Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧËТ**  
по лабораторной работе №9  
по дисциплине: «Арифметические и логические основы вычислительной техники»  
на тему «Деление в цифровых процессорах»

Выполнил студент группы 22ВВП1:  
Беляев Д.

Приняли:  
Калиниченко Е. И.

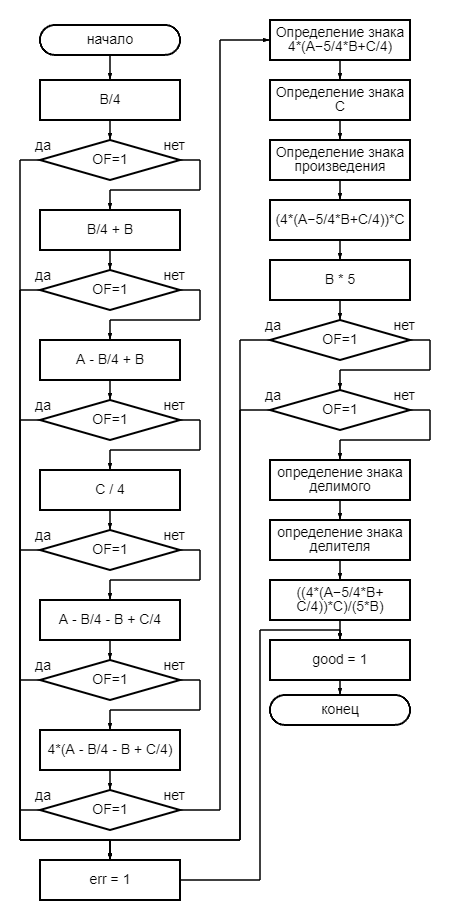
Пенза 2023

**Лабораторное задание:**

Написать на языке ассемблера программу вычисления выражения

**Y=((4\*(А−5/4\*В+С/4))\*C)/(5\*B)**.Процессор имеет разрядность – 16 бит. Исходные данные (значения переменных заданного выражения - целые, 16-битовые со знаком) располагаются в оперативной памяти, результат вычисления также поместить в оперативную память. Деление переменных при вычислении выражения осуществлено с помощью операций сложения, вычитания, сдвига. Умножение и деление на константы осуществлено с помощью операций сдвига, сложения и вычитания.

Общий алгоритм вычисления выражения приведен на **блок схеме**:



**Листинг Программы:**

; Y=((4\*(А−5/4\*В+С/4))\*C)/(5\*B)

data segment

;=============

A dw 500d

B dw -100d

C dw 40d

Y1 dw ?

Y2 dw ?

chast dw ?

ost dw ?

good db ?

err db ?

del dw ?

del\_neg dw ?

mask dw 8000h

;=============

data ends

code segment

assume cs: code, ds:data, ss: nothing

start: mov ax, data

mov ds,ax

;=============

mov good, 0

mov err, 0

mov bx, B

mov cl, 2

sar bx, cl; B/4

jno N1;

jmp ALARM

N1:

add bx, B; B/4 + B

jno N2;

jmp ALARM

N2:

mov ax, A

sub ax, bx; A - B/4 - B

jno N3;

jmp ALARM

N3:

mov bx, C

mov cl, 2

sar bx, cl; C/4

jno N4;

jmp ALARM

N4:

add ax, bx; A - B/4 - B + C/4

jno N5;

jmp ALARM

N5:

mov cl, 2

sal ax, cl; 4(A - B/4 - B + C/4)

jno N6;

jmp ALARM

N6:

; Умножение

mov bx, C

; проверка знака суммы

mov dx, mask

mov cx, dx

and dx, ax

jz PLUS\_LAST\_STEP

neg ax

mov dx, 1

PLUS\_LAST\_STEP:

; проверка знака C

and cx, bx

jz PLUS\_C

neg bx

mov cx, 1

PLUS\_C:

xor dx, cx

push dx

; Цикл умножения

xor dx, dx

mov cx, 15

MUL\_NUM:

rcr bx, 1

jnc a\_1

add dx, ax

a\_1:

rcr dx, 1

loop MUL\_NUM

rcr bx, 1

rcr dx, 1

rcr bx, 1

; Возврат знака

pop cx

test cl, 1

jz EXIT\_MUL

not dx

neg bx

EXIT\_MUL:

mov Y1, dx; старшая часть

mov Y2, bx; младшая часть

; Вычисление делителя

mov dx, B

mov ax, B

mov cl, 2

sal ax, cl ; B\*4

add dx, ax ; B\*4 + B = B\*5

jno N7;

jmp ALARM

N7:

cmp dx, 0

jne CHECK\_DELIMOE

jmp ALARM

; Проверка знака делимого

CHECK\_DELIMOE:

xor si, si

xor di, di

mov ax, Y1

mov bx, Y2

test ax, mask

jz CHECK\_DELITEL

not ax

not bx

add bx, 1

adc ax, 0

add si, 1

; Проверка знака делителя

CHECK\_DELITEL:

test dx, mask

jz PRED\_DIV

neg dx

add si, 2

; si = 1 отрицательно делимое

; si = 2 отрицателен делитель

; si = 3 отрицательны делитель и делимое

PRED\_DIV:

mov cx, 16

mov del, dx

neg dx

mov del\_neg, dx

add ax, del\_neg

jc ALARM

neg dx

jmp DELENIE

SAVE\_FLAGS:

popf

DELENIE:

rcl bx, 1

rcl ax, 1

jc PLUS

add ax, del\_neg

jmp NOT\_PLUS

PLUS:

add ax, del

NOT\_PLUS:

pushf

loop SAVE\_FLAGS

OSTATOK1:

popf

rcl bx, 1

test ax, mask

jz OSTATOK2

add ax, del

OSTATOK2:

test si, 1

jz OSTATOK\_PLUS

neg ax

add di, 1

OSTATOK\_PLUS:

test si, 2

jz CHASTNOE\_PLUS

add di, 1

CHASTNOE\_PLUS:

cmp di, 1

jne RESULT

neg bx

RESULT:

mov chast, bx

mov ost, ax

jmp ALLGOOD

ALARM:

mov err, 1

ALLGOOD:

mov good, 1

;=============

quit: mov ax,4c00h

int 21

code ends

end start

**Ход работы:**

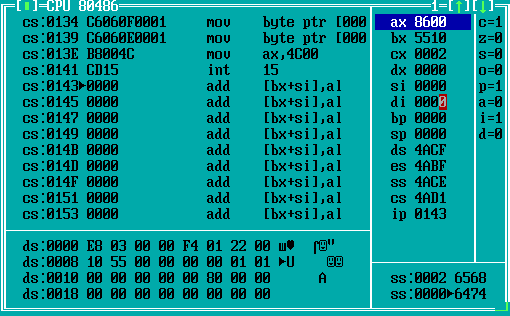
Все возможные исключения при выполнении **(4\*(А−5/4\*В+С/4))\*C** рассмотрены в лабораторной работе №8.

1. B\*5 = 0 и аварийное завершение (деление на ноль)

A = 1000

B = 0

C = 1000

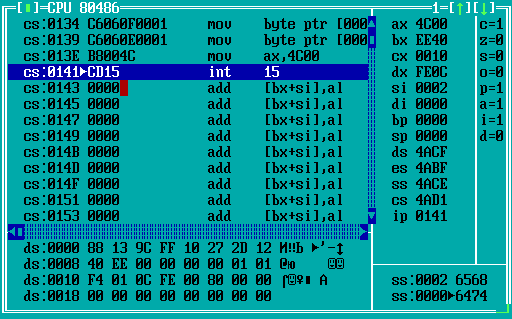


1. Переполнение при делении, частное выходит за пределы 16 бит.

A = 717

B = -100

C = 10

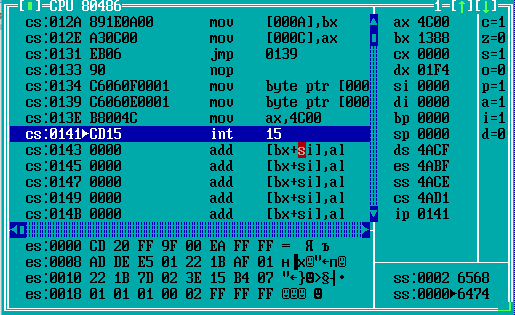


1. Нормальное завершение при положительном делимом и делителе, Y>0, без остатка.

A = 500

B = 100

C = 1000



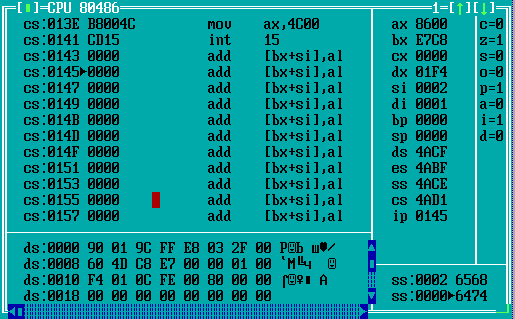
Y = ((4\*(500- 5/4\*100 + 1000/4))\*100)/(100\*5) = 500(10) = 1388(16)

1. Нормальное завершение при положительном делимом и отрицательном делителе, Y<0, без остатка.

A = 500

B = -100

C = 1000



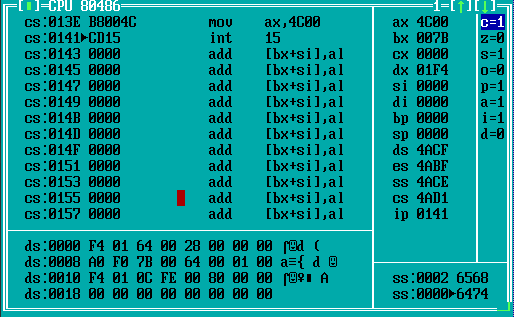
Y = ((4\*(500 + 5/4\*100 + 1000/4))\*1000)/(-100\*5) = -6200(10) => E7C8 (16)

1. Нормальное завершение при положительном делимом и делителе, Y>0, с остатком.

A = 500

B = 100

C = 40



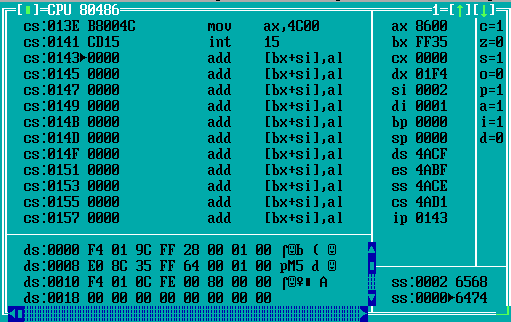
Y = ((4\*(500 - 5/4\*100 + 40/4))\*C)/(100\*5) = 123, ост = 100 (10) => 007B, ост = 64 (16)

1. Нормальное завершение при положительном делимом и отрицательном делителе, Y < 0, с остатком.

A = 500

B = -100

C = 40



Y = ((4\*(500 + 5/4\*100 + 40/4))\*C)/(-100\*5) = -203, ост = 100 (10) => FF35, ост = 64 (16)

**Вывод:**

Создали программу на языке ассемблера для вычисления выражения

**Y=((4\*(А−5/4\*В+С/4))\*C)/(5\*B)**.Умножение и деление были реализованы через операциисдвига, сложения и вычитания. Возможные исключения обработаны.