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. 0383 _____ Ханина М.И.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет: a) значения функций i1 = f1(a,b,i) и i2 = f2(a,b,i); b) 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.

Замечания:

1) при разработке программы нельзя использовать фрагменты, представленные на ЯВУ, в частности, для ввода-вывода данных. Исходные данные должны вводиться, а результаты контролироваться в режиме отладки;

2) при вычислении функций f1 и f2 вместо операции умножения следует использовать арифметический сдвиг и, возможно, сложение;

3) при вычислении функций f1 и f2 нельзя использовать процедуры;

4) при разработке программы следует минимизировать длину кода, для чего, если надо, следует преобразовать исходные выражения для вычисления функций.

Вариант 18:

$$i1 = f3 = \begin{cases} 7 - 4*i, & \text{при } a>b \\ 8 - 6*i, & \text{при } a<=b \end{cases}$$

$$i2 = f8 = \begin{cases} -(6*i+8), & \text{при } a>b \\ 9 -3*(i-1), & \text{при } a<=b \end{cases}$$

$$res = f6 = \begin{cases} |i1 - i2|, & \text{при } k<0 \\ \max(7, |i2|), & \text{при } k>=0 \end{cases}$$

Выполнение работы.

Числа для работы программы вводятся сразу в asm файл. Для реализации алгоритмов использовались команда сравнения cmp и различные условные переходы. Для функций f3 и f8 условия одинаковы, поэтому их вычисление проходит в одном блоке. Сначала командой cmp сверяются значения a и b. С помощью команды jle проверяется, что a <= b, и в зависимости от результата программа переходит к блоку, где рассчитываются соответствующие значения f3 и f8. Для операций умножения использовался битовый сдвиг влево(команда shl) и сложение (команда add).

Результаты тестирования представлены в табл. 1.

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

Таблица 1. Проверка работы программы.

№	Входные данные	Значение i1	Значение i2	Значение res	Комментарий
1	a = 2, b = 3, i = 1, k = 4	2	9	9	Программа работает корректно
2	a = 1, b = 4, i = 2, k = 0	-4	6	7	Программа работает корректно
3	a = 6, b = 4, i = 3, k = -2	-5	-26	21	Программа работает корректно
4	a = 10, b = -3, i = 0, k = -10	3	-8	11	Программа работает корректно

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа с целыми числами и условными переходами на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: **lr3.asm**

```
; Стек  программы
AStack SEGMENT  STACK
        DW 12 DUP(?)
AStack  ENDS
;Данные программы
DATA        SEGMENT
;Директивы описания данных
a       DW    6
b       DW    4
i       DW    3
k       DW    -2
i1      DW    0
i2      DW    0

DATA        ENDS

; Код программы
CODE        SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main        PROC  FAR
        push  DS
        sub   AX,AX
        push  AX
```

```
  mov   AX,DATA
  mov   DS,AX
mov   CX, 0

;вычисление f3
mov cx, i
mov ax, cx
shl cx, 1
shl cx, 1 ; 4i
cmp a, bx
jle f3second
  mov ax, cx
   mov cx, 7
   sub cx, ax
   jmp f3final
f3second:
  add cx, ax
   add cx, ax
   mov ax, cx
   mov cx, 8
   sub cx, ax
f3final:
mov i1, cx

;вычисление f8
mov cx, i
mov ax, cx
mov bx, b
cmp a, bx
```

```
jle f8second
    shl cx, 1
    shl cx, 1 ; 4i
    add cx, ax
    add cx, ax
    add cx, 8
    neg cx
    jmp f8final
f8second:
    add cx, -1
    mov ax, cx
    shl cx, 1
    shl cx, 1
    sub cx, ax
    neg cx
    add cx, 9
f8final:
mov i1, cx

mov cx, i2
cmp cx, 0
jge skip2
   neg cx
    mov ax, cx
skip2:

;рассчет f6
mov bx, k
cmp bx, 0
```

```
    jl f6Second
      mov bx, ax
       cmp bx, 7
       jl max1
         mov cx, bx          ; |i2| >= 7
         jmp MainFinal
       max1:
         mov cx, 7           ; |i2| < 7
         jmp MainFinal
      f6Second:
      mov cx, i1
       sub cx, i2
       cmp cx, 0
       jge MainFinal
              neg cx
              jmp MainFinal
    MainFinal:
      ret
Main        ENDP
CODE        ENDS
END Main
```

ПРИЛОЖЕНИЕ Б

ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: **lr3.lst**

```
Microsoft (R) Macro Assembler Version 5.10                10/24/21 01:46:1
                                                          Page   1-1


                        ; Стек  программы
0000                    AStack SEGMENT  STACK
0000  000C[                     DW 12 DUP(?)
        ????
                    ]

0018                    AStack  ENDS
                        ;Данные программы
0000                    DATA        SEGMENT
                        ;Директивы описания данных
0000  0006              a       DW   6
0002  0004              b       DW   4
0004  0003              i       DW   3
0006  FFFE              k       DW   -2
0008  0000              i1      DW   0
000A  0000              i2      DW   0

000C                    DATA        ENDS

                        ; Код программы
0000                    CODE        SEGMENT
                            ASSUME CS:CODE, DS:DATA, SS:AStack
```

```
                        ; Головная процедура
0000                    Main        PROC  FAR
0000  1E                       push  DS
0001  2B C0                          sub   AX,AX
0003  50                       push  AX
0004  B8 ---- R                mov   AX,DATA
0007  8E D8                          mov   DS,AX
0009  B9 0000                          mov   CX, 0

                                ;вычисление f3
000C  8B 0E 0004 R                    mov cx, i
0010  8B C1                           mov ax, cx
0012  D1 E1                           shl cx, 1
0014  D1 E1                           shl cx, 1 ; 4i
0016  39 1E 0000 R                    cmp a, bx
001A  7E 0A                           jle f3second
001C  8B C1                                 mov ax, cx
001E  B9 0007                                mov cx, 7
0021  2B C8                                  sub cx, ax
0023  EB 0C 90                               jmp f3final
0026                                f3second:
0026  03 C8                                 add cx, ax
0028  03 C8                                  add cx, ax
002A  8B C1                                  mov ax, cx
002C  B9 0008                                mov cx, 8
002F  2B C8                                  sub cx, ax
0031                                f3final:
0031  89 0E 0008 R                    mov i1, cx
```

```
                                ;вычисление f8
0035  8B 0E 0004 R                mov cx, i
Microsoft (R) Macro Assembler Version 5.10                  10/24/21 01:46:1
                                                            Page  1-2


0039  8B C1                       mov ax, cx
003B  8B 1E 0002 R                mov bx, b ;
003F  39 1E 0000 R                cmp a, bx  ; сравнение a и b
0043  7E 10                       jle f8second
0045  D1 E1                           shl cx, 1
0047  D1 E1                           shl cx, 1 ; 4i
0049  03 C8                           add cx, ax
004B  03 C8                           add cx, ax
004D  83 C1 08                        add cx, 8
0050  F7 D9                           neg cx
0052  EB 11 90                       jmp f8final
0055                          f8second:
0055  83 C1 FF                        add cx, -1
0058  8B C1                           mov ax, cx
005A  D1 E1                           shl cx, 1
005C  D1 E1                           shl cx, 1
005E  2B C8                           sub cx, ax
0060  F7 D9                           neg cx
0062  83 C1 09                        add cx, 9
0065                          f8final:
0065  89 0E 0008 R                mov i1, cx
```

```
0069  8B 0E 000A R              mov cx, i2
006D  83 F9 00                  cmp cx, 0
0070  7D 04                    jge skip2
0072  F7 D9                        neg cx
0074  8B C1                       mov ax, cx
0076                        skip2:

                            ;рассчет f6
0076  8B 1E 0006 R              mov bx, k
007A  83 FB 00                  cmp bx, 0
007D  7C 12                    jl f6Second
007F  8B D8                        mov bx, ax
0081  83 FB 07                     cmp bx, 7
0084  7C 05                        jl max1
0086  8B CB                          mov cx, bx        ; |i2| >= 7
0088  EB 19 90                       jmp MainFinal
008B                         max1:
008B  B9 0007                        mov cx, 7         ; |i2| < 7
008E  EB 13 90                       jmp MainFinal
0091                        f6Second:
0091  8B 0E 0008 R                 mov cx, i1
0095  2B 0E 000A R                  sub cx, i2
0099  83 F9 00                      cmp cx, 0
009C  7D 05                         jge MainFinal
009E  F7 D9                               neg cx
00A0  EB 01 90                            jmp MainFinal
00A3                        MainFinal:      ; в cx лежит зна
                    чение функции f8
00A3  CB                    ret
```

```
00A4                    Main        ENDP
00A4                    CODE        ENDS
                        END Main
```


Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	00A4	PARA	NONE	
DATA	000C	PARA	NONE	

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F3FINAL	L NEAR	0031	CODE
F3SECOND	L NEAR	0026	CODE
F6SECOND	L NEAR	0091	CODE
F8FINAL	L NEAR	0065	CODE
F8SECOND	L NEAR	0055	CODE

```
I . . . . . L WORD    0004    DATA
I1 . . . . . L WORD    0008    DATA
I2 . . . . . L WORD    000A    DATA

K . . . . . L WORD    0006    DATA

MAIN . . . . . F PROC    0000    CODE    Length = 00A4
MAINFINAL . . . . . L NEAR    00A3    CODE
MAX1 . . . . . L NEAR    008B    CODE

SKIP2 . . . . . L NEAR    0076    CODE

@CPU . . . . . TEXT  0101h
@FILENAME . . . . . TEXT  lr3
@VERSION . . . . . TEXT  510


  104 Source  Lines
  104 Total   Lines
  23 Symbols

 47990 + 461317 Bytes symbol space free

  0 Warning Errors
  0 Severe  Errors
```