NVSim-CAM: A Circuit-Level Simulator for Emerging Nonvolatile Memory based Content-Addressable Memory

Shuangchen Li[1], Liu Liu[1], Peng Gu[1], Cong Xu[2], and Yuan Xie[1] *
University of California, Santa Barbara[1] Hewlett Packard Labs[2]
{shuangchenli, yuanxie}@ece.ucsb.edu

ABSTRACT

Ternary Content-Addressable Memory (TCAM) is widely used in networking routers, fully associative caches, search engines, etc. While the conventional SRAM-based TCAM suffers from the poor scalability, the emerging nonvolatile memories (NVM, i.e., MRAM, PCM, and ReRAM) bring evolution for the TCAM design. It effectively reduces the cell size, and makes significant energy reduction and scalability improvement. New applications such as associative processors/accelerators are facilitated by the emergence of the nonvolatile TCAM (nvTCAM). However, nvTCAM design is challenging. In addition to the emerging device's uncertainty, the nvTCAM cell structure is so diverse that it results in a design space too large to explore manually. To tackle these challenges, we propose a circuit-level model and develop a simulation tool, NVSim-CAM, which helps researchers to make early design decisions, and to evaluate device/circuit innovations. The tool is validated by HSPICE simulations and data from fabricated chips. We also present a case study to illustrate how NVSim-CAM benefits the nvTCAM design. In the case study, we propose a novel 3D vertical ReRAM based TCAM cell, the 3DvTCAM. We project the advantages/disadvantages and explore the design space for the proposed cell with NVSim-CAM.

1. INTRODUCTION

Ternary Content-Addressable Memories (TCAMs) are used for a wide variety of applications, such as associative caches, networking routers, and search engines. TCAMs provide fast match/mismatch responses for in-memory content searching. Conventionally, TCAMs are implemented by SRAMs with 16 transistors per cell. The large cell area and the corresponding large power consumption result in poor scalability. However, the emergence of the nonvolatile memory (NVM) based TCAM (nvTCAM) offers an alternative to overcome the challenge. The emerging NVMs, i.e., Magnetoresistive RAM (MRAM) [22], Phase-Changing RAM (PCM) [16], and Resistive RAM (ReRAM) [19], provide small cell area, nonvolatility [17], and zero standby power consumption [20]. Consequently, it not only provides denser memory designs with higher power efficiency, but also makes evolution for the TCAM design: a new generation of the nvTCAM with significant area reduction, low power consumption, better scalability, and instant-on/off features.

The nvTCAM has an even boarder influence. It can also pave ways for corresponding architecture innovations, which otherwise, are impossible with conventional SRAM-TCAMs. For example, thanks to the nvTCAM's low power consumption, Imani *et al.* [11–15] has proposed a nvTCAM-based approximate computing engine bypassing the energy hungry GPGPU data path. SRAM-TCAM is not competent due to large leakage power. Similarly, taking advantages of the nvTCAM's high density feature, Ipek *et al.* [9] has proposed a nvTCAM-based accelerator for data intensive applications. SRAM-TCAM is not adoptable due to the poor scalability. These work has shown the trend that as the nvTCAM keeps developing, there will be more edge-cutting techniques that call for the architecture/system design rethinking.

In order to keep pace with the ever-changing nvTCAM technology, a circuit-level model and a simulation tool are essential. Previous work set foot in modeling either the SRAM-TCAM or the emerging NVM, but they are not adoptable for the nvTCAM modeling. CACTI [24] and its variants have modeled SRAM-based fully associate cache, but no emerging technologies are supported. Sherwood *et al.* [2] has proposed a power model for TCAMs, but again, they only focused on SRAM-TCAMs. On the other hand, NVSim [26] is widely used for emerging NVM performance, area, and power evaluation, but the support for nvTCAM is not yet developed. Most recently, Chen *et al.* [4] has made comprehensive comparisons and design space explorations among three types of MRAM-TCAM cells. However, they did not provide a universal model and simulation platform, either.

In this paper, for the first time, we develop a universal simulation tool for nvTCAMs, named NVSim-CAM. A following case study presents this tool's competency of early stage projection and design space exploration, with a novel 3D vertical ReRAM based nvTCAM design. Our specific contributions in this paper are listed as follows,

- We develop a circuit-level model of nvTCAM which provides full-support for the indispensable diversity and flexibility of the nvTCAM design given their many possible choices of cell structures and circuit optimizations. We implement our model on a simulator frame-

* This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award number DE-SC0013553 with disclaimer at http://seal.ece.ucsb.edu/doe/. It was also supported in part by NSF 1533933, 1461698, 1500848, and 1213052.

ICCAD '16, November 07-10, 2016, Austin, TX, USA

 ISBN 978-1-4503-4466-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2966986.2967059

work NVSim with heavy modifications of its code base.
- We validate our model with fabricated nvTCAM prototypes, and the results show that we can achieve 3.5% error on average for several chips with different designs. We also demonstrate the competency of the tool in exploring a huge design space of nvTCAM at an early design phase.
- We propose a novel and extremely high-density TCAM design based on the low-cost 3D vertical ReRAM. We use NVSim-CAM to evaluate the design and demonstrate 234× higher density than the state-of-the-art design. We then project the superiority and identify the limitation of 3DvTCAM based on our evaluations with the tool.

2. BACKGROUND

In this session, we briefly introduce the necessary NVM basics and the principle working mechanism of TCAMs.

Although the working mechanisms and the features vary, all of the emerging NVMs (i.e., MRAM, PCM, and ReRAM) share common basics [29]: They all use cell resistance (R_H or R_L) to represent logic "0" or "1". Typically, the NVM storage cell is built with the 1T1R (one access transistor with one resistive cell) structure, where it has a wordline (WL) controlling the access transistor, a bitline (BL) for data sensing, and a source line (SL) to provide the write current/voltage.

The TCAM cells are connected together with the matchline (ML). In order to describe three states ("0", "1", and "x"), the cell usually contains two single-level cells or one MLC. The querying data is transferred to each corresponding cells through the searchline (SrL) during search operations. By properly encoding the storing data and the querying data, the cell can output a logical "0" for mismatch and logic "1" for match. The ML performs an overall logic "AND" operation for all the cell matching result on it: If any mismatch shows up ("0"), the final result shows a mismatch.

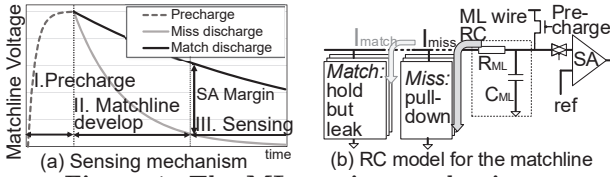

Figure 1: The ML sensing machanism.

Fig. 1 (a) shows the sensing mechanism for (nv)TCAM. There are three phases. In the first phase, the ML is charged to a high voltage. In the second phase, the SrLs are activated to evaluate matching. The ML starts to discharge. In the last phase, the ML voltage difference between match and mismatch is large enough for a sense amplifier (SA) to sense. We denote the minimal ML voltage difference between a match and a mismatch as the sense margin. Fig. 1 (b) shows the circuit model. A mismatch cell generates I_{miss} to discharge the ML. This current is much larger than I_{match}, which is the leaking discharge current from a match cell.

3. OVERVIEW

This section shows an overview of the micro-architecture of a general nvTCAM and NVSim-CAM's framework.

3.1 The bank organization and components

Fig. 2 (left) shows the bank-level architecture for a nvTCAM (please find preliminaries from NVSim [26]). As the lower level micro-block, Mats within a bank are connected with H-tree (the bus-like connection is also supported). In order to support a large query word size, the word is able to be partitioned among the Mats. In this situation, the search operations inside each Mat work simultaneously, and their results are merged (by AND logic) at the joint point of the H-tree routing.

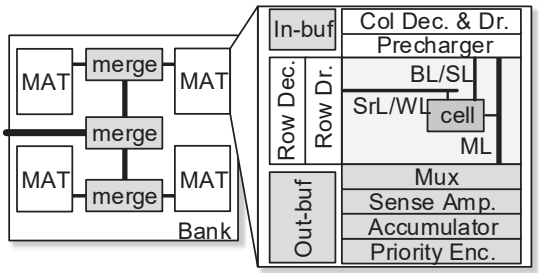

Figure 2: The bank organization (left) and components within a Mat (right). Glossary: Decoder (Dec.), Driver (Dr.), Encoder (Enc.)

Fig. 2 (right) shows the components that build a nvTCAM Mat. NVSim-CAM is based on NVSim [26] but there are plenty of differences between the TCAM and normal memories, as marked with dark colors. Besides the WL/BL/SL, there are also SrLs and MLs in nvTCAMs. There are also unique components in nvTCAM, including the accumulator and the priority encoder.

3.2 The framework of NVSim-CAM

The design knobs include data organizations, technologies, component settings, and cell designs. The detailed configurable items are listed in Table 1. These design knobs are either fixed for a certain design projection, or input as a range for DSEs. The optimization objective and design constraints are supported. The output of NVSim-CAM is the performance/power/area parameters of the best design that meets the design specification and constraints.

Organization:	**Component:**
Bank/Mat size: H-tree, partition	SA types (vol./cur.)
SA sharing: Mux, local/global	Buf., Acc., Priority enc.
Bit serial width (if applicable)	Drivers opt. target
Cell description:	
Cell type (Diode/NMOS/Direct); Device parameters	
Description for each port: transistor size, V/I in every op.	

Table 1: NVSim-CAM's input and design knobs.

4. NVSim-CAM DEVELOPMENT

In this section, we show the development of NVSim-CAM. We first show how NVSim-CAM models different cell structures. Then, in order to improve the simulation precision, we propose the customized-SA based modeling. In the end, we validate NVSim-CAM with fabricated nvTCAM prototypes.

4.1 Description of various cell structures

The nvTCAM cell structure design is flexible. For example, from simple cells of 2T2R [16] or 3T1R [19], to the complex cell of 6T2R [21], there are reasonable designs with fabricated prototypes. They are adopted according to different design targets. To model the cell diversity, we focus on three aspects: the cell's impact on peripheral circuits, the intra-cell currents, and the cell's impact on ML development.

4.1.1 Cell's impacts on peripheral circuits

A cell can have multiple ports connected to the row/column wires (WL/SrL/etc). For example, the 3T1R cell [6] is connected to three row wires and three column wires. These wires determine the corresponding row/column drivers and the multiplex's design. To capture these impacts, we need a description of each port, including the connected transistor's size and the connected wire's width. By these descriptions, the RC model of the row/column wire is established. Then, with the description of the voltage and/or current that applied to this port during search/write operations, the maximal current on the wire is calculated, and hence we have the parameters for driver/mux design (RC load and maximal current).

4.1.2 Intra-cell currents

Two kinds of intra-cell currents are considered in NVSim-CAM. First, there could be direct current (DC) in the cells during search, for example, in the 4T2R cell [10]. The DC needs to be counted for power consumption. Second, although based on NVM, the nvTCAM cell still suffers from leakage, which needs to be included in the leakage power.

4.1.3 Cell's impacts on ML development

The ML development is essential for calculating sensing latency and checking sense margin constraints. To model various cell structures' impact on ML, we classify all those cell designs into three categories: Diode-access, NMOS-access, and Direct-access, as shown in Fig. 3. To support various cell structure designs, the circuit in the dashed box is flexible with any design.

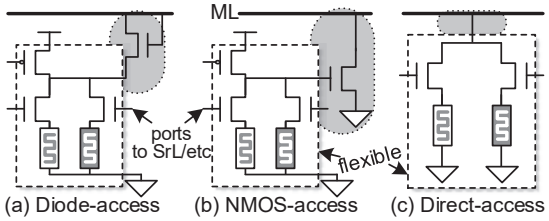

Figure 3: Three types of nvTCAM cell structures.

Diode-access nvTCAM. A diode (implemented by NMOS) is used to connect the cells to the ML. While mismatching, a low voltage is generated to the diode and turns it on, setting a path to discharge the ML. For matches, a higher-than-ML voltage is outputted to the diode and turns it off. The 4T2R cell structure [22] in Fig. 3 (a) is an example. A match operation connects one or two storage cells with R_H to the circuit, generating a high enough voltage to turn the diode off. If mismatch happens, it connects one storage cell with R_L to the diode with a low voltage that turns it on, and hence discharges the ML.

NMOS-access nvTCAM. A pull-down NMOS is used to connect the cell circuit to the ML. A mismatch/match generates an high/low voltage to the NMOS's gate. It further discharges the ML or keeps its voltage high. The 4T2R cell structure [10] in Fig. 3 (b) is an example,which is similar with the Diode-access example.

Direct-access nvTCAM. The cells are directly connected to the ML with the access transistors. A mismatch connects cells with R_L to the ML and generate large discharge current. A match connects at least one cell with large resistance R_H to the ML, results in a leaking current but small enough to keep the ML voltage high for a long time. The 2T2R cell structure [16] in Fig. 3 (c) is an example.

Different categories result in different discharging paths. The ML delay is able to be calculated accordingly. We show the detailed calculation in Section 4.2.

4.2 ML delay modeling

We focus on ML developing phase in the three phases for sensing shown in Fig. 1 (a). For the other two phases, previous methodologies are adapted for modeling. For example in the precharge phase, since no cell is turned on in that phase, the precharge latency and power are independent with store/search data pattern, and is able to be calculated with ML's RC parameters. We extend the BL delay model in NVSim [26] for ML modeling. Recall that the voltage dividing bitline delay model is described as follows,

$$\tau = R_M C_{\mathrm{ML}} + \frac{1}{2} R_{\mathrm{ML}} C_{\mathrm{ML}}, \quad V_{\mathrm{sense}} = V_s \cdot e^{-\frac{t_{\mathrm{ML}}}{\tau}}, \qquad (1)$$

where R_M is the equivalent cell resistance, R_{ML} and C_{ML} are the ML's RC parameters. V_s is the precharged voltage, and V_{sense} sets the timing that the SA is enabled to sense. t_{ML} is the ML development delay.

The nvTCAM ML delay modeling is different from conventional memory's BL model in two aspects. First, besides calculating the delay, we also need to check the sensing margin constraint for nvTCAM. To this end, we calculate the mismatch that provides the worst case for sensing (with largest R_M^{miss}, usually the case that only one cell misses) in Equation (2). In this worst case, the ML discharge is slower than any other mismatch cases, and the ML developed voltage is closest to the match case. Therefore, we calculate the sense margin with it. t_{ML} is defined as the latency that the ML discharges to V_{sense} in the worst mismatch, as follows,

$$V_{\mathrm{miss}}^{\mathrm{max}} = V_{\mathrm{sense}} = V_s \cdot e^{-\frac{t_{\mathrm{ML}}}{\tau_{\mathrm{miss}}}}, \quad V_{\mathrm{match}} = V_s \cdot e^{-\frac{t_{\mathrm{ML}}}{\tau_{\mathrm{match}}}}, \quad (2)$$

where the ML voltage at t_{ML} in the match situation is denoted as V_{match}. The voltage difference between the match (V_{match}) and the worst mismatch ($V_{\mathrm{miss}}^{\mathrm{max}}$) should be larger than the sensing margin.

Second, different from normal memories, the nvTCAM R_M calculation depends on both the cell categories and the match results. For Diode/NMOS-access cells, R_M in match and worst-case mismatch situations are calculated as follows,

$$R_M^{\mathrm{match}} = \frac{R_{\mathrm{off}}}{N}, \quad R_M^{\mathrm{miss}} = R_{\mathrm{on}} || \frac{R_{\mathrm{off}}}{N-1}, \qquad (3)$$

where R_{on} and R_{off} are the on/off equivalent resistance of a diode/NMOS transistor. N denotes the number of cells that are activated on the ML at one time. Note that different from NMOS-access cells, Diode-access cells have a fast voltage drop before the ML discharges.

The ML model for the three types of nvTCAMs are then validated with HSPICE simulations, as shown in Fig. 4. It shows that the model (lines) is able to represent the real delay from the HSPICE (dots) precisely for each category of the nvTCAM cells.

4.3 Customized SA

The support for customized SAs is essential. Unlike the well developed SRAM's SA, the NVM (especially nvTCAM) SA designs are very complex and flexible. They are usually designed with special consideration of a particular device or cell structure. Moreover, both of the SA performance and area take an important portion in the overall chip evaluation.

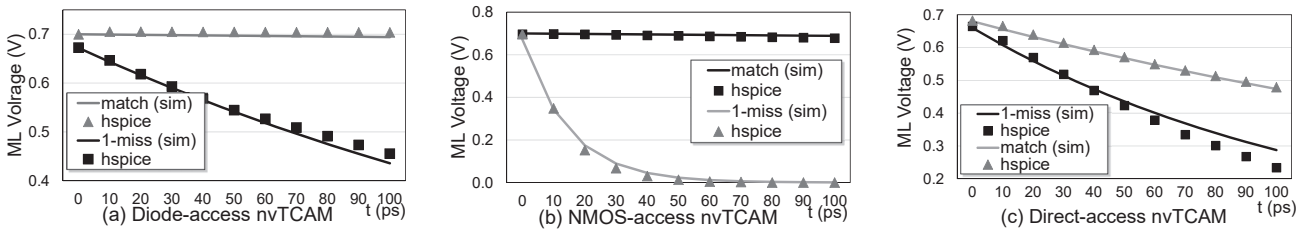

Figure 4: Validating ML model in NVSim-CAM with HSPICE simulations for all three cell categories.

Even though the SA design is so flexible and important, existing tools only support fixed SA designs with constant parameters. NVSim [26] supports three types of SA designs but still not sufficient. To cope with this problem, NVSim-CAM supports customized SAs, which provides an interface to import any SA parameters (either from HSPICE simulation or literature) into the tool, when a more precise result is expected.

Figure 5: Customized SA in NVSim-CAM's.

Fig. 5 shows the framework. First, we pre-run the NVSim-CAM to extract the model of the ML for the following customized SA design. The user could design their SA by HSPICE or simply gather data from literature, as long as the custom SA's latency and power parameters are given back to the NVSim-CAM. For the area, if it is not available, a netlist with transistor size is also acceptable, in which case, the NVSim-CAM estimates the layout footprint [26].

By applying the custom SA design, the validation error (in the case of Table 2) reduces from 8% to 4.3%.

4.4 Other components

Some of the nvTCAM designs require extra peripheral circuits. NVSim-CAM provides the bit serial accumulator and the priority encoder. We describe the modeling as follows.

Accumulator. The accumulator is used to support bit serial search, which activates only parts of the ML each time and searches the whole word serially. The accumulator gathers the partial matching results and generates the final result for the serial searching. The accumulator circuit includes a register and an AND logic [23]. It also contains a power gating transistor to gate the ML whenever a mismatch happens during the serial searching to save power. By applying the bit serial searching, the number of match leakage path is reduced and hence the sense margin is improved.

Priority Encoder. The priority encoder only generates the lowest address of all the matching entries. The encoder is implemented to facilitate particular applications [25] or to reduce global wiring for the results. The priority encoder is made of a multiple match resolver (MMR) block and a normal encoder block. NVSim-CAM models the MMRs according to a look ahead 3-level folding design [8]. The basic MMR block (8 entries) is based on dynamic logic [8], and there are two look ahead signals. The basic blocks are serially connected and the look ahead signals are connected in a hierarchical folding style so that the overall latency is reduced from $O(N)$ to $O(\log N)$. We validate NVSim-CAM's priority encoder model with a 256-bit encoder with the fabricated result [8]. It shows that NVSim-CAM achieves 12.96% latency error and 16.18% power error[1].

[1]We scale the 600nm data to 32nm for comparison and hence result in unexpected error.

	Metric	*Actual*	*Projected*	*Error*
MRAM, 4T2R, 32-bit, 64-entry, Diode, 90nm [22]	Area (μm^2):	17118.95$^+$	16378.50	-4.3%
	L_{Search} (ns):	2.50	2.571	2.6%
	E_{Search} (pJ):	–	4.606	–
ReRAM, 4T2R, 32-bit, 128-entry, NMOS, 180nm [10]	Area (μm^2):	–	83157.52	–
	L_{Search} (ns):	1.20	1.14	-5.34%
	E_{Search} (pJ):	–	51.661	–
PCM, 2T2R, 64-bit, 2048-entry, 8-mat, 90nm, Direct [16]	Area (μm^2):	–*	34636.95	–
	L_{Search} (ns):	1.90	1.85	-2.5%
	E_{Search} (pJ):	–	144.42	–

Table 2: Validations. ($^+$**Blank area is excluded.** ***Test and reference circuit is embedded.)**

4.5 Validation with fabricated prototypes

In order to validate NVSim-CAM, we compare the projected result from NVSim-CAM against the fabricated prototypes. Table 2 evaluates the all three types of cells. Non-volatile technologies of MRAM, PCM, and ReRAM are all examined by those validations. Despite of the limited data we achieve from the literature, NVSim-CAM manages to achieve an estimation with error around 5%. Even though the error rate is acceptable, we would like to point out that each of those chips is fabricated by a technology with in-house parameters, but NVSim-CAM is based on PTM. The errors from the technology library could be a major source for the error. Therefore, the significance of this tool lies in relative comparisons, such as DSE shown in the next section.

5. DESIGN SPACE EXPLORATION

In this section, we perform DSE with NVSim-CAM to show its competency.

5.1 Exploring search word size's impacts

Fig. 6 shows the search word size's impact on the ML delay and the sense margin. For the configurations, we use 14nm FinFET technology [1] and ReRAM [10]. We set the the ML length the same as the search word size. Three cell structures with 4T2R [22], 4T2R [10], and 2T2R [16] are selected as representatives for the three nvTCAM cell structure categories. The observations are presented as follows.

Diode-access nvTCAM cells provide **support for long search words, but suffer from large search latency**. It is able to minimize the leaking current (I_{match}) if matching, results in the large sense margin and hence good scalability. However, the current when it mismatches (I_{miss}) is also small, and it causes a longer ML discharge delay. Fig. 6 shows the trend. Its ML delay is larger than other types in most of the cases, and the sense margin almost stays the same, as the word size scales up.

NMOS-access nvTCAM provides **support for long words, and fast search for short words**. This is because it have both small I_{match} and large I_{miss}. However, the downside is the large cell area. As shown in Fig. 6, the ML delay gets larger when word gets longer. It is because the large

	Area Opt.	L_{Search} Opt.	E_{Search} Opt.	E_{Write} Opt.	Leakage Opt.
Area (μm^2):	**5746.551**	33327.997	21879.867	27913.421	5746.551
Search Latency (ns)	16.736	**1.332**	5.201	72.992	1121.232
Search Energy (fJ/bit)	27.878	25.533	**23.772**	599.912	1976.749
Write Energy (nJ)	102.992	106.26	168.25	**95.142**	102.992
Leakage (μW)	47.507	2089.383	418.904	63.778	**47.507**
Cell Sturcture	2T2R-direct	3T1R-nmos	3T1R-nmos	4T2R-diode	2T2R-direct
MLC	No	Yes	Yes	No	No
Bit Serial	64-bit	–	–	–	1-bit
Driver Opt.	area-opt	latency-opt	area-opt	area-opt	area-opt

Table 3: Design Space Exploration for 14nm ReRAM based 128-bit 64-entry nvTCAM.

cell area causes a long ML and hence a large RC delay. For sense margin, it is even better than the Diode-access nvTCAMs, because it does not contain a fast voltage drop before the ML development.

Direct-access nvTCAM offers best scalability but **cannot afford a long search word**. It benefits from a considerable small cell area. However, the leaking current I_{match} is large, and hence it turns to be a limitation for a long search word. Fig. 6 shows that when the word size is larger than 32-bit, the sense margin is below 80mV, which is difficult for the SA design. Therefore, Mat partition or bit serial searching is required to support longer search words.

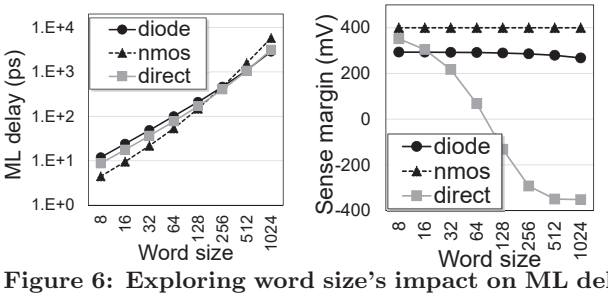

Figure 6: Exploring word size's impact on ML delay and sense margin for three types of cell structures.

5.2 Exploring technology's impact

We study the impact of technology scaling on the performance of nvTCAM design. For the configurations, we use the same cells in Section 5.1. The bank contains a single Mat with 64-bit word and 256 entries. We implement the FinFET technology models in NVSim-CAM and the device parameters are extracted from PTM [1, 30]. We observe from NVSim-CAM that, overall the latency and energy result scales well with the technology development. However, we notice that **for Direct-access nvTCAMs, the technology scaling hurts the sense margin**. Beyond the 22nm technology, 64-bit word size hits the 80mV sense margin constraint, and hence is difficult for sensing. It brings up the scaling challenges for the area-efficient Direct-access cells. Data encoding schemes [16] or ECC could be the feasible solutions.

5.3 An overall DSE example

In order to show the nvTCAM DSE is nontrivial, we show an overall DSE example with NVSim-CAM in Fig. 3. For the configurations, we set the data organization as a single Mat with 128-bit word and 64 entries. The cell library contains three cells from the three categories, i.e., the 4T2R [22] Diode-access cell, the 3T1R [6] NMOS-access cell, and the 2T2R [16] Direct-access cell. We optimize the design for different targets such as area and search latency.

We observe from the DSE result that **there is no such a design choice that wins for every design target**. For the area optimization, the Direct-access cell providing small cell area is adopted. However, the Direct-access cell suffers from word size scalability challenge. It has to apply bit serial search scheme to achieve the 128-bit search word requirement. As a result, the area optimized design scarifies search latency for a smaller area. For the search latency and energy optimizations, the NMOS-access cell wins, thanks to the large I_{miss} it provides. However, it is not optimistic considering write energy, since the 3T1R cell has to use the MLC feature, which is more difficult to write. In the end for the leakage optimization, the Direct-access cell is better for two reasons: first, its intra-cell leakage is much smaller; second, the smaller area leads to shorter wires, and hence smaller drivers with smaller leakage.

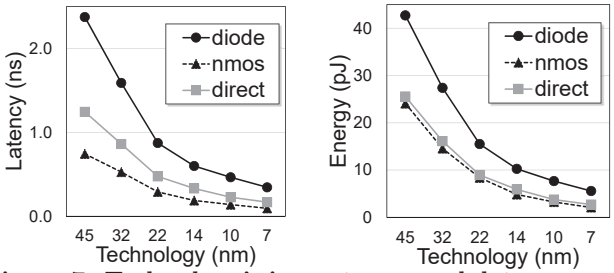

Figure 7: Technology's impact on search latency and energy for three types of cell structures.

6. A CASE STUDY: 3DvTCAM

In this section, we propose 3DvTCAM, a novel nvTCAM based on the 3D vertical ReRAM (3DVReRAM) as a case study showing that how the NVSim-CAM tool helps to project the emerging device developments.

6.1 The 3D Vertical ReRAM TCAM cell

We briefly introduce the background, since the proposed 3DvTCAM is based on the 3DVReRAM [3, 27]. As shown in Fig. 8 (a), the 3DVReRAM structure is similar with 3D-NAND. Each horizontal plane makes the WL. The vertical pillars with metal oxide around the central metal pillar provide the metal-oxide-metal sandwich structure when contacting with the horizontal planes, and hence build ReRAM cells (a clearer sectional view is shown in Fig. 8 (b)). The pillar is connected with an access transistor controlled by SL, and then a row of pillars are connected to the BL. The signal from a certain pillar is sent to the SA for read. Even though the 3DVReRAM is an ultra dense multi-layer transistor-less design that provides extreme cost efficiency [28], it faces the sneak path problem. Selector based device [5] is proposed to solve the problem by increasing the non-linearity.

We propose the 3DvTCAM based on the 3DVReRAM, as shown in Fig. 8 (a) and (c). The horizontal plane is used as the SrL, and the BL is used as the ML. A word is stored vertically along a vertical pillar, and a bit is built up with a couple of cells encoding the "0/1/x" states. For searching, a SrL inputs (0, 1/2V)/(1/2V,0). The SL signal only activates

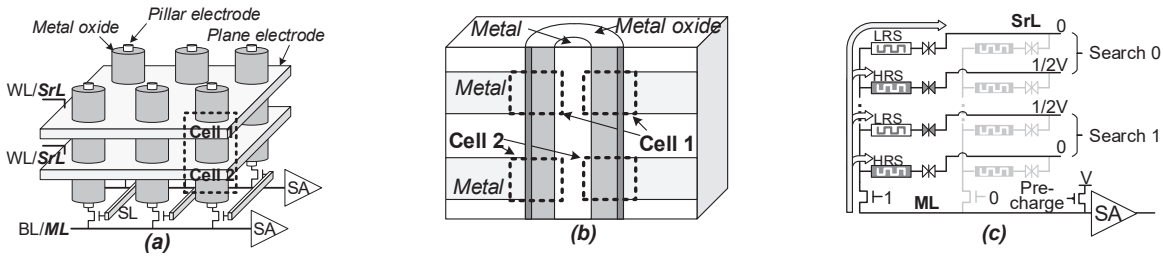

Figure 8: (a) 3D Vertical ReRAM based TCAM structure. (b) Cell section. (c) Circuit model when searching.

one of the pillars connected to the ML at one time. The first column and first row in Table 4 shows how 3DvTCAM searches and stores "0/1/x". For normal read and write, it remains the same as 3DVReRAM.

Cell Content	1 (L, H)	0 (H, L)	X (H, H)
Search 0 (0, 0.5V)	$\mathbf{I_{0L}+I_{1H}}$	$I_{0H}+I_{1L}$	$I_{0H}+I_{1H}$
Search 1 (0.5V, 0)	$I_{1L}+I_{0H}$	$\mathbf{I_{1H}+I_{0L}}$	$I_{1H}+I_{0H}$
Search X (0.5V, 0.5V)	$I_{1L}+I_{1H}$	$I_{1H}+I_{1L}$	$I_{1H}+I_{1H}$

Table 4: Cell current during ML developing (darker means larger).

In order to make sure the 3DvTCAM work, we prefer the I_{miss} to be large enough to discharge the ML, and the I_{match} small enough to hold the ML voltage and provide enough sense margins. We show the discharge current with all combination of SrL voltage and storage cell resistance as follows,

$$I_{0L}=\frac{V}{R_L},\ I_{0H}=\frac{V}{R_H},\ I_{1L}=\frac{V}{2K_rR_L},\ I_{1H}=\frac{V}{2K_rR_H},$$
$$K_r=\frac{R(\frac{1}{2}V_{\text{read}})}{R(V_{\text{read}})},\qquad I_{0L}\gg\max\{I_{0H},I_{1L},I_{1H}\},\qquad(4)$$

where I_{0L} and I_{1L} denotes the current with 0V or 1/2V input at SrL to a cell with R_L resistance, and K_r represents the cell nonlinearity. From the equation we observe that, if provided a large K_r and a large on/off resistance ratio, current I_{0L} will be much smaller than other possible currents and hence ensure the correctness of the TCAM. Table 4 shows the cell current of all possible combinations. A darker table cell represents a larger current. Based on this, we show the calculation of the equivalent cell resistance as follows,

$$R_M^{\text{match}}=\frac{2K_rR_L}{N}||\frac{R_H}{N},\quad R_M^{\text{miss}}=R_L||\frac{2K_rR_H}{2N-1}.\qquad(5)$$

Based on the resistance calculation and the ML model in Section 4.2, we show the ML discharge calculation for 3DvTCAM as follows,

$$V_o(t)=\frac{V_s(R_M+R_T)}{2R_{\frac{1}{2}V}}+(V_s-\frac{V_s(R_M+R_T)}{2R_{\frac{1}{2}V}})e^{-\frac{t}{\tau}}\qquad(6)$$

where $R_{\frac{1}{2}V}$ are the overall equivalent resistance connected to the SrLs with 1/2V as input. Different from other nvTCAM, the SrL the 1/2V input makes some of the discharge paths to the 1/2V instead of the ground. We then validate the model with HSPICE simulations in Fig. 9, where the lines represent results from NVSim-CAM and the points represent results from the HSPICE. The results show both mismatch and match scenarios with K_r as 20 and 500. It shows that our model fits well with the real data.

6.2 Exploring feature of the new cell

We project the advantages and disadvantages of the new cell with the help of 3DvTCAM in this subsection. For

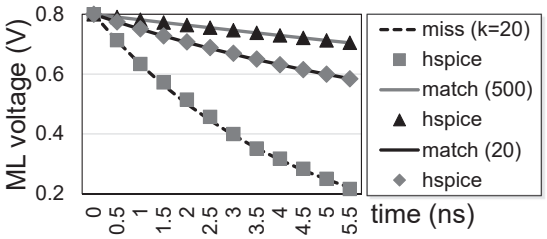

Figure 9: ML delay validation with HSPICE.

all the experiment, we apply 14nm FinFET technology and other configurations following the work from Cong *et al.* [27].

We have three observations after studying the impact of the number of layers and the nonlinearity factor K_r in Fig. 10. First, **more layers hurts the sense margin badly**. This is because a larger number of layers results in more discharge paths during match, and it makes the sense margin drop exponentially. Second, **a larger $\mathbf{K_r}$ helps** to provide smaller I_{match} and hence better sense margin. However, beyond $K_r=50$, increasing K_r barely enlarges the sense margin anymore. This observation shows that **aggressive nonlinearity device technology cannot completely overcome the layer scaling challenge**. Instead, we have to apply bit serial searching or Mat partition design to achieve a longer search word. Third, except for increasing the searching parallelism, **we do not have a motivation for a large number of layers**, from density point of view. Since the aspect ratio is fixed, more layers makes the array area larger, and hence longer wires and larger driver. Fig.10 shows that the density sweet point is 16 layers while only considering cell density and 64 layers for the overall density.

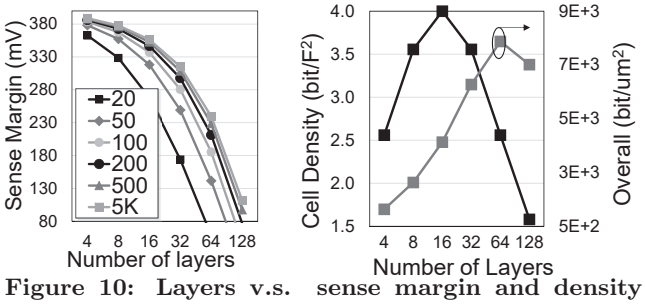

Figure 10: Layers v.s. sense margin and density (256×256 array).

We also explore the number of layer's impact on the overall search latency and power in Fig. 11. Although the number of layer (capacity) increases, the latency almost stays the same, which shows 3DvTCAM's **good scalability**. Also, the K_r's impact on latency is negligible. For the power, it doesn't change while layer (capacity) increases, either. The power per bit reduces exponentially.

We also compare the 3DvTCAM with conventional 2D nvTCAM design in Fig. 12. For configurations, we use 128-bit search word, 32 layer design with K_r=20. We take

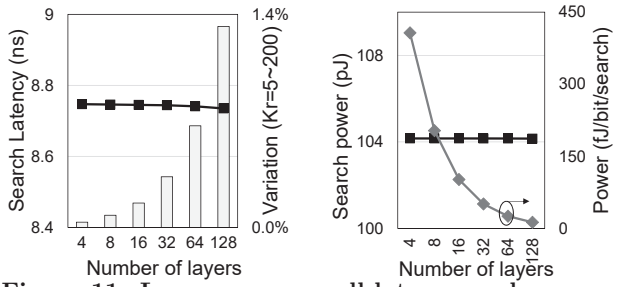

Figure 11: Layers v.s. overall latency and power.

the 4T2R NMOS-access cell [10] for the 2D baseline. For speed, **the 3DvTCAM is slower than a 2D nvTCAM** by ∼58% even when the capacity goes as large as 1GB. However, **3DvTCAM is more energy efficient**. since the power consumption results show ∼2.7× and ∼506× improvement than a 2D nvTCAM of 1MB and 1GB, respectively. That is owed to the good scalability of 3DvTCAM that the length of global wire increases such slower than that of the 2D case. For area comparison, **3DvTCAM provides a ultra dense solution** that saves up to ∼234× area, which is much larger than the 3D layer factor (32 layers). This is owed to the area reduction of the global wires and drivers.

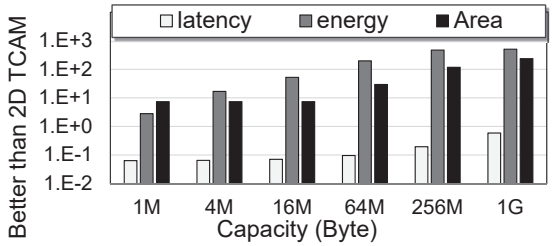

Figure 12: Compare 3DvTCAM with 2D design.

6.3 Discussion

We discuss the architecture-level innovation facilitated by the projection result from NVSim-CAM: a processing-in-storage architecture targeting at energy-efficient acceleration for DNA alignment algorithms. This is one more step from the NVM-based processing-in-memory ideas [7, 18]. In order to eliminate the unnecessary data movement, we can design the fast storage 3DVReRAM with part of that designed as 3DvTCAM. The search operation for hit sequence is then able to be performed inside the storage with the help of the TCAM. Hance the data movement is minimized that only useful data (a few hit sequences) is fetched to the processer.

7. CONCLUSION

In order to model and project the ever changing emerging NVM based TCAM design, we propose a circuit-level model and develop a simulation tool, NVSim-CAM. The tool is able to capture the flexibility of the nvTCAM design and is validated with both HSPICE simulations and fabricated prototypes. Based on NVSim-CAM, we perform the DSE for different types of nvTCAM cells. In order to show how NVSim-CAM helps for early stage projection of potential novel TCAM designs, we explore 3DvTCAM, a proposed 3D vertical ReRAM based TCAM, as a case study.

8. REFERENCES

[1] Predictive Technology Model. http://ptm.asu.edu/.

[2] B. Agrawal and T. Sherwood. Ternary CAM power and delay model: extensions and uses. *TVLSI*, 16(5):554–564, may 2008.

[3] I. G. Baek et al. Realization of vertical resistive memory (VRRAM) using cost effective 3D process. In *IEDM*, pages 31.8.1–31.8.4, dec 2011.

[4] I. Bayram and Y. Chen. NV-TCAM: Alternative interests and practices in NVM designs. In *NVMSA*, pages 1–6, aug 2014.

[5] E. Cha et al. Nanoscale (10nm) 3D vertical ReRAM and NbO2 threshold selector with TiN electrode. In *IEDM*, pages 10.5.1–10.5.4, dec 2013.

[6] M.-F. F. Chang et al. A 3T1R nonvolatile TCAM using MLC ReRAM with Sub-1ns search time. *JSSC*, 58:318–319, feb 2015.

[7] P. Chi et al. PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *ISCA*, volume 43, 2016.

[8] Chung-Hsun Huang et al. Design of high-performance CMOS priority encoders and incrementer/decrementers using multilevel lookahead and multilevel folding techniques. *JSSC*, 37(1):63–76, 2002.

[9] Q. Guo et al. AC-DIMM: associative computing with STT-MRAM. In *ISCA*, pages 189–200, 2013.

[10] L.-Y. Huang et al. ReRAM-based 4T2R nonvolatile TCAM with 7x NVM-stress reduction, and 4x improvement in speed-wordlength-capacity for normally-off instant-on filter-based search engines used in big-data processing. In *VLSIC*, pages 1–2, jun 2014.

[11] M. Imani et al. ACAM: Approximate computing based on adaptive associative memory with online learning. In *ISLPED*, 2016.

[12] M. Imani et al. Approximate computing using multiple-access single-charge associative memory. *TETC*, PP(99):1–1, 2016.

[13] M. Imani et al. Processing Acceleration with Resistive Memory-based Computation. In *Memsys*, 2016.

[14] M. Imani et al. ReMAM: low energy resistive multi-stage associative memory for energy efficient computing. In *ISQED*, pages 101–106, 2016.

[15] M. Imani et al. Resistive configurable associative memory for approximate computing. In *DATE*, pages 1327–1332, 2016.

[16] J. Li et al. 1 Mb 0.41 um2 2T-2R cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing. *JSSC*, 49(4):896–907, apr 2014.

[17] S. Li et al. Leveraging nonvolatility for architecture design with emerging NVM. In *NVMSA*, pages 1–5, 2015.

[18] S. Li et al. Pinatubo: a processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories. In *DAC*, page 173, 2016.

[19] C.-C. Lin et al. A 256b-Wordlength ReRAM-based TCAM with 1ns search-time and 14x improvement in wordlength-energyefficiency-density product using 2.5T1R cell. In *ISSCC*, pages 136–138, 2016.

[20] K. Ma et al. Nonvolatile processor architecture exploration for energy-harvesting applications. *IEEE Micro*, 35(5):32–40, 2015.

[21] S. Matsunaga et al. Fully parallel 6T-2MTJ nonvolatile TCAM with single-transistor-based self match-line discharge control. In *VLSIC*, pages 298–299, jun 2011.

[22] S. Matsunaga et al. A 3.14 um2 4T-2MTJ-cell fully parallel TCAM based on nonvolatile logic-in-memory architecture. In *VLSIC*, pages 44–45, jun 2012.

[23] S. Matsunaga et al. Implementation of a perpendicular MTJ-based read-disturb-tolerant 2T-2R nonvolatile TCAM based on a reversed current reading scheme. In *ASP-DAC*, pages 475–476, jan 2012.

[24] N. Muralimanohar et al. CACTI 6.0: A tool to model large caches. *HP Lab.*, pages 22–31, 2009.

[25] H.-J. Tsai et al. Energy-efficient non-volatile TCAM search engine design using priority-decision in memory technology for DPI. In *DAC*, pages 1–6, june 2015.

[26] Xiangyu Dong et al. NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *TCAD*, 31(7):994–1007, jul 2012.

[27] C. Xu et al. Architecting 3D vertical resistive memory for next-generation storage systems. In *ICCAD*, pages 55–62, nov 2014.

[28] C. Xu et al. Modeling and design analysis of 3D vertical resistive memory: A low cost cross-point architecture. In *ASP-DAC*, pages 825–830, jan 2014.

[29] J. Zhao et al. Memory and storage system design with nonvolatile memory technologies. *IPSJ*, 8:2–11, 2015.

[30] S. Zuloaga et al. Scaling 2-layer RRAM cross-point array towards 10 nm node: A device-circuit co-design. In *ISCAS*, pages 193–196, may 2015.