

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Кафедра ИИТ

Отчёт
о лабораторной работе №4
по дисциплине «Компьютерные системы и сети»

Тема: «Архитектура и программирование сопроцессора. Использование
целочисленных команд»

Выполнил студент 2 курса
группы ПО-11 Сымоник И.А.
Номер зачетной книжки: 220220

Проверил: Савицкий Ю.В.

Цель работы: изучение архитектуры и программной модели сопроцессора; практическая работа с системой целочисленных команд.

Вариант 6

Ход работы

Задание:

1. Изучить следующие теоретические сведения: архитектура и программная модель сопроцессора – регистры и их назначение; форматы чисел, типовые команды, целочисленные команды (используя материалы лекций и методические указания).

2. Составить и отладить программу на языке ассемблера для вычисления значения функции, используя регистры сопроцессора и целочисленные команды согласно варианту, выданному преподавателем (таблица 2.2). Оформить ее в виде отдельной функции. Каждую ветку алгоритма оформить в виде отдельной внутренней процедуры. Ввод и вывод данных осуществляется через консоль. Программу протестировать по всем условиям.

<p>Вариант 6</p> $y = \begin{cases} 5x + 2xy + 1, & x + y > 9; \\ \frac{2xy + 3}{y^2 + 4}, & x + y < -5; \\ 3x^2 - 2y^2 + 1, & -5 \leq x + y \leq 9. \end{cases}$

Исходный код:

```
.386
.model flat, stdcall
option casemap:none

include \masm32\include\windows.inc
include \masm32\include\masm32.inc
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
includelib \masm32\lib\kernel32.lib
```

```
include lib \masm32\lib\user32.lib
include lib \masm32\lib\masm32.lib
```

```
.data
```

```
prompt1 db "Enter x: ", 0
prompt2 db "Enter y: ", 0
resultPrompt db "Result: ", 0
```

```
buffer1 db 11 dup(?)
buffer2 db 11 dup(?)
result dword ?
```

```
x dword ?
y dword ?
```

```
nine dw 9
five dw 5
four dw 4
three dw 3
two dw 2
one dw 1
zero dw 0
mOne dw -1
mFive dw -5
```

```
.code
```

```
start:
```

```
    ; Ввод первого числа
    invoke StdOut, addr prompt1
    invoke StdIn, addr buffer1, 11
```

```
    ; Преобразование строки в число
    invoke atodw, addr buffer1
    mov dword ptr [x], eax
    cmp byte ptr [buffer1], '-'
    je negativeX
```

```
cont_1:
    fild x
```

```
    ; Ввод второго числа
    invoke StdOut, addr prompt2
    invoke StdIn, addr buffer2, 11
```

```
    ; Преобразование строки в число
```

```

invoke atodw, addr buffer2
mov dword ptr [y], eax
cmp byte ptr [buffer2], '-'
je negativeY

cont_2:

fiadd y
fistp dword ptr [result]

; Если  $x + y < -5$ 
fild result
ficompl mFive
fstsw ax ;сохранение swr в регистре ax
sahf
jc branch_two_

; Если  $x + y > 9$ 
fild result
ficompl nine
fstsw ax ;сохранение swr в регистре ax
sahf
jnc branch_one_

; Если  $-5 \leq x + y \leq 9$ 
call branch_three

```

```

end_:
; Конвертируем число в строку
invoke dwtoa, result, addr buffer1
; Вывод результата
invoke StdOut, addr resultPrompt
invoke StdOut, addr buffer1

; Выход из программы
invoke ExitProcess, 0

```

```

branch_one proc

```

```

    fild five
    fimul x
    fistp dword ptr [result]

```

```
    fild two
    fimul x
    fimul y
    fiadd result
    fiadd one
    fistp dword ptr [result]
    ret
branch_one endp
```

```
branch_two proc
```

```
    fild y
    fimul y
    fiadd four

    fistp dword ptr [result]

    fild two
    fimul x
    fimul y
    fiadd three

    fidiv result

    fistp dword ptr [result]
    ret
branch_two endp
```

```
branch_three proc
```

```
    fild three
    fimul x
    fimul x

    fistp dword ptr [result]

    fild two
    fimul y
    fimul y
    fimul mOne

    fiadd result
    fiadd one

    fistp dword ptr [result]
    ret
```

```
branch_three endp
```

```
negativeX:
```

```
    mov eax, dword ptr [x]
```

```
    not eax
```

;инвертирование всех битов

```
    add  eax, 2531
```

; прибавление 2531, т.к в atodw допущена

ошибка

```
    mov dword ptr [x], eax
```

```
    jmp cont_1
```

```
negativeY:
```

```
    mov eax, dword ptr [y]
```

```
    not eax
```

;инвертирование всех битов

```
    add  eax, 2531
```

; прибавление 2531, т.к в atodw допущена

ошибка

```
    mov dword ptr [y], eax
```

```
    jmp cont_2
```

```
branch_one_:
```

```
    call branch_one
```

```
    jmp end_
```

```
branch_two_:
```

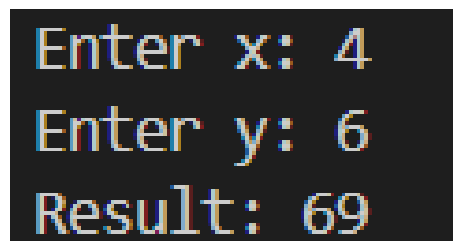
```
    call branch_two
```

```
    jmp end_
```

```
end start
```

Результат выполнения:

X+Y > 9



```
Enter x: 4
Enter y: 6
Result: 69
```

$$X+Y < -5$$

```
Enter x: -4  
Enter y: -6  
Result: 1
```

$$-5 \leq X+Y \leq 9$$

```
Enter x: 2  
Enter y: 3  
Result: -5
```

Вывод: изучили архитектуру и программную модель сопроцессора;
получили навыки практической работы с системой целочисленных команд.