe the result and carry in memory.

Observations and Calculations

Observation Table:

!Memory Location	IIData (Hex)	IIDescription
130H	IIXX	IIFirst number
131H	IIXX	IISecond number
132H	IIXX	1result (Sum)
133H	1100/ 01	IICarry Flag

Calculation:

Result= First Number+ Second Number

If Result<FFH - No Carry (CY= 0)

If Result> FFH- Carry generated (CY= 1)

Result

The addition of two 8-bit numbers using 8051 microcontroller was successfully performed. The result and carry were correctly stored in the specified memory locations.

VIVA BOOK

1. What is the function of ADD instruction in 8051?
2. What is Carry flag in 8051?
3. What happens when addition exceeds 8 bits?
4. What is the role of accumulator in 8051?
5. What is the difference between ADD and ADDC instruction?

EXPERIMENT N0:9

Aim

To write and execute an assembly language program for 8051 Microcontroller to subtract two 8-bit numbers and store the result along with borrow (carry).

Theory

The 8051 microcontroller performs subtraction using the SUBB (Subtract with Borrow) instruction. The subtraction operation is carried out using the accumulator (A register).

Before performing subtraction, the Carry flag must be cleared (CLR C), because SUBB subtracts both the operand and the borrow (CY).

Working principle:

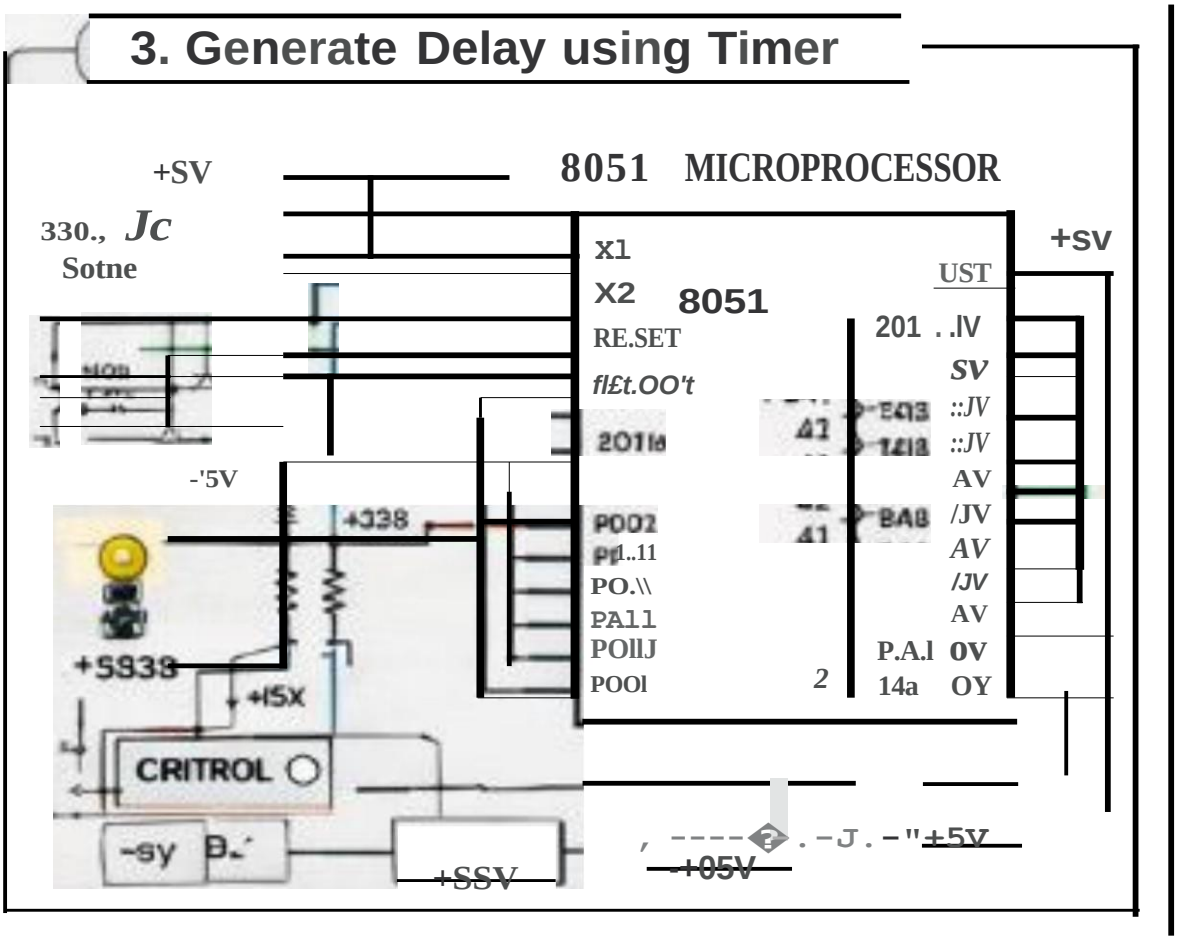
1. Load the first number (minuend) into accumulator.
2. Clear the Carry flag.
3. Subtract the second number (subtrahend) using SUBB instruction.
4. Store the result in memory.
5. Check Carry flag:
    - If CY= 0 - No borrow (without carry case)
    - If CY = 1 - Borrow occurred (with carry case)
6. Store carry/borrow in another memory location.

Important instructions:

- MOY: Data transfer
- CLR C: Clear carry flag
- SUBB: Subtract with borrow
- JNC: Jump if no carry
- JC: Jump if carry

Practical Set-up / Circuit Diagram

The experiment is performed using an 8051 trainer kit. The input data is stored in internal RAM locations. The program is written using assembly mnemonics and executed through the trainer kit or simulator. After execution, the result and borrow are stored in specified memory locations and verified using memory display.



Apparatus Required

8051 Trainer Kit, Power Supply, Connecting Wires, Computer (optional)

Precautions

1. Ensure proper power supply before starting.

2. Clear carry flag before subtraction.
3. Enter program carefully without errors.
4. Verify memory addresses before execution.
5. Reset the kit before running program.
6. Avoid loose connections.

Procedure

1. Switch ON the 8051 trainer kit.
2. Enter the assembly language program into memory.
3. Store first 8-bit number at memory location (e.g., 30H).
4. Store second 8-bit number at memory location (e.g., 31H).
5. Load first number into accumulator.
6. Clear the Carry flag using CLR C.
7. Subtract second number using SUBB instruction.
8. Store the result in memory location (e.g., 32H).
9. Check Carry flag using conditional jump.
10. Store carry/borrow (OOH or 0lH) in another location (e.g., 33H).
11. Execute the program.
12. Observe the result and carry in memory.

Observations and Calculations

Observation Table:

!Memory Location	11»ata (Hex)	11»escription
130H	‖XX	IIFirst number (Minuend)
131H	‖XX	IISecond number (Subtrahend)
132H	‖XX	IIResult
133H	1100/ 01	IICarry (Borrow)

Calculation:

Result= First Number - Second Number

If First Number≥Second Number - No Borrow (CY= 0)

If First Number< Second Number - Borrow (CY= **1)**

Result

The subtraction of two 8-bit numbers usmg 8051 microcontroller was successfully performed. The result and borrow were correctly stored in the specified memory locations.

VIVA BOOK

1. What is the function of SUBB instruction in 8051?
2. Why is CLR C used before subtraction?
3. What is the meaning of borrow in subtraction?
4. What is the difference between ADD and SUBB instruction?
5. What happens when the result of subtraction is negative in 8051?

EXPERIMENT NO:10

Aim

To write and execute an assembly language program using 8051 Microcontroller timer to generate a time delay.

Theory

The 8051 microcontroller has two 16-bit timers: Timer 0 and Timer 1. These timers can be used to generate accurate time delays.

Each timer consists of two registers: TH (Timer High) and TL (Timer Low). The timer counts machine cycles. When the timer overflows (from FFFFH to 0000H), the overflow flag (TF) is set.

Working principle:

1. Select timer mode using TMOD register.
2. Load initial count value into TH and TL registers.
3. Start the timer using TR (Timer Run) bit.
4. Wait until overflow flag TF becomes 1.
5. Stop the timer and clear TF flag.
6. Repeat if required.

Formula for delay:

Delay= (65536 - Initial Value) x Machine Cycle Time

Machine Cycle Time= 12 / Crystal Frequency

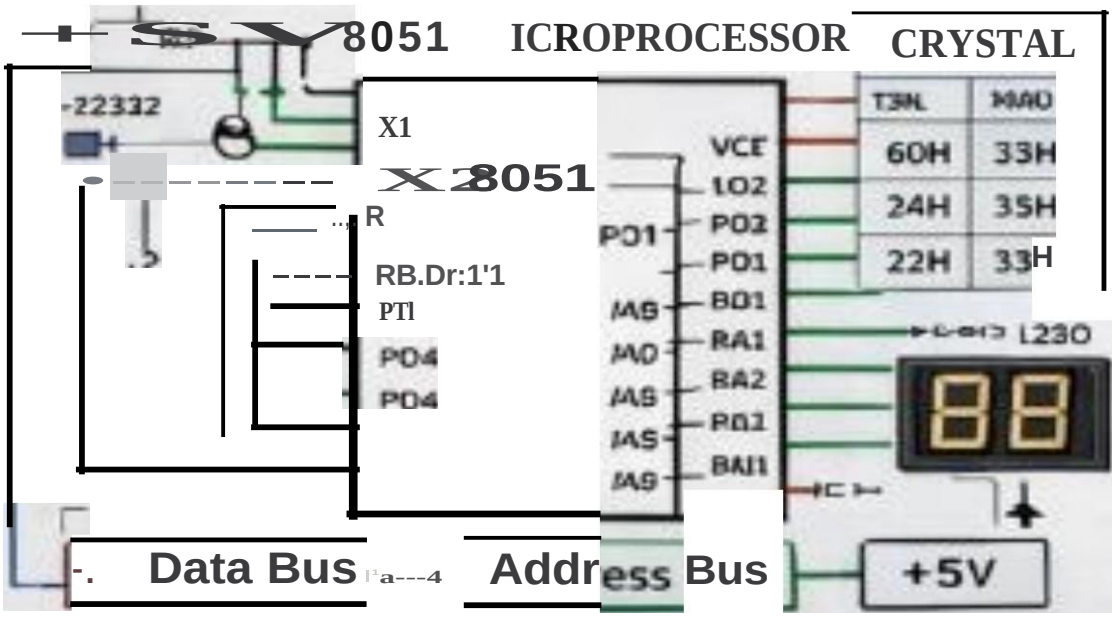
Example:

For 12 MHz crystal - Machine cycle= 1 μs

Practical Set-up / Circuit Diagram

The experiment is performed on an 8051 trainer kit. The timer registers are programmed using assembly instructions. The delay generated can be observed by connecting an LED and making it blink with the delay.

4. Interface 7 Segment Display



Apparatus Required

8051 Trainer Kit, Power Supply, LED, Connecting Wires

Precautions

1. Ensure correct power supply.
2. Load correct initial values in timer registers.
3. Clear overflow flag before reuse.
4. Stop timer after overflow.
5. Enter program correctly without errors.

6. Avoid loose connections.

Procedure

1. Switch ON the 8051 trainer kit.
2. Write and enter the ALP into memory.
3. Load TMOD register to select timer mode (e.g., Mode 1).
4. Load TH and TL registers with initial value.
5. Start timer by setting TR bit.
6. Wait until TF flag becomes 1.
7. Stop timer and clear TF flag.
8. Repeat steps to generate continuous delay.
9. Execute the program.
10. Observe delay (e.g., LED blinking).

Observations and Calculations

Observation Table:

1TH Valuell	TL Valuel	Delay Observed !	Result
DDITimeDelay			

Calculation:

Delay= (65536 - Initial Value) x Machine Cycle Time

Example:

IfTH = FC, TL= 66---+Initial value= FC66H

Result

The delay was successfully generated using the timer of 8051 microcontroller. The timer operation and delay calculation were verified.

VIVA BOOK

1. What is a timer in 8051?
2. What are different modes of timer in 8051?
3. What is TMOD register?
4. What is overflow flag (TF)?
5. What is the difference between timer and counter?

EXPERIMENT N0:11

Aim

To write and execute an assembly language program to interface a 7 segment display with 8051 Microcontroller and display numbers.

Theory

A 7 segment display is an electronic display device used to display digits from 0 to 9. It consists of seven LEDs (segments) labeled a to g. By turning ON or OFF these segments, different numbers can be displayed.

There are two types of 7 segment displays:

- Common Anode
- Common Cathode

In common cathode, all cathodes are connected together and segments glow when logic HIGH is applied. In common anode, all anodes are common and segments glow when logic LOW is applied.

Working principle:

1. Each segment is connected to an output port of 8051.
2. Binary data corresponding to digits is sent to the port.
3. The segments glow according to the data pattern.

Example codes (Common Cathode):

0 - 3FH

1-06H

2-SBH

3-4FH

4-66H

5---+ 6DH

6- --+7DH

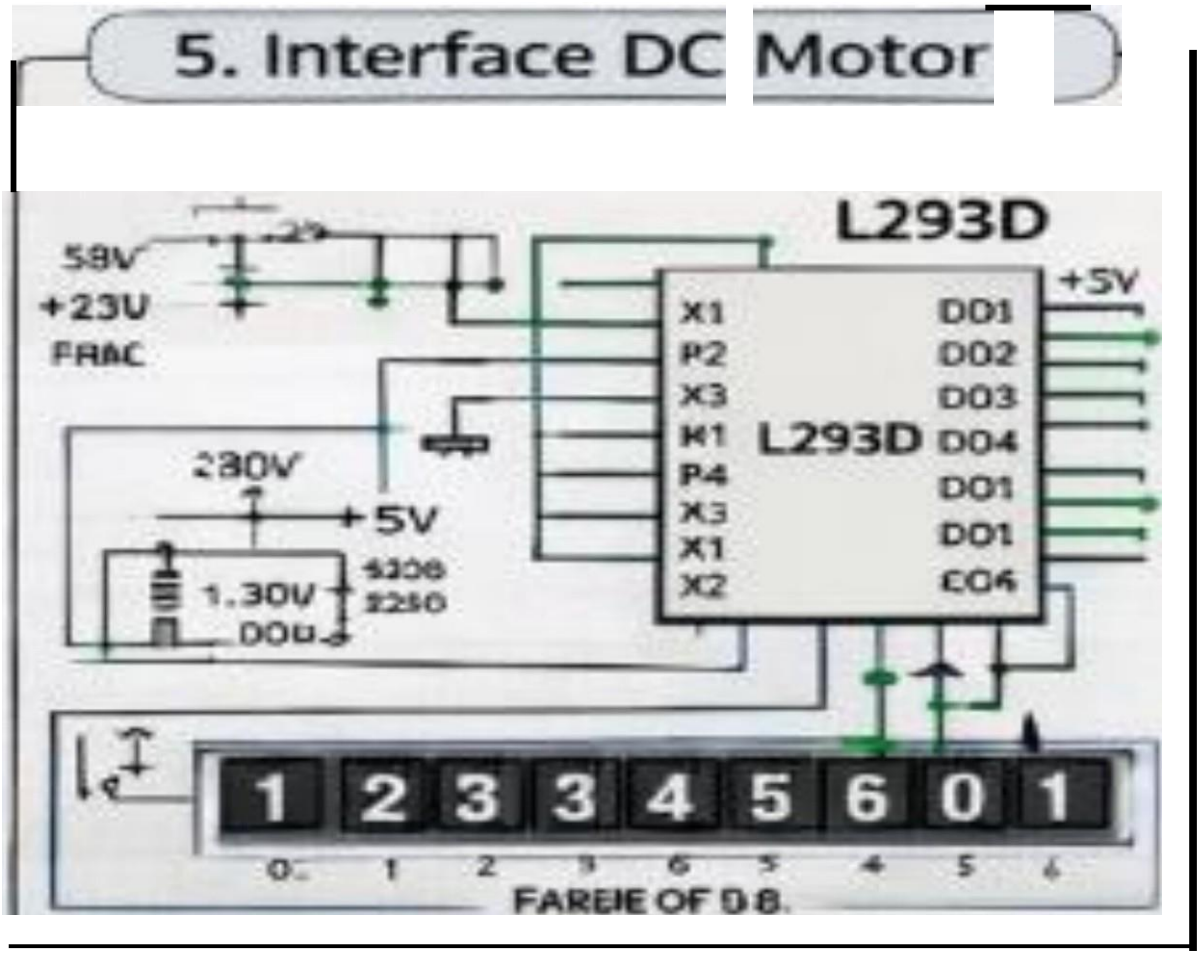
7- --+07H

8- --+7FH

9- --+6FH

Practical Set-up / Circuit Diagram

The 7 segment display is connected to one port of the 8051 (e.g., Port 1). Each segment is connected through current limiting resistors. The common terminal is connected to ground (for common cathode) or Vee (for common anode). The microcontroller sends data to the port to display digits.



Apparatus Required

8051 Trainer Kit, 7 Segment Display, Resistors, Connecting Wires, Power Supply

Precautions

1. Use proper current limiting resistors.
2. Check whether display is common anode or cathode.
3. Ensure correct port connections.
4. Avoid short circuits.
5. Enter program carefully.
6. Provide proper power supply.

Procedure

1. Connect the 7 segment display to Port 1 of 8051.
2. Connect resistors in series with each segment.
3. Identify the type of display (common anode/cathode).
4. Write the assembly program for displaying digits.
5. Load the program into the trainer kit.
6. Execute the program.
7. Observe the displayed digits.
8. Verify correct output sequence.

Observations and Calculations

Observation Table:

Input Data (Hex)	Digit Displayed	Segments ON
13FH	0	a, b, c, d, e, f

106H 11bc,

15BH 112 Ila,b, d,eg,

14FH 113 Ila,b, c,dg,

166H-114 11b, c

Calculation:

No mathematical calculation is required. Output depends on segment code.

Result

The 7 segment display was successfully interfaced with the 8051 microcontroller. The desired digits were displayed correctly according to the program.

VIVA BOOK

1. What is a 7 segment display?
2. What is the difference between common anode and common cathode display?
3. Why are resistors used with 7 segment display?
4. What is the role of microcontroller in display interfacing?
5. How are digits displayed using binary codes?

EXPERIMENT N0:12

Aim

To write and execute an assembly language program to interface a DC motor with 8051 Microcontroller and control its operation.

Theory

A DC motor converts electrical energy into mechanical energy. Since the 8051 microcontroller cannot supply sufficient current to drive a motor directly, a driver circuit such as a transistor or motor driver IC (like L293D) is used.

The microcontroller provides control signals to the driver circuit, which in turn drives the motor. By changing the logic levels at the output pins, the motor can be turned ON/OFF and its direction can also be controlled.

Working principle:

1. Output pin of 8051 is connected to motor driver.
2. Logic HIGH/LOW signals control motor operation.
3. Driver circuit amplifies current to run the motor.

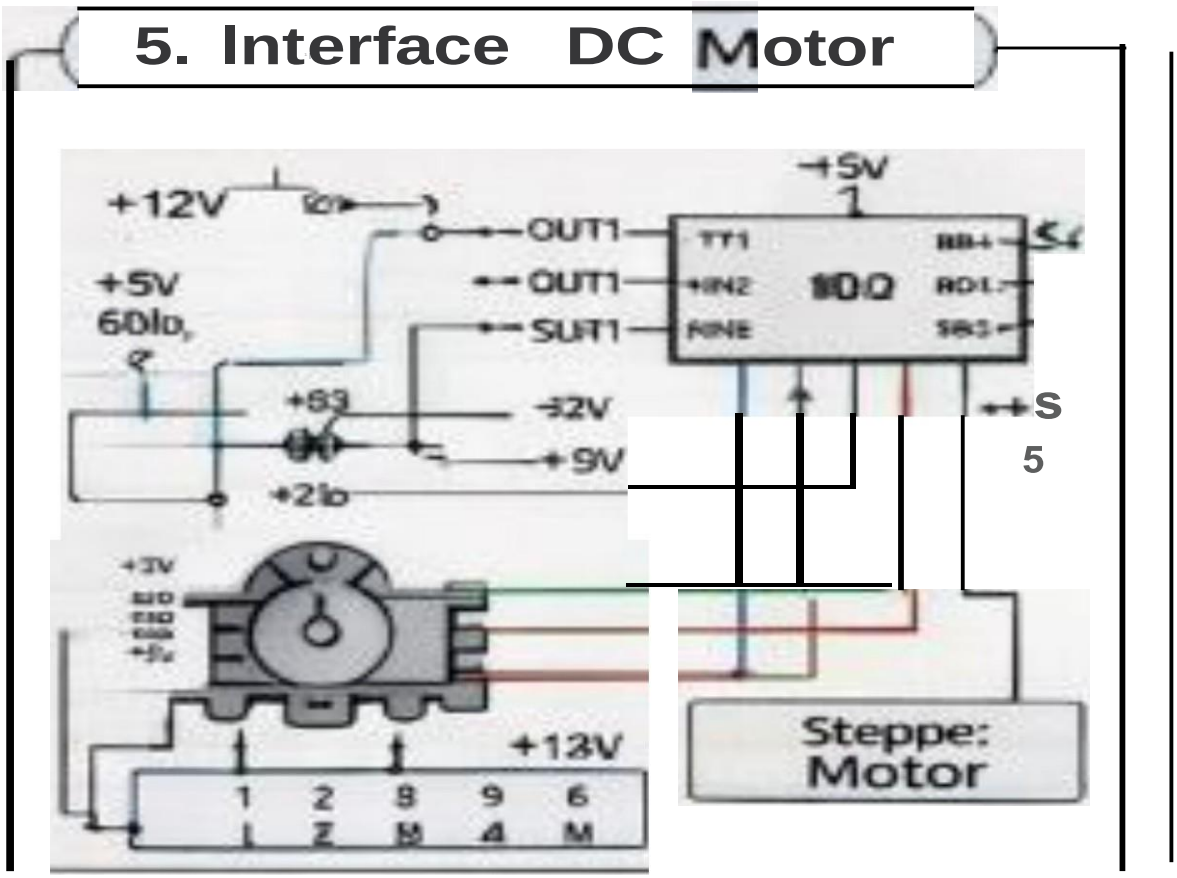
Control logic (using two pins):

- 1 0 - Clockwise rotation
- 0 1 - Anticlockwise rotation
- 1 1 or 0 0 - Motor stop

Practical Set-up/ Circuit Diagram

The DC motor is connected to the 8051 through a motor driver IC (L293D) or a transistor circuit. The input pins of the driver are connected to the output port (e.g., Port 1) of the microcontroller. The motor is powered using an external supply.

5. Interface DC Motor



Apparatus Required

8051 Trainer Kit, DC Motor, Motor Driver IC (L293D) / Transistor, Power Supply, Connecting Wires

Precautions

1. Do not connect motor directly to microcontroller.
2. Use proper driver circuit to avoid damage.
3. Ensure correct power supply for motor and IC.
4. Check all connections before switching ON.
5. Avoid short circuits.
6. Handle components carefully.

Procedure

1. Connect the motor to driver IC (L293D).
2. Connect driver inputs to 8051 output port pins.
3. Provide separate power supply to motor.
4. Switch ON the trainer kit.
5. Write the assembly program for motor control.
6. Load the program into memory.
7. Execute the program.
8. Observe motor rotation (ON/OFF or direction).
9. Verify correct working of motor control.

Observations and Calculations

Observation Table:

Port Output	Motor Action	Result
0lH	Clockwise Rotation	Working
02H	Anticlockwise	Working
OOH	Stop	Working
03H	Stop	Working

Calculation:

No mathematical calculation is required. Operation depends on control signals.

Result

The DC motor was successfully interfaced with the 8051 microcontroller. The motor operation and direction were controlled correctly using the program.

VIVA BOOK

1. What is a DC motor?
2. Why is a driver circuit required for motor interfacing?
3. What is the function of L293D IC?
4. How is motor direction controlled?
5. What is the difference between microcontroller output and motor requirement?

EXPERIMENT N0:13

Aim

To write and execute an assembly language program to interface a stepper motor with 8051 Microcontroller and control its rotation.

Theory

A stepper motor is an electromechanical device that converts electrical pulses into discrete mechanical steps. Each pulse causes the motor shaft to move by a fixed angle called step angle.

The stepper motor requires a sequence of pulses to rotate. Since the 8051 cannot drive the motor directly, a driver circuit like ULN2003 or L293D is used to provide sufficient current.

Working principle:

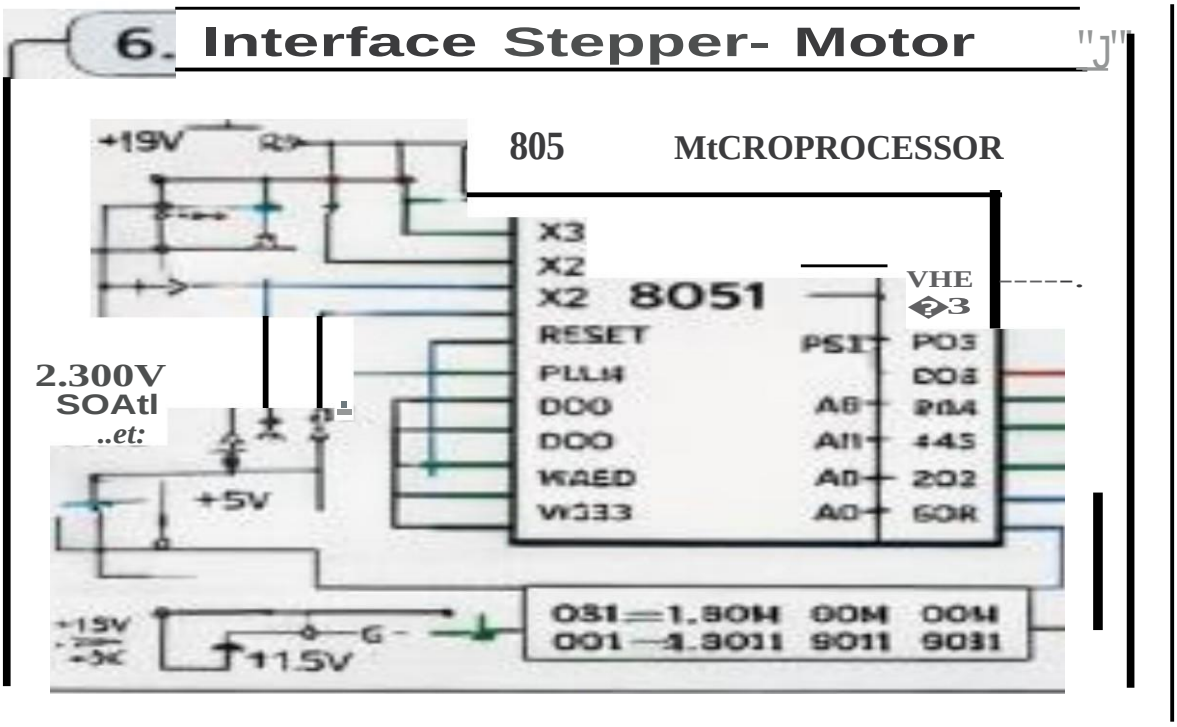
1. The microcontroller sends pulse sequences to motor coils.
2. Each pulse energizes a specific coil.
3. The sequence of energizing coils causes rotation.
4. Changing sequence changes direction of rotation.

Example sequence (4-step sequence):

- 09H - 0A H - 06H - 05H (Clockwise)
- Reverse sequence for anticlockwise

Practical Set-up / Circuit Diagram

The stepper motor is connected to the 8051 through a driver IC (ULN2003 or L293D). The output port pins (e.g., Port 1) are connected to the driver inputs. The driver outputs are connected to the motor windings. Aseparate power supply is used for the motor. The program sends step sequences to rotate the motor.



Apparatus Required

8051 Trainer Kit, Stepper Motor, ULN2003 / L293D Driver IC, Power Supply, Connecting Wires

Precautions

1. Do not connect motor directly to microcontroller.
2. Use proper driver IC for safe operation.
3. Ensure correct sequence of pulses.
4. Provide proper power supply.
5. Avoid loose connections.
6. Handle components carefully.

Procedure

1. Connect stepper motor to driver IC.
2. Connect driver inputs to 8051 port pins.

3. Provide power supply to motor and kit.
4. Write the assembly program for step sequence.
5. Load the program into memory.
6. Execute the program.
7. Observe motor rotation.
8. Change sequence to reverse direction if needed.
9. Verify correct stepping operation.

Observations and Calculations

Observation Table:

Step Sequence	Direction	Result
09H	Clockwise	Working
0AH	Clockwise	Working
06H	Clockwise	Working
05H	Clockwise	Working
Reverse Seq	Anticlockwise	Working

Calculation:

No mathematical calculation is required. Rotation depends on pulse sequence.

Result

The stepper motor was successfully interfaced with the 8051 microcontroller. The motor rotation and direction were controlled correctly using the program.

VIVA BOOK

1. What is a stepper motor?
2. Why is a driver IC required for stepper motor?
3. What is step angle?
4. How is direction controlled **in** stepper motor?
5. What is the difference between DC motor and stepper motor?