МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

#### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №3

«Использование математического сопроцессора»

Вариант №14

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИНБс-3301-01 | \_\_\_\_\_\_\_\_\_\_\_\_ | К.М.Огородникова |
| Проверил: доцент кафедры РЭС | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М. А. Земцов |

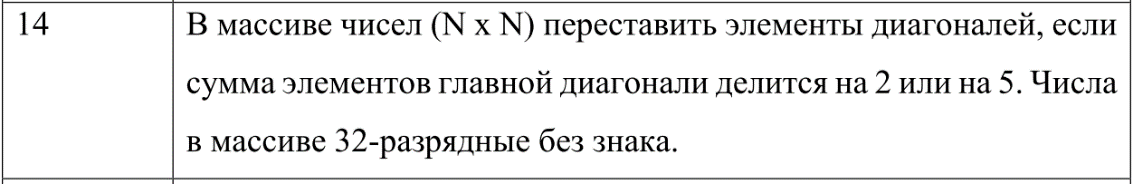
#### Киров 2025

**Цель работы:** изучение принципов использование математического сопроцессора микропроцессоров с архитектурой x86.

**Ход работы:**

**1**. Исходные данные представлены в таблице 1.

Таблица 1 – Исходные данные



**2**. Написание программы

Был написан код для ассемблера (masm), который приведён в п.3.

3. Анализ работы программы при разных значениях.

Результат работы программы тестировался при разных значениях массива размером 4 х 4, значения которого для первого случая приведены на рисунке 1.

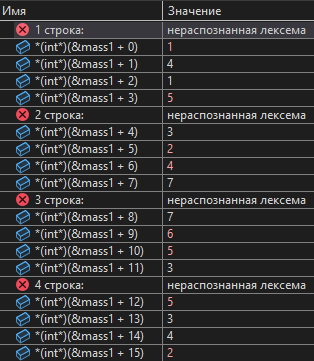


Рисунок 1 — Исходные значения массива в первом случае

Сумма элементов данного массива по главной диагонали равна: 1+2+5+2=10, что подходит для условия «делится на 2 или 5». По коду программы значение переменной «delimost» будет равно 1 в таком случае; если условие не выполнится, то значение будет равно 2.

Результат вычисления суммы и проверка делимости приведены на рисунке 2.



Рисунок 2 — Полученные значения

Сумма была рассчитана верно, делимость тоже. В результате успешной проверки условия были переставлены элементы диагоналей, что приведено на рисунке 3.

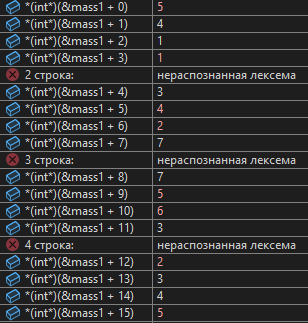


Рисунок 3 — Изменённый массив

Изменённый массив действительно имеет в себе переставленные элементы диагоналей. Красным цветом выделены те элементы, значения которых были изменены.

Далее был проверен случай значений массива, при котором условие не выполняется. Новые значения массива приведены на рисунке 4.

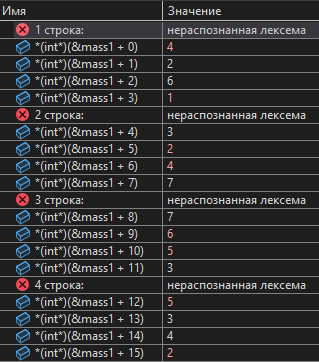


Рисунок 4 — Исходные значения нового массива

Сумма элементов главной диагонали данного массива равна: 4+2+5+2=13. При таком случае условие не выполняется.

Результат расчёта параметров при таком случае приведён на рисунке 5.



Рисунок 5 — Полученные значения

Полученные значения совпали с ожидаемыми. При этом значения массива не были изменены, так как условие не выполняется.

***Вывод:*** в ходе лабораторной работы были получены навыки работы с математическим сопроцессором при написании кода для ассемблера.

**3. Код программы:**

.686

.model flat, stdcall

.stack 100h

.data

mass1 dd 4, 2, 6, 1, 3, 2, 4, 7, 7, 6, 5, 3, 5, 3, 4, 2 ; массив N x N 32-битных

N dw 4 ; N

delimost dw ?

summa dd ? ; сумма элементов главной диагонали

deliteli dw 2, 5 ; массив чисел, проверить сумму на делимость которых надо проверить

buffer dd 4 ; переменная для промежуточных данных

buffer2 dd 4 ; переменная для промежуточных данных

i dw 0

j dw 0

.code

ExitProcess PROTO STDCALL :DWORD

Start: ; start code

finit ; инициализация мат сопроцессора

xor eax, eax ; зануление всех основных регистров

xor ebx, ebx

xor ecx, ecx

xor edx, edx

xor esi, esi

call summ ; вызов функции подсчёта суммы элементов диагонали

call proverka ; вызов функции проверки условия

; если условие выполняется, то delimost=1, иначе delimist=2

; проверка, что delimost=1; если не так, то переход к точке endd

cmp delimost, 1

jne endd

call smena ; функция для перестановки элементов диагоналей

endd:

ret

;------------------------------ конец "main" функции ---------------------------------------------

summ PROC ; функция для нахождения суммы элементов главной диагонали

xor ecx, ecx ; зануление регистра-счётчика количества итераций

xor esi, esi ; зануление регистра-счётчика итераций (типа "i" в С++)

mov cx, N ; задание количество итераций цикла (N раз)

mov ax, N

mov buffer, 4

inc eax

mul buffer

mov buffer, eax

fldz ; загрузка 0 в st(0)

loop\_1:

fild dword ptr [mass1 + esi] ; загрузка текущего элемента диагонали в st(0)

fadd st(1), st(0) ; результат суммы будет в st(1)

fxch st(1) ; меняем st(1) с st(0)

add esi, buffer ; типа i++ в С++, только +4\*(N+1), т.к. элементы массива по 4 байта, а не по 1 байту, и шагаем по диагонали

loop loop\_1

fist dword ptr [summa] ; выгрузка суммы в summa

ret

summ ENDP

proverka PROC ; функция для проверки условия

mov cx, 2

mov si, 0

; mov ukazatel, offset deliteli

loop\_2:

finit

fild dword ptr [summa]

mov buffer, 2

fild word ptr [deliteli + si]

; сейчас: st(0) = 2, st(1) = summa

;loop\_2:

; mov cx, 2 ; делаем цикл бесконечным

fdiv st(1), st(0) ; деление st(1) на st(0)

mov buffer, 0

fld st(1) ; добавление st(1) в вершину стека

frndint ; округление st(0) до ближайшего целого

; сейчас: st(0) = округлённый остаток, st(2) = изначальный остаток

fcom st(2) ; сравнение st(0) с st(2)

xor eax, eax

fstsw ax ; выгрузка регистра состояния FPU в AX

sahf ; выгрузка AX во флаги процессора

je delitsa ; прыжок на точку "delitsa", если число делится на 2, т.е. если округлённый и изначальный остатки равны

mov delimost, 2

add si, 2 ; +2, т.к. массив элементов по 2 байта (dw)

loop loop\_2

ret

delitsa:

mov delimost, 1

ret

proverka ENDP

smena PROC

mov cx, N

mov si, 0

; i -- шаг в байтах для главной диагонали, его расчёт:

mov i, 4

mov ax, N

inc eax

mul i

mov i, ax

; j -- шаг в байтах для побочной диагонали, его расчёт:

mov j, 4

mov ax, N

dec eax

mul j

mov j, ax

loop\_3: ; цикл для замены элементов диагоналей для каждой строки

; запись элемента гл. диаг. в buffer:

mov ax, si

mul i

mov ebx, [mass1 + ax]

mov buffer, ebx

; запись элемента поб. диаг. в buffer2:

mov ax, si

mul j

add ax, j

mov ebx, [mass1 + ax]

mov buffer2, ebx

; запись buffer на место элемента поб. диаг.:

mov ebx, buffer

mov [mass1 + ax], ebx

; запись buffer2 на место элемента гл. диаг.:

mov ax, si

mul i

mov ebx, buffer2

mov [mass1 + ax], ebx

add si, 1

loop loop\_3

ret

smena ENDP

exit:

Invoke ExitProcess,1

End Start