НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Системне програмування »

*Звiт з лабораторної роботи №2*

*Тема: «*Команди зсуву, тестування бітів*»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Харків – 2020

**Мета :**

* набути практичних навичок складання, налагодження та виконання програм, написаних мовою ассемблера для обчислення математичного рівняння за допомогою арифметичних команд і команд зсуву, порівняння часу їх виконання під МП платформу х64.

***Індивідуальне завдання***

Варіант 6.

* Задані числа: a, b, c, d, e, f, g, які є показниками ступеня 2. В одній програмі обчислити вираз двома способами: за допомогою арифметичних команд і команд зсуву, які замінюють арифметичні команди. Вивести через одну функцію MessageBox результати виконання і кількість тактів, які використані для їх обчислень.

Згідно номеру студента в групі вибрати варіант завдання і написати на асемблері програму обчислення одного з виразів: e/b + a/c – def

* Задані масиви А і В з N = 40 елементів. Привести програму визначення твори елементів масиву B, для яких біти 0 і 1 збігаються.

**Текст програми**

Програма 1: (LR2-1)

include \masm64\include64\masm64rt.inc ; библиотеки

;count PROTO arg\_a:QWORD,arg\_b:QWORD,arg\_c:QWORD,arg\_d:QWORD,arg\_e:QWORD, arg\_f:QWORD

.data ;

a1 dq 4 ; операнд а1 размерностью 64 разряда константа 2

b1 dq 16 ; операнд b1 размерностью 64 разряда ;змзмінна d

c1 dq 32 ; операнд c1 размерностью 64 разряда;змзмінна c

d1 dq 64 ; операнд d1 размерностью 64 разряда ;змзмінна d

e1 dq 128 ; операнд e1 размерностью 64 разряда ;змзмінна e

f1 dq 256 ; операнд f1 размерностью 64 разряда ;змзмінна f

res1 dq -2097144 ; операнд res1 размерностью 64 разряда; змінна результ

\_res dq 0

\_res1 dq 0

\_res2 dq 0

\_res3 dq 0

;Текст для MessageBox

title1 db "Лаб.2-1 Команди зсуву. masm64",0

txt1 db "Рівняння e/b + a/c – d\*e\*f",10,

"Змінні: a=%d, b=%d, c=%d, d=%i, e=%d, f=%d",10,10,

"Результат виконання арифм. команд: %d",0ah,"Число тактів: %d",0ah,"Адрес змінної в памяти: %ph",10,10,

"Результат виконання команд зсуву: %d",0ah,"Число тактів: %d",0ah,"Адрес змінної в памяти: %ph",10,10,

"Результат(Очікуваний): %d",10,"Адрес змінної в памяти: %ph",10,10,

"Автор: Гряник Г.В., гр.КІТ-119Д",0

buf1 dq 3 dup(0),0

.code;cекция кода

count proc arga:QWORD,argb:QWORD,argc:QWORD,argd:QWORD,arge:QWORD,argf:QWORD

rdtsc

xchg rdi,rax

mov rax,[rbp] ; исследование ; смотрим в x64dbg содержимое [rbp]

mov rax,[rbp+10h] ; a1;

mov rax,[rbp+18h] ; b1

mov rax,[rbp+20h] ; c1

mov rax,[rbp+28h] ; d1

mov rax,[rbp+30h] ; e1

mov rax,[rbp+38h] ; f1

mov r10,[rbp+18h] ;занести в r10 b1

mov rax,[rbp+30h] ; e1

mov rdx,0 ; обнулить рігіст rdx

div r10 ; e/d

mov \_res,rax; ; save result

mov rax, rcx ; занести а в rax

div r8 ; поділити на с

add \_res,rax ; e/b + a/c

mov rax,r9 ; занести с v rax

mov r8,[rbp+30h] ; e1

mov r9,[rbp+38h] ; f1

mul r8 ; d\*e

mul r9 ; d\*e\*f

sub \_res,rax ; e/b + a/c – d\*e\*f

rdtsc ; получение числа тактов

sub rax,rdi ; вычитание из последнего числа тактов предыдущего числа

mov \_res1,rax

ret

count endp;;;програма 1 кінець

count2 proc arg\_a:QWORD,arg\_b:QWORD,arg\_c:QWORD,arg\_d:QWORD,arg\_e:QWORD,arg\_f:QWORD;;;програма 2

rdtsc

xchg rdi,rax

mov rax ,[rbp+30h] ; занести в r10 rax

sar rax,4 ; ; e/d

mov \_res2,rax ; збереження результатів

sar rcx,5 ;a/c

add \_res2,rcx ;e/d+a/c

shl r9,7 ;d\*e

shl r9,8 ;d\*e\*f

sub \_res2,r9 ;e/d+a/c-d\*e\*f

rdtsc

sub rax, rdi ;получение числа тактов

mov \_res3,rax ; число тактів

ret

count2 endp;;програма 2 кінець

entry\_point proc

mov rax,e1

invoke count,a1,b1,c1,d1,e1,f1 ;виконання арифм. команд

invoke count2,a1,b1,c1,d1,e1,f1;виконання команд зсуву

;Створення MessageBox

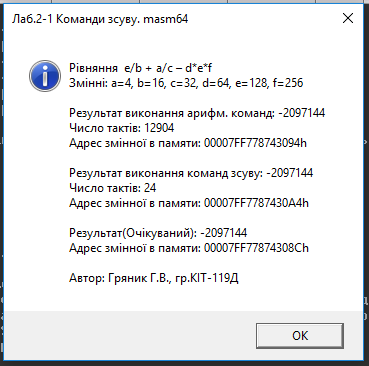
invoke wsprintf,ADDR buf1,ADDR txt1,a1,b1,c1,d1,e1,f1,\_res, \_res1,ADDR \_res,\_res2,\_res3,ADDR \_res2,res1,ADDR res1

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION

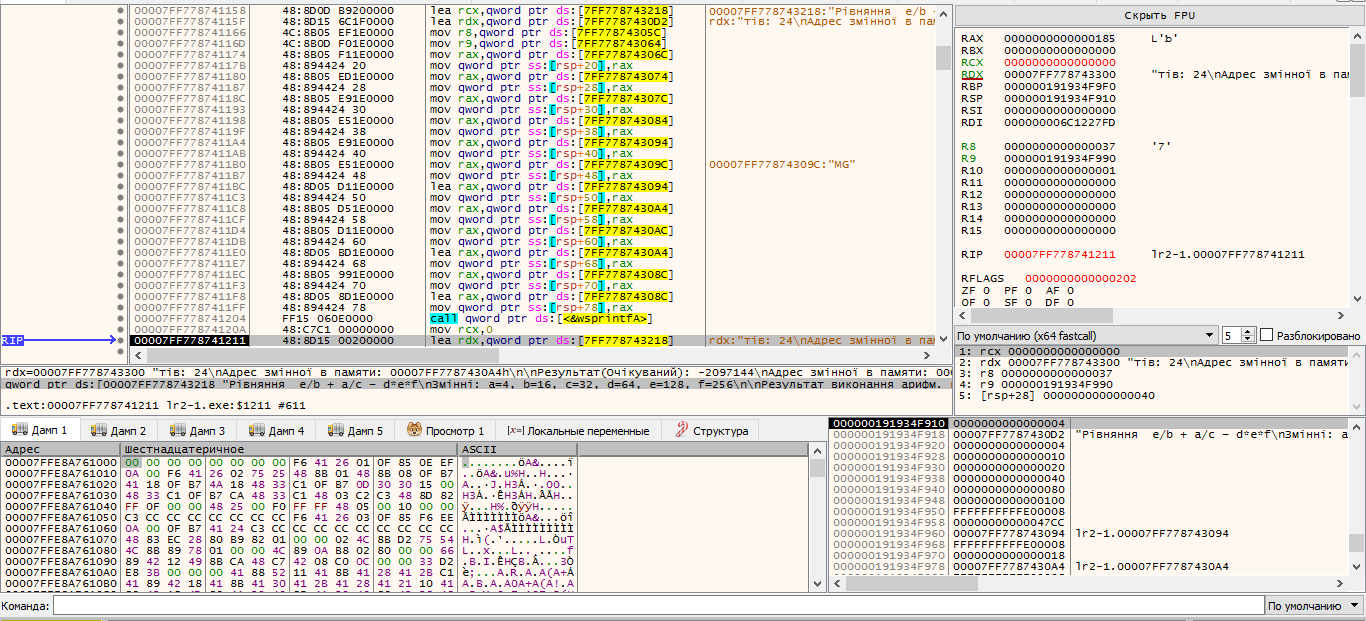
invoke ExitProcess,0

entry\_point endp; кінець роботи програми

end



**Рис 1 – результат виконання роботи програми** LR2-1



**Рис 2 – результат налагодження програми** LR2-1

Алгоритм: Прорама в з мінні заносить данні після чого ппрограма заходе в процедуру та розвязує приклад за допомогою алгебраїчних операції. Процедура поступово виконує математичні операції щоб отримати результат.

Для отримання числа тіків процесора на початку заміряється кількість тіків , а потім в кінці процедури виконується математична операція віднімання від кількості тіків віднімають кількість на початку і отримують результат.

По закінченню першої процедури програма переходить в другу процедуру де вирішення такого прикладу робиться за допомогою команд зсуву. Алгоритм подібен до алгебраїчних команд за винятком того ,що на місцях ділення або множення виконуються команди зсуву sar,shl. Алгоритм фіксування тіків ідентичний раніше описаному.

Висновок: програма яка робить обчислення за допомогою команд зсуву працює набагато швидше ніж з алгебраїчними командами. При роботів з командами зсуву треба передбачати всі можливі оптимальні вирішення задачі наприклад ділення від’ємних чисел абощо.

**Програма 2:**

include \masm64\include64\masm64rt.inc ; библиотеки

err1 PROTO arg\_a:QWORD

.data ;

mas1 dq 12,4,5,3,8

dq 10 dup(1)

dq 25 dup(5) ; массив mas1

mas2 dq 10,12,10,34,17

dq 20 dup(1)

dq 15 dup(2) ; массив mas2

buf dq ?,0 ;

len1 equ ($-mas1)/type mas2 ; вычисление количества слов в mas1

\_res dq 1 ; операнд res1 размерностью 64 разряда; змінна результ

\_res1 dq 0 ; операнд res1 размерностью 64 разряда; змінна результ

;Текст для MessageBox

title1 db "Лаб.2-2 Тест битов. masm64",0

txt1 db "Завдання:Задані масиви А і В з N = 50 елементів.",10," Привести програму визначення добутку елементів масиву B ,",10," для яких біти 0 і 1 збігаються.", 10,10,

"Кількість елементів : %d",10,"Адрес змінної в памяти: %ph",10,10,

"Добуток елементів : %d",10,"Адрес змінної в памяти: %ph",10,10,

"Автор: Гряник Г.В., гр.КІТ-119Д",0

buf1 dq 3 dup(0),0

.code;cекция кода

entry\_point proc

invoke err1,len1

;mov rdx,\_res

;Створення MessageBox

invoke wsprintf,ADDR buf1,ADDR txt1,\_res1,ADDR \_res1,\_res,ADDR \_res

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION

invoke ExitProcess,0

entry\_point endp; кінець роботи програми

mov r14,\_res

err1 proc arg\_a:QWORD; arg\_a передается в rcx

lea rsi,mas2 ; начальный адрес массива mas1

m1:

mov ax,[rsi] ; в ах заносится элемент массива

bt rax,0 ; выбор нулевого бита

setc bh ; если cf=1, то установление 1 в bh

bt rax,1 ; выбор первого бита

setc bl ; если pf = 1, то установление 1 в bl

cmp bh,bl ; сравнение битов

jne m2 ; если не равняется, то перейти на m2

mov \_res,rax;збереження результату

add \_res1,1;кількість елементів

mov r14,\_res

m2: add rsi,8 ; увеличение адреса mas2 для выборки нового числа

dec rcx ; уменьшение счетчика чисел в массиве mas1

cmp rcx,0

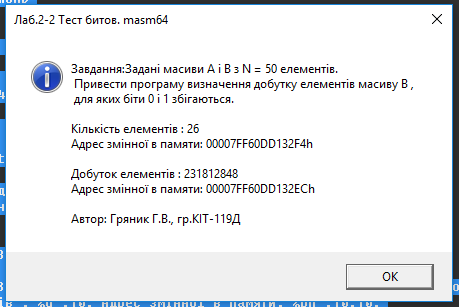
jnz m1 ; перейти на метку m1, если не нуль

ret

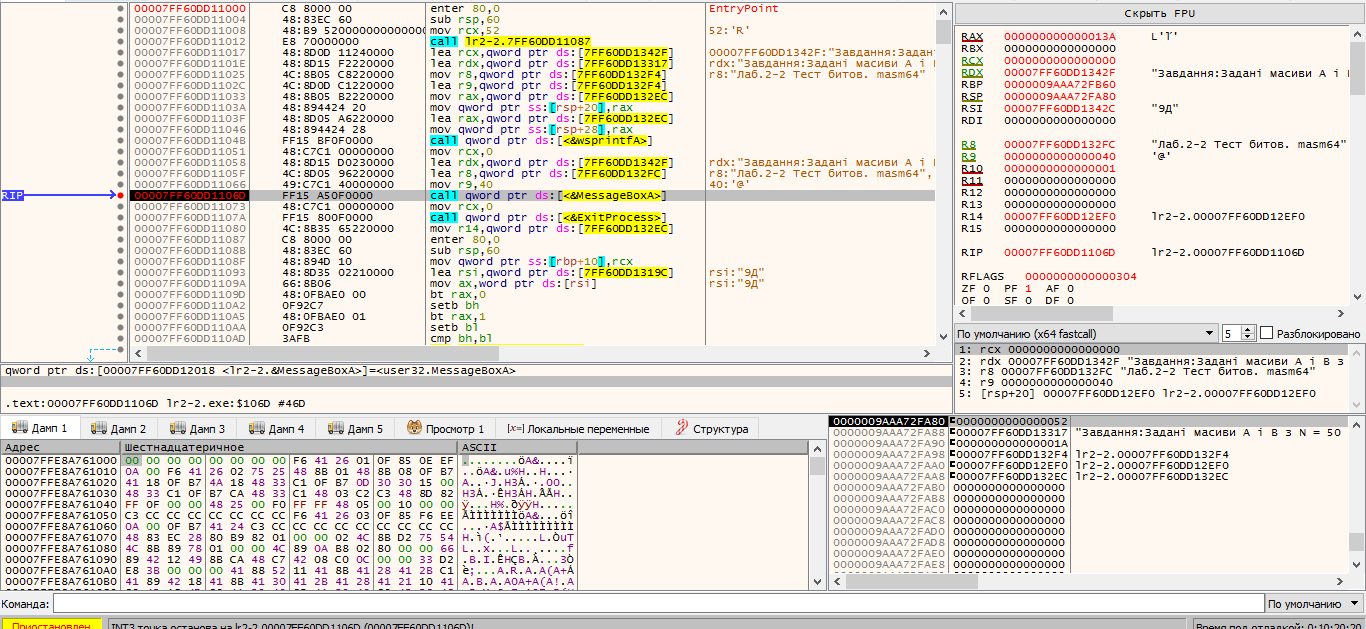
err1 endp

end

**Результат**



**Рис 3 – результат виконання роботи програми** LR2-2



**Рис 4 – результат налагодження програми** LR2-2

Алгоритм: Прорама в з мінні заносить данні. Програма викликає процедуру. В якій в циклі заноситься в регістри нульовий та перший біт масиву після чого ці два регістрва порівнюється і якщо вони однакові то +1 до лічильник та цей елемент множиться на результат. Цикл працює доти доки не закінчиться масив. По завершенню програма виведе результати роботи. В результаті ми отримаємо кількість елементів в яких перший та нульовий біт співпадають та добуток цих елементів.

Висновок: доступ до бітві елементу дає можливість перевірити правильність роботи програми.

* **Висновок:** Під час лабораторної роботи було набуто практичних навичок з роботи mas64, а саме, складання, налагодження та виконання програм, написаних мовою ассемблера для обчислення математичного рівняння за допомогою арифметичних команд і команд зсуву, порівняння часу їх виконання під МП платформу х64.