# Laboratory manual

# on

#  microprocessor 8085



**DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING**

**SANT LONGOWAL INSTITUTE OF ENGINEERING AND TECHNOLOGY**,

**LONGOWAL- 148106, PUNJAB**

# preface

This 8085 microprocessor lab manual is designed to provide a basic knowledge about 8085 microprocessor to the users and the students. Before reading this manual, the students must be familiar with the basics of 8085 and should be capable of performing basic logical programming on paper. The programs in the manual are in assembly language which in itself requires rigorous mental exercise. To perform to the fullest, the students must attend this lab course in parallel to the theory course.

The authors of this manual went through plenty of books and sources to gather a list of programs and experiments that test the students programming capabilities while preparing them to operate the 8085 kit without any hassle. The programs are preceded by some theory that captures the gist of the topic. For best results, the authors recommend using \_\_\_\_\_\_ along with the 8085 kit provided in the lab.

It is of utmost importance that the contents of this manual are error free and the authors have tried their best to make it so. If you come across any misprint or misinformation, kindly inform your lab in-charge.

Contents

[EXPERIMENT 1 2](#_Toc20074193)

[AIM 2](#_Toc20074194)

[ARCHITECTURE OF 8085 MICROPROCESSOR 2](#_Toc20074195)

[PIN DESCRIPTION 4](#_Toc20074196)

[SPECIFICATIONS 6](#_Toc20074197)

[EXPERIMENT 02 7](#_Toc20074198)

[AIM 7](#_Toc20074199)

[PREREQUISITES 7](#_Toc20074200)

[APPARATUS REQUIRED 7](#_Toc20074201)

[ALGORITHM 7](#_Toc20074202)

[FLOWCHART 8](#_Toc20074203)

[SOURCE CODE 9](#_Toc20074204)

[SAMPLE INPUT-OUTPUT 10](#_Toc20074205)

[QUESTIONS: 10](#_Toc20074206)

[EXPERIMENT 03 11](#_Toc20074207)

[AIM 11](#_Toc20074208)

[PREREQUISITES 11](#_Toc20074209)

[APPARATUS REQUIRED 11](#_Toc20074210)

[ALGORITHM 11](#_Toc20074211)

[FLOWCHART 12](#_Toc20074212)

[SOURCE CODE 13](#_Toc20074213)

[SAMPLE INPUT OUTPUT 14](#_Toc20074214)

[QUESTIONS: 14](#_Toc20074215)

[EXPERIMENT 04(a) 15](#_Toc20074216)

[AIM 15](#_Toc20074217)

[PREREQUISITES 15](#_Toc20074218)

[APPARATUS REQUIRED 15](#_Toc20074219)

[ALGORITHM 15](#_Toc20074220)

[FLOWCHART 16](#_Toc20074221)

[SOURCE CODE 17](#_Toc20074222)

[SAMPLE INPUT OUTPUT 18](#_Toc20074223)

[QUESTIONS 18](#_Toc20074224)

[EXPERIMENT 04(b) 19](#_Toc20074225)

[AIM: 19](#_Toc20074226)

[PREREQUISITES: 19](#_Toc20074227)

[APPARATUS REQUIRED: 19](#_Toc20074228)

[ALGORITHM: 19](#_Toc20074229)

[FLOWCHART: 20](#_Toc20074230)

[SOURCE CODE 20](#_Toc20074231)

[SAMPLE INPUT OUTPUT 21](#_Toc20074232)

[QUESTIONS 21](#_Toc20074233)

[EXPERIMENT 05(a) 22](#_Toc20074234)

[AIM 22](#_Toc20074235)

[PREREQUISITES 22](#_Toc20074236)

[APPARATUS REQUIRED 22](#_Toc20074237)

[ALGORITHM 22](#_Toc20074238)

[FLOWCHART 23](#_Toc20074239)

[SOURCE CODE 24](#_Toc20074240)

[SAMPLE INPUT AND OUTPUT 25](#_Toc20074241)

[QUESTIONS 25](#_Toc20074242)

[EXPERIMENT 05(b) 26](#_Toc20074243)

[AIM 26](#_Toc20074244)

[PREREQUISITES 26](#_Toc20074245)

[APPARATUS REQUIRED 26](#_Toc20074246)

[ALGORITHM 26](#_Toc20074247)

[FLOWCHART 27](#_Toc20074248)

[SOURCE CODE 28](#_Toc20074249)

[SAMPLE INPUT OUTPUT 28](#_Toc20074250)

[QUESTIONS 28](#_Toc20074251)

[EXPERIMENT 06 29](#_Toc20074252)

[AIM 29](#_Toc20074253)

[PREREQUISITES 29](#_Toc20074254)

[APPARATUS REQUIRED 29](#_Toc20074255)

[ALGORITHM 29](#_Toc20074256)

[FLOWCHART 30](#_Toc20074257)

[SOURCE CODE 31](#_Toc20074258)

[SAMPLE INPUT OUTPUT 32](#_Toc20074259)

[QUESTIONS 32](#_Toc20074260)

[EXPERIMENT 07 33](#_Toc20074261)

[AIM 33](#_Toc20074262)

[PREREQUISITES 33](#_Toc20074263)

[APPARATUS REQUIRED 33](#_Toc20074264)

[ALGORITHM 33](#_Toc20074265)

[FLOWCHART 33](#_Toc20074266)

[SOURCE CODE 34](#_Toc20074267)

[SAMPLE INPUT OUTPUT 34](#_Toc20074268)

[QUESTIONS 35](#_Toc20074269)

[EXPERIMENT 08 36](#_Toc20074270)

[AIM 36](#_Toc20074271)

[PREREQUISITES 36](#_Toc20074272)

[APPARATUS REQUIRED 36](#_Toc20074273)

[ALGORITHM 36](#_Toc20074274)

[FLOWCHART 37](#_Toc20074275)

[SOURCE CODE 38](#_Toc20074276)

[SAMPLE INPUT AND OUTPUT 39](#_Toc20074277)

[QUESTIONS 39](#_Toc20074278)

[EXPERIMENT 09 40](#_Toc20074279)

[AIM 40](#_Toc20074280)

[PREREQUISITES 40](#_Toc20074281)

[APPARATUS REQUIRED 40](#_Toc20074282)

[ALGORITHM 40](#_Toc20074283)

[FLOWCHART 40](#_Toc20074284)

[SOURCE CODE 41](#_Toc20074285)

[SAMPLE INPUT AND OUTPUT 41](#_Toc20074286)

[QUESTIONS 41](#_Toc20074287)

[EXPERIMENT 10 42](#_Toc20074288)

[AIM 42](#_Toc20074289)

[PREREQUISITES 42](#_Toc20074290)

[APPARATUS REQUIRED 42](#_Toc20074291)

[ALGORITHM 42](#_Toc20074292)

[FLOWCHART 43](#_Toc20074293)

[SOUCE CODE 44](#_Toc20074294)

[SAMPLE INPUT AND OUTPUT 45](#_Toc20074295)

[QUESTIONS 45](#_Toc20074296)

[EXPERIMENT 11 46](#_Toc20074297)

[AIM 46](#_Toc20074298)

[PREREQUISITES 46](#_Toc20074299)

[APPARATUS REQUIRED 46](#_Toc20074300)

[ALGORITHM 46](#_Toc20074301)

[FLOWCHART 47](#_Toc20074302)

[SOURCE CODE 48](#_Toc20074303)

[SAMPLE INPUT AND OUTPUT 49](#_Toc20074304)

[QUESTIONS 49](#_Toc20074305)

[EXPERIMENT 12 50](#_Toc20074306)

[AIM 50](#_Toc20074307)

[PREREQUISITES 50](#_Toc20074308)

[APPARATUS REQUIRED 50](#_Toc20074309)

[ALGORITHM 50](#_Toc20074310)

[FLOWCHART 51](#_Toc20074311)

[SOURCE CODE 52](#_Toc20074312)

[SAMPLE INPUT AND OUTPUT 52](#_Toc20074313)

[QUESTIONS 52](#_Toc20074314)

[APPENDIX I 53](#_Toc20074315)

# EXPERIMENT 1

AIMTo study the microprocessor 8085

## ARCHITECTURE OF 8085 MICROPROCESSOR

a) General purpose registers

They are six 8 bit register i.e. B, C, D, E, H, L. The combination of 8 bit register is known as register pair, which can hold 16 bit data. The HL pair is used to act as memory pointer is accessible to program.

b) Accumulator

It is an 8 bit register which hold one of the data to be processed by ALU and store`d the result of the operation.

c) Program Counter (PC)

It is a 16 bit pointer which holds the address of next instruction to be executed.

d) Stack pointer (Sp)

It is a 16 bit special purpose register which points to the top of stack.

e) Arithmetic and logical unit

It carries out arithmetic and logical operation by 8 bit address it uses the accumulator content as input the ALU result is stored back into accumulator.

f) Temporary register

It is an 8 bit register associated with ALU hold data, entering an operation, used by the microprocessor and not accessible to programs.

g) Flags

Flag register is a group of flip-flops which are set or reset after an operation according to the data conditions of the result in the accumulator. The five flags are

1. Carry flag (C)
2. Parity flag (P)
3. Zero flag (Z)
4. Auxiliary carry flag (AC)
5. Sign flag (S)

h) Timing and control unit

All microprocessor operation with the clock and generator and control signal from it necessary to maintain a synchronous communication between controller and peripheral.

i) Instruction register and decoder

Instruction is fetched from line memory and stored in line instruction register decoder the stored information.

j) Register Array

These are used to store 8 bit data during execution of some instruction.



Figure 1 Pin Diagram of 8085

## PIN DESCRIPTION

Address Bus

1. The pins Ao – A15 denote the address bus.
2. They are used for most significant bit.

Address / Data Bus

1. AD0 – AD7 constitutes the address / Data bus
2. These pins are used for least significant bit

ALE (Address Latch Enable)

1. The signal goes high during the first clock cycle and enables the lower order address bits.

IO / M

1. This distinguishes whether the address is for memory or input.
2. When this pin goes high, the address is for an I/O device.

S0 – S1

S0 and S1 are status signal which provides different status and functions.

RD

1. This is an active low signal
2. This signal is used to control READ operation of the microprocessor.

WR

1. WR is also an active low signal
2. Controls the write operation of the microprocessor.

HOLD

1. This indicates if any other device is requesting the use of address and data bus.

HLDA

1. HLDA is the acknowledgement signal for HOLD
2. It indicates whether the hold signal is received or not.

INTR

1. INTE is an interrupt request signal
2. IT can be enabled or disabled by using software

INTA

1. Whenever the microprocessor receives interrupt signal
2. It has to be acknowledged.

RST 5.5, 6.5, 7.5

1. These are nothing but the restart interrupts
2. They insert an internal restart junction automatically.

TRAP

1. Trap is the only non-maskable interrupt
2. It cannot be enabled (or) disabled using program.

RESET IN

1. This pin resets the program counter to 0 to 1 and results interrupt enable and HLDA flip flops.

X1, X2

These are the terminals which are connected to external oscillator to produce the necessary and suitable clock operation.

SID

This pin provides serial input data

SOD

This pin provides serial output data

VCC and VSS

1. VCC is +5V supply pin
2. VSS is ground pin

## SPECIFICATIONS

1. Processors

Intel 8085 at E144 MHz clock

|  |  |  |
| --- | --- | --- |
| 2. Memory |  |  |
| Monitor RAM: | 0000 | – IFFF |
| EPROM Expansion: | 2000 | – 3FFF’s |
|  | 0000 | – FFF |
| System RAM: | 4000 | – 5FFF |
| Monitor data area | 4100 | – 5FFF |
| RAM Expansion | 6000 | – BFFF |

3. Input / Output

Parallel: A8 TTL input timer with 2 number of 32-55 only input timer available in -85 EBI.

Serial: Only one number RS 232-C, Compatible, crucial interface using 8281A

4. Timer: 3 channel -16 bit programmable units, using 8253 channel ‘0’ used for no band late. Clock generator. Channel ‘1’ is used for single stopping used program.

5. Display: 6 digit – 7 segment LED display with filter 4 digit for adder display and 2 digit for data display.

6. Key board: 21 keys, soft keyboard including common keys and hexa decimal keys.

7. RES: Reset keys allow to terminate any present activity and retain to  - 85 its on initialize state.

8. INT: Maskable interrupt connect to CPU’s RST 7.5 interrupt

9. DEC: Decrement the adder by 1

10. EXEC: Execute line particular value after selecting address through go command.

11. NEXT: Increment the address by 1 and then display its content.

# EXPERIMENT 02

AIMAddition of two 8 bit numbers.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of binary number addition.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hexcodes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.

## APPARATUS REQUIRED

 8085 microprocessor kit

## ALGORITHM

|  |  |
| --- | --- |
| Step 1  | Start the microprocessor |
| Step 2 | Load 00 in register C  |
| Step 3 | Load the first 8 bit data into the accumulator |
| Step 4 | Copy the contents of accumulator into register B |
| Step 5 | Load the second 8 bit data into the accumulator. |
| Step 6 | Add the 2 - 8 bit data and check for carry. |
| Step 7 | Jump on if no carry |
| Step 8 | Increment C if there is carry |
| Step 9 | Store the added result in memory |
| Step 10 | Move the C value to accumulator |
| Step 11 | Store the accumulator value (Carry ) in memory  |
| Step 12 | Stop the program execution. |

## FLOWCHART

****

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4100 |  | MVI C,00H | 0E,00 | Clear the register C |
| 4102 |  | LDA 4300 | 3A,(00,43) | Load the first 8 bit data from the memory address 4300H |
| 4105 |  | MOV B,A | 47 | Copy the value of 8 bit data into register B from accumulator |
| 4106 |  | LDA 4301 | 3A (01,43) | Load the second 8 bit data from the memory address 4300H |
| 4109 |  | ADD B | 80 | Add contents of A and B |
| 410A |  | JNC LOOP | D2,0E,41 | Jump if no carry |
| 410D |  | INR C | 0C | If carry is there then increment C by 1 |
| 410E | LOOP | STA 4302 | 32 (02,43) | Store the added value at the address 4302H |
| 4111 |  | MOV A,C | 79 | Copy the value of carry to the accumulator from register C |
| 4112 |  | STA 4303 | 32 (03,43) | Store the value of carry at the address 4303 from accumulator |
| 4115 |  | HLT | 76 | Stop the execution of program |

## SAMPLE INPUT-OUTPUT

1. Without carry

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input Address |  | Value |
|  | 4300 |  |  08 |
|  | 4301 |  |  03 |
|  |  |  |  |  |
| Output Address |  | Value |
|  | 4302 |  |  0B |
|  | 4303 |  | 00 (carry) |

1. With carry

|  |  |  |
| --- | --- | --- |
| Input Address |  | Value |
| 4300 |  |  4F |
| 4301 |  | FF |
|  |  |
| Output Address |  | Value |
| 4302 |  |  4E |
| 4303 |  | 01 (carry) |

## QUESTIONS:

1. What is the function of LDA?
2. What is the function of STA?
3. What is the function of JNC?
4. Find the contents of Accumulator after the execution of the program:

|  |
| --- |
| MVI A, 45H |
| MVI B, 25H |
| ADD B |
| HLT |

# EXPERIMENT 03

AIM Subtraction of two 8 bit numbers.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of binary number subtraction.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about 1’s compliment and 2’s compliment

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

|  |  |
| --- | --- |
| Step 1 | Start the microprocessor |
| Step 2 | Clear register C |
| Step 3 | Load the first 8 bit data into the accumulator |
| Step 4 | Copy the contents of contents into the register ‘B’ |
| Step 5 | Load the second 8 bit data into the accumulator. |
| Step 6 | Subtract the 2 8 bit data and check for borrow. |
| Step 7 | Jump on if no borrow |
| Step 8 | Increment borrow if there is |
| Step 9 | 2’s compliment of accumulator is found out |
| Step 10 | Store the result in the memory |
| Step 11 | More the borrow value from ‘c’ to accumulator |
| Step 12 | Store the borrow value in the memory |
| Step 13 | Stop program execution |

## FLOWCHART

Clear the register C

Yes

No

Store the register C content (borrow) in memory

Store the result in the memory

2’s complement of accumulator value

C = C + 1

Check if borrow?

Subtract the values

Load the 2nd 8-bit data

Load the first 8-bit data to register B

Load the first 8-bit data

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hex codes** | **Comments** |
| 4100 |  | MVI C,00H | 0E,00 | Clear the register C |
| 4102 |  | LDA 4300 | 3A,(00,43) | Load the first 8 bit data from the memory address 4300H |
| 4105 |  | MOV B,A | 47 | Copy the value of 8 bit data into register B from accumulator |
| 4106 |  | LDA 4301 | 3A (01,43) | Load the second 8 bit data from the memory address 4300H |
| 4109 |  | SUB B | 90 | Subtract contents of B fromA |
| 410A |  | JNC LOOP | D2,0E,41 | Jump if no borrow |
| 410D |  | INR C | 0C | If carry is there then increment C by 1 |
| 410E |  | CMA | 2F | 1’s complement of accumulator data |
| 410F |  | ADI 01H | C6,01 | Add 1 to accumulator data |
| 410E | LOOP | STA 4302 | 32 (02,43) | Store the added value at the address 4302H |
| 4111 |  | MOV A,C | 79 | Copy the value of borrow to the accumulator from register C |
| 4112 |  | STA 4303 | 32 (03,43) | Store the value of borrow at the address 4303 from accumulator |
| 4115 |  | HLT | 76 | Stop the execution of program |

## SAMPLE INPUT OUTPUT

a. Without borrow

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input Address |  | Value |
|  | 4300 |  |  08 |
|  | 4301 |  |  03 |
|  |  |  |  |  |
| Output Address |  | Value |
|  | 4302 |  |  05 |
|  | 4303 |  | 00 (borrow) |

b. With borrow

|  |  |  |
| --- | --- | --- |
| Input Address |  | Value |
| 4300 |  |  4F |
| 4301 |  | FF |
|  |  |
| Output Address |  | Value |
| 4302 |  |  B0 |
| 4303 |  | 01 (borrow) |

## QUESTIONS:

1. What is the function of SUB?
2. What is the function of LDA?
3. What is the function of CMA?
4. What is the function of ADI?
5. Find the 2’s complement of 24H.
6. Write a program to find 2’s complement of a number stored at the address 2400H.

# EXPERIMENT 04(a)

AIM Multiplication of two 8 bit numbers using repeated addition method.

PREREQUISITES Following points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about increment and decrement instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

|  |  |
| --- | --- |
| Step 1 | Start the microprocessor |
| Step 2 | Get the 1st 8 bit numbers |
| Step 3 | Move the 1st 8it number to register ‘B’ |
| Step 4 | Get the 2nd 8 bit number |
| Step 5  | Move the 2nd 8 bit number to register ‘C’ (Counter) |
| Step 6 | Initialize the accumulator as zero |
| Step 7 | Initialize the carry as zero |
| Step 8 | Add both register ‘B’ value as accumulator |
| Step 9 | Jump on if no carry |
| Step 10 | Increment carry by 1 if there is |
| Step 11 | Decrement the 2nd value and repeat from step 8 till the 2nd value becomes zero |
| Step 12 | Store the multiplied value in memory |
| Step 13 | Move the carry value to accumulator |
| Step 14 | Store the carry value in memory |
| Step 15 | Stop execution of program |

## FLOWCHART

****

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4100 |  | LDA 4500H | 3A,00,45 | Load the first 8 bit number |
| 4103 |  | MOV B,A | 47 | Move the 1st bit data to register B |
| 4104 |  | LDA 4501 | 3A 01,45 | Load the 2nd number bit data |
| 4107 |  | MOV C,A | 4F | Move the 2nd bit data to register C (Counter) |
| 4108 |  | MVI A,00H | 3E, 00H | Initialize the accumulator as zero |
| 410A |  | MVI D, 00H | 16, 00H | Initialize the carry as Zero |
| 410C | LOOP | ADD B | 80 | Add the contents of register B with accumulator |
| 410D |  | JNC NEXT | D2 11,41 | Jump if no carry |
| 4110 |  | INR D | 14 | Increment carry if there is |
| 4111 | NEXT | DCR C | 0D | Decrement the Counter |
| 4112 |  | JNZ LOOP | C2 0C,41 | Jump until counter is not zero |
| 4115 |  | STA 4502H | 32 (02,45) | Store the result at the address 4502H |
| 4118 |  | MOV A,D | 7A | Move carry to accumulator |
| 4119 |  | STA 4503 | 32 (03,45) | Store the value of carry at the address 4503 from accumulator |
| 411C |  | HLT | 76 | Stop execution of program |

## SAMPLE INPUT OUTPUT

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input Address |  | Value |
|  |  4500 |  |  04 |
|  |  4501 |  |  03 |
|  |  |  |  |  |
| Output Address |  | Value |
|  |  4502 |  |  0C |
|  |  4503 |  | 00 (carry) |

## QUESTIONS

1. What is the function of counter?
2. What is conditional jump?
3. What is the function of JNZ?
4. Which flag is checked on execution of JNZ instruction?
5. State the function of INR and DCR.
6. Find the multiplication of 45H and 20H.

# EXPERIMENT 04(b)

AIM: Multiplication of a 8 bit number by 2 using rotation method.

PREREQUISITES: Following points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about rotation instruction.

## APPARATUS REQUIRED:

 8085 microprocessor kit.

## ALGORITHM:

|  |  |  |  |
| --- | --- | --- | --- |
| Step | 1 | : | Start the microprocessor |
| Step | 2 | : | Load the data of location 200A in the accumulator |
| Step | 3 | : | Set the carry flag to zero |
| Step | 4 | : | Rotate accumulator to left through carry |
| Step | 5 | : | Store the contents of accumulator into 200B |

## FLOWCHART:



## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4000 |  | LDA 200AH | 3A 0A 20 | Load 8-bit the number in accumulator |
| 4003 |  | STC | 37 | Set the carry flag to 1 |
| 4104 |  | CMC | 3F | Complement the carry flag |
| 4105 |  | RAL | 17 | Rotate accumulator to left through carry (Multiply by 2) |
| 4106 |  | STA 200B | 32 0B 20 | Store the result in 200B |
| 4109 |  | HLT | 76 | Stop the execution of program |

## **SAMPLE INPUT OUTPUT**

|  |  |
| --- | --- |
| Input Address | Contents |
| 200A | 04 |

|  |  |
| --- | --- |
| Output Address | Contents |
| 200B | 08 |

## QUESTIONS

1. What is the function of STC?
2. What is function of CMC?
3. What is the function of RAL?
4. Find the multiplication of 20H by 4.

# EXPERIMENT 05(a)

AIMDivision of two 8 bit numbers.

PREREQUISITES Following points should be noted before performing this experiment

1. Students must have knowledge of binary number division.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about increment and decrement instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Start the program by loading the HL pair registers with address of memory location.
2. Move the data to register B.
3. Load the 2nd data into accumulator.
4. Compare the two numbers to check the carry
5. Subtract the two numbers.
6. Increment the value of carry.
7. Check whether repeated subtraction is over.
8. Store the results in given memory location
9. Terminate the program.

## FLOWCHART

****

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4000 |  | LXI H,2050 | 21,50,20 | Load HL pair immediate from 2050H |
| 4003 |  | MOV B,M | 46 | Copy data from Memory to B |
| 4004 |  | MVI C,00 | 0E 00 | Clear the register C (Quotient) |
| 4006 |  | INX H | 23 | Increment HL pair |
| 4007 |  | MOV A,M | 7E | Copy data from memory to accumulator |
| 4008 | LOOP | CMP B | B8 | Compare content of B with accumulator |
| 4009 |  | JC NEXT | DA | Jump if carry (flag) |
| 400C |  | SUB B | 90 | A = A-B |
| 400D |  | INR C | 0C | C = C + 1 |
| 400E |  | JMP LOOP | C3 | Jump to Loop |
| 4011 | NEXT | STA 3050 | 32 50 30 | Store the remainder in address 3050H |
| 4014 |  | MOV A,C | 79 | Copy carry from C to A  |
| 4015 |  | STA 3051 | 32 51 30 | Store the Quotient in address 3051H |
| 4018 |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT AND OUTPUT

|  |  |
| --- | --- |
| Input Address | Contents |
| 2050 | FF |
| 2051 | FE |
| Input Address | Contents |
| 3050 | 01 |
| 3051 | 01 |

## QUESTIONS

1. What is the function of JMP?
2. What is function of LXI?
3. What is the function of SUB?
4. What is the function of INX?

# EXPERIMENT 05(b)

AIM Division of 8 bit number by 2 using rotation method.

PREREQUISITES Following points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about rotation instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

|  |  |  |  |
| --- | --- | --- | --- |
| Step | 1 | : | Start the microprocessor |
| Step | 2 | : | Load the data of location 200A in the accumulator |
| Step | 3 | : | Set the carry flag to zero |
| Step | 4 | : | Rotate accumulator to right through carry |
| Step | 5 | : | Store the contents of accumulator into 200B |

## FLOWCHART

****

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4000 |  | LDA 200AH | 3A 0A 20 | Load 8-bit the number in accumulator |
| 4003 |  | STC | 37 | Set the carry flag to 1 |
| 4004 |  | CMC | 3F | Complement the carry flag |
| 4005 |  | RAR | 1F | Rotate accumulator to right through carry (Multiply by 2) |
| 4006 |  | STA 200B | 32 0B 20 | Store the result in 200B |
| 4009 |  | HLT | 76 | Stop the execution of program |

## SAMPLE INPUT OUTPUT

|  |  |
| --- | --- |
| Input Address | Contents |
| 200A | 08 |

|  |  |
| --- | --- |
| Output Address | Contents |
| 200B | 04 |

## QUESTIONS

1. What is the function of STC?
2. What is function of CMC?
3. What is the function of RAR?
4. Find the division of 20H by 2.

# EXPERIMENT 06

AIMProgram to add two 8-bit BCD numbers.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about rotation instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Get the 1st BCD number.
2. Get the 2nd BCD number.
3. Add two BCD numbers
4. Adjust result to valid BCD number.
5. Check for carry. If carry then store in memory.
6. Store the result.
7. Stop

## FLOWCHART

NO

YES

Store the result

 Store the carry

Is carry set?

Adjust result to valid BCD number

A = A + B

Get the 2nd data into accumulator

Get the 1st data into a register (B) from memory

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4000 |  | MVI C,00H | 0E, 00 | Clear the register C (Carry) |
| 4002 |  | LDA 2050H | 3A 50 20 | Load data from address 2050H into accumulator |
| 4005 |  | MOV B,A | 47 | Move data from A to B |
| 4006 |  | LDA 2051 | 3A 51 20 | Load data from address 2051H into accumulator |
| 4009 |  | ADD B | 80 | Add content of B with A |
| 400A | LOOP | DAA | 27 | Add 06 if sum > 9 or AC = 1 |
| 400B |  | JNC NEXT | D2 | Jump if not carry (flag) |
| 400E |  | INR C | 0C | C = C + 1 |
| 400F |  | STA 2052H | 32 52 20 | Store content of accumulator at address 2052H |
| 4012 |  | MOV A,C | 79 | Move content from carry register to accumulator |
| 4013 | NEXT | STA 2053H | 32 5320 | Store content of accumulator into memory |
| 4016 |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT OUTPUT

|  |  |
| --- | --- |
| Input Address | Contents |
| 2050 | 42 |
| 2051 | 28 |

|  |  |
| --- | --- |
| Output Address | Contents |
| 2052 | 70 |

## QUESTIONS

1. What is BCD number?
2. What is function of DAA?
3. Find the BCD addition of 28H and 82H.

# EXPERIMENT 07

AIMAddition of two 16-bit numbers.

PREREQUISITES Following points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about rotation instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Load the first number in HL.
2. Move the content of HL to DE.
3. Load the second number in HL.
4. Add the two numbers.
5. Store the result.
6. Stop

## FLOWCHART

Store the result

Store the result

Move the content of HL to DE

Store the result

Add the two numbers using DAD

Load the 2nd number in HL

Get the 1st number in HL.

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics/Operand** | **Hexcodes** | **Comments** |
| 4000 |  | LHLD 2000 | 2A 00 20 | Load first 16 bit from 2000H and 2001H to HL register pair |
| 4003 |  | XCHG | EB | Exchange the contents of HL and DE register pair. |
| 4004 |  | LHLD 2002 | 2A 02 20 | Load first 16 bit from 2002H and 2003H to HL register pair |
| 4007 |  | DAD D | 19 | Add contents of HL and DE.  |
| 4008 |  | SHLD 2004 | 22 04 20 | Store the result in 2004H and 2005H |
| 400B |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT OUTPUT

|  |  |
| --- | --- |
| Input Address | Contents |
| 2000 | 42 |
| 2001 | 28 |
| 2002 | 22 |
| 2003 | 42 |

|  |  |
| --- | --- |
| Output Address | Contents |
| 2004 | 64 |
| 2005 | 6A |

## QUESTIONS

1. What is the function of LHLD?
2. What is the function of XCHG?
3. What is the function of SHLD?
4. What is the function of DAD D?
5. Find the addition of CAA7H and 6BB9H.
6. Write a program to find subtraction of two 16-bit numbers.

# EXPERIMENT 08

AIMProgram to find the largest number in array of data.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of binary number multiplication.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.
6. Students must know about rotation instruction.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Take the first element of array in A.
2. Compare it with other elements of array.
3. If A is smaller, then store that element in A otherwise compare with next element.
4. The value of A is the largest number.
5. Stop

## FLOWCHART

No

Yes

No

Yes

Store the maximum from B into memory

C = 0?

Z = 0?

C = C - 1

B = A

Store the maximum from B into memory

C = Block Size

B = Load the first number

HL = get the address of next number

A > B

A = item of address pointed by HL

HL = HL + 1

C = C - 1

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **Hexcodes** | **Comments** |
| 4000 |  | LXI H 8000H | 0E, 00 | Point to get array size |
| 4003 |  | MOV C,M | 3A 50 20 | Get the size of array |
| 4004 |  | INX H | 47 | Point to actual array |
| 4005 |  | MOV B,M | 3A 51 20 | Load the first number into B |
| 4006 |  | DCR C | 80 | C = C - 1 |
| 4007 | LOOP | INX H | 27 | Point to next location |
| 4008 |  | MOV A,M | D2 | Get the next number from memory to accumulator |
| 4009 |  | CMP B | 0C | Compare A with B |
| 400A |  | JC SKIP | 32 52 20 | If A < B then skip |
| 400D |  | MOV B,A | 79 | If CY = 0 then update B |
| 400E | SKIP | DCR C | 32 53 20 | C = C - 1 |
| 400F |  | JNZ LOOP | 76 | Count is not zero then go to LOOP |
| 4012 |  | LXI H, 9000H |  | Point to destination address |
| 4015 |  | MOV M,B |  | Store the largest number |
| 4016 |  | HLT |  | Terminate the program |

## SAMPLE INPUT AND OUTPUT

**Input:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 8000 | 06 |
| 8001 | 44 |
| 8002 | 22 |
| 8003 | 4F |
| 8004 | 1F |
| 8005 | 20 |
| 8006 | 32 |

**Output:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 9000 | 4F |

## QUESTIONS

1. What is the function of JC?
2. Write a program to find smallest number in given array.

|  |  |
| --- | --- |
| **Address** | **Data** |
| 3000 | 6 |
| 3001 | 23 |
| 3002 | 12 |
| 3003 | 5F |
| 3004 | 10 |
| 3005 | 1A |
| 3006 | 20 |

# EXPERIMENT 09

AIMProgram to check whether a given number is even or odd.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of binary number AND operation.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.

# APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Load the number in accumulator from the address.
2. Perform AND operation with the accumulator content and 01.
3. If the content of A is 00H then the number is even otherwise it is odd.
4. Stop.

## FLOWCHART

Yes

No

Load the number in accumulator from memory

EVEN

ODD

A =0?

A = (A)AND(01)

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics/Operand** | **Hexcodes** | **Comments** |
| 4000 |  | LDA 2050 | 3A 50 20 | Load 8 bit number from 2050H to accumulator. |
| 4003 |  | ANI 01H | E6 01 | Logical AND operation of accumulator with 01H. |
| 4005 |  | JZ NEXT | CA 0D 40 | If ZF=1 then jump to NEXT |
| 4008 |  | MVI A, 11 | 3E 11 | Move 11H in accumulator for odd number |
| 400A |  | JMP SKIP | C3 0F 40 | Jump to SKIP |
| 400D | NEXT | MVI A, 22 | 3E 22 | Move 22H in accumulator for even number |
| 400F | SKIP | STA 2051 | 32 51 20 | Store the result in address 2051H |
| 4012 |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT AND OUTPUT

**Input:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 2050 | 0A |

**Output:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 2051 | 22 |

## QUESTIONS

1. What is the function of JZ and JMP?
2. What is the function of ANI?
3. Find the even numbers in 10 memory locations starting from the address 2320H and store in memory locations starting from 1060H.

# EXPERIMENT 10

AIMProgram to countnumbers of 1’s in the given 8-bit number.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of RAR and RAL.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the conditional jumps.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Get the number in accumulator.
2. Set the counter C to 08H.
3. Set count D = 00H.
4. Rotate left or right it through carry.
5. C = C - 1
6. If CY = 1, D = D + 1
7. If counter is not zero then go to step 4.
8. Store the numbers of counts in memory.
9. Stop.

## FLOWCHART

RAR

Load 8-bit number in accumulator

Counter (C) = 08H

Count (D) = 00H

Is CY = 1?

 No

Yes

Yes

No

Is counter zero ?

Store the count in memory

C = C - 1

D = D + 1

## SOUCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics/Operand** | **Hexcodes** | **Comments** |
| 4000 |  | MVI B,00H | 06 00 | Load 00H in B (Count) |
| 4002 |  | MVI C,08H | 0E 08 | Load 08H in C (Counter) |
| 4004 |  | LDA 3000 | 3A 00 30 | Load contents of 3000H in accumulator |
| 4007 | LOOP | RAR | 1F | Rotate accumulator right with carry |
| 4008 |  | JNC SKIP | D2 0C 40 | Jump to SKIP if CY = 0 |
| 400B |  | INR B | 04 | B = B + 1 |
| 400C | SKIP | DCR C | 0D | C = C - 1 |
| 400D |  | JNZ LOOP | C2 07 40 | If counter = 0, then jump to LOOP |
| 4010 |  | MOV A,B | 78 | Move content of B (count) to accumulator |
| 4011 |  | STA 3001H | 32 01 30 | Store value of count at address 3001H |
| 4014 |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT AND OUTPUT

**Input:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 3000 | 0A |

**Output:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 3001 | 02 |

## QUESTIONS

1. What is the function of JNC and JC?
2. What is the difference between RAR and RRC?
3. Write a program to count number zeros in given 8-bit data.
4. Write a program to count number of 1’s and 0’s after addition of two 8-bit data stored at the address 4000H and 40001H.

# EXPERIMENT 11

AIMProgram to pack two unpacked BCD numbers stored in memory locations 2000H and 2001H and store result in memory location 2300H. Assume the least significant digit is stored at 2000H.

PREREQUISITESFollowing points should be noted before performing this experiment

1. Students must have knowledge of RLC and RRC.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about logical operations.
6. Student must know about BCD numbers.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Load the number for MSB of BCD digit.
2. Rotate 4 times to the left without carry and make LSB part of it zero.
3. Load the number for LSB of BCD digit.
4. Add it to the rotated number.
5. Store the result.
6. Stop.

## FLOWCHART

Store the result

Add the number for LSB BCD digit to rotated number

Rotate 4 times to the left without carry and make LSB zero

Load the number for MSB BCD digit

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics/Operand** | **Hexcodes** | **Comments** |
| 4000 |  | LDA 2001H | 3A 00 20 | Load the MSB BCD digit. |
| 4003 |  | RLC | 07 | Rotate to the left without carry |
| 4004 |  | RLC | 07 | Rotate to the left without carry |
| 4005 |  | RLC | 07 | Rotate to the left without carry |
| 4006 |  | RLC | 07 | Rotate to the left without carry |
| 4007 |  | ANI F0H | E6 F0 | Make least significant BCD digit zero |
| 4009 |  | MOV C,A | 4F | store the partial result |
| 400A |  | LDA 2000H | 3A 01 20 | Get the lower BCD digit |
| 400D |  | ADD C | 81 | Add lower BCD digit |
| 400E |  | STA 2300H | 32 00 23 | Store the result |
| 4011 |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT AND OUTPUT

**Input:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 2000 | 04 |
| 2001 | 05 |

**Output:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 2300 | 54 |

## QUESTIONS

1. What is BCD number?
2. What is packed and unpacked BCD numbers?
3. Write a program to unpack a packed BCD number.
4. Find the data stored at the address 4000H and 4001H after the execution of the program.

|  |
| --- |
| MVI B,45H |
| MOV A,B |
| ADI 0FH |
| STA 4000H |
| MOV A,B |
| ADI F0H |
| RRC |
| RRC |
| STA 4001H |
| HLT |

# EXPERIMENT 12

AIMProgram to convert binary to gray code.

PREREQUISITES Following points should be noted before performing this experiment

1. Students must have knowledge how to set and reset the carry.
2. Students must know about the microprocessor registers and arithmetic operations.
3. Students must know about the opcodes and their hex codes.
4. Students must have knowledge regarding data movement.
5. Students must know about the logical operation.

## APPARATUS REQUIRED

 8085 microprocessor kit.

## ALGORITHM

1. Set the Carry Flag (CY) to 0.
2. Load the data from address 2050 in A.
3. Move the data of A(accumulator) into register B.
4. Rotate the bits of A to right.
5. XOR the contents of register A and B.
6. Store the result at memory address 3050.
7. Stop

## FLOWCHART

Store the result

XOR the contents of A and B

Rotate the bits of A to right.

Move the data of A(accumulator) into register B.

Load the data in accumulator from memory

Reset the carry

## SOURCE CODE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics/Operand** | **Hexcodes** | **Comments** |
| 4000 |  | STC | 37 | Set the carry flag |
| 4001 |  | CMC | 3F | Compliment the carry |
| 4002 |  | LDA 2050H | 3A 50 20 | Load accumulator from address 2050H |
| 4005 |  | MOV B,A | 47 | Copy contents of accumulator to B |
| 4006 |  | RAR | 1F | Rotate to the right with carry |
| 4007 |  | XRA B | A8 | A=A ⊕ B |
| 4008 |  | STA 3050H | 32 5030 | Store the result at address 3050H |
| 400B |  | HLT | 76 | Terminate the program |

## SAMPLE INPUT AND OUTPUT

**Input:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 2050 | 74 |

**Output:**

|  |  |
| --- | --- |
| **Address** | **Data** |
| 3050 | 4E |

## QUESTIONS

1. What is gray code?
2. What is gray code of (0101 1111)2?

# APPENDIX I

Opcode sheet for 8085 Microprocessor with description

|  |  |  |
| --- | --- | --- |
| **Mnemonic** | **Opcode** | **Description** |
| ACI DATA | CE | Add with carry immediate |
| ADC REG | 8F | Add with carry |
| ADC M | 8E | Add with carry to memory |
| ADD REG | 87 | Add Register to accumlator |
| ADD M | 86 | Add to memory |
| ADI DATA | C6 | Add Immediate |
| ANA REG | A7 | AND Accumulator |
| ANA M | A6 | AND Accumulator and memory |
| ANI DATA | E6 | AND Immediate |
| CALL ADDR | CD | Call unconditional |
| CC ADDR | DC | Call on carry |
| CM ADDR | FC | Call on minus |
| CMA | 2F | Complement Accumulator |
| CMC | 3F | Complement carry |
| CMP REG | BF | Compare accumulator with register |
| CMP M | BF | Compare with memory |
| CNC ADDR | D4 | Call on no carry |
| CNZ ADDR | C4 | Call on no zero |
| CP ADDR | F4 | Call on plus |
| CPE ADDR | EC | Call on parity even |
| CPI DATA | FE | Compare immediate |
| CPO ADDR | E4 | Call on parity odd |
| CZ ADDR | CC | Call on zero |
| DAA | 27 | Decimal Adjust Accumulator |
| DAD B | 09 | Double Add BC to HL |
| DAD D | 19 | Double Add DE to HL |
| DAD H | 29 | Double Add HL to HL |
| DAD SP | 39 | Double Add SP to HL |
| DCR REG | 3D | Decrement register |
| DCR M | 35 | Decrement memory |
| DCX B | 0B | Decrement BC |
| DCX D | 1B | Decrement DE |
| DCX H | 2B | Decrement HL |
| INR REG | 3C | Increment register |
| INR M | 3C | Increment memory |
| INX B | 03 | Increment BC |
| INX D | 13 | Increment DE |
| INX H | 23 | Increment HL |
| INX SP | 33 | Increment Stack Point |
| JMP ADDR | C3 | Jump unconditional |
| JC ADDR | DA | Jump on carry |
| JM ADDR | FA | Jump on Minus |
| JNC ADDR | D2 | Jump on No carry |
| JNZ ADDR | C2 | Jump on non zero |
| JP ADDR | F2 | Jump on plus |
| JPE ADDR | EA | Jump on parity even |
| JPO ADDR | E2 | Jump on parity odd |
| JZ ADDR | CA | Jump on zero |
| LDA ADDR | 3A | Load Accumulator direct |
| LDAX B | 0A | Load Accumulator indirect |
| LDAX D | 1A | Load Accumulator indirect |
| LHLD ADDR | 2A | Load HL direct |
| LXI B, 16 bit | 01 | Load immediate BC |
| LXI D, 16 bit | 11 | Load immediate DE |
| LXI H, 16 bit | 21 | Load immediate HL |
| LXI SP, 16 bit | 31 | Load immediate Stack pointer |
| MOV R1,R2 | 7F | Move register to register |
| MOV M,R | 77 | Move register to memory |
| MOV R,M | 7E | Move memory to register |
| MVI R, | 3E | Move immediate |
| MVI M,DATA | 36 | Move immediate to memory |
| NOP | 00 | No operation |
| ORA REG | B7 | Inclusive OR Accumulator |
| ORA M | B6 | Inclusive OR Accumulator |
| ORI DATA | F6 | Inclusive OR Immediate |
| OUT PORT | D3 | Output |
| PCHL | E9 | Jump HL indirect |
| POP B | C1 | Pop BC |
| POP D | D1 | Pop DE |
| POP H | E1 | Pop HL |
| POP PSW | F1 | Pop program status word |
| PUSH B | C5 | Push BC |
| PUSH D | D5 | Push DE |
| PUSH H | E5 | Push HL |
| PUSH PSW | F5 | Push program status word |
| RAL | 17 | Rotate accumulator left through carry |
| RAR | 1F | Rotate accumulator right through carry |
| RET | C9 | Return |
| RC | D8 | Return on carry |
| RIM | 20 | Read interrupt mask |
| RM | F8 | Return on minus |
| RNC | D0 | Return on No Carry |
| RNZ | C0 | Return on non zero |
| RP | F0 | Return on plus |
| RPE | E8 | Return on parity even |
| RPO | E0 | Return on parity odd |
| RZ | C8 | Return on zero |
| RLC | 07 | Rotate accumulator left  |
| RRC | 0F | Rotate accumulator right  |
| RST | C7 | Restart |
| SBB REG | 9F | Subtract with borrow |
| SBB M | 9E | Subtract with borrow |
| SBI DATA | DE | Subtract with borrow immediate |
| SHLD ADDR | 22 | Store HL pair direct |
| SIM | 30 | Set Interrupt mask |
| SPHL | F9 | Move HL to SP |
| STA ADDR | 32 | Store accumulator |
| STAX B | 02 | Store accumulator indirect in memory address BC |
| STAX D | 12 | Store accumulator indirect in memory address DE |
| STC | 37 | Set carry |
| SUB REG | 97 | Subtract register from accumulator |
| SUB M | 96 | Subtract memory from accumulator |
| SUI DATA | D6 | Subtract immediate data |
| XCHG | EB | Exchange HL with DE |
| XRA REG | AF | Exclusive OR register with accumulator |
| XRA M | AE | Exclusive OR memory with accumulator |
| XRI DATA | EE | Exclusive OR immediate data |
| XTHL | E3 | Exchange stack top with HL pair |