# Лабораторная работа 9.Блок вычислений с плавающей точкой FPU x87

## 9.1. 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.

## 9.2 Самостоятельная работа студентов

Перед выполнением лабораторной работы студенты должны изучить регистровую модель FPU x87, форматы данных и особые случаи представления данных, ознакомиться с системой ассемблерных команд FPU ([приложение Б](file:///G:\Users\Dima\0501-KI%D0%B7%20-%2013%2014%20%D0%90%D0%9A\2020%20%D0%BE%D1%81%D0%B5%D0%BD%D1%8C\CA_appendix.pdf#%5B%7B%22num%22%3A64%2C%22gen%22%3A0%7D%2C%7B%22name%22%3A%22XYZ%22%7D%2C54%2C790%2C0%5D)). Выполнение заданий данной лабораторной работы предполагает наличие у студентов знаний и навыков, полученных и закрепленных в ходе выполнения [лабораторной работы 8](file:///G:\Users\Dima\0501-KI%D0%B7%20-%2013%2014%20%D0%90%D0%9A\2020%20%D0%BE%D1%81%D0%B5%D0%BD%D1%8C\Lab_08\index.html) («Архитектура прикладного уровня CPU x86-64»).

При подготовке к выполнению практических заданий следует обратить особое внимание на группы команд передачи данных, арифметические команды и команды сравнения.

С целью получения практических навыков по реализации операций FPU необходимо составить программы в соответствии с вариантом заданий (п. 9.5) и исследовать функционирование FPU, выполняя разработанную программу в пошаговом режиме (под управлением отладчика).

## 9.3 Примеры выполнения практических заданий

9.3.1 Вычислить значение функции S = (1/2)\*A\*B\*sin(α).

void main () // начало программы на языке С++

{ // описание операндов в памяти

long A=20, B=30, M2=2;

float ALPHA=0.7, Y;

\_asm

{ ; начало ассемблерной вставки

finit ; очистка регистров FPU

fld ALPHA ; загрузка аргумента в стек FPU

fsin ; вычисление синуса

fimul A ; умножение ST(0) на целое A

fimul B ; умножение ST(0) на целое B

fidiv M2 ; деление ST(0) на целое M2

fstp Y ; сохранение результата в ячейке памяти

} // окончание ассемблерной вставки

} // окончание программы на языке С++

9.3.2 Вычислить площадь треугольника по формуле Герона , где p – полупериметр треугольника.

void main () // начало программы на языке С++

{

long A=20, B=30, C=40, M2=2;// описание операндов в памяти

float RES; // ячейка памяти для результата

\_asm

{ ; начало ассемблерной вставки

finit ; очистка регистров FPU

fild A ;

fild B ;

fild C ;

fld ST ; C -> ST(0)

fadd ST,ST(2) ; C + B

fadd ST,ST(3) ; C + B + A

fidiv M2 ; (C + B + A) / 2 -> ST(0) (p)

fsubr ST(1),ST ; (p-C)-> ST(1)

fsubr ST(2),ST ; (p-B)-> ST(2)

fsubr ST(3),ST ; (p-A)-> ST(3)

fmul ; p\*(p-C)

fmul ; p\*(p-C)\*(p-B)

fmul ; p\*(p-C)\*(p-B)\*(p-A)

fsqrt ; (p\*(p-C)\*(p-B)\*(p-A))^(1/2)

fstp RES ; сохранение результата и очистка ST(0)

} // окончание ассемблерной вставки

} // окончание программы на языке С++

9.3.3 Определить номер (n) элемента последовательности an = 5,3^n + 5\*n, при котором сумма элементов превысит 20000.

void main () // начало программы на языке С++

{

float A=5.3; // описание операндов в памяти

long B=5, C=20000, N=0;

\_asm

{ ; начало ассемблерной вставки

finit ; очистка регистров FPU

fld1 ; подготовка к вычислению степенной функции

fldz ; подготовка к накоплению суммы прогрессии

m1: inc N ; наращивание аргумента

fld A ; загрузка 5,3 в стек FPU

fmulp ST(2),ST ; вычисление степенной функции 5,3^n

fild B

fimul N

fadd ST,ST(2) ; вычисление элемента последовательности

fadd ; накопление суммы последовательности

ficom C ; сравнение суммы с 20000

fstsw AX ; сохранение копии регистра состояния SW

; в регистре AX (CPU)

sahf ; сохранение старшего байта АХ в EFLAGS

jc m1 ; переход, если сумма меньше 20000

} // окончание ассемблерной вставки

} // окончание программы на языке С++

9.3.4 Вычислить функцию Y = 5 \* arcSin ((tg 60°)2/ 7). Результат перевести в градусы.

void main () // начало программы на языке С++

{

long A=60, B=5, C=7, D=180; // описание операндов в памяти

float Y;

\_asm{ ; начало ассемблерной вставки

finit ; очистка регистров FPU

fldpi ; загрузка константы «пи» в стек FPU

fimul A ; умножение числа «пи» на аргумент

fidiv D ; деление аргумента на 180

fptan ; вычисление частичного тангенса

fdiv ; нахождение тангенса

fmul ST,ST ; возведение в квадрат

fidiv C ; вычислен «z» – аргумент ArcSin

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

fimul B ; умножение на константу

fldpi ;

fdiv ; перевод из радианов – в градусы

fimul D ;

fstp Y ; сохранение результата в ячейке памяти

} // окончание ассемблерной вставки

} // окончание программы на языке С++

9.3.5 Возведение основания 2 в произвольную степень.

; фрагмент ассемблерной вставки:

; - аргумент z находится на вершине стека

; - результат 2^z будет помещен на вершину стека

fld ST(0) ; копирование вершины стека

frndint ; округление ST(0) до целого

fsub ST(1),ST(0) ; выделение дробной части в ST(1)

fxch ; обмен регистров ST(0) и ST(1)

f2xm1 ; возведение 2 в дробную степень (минус 1)

fld1 ; загрузка 1 в стек FPU

fadd ; добавление единицы

fscale ; возведение 2 в целую степень и умножение

fstp ST(1) ; удаление регистра и сдвиг вершины,

; результат находится на вершине стека

## 9.4 Порядок выполнения работы

Выполнение работы состоит в разработке, отладке и выполнении ассемблерных программ для FPU x87 в среде разработки Visual C++. Основные этапы работы в среде Visual C++ рассмотрены в [лабораторной работе 8 (п. 8.4)](file:///G:\Users\Dima\0501-KI%D0%B7%20-%2013%2014%20%D0%90%D0%9A\2020%20%D0%BE%D1%81%D0%B5%D0%BD%D1%8C\Lab_08\index.html#p2_4).

Программы выполняются в пошаговом режиме (под управлением отладчика) с одновременным контролем промежуточных результатов в регистровом стеке FPU. С этой целью, в окне отладки «Registers» необходимо выбрать для отображения (в дополнение к компонентам программной модели CPU) компонент «Floating Point» (рис. 9.1).

Для задания 1 необходимо построить диаграмму стека FPU на какой-либо итерации цикла выполнения программы по образцу, приведенному на рис. 9.2. Диаграмма должна отражать изменения состояния стека после каждой команды FPU. Таким образом, в нее необходимо включить команды FPU в порядке их появления в программе. Рекомендуется указывать значение мантиссы с точностью 3 разрядов после десятичной точки, значение порядка – 1 знак.

Результаты вычислений, сохраняемые в памяти, рекомендуется контролировать посредством элементов «Memory n» меню отладки «Memory» (рисунок 9.3; для представления результатов в окне «Memory 2» выбран формат «32-bit Floating Point»).

Исходные данные и результаты выполнения программ представить в десятичном (с плавающей точкой) и шестнадцатеричном форматах.

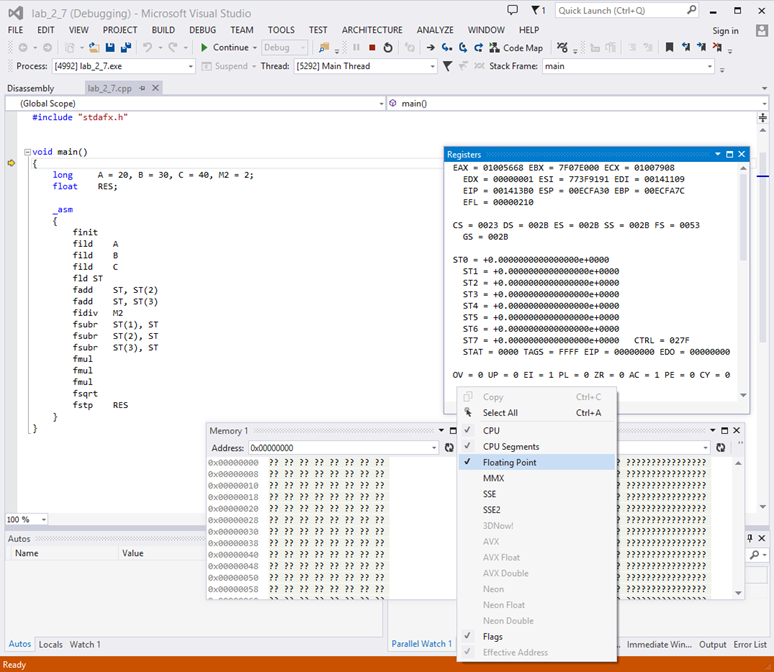


Рисунок 9.1 – Окно отладки «Registers»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Стек FPU** | **Команды FPU** | | | | | |
| **finit** | **fild A** | **fild B** | **fild C** | **fld ST** | **fadd ST,ST(2)** |
| ST(0) | 0.000e0 | 2.000e1 | 3.000e1 | 4.000e1 | 4.000e1 | 7.000e1 |
| ST(1) | 0.000e0 | 0.000e0 | 2.000e1 | 3.000e1 | 4.000e1 | 4.000e1 |
| ST(2) | 0.000e0 | 0.000e0 | 0.000e0 | 2.000e1 | 3.000e1 | 3.000e1 |
| ST(3) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 2.000e1 | 2.000e1 |
| ST(4) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(5) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(6) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(7) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| Тэги | FFFF | 3FFF | 0FFF | 03FF | 00FF | 00FF |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Стек FPU** | **Команды FPU** | | | | | |
| **fadd ST,ST(3)** | **fidiv M2** | **fsubr ST(1),ST** | **fsubr ST(2),ST** | **fsubr ST(3),ST** | **fmul** |
| ST(0) | 9.000e1 | 4.500e1 | 4.500e1 | 4.500e1 | 4.500e1 | 2.250e2 |
| ST(1) | 4.000e1 | 4.000e1 | 5.000e1 | 5.000e1 | 5.000e1 | 1.500e1 |
| ST(2) | 3.000e1 | 3.000e1 | 3.000e1 | 1.500e1 | 1.500e1 | 2.500e1 |
| ST(3) | 2.000e1 | 2.000e1 | 2.000e1 | 2.000e1 | 2.500e1 | 0.000e0 |
| ST(4) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(5) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(6) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(7) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 | 4.500e1 |
| Тэги | 00FF | 00FF | 00FF | 00FF | 00FF | 03FF |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Стек FPU** | **Команды FPU** | | | |
| **fmul** | **fmul** | **fsqrt** | **fstp RES** |
| ST(0) | 3.375e3 | 8.438e4 | 2.905e2 | 0.000e0 |
| ST(1) | 2.500e1 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(2) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(3) | 0.000e0 | 0.000e0 | 0.000e0 | 0.000e0 |
| ST(4) | 0.000e0 | 0.000e0 | 0.000e0 | 4.500e1 |
| ST(5) | 0.000e0 | 4.500e1 | 4.500e1 | 2.250e2 |
| ST(6) | 4.500e1 | 2.250e2 | 2.250e2 | 3.375e3 |
| ST(7) | 2.250e2 | 3.375e3 | 3.375e3 | 2.905e2 |
| Тэги | 0FFF | 3FFF | 3FFF | FFFF |

Рисунок 9.2 – Изменение стека FPU

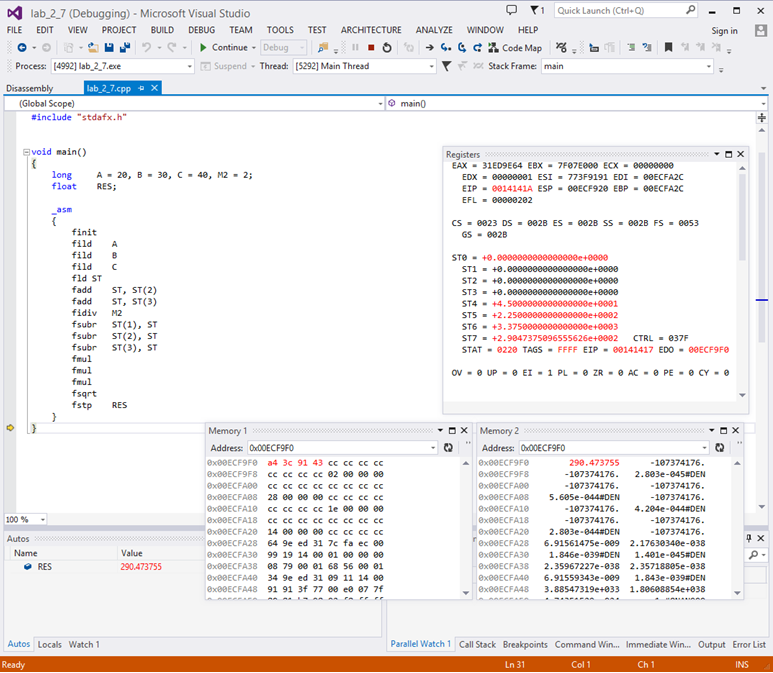


Рисунок 9.3 – Выбор областей памяти

## 9.5 Варианты индивидуальных заданий

#### 9.5.1 Исследование выполнения арифметических операций.

Вариант 1. Вычислить 7 значений функции Y = 2500 / (2 \* х^2 + 3,7) (х изменяется от 3 с шагом 2,5). Результат округлить до целого и разместить в памяти.

Вариант 2. Вычислить 6 значений функции Y = 3,5 \* x^2 + 7,2 \* x (х изменяется от 4,5 с шагом 3). Результат округлить до целого и разместить в памяти.

Вариант 3. Вычислить 5 значений функции Y = (x - 7) / [(x + 20)^(1/2)] (x изменяется от 3 с шагом 2,5). Результат разместить в памяти.

Вариант 4. Вычислить 6 значений функции Y = (14 \* x + 5,6)^(1/2) (x изменяется от 6 с шагом 3). Результат разместить в памяти.

Вариант 5. 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.

Вариант 6. Вычислить 5 значений суммы элементов последовательности an = 2,5 \* n^2 + 5,3 (для n от 4 с шагом 1). Результат разместить в памяти в целом формате.

Вариант 7. Вычислить 6 значений функции Y = (20 \* x) / (5 \* x^2 – 8,5) (x изменяется от 1 с шагом 4). Результат разместить в памяти.

Вариант 8. Вычислить 6 значений суммы элементов последовательности an = (4,5^n) / (n + 5) (для n от 1 с шагом 1). Результат округлить до целого и разместить в памяти.

Вариант 9. Вычислить 5 значений суммы суммы элементов последовательности an = 418 / (2 \* n^2 + 7,3), (для n от 1 с шагом 1). Результат разместить в памяти.

Вариант 10. Вычислить 6 значений функции Y = 1024 / (3,2 \* x^2 – 25), (x изменяется от 2 с шагом 2). Результат разместить в памяти.

#### 9.5.2 Исследование выполнения операций сравнения.

Вариант 1. Определить номер (n) элемента последовательности an = 3,3^n + 5, при котором сумма элементов последовательности превысит 15000.

Вариант 2. Найти целое значение аргумента, при котором функция Y = 20 / (x^2 + 2,5^x) станет меньше 0,2.

Вариант 3. Определить номер (n) элемента прогрессии Y = (2,5 + 3 \* n)^(1/2), при котором сумма элементов превысит 100.

Вариант 4. Найти целое значение аргумента х, при котором функция Y = 15 / (x^2 + 3,7) станет меньше 0,1 (для х от 1 с шагом 1).

Вариант 5. Определить номер (n) элемента последовательности an = 2,5 \* n^2 + 7,3, при котором сумма элементов последовательности превысит 1000.

Вариант 6. Найти целое значение аргумента х, при котором функция Y = (15 \* x + 32 \* x + 40)^(1/2) превысит 30.

Вариант 7. Задан массив с элементами a(i) = sin (5 \* i). Определить номер элемента массива, при котором сумма элементов превысит 3. Аргумент синуса задан в градусах.

Вариант 8. Найти целое значение аргумента, при котором функция Y = (2,5 \* x + 10)^(1/2) превысит 100.

Вариант 9. Задан массив с элементами b(i) = sin(2 \* i^2). Определить номер элемента, при котором сумма элементов превысит 3. Аргумент синуса задан в градусах.

Вариант 10. Найти целое значение аргумента, при котором функция Y = (5,6^x) / (3 \* x^2) превысит 200.

#### 9.5.3 Исследование выполнения команд с обратными тригонометрическими функциями.

Вычислить одно значение функции Y. Параметры аргументов выбирать с учетом области определения. Результат перевести в градусы.

Вариант 1. Y = 3 \* arcSin (2 \* cos(70°))^2.

Вариант 2. Y = (1/4) \* arcCosec ((5 \* A + B^2)/2).

Вариант 3. Y = (1/2) \* arcCosec (3 \* A^2 + 4 \* B).

Вариант 4. Y = 2 \* arcSin ((A^2 + B^2 )/3).

Вариант 5. Y = 3 \* arcCosec ( 4 \* A + tg 40°).

Вариант 6. Y = 5 \* arcSin (3 \* cos 25° \* sin 25°).

Вариант 7. Y = 3 \* arcCos (2 \* A^2 - B).

Вариант 8. Y = 4 \* arcCos (2 \* sin 30° \* cos 30°).

Вариант 9. Y = 5 \* arcSec (3 \* (tg 70°)^2).

Вариант 10. Y = (1/3) \* arcSin (3 \* A + sin 20°).

#### 9.5.4 Исследование выполнения команд с логарифмическими и показательными функциями.

Вариант 1. Вычислить 7 значений функции Y = 5(sin x), x изменяется в градусах от 10 с шагом 8.

Вариант 2. Вычислить 6 значений функции Y = 7 \* ln [x^2 + x^(1/2)], x изменяется от 2 с шагом 3.

Вариант 3. Вычислить 7 значений функции Y = 6 \* lg (cos x), x изменяется в градусах от 8 с шагом 12.

Вариант 4. Вычислить 6 значений функции Y = 5 \* ln (sin x), x изменяется в градусах от 10 с шагом 15.

Вариант 5. Вычислить 5 значений функции Y = 7^x, x изменяется от 0,5 с шагом 0,2.

Вариант 6. Вычислить 7 значений функции Y = 4 \* lg (tg x), x изменяется в градусах от 15 с шагом 10.

Вариант 7. Вычислить 6 значений функции Y = 3^(cos x), x изменяется в градусах от 10 с шагом 8.

Вариант 8. Вычислить 6 значений функции Y = 12^x, x изменяется от 0,5 с шагом 0,3.

Вариант 9. Вычислить 5 значений функции Y = 3 \* log8(x^2 +1), x изменяется от 0,2 с шагом 0,3.

Вариант 10. Вычислить 5 значений функции Y = 4^(x^2 + 1), x изменяется от 0,2 с шагом 0,4.

## 9.6 Схема вариантов заданий

| №№  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** |
|  | 9 | 5 | 1 | 1 |
|  | 7 | 4 | 2 | 2 |
|  | 1 | 6 | 5 | 9 |
|  | 2 | 4 | 10 | 10 |
|  | 8 | 7 | 6 | 1 |
|  | 10 | 8 | 7 | 2 |
|  | 2 | 9 | 8 | 6 |
|  | 7 | 8 | 5 | 9 |
|  | 1 | 8 | 6 | 10 |
|  | 9 | 4 | 2 | 1 |
|  | 6 | 4 | 6 | 2 |
|  | 5 | 9 | 4 | 3 |
|  | 8 | 2 | 10 | 4 |
|  | 10 | 5 | 4 | 5 |
|  | 9 | 10 | 3 | 6 |
|  | 4 | 3 | 9 | 7 |
|  | 4 | 1 | 3 | 3 |
|  | 7 | 3 | 2 | 6 |
|  | 5 | 6 | 4 | 7 |
|  | 4 | 9 | 1 | 8 |
|  | 8 | 10 | 6 | 4 |
|  | 5 | 3 | 4 | 9 |
|  | 6 | 7 | 3 | 10 |
|  | 10 | 1 | 8 | 1 |
|  | 8 | 4 | 2 | 2 |
|  | 2 | 5 | 6 | 3 |
|  | 4 | 10 | 5 | 4 |
|  | 6 | 7 | 9 | 5 |
|  | 3 | 7 | 1 | 9 |
|  | 1 | 2 | 9 | 10 |
|  | 5 | 9 | 10 | 1 |
|  | 10 | 6 | 7 | 2 |
|  | 6 | 5 | 8 | 3 |
|  | 7 | 5 | 3 | 4 |
|  | 9 | 2 | 5 | 5 |
|  | 10 | 5 | 7 | 6 |
|  | 1 | 7 | 6 | 7 |
|  | 3 | 6 | 10 | 7 |
|  | 3 | 6 | 7 | 9 |
|  | 5 | 7 | 8 | 10 |
|  | 6 | 10 | 9 | 1 |
|  | 10 | 9 | 1 | 2 |
|  | 9 | 3 | 8 | 5 |
|  | 9 | 7 | 6 | 8 |
|  | 5 | 10 | 9 | 6 |
|  | 1 | 1 | 1 | 7 |
|  | 2 | 4 | 9 | 8 |
|  | 4 | 4 | 8 | 9 |
|  | 9 | 1 | 3 | 10 |
|  | 8 | 7 | 10 | 1 |
|  | 6 | 8 | 4 | 2 |
|  | 4 | 3 | 2 | 3 |
|  | 2 | 1 | 10 | 8 |
|  | 8 | 1 | 4 | 4 |
|  | 3 | 10 | 1 | 5 |
|  | 10 | 7 | 3 | 9 |
|  | 7 | 6 | 6 | 10 |
|  | 1 | 1 | 8 | 1 |
|  | 9 | 3 | 1 | 2 |
|  | 2 | 8 | 9 | 3 |
|  | 3 | 10 | 7 | 4 |
|  | 6 | 9 | 10 | 5 |
|  | 8 | 2 | 5 | 6 |
|  | 5 | 8 | 2 | 7 |
|  | 3 | 10 | 4 | 8 |
|  | 4 | 8 | 5 | 9 |
|  | 9 | 8 | 7 | 10 |
|  | 8 | 9 | 4 | 1 |
|  | 1 | 10 | 8 | 2 |
|  | 5 | 6 | 10 | 3 |
|  | 10 | 4 | 5 | 10 |
|  | 2 | 1 | 3 | 4 |
|  | 7 | 3 | 6 | 5 |
|  | 3 | 7 | 10 | 3 |
|  | 2 | 10 | 8 | 4 |
|  | 4 | 6 | 9 | 5 |
|  | 2 | 3 | 6 | 8 |
|  | 7 | 1 | 2 | 3 |
|  | 3 | 2 | 1 | 4 |
|  | 1 | 5 | 7 | 8 |
|  | 6 | 2 | 9 | 6 |
|  | 10 | 5 | 2 | 7 |
|  | 1 | 9 | 3 | 5 |
|  | 3 | 6 | 9 | 6 |
|  | 6 | 2 | 10 | 7 |
|  | 4 | 5 | 7 | 8 |
|  | 2 | 2 | 5 | 8 |
|  | 3 | 5 | 3 | 9 |
|  | 4 | 8 | 1 | 10 |
|  | 9 | 8 | 4 | 1 |
|  | 7 | 9 | 1 | 8 |
|  | 7 | 2 | 7 | 2 |
|  | 5 | 2 | 5 | 3 |
|  | 10 | 3 | 2 | 4 |
|  | 6 | 9 | 8 | 5 |
|  | 1 | 4 | 5 | 6 |
|  | 8 | 6 | 2 | 7 |
|  | 7 | 1 | 4 | 6 |
|  | 8 | 3 | 7 | 7 |
|  | 5 | 4 | 3 | 9 |

## 9.7 Содержание отчета

Отчет о лабораторной работе должен содержать:

* цель работы;
* краткое описание теоретических основ;
* постановку задачи;
* экспериментально-практическую часть (включая анализ результатов);
* выводы по работе.

В экспериментально-практической части необходимо привести исходные тексты разработанных программ (на языке C++ с ассемблерными вставками) и результаты их выполнения, полученные в регистрах FPU и (или) ячейках памяти.

## 9.8 Контрольные вопросы

1. Приведите форматы данных, обрабатываемых FPU.
2. Что представляет собой регистровая модель блока FPU?
3. Каким образом адресуются регистры данных FPU?
4. Поясните принцип выполнения двухоперандных арифметических команд FPU.
5. Перечислите особые случаи представления вещественных чисел.
6. Какие особые случаи различают в работе FPU?
7. Что представляет собой мнемоника команд FPU?
8. Назовите основные группы команд FPU и объясните их назначение.
9. Как реализуются ветвления на основе анализа значений данных FPU?