Practical Implementation of a Low-Power Content-Addressable Memory

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— **This paper presents a practical implementation of a CAM oriented to low-power applications. This implementation takes into account every architectural module integrating the CAM system, as well as the relations among them that allow the minimization of the energy dissipation. This work has followed the complete design process in order to optimize the implementation results in terms of power consumption, and avoiding the performance penalty of previous approaches which did not consider the whole system.**

I. Introduction

Content-Addressable memories (CAM), especially fully parallel CAM, provide an exclusive fast data-search function by accessing data by its content rather than its memory location indicated by an address. Nowadays, it can be found a wide range of applications taking advantage of the CAM function: lookup tables, databases, associative computing, data compression, and others. Recently in the network computing era, fast lookup tables are required for address resolution in network switches and routers such as LAN bridges/switches, ATM switches, and layer-3 switches. Moreover, CAM's fast search functions are especially useful in supporting the quality of service (QoS) required for real-time applications like multimedia data transmission. Even faster search operations are desired for higher speed communications networks like OC-192 and OC-768 where address resolution within less than 10 ns is required.

The low memory density in CAMs (when comparing with DRAMs and SRAMs) and its relative high cost implementation, limits its use in applications where high memory capacity is not the deciding factor. CAM's low memory density is mainly due to its area-consuming memory cells and the difficulty of implementing the column address.

Another limiting factor to be considered is that power consumption in CAM is still high in comparison with RAM of similar capacity. The main reason for this is the large current and large power consumption in the search operation due to inherent nature of CAM's parallel search. In this operation, the input data which has been search for, is sent to all memory locations, activating all its arrays for simultaneous data comparison. Therefore, current flows in major data system circuits including long heavy data lines and bitlines in all cell arrays. Another main power consumers during the search operation are the match detection circuits (one for each data word) that include heavy word match lines, and other search-related circuits such as encoders needed for the selection of only one valid match. From this combination of the parallel search power consuming factor with the extra hardware cost needed for its implementation comes the difficulty to achieve a high density integration in CAM memories.

Several techniques to reduce the current flow during the search operation have been developed, as will be described in section IV. For example, a NAND-type match-line can be used in the match circuits instead of another type to get the word match signal. In the NAND-type implementation, the match-line driver devices of each bit cell of a memory location are connected in series so this implementation is inherently slower than the NOR-type, even though the power consumption is reduced.

Despite the case just been mentioned, several efforts have been carried out to improve the power consumption, performance and implementation area of the CAMs. Nevertheless, most of these approaches have forgotten to consider all the architectural modules that play part in this system focusing only on some of the modules. Moreover, the relations between them can overcome any of the power or performance improvements achieved in a specific place. In this paper we propose a practical implementation of a CAM architecture oriented to low-power applications. This work performs the complete design flow for every architectural module and selects those implementations with lower power consumption to minimize the energy dissipation of the whole system.

The paper structure is as follows. Next, an overview of the CAM architecture and operation is summarized in section II, while the main power consumers in this structure are described in section III. The previous works on this area are reported in section IV, and the proposed architecture is deeply described in section V. Finally, some conclusions are drawn.

II. CAM Overview

Figure 1 shows a block diagram of a simplified conventional (non-pipelined non-hierarchical) CAM architecture. The CAM has four horizontal words (CAM entries) and four bits per entry. The CAM compares the search data (i.e., 1001 in the figure) to all the entries in the CAM, and identifies the words that match. Whenever a search operation happens, the vertical search-lines (SLs) are reset to ground and the horizontal match-lines (MLs) to V_{DD}. The precharge of the MLs to V_{DD} puts them all temporarily in the match state while the discharge

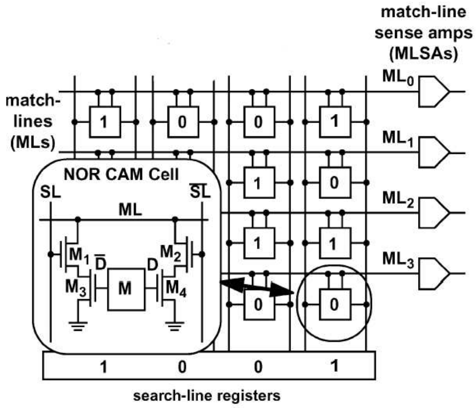

Fig. 1. Simplified CAM architecture [1]

of the SLs prevents wasting of direct-path current during the subsequent precharge of the MLs. When the search operation is complete, the match-lines that remain in the match state will identify the words that match the search data.

After the ML precharge, the search-line registers drive the search data onto the differential search-lines ($\bar{SL}$). Then each CAM cell compares its stored bit against the corresponding search bit on the SLs. The inset of Figure 1 illustrates the schematic of a NOR-based CAM cell. This cell consists of a memory cell, M, which is a SRAM cell in this work [1], and four compare transistors arranged in two pulldown paths between ML and ground. In the cell, a mismatch (or miss for short) between SL and D results in a series path from the ML to ground. On the other hand, a match between SL and D results in no path from ML to ground. The pulldown paths of the individual CAM cells combine on the ML like a dynamic NOR to form either a path to ground (in the case of any miss in the word) or no path to ground (in the case of a full match). In other words, any single miss in any of the cells of a word creates a path to ground that discharges the ML (indicating a miss). Conversely, if all bits of a word match, then the ML remains precharged high (indicating a match). In the example of Figure 1, the search data, 1001, matches the uppermost word in the array. Hence, its associated ML (ML_0) remains high indicating a match, while all the other match-lines discharge to ground, indicating misses. The match-line sense amplifiers (MLSAs) are used to distinguish a match from a miss, often using a threshold voltage as the reference.

As mentioned earlier, the two main sources of power consumption are the highly capacitive MLs and SLs. Next section will give the details of the power consumption in the CAM architecture.

III. Energy Consumption in CAMs

A. Match-Line Energy Consumption

About half the energy consumed in a CAM is due to the repeated precharging and discharging of all but one of the match lines in each access. This is due to the parallel (or NOR type) implementation of the match operation. Serial (or NAND type) CAM designs, search one bit at a time (for each row) so that they do not discharge a single large capacitance when there is no match. Unfortunately, they are generally slower than parallel CAMs, as their search speed depends on the number of cells in a row.

Let's consider a $n{-}words{\times}m{-}bits$ memory array. Regarding the NOR-type match line, there are *n* drain capacitances on it, so the effective capacitance is nC_d. In the clock precharge phase all match lines, except the one which is evaluated as *match* in the previous cycle, are charged from *0* volt to *V* volt. Therefore, we could model the evaluation power of a NOR-type CAM as

$$fmnC_dV^2$$

Regarding the evaluation power for a NAND-type match line, if each bit has equal possibility to be stored '1' or '0' regardless of its spatial location, the effective capacitance on a match line of a NAND-type CAM is

$$(1 \times \frac{1}{1} + 2 \times \frac{1}{2} + 2 \times \frac{1}{4} + \ldots + 2 \times \frac{1}{2^{n-1}} + 1 \times \frac{1}{2^n})C_d$$

which could be approximated as $3C_d$. But the voltage swing of the capacitance declines by a factor r_d. So the evaluation power of *m* NAND-type CAM words is

$$fm3C_dr_dV^2$$

B. Search-Line Energy Consumption

Traditional CAM cells combine the search lines with the bit-lines. This causes an increase in the capacitance of each search line as an extra transistor drain per cell is present (though it could be shared in the cell layout). Even after separating search and bit lines, driving the search lines accounts for almost half of the energy in search operations.

Apart from having a relatively high capacitance, in parallel CAMs, one search line per bit switches at every search, even when the same value is searched each time. This is because the search lines must be driven low while the match lines are being precharged to avoid a direct short-circuit from supply to ground through the cells that do not match. On the other hand, serial CAMs form chains of transistors that propagate a value when all the cells match; evaluation is coordinated by precharging the intermediate nodes so that the search lines do not have to be precharged (or predischarged) for every search. Therefore, the power dissipation related to the search-lines can be approximated by

$$f\frac{m}{2}nC_gV^2$$

This value is common for both NOR and NAND types.

IV. Related Work

Content addressable memory (CAM) is widely considered as the most efficient architecture for pattern matching required by the LZ77 compression process. In [2], a low-power CAM-based LZ77 data compressor is proposed. By shutting down the power for unnecessary comparisons between the CAM words and the input symbol, the proposed CAM architecture consumes less power than the conventional implementation without noticeable performance penalty.

Based on memory traces, which usually cause tag mismatch within the lower four bits, in [3] a new serial CAM organization is proposed which consumes just 45% more than a single tag RAM read and is only 25% slower than the conventional, parallel CAM. In [4] this work is extended to exploit the address patterns commonly found in application programs, where testing the four least significant bits of the tag is sufficient to determine over 90% of the tag mismatches.

In [5], the proposed CAM word structure adopts a static pseudo nMOS circuit that not only improves system reliability, but also prevents using clock signal to drive the overall system. In order to reduce static power occurred in the proposed CAM word structure, a precomputation approach is used to turn off most of pseudo nMOS circuits. This approach is extended in [6] with a design based on a precomputation skill that saves not only power consumption of the PB-CAM (Precomputation-Based CAM) system, but also reduces transistor count and operating voltage of the PB-CAM cell.

The work described in [7] derives power models for four low-power CAMs from the fCV^2 base model. Attending to this work, CAM has three major power-sinking sources: evaluation power, input transition power and clocking power.

Also, [8] presents a new CAM cell with a pMOS match-line driver which reduces search rush current and power consumption, allowing a NOR-type match-line structure suitable for high-speed search operations.

Despite the previous approaches to describe a low-power implementation of a CAM, these works have not considered the system as a whole, optimizing every architectural module in terms of power consumption. The work proposed in this paper overcomes such limitations by performing a complete analysis and design of the low-power CAM from a system perspective.

V. Proposed Architecture

A. General Organization

A general CAM architecture usually consists of the data memory with a valid bit field, the address decoder, the bitline precharger, the word match circuit, and the address priority encoder (see Figure 2). The valid bit field indicates the availability of stored data. In the data searching operation, the input data is sent into the CAM to be compared simultaneously with all those valid data stored in the CAM. An address from among those matches of comparison is sent to the output.

To minimize the power consumed during the comparison, one of the best approaches is to reduce the comparison operations to a minimum. This can be carried out by a scheme based on the precomputation of a simple parameter which, preferably in a unique way, characterizes every data word. An example of this kind of parameters may be the number of ones in the word. In this work, the CAM architecture adopts the ones count function to perform the parameter extraction, because the ones count function filters a large amount of unmatched data with a small bit length.

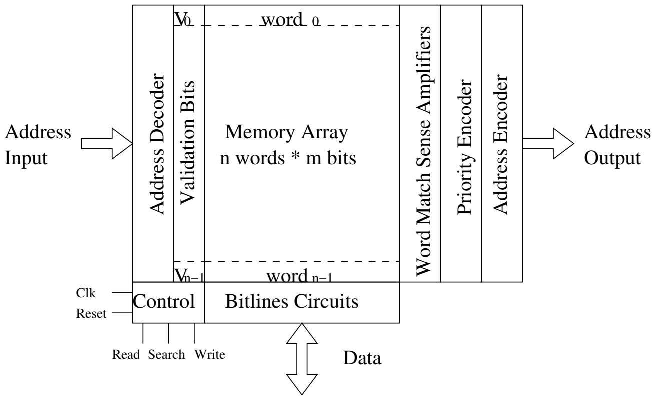

Fig. 2. General scheme for the the CAM architecture

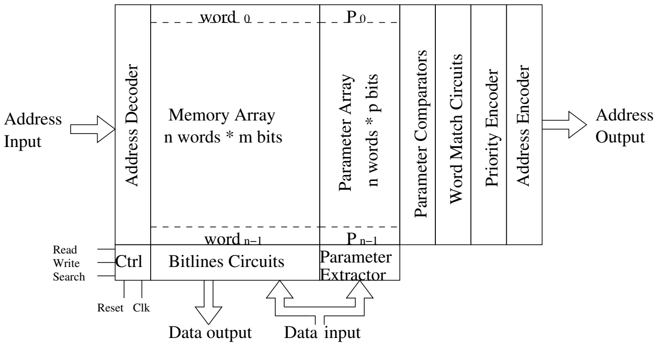

Fig. 3. General scheme for the the PB-CAM architecture

The memory organization (depicted in Figure 3) of the proposed CAM architecture is composed of the data memory, the parameter memory, and the parameter extractor. This parameter extractor has been implemented with a chain of full-adders to perform the ones-count function.

During the data writing operation, the parameter extractor calculates the parameter of the input data, and then stores the input data and its parameter into the data memory and the parameter memory, respectively.

In the data searching operation, in order to reduce the large amount of comparison operations, the operation is separated into two comparison processes. During the first stage, the parameter extractor calculates the parameter of the input data and performs a preliminary comparison (the parameter comparison circuits compare in parallel this parameter of the input data with all the parameters stored in the parameter memory). Based on the two comparison processes, if there is a mismatch between the stored parameter and the one belonging to the input data, then the number of comparisons in the second comparison process is largely reduced. The parameter comparison process is also known as the *precomputation process*.

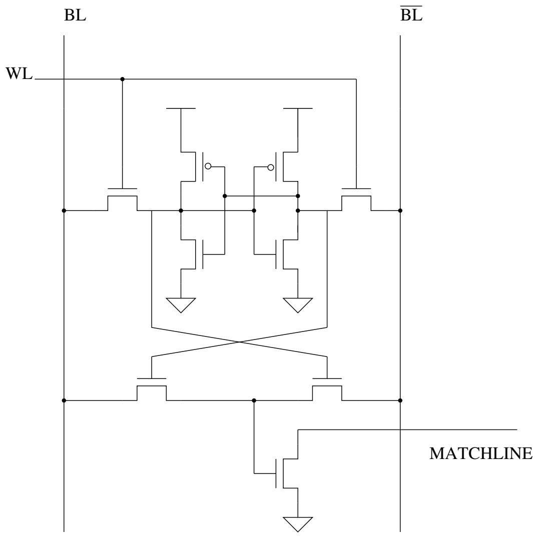

Fig. 4. 9-transistor CAM cell implementation

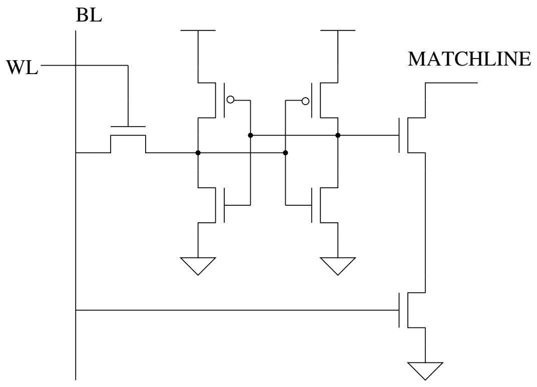

Fig. 5. 7-transistor CAM cell implementation

B. Memory Cell

In the traditional CAM circuit design, the CAM cell is constructed by a nine-transistor architecture as shown in Figure 4. The CAM cell consists of an ordinary six-transistor SRAM cell to store a data bit, an XOR-type comparison circuit containing two nMOS transistors, and an nMOS pull-down device to drive the word match line. Some characteristics of the traditional CAM cell structure are as follows.

1) The circuit is constructed by nine-transistor architecture.
2) The comparison circuit is performed by a PTL-type XOR gate.
3) Input circuits are two complementary heavy load bit-lines.

Unlike the traditional CAM cell design, the proposed CAM cell based on precomputation is a seven-transistor cell structure, as shown in Figure 5. This cell incorporates a standard five-transistor D-latch device to store a data bit and a NAND-type comparison circuit containing two nMOS transistors to drive the word match line. [9]

The parameter comparison function is performed by a static CMOS structure. Its function acts like the traditional CAM word circuit; it compares the parameter of the input data with the parameter of the stored data and then controls the output gate of its related PB-CAM.

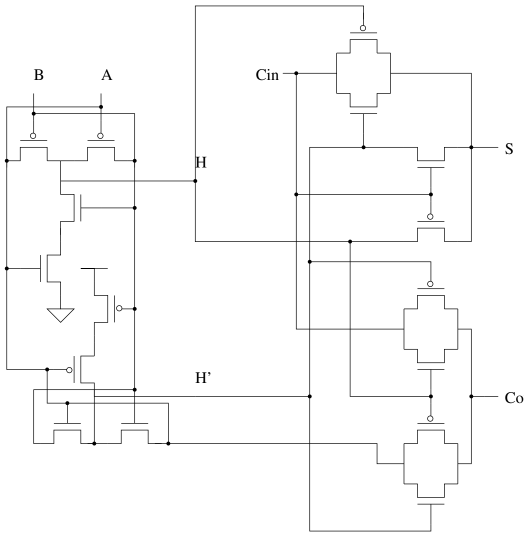

Fig. 6. New 16-transistor full adder implementation

Some functions are required to perform the parameter extraction in the proposed CAM architecture, such as ones count function, parity function, and remainder function. Also, the selection of this parameter function allows to reduce in two transistors the memory cell implementation due to the logic simplification occurred [9].

C. Full Adders

The full adder cell used in this design uses 16 transistors [10] (see Figure 6). The transistor number in this design is also the smallest one (transmission full adder has also 16 transistors), and there are fewer glitches in the new 16-transistor design as compared with the 16-transistor full adder because XOR and XNOR gates are generated simultaneously. Due to fewer glitches, the power can be saved and performance can be increased. Finally, the new implementation counts with no inverters in the design and no short-circuit power component.

As mentioned before, for this design, the full design flow has been followed. Figure 7 shows the proposed layout which shows the optimality achieved in terms of area.

When compared with the traditional 16-transistor implementation, the power consumption results show an improvement of 3% for the new 16-transistor architecture, and a 16% improvement for the performance (speed) results.

D. Priority Encoder

The proposed implementation of the priority encoder is based on an 8-bit cell which greatly simplifies the design (Figure 8). The transistor count is reduced from 102 to 62 and necessary precharge nodes are thus also reduced [11].

The selected parallel priority look-ahead architecture presents several advantages. First, the 8-bit cell found before

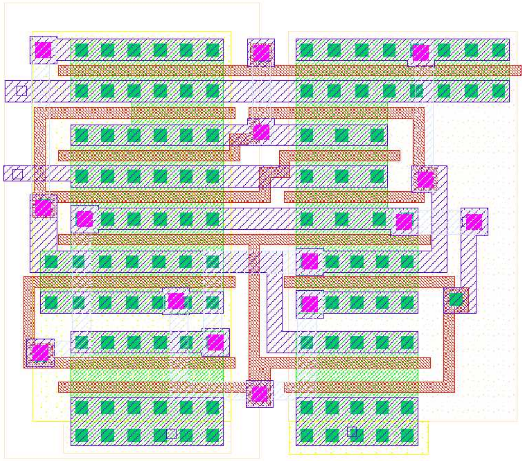

Fig. 7. Layout for the new 16-transistor full adder implementation

the priority encoder[1] performs the precomputation of the look-ahead signals. Therefore the less significant bits do not need to wait for the most significant cells. For a 64-bit priority encoder, the total gate delays are only three: one OR gate at the input, one AND gate inside the 8-bit cell of the look-ahead priority encoder, and one AND gate in the data priority encoder.

Second, the look-ahead signal routing is much more regular in this multilevel implementation, which simplifies the layout. And, finally, the architecture can be easily pipelined due to the division of the computation into two stages.

The obtained power consumption for this architecture achieves 2.68 mW, which greatly reduces the 6.18 mW of the conventional implementation [11].

E. Sense Amplifier

Due to the memory cell design, the small signal on the match-line and the bit-line must be detected by a single-ended sense amplifier. The current-mode latch sense amplifier shown in Figure 9 is selected for this implementation [12].

In the case of the precomputation-based architecture, the optimal sense amplifier implementation showed previously is changed to a buffer (only for the match-line). The reason for this is the improved performance results in terms of speed and consumption obtained when using only two inverters in comparison with the differential amplifier.

The applicability of the buffer-based implementation is limited to non-precharged lines in order to speed up the output of the matchline voltage.

VI. 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 in this work is a $.35\mu m$ one from Austria MicroSystems, dual-poly quadruple-metal CMOS process.

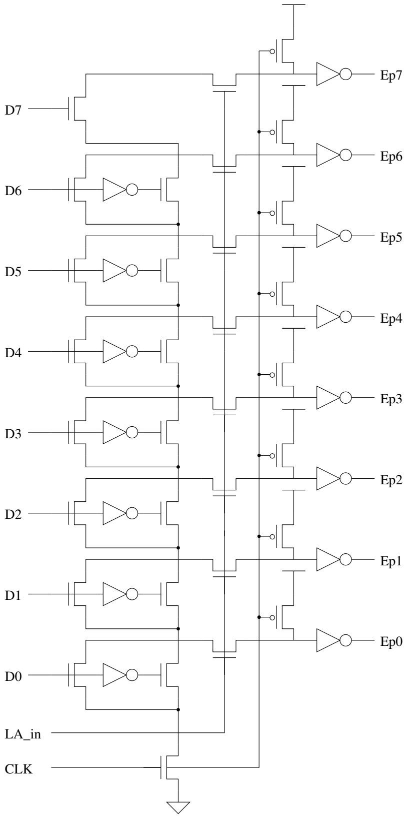

Fig. 8. 8-bit priority encoder cell

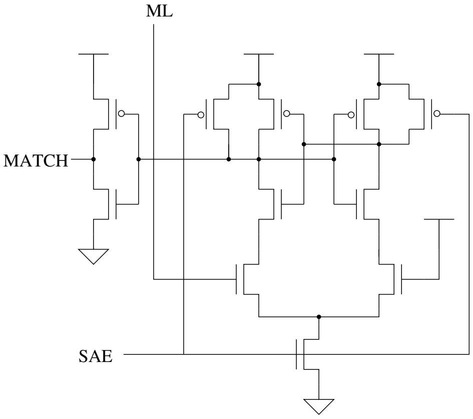

Fig. 9. Sense amplifier

Based on the proposed PB-CAM word structure and cell circuit design, the PB-CAM architecture achieves low-power, low-cost, and low-voltage features. The measured search access energy consumption for the 128 x 30 PB-CAM is 86 fJ/bit/search [9], quite reduced in comparison with 131 fJ/bit/search [8].

As was explained before, the most power consuming task in the CAM access is the search operation. An analysis of the number of memory cells excited during the search operation has been performed. This evaluation gives an overview of the behavior in terms of power consumption for several memory

[1] What can be considered a priority encoder itself.

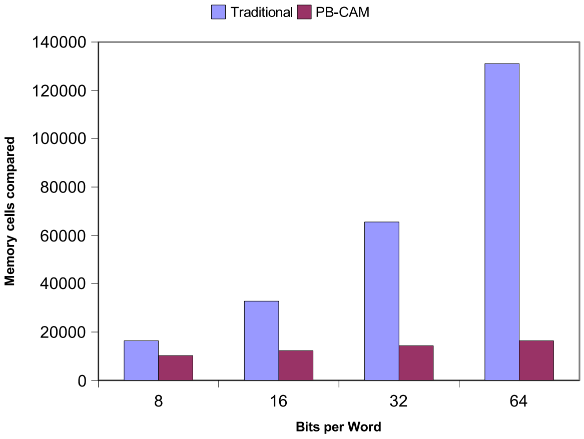

Fig. 10. Activated cells for a fixed memory capacity (2048b)

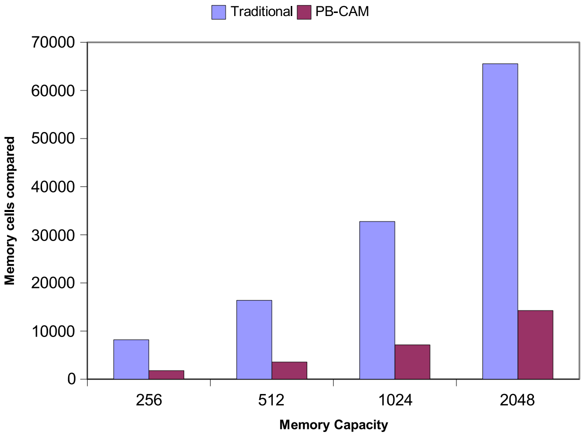

Fig. 11. Activated cells for a fixed word length (32b)

capacities and word lengths. These results are shown in Figures 10 and 11.

As can be seen in Figure 10, the traditional implementation shows a linear trend on the number of memory cells compared when the memory size is fixed (2048 positions) and the number of bits per word is ranged. On the contrary, the number of memory cells compared remains almost constant (logarithmic growth) for the PB-CAM implementation, due to the fact that the parameter size is $log(Nbits + 2)$. This graph also explains how for a reduced size of the word length, the parameter overhead could overcome the savings.

Figure 11 shows the linear tendency on the number of memory cells compared for both the traditional and the PB-CAM implementation when the word length is fixed (32b) and the memory size is ranged. The different slopes between both linear tendencies explain how for a large size of the memory, the savings with the PB-CAM are quite representative.

VII. Conclusions

Nowadays, the limiting factor in applications where the CAMs play a critical role is the power consumption of these devices. The integration levels achieved by current technology processes have turned the area and performance factors into secondary actors.

The work presented in this paper has shown a practical implementation of a CAM oriented to low-power applications. This implementation takes into account every architectural module integrating the CAM system, as well as the relations among them that allow the minimization of the energy dissipation.

The proposed design is easily adapted to a pipeline implementation and the performance results have not been compromised. Therefore, the use of this CAM architecture into the pipeline of a high-performance computer can be considered.

Acknowledgements

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

References

[1] K. Pagiamtzis and A. Sheikholeslami, “A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, September 2004.

[2] K.-J. Lin and C.-W. Wu, “A low-power CAM design for LZ data compression,” *IEEE Transactions On Computers*, vol. 49, no. 10, pp. 1139–1145, October 2000.

[3] A. Efthymiou and J. D. Garside, “An adaptive serial-parallel CAM architecture for low-power cache blocks,” in *IEEE International Symposium on Low Power Electronics and Design*, 2002, pp. 136–141.

[4] ——, “A CAM with mixed serial-parallel comparison for use in low energy caches,” *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 3, pp. 325–329, March 2004.

[5] C.-S. Lin, J. C. Chang, and B.-D. Liu, “Design for low-power, low-cost, and high-reliability precomputation-based content-addressable memory,” in *IEEE Asia Pacific Conference on Circuits and Systems*, 2002, pp. 319–324.

[6] C.-S. Lin, K.-H. Chen, and B.-D. Liu, “A low-power and low-voltage fully parallel content-addressable memory,” in *IEEE International Symposium on Circuits and Systems*, 2003, pp. 373–376.

[7] I. Y.-L. Hsiao and D.-H. W. amd Chein-Wei Jen, “Power modeling and low-power design of content addressable memories,” in *IEEE International Symposium on Circuits and Systems*, 2001, pp. 926–929.

[8] 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.

[9] C.-S. Lin, J.-C. Chang, and B.-D. Liu, “A low-power precomputation-based fully parallel content-addressable memory,” *IEEE Journal Of Solid-State Circuits*, vol. 38, no. 4, pp. 654–662, April 2003.

[10] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, “Performance analysis of low-power 1-bit CMOS full adder cells,” *IEEE Trans. on VLSI*, vol. 10, no. 1, pp. 20–29, February 2002.

[11] C. Kun, S. Quan, and A. Mason, “A power-optimized 64-bit priority encoder utilizing parallel priority look-ahead,” in *IEEE International Symposium on Circuits and Systems*, 2004, pp. 753–756.

[12] T. Kobayashi and al., “A current-mode lath sense amplifier and a static power saving input buffer for low-power architecture,” in *Symposium on VLSI Circuits*, 1992, pp. 28–29.