

Отчет к лабораторной работе cache benchmark

Процессор: AMD Ryzen 5 5600H ([Ryzen 5 5600H - AMD - WikiChip](#)).

L1 cache – 384 KB (L1D – 32 KB x 6 cores, L1I – 32 KB x 6 cores).

L2 cache – 3072 KB = 3 MB (512 KB x 6 cores)

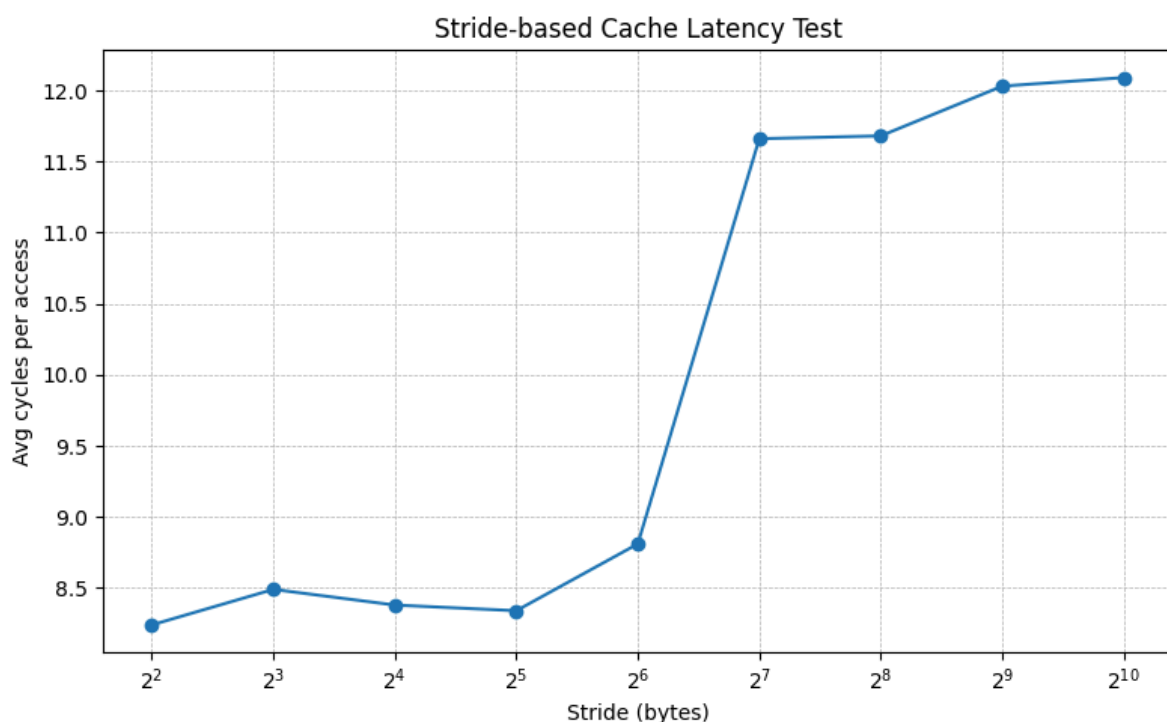
L3 cache – 16384 KB = 16 KB (unified for all cores)

RAM – 16 GB

Все таблицы с выводами программ находятся [здесь](#).

В самом начале во все программы была добавлена привязка к одному ядру.

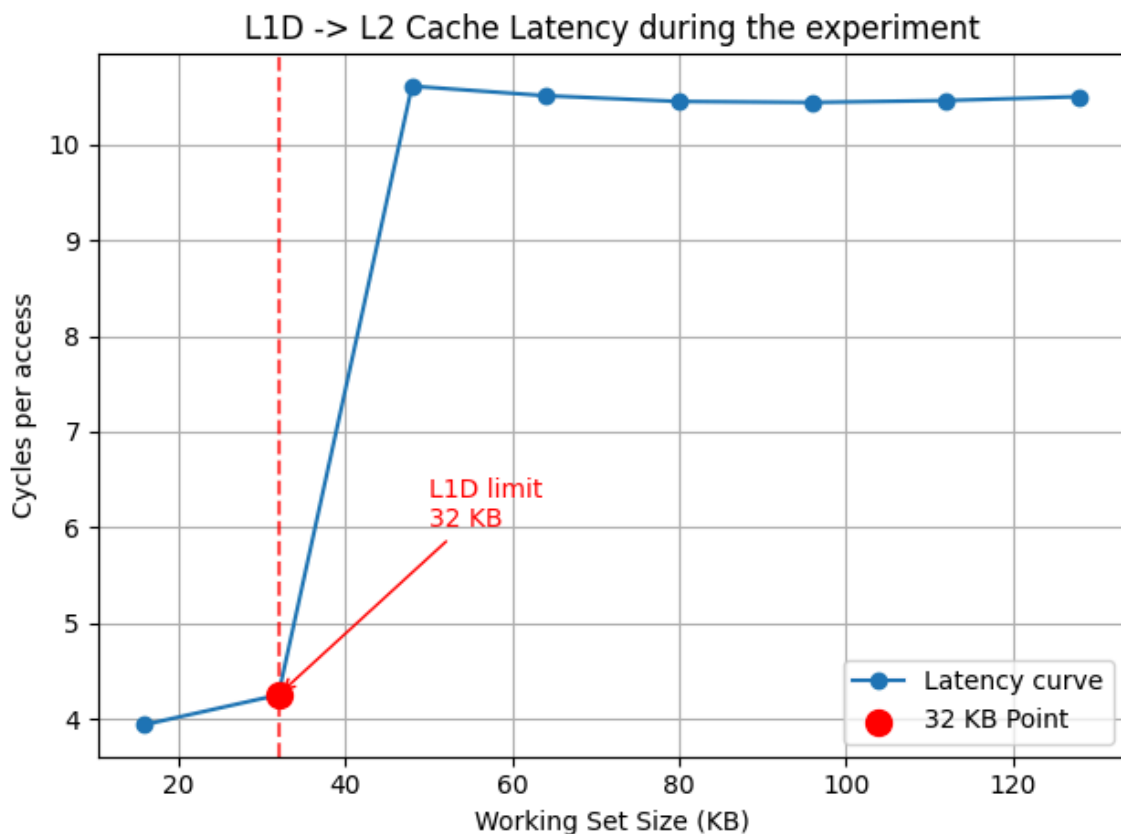
Сперва измерим размер кэш-линии. Для этого была написана программа (`cache_line_size.cpp`), в которой осуществлен read-write проход по массиву с меняющимся шагом (*stride-based*). График изменения тактов на доступ в зависимости от размера шага:



Здесь хорошо видно скачкообразное повышение числа тактов на доступ начиная с $2^6 = 64$ -байтового шага. Это говорит о том, что с этого момента произошел переход на другую кэш-линию. Следовательно, размер кэш-линии на тестируемом процессоре – 64 байта.

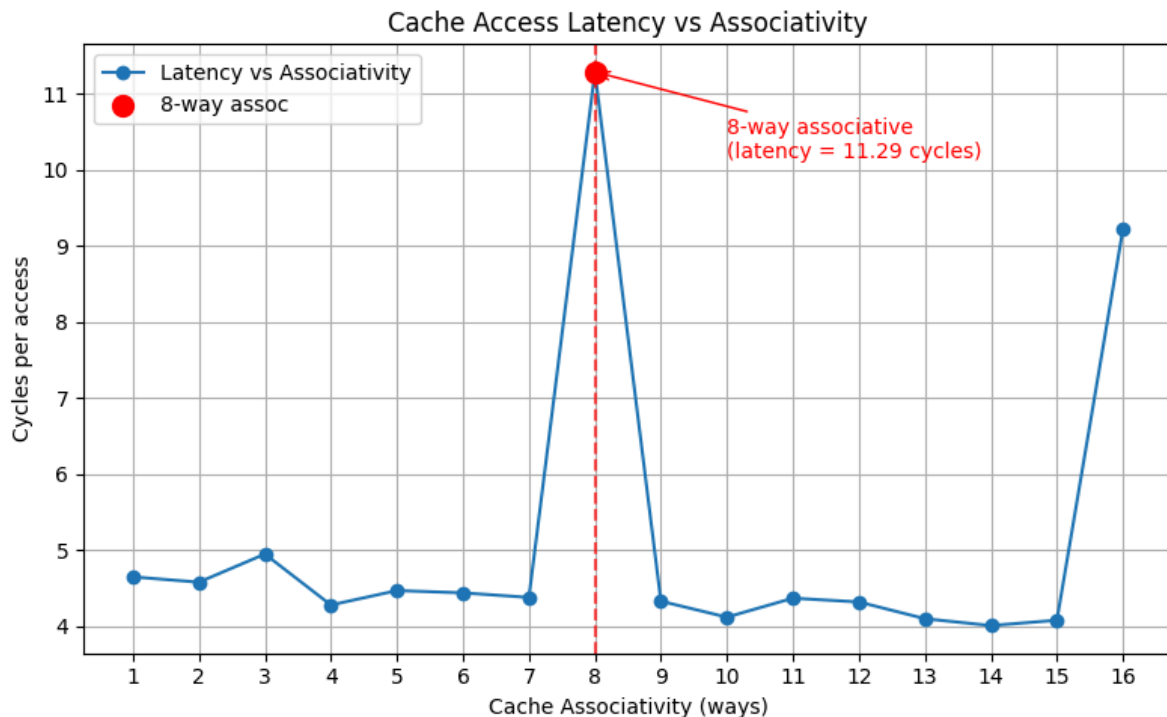
Для измерения размера L1 Data кэша была написана специальная программа (файл `L1d_cache_size.cpp`), в которой делается выравнивание по размеру кэш-линии. Она строит массив индексов, перемешивает его и на его основе создаёт односвязный кольцевой список из структур `Node`, выровненных на 64 байта (размер кэш-линии). Затем все данные кэша сбрасываются с помощью `_mm_clflush`, и мы итерируемся по указателям (`p = p->next`) фиксированное количество раз, измеряя разницу таймеров `__rdtscp` до и после итерации. Весь этот блок выполняется независимо 200 раз подряд (константа `RUNS`), после чего накопленные данные усредняются. Поскольку порядок переходов случайный (не последовательный), аппаратный prefetcher срабатывает, и мы видим реальную латентность L1D кэша.

График:



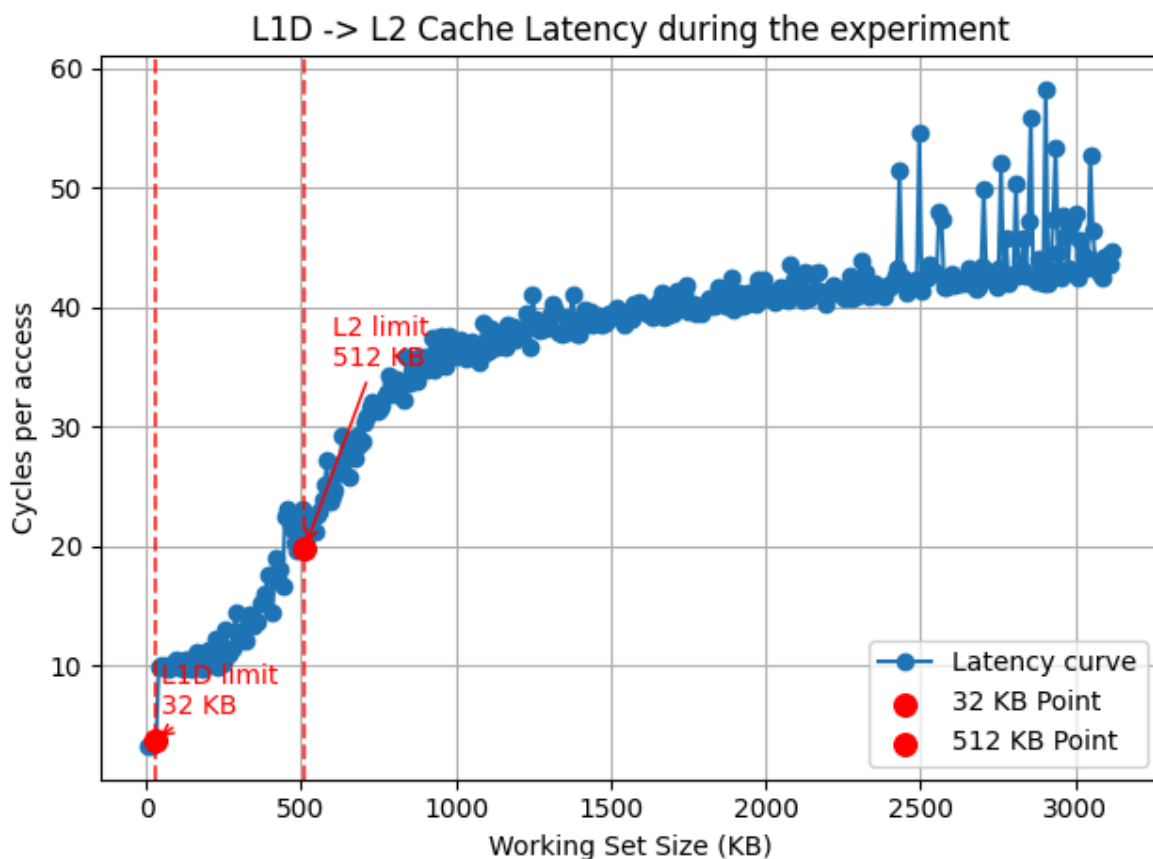
Как можно видеть из графика, начиная с размера рабочего набора, превышающего 32 КВ, заметно повышается среднее число тактов на доступ. Это связано с тем, что данные уже не помещаются в L1D, и поэтому возникают cache miss'ы, которые начинают обслуживаться из L2. Отсюда делаем вывод, что размер L1D-кеша на одно ядро равен 32 KiB. Поскольку тест выполнялся на шестиядерном процессоре, общий объём L1D-кеша составляет $32 \text{ KB} \times 6 = 192 \text{ KB}$. Полученные результаты совпадают с [официальными спецификациями](#) процессора. Латентность L1D кэша – примерно 4 цикла (такта).

Теперь выведем ассоциативность L1D кэша. Для этого была написана программа `L1D_assoc.cpp`. График:



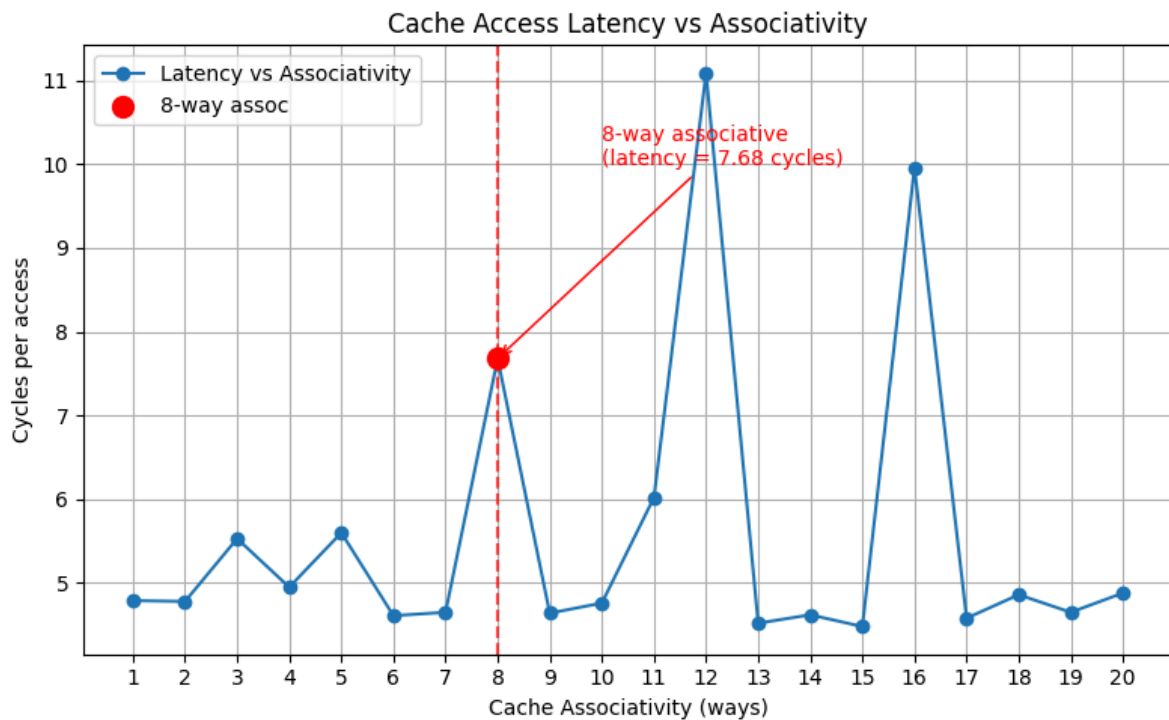
На графике при ассоциативности 8-way наблюдается резкий скачок латентности доступа (до ≈ 11.3 циклов), тогда как при остальных степенях ассоциативности значения лежат в диапазоне 4–5 циклов. Это указывает на то, что L1D кэш у тестируемого процессора имеет 8-way ассоциативность. Полученные результаты совпадают с официальными спецификациями.

Проводим аналогичные замеры для L2 кэша (программа `L2_L3_cache_size_latency.cpp`).
График:



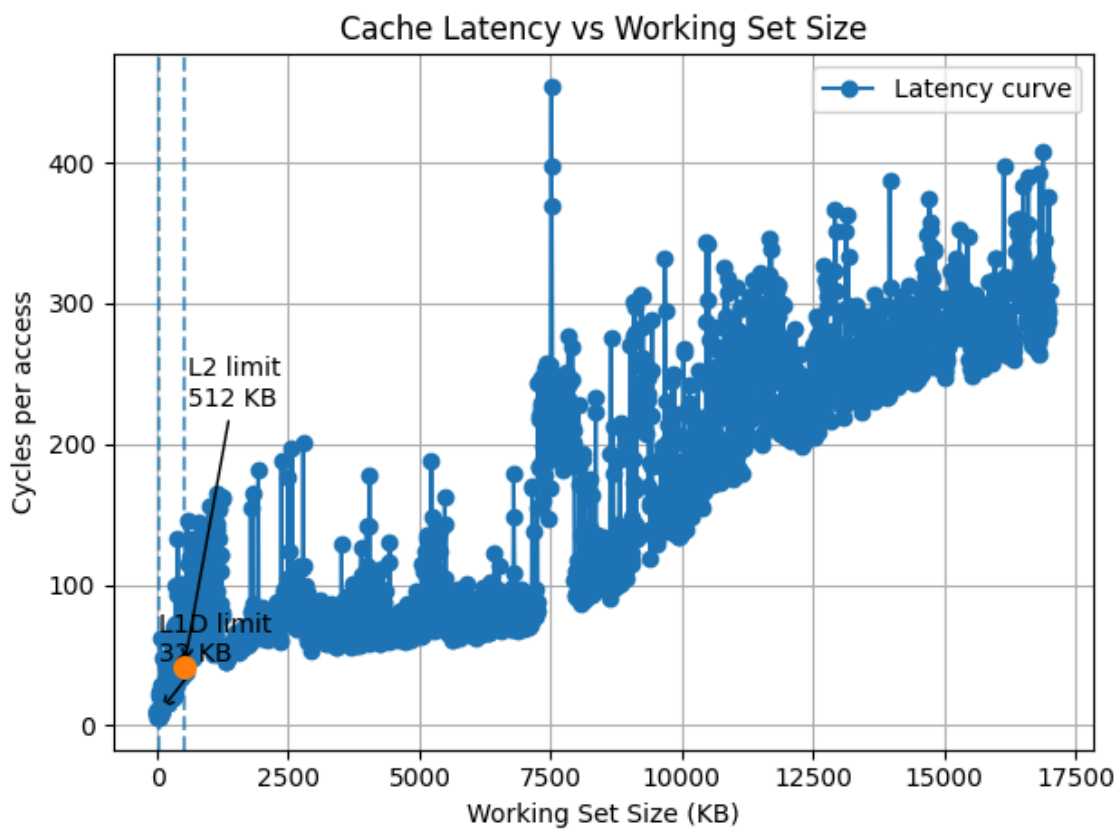
При размере рабочего набора от 32 KB до 512 KB рост латентности преимущественно обусловлен TLB-miss'ами и page walk'ами. Чёткий переход к более высокой латентности после 512 KB видно плохо из-за наложения эффектов TLB, однако там наблюдается рост латентности. Также по спецификации тестируемого процессора размер L2-кэша составляет 512 KB. L3-кэш (16 MB) начинает влиять на задержки при размерах свыше 512 KB, что тоже проявляется в дополнительном росте латентности при объёмах > 512 KB.

Латентность L2-кэша – ~11 циклов (тактов). Ассоциативность:



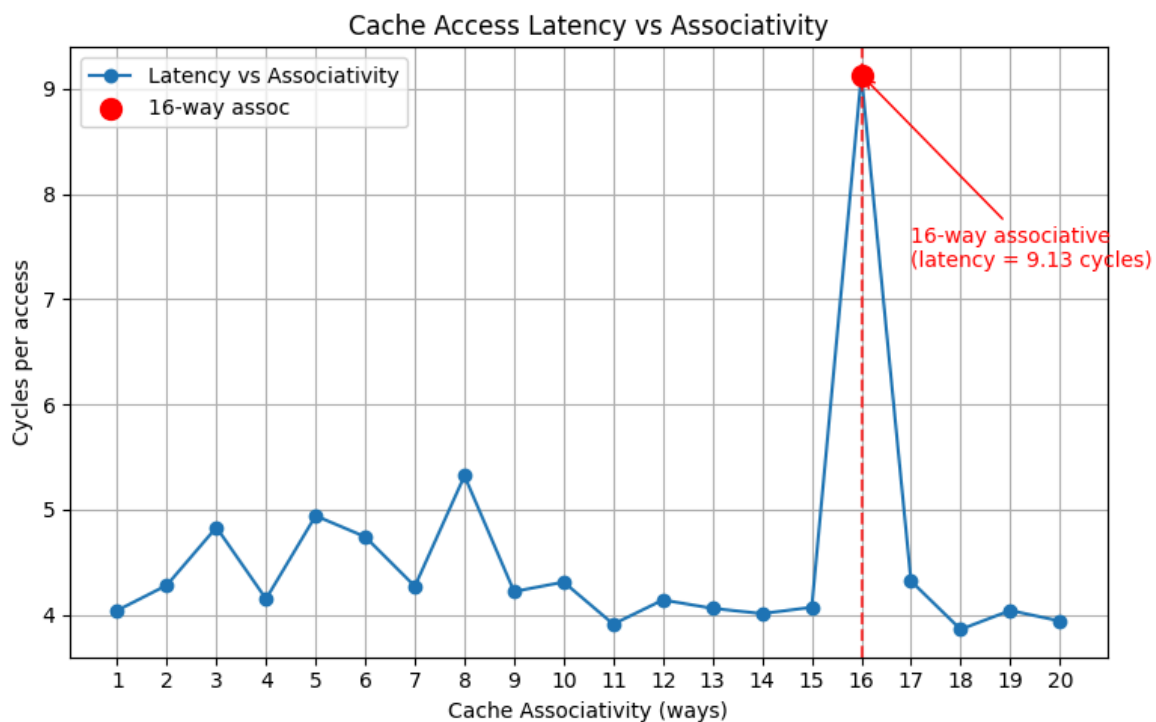
Из графика видно, что ассоциативность L2 кэша – 8-way.

Также график для L3:



Аналогично переходу L1 -> L2, есть (на графике не очень заметен, но он есть) резкий скачок (см. таблицу) задержки около 16 384 KB рабочего набора — это и свидетельствует о заполнении L3. Официальный объём L3 в спецификации процессора — 16 MB, что совпадает с нашим наблюдением.

Ассоциативность:



Из графика видно, что ассоциативность L3-кэша – 16-way. Латентность – ~50 циклов (тактов).

Вывод: получилось эмпирически вычислить примерные размеры, латентности и ассоциативности кэшей.

Уров ень	Эксп. объём	Специ ф. объём	Эксп. ассо ц.	Специ ф. ассоц.	Эксп. латентно сть	Специ ф.лат.
-------------	----------------	----------------------	---------------------	-----------------------	--------------------------	-----------------

L1D	32 KiB × 6 cores	32 KiB × 6 cores	8-way	8-way	≈4 циклов	≈4 циклов
L2	512 KiB × 6	512 KiB × 6	8-way	8-way	≈11 циклов	≈11 циклов
L3	16 MiB shared	16 MiB shared	16-way	16-way	≈70 циклов	≈50 циклов*