ELSEVIER

INTEGRATION, the VLSI journal 22 (1997) 165–171

INTEGRATION
the VLSI journal

A note on architectures for large-capacity CAMs

Behrooz Parhami*

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA

Received April 1 1996

Abstract

Recently, Schultz and Gulak have presented a taxonomy for content-addressable memories, discussed the relative advantages of the various serial/parallel architectures, and listed variations and options for implementing large systems. However, they do not cite the origins of this taxonomy that dates back to 1973, leaving the reader with the impression that it is new and being proposed for the first time. They also omit other relevant references to methods for implementing systolic and large-scale associative systems.

Keywords: Associative memory; Content-addressable memory; Classification; Serial/parallel implementations; Taxonomy for CAMs; VLSI architectures

1. Introduction

In [1], Schultz and Gulak present a taxonomy for content-addressable memories (CAMs). The taxonomy begins by dividing CAMs into four classes of fully parallel (word-parallel, bit-parallel), (word-parallel) bit-serial, word-serial (bit-parallel), and word-serial bit-serial. The fully parallel class is then divided into four subclasses reflecting, in effect, the extent of hardware sharing within the CAM array. This categorization involves two parameters: m, the number of CAM words sharing a match line within the memory matrix ($m \neq 1$ requires "post-encoding" of match results); Ω, the number of classes or bit-slices sharing a comparator ($\Omega \neq 1$ implies "pre-classification" to limit the search to a few sequentially examined classes).

In this note, we contend that the above-mentioned taxonomy is not new. Its primary structure was presented as early as 1973 by this author and was subsequently referenced and used by many authors and researchers. Another shortcoming is that many serial/parallel designs are misleadingly categorized as fully parallel systems. We also discuss several important concepts for the design of large CAMs that were omitted in Schultz and Gulak's treatment.

*Tel.: 805 893 3211; fax.: 805 893 3262, e-mail: parhami@ece.ucsb.edu.

PII S0167-9260(97)00010-2

2. On the Taxonomy and History

The main part of the taxonomy presented in [1] was first introduced by this author in a widely known survey paper on associative memories (AMs) and processors (APs) [2]. Following its publication in 1973, this taxonomy was used/cited in various sources, most notably books by Foster [3], Kohonen [4] (his Ref. [1.5]), and Su [5] as well as survey/review papers by Yau and Fung [6] (their Ref. [3]), Chisvin and Duckworth [7] (their Ref. [7]), and most recently Krikelis and Weems [8]. It is therefore quite surprising that the authors of [1] fail to identify the origins of this taxonomy and claim to have presented "the first such taxonomy in the literature".

The 1973 origin of the taxonomy may be somewhat unclear from some of the secondary sources listed above due to lack of immediate and proper citation, but each of the three survey/review papers [6–8] actually uses the top level of this taxonomy in a manner that is quite hard to miss. The claim of novelty is even more perplexing given the fact that the authors of [1] actually cite [4, 7] as their first two references.

A related, but less serious problem is that the authors also misrepresent the history of research on CAMs. Citing the 1966 survey by Hanlon [9] (their Ref. [3]), the authors state that "[CAMs/AMs] have been studied for over 30 years". This is at best misleading, though technically correct because of the word "over". Thirty years ago, AMs/APs had already passed the mega-bit-ops (bit operations per second) performance milestone and were well on their way to be first in giga-bit-ops performance. There are reasons to believe that they will again be first in reaching the peta-bit-ops milestone in high-performance computing (see Table 1).

As noted in [2] and many other sources, AMs/CAMs came into the forefront at least 40 years ago with the publication of a seminal paper by Slade and McMahon [10]. The origins of research on AM/AP technology and applications actually go back to the 1943 sketch of a relay-based AM by Konrad Zuse [11] and the 1945 visionary assessment of the need for associative access to information by Vannevar Bush [12], making the field 50+ years old.

3. Systolic associative memories

Arguments for the infeasibility of large AMs/APs based on the broadcasting of instructions and reduction by wired logic (wired-AND or wired-OR implied in a shared match or mismatch line)

Table 1
Entering the second-half century of associative processing [13]

Decade	Events & advances	Technology	Performance	Key ref.'s
1940s	Formulation of need & concept	Relays		[11], [12]
1950s	Emergence of cell technologies	Magn., Cryo.	Mega-bit-ops	[10]
1960s	Introduction of basic architectures	Transistors		[9], [14]
1970s	Commercialization & applications	ICs	Giga-bit-ops	[2], [6]
1980s	Focus on system/software issues	VLSI	Tera-bit-ops	[4], [7]
1990s	Scalable & flexible architectures	ULSI, WSI	Peta-bit-ops?	[8], [15]

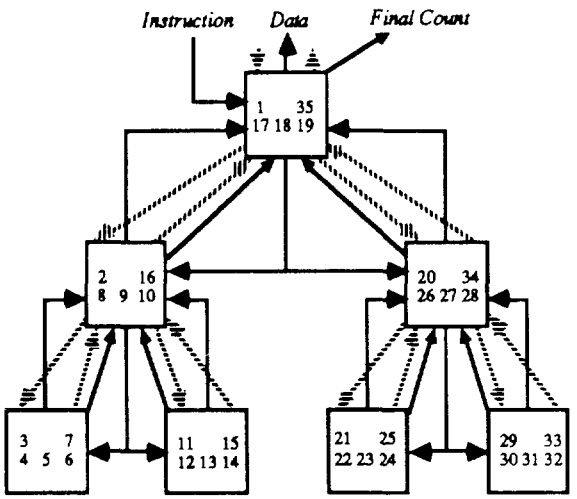

Fig. 1. Systolic AM built as a tree of simple building-block AMs.

were first presented by this author in 1990 [16–20]. It was then suggested that one needs to embody the systolic design paradigm [21] in the implementation of large AMs from smaller building-block subsystems, interconnected by low-fan-out, short local wires.

The systolic AM concept is needed since both broadcasting and global operations require time that increases substantially as we go from single-chip to multi-chip and multi-board systems and also because at extremely high speeds and integration densities, wire delays become dominant, thus invalidating many speedup claims based on the assumption of a fixed AM cycle time (Lipovski [22] uses the term "systolic AM" in a different sense to refer to his modified DRAM chip that communicates in a linear array and has simple search logic in its refresh path).

This author has proposed several architectures for building-block AMs and their incorporation into larger systems [16–20]. The approach is exemplified by Fig. 1 showing a tree-structured AM built from simple AM building blocks (BB-SAMs). The cells can also be arranged into linear or higher-dimensional arrays. In the binary-tree version, instructions are input at the root and move downward to the leaves of the tree, where they rebound and move back up towards the root, carrying with them various indicators and partial computation results. The BB-SAMs can operate asynchronously, but we assume lock-step operation here for ease of exposition.

All BB-SAMs located at the same tree level receive the common instruction simultaneously. Searching, reading, and writing is done independently within the BB-SAMs according to the SIMD mode of parallel computation [23]. The count of responders, if needed by the instruction, is produced by participating cells and is combined during the instruction's upward movement towards the root. The specific numbering scheme for the AM words/cells within each BB-SAM (5 in this pedagogical example) is devised to allow left and right circular shifts of the response tags which is a commonly used inter-word communication mechanism in many AMs.

4. Serial/parallel architectures

The need for including the post-encoding and pre-classification features in the taxonomy proposed in [1] stems in part from the recognition that any practically realizable large AM must have a serial/parallel, rather than a fully parallel, implementation. Post-encoding, or sharing of match lines among multiple words, necessitates sequential processing among those words. Similarly, incorporation of pre-classification gives rise to a sequentialization of searches that span multiple classes. The taxonomy proposed in [1] obscures this fact by including such mixed serial/parallel designs in the category of "fully parallel" systems.

In retrospect, it makes much more sense to envisage three categories along each of the bit and word dimensions corresponding to serial, partially parallel, and parallel processing. In fact, one can take this a step further and define the architectural classes as (b, w), meaning that b bits in each of w words are processed in parallel. The top 4 classes in the classification of [1] would then correspond to (all, all), (1, all), (all, 1), and (1, 1), with the other categories, currently included under (all, all), corresponding to b and/or w assuming intermediate values; e.g., (all/Ω, all/m) when both post-encoding and pre-classification are used.

One of the first attempts to build large CAMs in serial/parallel form was that of RAPID [24] which was based on circulating the contents of a head-per-track disk memory through a number of associative processing cells. This system was highly influential and has been described in [4, 6, 25] among other sources. With modern VLSI technology, the fully electronic version of such a design can be built in a cost-effective manner [26]. The resulting serial/parallel design offers interesting tradeoffs, with choices ranging from the low-cost version composed of a small number of shift registers to a massively parallel high-performance design.

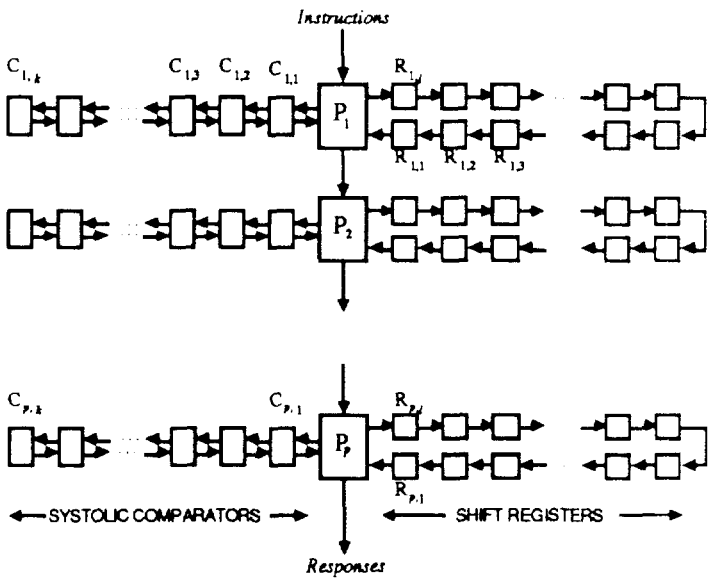

Fig. 2. Serial/parallel AM with high-speed shift registers and systolic comparators.

Another dimension of the tradeoff involves using complex processors composed of high-speed systolic comparators to allow faster shifting of data (and thus making the use of fewer, longer registers possible) versus a large number of slower processors with conventional comparators. Fig. 2 shows the block diagram of a shift-register-based serial/parallel architecture using systolic comparators [26]. Clearly, these design techniques and the attendant tradeoffs are relevant to a discussion of architectures for large AMs/CAMs.

In closing, we note that in algorithms such as maximum or minimum finding, even fully parallel AMs are often used in bit-serial fashion. Recently, strategies and algorithms for better utilizing the parallel match capability of such systems have been developed and analyzed [27, 28].

5. Conclusion

Within the field of computer science and engineering, it is quite easy to overlook relevant references in preparing an overview or survey paper and to claim novelty for ideas that have in fact been published previously, given the immense volume of technical literature. However, in the area of associative processing, researchers have been quite diligent in properly attributing ideas to their originators and in publishing surveys that summarize the state of the associative computing technology and applications every few years. It is only fair to expect authors of new overviews or surveys to study all previously published reviews in order to familiarize themselves with the progress of the field and to give proper credit where due.

6. Note added in press

The authors' reply to this note focuses on three main points. First, the authors state that their paper was intended as "an overview of how one would build a CAM on a VLSI chip today". This limited scope should have been explicated in the title, abstract, and in the opening paragraph of the introduction. Since *INTEGRATION* also covers VLSI systems engineering, algorithms, and theories, publication in this journal does not automatically mean that the paper covers only existing implementations. Furthermore, large CAMs, like large RAMs, may have to be built from multiple chips, no matter how large individual chips become. Second, the authors attribute my characterization of their "fully parallel" architectures as including serial-parallel designs to a misunderstanding of post-encoding and pre-classification. A post-encoded CAM is "fully parallel" only in the sense that a ripple-carry adder is fully parallel: While individual bits are added concurrently by dedicated hardware cells, combining and propagating the carries by constant-fan-in logic elements implies some degree of serialization or propagation for non-trivial word lengths. Similarly, requiring that the search be limited to a subarray through pre-classification of data leads to serialization when pre-classification is impractical (for searches based on multiple attributes, e.g.). Carrying the authors' reasoning to an extreme would lead to a RAM being classified as a fully parallel CAM with single-word classes. Third, the authors judge my "argument for inclusion of systolic methods to be technically unconvincing". The discrepancy between processor and memory cycle times has already motivated systolic RAM designs [29]. It is only a matter of time before CAM designers will be forced to do the same. This may not apply to current working chips, but is

definitely a valid point in a discussion of future large-capacity CAMs, as even on-chip signal propagation delays exceed switching delays. In conclusion, it is noteworthy that products such as processor-in-memory chips [30] have all but eradicated the distinction between associative memories and associative processors.

References

[1] K.J. Schultz, P.G. Gulak, Architectures for large-capacity CAMs, *Integration the VLSI J.* 18 (2/3) (1995) 151–171.
[2] B. Parhami, Associative memories and processors: an overview and selected bibliography, *Proc. IEEE* 61 (6) (1973) 722–730.
[3] C.C. Foster, Content Addressable Parallel Processors, Van Nostrand Reinhold, New York, 1976.
[4] T. Kohonen, Content-Addressable Memories, 2nd ed., Springer, Berlin, 1987, Section 1.1.2, pp. 3–6, Section 3.6.3. pp. 166–172.
[5] S. Y. W. Su, Database Computers: Principles, Architectures & Techniques, McGraw-Hill, New York, 1988, Section 5.1, pp. 181–184.
[6] S.S. Yau, H.S. Fung, Associative processor architecture – a survey, *Comput. Surv.* 9 (1) (1977) 3–27.
[7] L. Chisvin, R.J. Duckworth, Content-addressable and associative memory: alternatives to the ubiquitous RAM, *Computer* 22 (7) (1989) 51–64.
[8] A. Krikelis, C.C. Weems, Associative processing and processors (Introduction to a special issue on this topic), *Computer* 27 (11) (1994) 12–17.
[9] A.G. Hanlon, Content-addressable and associative memory systems: a survey, *IEEE Trans. Electron. Comput. EC*-15, (1966) 509–521.
[10] A. Slade, H.O. McMahon, A cryotron catalog memory system, *Proc. Eastern Joint Computer Conf.*, 1956, pp. 115–120.
[11] K. Zuse, Der Computer-Mein Lebenswerk, Springer, Berlin, 1986 (Diagram on p. 77 shows an associative relay circuit sketched by Zuse in 1943).
[12] V. Bush, As we may think, Atlantic Monthly 176 (1945). Reprinted in *Comput. Bull.* 4 Part (1) (1988) 35–40.
[13] B. Parhami, Search and data selection algorithms for associative processors, in: Associative Processing and Processors, in: A. Krikelis, C. Weems (Eds.) *IEEE Computer Society Press*, (1997) in press.
[14] R.H. Fuller, Associative parallel processing, *Proc. Spring Joint Computer Conf.*, 1967, pp. 471–475. See also Fuller's Ph.D. Dissertation, EE Dept., Univ. of California, L.A. 1960.
[15] K.E. Grosspietsch, (Ed), Special issue on associative processors and memories (in 2 parts), *IEEE Micro* 12 (6 & 12) (1992).
[16] B. Parhami, Associative memory designs for VLSI implementation, *Proc. Internat. Conf. on Databases, Parallel Architectures, and Their Applications*, Miami, FL, March 1990 pp. 359–366.
[17] B. Parhami, Massively parallel search processors: history and modern trends, *Proc. 4th Parallel Processing Symp.*, Fullerton, CA, April 1990, pp. 91–104.
[18] B. Parhami, Systolic associative memories, *Proc. Internat. Conf. on Parallel Processing.* vol. I, St. Charles, IL, August 1990 pp. 545–548.
[19] B. Parhami, Scalable architectures for VLSI-based associative memories, in Parallel Architectures, N. Rishe, S. Navathe, D. Tal (Eds.) *IEEE Computer Society Press*, Silver Spring, MD, 1991, pp. 181–200.
[20] B. Parhami, Architectural tradeoffs in the design of VLSI-based associative memories, *Microprocessing and Microprogramming* 36 (1992) 27–41.
[21] H.T. Kung, Why systolic architectures, *Computer* 15 (1) (1982) 37–46.
[22] G.J. Lipovski, A four megabit dynamic systolic associative memory chip, *J. VLSI Signal Processing* 4 (1) (1992) 37–51.
[23] B. Parhami, Panel assesses SIMD's future, Computer 28 (6) (1995) 89–91. The full report on which this article is based appears in IEEE Computer Society Technical Committee on Computer Architecture Newsletter, August 1995, pp. 23–26, and *ACM Computer Architecture News* 23 (4) (1995) 19–22.

[24] B. Parhami, A highly parallel computing system for information retrieval, *AFIPs Conf. Proc.*, vol. 41 (1972 Fall Joint Computer Conf.), AFIPS Press, Montvale, NJ, pp. 681–690.
[25] E. Ozkarahan, Database Machines and Database Management, Prentice-Hall, Englewood Cliffs, NJ, 1986, Section 6.5.4, pp. 220–221.
[26] B. Parhami, The mixed serial/parallel approach to VLSI search processor, *Proc. Hawaii Internat. Conf. on System Sciences (Minitrack on Advances in the Architecture of Associative Processors)*, vol. I, Koloa, Hawaii, January 1991, pp. 202–211.
[27] B. Parhami, Performance analysis and optimization of search and selection algorithms for highly parallel associative memories, *Proc. Internat. Conf. on Modeling, Analysis, & Simulation of Computer & Telecommunication Systems*, San Jose, Cal, February 1996, pp. 217–221.
[28] B. Parhami, Extreme-value search and general selection algorithms for fully parallel associative memories, *The Computer J.*, 39 (3) (1996) 241–250.
[29] C.J. Nicol, A.G. Dickinson, A scalable pipelined architecture for fast buffer SRAMs, *IEEE J. Solid-State Circuits* 31 (3) (1996) 419–429.
[30] M. Gokhale, B. Holmes, K. Iobst, Processing in memory: The Terasys massively parallel PIM array, *Computer* 28 (4) (1995) 23–31.