

## ОРГАНИЗАЦИЯ КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВ УПРАВЛЕНИЯ С ПОЛНОСТЬЮ АССОЦИАТИВНОЙ КЭШ-ПАМЯТЬЮ

Бабаков Р.М.

Кафедра ЭВМ, ДонНТУ

### **Abstract**

*Babakov R.M. The organization of composite microprogram control devices with fully-associative cache-memory. The structural organization and principle of work of the cache-controller with fully-associative cache-memory in structure of the composite microprogram control device are considered.*

### **Введение**

Одним из центральных блоков цифровых устройств является устройство управления, в качестве которого может использоваться композиционное микропрограммное устройство управления (КМУУ) [1]. Одной из важных задач, возникающих при синтезе устройств управления, является повышение быстродействия работы схемы, что возможно за счет использования кэш-памяти для кэширования микрокоманд [2]. Одним из способов архитектурной организации кэш-памяти является кэш-память полностью ассоциативного типа [3, 4]. В настоящей работе рассматривается организация и принцип работы контроллера кэш-памяти полностью ассоциативного типа в составе композиционного микропрограммного устройства управления.

### **Принцип работы ассоциативной кэш-памяти**

Один из способов увеличения быстродействия КМУУ заключается в использовании в структуре КМУУ кэш-памяти, предназначенной для хранения часто используемых микрокоманд (МК) и выполненной на быстродействующих микросхемах типа SRAM [2]. При этом время доступа к кэш-памяти во много раз меньше, чем к УП, а размер кэш-памяти обычно составляет несколько процентов от общего размера УП. Это позволяет многократно повысить быстродействие схемы при незначительном увеличении аппаратных затрат и стоимости.

В [2] предлагается встраивать кэш-память так, как показано на рис. 1. Помимо самой кэш-памяти микрокоманд КПК в схеме присутствует кэш-контроллер К, управляющий процессом выборки микрокоманды из памяти на основании адреса микрокоманды, поступающего из счетчика адреса микрокоманд СЧАМК.

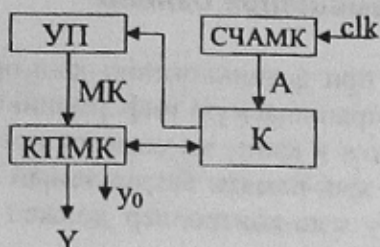


Рисунок 1 – Расположение кэш-памяти в структуре КМУУ

В отличие от кэш-памяти с прямым отображением, в которой каждый блок микрокоманд может быть размещен лишь в строго определенной строке кэш-памяти, в полностью ассоциативном кэше любой блок микрокоманд может быть размещен в любой строке кэша. Такая организация позволяет увеличить эффективность использования кэш-памяти, однако связана со значительными затратами аппаратуры, поскольку в каждом такте работы устройства поиск запрашиваемой микрокоманды должен производиться во всех строках кэша одновременно [3].

При использовании полностью ассоциативной кэш-памяти адрес МК представляется состоящим из двух полей: поля тэга и поля слова. Поле тэга Tag разрядности  $t_t$  определяет блок микрокоманд в управляющей памяти (УП) КМУУ, а поле слова Word разрядности  $t_w$  определяет порядковый номер слова в блоке МК.

Общий алгоритм одного такта работы полностью ассоциативной кэш-памяти проиллюстрирован рисунком 2.

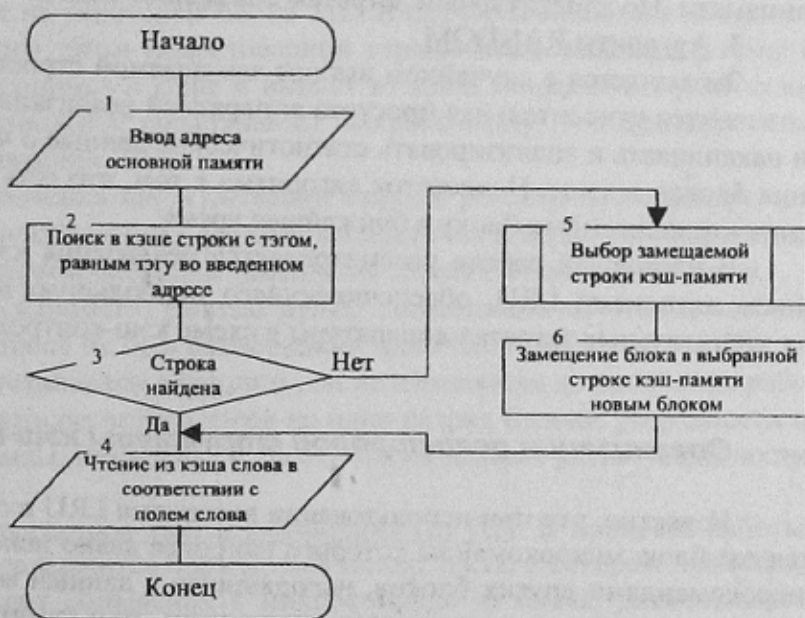


Рисунок 2 – Общий алгоритм одного такта работы кэш-памяти полностью ассоциативного типа

### **Алгоритм замещения данных**

Как известно, при возникновении кэш-промаха происходит запись блока данных, содержащего запрашиваемую информацию, из УП в кэш-память. При этом один из блоков, находящихся в кэше, замещается новым блоком. При этом в случае полностью ассоциативной кэш-памяти загружаемый блок может быть размещен в любой строке кэша, поэтому кэш-контроллер должен выбрать строку, подлежащую замене (шаг 5 на рис. 2).

Алгоритм, по которому выбирается замещаемая строка, называется алгоритмом или стратегией замещения данных. Алгоритм замещения обычно реализуется аппаратно и является важнейшей характеристикой кэш-памяти полностью ассоциативного типа.

Рассмотрим несколько наиболее известных алгоритмов замещения данных [3, 5]:

#### **1. Алгоритм LRU (Least Recently Used – наиболее давно использовавшийся).**

Данный алгоритм является приближением оптимального алгоритма. Его идея заключается в том, что в случае кэш-промаха замещается тот блок, обращение к которому происходило раньше, чем к другим блокам, находящимся в данный момент в кэше. Таким образом, замещается *наиболее давно использовавшийся* блок, то есть данные, которые давно не использовались (и поэтому предположительно не будут затребованы в ближайшее время) замещаются данными, затребованными в данный момент и, возможно, в ближайшем будущем.

Недостаток алгоритма – значительная сложность аппаратной реализации, резко возрастающая с увеличением строк кэш-памяти.

#### **2. Алгоритм FIFO.**

В соответствии с этим алгоритмом в случае кэш-промаха замещается тот блок, который был загружен в кэш раньше других блоков, находящихся в данный момент в кэш-памяти. По аппаратным затратам соизмерим с LRU.

#### **3. Алгоритм RANDOM.**

Заключается в случайном выборе замещаемой строки. К достоинствам алгоритма относится относительная простота аппаратной реализации, так как нет необходимости накапливать и анализировать статистические данные о частоте и порядке использования блоков в кэше. Недостаток алгоритма в том, что есть высокая вероятность обращения в замещаемому блоку в ближайшее время.

В настоящей работе рассматривается реализация кэш-контроллера с использованием алгоритма LRU, обеспечивающего наибольшую вероятность кэш-попаданий при значительных затратах аппаратуры в схеме кэш-контроллера.

### **Организация регистровой структуры кэш-памяти**

Известно, что при использовании алгоритма LRU в случае кэш-промаха замещается тот блок, микрокоманды которого наиболее давно использовались по сравнению с микрокомандами других блоков, находящихся в данный момент в кэше. Для реализации этого алгоритма предлагается представить кэш-память состоящей из следующих регистровых блоков (рис. 3):

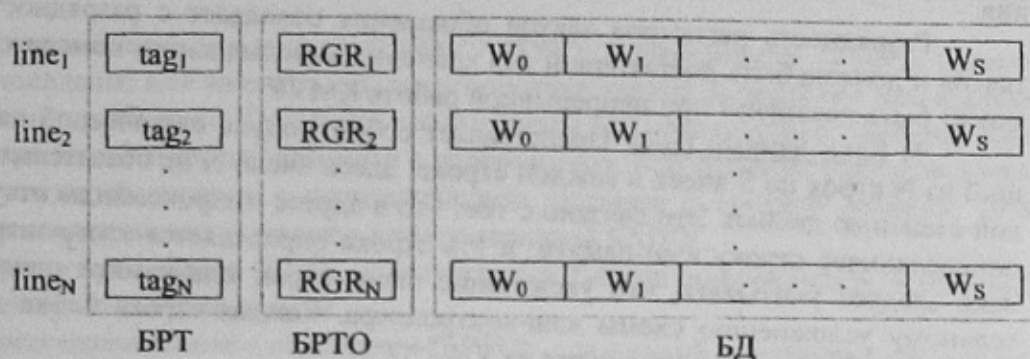


Рисунок 3 – Регистровая структура кэш-памяти полностью ассоциативного типа, реализующей алгоритм LRU

1) **Блок регистров тэгов (БРТ)**. Содержит тэги блоков микрокоманд, находящихся в данный момент в кэш-памяти. В процессе работы устройства КМУУ значения каждого из этих регистров равно значению поля Tag адреса первой ячейки блока микрокоманд, находящегося в данной строке. Обновление регистра tag<sub>i</sub> происходит при замещении блока, находящегося в i-й строке.

Регистры тэгов анализируются в каждом такте на предмет совпадения с тэгом запрашиваемой микрокоманды. Особой ситуацией в работе кэш-контроллера являются первые такты работы КМУУ, когда в строках кэш-памяти еще не содержатся достоверные данные (в строку пока не был загружен ни один блок микрокоманд), а регистры тэгов указывают на некоторые блоки МК. В случае выполнения микрокоманды из блока, определяемого тэгом недостоверной строки кэша, кэш-контроллер будет считать этот блок находящимся в кэше и выдаст из кэша запрашиваемую (но недостоверную, имеющую произвольное содержимое) микрокоманду, что приведет к сбою в работе устройства управления.

Для исключения такой ситуации каждый регистр тэга снабжен дополнительным битом достоверности (v-бит), устанавливаемым в ноль при инициализации КМУУ и в единицу при первой операции замещения данных в этой строке кэша.

Бит достоверности, равный нулю, свидетельствует о том, что с начала работы КМУУ в этой строке не был размещен ни один блок микрокоманд. После того, как бит достоверности установлен в единицу, он не изменяется до окончания работы КМУУ.

Разрядность регистров тэгов на один разряд больше разрядности поля тэга в адресе микрокоманды. Разрядность ячейки блока данных равна разрядности микрокоманды.

2) **Блок регистров тактов обращения (БРТО)**. В процессе работы устройства в этих регистрах содержится порядковый номер такта, в котором происходило последнее на данный момент обращение к микрокоманде из блока, размещенного в соответствующей строке кэша. Порядковый номер такта работы берется из счетчика тактов, находящегося в кэш-контроллере и инкрементирующегося в каждом такте работы КМУУ. Эта информация необходима для реализации алгоритма LRU: при очередном кэш-

промахе замещается строка, имеющая наименьшее значение в регистре такта обращения.

Разрядность регистров тактов обращения совпадает с разрядностью счетчика тактов и должна быть достаточной для хранения максимального номера такта, который может быть достигнут при непрерывной работе КМУУ.

3) Блок данных (БД). Представляет собой модуль статической памяти, состоящий из N строк по S ячеек в каждой строке. Здесь число N не обязательно является целой степенью двойки. Это связано с тем, что в адресе микрокоманды отсутствует поле, определяющее строку кэш-памяти, и эта строка определяется кэш-контроллером. Однако следует учитывать, что увеличение числа строк кэш-памяти приводит к значительному усложнению схемы кэш-контроллера. Каждая строка блока данных может хранить любой блок микрокоманд из УП.

### Организация структурной схемы кэш-контроллера

Структурная схема подсистемы кэш-памяти КМУУ представлена на рис. 4 и включает следующие блоки:

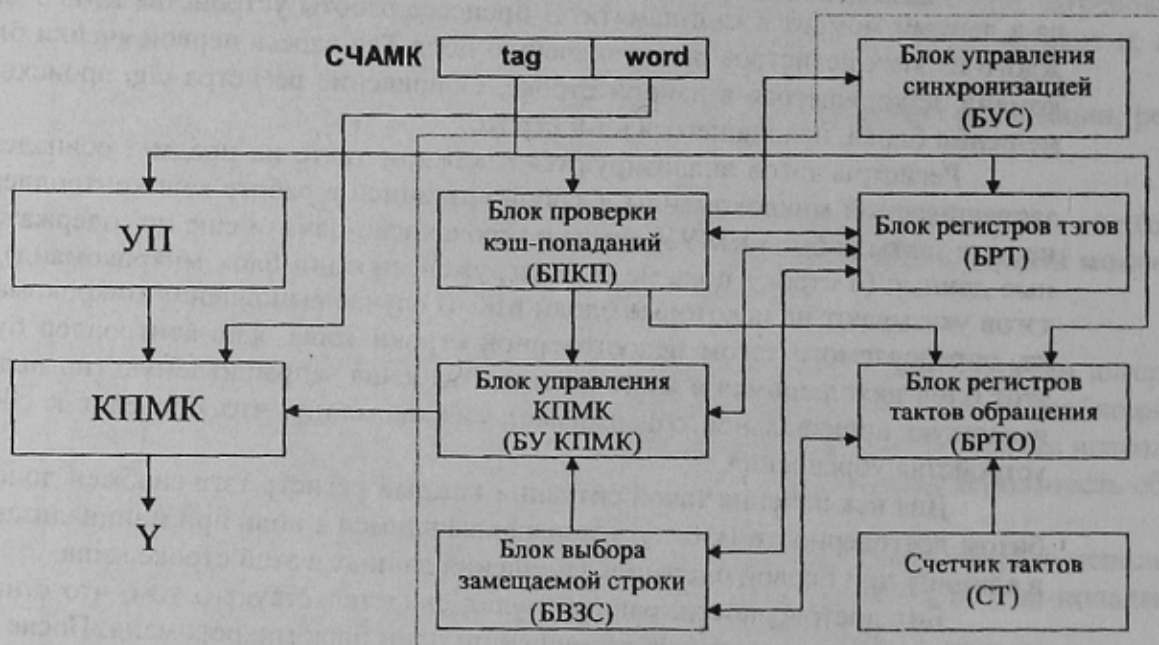


Рисунок 4 – Структурная схема кэш-контроллера в КМУУ

- *Блок проверки кэш-попаданий (БПКП)*: на основе тэга текущей микрокоманды, регистров тэгов и битов достоверности проверяет, что именно произошло: кэш-попадание или кэш-промах.
- *Блок управления КПМК (БУ КПМК)*: управляет чтением и записью данных в кэш-память микрокоманд КПМК.

- *Блок выбора замещаемой строки (БВЗС):* выбирает строку, замещаемую в случае кэш-промаха новым блоком данных из УП в соответствии с алгоритмом замещения LRU.
- *Блок управления синхронизацией КМУУ (БУС):* в зависимости от того, произошло кэш-попадание или кэш-промах, задает соответственно «короткий» или «длинный» такты синхронизации КМУУ. На работу схемы БУС также влияет значение сигнала  $u_0$ , разрешающее работу автомата с программируемой логикой: так, если выполняется условный переход с использованием автомата с жесткой логикой, то кэш-контроллер не может управлять синхронизацией КМУУ.
- *Блок регистров тэгов (БРТ):* содержит тэги блоков микрокоманд, хранящихся в данный момент в кэш-памяти.
- *Блок регистров тактов обращения (БРТО):* для каждой строки кэш-памяти содержит порядковый номер такта, в котором производилось последнее обращение к микрокоманде из этой строки.
- *Счетчик тактов (СТ):* инкрементируется в каждом такте работы КМУУ и таким образом хранит порядковый номер такта работы устройства.

### **Заклучение**

Эффективность работы кэш-памяти в значительной степени зависит от ее архитектурной организации. Рассмотренная в настоящей работе кэш-память полностью ассоциативного типа позволяет достичь максимальной частоты кэш-попаданий при реализации произвольного алгоритма. Важной характеристикой кэш-памяти ассоциативного типа является алгоритм замещения данных, реализуемый аппаратно посредством кэш-контроллера. При этом значительный выигрыш в быстродействии в большинстве случаев оправдывает увеличение аппаратурных затрат к схеме КМУУ.

### **Литература**

1. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. – Киев: ИК НАН Украины, 1997. – 135 с.
2. Ковалев С.А., Бабаков Р.М., Баркалов А.А. (ВТ-95д). Применение методов кэширования в композиционных микропрограммных устройствах управления. – В кн.: Наукові праці Донецького державного технічного університету. Серія "Інформатика, кібернетика та обчислювальна техніка". Випуск 15. – Донецьк, ДонДТУ, 2000. – с. 118-123.
3. Hill Mark Donald. Aspects of Cache Memory and Instruction Buffer Performance. Ph. D. Dissertation, Computer Science Division (EECS), University of California, Berkeley, November 1987.
4. Przybylski S.A. Cache and Memory Hierarchy Design: A Performance-Directed Approach. San Mateo, California: Morgan Kaufmann, 1990.
5. Belady L.A. A Study of Replacement Algorithms for Virtual Storage Computers. IBM Systems Journal, 5(2):78-101, 1966.

Поступила в редакційну колегію 1. 11.2001 р.