Учреждение образования Республики Беларусь

«Гомельский государственный технический университет им. П. О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №6

по дисциплине «Архитектура ЭВМ»

по теме «Исследование работы арифметического сопроцессора»

Выполнил:

студент группы ИТП-21

Иванов Д. А.

Проверил:

преподаватель

Савельев В. А.

Гомель, 2022

**Цель:** знакомство с работой арифметического сопроцессора; исследование инструкций сопроцессора; получение навыков работы с отладчиком.

**Ход работы**

Задание

10. Вычислить 6 значений функции Yn = 256 / (3⋅x 2 + 31) (x изменяется от 2 с шагом 3).

Листинг:

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

; Программа на masm32 для Windows

; Исследование работы арифметического сопроцессора

; Вычислить 5 значений функции Yn = 4,3\*(x^2 + 1)

; (x изменяется с шагом 1,7)

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

.686 ; в программе будут использоваться

; команды процессора Pentium Pro

.model flat, stdcall ; модель памяти и соглашение

; о передаче параметров

option casemap :none ; включается чувствительность

; к регистру

; Библиотеки и подключаемые файлы проекта

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

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

include C:\masm32\include\user32.inc

include C:\masm32\include\fpu.inc

; содержит прототип функции FpuFLtoA

includelib C:\masm32\lib\kernel32.lib

includelib C:\masm32\lib\user32.lib

includelib C:\masm32\lib\fpu.lib

; Сегмент даных

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

.data ; инициализированные данные

MsgBoxTitle byte "Soproc x87 operations", 0

; заголовок окна

MsgBoxText db "Function calculation Yn = 256\*(3\*x^2 + 31),", 13,

"where x changes with step 3", 13, 13,

"y1="

res1 db 14 DUP(0), 10, 13 ; зарезервировать 14 байт для первого

; результата и поместить туда 0

db "y2="

res2 db 14 DUP(0), 10, 13

db "y3="

res3 db 14 DUP(0), 10, 13

db "y4="

res4 db 14 DUP(0), 10, 13

db "y5="

res5 db 14 DUP(0), 10, 13

db "y6="

res6 db 14 DUP(0), 10, 13

CrLf equ 0A0Dh

y1 TBYTE 0.0 ; тип 80 бит без знака (TBYTE = dt)

y2 dt 0.0

y3 dt 0.0

y4 dt 0.0

y5 dt 0.0

y6 dt 0.0

x DWORD 2.0 ; тип 32 бита без знака (DWORD = dd)

op1 dd 256.0

op2 dd 31.0

op3 dd 3.0

zero dd 0.0

step dd 3.0

.data? ; неинициализированные данные

.const ; константы

; Cегмент кода

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

.code

start: ; метка (точка входа в программу)

finit ; инициализация регистров FPU

; (CWR = 037Fh, SWR = 0h, TWR = FFFFh,

; DPR = 0h, IPR = 0h)

mov ecx, 6 ; счётчик X

m1: ; метка начала цикла

fld x ; x^2

fmul x

fmul op3 ; 3\*x^2

fadd op2 ; 3\*x^2+31

fdivr op1 ; 256/(x^2+1)

fld x ; увеличение X на величину шага

fadd step

fstp x

loop m1 ; если ecx = ecx - 1 != 0,

; переходим на m1

fstp y6 ; сохраняем стек в память

fstp y5

fstp y4

fstp y3

fstp y2

fstp y1

; преобразование результатов вычислений в массив символов

invoke FpuFLtoA, addr y1, 10, addr res1, SRC1\_REAL or SRC2\_DIMM

mov word ptr res1 + 13, CrLf

invoke FpuFLtoA, addr y2, 10, addr res2, SRC1\_REAL or SRC2\_DIMM

mov word ptr res2 + 13, CrLf

invoke FpuFLtoA, addr y3, 10, addr res3, SRC1\_REAL or SRC2\_DIMM

mov word ptr res3 + 13, CrLf

invoke FpuFLtoA, addr y4, 10, addr res4, SRC1\_REAL or SRC2\_DIMM

mov word ptr res4 + 13, CrLf

invoke FpuFLtoA, addr y5, 10, addr res5, SRC1\_REAL or SRC2\_DIMM

mov word ptr res5 + 13, CrLf

invoke FpuFLtoA, addr y6, 10, addr res6, SRC1\_REAL or SRC2\_DIMM

; вывод результатов вычислений

invoke MessageBox, NULL, addr MsgBoxText, addr MsgBoxTitle,

MB\_ICONINFORMATION

invoke ExitProcess, NULL; функция завершения с параметром NULL

end start ; окончание программы

На рисунке 1 представлен результат выполнения программы:

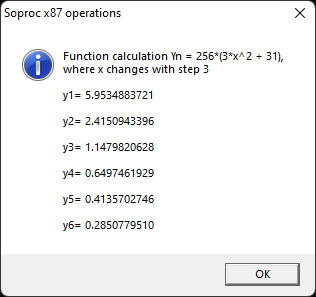


Рисунок 1 – Результат выполнения программы

2. Выполнить программу в пошаговом режиме. После выполнения каждого шага заносить данные в таблицу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ST0 | ST1 | ST2 | ST3 | ST4 | ST5 | ST6 | ST7 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 43.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 |
| 5.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 25.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 75.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 106.0 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 |
| 8.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 64.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 192.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 223.0 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 11.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 | 11.0 |
| 11.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 121.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 363.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 394.0 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 0.0 |
| 11.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 14.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 | 14.0 |
| 14.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 196.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 588.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 619.0 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 0.0 |
| 14.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 17.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 | 17.0 |
| 17.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 289.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 867.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 898.0 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 0.2850 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 0.0 |
| 17.0 | 0.2850 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 |
| 20.0 | 0.2850 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 |
| 0.2850 | 0.4135 | 0.6497 | 1.1479 | 2.4150 | 5.9534 | 0.0 | 20.0 |

4. Выводы по пункту 3: Изначально регистры сопроцессора не содержат никаких значений. После выполнения первой операции в регистр ST0 заносится значение 3.0. Все последующие команды умножения, сложения, округления и т. д. будут заносится в этот регистр. При сохранении значения с выталкиванием значение удаляется из стека и записывается в соответствующую переменную. Стек очищается при использовании команды finit.

**Вывод:** Арифметический сопроцессор – специализированный процессор, расширяющий возможности центрального процессора компьютерной системы. Предназначен для выполнения операция над числами с плавающей точкой.К основным операциям относятся fld (загрузка в стек), fst (сохранение в память), fstp (сохранение в память с выгрузкой), fadd (сложение вещественное), fsub (вычитание вещественное), fmul (умножение вещественное), fdiv (деление вещественное).