

# Комп'ютерна Модель Блоку LRU Кеш - пам'яті Процесорного Ядра Архітектури IA - 32

Вадим Пуйденко

Циклова комісія "Комп'ютерні науки"  
Харківський радіотехнічний технікум  
vapuydenko@gmail.com  
+380503435493

## The Computer's Model of the Memory Cache LRU Unit of the Processor's Core of the Architecture IA-32

Vadym Puydenko

The department "Computer's sciences"  
Kharkiv radio engineering technical school  
vapuydenko@gmail.com  
+380503435493

Анотація—в представленій науково-дослідницькій праці автором вирішується задача синтезу функціональної логіки роботи блоку LRU внутрішньої кеш-пам'яті центрального процесора при умовах промахів, влучань та заповнень рядків блоку даних внутрішньої кеш-пам'яті. Аналізується архітектура моделей логіки обрання рядків серед недостовірних певної множини для заповнення та логіки керування заміщенням рядків при повній достовірності обраної множини блоку даних. Проводиться мінімізація неповністю визначених функцій, що перемикаються:  $B^+=f(L,B)$  обрання рядків множини серед достовірних та формування значень бітів  $B_2^+B_1^+B_0^+$  блоку LRU з урахуванням попереднього стану цих бітів. В результаті синтезу отримані мінімальні логічні рівняння, які описують певну функціональну логіку реальної роботи блоку LRU внутрішньої кеш-пам'яті, як компоненти центрального процесора архітектури IA-32.

Abstract—in the presented scientific research work the author solves the problem of synthesis of the functional logic of the work of the unit of LRU of the internal cache memory of the central processor under the conditions of misses, hits and filling the rows of the data block of the internal cache memory. The architecture of models of the logic of selection of row among the unreliable certain set for filling and the logic of row replacement management with full accuracy of the selected set of data blocks are analyzed. Minimization of completely and not fully defined switching functions is carried out : function of  $L=f(B)$  and

function of  $B^+=f(L,B)$  selecting of sets of rows among reliable and forming values of bits  $B_2^+B_1^+B_0^+$  of the LRU block taking into account the previous state of these bits. As a result of synthesis minimal logical equations that describe a certain functional logic of the actual work of the unit of reliability/LRU of the on-chip cache memory unit as components of the central processor architecture of IA – 32 are obtained.

*Ключові слова—алгоритм pseudo-LRU, функціональна логіка, внутрішня кеш-пам'ять, блок LRU.*

*Keywords—algorithm of the pseudo-LRU, functional logic, on-chip memory cache, LRU unit.*

### I. ВСТУП

Процесори базису x86 мають досить високу швидкодію виконання операцій всього за один такт циклу. Відомо, що зовнішня пам'ять невзможливо забезпечувати дані у кожному такті, тому для підвищення загальної продуктивності потребується внутрішня кеш-пам'ять, яка гарантує досить швидке звертання до даних. Якщо всі операції програми використовують внутрішні ресурси, тоді досягається її найшвидке виконання, оскільки всі запити задовольняють внутрішні регістри і кеш-пам'ять. Однак при промахах в операціях читання і у всіх операціях запису процесору доводиться звертатися до зовнішньої шини, що знижує продуктивність системи. Зокрема, в операціях запису у внутрішню кеш-пам'ять, процесор

може наштовхнутися з проблемою вибору певного рядка множини блоку даних за умови достовірності всіх рядків. В цьому випадку процесор залучає блок достовірності/LRU, функціональна логіка якого гарантує обрання певного рядка множини для запису даних з використанням алгоритму pseudo-LRU.

## II. ПОСТАНОВКА ЗАДАЧІ СИНТЕЗУ БЛОКУ LRU

Алгоритм pseudo-LRU діє наступним чином. Коли у циклі читання відбувається промах і в кеш-пам'яті необхідно передати з пам'яті новий рядок, приходиться обирати для заповнення один з чотирьох рядків множини. Якщо у множині є недостовірний рядок (його біт достовірності R дорівнює 0), то для заповнення обирається саме цей рядок. Коли ж всі рядки у множині достовірні (всі чотири достовірності біти R місять 1), то рядок, що замінюється, обирається із залученням бітів з блоку LRU. Позначимо рядки у множині через  $L_0, L_1, L_2$  та  $L_3$ . Кожній множині у блоці LRU відповідає три біти  $B_0, B_1$  та  $B_2$ , котрі модифікуються при кожному влучанні або заповненні наступним чином (рис.1) :

- якщо останнє звертання було до рядка  $L_0$  або  $L_1$ , то біт  $B_0=1$ , а при звертанні до рядка  $L_2$  або  $L_3$  біт  $B_0=0$ ;
- якщо останнє звертання у парі  $L_0-L_1$  було до рядка  $L_0$ , то біт  $B_1=1$ , а при звертанні до рядка  $L_1$  біт  $B_1=0$ ;
- якщо останнє звертання у парі  $L_2-L_3$  було до рядка  $L_2$ , то біт  $B_2=1$ , а при звертанні до рядка  $L_3$  біт  $B_2=0$ ;

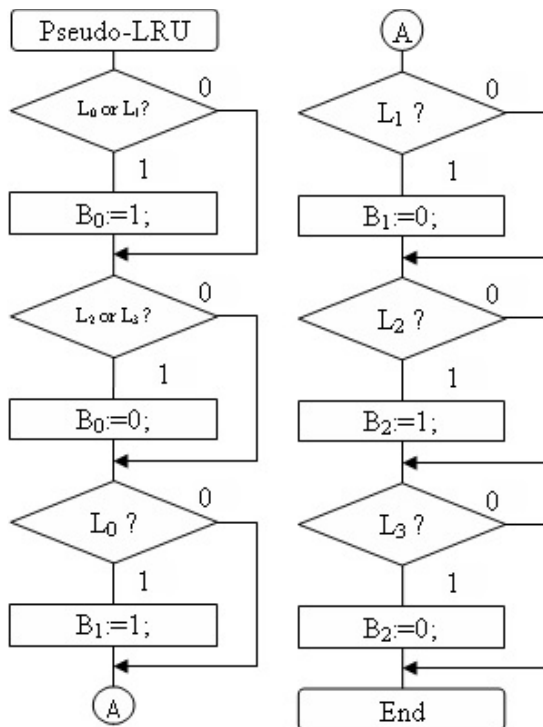


Рис. 1. Блок-схема алгоритму псевдо-LRU

## III. СИНТЕЗ ЗАДАЧІ ДОСЛІДЖЕННЯ

Згідно з логікою функціонування алгоритму створимо повну таблицю істини обрання певного рядка (елемента) відповідної множини блоку даних (табл.1):

ТАБЛИЦЯ I. ОБРАННЯ ПЕВНОГО РЯДКА  $L_i$

Біти LRU			Елемент $L=f(b)$
$B_2$	$B_1$	$B_0$	
0	0	0	$L_0$
0	0	1	$L_2$
0	1	0	$L_1$
0	1	1	$L_2$
1	0	0	$L_0$
1	0	1	$L_3$
1	1	0	$L_1$
1	1	1	$L_3$

Спрощена таблиця істини обрання певного рядка (елемента) відповідної множини блоку даних набуде вигляду (табл. 2):

ТАБЛИЦЯ II. ОБРАННЯ ПЕВНОГО РЯДКА  $L_i$

Біти LRU			Елемент $L=f(b)$
$B_2$	$B_1$	$B_0$	
x	0	0	$L_0$
x	1	0	$L_1$
0	x	1	$L_2$
1	x	1	$L_3$

Складемо відповідні карти Карно з метою отримання МДНФ логічних функцій  $L_i=f(b)$ , що перемикаються, (де  $i \in \{0,1,2,3\}$ ) (рис. 2).

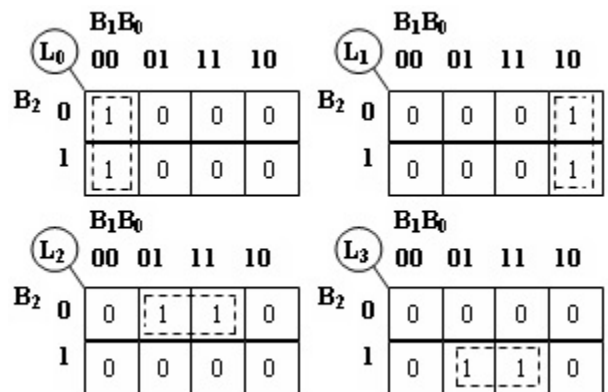


Рис. 2. Мінімізація логіки обрання рядків  $L_i$

Отже, логічні рівняння обрання певного рядка  $L_i$  відповідної множини набудуть вигляду (1):

$$L_0 = \overline{B_1} \& \overline{B_0}; \quad L_1 = B_1 \& \overline{B_0}; \quad L_2 = \overline{B_2} \& B_0; \quad L_3 = B_2 \& B_0; \quad (1)$$

З урахуванням отриманих вище логічних рівнянь (1) фрагмент функціональної логіки обрання певного рядка  $L_i$  набуде вигляду (рис. 3)

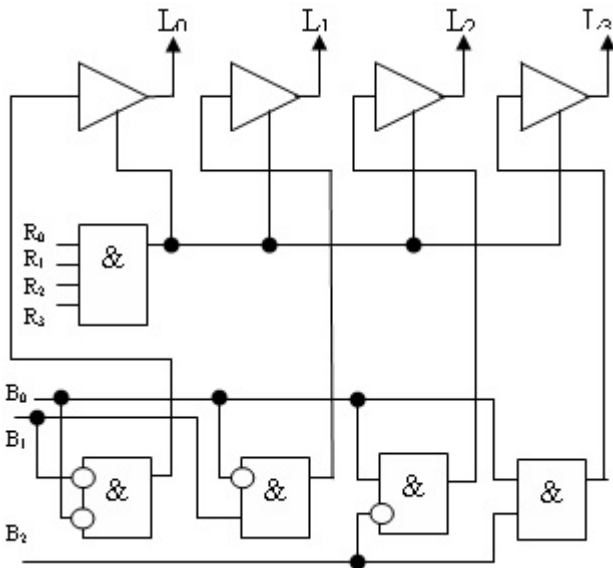


Рис. 3. Логіка обрання певного рядка  $L_i$

Складемо таблиці істини відновлення бітів  $V_i^+$ .  
Таблиця істини відновлення біту  $V_0^+$  буде такою (табл. 3):

ТАБЛИЦЯ III. Відновлення біту  $V_0^+$

Рядки множини				Біти блоку LRU	
$L_3$	$L_2$	$L_1$	$L_0$	$V_0$	$V_0^+$
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	x	x
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	x	x
0	1	1	0	x	x
0	1	1	1	x	x
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	x	x
1	0	1	x	x	x
1	1	x	x	x	x

Складемо відповідні карти Карно з метою отримання МДНФ логічних функцій  $V_i^+=f(L)$ , що перемикаються, (де  $i \in \{0,1,2\}$ ) для створення логічних рівнянь відновлення прапорців  $V_2^+V_1^+V_0^+$ . Карта Карно відновлення біту  $V_0^+$  набуде вигляду (рис. 4):

$V_0^+$	$L_1L_0B_0$							
$L_3L_2$	000	001	011	010	110	111	101	100
00	0	1	1	1	x	x	x	1
01	0	0	x	x	x	x	x	x
11	x	x	x	x	x	x	x	x
01	0	0	x	x	x	x	x	x

Рис. 4. Мінімізація логіки відновлення біту  $V_0$

Таким чином, логічне рівняння відновлення значення біта  $V_0^+$  має вигляд (2):

$$V_0^+ = \overline{L_3} \& \overline{L_2} \& B_0 \vee L_1 \vee L_0 = \overline{L_3 \& L_2 \& B_0} \& L_1 \& L_0; \quad (2)$$

Таблиця істини відновлення біту  $V_1^+$  набуде вигляду (табл. 4):

ТАБЛИЦЯ IV. Відновлення біту  $V_1^+$

Рядки множини		Біти блоку LRU	
$L_1$	$L_0$	$V_1$	$V_1^+$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	x

Відповідна карта Карно буде такою (рис.5):

$V_1^+$		$L_0B_1$			
$L_1$	0	00	01	11	10
0	0	1	1	1	
1	0	0	x	x	

Рис. 5. Мінімізація логіки відновлення біту  $V_1$

Логічне рівняння формування значення біта  $V_1^+$  виглядатиме так (3):

$$V_1^+ = \overline{L_1} \& B_1 \vee L_0 = \overline{L_1 \& B_1} \& L_0; \quad (3)$$

Таблиця істини відновлення біту  $B_2^+$  набуде вигляду (таблиця 5):

ТАБЛИЦЯ V. ВІДНОВЛЕННЯ БІТУ  $B_2^+$

Рядки множини		Біти блоку LRU	
$L_3$	$L_2$	$B_2$	$B_2^+$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	x

Відповідна карта Карно буде такою (рис.6):

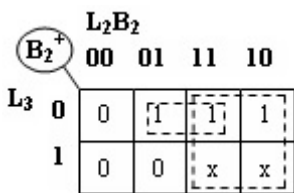


Рис. 6. Мінімізація логіки відновлення біту  $B_2^+$

Таким чином, логічне рівняння відновлення значення біта  $B_2^+$  має вигляд (4):

$$B_2^+ = \overline{L_3} \& B_2 \vee L_2 = \overline{\overline{L_3} \& B_2 \& \overline{L_2}} \quad (4)$$

Отже, фрагмент функціональної логіки відновлення бітів  $B_2^+ B_1^+ B_0^+$  набуде вигляду (рис. 7):

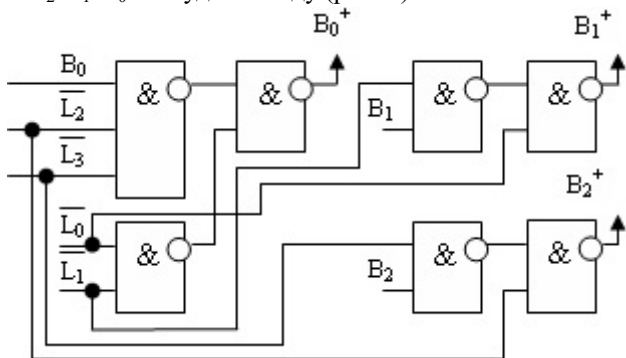


Рис. 7. Логіка відновлення бітів  $B_2^+ B_1^+ B_0^+$

Поєднуючи отримані фрагменти функціональних логік обрання певного рядка  $L_i$  та відновлення бітів  $B_2^+ B_1^+ B_0^+$  в єдиний синхронний цифровий автомат, де в якості синхронних елементів пам'яті для бітів  $B$  пропонуються синхронні D-триггери, а в якості синхросигналу – сигнал запису/читання  $W/R$ , отримаємо схематехнічне рішення блоку LRU внутрішньої кеш-пам'яті (рис. 8):

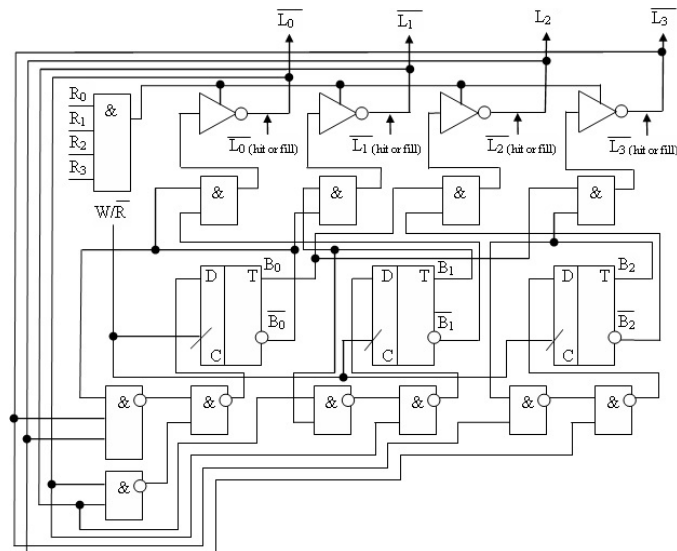


Рис. 8. Схематехнічне рішення блоку LRU

Схематехнічне рішення блоку LRU функціонує наступним чином: процесор, виконуючи свої цикли, в першу чергу адресно звертається до внутрішньої кеш-пам'яті по інструкції та операнди у вигляді послідовних 16 байтів. Фіксуєчи подію промаху у внутрішній кеш-пам'яті, процесор ініціює свій шинний інтерфейс на цикл звертання до сегменту або сторінки коду з метою розташування в адресованому рядку певної множини чергової 16 байтної порції, але наштовхується на факт повної достовірності рядків обраної множини. Процесор вирішує цю проблему, активуючи блок LRU. Активна функціональна логіка блоку LRU, покладаючись на алгоритм псевдо-LRU заміщення рядків, почне приймати рішення про запис у найменш нещодавно використаний рядок множини і першим на черзі стане рядок  $L_0$ , про що свідчитиме вміст бітів  $B_2 B_1 B_0 = 000$ , оскільки останнім рішенням функціональної логіки блоку достовірності було обрання рядка  $L_3$ . Наступний промах процесора при звертанні до обраної множини блоку даних внутрішньої кеш-пам'яті спричинить рішення функціональної логіки блоку LRU про обрання для запису у рядок  $L_2$ , про що свідчитиме вміст бітів  $B_2 B_1 B_0 = 011$ , який утвориться на підставі звертання попередньо до рядка  $L_0$  та попереднього стану бітів  $B_2 B_1 B_0 = 000$ . Далі, у разі промаху, функціональної логіки блоку LRU прийме рішення про обрання для запису рядок  $L_1$ , про що свідчитиме вміст бітів  $B_2 B_1 B_0 = 110$ , який утвориться на підставі звертання попередньо до рядка  $L_2$  та попереднього стану бітів  $B_2 B_1 B_0 = 011$ . Промах наступного циклу звертання процесора до обраної множини блоку даних внутрішньої кеш-пам'яті змусить функціональну логіку блоку LRU прийняти рішення про обрання для запису рядка  $L_3$ , про що свідчитиме вміст бітів  $B_2 B_1 B_0 = 101$ , який утвориться на підставі звертання попередньо до рядка  $L_1$  та попереднього стану бітів

$B_2B_1B_0=110$ . Ймовірна асинхронна активність сигналу FLUSH# знову перетворить вміст блоку даних внутрішньої кеш-пам'яті на недостовірний і цикли заповнення та заміщення рядків знову повторяться. Взагалі існує три варіанти очистки внутрішньої кеш-пам'яті, що перетворюють достовірний вміст певного рядка відповідної множини блоку даних в недостовірний:

- вплив вхідного сигналу очистки внутрішньої кеш-пам'яті FLUSH#;
- цикл очистки кеш-пам'яті;
- цикл зворотного запису та очистки кеш-пам'яті;

Слід додати, що цикл очистки кеш-пам'яті виконує дві дії: перетворює увесь вміст внутрішньої кеш-пам'яті в недостовірний і запитує зовнішню кеш-пам'ять оголосити весь її вміст недостовірним. При чому зовнішня кеш-пам'ять не повинна записувати її вміст в основну пам'ять перед очисткою. Цей цикл ініціює команда INVD (INValidDate cache). Цикл зворотного запису та очистки кеш-пам'яті виконує три дії: перетворює увесь вміст внутрішньої кеш-пам'яті в недостовірний, запитує зовнішню кеш-пам'ять записати весь достовірний її вміст в основну пам'ять (здійснити зворотний запис або вивантаження) та запитує зовнішню кеш-пам'ять оголосити весь її вміст недостовірним після зворотного запису. Даний ініціює команда WBINVD (Write-Back INValidDate cache).

#### IV. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ БЛОКУ LRU

Остаточна комп'ютерна модель блоку LRU внутрішньої кеш-пам'яті процесорного ядра відтворена на рисунку 9.

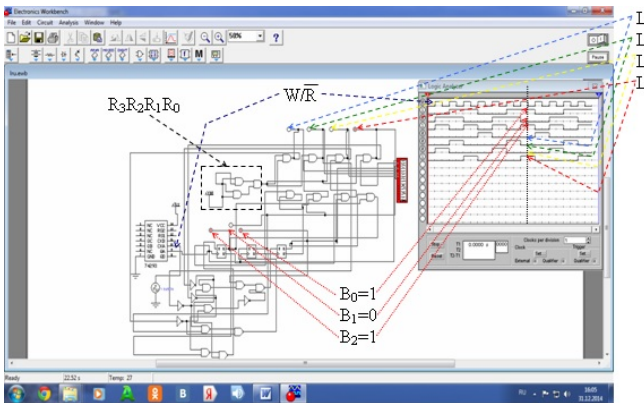


Рис. 9. Комп'ютерна модель блоку LRU

Синтезована модель функціональної логіки блоку LRU внутрішньої кеш-пам'яті процесорного ядра є оригінальним рішенням поставленої задачі. В процесі досліджень отриманої моделі модуля LRU в середовищі комп'ютерного моделювання EWB512 одержані результати у вигляді діаграм часу та значень логічних аналізаторів, де чітко відтворюється певна послідовність

відновлень значень бітів  $B_2B_1B_0$ : 000,011,110,101 (0,3,6,5) при кожній активності сигналу запису W/R, що відповідає влученню або заповненню рядків L0,L2,L1,L3 відповідної множини блоку даних кеш-пам'яті. У разі промаху, з урахуванням вище наведених відновлень значень бітів  $B_2B_1B_0$ , на поточному циклі заповнення рядка за умови достовірності всіх рядків ( $R_3R_2R_1R_0=1111$ ), активний модуль LRU вже має рішення про обрання рядка для заповнення даними внаслідок попередніх влучень та заповнень, а його функціональна логіка наприкінці циклу вже готує певний рядок Li для наступних циклів звертання. Слід зазначити, що внаслідок синтезу функціональної логіки модуля LRU, стани бітів  $B_2B_1B_0$ : 001,010,100,111 (1,2,4,7) математично спростовані апаратом графічного методу мінімізації алгебри логіки, що вдвічі зменшує частоту використання одного й того ж рядка для заповнення.

#### ВИСНОВКИ

Сучасні джерела комп'ютерної літератури досить стримано висвітлюють апаратні рішення компонент процесорних ядер на рівні комп'ютерних схемотехніки та логіки. Цей факт торкається і такої компоненти, як модуль LRU внутрішньої кеш-пам'яті та асоціативного кеш-буфера сторінкового перетворення. Як правило в літературі ця компонента процесорного ядра описується на рівні архітектури.

Ця обставина надихнула автора, використовуючи математичний апарат комп'ютерної логіки, синтезувати модуль достовірності, блок LRU з алгоритмом pseudo-LRU та на підставі отриманих мінімальних логічних рівнянь отримати певне схемотехнічне рішення.

За схемотехнічним рішенням в середовищі комп'ютерного моделювання EWB512 була побудована комп'ютерна модель та досліджена комп'ютерна логіка її функціонування.

Представлена та досліджена комп'ютерна модель блоку LRU складає альтернативу існуючим блокам LRU реальних процесорів архітектури IA-32 та створює конкретну схемотехнічну уяву про досить абстрактну архітектуру пристрою.

#### ЛІТЕРАТУРА REFERENCES

- [1] В.Л. Григорьев Микропроцессор i486. Архитектура и программирование. Книга 2,3,4. Аппаратная архитектура. – М., ГРАНАЛ, 1993. – с. 111, ил. 54
- [2] Б.Э. Смит, М.Т. Джонсон Архитектура и программирование микропроцессора INTEL 80386 – М.:ТОО «Конкорд», 1992. – 334 с.
- [3] Брей Б. Микропроцессоры Intel: 8086/8088, 80186/80188, 80286, 80386, i486, Pentium, Pentium Pro, Pentium 2, Pentium 3, Pentium 4. Архитектура, программирование и интерфейсы. Шестое издание: Пер. с англ. - СПб.: БХВ-Петербург, 2005. – 1328 с.: ил.
- [4] Intel® 64 and IA-32 Architectures Software Developer's Manual <http://www.intel.com>
- [5] Гук М. Процессоры P-II, P-Pro и просто Pentium. С-Петербург.: Питер Ком, 1999.- 288 с.
- [6] Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron.- СПб.: Питер, 2001.- 512 с.