МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования  
«Гомельский государственный технический университет имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №6

Вариант 1

по дисциплине: «Архитектура ЭВМ»

на тему: **«Исследование работы арифметического сопроцессора»**

Выполнил: студент гр. ИТД-11

Фамилия И.О.

Принял: преподаватель

Ястребов А.А.

Гомель 2024

**Цель работы**: знакомство с работой арифметического сопроцессора. Исследование инструкций сопроцессора. Получение навыков работы с отладчиком.

**Задание:**

1. Согласно номера студента в журнале группы выбрать вариант задания и написать программу вычисления заданного выражения на языке Ассемблера.

2. Выполнить программу в пошаговом режиме в отладчике и заполнить таблицу, отображающую состояние регистров и флагов сопроцессора.

3. Прокомментировать полученный результат.

Вариант задания: вычислить пять значений функции  (*x* изменяется от 1 с шагом 4).

**Ход работы и результаты выполнения:**

Для вычисления значения функции при переменном ее аргументе разработана программа на языке Ассемблера. Вычисления выполнялись с использованием команд сопроцессора (библиотека *fpu*). Для хранения промежуточных результатов использован стек сопроцессора.

Результаты работы программы представлены на рисунке 1.

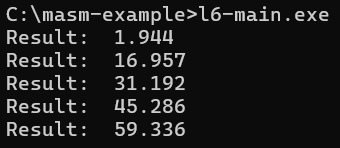


Рисунок 1 – Результат работы программы

Для отладки разработанной программы использован отладчик *OLLYDBG.* На рисунке 2 представлена часть графического интерфейса отладчика в процессе выполнения разработанной программы на этапе окончания работы цикла, вычисляющего значение функции при переменном ее аргументе.

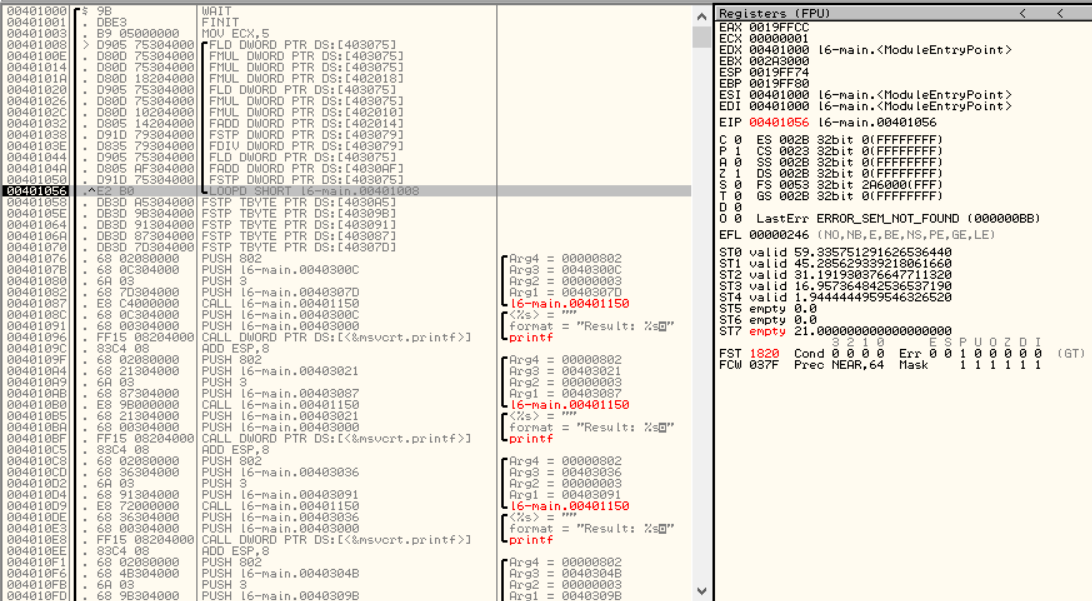


Рисунок 2 – Часть графического интерфейса отладчика *OLLYDBG* в процессе выполнения разработанной программы

В таблицах 1 и 2 представлено состояние стека сопроцессора на первой и последней итерации цикла соответственно.

Таблица 1 – Состояние стека сопроцессора на первой итерации цикла

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ST0** | **ST1** | **ST2** | **ST3** | **ST4** | **ST5** | **ST6** | **ST7** |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 7,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1,0 | 7,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 2,0 | 7,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 3,6 | 7,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 7,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 3,6 |
| 1,94 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 3,6 |
| 1,0 | 1,94 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 5,0 | 1,94 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1,94 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 5,0 |

Таблица 2 – Состояние стека сопроцессора на последней итерации цикла

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ST0** | **ST1** | **ST2** | **ST3** | **ST4** | **ST5** | **ST6** | **ST7** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 17,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 0,0 |
| 289,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 0,0 |

Продолжение таблицы 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 4913,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 0,0 |
| 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 0,0 |
| 17,0 | 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 |
| 289,0 | 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 |
| 578,0 | 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 |
| 579,6 | 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 |
| 34391,0 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 579,6 |
| 59,34 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 579,6 |
| 17,0 | 59,34 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 |
| 59,34 | 45,29 | 31,19 | 16,96 | 1,94 | 0,0 | 0,0 | 21,0 |

Текст программы представлен в приложении А.

**Вывод:** в результате выполнения лабораторной работы изучены принципы работы арифметического сопроцессора. Исследованы инструкций сопроцессора. Разработана программа на языке Ассемблера для вычисления значения функции при переменном ее аргументе. Выполнена отладка программы в отладчике *OLLYDBG*.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

.386

.model flat, stdcall

option casemap:none

include ../masm32/include/fpu.inc

include ../masm32/include/masm32rt.inc

includelib ../masm32/lib/fpu.lib

includelib ../masm32/lib/masm32rt.lib

.data

; для вывода строки с результатами

message db "Result: %s", 0Ah, 0

; для хранения результатов в строковом виде

res1 db 21 dup(0)

res2 db 21 dup(0)

res3 db 21 dup(0)

res4 db 21 dup(0)

res5 db 21 dup(0)

; переменная уравнения

x dd 1.0

; доп. переменная-буфер

tmp dd 0.0

; для хранения результатов

y1 dt 0.0

y2 dt 0.0

y3 dt 0.0

y4 dt 0.0

y5 dt 0.0

; величина шага

step dd 4.0

.const

; константы уравнения

op1 dd 2.0

op2 dd 1.6

op3 dd 7.0

.code

main:

; вычисление Yn = 7 \* x^3 / (2 \* x^2 + 1.6)

; нач. знач. x: 1

; шаг: 4

finit ; инициализация регистров FPU

mov ecx, 5 ; итератор цикла

cycle: ; метка начала цикла

; примечание: команды сопроцессора в качестве левого операнда

; используют значение на вершине стека сопроцессора (ST)

fld x ; загрузка значения x в вершину стека сопроцессора

fmul x ; x^2 (x тот, который находится в стеке)

fmul x ; x^3

fmul op3 ; 7 \* x^3

; в стеке хранится результат вычисления 7 \* x^3

fld x ; загрузка значения x в вершину стека сопроцессора

fmul x ; x^2

fmul op1 ; 2 \* x^2

fadd op2 ; 2 \* x^2 + 1.6

; в стеке теперь хранятся следующие величины:

; 2 \* x^2 + 1.6 (вершина стека)

; 7 \* x^3

fstp tmp ; выгрузка вершины стека в переменную-буфер

fdiv tmp ; 7 \* x^3 / (2 \* x^2 + 1.6)

fld x ; загрузка значения x в вершину стека сопроцессора

fadd step ; увеличение x на величину шага

fstp x ; выгрузка вершины стека в память

; результат вычисления выражения на каждой

; итерации цикла будет сохраняться на стеке

loop cycle ; если exc = exc - 1 != 0,

; то переход на след. итерацию цикла

; (изменение итератора цикла

; выполняется автоматически)

; выгрузка всего стека в память

fstp y5

fstp y4

fstp y3

fstp y2

fstp y1

; преобразование float в string и вывод

invoke FpuFLtoA, addr y1, 3, addr res1, SRC1\_REAL or SRC2\_DIMM

invoke crt\_printf, addr message, addr res1

invoke FpuFLtoA, addr y2, 3, addr res2, SRC1\_REAL or SRC2\_DIMM

invoke crt\_printf, addr message, addr res2

invoke FpuFLtoA, addr y3, 3, addr res3, SRC1\_REAL or SRC2\_DIMM

invoke crt\_printf, addr message, addr res3

invoke FpuFLtoA, addr y4, 3, addr res4, SRC1\_REAL or SRC2\_DIMM

invoke crt\_printf, addr message, addr res4

invoke FpuFLtoA, addr y5, 3, addr res5, SRC1\_REAL or SRC2\_DIMM

invoke crt\_printf, addr message, addr res5

invoke ExitProcess, 0

end main