

ВВЕДЕНИЕ В КЭШИ: МИРОАХИТЕКТУРА

Занятие №22
29 марта 2025



ШКОЛА СИНТЕЗА
ЦИФРОВЫХ СХЕМ

ПРИ ПАРТНЕРСТВЕ





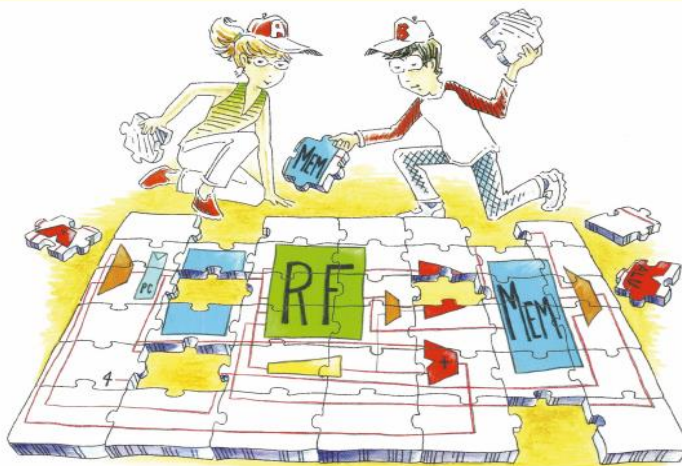
Александр Кириченко
RTL разработчик Yadro

МАТЕРИАЛ ЛЕКЦИИ

Глава 8. Системы памяти.

Параграф 8.3. Кеш-память.

Цифровая схемотехника и архитектура компьютера: RISC-V



Дэвид М. Харрис
Сара Л. Харрис



МИЭМ

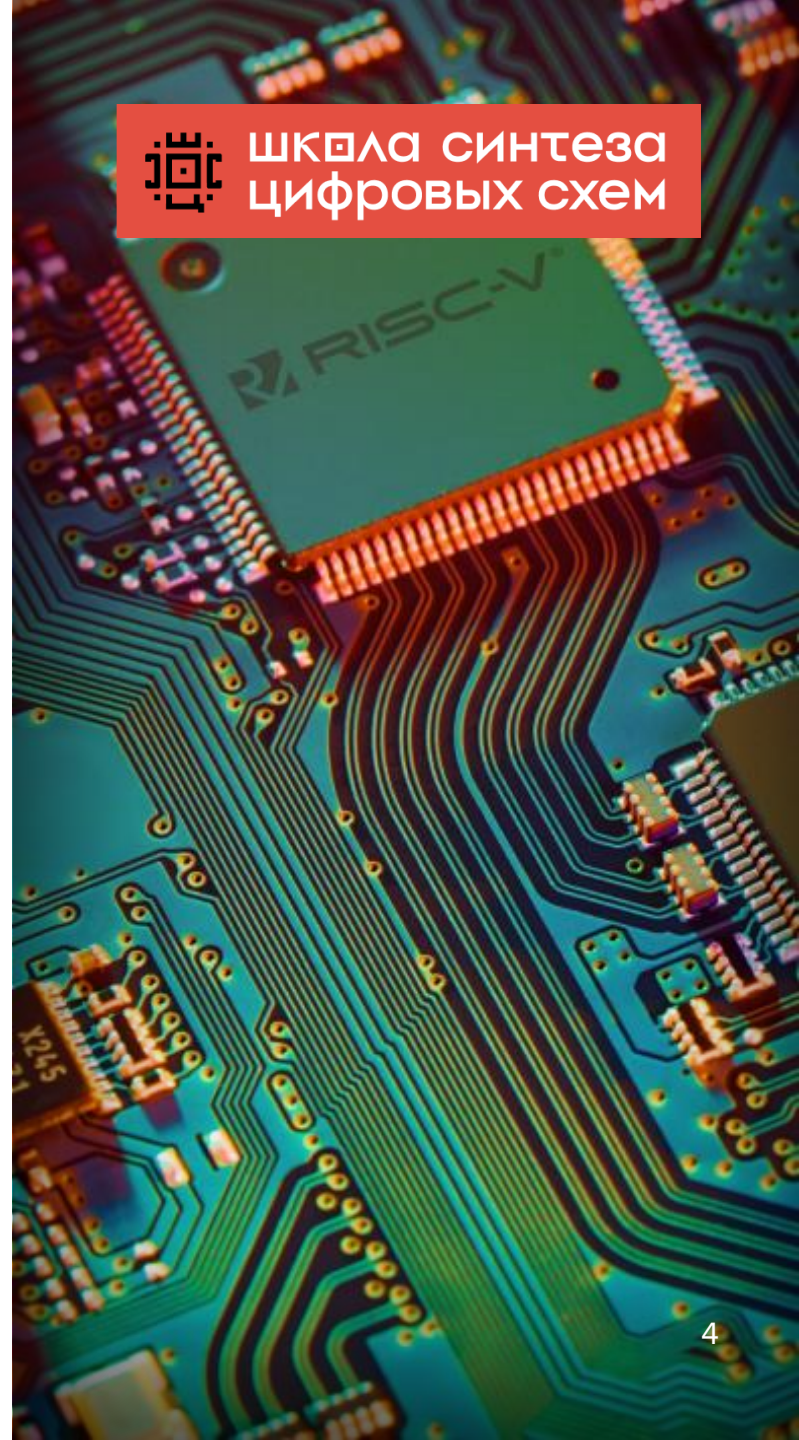
ДМК
ИЗДАТЕЛЬСТВО



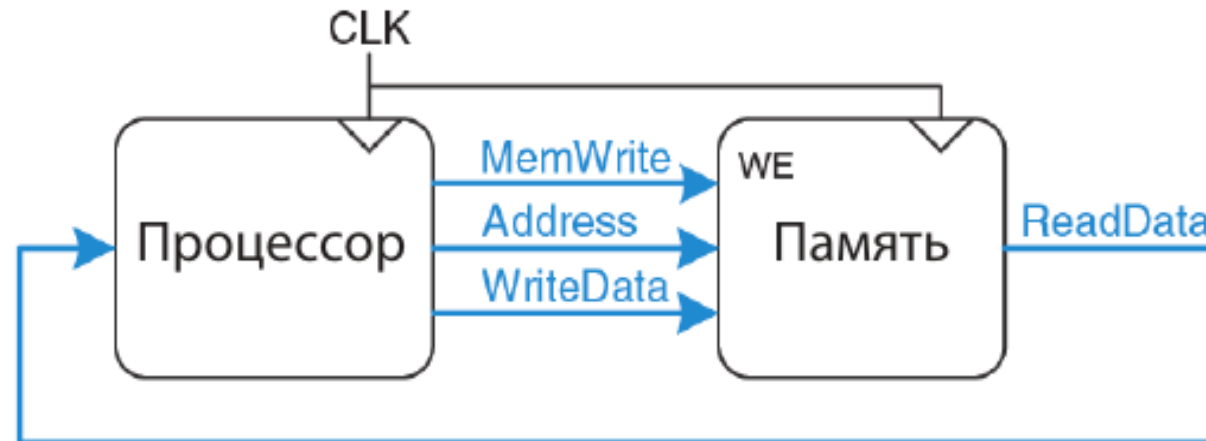
ШКОЛА СИНТЕЗА
ЦИФРОВЫХ СХЕМ

ПЛАН ЛЕКЦИИ

- Проблема производительности подсистемы памяти;
- Кеш-память: хранение данных (какие данные хранятся, характеристики кеша);
- Виды кеш-памяти: прямого отображения, наборно-ассоциативный кеш, полностью ассоциативный;
- Замещение данных в кеше;
- Демонстрация: schoolriscv с полностью ассоциативным кешем инструкций;
- Улучшенная кеш-память.



ВВЕДЕНИЕ



MIRISCV для памяти данных использует интерфейс аналогичный изображенному на рисунке.

В настоящее время оперативная память типа DRAM (Dynamic Random Access Memory, динамическая память с произвольным доступом) медленнее процессора от 10 до 100 раз.

ПРИНЦИПЫ ВРЕМЕННОЙ И ПРОСТРАНСТВЕННОЙ ЛОКАЛЬНОСТИ

На примере книг в библиотеке.

Временная локальность — если вы только что использовали книгу, то, вероятно, она скоро понадобится вам снова.

Пространственная локальность — когда вам понадобилась определенная книга, то, вероятно, вас заинтересуют и другие книги с той же полки.



ПОДСИСТЕМЫ ПАМЯТИ

Компьютерная память в основном построена на базе динамической (DRAM) и статической (SRAM) памяти.

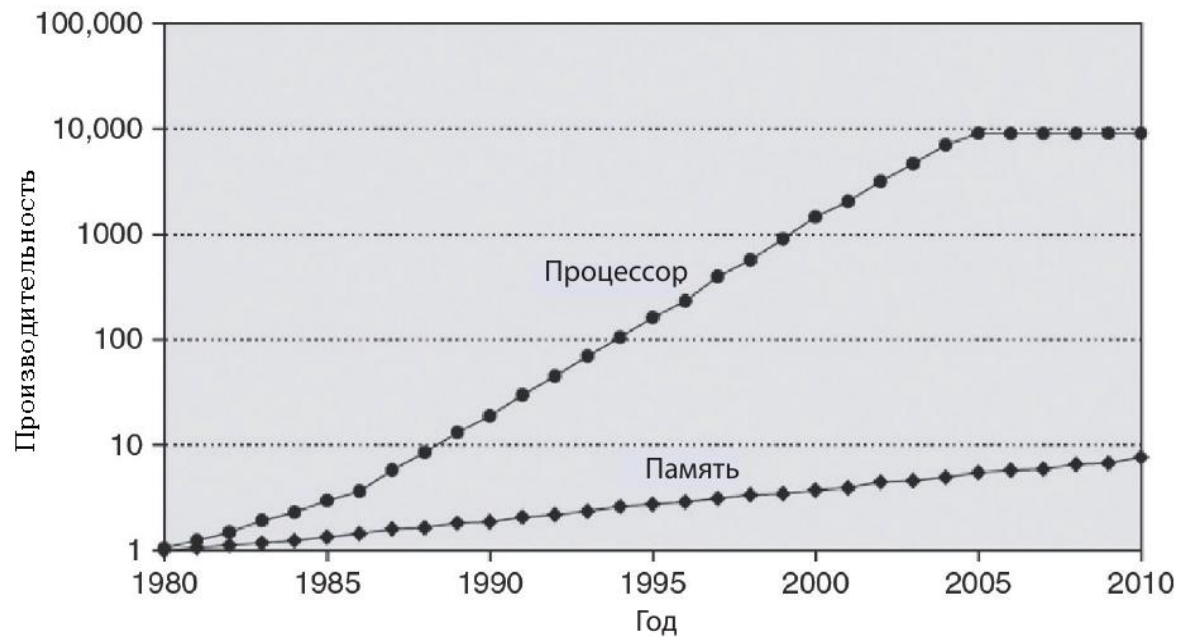


ПРОБЛЕМА ПРОИЗВОДИТЕЛЬНОСТИ ПАМЯТИ



Цены на DRAM падали в среднем на 15–25 % в год.

К сожалению, скорость работы самих микросхем DRAM возрастала только на 7% в год, в то время как производительность процессоров возрастала на 25–50 % в год.



Разница в производительности процессора и памяти (график из книги Hennessy and Patterson, Computer Architecture: A Quantitative Approach)

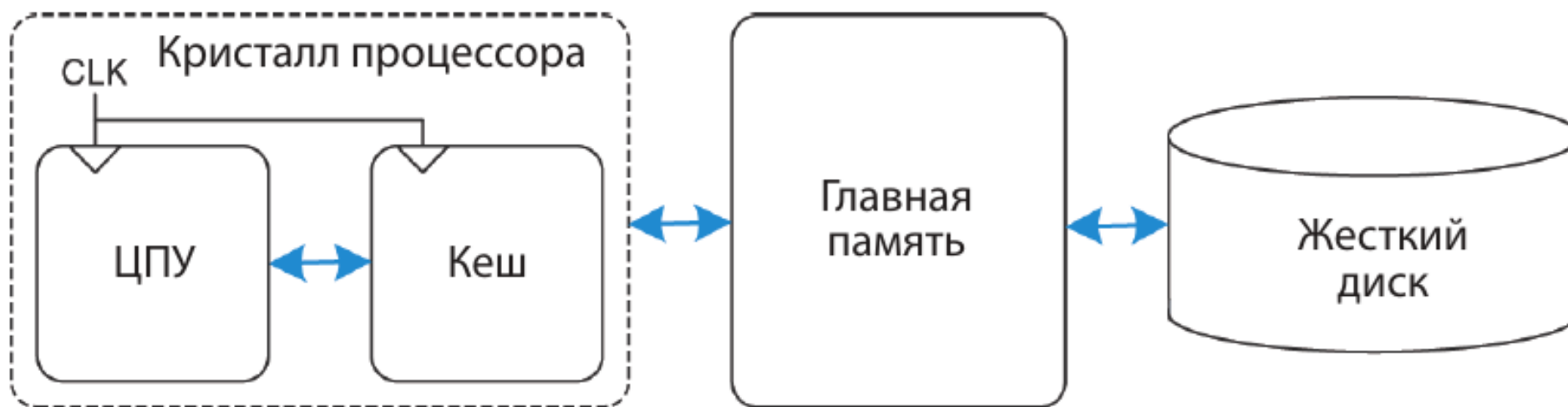
ПОПАДАНИЕ И ПРОМАХ КЕША

Если процессор запрашивает данные, которые уже находятся в кеше, то он получает их очень быстро. Это называется **попаданием в кеш** (cache hit).

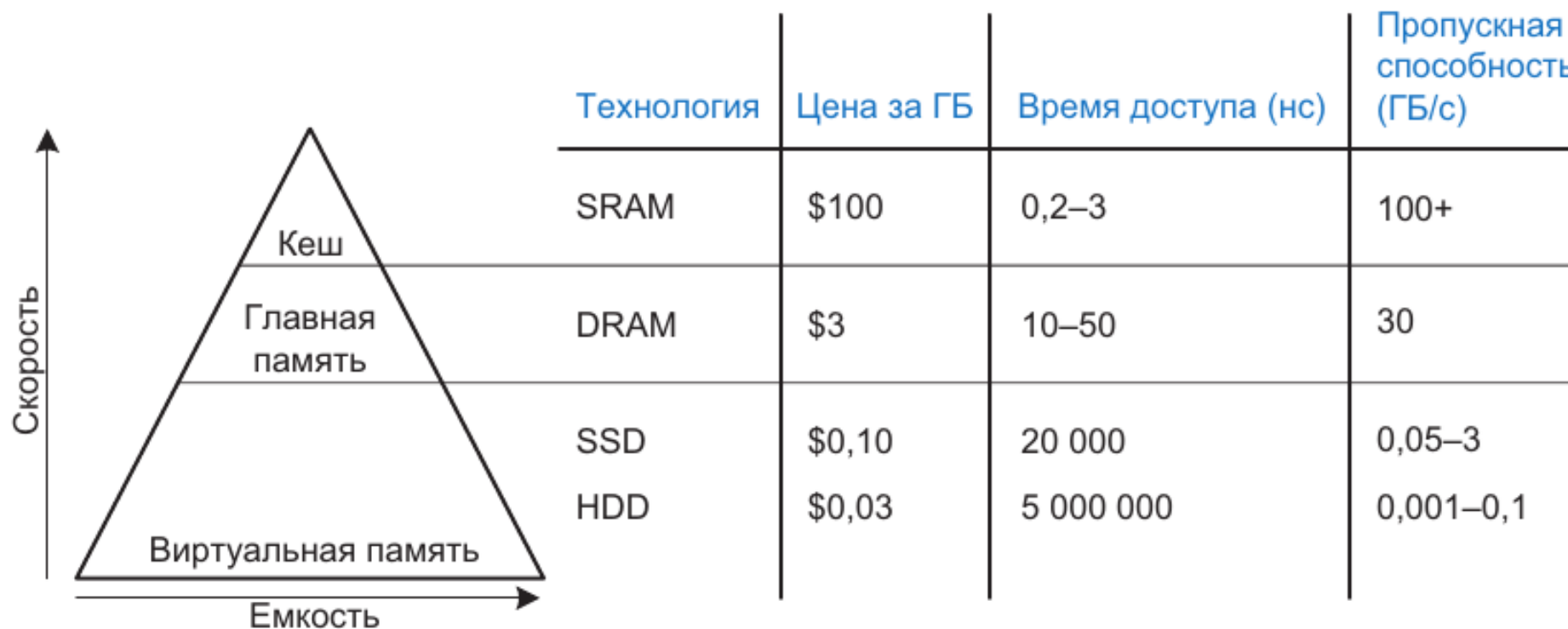
В противном случае процессор вынужден читать данные из оперативной памяти (DRAM). Это называется **промахом кеша**, кеш-промахом или промахом доступа в кеш (cache miss).



ТИПИЧНАЯ ИЕРАРХИЯ ПАМЯТИ



СООТНОШЕНИЕ СТОИМОСТИ И СКОРОСТИ РАЗНЫХ ТИПОВ ПАМЯТИ



АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ПАМЯТИ



Мерами измерения производительности систем памяти являются доля попаданий (hit rate) или промахов (miss rate), а также среднее время доступа.

$$\text{Доля промахов} = \frac{\text{Количество промахов}}{\text{Общее количество доступов к памяти}} = 1 - \text{Доля попаданий};$$

$$\text{Доля попаданий} = \frac{\text{Количество попаданий}}{\text{Общее количество доступов к памяти}} = 1 - \text{Доля промахов}.$$

Предположим, что программа содержит 2000 команд обращения к данным (чтения и записи), но только 1250 из этих команд нашли запрошенные ими данные в кеш-памяти. Остальным 750 командам пришлось получать данные из оперативной памяти или с жесткого диска. Чему равны доли промахов и попаданий в кеш-память в этом случае?



АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ПАМЯТИ

Среднее время доступа (average memory access time, AMAT) — это среднее время, которое процессор тратит, ожидая доступа к памяти при выполнении команд загрузки или сохранения данных.

$$AMAT = t_{cache} + MR_{cache}(t_{MM} + MR_{MM}t_{VM})$$

где t_{cache} , t_{MM} и t_{VM} — это времена доступа к кешу, оперативной памяти и диску соответственно, а MR_{cache} и MR_{MM} — это доли промахов кеша и оперативной памяти.

Предположим, что компьютерная система имеет память всего с двумя уровнями иерархии: кешем и оперативной памятью. Чему равно среднее время доступа?

Время доступа и процент промахов заданы в таблице ниже.

Уровень памяти	Время доступа в тактах	Доля промахов
Кеш-память	1	10 %
Оперативная память	100	0 %



АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ПАМЯТИ

Среднее время доступа к памяти в 11 тактов означает, что процессор тратит 10 тактов на ожидание данных на каждый такт реального использования этих данных. Какой процент промахов в кеш необходим для уменьшения среднего времени доступа к памяти до 1,5 такта при времени доступа к памяти заданном ниже?

Уровень памяти	Время доступа в тактах	Доля промахов
Кеш-память	1	10 %
Оперативная память	100	0 %

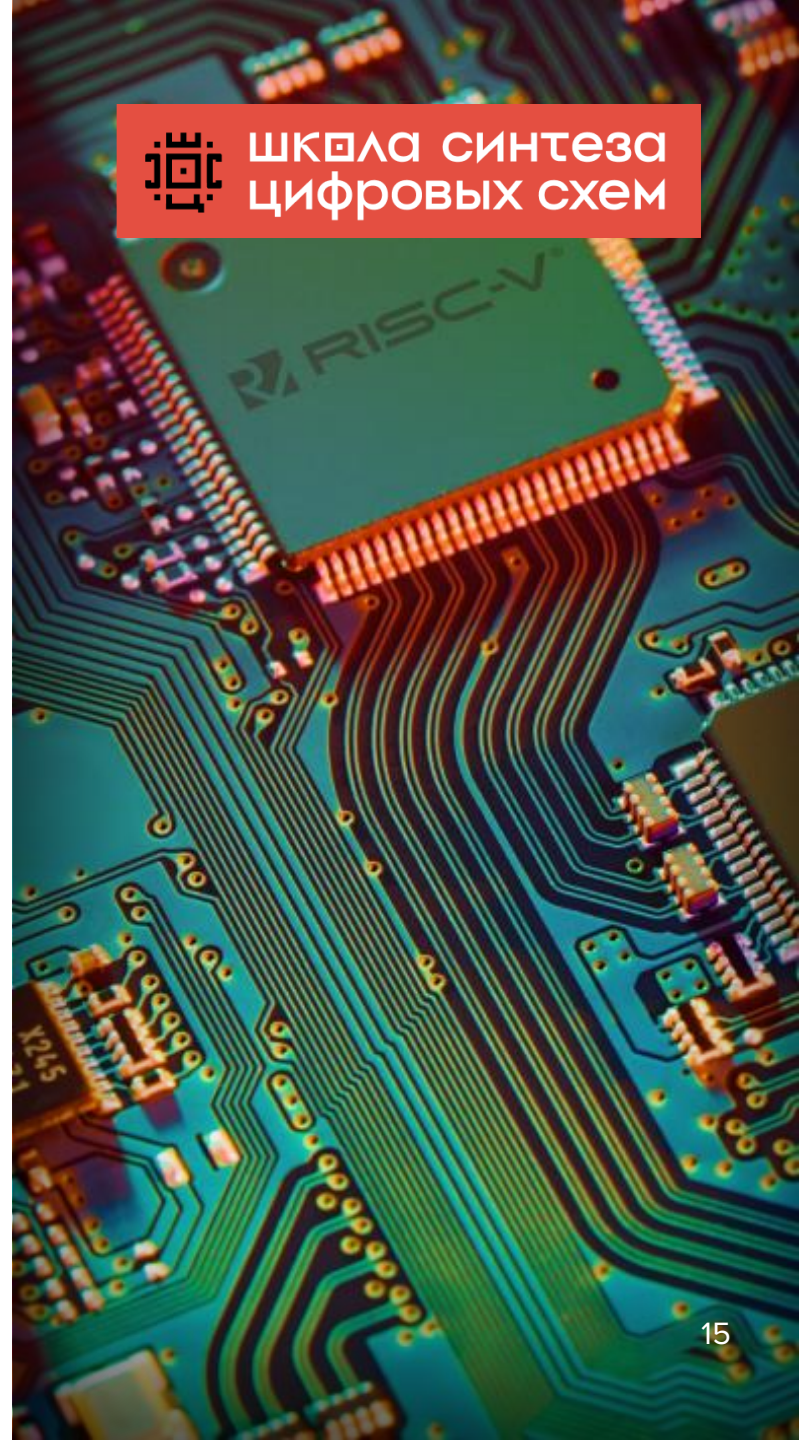
Найти MR_{cache} с помощью формулы:

$$AMAT = t_{cache} + MR_{cache}(t_{MM} + MR_{MM}t_{VM})$$

КЕШ-ПАМЯТЬ: ХАРАКТЕРИСТИКИ КЕША

Кеш-память характеризуется:

- емкостью **C** (capacity);
- количеством наборов **S** (set);
- длиной строки, иногда называемой размером блока **b** (block);
- количеством строк или блоков **B**;
- степенью ассоциативности **N**.



КЕШ-ПАМЯТЬ: КАКИЕ ДАННЫЕ ХРАНИТ КЕШ?

Временная локальность — процессор, вероятно, еще раз обратится к тем данным, которые он недавно использовал.

Пространственная локальность — если процессор обращается к каким-либо данным, то, вероятно, ему понадобятся и данные расположенные рядом. Для реализации этого принципа данные в кеше размещаются строками.

Строка кеша (cache line или «блок кеша» (cache block) — это совокупность из нескольких подряд идущих слов данных прочитанных из оперативной памяти.

Количество слов в строке b называется длиной строки (line size или block size). Кеш емкостью C содержит $V = C / b$ строк.





ВИДЫ КЕШ-ПАМЯТИ

- Прямое отображения (direct mapped cache);
- Наборно-ассоциативный кеш с N секциями (N-way set associative cache);
- Полностью ассоциативный кеш (fully associative cache).

Любой кеш состоит из S наборов, каждый из которых содержит одну или несколько строк (блоков данных). Количество строк в наборе зависит от типа кеша. Для кеша прямого отображения оно минимально. Полностью ассоциативный кеш хранит данные одним набором.

Взаимосвязь между адресом данных в оперативной памяти и расположением этих данных в кеше называется **отображением**.

В кеше **прямого отображения** (direct mapped cache) каждый набор содержит только одну строку (один блок), так что кеш содержит $S = B$ наборов.

В случае же **наборно-ассоциативного кеша с N секциями** (N-way set associative cache) каждый набор состоит из N строк. Количество наборов в этом случае равно $S = B / N$, а данные могут оказаться в любой из N строк этого набора.

В отличие от кеша прямого отображения и наборно-ассоциативного кеша, **полностью ассоциативный кеш** (fully associative cache) имеет только один набор ($S = 1$), и данные могут оказаться в любой из B строк этого набора.

ПОДСИСТЕМА ПАМЯТИ ПРОЦЕССОРА RISC-V



Address: 32-бита.

ReadData и WriteData: 32-бит слово.

Слова в памяти выровнены по 4-байтной границе. Память содержит 2^{30} слов.

Емкость кеша $C = 8$ слов.

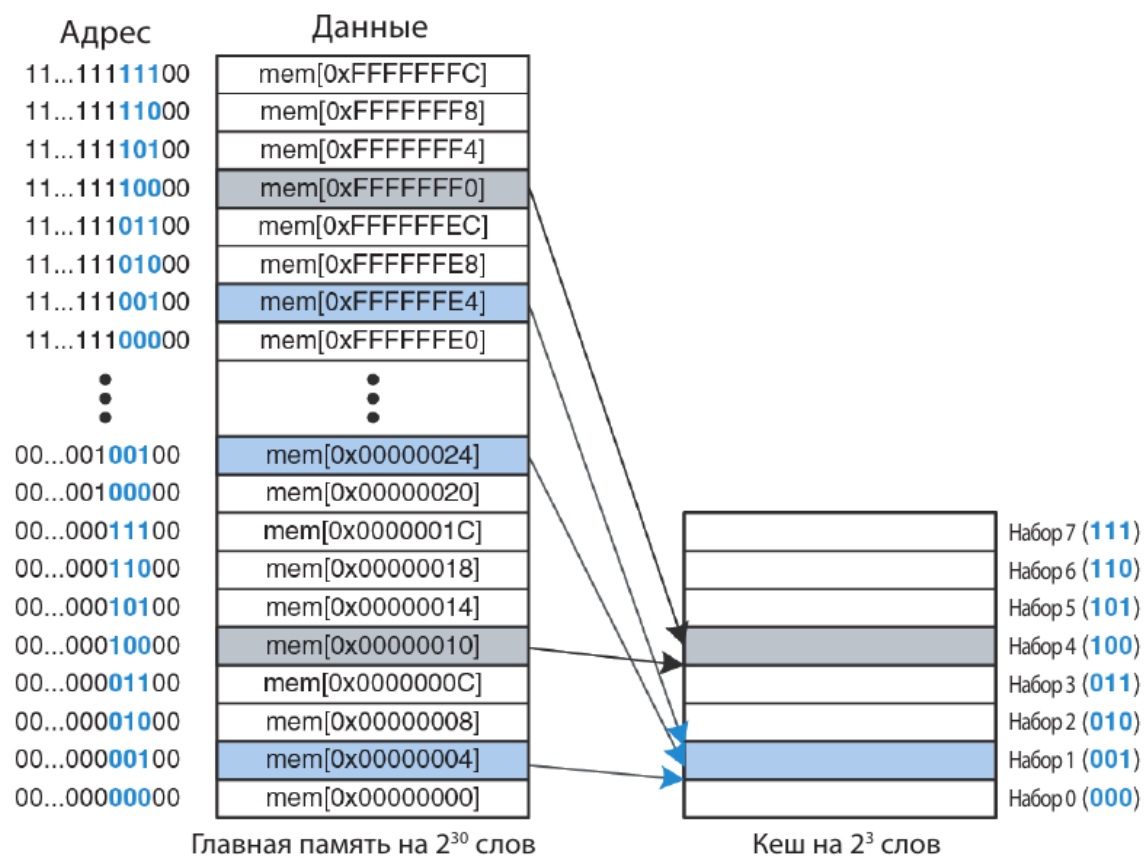
Длина строки $b = 1$ слово.



ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

Каждый набор содержит только одну строку (блок данных), так что $S = B$ наборов и строк.

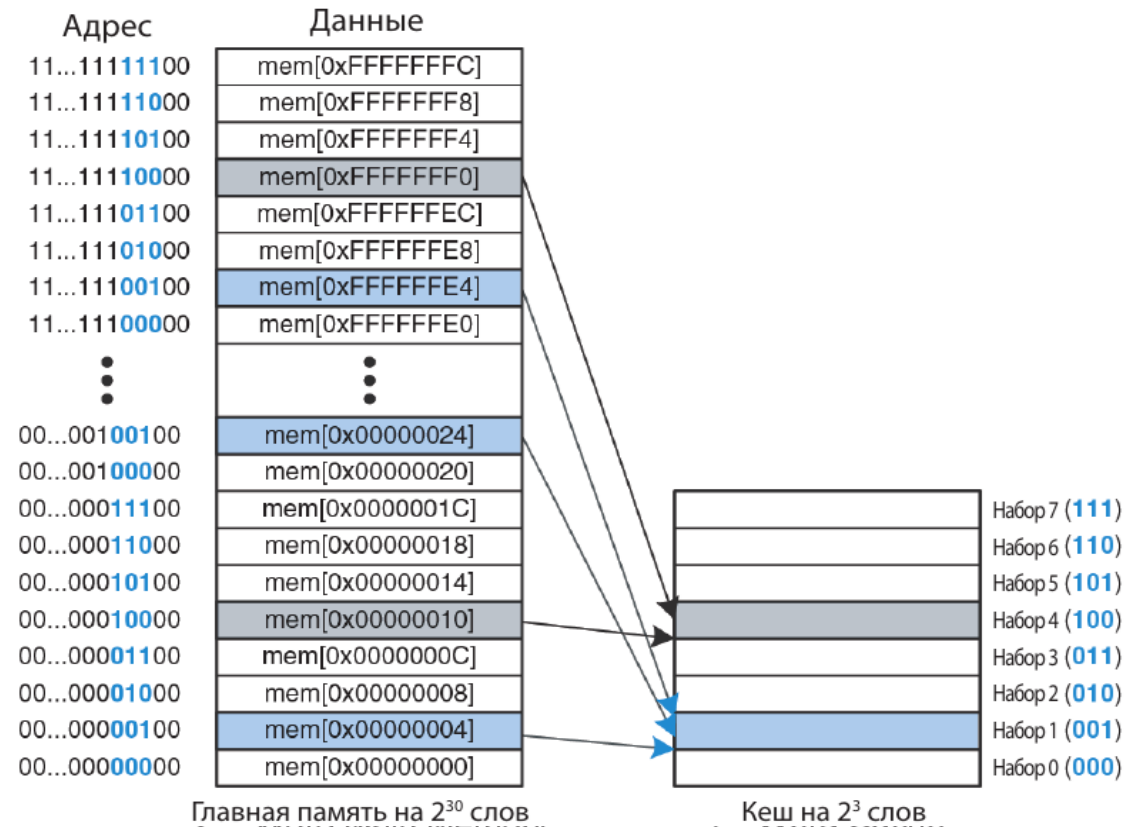
Отображение оперативной памяти на кеш прямого отображения





ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

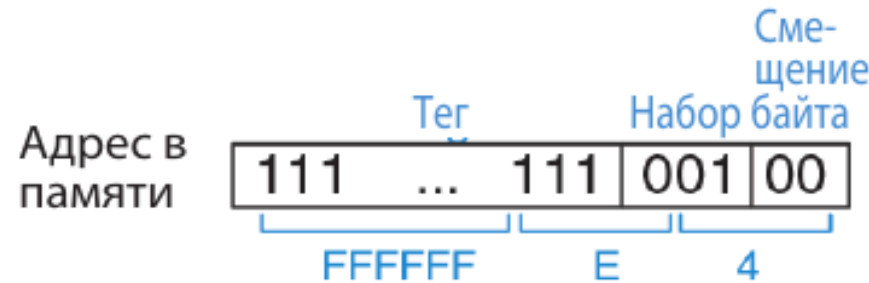
В какой набор кеша будет отображено слово с адресом 0x00000014? Назовите другой адрес, который отображается в этот же самый набор.





ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

Части адреса 0xFFFFFE4 при отображении в кеш.



Два младших бита адреса называются **байтовым смещением** (byte offset), поскольку они указывают на номер байта внутри слова.

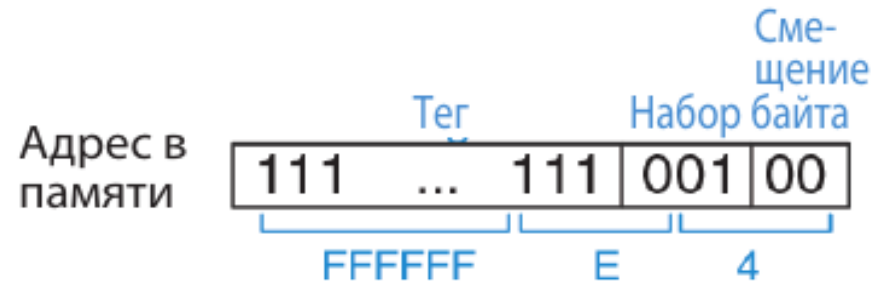
Следующие три бита называются **индексом** (cache index) или **номером набора** (set bits), так как они указывают на номер набора, в который отображается этот адрес.

Оставшиеся 27 бит **тега** указывают на адрес слова, которое в текущий момент находится в конкретном наборе кеша.



ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

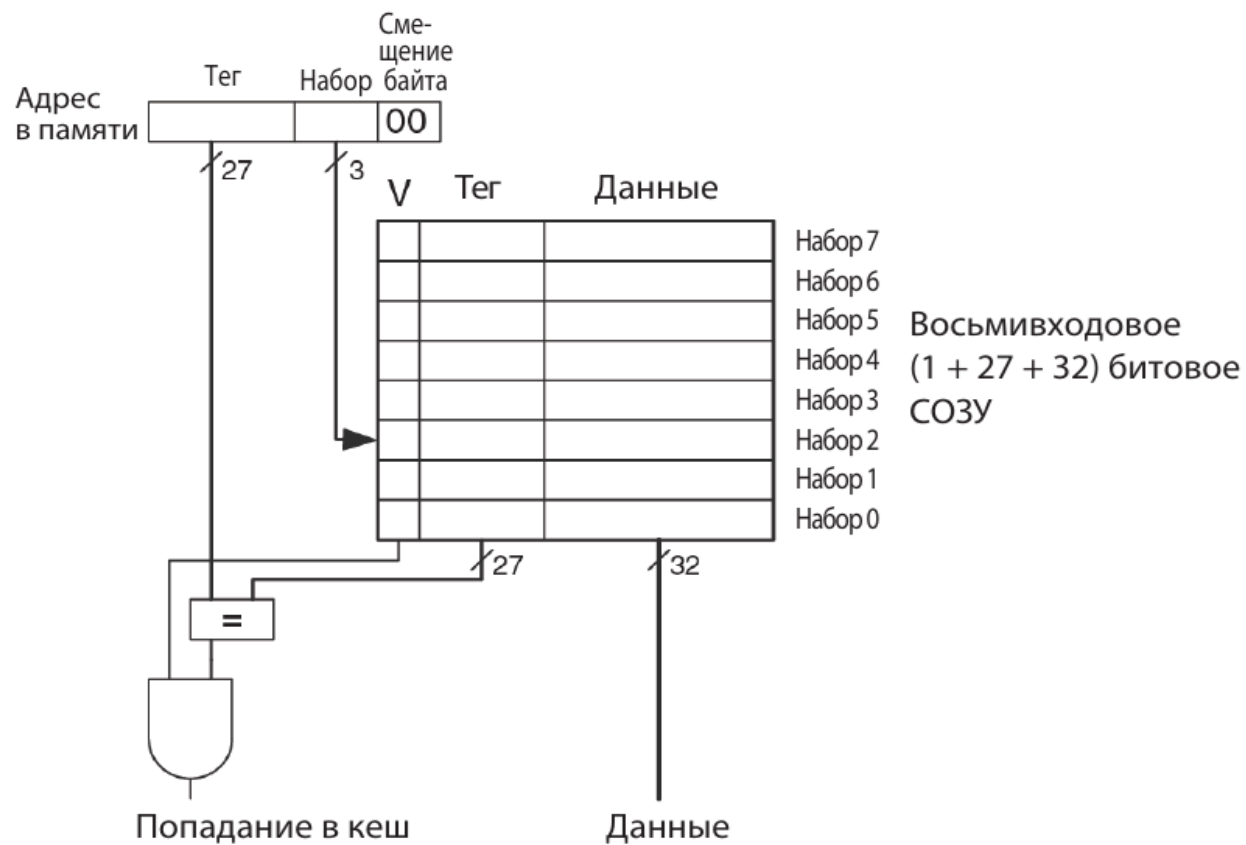
Найти количество битов тега и номера набора (индекса) для кеш-памяти прямого отображения с 1024 (2^{10}) наборами и длиной строки, равной одному слову. Размер адреса - 32 бита.





ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

Блок-схема аппаратной реализации кеша прямого отображения с восемью наборами.





ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

Пример временной локальности для кеш-памяти прямого отображения.

Циклы – это типичный источник временной и пространственной локальности данных в приложениях. Используя кеш с восемью ячейками, покажите, чему будет равно содержимое кеша после выполнения следующего небольшого цикла на языке ассемблера RISC-V. Считайте, что изначально кеш пуст. Каким будет процент промахов?

```
        addi s0, zero, 5
        addi s1, zero, 0
LOOP:   beq s0, zero,
DONE
        lw s2, 4(s1)
        lw s3, 12(s1)
        lw s4, 8(s1)
        addi s0, s0, -1
        j LOOP
DONE:
```




ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ



Ответ: процент промахов кеша: $3 / 15 = 20 \%$.



ВИДЫ КЕШ-ПАМЯТИ: КЕШ ПРЯМОГО ОТОБРАЖЕНИЯ

При обращении к адресам памяти отображающихся в одну и ту же строку кеша возникает конфликт. Новые данные **вытесняют** (evict), перезаписывают данные в кеше.

Пример конфликтов: чему будет равен процент промахов при выполнении следующего цикла при наличии кеша прямого отображения емкостью 8 слов? Считайте, что изначально кеш пуст.

```

    addi s0,
zero, 5

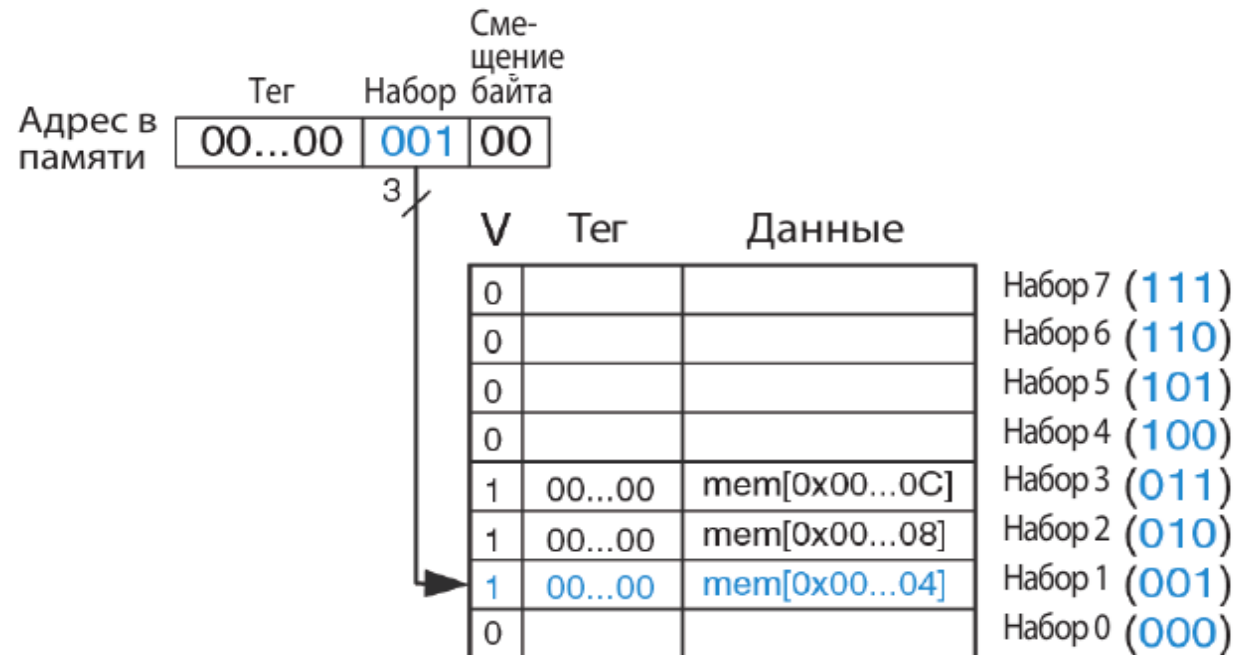
    addi s1,
zero, 0

LOOP: beq s 0,
zero, DONE

    lw s2,
0x4(s1)

    lw s4,
0x24(s1)

```



ВИДЫ КЕШ-ПАМЯТИ: МНОГОСЕКЦИОННЫЙ НАБОРНО-АССОЦИАТИВНЫЙ КЕШ

N-секционный наборно-ассоциативный кеш (N-way set associative cache) уменьшает количество конфликтов путем расширения набора до N строк.

Можно сказать, что кеш прямого отображения – это односекционный наборно-ассоциативный кеш.

Число N называют степенью ассоциативности кеша.

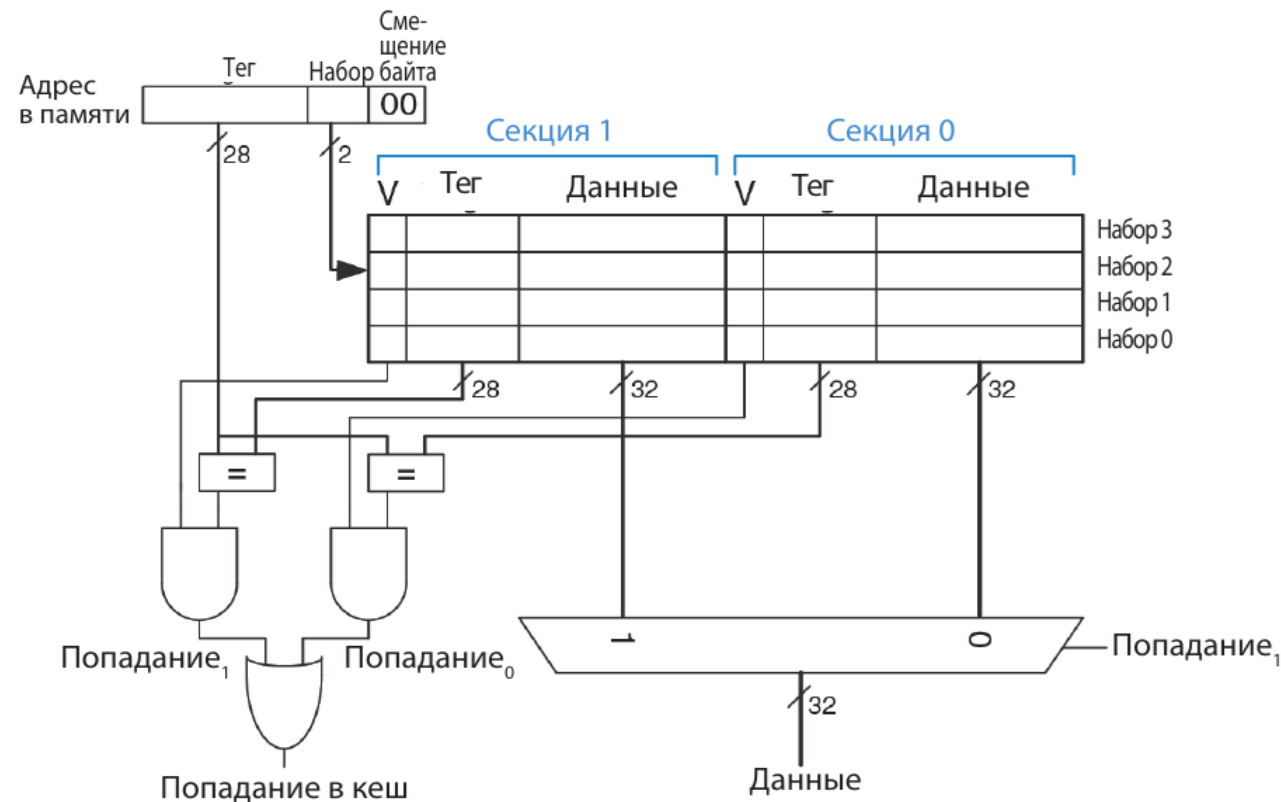




ВИДЫ КЕШ-ПАМЯТИ: МНОГОСЕКЦИОННЫЙ НАБОРНО-АССОЦИАТИВНЫЙ КЕШ

Блок-схема аппаратной реализации наборно-ассоциативного кеша емкостью $C = 8$ слов с $N = 2$ секциями.

Число наборов $S = B / N$. Число строк $B = C / b$. $b = 1$ слово. Отсюда $B = 8$, $S = 8 / 2 = 4$.





ВИДЫ КЕШ-ПАМЯТИ: МНОГОСЕКЦИОННЫЙ НАБОРНО-АССОЦИАТИВНЫЙ КЕШ

Чему будет равен процент промахов при выполнении следующего цикла при наличии многосекционного наборно-ассоциативного кеша емкостью 8 слов? Считайте, что изначально кеш пуст.

```

        addi s0,
zero, 5
        addi s1,
zero, 0
LOOP: beq s 0,
zero, DONE
        lw s2,
0x4(s1)
        lw s4,
0x24(s1)
        addi s0, s0,

```





ВИДЫ КЕШ-ПАМЯТИ: ПОЛНОСТЬЮ АССОЦИАТИВНЫЙ КЕШ

Полностью ассоциативный кеш (fully associative cache) состоит из одного набора с B секциями, где B – количество строк (блоков данных). Адрес памяти может быть отображен в строку любой из этих секций.

Можно сказать, что полностью ассоциативный кеш – это B -секционный наборно-ассоциативный кеш с одним набором.

Если емкость кеша $C = 8$ слов, то число строк $B = 8$. Так как размер блока $b = 1$ слово.

Число секций $N = B = 8$. Поэтому число наборов $S = B / N = 8 / 8 = 1$.



Полностью ассоциативные кешы обеспечивают при прочих равных условиях минимально возможное количество конфликтов, но требуют еще больше аппаратуры для дополнительных сравнений тегов.

Из-за этого они применяются лишь в относительно маленьких кешах.

ДЛИНА СТРОКИ КЕША

Чтобы воспользоваться пространственной локальностью, в кеш-памяти используют строки, содержащие несколько слов, расположенных последовательно в памяти.

Требуется больше времени на чтение данных в строку после промаха, т. к. из памяти необходимо будет прочитать не одно, а несколько слов. Время, требуемое для загрузки данных в строку кеша после промаха, называется **ценой промаха** (miss penalty).



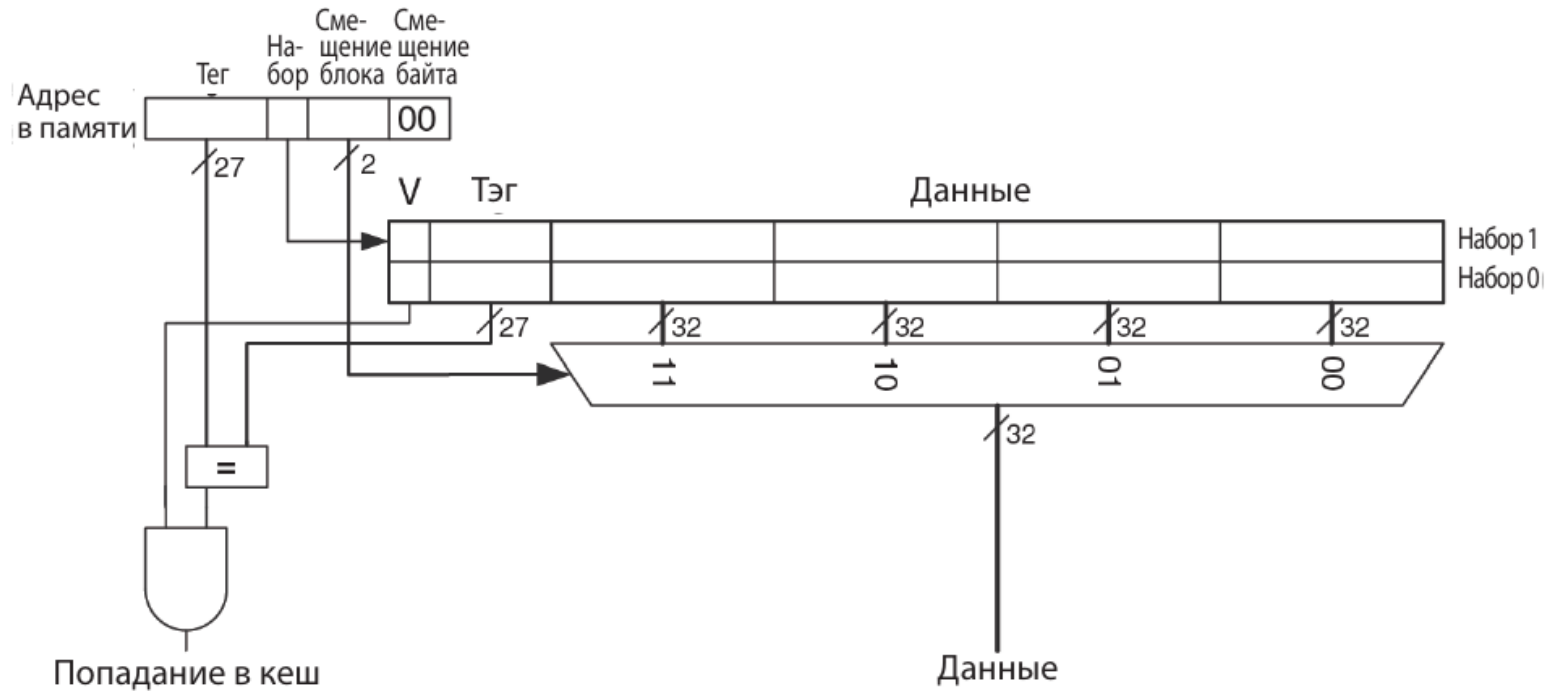
Клик правой кнопкой —
изменить изображение



ДЛИНА СТРОКИ КЕША

Кеш прямого отображения емкостью $C = 8$, с длиной строки $b = 4$ словам.

Число строк $V = C / b = 2$. Число наборов $S = V$.

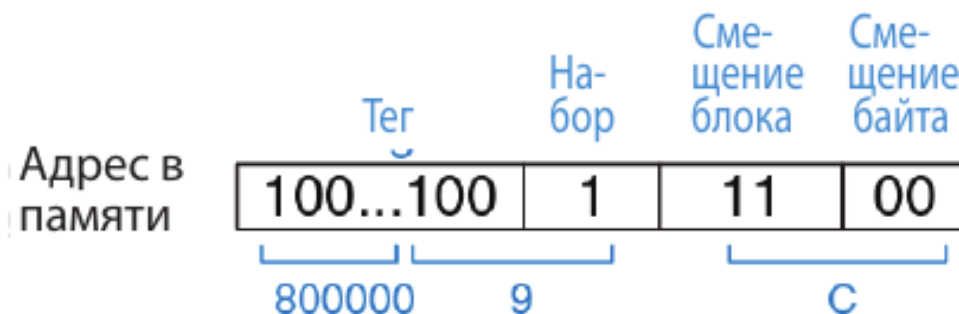


Мультиплексор управляется 2 битами адреса, которые называются **смещением в строке** (line offset или block offset).

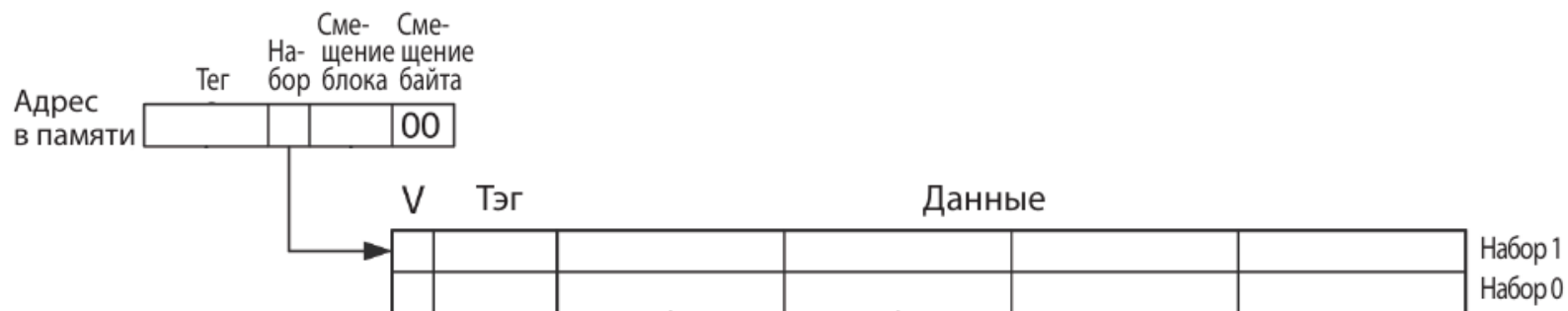
ДЛИНА СТРОКИ КЕША



Части адреса 0x8000009C при доступе в кеш прямого отображения с длиной строки $b = 4$.



Слово по адресу 0x8000009C отображается в третье слово набора 1.





ДЛИНА СТРОКИ КЕША

Пример временной локальности для кеш-памяти прямого отображения емкостью $C = 8$, с длиной строки $b = 4$ словам.

Покажите, чему будет равно содержимое кеша после выполнения следующего небольшого цикла на языке ассемблера RISC-V. Считайте, что изначально кеш пуст. Каким будет процент промахов?

```

    addi s0, zero,
5
    addi s1, zero,
0
LOOP: beq s0, zero,
DONE
    lw s2, 4(s1)
    lw s3, 12(s1)
    lw s4, 8(s1)
    addi s0, s0,

```

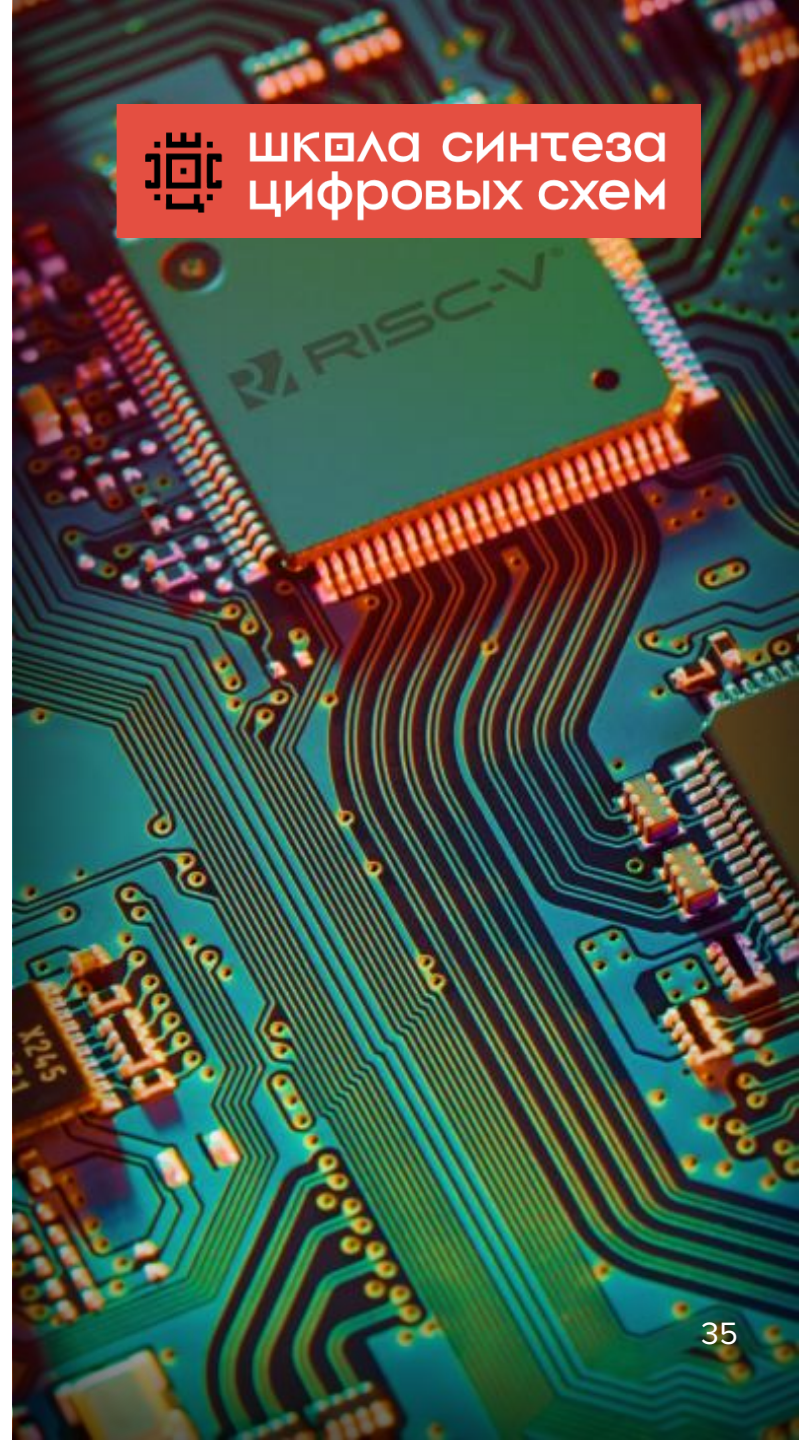


ПРОМЕЖУТОЧНЫЕ ИТОГИ

Кеш представляет собой двумерный массив. Строки этого массива называют наборами, а колонки – секциями. Каждый элемент массива содержит строку (т.е. блок данных) и связанные с ней тег и бит достоверности.

Кеш характеризуется:

- емкостью **C**;
- длиной строки **b** и соответствующим количеством строк $\mathbf{B} = C / b$;
- количеством строк в наборе (**N**).



ПРОМЕЖУТОЧНЫЕ ИТОГИ



Способы организации кеш-памяти.

Способ организации	Количество секций (N)	Количество наборов (S)
Прямого отображения	1	B
Наборно-ассоциативный	$1 < N < B$	B / N
Полностью ассоциативный	B	1

Емкость кеша, степень ассоциативности, количество наборов и длина строки обычно кратны степени двойки. Это позволяет однозначно соотносить биты адреса с битами тега, индекса (номера набора) и смещения в строке.

ЗАМЕЩЕНИЕ ДАННЫХ В КЕШ-ПАМЯТИ

Кеш прямого отображения: каждый адрес всегда отображается в одну и ту же строку одного и того же набора, поэтому новые данные просто перезаписывают старые.

В наборно-ассоциативной и полностью ассоциативной кеш-памяти нужно решать, какую именно из нескольких строк в наборе вытеснить.

Учитывая принцип временной локальности большинство кешей используют стратегию замены **редко используемых данных** (least recently used, LRU).





ЗАМЕЩЕНИЕ ДАННЫХ В КЕШ-ПАМЯТИ

В двухсекционном наборно-ассоциативном кеше бит использования U (от англ. used) содержит номер той секции в наборе, которая дольше не использовалась.

Секция 1				Секция 0			
V	U	Тег	Данные	V	Тег	Данные	
0	0			0			Набор 3 (11)
0	0			0			Набор 2 (10)
0	0			0			Набор 1 (01)
0	0			0			Набор 0 (00)

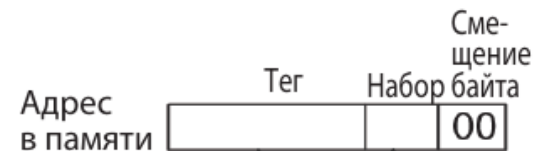
При увеличении числа секций отслеживать самые редко используемые строки становится сложно. Для упрощения реализации секции часто делят на две группы, а бит использования указывает на ту группу, которая дольше не использовалась. Замещается случайная строка в той группе, на которую указывает бит использования.

Такая стратегия называется **псевдо-LRU** (pseudo-LRU) и на практике она работает достаточно хорошо.

ЗАМЕЩЕНИЕ ДАННЫХ В КЕШ-ПАМЯТИ

Покажите содержимое двухсекционного наборно-ассоциативного кеша емкостью $C = 8$ слов после выполнения следующего кода. Используйте стратегию замещения LRU, длину строки $b = 1$ слов. Считайте, что изначально кеш пуст.

```
addi t0,  
zero, 0  
  
lw s1,  
0x4(t0)  
  
lw s2,  
0x24(t0)
```



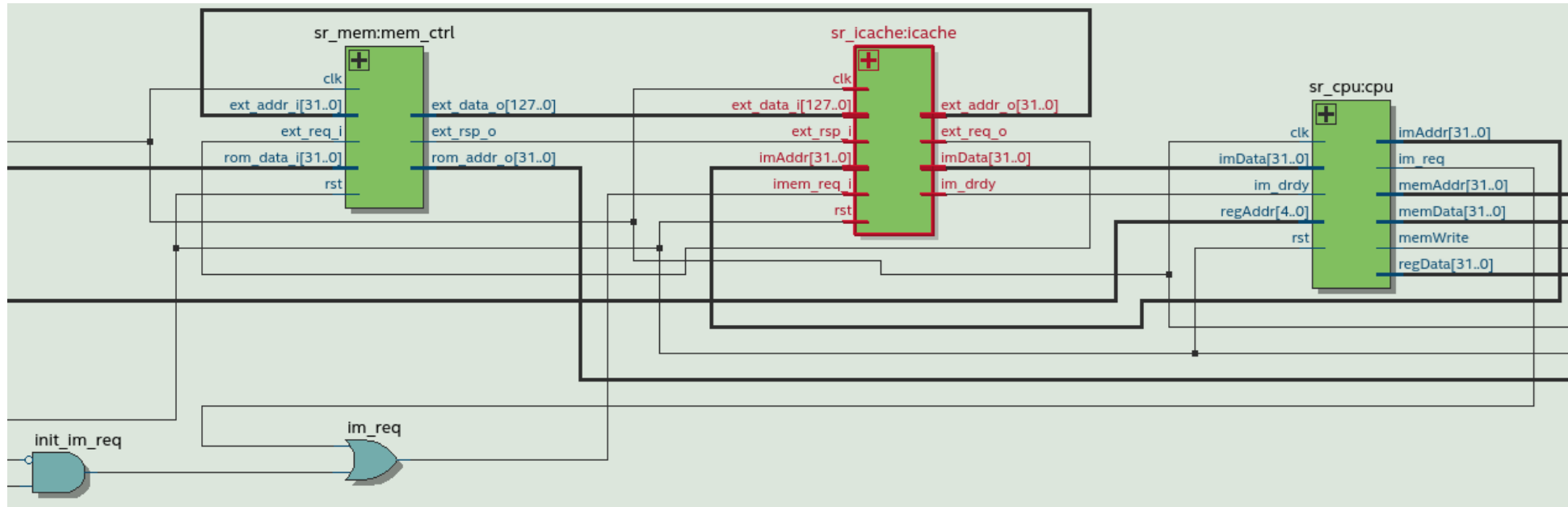
Секция 1				Секция 0				
V	U	Тег	Данные	V	Тег	Данные		
0	0			0			Набор3	(11)
0	0			0			Набор2	(10)
1	0	00...010	mem[0x00...24]	1	00...000	mem[0x00...04]	Набор1	(01)
0	0			0			Набор0	(00)

(a)

Секция 1				Секция 0				
V	U	Тег	Данные	V	Тег	Данные		
0	0			0			Набор3	(11)
0	0			0			Набор2	(10)
1	1	00...010	mem[0x00...24]	1	00...101	mem[0x00...54]	Набор1	(01)
0	0			0			Набор0	(00)

(b)

ПРИМЕР: schoolriscv_cache



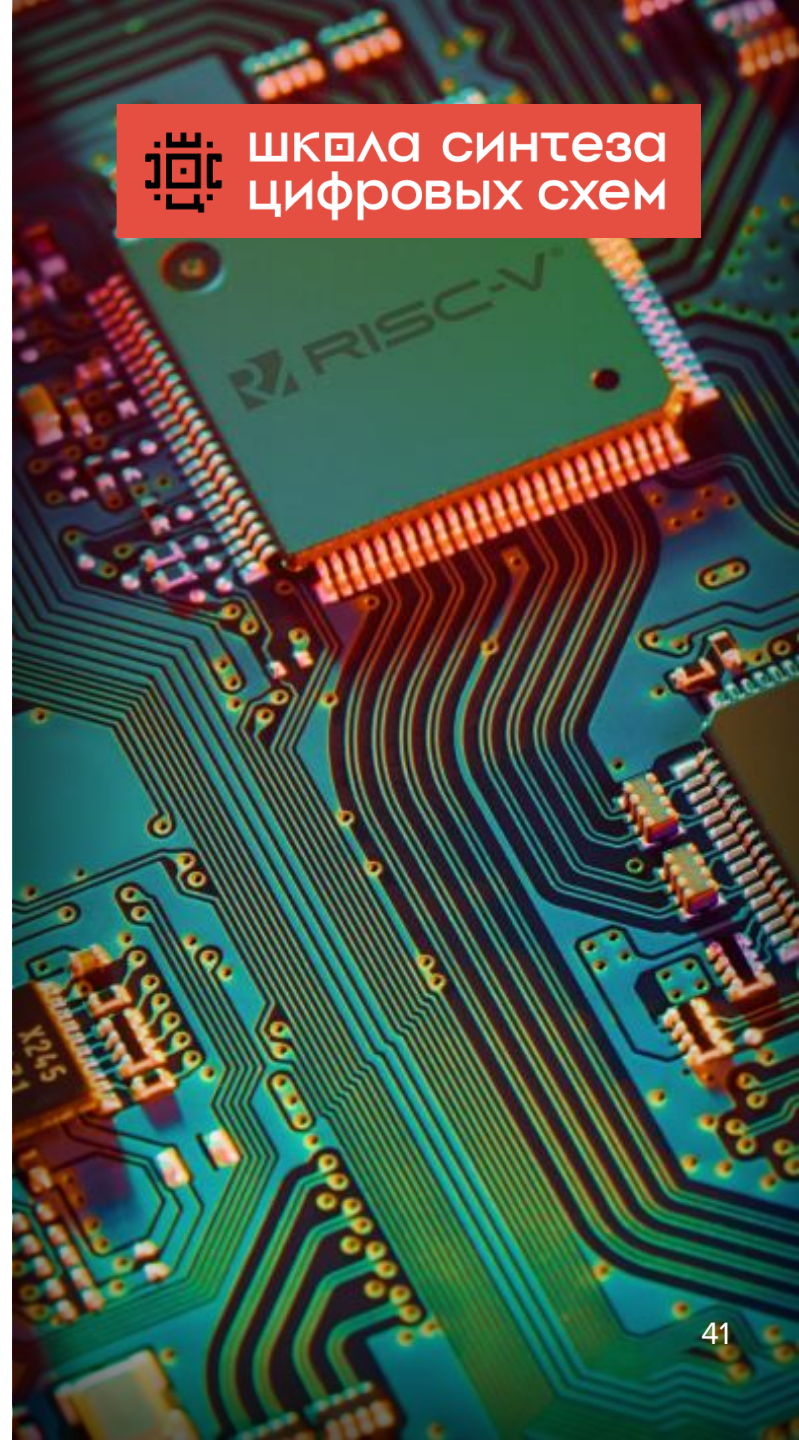
Пример процессора с полностью ассоциативным кешем инструкций.

Параметры кеша:

емкость $C = 16$ слов, длина строки $b = 4$ слова, число строк/секций $B = N = 4$, число наборов $S = 1$.

УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ

- Двухуровневый кеш
- Характеристики кеша и частота промахов
- Стратегии записи в память в системе с кешем



УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: ДВУХУРОВНЕВЫЙ КЕШ



Чем больше размер кеша, тем меньше у него будет частота промахов. Но увеличение размера памяти снижает ее быстродействие.

Выход: использование нескольких уровней кеша, которые в разы отличаются по размеру.





УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: ДВУХУРОВНЕВЫЙ КЕШ

Быстродействие системы с двухуровневым кешем.

Используйте систему со временем доступа 1, 10 и 100 тактов для кеша L1, кеша L2 и основной памяти соответственно. Предположим, что кеши L1 и L2 имеют процент промахов 5 % и 20 % соответственно. В частности, из 5 % обращений без кеша L1 20 % из них также не попадают в кеш L2.

Какое **среднее время доступа к памяти** (average memory access time, AMAT) в такой системе?

$$AMAT = t_{cache} + MR_{cache}(t_{MM} + MR_{MM}t_{VM})$$

где t_{cache} , t_{MM} , и t_{VM} — это времена доступа к кешу, оперативной памяти и диску соответственно, а MR_{cache} и MR_{MM} — это доли промахов кеша и оперативной памяти.

Формула выше написана для трехуровневой системы памяти, где используется только один уровень кеш-памяти и самая медленная память — это виртуальная память.

УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: СОКРАЩЕНИЕ ЧАСТОТЫ ПРОМАХОВ



Долю промахов кеша можно сократить, изменяя его емкость, длину строки и/или ассоциативность.

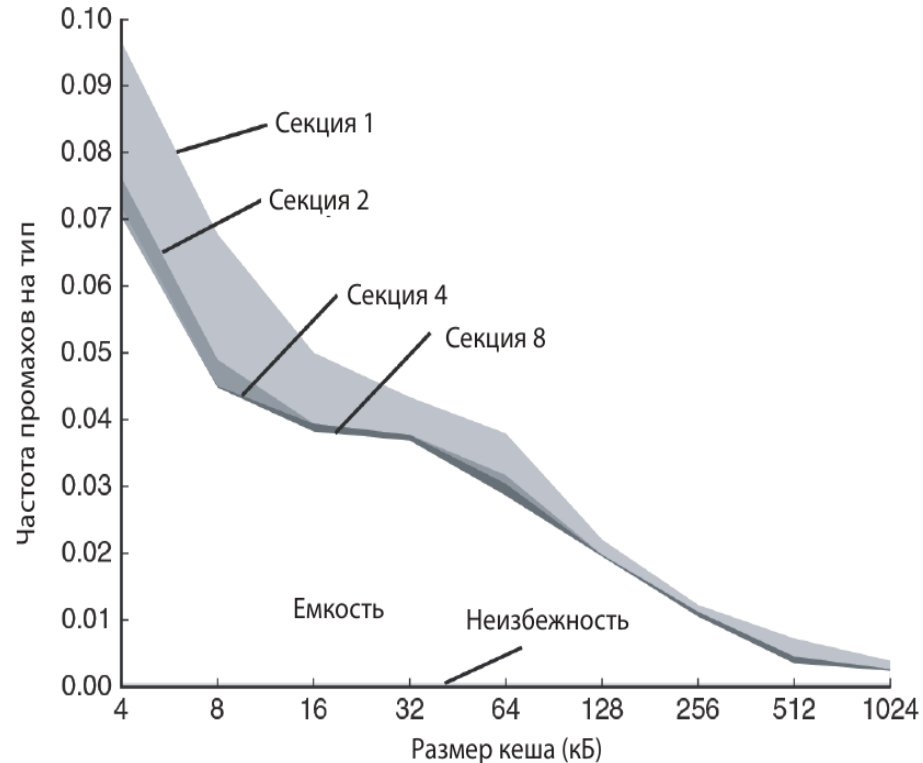
Причины промахов:

- неизбежные промахи (compulsory misses);
- промахи из-за недостаточной емкости (capacity misses);
- промахи из-за конфликтов (conflict misses).

Увеличение размера кеша может сократить промахи из-за конфликтов и промахи из-за недостатка емкости, но никак не повлияет на количество неизбежных промахов.

С другой стороны, увеличение длины строки может сократить количество неизбежных промахов (благодаря пространственной локальности данных), но одновременно может увеличить частоту промахов из-за конфликтов, поскольку большее количество адресов будет отображаться на один и тот же набор.

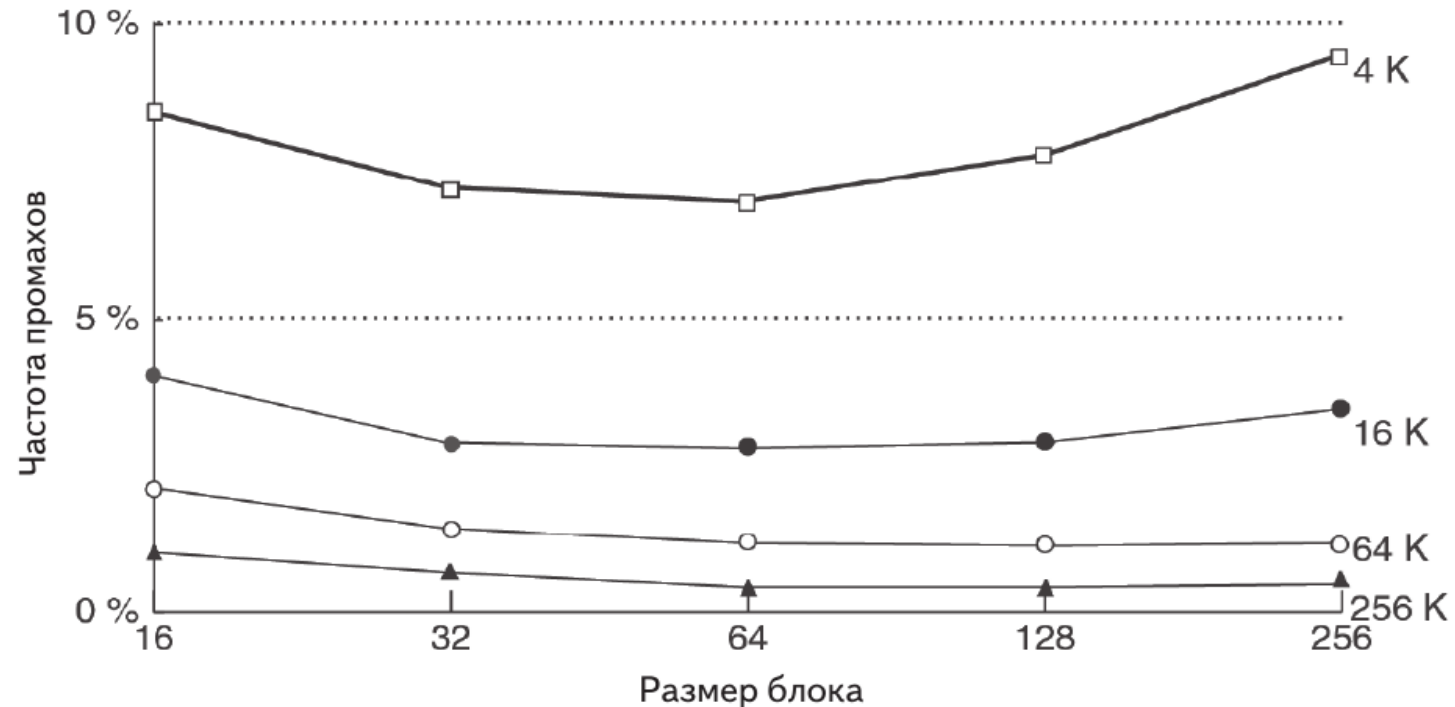
УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: СОКРАЩЕНИЕ ЧАСТОТЫ ПРОМАХОВ



На рисунке представлена зависимость частоты промахов от размера и ассоциативности кеша на тестах **SPEC2000** (график из книги Hennessy and Patterson, Computer Architecture: A Quantitative Approach, 5th ed., Morgan Kaufmann, 2012)

Системы памяти настолько сложны, что лучший способ оценивать их производительность – это запускать тестовые программы, варьируя параметры кеша.

УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: СОКРАЩЕНИЕ ЧАСТОТЫ ПРОМАХОВ



На рисунке представлена зависимость частоты промахов от длины строки и размера кеша на тестах

SPEC92 (график из книги Hennessy and Patterson, Computer Architecture: A Quantitative Approach, 5th ed., Morgan Kaufmann, 2012)

Частоту промахов можно уменьшить, используя пространственную локальность данных с помощью более длинных строк кеша. Но увеличение длины строки в кеше уменьшает количество наборов, что увеличивает вероятность конфликтов и может вызвать увеличение времени выполнения из-за более высокой **цены промаха** (miss penalty).

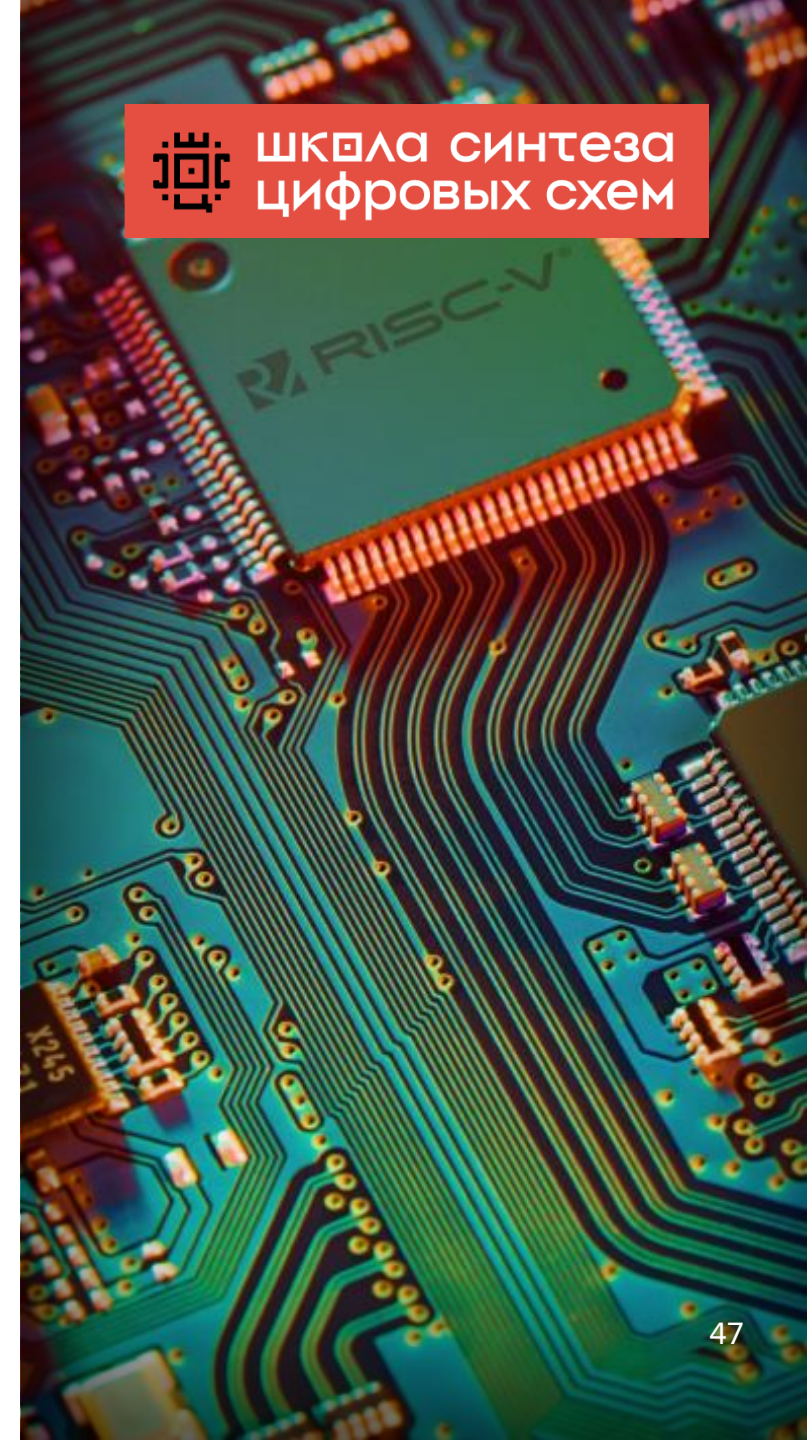
УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: СТРАТЕГИИ ЗАПИСИ

Запись в системе памяти с кешем выполняется примерно так же, как и чтение. При выполнении команды сохранения данных процессор сначала проверяет кеш. В случае промаха кеша соответствующая строка выбирается из оперативной памяти в кеш, а затем в нее записывается нужное слово. В случае попадания в кеш слово просто записывается в строку.

По стратегии записи кеша делятся на два типа:

- со **сквозной записью** (write-through);
- с **отложенной записью** (write-back).

В современных системах обычно используют кеша с отложенной записью.





УЛУЧШЕННАЯ КЕШ-ПАМЯТЬ: СТРАТЕГИИ ЗАПИСИ

Допустим, что длина строки кеша – четыре слова. Сколько обращений к оперативной памяти потребуется при выполнении кода, приведенного ниже, если используется стратегия сквозной записи, и сколько, если используется отложенная запись?

```
addi  t5,  
zero, 0  
  
sw  t1, 0(t5)  
  
sw  t2, 12(t5)  
  
sw  t3, 8(t5)  
  
sw  t4, 4(t5)
```

Решение.

Все четыре команды сохранения изменяют одну и ту же строку кеша (адреса идут друг за другом с шагом в 32 бита).

При сквозной записи каждая команда сохраняет слово в оперативную память, соответственно, потребуется четыре обращения к памяти. При отложенной записи потребуется только одно обращение – тогда, когда эта строка будет вытеснена из кеша.

КЕШ ПАМЯТЬ В КОММЕРЧЕСКИХ СИСТЕМАХ



CPU | Mainboard | Memory | SPD | Graphics | Bench | About

Processor

Name	Intel Core i7 4770K		
Code Name	Haswell	Max TDP	84.0 W
Package	Socket 1150 LGA		
Technology	22 nm	Core Voltage	1.252 V

Specification: Intel® Core™ i7-4770K CPU @ 3.50GHz

Family	6	Model	C	Stepping	3
Ext. Family	6	Ext. Model	3C	Revision	C0

Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, AES, AVX, AVX2, FMA3

Clocks (Core #0)

Core Speed	4242.05 MHz
Multiplier	x 43.0 (8 - 43)
Bus Speed	98.65 MHz
Rated FSB	

Cache

L1 Data	4 x 32 KBytes	8-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 256 KBytes	8-way
Level 3	8 MBytes	16-way

Selection: Socket #1 | Cores: 4 | Threads: 8

CPU | Caches | Mainboard | Memory | SPD | Graphics | Bench | About

Processor

Name	AMD Ryzen 7 1700		
Code Name	Summit Ridge	Max TDP	65 W
Package	Socket AM4 (1331)		
Technology	14 nm	Core Voltage	1.373 V

Specification: AMD Ryzen 7 1700 Eight-Core Processor

Family	F	Model	1	Stepping	1
Ext. Family	17	Ext. Model	1	Revision	ZP-B1

Instructions: MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, FMA3, SHA

Clocks (Core #0)

Core Speed	3990.48 MHz
Multiplier	x 40.0
Bus Speed	99.76 MHz
Rated FSB	

Cache

L1 Data	8 x 32 KBytes	8-way
L1 Inst.	8 x 64 KBytes	4-way
Level 2	8 x 512 KBytes	8-way
Level 3	2 x 8192 KBytes	16-way

Selection: Processor #1 | Cores: 8 | Threads: 16