

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Организация ЭВМ и систем»
ТЕМА: Представление и обработка целых чисел. Организация
ветвящихся процессов

Студентка гр. 9382

Бочаров Г.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить арифметические команды ассемблера, разработать программу, вычисляющую необходимые переменные и углубить свои знания, в процессе написания программы.

Задание:

2 Вариант - 1.3.2

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a , b , i , k вычисляет: а) значения функций $i1 = f1(a,b,i)$ и $i2 = f2(a,b,i)$; б) значения результирующей функции $res = f3(i1,i2,k)$, где вид функций $f1$ и $f2$ определяется из табл. 2, а функции $f3$ - из табл.3 по цифрам шифра индивидуального задания ($n1,n2,n3$), приведенным в табл.4. Значения a , b , i , k являются исходными данными, которые должны выбираться студентом самостоятельно и задаваться в процессе исполнения программы в режиме отладки. При этом следует рассмотреть всевозможные комбинации параметров a , b и k , позволяющие проверить различные маршруты выполнения программы, а также различные знаки параметров a и b .

$$/15-2*i, \text{ при } a>b$$

$$f1 = <$$

$$\backslash 3*i+4, \text{ при } a\leq b$$

$// 7 - 4*i$, при $a > b$

$f3 = <$

$\backslash 8 - 6*i$, при $a \leq b$

$/ \max(i1, 10-i2)$, при $k < 0$

$f2 = <$

$\backslash |i1 - i2|$, при $k \geq 0$

Ход работы

В ходе выполнения работы были использованы следующие команды арифметических операций:

Add – выполняет арифметическое сложение приемника и источника и помещает сумму в приемник.

Sub – вычитает источник из приемника и помещает разность в приемник.

Cmp - сравнивает приемник и источник и устанавливает флаги.

Neg - выполняет над числом, содержащимся в приемнике, операцию дополнения до двух.

В ходе выполнения работы были использованы следующие сдвиговые команды:

Sal – выполняет арифметический сдвиг влево.

В ходе выполнения работы были использованы следующие команды передачи управления:

Jg - переход, если больше ($ZF = 0$ и $SF = OF$).

Jz - переход, если 0 ($ZF = 1$).

Jl - переход, если меньше ($SF \neq OF$).

Jmp - передает управление в другую точку программы.

Тестирование.

Вводные данные	Результат
a = 3	i1 = 22
b = 5	i2 = -28
i = 6	res = 50
k = 1	
a = 2	i1 = 13
b = 1	i2 = 3

i = 1 k = 1	res = 10
a = 1 b = 1 i = 1 k = -1	i1 = 7 i2 = 2 res = 8
a = 2 b = 1 i = 1 k = -1	i1 = 13 i2 = 3 res = 13

Выводы.

В результате выполнения лабораторной работы был разработан код для подсчитывания определенных выражений, оптимизировано умножение. Были улучшены навыки письма в ассемблере.

Приложение.

Текст файла 1.ASM

```
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
```

DATA SEGMENT

a DW 2
b DW 1
i DW 1
k DW -1
i1 DW 0
i2 DW 0
res DW 0

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

Main PROC FAR

push ds
sub ax,ax
push ax
mov ax, data
mov ds, ax

f1:
mov ax, a
cmp ax, b
jle f1_1 ; a <= b
jmp f1_0 ; a > b

f1_0:
mov ax, i
add ax, ax; i*2
mov bx, 15; 15
xchg bx, ax
sub ax, bx

mov i1, ax
jmp f2_0

f1_1:

mov ax, i
shl ax, 1; i*3
add ax, i
add ax, 4

mov i1, ax

jmp f2_1

```

f2_0:
    mov ax, i1
    shl ax, 1; 30 - 4i

    sub ax, 23 ; 7 - 4i

    mov i2, ax

    jmp f3

```

```

f2_1:
    mov ax, i1
    shl ax, 1; 6i+8
    neg ax
    add ax, 16

    mov i2, ax

    jmp f3

```

```

f3:
    cmp k, 0
    jge f3_0 ; k >= 0
    jmp f3_1 ; k < 0

```

```

f3_1:

    mov ax, i2

    sub ax, 10

    neg ax

    cmp i1, ax;
    jge res1 ; >=

    jmp endLL

```

```

    res1:
    mov ax, i1

```

```

    jmp endLL

```

```

f3_0:
    mov ax, i1;
    sub ax, i2

    cmp ax, 0
    jl f3_0_abs

```

```
        jmp endLL

f3_0_abs:

        neg ax

        jmp endLL

endLL:

        mov res, ax
        mov ah, 4ch
        int 21h

Main ENDP
CODE ENDS
        END Main
```