

# Дополнительное задание №2

## Изучение команд сопроцессора FPU (x86)

*Цель работы:* познакомиться с архитектурой и системой команд математического сопроцессора FPU (floating point unit) процессоров семейства x86.

### 1 Краткие теоретические сведения

#### 1.1 Архитектура FPU

Арифметический сопроцессор (FPU) расширяет сферы применения процессора (CPU) на область математических расчетов, в которых требуется очень широкий диапазон и высокая точность представления чисел.

FPU имеет стековую организацию, удобную для организации вычислений. В FPU результат операции может часто заместить один или несколько исходных операндов и может служить операндом следующей команды.



Рисунок 1 – Стек регистров процессора FPU

Стек регистров в FPU имеет кольцевую организацию. За правильное использование стека отвечает программист. В частности нужно помнить, что максимальное число данных, которое можно хранить в стеке одновременно равно 8 (восемью).

#### 1.2 Команды FPU

Перечень команд FPU можно найти в книге:

**Юров В.И. *Assembler. Учебник для вузов* / СПб.: Питер, 2003. – 637 с.**

Особенность команд сопроцессора в том, что все они начинаются с буквы «f». Ниже приведен неполный перечень наиболее часто используемых команд FPU.

Название	Действие
finit	устанавливает во всех управляющих регистрах FPU значение по умолчанию
fld mem	помещает переменную mem в вершину стека FPU ST(0)
fst mem	копирует значение из вершины стека в переменную mem.
fstp mem	копирует значение из вершины стека в переменную mem, но далее выталкивает это значение из стека.
fxch	Меняет местами содержимого ST(0) и ST(1).

fadd	$ST(0)=ST(0)+ST(1)$ . Сохраняется только результат на вершине стека
fsub	$ST(0)=ST(0)-ST(1)$ . Сохраняется только результат на вершине стека
fmul	$ST(0)=ST(0)*ST(1)$ . Сохраняется только результат на вершине стека.
fdiv ST(0),ST(i)	$ST(0)=ST(0)/ST(i)$ . Сохраняется только результат на вершине стека.
fsqrt	Извлечение квадратного корня из ST(0).
fsin/fcos	Вычисление синуса/косинуса от аргумента хранящегося в ST(0). Полагается, что в ST(0) хранится радианная мера угла.
frndint	Округляет операнд, находящийся в ST(0), до ближайшего целого числа.
fscale	$ST(0) = ST(0) * 2^{\text{RoundTowardZero } ST(1)}$
fyl2x	$ST(0) = ST(1)*\log_2 ST(0)$ . Операнд в ST(0) должен быть ненулевым положительным числом
f2xm1	$ST(0) = 2^{ST(0)} - 1$ . Значение ST(0) должно находиться в диапазоне $[-1; 1]$ .

Ниже приведён пример использования команд FPU:

```
void main () {
float a,b,c;
a = 1;
b = 2;
__asm{
    finit
    fld a
    fld b
    fadd
    fst c
}
printf("%3.2f + %3.2f = %3.2f \n",a,b,c);
getch();
}
```

Чтобы войти в режим отладки (Debug) и просмотреть содержимое регистров FPU необходимо:

- 1) поставить точку останова (F9) на строке finit
- 2) запустить программу в режиме отладки (F5)
- 3) открыть окно дизассемблера (Debug->Window->Disassembly)
- 4) открыть окно с регистрами процессора (Debug->Window->Registers)
- 5) далее программу можно выполнять в пошаговом режиме с помощью клавиши F10.

### 1.3 Вычисление функции $x^y$

В системе команд FPU нет команды для вычисления функции  $x^y$ . В этом разделе дается совет, как используя имеющиеся команды вычислить эту функцию. Итак, нет команды, чтобы возвести произвольное число  $x$  в произвольную степень  $y$ , но есть команда для возведения 2-ки в заданную степень (f2xm1). Поэтому необходимо сделать следующий математический переход:

$$x^y = 2^{y \cdot \log_2 x} = 2^a, \text{ где } a = y \cdot \log_2 x$$

Далее мы сталкиваемся с той неприятностью, что f2xm1 работает только, когда степень в которую возводится двойка лежит в диапазоне от -1 до 1. Поэтому нам необходимо число представить в виде суммы целой и дробной части:

$$a = a_{\text{дробная часть}} + a_{\text{целая часть}}$$

Получить целую часть числа  $a$  можно командой frndint, тогда дробная часть числа легко вычисляется:  $a_{\text{дробная часть}} = a - a_{\text{целая часть}}$ .

Таким образом, мы приходим к следующему алгоритму вычисления функции  $z = x^y$ :

1. Вычислить  $a = y \cdot \log_2 x$  (команда fyl2x);
2. Найти  $a_{\text{дробная часть}}$  и  $a_{\text{целая часть}}$ ;
3. Вычислить  $z_1 = 2^{a_{\text{дробная часть}}}$  (при помощи команды f2xm1)
4. Вычислить  $z_2 = 2^{a_{\text{целая часть}}}$  (при помощи функции fscale)
5. Вычислить  $z = z_1 + z_2$ .

## 2 Выполнение лабораторной работы

### 2.1 Задание на лабораторную работу

В данной лабораторной работе необходимо реализовать программу вычисления значения функции с использованием команд FPU двумя способами: на языке C++ и с помощью ассемблерной вставки. Значение функции необходимо определить для 10 значений аргумента функции, а результат отобразить в виде следующей форматированной таблицы:

Значение $x$	$R(x, y)$ ассемблер	$R(x, y)$ C++	Разность: $R(x, y)$ ассемблер - $R(x, y)$ C++
--------------	---------------------	---------------	---

Номер	Задание
1	$R = \frac{\cos(x)}{y \cdot \log_2(x+1)} \cdot \sqrt{\sin(y) \cdot x^y} + \exp(y) - \operatorname{tg}\left(\frac{x}{y}\right) - 2^{x-1}$
2	$R = \frac{\cos(x)}{x \cdot \log_2(y)} \cdot \cos(y) \cdot y^{x+1} + \frac{1}{x} \cdot \exp(x) + \sqrt{\sin\left(\frac{x}{y}\right)} - 2^{y-1}$
3	$R = \left( \frac{\cos(x)}{y \cdot \log_2(x+1)} \cdot \sin(y) \cdot x^y + y \cdot \log_2(x+1) \right)^3 \cdot \sqrt{\exp(y)} - \operatorname{tg}\left(\frac{x}{y}\right) - 2^{x-1}$
4	$R = \frac{\cos(x)}{y \cdot \log_2(x+1)} \cdot \sqrt{\sin(y) \cdot x^y} + \exp(y) - \operatorname{tg}\left(\frac{\sqrt{x}}{y}\right) + x \cdot 2^{x-1}$
5	$R = \left( \frac{\cos(x)}{y \cdot \log(x+1)} + \sin(y) \right) \cdot x^y + \sqrt{\exp(y) - \operatorname{tg}\left(\frac{x^{y-1}}{y}\right)} - 2^x$

6	$R = \frac{\cos(y)}{x \cdot \log(x+1)} \cdot \sin(y) \cdot \sqrt{x^{y-1} - \exp(y) - \operatorname{tg}\left(\frac{\sqrt{y}}{x}\right)} - 2^{x-1}$
---	---

### 3 Результаты выполнения лабораторной работы

В результате выполнения данной лабораторной работы необходимо составить отчёт, содержащий следующие пункты:

- 1) Титульный лист
- 2) Цель лабораторной работы
- 3) Индивидуальное задание
- 4) Ход выполнения лабораторной работы (код программы)
- 5) Результаты выполнения лабораторной работы (скриншоты)
- 6) Выводы