МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра информационных технологий**

**Отчет о выполнении лабораторной работы №2**

**по дисциплине «Системы реального времени»**

Работу выполнил студент 45/2 группы А.А. Козин

Руководитель

доц. каф. ИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Н. Полетайкин

Краснодар

2023

**ЛАБОРАТОРНАЯ РАБОТА 2**

**ИЗУЧЕНИЕ КОМАНД АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ОПЕРАЦИЙ**

Цель: изучение команд арифметических и логических операций, приобретение практических навыков при их использовании в различных вычислительных задачах.

Задание (Вариант 24):

1. Составить программу для расчета заданного арифметического выражения (A+891/(B\*59+C))\*76. Длину и значение переменных A, B, C выбрать самостоятельно. Константы, заданные в выражении, использовать в кодовом сегменте.
2. Описать команды умножения и деления, используемые в программе на предмет длины операндов, участвующих в операции. Охарактеризовать длину результата и место его хранения.
3. Получить загрузочный модуль и протестировать выполнение программы в отладчике.
4. На основе составленной программы выполнить следующие действия:
   * загрузить в аккумулятор число 1 и при помощи команд логических операций установить маскирующее слово, позволяющее определить заданную характеристику содержимого регистра DX;
   * выполнить заданную проверку и ее результат сохранить в переменной RESULT, объявленной в сегменте данных.
5. Перекомпилировать загрузочный модуль и протестировать выполнение программы в отладчике.
6. Произвести расчет времени выполнения программы.
7. Сделать выводы.

**ХОД РАБОТЫ**

Задание 1. Код программы

%include "io.inc"

section .data

A dd 10

B dd 10

C dd 10

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

mov eax, [B]

mov ebx, 59

imul ebx ; B\*59 in eax

mov ebx, [C]

add eax, ebx

mov ebx, eax ; (B\*59+C) in ebx

mov eax, [A]

mov ecx, 891

add eax, ecx ; A+891 in eax

idiv ebx ; (A+891/(B\*59+C)) in eax

mov ebx, 76

imul ebx ; result in eax

PRINT\_DEC 4, eax ; macro for print decimal value answer

mov eax, 1

ret

Анализ арифметических операций.

В программе используются операции умножения imul для регистра ebx размером 4 байта, результат будет записан в первый указанный в операции регистр, то есть eax.

Также используется операция деления idiv для регистра ebx. При этом в команде указывается только регистр, содержащий делитель, делимое берется из регистра eax.

Для операций вычитания и сложения используются два любых регистра общего назначения, результат записывается в регистр, указанный первым (слева).

3. Проверим корректность всех арифметических операций в отладчике:

A screenshot of a computer program

Description automatically generated

Рисунок 1 – Значение регистров до умножения.

A screenshot of a computer program

Description automatically generated

Рисунок 2 – Значение регистров после умножения.

A screenshot of a computer

Description automatically generated

Рисунок 3 – Значение регистров до сложения.

A screenshot of a computer

Description automatically generated

Рисунок 4 – Значение регистров после сложения.

A screenshot of a computer

Description automatically generated

Рисунок 5 – Значение регистров до деления.

A screenshot of a computer

Description automatically generated

Рисунок 6 – Значение регистров после деления.

A screenshot of a computer

Description automatically generated

Рисунок 7 – Вывод результата вычислений.

Контрольные расчет:

A = 10, B = 10, C = 10.

A + 891 = 901;

B\*59+C = 600;

901/600 = 1

1 \* 76 = 76

Результат вычислений: 76

4. Для проверки значения бита регистра edx после операции деления использую следующие инструкции:

mov ebx, 1

and ebx, edx

jz is\_zero

Операция and выставляет флаг ZF, поэтому можно использовать операцию условного перехода jz на метку is\_zero, где запишется результат в переменную RESULT.

Проведем тестирование программы.

Просмотрим значения регистров edx, в котором находится значение остатка от деления и ebx, содержащий маскирующее слово:

A screenshot of a computer

Description automatically generated

Рисунок 8 – Значения регистров перед побитовым умножением.

A screenshot of a computer

Description automatically generated

Рисунок 9 – Значения регистров после побитового умножения.

Рассчитаем количество процессорных тактов, необходимых для выполнения программы:

%include "io.inc"

section .data

A dd 10

B dd 10

C dd 10

RESULT dd 0

section .text

global CMAIN

CMAIN:

mov eax, [B] ;12+5

mov ebx, 59 ;4

imul ebx ;154

mov ebx, [C] ;12+5

add eax, ebx ;3

mov ebx, eax ; 4

mov eax, [A] ;12+5

mov ecx, 891 ;4

add eax, ecx ;3

idiv ebx ; 184

mov ebx, 76 ;4

imul ebx ; 154

PRINT\_DEC 4, eax ;

mov ebx, 1 ;4

and ebx, edx ;3

jz is\_zero ;15

mov eax, 0 ;4

mov [RESULT], eax ;13+5

mov eax, 1 ;4

ret

is\_zero:

mov eax, 1 ;4

mov [RESULT], eax ;13+5

Время работы программы: 631 процессорных тактов для полного выполнения. При частоте процессора M1 Pro в 3.2 ГГц займет 197.2 наносекунд.

**ВЫВОД**

В результате работы, была написана программа для выполнения лабораторной работы №2. Были изучены команды арифметических и логических операций.