

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина: Архитектура вычислительных систем

**ОТЧЁТ**  
к лабораторной работе №2  
на тему  
Программирование арифметического сопроцессора

Выполнил: студент группы 253505  
Снежко Максим Андреевич  
Проверил: Калиновская Анастасия Александровна

Минск 2024

**Цель:** научиться программировать, используя возможности арифметического сопроцессора, научиться работать с процессорами различных семейств и поколений в режиме множества ядер.

**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  $\varepsilon = |Y(x) - S(x)|$ . Результат вывести в виде таблицы. Значения  $a$ ,  $b$ ,  $h$  и  $\varepsilon$  вводятся с клавиатуры.

### Вариант 8

$$8. \quad S(x) = \sum_{k=0}^n \frac{2k+1}{k!} x^{2k}, \quad Y(x) = (1 + 2x^2)e^{x^2}.$$

**Ход выполнения:**

Платформа: ноутбук Honor MagicBook 16 HYM-WXX

Процессор: AMD Ryzen 5 5600H with Radeon Graphics.

Операционная система: Linux MANJARO.

Начальные условия ( $a = 0.8$ ,  $b = 0.85$ ,  $h = 0.0001$ ,  $\varepsilon = 1e-11$ ) взяты такими для получения достоверных результатов эксперимента.

Все вычисления происходят в консольном режиме Manjaro.

В первом тесте запустим вычисления на всех ядрах с включенной функцией Simultaneous Multithreading. Количество потоков на ядро равно 2.

Часть результата выполнения на всех ядрах с включенной технологией Simultaneous Multithreading:

S	Y	Итерации	Время
4.640482	4.640482	14	551
4.534857	4.334857	14	271
4.475973	4.475973	14	71
4.463848	4.463848	14	355
4.174985	4.174985	14	340
4.579947	4.579947	14	271
4.331777	4.708207	14	340
4.733078	4.383297	14	290
4.105235	4.105235	14	350
4.356837	4.670404	14	381

4.989729	4.890729	14	351
4.707282	4.707282	14	651
4.564433	4.394433	14	531
4.469076	4.389076	14	421
4.542123	4.822123	14	340
4.420913	4.742091	14	361
4.468142	4.348142	14	341
4.852321	4.952321	14	431
4.562937	4.362937	14	331
4.705315	4.305315	14	341
4.196476	4.496476	14	691
4.807977	4.847977	14	611
4.947734	4.447734	14	370
4.895767	4.595767	14	331
4.92094	4.492094	14	371
4.936733	4.436733	14	341
4.829704	4.829704	14	341
4.461833	4.941833	14	390
4.918282	4.918282	14	431
4.421688	4.421688	14	340
4.972068	4.972068	14	340
4.356944	4.156944	14	371
4.213822	4.213822	14	341
4.295233	4.205233	14	340
4.343689	4.643689	14	341
4.429209	4.429209	14	341
4.526811	4.261811	14	330
4.341513	4.141513	14	331
4.968334	4.068334	14	340
4.384229	4.304229	14	331
4.773311	4.773311	14	731
4.895295	4.395295	14	340
4.132895	4.132895	14	351
4.318133	4.418133	14	361
4.751027	4.751027	14	340
4.513596	4.131596	14	340
4.855919	4.585919	14	541
4.370483	4.710483	14	521
4.697482	4.697482	14	411

Во втором тесте вычисления будем производить на всех ядрах с отключенной функцией Simultaneous Multithreading. В данном тесте на каждом ядре работает один поток.

Часть результата:

S	Y	Итерации	Время
4.934393	4.934393	14	440
4.678816	4.678816	14	293
4.477243	4.477243	14	131
4.329695	4.329695	14	150
4.236193	4.236193	14	160
4.271748	4.271748	14	241
4.527748	4.427748	14	150
4.576352	4.676352	14	161
4.694127	4.894243	14	180
4.204547	4.604547	14	130
4.566481	4.456481	14	120

4.979947	4.979947	14	180
4.444967	4.344967	14	160
4.596156	4.696156	14	121
4.599746	4.529746	14	170
4.496545	4.6149545	14	120
4.820977	4.820977	14	141
4.254062	4.544062	14	140
4.319882	4.531882	14	140
4.814527	4.814527	14	140
4.023433	4.502433	14	140
4.957329	4.953329	14	161
4.532192	4.532192	14	210
4.640482	4.640482	14	230
4.534857	4.534857	14	180
4.475973	4.345973	14	150
4.763848	4.463848	14	140
4.349985	4.974985	14	150
4.579947	4.579947	14	140
4.708207	4.708807	14	141
4.883297	4.883297	14	150
4.805235	4.305235	14	140
4.837404	4.837404	14	140
4.869729	4.689729	14	140
4.902321	4.052321	14	130
4.254645	4.254645	14	341
4.571479	4.571479	14	151
4.940603	4.940603	14	150
4.362035	4.733620	14	140
4.735797	4.835797	14	141
4.607755	4.167755	14	231
4.620197	4.620197	14	140
4.687523	4.687523	14	141
4.809755	4.809755	14	131
4.386914	4.986914	14	140
4.219022	4.919022	14	140
4.485061	4.935061	14	140
4.948169	4.848169	14	140
4.924251	4.125251	14	130

На рисунках 1 и 2 можно увидеть графики зависимости времени вычисления от количества итераций для двух тестов.

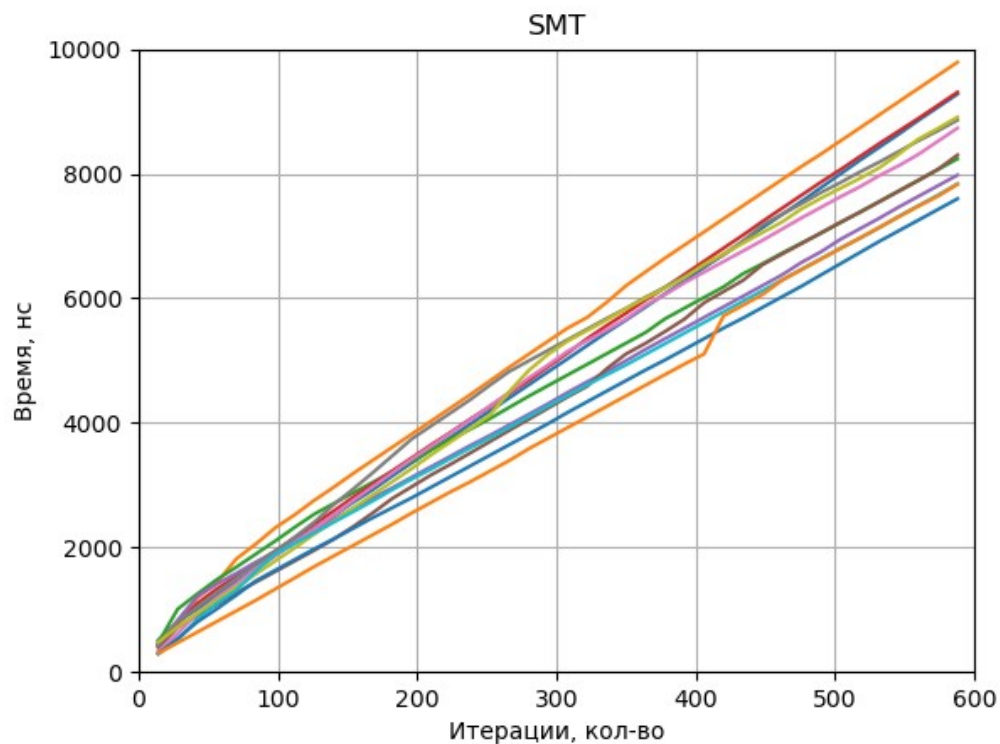


Рисунок 1 — График зависимости времени от количества итераций для включенного режима Simultaneous Multithreading на всех ядрах.

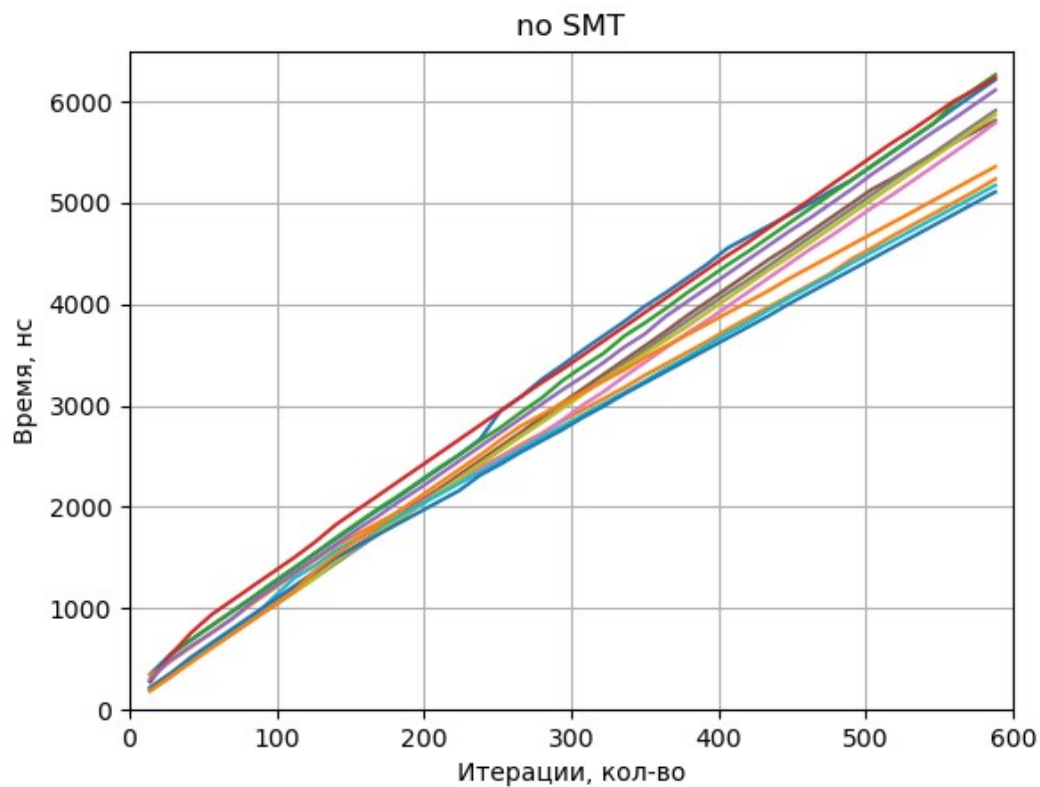


Рисунок 2 — График зависимости времени от количества итераций для выключенного режима Simultaneous Multithreading на всех ядрах.

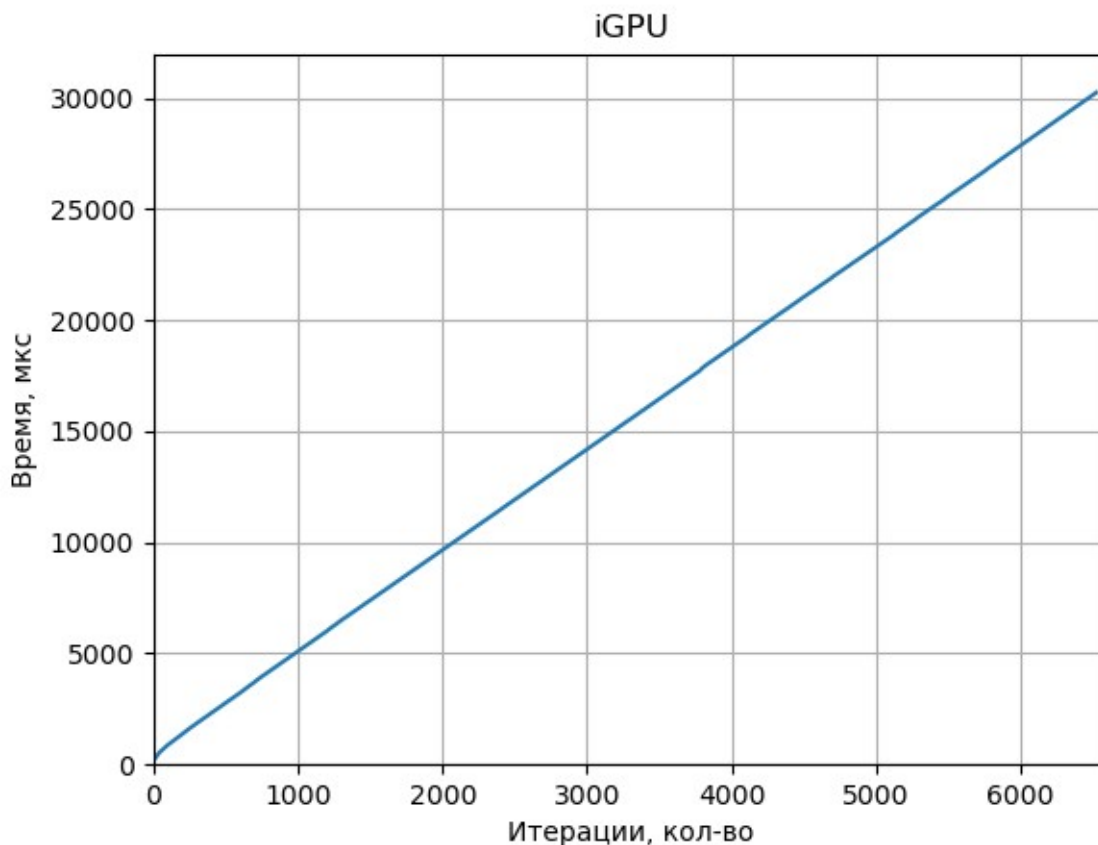


Рисунок 2 — График зависимости времени от количества итераций на iGPU

**Вывод:** Как видно из графика и результатов в таблице режим Simultaneous Multithreading увеличил время выполнения программы. В ходе лабораторной работы я научился программировать с использованием арифметического сопроцессора. Протестировал программу в двух режимах: в многоядерном режиме с использованием Simultaneous Multithreading и в многоядерном режиме без использования Simultaneous Multithreading.