Two complementary notes on skewed-associative caches

André Seznec

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche
1992
25ème anniversaire
N° 1737

Programme 1
Architectures parallèles, Bases de données, Réseaux et Systèmes distribués

TWO COMPLEMENTARY NOTES ON SKEWED-ASSOCIATIVE CACHES

André SEZNEC

Juillet 1992

RR-1737

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTEMES ALEATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CEDEX FRANCE
Tél. : 99 84 71 00 - Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Deux notes complémentaires sur les antémoires associatives brouillées
Two complementary notes on skewed-associative caches
Programme 1
Projet CALCPAR

André Seznec

27 juillet 1992

Publication Interne n° 668 - Juillet 1992 - 10 pages

Résumé

L'organisation de l'antémémoire associative brouillée a été définie précédemment dans le rapport IRISA 645.

Nous présentons ici deux notes complémentaires sur la mise en œuvre d'une antémémoire associative brouillée.

Abstract

The organization of the skewed-associative cache has been presented in the IRISA report 645.

We present here two complementary notes on the implementation of a skewed-associative cache.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (U.R.A. 227) UNIVERSITE DE RENNES I INSA DE RENNES
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (UNITE DE RECHERCHE DE RENNES)

A family of skewing functions for two-way skewed-associative caches

André Seznec
IRISA, Campus de Beaulieu
35042 Rennes Cedex
FRANCE
tel : (33) 99 84 73 36
FAX : (33) 99 38 38 32
e-mail : seznec@irisa.irisa.fr

June 30, 1992

In this short note, we focuss on defining skewing functions for two-way skewed associative caches [1]. Our basic goals are here to obtain a minimum hardware implementation cost and a minimum extra delay on cache hit time.

An example

Let us consider the particular case of 64 bytes lines and 4096 bytes cache banks. An organization of a two-way skewed associative cache is illustrated in fig 3.

The skewing functions used in this example clearly verify the criterions for "good" skewing functions cited in [1].

On this example, only three XOR gates are added to a classical cache bank. It appears in the proposed design that the access time to the cache bank shall not be lenghtened a lot by the skewing functions: at most the delay for crossing a XOR gate.

We think that the access time may even be exactly the same as in a classical two-way set-associative cache:

1. In a microprocessor, when using a one-cycle access cache, the access cache time generally determines the machine cycle. The address computation is performed in a less critical stage: the XOR gates may be added at the end of the stage.

2. When using a pipelined cache, row selection may be done on the second cycle.

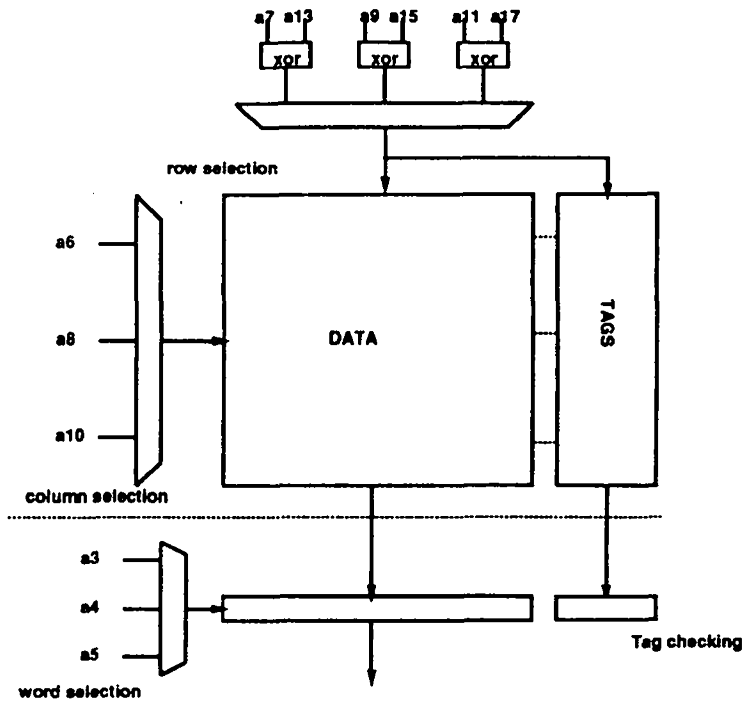

Figure 1: Cache bank 0

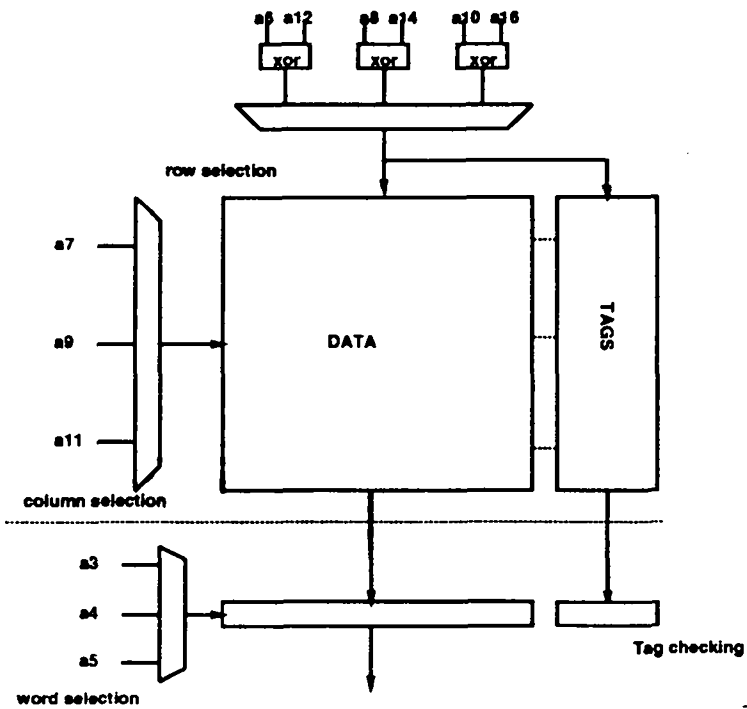

Figure 2: Cache bank 1

Figure 3: A two bank skewed-associative cache

A formal description of the family of skewing functions

Let 2^c be the size of the line.

Let 2^n be the number of cache lines in a cache bank.

Let us consider the decomposition of a binary representation of an address A in bit substrings A = (A_3 ,A_2, A_1, A_0), A_0 is a c bits string: the displacement in the line. A_1 and A_2 are two n bits strings and A_3 is the string of the $q-(2*n+c)$ most significant bits.

Let us consider T an integer such that $0 \leq T < 2^n$ and $\bar{T}$ its binary opposite,($\bar{T} = 2^n - 1 - T$)..

Let ϕ be a Bit Permute Permutation on the set $\{0,..,2^n-1\}$ (e.g. Identity, Perferct Shuffle, Bit Reversal).

We consider the mapping functions defined respectively by[1]:

$$\begin{array}{llcl} F_0^{T,\phi}: & S & \longrightarrow & \{0,..,2^n-1\} \\ & (A_3, A_2, A_1, A_0) & \longrightarrow & A_1) \oplus (\phi(A_2) \bullet T) \end{array}$$

$$\begin{array}{llcl} F_1^{T,\phi}: & S & \longrightarrow & \{0,..,2^n-1\} \\ & (A_3, A_2, A_1, A_0) & \longrightarrow & A_1 \oplus (\phi(A_2) \bullet \bar{T}) \end{array}$$

$\oplus$ is the exclusive OR and $\bullet$ is the bitwise product.

These functions satisfy the criterions for "good" skewing functions defined in the paper (Equitability, inter-bank dispersion and local dispersion).

Each bit of the $F_0^{T,\phi}(A)$ or $F_1^{T,\phi}(A)$ is either directly a bit of the binary decomposition of address A or the XOR of two bits of this binary decomposition.

T may be chosen in order to allow symmetric design of the two cache banks: when n is even, having the same number of bits equal to one and zero seems an interesting approach.

In the previous example in figure 3, T= 44 (binary decomposition 101010) and the Bit Permute Permutation is the identity.

First simulations results show that when using these particular skewing functions, the two-way skewed-associative caches exhibit approximatively the same behavior as when using the skewing functions described in [1].

References

[1] A.Seznec, F. Bodin "Skewed Associative Caches", IRISA report 645, March 1992

[1]a line in main memory is represented by the address of its first byte

Virtual address translation and skewed-associative caches

André Seznec
IRISA, Campus de Beaulieu
35042 Rennes Cedex
FRANCE
tel : (33) 99 84 73 36
FAX : (33) 99 38 38 32
e-mail : seznec@irisa.irisa.fr

July 2, 1992

Introduction

In RISC and superscalar microprocessors, cache access is often the critical path in the pipeline. In this short note, we present a page mapping in physical memory which allow to execute translation from virtual to physical addres in parallel with cache access, where using skewed-associative caches [1].

Virtual caching *or* physical caching

Two types of addressing may be used for caches in microprocessors:

1. Virtual addressing: the virtual address is used for indexing the cache. Tags allowing to reconstruct the whole virtual address of a word stored in the cache are stored with the cache line. Virtual address is checked against these *virtual tags.*

2. Physical address: the physical address is used for indexing cache. Tags allowing to reconstruct the physical address of a word stored in the cache. Physical address is checked against these *physical tags.*

   Notice that in this case the physical address is needed for this checking: translation from virtual address to physical address must be done "before" the checking.

Physical addressing is desirable

When sharing pages between processes, physical addressing of the cache allows to avoid multiple copies of the same line of data in the cache and then the data in the cache always remain coherent.

Parallel address translation and cache access

From now, we consider only physical caching.

When the whole physical address is needed for indexing the cache, the address translation and the cache access must be sequentialized; for example, these two steps may be executed in two consecutive pipeline stages, this solution was adopted in the MIPS R3000.

But this sequentialization introduced some performance loss by delaying the availability of data and instructions.

Set-associative caches: page size $\geq$ cache bank size

To avoid this sequentiality, an artifice is used in many microprocessor designs:

> When the minimum size of a page is 2^c bytes, the address translation do not affect the c lowest significant bits.
>
> When the size of a cache bank is smaller than or equal to 2^q bytes, the q lowest significant bits are used for indexing the caches: address translation and cache access may begin at the same time.

It has to be pointed out that such an approach leads to increase the minimum page size when the size of cache grows. This will soon become unacceptable: using pages of size 32Kbytes in place of 4Kbytes pages leads to a significant increase of the working set size of an application needed in memory [2] (sometimes 60%).

Case of skewed-associative caches

Unfortunately an adaptation of the previous artifice can not be used directly when using skewed-associative caches, it would lead to unrealistic minimum size of the page.

Nevertheless, we propose skewing on virtual addresses:

> When the size of the line is 2^c bytes, and the number of lines per cache bank is 2^n, the skewing functions proposed in [1, 4] use the $c + 2n$ lowest significant bits of the address.
>
> We may impose that address translation does not affect these $c + 2n$ bits:
>
> Let us assume for example that the page size is $2n + c$ bytes, virtual page beginning at address $X3 * 2^{2n+c} + X2 * 2^{n+c}$ can only be mapped on a physical page beginning at $Y3 * 2^{2n+c} + X2 * 2^{n+c}$.

This limitation is not a major restriction, as shown below:

> Let us consider the case of a cache bank of 4096 bytes and a cache line of 64 bytes:
>
> in order to allow parallel address translation and cache access in a skewed-associative caches, we impose the virtual address and the physical address to correspond on the 18 lowest bits (i.e modulo 256K).
>
> If we consider a 8 Megabytes physical memory, then each virtual page of 4K bytes may be mapped in anyone of the physical pages in a set of 32 physical pages:
>
> The physical memory may be considered as a 32-way set-associative cache of 2048 lines of 4096 bytes. It is well known that the behavior of such a cache is approximately the same as a the behavior of a fully-associative cache!

Physical page allocations procedure will have to be rewritten with this particular constraint.

Notice that a similar solution will certainly also be adopted in near future in microprocessors using classical set-associative caches or direct-mapped caches in order to keep a reasonable minimum page size and has already be envisaged [3].

References

[1] A. Seznec, F. Bodin "Skewed Associative Caches", IRISA report 645.

[2] M. Talluri, S. Kong, M.D. Hill, D.A. Patterson "Tradeoffs in Supporting Two Page Sizes" Proceedings of the 19^{th} International Symposium on Computer Architecture, May 1992

[3] B.K. Bray, W.L. Lynch, M.J. Flynn "Page allocation to reduce access time of physical caches" Stanford University, Technical Report No CSL-TR-90-454 Nov. 1990

[4] A. Seznec "A family of skewing functions for two-way skewed-associative caches"

LISTE DES DERNIERES PUBLICATIONS INTERNES PARUES A L'IRISA

PI 661 REACHABILITY ANALYSIS ON DISTRIBUTED EXECUTIONS
Claire DIEHL, Claude JARD, Jean-Xavier RAMPON
Juin 1992, 18 pages.

PI 662 RECONSTRUCTION 3D DE PRIMITIVES GEOMETRIQUES PAR VISION ACTIVE
Samia BOUKIR, François CHAUMETTE
Juin 1992, 40 pages.

PI 663 FILTRES SEMANTIQUES EN CALCUL PROPOSITIONNEL
Raymond ROLLAND
Juin 1992, 22 pages.

PI 664 REGION-BASED TRACKING IN AN IMAGE SEQUENCE
François MEYER, Patrick BOUTHEMY
Juin 1992, 50 pages.

PI 665 CORRECTNESS OF AUTOMATED DISTRIBUTION OF SEQUENTIAL PROGRAMS
Cyrille BARREAU, Benoît CAILLAUD, Claude JARD, René THORAVAL
Juin 1992, 32 pages.

PI 666 AGREGATION FAIBLE DES PROCESSUS DE MARKOV ABSORBANTS
James LEDOUX, Gerardo RUBINO, Bruno SERICOLA
Juillet 1992, 30 pages.

PI 667 MODELES D'EVALUATION DE LA FIABILITE DU LOGICIEL ET TECHNIQUES DE VALIDATION DE SYSTEMES DE PREDICTION : ETUDE BIBLIOGRAPHIQUE
James LEDOUX
Juillet 1992, 76 pages.

PI 668 TWO COMPLEMENTARY NOTES ON SKEWED-ASSOCIATIVE CACHES
André SEZNEC
Juillet 1992, 10 pages.

PI 669 PARALLELISATION D'UN ALGORITHME DE DETECTION DE MOUVEMENT SUR UNE ARCHITECTURE MIMD
Fabrice HEITZ, Sergui JUFRESA, Etienne MEMIN, Thierry PRIOL
Juillet 1992, 34 pages.

ISSN 0249-6399