**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**РУТ (МИИТ)**

Институт управления и цифровых технологий

Кафедра «Цифровые технологии управления транспортными процессами»

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

|  |  |  |
| --- | --- | --- |
| **по дисциплине:** | | Ассемблер |
| **на тему:** | АРИФМЕТИЧЕСКИЕ ВЫЧИСЛЕНИЯ | |
| **цель работы:** | Изучение арифметических команд сложения, вычитания, умножения и деления, изучение команд сравнения и перехода, написание и отладка программы вычисления арифметических выражений с ветвлением. | |

|  |  |
| --- | --- |
| **Выполнил студент группы УИС-312** | **(Рыжов В. Р.)** |
| **Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **(Преподаватель кафедры ЦТУТП Победоносцев И. Н.)** |

**Москва 2022 г.**

Оглавление

[**Требования к отчету** 3](#_Toc117530324)

[**Цель работы:** 4](#_Toc117530325)

[**1 Блок-схема программы** 6](#_Toc117530326)

[**2 ТЕКСТ ПРОГРАММЫ С КОМЕНТАРИЯМИ** 7](#_Toc117530327)

[**3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ** 9](#_Toc117530328)

[**4 РЕЗУЛЬТАТЫ ПОШАГОВОГО ВЫПОЛНЕНИЯ ПРОГРАММЫ В ОТЛАДЧИКЕ** 12](#_Toc117530329)

[**5 ВОПРОСЫ** 14](#_Toc117530330)

[**ЗАКЛЮЧЕНИЕ** 16](#_Toc117530331)

# **Требования к отчету**

Отчет должен включать:

1. Продемонстрируйте работающую программу преподавателю с предъявлением результатов в командной строке.

2. Подготовьте отчет, включающий:   
- текст варианта задания с указанием типа переменных X и Y;   
- укрупненную блок-схему алгоритма разработанной программы;   
- текст разработанной программы с обязательными комментариями;   
- результаты работы программы в виде заполненной таблицы и скриншотов окон отладчика.  
- результаты пошагового выполнения программы в отладчике в виде заполненной таблицы.

# **Цель работы:**

Изучение арифметических команд сложения, вычитания, умножения и деления, изучение команд сравнения и перехода, написание и отладка программы вычисления арифметических выражений с ветвлением.**Задачи:**

Написать и отладить программу вычисления функции Y(X), смотри рисунок 0.1, с учетом следующих требований:

- типы переменных X и Y задаются преподавателем;   
- при выполнении арифметических операций там, где это необходимо, производить контроль флагов переноса (СF) и переполнения (OF) с целью выявления ошибочного результата;   
- арифметические ошибки фиксировать с помощью переменной MYERR, которой присваивается значение 1 при наличии ошибки, и 0 в противном случае;   
- в шапку программы включить в виде комментария сведения об авторе, номер варианта и расчетную формулу.

Изображение выглядит как текст

Автоматически созданное описание  
Рис. 0.1 – функция, которую нужно реализовать

# **1 Блок-схема программы**

На рисунке 1.1 представлена блок схема для программы arithmetic.

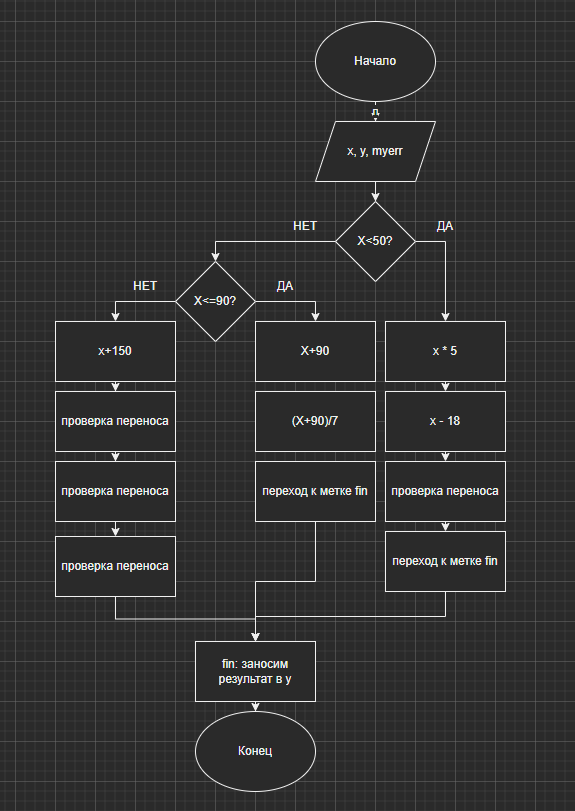


Рис 1.1 – Блок схема для приложения arithmetic

# **2 ТЕКСТ ПРОГРАММЫ С КОМЕНТАРИЯМИ**

;Лабораторная работа №2

;Выполнил ст. гр. УИС-312 Рыжов Виталий Романович

;Вариант 20

; 5x - 18 ; x<50

; y= (90+x)/7 ; 50<=x<=90

; x+150 ; x>90

.486

.model flat, stdcall

option casemap: none

.stack 100h

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

include <\masm32\include\kernel32.inc>

includelib <\masm32\lib\kernel32.lib>

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

.data

x db 140

y db ?

myerr db 0

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

.code

main:

; 5x - 18, x<50

cmp x, 50 ; х-50 ?

jae int2 ; если x<50 перейти к метке int2

; выполним 5x - 18

mov ax, 0 ; очистим регистр

mov al, 5 ; помещаем в рег al значение 5

mul x ; значение рег ax = x \* al(значение 5)

sub ax, 18 ; вычитаем из значение регистра al(x\*5) число 18

jc osh ; Переход, если есть перенос

jmp fin ; перейти к метке fin

;--------------------------------

int2:

; (90+X)/7, 50<=X<=90

cmp x, 91 ; x-90?

jae int3 ; если x>=90 перейти к метке int3

mov ax, 0 ; очистим регистр

; выполним (90+X)/7

mov al, x ; помещаем в регистр al значение x

add al, 90 ; прибавляем к x значение регистра al(90)

mov bl, 7 ; помещаем 7 в рег bl

div bl ; делим значение регистра ax(x+90)/bl ответ получаем в регистр al

jmp fin ; перейти к метке fin

;--------------------------------

int3:

;выполним X+150, X>90

mov ax, 0 ; очистим регистр

mov al, x ; помещаем значение x в регистр al

add al, 150 ; прибавляем к x 150

jc osh ; Переход, если есть перенос

jmp fin ; перейти к метке fin

osh: mov myerr, 1 ; установить код ошибки

;--------------------------------

fin: mov y, al ; сформировать результат в переменной y

exit: mov al, myerr ; код завершения программы

invoke ExitProcess, al

end main

# **3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ**

На рисунке 3.1 видны результаты работы программы при заданном x = 30.

Изображение выглядит как текст

Автоматически созданное описание  
Рис 3.1 результат работы программы при заданном x=30.

На рисунке 3.2 видны результаты работы программы при заданном x = 60.

Изображение выглядит как текст

Автоматически созданное описание  
Рис 3.2 результат работы программы при заданном x=60.

На рисунке 3.3 видны результаты работы программы при заданном x = 100.

Изображение выглядит как текст

Автоматически созданное описание  
Рис 3.3 результат работы программы при заданном x=100.

На рисунке 3.4 видны результаты работы программы при заданном x = 140.

Изображение выглядит как текст

Автоматически созданное описание

Рис 3.4 результат работы программы при заданном x=140.

Далее в таблице 3.1 приведены результаты выполнения программы при разных исходных значениях.

Таблица 3.1 - результаты работы программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **X1 =30** | **X2 = 60** | **X3 = 100** | **X4 = 140** |
| **Y (hex)** | 84 | 15 | FA | ~ |
| **Y (dec)** | 132 | 21 | 250 | ~ |

# **4 РЕЗУЛЬТАТЫ ПОШАГОВОГО ВЫПОЛНЕНИЯ ПРОГРАММЫ В ОТЛАДЧИКЕ**

Результаты выполнения программ в пошаговом режиме при X=X1 и X=X4 приведены в таблицах 4.1 и 4.2  
**Таблица 4.1** - результат при х = х1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № ПП | Команда | EAX | EBX | ESP | EBP | EIP | X | Y | MYERR | ФЛАГИ | | | |
| CF | OF | ZF | SF |
| 1 | 803D | 0019FFCC | 002C9000 | 0019FF74 | 0019FF80 | 00401000 | 1E | 00 | 00 | 0 | 0 | 1 | 0 |
| 2 | 73 14 |  |  |  |  | 00401007 |  |  |  | 1 | 0 | 0 | 1 |
| 3 | 66:B8 0000 |  |  |  |  | 00401009 |  |  |  |  |  |  |  |
| 4 | B0 05 | 00190000 |  |  |  | 0040100D |  |  |  |  |  |  |  |
| 5 | F625 | 00190005 |  |  |  | 0040100F |  |  |  |  |  |  |  |
| 6 | 66:83E8 12 | 00190096 |  |  |  | 00401015 |  |  |  | 0 |  |  |  |
| 7 | 72 2B | 00190084 |  |  |  | 00401019 |  |  |  |  |  |  | 0 |
| 8 | EB 30 |  |  |  |  | 0040101B |  |  |  |  |  |  |  |
| 9 | A2 01204000 |  |  |  |  | 0040104D |  | 84 |  |  |  |  |  |
| 10 | A0 02204000 |  |  |  |  | 00401052 |  |  |  |  |  |  |  |
| 11 | 6A 00 | 00190000 |  |  |  | 00401057 |  |  |  |  |  |  |  |
| 12 | 66:0FB6C0 |  |  | 0019FF70 |  | 00401059 |  |  |  |  |  |  |  |
| 13 | 66:50 |  |  |  |  | 0040105D |  |  |  |  |  |  |  |
| 14 | E8 00000000 |  |  | 0019FF6E |  | 0040105F |  |  |  |  |  |  |  |
| 15 | -FF25 34204000 |  |  | 0019FF6A |  | 00401064 |  |  |  |  |  |  |  |

**Таблица 4.2** - результат при х = х4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № ПП | Команда | EAX | EBX | ESP | EBP | EIP | X | Y | MYERR | ФЛАГИ | | | |
| CF | OF | ZF | SF |
| 1 | 803D | 0019FFCC | 002C9000 | 0019FF74 | 0019FF80 | 00401000 | 8C | 00 | 00 | 0 | 0 | 1 | 0 |
| 2 | 73 14 |  |  |  |  | 00401007 |  |  |  |  | 1 | 0 |  |
| 3 | 803D 00204000 |  |  |  |  | 0040101D |  |  |  |  |  |  |  |
| 4 | 73 11 |  |  |  |  | 00401024 |  |  |  |  |  |  |  |
| 5 | 66:B8 0000 |  |  |  |  | 00401037 |  |  |  |  |  |  |  |
| 6 | A0 00204000 | 00190000 |  |  |  | 0040103B |  |  |  |  |  |  |  |
| 7 | 84 96 | 0019008C |  |  |  | 00401040 | 8C |  |  |  |  |  |  |
| 8 | 72 02 | 00190022 |  |  |  | 00401042 |  |  |  | 1 |  |  |  |
| 9 | C605 02204000 |  |  |  |  | 00401046 |  |  |  |  |  |  |  |
| 10 | A2 01204000 |  |  |  |  | 0040104D |  |  |  |  |  |  |  |
| 11 | A0 02204000 |  |  |  |  | 00401052 |  |  |  |  |  |  |  |
| 12 | A6 00 | 00190001 |  |  |  | 00401057 |  | 00 | 01 |  |  |  |  |
| 13 | 66:0FB6C0 |  |  | 0019FF70 |  | 00401059 |  |  |  |  |  |  |  |
| 14 | 66:50 |  |  |  |  | 0040105D |  |  |  |  |  |  |  |
| 15 | E8 00000000 |  |  | 0019FF6E |  | 0040105F |  |  |  |  |  |  |  |
| 16 | -FF 34204000 |  |  | 0019FF6A |  | 00401064 |  |  |  |  |  |  |  |

# **5 ВОПРОСЫ**

1) **Поясните действие команд умножения и деления в вашей программе.**

**Почему команды умножения и деления используют только один явный**

**операнд?**

a) Второй множитель должен находится в регистре или в памяти. В моей

программе, при операциях умножения или деления все храниться в

регистре.

2) **Что означают и как формируются флаги переноса (CF), переполнния (OF),**

**знака (SF) и нуля (ZF) при выполнении различных арифметиких операций?**

a) CF (Carry Flag) - флаг переноса. Содержит значение «переносов» или

«заемов», затрагивающих старший разряд результирующего

операнда при выполнении арифметических операций и логических

операций сдвига.

b) OF (Overflow Flag) - флаг переполнения. Фиксирует арифметическое

переполнение, т.е. перенос (заем) в/из старшего (знакового) бита при

выполнении знаковых арифметических операций.

c) ZF (Zero Flag) - флаг нуля. Формируется в качестве результата

выполнения арифметических команд и команд сравнения. Ненулевой

результат приводит к установке нулевого значения этого флага, а

нулевой - к установке единичного значения. Команды условного

перехода JE и JZ проверяют значение именно этого флага.

d) SF (Sign Flag) - флаг знака. Устанавливается в соответствии со знаком

старшего бита результата после выполнения арифметических

операций: положительный результат устанавливает 0, а

отрицательный - 1. Команды условного перехода JG и JL проверяют

именно этот флаг.

3) **В чем сходство и различие операций вычитания (SUB) и сравнения (CMP)?**

a) Основное различие в получаемом результате cmp – помогает

определить одинаковые ли числа, а sub – выполняет вычитание

одного числа из другого.

b) Сходство в том, что при сравнении используется вычитание.

4) **Почему следует использовать разные команды условного перехода при**

**сравнении знаковых и беззнаковых чисел?**

a) Признаком правильности результата вычитания беззнаковых чисел

является значения флага переноса CF=0 (CF=1 – ошибка). Признаком

правильности результата сложения чисел со знаком является

значения флага переполнения OF=0 (OF=1 – ошибка).

5) **Поясните различия между длинным, коротким, ближним и дальним**

**переходами.**

a) Прямой короткий (short) переход. Прямым называется переход, в

команде которого в явной форме указывается метка, на которую

нужно перейти.

b) Прямой ближний (near), или внутрисегментный переход. Этот

переход отличается от предыдущего только тем, что под смещение к

точке перехода отводится целое слово.

c) Прямой дальний (far), или межсегментный переход. Этот переход

позволяет передать управление в любую точку любого сегмента.

6) **Запишите команды для вычисления выражений**: 2\*17-3; 250/13+8; 47\*10/7;

(720-350)/3.

a) Нужные команды по порядку: (mul, sub; div, add; mul, div; sub, div)

# **ЗАКЛЮЧЕНИЕ**

В ходе данной работы я познакомился арифметическими возможностями кода ассемблера. А также лучше понял алгоритмами работы программ, написанных для ассемблера.