VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Implementing Affine Cipher IN 8086 Assembly language

02.06.2020

Divyanshu Kantol -18BCE1102

Shubham Bajaj- 18BCE1114

Saurabh Mohata -18BCE1207

Jay Daftari -18BCE1299

Professor name - Prathiba A. (Slot- A2)
Course Code - CSE2006

Overview

The affine is a type of monoalphabetic substitution cipher, where each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter The formula used means that each letter encrypts to one other letter, and back again, meaning the cipher is essentially a standard substitution cipher with a rule governing which letter goes to which.

Goal

To Encrypt The Given String Using Affine Cipher

Properties

The whole process relies on working modulo m (the length of the alphabet used). In the affine cipher, the letters of an alphabet of size m are first mapped to the integers in the range 0 … m-1.

The ‘key’ for the Affine cipher consists of 2 numbers, we’ll call them a and b. The following discussion assumes the use of a 26 character alphabet (m = 26). a should be chosen to be relatively prime to m (i.e. a should have no factors in common with m).

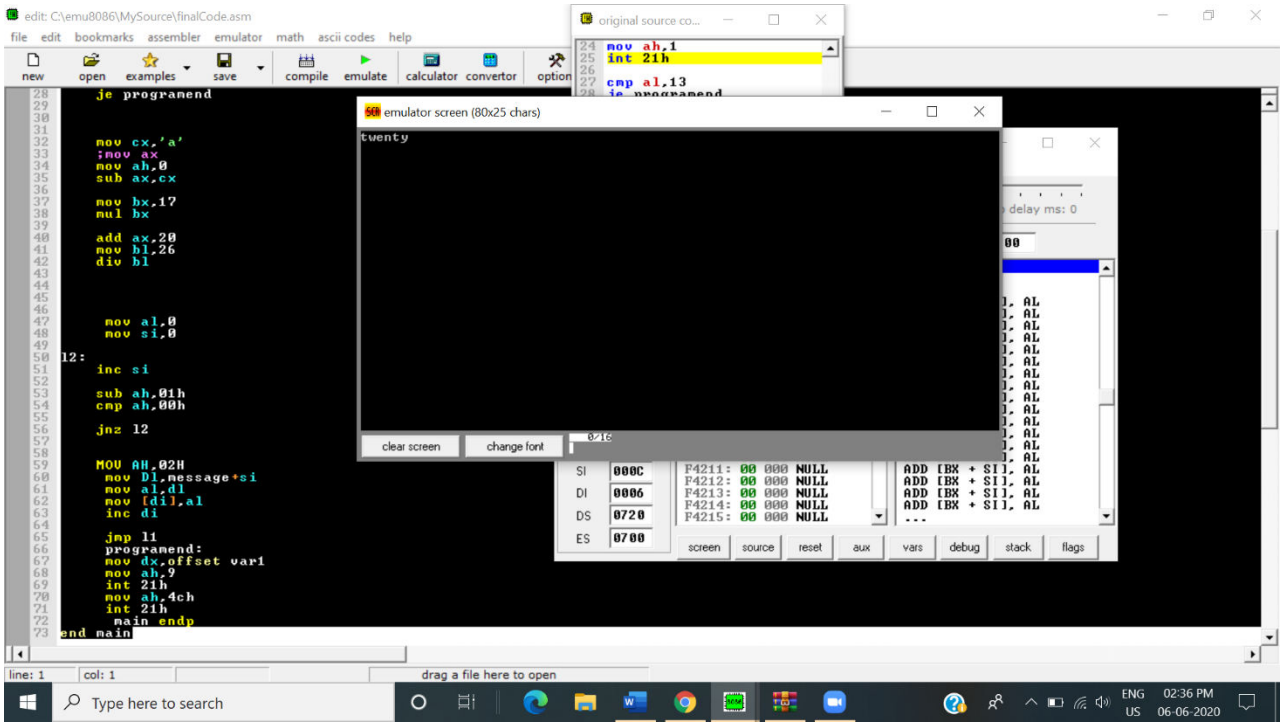
8086 Assembly Language Code (Software Used - EMU 8086)-

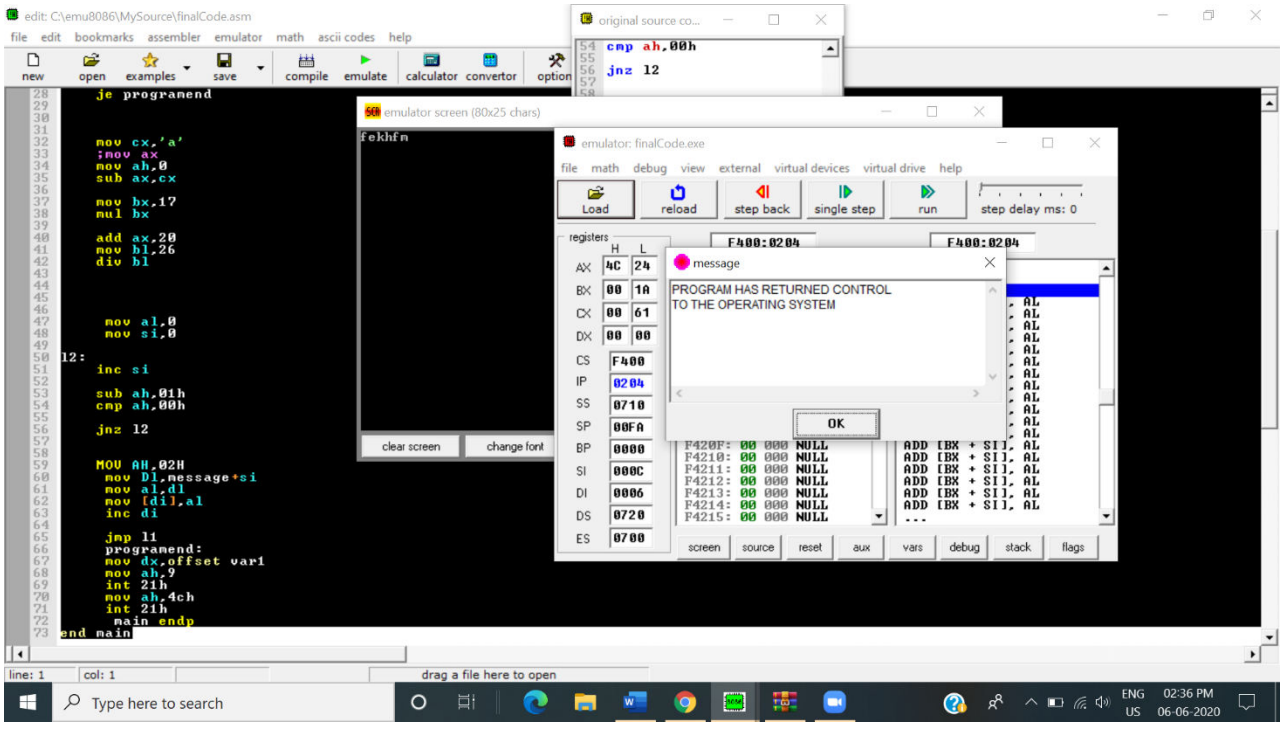
Code:

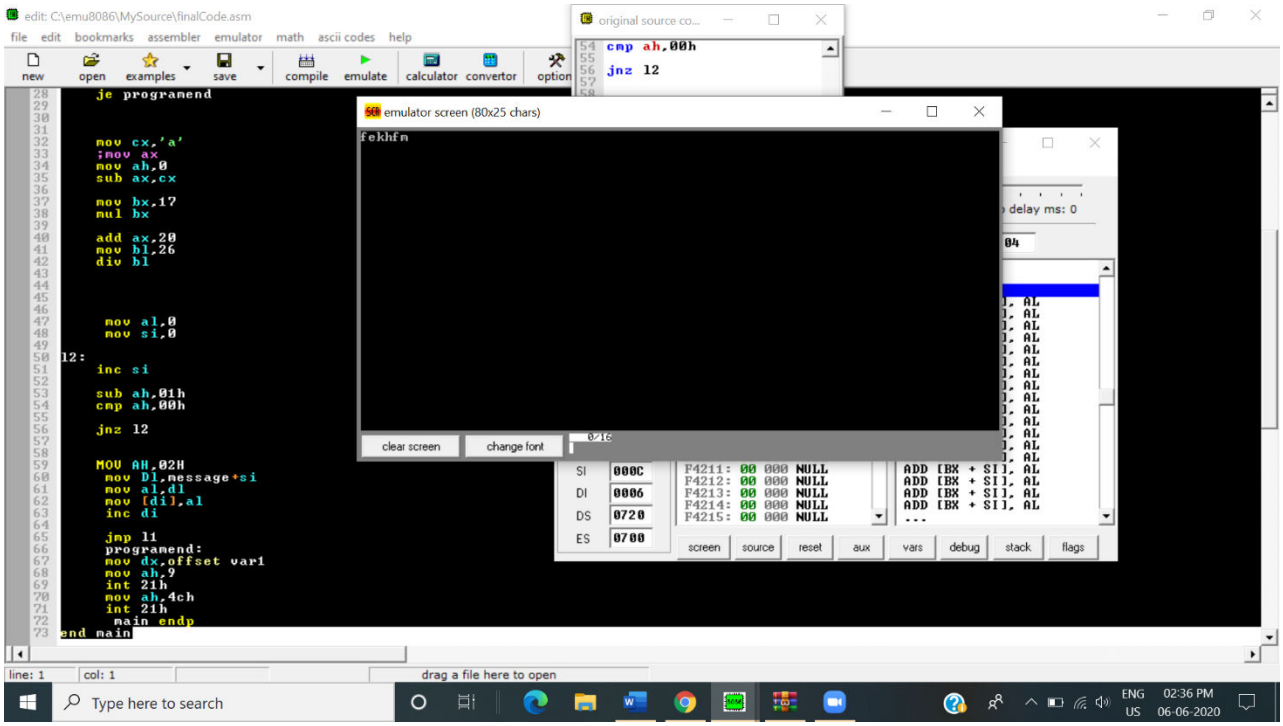
```
.model small
.stack 100h
.data
var1 db 100 dup('$')

    message db "abcdefghijklmnopqrstuvwxyz", 0dh, 0ah
    string db '$'


.code



main proc

    mov ax,@data
    mov ds,ax



    mov di,offset var1
    l1:
    mov ah,1
    int 21h

    cmp al,13
    je programend



    mov cx,'a'
    ;mov ax
    mov ah,0
    sub ax,cx

    mov bx,17
    mul bx

    add ax,20
    mov bl,26
```

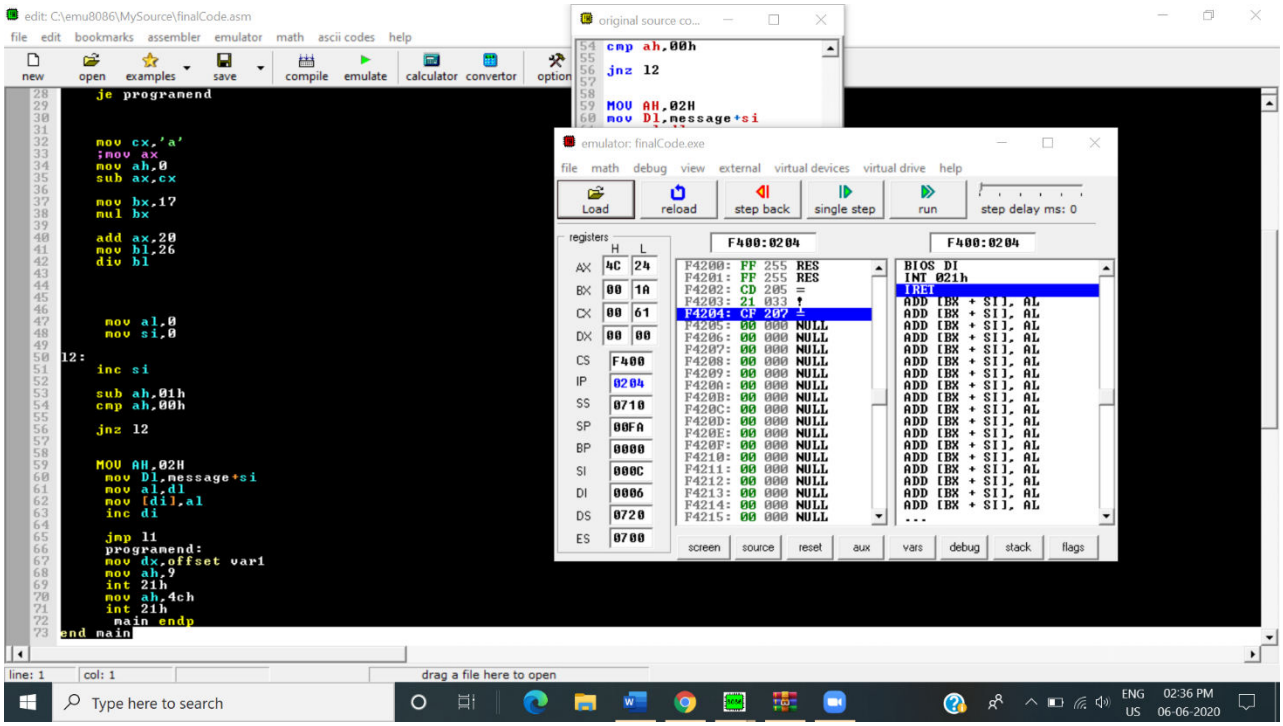
```
    div bl



     mov al,0
     mov si,0

l2:
    inc si

    sub ah,01h
    cmp ah,00h

    jnz l2


    MOV AH,02H
     mov Dl,message+si
     mov al,dl
     mov [di],al
     inc di

     jmp l1
     programend:
     mov dx,offset var1
     mov ah,9
     int 21h
     mov ah,4ch
     int 21h
     main endp

end main
```

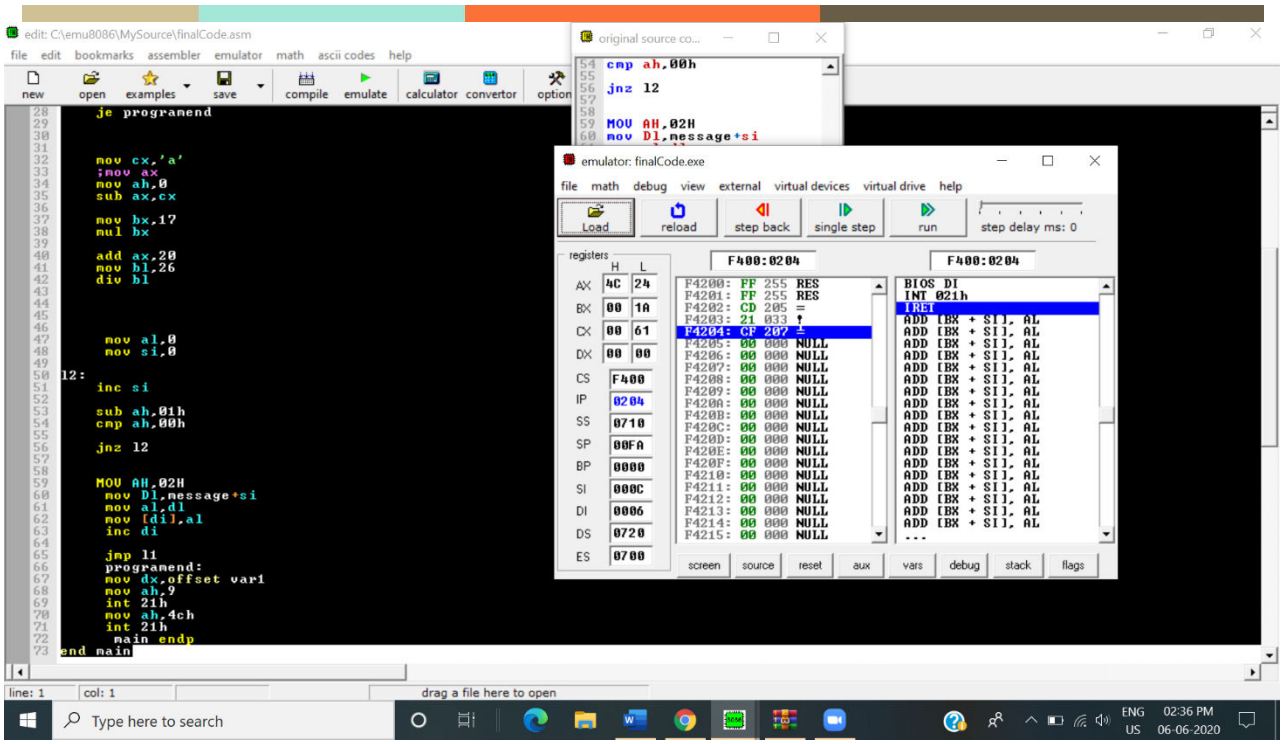
Screenshots:


edit: C:\emu8086\MySource\finalCode.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor option
original source co...
mov ah,1
int 21h
cmp al,13
je programend
emulator screen (80x25 chars)
twenty
clear screen
change font
delay ms: 0
SI 000C
DI 0006
DS 0720
ES 0700
screen source reset aux vars debug stack flags
je programend
mov cx,'a'
;mov ax
mov ah,0
sub ax,cx
mov bx,17
mul bx
add ax,20
mov bl,26
div bl
mov al,0
mov si,0
l2:
inc si
sub ah,01h
cmp ah,00h
jnz l2
MOV AH,02H
mov Dl,message+si
mov al,dl
mov [di],al
inc di
jmp l1
programend:
mov dx,offset var1
mov ah,9
int 21h
mov ah,4ch
int 21h
main endp
end main
line: 1
col: 1
drag a file here to open
Type here to search
ENG US
02:36 PM
06-06-2020


edit: C:\emu8086\MySource\finalCode.asm
file edit bookmarks assembler emulator math ascii codes help
original source co...
cmp ah,00h
jnz l2
emulator screen (80x25 chars)
fekhfn
clear screen
change font
emulator: finalCode.exe
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
AX 4C 24
BX 00 1A
CX 00 61
DX 00 00
CS F400
IP 0204
SS 0710
SP 00FA
BP 0000
SI 000C
DI 0006
DS 0720
ES 0700
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
OK
screen source reset aux vars debug stack flags
je programend
mov cx,'a'
;mov ax
mov ah,0
sub ax,cx
mov bx,17
mul bx
add ax,20
mov bl,26
div bl
mov al,0
mov si,0
l2:
inc si
sub ah,01h
cmp ah,00h
jnz l2
MOV AH,02H
mov Dl,message+si
mov al,dl
mov [di],al
inc di
jmp l1
programend:
mov dx,offset var1
mov ah,9
int 21h
mov ah,4ch
int 21h
main endp
end main
line: 1
col: 1
drag a file here to open
Type here to search
ENG US
02:36 PM
06-06-2020


edit: C:\emu8086\MySource\finalCode.asm
original source co...
emulator screen (80x25 chars)
fekhfn
clear screen
change font
delay ms: 0
Type here to search


edit: C:\emu8086\MySource\finalCode.asm
original source co...
emulator: finalCode.exe
Load
reload
step back
single step
run
step delay ms: 0
Type here to search




```
    je programend


    mov cx,'a'
    ;mov ax
    mov ah,0
    sub ax,cx

    mov bx,17
    mul bx

    add ax,20
    mov bl,26
    div bl



     mov al,0
     mov si,0

12:
    inc si

    sub ah,01h
    cmp ah,00h

    jnz 12

    MOV AH,02H
     mov Dl,message+si
     mov al,dl
     mov [di],al
     inc di

     jmp 11
     programend:
     mov dx,offset var1
     mov ah,9
     int 21h
     mov ah,4ch
     int 21h
      main endp
end main
```