

# Saad Ahmad (20P-0051)

First we initialize multiplicand by 13 in db in 8 bit because we need to multiply by 4 bit number so the result will be in 8 bits and we will also initialize multiplier by 5. Now we need to apply this loop 4 times to move cl to 4. Now we will move multiplicand value to bl and multiplier to dl. Now loop will start and shr dl, mean will start the revolution so that right bit will move to carry flag if the carry flag do not set then shift to next bit and if set then add to result after the loop is completed the decrement cl and jump to loop again so that it will Run again.

Output:

The screenshot shows a Linux desktop environment with a terminal window titled "DOSBox Emulator". The terminal displays assembly code for a multiplication program. The code initializes variables, sets up a loop counter, and performs a 4-bit multiplication using a bit-shifting technique. The right side of the terminal shows the CPU dump, which is a memory dump of the CPU's state. The dump includes registers (AX, BX, CX, DX, SI, DI, IP, CS, DS, SS, ES, FS, GS), stack information, and flags. The CPU dump shows the progression of the multiplication process, with values like 13, 5, and their product 65 being manipulated through shifts and adds. The DOSBox interface at the bottom provides various control options for the emulator.

```
Activities DOSBox Emulator ▾
File Edit Selection View Go Run Terminal Help
code > c04-01.asm X
1 [org 0x100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1         ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl   ; only add if CF IS SET
20
21     skip:
22         shl bl, 1         ; always shift the multiplicand
23
24         dec cl
25         jnz checkbit
26
27     mov ax, 0x4c00
28     int 0x21
29
30
31
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

IP	DS:0000	0 1 2 3 4 5 6 7
0x0000	SI 0000 CS 19F5 IP 0100	Stack +0 0000 Flags 7202
0x0000	DI 0000 DS 19F5	+2 20CD
0x0020	BP 0000 ES 19F5 HS 19F5	+4 9FFF OF DF IF SF ZF RF PF CF
0x0000	SP FFFF SS 19F5 FS 19F5	+6 E908 0 0 1 0 0 0 0 0 0

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

IP	DS:0000	0 1 2 3 4 5 6 7
0x0000	OR AX,0005	DS:0000 CB 20 10 05 C5 00 00
0x0003	0103 000500 OR AX,0005	DS:0018 10 01 10 01 10 01 92 01
0x0006	B104 MOU CL,04	DS:0028 FF FF FF FF FF FF FF
0x0009	0000000001 MOU DS,0003	DS:0038 00 00 00 00 00 00 00 00
0x000C	00100401 MOU DL,101041	DS:003B 02 01 14 00 10 00 F5 19
0x0010	D0E9 SHR DL,1	DS:0040 FF FF FT FF 00 00 00 00
0x0012	7304 JNC 0118	DS:0044 05 00 00 00 00 00 00 00
0x0014	00010501 ADD (0105),BL	DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00

DS:0003 02 01 14 00 10 00 F5 19 FF FF FT FF EB 19 C0 11

DS:0020 FF FF FT FF FT FF FF FF FF FF FF EB 19 C0 11

DS:0030 02 01 14 00 10 00 F5 19 FF FF FT FF 00 00 0

Activities DOSBox Emulator ▾ Wed 27 Oct 1:42:37 PM

c04-01.asm - Desktop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

c04-01.asm X

coal > c04-01.asm

```
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12 mov cl, 4              ; how many times we need to run the loop
13 mov bl, [multiplicand]
14 mov dl, [multiplier]
15
16
17 checkbit:
18 shr dl, 1              ; do the rotation so that right bit is thrown in CF
19 jnc skip
20 add [result], bl        ; only add if CF IS SET
21
22
23 skip:
24 shr bl, 1              ; always shift the multiplicand
25
26 dec cl
27 jnz checkbit
28
29
30 mov ax, 0x4c00
31 int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

	0	1	2	3	4	5	6	7	C	D	E	F	
DS:0000	CD	20	FF	9F	06	EB	FF	AD	DE	1B	65	C5	00
DS:0001	10	10	10	10	10	10	10	01	01	01	01	01	00
DS:0002	01	FF											
DS:0003	02	01	14	00	18	05	19	FF	FF	FF	FF	FF	FF
DS:0004	05	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStop 3 Retrieve 4 Help On 5 DIR Mem 6 Up 7 Up 8 dn 9 Go Live 10 Help

The screenshot shows a Linux desktop environment with the Unity interface. A terminal window titled 'c04-01.asm - Desktop - Visual Studio Code' is open, displaying assembly code for a multiplication program. The code uses registers AX, BX, CX, DX, SI, DI, BP, SP, and ES. It includes instructions like MUL, SHL, ADD, DEC, and JNC. The assembly code is as follows:

```
c04-01.asm
[org 0x100]
jmp start

multiplicand: db 13      ; 4-bit number, save space of 8-bits
multiplier:  db 5        ; 4-bit
result:      db 0        ; 8-bit result

start:
    mov cl, 4            ; how many times we need to run the loop
    mov bl, [multiplicand]
    mov dl, [multiplier]

    checkbit:
        shr dl, 1         ; do the rotation so that right bit is thrown in CF
        jnc skip
        add [result], bl   ; only add if CF IS SET

        skip:
        shl bl, 1          ; always shift the multiplicand

    dec cl
    jnz checkbit

    mov ax, 0x4c00
    int 0x21
```

The DOSBox emulator window is visible in the background, showing the assembly code and the CPU register dump. The CPU register dump table is as follows:

	0	1	2	3	4	5	6	7	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	00	00	00	=	f..=	4..+	..	..
DS:0010	10	10	10	10	01	10	01	01	01	01	00	FF	00
DS:0020	FF	FT	FF	FF	FF	EB							
DS:0030	02	00	00	00	00	00	00	00	19	00	00	00	11
DS:0040	02	00	00	00	00	00	00	00	00	00	00	00	00

The DOSBox interface also shows various status indicators and a menu bar at the top.

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X:0000	SI:0000	CS:19F5	IP:010C	Stack:0 0000	Flags:2209
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF OF FF CF
DX:0000	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0100 00100001	MUL BL,[0101]	DS:0000	0 1 2 3 4 5 6 7
0110 00100001	MUL BL,[0101]	DS:0000	CD 20 FF 9F 00 Ed F6 FF
0111 0000	SHL BL,1	DS:0000	AD DE 18 05 C5 06 00 00
0112 7304	JNC 0118	DS:0000	10 01 10 01 10 01 01 01
0113 00100561	MUL [0105],BL	DS:0000	01 01 01 00 FF 00 01 FF
0114 0000	SHL BL,1	DS:0000	DS:0000
0115 00E3	MUL [0105],BL	DS:0000	FF FF FF EB 19 C8 11
0116 FE03	MUL BL,1	DS:0000	00 00 00 00 00 00 00 00
0117 FE03	JNC 0119	DS:0000	FF FF FF EB 19 C8 11
0118 BB004C	MUL AX,4000	DS:0000	05 00 00 00 00 00 00 00
0119 CD21	INT 21	DS:0000	00 00 00 00 00 00 00 00

! Step 2ProcStep 3Retrieve 4Help 5M 6BRK Memd G 7Up 8dn 9le 10ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X:0000	SI:0000	CS:19F5	IP:0110	Stack:0 0000	Flags:2209
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF OF FF CF
DX:0000	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0100 00100001	MUL BL,[0104]	DS:0000	0 1 2 3 4 5 6 7
0110 0000	SHL BL,1	DS:0000	CD 20 FF 9F 00 Ed F6 FF
0111 7304	JNC 0118	DS:0000	AD DE 18 05 C5 06 00 00
0112 00100561	MUL [0105],BL	DS:0000	10 01 10 01 10 01 01 01
0113 0000	SHL BL,1	DS:0000	01 01 01 00 FF 00 01 FF
0114 00E3	MUL [0105],BL	DS:0000	DS:0000
0115 FE03	MUL BL,1	DS:0000	FF FF FF EB 19 C8 11
0116 FE03	JNC 0119	DS:0000	00 00 00 00 00 00 00 00
0117 BB004C	MUL AX,4000	DS:0000	FF FF FF EB 19 C8 11
0118 CD21	INT 21	DS:0000	05 00 00 00 00 00 00 00

! Step 2ProcStep 3Retrieve 4Help 5M 6BRK Memd G 7Up 8dn 9le 10ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1           ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:0112	Stack:0 0000	Flags: Z211
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0002	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 1 0 1

Registers:

DS:0000	0 1 2 3 4 5 6 7
SI:0000	CD 20 FF 9F 00 Ed F0 FE
DI:0000	AD DE 1B 05 C5 06 00 00
CX:0004	01 01 01 01 10 01 92 01
BP:0000	01 01 00 FF 00 01 FF
ES:0000	00 00 00 00 00 00 00 00
HS:0000	00 00 00 00 00 00 00 00
DX:0000	00 00 00 00 00 00 00 00
SP:FFFE	FF FF FF FF FF FF FF FF
SS:19F5	FF FF FF EB 19 C8 11
FS:19F5	00 00 00 00 00 00 00 00

Stack:

DS:0000	0 1 2 3 4 5 6 7
0110 D9E5	SHL BL,1
0111 73B1	JNC 0118
0114 001E9501	ADD (0105),BL
0118 D9E3	SHL BL,1
011C FEC9	DEC CL
011C 75F2	JNZ 0119
011D B8004C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,(BP-01)

Memory Dump:

DS:0000	0 1 2 3 4 5 6 7
0110 D9E5	SHL BL,1
0111 73B1	JNC 0118
0114 001E9501	ADD (0105),BL
0118 D9E3	SHL BL,1
011C FEC9	DEC CL
011C 75F2	JNZ 0119
011D B8004C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,(BP-01)

Registers:

DS:0000	0 1 2 3 4 5 6 7
SI:0000	CD 20 FF 9F 00 Ed F0 FE
DI:0000	AD DE 1B 05 C5 06 00 00
CX:0004	01 01 01 01 10 01 92 01
BP:0000	01 01 00 FF 00 01 FF
ES:0000	00 00 00 00 00 00 00 00
HS:0000	00 00 00 00 00 00 00 00
DX:0000	00 00 00 00 00 00 00 00
SP:FFFE	FF FF FF FF FF FF FF FF
SS:19F5	FF FF FF EB 19 C8 11
FS:19F5	00 00 00 00 00 00 00 00

Stack:

DS:0000	0 1 2 3 4 5 6 7
0110 D9E5	SHL BL,1
0111 73B1	JNC 0118
0114 001E9501	ADD (0105),BL
0118 D9E3	SHL BL,1
011C FEC9	DEC CL
011C 75F2	JNZ 0119
011D B8004C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,(BP-01)

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1           ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:0114	Stack:0 0000	Flags: Z211
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0002	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 1 0 1

Registers:

DS:0000	0 1 2 3 4 5 6 7
SI:0000	CD 20 FF 9F 00 Ed F0 FE
DI:0000	AD DE 1B 05 C5 06 00 00
CX:0004	01 01 01 01 10 01 92 01
BP:0000	01 01 00 FF 00 01 FF
ES:0000	00 00 00 00 00 00 00 00
HS:0000	00 00 00 00 00 00 00 00
DX:0000	00 00 00 00 00 00 00 00
SP:FFFE	FF FF FF FF FF FF FF FF
SS:19F5	FF FF FF EB 19 C8 11
FS:19F5	00 00 00 00 00 00 00 00

Stack:

DS:0000	0 1 2 3 4 5 6 7
0110 D9E5	SHL BL,1
0111 73B1	JNC 0118
0114 001E9501	ADD (0105),BL
0118 D9E3	SHL BL,1
011C FEC9	DEC CL
011C 75F2	JNZ 0119
011D B8004C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,(BP-01)

Registers:

DS:0000	0 1 2 3 4 5 6 7
SI:0000	CD 20 FF 9F 00 Ed F0 FE
DI:0000	AD DE 1B 05 C5 06 00 00
CX:0004	01 01 01 01 10 01 92 01
BP:0000	01 01 00 FF 00 01 FF
ES:0000	00 00 00 00 00 00 00 00
HS:0000	00 00 00 00 00 00 00 00
DX:0000	00 00 00 00 00 00 00 00
SP:FFFE	FF FF FF FF FF FF FF FF
SS:19F5	FF FF FF EB 19 C8 11
FS:19F5	00 00 00 00 00 00 00 00

Stack:

DS:0000	0 1 2 3 4 5 6 7
0110 D9E5	SHL BL,1
0111 73B1	JNC 0118
0114 001E9501	ADD (0105),BL
0118 D9E3	SHL BL,1
011C FEC9	DEC CL
011C 75F2	JNZ 0119
011D B8004C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,(BP-01)

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

```
c04-01.asm X
```

```

coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:011B	Stack:0 0000	Flags: 7200
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0002	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0114 001E9561	ADD	(0105),BL		DS:0000	CD 20 FF 9F 00 Ed F6 FE
0115 FE03	SHL	BL,1		DS:0000	AD DE 1B 05 C5 06 00 00
0116 FE03	DEC	CL		DS:0018	10 01 10 01 10 01 00 01 01
011C 75F2	JNZ	0110		DS:0020	FF FF FF FF FF FF FF FF
011D B9004C	MUL	AX,4000		DS:0020	FF FF FF FF FF FF FF FF
0121 CD21	INT	21		DS:0008	AZ 01 14 00 10 00 F5 19
0123 BB46F6	MUL	AX,[BP-001]		DS:0008	FF FF FF FF 00 00 00 00
0126 D1E9	SHL	AX,1		DS:0040	05 00 00 00 00 00 00 00 00 00
0128 D1E9	SHL	AX,1		DS:0040	00 00 00 00 00 00 00 00 00 00

Cmd >

DS:0000	CD 20 FF 9F 00 Ed F6 FE
DS:0010	AD DE 1B 05 C5 06 00 00
DS:0020	FF FF FF FF FF FF FF FF
DS:0030	FF FF FF FF FF FF FF FF
DS:0040	FF FF FF FF FF FF FF FF

Step ProcStep Retrieve Help DM SBRK Memd G ? up ? dn ? le ? rri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

```
c04-01.asm X
```

```

coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:011A	Stack:0 0000	Flags: 7210
DX:0010	DI:0000	BS:19F5		+2 20CD	
CX:0004	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0002	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0

0110 000E	SHL	BL,1		DS:0000	CD 20 FF 9F 00 Ed F6 FE
0111 FE03	DEC	CL		DS:0000	AD DE 1B 05 C5 06 00 00
011C 75F2	JNZ	0119		DS:0018	10 01 10 01 10 01 00 01 01
011E B9004C	MUL	AX,4000		DS:0020	FF FF FF FF FF FF FF FF
0121 CD21	INT	21		DS:0020	FF FF FF EB 19 C8 11
0123 BB46F6	MUL	AX,[BP-001]		DS:0008	AZ 01 14 00 10 00 F5 19
0126 D1E9	SHL	AX,1		DS:0008	FF FF FF FF 00 00 00 00
0128 D1E9	SHL	AX,1		DS:0040	05 00 00 00 00 00 00 00 00 00
0129 C5E9B0	LBS	EX,[BP-20]		DS:0040	00 00 00 00 00 00 00 00 00 00

Cmd >

DS:0000	CD 20 FF 9F 00 Ed F6 FE
DS:0010	AD DE 1B 05 C5 06 00 00
DS:0020	FF FF FF FF FF FF FF FF
DS:0030	FF FF FF FF FF FF FF FF
DS:0040	FF FF FF FF FF FF FF FF

Step ProcStep Retrieve Help DM SBRK Memd G ? up ? dn ? le ? rri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾ Wed 27 Oct 1:42:46 PM

c04-01.asm - Desktop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

c04-01.asm X

```
coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13 ; 4-bit number, save space of 8-bits
6 multiplier: db 5 ; 4-bit
7
8 result: db 0 ; 8-bit result
9
10 start:
11
12 mov cl, 4 ; how many times we need to run the loop
13 mov bl, [multiplicand]
14 mov dl, [multiplier]
15
16
17 checkbit:
18 shr dl, 1 ; do the rotation so that right bit is thrown in CF
19 jnc skip
20 add [result], bl ; only add if CF IS SET
21
22
23 skip:
24 shr bl, 1 ; always shift the multiplicand
25
26 dec cl
27 jnz checkbit
28
29
30 mov ax, 0x4c00
31 int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

	0	1	2	3	4	5	6	7	C	D	E	F				
DS:0000	CD	20	FF	9F	06	EB	F0	FF	A0	DE	1B	65	C5	00	00	= f...# +!...+
DS:0001	1B	00	00	00	00	00	00	00	01	01	01	01	01	01	01	.....R.
DS:0002	0F	FF	01	FF	FF	FF	EB	19	00	11						
DS:0003	02	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	6.....J.
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

1 Step 2 ProcStep 3 Retrieve 4 Inp On 5 Drk Mem 6 Up 7 Up 8 dn 9 In 10 r/r

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

The screenshot shows a Linux desktop environment with a window titled "c04-01.asm - Desktop - Visual Studio Code". The code editor displays assembly instructions for a 64-bit program. The assembly code includes labels like .org 0x1000, jmp start, and various arithmetic operations involving registers AX, BX, CX, DX, SI, DI, BP, ES, SS, and FS. The CPU register pane shows values for AX, BX, CX, DX, SI, DI, BP, ES, SS, and FS. The memory dump pane shows the state of memory at address 0x0000. The DOSBox emulator window shows the assembly code and its execution results.

```
c04-01.asm
File Edit Selection View Go Run Terminal Help

dos > c04-01.asm
[org 0x1000]
jmp start
multiplicand: db 13      ; 4-bit number, save space of 8-bits
multiplier:  db 5        ; 4-bit
result:      db 0        ; 8-bit result
start:
    mov cl, 4            ; how many times we need to run the loop
    mov bl, [multiplicand]
    mov dl, [multiplier]
    ...

    checkbit:
        shr dl, 1        ; do the rotation so that right bit is thrown in CF
        jnc skip
        add [result], bl   ; only add if CF IS SET
        ...

        skip:
        shl bl, 1          ; always shift the multiplicand
        ...

dec cl
jnz checkbit
mov ax, 0x4c00
int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

AX	SI	DS	CX	IP	Stack	Flags
0000	0000	19F5	0110	+0 0000	Flags: 7204	-Z 200D
BX	0010	DI	0000	DS 19F5		
CX	0003	BP	0000	ES 19F5		
DX	0002	SP	FFFF	SS 19F5		
				FS 19F5		
						-4 E9FF OF DF IF SF ZF OF PF CF
						-6 E966 0 0 1 6 0 0 1 0

CDP >	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
011C 20F2	JNC	0110														
011D B80A	SHL	BL,1														
011E 7304	JNC	011B														
011F 001E6501	ADD	(0101),BL														
011B D0E3	SHL	HL,1														
011C 43E9	DEC	CL														
011C 70F9	IM2	0110														
011E B8004C	MUL	AX,4000														
0121 CD21	INT	Z1														
DS:0000	CD 20 FF 00 E9 F0 FE	A0 DE 1B 05	C5 00 00	= f...= 4...+...												
DS:0010	10 10 10 01 10 01 92 01	01 01 01 00 FF 00 01 FF		.....A. ....A.												
DS:0020	FF FF FF FF FF FF FF	FF FF FF FF EB 19 CC 11														
DS:0030	02 01 00 00 00 00 00 00	FF FF FF FF FF FF FF 00 00 00														
DS:0040	02 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00														

i Step 2ProcStep 3Retrieve 4Help ON 5ORK Menu 6 7 up 8 dn 9 Ic 10 ri

Activities DOSBox Emulator ▾ Wed 27 Oct 1:42:48 PM

c04-01.asm - Desktop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

c04-01.asm X

```
coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13 ; 4-bit number, save space of 8-bits
6 multiplier: db 5 ; 4-bit
7
8 result: db 0 ; 8-bit result
9
10 start:
11
12 mov cl, 4 ; how many times we need to run the loop
13 mov bl, [multiplicand]
14 mov dl, [multiplier]
15
16
17 checkbit:
18 shr dl, 1 ; do the rotation so that right bit is thrown in CF
19 jnc skip
20 add [result], bl ; only add if CF IS SET
21
22
23 skip:
24 shr bl, 1 ; always shift the multiplicand
25
26 dec cl
27 jnz checkbit
28
29
30 mov ax, 0x4c00
31 int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

	0	1	Z	3	4	5	6	7	C	D	E	F	
DS:0000	CD	20	FF	9F	06	EB	FF	A0	DE	1B	65	C5	00
DS:0001	10	00	00	00	00	00	00	01	01	01	01	01	00
DS:0002	00	FF	FF	FF	FF	FF	FF	01	FF	FF	FF	FF	11
DS:0003	02	01	14	00	18	00	F5	19	FF	FF	FF	00	00
DS:0004	05	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStep 3 Retrieve 4 In 5 Out 6 Print Mem 7 Up 8 Down 9 Reg 10 Run

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

The screenshot shows a Linux desktop environment with a window titled "c04-01.asm - Desktop - Visual Studio Code". The code editor displays assembly code for a 16-bit program. The assembly code includes instructions for setting up registers, performing a multiplication loop, and handling carry flags. The DOSBox window shows the CPU register state (AX=0000, BX=0010, CX=0003, DX=0001, SI=0000, DI=0000) and memory dump windows for various segments like DS, ES, SS, and FS.

```
c04-01.asm
File Edit Selection View Go Run Terminal Help
File c04-01.asm ×
dos > c04-01.asm
[org 0x100]
jmp start
multiplicand: db 13 ; 4-bit number, save space of 8-bits
multiplier: db 5 ; 4-bit
result: db 0 ; 8-bit result
start:
    mov cl, 4 ; how many times we need to run the loop
    mov bl, [multiplicand]
    mov dl, [multiplier]
    mov dl, [multiplier]

    checkbit:
        shr dl, 1 ; do the rotation so that right bit is thrown in CF
        jnc skip
        add [result], bl ; only add if CF IS SET

        skip:
        shl bl, 1 ; always shift the multiplicand

    dec cl
    jnz checkbit
    mov ax, 0x4c00
    int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

AX 0000	SI 0000	CS 1975	IP 0110	Stack +0 0000	Flags 2110
BX 0010	DI 0000	DS 1975		+2 200D	
CX 0003	BP 0000	ES 1975	HS 1975	+4 9FFF	OF DF IF SF ZF OF PF CF
DX 0001	SP FFFF	SS 1975	FS 1975	+6 E966	0 0 1 6 0 1 6 0

[CPU] >	0 1 2 3 4 5 6 7
G115 2704 JNC G116	DS:0000 CD 20 FF 9F 00 00
G116 B5E9 SHL BL,I	DS:0000 AD DE 1B 05 C5 00 00 00
G116 FE29 DEC CL	DS:0000 00 00 00 00 00 00 00 00
G11C 75F2 JMZ G110	DS:0000 01 01 01 00 FF 00 00 FF
G11F B0094C MOV AX, 4C00	DS:0000 00 00 00 00 00 00 00 00
G122 8002E1 IMUL AX, DS:BX	DS:0000 00 00 00 00 00 00 00 00
G123 8002E6 IMUL AX, DS:[BX+00]	DS:0000 00 00 00 00 00 00 00 00
G126 B1D9 SHL AX, I	DS:0000 00 00 00 00 00 00 00 00
G128 B1E0 SHL AX, I	DS:0000 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 00 00 00	= f...= 4 ...+
DS:0010 00 00 00 00 00 00 00 00	.....,.....
DS:0020 FF FF FF FF FF FF FF	.....,.....
DS:0030 00 00 00 00 00 00 00 00	.....,.....
DS:0040 00 00 00 00 00 00 00 00	.....,.....

i Step | 2ProcStep | 3Retrieve | 4Help On | 5ORK Menu | 6 7 up | 8 dn | 9 1e | 10 ri

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	CD 20 FF 9F 00	Ed F0 FE
DS:0010	AD DE 1B 05 C5 06	00 00 00
DS:0018	01 01 01 00 FF 00 01	FF
DS:0020	FF FF FF FF FF FF	FF
DS:0028	02 01 04 00 FF 00 00	FF
DS:0030	00 00 00 00 00 00 00	00
DS:0040	00 00 00 00 00 00 00	00

DS:0000 0 1 2 3 4 5 6 7 DS:0000 CD 20 FF 9F 00 Ed F0 FE
DS:0010 AD DE 1B 05 C5 06 00 00 00
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF
DS:0028 02 01 04 00 FF 00 00 FF
DS:0030 00 00 00 00 00 00 00
DS:0040 00 00 00 00 00 00 00

| Step | ProcStep | Retrieve | Help | DM | SBRK | Mem | G | ? up | ? up | B dn | S le | I G r |

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	CD 20 FF 9F 00	Ed F0 FE
DS:0010	AD DE 1B 05 C5 06	00 00 00
DS:0018	01 01 01 00 FF 00 01	FF
DS:0020	FF FF FF FF FF FF	FF
DS:0028	02 01 04 00 FF 00 00	FF
DS:0030	00 00 00 00 00 00 00	00
DS:0040	00 00 00 00 00 00 00	00

DS:0000 0 1 2 3 4 5 6 7 DS:0000 CD 20 FF 9F 00 Ed F0 FE
DS:0010 AD DE 1B 05 C5 06 00 00 00
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF
DS:0028 02 01 04 00 FF 00 00 FF
DS:0030 00 00 00 00 00 00 00
DS:0040 00 00 00 00 00 00 00

| Step | ProcStep | Retrieve | Help | DM | SBRK | Mem | G | ? up | ? up | B dn | S le | I G r |

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X:0000	SI:0000	CS:19F5	IP:0110	Stack:0 0000	Flags:7200
0X:0004	DI:0000	BS:19F5		+2 20CD	
CX:0002	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0001	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[CMD >]	0 1 2 3 4 5 6 7
011C 75F2	JNZ 0119
0110 B9E9	SUB BL,1
0112 7301	JNC 0118
0114 001E9501	ADD (0105),BL
0116 B9E9	SUB BL,1
0118 7302	JNC CL,
011C 75F2	JNZ 0119
011E B9E94C	MUL AX,4000
0121 CD21	INT 21

DS:0000	CD 20 FF 9F 00 Ed F0 FE
DS:0010	10 01 10 01 10 01 92 01
DS:0020	01 01 01 00 FF 00 01 FF
DS:0030	FF FF FF EB 19 C0 11
DS:0040	02 00 14 00 00 00 00 00

! Step 2ProcStep 3Retrieve 4Help 5IN 6BRK MemD G 7Up 8Down 9Left 10Right

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X:0000	SI:0000	CS:19F5	IP:0112	Stack:0 0000	Flags:7255
0X:0004	DI:0000	BS:19F5		+2 20CD	
CX:0002	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0001	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[CMD >]	0 1 2 3 4 5 6 7
0110 B9E9	SUB BL,1
0112 7301	JNC 0119
0114 001E9501	ADD (0105),BL
0116 D9E3	SHL BL,1
0118 FEC9	DEC CL
011A 7302	JNC 0119
011C B9E94C	MUL AX,4000
0121 CD21	INT 21
0123 BB46F6	MUL AX,[BP+01]

DS:0000	CD 20 FF 9F 00 Ed F0 FE
DS:0010	10 01 10 01 10 01 92 01
DS:0020	01 01 01 00 FF 00 01 FF
DS:0030	FF FF FF EB 19 C0 11
DS:0040	02 00 14 00 00 00 00 00

! Step 2ProcStep 3Retrieve 4Help 5IN 6BRK MemD G 7Up 8Down 9Left 10Right

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16
17     checkbit:
18         shr dl, 1          ; do the rotation so that right bit is thrown in CF
19         jnc skip
20         add [result], bl    ; only add if CF IS SET
21
22     skip:
23         shl bl, 1          ; always shift the multiplicand
24
25     dec cl
26     jnz checkbit
27
28
29     mov ax, 0x4c00
30     int 0x21
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X 0000	SI 0000	CS 19F5	IP 0114	Stack +0 0000	Flags 7255
0X 0004	DI 0000	BS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 E900	6 0 1 0 1 1 1 1

[CMD >]	0 1 2 3 4 5 6 7			
0112 7204	JNC 0118	0000	CD 20 FF 9F 00 Ed F0 FE	
0114 00195601	SHL 0119	(0119),BL	0000	AD DE 1B 05 C5 06 00 00
0118 D0E3	SHL BL,1		0000	AD DE 1B 05 C5 06 00 00
011A FEC9	BEC CL		0000	AD DE 1B 05 C5 06 00 00
011C 75F2	JNE 0110		0000	AD DE 1B 05 C5 06 00 00
011E B0094C	MUL AX,4000		0000	FF FF FF EB 19 C8 11
0120 CD21	INT 21		0000	AD DE 1B 05 C5 06 00 00
0123 B846F6	MUL AX,[BP-04]		0000	FF FF FF EB 19 C8 11
0126 D1E9	SHL AX,1		0000	AD DE 1B 05 C5 06 00 00
0128 D1E9	SHL AX,1		0000	AD DE 1B 05 C5 06 00 00

DS:0000 CD 20 FF 9F 00 Ed F0 FE  
DS:0010 10 01 10 01 10 01 92 01 01 01 01 00 FF 00 01 FF  
DS:0018 01 01 01 00 FF 00 01 FF  
DS:0020 FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF FF FF FF  
DS:0030 AD 01 14 00 10 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

2 Step ProcStep Retrieve Help DM SBRK MemD G ? up B dn S le I G ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16
17     checkbit:
18         shr dl, 1          ; do the rotation so that right bit is thrown in CF
19         jnc skip
20         add [result], bl    ; only add if CF IS SET
21
22     skip:
23         shl bl, 1          ; always shift the multiplicand
24
25     dec cl
26     jnz checkbit
27
28
29     mov ax, 0x4c00
30     int 0x21
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X 0000	SI 0000	CS 19F5	IP 0114	Stack +0 0000	Flags 7214
0X 0004	DI 0000	BS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 E900	6 0 1 0 1 1 1 1

[CMD >]	0 1 2 3 4 5 6 7			
0114 00195601	SHL 0119	(0119),BL	0000	CD 20 FF 9F 00 Ed F0 FE
0116 FEC9	BEC CL	0119	0000	AD DE 1B 05 C5 06 00 00
0118 FEC9	BEC CL		0000	AD DE 1B 05 C5 06 00 00
011C 75F2	JNE 0110		0000	AD DE 1B 05 C5 06 00 00
011E B0094C	MUL AX,4000		0000	FF FF FF EB 19 C8 11
0120 CD21	INT 21		0000	AD DE 1B 05 C5 06 00 00
0123 B846F6	MUL AX,[BP-04]		0000	FF FF FF EB 19 C8 11
0126 D1E9	SHL AX,1		0000	AD DE 1B 05 C5 06 00 00
0128 D1E9	SHL AX,1		0000	AD DE 1B 05 C5 06 00 00

DS:0000 CD 20 FF 9F 00 Ed F0 FE  
DS:0010 10 01 10 01 10 01 92 01 01 01 01 00 FF 00 01 FF  
DS:0018 01 01 01 00 FF 00 01 FF  
DS:0020 FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF FF FF FF  
DS:0030 AD 01 14 00 10 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

2 Step ProcStep Retrieve Help DM SBRK MemD G ? up B dn S le I G ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾ Wed 27 Oct 1:42:56 PM

c04-01.asm - Desktop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

c04-01.asm X

```
coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12 mov cl, 4              ; how many times we need to run the loop
13 mov bl, [multiplicand]
14 mov dl, [multiplier]
15
16
17 checkbit:
18 shr dl, 1              ; do the rotation so that right bit is thrown in CF
19 jnc skip
20 add [result], bl        ; only add if CF IS SET
21
22
23 skip:
24 shr bl, 1              ; always shift the multiplicand
25
26 dec cl
27 jnz checkbit
28
29
30 mov ax, 0x4c00
31 int 0x21
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value	Reg	Value						
AX	0000	S1	0000	CS	19'F	IP	011A	Stack	+0 0000	Flags	7210																								
BX	0008	DI	0009	DS	19'F							+2 20CD																							
CX	0000	BP	0000	ES	19'F	HS	19'F5				+4 20E0	OF	RF	IF	SF	ZF	HF	PF	CF																
DX	0000	SI	1117	SS	19'F	FS	19'F5				+6 E568	0	0	1	0	0	1	0	0																

[CPU >]

Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op	Op							
011B	00E3	SHL	BL, 1																																			
011B	FEC9	DEC	CL																																			
011C	75F2	JNZ	0110																																			
011D	00994C	MUL	AX, 0000																																			
0121	0000	IMR	ZF																																			
0121	00H4F6	MUL	AX, [BP-0014]																																			
0126	00E9	SHL	AX, 1																																			
0128	00E9	SHL	AX, 1																																			
012A	C5E9E6	LBS	BX, [BP-20]																																			

1 Step 2 ProcStep 3 Retrieve 4 InIp On 5 DRk Mem 6 Up 7 Up 8 dn 9 In 10 ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

A screenshot of a Linux desktop environment. The window title is "DOSBox Emulator". Inside the window, a terminal-like interface shows assembly code for a program named "c04-01.asm". The assembly code includes instructions like `jmp start`, `multiplicand: db 13`, `multiplier: db 5`, and various loop and conditional instructions. To the right of the assembly code, there is a CPU register dump and a memory dump window. The CPU dump shows registers AX through DX with their corresponding values. The memory dump shows memory starting at address 0x0000 with various byte values. At the bottom of the DOSBox window, there is a toolbar with icons for step operations: Step, ProcStep, Retire, Help, ROM, SPARK Menu, Up, Down, Left, Right, and Exit.

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

```
c04-01.asm X
```

```

coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X 0000	SI 0000	CS 19F5	IP 0110	Stack +0 0000	Flags 7200
0X 0000	DI 0000	BS 19F5		+2 20CD	
0X 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
0X 0000	SP FFFE	SS 19F5	FS 19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[CMD >]	0 1 2 3 4 5 6 7
011C 75F2	JNC 0110
0110 B909	SHL BL,1
0112 7304	JNC 0110
0114 001E9501	ADD [0105],BL
0116 B909	SHL BL,1
0118 7303	BEC CL
011C 75F2	JNC 0110
011E B9094C	MUL AX,4000
0121 CD21	INT 21

[DS:0000]	0 1 2 3 4 5 6 7
011C 75F2	JNC 0110
0110 B909	SHL BL,1
0112 7304	JNC 0110
0114 001E9501	ADD [0105],BL
0116 B909	SHL BL,1
0118 7303	BEC CL
011C 75F2	JNC 0110
011E B9094C	MUL AX,4000
0121 CD21	INT 21

! Step 2 ProcStep 3 Retrieve 4 Help 5N 5BRK MemD G 7 up 8 dn 9 le 10 ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

```
c04-01.asm X
```

```

coal > c04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

0X 0000	SI 0000	CS 19F5	IP 0110	Stack +0 0000	Flags 7254
0X 0000	DI 0000	BS 19F5		+2 20CD	
0X 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
0X 0000	SP FFFE	SS 19F5	FS 19F5	+6 E900	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[CMD >]	0 1 2 3 4 5 6 7
0112 7304	JNC 0110
0110 B909	SHL BL,1
0112 FE09	BEC CL
0114 75F2	JNC 0110
0116 B9094C	MUL AX,4000
0118 B909	SHL BL,1
011A 7303	BEC CL
011C B9094C	MUL AX,[BP-01]
0126 D1E9	SHL AX,1
0120 D1E9	SHL AX,1

[DS:0000]	0 1 2 3 4 5 6 7
0112 7304	JNC 0110
0110 B909	SHL BL,1
0112 FE09	BEC CL
0114 75F2	JNC 0110
0116 B9094C	MUL AX,4000
0118 B909	SHL BL,1
011A 7303	BEC CL
011C B9094C	MUL AX,[BP-01]
0126 D1E9	SHL AX,1
0120 D1E9	SHL AX,1

! Step 2 ProcStep 3 Retrieve 4 Help 5N 5BRK MemD G 7 up 8 dn 9 le 10 ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:0110	Stack +0 0000	Flags 7000
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0001	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0000	SP:FFFE	SS:19F5	FS:19F5	+6 E900	1 0 1 1 0 1 0 0

0:0000 0 1 2 3 4 5 6 7 DS:0000 CD 20 FF 9F 00 Ed F0 FE
0:0010 10 01 10 01 10 01 92 01 DS:0000 AD DE 18 05 C5 06 00 00
0:0020 01 01 01 00 FF 00 01 FF DS:0000 00 00 00 00 00 00 00 00
0:0030 02 01 04 00 00 00 00 15 DS:0000 FF FF FF EB 19 C8 11
0:0040 05 00 00 00 00 00 00 00 DS:0000 00 00 00 00 00 00 00 00

1: Step 2:procStep 3:Retrieve 4:help 5:N 6:BRK Memd G 7:up 8:dn 9:le 10:ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17         shr dl, 1          ; do the rotation so that right bit is thrown in CF
18         jnc skip
19         add [result], bl    ; only add if CF IS SET
20
21     skip:
22         shl bl, 1          ; always shift the multiplicand
23
24     dec cl
25     jnz checkbit
26
27     mov ax, 0x4c00
28
29     int 0x21
30
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	SI:0000	CS:19F5	IP:011C	Stack +0 0000	Flags 7244
DX:0000	DI:0000	BS:19F5		+2 20CD	
CX:0001	BP:0000	ES:19F5	HS:19F5	+4 9FFF	OF DF IF SF ZF AF FF CF
DX:0000	SP:FFFE	SS:19F5	FS:19F5	+6 E900	0 0 1 6 1 0 1 0

0:0000 0 1 2 3 4 5 6 7 DS:0000 CD 20 FF 9F 00 Ed F0 FE
0:0010 10 01 10 01 10 01 92 01 DS:0000 AD DE 1B 05 C5 06 00 00
0:0020 01 01 01 00 FF 00 01 FF DS:0000 00 00 00 00 00 00 00 00
0:0030 01 1E 02 01 00 00 00 00 DS:0000 FF FF FF EB 19 C8 11
0:0040 05 00 00 00 00 00 00 00 DS:0000 00 00 00 00 00 00 00 00

1: Step 2:procStep 3:Retrieve 4:help 5:N 6:BRK Memd G 7:up 8:dn 9:le 10:ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17
18         shr dl, 1          ; do the rotation so that right bit is thrown in CF
19         jnc skip
20         add [result], bl    ; only add if CF IS SET
21
22     skip:
23         shl bl, 1          ; always shift the multiplicand
24
25     dec cl
26     jnz checkbit
27
28
29     mov ax, 0x4c00
30     int 0x21
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	CD 20 FF 9F 00	Ed F0 FE
DX:0000	DI 0000	BS 19F5
CX:0000	BP 0000	ES 19F5 HS 19F5
DX:0000	SP FFTE	SS 19F5 FS 19F5
DS:0000	0 1 2 3 4 5 6 7	
DS:0000	CD 20 FF 9F 00	Ed F0 FE
DS:0010	10 01 10 01 10 01 92 01	01 01 01 00 FF 00 01 FF
DS:0010	FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11
DS:0010	02 00 04 00 00 00 00 00	00 00 00 00 00 00 00 00
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

! Step 2 ProcStep 3 Retrieve 4 Help 5N 5BRK MemD G 7 up 8 dn 9 le 10 ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live

Activities DOSBox Emulator ▾

File Edit Selection View Go Run Terminal Help

c04-01.asm x

```

coal > cd04-01.asm
1 [org 0x0100]
2
3 jmp start
4
5 multiplicand: db 13      ; 4-bit number, save space of 8-bits
6 multiplier:  db 5        ; 4-bit
7
8 result:    db 0          ; 8-bit result
9
10 start:
11
12     mov cl, 4            ; how many times we need to run the loop
13     mov bl, [multiplicand]
14     mov dl, [multiplier]
15
16     checkbit:
17
18         shr dl, 1          ; do the rotation so that right bit is thrown in CF
19         jnc skip
20         add [result], bl    ; only add if CF IS SET
21
22     skip:
23         shl bl, 1          ; always shift the multiplicand
24
25     dec cl
26     jnz checkbit
27
28
29     mov ax, 0x4c00
30     int 0x21
31

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -

DS:0000	CD 20 FF 9F 00	Ed F0 FE
DX:0000	DI 0000	BS 19F5
CX:0000	BP 0000	ES 19F5 HS 19F5
DX:0000	SP FFTE	SS 19F5 FS 19F5
DS:0000	0 1 2 3 4 5 6 7	
DS:0000	CD 20 FF 9F 00	Ed F0 FE
DS:0010	10 01 10 01 10 01 92 01	01 01 01 00 FF 00 01 FF
DS:0010	FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11
DS:0010	02 00 04 00 00 00 00 00	00 00 00 00 00 00 00 00
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

! Step 2 ProcStep 3 Retrieve 4 Help 5N 5BRK MemD G 7 up 8 dn 9 le 10 ri

Ln 31, Col 15 Spaces: 4 UTF-8 LF x86 and x86\_64 Assembly Go Live