

Модель стратегии случайного замещения данных в КМУУ с кэш-памятью ассоциативного типа

Ковалев С.А., Бабаков Р.М., Шишко С.Н.
Кафедра ЭВМ
shishko@mail.ru

Abstract

Kovalov S.A., Babakov R.M., Shishko S.N. Strategy model of accidental data replacement in combinational microprogram control unit with associative cache. Development of an imitation analytical software model, which realizes analytical method of determining probability of cache-hits for CMCU fully associative cache according to flow chart, determining efficient cache memory application areas in CMCU.

Введение

В основе современных цифровых систем находится принцип микропрограммного управления, предполагающий наличие в системе управляющего автомата, координирующего работу всех ее блоков.

Управляющий автомат может быть реализован как композиция автоматов с «жесткой» и «программируемой» логикой. Разработкой таких автоматов занимался проф. А.А. Баркалов и они были названы композиционными микропрограммными устройствами управления (в дальнейшем КМУУ). В КМУУ достигается минимально возможная ёмкость управляющей памяти и максимальное быстродействие [1].

Постановка задачи

Одним из уровней иерархии памяти, используемым в вычислительных устройствах, может выступать *кэш-память*, структурно расположенная между вычислителем и основной памятью [2,3,4,5,6]. Выполненная обычно на микросхемах статической памяти SRAM, кэш-память используется для хранения команд и данных, которые требуются ВУ на протяжении текущего такта. Статистический анализ работы современных вычислительных устройств показывает, что около 90% всех запрашиваемых данных обычно находится в кэш-памяти; эта величина называется коэффициентом кэш-попадания. Если ВУ находит нужную ему информацию в кэш-памяти, это называется *кэш-попаданием*, иначе ему приходится искать нужную информацию в основной памяти; такая ситуация называется *кэш-промахом*. Использование кэш-памяти также предполагает наличие *кэш-контроллера* – специального блока,

управляющего работой такой памяти [5,6].



Рисунок 1 - Структурное расположение кэш-памяти в вычислительном устройстве

Одним из способов повышения быстродействия КМУУ является использование в структуре КМУУ кэш-памяти, предназначенной для хранения наиболее часто используемых микрокоманд и выполненной на быстродействующих микросхемах типа SRAM. При этом возможным является встраивать кэш-память (КП) так, как показано на рис. 2.

Здесь К – кэш-контроллер, предназначенный для управления процессом выбора микрокоманды из памяти. Кэш-контроллер, помимо управления УП и КП, также управляет подачей синхронизации на входы счётчика адреса микрокоманд (СЧАМК) и регистра памяти (РП).

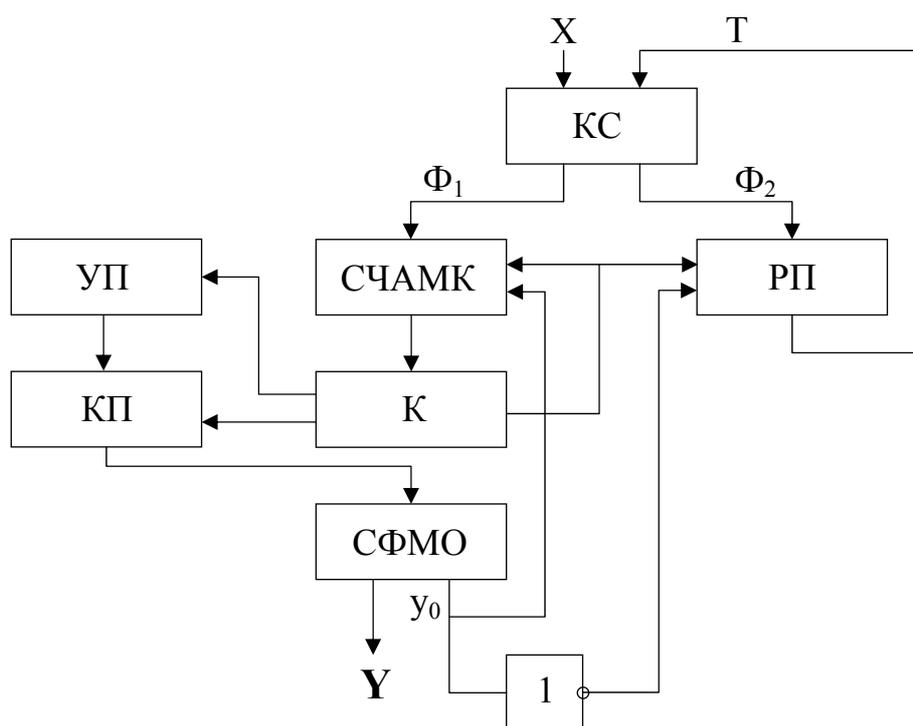


Рисунок 2 - Структурная схема КМУУ с кэш-памятью

Одним из основных способов архитектурной организации кэш-памяти является полностью ассоциативная кэш-память [7]. В полностью ассоциативном кэше любой блок 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 запрашиваемых данных должен производиться во всех строках кэша одновременно.

Стратегия замещения данных Random

Алгоритм, по которому выбирается замещаемая строка, называется *алгоритмом* или *стратегией замещения данных* [8]. Алгоритм замещения обычно реализуется аппаратно и является важнейшей характеристикой кэш-памяти полностью ассоциативного типа [9]

В качестве алгоритма замещения данных предлагается проанализировать алгоритм замещения типа Random, который заключается в случайном выборе замещаемой строки. [9,10].

Применительно к КМУУ алгоритм замещения Random может быть представлен следующим образом (рис. 3):

На первом шаге из адреса запрашиваемой микрокоманды выделяется поле тэга.

На втором шаге проверяется, произошло кэш-попадание или кэш-промах. Последовательно для каждой из N строк кэш-памяти проверяется следующее:

- Если содержимое i-й строки достоверно и тэг блока, находящегося в этой строке, совпадает с тэгом в адресе МК, то имеет место кэш-попадание и выполняется «короткий» цикл чтения из i-й строки микрокоманды, соответствующей полю Word в адресе МК.
- В противном случае проверяется следующая строка. Если проверены все строки и совпадение тэгов не обнаружено, то имеет место кэш-промах.

На третьем шаге выполняется последовательная проверка всех строк кэша на достоверность их содержимого. Для замещения выбирается первая обнаруженная строка с недостоверным содержимым. Этот шаг эффективен лишь до тех пор, пока не будут выполнены первые N кэш-промахов, после чего проверка строк на достоверность будет излишней – все строки будут содержать некоторые блоки микрокоманд из УП. Однако в дальнейшем выполнение этого шага не замедляет работу кэш-контроллера, поскольку фактически шаг 3 выполняется аппаратно одновременно с шагом 4, и имеет лишь логический приоритет в выборе замещаемой строки.

На четвертом шаге реализуется непосредственно алгоритм Random. Он заключается в случайном выборе замещаемой строки. Выбранная таким образом строка подвергается замещению.

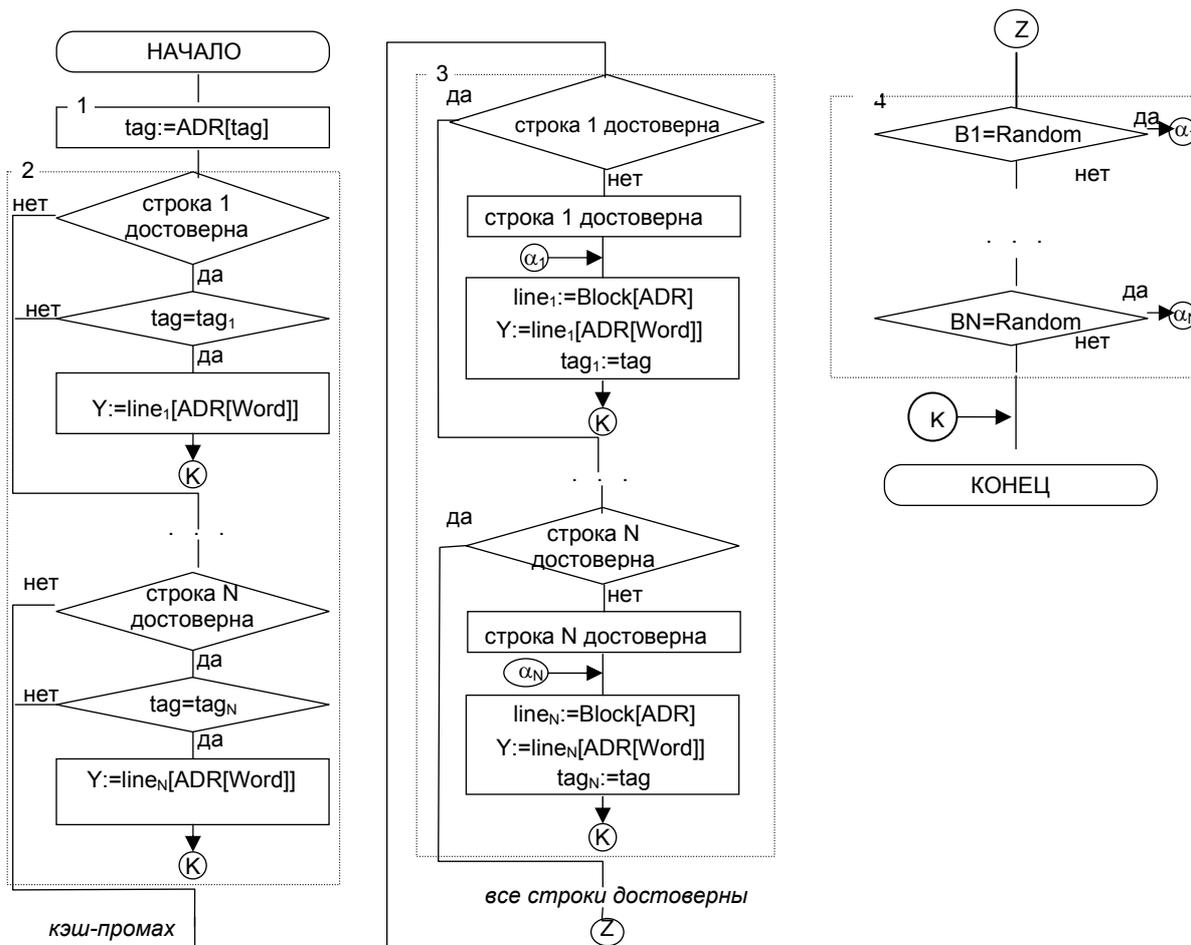


Рисунок 3 - Блок-схема алгоритма замещения данных Random

В настоящее время не существует точных аналитических методов, определяющих оптимальные значения таких характеристик кэш-памяти, как ее размер, длина строки, архитектурная организация. Их значения во многом определяются областью использования цифровых устройств, характером реализуемых алгоритмов, элементной базой, и обычно подбираются опытным путем. Поскольку КМУУ содержит в своей структуре автомат с жесткой логикой, синтезируемый для конкретного алгоритма, то возникает необходимость в разработке методики оценки эффективности применения кэш-памяти при определенных ее параметрах для заданного алгоритма.

Предлагается метод аналитического определения вероятности кэш-попаданий для заданного алгоритма, который основывается на выявлении ситуаций, приводящих к возникновению кэш-попаданий или кэш-промахов, и на подсчете частоты возникновения таких ситуаций в процессе выполнения алгоритма.

Построение программной имитационно-аналитической модели

Практической реализацией разработанного метода является программная имитационно-аналитическая модель КМУУ с кэш-памятью микрокоманд. Структурно программа состоит из 2 модулей, изображенных на рис. 4.

Модуль 1 реализует экспериментальное определение вероятности кэш-попаданий для КМУУ с кэш-памятью полностью ассоциативного типа при использовании алгоритма замещения Random. Время, затрачиваемое на эксперимент, зависит как от среднего количества микрокоманд, выполняющихся за один проход алгоритма, так и от количества строк в кэш-памяти, и возрастает с их увеличением.

Модуль 2 реализует метод аналитического определения точного значения вероятности кэш-попаданий по граф-схеме алгоритма для КМУУ с кэш-памятью полностью ассоциативного типа при использовании алгоритма замещения Random.

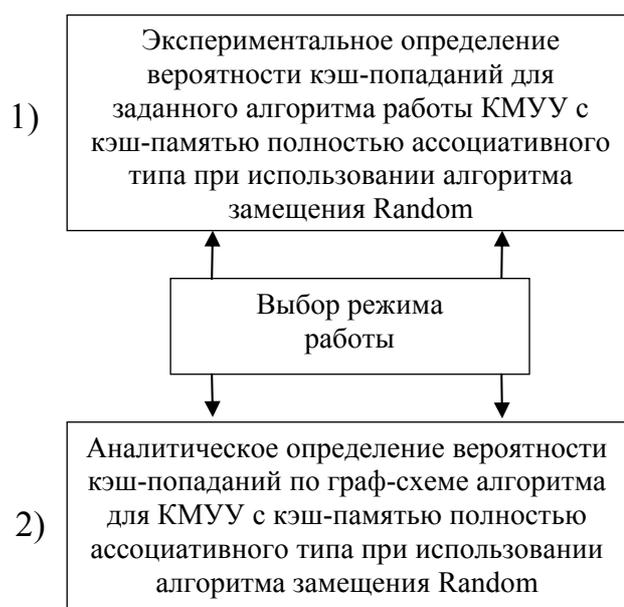


Рисунок 4 - Структурная организация программной имитационно-аналитической модели КМУУ с кэш-памятью

Данная имитационно-аналитическая модель показывает насколько эффективнее с точки зрения быстродействия оказывается разработанный метод по сравнению с экспериментальным определением вероятности кэш-попаданий.

Для более точного определения количества кэш-попаданий и кэш-промахов алгоритм должен быть выполнен большое количество раз. Это связано с тем, что переходы по ветвям логических условий выбираются

каждый раз случайным образом, и при каждом проходе алгоритма может выполняться разное количество микрокоманд. Опытным путем установлено, что для того, чтобы экспериментальные и аналитические значения вероятности кэш-попаданий были равны с точностью до третьего значащего разряда, необходимо выполнить не менее 10000 проходов алгоритма.

Для оценки эффективности разработанных методов аналитического определения вероятности кэш-попаданий с точки зрения быстродействия сравнивается время, затрачиваемое на моделирование, со временем, затрачиваемым на аналитический расчет. Подобные сравнения проведены для различных размеров кэш-памяти при неизменных характеристиках алгоритма. Все временные значения получены на компьютере с процессором iP-III 733 МГц под управлением операционной системы Windows XP в оконном режиме при отсутствии других выполняющихся задач.

Таблица 1 - Временные затраты на экспериментальное и аналитическое определение вероятности кэш-попаданий для кэш-памяти полностью ассоциативного типа

N/S	2	4	8	16	32	64	128	256
2	318 0,3	316 0,3	302 0,2	315 0,2	342 0,2	320 0,2	326 0,2	359 0,2
3	329 0,4	321 0,3	309 0,2	339 0,2	315 0,2	335 0,2	324 0,2	
4	329 0,4	339 0,3	317 0,3	326 0,3	319 0,3	345 0,3	300 0,3	
5	360 0,5	350 0,3	369 0,3	324 0,3	326 0,3	335 0,3		
8	377 0,8	361 0,6	362 0,5	353 0,4	358 0,5	353 0,7		
10	394 1,0	352 0,7	368 0,6	378 0,6	344 0,8			
14	454 1,5	385 1,1	400 1,0	363 1,2	371 1,8			
16	476 1,8	413 1,3	407 1,2	421 1,6	357 2,5			

В таблице показаны временные затраты на экспериментальное и аналитическое определение вероятности кэш-попаданий. При этом в верхней части ячейки указано количество секунд, затраченное на экспериментальное определение вероятности, а в нижней части – на аналитическое.

Исходя из содержимого таблицы можно сделать следующие выводы:

- Увеличение числа строк приводит к увеличению длительности эксперимента.
- Увеличение числа слов в строке не оказывает существенного влияния на длительность эксперимента. Колебания значений вызваны случайной генерацией переходов по ветвям условных вершин.
- Увеличение числа строк приводит к увеличению длительности анализа.
- Увеличение слов в строке кэш-памяти сначала приводит к уменьшению времени анализа, а затем – к его увеличению (соответствующий график зависимости для 10, 14 и 16 строк кэш-памяти изображен на рис.5).

Таким образом, аналитическое определение вероятности кэш-попаданий по граф-схеме алгоритма с использованием разработанного метода, оказывается в большинстве случаев в несколько раз эффективнее по сравнению с экспериментальным методом.

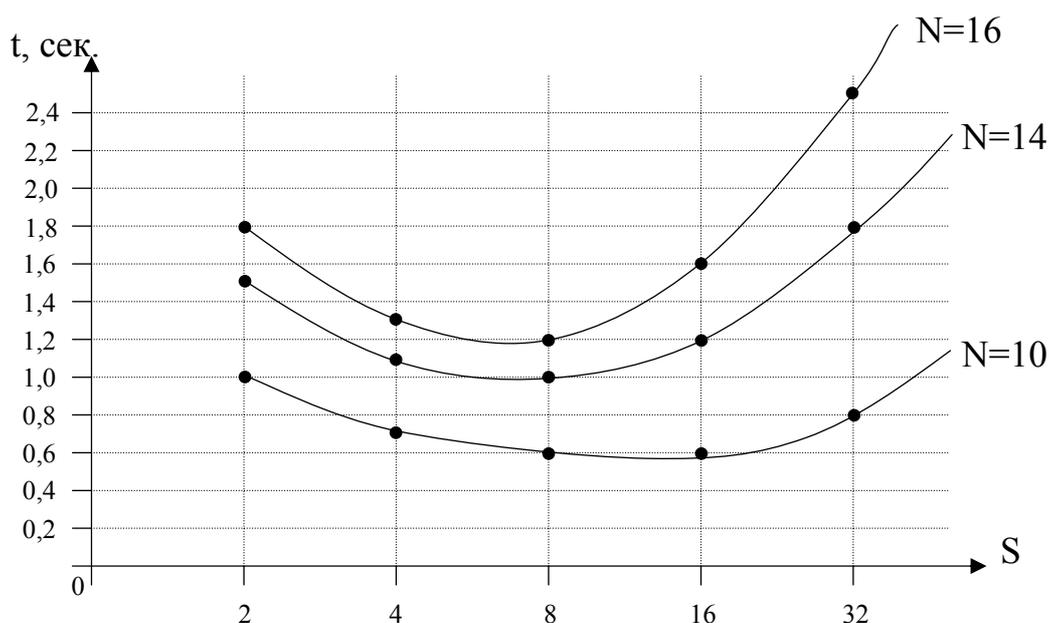


Рисунок 5 - Зависимость времени анализа от числа слов в строке кэш-памяти полностью ассоциативного типа

Заключение

Разработана имитационно-аналитическая модель определения вероятности кэш-попаданий для заданной граф-схемы алгоритма управления и характеристик кэш-памяти. Модель разработана для кэш-памяти полностью ассоциативного типа при использовании алгоритма замещения Random. Получены аналитические оценки увеличения

быстродействия при использовании кэш-памяти в композиционных микропрограммных устройствах управления. Предложенный метод аналитического определения значения вероятности кэш-попаданий по граф-схеме алгоритма позволяет получить результат в общем случае в несколько раз быстрее, чем при использовании экспериментальных методов.

Литература

1. Баркалов А.А., Матвиенко А.В. Реализация микропрограммного устройства управления композицией автоматов с жесткой и программируемой логикой // Микропроцессорные средства, разработка и применение. – К.: ИК АН УССР, 1985. – С. 38-42.
2. Гук М. Процессоры Intel от 8086 до Pentium II. – СПб.: Питер, 1997. – 224с.
3. Шагурин И.И., Бердышев Е.М. Процессоры семейства P6 – Pentium II, Pentium III, Celeron и другие. Архитектура, программирование, интерфейс. – М.: Горячая линия – Телеком. – 2000. – 248 с.
4. Faust B. Designing Alpha-based systems. – Byte Magazine. – June 1995. – P. 239-240.
5. Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach. – Morgan Kaufmann Publishers, San Mateo, CA, second ed., 1996.
6. Hill Mark Donald. Aspects of Cache Memory and Instruction Buffer Performance. – Ph. D. Dissertation, Computer Science Division (EECS), University of California, Berkeley, November 1987.
7. Cheremisinova L. Synthesis of single-level circuits in programmable array logic basis. / In: Computer Aided. – Design of Discrete Devices. VI. – Minsk, 1997. – P. 81-85.
8. Hosseini-Khayat S. On Optimal Replacement of Nonuniform Cache Objects // IEEE Transactions on Computers, Vol. 49, № 8, August 2000. – P. 769-778.
9. Johnson T., Shasha D. A Low Overhead High Performance Buffer Management Replacement Algorithm // VLDB 1994: P. 439-450.
10. Kessler R., Hill M. Page Placement Algorithm for Large Real-Indexed Caches // ACM Trans. Computer Systems. – Nov. 1992, vol. 10, no. 4. – P. 338-359.

Дата надходження до редакції 09.10.2007 р.