

Механизм ассоциативной памяти

Выполнил
Сизов
Владислав
2ПКС-215

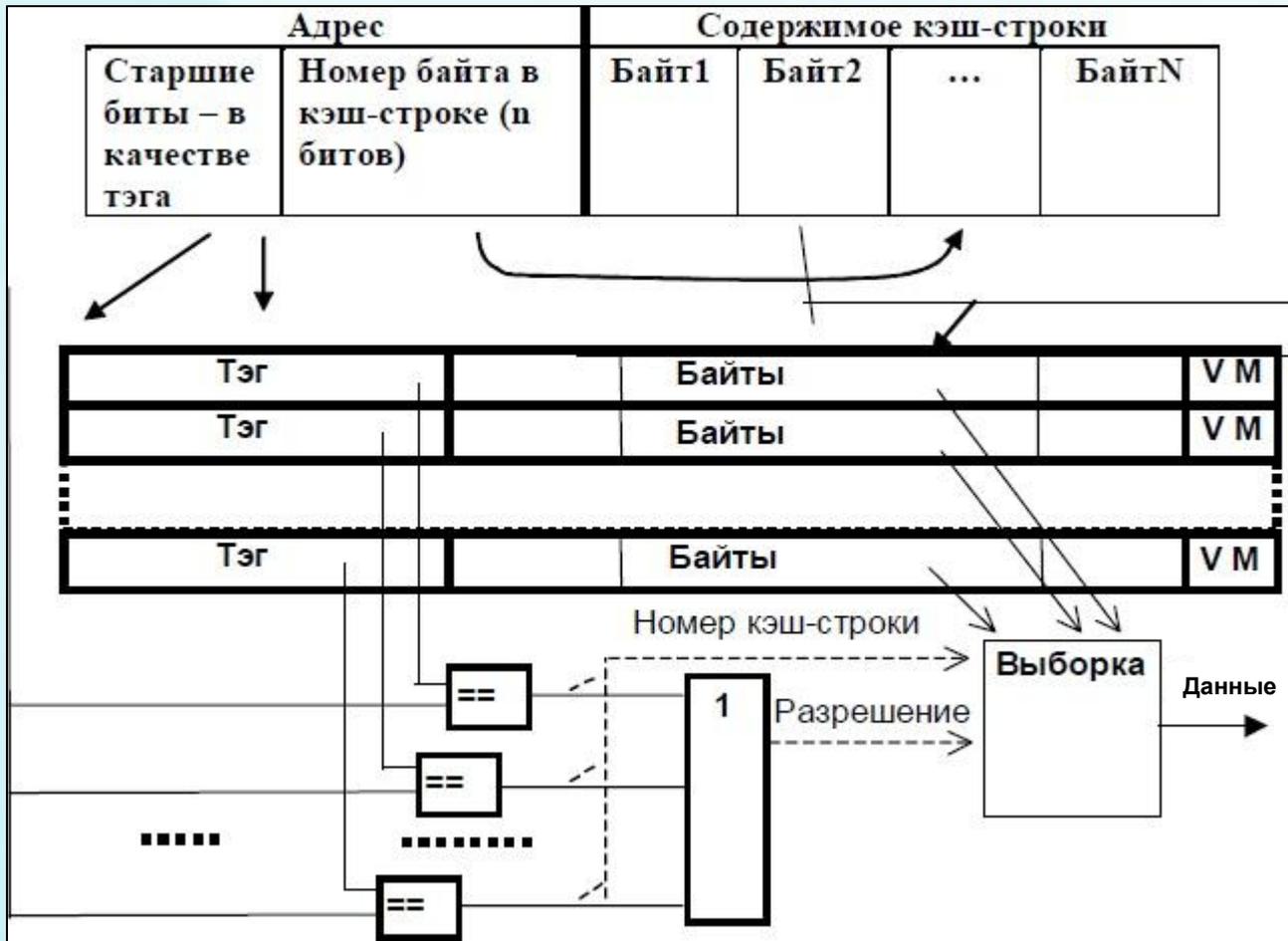
В ассоциативной памяти элементы выбираются не по адресу, а по содержимому. Единицу хранения информации в ассоциативной памяти является строка ассоциативной памяти. Каждая строка содержит два поля: поле и поле данных. Запрос на чтение к ассоциативной памяти словами можно выразить следующим образом: выбрать строку (строки), у которой (у которых) тег равен заданному значению.

При таком запросе возможен один из трех результатов:

- имеется в точности одна строка с заданным тегом
- имеется несколько строк с заданным тегом
- нет ни одной строки с заданным тегом

Об ассоциативной памяти говорят тогда, когда ассоциативная выборка данных из памяти поддерживается аппаратно. При записи в ассоциативную память элемент данных помещается в строку вместе с присущим этому элементу тегом. Ассоциативная память используется в КЭШ

Полностью ассоциативный КЭШ



В начале работы КЭШ-память пуста.

При выполнении первой же команды во время выборки ее код, а также еще несколько соседних байтов программного кода, - будут перенесены (медленно) в одну из строк КЭШа, и одновременно старшая часть адреса будет записана в соответствующий тег. Так происходит заполнение КЭШ-строки.

Если следующие выборки возможны из этого участка, они будут сделаны уже из КЭШа (быстро) - "КЭШ-попадание". Если же окажется, что нужного элемента в КЭШе нет, - "КЭШ-промахом". В этом случае обращение происходит к ОЗУ (медленно), и при этом одновременно заполняется очередная КЭШ-строка.

После формирования исполнительного адреса его старшие биты, образующие тег, аппаратно (быстро) и одновременно сравниваются с тегами всех КЭШ-строк. При этом возможны только две ситуации из трех, перечисленных ранее: либо все сравнения дадут отрицательный результат (КЭШ-промах), либо положительный результат сравнения будет зафиксирован в точности для одной строки (КЭШ-попадание).

При считывании, если зафиксировано КЭШ-попадание, младшие разряды адреса определяют позицию в КЭШ-строке, начиная с которой следует выбирать байты, а тип операции определяет количество байтов. Очевидно, что если длина элемента данных превышает один байт, то возможны ситуации, когда этот элемент (частями) расположен в двух (или более) разных КЭШ-строках, тогда время на выборку такого элемента увеличится. Противодействовать этому можно, выравнивая операнды и команды по границам КЭШ-строк, что и учитывают при разработке оптимизирующих трансляторов или при ручной оптимизации кода.

Если произошел КЭШ-промах, а в КЭШе нет свободных строк, необходимо заменить одну строку КЭШа на другую строку.

Основная цель стратегии замещения - удерживать в КЭШ-памяти строки, к которым наиболее вероятны обращения в ближайшем будущем, и заменять строки, доступ к которым произойдет в более отдаленном времени или вообще не случится. Очевидно, что оптимальным будет алгоритм, который замещает ту строку, обращение к которой в будущем произойдет позже, чем к любой другой строке-КЭШ.

К сожалению, такое предсказание практически нереализуемо, и приходится привлекать алгоритмы, уступающие оптимальному. Вне зависимости от используемого алгоритма замещения, для достижения высокой скорости он должен быть реализован аппаратными средствами.

Среди множества возможных алгоритмов замещения наиболее распространенными являются четыре, рассматриваемые в порядке уменьшения их относительной эффективности. Любой из них может быть применен в полностью ассоциативном КЭШ.

LRU - Least Recently Used

Этот алгоритм основан на замещении строк, к которой дольше всего не было обращений.

Наиболее известны два способа аппаратурной реализации этого алгоритма. В первом из них с каждой строкой КЭШ-памяти ассоциируют счетчик. К содержимому всех счетчиков через определенные интервалы времени добавляется единица. При обращении к строке ее счетчик обнуляется. Таким образом, наибольшее число будет в счетчике той строки, к которой дольше всего не было обращений и эта строка - первый кандидат на замещение.

Второй способ реализуется с помощью очереди, куда в порядке заполнения строк КЭШ-памяти заносятся ссылки на эти строки. При каждом обращении к строке ссылка на нее перемещается в конец очереди. В итоге первой в очереди каждый раз оказывается ссылка на строку, к которой дольше всего не было обращений. Именно эта строка прежде всего и заменяется.

FIFO - First In First Out

«Первый вошел, первый вышел». Здесь заменяется строка, дольше всего находившаяся в КЭШ-памяти. Алгоритм легко реализуется с помощью рассмотренной ранее очереди, с той лишь разницей, что после обращения к строке положение соответствующей ссылки в очереди не меняется.

LFU - Least Frequently Used

Заменяется та строка в КЭШ-памяти, к которой было меньше всего обращений. Принцип можно воплотить на практике, связав каждую строку со счетчиком обращений, к содержимому которого после каждого обращения добавляется единица. Главным претендентом на замещение является строка, счетчик которой содержит наименьшее число.

Простейший алгоритм - произвольный выбор строки для замены. Замещаемая строка выбирается случайным образом. Реализовано это может быть, например, с помощью счетчика, содержимое которого увеличивается на единицу с каждым тактовым импульсом, вне зависимости от того, имело место попадание или промах. Значение в счетчике определяет заменяемую строку.

Кроме тега и байтов данных в КЭШ-строке могут содержаться дополнительные служебные поля, среди которых в первую очередь следует отметить бит достоверности V (от valid - действительный имеющий силу) и бит модификации M (от modify - изменять, модифицировать).

При заполнении очередной КЭШ-строки V устанавливается в состояние "достоверно", а M - в состояние "не модифицировано". В случае, если в ходе выполнения программы содержимое данной строки было изменено, переключается бит M, сигнализируя о том, что при замене данной строки ее содержимое следует переписать в ОЗУ.

Если по каким-либо причинам произошло изменение копии элемента данной строки, хранимого в другом месте (например в ОЗУ), переключается бит V. При обращении к такой строке будет зафиксирован КЭШ-промах (несмотря на то, что тег совпадает), и обращение произойдет к основному ОЗУ.

Структура полностью ассоциативной КЭШ-памяти (ассоциативная память) реализуема только при малом количестве строк в КЭШе, т.е. при малом объеме КЭШа (практически не более 32...64 строк). КЭШ большего объема строят по другой структуре.

Ресурсы

- <http://perscom.ru>

Спасибо за внимание