MASC: Ultra-Low Energy Multiple-Access Single-Charge TCAM for Approximate Computing

Mohsen Imani, Shruti Patil, Tajana S. Rosing
CSE Department, University of California San Diego, La Jolla, CA 92093, USA
{moimani, patil, tajana}@ucsd.edu

***Abstract*— Memory-based computing using associative memory has emerged as a promising solution to reduce the energy consumption of important classes of streaming applications such as multimedia by avoiding redundant computations. In associative memory, a set of frequent patterns that represent basic functions are pre-stored in ternary content addressable memory (TCAM) and reused. The primary limitation to using associative memory in modern parallel processors is the large search energy required by TCAMs. In TCAMs, all match rows, except hit rows, precharge and discharge in every search operation, resulting in high and undesirable energy consumption. In this paper, we propose a new multiple-access single-charge (MASC) TCAM architecture which is capable of searching TCAM contents multiple times with a single precharging cycle. In contrast to previous designs, the MASC TCAM keeps the match-line voltage of all miss-rows high and uses their charge for the next search operation, while only the hit rows discharge. We use a periodic refresh scheme to guarantee the accuracy of the search. We also implement a new type of approximate associative memory by setting longer refresh times for MASC TCAMs, which yields search results within 1-2 bit Hamming distances of the exact result. Our evaluation on AMD Southern Island GPU shows that using MASC associative memory can improve the average GPGPU energy efficiency by 36.6%, 40.2% and 39.4% for exact matching, selective 1-HD and 2-HD approximations respectively, with acceptable quality of service (PSNR>30dB). These energy savings are 1.8X and 1.6X higher than GPGPU using exact matching TCAM and approximation TCAM that uses voltage overscaling, respectively.**

I. Introduction

The massive computation needs of big data requires efficient parallel processors. There is a significant amount of redundant data when processing streaming applications such as multimedia, [1], [2]. The idea of associative memory was introduced to exploit this observation and decrease the number of redundant computation [3], [4], [6], [7], [8]. In hardware, associative memories are implemented as look up tables using ternary content addressable memories (TCAMs) [4]. However, TCAMs based on CMOS technology suffer from low density and high energy consumption compared to SRAMs [9], [5]. This energy limits the application of TCAMs to classification [10] and IP look-up [11]. To decrease the energy consumption on TCAM, voltage overscaling (VOS) has been applied on CMOS-based TCAMs [12], [13]. However, this increases the system error-rate due to process variations and timing errors.

Non-volatile memories (NVMs) present a new opportunity for efficient memory-based computation using low energy TCAMs [14], [15], [16]. Resistive RAM (ReRAM) and Spin Torque Transfer RAM (STT-RAM) are two types of fast and dense non-volatile memories based on memristor and magnetic tunneling junction (MTJ) devices [17][18]. Previous work has used NVMs to design fast and energy efficient TCAMs. However, the energy consumption of NVM-based TCAMs is still high because of high number of charges and discharges in TCAM lines at each search operation. To further reduce NVM-based energy consumption, VOS has been applied on memristive-based TCAM while accepting results within 1-2 bits Hamming distance between input and pre-stored TCAM patterns [6]. This aggressive voltage relaxation on the full TCAM bitline limits the TCAM size and degrades the computation quality of service below acceptable range.

Our goal in this paper is to design TCAMs such that the energy consumption is decreased even in wide TCAM sizes, while controlling the output quality. To that end, we propose an ultra-low energy multi-access single-cycle TCAM (MASC TCAM) which improves the energy consumption of the TCAM by performing multiple search operations after a single precharging cycle. In conventional TCAMs, the match-lines (MLs) of all TCAM rows precharge to VDD voltage. During the search operation, all MLs, except the hit rows (if any) discharge. This causes the TCAM to consume high energy independent of a hit or a miss. In contrast, MASC TCAM is designed with a complementary operation. It only discharges the voltage of hit rows. Consequently, majority of the miss rows of the TCAM retain their charge after the search operation for more search cycles without the need to precharge each time. The proposed MASC design uses encoding scheme to split the search operation in several short word-sizes so that they can be performed in parallel.

Instead of using voltage overscaling to decrease TCAM energy consumption, the MASC TCAM architecture varies the period of precharging cycles on selective TCAM blocks to decrease the energy and control the computational error. The selective implementation of approximate MASC allows the system to balance the TCAM energy savings and computational quality of service based on the running application. Our evaluations on AMD Southern Island GPU architecture using four image processing applications show that the proposed design improves the energy consumption of GPGPU by 36.6%, 40.2% and 39.4% on average for exact matching, 1 and 2 hamming distance approximation with acceptable peak signal-to-noise ratio (PSNR>30dB). This energy saving is 1.8X and 1.6X higher than conventional TCAM for exact matching and VOS approximation respectively.

II. Related Work

Associative memory in the form of a look-up table has been used with parallel streaming processors to avoid doing redundant computations [3], [4], [5], [6], [7]. In hardware, associative memories are implemented using TCAM blocks. CMOS-based TCAMs consist of two SRAM cells but their cost per bit is 8X more than SRAM [9]. High density and low energy consumption of NVMs such as ReRAM and STT-RAM improve the energy efficiency of memory based computation. ReRAMs have comparable read operation to SRAMs, but they

978-3-9815370-7-9/DATE16/ ©2016 EDAA

suffer from limited endurance (10^6-10^7 write operation), which degrades their lifetime [18]. On the other hand, STT-RAMs have fast read operation as well as high endurance (>10^{15}). However, the bi-directional write current and low ON/OFF ratio (~2) usually increases the area of the MTJ-based TCAM with respect to ReRAM-based TCAM cells [19], [20]. High endurance is necessary for TCAMs since they must be periodically updated. Our design addresses endurance issue by limiting the write stress only at the start of kernel execution.

Several previous works have used NVMs to design stable and efficient TCAMs [14], [21], [20], [22]. Li et al. [14] designed a 1Mb energy efficient 2T-2R TCAM which is 10X smaller than SRAM-based TCAM. Another 3T-1R TCAM structure has been introduced in [21], which can search the entire CAM array in less than 1ns with very low energy consumption. An efficient 2Kb 4T-2MTJ based TCAM cell is proposed in [22]. This cell is for standby-power-free TCAM and has 86% area reduction respect to SRAM-based TCAM. Hanyu, et al. in [20] introduced 5T-4MTJ TCAM cell which searches input data on cell complementary with very high sense margin. However, the energy consumption of NVM-based TCAMs is still high because of high number of charge and discharge cycles at each search operation [13], [7], [6]. Approximate TCAM using voltage overscaling is one method to decrease the search energy consumption of associative memories [23],[6],[30]. However, aggressive VOS implementation on the entire TCAM bitline limits the number of TCAM size to just a few number of rows and increases the computation quality of service below acceptable range.

In contrast to previous efforts, we design a new MASC TCAM which can perform multiple search operations with a single precharging cycle. In MASC TCAM the MLs discharge only in the case of a hit while all miss-rows stay charged after the search operation. The proposed design introduces selective hit-line precharging and long precharging refresh scheme to improve the search energy consumption of error-free MASC TCAM. Selective MASC block approximation balances processor energy consumption and quality of service using multiple precharging periods based on running application.

III. RESISTIVE ASSOCIATIVE MEMORY

Resistive associative memory consists of TCAM and resistive memory (1T-1R memory). The frequent input patterns and the related outputs are pre-stored in TCAM and resistive memory respectively. Hit on a TCAM block stops the computation by clock-gating the processor computation. Low energy consumption of the NVM-based associative memory enables application of these memories to query processing [24], search engine [25], text processing [26], image processing [27], pattern recognition [3], data mining [28] and image coding [29]. Several of these applications need large TCAMs with respect to word-size and number of rows to cover a variety of input operands. However, designing large TCAMs has following challenges:

- Due to finite ON/OFF resistance ratio of NVMs in TCAM structure, the reliable search operation can be done on TCAMs with short word-size. Large word-sizes increase the leakage current of TCAM lines such that a hit may be incorrectly considered as a miss. Further, the effect of process variations makes the TCAM more sensitive to word-size.
- A TCAM with high number of rows requires a large input buffer block to distribute the input signal among all rows. At large sizes, this buffer degrades the search delay and energy consumption of the TCAM.
- The primary factor that limits the number of TCAM rows and word-size is the TCAM search energy. The search energy of a large TCAM hides the advantage of using associative memory. In other words, there is a tradeoff between the processor and associative memory energy consumption effected by the TCAM size. A large TCAM consumes high energy which degrades the energy efficiency of the computation. But the high hit-rate of large TCAM decreases the percentage of the time that the processor is doing computation by using clock gating.

Any technique to reduce the energy consumption of associative memory will allow us to push the minimum energy point to larger TCAMs in order to improve TCAM hit rate and FPU energy efficiency. We explore design techniques that can enable high search energy savings with controllable quality of service. Our proposed design allows multiple search operations on a TCAM block with single precharging. The proposed design implements selective approximation on TCAM blocks using the novel technique of long precharging refresh time to adaptively balance the computational energy and quality of service.

IV. MASC TCAM ARCHITECTURE

A. Motivation

In this paper we propose a multiple-access single-charge TCAM which can perform multiple search operations with single precharging of MLs (see Figure 1). In conventional TCAM, all TCAM rows (MLs) precharge to VDD voltage. In search mode, if the input pattern has a mismatch with any pre-

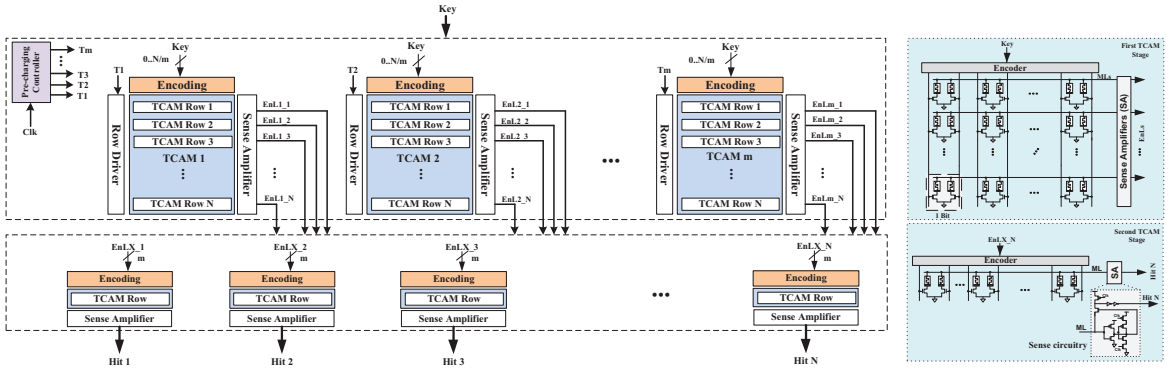


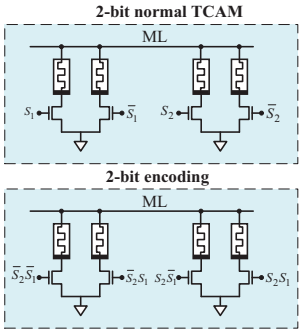
Figure 1. Proposed multiple-access single-charge TCAM architecture

stored TCAM rows, the ML starts discharging. In the case of a hit in the TCAM, the voltage on hit ML(s) stays high, while the other rows discharge to zero. For the next search operation we must precharge all TCAM rows again. This high energy consumption of precharging and discharging of MLs is the primary cause of the high TCAM search energy consumption.

To decrease the charge and discharge energy, we propose MASC TCAM which can perform search operations with extremely low energy consumption. The functionality of proposed cell is similar to conventional TCAM with the difference that in search operation, MLs discharge only when there is a match with the input pattern. In case of a mismatch, TCAM rows retain their precharge voltage. This allows us to use ML voltage of miss-rows to perform more search operations without another precharge cycle. After each search, we selectively precharge the hit MLs using a simple circuit. A complete refresh of MLs is performed after a specific number of cycles. This significantly reduces the energy consumption of the TCAM by decreasing the number of charges and discharges.

B. TCAM Cell and Encoding

We use an encoding technique [14] to design the MASC TCAM, as shown in Figure 2. Each TCAM cell has two memristor devices and two access transistors. The value of these memristors are pre-stored in the TCAM such that each block has only one low resistance (L). The tables of store and search operations of 2-bit TCAM with and without encoding schemes are shown in Figure 2. For the search operation, the input patterns are first transformed by the encoding block. This block activates one of the four possible combinations of the input signals ($\bar{S}_1\bar{S}_2$, $\bar{S}_1S_2$, $S_1\bar{S}_2$, S_1S_2) based on Figure 2. These signals activate only one access transistor per 2-bit encoding block. For a hit, the access transistor activates the memristor with low resistance and ML starts to discharge. In case of a mismatch, the access transistor related to just one of the high resistance (H) devices will be connected. This limits ML leakage currents to one transistor in each encoding block.



Store Data	Without Encoding	Two-bit Encoding
1 1	L H L H	H H H L
1 0	L H H L	H H L H
0 1	H L H L	H L H H
0 0	H L H L	L H H H
1 ×	L H H H	H H L L
0 ×	H L H H	L L H H
× 1	H H L H	H L H L
× 0	H H H L	L H L H
× ×	H H H H	L L L L

Search Data	Without Encoding	Two-bit encoding
1 1	0 1 0 1	0 0 0 1
1 0	0 1 1 0	0 0 1 0
0 1	1 0 0 1	0 1 0 0
0 0	1 0 1 0	1 0 0 0

Figure 2. TCAM with and without encoding

In conventional TCAM, the number of leaky cells depends on the length of bitline. For large word-sizes, the access transistors of many cells may be activated depending on the input pattern. This can discharge the ML unintentionally and yield incorrect search results. For efficient and reliable search in TCAM, sufficient margin must be present between the match and mismatch currents. The worst-case difference between the currents occurs when all cells are matched (no discharge), and when just one of the cells is unmatched. In contrast, in the proposed MASC TCAM, increasing the size of encoding blocks improves the TCAM sense margin and makes the search operation easier since in any encoding block size, the number of leaky cells from ML on a miss is limited to a single cell.

C. MASC TCAM architecture

The architecture of proposed MASC TCAM is shown in Figure 1. The MASC search operation is done in two stages; (i) to avoid long search delay and energy of large word-size TCAMs, we split the search to several short word-size TCAM searches. Each partial TCAM searches a part of the input data. In GPGPU with 32-bit search operation, we separated the bitline at 2:16 (sixteen 2-bit TCAMs), 4:8 TCAM (eight 4-bit TCAMs), and 8:4 TCAM (four 8-bit TCAMs) searches. This split sends 16, 8 and 4 output signals to the second TCAM stage (see Figure 1). (ii) The second stage logically *OR*s the *EnL* signals of the first stage TCAM using another TCAM stage. If the input pattern matches on the same row of all partial TCAMs, the input pattern is considered as a hit on that row. This requires a single TCAM cell and small encoder block that produces complement signals (e.g. $\overline{EnL1}\_1$, $\overline{EnL1}\_2$, …, $\overline{EnL1}\_m$ for *m*-bit encoding) to activate the first TCAM cell. In other words, the data in the second TCAM stage is fixed, and if the data matches in the same row of all partial TCAMs, a single cell is activated.

D. Refresh Schemes and MASC Approximation

We define refresh period as the number of search cycles that can be performed without precharging. This number depends on the word-size of the partial TCAM blocks. This precharging cycle is determined by the amount of the cell leakage through the ML in every search operation. In the proposed MASC TCAM, the best and worst case leakage scenarios are the same since there is always only one leaky cell for a miss independent to the block size. The ML voltages and average TCAM energy consumption (per cycle) for 8-bit encoding TCAM block is shown in Figure 3. To calculate this refresh cycle, we considered a 10% process variation on the resistance value, the size, and threshold voltage of access transistors. We call a refresh period as acceptable if it results in correct matches in 1000 Monte Carlo simulations. The 8-bit MASC TCAM can perform four consecutive search cycles without a complete precharging. With 2-bit and 4-bit MASC TCAM, the refresh period is 7-cycle and 5-cycles. The proposed technique reduces the search energy consumption of the TCAM significantly by reducing precharge requirement.

Performing more search operations than these refresh periods results in TCAM search error. Figure 4 shows the normalized ML voltage on TCAM with different precharging cycles at the last cycle of search. We set the search clock period as the maximum TCAM delay at the last period. We consider having mismatch in every search cycle. Our circuit level simulation on 8-bit TCAM shows that in order to have an error-free search operation we need the ML voltage higher than 850mV (see Figure 3). This means that for exact matching we can limit the period of precharging to 4-cycles. Measuring the average energy of a search operation after 4-cycles shows that the proposed design can achieve up to 3.2X lower search energy with respect to conventional TCAM design. Using longer precharging periods makes the search operation unstable and increases the probability of error. Our evaluations also show that ML voltages of about 775mV (6-cycle) and 650mV

(8-cycle) correspond to one and two bits Hamming distance in a TCAM search.

Another advantage of the proposed TCAM is its ability to control approximation by implementing long precharging cycles on selective TCAM blocks. In the proposed architecture, all TCAM blocks do *not* have the same effect on the result of computation. Applying long precharge cycle on the least significant bits has lower impact on the results of computation compared to most significant bits. The MASC TCAM allows us to implement 1-HD and 2-HD approximation on selective TCAM blocks using multiple precharging cycles. A simple precharging controller shown in Figure 1, sets the refresh time of each partial TCAM based on the running application and quality of service. We will show the effect of approximation on the GPGPU energy saving and output PSNR in section V.

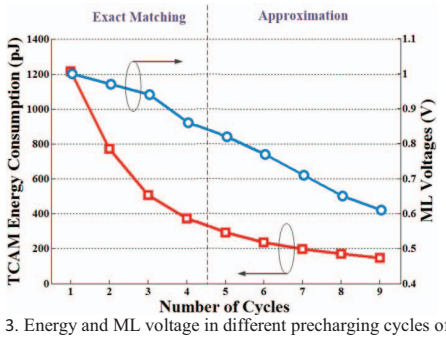


Figure 3. Energy and ML voltage in different precharging cycles of 8-bit TCAM

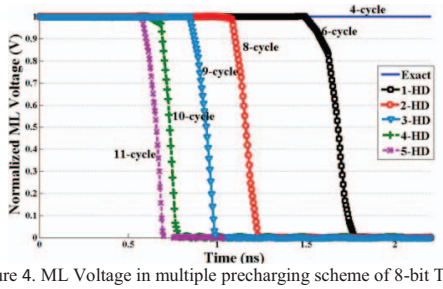


Figure 4. ML Voltage in multiple precharging scheme of 8-bit TCAM

V. EXPERIMENTAL RESULTS

A. Experimental Setup

We evaluate the MASC-based associative memory on AMD Southern Island GPU Radeon HD 7970 device, a recent GPU architecture. The image processing application has been adopted from AMD APP SDK v2.5 [31] in OpenCL to make it suitable for streaming applications. We use Multi2sim, a cycle accurate CPU-GPU simulator [32] and modified the kernel code to do profiling and run-time simulation. Four image processing applications *Sobel, Robert, Sharpen* and *Shift* are used to show the efficiency of the proposed MASC associative memory. The 6-stage balanced FPUs are designed using *Synopsys Design Compiler* [33] in 45-nm ASIC flow. Due to tradeoff between energy and delay, the FPUs are optimized for power based on different TCAM delay.

We extracted the frequent patterns for adder (ADD), multipliers (MUL), SQRT, multiply-accumulator (MAD) FPUs. In GPGPU computation, the FPU operations have a variety number of inputs. The ADD and MUL accept two 32-bit, SQRT one 32-bit and MAD three 32-bit input operands. Thus, the corresponding TCAMs require 64-bit, 32-bit and 96-bit word sizes respectively. The circuit level simulation of TCAM design is done using HSPICE simulator on 45-nm technology. Our framework executes in two modes: training mode (design time) and execution mode (run time). In training mode, we profile 10% of the input data for each applications using Caltech 101 computer vision dataset [34]. For each application, we train the system for 10 random images and run the system for 100 different images. During profiling, the system counts and ranks all operations based on the frequency of their occurrence for each FPUs. Then the host code chooses the most frequent patterns for each FPUs to fill the TCAM and 1T-1R memory. At run-time, the TCAM values are fixed and the system checks the input patterns in FPUs and associative memory simultaneously. Any hit in TCAM stops FPU computation and activates the corresponding row of 1T-1R memory to read the computation result.

B. Block Size

The MASC structure consists of two TCAM search stages. The second level TCAM is fast, area and energy efficient, such that it does not have a major impact on system efficiency. In the first stage of MASC TCAM, the search operation can be done in 2-bit, 4-bit or 8-bit encoding granularities. The size of encoding block shows a tradeoff between TCAM area and energy consumption; (i) MASC TCAM with small encoding blocks (e.g. MASC 2:16) improves the total area efficiency of the TCAM. In contrast, large blocks such as 4-bit and 8-bit encoding need more cells to design a TCAM. (ii) Large encoding blocks improve TCAM sense margin, since in the MASC structure the number of leaky cells from ML is always one independent of the block size. (iii) Additionally, TCAMs with large encoding blocks reduce the number of partial matches on the TCAMs, further improving the energy efficiency. This is because hits in partial TCAMs increase the number of undesired charging and precharging.

We define *effective TCAM states (ETS)* as the ratio of number of TCAM rows to the number of available states at each MASC granularity, which indicates the probability of hit in the TCAM. For 32-row 2:16 TCAM, *ETS* is $32 / 2^2 = 8$. This number decreases to 2 and 1/8 for 4-bit and 8-bit encoding TCAM blocks respectively. Lower *ETS* indicates that the system has lower probability of undesired hits. Our evaluation on four applications shows that MASC with *ETS* > 8 degrades TCAM search energy consumption to below that of the conventional TCAM. This limits the number of stages in 32-row and 64-row TCAMs, to 15-stages and 8-stages respectively. Thus, stability and energy efficiency of large encoding blocks come at the expense of TCAM area. However, the area overhead of resistive associative memories is negligible compared to the FPU area in GPU architecture. Further, if the associative memory area becomes an important design parameter, the MASC TCAM can be designed with smaller blocks (MASC 2:16). For energy efficient associative memory, employing MASC 8:4 results in the best energy and sense margin.

Figure 5 shows the delay of a 96-bit TCAM at different sizes. In the proposed TCAM, the search operation is performed using several parallel partial TCAM searches. At small sizes, the conventional TCAM has lower delay than MASC TCAM because of its single-stage search operation.

However, at large sizes, the search operation of the conventional TCAM slows down due to the precharging delay. A highly partitioned MASC uses short and parallel precharging cycles, which offsets the overhead of multiple stages. This results in lower MASC delay especially in highly partitioned TCAMs at large TCAM sizes.

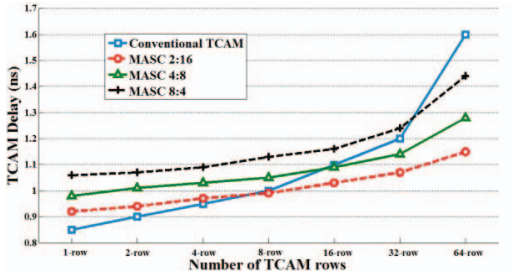


Figure 5. TCAM latencies in different TCAM size

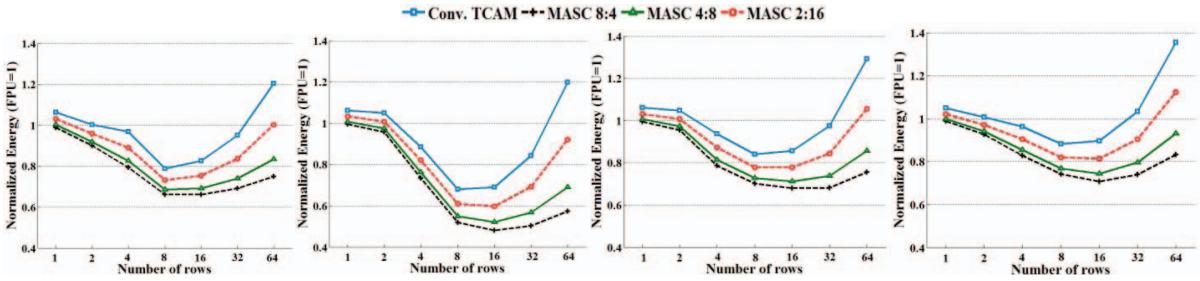
C. Multi-Search Associative Memory

To integrate the proposed design as an associative memory, we use 1T-1R memory structure along with MASC TCAM to pre-store the computational results for various operands. The normalized energy consumption of the GPGPU at different TCAM sizes is shown in Figure 6. The FPU energy is calculated based on the measured TCAM delay at each size. There are two factors that affect the total GPGPU energy consumption:

- **FPU energy:** In large TCAMs, higher hit-rate increases the percentage of the time that the processor is in clock-gate mode. Small TCAMs clearly benefit from doubling TCAM size where new TCAM row can pre-store a pattern with a high percentage of hit. But large TCAM already covers most of the frequent patterns so that adding new lines does not have a major impact on hit-rate improvement and energy efficiency. In addition, the delay of large TCAM allows the design compiler to optimize the FPU energy consumption.
- **TCAM energy:** High search energy consumption of a large TCAM limits the energy efficiency of the GPGPU processing. Therefore, in conventional associative memory, increasing the TCAM size to larger than 8-rows does not improve the hit-rate (and hence the FPU energy) enough to compensate for the large TCAM energy. Considering these factors, the minimum energy point of GPGPU using conventional associative memory occurs at 8-row TCAM size (see Figure 6).

D. Approximate Multi-search Associative memory

Approximation in MASC TCAM is defined by the period of the precharging. With a period of 4-cycles, an 8-bit MASC TCAM performs error-free searches. Increasing the refresh period of 8-bit TCAM to 6-cycle and 8-cycle creates one and two bits Hamming distance respectively. Our framework implements approximation in TCAM starting from the lower level TCAM blocks, since the mismatch on these blocks has lower effect on the computation result compared to error on most significant bits. TABLE 1 lists the maximum number of MASC blocks in 1-HD and 2-HD approximation, and hit-rate improvement compared to exact matching for different applications such that the output PSNR does not drop below 30dB. The system is able to apply the approximation on *m* lower blocks of each MASC TCAM based on the running applications. The small controller block in Figure 1 sends appropriate signals to row driver of each TCAM block to set the MLs refresh periods based on the running applications.

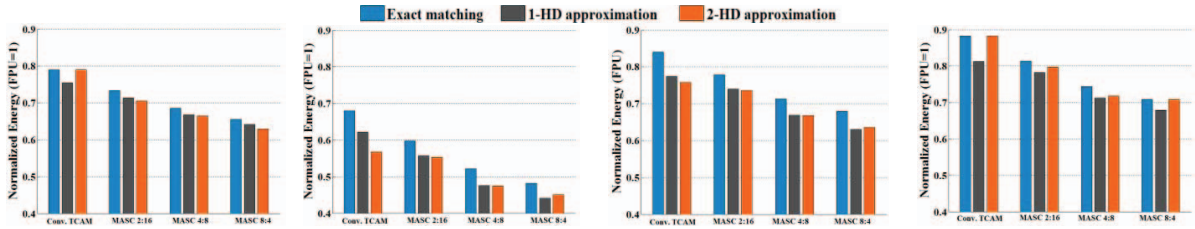


(a) Sobel (b) Robert (c) Sharpen (d) Shift

Figure 6. Normalized energy consumption of GPU in in different TCAM sizes.

TABLE 1. APPROXIMATION ON SELECTIVE TCAM BLOCKS

Application Type		*Sobel*			*Robert*			*Sharpen*			*Shift*		
TCAM Configuration		***2:16***	***4:8***	***8:4***	***2:16***	***4:8***	***8:4***	***2:16***	***4:8***	***8:4***	***2:16***	***4:8***	***8:4***
1-HD	# of blocks	3 (18.7%)	2 (25%)	1 (25%)	3 (18.7%)	2 (25%)	2 (50%)	3 (18.7%)	3 (37.5%)	2 (50%)	2 (18.7%)	2 (25%)	1 (25%)
	PSNR	31.0dB	32.4dB	41.5dB	33.4dB	34.7dB	36.4dB	35.5dB	32.8dB	36.3dB	34.9dB	32.3dB	40.3dB
	Hit-rate improvement	9.6%	8.3%	5.1%	7.7%	9.3%	13.5%	7.8%	10.5%	12.2%	6.8%	7.6%	4.8%
2-HD	# of blocks	2 (12.5%)	1 (12.5%)	1 (25%)	2 (12.5%)	1 (12.5%)	1 (25%)	2 (12.5%)	2 (25%)	1 (25%)	1 (6.2%)	1 (12.5%)	0 (0%)
	PSNR	30.3dB	34.6dB	32.2dB	31.2dB	42.6dB	39.1dB	32.4dB	30.4dB	39.5dB	37.5dB	35.2dB	Original
	Hit-rate improvement	3.2%	4.5%	8.2%	7.9%	6.4%	9.1%	6.8%	8.1%	9.3%	2.8%	4.4%	0%



(a) Sobel (b) Robert (c) Sharpen (d) Shift

Figure 7. Normalized Energy consumption of system in 1-HD and 2-HD approximation.

There is a trade-off between the quality of service and GPGPU energy consumption. As Table 1 shows, approximations of TCAM increases TCAM hit-rate based on the number and size of blocks in approximate mode, and the depth of approximation (1-HD or 2-HD). Implementing refresh time relaxation on a large number of bitlines improves energy efficiency by increasing the system hit-rate and reducing search energy, however this improvement is achieved at the expense of quality of service. In MASC 8:4, using long refresh periods on one block relaxes 25% of the entire TCAM bitline, while approximating a MASC 2:16 block relaxes 6.2% of the entire TCAM bitline. This wide range of relaxation significantly improves the system hit-rate and GPGPU energy saving. In addition, the system energy savings increase with TCAM sizes because large TCAMs with approximation benefit more from hit-rate improvement compared to small TCAMs.

Figure 7 shows the energy improvement of the GPGPU using conventional and MASC TCAMs for exact matching, 1-HD and 2-HD implementations for different applications. Our results show that the total energy consumption of the GPGPU with MASC 8:4 MASC 4:8 and MASC 2:16 decreases by 40.2%, 36.8% and 30.1% (39.4%, 37.8% and 30.2%) on average for 1-HD (2-HD) implementation respectively, using the number of blocks listed in TABLE 1 and ensuring acceptable quality of service (PSNR>30dB). This indicates that a GPGPU using proposed MASC achieves 1.8X higher energy savings for exact matching with respect to conventional TCAM. In the approximate mode, using long MASC precharging refresh periods improves GPGPU energy efficiency by 1.6X with respect to applying conventional voltage overscaling.

ACKNOWLEDGEMENT

This work was sponsored by NSF grant #1527034.

VI. CONCLUSION

In this paper, we propose an ultra-low energy multiple-access single-charge TCAM which significantly decreases the energy consumption of the associative memory. We observe that a conventional TCAM discharges all miss rows when performing a search operation, consuming a large amount of search energy. We propose the MASC TCAM design, which discharges only the hit row(s) while miss rows stay precharged. This allows us to use the charge of the miss-lines to perform multiple search operations. Our evaluation shows that the proposed 8-bit TCAM can achieve error-free search operations using a period of 4-cycles for precharging. Increasing the period of precharging improves the TCAM search energy efficiency at the expense of adding error to the matching patterns. This proposed approximate associative memory decreases the energy consumption of the GPGPU by 40.2% and 39.4% for 1-HD and 2-HD implementation with acceptable quality of service (PSNR>30dB). These savings are 1.6X higher than approximation using conventional voltage overscaling techniques.

REFERENCES

[1] A. Katal, et al., "Big data: Issues, challenges, tools and Good practices," *IEEE Contemporary Computing*, pp. 404-409, 2013.
[2] C. Ji, et al., "Big data processing in cloud computing environments," *IEEE ISPAN*, pp. 17-23, 2012.
[3] T. Kohonen, "Associative memory: A system-theoretical approach," *Springer Science & Business Media*, vol. 17, 2012.
[4] K. Pagiamtzis, et al., "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE JSSC*, vol. 41, pp. 712-727, 2006.
[5] M. Imani, et al., "Hierarchical design of robust and low data dependent FinFET based SRAM array." IEEE/ACM Nanoscale Architectures (NANOARCH), pp. 63-68, 2015.
[6] A. Rahimi, et al., "Approximate associative memristive memory for energy-efficient GPUs," *IEEE DATE*, pp. 1497-1502, 2015.
[7] M. Imani, et al., " ReMAM: low energy resistive multi-stage associative memory for energy efficient computing," *IEEE ISQED*, 2016.
[8] M. Breuer, "Multi-media applications and imprecise computation," *IEEE Digital System Design*, pp. 2-7, 2005.
[9] A. Goel, et al., "Small subset queries and bloom filters using ternary associative memories, with applications," *ACM SIGMETRICS Performance Evaluation Review*, pp. 143-154, 2010.
[10] K. Lakshminarayanan, et al., "Algorithms for advanced packet classification with ternary CAMs," *ACM SIGCOMM Computer Communication Review*, pp. 193-204, 2005.
[11] S. Kaxiras, et al., "IPStash: a set-associative memory approach for efficient IP-lookup," *IEEE INFOCOM*, pp. 992-1001, 2005.
[12] H. Zhang, et al., "Low power gpgpu computation with imprecise hardware," *ACM/IEEE DAC*, pp. 1-6, 2014.
[13] A. Rahimi, et al., "Temporal memoization for energy-efficient timing error recovery in gpgpus," *IEEE DATE*, p. 100, 2014.
[14] J. Li, et al., "1 Mb 0.41 μm² 2T-2R Cell Nonvolatile TCAM With Two-Bit Encoding and Clocked Self-Referenced Sensing," *IEEE JSSC*, vol. 49, pp. 896-907, 2014.
[15] S. Paul, et al., "Nanoscale reconfigurable computing using non-volatile 2-d sttram array," *IEEE Nanotechnology*, pp. 880-883, 2009.
[16] J. Cong, et al., "Energy-efficient computing using adaptive table lookup based on nonvolatile memories," *IEEE ISLPED*, pp. 280-285, 2013.
[17] S, N. Mozaffari, et al, "Fast march tests for defects in resistive memory", *IEEE Nanoarch*, pp. 88-93, 2015.
[18] Y. Kim, et al. "CAUSE: critical application usage-aware memory system using non-volatile memory for mobile devices." *IEEE/ACM ICCAD*, 2015.
[19] Q. Guo, et al., "AC-DIMM: associative computing with STT-MRAM," *ACM SIGARCH Computer Architecture News*, pp. 189-200, 2013.
[20] T. Hanyu, et al., "Spintronics-based nonvolatile logic-in-memory architecture towards an ultra-low-power and highly reliable VLSI computing paradigm," *IEEE DATE*, pp. 1006-1011, 2015.
[21] M.-F. Chang, et al., "A 3T1R Nonvolatile TCAM Using MLC ReRAM with Sub-1ns Search Time," *IEEE ISSCC*, pp. 318-U449, 2015.
[22] S. Matsunaga, et al., "A 3.14 um 2 4T-2MTJ-cell fully parallel TCAM based on nonvolatile logic-in-memory architecture," *IEEE VLSIC*, pp. 44-45, 2012.
[23] D. Mohapatra, et al., "Design of voltage-scalable meta-functions for approximate computing," *IEEE DATE*, pp. 1-6, 2011.
[24] N. Bandi, et al., "Fast data stream algorithms using associative memories," *ACM SIGMOD*, pp. 247-256, 2007.
[25] K. Eshraghian, et al., "Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines," *IEEE TVLSI*, vol. 19, pp. 1407-1417, 2011.
[26] C. Ranger, et al., "Evaluating mapreduce for multi-core and multiprocessor systems," *IEEE HPCA*, pp. 13-24, 2007.
[27] R. Agrawal, et al., "Fast algorithms for mining association rules," *IEEE VLDB*, pp. 487-499, 1994.
[28] T. Kohonen, "Content-addressable memories," *Springer Science & Business Media*, vol. 1, 2012.
[29] S. Panchanathan, et al., "A content-addressable memory architecture for image coding using vector quantization," *IEEE Signal Processing*, vol. 39, pp. 2066-2078, 1991.
[30] M. Jafari, et al. "Bottom-up design of a high performance ultra-low power DFT utilizing multiple-V DD, multiple-Vth and gate sizing." IEEE DTIS, 2013.
[31] "AMD APP SDK v2.5" Available: http://www.amd.com/stream".
[32] R. Ubal, et al., "Multi2Sim: a simulation framework for CPU-GPU computing," *IEEE PCAT,* pp. 335-344, 2012.
[33] Design Compiler, "Synopsys Inc,".
[34] "http://www.vision.caltech.edu/Image_Datasets/Caltech101/".