PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

# Optimized large-capacity content addressable memory (CAM) for mobile devices

Khader Mohammad, Iyad Tumar

Khader Mohammad, Iyad Tumar, "Optimized large-capacity content addressable memory (CAM) for mobile devices," Proc. SPIE 9411, Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2015, 94110E (11 March 2015); doi: 10.1117/12.2084505

SPIE.

Event: SPIE/IS&T Electronic Imaging, 2015, San Francisco, California, United States

# Optimized Large-Capacity Content Addressable Memory (CAM) for Mobile Devices

Khader Mohammad[a], Iyad Tumar[b]
[a,b]Birzeit University, P.O.BOX 14, Ramallah, Palestine

## ABSTRACT

A content addressable memory system includes CAM cells, each having a compare circuit and a memory bit cell that stores complementary bits. The main CAM design challenge is to reduce power consumption associated with large amount of parallel switching circuitry, without sacrificing speed or density. In this paper, we present a new technique to eliminate crowbar current during bit-cell write operation (saving 0.0114mA per cell in 22nm process), reduce average current consumption during cam operation and eliminate the need for routing the complementary data to every cam cell, saving routing track in smaller node technology where wire cap is dominant.

**Keywords:** Low Power, CAM design, memory, content addressable memory, mobile device CAM

## 1. INTRODUCTION

In mobile devices and systems, CAM cells are essential in any application that requires look up and search operation of data. The CAM can perform all the functions of an SRAM cell, including read or write operations given address and data information. It's also capable of performing matching operations. The key is to compare the cam data lines of the cell, and if the data is matched to the contents of a certain bits, the match lines of the bit are raised. Based on the hits, it returns the addresses at which the target data could be found. There have been many attempts to reduce the transistor count and resulting area for the CAM XOR block. A comprehensive review of different varieties of CAM cells, which can be equally applicable to TCAM cells, was presented in [7]. Other design approaches where presented in [2-11].

Content-addressable memories are hardware search engines that are much faster than algorithmic approaches for search-intensive applications [1]. They get used in many microprocessor design.

A typical CAM bit-cell implementation is shown in Figure 1. The bit-cell (BIT) is written into by enabling the write wordline (WRWL) & driving desired value through write bitlines (WRBL/WRBLY). During CAM operation, BIT value is compared against CAMDATA and MATCH is asserted when BIT & CAMDATA values are the same. However this particular implementation has a crow-bar current issue. There is one gate delay between BIT and BITX which opens one pass gate before closing the second, resulting in crowbar current.

Bit cells of a CAM system may include compare circuits to compare contents of the bit cells with reference bit values provided to the compare circuits. Conventional CAM compare circuits are implemented with complementary or differential reference bit lines, which disadvantageously increase routing complexity and space requirements. Typically the compare circuits include separate pass circuits associated with the differential reference bit lines. Switching delays in the CAM cell can cause unwanted current contention between the separate pass circuits, which manifests itself as a crowbar current that wastes power and slows down CAM speed.

Since there is strong relationship between the parametric failures in SRAM-based memory and supply voltage, other approaches has been proposed to use different supplies to the memory to minimize the impact of raising all the chip supply on power [14].

This invented paper will lead to further reduction in power for mobile processors which are targeted low power applications (cell phone and tablet).

Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2015, edited by Reiner Creutzburg, David Akopian, Proc. of SPIE-IS&T Electronic Imaging, Vol. 9411, 94110E

The paper is organized as follows: Section II introduce proposed approach (for typical and custom CAM cell) including two proposed methods and in section IV the simulation results are studied and analyzed

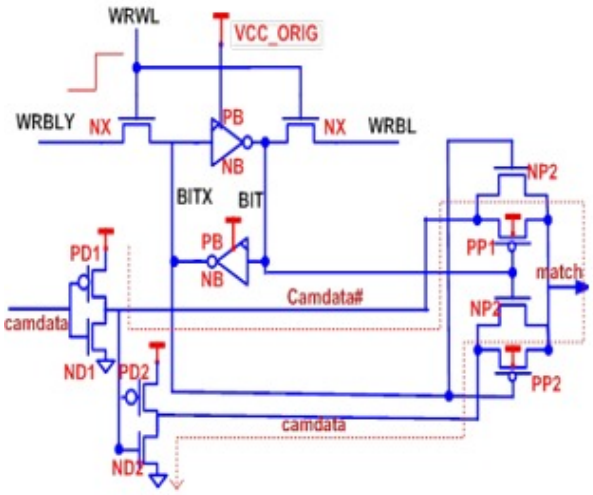


Figure 1: Abstract view of the CAM cell & crowbar current during BIT write operation

## 2. PROPOSED CAM CELL

A new technique to eliminate crowbar current during bit-cell write operation, reduce average current consumption during cam operation eliminate the need for routing camdata# to every cam cell, saving precious routing track and generating power saving in smaller node technology where wire cap is dominant. This invention has two methods of reducing power consumption with less routing tracks:

### 2.1 Method 1

While traditional implementation uses CAMDATA & its complement to compare bitcell, the new circuit eliminates of routing CAMDATA signal and manages compare using only the complement. The proposed new circuit is shown in figure 2. This method:

- Eliminates the CAMDATA signal hence saving precious routing track & power from not having to switch it. Although CAMDATA driver is upsized to drive additional gate load in CAM bit cell, overall power savings are attained due to route & its switching power elimination.
- Reduces power consumption when stored bit is 0 and CAM operation is performed by effectively switching less cap.
- Achieves ISO cell delay and output slope in configurations where CAMDATA inverter drives single or 8 bit cells. Cell delays and output slopes improve for 16 bit cell configuration. The 8 and 16 bit cell configuration is shown in figure 3.

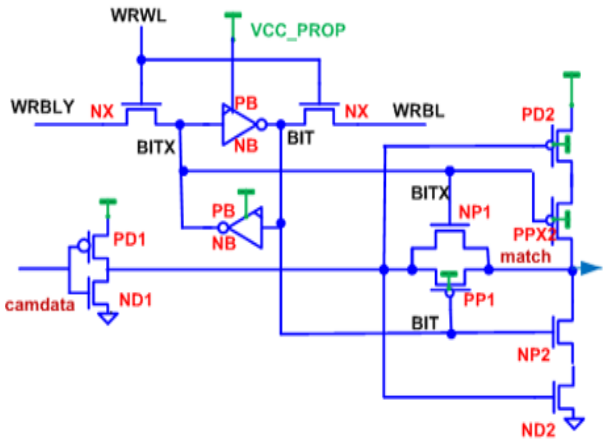


Figure 2: Abstract view of the proposed cell

### 2.2 Simulation results for method 1

Figure 4 shows the simulation results performed in 22nm process technology using single bit cell. The 3rd chart shows the current for both the proposed (blue) and original (yellow) designs. Table 1 shows the average current savings of 40% for typical usage of this structure in microprocessor core's front end cluster (for one write followed by 20 CAM operations). The implementation has no performance impact as shown in Table 2. The two devices added per CAM bit-cell increase its total device width by 6% but it does not increase the net cell area. Even though there is growth in single cam bit-cell area, block area does not grow as column width (in this array implementations) is set by metal routing and spacing for word lines above the cell. This method is simulated for more realistic case (8 cells connected to the same CAMDATA for each bit, figure 3 (A)). The top level effective load simulation results graph is shown in figure 4. The 40% current saving seen in single cell configuration is now reduced to 14% during CAM operation and 12% during write operation – with no performance degradation as shown in Table 3.

The practical usage model (48 entries by 48 bits CAM structure) is shown in figure 5. The total current saving is 985uA during cam operation and 283uA during write operation. Table 3 (A) shows a summary of current savings for the 8 cell implementation. The savings depends on the data stored in the memory bit cell and the wire cap. Table 3 (B) presents the current saving for 48 bit in 48 entries (1 write and 20 cam operation).

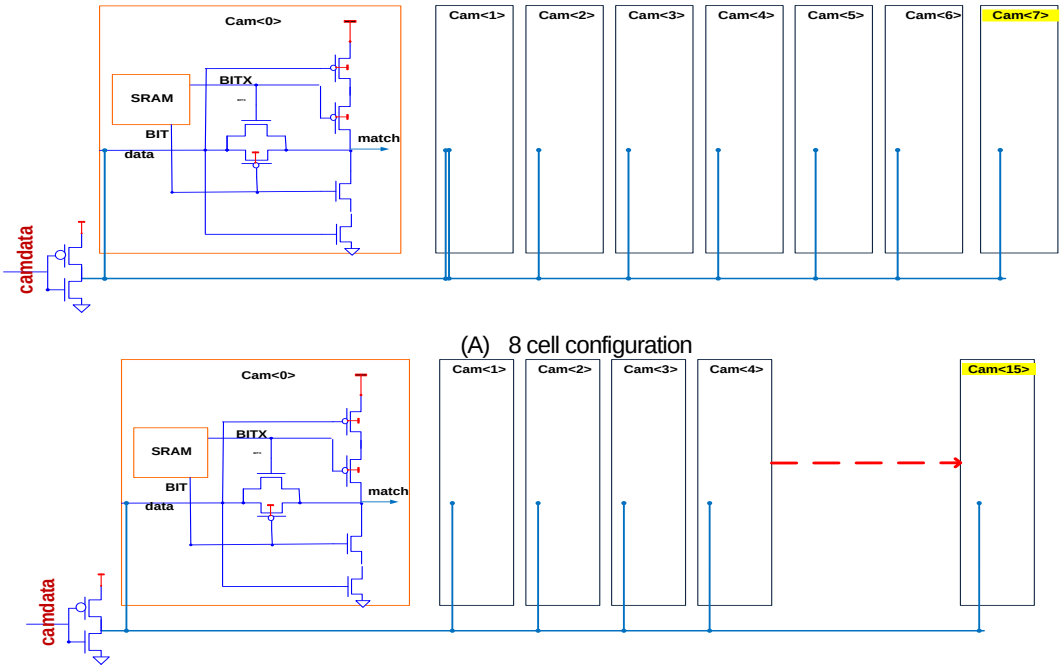


Figure 3: Typical implementation of CAM (8 and 16 cell configuration)

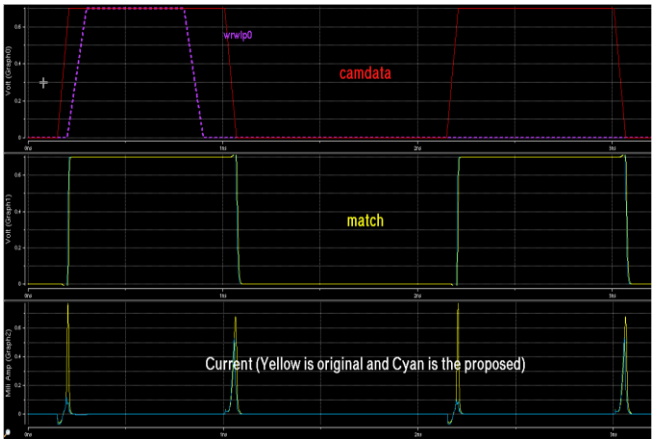


Figure 4: Method 1 Simulation Results

**Method 1 Result & Summary**

The proposed method:

- Saves routing tracks per cell. For example it can save 4 tracks in a cell with 4-CAM
- Power saving increases if architecture requires more writes to the CAM cell
- Single cell area stay the same.
    - Lower the amount of power by: 40% average current saving (one cell)

- For more realistic case (8-cells), average power saving of 13% -20% (bit is 0)
- Average power saving across all CAM blocks is 0.009 mW (assuming AF=2%, 1write 20 cam)

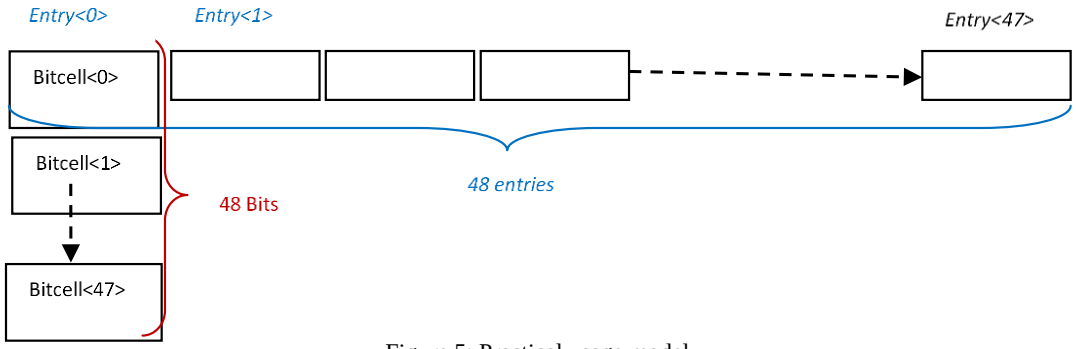


Figure 5: Practical usage model

**Table 1: Current savings comparison between original and proposed (1 write and 20 cam)**

| One cell (BIT IS SET TO 1) | | | |
|---|---|---|---|
| | Io(uA) | Ip((uA) | Im % |
| Avg_Current_NO_WRITE(CAM Operation) | 11.10 | 6.64 | 40% |
| Avg_Current_During_WRITE | 18.15 | 12.26 | 32% |
| One cell (BIT IS 0) | | | |
| Avg_Current_NO_WRITE(CAM Operation) | 11.03 | 5.51 | 50% |
| Avg_Current_During_WRITE | 14.4 | 7.32 | 49% |
| Avg_Current_SAVING_NO_WRITE(CAM Operation) | | | 45% |
| Avg_Current_SAVING_ During_WRITE | | | 41% |

*Io: Original Current, Ip: Proposed current* Im: Current improvement

**Table 2: Delay comparison (ps) between original and proposed designs**

| One cell | Original Delay(ps) | Proposed Delay(ps) |
|---|---|---|
| Delay | 40.9 | 39.6 |
| match_slope_fall | 7.47 | 7.9 |
| match_slope_rise | 3.67 | 3.94 |

**Table 3 (A) : Current saving for 8 cells with different cap and different stored bit value (1 write and 20 cam operation)**

| BIT = 1 | Cap(xff) | | | Cap(2xff) | | |
|---|---|---|---|---|---|---|
| | Io(uA) (C=2ff) | Ip(uA) (C=2.18ff) | % | Io(uA) (C=2ff) | Ip(uA) (C=2.18ff) | % |
| Avg_Current_NO_WRITE(CAM Operation) | 26.09 | 24.45 | 6% | 26.92 | 24.77 | 8% |
| Avg_Current_During_WRITE | 63.3 | 59.81 | 6% | 64.38 | 60.23 | 6% |
| **BIT = 1** | **Cap(xff)** | | | **Cap(xff)** | | |
| Avg_Current_NO_WRITE(CAM Operation) | 27.89 | 22.57 | 19% | 30.7 | 23.92 | 22% |
| Avg_Current_During_WRITE | 36.24 | 29.62 | 18% | 26.24 | 29.62 | 18% |
| Avg_Current_SAVING_NO_WRITE(CAM Operation) | | | 13% | | | 15% |
| Avg_Current_SAVING_During_WRITE | | | 12% | | | 12% |

**Table 3 (B): Total Current saving example for 48 bit in 48 entries (1 write and 20 cam operation)**

| | | 48 Entry | 48 Bit |
|---|---|---|---|
| | Current Saving (mA) | Total(uA) | Total(uA) |
| CAM Operation | 0.003420219 | 0.985023 | |
| Write Operation | 0.005916612 | | 0.283997 |

### 2.3 Method 2

The two devices ND2 & PD2 can be shared across 8 cells as shown in figure 6. This option will reduce the total device width (Z) penalty per cell and yields current savings of 14% as tabulated in table 4 with no performance degradation, table 5. The only disadvantage of this option is that the diffusion node is routed across the 8 cells leaving it susceptible to noise injection.

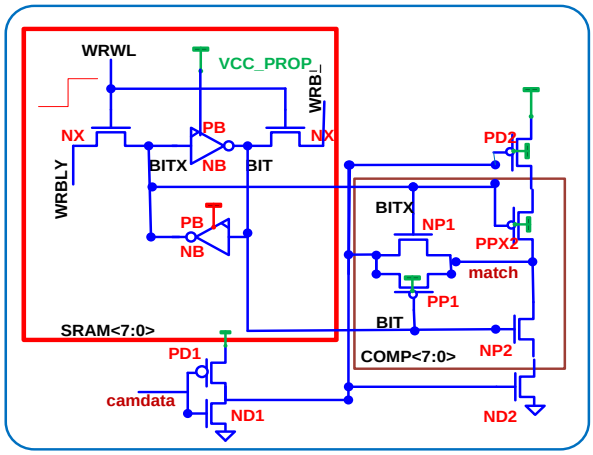


Figure 6: Method 2 circuit

**Method 2 Result & Summary**

Using this method:

- Average current saving of 14%.
- Has no impact to area or performance

Has to route the diffusion node across 8 cells

## 3. CONCLUSION

A content addressable memory (CAM) system includes CAM cells, each having a compare circuit and a memory bit cell that stores complementary bits. The compare circuit includes complementary inputs to receive the complementary stored

**Table 4: Method 2 current saving**

| One cell (BIT IS SET TO 1) | | | |
|---|---|---|---|
| | Io(uA) | Ip((uA) | Im% |
| Avg_Current_NO_WRITE(CAM Operation) | 27.89 | 29.99 | 25% |
| Avg_Current_During_WRITE | 36.24 | 27.67 | 24% |
| One cell (BIT IS 0) | | | |
| Avg_Current_NO_WRITE(CAM Operation) | 28.14 | 27.41 | 3% |
| Avg_Current_During_WRITE | 65.97 | 63.3 | 4% |
| Avg_Current_SAVING_NO_WRITE(CAM Operation) | | | 14% |
| Avg_Current_SAVING_ During_WRITE | | | 14% |

**Table 5: Delay and slope**

| 8 cells With Shared FET | |
|---|---|
| Delay_camdata_match_orig(ps) | 40.42 |
| Delay_camdata_match_prop(ps) | 38.18 |
| Slope_fall_orig<ps> | 10.28 |
| Slope_rise_orig<ps> | 5.99 |
| Slope_fall_prop<ps> | 5.47 |
| Slope_rise_prop<ps> | 9.48 |

bits, and an input node to receive a single-ended reference bit. The compare circuit includes circuitry controlled by the logic values of the single-ended reference bit and the complementary stored bits to provide a match output indicating a result of a compare between the stored complimentary bits and the reference bit. The CAM cells have respective or per-cell compare circuitry, but also share compare circuitry among the cells.

This invented paper presents two methods to reduce power in cam cells without impacting performance or area (method 2). Both methods have almost the same saving of 13%. Method 1 is easier to design and implement though it has cost penalty. While the second method has the same saving with no area impact, it adds more complexity for the design.

## REFERENCES


[1] Scott Beamer, Mehmet Akgul, "Design of a Low Power Content Addressable Memory (CAM)," University of California, Berkeley

[2] Hisatada Miyatake, Masahiro Tanaka, and Yotaro Mori. "A Design for High-Speed Low-Power CMOS Fully Parallel Content-Addressable Memory," Macros IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 36, NO. 6, JUNE 2001

[3] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE Journal of Solid-State Circuits, vol. 41, no. 3, pp. 712–727, March 2006.

[4] K. J. Schultz, "Content-addressable memory core cells: A survey," Integration, the VLSI Journal, vol. 23, no. 2, pp. 171–188, November 1997.

[5] Mohamed Elgebaly (2005), Energy Efficient Design for Deep Sub-micron CMOS VLSIs, Ph D Thesis, University of Waterloo, Waterloo, Ontario, Canada.

[6] Atila Alvandpour, Ram Krishnamurthy, K. Soumyanath, and Shekhar Borkar, "A Low-leakage Dynamic Multi-Ported Register File in 0.13um CMOS," Proceedings of the 2001 international symposium on Low power electronics and design, pp. 68-71, August 2001.

[7] Ataur R. Patwary, Hans Greub, Zhongfeng Wang, and Bibiche M. Geuskens, "Bit-line Organization in Register Files for Low-power and High-performance Applications," 4th International Conference on Electrical and Computer Engineering ICECE 2006, pp. 505-508, Dec. 2006.

[8] S. Thompson, I. Young, J. Greason, and M. Bohr, "Dual Threshold Voltages and Substrate Bias: Keys To High Performance, Low Power, 0.1 μm Logic Designs," Symposium on VLSI Technology Digest, pp. 69-70, June 1997.

[9] Khader Mohammad, Ahsan Kabeer, and Tarek Taha, "On-Chip Power Minimization Using Serialization-Widening with Frequent Value Encoding", VLSI Design Journal, Volume 2014 (2014), Article ID 801241, 14 pages

[10] Ataur R. Patwary, Bibiche M. Geuskens, and Shih-Lien L. Lu, "Content Addressable Memory for Low-Power and High-Performance Applications," accepted for publication in World Congress on Computer Science and Information Engineering, Los Angeles/Anaheim, California, USA, March/April 2009.

[11] Kostas Pagiamtzis, Stu, Ali Sheikholeslami, "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey, " IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 3, MARCH 2006