Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 30/3/22

EXPERIMENT 8

Aim: Program to count number of1's and 0;s in a given 8 bit number

LO: 4

LO STATEMENT: Develop the assembly level programming using 8086 loop instruction set

Software and Hardware Requirements: TASM Software

Theory:

1. MOV Instruction

The MOV instruction is the most important command in the 8086 because it moves data from one location to another. It also has the widest variety of parameters; so the assembler programmer can use MOV effectively, the rest of the commands are easier to understand. MOV copies the data in the source to the destination. The data can be either a byte or a word. Sometimes this has to be explicitly stated when the assembler cannot determine from the operands whether a byte or word is being referenced.

Syntax:
Move Destination, Source

Example:
MOV Ax, Bx

2. SHR Instruction

SHR shifts the bits within the destination operand to the right, where right is toward the least-significant bit (LSB). The number of bit positions shifted may be specified either as an 8-bit immediate value, or by the value in CL—*not* CX or ECX. (The 8086 and 8088 are limited to the immediate value 1.) Note that while CL may accept a value up to 255, it is meaningless to shift by any value larger than 16—or 32 in 32-bit mode—*even though the shifts are actually performed on the 8086 and 8088*. (The 286 and later limit the number of shift operations performed to the native word size except when running in Virtual 86 mode.) The rightmost bit of the operand is shifted into the Carry flag; the leftmost bit is cleared to 0. The Auxiliary carry flag (AF) becomes undefined after this instruction. OF is modified *only* by the shift-by-one forms of SHL; after shift-by-CL forms, OF becomes undefined.

Syntax:

SHR Register, Bits to be shifted

Example:

SHR AX, 2

Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 30/3/22

3. INT instruction:

Interrupt is the method of creating a temporary halt during program execution and allows peripheral devices to access the microprocessor. The microprocessor responds to that interrupt with an ISR (Interrupt Service Routine), which is a short program to instruct the microprocessor on how to handle the interrupt.

Example:

INT 21H

4. JC Instruction:

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.

Example:

JC me

5. JMP Instruction:

Conditional execution often involves a transfer of control to the address of an instruction that does not follow the currently executing instruction. Transfer of control may be forward, to execute a new set of instructions or backward, to re-execute the same steps. The JMP instruction provides a label name where the flow of control is transferred immediately.

Syntax:

JMP label

Example:

JMP next

6. INC Instruction:

The INC instruction adds one to the destination operand,
while preserving the state of the carry flag CF. 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.

Syntax: INC destination

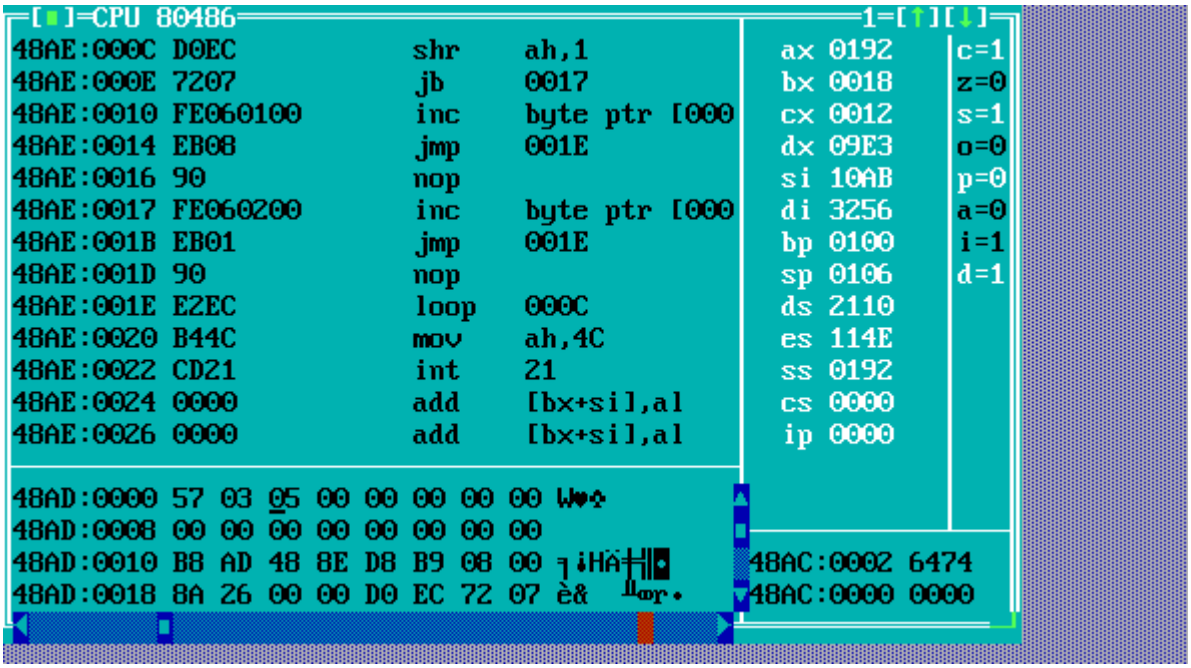
Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 30/3/22

Code:

```
assume ds:data,cs:code

data segment

 no db 57H

 c0 db 01 dup(?)

 c1 db 01 dup(?)

data ends

code segment

start:mov Ax,data

      mov Ds,Ax

      mov cx,08H

      mov Ah,no

up:   SHR Ah,1

      JC down

      INC c0

      JMP next

down: INC c1

      JMP next

next: LOOP up

      mov AH,4CH

      INT 21H

code ends

end start
```

Output:

Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 30/3/22



Conclusion:

From this experiment we have learned how to use different types of commands and to count the number of 0 and 1 in a binary number.