ФИО	Завьялов Никита Аркадьевич
Номер группы	M3138
Название работы	Кэш

2. Ссылка на репозиторий: https://github.com/NEKAfk/comp-ach
3. Инструментарий: Работа была выполнена с использованием среды разработки VSCode, язык программирования C++, компилятор g++ (Debian 12.2.0-14) 12.2.0.

4. Вывод программы:

LRU: hit perc. 99.9335% time: 3428471
pLRU: hit perc. 99.9335% time: 3428471

5. Параметры системы

ADDR_LEN	20
MEM_SIZE	(1LL << ADDR_LEN) = 1MB
CACHE_WAY	4
CACHE_LINE_SIZE	128
CACHE_LINE_COUNT	CACHE_WAY * CACHE_SETS_COUNT = 64
CACHE_SIZE	(CACHE_LINE_SIZE * CACHE_LINE_COUNT) = 8KB
CACHE_SETS_COUNT	(1 << CACHE_IDX_LEN) = 16
CACHE_IDX_LEN	ADDR_LEN – CACHE_TAG_LEN – CACHE_OFFSET_LEN = 4
CACHE_TAG_LEN	9
CACHE_OFFSET_LEN	log2(CACHE_LINE_SIZE) = 7
FL_LEN	pLRU: 3, LRU: 4

Ещё параметры:

ADDR1_BUS_LEN, ADDR2_BUS_LEN	ADDR_LEN = 20, ADDR_LEN-OFFSET = 13
DATA1_BUS_LEN, DATA2_BUS_LEN	16
C1_BUS_LEN, C2_BUS_LEN	3 bits, 2 bits

Были реализованы обе политики вытеснения. Кэш реализован при помощи двумерного массива, содержащего объекты класса CasheLine. Также был реализован класс Mem для выполнения обращений в RAM. Политики вытеснения реализованы так:

pLRU: когда мы обращаемся к кэш линии мы устанавливаем один бит флага на 1, если в ассоциативном блоке все линии имеют данный бит, равный единице, то они все обнуляются. Когда мы обращаемся в RAM, в первую очередь ищем не занятую линию, иначе линию с нулевым битом.

LRU: обновление флага: устанавливаем кэш линии максимальный номер, всем кэш линиям с номером большим предыдущего значения декрементируем номер. Когда обращаемся в RAM, в первую очередь ищем не занятую линию, иначе линию с минимальным номером.