

УДК 681.324

Д.В. НИКОЛАЕНКО

Автомобильно-дорожный институт Донецкого национального технического университета «АДИ ДонНТУ», Украина

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ЭВРИСТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ РАЗМЕЩЕНИЯ ДАННЫХ В КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВАХ УПРАВЛЕНИЯ С РАЗДЕЛЕНИЕМ КОДОВ И КЭШ-ПАМЯТЬЮ МИКРОКОМАНД

В работе выполнен анализ эффективности эвристического алгоритма оптимизации размещения микрокоманд в адресном пространстве управляющей памяти, позволяющего повысить величину вероятности кэш-попаданий модуля кэш-памяти в составе композиционных микропрограммных устройств управления с разделением кодов. На примере граф-схемы алгоритма средней сложности исследованы эффективности шести различных стратегий выбора первой операторной линейной цепи при формировании блока управляющей памяти. Получены значения прироста вероятности кэш-попаданий для различных размеров строки модуля кэш-памяти.

Ключевые слова: микропрограммное устройство управления, кэш-память, микрокоманда, граф-схема алгоритма, оптимизация вероятности кэш-попаданий, эвристический алгоритм, управляющая память.

Введение

Функционирование любой цифровой системы (ЦС) невозможно без использования в ее составе устройства управления (УУ), координирующего работу всех блоков системы [1]. Такие характеристики УУ, как быстродействие и аппаратные затраты, во многом определяют характеристики системы в целом. Актуальной научно-технической задачей является расширение области применения цифровых систем. Одним из способов решения данной задачи является увеличение быстродействия ЦС за счет увеличения быстродействия УУ.

Одним из способов реализации УУ является композиционное микропрограммное устройство управления (КМУУ), представляющее собой композицию автоматов с «жесткой» и «программируемой» логикой [2]. В КМУУ схема адресации традиционно реализуется в быстродействующем элементном базисе программируемых интегральных схем (ПЛИМ, ПЛИС), в то время как управляющая память (УП) реализуется в относительно дешевом и медленном базисе ПЗУ. В связи с невысокой по сравнению с ПЛИС скоростью работы схем ПЗУ время доступа к управляющей памяти (УП) может составлять значительную часть длительности такта работы КМУУ.

Композиционные микропрограммные устройства управления с разделением кодов являются одной из разновидностей КМУУ и используют в своих структурах специальные методы оптимизации аппа-

ратурных затрат в схеме адресации [3].

В работе [4] предложен способ увеличения быстродействия КМУУ с разделением кодов, заключающийся в использовании в структуре КМУУ модуля кэш-памяти микрокоманд. При этом фактором, определяющим эффективность использования модуля кэш-памяти и структуры в целом, является вероятность кэш-попаданий p_h , значение которой зависит как от параметров модуля кэш-памяти, так и от характеристик реализуемой граф-схемы алгоритма управления (ГСА). В работе [5] предложен подход к повышению величины p_h за счет оптимизации размещения операторных линейных цепей (ОЛЦ) в адресном пространстве управляющей памяти. В основу подхода положен ряд эвристических правил, справедливость которых была доказана экспериментально. На основании данного подхода в работе [6] был разработан эвристический алгоритм оптимизации размещения ОЛЦ, учитывающий особенности организации КМУУ с разделением кодов.

Следует отметить, что эффективность данного эвристического алгоритма с точки зрения повышения величины p_h в данное время остается неисследованной. Настоящая работа посвящена решению научной задачи исследования эффективности эвристического алгоритма оптимизации размещения ОЛЦ в адресном пространстве управляющей памяти композиционного микропрограммного устройства управления с разделением кодов и кэш-памятью микрокоманд. При этом целью работы является

обоснование эффективности и целесообразности применения рассматриваемого эвристического алгоритма.

1. Особенности проведения экспериментальных исследований

Предложенный в [6] эвристический алгоритм оптимизации размещения ОЛЦ может быть одинаково применен как для неэлементарных, так и для элементарных ОЛЦ [3], однако его эффективность для каждого типа ОЛЦ может быть различной. Под эффективностью эвристического алгоритма будем понимать прирост (в процентах) значения вероятности кэш-попаданий p_h заданного алгоритма управления по сравнению со значением вероятности до выполнения эвристического алгоритма (то есть при последовательном, неоптимизированном размещении ОЛЦ). В качестве средства моделирования будем использовать специально разработанную программную модель КМУУ с разделением кодов, позволяющую определить величину p_h статистическим путем.

Для проведения экспериментов были составлены две тестовые ГСА, одна из которых содержит неэлементарные, другая – элементарные ОЛЦ. Пусть каждая ГСА включает 100 ОЛЦ по 10 микрокоманд каждая. Таким образом, число микрокоманд в каждой ГСА равно 1000, что соответствует ГСА средней сложности [3]. Равное количество микрокоманд в ОЛЦ (не кратное числу слов в строке кэш-памяти) упрощает их адресацию в ПЗУ схемы УП. В КМУУ с разделением кодов для адресации 10 микрокоманд требуется 4 двоичных разряда. Таким образом, каждая ОЛЦ фактически занимает 16 адресов памяти и начинается с адреса, кратного 16.

Пусть в ГСА анализируются 4 логических условия x_1 - x_4 с заданными вероятностями переходов $p(x_1)$ - $p(x_4)$. Результаты исследований, полученные с использованием лишь одной ГСА с фиксированными значениями вероятностей логических условий (ЛУ), нельзя считать обобщенными – для обобщения потребовалось бы множество различных ГСА. Такое множество можно получить двумя способами:

1) изменяя структуру ГСА (количество и содержимое операторных и условных вершин, и связи между ними);

2) изменяя значения вероятностей выполнения логических условий $p(x_i)$ при неизменной структуре ГСА.

Для используемой программной модели, не имеющей генератора псевдослучайных ГСА с заданными параметрами, но учитывающей при моделировании значения вероятностей выполнения ЛУ, подходящим является второй способ. При про-

ведении исследований условимся выбирать для моделируемой ГСА различные значения вероятностей ЛУ следующим образом. Пусть вероятности выполнения каждого ЛУ принимают одно из двух возможных значений: $p(x_i)=0,1$ (ЛУ выполняется редко) и $p(x_i)=0,9$ (ЛУ выполняется часто). Если для четырех ЛУ, присутствующих в тестовых ГСА, рассматривать все возможные варианты их возможных значений, то таких вариантов оказывается 16: от варианта $\{p(x_1)=0,1; p(x_2)=0,1; p(x_3)=0,1; p(x_4)=0,1\}$ до варианта $\{p(x_1)=0,9; p(x_2)=0,9; p(x_3)=0,9; p(x_4)=0,9\}$. Каждый из вариантов определяет различный порядок выполнения микрокоманд, то есть по сути определяет отдельную ГСА.

Подчеркнем, что содержимое микрокоманд не является принципиальным и не влияет на работу модуля кэш-памяти и КМУУ. Это обуславливает абстрактный характер тестовых ГСА.

Также следует отметить, что одним из этапов эвристического алгоритма является выбор первой ОЛЦ при формировании нового блока памяти [6]. Такие способы выбора первой ОЛЦ в блоке названы стратегиями, что подчеркивает их значительное влияние на величину кэш-попаданий p_h . Таким образом, для исследования эвристического алгоритма важным является рассмотрение влияния каждой из стратегий.

При проведении исследований было принято размеры строки кэш-памяти выбирать различными, чтобы дать возможность эвристическому алгоритму работать с блоками памяти различной длины.

2. Анализ полученных результатов

В результате проведенных автором исследований были получены следующие основные результаты:

1. В большинстве случаев эвристический алгоритм *позволяет* добиться прироста величины вероятности кэш-попаданий для заданной ГСА. Примерно в 13% случаев для всех стратегий алгоритма величина p_h оказывалась ниже, чем при неоптимизированном размещении ОЛЦ. Данный факт подчеркивает эвристический характер алгоритма, не гарантирующий получение положительного результата в 100% случаев.

2. В таблицах 1 и 2 собраны величины средней и максимальной эффективностей эвристического алгоритма, т.е. значения прироста вероятности кэш-попаданий, достигнутые за счет оптимизации размещения микрокоманд. Таблица 1 соответствует неэлементарным ОЛЦ, таблица 2 – элементарным.

Из таблицы видно, что с ростом размера строки кэш-памяти средняя эффективность алгоритма снижается. Это объясняется тем, что при большом раз-

мере строки кэш-памяти «близкие» ОЛЦ оказываются в одном блоке памяти в основном благодаря большому размеру блока ПЗУ схемы УП, равного размеру строки кэш-памяти.

Таблица 1

Эффективность эвристического алгоритма для неэлементарных ОЛЦ

| Размер строки модуля кэш-памяти | Эффективность эвристического алгоритма, % | |
|---------------------------------|---|--------------|
| | Средняя | Максимальная |
| 1x32 | 3,038844 | 8,165900 |
| 1x64 | 1,514025 | 3,373400 |
| 1x128 | 0,291094 | 1,922900 |

Таблица 2

Эффективность эвристического алгоритма для неэлементарных ОЛЦ

| Размер строки модуля кэш-памяти | Эффективность эвристического алгоритма, % | |
|---------------------------------|---|--------------|
| | Средняя | Максимальная |
| 1x32 | 2,767319 | 6,647300 |
| 1x64 | 1,113588 | 2,752300 |
| 1x128 | 0,269031 | 1,032100 |

Для тестовых ГСА максимальный прирост величины r_h в проведенных исследованиях составил 8,16% (подчеркнем, что данный прирост является беспроегрешным, не влияет на характеристики устройства, в частности, на аппаратные затраты).

Также проведен анализ эффективности предложенных в работе стратегий 1-6 выбора первой ОЛЦ в блок данных управляющей памяти. В таблицах 2 и 3 собраны результаты применения шести стратегий эвристического алгоритма. Цифра в ячейке показывает, сколько раз из проведенных 16 экспериментов данная стратегия давала максимальный процент прироста величины r_h среди остальных стратегий, то есть оказывалась в числе лучших. Таблица 3 соответствует неэлементарным ОЛЦ, таблица 4 – элементарным.

Таблица 3

Эффективность стратегий для неэлементарных ОЛЦ

| Размер строки | Стратегия | | | | | |
|---------------|-----------|----|----|----|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1x32 | 1 | 12 | 1 | 11 | 4 | 8 |
| 1x64 | 6 | 7 | 7 | 7 | 2 | 3 |
| 1x128 | 10 | 4 | 10 | 4 | 4 | 3 |

С увеличением размера строки кэш-памяти эффективность стратегий 1 и 3 возрастает, а 2 и 4 – уменьшается. Стратегии 5 и 6 для неэлементарных и элементарных ОЛЦ ведут себя неодинаково, однако в целом можно утверждать, что их эффективность не возрастает.

Таблица 4

Эффективность стратегий для неэлементарных ОЛЦ

| Размер строки | Стратегия | | | | | |
|---------------|-----------|----|----|----|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1x32 | 1 | 13 | 1 | 13 | 8 | 7 |
| 1x64 | 7 | 6 | 7 | 6 | 6 | 6 |
| 1x128 | 12 | 4 | 11 | 4 | 2 | 4 |

Наибольшую эффективность показали стратегии 2 и 4 в случае элементарных ОЛЦ и кэш-памяти размером 1x128 – они были в числе наиболее эффективных стратегий в 13 случаях каждая. Наименьшую эффективность показали стратегии 1 и 3 для обоих типов ОЛЦ и кэш-памяти размером 1x32. Однако данные стратегии оказались наиболее эффективными для больших размеров строки кэш-памяти.

Если просуммировать значения столбцов обеих таблиц, то по суммарным значениям лидируют стратегии 2 и 4 со значениями 46 и 45 соответственно. Стратегия 5 со значением 26 оказалась в сумме наименее эффективной. Стратегии 5 и 6 занимают средние позиции среди остальных стратегий. Интересно отметить, что максимальные величины эффективности эвристического алгоритма (8,16% в табл. 1 и 6,64% в табл. 2) были получены (в том числе) с помощью стратегий 5 и 6.

В целом можно утверждать, что все шесть предложенных стратегий «подстраховывают» друг друга, обеспечивая положительную эффективность эвристического алгоритма для различных ГСА и размеров строки кэш-памяти.

Заключение

Предложенный в работе [6] эвристический алгоритм оптимизации размещения ОЛЦ позволяет аналитическим путем повысить величину вероятности кэш-попаданий, что, в конечном итоге, приводит к повышению эффективности структур КМУУ с разделением кодов и кэш-памяти, расширяя тем самым область их эффективного применения.

Научная новизна работы заключается в получении численных значений эффективности эвристического алгоритма, что позволяет сделать вывод о целесообразности его применения для любой реализуемой ГСА. В случае, если эвристический алгоритм дал худшие результаты по сравнению с последовательным размещением ОЛЦ, разработчик может использовать последовательное размещение ОЛЦ как наиболее эффективное в данном конкретном случае. Практическая значимость работы заключается в том, что применение эвристического алгоритма не вносит каких-либо изменений в струк-

туру устройства управления. По этой причине схемы устройств, спроектированных с использованием эвристического алгоритма и без него, отличаются друг от друга лишь значением вероятности кэш-попаданий, причем в 84% случаев – в пользу эвристического алгоритма.

В перспективе результаты данной работы планируется использовать при разработке комплексного аналитического метода оценки эффективности структур КМУУ с разделением кодов и кэш-памятью микрокоманд. Это, в свою очередь, позволит реализовать полученные результаты работы как часть САПР цифровых устройств управления.

Литература

1. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М.: Физматгиз, 1962. – 476 с.

2. Баркалов А.А. Синтез микропрограммных устройств управления / А.А. Баркалов, А.В. Палагин. – К.: Институт кибернетики НАН Украины, 1997. – 135 с.

3. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДонНТУ, 2002. – 262 с.

4. Николаенко Д.В. Организация композиционных микропрограммных устройств управления с разделением кодов и кэш-памятью / А.А. Баркалов, С.А. Ковалев, Р.М. Бабаков, Д.В. Николаенко // Искусственный интеллект. – 2007. – №3. – С. 135-138.

5. Николаенко Д.В. Эвристический подход к адресации микрокоманд в композиционных микропрограммных устройствах управления с разделением кодов и кэш-памятью / А.А. Баркалов, С.А. Ковалев, Р.М. Бабаков, Д.В. Николаенко // Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика і обчислювальна техніка» (ІКОТ-2008). Випуск 9 (132) – Донецьк: ДонНТУ. – 2008. – С. 188-194.

6. Николаенко Д.В. Эвристический алгоритм оптимизации размещения микрокоманд в композиционном микропрограммном устройстве управления с разделением кодов и кэш-памятью / А.А. Баркалов, С.А. Ковалев, Р.М. Бабаков, Д.В. Николаенко // Искусственный интеллект. – 2008. – №1. – С. 20-29.

Поступила в редакцию 1.02.2010

Рецензент: д-р техн. наук, проф., проф. кафедры ЭВМ А.А. Баркалов, Донецкий национальный технический университет, Донецк, Украина.

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЕВРИСТИЧНОГО АЛГОРИТМУ ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ДАНИХ В КОМПОЗИЦІЙНИХ МІКРОПРОГРАМНИХ ПРИСТРОЯХ КЕРУВАННЯ З ПОДІЛОМ КОДІВ І КЕШ-ПАМ'ЯТТЮ МІКРОКОМАНД

Д.В. Ніколаєнко

У роботі виконаний аналіз ефективності евристичного алгоритму оптимізації розміщення мікрокоманд в адресному просторі керуючої пам'яті, що дозволяє підвищити величину ймовірності кэш-влучень модуля кэш-пам'яті у складі композиційних мікропрограмних пристроїв керування з поділом кодів. На прикладі граф-схеми алгоритму середньої складності досліджені ефективності шести різних стратегій вибору першого операторного лінійного ланцюга при формуванні блоку керуючої пам'яті. Отримано значення приросту ймовірності кэш-влучень для різних розмірів рядка модуля кэш-пам'яті.

Ключові слова: мікропрограмний пристрій керування, кэш-пам'ять, мікрокоманда, граф-схема алгоритму, оптимізація ймовірності кэш-влучень, евристичний алгоритм, керуюча пам'ять.

RESEARCH OF EFFICIENCY OF HEURISTIC ALGORITHM OF OPTIMIZATION OF ALLOCATION OF DATA IN COMPOSITIONAL MICROPROGRAM CONTROL UNITS WITH DIVISION OF CODES AND CACHE MEMORY OF MICROCOMMANDS

D.V. Nikolaenko

In this paper the analysis of efficiency of heuristic algorithm of optimisation of allocation of microinstructions in address space of the control memory, allowing to raise the value of probability of caches-hits of the cache memory unit as a part of compositional microprogram control units with division of codes, is made. On an example a flow-chart of average complexity efficiency of six various strategies of a choice of the first operator linear chain at creation of the block of the control memory, are researched. Values of increase of probability of caches-hits for the various sizes of cache line are received.

Keywords: microprogram control unit, cache memory, microinstruction, flow-chart, optimization of probability of caches-hits, heuristic algorithm, control memory.

Николаенко Денис Владимирович – ассистент кафедры экономической кибернетики, Автомобильно-дорожный институт Донецкого национального технического университета, Горловка, Украина, e-mail: adi-eki@yandex.ru.