Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра ЭВМ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ 9

по предмету

«Архитектура компьютеров»

на тему:

«Блок вычислений с плавающей точкой FPU x87»

Вариант №12

Выполнил: Принял:

студент гр. КИУКИ-18-4 Голубничий Д.Ю.

Кравченко Н.С. Дзюбенко В.Ф.

Харьков 2020

1. Цель работы

Углубление и закрепление знаний архитектуры блока вычислений с плавающей точкой FPU x87; получение практических навыков по составлению программ для FPU x87 на языке ассемблера, их отладке и выполнению.

2. Постановка задачи

Выполнение работы состоит в разработке, отладке и выполнении ассемблерных программ в среде разработки Visual C++.

| №№  варианта | Комбинация вариантов заданий | | | |
| --- | --- | --- | --- | --- |
| № задания из пп.**9.5.1** | № задания из пп.**9.5.2** | № задания из пп.**9.5.3** | № задания из пп.**9.5.4** |
| 12 | 5 | 9 | 4 | 3 |

Таблица 1 – Схема вариантов заданий

#### 9.5.1 - Задание 5 (Исследование выполнения арифметических операций)

#### Вычислить 6 значений функции Y = 3000 / (x^2 + 3,6 \* x – 7,5), (x изменяется от 2 с шагом 2,7). Результат разместить в памяти.

#### 9.5.2 - Задание 9 (Исследование выполнения операций сравнения)

#### Задан массив с элементами b(i) = sin(2 \* i^2). Определить номер элемента, при котором сумма элементов превысит 3. Аргумент синуса задан в градусах.

#### 9.5.3 - Задание 4 (Исследование выполнения команд с обратными тригонометрическими функциями)

#### Вычислить одно значение функции Y = 2 \* arcSin ((A^2 + B^2 )/3). Параметры аргументов выбирать с учетом области определения. Результат перевести в градусы.

#### 9.5.4 - Задание 3 (Исследование выполнения команд с логарифмическими и показательными функциями)

#### Вычислить 7 значений функции Y = 6 \* lg (cos x), x изменяется в градусах от 8 с шагом 12.

3. Теоретическая часть

Под регистровой моделью понимается описание набора регистров процессора, их разрядности, способа организации, методов доступа, основных особенностей и пр.

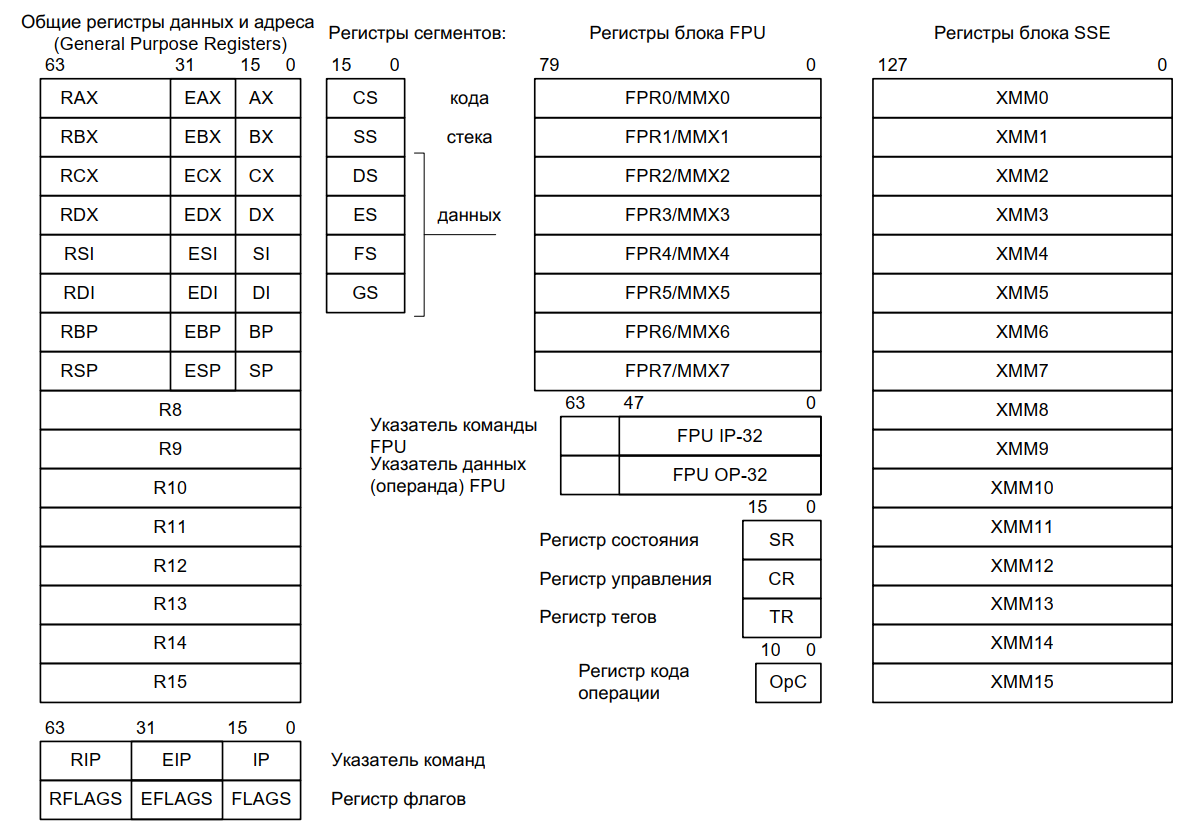


Рисунок 1 – Регистровая модель микропроцессоров

На рисунке 1 показаны регистры, доступные программисту при использовании машинных кодов (или языка ассемблера).

Наиболее универсальными являются регистры общего назначения (РОН), которые в 8086 обозначаются с использованием символов латинского алфавита A, B, C, D. Регистры имеют 16 разрядов, их младшая часть дополнительно обозначается символом L (Low), а старшая ­– H (High). Весь регистр обозначается дополнительным символом X. Таким образом, AX соответствует всем 16 разрядам первого регистра общего назначения, а доступ к двум его 8-разрядным половинам производится по именам AH, AL.

Для операций с объектами в памяти используются индексные регистры. Их имена являются сокращениями от SourceIndex (SI) и DestinationIndex (DI), т.е. «индекс источника» и «индекс приемника» соответственно.

Также в 8086 присутствуют указатели на области памяти стека и стекового кадра SP и BP, сводный регистр флагов (особых признаков результата последней операции) и сегментные регистры.

Система команд сопроцессора включает около 80 машинных команд

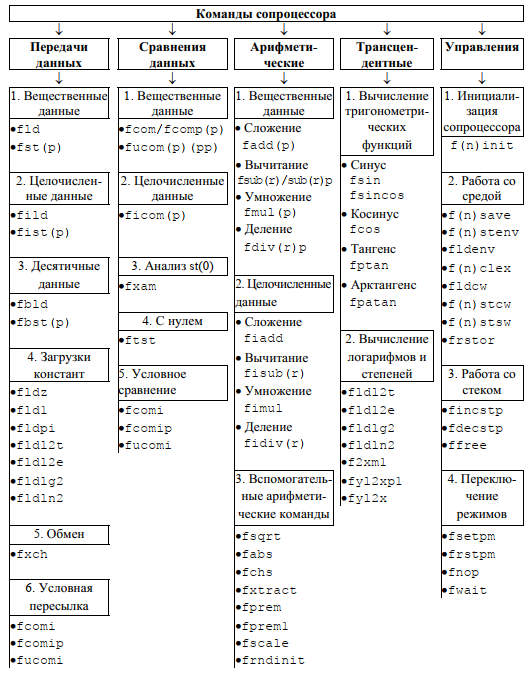


Рисунок 2 – Функциональная классификация команд сопроцессора

Мнемоническое обозначение команд сопроцессора характеризует особенности их работы:

1. Все мнемонические обозначения начинаются с символа f (float).

2. Вторая буква мнемонического обозначения определяет тип  
операнда в памяти, с которым работает команда:

* i – целое двоичное число;
* b – целое десятичное число;
* отсутствие буквы – вещественное число.

3. Последняя буква p в мнемоническом обозначении команды означает, что последним действием команды обязательно является извлечение операнда из стека.

4. Последняя или предпоследняя буква r (reversed) в мнемоническом обозначении команды означает реверсивное следование операндов при выполнении команд вычитания и деления, так как для них важен порядок следования операндов.  
Система команд сопроцессора отличается большой гибкостью в  
выборе вариантов задания команд, реализующих определенную операцию, и их операндов.

Минимальная длина команды сопроцессора – 2 байта.

Методика написания программ для сопроцессора имеет свои особенности. Главная причина – в стековой организации сопроцессора.

При разработке программ необходимо учитывать:

* ограниченность глубины стека сопроцессора;
* несовпадение форматов операндов;
* отсутствие поддержки на уровне команд сопроцессора некоторых операций, таких как возведение в степень, вычисление тригонометрических функций.

4. Экспериментально-практическая часть

#### 9.5.1 - Задание 5 (Исследование выполнения арифметических операций)

#### Вычислить 6 значений функции Y = 3000 / (x^2 + 3,6 \* x – 7,5), (x изменяется от 2 с шагом 2,7). Результат разместить в памяти.

Теоретические расчеты:

*x = 2 Y = 810.811*

*x = 4.7 Y = 95.2079*

*x = 7.4 Y = 40.5954*

*x = 10.1 Y = 22.9235*

*x = 12.8 Y = 14.8207*

*x = 15.5 Y = 10.3968*

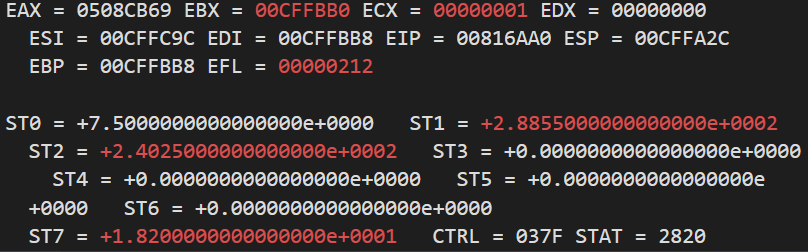


Рисунок 3 – Окно отладки «Registers»

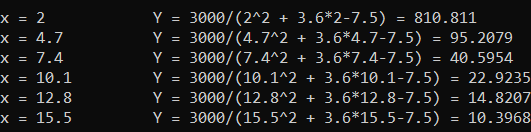


Рисунок 4 – Результат работы программы

Код программы:

double REZ[6];

double X = 2;

double A = 3.6;

double B = 7.5;

double C = 3000;

double D = 2.7;

\_asm

{

lea EBX, REZ ; загрузка адреса результатов в регистр EBX

mov ECX, 6 ; счетчик количества повторений цикла

m1 :

finit ; очистка регистров FPU

fld X ; загрузка аргумента в стек FPU

fmul X

fld A

fmul X

fadd ST, ST(1)

fld B

fsub ST(1), ST

fld C

fdiv ST(0), ST(2)

fstp qword ptr[EBX]

fld X ; загрузить в FPU X

fadd D

fstp X

add EBX, 8 ; увеличение адреса результатов

loop m1

}

for (auto element : REZ)

cout <<"Y = " << element << endl;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Стек FPU | Команды FPU | | | | | | | | | | | | | |
| **finit** | **fld X** | **fmul X** | **fld A** | **fmul X** | **fadd** | **fld B** | **fsub** | **fld C** | **fdiv** | **fstp** | **fld X** | **fadd D** | **fstp X** |
| ST(0) | 0.0e0 | 2.0e0 | 4.0e0 | 3.6e0 | 7.2e0 | 1.12e1 | 7.5e0 | 7.5e0 | 3.3e0 | 8.108e2 | 7.5e0 | 2.0e0 | 4.7e0 | 7.5e0 |
| ST(1) | 0.0e0 | 0.0e0 | 0.0e0 | 4.0e0 | 4.0e0 | 4.0e0 | 1.12e1 | 3.7e1 | 7.5e0 | 7.5e0 | 3.7e1 | 7.5e0 | 7.5e0 | 3.7e1 |
| ST(2) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 4.0e0 | 4.0e0 | 3.7e1 | 3.7e1 | 4.0e0 | 3.7e1 | 3.7e1 | 4.0e0 |
| ST(3) | 00e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 4.0e0 | 4.0e0 | 0.0e0 | 4.0e0 | 4.0e0 | 0.0e0 |
| ST(4) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(5) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(6) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(7) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 8.108e2 | 0.0e0 | 0.0e0 | 4.7e0 |
| Тэги | FFFF | 3FFF | 3FFF | 0FFF | 0FFF | 00FF | 03FF | 03FF | 00FF | 00FF | 03FF | 00FF | 00FF | 03FF |

#### 9.5.2 - Задание 9 (Исследование выполнения операций сравнения)

#### Задан массив с элементами b(i) = sin(2 \* i^2). Определить номер элемента, при котором сумма элементов превысит 3. Аргумент синуса задан в градусах.

Теоретические расчеты:

*i = 1: sin(2°\*1^2) = 0.0348995 sum = 0.0348995*

*i = 2: sin(2°\*2^2) = 0.139173 sum = 0.174073*

*i = 3: sin(2°\*3^2) = 0.309017 sum = 0.48309*

*i = 4: sin(2°\*4^2) = 0.52992 sum = 1.01301*

*i = 5: sin(2°\*5^2) = 0.766045 sum = 1.77905*

*i = 6: sin(2°\*6^2) = 0.951057 sum = 2.73011*

*i = 7: sin(2°\*7^2) = 0.990268 sum = 3.72038*

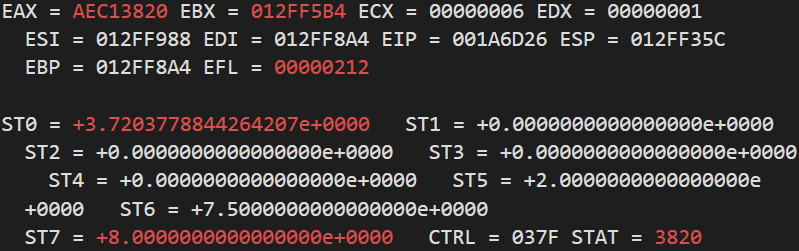


Рисунок 5 – Окно отладки «Registers»



Рисунок 6 – Результат работы программы

Код программы:

double REZ[10];

double X = 1;

double A = 2;

double B = 3;

double N = 1;

double sum = 0;

\_asm

{

lea EBX, REZ ; загрузка адреса результатов в регистр EBX

mov ECX, 6 ; счетчик количества повторений цикла

m1 :

finit ; очистка регистров FPU

fld X ; загрузка аргумента в стек FPU

fmul X

fmul A

fsin

fst qword ptr[EBX]

fld sum

fadd ; накопление суммы последовательности

fst sum

fld X ; загрузить в FPU X

fadd N

fstp X

add EBX, 8 ; увеличение адреса результатов

fcom B ; сравнение суммы с 3

fstsw AX ; сохранение копии регистра состояния SW в регистре AX(CPU)

sahf ; сохранение старшего байта АХ в EFLAGS

jb m1 ; переход, если сумма меньше 3

}

cout << "i = " << X - 1 << " \t SUM(" << X - 1 <<") = " << sum << endl;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Стек FPU | Команды FPU | | | | | | | | | | | | | |
| **finit** | **fldpi** | **fmul X** | **fidiv D** | **fmul X** | **fmul A** | **fsin** | **fst** | **fld sum** | **fadd** | **fst** | **fld X** | **fadd N** | **fstp X** |
| ST(0) | 0.0e0 | 3.14e0 | 3.14e0 | 1.745e-2 | 1.745e-2 | 3.4906e-2 | 3.4899e-2 | 3.4899e-2 | 0.0e0 | 3.4899e-2 | 3.4899e-2 | 1.0e0 | 2.0e0 | 3.4899e-2 |
| ST(1) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 3.4899e-2 | 0.0e0 | 0.0e0 | 3.4899e-2 | 0.0e0 | 0.0e0 |
| ST(2) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(3) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(4) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(5) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(6) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(7) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 2.0e0 |
| Тэги | FFFF | 3FFF | 3FFF | 3FFF | 3FFF | 3FFF | 3FFF | 3FFF | 1FFF | 3FFF | 3FFF | 0FFF | 0FFF | 3FFF |

#### 9.5.3 - Задание 4 (Исследование выполнения команд с обратными тригонометрическими функциями)

#### Вычислить одно значение функции *Y = 2 \* arcSin ((A^2 + B^2 )/3*. Параметры аргументов выбирать с учетом области определения. Результат перевести в градусы.

Теоретические расчеты:

*A = 0.5, B = 0.5;*

*Y = 2\*arcSin(z) = 19.1881°*

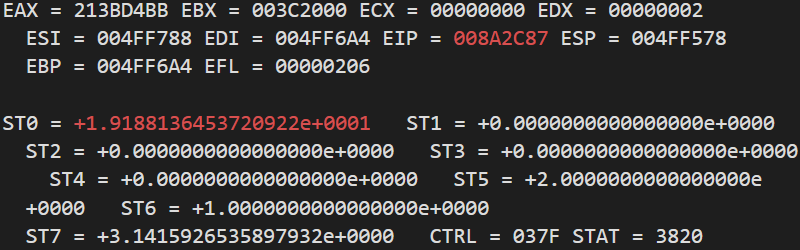


Рисунок 7 – Окно отладки «Registers»



Рисунок 8 – Результат работы программы

Код программы:

double A = 0.5;

double B = 0.5;

double C = 3;

int E = 2;

long D = 180;

double res;

\_asm

{

finit ; очистка регистров FPU

fld A

fmul A

fld B

fmul B

fadd

fdiv C

fld ST ; копирование вершины стека

fmul ST, ST ; возведение в квадрат

fld1 ; загрузка 1 в стек FPU

fsubr ; вычитание с реверсом : 1 – z ^ 2

fsqrt ; корень квадратный

fpatan ; вычисление арктангенса

fldpi ; загрузка в стек пи

fdiv ; перевод в градусы - деление на пи

fimul D ; перевод в градусы - умножение на 180

fimul E

fstp res ; сохранение результата в ячейке памяти

}

cout << "A = "<< A <<", B = " << B <<"\tY = " << res << "\xB0" << endl;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Стек FPU | Команды FPU | | | | | | | | | | | | | | | | | |
| **finit** | **fld A** | **fmul A** | **fld B** | **fmul B** | **fadd** | **fdiv C** | **fld ST** | **fmul ST** | **fld1** | **fsubr** | **fsqrt** | **fpatan** | **fldpi** | **fdiv** | **fimul D** | **fimul E** | **fstp** |
| ST(0) | 0.0e0 | 5.0e-1 | 2.5e-1 | 5.0e-1 | 2.5e-1 | 5.0e-1 | 1.6e-1 | 1.6e-1 | 2.77e-2 | 1.0e0 | 9.722e-1 | 9.86e-1 | 1.67e-1 | 3.14e0 | 5.33e-2 | 9.59e0 | 1.919e1 | 0.0e0 |
| ST(1) | 0.0e0 | 0.0e0 | 0.0e0 | 2.5e-1 | 0.0e0 | 0.0e0 | 0.0e0 | 1.6e-1 | 1.6e-1 | 2.77e-2 | 1.6e-1 | 0.0e0 | 0.0e0 | 1.67e-1 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(2) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 1.6e-1 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(3) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(4) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(5) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 1.0e0 |
| ST(6) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 1.0e0 | 0.0e0 | 0.0e0 | 1.0e0 | 1.0e0 | 3.14e0 |
| ST(7) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 2.5e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 1.0e0 | 0.0e0 | 9.86e-1 | 1.0e0 | 3.14e0 | 3.14e0 | 3.14e0 | 1.919e1 |
| Тэги | FFFF | 3FFF | 3FFF | 0FFF | 0FFF | 3FFF | 3FFF | 0FFF | 0FFF | 03FF | 0FFF | 0FFF | 3FFF | 0FFF | 3FFF | 3FFF | 3FFF | FFFF |

#### 9.5.4 - Задание 3 (Исследование выполнения команд с логарифмическими и показательными функциями)

#### Вычислить 7 значений функции Y = 6 \* lg (cos x), x изменяется в градусах от 8 с шагом 12.

Теоретические расчеты:

*x = 8° 6\*lg(cos x) = -0.0254833*

*x = 20° 6\*lg(cos x) = -0.162085*

*x = 32° 6\*lg(cos x) = -0.429477*

*x = 44° 6\*lg(cos x) = -0.858395*

*x = 56° 6\*lg(cos x) = -1.51463*

*x = 68° 6\*lg(cos x) = -2.55855*

*x = 80° 6\*lg(cos x) = -4.56198*

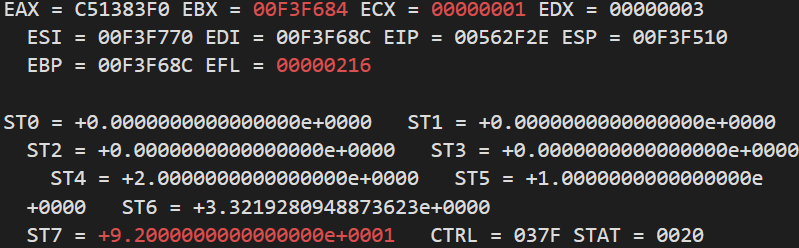


Рисунок 9 – Окно отладки «Registers»

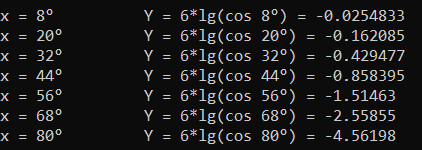


Рисунок 10 – Результат работы программы

Код программы:

double REZ[7];

int A = 6;

double X = 8;

long D = 180;

double C = 12;

\_asm

{

lea EBX, REZ ; загрузка адреса результатов в регистр EBX

mov ECX, 7 ; счетчик количества повторений цикла

m1 :

finit ; очистка регистров FPU

FLD1 ; загрузка константы "1" в стек

Fldpi ; загрузка константы «пи» в стек FPU

fmul X ; умножение числа «пи» на аргумент

fidiv D ; деление аргумента на 180

fcos ; вычисление cos(x)

FYL2X ; st = log2(X)

FLDL2T ; st = log2(10)

FDIVP st(1), st(0) ; st = log10(X)

fimul A ; умножение на 6

fstp qword ptr[EBX] ; сохранение значения в массив

fld X ; загрузить в FPU X

fadd C ; инкриментирование

fstp X ; инкриментирование

add EBX, 8 ; увеличение адреса результатов

loop m1

}

for (auto element : REZ)

cout << "Y = " << element << endl;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Стек FPU | Команды FPU | | | | | | | | | | | | | |
| **finit** | **fld1** | **fldpi** | **fmul X** | **fidiv D** | **fcos** | **FYL2X** | **FLDL2T** | **FDIVP** | **fimul** | **fstp** | **fld X** | **fadd C** | **fstp X** |
| ST(0) | 0.0e0 | 1.0e0 | 3.14e0 | 2.51e0 | 1.39e0 | 9.9e-1 | -1.410 e-2 | 3.322e0 | -4.247 0 e-3 | -2.548 e-2 | 0.0e0 | 8.0e0 | 2.0e0 | 2.0e0 |
| ST(1) | 0.0e0 | 0.0e0 | 1.0e0 | 1.0e0 | 1.0e0 | 1.0e0 | 0.0e0 | 1.410 e-2 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(2) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(3) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(4) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(5) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 |
| ST(6) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 3.322e0 | 0.0e0 | 0.0e0 | 3.322e0 |
| ST(7) | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 0.0e0 | 9.9e-1 | 0.0e0 | 3.322e0 | 3.322e0 | -2.548 e-2 | 3.322e0 | 3.322e0 | 2.0e0 |
| Тэги | FFFF | 3FFF | 0FFF | 0FFF | 0FFF | 0FFF | 3FFF | 0FFF | 3FFF | 3FFF | FFFF | 3FFF | 3FFF | FFFF |

5. Анализ результатов и выводы по работе

В ходе выполнения лабораторной работы ознакомился с системой команд сопроцессора и получил практические навыки по составлению ассемблерных вставок, встраиваемых в тело программы на языке C++ и отладке приложения.

Выполнение заданий проходило в среде разработки Visual C++, которая содержит удобный функционал режима отладки для контроля значений переменных и регистров.

В процессе работы были решены 4 задачи, направленные на вычисление с плавающей точкой FPU x87, арифметических операций, *sin(x), cos(x), arcSin(x), lg(x),* использования команд условных переходов, сохранения результатов в регистр или память и перевод из радианов в градусы.

В результате исполнения работы получен опыт работы с вещественными числами и использованы команды передачи и сравнения данных, арифметические, трансцендентные и команды управления.

Результаты выполнение не отличаются от теоретических вычислений, так как в работе использовался формат вещественных данных.