

8-Bit Arithmetic Operations using 8085

Aim:

To perform 8-bit arithmetic operations such as addition, subtraction, multiplication, and division using the 8085 microprocessor.

Apparatus Required:

- Laptop with internet connection

Algorithm:

For Addition (With Carry Consideration):

- Load the first number into register A.
- Load the second number into register B.
- Add the contents of registers A and B.
- If carry is generated, store carry in a separate location.
- Store the sum in another location.

For Subtraction (Considering Greater Number):

- Load the first number into register A.
- Load the second number into register B.
- Compare A and B.
- If $A < B$, swap the values of A and B to ensure positive result.
- Subtract the content of B from A.
- Store the result in a specified location.

For Multiplication:

- Load the first number into register A.
- Load the second number into register B.
- Multiply A and B using repeated addition.
- Store the result in suitable locations (including extra space if needed for higher bits).

For Division:

- Load the dividend into register A.
- Load the divisor into register B.
- Perform division using repeated subtraction.

- Store the quotient in one location and remainder in another.

Program:

Addition of Two 8-bit Numbers:

```
IN 01H    ; Read input from port 0x01 into A
MOV B,A    ; Save it in B
IN 02H    ; Read input from port 0x02 into A
ADD B      ; A = A + B
OUT 03H    ; Send sum to port 0x03
HLT        ; Stop execution
```

Subtraction (First number - Second number)

```
IN 01H    ; Read first number (A) from port 0x01 into A
MOV B,A    ; Save it in B
IN 02H    ; Read second number (B) from port 0x02 into
A
MOV C,A    ; Save B in C
MOV A,B    ; Restore first number into A
SUB C      ; A = A - B
OUT 03H    ; Output result to port 0x03
HLT        ; End program
```

Multiplication using repeated addition:

```
IN 01H    ; Read first number (Multiplicand) into A
MOV C, A   ; Store in C

IN 02H    ; Read second number (Multiplier) into A
MOV B, A   ; Store in B

MVI A, 00H ; Clear A to hold result

LOOP:
ADD C      ; A = A + C
DCR B      ; B = B - 1
JNZ LOOP   ; Repeat until B = 0

OUT 06H    ; Output the result to port 06H
```

HLT ; End of program

Division (Using Repeated Subtraction):

IN 01H ; Read dividend into A
MOV C, A ; Store dividend in C (for remainder tracking)
MVI A, 00H ; Clear A for quotient
MOV D, A ; Use D to store quotient

IN 02H ; Read divisor into A
MOV B, A ; Store divisor in B

DIV_LOOP:
MOV A, C ; Load current remainder into A
CMP B ; Compare remainder with divisor
JC END_DIV ; If A < B, jump to END_DIV
SUB B ; A = A - B
MOV C, A ; Update remainder in C
INR D ; Increment quotient
JMP DIV_LOOP ; Repeat loop

END_DIV:
MOV A, D ; Move quotient to A
OUT 03H ; Output quotient to port 03H

MOV A, C ; Move remainder to A
OUT 04H ; Output remainder to port 04H

HLT ; End program

Output:

Addition of Two 8-bit Numbers:

Input Ports:

- **01H** → First number
- **02H** → Second number

Output Ports:

- **03H** → Sum
- **04H** → Carry (if generated)

The screenshot displays a microcontroller development environment with three main panels:

- I/O Ports:** A list of ports from 0x00 to 0x10. Port 0x01 is set to 20, 0x02 to 10, and 0x03 to 30. All other ports are set to 00.
- Assembly Code:** A list of instructions:

```
1 IN 01H ; Read input from port 0x01 into A
2 MOV B,A ; Save it in B
3
4 IN 02H ; Read input from port 0x02 into A
5 ADD B ; A = A + B
6
7 OUT 03H ; Send sum to port 0x03
8 HLT ; Stop execution
9
```
- Machine Code:** A table showing the compiled machine code:

Line	Address	Machine Code	Source Code
1	0x0	DB 01	IN 01H ; Read in
2	0x2	47	MOV B,A ; Save it
3			
4	0x3	DB 02	IN 02H ; Read in
5	0x5	80	ADD B ; A = A +
6			
7	0x6	D3 03	OUT 03H ; Send su
8	0x8	76	HLT ; Stop ex
9			

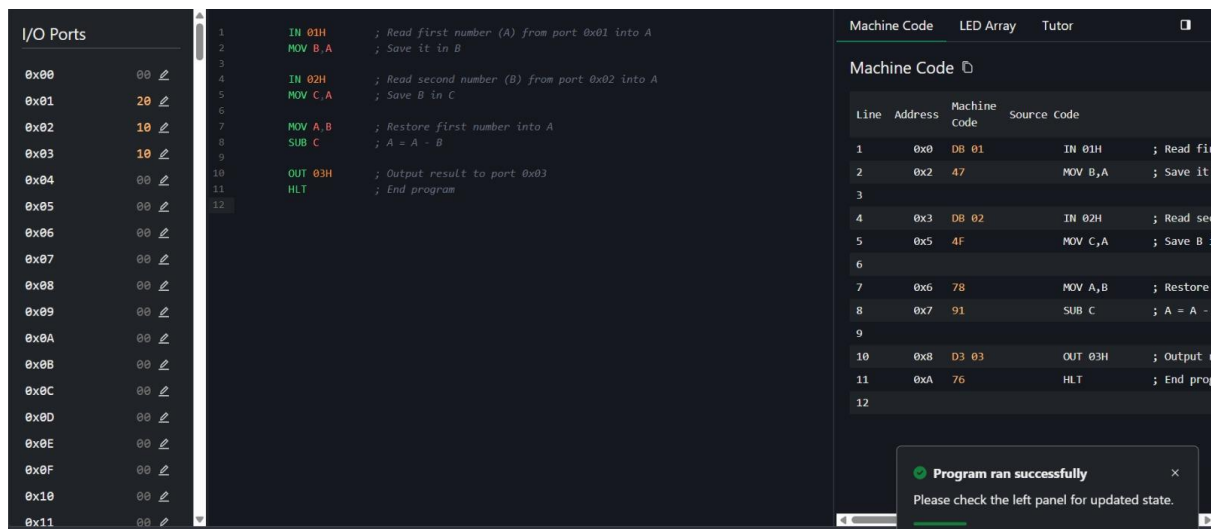
Subtraction (First number - Second number)

Input Ports:

- **01H** → First number
- **02H** → Second number

Output Ports:

- **03H** → Result (Difference)



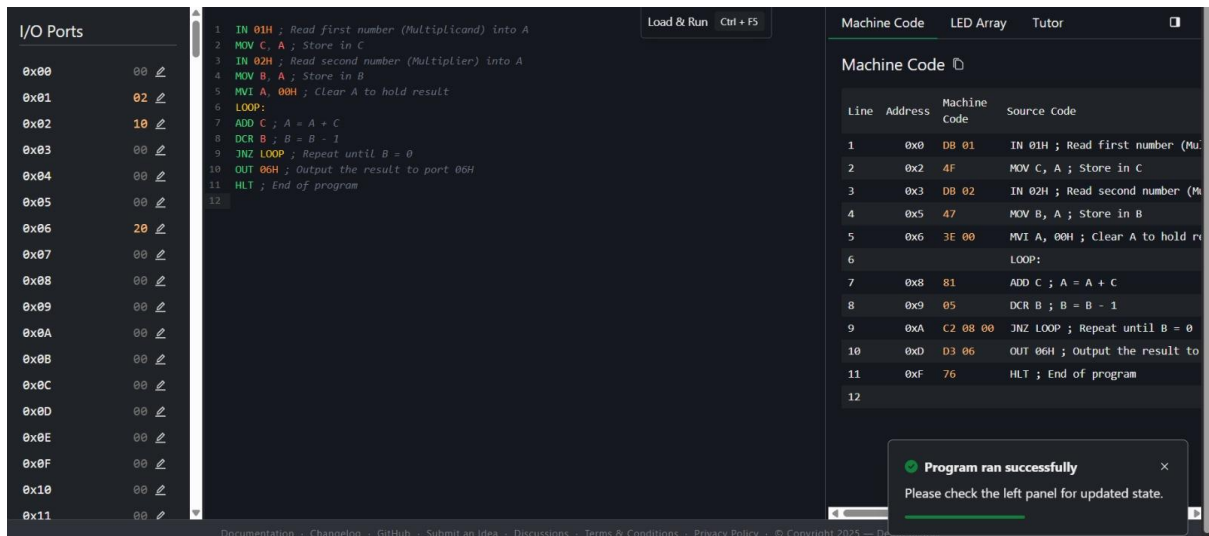
Multiplication using repeated addition:

Input Ports:

- **01H** → Multiplicand
- **02H** → Multiplier

Output Ports:

- **06H** → Product



Division (Using Repeated Subtraction):

Input Ports:

- **01H** → Dividend
- **02H** → Divisor

Output Ports:

- **03H** → Quotient
- **04H** → Remainder

I/O Ports				Machine Code				LED Array		Tutor		
0x00	00			1	IN 01H		; Read dividend into A					
0x01	20			2	MOV C, A		; Store dividend in C (for remainder tracking)					
0x02	10			3	MVI A, 00H		; Clear A for quotient					
0x03	02			4	MOV D, A		; Use D to store quotient					
0x04	00			5								
0x05	00			6	IN 02H		; Read divisor into A					
0x06	00			7	MOV B, A		; Store divisor in B					
0x07	00			8								
0x08	00			9	DIV_LOOP:							
0x09	00			10	MOV A, C		; Load current remainder into A					
0x0A	00			11	CMP B		; Compare remainder with divisor					
0x0B	00			12	JC END_DIV		; If A < B, jump to END_DIV					
0x0C	00			13	SUB B		; A = A - B					
0x0D	00			14	MOV C, A		; Update remainder in C					
0x0E	00			15	INR D		; Increment quotient					
0x0F	00			16	JMP DIV_LOOP		; Repeat loop					
0x10	00			17								
				18	END_DIV:							
				19	MOV A, D		; Move quotient to A					
				20	OUT 03H		; Output quotient to port 03H					
				21								
				22	MOV A, C		; Move remainder to A					
				23	OUT 04H		; Output remainder to port 04H					
				24								
				25	HLT		; End program					

Result:

The 8-bit arithmetic operations using the 8085 microprocessor have been successfully executed and verified using memory access for input and output.