

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 системы»
Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов

Студента гр. 1383

Самулевич С.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучение алгоритма построения оператора ветвления на языке ассемблера и методов взаимодействия с целыми числами и их обработки.

Задание.

(f3, f6, f4)

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

Выполнение работы.

- 1) формула $7 - 4i$ не нуждается в оптимизации
- 2) формулу $8 - 6i$ можно представить в виде $8 - 2((2+1)i)$
- 3) формулу $2(i+1) - 4$ можно представить в виде $2i-2$
- 4) формулу $5-3(i+1)$ можно представить в виде $2-3i$

Так же для уменьшения количества кода заранее считается $2i$.

Умножение на 2 реализовано командой SAL, умножение на 3 – есть сложение $2x+x$. Ветвление реализовано с помощью команд `jl` и `jmp`

Тесты:

Номер	Входные данные	Результат
1	$a = 5$ $b = 4$	$i1 = -21$ $i2 = 12$

	k = -2 i = 7	i3 = 2
2	a = 2 b = 1 k = 1 i = 2	i1 = -1 i2 = 2 i3 = -2
3	a = 1 b = 2 k = 1 i = 2	i1 = -4 i2 = -4 i3 = 4
4	a = -1 b = 1 k = 0 i = 1	i1 = 2 i2 = -1 i3 = 1
5	a = 1 b = 1 k = 1 i = -1	i1 = 14 i2 = 5 i3 = -5

Выводы.

В ходе работы были изучены ветвления и обработка с целых чисел на языке ассемблер.

Исходный код

EOFLine EQU '\$'

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

DATA SEGMENT

a DW 1h

b DW 1h

i DW -1h

k DW 1h

i1 DW ?

i2 DW ?

i3 DW ?

x DW ?

HELLO DB 'Assembler is the best!', 0AH, 0DH,EOFLine

DATA ENDS

AStack SEGMENT STACK

DW 12 DUP('0')

AStack ENDS

CODE SEGMENT

Main PROC FAR

push DS

sub AX,AX

push AX

mov AX, DATA

mov DS, AX

mov BX, a

mov CL, 1

mov AX, i

sal AX, CL

```
cmp BX, b
jle Less
```

```
sub AX, 2
mov i2, AX
```

```
add AX, 2
sal AX, CL
neg AX
add AX, 7
mov i1, AX
```

```
jmp Eif
```

```
Less:
add AX, i
mov x, AX
```

```
sal AX, CL
neg AX
add AX, 8
mov i1, AX
```

```
mov AX, x
neg AX
add AX, 2
mov i2, AX
```

```
Eif:
```

```
mov BX, k
```

```
cmp BX, 0
jl Less1
```

```
mov AX, i2  
neg AX
```

```
cmp AX, -6  
jl Less2  
mov i3, AX  
jmp Eif2
```

```
Less2:  
mov i3, -6
```

```
Eif2:
```

```
jmp Eif1
```

```
Less1:  
mov AX, i1  
sub AX, i2
```

```
cmp AX, 0  
jl Less3
```

```
jmp Eif3
```

```
Less3:  
neg AX
```

```
Eif3:
```

```
cmp AX, 2  
jl Less4  
mov i3, 2  
jmp Eif4
```

```
Less4:
```

mov i3, AX

Eif4:

Eif1:

mov DX, OFFSET HELLO

mov AH,9

int 21h

ret

Main ENDP

CODE ENDS

END Main