Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЕТ

По лабораторной работе №7

по дисциплине: «Арифметические и логические основы вычислительной техники»

на тему: «Поразрядные логические операции и операции сдвига в

цифровых процессорах»

Бригада №3

Выполнил:

студент группы 23ВВВ3:

Абсалямов Дамир К.

Принял:

Калиниченко Е.И.

Пенза 2024

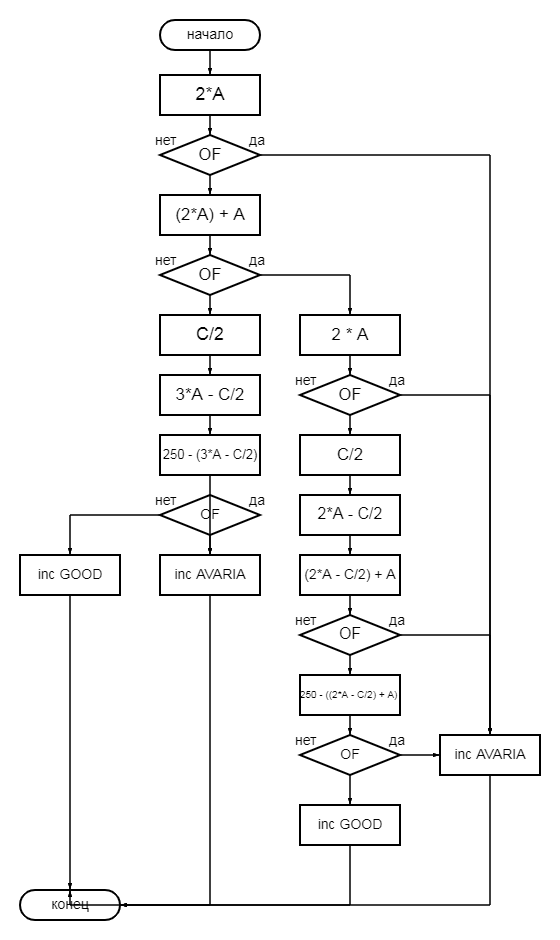
**Лабораторное задание:**

Для процессора i8086 написать на языке ассемблера программу вычисления

выражения **Y=250 - (3 \* А - С/2)**. Формат переменных – 16 бит, целые со знаком (занимают одно слово).

Умножение и деление на константы делать только с использованием операций сдвига, сложения и вычитания.

Общий алгоритм вычисления выражения приведен на блок схеме:



**Листинг программы:**

data segment

A dw ?

C dw ?

Y dw ?

GOOD db ?

AVARIA db ?

data ends

code segment

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

start:

mov ax, data

mov ds, ax

mov GOOD, 0

mov ax, A

sal ax, 1

add ax, A

jo izmen

mov bx, C

sar bx, 1

sub ax, bx

mov cx, 250

sub cx, ax

jo FAVARIA

mov y, cx

jmp FGOOD

izmen:

mov ax, A

sal ax, 1

jo FAVARIA

mov bx, C

sar bx, 1

sub ax, bx

add ax, A

jo FAVARIA

mov cx, 250

sub cx, ax

jo FAVARIA

mov y, cx

FGOOD:

inc GOOD

jmp quit

FAVARIA:

inc AVARIA

quit:

mov ax, 4c00h

int 21

code ends

end start

Чтобы выполнить полное тестирование программы, нужны следующие

расчёты:

1) Нормальное завершение, 2 случая (для Y > 0 и Y < 0), установка флага GOOD = 1

2) Переполнение при (2\*A) + A, переход по метке izmen и установка флага GOOD

3) Переполнение при 2\*A. 2 случая (При основной работе программы и при работе программы через метку izmen). Установка флага AVARIA.

4) Переполнение при (2\*A – C/2) + A, установка флага AVARIA.

5) Переполнение при 250 – (3\*A – C/2) и при 250 – ((2\*A – C/2) + A).

**Прогоны программы:**

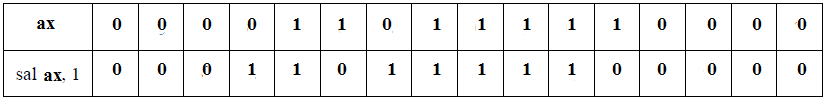
**1) Нормальное завершение, Y > 0:**

A = 356810 = DF016

C = 2140010 = 539816

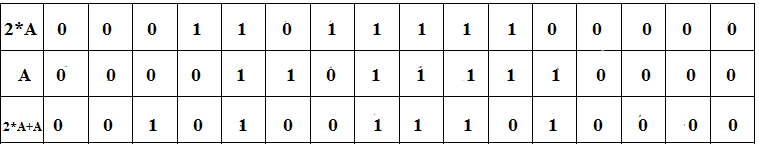
Y = 250 – (3\*3568 – 21400/2) = 250 – (10704 – 10700) = 250 – 4 = 24610

2\*A



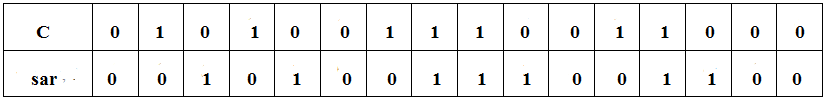
SF = 0 CF = 0 OF = 0 ZF = 0

(2\*A) + A



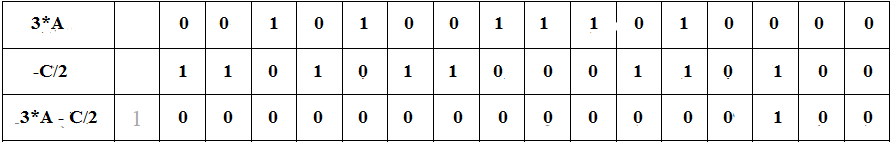
SF = 0 CF = 0 OF = 0 ZF = 0

C/2



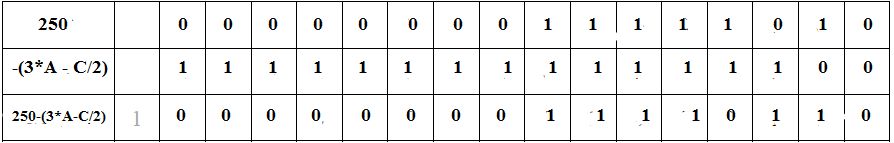
SF = 0 CF = 0 OF = 0 ZF = 0

3\*A – C/2



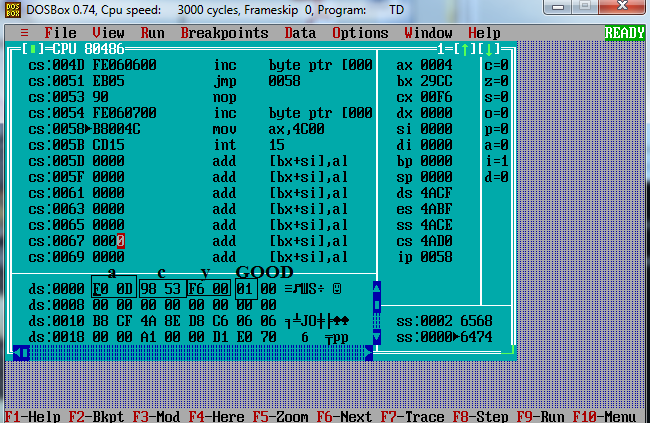
SF = 0 CF = 1 OF = 0 ZF = 0

250 – (3\*A – C/2)



SF = 0 CF = 1 OF = 0 ZF = 0

Результат



Y = 0000 0000 1111 0110 = 00F616=24610

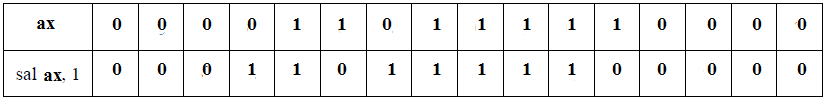
**2) Нормальное завершение, Y < 0:**

**A = 356810 = DF016**

**C = 2037610 = 4F9816**

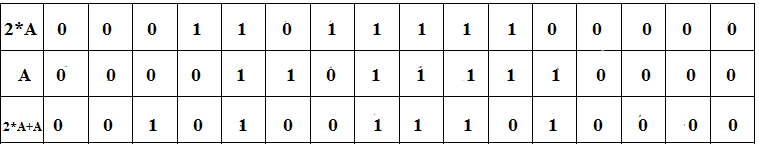
**Y = 250 – (3\*A –C/ 2) = 250 – (3\*3568 – 20376/2) = 250 – (10704 – 10188) = 250 – 516 = -26610**

**2\*A**

****

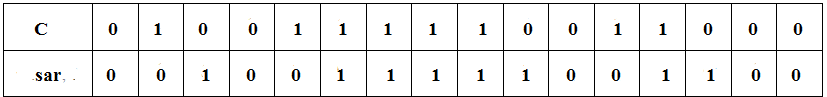
SF = 0 CF = 0 OF = 0 ZF = 0

**(2\*A) + A**

****

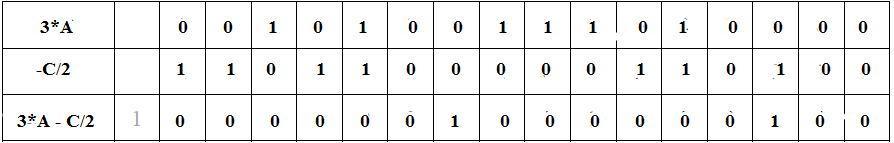
SF = 0 CF = 0 OF = 0 ZF = 0

**C/2**

****

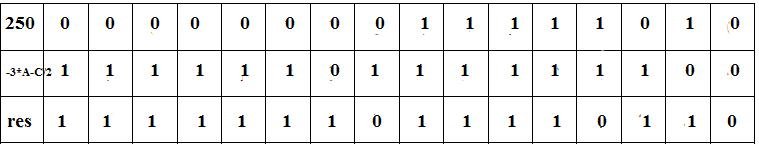
SF = 0 CF = 0 OF = 0 ZF = 0

**3\*A – C/2**

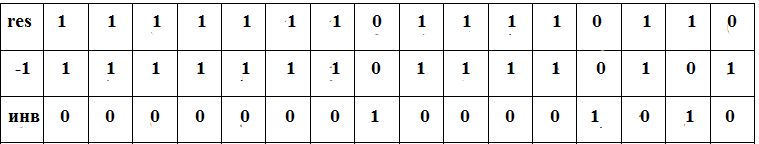
****

SF = 0 CF = 1 OF = 0 ZF = 0

250 – (3\*A – C/2)

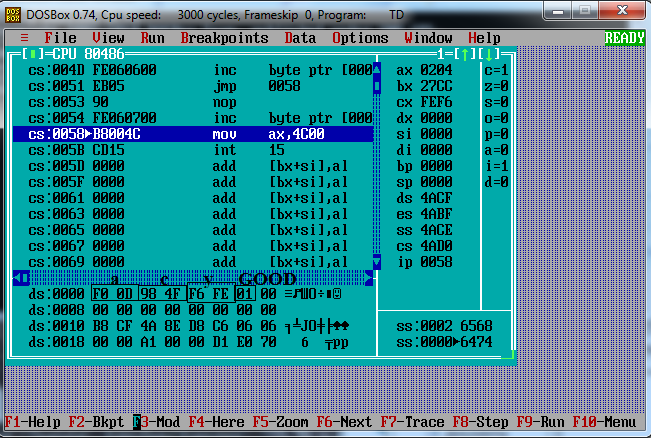
****

**ПЕРЕВОД РЕЗУЛЬТАТА**

****

SF = 0 CF = 0 OF = 0 ZF = 0

**Результат**

****

**Y =[1111 1110 1111 0110]= -0000 0001 0000 1010 = -010A16 = -26610**

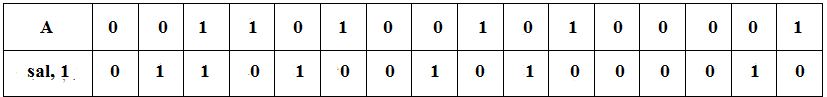
**3) Переполнение при (2\*A) + A, переход по метке izmen и установка флага GOOD:**

**A = 1347310=34A116**

**C = 2694010=693C 16**

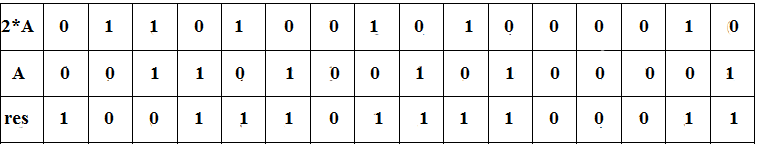
**Y = 250 – ((2 \* A – C/2) + A) = 250 – ((2\*13473 – 26940/2) + 13473) = 250 – ((26946 -13470) + 13473) = 250 – (13476 + 13473) = 250 – 26946 = -2669910 = -684B16 = 97B516**

**2\*A**

****

SF = 0 CF = 0 OF = 0 ZF = 0

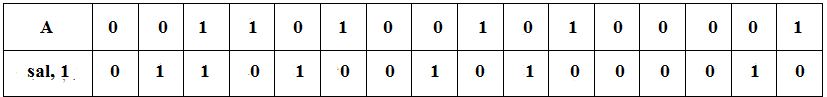
(2\*A) + A



SF = 0 CF = 0 OF = 1 ZF = 0

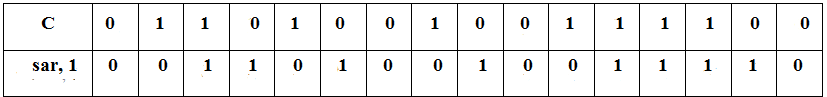
Возникло переполнение, программа переходит по метке izmen:

**2\*A**

****

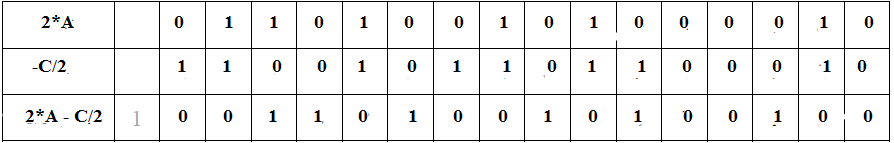
SF = 0 CF = 0 OF = 0 ZF = 0

C/2



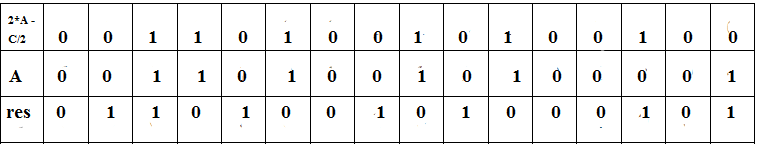
SF = 0 CF = 0 OF = 0 ZF = 0

2\*A – C/2



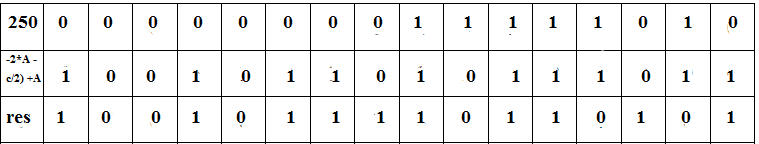
SF = 0 CF = 1 OF = 0 ZF = 0

(2\*A – C/2) + A



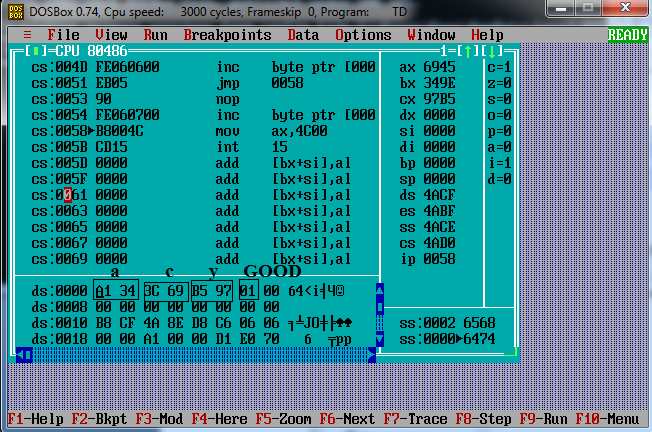
SF = 0 CF = 0 OF = 0 ZF = 0

250 – ((2\*A – C/2) + A)



SF = 1 CF = 0 OF = 0 ZF = 0

РЕЗУЛЬТАТ:



Y = [1001 0111 1011 0101] = -0110 1000 0100 1011 = -684B16 = -2669910

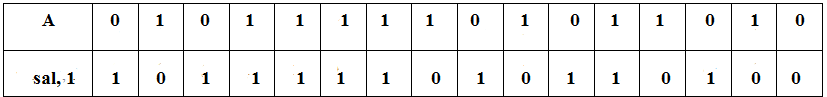
**4) Переполнение при 2\*A, установка флага AVARIA:**

**A = 0101 1111 0101 1010**

**C = в данном случае бессмысленно, так как программа не пойдёт работать дальше 2\*A ввиду переполнения.**

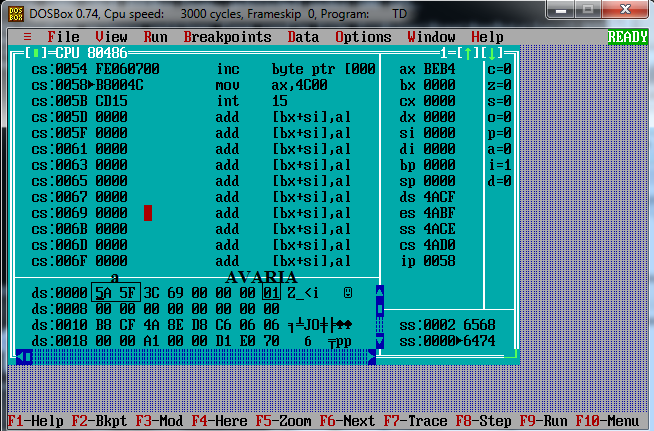
**Y = в данном случае бессмысленно, так как программа не пойдёт работать дальше 2\*A ввиду переполнения.**

**2\*A**

****

SF = 1 CF = 0 OF = 1 ZF = 0

РЕЗУЛЬТАТ:



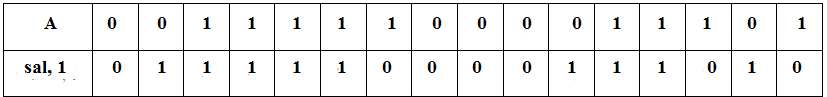
**5) Переполнение при (2\*A – C/2) + A, установка флага AVARIA:**

**A = 3E1D**

**C = 71E0**

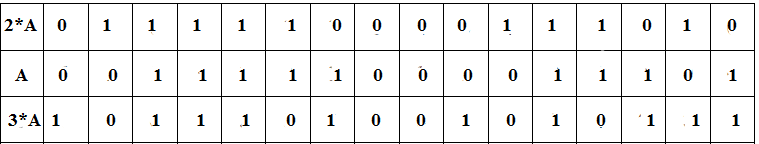
**Y = в данном случае бессмысленно, так как программа не пойдёт работать дальше (2\*A – С/2) + A ввиду переполнения.**

**2\*A**

****

SF = 0 CF = 0 OF = 0 ZF = 0

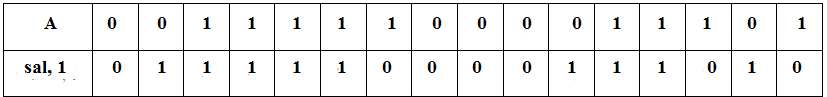
**(2\*A) + A**

****

SF = 1 CF = 0 OF = 1 ZF = 0

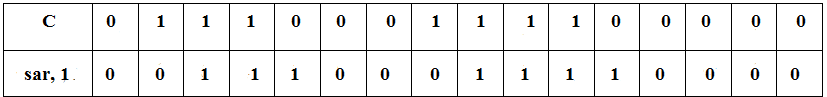
Переполнение. Переход по метке izmen

**2\*A**

****

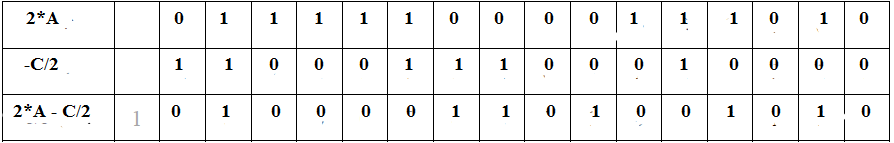
SF = 0 CF = 0 OF = 0 ZF = 0

C/2



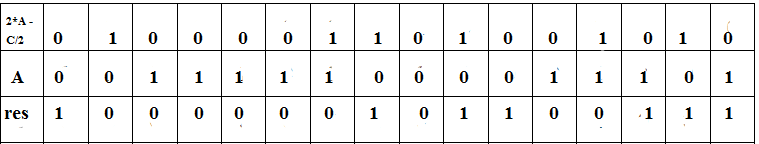
SF = 0 CF = 0 OF = 0 ZF = 0

(2\*A – C/2)



SF = 0 CF = 1 OF = 0 ZF = 0

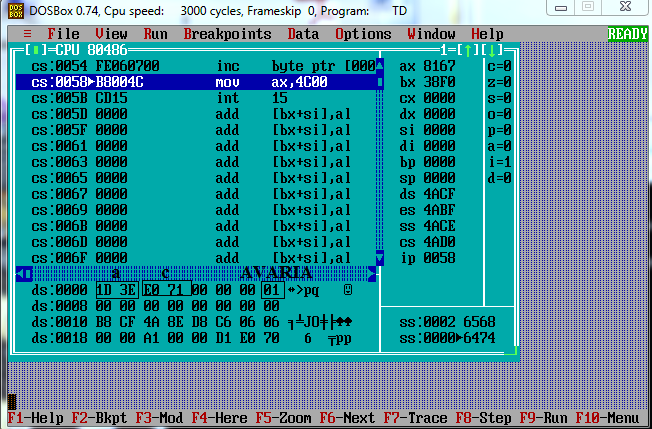
(2\*A – C/2) + A



SF = 1 CF = 0 OF = 1 ZF = 0

Переполнение.

РЕЗУЛЬТАТ:



**6) Переполнение при 250 – (3\*A – C/2):**

В таком случае переполнения не будет, так как чтобы получить данное переполнение, программа должна вычесть 250 – (- 32769), а это невозможно ввиду переполнения, получаемого ранее в программе.

**7) Переполнение при 250 – ((2\*A – C/2) + A:**

В таком случае переполнения не будет, так как чтобы получить данное переполнение, программа должна вычесть 250 – (- 32769), а это невозможно ввиду переполнения, получаемого ранее в программе.

**Вывод:** создал программу на языке ассемблера для вычисления выражения

**Y=250 - (3 \* А - С/2)**. Умножение и деление на константы реализовано

операциями сдвига, сложения и вычитания. Исключение – переполнение -

обработано для пересчета и получения верного результата.