

КЭШ-ПАМЯТЬ МИКРОКОМАНД В КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВАХ УПРАВЛЕНИЯ

Баркалов А.А., Ковалев С.А., Бабаков Р.М.

University Zelenogursky Institution of Informatics and Electronics, Poland

Кафедра ЭВМ, ДонНТУ

A.Barkalov@iie.uz.zgora.pl

Abstract

Barkalov A.A., Kovalyov S.A., Babakov R.M. Cache-memory of microinstructions in compositional microprogram control units. The method of increase of speed of circuits of control units is offered due to use of cache-memory of microinstructions. Analytical methods of an estimation of efficiency of use of cache-memory in compositional microprogram control units are offered. The comparative analysis of experimental and analytical methods of an estimation of efficiency of cache-memory is carried out.

Одной из проблем, возникающих при синтезе микропрограммных устройств управления (МУУ), является проблема увеличения быстродействия [1, 2]. В настоящей работе представляется метод решения этой проблемы, основанный на использовании в структуре МУУ кэш-памяти микрокоманд.

Микропрограммное устройство управления может быть реализовано в виде композиции автоматов с «жесткой» и «программируемой» логикой. Такие устройства получили название композиционных микропрограммных устройств управления (КМУУ) [2, 3]. Базовая структура КМУУ изображена на рис. 1.

Здесь автомат с «жесткой» логикой S_1 включает комбинационную схему КС, формирующую адреса переходов при нарушении естественного порядка выполнения микрокоманд, и регистр памяти РП, предназначенный для хранения состояний автомата S_1 . Автомат с «программируемой» логикой включает счетчик адреса микрокоманд СЧАМК, формирующий адреса переходов в линейных последовательностях операторных вершин, и управляющую память УП, предназначенную для хранения микропрограммы реализуемого алгоритма управления. Функционирование КМУУ, а также этапы синтеза базовой структуры подробно рассмотрены в [2].

В настоящее время существует множество методов оптимизации характеристик схемы КМУУ при ее реализации в базисе программируемых БИС (ПЗУ, ПЛМ, ПЛИС) [4]. Эти методы преимущественно направлены

на уменьшение аппаратных затрат, и обычно требуют введения в структуру КМУУ дополнительных преобразующих схем, включенных последовательно и в общем случае увеличивающих длительность такта работы КМУУ.

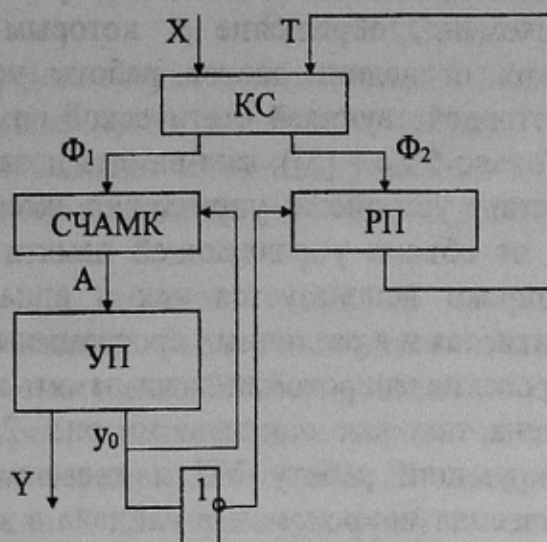


Рис. 1. Структура композиционного микропрограммного устройства управления

С другой стороны, существует множество задач, в которых быстродействие схемы устройства управления является важнейшим (а иногда решающим) фактором возможности и целесообразности применения КМУУ для решения задачи (например, управление процессом, требующим большого количества вычислений за относительно малое время). По этой причине важной задачей при синтезе схем КМУУ является увеличение быстродействия, даже если это приводит к некоторому увеличению аппаратных затрат в схеме. Это возможно как за счет увеличения быстродействия элементного базиса, так и за счет различных архитектурных решений.

При синтезе схем КМУУ управляющая память обычно реализуется на ПЗУ или ОЗУ [2, 3], что приводит к относительно большому времени доступа к УП при выборе микрокоманды. Если считать, что управляющая память реализована на ПЗУ с временем срабатывания $t_{УП}=200$ нс [5], а время срабатывания СЧАМК $t_{СЧАМК}=20$ нс, то становится очевидным, что общее время срабатывания автомата S_2 составляет 220 нс, причем время срабатывания УП здесь составляет 90%. В свою очередь, время срабатывания автомата S_1 определяется временем срабатывания КС $t_{КС}$ (будем считать, что при реализации на ПЛИМ или ПЛИС $t_{КС}=20$ нс) и временем срабатывания РП $t_{РП}$ (как и для СЧАМК, $t_{РП}=20$ нс) и составляет таким образом 40 нс, что значительно меньше времени срабатывания автомата S_2 . Следовательно, важной задачей, возникающей

при синтезе КМУУ, является увеличение быстродействия КМУУ путем снижения времени доступа к управляющей памяти.

С целью увеличения быстродействия схемы устройства авторами предлагается использовать в структуре КМУУ модуль кэш-памяти. Кэш-память предназначена для временного хранения подмножества микрокоманд, обращение к которым происходило на протяжении нескольких последних тактов работы устройства. Будучи реализованной на быстродействующей статической памяти и имея малое время срабатывания (около 5 нс – [5]), кэш-память позволяет значительно увеличить быстродействие устройства управления, имея при это размер в несколько процентов от объема управляющей памяти [6]. В настоящее время кэш-память широко используется как в аппаратных средствах вычислительной техники, так и в различных программных продуктах.

Для кэширования микрокоманд кэш-память в структуре КМУУ должна быть размещена так, как показано на рис. 2. Здесь К – кэш-контроллер, координирующий работу УП и кэш-памяти микрокоманд КПМК. Если запрашиваемая микрокоманда найдена в кэш-памяти (имело место кэш-попадание), то кэш-контроллер разрешает прохождение импульса синхронизации на вход СЧАМК (при этом длительность синхроимпульса достаточна для чтения данных из быстрой кэш-памяти). Если же искомая МК не найдена в кэш-памяти (ситуация кэш-промаха), то выполняется «длинный» цикл чтения данных из УП; при этом синхронизация на входы СЧАМК и РП не подается до тех пор, пока цикл чтения не будет завершен. Это может занять несколько тактов синхронизации (в зависимости от длительности срабатывания схемы УП).

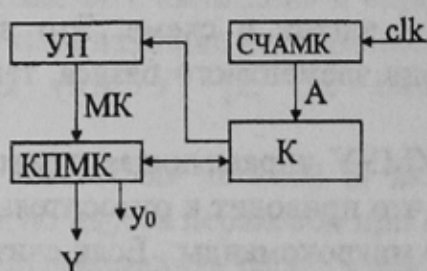


Рис. 2. Расположение кэш-памяти в структуре КМУУ

Основными видами архитектурной организации кэш-памяти являются кэш-память с прямым отображением и кэш-память полностью ассоциативного типа [6]. В первом случае замещаемые данные определяются частью адреса запрашиваемых данных, во втором случае – алгоритмом, основанным на частоте использования данных, находящихся в кэше. Использованию эффективных алгоритмов замещения данных

обычно сопутствуют высокие затраты аппаратуры в схеме кэш-контроллера.

Эффективность использования кэш-памяти в КМУУ может быть определена по формуле

$$E = \frac{t_{УП}}{t_E} = \frac{t_{УП}}{P_h \cdot t_C + (1 - P_h) \cdot t_m}, \quad (1)$$

где $t_{УП}$ – время доступа к управляющей памяти в системе без кэш-памяти; t_E – среднее время доступа в системе с кэш-памятью; P_h – вероятность кэш-попаданий алгоритма управления; t_C – время доступа к данным в случае кэш-попадания; t_m – время доступа к данным в случае кэш-промаха.

В формуле (1) величины $t_{УП}$, t_C и t_m зависят только от используемой элементной базы и являются известными. Величина P_h зависит от реализуемого алгоритма управления и от характеристик кэш-памяти (числа строк, числа слов в строке и типа архитектуры). В зависимости от этих параметров величина P_h может изменяться от 0 до 1 и не является очевидной из структуры граф-схемы (ГСА) реализуемого алгоритма управления.

Поскольку, в отличие от универсальных микропроцессоров, структура КМУУ жестко привязана к реализуемому алгоритму, имеется возможность подобрать такие характеристики кэш-памяти, которые для реализуемого алгоритма управления позволяли бы получить максимально возможную величину вероятности кэш-попаданий, а следовательно, максимальное увеличение быстродействия.

Значение P_h может быть определено экспериментально путем программного моделирования работы узлов КМУУ и сбора статистической информации о количестве возникновения ситуаций кэш-попаданий и кэш-промахов, либо аналитически. Авторами разработаны методы аналитического определения точного значения вероятности кэш-попаданий по граф-схеме алгоритма для двух основных типов кэш-памяти: с прямым отображением и полностью ассоциативного типа. При этом в качестве исходных данных выступают граф-схема алгоритма управления, вероятности переходов по логическим условиям, адреса микрокоманд, а также размеры и тип архитектуры кэш-памяти. В основе методов лежит известное вероятностное представление ГСА [7].

Основными этапами аналитических методов определения вероятности кэш-попаданий по ГСА являются:

1. Формирование и решение системы уравнений, отражающих переходы в каждую микрокоманду ГСА, с целью определения вероятностей ее выполнения.
2. Формирование множества V микрокоманд, для которых возможен кэш-промах.

3. Определение вероятности возникновения ситуации кэш-попадания для каждой микрокоманды ГСА. Возможность возникновения ситуаций кэш-попаданий или кэш-промахов определяется в соответствии с алгоритмом замещения данных.
4. Определение вероятности кэш-попаданий для рассматриваемого алгоритма как суммы вероятностей кэш-попаданий для каждой микрокоманды ГСА.

Рассмотренные методы нашли практическую реализацию в программной имитационно-аналитической модели. Данная модель позволяет определить вероятность кэш-попаданий по заданной ГСА для кэш-памяти с прямым отображением и полностью ассоциативного типа как экспериментальным способом, так и аналитическим, обеспечивая точность расчетов до третьего значащего разряда.

Для оценки эффективности использования кэш-памяти в КМУУ авторами был проведен ряд экспериментов. Эксперименты были проведены на примере абстрактного алгоритма управления, включающего 1000 микрокоманд, 33 ОЛЦ, 10 логических условий и 26 условных вершин.

Уточненная формула определения эффективности использования кэш-памяти в КМУУ имеет следующий вид:

$$E = \frac{t_{y\Pi} + t_a}{t_c + t_{y\Pi} + t_a - t_{y\Pi} \cdot P_h}, \quad (2)$$

где t_a – время срабатывания схем, асинхронных относительно сигнала синхронизации КМУУ. К таким схемам относятся, например, операционный автомат, схема КС, а также другие схемы, применяемые при использовании различных методов минимизации аппаратурных затрат.

По формуле (2) были получены зависимости эффективности от каждого из параметров формулы (2) при фиксированных значениях остальных параметров ($t_{y\Pi}=200$ нс, $t_c=10$ нс, $t_a=40$ нс, $P_h=0.9$). При этом были сделаны следующие выводы:

1. Зависимость $E(t_{y\Pi})$ имеет нелинейно возрастающий характер, причем при $t_{y\Pi} \rightarrow \infty$ значение E стремится к величине $t_{y\Pi}/(t_{y\Pi} - 0.9t_{y\Pi}) = 10$.
2. Зависимость $E(t_c)$ практически линейна и убывает с увеличением t_c . При $t_c \rightarrow \infty$ значение E стремится к нулю.
3. Зависимость $E(t_a)$ нелинейно убывает с увеличением t_a . При $t_a \rightarrow \infty$ значение E стремится к единице.
4. Зависимость $E(P_h)$ нелинейно возрастает с увеличением P_h . При $P_h \rightarrow 100\%$ величина $E \rightarrow (t_{y\Pi} + t_a)/(t_c + t_a) = 4.8$.

Анализ полученных зависимостей позволяет определить область эффективного использования кэш-памяти в композиционных микропрограммных устройствах управления.

Также авторами получены зависимости вероятности кэш-попаданий от размеров кэш-памяти. В таблице 1 приведены значения P_h для кэш-

памяти с прямым отображением в зависимости от количества строк (N) и слов в строке (S) кэш-памяти. Данные значения были получены с помощью разработанных методов аналитического определения вероятности кэш-попаданий по граф-схеме алгоритма. Следует отметить, что в тех случаях, когда размер кэш-памяти соизмерим с размерами основной памяти (ячейки таблицы, расположенные под побочной диагональю), вероятность кэш-попаданий стремится к 100%. Однако в этом случае также имеет место значительное удорожание схемы, что в большинстве случаев является неприемлемым [6].

Таблица 1

Зависимость вероятности кэш-попаданий от размеров кэш-памяти для кэш-памяти с прямым отображением

N/S	2	4	8	16	32	64	128	256
2	48,7	73,1	86,0	92,7	95,5	97,3	98,7	99,2
4	49,3	76,2	88,7	94,0	96,3	98,0	99,1	≈1
8	57,6	80,2	90,0	94,3	97,4	98,8	≈100	≈100
16	61,7	81,6	90,8	96,3	98,5	≈100	≈100	≈100
32	64,0	83,3	94,2	97,7	≈100	≈100	≈100	≈100
64	67,3	89,6	96,5	≈100	≈100	≈100	≈100	≈100
128	79,9	93,6	≈100	≈100	≈100	≈100	≈100	≈100
256	87,5	≈100	≈100	≈100	≈100	≈100	≈100	≈100

Кривые зависимостей $E(S)$ при заданном количестве строк имеют нелинейно возрастающий характер. На рис. 3 показаны зависимости вероятности кэш-попаданий P_h (1) и эффективности E (2) от числа слов в строке кэш-памяти с прямым отображением при числе строк, равном двум. Очевидно, что при значениях P_h , превышающих 0,8, эффективность использования кэш-памяти в КМУУ при заданных временных параметрах модулей КМУУ достигает значения 400% и выше.

Авторами получены аналогичные экспериментальные зависимости для кэш-памяти полностью ассоциативного типа.

При определении оптимальных характеристик кэш-памяти могут быть использованы как экспериментальный, так и аналитический методы определения вероятности кэш-попаданий. Для выявления более эффективного метода авторами проведено сравнение быстродействия аналитических и экспериментальных методов для выбранного алгоритма управления. Анализ эффективности проведен с помощью разработанной программной имитационно-аналитической модели.

В таблице 2 показаны временные затраты на экспериментальное и аналитическое определение вероятности кэш-попаданий для кэш-памяти с прямым отображением. Количество времени, затраченное на эксперимент и аналитические расчеты, указано в секундах. Исследования также показали, что временные затраты для кэш-памяти полностью

ассоциативного типа имеют сходные значения. Все временные значения получены на компьютере с процессором iP-III 733 МГц под управлением операционной системы Windows 98 в оконном режиме при отсутствии других выполняющихся программ.

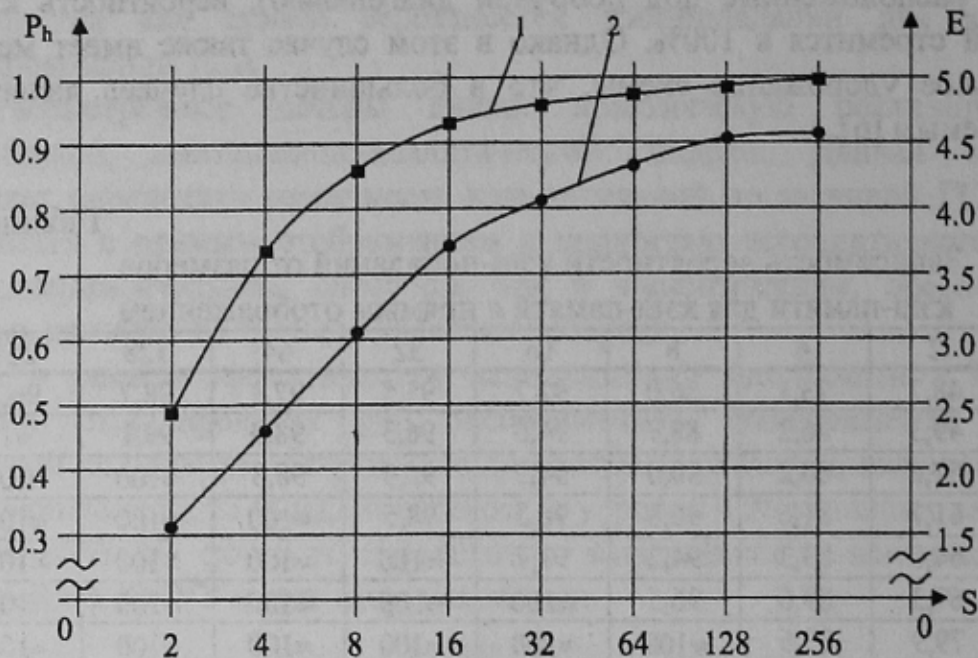


Рис. 3. Зависимость $P_h(S)$ (1) и $E(S)$ (2) для кэш-памяти с прямым отображением при $N=2$

Таблица 2

Временные затраты на экспериментальное и аналитическое определение вероятности кэш-попаданий для кэш-памяти с прямым отображением

Размер кэш-памяти (N/S)	Время на эксперимент, с	Время на аналитич. расчет, с	Размер кэш-памяти (N/S)	Время на эксперимент, с	Время на аналитич. расчет, с
2/2	248	0,5	8/16	238	1,9
2/4	246	0,4	8/64	246	1,6
2/16	248	0,3	16/2	237	9,1
2/128	262	0,3	16/8	222	6,6
2/256	263	0,3	16/32	224	4,1
4/2	251	1,0	32/2	230	46,2
4/16	246	0,7	32/16	216	13,2
4/128	254	0,7	64/2	230	223
8/2	238	2,6	64/8	213	59

Очевидно, что в большинстве случаев время, затрачиваемое на эксперимент, значительно превышает время аналитических расчетов при сохранении точности результатов, что говорит о высокой эффективности разработанных аналитических методов определения вероятности кэш-попаданий по ГСА.

Таким образом, применение кэш-памяти микрокоманд в композиционных микропрограммных устройствах управления приводит в общем случае к увеличению среднего быстродействия схемы устройства в 2-5 раз и выше в зависимости от характеристик элементного базиса и структуры реализуемого алгоритма управления. Предложенные авторами методы аналитического определения значения вероятности кэш-попаданий по граф-схеме алгоритма позволяют получить результат в общем случае в несколько раз быстрее, чем при использовании экспериментальных методов, и могут быть использованы в системах автоматического проектирования схем микропрограммных устройств управления с кэш-памятью микрокоманд.

Литература

1. Баранов С.И. Синтез микропрограммных автоматов. – Л.: Энергия, 1979. – 232 с.
2. Баркалов А.А. Палагин А.В. Синтез микропрограммных устройств управления. – К., 1997. – 136 с.
3. 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.
4. Баркалов А.А., Джалиашвили З.О., Струнилин В.Н. Оптимизация композиционного микропрограммного устройства управления // Известия вузов СССР. Приборостроение. – 1989, – №3. – С. 36-39.
5. Цифровые интегральные микросхемы: Справочник / П.П. Мальцев, И.И. Долидзе, М.И. Критенко и др. – М.: Радио и связь, 1994. – 240 с.
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. Поспелов Д.А. Вероятностные автоматы. – М.: Энергия, 1970. – 88 с., ил.

Дата надходження до редколегії: 16.10.2003 р.