

**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 ЛАБОРАТОРНОЙ РАБОТЫ
«ИЗМЕРЕНИЕ СТЕПЕНИ АССОЦИАТИВНОСТИ КЭШ-ПАМЯТИ»**

студента 2 курса, 21202 группы

Куращенко Льва Владиславовича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
Перепелкин В.А.

Новосибирск 2022

1. Цель

Экспериментально определить степени ассоциативности кэш-памяти разных уровней.

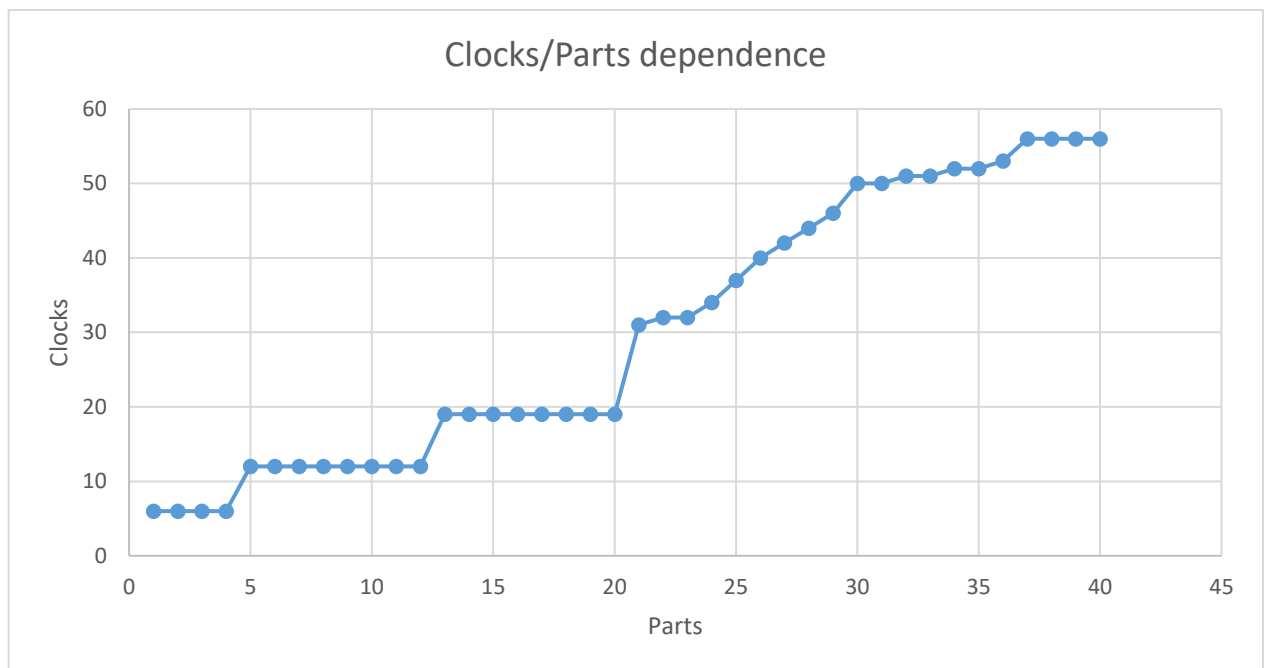
2. Задание

На основе обхода циклического массива, что вызывает кэш-букование, сделать среднюю оценку количества тактов для обращения к одному элементу массива. На основе полученных данных выяснить степени ассоциативности кэш-памяти процессора.

3. График роста тактов обращения

В лабораторной работе создавался массив с фрагментами со смещением, равному 16мб. Количество элементов в фрагменте равнялось ~7мб, что равняется размеру кэш памяти. На основе множественных тестов выяснилось, что данные параметры лучше всего показывали степень ассоциативности памяти.

Следующий график показывает зависимость времени обращения к одному элементу в тактах от количества фрагментов в массиве:



Вывод программы, на основе которого заполнялась таблица:

1	6
2	6
3	6
4	6
5	12
6	12
7	12
8	12

9	12
10	12
11	12
12	12
13	19
14	19
15	19
16	19
17	19
18	19
19	19
20	19
21	31
22	32
23	32
24	34
25	37
26	40
27	42
28	44
29	46
30	50
31	50
32	51
33	51
34	52
35	52
36	53
37	56
38	56
39	56
40	56

4. Листинг программы

```

5. #include <iostream>

const unsigned long long int N = 1024 * 1024 * 300;
const unsigned long long int RUN_ARRAY_COUNT = 8;
const unsigned long long int offset = 16 * 1024 * 1024 / sizeof(int);
const unsigned long long int size = 7 * 1024 * 1024 / sizeof(int);
unsigned long long int rdtsc()
{
    unsigned long long int hi, lo;
    __asm__ __volatile__ ("rdtsc" : "=a"(lo), "=d"(hi));
    return lo | (hi << 32ull);
}

void bypass(unsigned int *array, unsigned int fragments, int offset, int size)
{
    size_t i = 0;
    size_t j = 1;

    for(i = 0; i < size; i++)
    {

```

```

        for(j = 1; j < fragments; j++)
            array[i + (j - 1) * offset] = i + j * offset;

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

    array[i - 1 + (j - 1) * offset] = 0;
    volatile size_t counter;
    volatile size_t temp_k;
    volatile size_t temp_j;
    volatile unsigned long long int t1, t2;
    unsigned long long int tmin = ULLONG_MAX;
    for(temp_j = 0; temp_j < RUN_ARRAY_COUNT; temp_j++)
    {
        t1 = rdtsc();

        for(temp_k = 0, counter = 0; counter < N; counter++)
            temp_k = array[temp_k];

        t2 = rdtsc();

        if(tmin > t2 - t1)
            tmin = t2 - t1;
    }

    printf("%u\t %llu\n", fragments, tmin / N);
}

int main()
{
    auto *array = (unsigned int *) malloc(N * sizeof(unsigned int));
    if(!array)
        return 1;
    for(int n = 1; n <= 40; n++)
        bypass(array, n, offset, size);
    free(array);
    return 0;
}

```

6. Оценка ассоциативности кэша

На графике заметно несколько скачков. Первый скачок происходит после 4 фрагментов, что, скорее всего, говорит о степени ассоциативности TLB-кэша (буфера трансляции адресов). Следующие скачки роста времени обращения произошли на 13м и 21м фрагментах, что явно говорит о том, что степени ассоциативности кэшей L1 и L2 равны 12 и 20.

Данные, полученные с помощью CPU-Z, подтверждают данные предположения:

Cache		
L1 Data	4 x 48 KBytes	12-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 1.25 MBytes	20-way
Level 3	8 MBytes	8-way

Данные верны для процессора Intel Core i5 11300h. Расчёты проводились при тактовой частоте 3.5GHz.

7. Заключение

В ходе данной работы я смог установить степени ассоциативности кэш-ей собственного процессора и выяснить, насколько существенен прирост времени обращения при "переходе" с одного кэша на другой.