

Міністерство освіти і науки, молоді та спорту України

Національний аерокосмічний університет
ім. М.Є. Жуковського

Кафедра 503

Лабораторна робота № 3

з дисципліни

«Архітектура комп'ютерів»

Тема: «Вивчення логічних команд МП Intel x86»

Варіант № 37

Виконав: ст. гр. 525д

Петренко Р.Е.

Перевірів: ст. викладач

Дужий В.І.

Харків 2012

1. Задание. Дано выражение:

$$X = \begin{cases} A^2 / B - 1, & \text{если } a > b \\ -A, & \text{если } a = b \\ (70 - B) / (A + 1), & \text{если } a < b \end{cases}$$

где исходные значения необходимо разделить на заданную константу, т.е. $A = A / (\text{var} + 2)$, $B = B / (\text{var} + 3)$;

var – номер варианта, равный 35.

Написать ветвящийся алгоритм вычисления значения арифметического выражения и реализовать его в виде программы для МП Intel x86.

2. Исходные данные.

A, B – переменные, длинные целые знаковые числа.

70, 35 – константы, длинное целое знаковое число.

3. Требуемый результат.

X – переменная, длинное целое знаковое число.

4. Описание алгоритма на псевдокоде.

```
Ввод исходных переменных и констант;  
Вычислить условное арифметическое выражение на C;  
// Разделить исходные переменные A и B на константу 37;  
A1 = A/var+2;  
B1 = B/var+3;  
// Сбросить признак ошибки err;  
err = 0;  
Если (A==B)  
    Вычислить выражение X=-A; //(1)  
иначе  
    Если (A>B)  
    {  
        // Проверить знаменатель в выражении (2)  
        Если (B==0)  
            err=1;  
        иначе  
            Вычислить выражение X= A2/B-1; //(2)  
    }  
иначе  
    {  
        // Вычислить знаменатель в выражении (3)  
        tmp = A-1;  
        // Проверить знаменатель в выражении (3)  
        Если (tmp==0)  
            err=1;  
        иначе  
            Вычислить выражение X=(70-B)/(A-1); //(3)  
    }
```

Вывод полученного результата

Детальный алгоритм решения задачи представлен ниже (рис.1).

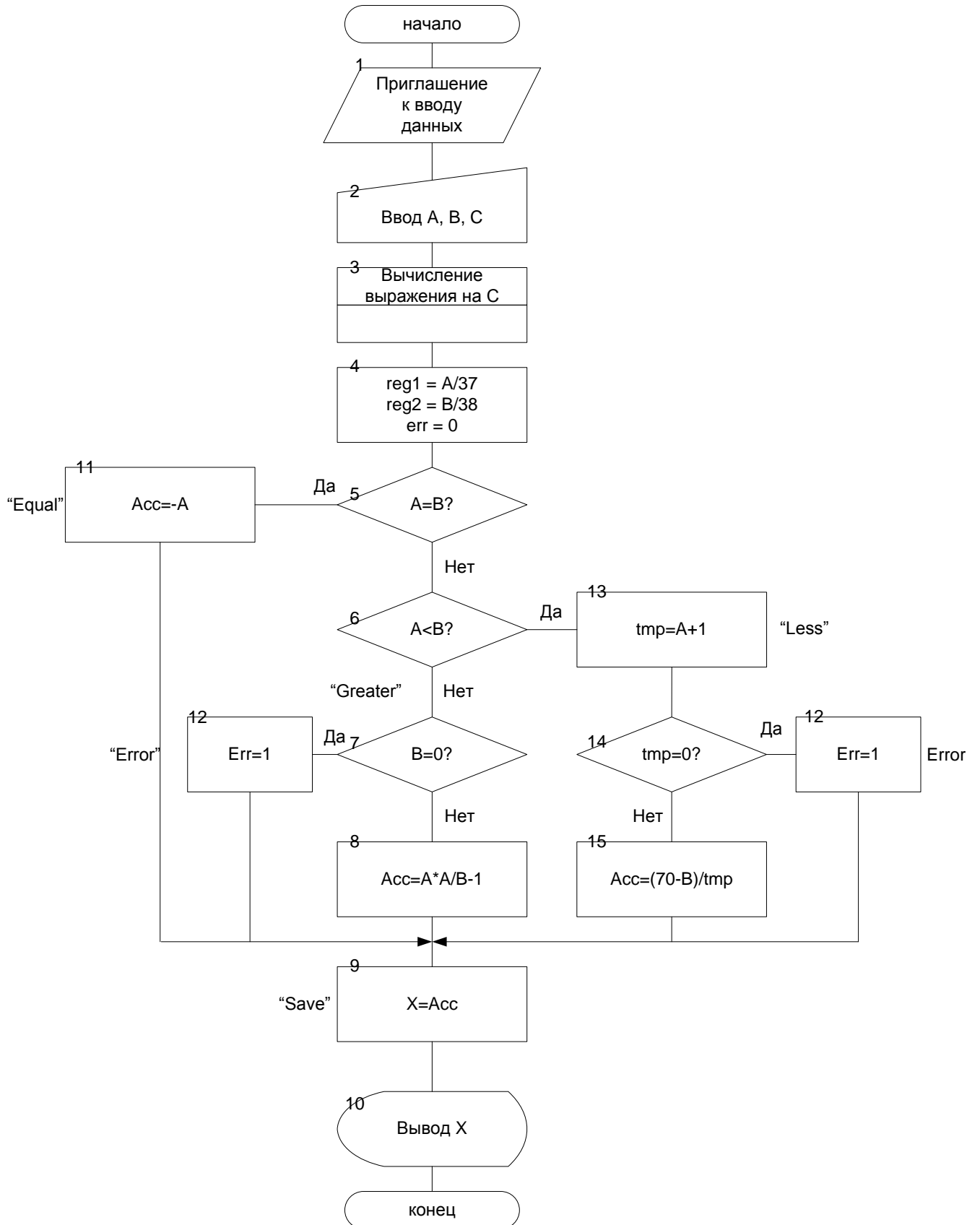


Рис.1. Схема алгоритма вычисления значения выражения

$$X = \begin{cases} A^2 / B - 1, & \text{если } a > b \\ -A, & \text{если } a = b \\ (70 - B) / (A + 1), & \text{если } a < b \end{cases}$$

5. Текст программы с комментариями.

```
//+=====
// File lab_3.cpp
// Ветвящаяся программа
// Вариант 35
// Эта программа вычисляет условное арифметическое выражение
//      { a^2/b-1,      a>b }
// X = < -a,      a=b >
//      { (70-b)/(a+1), a<b }
// где a=a/(35+2), b=b/(35+3)
//
// Выполняется проверка на переполнение
//
// (C) Дужий В.И., 2012
//-----
#include <stdio.h>

#define    VAR    35
long int  a, a1, b, b1, x, x_a, tmp;
int err;  // Ошибка в выражении на C
int err_a; // Ошибка в выражении на ассемблере

int main()
{
    printf("\n\t\t(C) Дужий В.И., 2012");
    for (;;)
    {
        printf("\n\tВычислить выражение:");
        printf("\n\t      { a^2/b-1,      a>b }");
        printf("\n\tX = < -a,      a=b >");
        printf("\n\t      { (70-b)/(a+1), a<b }");
        printf("\n\tгде a=a/37, b=b/38");
        printf("\nПожалуйста, введите целые числа А и В : ");
        scanf("%li%li",&a,&b);
//===== C =====
// Разделить исходные переменные
    a1 = a/(VAR+2);
    b1 = b/(VAR+3);
// Сбросить признак ошибки на C
    err = 0;
// Вычислить выражение
// (a==b)?
    if (a1==b1)
        x = -a1;
    else
// (a>b)?
        if (a1>b1)
        {
            if (b1==0)
                err = 1;
            else
                x = a1*a1/b1-1;
        }
// (a<b)?
    else
    {
        tmp = a1+1;
        if (tmp==0)
```

```

        err = 1;
    else
        x = (70-b1)/tmp;
};
//===================================================== Assembler =====
// err_a=0; Нет ошибок
__asm{
    mov     err_a,0
}
// Разделить исходные переменные на знаменатель VAR
__asm{
// b1=b/DENOM_A+3
    mov     ebx,VAR+3
    mov     eax,b
    cdq
    idiv    ebx
    mov     edi,eax        // b1 -> edi
// a1=a/DENOM_A+2
    mov     ebx,VAR+2
    mov     eax,a
    cdq
    idiv    ebx            // a1 -> esi
}
__asm{
// Вычислить выражение
// if (a1==b1)
    cmp     eax,edi
    je      Equal
    jlt     Less
Greater:
// if (b1==0) err_a = 1
    test    edi,edi
    je      Error
// else x = a1*a1/b1-1;
    imul    eax,eax
    cdq
    idiv    edi
    dec     eax
    jmp     Save
Equal:
// x = -a1;
    neg     eax
    jmp     Save
// else x = (70-b1)/tmp
Less:
// tmp = a1+1 -> eax
    inc     eax
// if ((a1+1)==0) err_a = 1
    je      Error
    sub     edi,70        // b1-70 -> edi
    neg     edi           // 70-b1 -> edi
    xchg    edi,eax       // tmp <-> (70-b1)
    cdq
    idiv    edi           // (70-b1)/tmp -> eax
Save:
    mov     x_a,eax       // eax -> x
    jmp     End
// err_a = 1
Error:                    // err = 1

```

```

        inc     err_a
End:
}
// Вывод результатов
if (err)
    printf("****( C )*** Ошибка: попытка деления на 0\n");
else
    printf("Результат ( C ): %li\n",x);
if (err_a)
    printf("****(Asm)*** Ошибка: попытка деления на 0\n");
else
    printf("Результат (Asm): %li\n",x_a);
}
return 0;
}

```

6. Тестовые примеры.

Номер	Исходные данные		Ожидаемый результат	Полученный результат	Цель теста
	A	B			
1	300	200	11		(a>b) & (b<>0)
2	-300	-500	-5		(a>b) & (b<>0)
3	800	30	???		(a>b) & (b=0) , деление на 0
4	600	-30	???		(a>b) & (b=0) , деление на 0
5	500	500	-500		a=b
6	-700	-700	700		a=b
7	-50	400	???		a<b) & (a=0) , деление на 0