МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №3**

**з дисципліни «Архітектура комп’ютера»**

**на тему: « Арифметичні команди процесора 8086»**

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

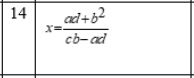
Нежуміра О. І.

Дніпро, 2020

**Тема.** Арифметичні команди процесора 8086.

**Мета.** Вивчити арифметичні команди мови Асемблера, набути практичних навичок їх використання.

Розробити програму для проведення розрахунків за заданим виразом відповідно до варіантів з таблиці:



A = -12

B = 7

C = -17

D = 9

3. Покроковий розрахунок виразу з зазначенням результатів у десятковій і шістнадцятковій системах числення;

|  |  |  |  |
| --- | --- | --- | --- |
| Програмний код | Формула | Результати | |
| Десяткове значення | Машинний код |
| a |  | -12 | FFF4 |
| b |  | 7 | 0007 |
| c |  | -17 | FFEF |
| d |  | 9 | 0009 |
| mov ax, c | ax = c | -17 | FFEF |
| imul b | ax = c \* b | -119 | FF89 |
| mov cx, ax | cx = c \* b | -119 | FF89 |
| mov ax, a | ax = a | -12 | FFF4 |
| imul d | ax = a \* d | -108 | FF94 |
| sub cx, ax | cx = c \* b - a \* d | -11 | FFF5 |
| mov ax, a | ax = a | -12 | FFF4 |
| imul d | ax = a \* d | -108 | FF94 |
| mov bx, ax | bx = a \* d | -108 | FF94 |
| mov ax, b | ax = b | 7 | 0007 |
| imul ax | ax = b \* b | 49 | 31 |
| add ax, bx | ax = a \* d + b \* b | -59 | FFC5 |
| idiv cx | ax = (a \* d + b \* b)/(c \* b - a \* d) | 5 | 0005 |

4. Лістинг программи

Turbo Assembler Version 3.0 10/22/20 01:47:35 Page 1

lb3.asm

1 ;Программа для проведення розрахунків за заданим +

2 виразом відповідно до варіантів з таблиці

3 ;----------------------------------------------- сегмент даних

4 0000 data segment para public 'data'

5 0000 FFF4 a dw -12

6 0002 0007 b dw 7

7 0004 FFEF c dw -17

8 0006 0009 d dw 9

9 0008 ???? Result dw ?

10 000A data ends

11 ;----------------------------------------------- сегмент команд

12 0000 code segment para public 'code'

13 assume cs:code, ds:data, ss:stk

14 0000 start:

15 0000 B8 0000s mov ax, data ; ініціалізація сегментного регістра ds

16 0003 8E D8 mov ds, ax ; на програмний сегмент даних data

17 0005 A1 0004r mov ax, c ; ax = c

18 0008 F7 2E 0002r imul b ; ax = c \* b

19 000C 8B C8 mov cx, ax ; cx = c \* b

20 000E A1 0000r mov ax, a ; ax = a

21 0011 F7 2E 0006r imul d ; ax = a \* d

22 0015 2B C8 sub cx, ax ; cx = c \* b - a \* d

23 0017 A1 0000r mov ax, a ; ax = a

24 001A F7 2E 0006r imul d ; ax = a \* d

25 001E 8B D8 mov bx, ax ; bx = a \* d

26 0020 A1 0002r mov ax, b ; ax = b

27 0023 F7 E8 imul ax ; ax = b \* b (b^2)

28 0025 03 C3 add ax, bx ; ax = a \* d + b \* b

29 0027 99 cwd ;

30 0028 F7 F9 idiv cx ; ax = (a \* d + b \* b)/(c \* b - a \* d)

31 002A A3 0008r mov Result, ax ; Result = ax =

32 002D B8 4C00 mov ax, 4C00h

33 0030 CD 21 int 21h

34 0032 code ends

35 ;----------------------------------------------- сегмент стека

36 0000 stk segment stack

37 0000 0100\*(??) db 256 dup (?) ;поле для стека розміром 256 байт

38 0100 stk ends

39 end start

\_Turbo Assembler Version 3.0 10/22/20 01:47:35 Page 2

Symbol Table

Symbol Name Type Value

??DATE Text "10/22/20"

??FILENAME Text "lb3 "

??TIME Text "01:47:35"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text STK

@FILENAME Text LB3

@WORDSIZE Text 2

A Word DATA:0000

B Word DATA:0002

C Word DATA:0004

D Word DATA:0006

RESULT Word DATA:0008

START Near CODE:0000

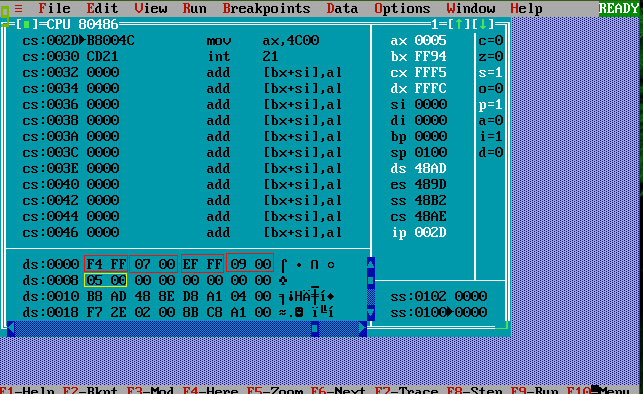
Groups & Segments Bit Size Align Combine Class

CODE 16 0032 Para Public CODE

DATA 16 000A Para Public DATA

STK 16 0100 Para Stack

\_



На скріншоті відмічені результати виконання програми. У червоних прямокутниках позначені вхідні дані, а у жовтому результат обчислень програми.

F4 FF (-12)

07 00 (7)

EF FF (-17)

09 00 (9)

05 00 (5)

**Висновок**

Під час виконання лабараторної роботи дізнався, що для множення числа необхідно переписувати дані регістру AX в інші регістри, інакше ви втратите результат обчислень. Дізнався, що в ассемблері є команди знакового і беззнакового ділення та множення. Операції додавання та віднімання мають більше операндів за якими проводяться арифметичні операції аніж множення та ділення. Існують операції інкременту да декременту, які збільшують або зменшують операнд на 1.