**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине «Архитектура ЭВМ»

на тему: «Исследование работы арифметического процессора»

Выполнил: студент гр. ИТП-21

Ковтун А.В.

Принял: преподаватель

Савельев В. А.

Гомель 2022

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

Задание: вычислить 5 значений функции Yn=130\*(2\*x2+7.3) (х = 1 с шагом 2)

**Листинг:**

.686 ; в программе будут использоваться

; команды процессора Pentium Pro

.model flat, stdcall ; модель памяти и соглашение

; о передаче параметров

option casemap :none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

include C:\masm32\include\user32.inc

include C:\masm32\include\fpu.inc

includelib C:\masm32\lib\kernel32.lib

includelib C:\masm32\lib\user32.lib

includelib C:\masm32\lib\fpu.lib

.

MsgBoxTitle byte "Операции в сопроцессоре x87", 0

MsgBoxText db "Вычисление функции Yn = 130\*(x^2 + 7.3),", 13,

"где x изменяется от 1 с шагом 2", 13, 13,

"y1="

res1 db 16 DUP(0), 10, 13 ; зарезервировать 14 байт для первого

; результата и поместить туда 0

db "y2="

res2 db 16 DUP(0), 10, 13

db "y3="

res3 db 16 DUP(0), 10, 13

db "y4="

res4 db 16 DUP(0), 10, 13

db "y5="

res5 db 16 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

x DWORD 1.0

op1 dd 130.0

op2 dd 7.3

zero dd 0.0

step dd 2.0

two dd 2.0

.data?

.const

.code

start:

finit ; инициализация регистров FPU

mov ecx, 5

m1:

fld x ; x

fmul x ; x^2

fmul two : 2\*x^2

fadd op2 ; 2\*x^2+7.3

fmul op1

fld x

fadd step

fstp x

loop m1

fstp y5

fstp y4

fstp y3

fstp y2

fstp y1

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

mov word ptr res1 + 16, CrLf

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

mov word ptr res2 + 16, CrLf

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

mov word ptr res3 + 16, CrLf

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

mov word ptr res4 + 16, CrLf

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

mov word ptr res5 + 16, CrLf

invoke MessageBox, NULL, addr MsgBoxText, addr MsgBoxTitle,

MB\_ICONINFORMATION

invoke ExitProcess, NULL

end start

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ST0 | ST1 | ST2 | ST3 | ST4 | ST5 | ST7 | ECX |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| **9.3** | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| **1209** | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| **1** | **1209** | 0 | 0 | 0 | 0 | 0 | 5 |
| **3** | **0** | 0 | 0 | 0 | 0 | 0 | 5 |
| **1209** | 0 | 0 | 0 | 0 | 0 | **3** | 5 |
| 1209 | 0 | 0 | 0 | 0 | 0 | 3 | **4** |
| **3** | **1209** | 0 | 0 | 0 | 0 | **0** | 4 |
| **9** | 1209 | 0 | 0 | 0 | 0 | 0 | 4 |
| **18** | 1209 | 0 | 0 | 0 | 0 | 0 | 4 |
| **25.3** | 1209 | 0 | 0 | 0 | 0 | 0 | 4 |
| **3289** | 1209 | 0 | 0 | 0 | 0 | 0 | 4 |
| **3** | **3289** | **1209** | 0 | 0 | 0 | 0 | 4 |
| **5** | 3289 | 1209 | 0 | 0 | 0 | 0 | 4 |
| **3289** | 1209 | 0 | 0 | 0 | 0 | **5** | 4 |
| **5** | 3289 | 1209 | 0 | 0 | 0 | **0** | **3** |
| **25** | 3289 | 1209 | 0 | 0 | 0 | 0 | 3 |
| **50** | 3289 | 1209 | 0 | 0 | 0 | 0 | 3 |
| **57.3** | 3289 | 1209 | 0 | 0 | 0 | 0 | 3 |
| **7449** | 3289 | 1209 | 0 | 0 | 0 | 0 | 3 |
| **5** | **7449** | **3289** | **1209** | 0 | 0 | 0 | 3 |
| **7** | 7449 | 3289 | 1209 | 0 | 0 | 0 | 3 |
| **7449** | **3289** | **1209** | 0 | 0 | 0 | **7** | 3 |
| 7449 | 3289 | 1209 | 0 | 0 | 0 | 7 | **2** |
| **7** | **7449** | **3289** | **1209** | 0 | 0 | **0** | 2 |
| **49** | 7449 | 3289 | 1209 | 0 | 0 | 0 | 2 |
| **98** | 7449 | 3289 | 1209 | 0 | 0 | 0 | 2 |
| **105.3** | 7449 | 3289 | 1209 | 0 | 0 | 0 | 2 |
| **13689** | 7449 | 3289 | 1209 | 0 | 0 | 0 | 2 |
| **7** | **11** | **7449** | **3289** | **1209** | 0 | 0 | 2 |
| **9** | 11 | 7449 | 3289 | 1209 | 0 | 0 | 2 |
| **13689** | **7449** | **3289** | **1209** | 0 | 0 | **9** | 2 |
| 13689 | 7449 | 3289 | 1209 | 0 | 0 | **0** | **1** |
| **9** | **13689** | **7449** | **3289** | **1209** | 0 | 0 | 1 |
| **81** | 13689 | 7449 | 3289 | 1209 | 0 | 0 | 1 |
| **162** | 13689 | 7449 | 3289 | 1209 | 0 | 0 | 1 |
| **169.3** | 13689 | 7449 | 3289 | 1209 | 0 | 0 | 1 |
| **22009** | 13689 | 7449 | 3289 | 1209 | 0 | 0 | 1 |
| **9** | **22009** | **13689** | **7449** | **3289** | **1209** | 0 | 1 |
| 11 | 2209 | 13689 | 7449 | 3289 | 0 | 0 | 1 |
| **22009** | **13689** | **7449** | **3289** | **1209** | 0 | **11** | 1 |
| 22009 | 13689 | 7449 | 3289 | 1209 | 0 | 11 | **0** |

Таблица 1 – Значения регистров общего назначения

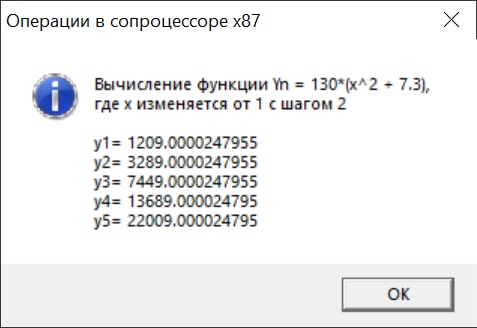


Рисунок 1 – Результат работы программы

**Вывод:** выполнять арифметические операции с помощью регистров общего назначения можно с помощью команд fadd – сложение, fmul - умножение, fld – загрузка в ST(0), fstp - извлечение из ST(0), fdiv – деление. С помощью функции FpuFLtoA можно конвертировать число с плавающей точкой в строку.