Eva-CAM: A Circuit/Architecture-Level Evaluation Tool for General Content Addressable Memories

Liu Liu[1], Mohammad Mehdi Sharifi[1], Ramin Rajaei[1], Arman Kazemi[1],
Kai Ni[2], Xunzhao Yin[3], Michael Niemier[1], and Xiaobo Sharon Hu[1]
[1]University of Notre Dame, USA, [2]Rochester Institute of Technology, USA, [3]Zhejiang University, China

***Abstract*—Content addressable memories (CAMs), a special-purpose in-memory computing (IMC) unit, support parallel searches directly in memory. There is growing interest in CAMs for data-intensive applications such as machine learning and bioinformatics. The design space for CAMs is rapidly expanding. In addition to traditional ternary CAMs (TCAMs), analog CAM (ACAM) and multi-bit CAM (MCAM) designs based on various non-volatile memory (NVM) devices have been recently introduced and may offer higher density, better energy efficiency, and non-volatility. Furthermore, aside from the widely-used exact match based search, CAM-based approximate matches have been proposed to further extend the utility of CAMs to new application spaces. For this memory architecture, evaluating different CAM design options for a given application is becoming more challenging. This paper presents Eva-CAM, a circuit/architecture-level modeling and evaluation tool for CAMs. Eva-CAM supports TCAM, ACAM, and MCAM designs implemented in non-volatile memories, for both exact and approximate match types. It also allows for the exploration of CAM array structures and sensing circuits. Eva-CAM has been validated with HSPICE simulation results and chip measurements. A comprehensive case study is described for FeFET CAM design space exploration.**

I. Introduction

Content addressable memories (CAMs), an in-memory computing (IMC) unit, support parallel search over stored entries directly in the memory array, and are promising candidates for addressing processor-memory bottlenecks. Besides being widely used in network routers and caches with high associativity, CAMs have been employed in a variety of emerging applications, such as hyperdimensional computing, one/few-shot learning [1], etc. In recent years, non-volatile memory (NVM) based CAMs, i.e., NV-CAMs [1]–[8], have become attractive due to their compact designs, high energy-efficiency, and non-volatility. The CAM design space is quickly growing. **First**, CAMs can be implemented in various device technologies with diverse circuit designs, such as CMOS, resistive RAM (RRAM) [2], [6], spin-transfer torque MRAM (STT-MRAM) [9], phase change memory (PCM) [3], ferroelectric FETs (FeFETs) [4], [5], [7], and floating-gate MOSFETs (Flash) [8]. **Second**, emerging analog CAMs (ACAMs) [6], [7] and multi-bit CAMs (MCAMs) [5], [8] have recently been proposed to achieve higher density and lower search energy. **Third**, emerging applications motivate the exploration of additional CAM-supported search functions. CAMs are mostly used for exact (EX) match search, while for some applications (especially in the machine learning domain), alternative match types including best (BE) match [5] and threshold (TH) match are extremely beneficial.

Circuit/array-level evaluation efforts for CAMs are essential given the rapid development of CAM designs and their various applications. While SPICE-based circuit simulations are accurate, they are time-consuming, and can only be used to evaluate very small CAM arrays. Alternatively, end-to-end evaluations (from devices to applications) are critical for emerging IMC designs, where system/application-level evaluation tools rely on circuit/architecture data, such as area, latency, and energy, to estimate the performance of specific tasks. Existing modeling tools can only support a subset of memory technologies or memory structures. For example, NVsim [10] can model NVM based caches or RAM, but they do not support CAMs. Though NVsim-CAM [11] provides NVM-based TCAMs modeling, it does not consider three-terminal devices, such as FeFETs, emerging CAM designs, such as ACAMs and MCAMs, or match types beyond EX-match.

In this paper, we present Eva-CAM, a circuit/architectural-level evaluation tool for general NV-CAMs. Eva-CAM leverages the basic structure of NVsim-CAM but significantly extends the NVsim-CAM functionality. Its capabilities include:

- Support for both exact and approximate match types.
- Support for both analog CAM and multi-bit CAM designs besides TCAM designs.
- Support for both two-terminal and three-terminal NVM devices, including FeFETs.

To the best of our knowledge, Eva-CAM is the first circuit/architecture-level evaluation tool that comprehensively considers the large CAM design space. Eva-CAM is validated against several state-of-the-art fabricated CAM chips and SPICE simulations and shows less than 20% error. We have also validated the new functionalities of Eva-CAM by using the FeFET CAMs to evaluate the aforementioned CAM design types, as well as search functions.

II. Background

Below, we summarize emerging CAM designs and CAM types, and review related work.

A. Emerging ACAM/MCAM designs

NVMs can be divided into two-terminal devices (e.g., RRAM, STT-MRAM, and PCM) and three-terminal devices (e.g., FeFET, Flash). Based on NVMs, a variety of new NV-CAMs have been proposed recently [2]–[9]. Specifically, ACAMs and MCAMs have been shown to achieve higher array density [5]–[8]. MCAMs [5] store contiguous non-overlapping, predefined value ranges. Search inputs are also constrained to these redefined ranges. An ACAM can store and search for analog values within a continuous value range [6]. Each ACAM

978-3-9819263-6-1/DATE22/©2022 EDAA

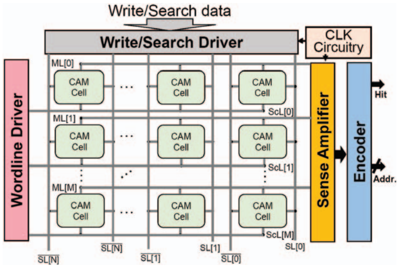

Fig. 1. Architecture of a general $M \times N$ CAM array

cell stores the upper and lower bound of a range using the analog programming capabilities of NVMs. A match occurs if the query range is the same as the stored range.

B. Exact match and approximate match

CAMs compare an input query against an array of stored entries and return the address of the matching entry or row. According to the matching degree (i.e., number of mismatched elements) of the returned entry, a match type can be classified into exact (EX) match or approximate match; an approximate match can be further divided into best (BE) match and threshold (TH) match. The match function can be modeled by the behavior of the machine line (ML) that we will discuss later. Let an input query element and a stored element be Q_j and C_{ij}, where i, j represents the row and column index, respectively. Assume that the query and the stored content are N-dimensional entries and M vectors are stored in an array. In the match state, ML is denoted as 1, and the distance function as $\sum_j d(C_{ij}, Q_j)$. The match types can be formally defined as follows:

- EX-match: $ML_i = 1$ if $C_{ij} == Q_j$ for all $j \in N$; else $ML_i = 0$.
- BE-match: $ML_i = 1$ if $\min \sum_{j=1}^{N} d(C_{ij}, Q_j)$ for all $i \in N$, $ML_i = 1$; else $ML_i = 0$.
- TH-match: $ML_i = 1$ if $\sum_{j=1}^{N} d(C_{ij}, Q_j) \leq k$; else $ML_i = 0$. k is a threshold value.

C. NV-CAM modeling and related work

NV-CAM designs are diverse due to various physical characteristics of NVM devices and CAM cell circuit designs. However, some common modeling concepts are applicable. The resistance and capacitance of CAM cells are key parameters for determining the stored states, search latency and energy consumption. Generally, the search latency can be estimated by time constant $\tau = R \times C$. The NVM devices and access transistors all contribute to the resistance and capacitance of an NV-CAM cell. Unlike two-terminal NVM devices which are modeled as resistance devices, three-terminal NVM devices also need to consider the capacitance. Additionally, the CAM cells in a row share a ML in a NOR-type connection and the ML is sensed by a sense amplifier (SA) (Fig. 1).

NVsim-CAM [11] is the first circuit-level simulation tool to model TCAM arrays. It provides an efficient way to model a diverse set of existing TCAM designs with peripherals support. However, NVsim-CAM only supports TCAM structures and does not consider emerging ACAM/MCAM designs and the approximate match types.

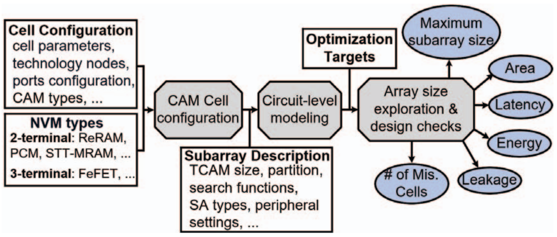
Cell Configuration
cell parameters, technology nodes, ports configuration, CAM types, ...
NVM types
2-terminal: ReRAM, PCM, STT-MRAM, ...
3-terminal: FeFET, ...
CAM Cell configuration
Circuit-level modeling
Subarray Description
TCAM size, partition, search functions, SA types, peripheral settings, ...
Optimization Targets
Array size exploration & design checks
Maximum subarray size
Area
Latency
Energy
Leakage
# of Mis. Cells

Fig. 2. High-level overview of the Eva-CAM framework.

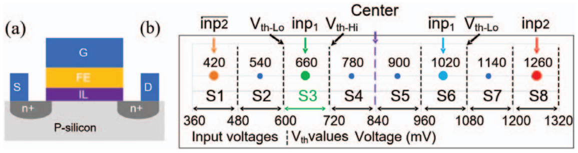

Fig. 3. (a) FeFET structure and (b) the states of the 3-bit MCAM cell denoted with S1 to S8, its 8 inputs in their corresponding states.

III. Eva-CAM Framework

Eva-CAM follows the general hierarchical CAM architecture including banks, mats, and subarrays [10]. Similar to NVsim-CAM, at the subarray level, aside from basic components, optional peripherals can be selected, including accumulators, buffers, priority encoders, etc. Based on this, Eva-CAM projects area, latency, energy and leakage of the CAM design. Additionally, it evaluates the largest achievable array size, the largest tolerable distance for approximate match. (i.e. the number of mismatched cells for TCAM array based on a Hamming distance function). Fig. 2 presents a high-level overview of the Eva-CAM framework. It describes the major evaluation stages of Eva-CAM (grey boxes), user-defined input (white boxes), and the outputs (blue circles). Eva-CAM shares the similar basic circuit modeling approach used in NVsim and NVsim-CAM, and we refer the reader to [10], [11] for more details.

Below, we present the details of Eva-CAM, including (i) TCAM, ACAM and MCAM cell modeling; and (ii) modeling of the exact and approximate match types, as well as the methodologies for exploring the achievable array size based on the sense margins of ML.

A. CAM cell modeling

Eva-CAM supports TCAM, ACAM, and MCAM modeling. TCAM cells use the low and high resistance states to store logical “0s” and “1s”, and two voltage levels are employed. ACAMs and MCAMs use multiple resistance states and search/write voltage. This directly impacts the latency and energy of CAM arrays. In MCAMs, each logical state, S_1 to S_n, is associated with a specific, non-overlapping resistance range defined by its upper and lower bound. The search voltage uses n levels to match the n states. An ACAM also uses resistance ranges defined by lower/upper bounds for stored values. However, an ACAM’s search voltage can be any value between the lowest/highest voltage. Eva-CAM provides interfaces designed for ACAMs and MCAMs. With said interfaces, the user specifies the CAM type and corresponding resistance value of each state. The search/write voltages on each port corresponding to each state must be specified.

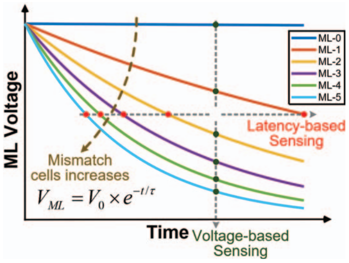

Fig. 4. ML discharge behavior is captured by the ML voltage vs. time curves.

Eva-CAM also provides interfaces for modeling three-terminal NVM devices, e.g., FeFETs. For three-terminal NVM devices, besides ON/OFF resistance values, the capacitance of NVM devices also needs to be considered. To the best of our knowledge, there is no FeFET small-signal model reported. Since the device structure of a FeFET is similar to the underlying MOSFET except for the additional FE layer (Fig. 3), we derive an equivalent thickness of the FE layer based on its capacitive behavior, and estimate the gate capacitance of the FeFET by scaling the dielectric thickness in the MOSFET capacitance model embedded in Eva-CAM. The drain capacitance of an FeFET is assumed to be the same as the underlying MOSFET. We will further improve the accuracy of the FeFET capacitance model when the related parameter values become available. For other three-terminal NVM devices such as flash, we can similarly estimate their capacitance values.

B. *Match types and achievable array size*

Eva-CAM supports three match types: EX-match, BE-match, and TH-match. We explain the circuit-level modeling approach for these match types in TCAMs using the Hamming distance function. Eva-CAM models three match types through the ML discharge behavior, as shown in Fig. 4. Two different sensing schemes can be adopted. The latency sensing scheme samples the ML latency at a specific voltage, while the voltage sensing scheme samples ML voltage at a specific time. A lower ML voltage or smaller ML latency indicates more mismatched cells.

One key concept is the *sense margin (SM) of ML*, which is the difference in sensed voltage (or latency) for two MLs with the different number of matching cells. Quantifying the worst-case SMs depends on the match types. For EX-match, the SM is defined based on differentiating the match case from a one-cell mismatch. For TCAM, one-cell mismatch implies a one-bit mismatch while for ACAM/MCAM, this implies a one-resistance state mismatch case, i.e., ML-0 and ML-1 in Fig. 4. We use ML-k to indicate entries with k mismatched cells.

BE-match has more strict sensing requirements on the MLs, and the definition of ML's SM is more involved. Assuming that an entry with x mismatched cells is the BE-match entry for a given input query, SA needs to distinguish ML-x from all other MLs with more than x mismatched cells. In the worst case, there may exist an entry with $(x + 1)$ mismatched cells, i.e., ML-$(x + 1)$. In this worst case, the SM must be large enough to differentiate ML-x from ML-$(x + 1)$. We use ML-2 in Fig. 4 as an example. If ML-2 is the best match case against all stored contents, the worst-case SM must differentiate ML-2 and ML-3. As shown in Fig. 4, as the number of mismatched cells increases (i.e., larger x), the ML discharge rate decreases while the latency/voltage differences between adjacent MLs decreases. That is, the worst-case SM becomes smaller and more difficult to be sensed. When the worst-case SM of ML is smaller than the SM of the selected SAs (SA's SM is defined as the smallest voltage/current difference detectable by the SA), the SA would not be able to differentiate the MLs with the corresponding SM. Based on the SM of a given SA, Eva-CAM can determine the maximum tolerable number of mismatched cells for a BE-match. For TH-match, the evaluation of the worst-case SM of ML is similar to BE-match.

For a given CAM design, the achievable array size is a key consideration for architecture-level design, and impacts area, delay, and energy. However, CAM array designs must consider another factor, i.e., the SM of ML, especially for approximate match types. The relatively small ON/OFF resistance ratios of NVM devices limit the SM of ML [3] and the number of cells on a ML. Eva-CAM can be used to determine the maximum number of cells on a ML (i.e., the number of columns) in a CAM subarray based on a SM check, as explained in III-B. Besides array size, Eva-CAM can also be used to determine the maximum identifiable number of mismatched cells in the approximate match case. For BE/TH-match, if the number of mismatched cells or mismatch states exceed certain values, the SA would not be able to exactly differentiate the ML(s) of the BE-match or TH-match case from the MLs with one more mismatched cell. Eva-CAM uses the circuit model and the user-specified/default SM values to determine the maximum tolerable mismatched cells in an array for approximate matches.

IV. Evaluation

We first present the validation results against fabricated NV-CAM chips. Three state-of-the-art TCAM chips based on RRAM, PCM, and MRAM [2], [3], [9], respectively, are used in the validation. FoM considered includes area, search latency and search energy. Note that due to the limited availability of measurement data, not all types of FoM are included. For the RRAM TCAM chip in [2], we only consider the area of the RRAM array and peripherals according to die photo, and the digital logic is estimated directly by the die photo due to lack of details. The results are summarized in Table I achieving 1.0% to 11.6% error rate, which is acceptable for chip-level validation. The errors mainly stem from the lack of information, e.g., certain low-level data and design details, and unavailable in-house technologies.

We also compare projected FoM by Eva-CAM with HSPICE simulations through a case study for FeFET CAMs. We consider FeFET TCAM/MCAM designs, achievable array size exploration for approximate match, and ML latency. The FeFET device parameters are obtained from HSPICE model [12]. We first validate Eva-CAM in modeling the two FeFET TCAM design proposed in [4] at the 45nm technology node. We include the same peripherals as [4] and validate the EX-match function with a 64×64 array size. The validation results are summarized in Table II, and exhibit errors of less than 6%. We next validate the FeFET MCAM design proposed in [5] for

TABLE I
THE VALIDATION FOR STATE-OF-THE-ART NV-CAM CHIPS.

References	FoMs	Actual	Eva-CAM	Error
RRAM 2T2R 40nm [2]	Area* (um^2)	98000	86600	-11.6%
	Search Latency (ns)	≥ 5	2-4.4	–
	Search Energy(pJ)	270	268.5	-1.0%
PCM 2T2R 90nm [3]	Area (um^2)	–	–	–
	Search Latency (ns)	1.9	2.1	9.4%
	Search Energy(pJ)	–	–	–
MRAM 4T2R 90nm [9]	Area (um^2)	17200	18270	6.2%
	Search Latency (ps)	2.5	2.72	8.6%
	Search Energy(pJ)	–	–	–

$*$ The actual area only includes RRAM array and peripherals.

TABLE II
VALIDATION FOR 2 FEFET TCAM ARRAYS [4].

FoMs		Actual	Eva-CAM	Error
Area	Subarray Area (um^2)	–	3274	–
Timing	Search Latency (ps)	350	345	-1.3%
	Write Latency (ns)	>10	10.2	
Dyn. Energy	Search Energy(pJ)	1.5	1.48	-1.3%
	Write Energy(pJ) (for a row)	0.1	0.1	–

EX-match. Though the MCAM cell structure is the same as TCAM [4], the MCAM cell can be programmed to eight logical states; and search and write schemes are different from [4], as shown in Fig. 3(a). We simulate a 1×64 MCAM at the 22nm technology node. The simulated ML latency values for the one-cell mismatch case with varying distances are depicted from 0 to 3, as the red triangles in Fig. 6. The Eva-CAM projected ML latency of MCAM (shown by the red curve in Fig. 6 incurs about 1% error on average compared with the simulation results. The projected total conductance of 64 MCAM cells is also provided as the blue curve in 6(a).

We have also used Eva-CAM to explore the impact of the array size on ML latency and evaluate the SM of ML for approximate match types. The studies are based on 2FeFET TCAM arrays implemented at the 45nm technology node. First, we simulate three TCAM arrays of size 16 bits, 32 bits, and 64 bits. The ML latency and the SM of ML for these three TCAM arrays are collected for the different numbers of mismatched cells from 1 to 5. The simulation results are shown as triangles in Fig. 5. The Eva-CAM projection results are depicted as the curves in Fig. 5(a). Furthermore, Eva-CAM evaluates the SM of ML based on the latency sensing scheme as shown in Fig. 5(b). The results in 5 show that the projected FoMs by Eva-CAM match the detailed HSPICE simulation data well.

V. CONCLUSION

This paper introduces Eva-CAM, a circuit/architecture-level evaluation tool for a variety of CAM designs. Eva-CAM supports CAM cells storing different types of data (ternary, analog, and multi-bits), different CAM types (EX-match, BE-match, and TH-match), and CAMs built with either two- or three-terminal NVM devices as well as CMOS transistors. Validations against both measured data from fabricated chips and simulation data from detailed SPICE models show that Eva-CAM incurs tolerable errors for area, latency, and energy projections. Eva-CAM can be a powerful tool for comparing CAM designs and CAM design space exploration.

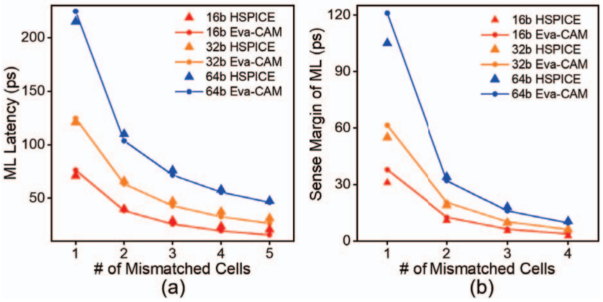

Fig. 5. (a) ML latency; and (b) sense margin of ML based on 16-bit, 32-bit, and 64-bit 2FeFET TCAM arrays, as the number of mismatched cells varys from 1 to 5.

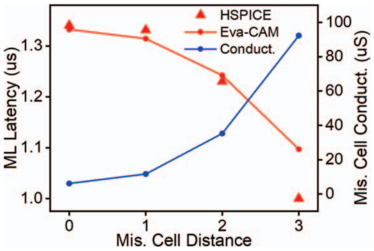

Fig. 6. ML latency and conductance vs. mismatched cell distance between the stored word and query, The distance is obtained by changing the state of singe MCAM cell from S1 to S4.

ACKNOWLEDGEMENT

This work was supported in part by ASCENT, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. We thank Drs. Shuangchen Li and Haitong Li for the insightful discussions. Xunzhao Yin is the corresponding author.

REFERENCES

[1] K. Ni *et al.*, “Ferroelectric ternary content-addressable memory for one-shot learning,” *Nature Electronics*, vol. 2, no. 11, pp. 521–529, Nov. 2019.
[2] H. Li *et al.*, “SAPIENS: A 64-kb RRAM-Based Non-Volatile Associative Memory for One-Shot Learning and Inference at the Edge,” *TED*, pp. 1–7, 2021.
[3] J. Li *et al.*, “1 Mb 0.41 µm² 2T-2R Cell Nonvolatile TCAM With Two-Bit Encoding and Clocked Self-Referenced Sensing,” *JSSC*, vol. 49, no. 4, pp. 896–907, Apr. 2014.
[4] X. Yin *et al.*, “An Ultra-Dense 2FeFET TCAM Design Based on a Multi-Domain FeFET Model,” *TCAS-II*, vol. 66, no. 9, pp. 1577–1581, Sep. 2019.
[5] A. Kazemi *et al.*, “In-Memory Nearest Neighbor Search with FeFET Multi-Bit Content-Addressable Memories,” in *DATE*, Feb. 2021, pp. 1084–1089.
[6] C. Li *et al.*, “Analog content-addressable memories with memristors,” *Nature Communication*, vol. 11, no. 1, p. 1638, Apr. 2020.
[7] X. Yin *et al.*, “FeCAM: A Universal Compact Digital and Analog Content Addressable Memory Using Ferroelectric,” *TED*, vol. 67, no. 7, pp. 2785–2792, Jul. 2020.
[8] A. Kazemi *et al.*, “A Flash-Based Multi-Bit Content-Addressable Memory with Euclidean Squared Distance,” in *ISLPED*, Jul. 2021, pp. 1–6.
[9] S. Matsunaga *et al.*, “A 3.14 um^2 4T-2MTJ-cell fully parallel TCAM based on nonvolatile logic-in-memory architecture,” in *VLSIC*, Jun. 2012, pp. 44–45.
[10] X. Dong *et al.*, “Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory,” *TCAD*, vol. 31, no. 7, pp. 994–1007, 2012.
[11] S. Li *et al.*, “NVSim-CAM: A circuit-level simulator for emerging nonvolatile memory based Content-Addressable Memory,” in *ICCAD*, 2016, pp. 1–7.
[12] K. Ni *et al.*, “A Circuit Compatible Accurate Compact Model for Ferroelectric-FETs,” in *VLSI*, Jun. 2018, pp. 131–132.