

## **8-BIT ARITHMETIC OPERATIONS USING 8051**

**Exp No.:** 12

**Name:** S Vishakan

**Date:** 21-10-2020

**Reg. No:** 18 5001 196

---

**AIM:**

To write assembly language programs to perform the following arithmetic operations using an 8051 microcontroller:

1. 8-bit addition
2. 8-bit subtraction
3. 8-bit multiplication
4. 8-bit division

### **PROGRAM – 1: 8-BIT ADDITION:**

#### **ALGORITHM:**

1. Begin
2. Initialize R0 with 00h.
3. Move the value in R1 to A.
4. Add the value in A to with value in R2.
5. Increment R0 if carry is produced.
6. Move R0 to R3 (carry) and A to R4 (sum).
7. End.

PROGRAM	COMMENTS
MOV R0, #00	R0 has address of 0x00
MOV A, R1	0x01 has 1st 8-bit number
ADD A, R2	0x02 has 2nd 8-bit number. Add it with A
JNC LABEL	If no carry, jump to "LABEL".
INC R0	If carry, increment R0
LABEL:	
MOV R4, A	Move output to R4 from A
MOV 03, R0	Move carry to R3. (MOV R3, R0) is invalid
HALT:	
SJMP HALT	Halt the program with a loop.

## SAMPLE I/O SNAPSHOT:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBVF

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x00	0x00	0x00	0x00

RxD	TxD	TMOD	R4	ACC
1	1	0x00	0xFE	0xFE

SCON	TCON	R3	PSW
0x00	0x00	0x01	0xC1

pins	bits	TH1	TL1	R2	IP
0xFF	0xFF	0x00	0x00	0xFF	0x00

bits	PC	R1	IE
0xFF	0x000A	0xFF	0x00

bits	P0	R0	PCON
0xFF	0xFF	0x01	0x00

8051

Modify RAM

addr	0x01	0xFF	value
0	00	01	FF
1	FF	01	FE
2	00	00	00
3	00	00	00
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00

Copyright ©2005-2016 James Rogers

Remove All Breakpo...

RST Step Run New Load Save Copy Paste

Time: 31us - Instructions: 18

8051 ALP TO ADD 2 8-BIT NUMBE

```
0000| MOV R0, #00 ;R0 has ac
0002| MOV A, R1 ;0x01 has
0003| ADD A, R2 ;0x02 has
0004| JNC LABEL
0006| INC R0 ;If carry,
LABEL:
0007| MOV R4, A ;Move outp
0008| MOV 03, R0 ;Move carz
HALT:
000A| SJMP HALT
```

P0.7	1	Display-select Decoder CS DAC WR
P0.6	1	Keypad Column 2
P0.5	1	Keypad Column 1
P0.4	1	Keypad Column 0
P0.3	1	Keypad Row 3
P0.2	1	Keypad Row 2
P0.1	1	Keypad Row 1
P0.0	1	Keypad Row 0
P1.7	1	LED 7 Seg. dp DAC DB7 LCD DB7
P1.6	1	LED 6 Seg. g DAC DB6 LCD DB6
P1.5	1	LED 5 Seg. f DAC DB5 LCD DB5
P1.4	1	LED 4 Seg. e DAC DB4 LCD DB4
P1.3	1	LED 3 ... d ..DB3 ..DB3 .. RS
P1.2	1	LED 2 ... c ..DB2 ..DB2 LCD E
P1.1	1	LED 1 Seg. b DAC DB1 LCD DB1
P1.0	1	LED 0 Seg. a DAC DB0 LCD DB0
P2.7	1	SW 7 ADC DB7
P2.6	1	SW 6 ADC DB6
P2.5	1	SW 5 ADC DB5
P2.4	1	SW 4 ADC DB4
P2.3	1	SW 3 ADC DB3
P2.2	1	SW 2 ADC DB2
P2.1	1	SW 1 ADC DB1
P2.0	1	SW 0 ADC DB0
P3.7	1	ADC RD Comparator Output
P3.6	1	ADC WR
P3.5	1	Motor Sensor
P3.4	1	Display-select Input 1
P3.3	1	AND Gate Output Display-se..t 0
P3.2	1	ADC INTR
P3.1	1	Motor Control Bit 1 Ext. UART Rx
P3.0	1	Motor Control Bit 0 Ext. UART Tx

DI LD

1 2 3 AND Gate Disabl...

4 5 6 Key Bounce Disabl...

7 6 5 4 3 2 1 0

Standard

U Odd Parity 8-bit UART @ 4800 Baud

Rx Rx Reset

Tx Tx Send

0.0 V output

Scope DAC

BF 0 AC 0x00 IR 0x00 DR 0x00

0.0 V input

11111111

ADC

MAX MIN

Motor Enabled

## **PROGRAM – 2: 8-BIT SUBTRACTION:**

### **ALGORITHM:**

1. Begin.
2. Initialize R0 with 00h
3. Move the value in R1 to A.
4. Subtract the value in A to with value in R2.
5. Increment R0 if carry is produced and take 2's complement of A.
6. Move R0 to R3 (borrow) and A to R4 (difference)
7. End.

PROGRAM	COMMENTS
MOV R0, #00	R0 has address of 0x00
MOV A, R1	0x01 has 1st 8-bit number
SUBB A, R2	;0x02 has 2nd 8-bit number. Subtract it from A.
JNC LABEL	If no carry, jump to "LABEL".
INC R0	If carry, increment R0
CPL A	1's complement the difference
INC A	2's complement the difference
LABEL:	
MOV R4, A	Move output to R4 from A
MOV 03, R0	Move carry to R3. (MOV R3, R0) is invalid
HALT:	
SJMP HALT	Halt the program with a loop.

## SAMPLE I/O SNAPSHOT:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 Update Freq. 1

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0x00
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x01
R/D	TXD	TMOD	0x00	R5	0x00	PSW	0xC1
1	1	0x00	0x00	R4	0x01	IP	0x00
SCON	0x00	TCON	0x00	R3	0x01	IE	0x00
				R2	0xFF	PCON	0x00
pins	bits	TH1	TL1	R1	0xFE	DPH	0x00
0xFF	0xFF	0x00	0x00	R0	0x01	DPL	0x00
0xFF	0xFF	PC	0x000C			SP	0x07
0xFF	0xFF						
0xFF	0xFF						

8051

Modify RAM

Data Memory

addr	0x05	0x00	value
0	1	2	3
00	01	FE	FF
01	01	01	00
02	00	00	00
03	00	00	00
04	00	00	00
05	00	00	00
06	00	00	00
07	00	00	00
08	00	00	00
09	00	00	00
0A	00	00	00
0B	00	00	00
0C	00	00	00
0D	00	00	00
0E	00	00	00
0F	00	00	00
10	00	00	00
11	00	00	00
12	00	00	00
13	00	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	00	00	00
19	00	00	00
1A	00	00	00
1B	00	00	00
1C	00	00	00
1D	00	00	00
1E	00	00	00
1F	00	00	00

Remove All Breakpo...

RST Step Run New Load Save Copy Paste

Time: 21us - Instructions: 14

```

;8051 ALP TO SUBTRACT 2 8-BIT
0000| MOV     R0, #00      ;R0 has ac
0002| MOV     A, R1        ;0x01 has
0003| SUBB    A, R2        ;0x02 has
0004| JNC     LABEL
0006| INC     R0           ;If borrow
0007| CPL     A            ;1's compl
0008| INC     A            ;2's compl

LABEL:
0009| MOV     R4, A        ;Move outp
000A| MOV     03, R0       ;Move bozz

HALT:
000C| SJMP    HALT
  
```

P0.7 1 Display-select Decoder CS|DAC WR  
P0.6 1 Keypad Column 2  
+0.5 1 Keypad Column 1  
P0.4 1 Keypad Column 0  
P0.3 1 Keypad Row 3  
P0.2 1 Keypad Row 2  
P0.1 1 Keypad Row 1  
P0.0 1 Keypad Row 0  
P1.7 1 LED 7|Seg. dp|DAC DB7|LCD DB7  
P1.6 1 LED 6|Seg. g|DAC DB6|LCD DB6  
+1.5 1 LED 5|Seg. f|DAC DB5|LCD DB5  
P1.4 1 LED 4|Seg. e|DAC DB4|LCD DB4  
P1.3 1 LED 3|... d|..DB3|..DB3|.. RS  
P1.2 1 LED 2|... c|..DB2|..DB2|LCD E  
P1.1 1 LED 1|Seg. b|DAC DB1|LCD DB1  
P1.0 1 LED 0|Seg. a|DAC DB0|LCD DB0  
P2.7 1 SW 7|ADC DB7  
P2.6 1 SW 6|ADC DB6  
+2.5 1 SW 5|ADC DB5  
P2.4 1 SW 4|ADC DB4  
P2.3 1 SW 3|ADC DB3  
P2.2 1 SW 2|ADC DB2  
P2.1 1 SW 1|ADC DB1  
P2.0 1 SW 0|ADC DB0  
P3.7 1 ADC RD|Comparator Output  
P3.6 1 ADC WR  
+3.5 1 Motor Sensor  
P3.4 1 Display-select Input 1  
P3.3 1 AND Gate Output|Display-se..t 0  
P3.2 1 ADC INTR  
P3.1 1 Motor Control Bit 1|Ext. UART Rx  
P3.0 1 Motor Control Bit 0|Ext. UART Tx

DI LD

7 6 5 4 3 2 1 0

0.0 V output

Scope

DAC

BF 0 AC 0x00 IR 0x00 DR 0x00

AND Gate Disabl...

Key Bounce Disabl...

Standard

U Odd Parity 8-bit UART @ 4800 Baud

Rx Rx Reset

Tx Tx Send

0.0 V input

1111111

ADC

MAX

MIN

Motor Enabled

### **PROGRAM – 3: 8-BIT MULTIPLICATION:**

#### **ALGORITHM:**

1. Begin.
2. Initialize R0 with 00h
3. Move the value in R1 to A.
4. Move the value in R2 to B.
5. Multiply A and B.
6. Move B to R4 (MSB of product) and A to R5 (LSB of product)
7. End.

PROGRAM	COMMENTS
MOV R0, #00	R0 has address of 0x00
MOV A, R1	0x01 has 1st 8-bit number
MOV B, R2	0x02 has 2nd 8-bit number
MUL AB	BA = A * B
MOV R5, A	Move lower byte to R5 from A
MOV R4, B	Move higher byte to R4 from B
HALT:	
SJMP HALT	Halt the program with a loop.

## SAMPLE I/O SNAPSHOT:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x00	0x00	0x00	0xFF

RXD	TXD	TMOD	TCON	R6	ACC
1	1	0x00	0x00	0x00	0x01

SCON	PCON	R5	PSW
0x00	0x00	0x01	0x05

R4	IP	R3	IE
0xFF	0x00	0x00	0x00

R2	DPH	R1	DPL
0xFF	0x00	0xFF	0x00

R0	SP
0x00	0x07

pins bits TH1 TL1 PC 8051 PSW 0 0 0 0 0 1 0 1

Modify RAM

Data Memory	addr	0x05	0x00	value
0	0	00	00	00
1	1	00	00	00
2	2	00	00	00
3	3	00	00	00
4	4	00	00	00
5	5	00	00	00
6	6	00	00	00
7	7	00	00	00
8	8	00	00	00
9	9	00	00	00
A	A	00	00	00
B	B	00	00	00
C	C	00	00	00
D	D	00	00	00
E	E	00	00	00
F	F	00	00	00

Copyright ©2005-2016 James Rogers Remove All Breakpo...

RST Step Run New Load Save Copy Paste

Time: 25us - Instructions: 13

```
;8051 ALP TO MULTIPLY 2 8-BIT
0000| MOV R0, #00 ;R0 has ac
0002| MOV A, R1 ;0x01 has
0003| MOV B, R2 ;0x02 has
0005| MUL AB ;BA = A *
0006| MOV R5, A ;Move lowe
0007| MOV R4, B ;Move high
HALT:
0009| JMP HALT
```

P0.7	1	Display-select Decoder CS DAC WR
P0.6	1	Keypad Column 2
P0.5	1	Keypad Column 1
P0.4	1	Keypad Column 0
P0.3	1	Keypad Row 3
P0.2	1	Keypad Row 2
P0.1	1	Keypad Row 1
P0.0	1	Keypad Row 0
P1.7	1	LED 7 Seg. dp DAC DB7 LCD DB7
P1.6	1	LED 6 Seg. g DAC DB6 LCD DB6
P1.5	1	LED 5 Seg. f DAC DB5 LCD DB5
P1.4	1	LED 4 Seg. e DAC DB4 LCD DB4
P1.3	1	LED 3 ... d ..DB3 ..DB3 .. RS
P1.2	1	LED 2 ... c ..DB2 ..DB2 LCD E
P1.1	1	LED 1 Seg. b DAC DB1 LCD DB1
P1.0	1	LED 0 Seg. a DAC DB0 LCD DB0
P2.7	1	SW 7 ADC DB7
P2.6	1	SW 6 ADC DB6
P2.5	1	SW 5 ADC DB5
P2.4	1	SW 4 ADC DB4
P2.3	1	SW 3 ADC DB3
P2.2	1	SW 2 ADC DB2
P2.1	1	SW 1 ADC DB1
P2.0	1	SW 0 ADC DB0
P3.7	1	ADC RD Comparator Output
P3.6	1	ADC WR
P3.5	1	Motor Sensor
P3.4	1	Display-select Input 1
P3.3	1	AND Gate Output Display-se..t 0
P3.2	1	ADC INTR
P3.1	1	Motor Control Bit 1 Ext. UART Rx
P3.0	1	Motor Control Bit 0 Ext. UART Tx

DI / LD

AND Gate Disabl...

Key Bounce Disabl...

Standard

U Odd Parity 8-bit UART @ 4800 Baud

Rx Rx Reset

Tx Tx Send

0.0 V output

Scope

DAC

BF 0 AC 0x00 IR 0x00 DR 0x00

8888

0.0 V input

1111111

ADC

MAX

MIN

Motor Enabled

#### **PROGRAM – 4: 8-BIT DIVISION:**

##### **ALGORITHM:**

1. Begin.
2. Initialize R0 with 00h.
3. Move the value in R1 to A.
4. Move the value in R2 to B.
5. Divide A by B.
6. Move A to R4 (quotient) and B to R5 (remainder)
7. End.

PROGRAM	COMMENTS
MOV R0, #00	R0 has address of 0x00
MOV A, R1	0x01 has 1st 8-bit number
MOV B, R2	0x02 has 2nd 8-bit number
DIV AB	BA = A / B, A: Quotient, B: Remainder
MOV R5, A	Move quotient to R4 from A
MOV R4, B	Move remainder to R5 from B
HALT:	
SJMP HALT	Halt the program with a loop.



## SAMPLE I/O SNAPSHOT:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0x01
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x04
RXD	TXD	TMOD	0x00	R5	0x01	PSW	0x01
1	1	TCON	0x00	R4	0x04	IP	0x00
SCON	0x00	PCON	0x00	R3	0x00	IE	0x00
pins	bits	TH1	TL1	R2	0x02	DPH	0x00
0xFF	0xFF	P3	0x00	R1	0x09	DPL	0x00
0xFF	0xFF	P2	0x00	R0	0x00	SP	0x07
0xFF	0xFF	P1	0x00	PC	0x0009	PSW	0 0 0 0 0 0 0 1
0xFF	0xFF	P0	0x00				

8051

Modify RAM

Data Memory

addr	0x02	0x02	value
0	00	09	02
1	00	04	01
2	00	00	00
3	00	00	00
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00

Copyright ©2005-2016 James Rogers

Remove All Breakpo...

RST Step Run New Load Save Copy Paste

Time: 23us - Instructions: 12

```

;8051 ALP TO DIVIDE 2 8-BIT NU
0000| MOV     R0, #00      ;R0 has ac
0002| MOV     A, R1        ;0x01 has
0003| MOV     B, R2        ;0x02 has
0005| DIV     AB           ;AB = A/B,
0006| MOV     R4, A        ;Move quot
0007| MOV     R5, B        ;Move rema
HALT:
0009| SJMP    HALT
  
```

P0.7 1 Display-select Decoder CS|DAC WR  
P0.6 1 Keypad Column 2  
P0.5 1 Keypad Column 1  
P0.4 1 Keypad Column 0  
P0.3 1 Keypad Row 3  
P0.2 1 Keypad Row 2  
P0.1 1 Keypad Row 1  
P0.0 1 Keypad Row 0  
P1.7 1 LED 7|Seg. dp|DAC DB7|LCD DB7  
P1.6 1 LED 6|Seg. g|DAC DB6|LCD DB6  
P1.5 1 LED 5|Seg. f|DAC DB5|LCD DB5  
P1.4 1 LED 4|Seg. e|DAC DB4|LCD DB4  
P1.3 1 LED 3|... d|..DB3|..DB3|.. RS  
P1.2 1 LED 2|... c|..DB2|..DB2|LCD E  
P1.1 1 LED 1|Seg. b|DAC DB1|LCD DB1  
P1.0 1 LED 0|Seg. a|DAC DB0|LCD DB0  
P2.7 1 SW 7|ADC DB7  
P2.6 1 SW 6|ADC DB6  
P2.5 1 SW 5|ADC DB5  
P2.4 1 SW 4|ADC DB4  
P2.3 1 SW 3|ADC DB3  
P2.2 1 SW 2|ADC DB2  
P2.1 1 SW 1|ADC DB1  
P2.0 1 SW 0|ADC DB0  
P3.7 1 ADC RD|Comparator Output  
P3.6 1 ADC WR  
P3.5 1 Motor Sensor  
P3.4 1 Display-select Input 1  
P3.3 1 AND Gate Output|Display-se...t 0  
P3.2 1 ADC INTR  
P3.1 1 Motor Control Bit 1|Ext. UART Rx  
P3.0 1 Motor Control Bit 0|Ext. UART Tx

DI LD

1 2 3 AND Gate Disabl...  
4 5 6 Key Bounce Disabl...  
7 8 9 Standard  
\* 0 #

U Odd Parity 8-bit UART @ 4800 Baud  
Rx Rx Reset  
Tx Tx Send

0.0 V output  
Scope  
DAC

BF 0 AC 0x00 IR 0x00 DR 0x00

0.0 V input  
1111111  
ADC

MAX  
MIN  
Motor Enabled

8888

**RESULT:**

The assembly level programs were written to perform the above specified 8-bit arithmetic operations using an 8051 microcontroller and the outputs were verified.