Lifetime Improvement and Low Power Concept of MemCAM/MemTCAM

Isha[1] Vijay Nandal[2] Manisha[3]
[1]Student [2,3]Assistant Professor
[1,2,3]Department of Electronics & Communication Engineering
[1,2,3]MRIEM, Rohtak

Abstract— *Rapid data growth makes it more critical to reduce search time to improve the extensive performance of search-intensive applications. Present conventional approaches such as CAM to reduce search time are no longer efficient because of the huge data: CMOS-based CAM has low capacity which cannot be increased further through CMOS scaling, and performance degradation occurs as data size increases. In order to fully utilize the computation ability of memristors and deal with the lifetime problem, we hereby explore the design space of memristor-based memory storage structures. We first propose MemCAM/MemTCAM, a configurable memristor-based CAM/TCAM design, which improves the power consumption. Here we mention some techniques to improve lifetime of MemCAM and low power consumption methods for CAM.*

Keywords— MEMTCAM, MEMCAM

I. INTRODUCTION

A CAM is a memory that implements the look-up table function using dedicated comparison circuitry in a single clock cycle. Content-Addressable Memory (CAM) compares input search data against a table of stored data, and returns the address of the matched data. CAM comes at the cost of increased silicon area and power consumption, two design parameters that designers strive to reduce. TCAM is specialized type of high speed memory that searches its entire content in a single clock cycle. The term "ternary" refers to the memory's ability to store and query data using three different inputs: 0,1 and X. the "X" input, which is often referred to as a "don't care" or "wildcard" state, enables TCAM to perform broader searches based on pattern matching, as opposed to binary CAM which performs exact match searches using only 0's and 1's[2]. A memristor is an electrical component that limits or regulates the flow of electric currents in a circuit and remembers the amount of charge that has previously flowed through it. Memristors are important because they are non-volatile, meaning that they retain memory without power[1].

II. LIFETIME IMPROVEMENT METHOD OF TCAM

Memristor have low endurance which limits its use in many applications. In this paper I suggest different methods for solving this problem. Lower endurance means it has 10^{10} write cycles as compared to other memory devices like DRAM/SRAM which have 10^{16} write cycles. The comparison and match both require write operation into the memristor which limits its lifetime to several minutes only. So we need to reduce its endurance problem to take advantage of memristor properties.

Two technologies have been discussed here:

- Wear-leveling technique
- Write avoidance method

A. Wear-Leveling

This technique is used to uniform the writes to improve lifetime of storage devices built using technologies like PCM and flash memory[4]. We can avoid wear out by remapping most frequently written lines to less frequently written lines. Non- uniformity in writes decrease the achievable lifetime. *Wear leveling* is a mechanism that tries to make the writes uniform by remapping heavily written lines to less frequently written lines[6]. Wear-leveling need tables to track write counts associated with each line and an indirection table to perform address mapping to achieve uniform wear-out of the system. Unfortunately, the hardware required for these structures scales linearly with the memory being tracked and is typically in the range of several Mega Bytes (MB). Also, table look-up adds significant latency to each access and also increases the overall power consumption. The storage tables of wear leveling can be eliminated if an algebraic mapping can be provided between the logical and physical address. Based on this people of IBM proposed a technique named START-GAP. A simple and effective technique that uses two registers Start and Gap to do wear leveling. Start Gap moves one line from its location to a neighbouring location. Gap register keeps tracks of how many lines have moved. When all the lines have moved, the start register is incremented to keep track of the no of times all lines have moved. The mapping of lines from logical address to physical address is done by a simple arithmetic operation of Gap and Start registers with the logical address. START_GAP can obtain average lifetime of more than 97% of the ideal.

B. Write Avoidance

Even after uniform distributing writes, we can also reduce the average write frequency to increase the overall lifetime of any memory systems. Some of the write avoidance mechanisms are hardware implemented and rest are software implemented. The software implemented mechanisms are transparent to user level application programmers but others are not transparent to system level application programmers. We reduce the average write frequency to MemCAM/MemTCAM[3] through composite designs in this section and make an estimation how the efficiency is affected.

Two write avoidance approaches are:

1) Composite memristor based CAM/TCAM memory
2) Composite CMOS-memristor based CAM/TCAM

1) Composite Memristor Based CAM/TCAM Memory:

We can reduce the average write frequency by utilizing the different composite structures of a memristor array. One advantage of memristor-based storage structures is that a memristor array can function as CAM/TCAM or memory based on the voltage supplied on to the memristors. We can divide a memristor array into multiple partitions where each

partition have the same capacity and recognize one partition as CAM/TCAM and rest partitions as memory. We can then 'rotate' the CAM/TCAM partition within the memristor array as shown in Figure. The improvement of lifetime by using the hybrid memristor-based CAM/TCAM memory design is directly proportional to the number of partitions. However this design requires a large memristor array to obtain acceptable lifetime of a small MemCAM/TCAM.

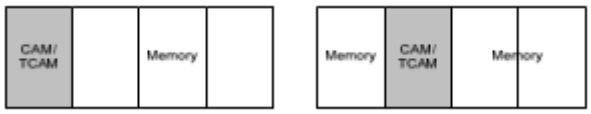

Fig. 1: Composite Memristor-based CAM/TCAM-memory

2) Composite CMOS-Memristor-Based CAM/TCAM

Hierarchical storage structure is also another option to reduce write frequency. We can use a CMOS-based CAM/TCAM as a buffer of MemCAM/MemTCAM as shown in Figure. We store hot data (more frequently searched data) in CMOS CAM/TCAM buffer and store cold data In MemCAM/MemTCAM. The search frequency of MemCAM/MemTCAM is reduced and hence the write frequency.

The improvement of lifetime by using the hybrid CMOS-memristor based CAM/TCAM design is somehow dependent on the capacities of both CAMs/TCAMs and the access frequencies of both hot and cold data how frequently they are used. Hot data has to be accessed $4x10^5$ times more frequently than cold data in order to achieve a one-year lifetime.

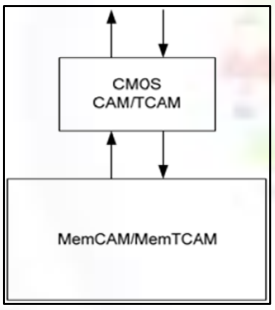

Fig. 2: Composite CMOS Memristor based CAM/TCAM

III. LOW POWER CONCEPT FOR CONTENT ADDRESSABLE MEMORY

A basic CAM cell function could be observed as twofold: bit storage as in RAM and bit comparison which is unique to CAM. The input signal is one bit value from the search data register i.e. the input word to be compared against all the values stored in CAM arrays or the value to be stored in the CAM cell. Cell enable signal allows or prevent comparison i.e. matching process meaning XOR-ing the stored bit value in the Flip-flop and the input bit[5].

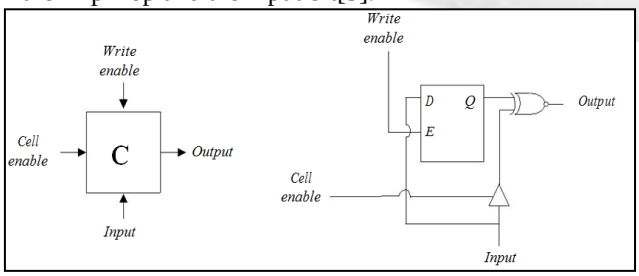

Fig. 3: CAM cell (1) logic symbol; (2) circuit

A. One-Cell Segmentation Power Reducing Scheme

Selective pre-charge scheme basically divides the mach line in two segments. In general following the same concept it can be divided in many numbers of segments thus forming a *pipeline.* When performing a search if the first few bits do not match there is no point in checking the remaining bits. *Selective pre-charge* initially searches only first *n* bits and only searches the remaining bits for words that matches first n bits. The drawback of this scheme is the increased latency and area overhead due to the pipeline stages.

B. Parity Check Pre-Computation Power Reducing Scheme

One method uses pre-computation circuits to count the number of ones and stores this data along with the word in binary format are compared for every stored word and for those that match the process continues with the data search. To perform the computation the data word bits are grouped in three bit segments. It worth noting that simple FA[30] at circuit level is implemented with two AND, two XOR and one OR gates. Other method is instead of counting the number of 1's (or 0's) we only check the parity of 1's in the data word.

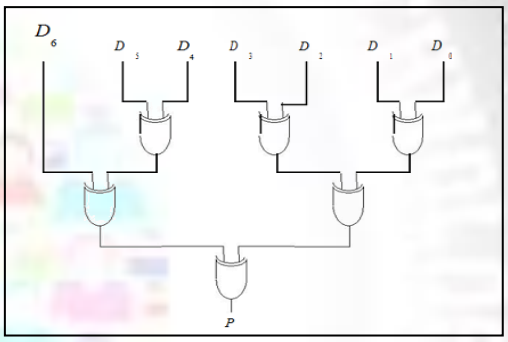

Fig. 4: logical circuit for bit parity parameter extractor

IV. RESULTS

A. Wear Leveling: Swap Algorithm

Let L be any logical page to write and
P be any physical page that is being allocated to L
Swapping is done on pages, not on the subpages, to minimize implementation overhead.
If
SWAP condition ( ) == FALSE then
Write L data on P
Else
P' = select SWAP destination page ( )
P ← P' (copy data from P' to P)
Write L on P'
End if

B. Gap Movement

If
Ψ<= K then
Write to same line
Else
[Gap] = [Gap] = [Gap-1]
End if

1) Algorithm:

If
GAP== 0 then
Start = Start + 1;

[Gap] = [N]
Else
[Gap] = [Gap - 1]
End if

Power reducing techniques is always a counter balance to the large logic area and the speed of processing. Power reducing is at the cost of adding delays in the system. Since the whole process of matching should be completed in one clock cycle, the overall system delay should be completely fit into the one clock cycle period time.

The overall delay T is the sum of all the successive delays of each CAM cell and the delay added by parameter extractor.

$$T = t + n\tau$$

t = parameter extractor delay
$\tau =$ delay of CAM cell
Which lies in single clock cycle.

REFERENCES

[1] Williams, R. Stenley, "How we found the missing memristor". IEEE spectrum.NA, 2008, pp. 29-35.
[2] Pagiamtzis Kostas and Sheikholeslami Ali, "content-addressable memory (CAM) circuits and architectures: a tutorial and survey. IEEE journal of solid-state circuits", VOL. 41, NO. 3, MARCH 2006, pp. 712-725.
[3] K. Eshraghian, et al., ″Memristor MOS content addressable memory (MCAM): hybrid architecture for future high performance search engines,″ IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. PP, 2010.
[4] Moinuddin K. Qureshi, Michele Franchucini and Vijaylakshmi Srinivasan,"Enhancing Lifetime and Security of PCM based main Memory with Start Gap Wear Leveling", MICRO'09, December 12-16,2009.
[5] Sparsh Mittal and Jeffrey Vetter," A Technique for Improving Lifetime of Non-volatile Caches using Write Minimization", Journal of Low Power Electronics and Applications, 18 january, 2016.
[6] S. Chen, et al., ″Rethinking database algorithms for phase change memory,″ in Proceedings of the 5th Biennial Conference on Innovative Data System Research (CIDR ′11), 2011.