ПРИМЕНЕНИЕ ПРИНЦИПОВ КЭШИРОВАНИЯ В КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВАХ УПРАВЛЕНИЯ

Ковалев С.А., Бабаков Р.М., Баркалов А.А. (ВТ-95 д)
кафедра ЭВМ, ДГТУ
kovalev@pop.dgtu.donetsk.ua

Abstract

Kovalev S.A., Babakov R.M., Barkalov A.A. (VT-95d). Application of principles of caching in compositional microprogram control units. *The article is devoted to the necessity of application of the cache memory in control units for increase of speed. The formulas for a numerical evaluation of efficiency of usage of the cache memory are given, and also a structure of compositional microprogram control unit with the cache memory is considered.*

Введение

Одним из центральных блоков цифровых устройств является устройство управления (УУ), в качестве которого может использоваться композиционное микропрограммное устройство управления (КМУУ), обладающее лучшими характеристиками по сравнению с УУ, использующими операционно-адресную структуру управляющих слов [1]. Одной из важных задач, возникающих при синтезе КМУУ, является повышение быстродействия работы схемы. В настоящей работе предлагается использование в структуре КМУУ элементов кэш-памяти, предназначенной для уменьшения среднего времени выполнения алгоритма. Дано обоснование необходимости и возможности использования кэш-памяти, рассмотрены основные принципы кэширования, получены математические оценки для определения эффективности использования кэш-памяти. Рассмотрена структура КМУУ с кэш-памятью и особенности ее организации и функционирования.

1. Принципы организации композиционных микропрограммных устройств управления

Композиционные микропрограммные устройства управления представляют собой композицию автоматов с «жесткой» и «программируемой» логикой [2]. Одной из характерных черт КМУУ является использование естественной адресации при реализации функций переходов. При этом в управляющей памяти (УП) отсутствуют управляющие микрокоманды (МК), вследствие чего содержимое УП представляет собой множество наборов микрокоманд из последовательно идущих операторных вершин, образовывающих операторные линейные цепи (ОЛЦ) (рис.1) [3].

1. Автомат с жесткой логикой S_1, формирующий адреса переходов при нарушении естественного порядка следования микрокоманд. Этот автомат включает комбинационную схему КС и регистр памяти РП.
2. Автомат с программируемой логикой S_2, формирующий адреса переходов при естественном следовании микрокоманд, он включает счетчик адреса микрокоманд СЧАМК, управляемый сигналом y_0, и управляющую память УП. Для реализации системы микроопераций используется схема формирования микроопераций СФМО.

Переход в пределах любой ОЛЦ происходит под действием управляющего сигнала y_0 # СЧАМК:=СЧАМК+1. Переход автомата S_1 возможен только при y_0=0, в этом случае в РП записывается код $Ф_2$. Иначе в СЧАМК заносится адрес следующей МК.

Регистры СЧАМК и РП синхронизируются сигналом clk, причем в каждом такте срабатывает только один из них (в зависимости от значения сигнала y_0). При этом СЧАМК формирует R-разрядный адрес управляющей памяти, по которому из УП выбирается микрокоманда разрядности K.

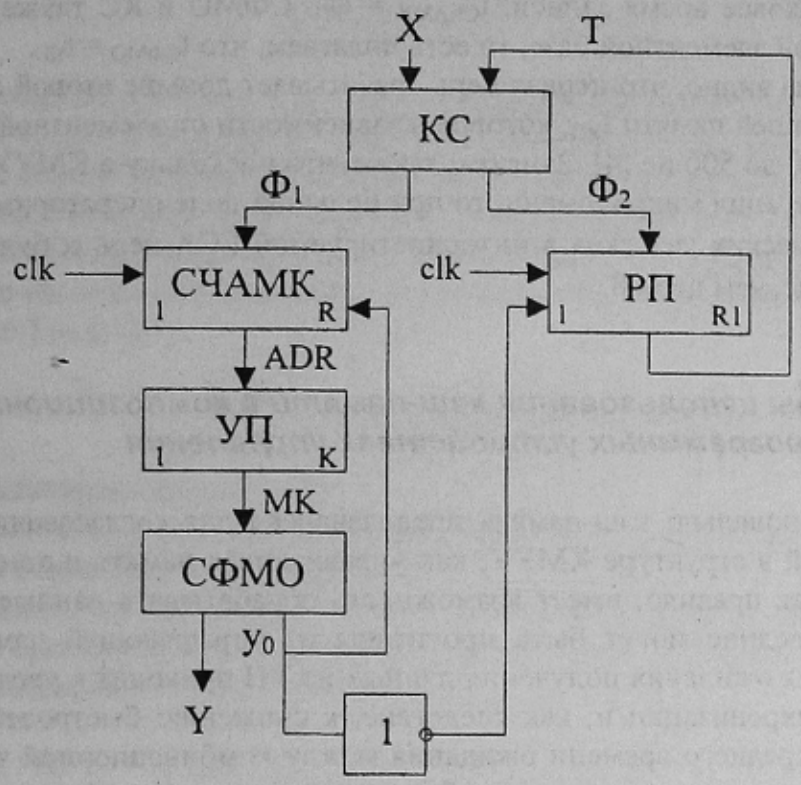

Рис.1. Функциональная схема КМУУ

Рассмотрим временные диаграммы одного такта работы КМУУ (рис.2).

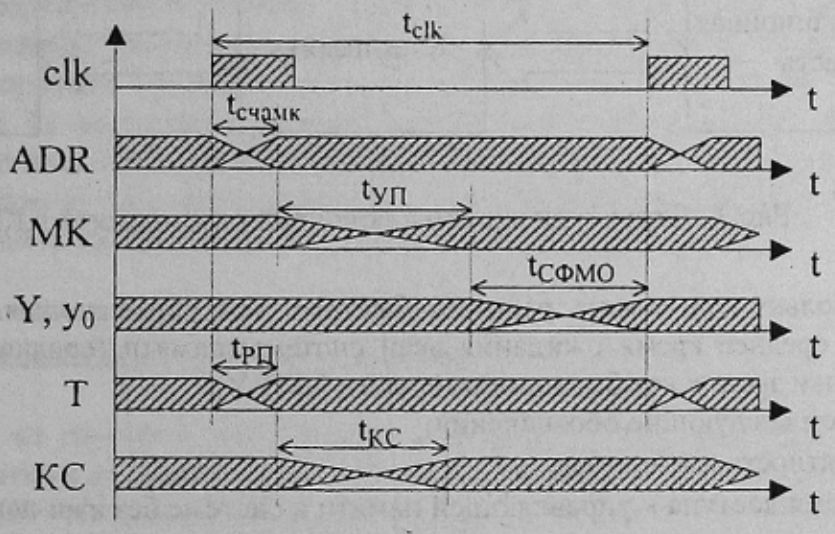

Рис.2. Временные диаграммы работы КМУУ

Так как по сигналу clk срабатывают либо СЧАМК, либо РП, то в схеме КМУУ существует две цепи прохождения сигналов: цепь α = <СЧАМК, УП, СФМО> и цепь β = <РП, КС> (временем срабатывания инвертора пренебрегаем). Поскольку эти цепи включены параллельно, то длительность синхроимпульса t_{clk} равна максимальному времени прохождения сигнала по каждой из цепей. То есть $t_\alpha = t_{СЧАМК} + t_{УП} + t_{СФМО}$; $t_\beta = t_{РП} + t_{КС}$; $t_{clk} = \max(t_\alpha, t_\beta)$.

Если принять, что СЧАМК и РП выполнены на одинаковых элементах, то они имеют одинаковое время записи: $t_{СЧАМК} = t_{РП}$. СФМО и КС также считаем выполненными на одной элементной базе, то есть полагаем, что $t_{СФМО} = t_{КС}$.

Отсюда видно, что первая цепь срабатывает дольше второй на время срабатывания управляющей памяти $t_{УП}$, которое в зависимости от элементной базы УП может составлять от 20 до 500 нс [4]. Заметим также, что поскольку в КМУУ используется естественная адресация микрокоманд, то при большой доле операторных вершин и при наличии циклических участков в интерпретируемой ГСА цепь α будет срабатывать значительно чаще, чем цепь β.

3. *Принципы использования кэш-памяти в композиционных микропрограммных устройствах управления*

Функционально кэш-память предназначена для согласования скорости работы таких модулей в структуре КМУУ, как управляющая память и комбинационная часть. Последняя, как правило, имеет возможность обрабатывать данные значительно быстрее, чем последние могут быть прочитаны из управляющей памяти. Относительно большое время ожидания получения данных из УП приводит к увеличению длительности такта синхронизации и, как следствие, к снижению быстродействия КМУУ. Для уменьшения среднего времени ожидания между комбинационной частью и управляющей памятью предлагается ввести быстродействующую память небольшого объема, называемую обычно кэш-памятью (рис.3). Как правило, кэш-память подразумевает также наличие кэш-контроллера – специального модуля, осуществляющего управление обменом данными между комбинационной частью, управляющей и кэш-памятью [5].

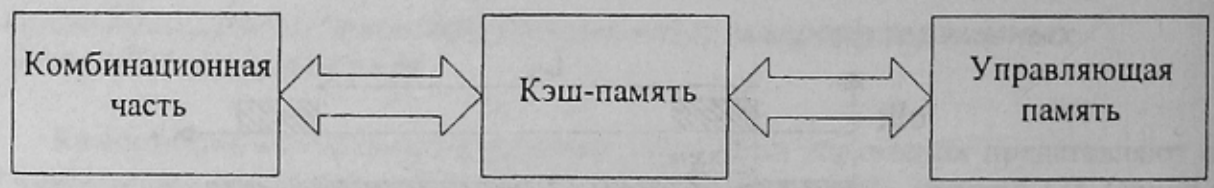

Рис.3. Структурная схема включения кэш-памяти в КМУУ

Поскольку кэш-память работает быстрее, чем управляющая память, то можно определить среднее время ожидания всей системы памяти (среднее время доступа к памяти с точки зрения комбинационной части КМУУ).

Введем следующие обозначения:

- H – вероятность кэш-попаданий;
- $T_{УП}$ – время доступа к управляющей памяти в системе без кэш-памяти;
- T_c – время поиска в кэш-памяти и выбор требуемых данных в случае кэш-попадания;

- T_m – время доступа к данным в случае кэш-промаха; можно записать, что $T_m=T_{УП}+T_п$, где $T_п$ – так называемое время потерь при кэш-промахе, затрачиваемое на поиск запрашиваемых данных в кэш-памяти и помещение данных из управляющей памяти в кэш.

Тогда среднее время ожидания системы памяти, включающей в себя кэш-память, определяется как

$$T_E = H \cdot T_C + (1-H) \cdot T_m . \qquad (1)$$

Легко заметить, что для достижения высокой скорости доступа, близкой к скорости доступа чистой кэш-памяти, потребуется очень высокая частота попаданий.

Другой способ оценки эффективности использования кэш-памяти состоит в определении того, насколько кэш-память повышает скорость доступа к системе памяти. Полагая, что в общем случае $T_c < T_{УП} < T_m$, выигрыш в скорости доступа к данным в системе с кэш-памятью по сравнению с системой без кэш-памяти можно определить как отношение $T_{УП}$ к T_E:

$$S = T_{УП} / T_E = T_{УП} / (H \cdot T_C + (1-H) \cdot T_m). \qquad (2)$$

Проанализируем формулу (2).

1) Как показали результаты исследований, зависимость S(H) имеет экспоненциальный характер и возрастает с увеличением H.
2) Предположим, что $T_m \rightarrow T_{УП}$, то есть в случае кэш-промаха имеем цикл доступа, равный по времени циклу обращения в системе без кэш-памяти. Тогда формулу (2) можно преобразовать следующим образом:

$$S = T_{УП} / (H \cdot T_C + (1-H) \cdot T_{УП}) = 1 / (H \cdot T_C / T_{УП} + 1 - H) = 1 / (1 - H \cdot (1 - T_C / T_{УП})). \qquad (3)$$

Формула (3) наглядно показывает, что при $T_{УП}=T_m$ значение $S > 1$ при любом $H > 0$, то есть имеем постоянный выигрыш в быстродействии. Поскольку $T_m=T_{УП}+T_п$, то очевидным является стремление уменьшить $T_п$ с целью приближения T_m к $T_{УП}$.

3) Если $T_{УП} >> T_c$, то при $H>0,9$ имеем значительный (в 10 раз и выше) выигрыш S.

На основании вышесказанного можно сделать вывод, что для увеличения выигрыша в быстродействии в системе с кэш-памятью по сравнению с системой без кэш-памяти необходимы: увеличение вероятности кэш-попаданий H; снижение времени потерь при кэш-промахе $T_п$; уменьшение времени доступа к кэш-памяти T_c. Увеличение H и уменьшение $T_п$ достигается за счет использования оптимальной архитектуры кэш-памяти, алгоритма кэширования и параметров кэш-памяти. Уменьшение T_c возможно главным образом за счет использования для построения кэш-памяти быстродействующей элементной базы (например, микросхем памяти типа SRAM).

4. Организация КМУУ с кэшированием

Одним из способов уменьшения величины $t_{УП}$ является введения в структуру КМУУ элементов кэш-памяти, выполняющей кэширование УП по чтению (рис.4).

Здесь КК – кэш-контроллер, предназначенный для управления процессом выбора микрокоманды из памяти; КП – модуль кэш-памяти, время доступа к которому значительно меньше, чем к УП.

Кэш-контроллер, помимо управления УП и КП, также управляет подачей синхронизации на входы СЧАМК и РП. Если искомое слово найдено в кэше (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, 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 быстродействия схемы УП).

УП и КП организованы следующим образом. Известно, что в КМУУ управляющая память содержит только операционные микрокоманды. Пусть разрядность адреса $R =]\log_2 M[$, где M – число операторных вершин в ГСА. Пусть также число разрядов в микрокоманде равно K. Таким образом, УП может содержать до 2^R слов (далее под словом будем понимать микрокоманду разрядности K).

Управляющую память можно представить как набор последовательно идущих блоков по S слов в каждом. Всего таких блоков будет $2^R/S$. Количество слов в каждом блоке будем считать кратным степени двойки. Тогда кэш-память можно представить как некоторое множество строк, каждая из которых имеет длину, равную длине одного блока УП.

В каждой из N строк может быть размещен блок из S слов. Число строк N много меньше количества блоков в УП: $N << 2^R/S$. В любой момент времени в кэш-памяти содержится некоторое подмножество блоков из управляющей памяти. Если СЧАМК формирует адрес некоторого слова в УП, то в строку кэш-памяти переписывается весь блок, в котором находится требуемое слово. Это позволяет предупредить возможность того, что в следующем такте будет затребовано следующее слово из данного блока (что весьма вероятно при естественной адресации микрокоманд).

Поскольку число строк в кэш-памяти много меньше числа блоков в УП, то конкретная строка не может однозначно соответствовать конкретному блоку в УП. Чтобы определить, какой блок УП размещен в данный момент в некоторой строке кэш-памяти, каждая строка содержит специальное поле, называемое тэгом (tag). Обычно тэг содержит часть старших разрядов полного адреса УП, поступающего из СЧАМК.

Схема функционирует следующим образом. Кэш-контроллер получает из СЧАМК адрес очередной МК. В случае обнаружения необходимой МК в кэше (при совпадении определенной части разрядов адреса МК и значения тэга в одной из строк кэш-памяти) кэш-контроллер направляет прочитанную из кэша МК в СФМО. Если же требуемая МК не найдена, кэш-контроллер читает из УП тот блок, в котором находится нужная МК, выбирает и направляет МК в СФМО, после чего сохраняет прочитанный блок в некоторой строке кэш-памяти.

Кэш-память имеет ряд специфических характеристик, определяющих эффективность ее использования в каждом конкретном случае. К таким характеристикам относятся:

- размер кэш-памяти: составляет, как правило, от 1 до 10% основной памяти;
- способ отображения: прямое, ассоциативное и множественно-ассоциативное отображение;
- алгоритм замещения данных в кэш-памяти: LRU (Least-recently used), FIFO (First-in-first-out), Random и др.;
- размер блока и т.п.

В настоящее время не существует оптимальных значений для данных характеристик. Они во многом определяются областью использования цифровых устройств, характером реализуемых алгоритмов, элементной базой, и зачастую подбираются опытным путем.

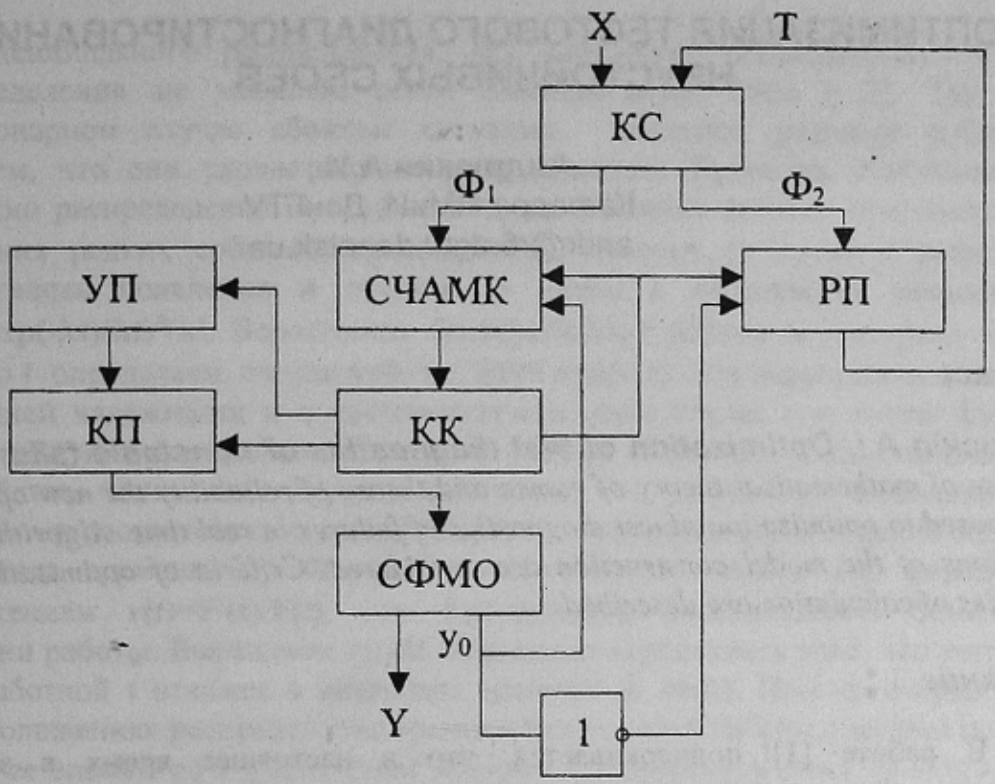

Рис.4. Структурная схема КМУУ с кэш-памятью

Заключение

Использование кэш-памяти в схемах КМУУ является компромиссным решением, дающим возможность значительного увеличения быстродействия при незначительном увеличении аппаратурных затрат. Принципы использования кэш-памяти в КМУУ могут быть использованы в дальнейшем для разработки методов синтеза микропрограммных устройств управления с кэш-памятью. С этой целью необходимо исследовать возможности реализации в составе КМУУ конкретных архитектур кэш-памяти, а также современной элементной базы, пригодной для построения кэш-памяти и кэш-контроллера.

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. Микропрограммное устройство управления как композиция автоматов с программируемой и жесткой логикой // Автоматика и вычислительная техника. – 1983. – № 4. – с. 42-50.
4. Интегральные микросхемы: Справочник / Под ред. Б. В. Тарабрина. – М.: Радио и связь, 1984 – 528 с.
5. J.-K. Peir, W.W. Hsu, and A.J. Smith. Functional Implementation Techniques for CPU cache memories // IEEE Transactions on Computers. – 1999. – № 2. – p. 100-104.

Поступила в редакційну колегію 20.04.2000 р.