

Name : _____

Subject : _____

Year : _____ Semester : _____

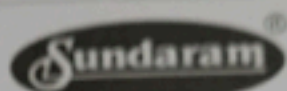
Class : _____ Roll No. : _____

Academic Year : _____

Professor-in-Charge : _____

Department : _____ Head of the Department : _____

University : _____



Books for Success...

CERTIFICATE

Class : _____

Year : _____

This is to certify that the work entered in this journal is the
Kumar / Kumari _____

who has worked for the 1st / 2nd / both Term(s) of
_____ in the college laboratory.

Professor-in-charge

Head of the

Date :

EMP Exp-2

Aim: To perform arithmetic operations on 8bit and 16 bit data

Under arithmetic operation, 8086 provides an addition, subtraction, multiplication and division. After all operations are performed on the operand (data).

Arithmetic Group

- i) Addition
- ii) Subtraction
- iii) Multiplication
- iv) Division

1) Addition:

ADD

ADC

INC

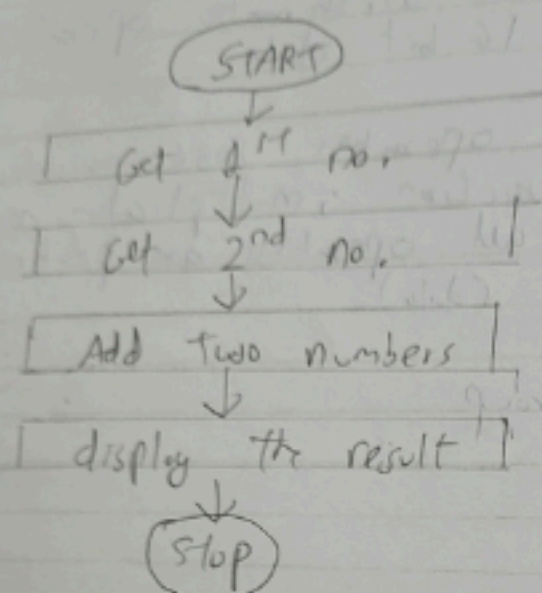
DAA

AAA

ADD: Add byte or word
adds a number from source to destination.

Mnemonic: ADD Destination, source
ADD operand 1, operand 2

Flowchart



atpt:

04

WAP:

Algorithm

1) Initialize

Get

get

Add

Displ

stop

Program:

.model

.data

a db

b db

.code

mov

mov

mov

mov

mov

WAP: addition of two 8bit numbers

Algorithm:

- 1) Initialize the data segment
- 2) Get the first number in AL register
- 3) Get the second number in BL register
- 4) Add the two numbers
- 5) Display result
- 6) Stop

Program:

.model small

.data

a db 02H

b db 02H

.code

mov ax, @data ; initialize data ~~segment~~ ^{section}

mov ds, ax

mov al, a ; load no. 1 in al

mov bl, b ; load no. 2 in bl

mov al, b1 ; add numbers and result in al

mov cx, 02H ; count of digits to be displayed
mov cx, 04H ; count to roll by 4553
mov bx, al ; result in reg bx

j2: rol bx, cl ; roll bx, so msb \rightarrow to lsb
mov dl, bh ; load dl with data to be displayed
and dl, 0FH ; get only lsb
cmp dl, 09 ; check if digit 0-9 or A-F

jbe 14
add dl, 07 ; if digit 0-9 or letter A-F with
32h, then only add 30H

j4: add dl, 30H

mov ah, 02h
int 21H
dec cx

Jnz J2
mov ah, 0CH
int 21H

end

2) Subtraction

SUB

SBB

DEC

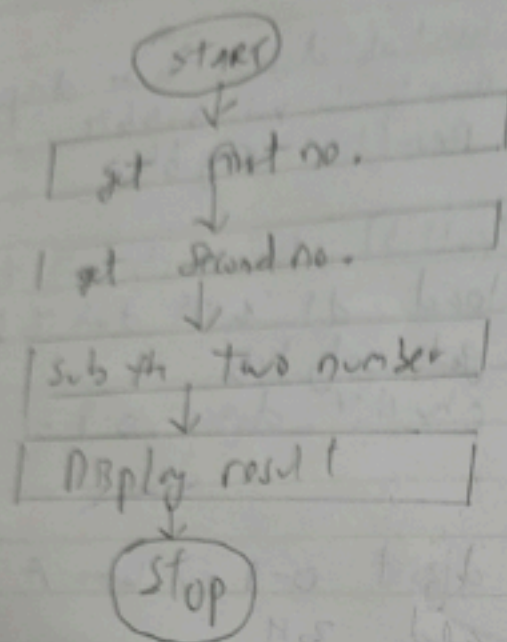
NEG

CMP

DAS

AAS

Flowchart



M50, d0
 M100, 12
 10, dd

10, dd 100
 M70, 16
 P0, 16

M0E, 16 160

M50, d0
 M15, 100
 d0, 100

M10, d0
 M15, 100

100

redacted

100

100

Al

In

get

g

do

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

pro

Algorithm

- 1) Initialize the data segment
- 2) get the first number in AX register
- 3) get the second number in BX register
- 4) Add two numbers
- 5) display result.

Program:

.model small

.data

a db 02h

b db 02h

.code

mov ax, @data

mov dx, ax

mov al, a

mov bl, b

sub al, bl

mov ch, 02h

mov cl, 04h

mov bh, al

atp t: 0

metabolism

longer than the other
the first number is the
the second number is the
the third number is the
the fourth number is the
the fifth number is the
the sixth number is the
the seventh number is the
the eighth number is the
the ninth number is the
the tenth number is the
the eleventh number is the
the twelfth number is the
the thirteenth number is the
the fourteenth number is the
the fifteenth number is the
the sixteenth number is the
the seventeenth number is the
the eighteenth number is the
the nineteenth number is the
the twentieth number is the
the twenty-first number is the
the twenty-second number is the
the twenty-third number is the
the twenty-fourth number is the
the twenty-fifth number is the
the twenty-sixth number is the
the twenty-seventh number is the
the twenty-eighth number is the
the twenty-ninth number is the
the thirtieth number is the
the thirty-first number is the
the thirty-second number is the
the thirty-third number is the
the thirty-fourth number is the
the thirty-fifth number is the
the thirty-sixth number is the
the thirty-seventh number is the
the thirty-eighth number is the
the thirty-ninth number is the
the fortieth number is the
the forty-first number is the
the forty-second number is the
the forty-third number is the
the forty-fourth number is the
the forty-fifth number is the
the forty-sixth number is the
the forty-seventh number is the
the forty-eighth number is the
the forty-ninth number is the
the fiftieth number is the
the fifty-first number is the
the fifty-second number is the
the fifty-third number is the
the fifty-fourth number is the
the fifty-fifth number is the
the fifty-sixth number is the
the fifty-seventh number is the
the fifty-eighth number is the
the fifty-ninth number is the
the sixtieth number is the
the sixty-first number is the
the sixty-second number is the
the sixty-third number is the
the sixty-fourth number is the
the sixty-fifth number is the
the sixty-sixth number is the
the sixty-seventh number is the
the sixty-eighth number is the
the sixty-ninth number is the
the seventieth number is the
the seventy-first number is the
the seventy-second number is the
the seventy-third number is the
the seventy-fourth number is the
the seventy-fifth number is the
the seventy-sixth number is the
the seventy-seventh number is the
the seventy-eighth number is the
the seventy-ninth number is the
the eightieth number is the
the eighty-first number is the
the eighty-second number is the
the eighty-third number is the
the eighty-fourth number is the
the eighty-fifth number is the
the eighty-sixth number is the
the eighty-seventh number is the
the eighty-eighth number is the
the eighty-ninth number is the
the ninetieth number is the
the ninety-first number is the
the ninety-second number is the
the ninety-third number is the
the ninety-fourth number is the
the ninety-fifth number is the
the ninety-sixth number is the
the ninety-seventh number is the
the ninety-eighth number is the
the ninety-ninth number is the
the hundredth number is the

metabolism

metabolism

metabolism

metabolism

metabolism

Algo

Trin


```
i2: rol bh,cl  
mov dl,bh  
and dl,0fh  
cmp dl,04  
jbe i4  
add dl,07
```

```
i5: add dl,30h  
mov ah,02h  
int 21h
```

```
dec ch  
jnz i2
```

```
mov ah,5ch  
int 21h
```

end

WAP for addition of two 16-bit numbers.

Algorithm:

- 1) Initialize the data segment
- 2) Get 1st no. in AX register
- 3) Get 2nd no. in BX register
- 4) Add two numbers
- 5) Display result.

output: 3

add 10

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

add 16

longer than

longer than

comp dl

the it

add 16

.model small

.data

a dw 0001h

b dw 0002h

.code

mov ax, @data

mov ds, ax

mov ax, a

mov bx, b

add ax, bx

mov cx, 04h

mov cl, 04h

mov bx, ax

i2: rol bx, cl

mov ds, bx

and di, 0fh

cmp di, 09

jbe i4

add di, 07

```
in: add dl, 30h
mov ah, 02h
int 21h
dec ch
```

```
int 12
int 12
```

```
mov ah, 1ch
int 21h
```

end

WAP to subtract two 16-bit numbers.

Algorithm

- 1) initialize the data segment
- 2) get 1st no. in AX register
- 3) get 2nd no. in BX register
- 4) subtract two numbers.
- 5) display result
- 6) stop

program :-

- model small

.data

~~model~~ a dw 0001h
b dw 0002h

-code

mov ax, @data
mov ds, ax

mov ax, a
mov bx, b
add /x sub ax, bx

mov ch, 04h
mov cl, 04h
mov bx, ax
i2: rol bx, cl

mov dx, bx
and dl, 0Fh

cmp dl, 09
jbe i4
add dl, 07

i4: add dl, 30h
mov ah, 02h
int 21h
dec ch

in 2 12

mov ah, 5ch
int 21h

end

3) Multiplication:

MUL
IMUL
AAM

This instruction multiplies an unsigned byte from source with an byte in the AL register or an unsigned word from source with an unsigned word in AX.

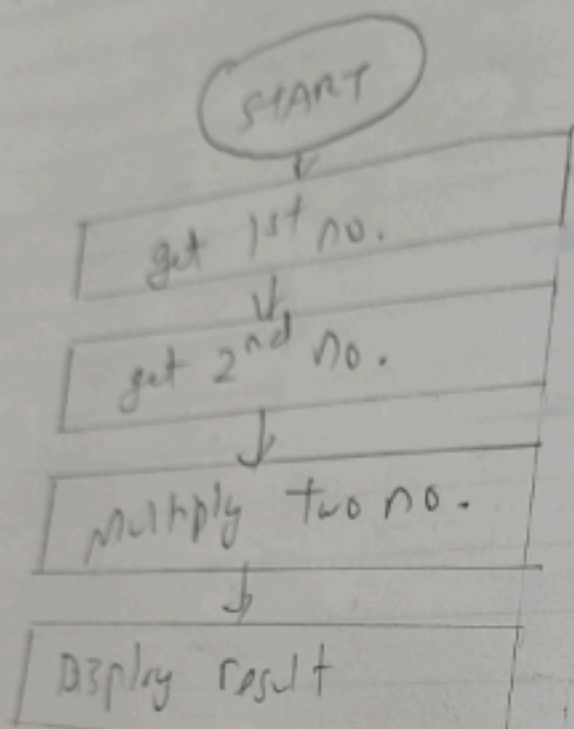
When a byte is multiplied by contents of AL, the result is stored in AX.

The MSB of result is stored in AH register and the LSB of result is stored in AL register.

Memory: MUL Multiplier

WAP: multiplication of two 8 bit numbers.

Flowchart



stop

P Algorithm :-

- 1) Initialize the data segment
- 2) get 1st no. in AX register
- 3) get 2nd no. in BX register
- 4) Multiply two number
- 5) stop display result

model small

data

a db 05h

b db 02h

code

mov ax, @data
mov ds, ax

mov al, a

mov bl, b

mul bl

mov ah, 02h

mov ch, 02h

mov cl, 03h

mov bh, al

int 21h

mov dl, bh

and dl, 0fh

cmp dl, 09

jbe ih

add dl, 07

ih : add dl, 30h

mov ah, 02h

int 21h

dec ch

jnz ??

mov ah, 4ch

int 21h

end

9) Division:-

DIV

FDIV

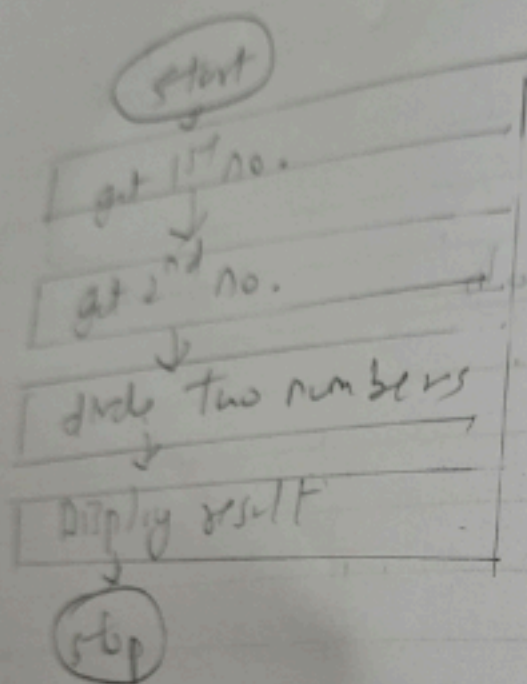
AAD

CBW

CWD

This instruction divides an unsigned byte from the source with an byte in the AL register or an unsigned word from source with an unsigned word in AX.

Flowchart



add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

add lb

M

W

A

1) Zr

2) g

3) g

1) C

7) d

7) d

7) end

prog

mode

data

a dw

b dw

code

mov

m

Memory: Div ~~mult~~ div multiplier.

WAP: division of two 8-bit numbers.

Algorithm:

- 1) Initialize data segment
- 2) get dividend in AX register
- 3) get divisor in BX register
- 4) Clear the AX register
- 5) divide the number
- 6) display result
- 7) end

Program

.model small

.data

a dw 08h

b dw 04h

.code

mov ax, @data

mov ds, ax

mov ax, a

mov bx, b

div bx


```
mov ch, 04h  
mov cl, 04h  
mov bx, ax
```

```
jz: rol bx, cl  
mov dx, bx  
and dl, 0fh  
cmp dl, 09
```

```
jbe ih  
add dl, 07
```

```
ih: add ax dl, 30h  
mov ah, 02h  
int 21h
```

```
dec ch  
jnz ih  
mov dh, 4ch  
int 21h
```

end

Conclusion:- Hence we successfully performed arithmetic operations on 8 bit and 16 bit data.