

**FR. Conceicao Rodrigues College of Engineering**  
**Department of Computer Engineering**

## 1. Addition of Two 8/16/32 bit numbers

### 1. Course, Subject & Experiment Details

Academic Year	2023-24	Estimated Time	Experiment No. 1– 02 Hours
Course & Semester	S.E. (Comps) – Sem. IV	Subject Name	Microprocessor
Chapter No.	2	Chapter Title	Instruction Set and Programming
Experiment Type	Software	Subject Code	CSC405

### Rubrics

Timeline (2)	Practical Skill & Applied Knowledge (2)	Output (3)	Postlab (3)	Total (10)	Sign

### 2. Aim & Objective of Experiment

**TO ADD TWO 8/16/32 BIT NUMBERS**

**Objective :** Program involves storing the two 8-bit no in memory locations and adding them taking into consideration the carry generated. The objective of this program is to give an overview of arithmetic instructions of 8086 for 8-bit operands

### 3. Software Required

TASM Assembler

**Prepared by : Prof. Heenakausar Pendhari**

## 4 . Brief Theoretical Description

**Pre-Requisites:**

1. Instructions of microprocessor 8086
2. Addressing mode of microprocessor 8086.
3. Knowledge of TASM directories.

**Theory:** The addressing modes used in program are:

- 1) Direct addressing mode: in this mode address of operand is directly specified in the instruction. This address is offset address of the segment being indicated by an instruction.

E.g. MOV AL,[2000h]

$$EA = DS \times 10H + 2000H$$

- 2) Register Addressing Mode: In this mode operand are specified using registers. Instructions are shorter but operations cannot be identified looking at instruction.

E.g. MOV CL, DL

- 3) Based Indexed Addressing Mode: The operand address is calculated using base register and index register.

E.g. MOV DX, [BX + SI] moves word from address pointed by BX + SI in data segment to DX.

$$EA = DS \times 10H + BX + SI$$

- 4) Base indexed plus displacement: In this mode address of operand is calculated using base register , index register and displacement.

E.g. MOV CX, [BX+DI+10h]

This moves a word from address pointed by BX + DI +10h of segment to CX.

- 5. Algorithm:**
1. Initialize the data segment.
  2. Store two 8/16 -bit numbers in memory locations.
  3. Move the 1<sup>st</sup> number in any one of the general purpose register.
  4. Move the 2<sup>nd</sup> number in any other general purpose register.
  5. Add the 2 numbers.
  6. Store the result in memory location.
  7. Check for carry flag. If carry flag is set then store '1' as MSB of result.
  8. Stop

## 6. Conclusion:

### 1. 8 bit addition without carry:

```
.8086
.model small
.stack 100h

.data
a db 09h
b db 02h

.code
start:
mov ax,@data
mov ds,ax
mov al,a
mov bl,b
add al,b
mov cl,al
mov ah,4ch
int 21h
end start
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

CPU 80486

CS	IP	Instruction	AX	BX	CX	DX	SI	DI	BP	SP
0000	8B4F	mov ax,44AF	44AF	0002	0000	0000	0000	0000	0000	0100
0003	8ED8	mov ds,ax	44AF	0002	0000	0000	0000	0000	0000	0100
0005	A00000	mov al,[0000]	0000	0002	0000	0000	0000	0000	0000	0100
0008	B1E0100	mov bl,[0001]	0000	0002	0000	0000	0000	0000	0000	0100
000C	02C3	add al,bl	02C3	0002	0000	0000	0000	0000	0000	0100
000E	9ACB	mov cl,al	02C3	0002	0000	0000	0000	0000	0000	0100
0010	B44C	mov ah,4C	02C3	0002	0000	0000	0000	0000	0000	0100
0012	CD21	int 21	02C3	0002	0000	0000	0000	0000	0000	0100

es:0000 CD 20 FF 9F 00 EA FF FF = f 9  
es:0000 AD DE E5 01 00 15 AF 01 i 9 8 S 8  
es:0010 00 15 7D 02 1C 0F 92 01 S 10 4 8  
es:0018 01 01 01 00 02 FF FF FF 8 8 8

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

### 2. 8 bit addition with carry:

```
.8086
.model small
.stack 100h

.data
a db 99h
b db 88h
sum db ?
carry db ?

.code
start:
mov ax,@data
mov ds,ax
mov al,a
mov bl,b
add al,b
jnc skip
inc carry
skip:
mov sum,al
int 03h
end start
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

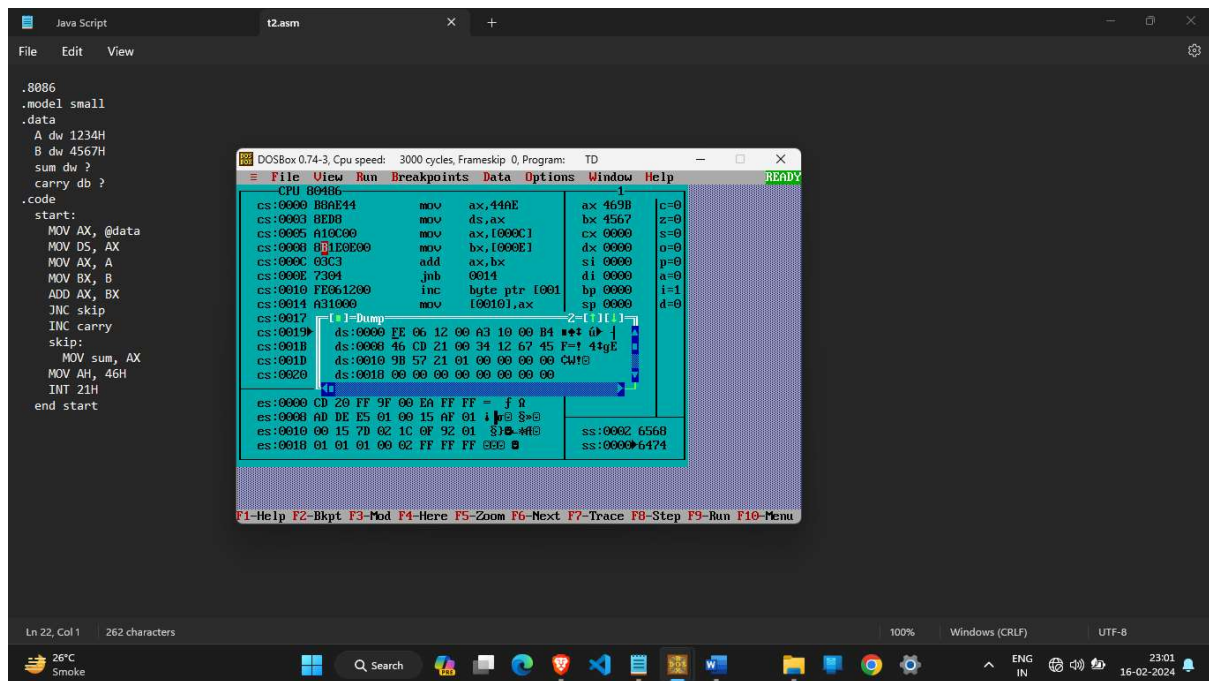
CPU 80486

CS	IP	Instruction	AX	BX	CX	DX	SI	DI	BP	SP
0000	8B4F	mov ax,44AF	44AF	0002	0000	0000	0000	0000	0000	0100
0003	8ED8	mov ds,ax	44AF	0002	0000	0000	0000	0000	0000	0100
0005	A00000	mov al,[0000]	0000	0002	0000	0000	0000	0000	0000	0100
0008	B1E0100	mov bl,[0001]	0000	0002	0000	0000	0000	0000	0000	0100
000C	02C3	add al,bl	02C3	0002	0000	0000	0000	0000	0000	0100
000E	7304	jnb 0014	02C3	0002	0000	0000	0000	0000	0000	0100
0010	FD060300	inc byte ptr [0000]	02C3	0002	0000	0000	0000	0000	0000	0100
0014	628200	mov [0002],al	02C3	0002	0000	0000	0000	0000	0000	0100

es:0000 CD 20 FF 9F 00 EA FF FF = f 9  
es:0000 AD DE E5 01 00 15 AF 01 i 9 8 S 8  
es:0010 00 15 7D 02 1C 0F 92 01 S 10 4 8  
es:0018 01 01 01 00 02 FF FF FF 8 8 8

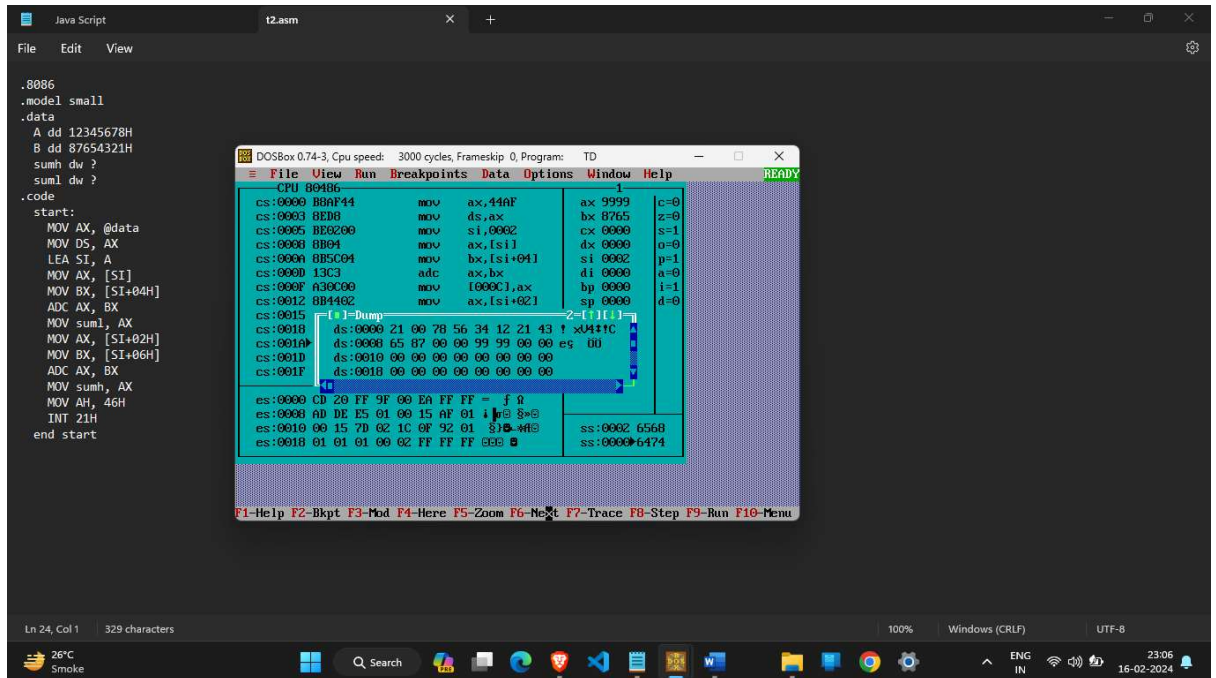
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

### 3. 16 bit addition with carry:



### Postlab:

1. Write a program for addition of two 32 bit numbers ,execute and take the screen shots of the results.



2. Write a program to Subtract two 16 bit numbers.

