Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №1

По дисциплине «Архитектура вычислительных систем» По теме «Программирование арифметического сопроцессора»

Выполнила:

студентка гр. 253503

Ярмак В.С.

Проверила:

ассистент каф. информатики

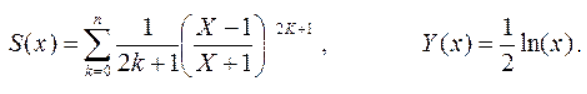
Калиновская А.А.

Минск 2024

**Цель работы**: Ознакомиться с предназначением арифметического сопроцессора. Рассмотреть строение и основные принципы работы сопроцессора. Изучить команды, доступные при использовании сопроцессора. Научиться работать с сопроцессором в ходе выполнения лабораторного задания.

**Задание:** Значение аргумента 𝑥 изменяется от 𝑎 до 𝑏 с шагом ℎ. Для каждого 𝑥 найти значения функции 𝑌(𝑥), суммы 𝑆(𝑥) и число итераций 𝑛, при котором достигается требуемая точность ε = |𝑌(𝑥) − 𝑆(𝑥)|. Результат вывести в виде таблицы. Значения 𝑎, 𝑏, ℎ и ε вводятся с клавиатуры.

**Вариант 10**



Листинг 1 – исходный код программы

#include <iostream>

#include <cmath>

#include <chrono>

#include <fstream>

float Y(float x)

{

  const float one = 1.0f;

  const float two = 2.0f;

  const float four = 4.0f;

  float Y = 0.0f;

  \_\_asm\_\_(

    "fld %2;" //2

    "fld %1;" //1

    "fdivp;" // 1/2

    "fldln2;" //ln2

    "fld %3;" //x

    "fyl2x;" //log2(x)\*ln(2) = ln(x)

    "fmulp;" // 1/2\*(ln(x))

    "fstp %0;"

    : "=m" (Y)

    : "m" (one), "m" (two), "m" (x)

  );

  return Y;

}

float S(float x, float eps, float& n)

{

  const float one = 1.0f;

  const float two = 2.0f;

  float sum = 0.0f;

  float res;

  const float pi = 3.14159265358979323846f;

  const float four = 4.0f;

  n = 1.0f;

  float S = 0.0f;

  float totalS = 0.0f;

  while (std::abs(totalS - Y(x)) > eps)

  {

    //std::cout << "Before " << n << std::endl;

    float tempS1 = 0.0f;

    float tempS2 = 0.0f;

    //1/(2\*k+1)

    asm

    (

      "fld %2;" //2

      "fld %3;" //k

      "fmulp;"  //2\*k

      "fld %1;" //1

      "faddp;"  //2\*k+1

      "fld %1;" //1

      "fdivp;"  //1/(2\*k+1)

      "fstp %0;"

      : "=m" (tempS1)

      : "m" (one), "m" (two), "m" (n), "m" (x)

    );

    //std::cout << "x = " << x << "k = " << n << std::endl;

   // std::cout << "tempS1 = " << tempS1 << std::endl;

    //(x-1)/(x+1)

    asm

    (

      "fld %3;" //x

      "fld %1;" //1

      "fadd;"   //x+1

      "fld %1;" //1

      "fld %3;" //x

      "fsub;"   //x-1

      "fdivp;" //(x-1)/(x+1)

      "fstp %0;"

      : "=m" (tempS2)

      : "m" (one), "m" (n), "m" (x)

    );

    //std::cout << "tempS2 = " << tempS2 << std::endl;

    float tempS2mul = tempS2;

    //std::cout<<"n = "<<n<<"\n";

    for(int i = 1; i < (2\*n + 1); i++)

    {

      tempS2 \*= tempS2mul;

      //std::cout<<"Up = "<<tempS2<<"\n";

    }

    //std::cout << "tempS2\* = " << tempS2 << std::endl;

    asm

    (

      "fld %2;" //1/(2\*k+1)

      "fld %1;" //((x-1)/(x+1))^(2\*n + 1)

      "fmulp;"  //S

      "fld %0;"

      "faddp;"

      "fstp %0;"

      : "=m" (S)

      : "m" (tempS2), "m" (tempS1)

    );

    totalS+=S;

    //std::cout<< "S = " << totalS << std::endl;

    //0.sum=S;1

    if (n>1000) break;

    n+=1.0f;

    //std::cout<< "totalS =  " <<totalS<< "Y(x) = " << Y(x) << " std::abs(totalS - Y(x))" << std::abs(totalS - Y(x)) << std::endl;

  }

  //return sum;

  return totalS;

}

int main()

{

  float a, b, h, eps;

  float k=1.0f;

  std::cout << "Enter a, b, h, eps: ";

  std::cin >> a >> b >> h >> eps;

  for (double x = a; x < b; x += h)

  {

    float n=1.0f;

    auto start = std::chrono::high\_resolution\_clock::now();

    float y = Y(x);

    float s = S(x, eps, n);

    auto end = std::chrono::high\_resolution\_clock::now();

    auto duration = std::chrono::duration\_cast<std::chrono::nanoseconds>(end - start).count();

    //std::cout << "x = " << x << ", Y(x) = " << y << ", S(x) = " << s << ", n = " << n << ", time = " << duration << " ns" <<

    //std::endl;

    std::cout<<n<<" "<<duration<<std::endl;

  }

  return 0;

}



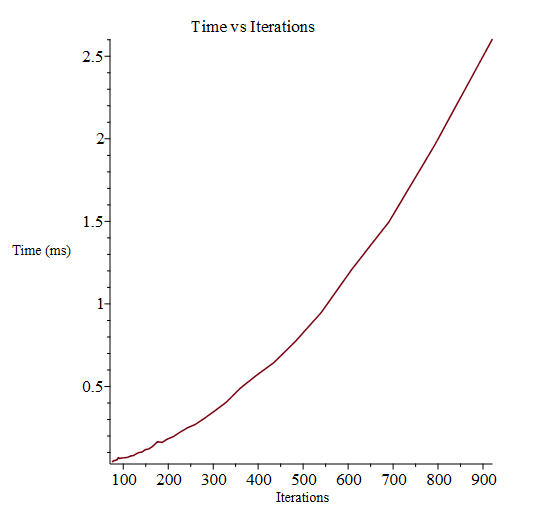


Рисунок 1 – работа программы на промежутке [1.12;1.5] с шагом 0.01 и точностью 0.001

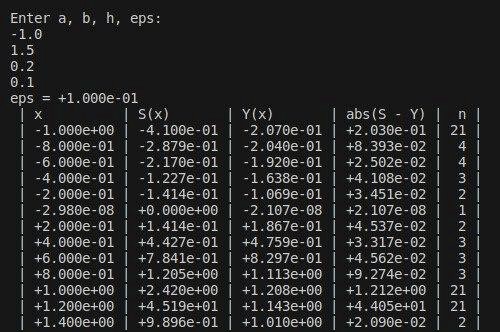


Рисунок 2 – работа программы на промежутке [-1;1.5] с шагом 0.2 и точностью 0.1

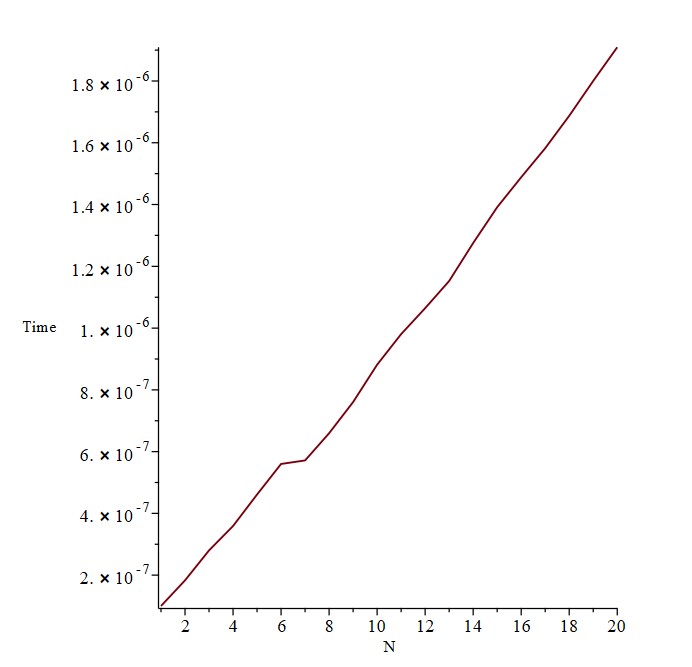


Рисунок 3 – график количества времени в секундах от итераций программы при x равном -1 и точностью 0.1

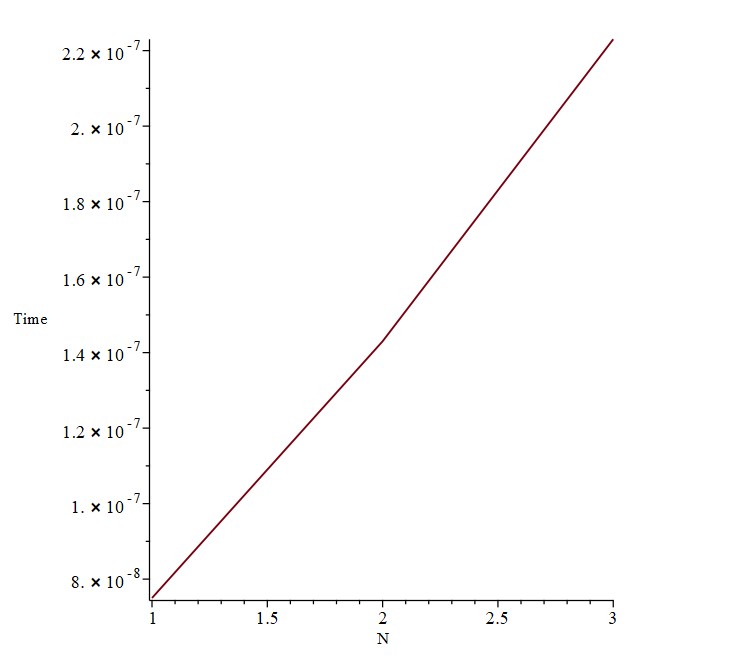


Рисунок 4 – график количества времени в секундах от итераций программы при x равном -0.8 и точностью 0.1

**Выводы:** В ходе выполнения лабораторной работы были изучены понятия арифметического сопроцессора, сопроцессорных конфигураций, программной модели сопроцессора. Рассмотрены различные форматы представления численных данных: двоичные целые числа, упакованные десятичные числа, вещественные числа. Выяснены режимы работы и состояния арифметического сопроцессора, система команд сопроцессора, особенности задания команд, различные группы команд сопроцессора (команды управления, команды передачи данных, команды загрузки, команды запоминания, арифметические команды).