###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ

«Измерение степени ассоциативности кэш-памяти»

студента 2 курса, группы 20204

**Дронова Дениса Юрьевича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Кандидат технических наук

А. Ю. Власенко

Новосибирск 2021

**СОДЕРЖАНИЕ**

[ЦЕЛИ 3](#_Toc89469519)

[ЗАДАНИЕ 3](#_Toc89469520)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc89469521)

[Приложение 1. *Листинг программы* 6](#_Toc89469522)

# ЦЕЛЬ

Экспериментальное определение степени ассоциативности кэш-памяти

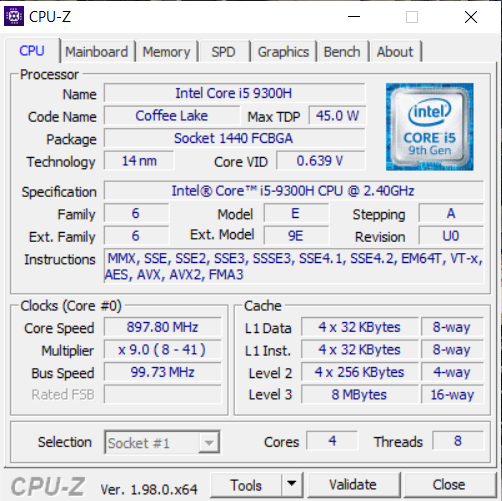
# ЗАДАНИЕ

1. Написать программу, выполняющую обход памяти в соответствии с заданием.
2. Измерить среднее время доступа к одному элементу массива (в тактах процессора) для разного числа фрагментов: от 1 до 32. Построить график зависимости времени от числа фрагментов.
3. По полученному графику определить степень ассоциативности кэш-памяти, сравнить с реальными характеристиками исследуемого процессора.

# ОПИСАНИЕ РАБОТЫ

В ходе выполнения практической работы было произведено три варианта обхода массива с различными вариантами смещения (offset) для определения степеней ассоциативности уровней кэш-памяти L1(offset: 8192), L2(offset: 131072), L3(offset: 16777216). Каждый вариант производил обход 10 раз для более точного определения среднего времени. Предварительно были получены размеры уровней кэш-памяти и их степени ассоциативности для конкретного процессора при помощи утилиты CPU-Z. По результатам замеров были получены значения числа тактов для каждого варианта обхода массива (с разными вариантами offset). На основе полученных значений числа тактов были построены графики, приведенные ниже.

Строка компиляции: g++ -O1 cache\_assoc.cpp -o a.exe

**

**ЗАКЛЮЧЕНИЕ**

В процессе выполнения практической работы были исследованы графики времени доступа к данным, находящимся в кэш-памяти, в зависимости от шага обхода элементов массива. По результатам исследования можно выдвинуть гипотезу о степенях ассоциативности уровней кэш-памяти процессора, установленного на тестовый ПК: L1 – 8 ассоциативный, L2 – 4 ассоциативный, L3 – 16 ассоциативный. Сопоставив полученные данные с данными, предоставляемыми утилитой CPU-Z, можно убедиться в истинности выдвинутой гипотезы.

# Приложение 1. *Листинг программы*

#include <cstdlib>

#include <iostream>

#include <intrin.h>

using namespace std;

#define L1\_cache 131072  // 128 Kb - for 4 cores (32 Kb - on each core)

#define L2\_cache 1048576 // 1 MB - for 4 cores (256 Kb - on each core)

#define L3\_cache 8388608 // 8 MB

//L1 bank = 4096 bytes

//L2 bank = 65536 bytes

//L3 bank = 524288 bytes

#define L1\_offset 8192    //2 \* L1 bank size

#define L2\_offset 131072  //2 \* L2 bank size (also may use 524288)

#define L3\_offset 8388608 //16 \* L3 bank size

double count(int \*arr, size\_t offset)

{

    volatile size\_t k = 0;

    double time = \_\_rdtsc();

    for (volatile size\_t i = 0; i < 10 \* offset; ++i){

        k = arr[k];

    }

    time = \_\_rdtsc() - time;

    return time / (10 \* offset); //average time

}

void FillArray(int \*array, int const offset, int n){

    for (size\_t i = 0; i < offset; ++i){

        for (size\_t j = 0; j < n - 1; ++j){

            array[j \* offset + i] = (j + 1) \* offset + i;

        }

        array[(n - 1) \* offset + i] = (i + 1) % offset; //next row

    }

}

int main(){

    const int offset = L3\_offset;

    int \*array = new int[32 \* offset];

    for (int i = 1; i <= 32; ++i){

        FillArray(array, offset, i);

        cout << count(array, offset \* i) << endl;

    }

    delete[] array;

    return 0;

}