ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Жиданов К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
|  |
| по курсу: «Информатика» |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 3145 |  |  |  | Маракуева А.Р. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Задание 1:** написать на языке ассемблера программу, реализующую умножение и сложение в целых числах по заданной формуле. Проверить корректность работы программы с помощью отладчика на заданных значениях.

**Ход выполнения:**

1. Индивидуальное условие задания:

**Выражение**: a / (b+c)

**Значения**: [{6,25,13}, {-7,-8,11}]

**Разрядность** **входных** **значений**: 16

**Код**: дополнительный

1. Вычислим значения выражения для заданных значений:

Для первого набора данных: 6/(25+13) = 0 (остаток 6)

Для второго набора данных: -7/(-8+11) = -2 ( остаток -1)

Перевод всех чисел в 16-ричную систему с учётом разрядности и способа кодирования:

Первый набор данных: Второй набор данных:

610  = 616 -710 = FFF916

2510 = 1916 -810 = FFF816

1310 = D16 1110 = B16

1. Регистры для входных и выходных значений:

**Входные значения**: **Выходные значения**:

**DX AX**

**AX DX**

**BX**

1. Заносим программу в отладчик и проверяем значения во всех используемых регистрах
2. Проверяем результаты программы:

Первый набор данных:

После выполнения программы в регистрах находятся числа 6 и 0

Второй набор данных:

После выполнения программы в регистрах находится числа -2 и -1

Данные значения совпадают со значениями, полученными вручную.

**Код программы:**

%include "io.inc"

; a/(b+c)

; {6, 25, 13}

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

MOV DX, 0x0000 ;заполняем ведущий регистр единицами для деления

MOV AX, [a]

MOV BX, [b]

ADD BX, [c]

IDIV BX ;деление со знаком

PRINT\_DEC 4, AX ;частное

NEWLINE

PRINT\_DEC 4, DX ;остаток

RET

section .data

a dw 0x06 ; прямой код 6

b dw 0x19 ; прямой код 25

c dw 0x0D ; прямой код 13

или

%include "io.inc"

; a /(b+c)

; {-7, -8, 11}

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

MOV DX, 0xFFFF ;заполняем ведущий регистр единицами для деления

MOV AX, [a]

MOV BX, [b]

ADD BX, [c]

IDIV BX ;деление со знаком

PRINT\_DEC 4, AX ;частное

NEWLINE

PRINT\_DEC 4, DX ;остаток

RET

section .data

a dw 0xFFF9 ; доп код -7

b dw 0xFFF8 ; доп код -8

c dw 0x0B ; доп код 11

**Задание 2:** написать на языке ассемблера программу, реализующую вычисления в числах с плавающей точкой по заданной формуле. Проверить корректность работы программы с помощью отладчика на заданных значениях

**Ход выполнения:**

1. Индивидуальное условие задания:

**Выражение**: a / (b+c)

**Значения**: [{6,25,13}, {-7,-8,11}]

**Разрядность** **входных** **значений**: 16

**Код**: дополнительный

1. Вычислим значения выражения для заданных значений:
2. Перевод всех чисел в формат чисел с плавающей точкой в соответствии с форматом IEEE 754

Первый набор данных: (0x40C00000, 0x41C80000, 0x41500000)

Второй набор данных: (0xC0E00000, 0xC1000000, 0x41300000)

Перевод выполняется следующим образом (На примере A[0] = 6):

1. Число переводится в **bin** и приводится к виду нормализованнной мантиссы и экспоненты *(Для 110b: мантисса — 1.1b, экспонента — 2)*
2. К значению экспоненты **прибавляется 127** *(экспонента — 129 = 10000010b)*
3. **Знаковый бит** равен 0, т.к. число положительное.
4. 32-ой бит занимает знаковый, 31-24 биты занимает экспонента, оставшиеся биты — дробная часть мантиссы *(1.10000000000000000000000=* *0x40c00000)*
5. Переменные, используемые в section. data для ввода и вывода данных:

Ввод данных: Вывод данных:

a: с r: result

b: b

c: a

1. Заносим программу в отладчик и проверяем значения во всех используемых регистрах
2. Проверяем результаты программы:

Первый набор данных:

После выполнения программы переменной r присваивается значение 3e21af28

Второй набор данных:

После выполнения программы переменной r присваивается значение С1055555

Код программы:

%include "io.inc"

; a /(b+c)

; {6, 25, 13}

section .text

global CMAIN

CMAIN:

MOV EBP, ESP ;запись в обратной польской нотации

FLD DWORD [a]

FLD DWORD [b]

FLD DWORD [c]

FADD

FDIV

FST DWORD [result]

PRINT\_HEX 4, [result]

RET

section .data

a dd 0x40C00000 ; 6 в ieee754

b dd 0x41C80000 ; 25 в ieee754

c dd 0x41500000 ; 13 в ieee754

result dd 0x00000000

или

%include "io.inc"

; a /(b+c)

; {6, 25, 13}

section .text

global CMAIN

CMAIN:

MOV EBP, ESP ;запись в обратной польской нотации

FLD DWORD [a]

FLD DWORD [b]

FLD DWORD [c]

FADD

FDIV

FST DWORD [result]

PRINT\_HEX 4, [result]

RET

section .data

a dd 0x40C00000 ; 6 в ieee754

b dd 0x41C80000 ; 25 в ieee754

c dd 0x41500000 ; 13 в ieee754

result dd 0x00000000

**Задание 3:**

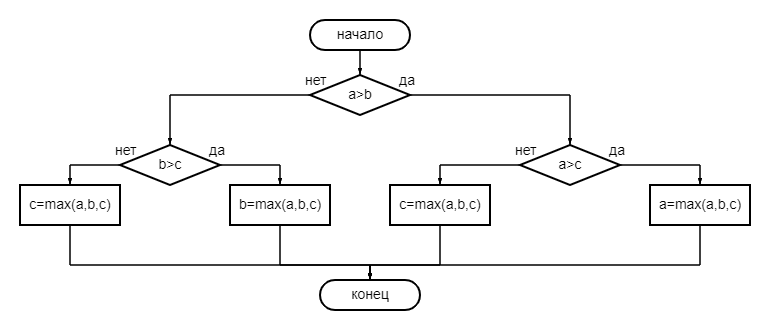
Написать на языке ассемблера программу, которая находит наибольшее из 3 чисел. Проверить корректность работы программы с помощью отладчика на заданных значениях.

**Ход работы:**

1. Индивидуальное условие задания: найти максимальное из 3 чисел a, b и c

Значения (5,58,46)

1. Составим алгоритм:



1. Заносим программу в отладчик и проверяем значения во всех используемых регистрах
2. Проверяем результаты программы:

При данных тестовых наборах программа выводит 58, следовательно программа работает верно

Код программы:

%include "io.inc"

;Наибольшее из трех чисел

section .text

global CMAIN

CMAIN:

MOV AH, [a]

MOV AL, [b]

CMP AH, AL

JA AC

MOV AH, [c]

CMP AL, AH

JA B

JMP C

B:

PRINT\_DEC 1, [b]

RET

C:

PRINT\_DEC 1, [c]

RET

AC:

MOV AL, [c]

CMP AH, AL

JA A

PRINT\_DEC 1, [c]

RET

A:

PRINT\_DEC 1, [a]

RET

section .data

a: db 5

b: db 58

c: db 46