ROLL NO :71 MPL LAB

EXPERIMENT 07

AIM :-

Program to count number of 1's and 0;s in a given 8 bit number

LO: (LO4): Program to move set of numbers from one memory block to another.

SOFTWARE:- Tasm Software

Theory:-

Instructions used in this program

MOV

The MOV instruction is the most important command in the 8086 because it moves data from one location to another.

Syntax: Mov source, destination

Example: Mov Ax,1234H

SHR

The SHR instruction is an abbreviation for 'Shift Right'. This instruction simply shifts the mentioned bits in the register to the right side one by one by inserting the same number (bits that are being shifted) of zeroes from the left end. The rightmost bit that is being shifted is stored in the Carry Flag (CF).

Syntax: SHR Register, Bits to be shifted

Example: SHR AX, 2

LOOP

The loop instructions cause the microprocessor to execute a series of instructions repeatedly. Basically, the LOOP instructions are short jump instructions on a condition i.e., when the condition satisfies a short jump is taken whose destination or target address is in the range of -128 bytes to +127 bytes from the instruction address after LOOP instruction.

ROLL NO :71 MPL LAB

Syntax: LOOP label

INTERRUPT

int 21h means, call the interrupt handler 0x21 which is the DOS Function dispatcher. the "mov ah,01h" is setting AH with 0x01, which is the Keyboard Input with Echo handler in the interrupt.

Syntax: int 21H

Example: int 21H

JMP

The basic instruction that transfers control to another point in the program is JMP.

Syntax: JMP label

To declare a label in your program, just type its name and add ":" to the end, label can be any character combination but it cannot start with a number, for example here are 3 legal label definitions:

label1:

label2:

a:

JC

Stands for 'Jump if Carry'

It checks whether the carry flag is set or not. If yes, then jump takes place, that is: If CF = 1, then jump.

INC

Increment Register or memory by 1.

Adds 1 to the destination operand, while preserving the state of the CF flag. The destination operand can be a register or a memory location. This instruction allows a loop counter to be updated without disturbing the CF flag.

ROLL NO :71 MPL LAB

(Use a ADD instruction with an immediate operand of 1 to perform an increment operation that does updates the CF flag.)

This instruction can be used with a LOCK prefix to allow the instruction to be executed atomically.

Example: MOV AL, 4 INC AL; AL = 5

Code :-

Assume CS: Code, DS: Data

Data Segment

no db 0FFH

c0 db 01 dup(?)

c1 db 01 dup(?)

Data Ends

Code Segment

START: MOV AX, Data

MOV DS, AX

MOV AH, no

TO: SHR AH, 1

JC DOWN

INC c0

JMP NEXT

DOWN: INC c1

JMP NEXT

NEXT: LOOP TO

NAME : Soham Manoj Pawar	Class : SE-IT
ROLL NO :71	MPL LAB
MOV AH, 4CH	

INT 21H Code Ends

End Start

Output :-

ROLL NO :71 MPL LAB

≡ File Edit View	Run Breakpoints Dat	a Options Window	ı Help READY
[-[1] = CPU 80486		1-[†][↓]	
cs:000F DOEC	shr ah,1	🔼 ax 4000 c=	1
cs:0011 7207	jb 001A	bx 0000 z=	<u>o</u>
cs:0013 FE060100	inc byte ptr [6		
cs:0017 EB08	jmp 0021	dx 0000 o=	
cs:0019 90	0.1	si 0000 p=	
cs:0013 56 cs:0014 FE060200	nop		
	inc byte ptr [6		
cs:001E EB01	jmp 0021	bp 0000 i=	
cs:0020 90	nop	sp 0000 d=	•
cs:0021 EZEC	loop 000F	ds 48AD	
cs:0023 B44C	mo∨ ah,4C	es 489D	
cs:0025▶CD21	int 21	ss 48AC	
cs:0027 4C	dec sp	cs 48AE	
cs:0028 CD21	int 21	ip 0025	
∢ ∎			
es:0000 CD 20 FF 9F	$\Theta\Theta$ EA FF FF = $f \Theta$		
	C5 15 AA 01		
es:0010 C5 15 99 02	20 10 92 01 §e B ►ff©	ss:0002 6474	
es:0018 01 03 01 00		ss:0000*0000	
68.0010 01 03 01 00	OZ FF FF FF GFG G	88.000070000	- 1
F1-Help F2-Bkpt F3-Mod	F4-Here F5-Zoom F6-Ne	ext <mark>F7-Trace F8-Ste</mark>	p F9-Run F10-Menu
≡ File Edit View	F4-Here F5-Zoom F6-Ne Run Breakpoints Dat	a Options Window	Help READY
≡ File Edit View	Run Breakpoints Dat	a Options Window 1=[†][↓]	Help READY
≡ File Edit Uiew -[■1-CPU 80486- cs:00000-BBAD48	Run Breakpoints Dat mo∨ a×,48AD	a Options Window 1-[1][4] ax 0000 c=	Help READY
= File Edit View -[1]=CPU 80486- cs:0000>B8AD48 cs:0003 8ED8	Run Breakpoints Dat mo∨ ax,48AD mo∨ ds,ax	a Options Window 1-[1][1] ax 0000 c= bx 0000 z=	Help RDADY
≡ File Edit View	Run Breakpoints Date mov ax,48AD mov ds,ax mov ax,0000	a Options Window 1-[1][1] ax 00000 c= bx 00000 c= cx 00000 s=	Help RDADY
= File Edit View	Run Breakpoints Date mov ax,48AD mov ds,ax mov ax,0000 mov ah,[0000]	a Options Window 1=[1][1] ax 00000 c= bx 00000 c= cx 00000 s= dx 00000 o=	Help READY
= File Edit View	Run Breakpoints Date mov ax,48AD mov ds,ax mov ax,0000 mov ah,[0000] mov cx,0008	a Options Window 1=[1][1] ax 00000 c= bx 00000 c= cx 00000 s= dx 00000 c= si 00000 p=	Help READY
= File Edit View	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1	a Options Window 1 - [] [] []	Help RDADY
= File Edit View	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A	a Options Window	Help READY 0
= File Edit View	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr	a Options Window	Help READY 0
= File Edit View	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr jmp 0021	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= sp 09000 d= dx 489D	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0005 B80000 Cs:0008 B4260000 Cs:0006 B90800 Cs:0007 D0EC Cs:0011 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr jmp 0021 nop	ax 9000 c= bx 9000 c= cx 9000 c= cx 9000 c= dx 9000 p= di 9000 a= bp 9000 i= sp 9000 d= dx 489D es 489D	Help READY 0
■ File Edit View Cs:0000 B8AD48 Cs:0003 BED8 Cs:0005 B80000 Cs:0008 BA260000 Cs:000C B90800 Cs:000F D0EC Cs:0011 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr jmp 0021 nop inc byte ptr	ax 99699 c= bx 99690 c= cx 99690 c= cx 99690 c= dx 99690 c= di 99690 c= bp 99690 c= bp 99690 d= dx 489D ex 489D ex 489D ex 489D	Help READY 0
■ File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0000 B80000 Cs:0000 B90800 Cs:0000 B90800 Cs:0000 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200 Cs:001E EB01	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr jmp 0021 nop inc byte ptr jmp 0021	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= bp 09000 d= dx 489D ex 489D ex 48AC cx 48AE	Help READY 0
■ File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0000 B80000 Cs:0000 B90800 Cs:0000 B90800 Cs:0000 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200 Cs:001E EB01 Cs:0020 90	mov ax,48AD mov ds,ax mov ax,0000 mov ah,100001 mov cx,0008 shr ah,1 jb 001A inc byte ptr jmp 0021 nop inc byte ptr	ax 99699 c= bx 99690 c= cx 99690 c= cx 99690 c= dx 99690 c= di 99690 c= bp 99690 c= bp 99690 d= dx 489D ex 489D ex 489D ex 489D	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0003 BED8 Cs:0005 B80000 Cs:0006 B90800 Cs:0006 B90800 Cs:0007 DOEC Cs:0011 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:0014 FE060200 Cs:0016 EB01 Cs:0020 90 Cs:00	Run Breakpoints Date mov ax,48AD mov ds,ax mov ax,0000 mov ah,[0000] mov cx,0008 shr ah,1 jb 001A inc byte ptr [6] jmp 0021 nop inc byte ptr [6] jmp 0021 nop	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= bp 09000 d= dx 489D ex 489D ex 48AC cx 48AE	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0003 BED8 Cs:0005 B80000 Cs:0006 B90800 Cs:0006 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:0014 FE060200 Cs:0016 EB01 Cs:0020 90 Cs:0020 90 Cs:0020 Cs:0020	Run Breakpoints Date mov	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= bp 09000 d= dx 489D ex 489D ex 48AC cx 48AE	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0003 BED8 Cs:0005 B80000 Cs:0006 B90800 Cs:0006 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:0014 FE060200 Cs:0016 EB01 Cs:0020 90 Cs:0020 90 Cs:0020 Cs:0020	Run Breakpoints Date mov	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= bp 09000 d= dx 489D ex 489D ex 48AC cx 48AE	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0005 B80000 Cs:0006 B4260000 Cs:0006 B90800 Cs:0007 D0EC Cs:0011 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:0014 FE060200 Cs:0016 EB01 Cs:0020 90 Cs:0020 90 Cs:0008 AD DE EO 01	Run Breakpoints Date mov	ax 9000 c= bx 9000 c= cx 9000 s= dx 9000 p= di 9000 a= bp 9000 i= sp 9000 d= dx 489D es 489D es 48AC cs 48AE ip 9000	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0005 B80000 Cs:0006 B4260000 Cs:0006 B90800 Cs:0007 D0EC Cs:0011 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:0014 FE060200 Cs:0016 EB01 Cs:0020 90 Cs:0020 90 Cs:0008 AD DE EO 01	Run Breakpoints Date mov	ax 09000 c= bx 09000 c= cx 09000 c= cx 09000 c= dx 09000 c= di 09000 c= bp 09000 i= bp 09000 d= dx 489D ex 489D ex 48AC cx 48AE	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0000 B80000 Cs:0000 B80000 Cs:0000 B90800 Cs:0000 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200 Cs:001E EB01 Cs:0020 90 Cs:0000 CD 20 FF 9F ds:0000 CD 20	Run Breakpoints Date mov	ax 90000 c= bx 90000 c= cx 90000 s= dx 90000 p= di 90000 a= bp 90000 d= ds 489D es 489D es 48AC cs 48AE ip 90000	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0000 B80000 Cs:0000 B80000 Cs:0000 B90800 Cs:0000 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200 Cs:001E EB01 Cs:0020 90 Cs:0000 CD 20 FF 9F ds:0000 CD 20	Run Breakpoints Date mov	ax 90000 c= bx 90000 c= cx 90000 s= dx 90000 p= di 90000 a= bp 90000 d= ds 489D es 489D es 48AC cs 48AE ip 90000	Help READY 0
File Edit View Cs:0000 B8AD48 Cs:0000 B8AD48 Cs:0000 B80000 Cs:0000 B80000 Cs:0000 B90800 Cs:0000 B90800 Cs:0001 7207 Cs:0013 FE060100 Cs:0017 EB08 Cs:0019 90 Cs:001A FE060200 Cs:001E EB01 Cs:0020 90 Cs:0000 CD 20 FF 9F ds:0000 CD 20	Run Breakpoints Date mov	ax 90000 c= bx 90000 c= cx 90000 s= dx 90000 p= di 90000 a= bp 90000 d= ds 489D es 489D es 48AC cs 48AE ip 90000	Help READY 0
■ File Edit View C: 1-CPU 80486 CS: 00000 B8AD48 CS: 00003 8ED8 CS: 00005 B800000 CS: 00006 B908000 CS: 00006 B908000 CS: 00011 7207 CS: 00113 FE0601000 CS: 0017 EB08 CS: 0019 90 CS: 001A FE0602000 CS: 001E EB01 CS: 0020 90 CS: 0008 AD DE E0 01 dS: 0000 C5 15 89 02	Mov	ax 90000 c= bx 90000 c= cx 90000 s= dx 90000 p= di 90000 a= bp 90000 d= ds 489D es 489D es 489D es 48AC cs 48AE ip 90000 ss:90000 6474 ss:90000 90000	Help READY

Conclusion : We learned to built a program on microprocessor using arithmetic and logical instructions and performed BCD addition on 16-bit values.