БГУИР

Кафедра ЭВМ

Отчёт по лабораторной работе

Тема: “Математический сопроцессор”

Выполнил:

Студент группы 650503 Юревич А. С.

Проверил:

к.т.н., доцент Одинец Д. Н.

Минск

2017

**Постановка задачи**

В данной лабораторной работе необходимо было создать консольное приложение, которое вычисляет функцию двумя способами:

1. С использованием стандартной библиотеки math.h
2. С использованием команд математического сопроцессора

Было необходимо выводить время выполнения каждого из способов.

**Алгоритм решения задачи**

В случае использования стандартной библиотеки math.h:

1. В цикле пока не вычислим сумму значений функций на интервале от *a* до *b* находим сумму значений функции и прибавляем шаг.

В случае использования сопроцессора:

1. Инициализируем сопроцессор.
2. Загружаем в вершину стека сопроцессора верхнюю границу интервала и изначальное значение функции.
3. Проводим проверку, достигнута ли граница интервала. Если нет, то переходим к шагу 4, иначе заканчиваем работу с сопроцессором и синхронизируем работу сопроцессора с ЦП.
4. Загружаем константу “0” в вершину стека.
5. Прибавляем текущее значение функции к предыдущему.
6. Загружаем значение функции в вершину стека.
7. Возводим значение функции во вторую степень и прибавляем 10.
8. Делим числитель на знаменатель.
9. Прибавляем шаг, находим новое итогоове значение функции и сохраняем его в памяти. Переходим к шагу 3.

**Листинг программы**

// 16. F(x) = sin(x)/(x^2+10)

#include <math.h>

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b, step, f, x;

int ten = 10;

clock\_t start\_asm, end\_asm, start\_math, end\_math;

while (1)

{

system("cls");

scanf("%lf %lf %lf", &a, &b, &step);

puts("");

x = a;

f = 0;

start\_math = clock();

while (x <= b) //Вычисление функции при помощи библиотеки math.h

{

f += sin(x) / (pow(x, 2) + 10);

x += step;

}

end\_math = clock();

printf("Библиотека <math.h>\n");

printf("Значение функции: %f\n", f);

printf("Время: %f\n\n", (double)(end\_math - start\_math) / CLK\_TCK);

x = a;

f = 0;

start\_asm = clock();

\_asm

{

finit //Инициализация сопроцессора

fld b //Загружаем в ST(0) верхнюю границу интервала

fld x //Загружаем значение функции

loop\_start : //метка начала цикла

fcom //проверка границы интервала

fstsw ax

and ah, 01000101b

jz loop\_end

fldz //Загружаем ноль

fadd x //Прибалвяем х

fsin //Вычисляем синус

fld x //Загружаем х

fmul x //Умножаем x\*x

fiadd ten //Прибавляем 10

fdiv //Делим

fadd f //Прибавляем значение функции

fstp f //Сохраняем в памяти значение

fadd step //Прибавляем шаг

fst x //Сохраняем х в памяти

jmp loop\_start //Переходим к метке

loop\_end : //Метка окончания работы

fwait

}

end\_asm = clock();

printf("Работа сопроцессора\n");

printf("Значение функции: %f\n", f);

printf("Время: %f\n\n", (double)(end\_asm - start\_asm) / CLK\_TCK);

printf("Для выхода нажмите 0");

char c = \_getch();

if (c == '0') break;

}

return 0;

}

**Тестовые примеры**

На рисунках 1 и 2 представлены результаты работы программы.

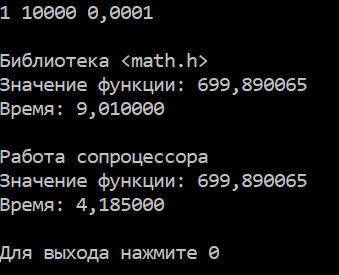


Рисунок 1

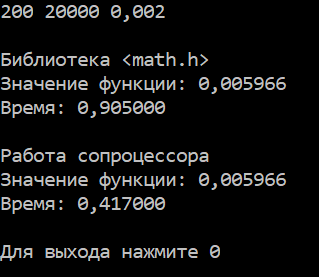


Рисунок 2

**Заключение**

Как показала программа, математический сопроцессор проводит расчёт данной функции быстрее в среднем в 2.15 раз. Это связано с тем, что математический сопроцессор расширяет вычислительные возможности основного процессора и работает параллельно с ним. Ассемблерные вставки позволяют отказаться от подключения дополнительных библиотек и позволяют писать оптимальный код, что также ускоряет вычисления. Использование математического сопроцессора значительно ускоряет математические вычисления, что говорит о важном значении этого устройства.