МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Розрахункова робота**

з дисципліни « Програмування на асемблері »

*назва дисципліни*

на тему: АРИФМЕТИЧНІ ОПЕРАЦІЇ. БІТОВІ КОМАНДИ

Виконав: студент 2 курсу групи № 622п

напряму підготовки (спеціальності)

121 інженерія программного забезпечення

(шифр і назва напряму підготовки (спеціальності))

Зайченко Я. І.

(прізвище й ініціали студента)

Прийняв: ст. викладач каф. 603

Дем’яненко В. А.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала:

Кількість балів:

Харків – 2024

**Мета роботи:** вивчити арифметичні операції мовою Асемблер та їх використання при складанні програм; вивчити бітові команди і їх застосування.

Структура звіту

1. написання коду програми 1 з ім'ям *RR1.asm.*
2. процес створення виконуваного файлу розширенням \*.EXE
3. відладка та запуск в TURBODEBUG;
4. виконання програми з п’ятьма варіантами різних початкових значень змінних A, B, C, D по кроках (табл.1). Записані результати виконання в таблиці 1. Переведення результату в десяткову систему.
5. написання коду програми 2 з ім'ям RR2.asm
6. процес створення виконуваного файлу розширенням \*.EXE
7. Виконання програми 2 згідно варіанту початкових значень регістра AL.
8. Записані результати в таблиці 2 у двійковій і десятковій системах.
9. висновок.

Виконання роботи

Створюємо новий asm файл з кодом поданим нижче:

;Програма 1

;x =(a\*2+b\*c)/(d-3)

.model small

.stack 100h

.data

a db ?

b db ?

c db ?

d db ?

x dw ?

;Резервуємо пам'ять для змінних

;A,B,C,D,X

.code start:

mov ax,@data

mov ds,ax

mov a,3

mov b,4

mov c,2

mov d,5

mov al,2

mul a

mov cx,ax

mov al,b

mul c

add ax,cx

mov cl,d

sub cl,3

div cl

mov x,ax

mov ah,4ch

int 21h

end start

Для відладки було використано емулятор emu8086, результат показано на рисунку 1.

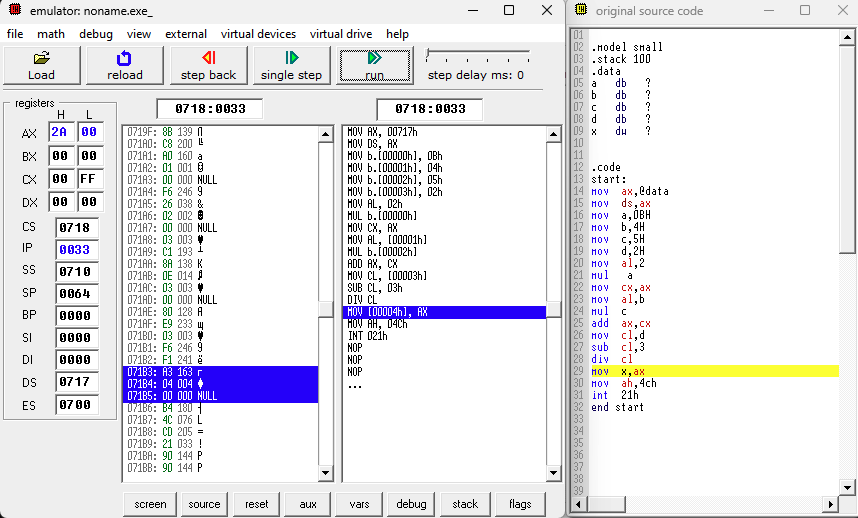


Рисунок 1 – відладка коду на асемблері

Таблиця 1 – Варіанти початкових значень змінних A, B, C, D

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | | **1** | **2** | **3** | **4** | **7** |
| Значення  змінних | A | 3 | 0AH | 30 | 60 | 0ВН |
| B | 4 | 5H | 4 | 16 | 4H |
| C | 2 | 8H | 15 | 5 | 5H |
| D | 5 | 9H | 6 | 18 | 2H |
| Частка  AL |  | 07 | 0A | 28 | 0D | 00 |
| Остача  AH |  | 00 | 00 | 00 | 05 | 2A |
| Десяткова система |  | 7 | 10 | 40 | 13,5 | 0,42 |

Створюємо новий asm файл з кодом поданим нижче:

;Програма 2

.model tiny

.stack 100h

.code

start:

mov ah, 10101100b

shr ah,1

mov ah, 10101100b

shl ah,1

mov ah, 10101100b

sar ah,1

mov ah, 10101100b

ror ah,1

mov ah, 10101100b

rol ah,1

mov ax,4c00h

int 21h

end start

Для відладки було використано емулятор emu8086, результат показано на рисунку 2.

Рисунок 2 – відладка коду на асемблері

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Команда | SHL | SHR | SAR | ROR | ROL |
| 10 | 10101100b | 56/86/  01010110 | 58/88/  01011000 | D6/214/  11010110 | 56/86/  01010110 | 59/89/  01011001 |

Висновок

Під час виконання лабораторної роботи ми ознайомились з основами роботи з процедурами у мові асемблера, вивчили різновиди адресації процедур (NEAR, FAR) та навчились використовувати локальні змінні для зберігання тимчасових даних.

Основними результатами роботи стало здобуття практичних навичок у написанні програм на асемблері з використанням процедурного підходу. Ми ретельно вивчили процес створення виконуваних файлів з допомогою асемблера та впровадили розгалуження та змінні для забезпечення більшої гнучкості та читабельності коду.

Отримані знання дозволили краще розуміти принципи адресації в асемблері, що є важливим елементом управління потоком виконання програм. Дослідження різних типів адресації процедур дало змогу зрозуміти їх відмінності та сфери застосування.

У цілому, лабораторна робота була корисною для поглиблення знань у мові асемблера та відкрила нові можливості для подальшого вивчення аспектів програмування на рівні машинного коду.