

Математичні моделі й методи в системах автоматизованого управління, проектування

УДК 681.324

Д. В. Ніколаєнко, канд. техн. наук, доц.,
Автомобільно-дорожній інститут Донецького національного технічного університету
adi-eki@yandex.ru

Результативність евристичного алгоритму розміщення даних в КМПК з кеш-пам'яттю мікрокоманд

У роботі виконаний аналіз ефективності евристичного алгоритму оптимізації розміщення мікрокоманд в адресному просторі керуючої пам'яті, що дозволяє підвищити величину ймовірності кеш-влучень модуля кеш-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 кеш-влучень для різних розмірів рядка модуля кеш-пам'яті.

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

Вступ

Функціонування будь якої цифрової системи (ЦС) неможливо без використання в її складі пристрою керування (ПК), який координує роботу всіх блоків системи [1]. Такі характеристики ПК, як швидкодія та апаратурні витрати, багато в чому визначають характеристики системи в цілому.

Актуальною науково-технічною задачею є розширення галузі застосування цифрових систем. Одним із способів вирішення цієї задачі є збільшення швидкодії ЦС за рахунок збільшення швидкодії ПК.

Одним із способів реалізації ПК є композиційні мікропрограмні пристрої керування (КМПК), що представляють собою композицію автоматів з «жорсткою» та «програмованою» логікою [2]. В КМПК схема адресації традиційно реалізується в швидкодіючому елементному базисі прогамованих інтегральних схем (ПЛМ, ПЛІС), в той час як керуюча пам'ять (КП) реалізується у відносно дешевому і повільному базисі ПЗП. У зв'язку з невисокою в порівнянні з ПЛІС швидкістю роботи схем ПЗП час доступу до керуючої пам'яті (КП) може становити значну частину тривалості такту роботи КМПК.

Композиційні мікропрограмні пристрої керування з розподілом кодів є одним з різновидів КМПК і використовують у своїх структурах спеціальні методи оптимізації апаратурних витрат у схемі адресації [3].

В роботі [4] запропонований спосіб збільшення швидкодії КМПК з розподілом кодів, що полягає у використанні в структурі КМПК модуля кеш-пам'яті мікрокоманд. При цьому фактором, що визначає ефективність використання модуля кеш-пам'яті і структури в цілому, є ймовірність кеш-влучень ph , значення якої залежить як від параметрів модуля кеш-пам'яті, так і від характеристик реалізованої граф-схеми алгоритму керування (ГСА). В роботі [5] запропоновано підхід до підвищення величини ph за рахунок оптимізації розміщення операторних лінійних ланцюгів (ОЛЛ) в адресному просторі керуючої пам'яті.

В основу підходу покладено ряд евристичних правил, справедливості яких була доведена експериментально. На підставі даного підходу в роботі [6] був розроблений евристичний алгоритм оптимізації розміщення ОЛЛ, що враховує особливості організації КМПК з розподілом кодів.

Слід зазначити, що ефективність даного евристичного алгоритму з точки зору підвищення величини ph в даний час залишається недослідженою. Дана робота присвячена вирішенню наукової задачі дослідження ефективності евристичного алгоритму оптимізації розміщення ОЛЛ в адресному просторі керуючої пам'яті композиційного мікропрограмного пристрою керування з розподілом кодів і кеш-пам'яттю мікрокоманд. При цьому метою роботи є обґрунтування ефективності та доцільності застосування розглянутого евристичного алгоритму.

1. Умови проведення експериментів

Запропонований в [6] евристичний алгоритм оптимізації розміщення ОЛЛ може бути однаково застосований як для неелементарних, так і для елементарних ОЛЛ [3], проте його ефективність для кожного типу ОЛЛ може бути різною. Під ефективністю евристичного алгоритму будемо розуміти приріст (у відсотках) значення ймовірності кеш-влучень ph заданого алгоритму управління в порівнянні зі значенням ймовірності до виконання евристичного алгоритму (тобто при послідовному, неоптимізованих розміщенні ОЛЛ). В якості засобу моделювання будемо використовувати спеціально розроблену програмну модель КМПК з розділення кодів, що дозволить визначити величину ph статистичним шляхом.

Для проведення експериментів були складені дві тестові ГСА, одна з яких містить неелементарні, інша - елементарні ОЛЛ. Нехай кожна ГСА включає 100 ОЛЛ по 10 мікрокоманд кожна. Таким чином, кількість мікрокоманд у кожній ГСА одне і дорівнює 1000, що відповідає ГСА середньої складності [3]. Рівна кількість мікро-команд в ОЛЛ (не кратне числу слів у рядку кеш-пам'яті) спрощує їх адресацію в ПЗП схеми КП. В КМПК з розділенням кодів для адресації 10 мікрокоманд потрібно 4 двійкових розряди. Таким образом, кожна ОЛЛ фактично займає 16 адрес пам'яті і починається з адреси, кратного 16.

Нехай в ГСА аналізуються 4 логічних умови x_1 - x_4 із заданими ймовірностями переходів $p(x_1)$ - $p(x_4)$. Результати досліджень, отримані з використанням лише однієї ГСА з фіксованими значеннями ймовірностей логічних умов (ЛУ), не можна вважати узагальненими – для узагальнення було б потрібно безліч різних ГСА. Таку безліч можна отримати двома способами:

- 1) змінюючи структуру ГСА (кількість і вміст операторних і умовних вершин, і зв'язку між ними);
- 2) змінюючи значення ймовірностей виконання логічних умов $p(x_i)$ при незмінній структурі ГСА.

Для використовуваної програмної моделі, яка не має генератора псевдовипадкових ГСА із заданими параметрами, але враховує при моделюванні значення ймовірностей виконання ЛУ, придатним є другий спосіб. При проведенні досліджень умовимося вибирати для моделюваної ГСА різні значення ймовірностей ЛУ наступним чином. Нехай ймовірності виконання кожного ЛУ приймають одне з двох можливих значень: $p(x_i) = 0,1$ (ЛУ виконується рідко) і $p(x_i) = 0,9$ (ЛУ виконується часто). Якщо для чотирьох ЛУ, присутніх в тестових ДСА, розглядати всі можливі варіанти їх можливих значень, то таких варіантів виявляється 16: від варіанту $\{p(x_1) = 0,1; p(x_2) = 0,1; p(x_3) = 0,1; p(x_4) = 0,1\}$ до варіанту $\{p(x_1) = 0,9; p(x_2) = 0,9; p(x_3) = 0,9; p(x_4) = 0,9\}$. Кожен з варіантів визначає різний порядок виконання мікрокоманд, тобто по суті визначає окрему ГСА.

Підкреслимо, що вміст мікрокоманд не є принциповим і не впливає на роботу модуля кеш-пам'яті і КМПК. Це обумовлює абстрактний характер тестових ГСА.

Також слід зазначити, що одним з етапів евристичного алгоритму є вибір першої ОЛЛ при формуванні нового блоку пам'яті [6]. Такі способи вибору першої ОЛЛ в блоці названі стратегіями, що підкреслює їх значний вплив на величину кеш-влучень ph . Таким чином, для дослідження евристичного алгоритму важливим є розгляд впливу кожної із стратегій.

При проведенні досліджень було прийнято розміри рядка кеш-пам'яті вибирати різними, щоб дати можливість евристичному алгоритму працювати з блоками пам'яті різної довжини.

2. Аналіз результатів досліджень

В результаті проведених автором досліджень були отримані наступні основні результати:

1. У більшості випадків евристичний алгоритм дозволяє добитися приросту величини ймовірності кеш-влучень для заданої ГСА. Приблизно в 13% випадків для всіх стратегій алгоритму величина ph виявлялася нижче, ніж при неоптимізованих (послідовному) розміщенні ОЛЛ. Даний факт підкреслює евристичний характер алгоритму, не гарантує отримання позитивного результату у 100% випадків.

2. У таблицях 1 і 2 зібрані величини середньої та максимальної ефективностей евристичного алгоритму, тобто значення приросту ймовірності кеш-влучень, досягнуті за рахунок оптимізації розміщення мікрокоманд. Таблиця 1 відповідає неелементарним ОЛЛ, таблиця 2 – елементарним.

Аналіз таблиці показує, що з ростом розміру рядка кеш-пам'яті середня ефективність евристичного алгоритму знижується. Це пояснюється тим, що при великому розмірі рядка кеш-пам'яті «близькі» ОЛЛ опиняються в одному блоці пам'яті не стільки завдяки оптимізації їх розміщення, скільки завдяки великому розміру блоку ПЗУ схеми УП, рівного розміру рядка кеш-пам'яті.

У таблиці 1 наведені величини середньої та максимальної ефективності евристичного алгоритму отримані експериментальним шляхом.

Таблиця 1.

Ефективність евристичного алгоритму
для неелементарних ОЛЛ

Розмір рядка модуля кеш-пам'яті	Ефективність евристичного алгоритму, %	
	Середня	Максимальна
32	3,04	8,17
64	1,51	3,37
128	0,29	1,92

При цьому для тестових ГСА максимальний приріст величини ph в проведених дослідженнях склав 8,16% (підкреслимо, що даний приріст є безпрограшним, тобто ніяк не впливає на інші характеристики пристрою, зокрема, на апаратурні витрати).

Таблиця 2.
Ефективність евристичного алгоритму
для неелементарних ОЛЛ

Розмір рядка модуля кеш-пам'яті	Ефективність евристичного алгоритму, %	
	Середня	Максимальна
32	2,77	6,65
64	1,11	2,75
128	0,27	1,03

Також проведено аналіз ефективності запропонованих в роботі стратегій 1-6 вибору першої ОЛЛ у блок даних керуючої пам'яті. У таблицях 2 і 3 зібрані результати застосування шести стратегій евристичного алгоритму. Цифра в комірці показує, скільки разів із проведених 16 експериментів дана стратегія давала максимальний відсоток приросту величини ph серед інших стратегій, тобто виявлялася в числі кращих. Таблиця 3 відповідає неелементарним ОЛЛ, таблиця 4 - елементарним.

Таблиця 3.
Ефективність стратегій
для неелементарних ОЛЛ

Розмір рядка	Стратегія					
	1	2	3	4	5	6
32	1	12	1	11	4	8
64	6	7	7	7	2	3
128	10	4	10	4	4	3

Таблиця 4.
Ефективність стратегій
для елементарних ОЛЛ

Розмір рядка	Стратегія					
	1	2	3	4	5	6
32	1	13	1	13	8	7
64	7	6	7	6	6	6
128	12	4	11	4	2	4

Із збільшенням розміру рядка кеш-пам'яті ефективність стратегій 1 і 3 зростає, а 2 і 4 - зменшується. Стратегії 5 і 6 для неелементарних і

елементарних ОЛЛ поведуться неоднаково, проте в цілому можна стверджувати, що їхня ефективність не зростає.

Найбільшу ефективність показали стратегії 2 і 4 у разі елементарних ОЛЛ і кеш-пам'яті розміром 1x128 – вони були в числі найбільш ефективних стратегій в 13 випадках кожна. Найменшу ефективність показали стратегії 1 і 3 для обох типів ОЛЛ і кеш-пам'яті розміром 1x32. Однак дані стратегії виявилися найбільш ефективними для більших розмірів рядка кеш-пам'яті.

Якщо підсумувати значення стовпців обох таблиць, то за сумарними значеннями ведуть стратегії 2 і 4 зі значеннями 46 і 45 відповідно. Стратегія 5 зі значенням 26 виявилася в сумі найменш ефективною. Стратегії 5 і 6 посідають середні позиції серед інших стратегій.

Цікаво відзначити, що максимальні величини ефективності евристичного алгоритму (8,16% в табл. 1 і 6,64% в табл. 2) були отримані (у тому числі) за допомогою стратегій 5 і 6. В цілому можна стверджувати, що всі шість запропонованих стратегій «підстраховують» один одного, забезпечуючи позитивну ефективність евристичного алгоритму для різних ГСА та розмірів рядка кеш-пам'яті.

Висновки

Запропонований в роботі [6] евристичний алгоритм оптимізації розміщення ОЛЛ дозволяє аналітичним шляхом підвищити величину ймовірності кеш-влучень, що, в остаточному підсумку, призводить до підвищення ефективності структур КМПК з розподілом кодів та кеш-пам'яттю, розширюючи тим самим область їх ефективного застосування.

Наукова новизна роботи полягає в отриманні чисельних значень ефективності евристичного алгоритму, що дозволяє зробити висновок про доцільність його застосування для будь-якої реалізованої ГСА. У разі, якщо евристичний алгоритм дав гірші результати в порівнянні з послідовним розміщенням ОЛЛ, розробник може використовувати послідовне розміщення ОЛЛ як найбільш ефективне в даному конкретному випадку. Практична значимість роботи полягає в тому, що застосування евристичного алгоритму не вносить жодних змін у структуру пристрою керування. З цієї причини схеми пристроїв, спроектованих з використанням евристичного алгоритму і без нього, відрізняються один від одного лише значенням ймовірності кеш-влучень, причому в 84% випадків - на користь евристичного алгоритму.

Список літератури

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. – Киев: Институт кибернетики НАН України, 1997. – 135 с.

3. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДонНТУ, 2002. – 262 с.
4. 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. 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). – 2008. – Випуск 9 (132). – С. 188-194.
6. 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.

Надійшла до редакції 20.05.2013

Д. В. НИКОЛАЕНКО

Автомобильно-дорожный институт Донецкого национального технического университета

РЕЗУЛЬТАТИВНОСТЬ ЭВРИСТИЧЕСКОГО АЛГОРИТМА РАЗМЕЩЕНИЯ ДАННЫХ В КМУУ С 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.

D.V. NIKOLAIEENKO

Automobile and Road Institute of Donetsk National Technical University

EFFECTIVENESS OF THE HEURISTIC ALGORITHM OF DATA ALLOCATION IN CMCU WITH CACHE MEMORY OF MICROINSTRUCTIONS

Relevant scientific challenge is to expand the scope of digital systems. One way of solving this problem is to increase the performance of digital systems by increasing the speed control memory.

Based on the previously proposed method for increasing the speed of compositional microprogram control units with codes division, which employs the composite structure of the microprogram control unit cache microinstructions, we analyzed the effectiveness of the heuristic optimization algorithm for allocating microinstructions in the control memory address space, which allows increasing the probability of cache hits of the cache memory module in compositional microprogram control devices with codes division. The factor, which determines the efficiency of cache memory and the whole structure, is cache hits probability ph , the value of which depends on the cache memory parameters and the characteristics of the implemented flowgraph of the control algorithm.

For the experiments we made up two test flowgraphs of the algorithm. One of them contained non-elementary linear chains and the other contained elementary operational linear chains. Each flowgraph included 100 operational linear chains, each with 10 microinstructions. Thus, the number of microinstructions in each flowgraph of the algorithm is 1000, which corresponds to the flowgraph of an algorithm of medium complexity. On the example of the flowgraph of such an algorithm we studied the effectiveness of six different strategies for selecting the first operator linear chain in the formation of a control memory unit. We obtained the values of cache hits probability increment for various sizes of cache memory module line.

The aim is to study the effectiveness and feasibility of this heuristic algorithm application. As a result, we obtained numerical values of heuristic algorithm efficiency.

The importance of the work is in the fact that heuristic algorithm application does not affect the control device structure. That is why the circuits of the devices designed with the use of the heuristic algorithm and without it differ from each other only by cache hits probability, and in 84% of cases the heuristic algorithm has advantages.

Key words: microprogram control unit, cache memory, microinstruction, flowgraph, cache hits probability optimization, heuristic algorithm, control memory.