A Banked Precomputation-Based CAM Architecture for Low-Power Storage-Demanding Applications

Pedro Echeverría, José L. Ayala, Marisa López-Vallejo
Departamento de Ingeniería Electrónica
Universidad Politécnica de Madrid (Spain)
Email: {petxebe,jayala,marisa}@die.upm.es

***Abstract*— The content-based access of CAMs makes them of great interest in look-up based operations. However, the large amounts of parallel comparisons required cause an expensive cost in power dissipation. In this work we present a novel banked pre-computation based architecture for low-power and storage-demanding applications. Experimental results show that the proposed banked architecture reduces up to a 76% of power consumption during the search process and decreases the active area in a 10% while performance is also improved by a 25%.**

I. Introduction

A broad range of modern applications demands large storage devices with fast response. Content-Addressable Memories (CAMs) have emerged as one of the favorite devices for such applications [1]–[4]. In CAMs data are accessed based on their content, rather than their physical address. This functionality has shown to be specially efficient in lookup-based applications like TLBs [1] and parameterized mathematical functions in complex algorithms such as neural networks [2], associative computing and data compression [3]. Network search engines [4] and high-speed networks such as gigabit Ethernet and ATM switches also benefit from this particular structure [5].

However, CAMs pay a high hardware price for this content-based access because the memory cell must include comparison circuitry, negatively impacting the size/speed trade-off and complexity of the implementation. Usually, a 9-transistors cell is required instead of the 6-transistors cell used in SRAM. Moreover, the large amounts of parallel comparisons performed in conventional CAM make the device to consume too much power, preventing the implementation of large scale CAMs in a single chip as the leading-edge applications demand.

In this paper we present a novel implementation of a CAM with low-power constraints. The proposed architecture is highly scalable and provides high-performance functioning at large sizes. Therefore, this architecture overcomes previous limitations of the CAM implementation and makes it suitable to all the applications where a high-performance low-power data search functioning is needed.

Previous work on CAM design has focused only on either reducing the power consumption of the match line [6] or enhancing the search speed [7]. Nevertheless, both goals could not have been easily combined in previous approaches.

Although many approaches addressing power dissipation have been reported [8]–[10], resulting circuit techniques have either substantial area overhead, deficiencies in noise inmunity, or cannot be easily scaled without a negative impact on performance. Our work overcomes these limitations by a novel and effective design of the CAM architecture.

The work presented here shares some common ideas with the approach presented by Li et al. in [11], and recently extended by Noda [12] and Choi [10]. These recent works provide a low power implementation of the CAM based on the precomputation of an index parameter. Nevertheless, they are constrained to specific small sizes, lack of scalability and present an increased search delay.

The work presented in this paper is also based on a parameter precomputation-based architecture (Pb-CAM from now on); however, we are able to reduce the parameter word's size with respect to [11], decreasing in this way the logic complexity, area and power consumption related to this parameter. Moreover, the energy savings obtained with the proposed banked architecture (up to 21% during the comparison process) improve the previous implementations of similar technologies and also improve the scalability capabilities of architectures like [12] and [10].

The paper is composed as follows: section II describes the baseline architecture, while the proposed approach is carefully presented in section III. Section IV shows the experimental work and, finally, some conclusions are drawn.

II. Low-Power Implementation: Baseline Architecture

In the traditional implementation of the CAM architecture [13], the comparisons performed during the search operations consume most of the total CAM power. Therefore, one of the best approaches to reduce the power consumption is to minimize the number of these comparisons. As Figure 1 shows, our low-power implementation of the CAM introduces a precomputed parameter [11]. During the *writing* phase, the parameter extractor composes the parameter based on the input data. During the data searching operation, the parameter extractor computes the parameter from the input data and this is compared in parallel with all the parameters stored in the memory. Using this first comparison process result, the input data is only compared with those lines that match the input parameter.

In this architecture, an important part of the comparison power is dissipated by the parameter extraction and comparison, therefore its complexity must be minimized. Moreover,

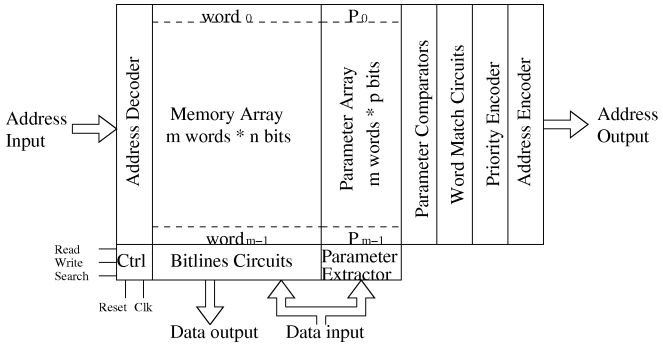

Fig. 1. Parameter precomputation-based architecture.

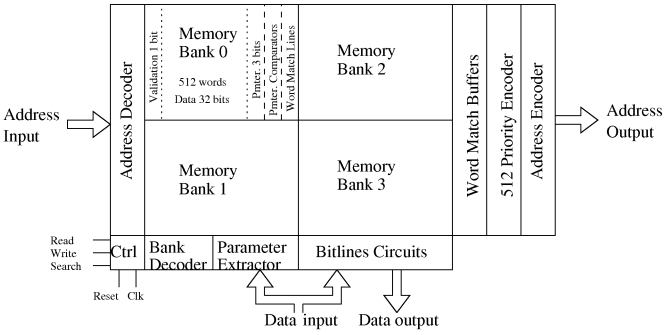

Fig. 2. Proposed banked implementation (2048b x 32b).

the design idea of the parameter extractor is to filter as many unmatched data as possible, with the shortest bit-length and the simplest computation effort. In this approach, a ones count function has been selected and the practical implementation of the parameter extractor is based on a multi-level adder.

Our precomputation-based architecture extends the work presented in [11] and solves its main limitations. This related work shows a Pb-CAM with several advantages like the simplification of the memory cell to a 7-transistor implementation as compared with the 9-transistor one of the traditional architecture (which has a clear impact on the area savings), or the power savings achieved with the parameter precomputation approach. The minimal bit length of the parameter is equal to $\lceil \log_2(n+2) \rceil$ (where n represents the word size), which implies that for long data words a substantial portion of the CAM area is reserved to the parameter. Also, the work we propose here reduces the *pseudo-static* power consumption[1] (the power consumed by the match-lines during the search operation) by using a specific clocked circuit which performs the parameter comparison and manages this situation efficiently.

The design of the CAM proposed in this paper overcomes these limitations and presents an efficient architecture with lower power consumption and increased scalability skills. The proposed architecture is based on the previously referred implementation but includes, among other improvements, a banked implementation of the memory with improved search locality among banks, and a simplified parameter word.

III. Improved Implementation

A. Banked Architecture

The first modification performed in the CAM architecture is to split the storage array into a set of independent banks with equal number of data words per bank (see Figure 2). Once the working bank is selected, the search operation is locally performed in such bank. The selection of the specific bank where the search or write operation occurs is enabled by the least significant bits of the parameter (p_LSBs) and a simple decoding logic. To split the bank architecture into banks implies a negligible extra cost (area and delay) because most of the required work is done by the parameter extractor. Due to the smaller bank size compared with the whole CAM, the energy consumption of the search operation is reduced. Moreover, once the required bank is selected, the bitlines that feed the others can be disabled to avoid any power waste.

Those p_LSBs used in the bank selection do not need to be stored with the data word because all the words stored in the same bank share the p_LSBs. Thus, the number of memory cells used for the parameter is decreased due to the same reason, diminishing in this way the area and complexity of the device. In this way, the energy consumption of the parameter comparison is reduced because so it is the parameter size.

Furthermore, using the LSBs of the parameter instead of the most significant bits (MSBs) to select the bank has two main implications. First, the data search and write operations are homogeneously distributed among all the memory banks (supposing inputs with equal probability) and, in this way, every bank can provide the required data without increasing the probability of a miss. Figure 3 shows how two out of four banks are mostly used when the MSBs are employed to select the working bank, while this usage is equally distributed when the LSBs are used. And secondly, the parameter size can be additionally reduced due to this decoding mechanism, as will be carefully described in the next section.

Finally, the use of a banked architecture allows to decrease the complexity of the logic in the output circuits (priority encoders and match-detection buffers).

There are other memory architectures which propose a banked structure to reduce the power consumption and logic complexity in the hardware implementation. However, these realizations have been mainly proposed for RAMs, while the extensions to CAMs present serious problems in terms of scalability and power consumption.

B. Valid Bit and Parameter's Size Reduction

The work presented by Lin et al. [11] includes the valid bit into the parameter word instead of reserving a specific memory cell for it. This can be done because the precharge of the match-line is controlled by the first comparison stage. Therefore, the size of the parameter word is $\log_2(n+2)$. Our design retrieves the valid bit out of the parameter word (using $\log_2(n+1)$ bits for the parameter word) and assigns a specific memory cell for it.

The idea of retrieving the valid bit out of the parameter word, in combination with a decoding logic, allows us to

[1]Described in detail in [11].

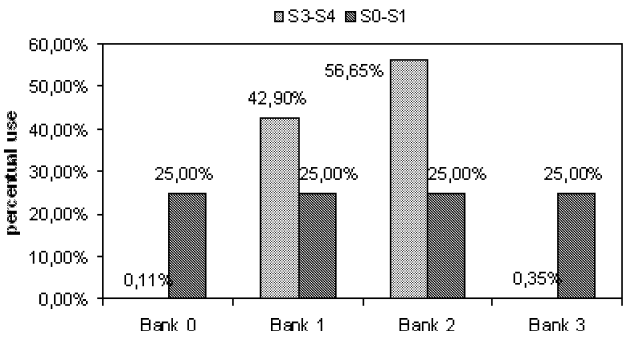

Fig. 3. Percentual use of memory banks (MSBs and LSBs).

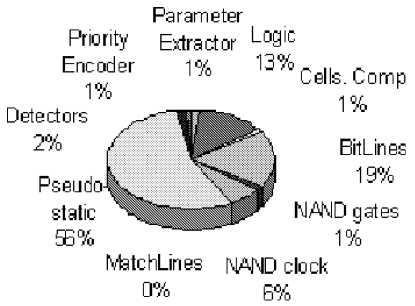

Fig. 4. Distribution of power consumption (banked PB-CAM).

simplify the parameter's size in one more bit. The next example clarifies this improvement.

For a 5-bit parameter, the invalid word in the baseline architecture ([11]) is represented by any parameter value bigger than the word size. The reason is that the ones count can never be bigger than the word size.

In those memory sizes which are an integer exponent of 2 (2, 4, 8, 16, 32...), which correspond to the common implementations, present just one case with the MSB of the parameter equal to '1' (10, 100, 1000, 10000...). In our example (16-bits data word), this is

S_4	S_3	S_2	S_1	S_0
1	0	0	0	0

thus, the next word can be taken as an invalid value,

S_4	S_3	S_2	S_1	S_0
1	1	1	1	1

Our designed decoding logic detects the specific case of 10000 and allows us to detect the specific case of MSB '1' and rest '0' before the comparison is performed. Therefore, the MSB of the parameter (S_4) can be skip and the parameter's size is reduced from 5 to 2 bits (S_3 and S_2) because S_1 and S_0 are used to select the required bank. In this way, now the parameter word size is reduced to $\log_2(n)$.

C. Pseudo-Static Power Consumption

One of the main actors in the power consumption of the Pb-CAM architecture is the pseudo-static power consumption. This is due to the energy wastage in the match-lines during the search operation since these lines remain driven.

A specific parameter comparison circuit has been designed to control this pseudo-static power consumption. Also, the clock distribution to the CAM banks is gated by the decoding circuitry when they are not selected. In this way, the pseudo-static power consumption is reduced to a minimum in the bank where the search operation is performed while it is fully eliminated in the others.

IV. Experimental Results

The experimental work and evaluation of the previous architectures have been carried out with Spice simulations in the Cadence environment. The technology used is .35 μm from Austria MicroSystems.

The proposed architecture has been firstly evaluated in terms of the energy savings obtained. The simulated memory resembles the architecture described in Figure 2 and decreases the energy consumption by a 21% (68.4 fJ/bit in the banked architecture with respect to 86 fJ/bit in the referenced one). This is obtained with the banked implementation of the CAM without gating the pseudo-static power consumption during the search operation. When the circuit that controls the pseudo-static power consumption is used, the power savings reach a 76% (20.68 fJ/bit).

Figure 4 shows the power distribution in the banked implementation of the PB-CAM when the control circuit is also used. As can be seen, most of the power consumption is still due to the pseudo-static power consumption even though the logic is now activated just during the working cycle.

The area improvement achieved with the proposed architecture has also been evaluated. Figure 5 shows the number of transistors in the traditional, Pb-CAM and banked Pb-CAM implementations when the memory size is fixed (2048 positions) and the number of bits per word is ranged. As can be seen, there is a reduced area improvement of the banked implementation with respect to the original Pb-CAM, and almost constant when the number of bits per word is shifted. This difference is due to the savings in the parameter word length and comparison logic. On the other hand, the area savings achieved by both designs of the Pb-CAM with respect to the traditional implementation are quite representative for architectures with more than 16 bits (up to 17% in the range considered).

Figure 6 shows the rapid growth on the number of transistors (area) in the traditional, the Pb-CAM and the banked Pb-CAM implementation when the word length is fixed (32b) and the memory size is ranged. As can be seen, the area of all three implementations grows dramatically with the memory capacity. However, the area savings obtained with the banked Pb-CAM become even more notorious for larger implementations (10% for the 2048b architecture) because the

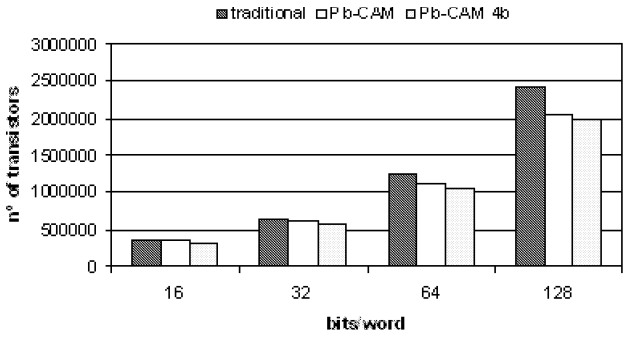

Fig. 5. Number of transistors for a fixed memory capacity (2048b).

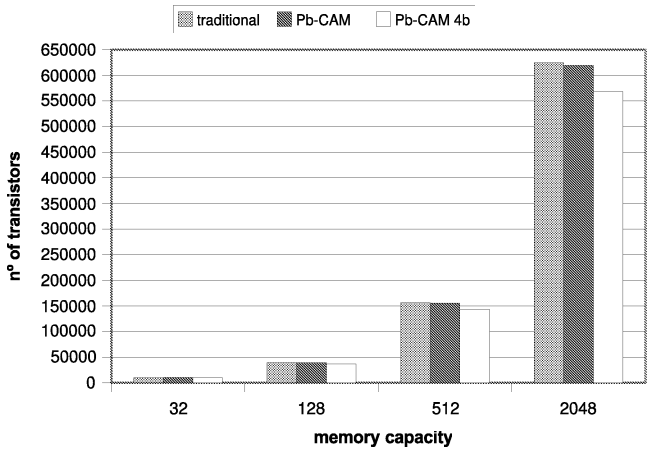

Fig. 6. Number of transistors for a fixed word length (32b).

overhead incurred by the parameter word is lower than in the Pb-CAM. Also, these two implementations of the Pb-CAM reduce the number of active elements (transistors) by employing a 7-transistor memory cell as compared with the 9-transistor one of the traditional architecture.

Finally, the performance of the design has been analyzed to assure the required fast response. Figure 7 shows the electrical simulation of a comparison process, where the Data Search (activation of the comparison process), Data Match (a matching result), Nand's Clock (clock signal of the parameter comparison circuit), 1st Comparison (output signal of the first comparison) and EP511 (lowest-priority output of the priority encoder) signals have been plotted. These results show a 7.5 ns delay for the search operation, which also includes the data write into a RAM memory. The comparison of these performance results with those described by Lin in [11] shows how the delay occurred by the banked architecture is a 25% lower than the original Pb-CAM (10 ns).

V. Conclusions

This paper presents a low-power implementation of a CAM based on a banked approach for a precomputation-based architecture. The banked Pb-CAM that has been presented is suitable for applications demanding a large size of the storage device while high performance is still required.

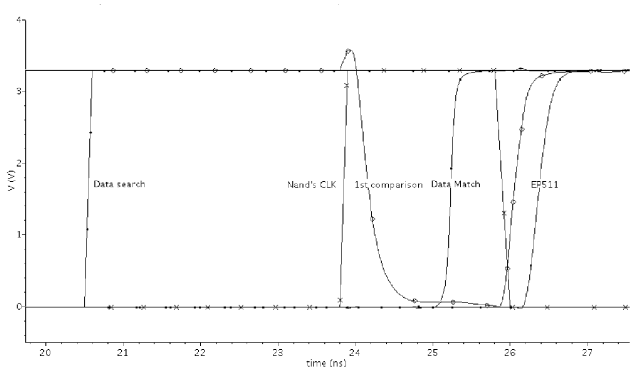

Fig. 7. Performance analysis (electrical simulation).

The experimental work has analyzed the reduction in complexity and power consumption in comparison with other approaches, overcoming previous results. Also, the effect of the validity of words in the memory has been studied, showing how the proposed implementation takes advantage of this fact, specially for two's power sizes.

Acknowledgements

This work was supported by the Spanish Ministry of Science and Education under contract TIC2003-07036.

References

[1] S. Swaminathan, S. B. Patel, J. Dieffenderfer, and J. Silberman, "Reducing Power Consumption during TLB Lookups in a PowerPC™Embedded Processor," in *International Symposium on Quality of Electronic Design*, 2005.

[2] S. Stas, "Associative Processing with CAMs," in *Northcon*, 1993.

[3] K. J. Lin and C. W. Wu, "A Low-Power CAM Design for LZ Data Compression," *IEEE Trans. on Computers*, vol. 49, no. 10, pp. 1139–1145, October 2000.

[4] Y. Tang *et al.*, "CAM-Based Label Search Engine for MPLS over ATM Networks," in *IEEE GLOBECOM*, 2001.

[5] H. Liu, "Reducing Routing Table Size Using Ternary-CAM," in *Symposium on High Performance Interconnects*, 2001.

[6] I. Arsovski and A. Sheikholeslami, "A current-saving match-line sensing scheme for content-addressable memories," in *IEEE Int. Solid-State Circuits Conf.*, 2003.

[7] F. Shafai *et al.*, "Fully parallel 30-MHz, 2.5-Mb CAM," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1690–1696, November 1998.

[8] I. Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content-addressable memories," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1958–1966, November 2003.

[9] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in *IEEE Custom Integrated Circuits Conf.*, 2003.

[10] S. Choi, K. Sohn, and H.-J. Yoo, "A 0.7-fJ/Bit/Search 2.2-ns Search Time Hybrid-Type TCAM Architecture," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 254–260, January 2005.

[11] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A Low-Power Precomputation-Based Fully Paralel Content-Addressable Memory," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 654–662, April 2003.

[12] H. Noda, K. Inoue, and M. Kuroiwa, "A Cost-Efficient High-Performance Dynamic TCAM With Pipelined Hierarchical Searching and Shift Redundancy Architecture," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 245–253, January 2005.

[13] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE Journal Of Solid-State Circuits*, vol. 36, no. 6, pp. 956–968, June 2001.