Multiplication of Two numbers using shift logical operators

[org 0x0100]

jmp start

multiplicand: db 13

multiplier: db 5

result : dw 0

start:

mov cx,4

mov al, [multiplier]

mov dl, [multiplicand]

lup:

shr al,1

jnc skip

add [result], dl

skip:

shl dl, 1

sub cx, 1

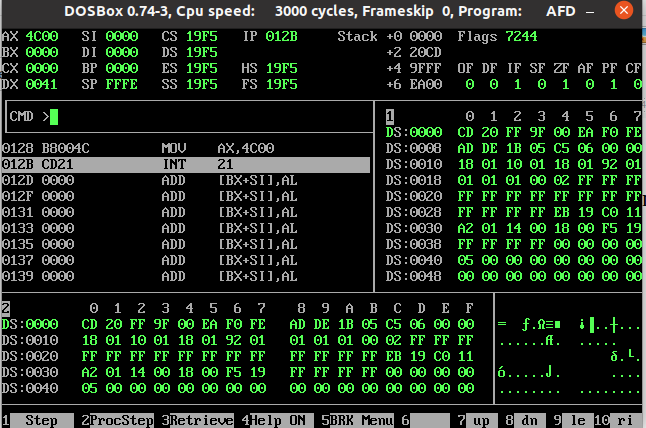
jnz lup

mov dx, 0

mov dx, [result]

mov ax, 0x4c00

int 0x21



1: Here we are using logical operators to multiply two numbers.

2: Firstly we are moving 13 in Multiplcand label and 3 in multiplier label

3: We have created the Result

4: Then we have moved 4 in cx becauses it is 4-bit multiplication

5: And we are moving multipler in al register and multiplicand in dl register because we want to shift both of these

6: And we have created a loop

7: now we are shifting ri8 the al means multiplier if there will be carry generated then the multiplicand will be added in result and if carry not generated it will not add the multiplicand in the result

8: And at the last we are moving the result in dx register..

9: The multiplication of 13 and 5 is 65 in decimal and in hexadecimal value it is 41