

Research Article

Power Considerations in Banked CAMs: A Leakage Reduction Approach

Pedro Echeverría, José L. Ayala, and Marisa López-Vallejo

Departamento de Ingeniería Electrónica, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to Pedro Echeverría, petxebe@die.upm.es

Received 26 April 2007; Accepted 9 December 2007

Recommended by Jean-Baptiste Begueret

The content-based access of CAMs makes them of great interest in lookup-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 precomputation-based architecture for low-power and storage-demanding applications where the reduction of both dynamic and leakage power consumption is addressed. Experimental results show that the proposed banked architecture reduces up to an 89% of dynamic power consumption during the search process while the leakage power consumption is also minimized up to a 91%.

Copyright © 2008 Pedro Echeverría et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

A broad range of modern applications demand large storage devices with fast search capabilities. Content-addressable memories (CAMs) have emerged as one of the favorite devices for such applications [1–3]. 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], associative computing and data compression [2]. High-speed networks such as gigabit Ethernet and ATM switches [3] also benefit from this particular structure [4].

However, CAMs pay a high hardware cost for this content-based access because the memory cell must include comparison circuitry, negatively impacting the size/speed tradeoff and complexity of the implementation. Usually, a 9-transistor cell is required instead of the 6-transistor cell used in SRAM. Moreover, the large amounts of parallel comparisons performed in conventional CAMs make the device 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 the implementation and characterization of a CAM with low-power constraints. The proposed architecture is highly scalable and provides high-performance functioning at large sizes. Moreover, it achieves great savings for both dynamic (89%) and leakage (91%)

power consumption. 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 reducing the dynamic power consumption of the match line [5] and enhancing the search speed [6]. Although many approaches addressing dynamic power dissipation have been reported [7–9], the resulting circuit techniques have either substantial area overhead, deficiencies in noise immunity, 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 and also addresses leakage energy reduction in an efficient way. The architecture presented here improves the energy savings obtained by Lin et al. in [10], and recently extended by Noda et al. [11] and Choi et al. [9]. 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.

Moreover, the leakage energy consumption (which is one of the main issues regarding current electronic design) has not been addressed by previous approaches. Static power dissipation is becoming as important as dynamic dissipation as transistor gate sizes are being reduced [12]. Our work

improves the dynamic power savings of the referred approaches and also reduces the leakage energy consumption of the memory to a minimum.

As stated before, this work is based on a parameter precomputation-based architecture [10] (PB-CAM from now on); however, we are able to reduce the parameter word's size with respect to [10], 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 89% of the dynamic power consumption and a 91% of the leakage power consumption) improve the previous implementations of similar technologies and also improve the scalability capabilities of architectures like [9, 11].

Our research work on this field has already shown good results in terms of area and dynamic power consumption [13, 14]. This paper expands [14] where an improved architecture was presented (with a novel hardware mechanism to reduce the static power consumption and increase the dynamic energy savings) with new experimental results and a deeper analysis of the consequences of applying leakage reduction techniques over CAM memories.

The paper is composed as follows. Section 2 presents the first designed approach where the minimization of the dynamic power consumption is addressed, while the leakage power reduction technique is shown in Section 3. The experimental results are introduced in Section 4. Finally, some conclusions are drawn.

2. DEvised ARCHITECTURE: BANKED APPROACH

The first architecture presented in this section describes a banked implementation of the mentioned PB-CAM [10] in order to reduce the dynamic power consumption during the search operation. The main idea of the PB-CAM is to store a parameter word (obtained by a formula, e.g., a one's count) to perform the comparison process in a reduced number of memory positions, saving dynamic power consumption. However, the total power consumption of the logic can still be too high for low-power applications. Additionally, the PB-CAM architecture is also based on a novel seven-transistor data memory cell (that can only be used on conventional architectures) instead of the common one of nine or ten transistors, that can only be used on conventional architectures.

Our architecture employs the precomputed parameter to perform a power-aware ordering of the data. The extracted parameter allows us the classification of the memory contents based on the one's count. This classification can be used to store the memory contents in such an efficient way that the search operation is restricted to a smaller memory size.

The order of the memory data attending to the one's count parameter makes possible to split the memory architecture into independent banks where every data in the bank has the same value for one subset of the parameter (e.g., *N-least significant bits*). Moreover, the logic needed for this ordering is very simple and does not present a serious overhead in terms of delay and energy consumption [13].

In Figure 1 our architecture is depicted (RAM area corresponds to the memory where the output address associated

to each tag will be found to complete a CAM-search engine). It can be observed that each memory word is composed of a validation bit, the data word and part of its parameter. The parameter extractor computes the parameter of the input tag and a bank decoder selects the proper bank with a subset of the bits of the calculated parameter. Then, the rest of the parameter bits and the input data are searched in the decoded bank.

Due to the banked implementation of the memory, the operation of the architecture is restricted to just one bank every cycle. One of the advantages of this banked structure is the reduction of the dynamic power consumption as the charge in the bit lines is limited to one bank (the driven line is simplified to the bit line of the accessed bank of the memory). This behavior is also shown by the parameter lines and also has a positive influence in the memory speed. The complexity of the logic shared for the banks (buffers, priority encoders, and address decoders) is reduced when the bank approach is applied. This simplification saves area, power consumption and improves the delay of these devices.

3. DEvised ARCHITECTURE: DROWSY APPROACH

The banked PB-CAM presented in Section 2 has been improved to reduce the static power consumption of the architecture. In this case, the unused memory banks are put into a low-power state and a pipeline carefully manages the operations to avoid any performance penalty. This approach is based on the design of a low-power cell and a further pipelined subbanked implementation.

3.1. Low-power cells (data and parameter)

As has been previously described, the memory area has been split into several banks, which can be independently accessed. Then, a dynamic voltage scaling (DVS) technique is applied to turn the unused banks into a low-power state and thus save as much energy as possible in the system. However, when the objective is a memory device, the cost of recovering the lost information could hide any power saving or, at least, represent a very significant time penalty. Moreover, something has to be done in the powered-down banks to prevent the information from being lost.

An efficient approach to achieve the low-power (drowsy) state is proposed by [15], where a DVS technique is exploited to reduce static power consumption. As is well known, both dynamic and static power consumption are proportional to the supply voltage. DVS technique benefits of this fact by turning down the power supply to reduce power consumption. However, reducing the supply voltage also has a negative impact in other parameters such as speed, hence DVS techniques look for combining different voltages while not affecting other parameters.

The method proposed by Flautner et al. implements DVS to reduce the leakage power of cache cells by scaling the voltage of the cell to a lower voltage that ensures the preservation of the state of the memory cells. This voltage can be conservatively approximated to 1.5 times V_{th} [15], but further

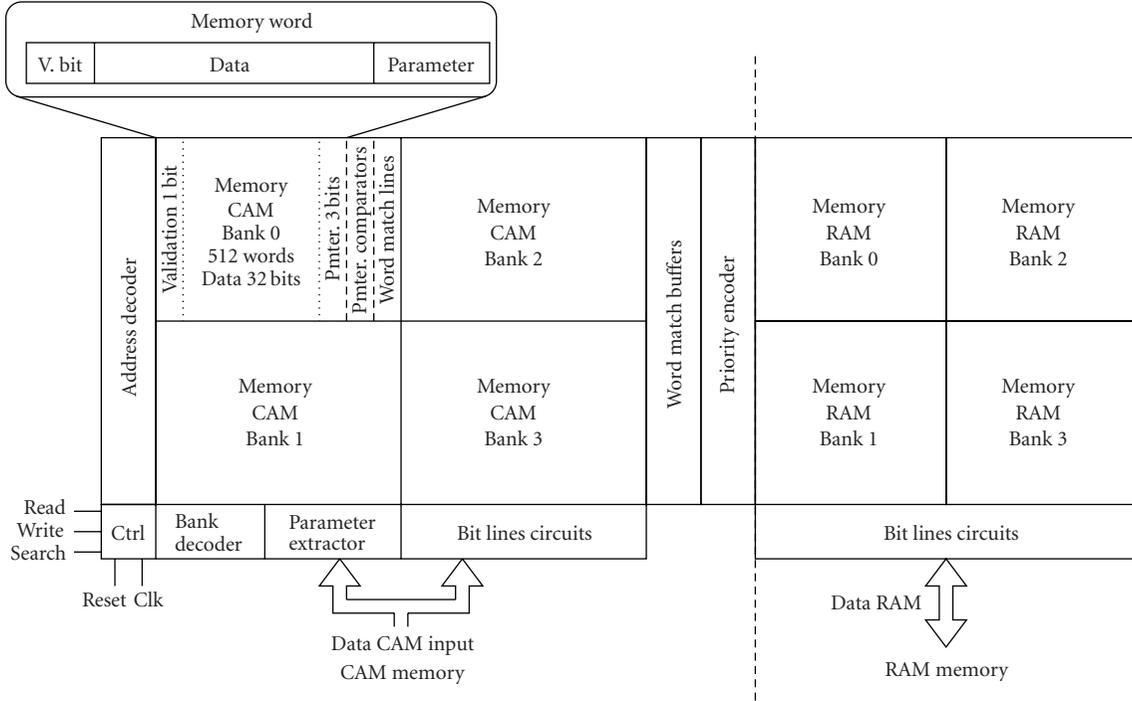


FIGURE 1: Banked architecture (4-bank implementation).

reductions of the scaled voltage would increase static power savings [16].

Figures 2 and 3 show the modified memory cells (data and parameter) used to support the drowsy state with a dual power supply (not part of the cell, it is shared by all the cells in a same bank). As can be observed, the dual power supply is switched to low V_{DD} when the cell is in drowsy state. It is necessary to use *high- V_{th}* devices as pass transistors because the voltage on bit lines could destroy the cell contents. Before a memory position can be accessed, the power supply has to be switched to high V_{DD} (wake up) to restore the contents and allow the access. The careful management of these operations along the pipeline presented in the next section takes care of this extra clock cycle to avoid any performance penalty.

As mentioned before, each bank of the CAM architecture counts with the additional logic required to implement the DVS mechanism. Since the low-power consumption state is selected for the whole bank instead of a specific memory position, the overhead of the control logic is greatly minimized.

3.2. Pipeline

A clear way to improve the access time of a CAM is the use of a pipeline structure, which additionally provides greater scalability in the performance and density of the applications that make use of CAMs. The aforementioned DVS technique can take advantage of this pipeline to awake the drowsy cells one clock cycle before the access. Only one of the banks needs to be on while the rest can remain in the drowsy state saving leakage energy.

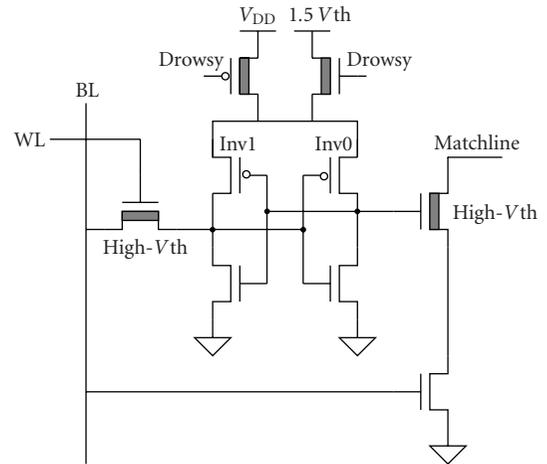


FIGURE 2: 7-transistor CAM cell with drowsy support.

In our approach, the devised pipeline configuration includes the three operations needed in a CAM-search engine: READ, OVERWRITE, and WRITE. READ is the read operation in the associated RAM memory after the tag is found in the CAM. In OVERWRITE, after the tag is found in the CAM, a write operation is done in the RAM memory, and WRITE is the operation to write a tag and its data in both CAM and RAM memories.

The pipeline stages defined within those operations are EXT (parameter extraction of the input tag and selection of the working bank), SEARCH (tag comparison in the CAM), DEC (decodification of internal address, common for

Instruction	1	2	3	4	5	6
Read	EXT	SEARCH	READ_R			
	Pmt. Extr.	C. Mem	R. Mem			
Write		EXT	DEC	WRITE_CR		
		Pmt. Extr.	Addr. Dec	C. Mem R. Mem		
Overwrite			EXT	SEARCH	WRITE_R	
			Pmt. Extr.	C. Mem	R. Mem	

(a) 3-stage pipeline

Instruction	1	2	3	4	5	6
Read	EXT	NOP	SEARCH	READ_R		
	Pmt. Extr.		C. Mem	R. Mem		
Write		EXT	DEC	WRITE_C	WRITE_R	
		Pmt. Extr.	Addr. Dec	C. Mem	R. Mem	
Overwrite			EXT	NOP	SEARCH	WRITE_R
			Pmt. Extr.		C. Mem	R. Mem

(b) 4-stage pipeline

FIGURE 4: Structure of the proposed pipeline.

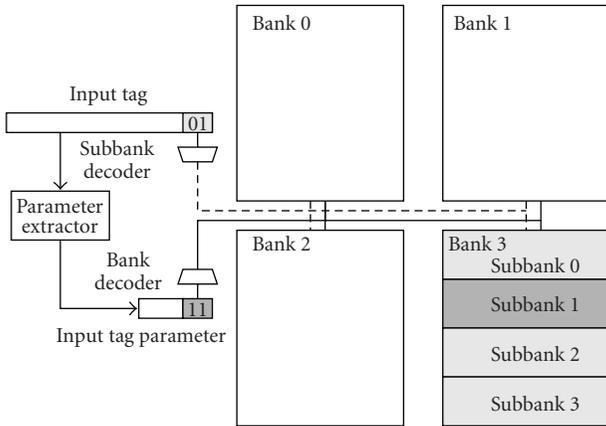


FIGURE 5: Subbanking scheme.

TABLE 1: Drowsy PB-CAM with two different scaled voltages.

Voltage	1.5 V _{th}	1.25 V _{th}
Drowsy state; leakage reduction	92%	98%
Leakage savings	86%	91%
Dynamic wake-up penalty	2.77 fJ/bit	3.06 fJ/bit

to implement the designed architecture has been 0.35 μm from Austria MicroSystems (as the base architecture is also a 0.35 μm design), while the estimation of the leakage energy savings has been carried out using the 70 nm BPTM [17] models, as the leakage power cannot be studied with the technology selected for the implementation.

4.1. Banked implementation

The banked architecture has been firstly evaluated in terms of the energy savings obtained after reducing dynamic power.

The simulated memory is the architecture described in Figure 1, implemented as a memory of 2048 positions and 32 bits per word, and split into 4 independent banks. Our approach decreases the dynamic power consumption by a 78% (18.86 fJ/bit in the banked architecture with respect to 86 fJ/bit in [10]).

The area improvement achieved with the proposed architecture has also been evaluated in terms of number of transistors. When the memory size is fixed and the number of bits per word is varied, there is a reduced area improvement of the banked implementation with respect to the original PB-CAM due to the savings in the parameter word length and comparison logic. Compared to a traditional implementation, area savings are quite representative for architectures with more than 16 bits (up to 17.5% in the range considered: 8 to 128 bits per word and a 2048 words memory size) due to the use of 7-transistor memory data cells instead of the 9-transistor one of a traditional architecture.

For the implemented architecture, the area savings obtained are 13.5% with respect to a traditional architecture and 10.7% respect to the base PB-CAM.

Finally, the performance of the design has been analyzed to assure the required fast response. The 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 the ones described by Lin in [10] shows how the operation time in the banked architecture is a 25% faster than in the original PB-CAM (10 nanoseconds).

4.2. Drowsy implementation

Regarding the energy savings obtained, this approach presents further improvements both in dynamic and leakage energy due to the subbanking and drowsy techniques. The simulated architecture is a CAM with 8192 positions (notice the larger implementation with respect to the baseline

TABLE 2: Comparison of the three approaches (simple, banked, and drowsy PB-CAM [10]).

	PB-CAM [10]	B. PB-CAM	D. PB-CAM
Technology	0.35 μm	0.35 μm	0.35 μm
CAM configuration	128 \times 30	2048 \times 32	8192 \times 32
Dynamic power	86 fJ/bit	18.86 fJ/bit	9.06 fJ/bit
Operation delay	10 ns	7.5 ns	12 ns
Throughput	100 MIPS	133 MIPS	333 MIPS
Static power reduction	—	—	86%–91%

architecture) and 32 bits per word, implemented with 4 independent banks, 4 subbanks per bank, and the described 4-stage pipeline.

With this design, the dynamic consumption is reduced to 9.06 fJ/bit (a decrease of an 89% and 52% with respect to the baseline architecture and our first approach, resp.) when the subbanking technique is used without the drowsy technique.

To apply the drowsy technique for leakage reduction, we have considered two scaled voltages, the one referred as the conservative approach [15] (scaled voltage 1.5 V_{th}) and a smaller one [16] (1.25 V_{th}). The BPTM simulations estimate a 92% of leakage energy savings for the memory cells at a low power state for the conservative voltage, that is increased up to a 98% for the second voltage. Given that the working subbank has to remain at full voltage, the total power consumption, for the simulated architecture, is reduced an 86% and a 91%, respectively.

However when the drowsy technique is applied, the waking up of the accessed subbank is a new source of dynamic power consumption. The total value of this dynamic consumption depends on the scaled voltage used for the low-power state, but in the opposite way for the leakage consumption. When the scaled voltage is reduced, the leakage power consumption is decreased while the waking up dynamic power consumption is increased (the swing voltage between low-power state and working state is increased).

If we extrapolate BPTM simulations to the 0.35 μm simulated architecture and the two proposed voltages, the waking up of a subbank means an additional dynamic power consumption of 2.77 fJ/bit for the 1.5 V_{th} and 3.06 fJ/bit for the 1.25 V_{th} . In Table 1 the experimental results for the two voltages are summarized.

These energy savings can be easily increased if an architecture with more banks or more subbanks per bank is selected. As stated before, and as can be seen in Figure 6, using an 8-banks implementation unbalances the usage of banks and increases the failed search rate, due to the different probabilities of each parameter. While for parameter zero or 32 there is only one possible input data, there are 601 hundred million different input data for parameter 16.

However, using more subbanks per bank has a very weak impact in that factor because all the input data bits have the same probability (and so the ones used to select the subbank), as can be seen in Figure 7. For a 4-bank implementation, the percentage of use between the most and the least used bank is 0.006%. When subbanks are introduced, it can be observed how that percentage duplicates when we qua-

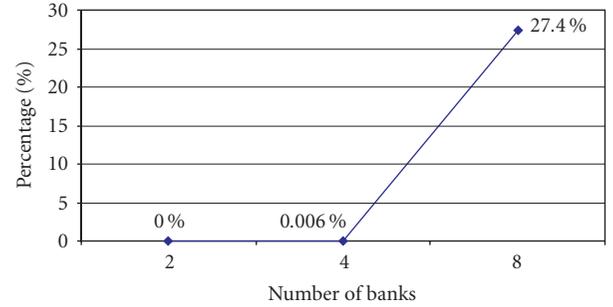


FIGURE 6: Unbalance between banks (no subbanking).

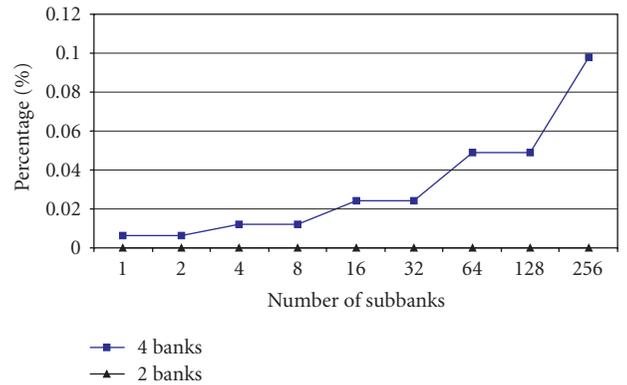


FIGURE 7: Unbalance between subbanks.

druplicate the number of subbanks, and has a very small value for a very wide range of subbanks.

In this way, using more subbanks only increases slightly the unbalance between banks. However, the energy saved while using more subbanks is considerable, as the active part of the memory is smaller: for dynamic power consumption there is less comparison power consumed during a search and less wake up power consumption, and for leakage power more parts of the memory can stay at a low-power state.

The drowsy implementation with subbanking also presents area improvements because two less memory cells per data are stored, while the overhead of the DVS control has a very small impact given that it is shared by all the cells on the same subbank. In Figure 8 it can be observed the area savings when the word length is fixed (32b) and the memory size is ranged. For our simulated architecture we obtained an improvement of 16.1% with respect to the traditional

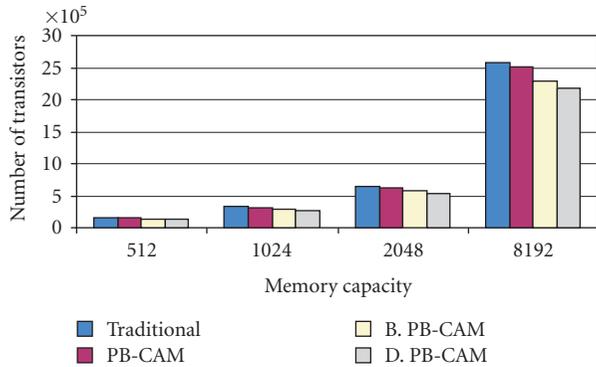


FIGURE 8: Number of transistors for a fixed word length.

implementation, 13.4% respect the base PB-CAM and a 5% compared with our banked implementation.

The throughput of the system is also improved (70% when compared with the baseline architecture, 60% when compared with the banked PB-CAM) due to the pipeline. A comparison of the two approaches presented in this paper, as well as the baseline architecture, can be found in Table 2.

5. 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. Search-based applications with high-performance constrains demand efficient implementations of content-addressable memories to cover the astringent constraints. The work presented in this paper has shown efficient mechanisms to reduce the dynamic and static power consumption by means of hardware modifications. These approaches do not compromise the performance and area improvements achieved with the architecture.

ACKNOWLEDGMENT

This work was supported by the Spanish Ministry of Science and Education under Contract TEC2006-00739.

REFERENCES

- [1] S. Swaminathan, S. B. Patel, J. Dieffenderfer, and J. Silberman, "Reducing power consumption during TLB lookups in a PowerPC/spl trade/ embedded processor," in *Proceedings of the 6th International Symposium on Quality of Electronic Design (ISQED '05)*, pp. 54–58, San Jose, Calif, USA, March 2005.
- [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, 2000.
- [3] Y. Tang, Y. Jiang, and Y. Wang, "CAM-based label search engine for MPLS over ATM networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 1, pp. 45–49, San Antonio, Tex, USA, November 2001.
- [4] H. Liu, "Reducing routing table size using ternary-CAM," in *Proceedings of the 9th Symposium on High Performance Interconnects (HOTI '01)*, pp. 69–73, Stanford, Calif, USA, August 2001.
- [5] I. Arsovski and A. Sheikholeslami, "A current-saving match-line sensing scheme for content-addressable memories," in *Proceedings of IEEE International Solid-State Circuits Conference (ISSCC '03)*, vol. 1, pp. 304–494, San Francisco, Calif, USA, February 2003.
- [6] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed lowpower CMOS fully parallel content-addressable memory macros," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 6, pp. 956–968, 2001.
- [7] 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, 2003.
- [8] K. Pagiampzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '03)*, pp. 383–386, San Jose, Calif, USA, September 2003.
- [9] 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, 2005.
- [10] 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, 2003.
- [11] H. Noda, K. Inoue, M. Kuroiwa, et al., "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, 2005.
- [12] N. S. Kim, T. Austin, D. Blaauw, et al., "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, 2003.
- [13] P. Echeverría, J. L. Ayala, and M. López-Vallejo, "A banked precomputation-based CAM architecture for low-power storage-demanding Applications," in *Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference (MELECON '06)*, pp. 57–60, Malaga, Spain, May 2006.
- [14] P. Echeverría, J. L. Ayala, and M. López-Vallejo, "Leakage energy reduction in banked content addressable memories," in *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS '06)*, pp. 1196–1199, Nice, France, December 2006.
- [15] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. N. Mudge, "Drowsy caches: simple techniques for reducing leakage power," in *Proceedings of the 29th Annual International Symposium on Computer Architecture (ISCA '02)*, pp. 148–157, Anchorage, Alaska, USA, May 2002.
- [16] N. S. Kim, K. Flautner, D. Blaauw, and T. N. Mudge, "Single-VDD and single-VT super-drowsy techniques for low-leakage high-performance instruction caches," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '04)*, pp. 54–57, Newport, Calif, USA, August 2004.
- [17] <http://www.eas.asu.edu/ptm/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

