NAME : Soham Manoj Pawar **Class : SE-IT**

ROLL NO :71 **MPL LAB**

EXPERIMENT 07

AIM :-

Program to count number of1'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.

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.

NAME : Soham Manoj Pawar **Class : SE-IT**

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 :-

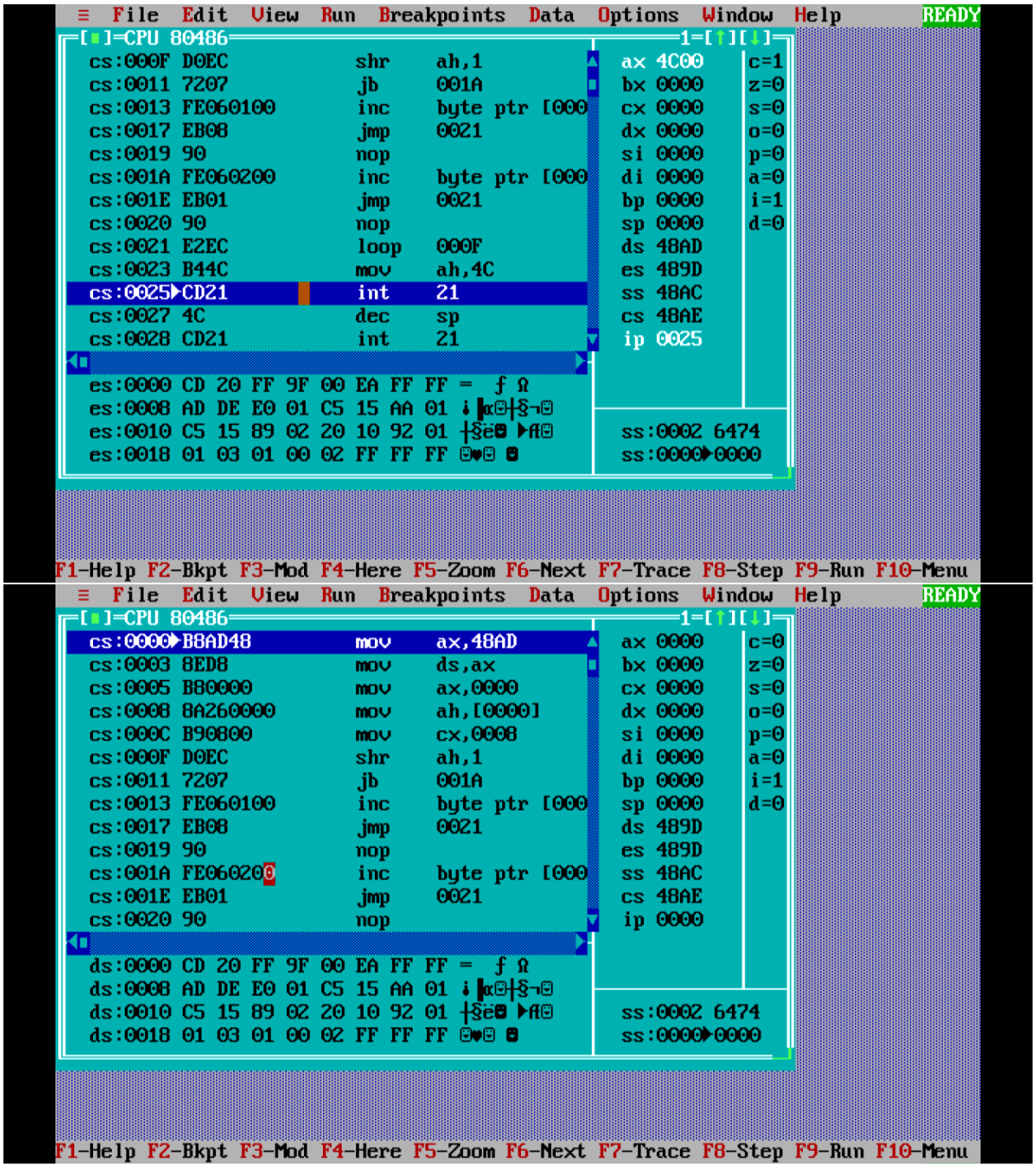
```
Assume CS: Code, DS: Data

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
```

```
MOV AH, 4CH

INT 21H

Code Ends

End Start
```

Output :-

NAME : Soham Manoj Pawar **Class : SE-IT**

ROLL NO :71 **MPL LAB**



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