International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

Reduction of Power in TCAM Using Precharge Controller

D. REVATHI [1], V.Venkata Rao[2]

[1]PG Student, [2]Professor& HOD
Department of ECE, Narasaraopeta Engineering College, Narasaraopet, Guntur, AP, INDIA
[1]revathid.95@gmail.com

ABSTRACT:

CAM is a hardware search algorithm based on the software search algorithm. This compares input search data against a table of stored data, and returns the address of the matching data. CAM is an important lookup table hardware for applications like network routers, data compression, image processing and data management. Ternary cells, in addition, store an "X" value. The "X" value is a don't care, that represents both "0" and "1", allowing a wildcard operation. Wildcard operation means that an "X" value stored in a cell causes a match regardless of the input bit. This is a feature used in packet forwarding in Internet routers. The main challenges in the CAM are power consumption without affecting the search speed and the power consumption in the TCAM can be reduced by identifying the missing matchline and not allowing charging fully.

Keywords:

CAM (Content Addressable Memory), TCAM (Ternary Content Addressable Memory)

1. Introduction

A Content-Addressable Memory (CAM) compares input search data against a table of stored data, and returns the address of the matching data. CAMs can be used in a wide variety of applications requiring high search speeds. The main application of CAMs today is to classify and forward Internet protocol (IP) packets in network routers. In networks like the Internet, a message such an as e-mail or a Web page is transferred by first breaking up the message into small data packets of a few hundred bytes, and, then, sending each data packet individually through the network(routers). The function of a router is to compare the destination address of a packet to all possible routes, in order to choose the appropriate one. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

1.1 CAM Architecture

A small model of CAM architecture is shown in Fig. 1. The figure shows a CAM consisting of 4 words, with each word containing 3 bits arranged horizontally (corresponding to 3 CAM cells). There is a Matchline corresponding to each word (ML0, ML1, etc.) feeding into Matchline sense amplifiers (MLSAs), and there is a differential Searchline pair corresponding to each bit of the search word ($SL_{0},\overline{SL}_{0}, \overline{SL}_{1},SL_{1}$,etc.). A CAM search operation begins with loading the search-data word into the search-data registers followed by precharging all matchlines high, putting them all temporarily in the match state. Next, the Searchline drivers broadcast the search word onto the differential Searchlines and each CAM core cell compares its stored bit against the bit on its corresponding Searchlines. Matchlines on which all bits match remain in the Precharged-high state. Matchlines that have at least one bit that misses, discharge to ground. The MLSA then detects whether its Matchline has a matching condition or miss condition. Finally, the encoder maps the Matchline of the matching location to its encoded address.

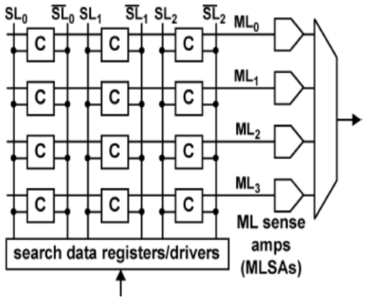


Fig. 1: CAM Architecture

2. CONTENT-ADREESSABLE MEMORY (CAM)

CAM is a hardware search algorithm based on the software search algorithm [2]. Where compares input search data against a table of stored data, and returns the address of the matching data. CAM is an important lookup table hardware for applications like network routers [6] data compression [4], image processing [5] and data management [3]. The parallel active circuits in CAM make it power consumption. The challenge is to reduce the CAM power consumption without affecting the search speed. CAM cells connected to a single match-line (ML) form a CAM word. Precomputation-based CAM is widely used. Each word stored in a CAM has memory storage with it to store the word. Search word is precompiled when it is given. The search word and the each word in CAM are first compared in parallel [1]. The CAM words which do not match are eliminated from the search it takes more time for detecting the match line and also make power hungry. So, to get an efficient design of CAM, NOR CAM which is having off, high-speed performance cell and NAND CAM of low power feature cell are combined. CAM is segmented into two, and the second segment is obtained for search conditionally only if match is found in the first segment. If inconsistent results are occurred between the two segments, then, late correction is obtained. The more power is consumed for the mismatch MLs. To reduce the power consumption adds a dummy row to pull the precharge signal as soon as the match is found; reducing the swing level of the mismatching MLs. CAM architectures have NOR CAM as the basic building block. The NOR CAM has a short circuit problem, and a precharge-free CAM is proposed. The design is obtained with a precharge signal, free from short-circuits current and results the better performance with dynamic precharging of the MLs without using the dummy row. With the help of the CAM we can store only 0's and 1's called as the Binary CAM. Where as in the TCAM we can store don't care condition also irrespective of the input is taken.

2.1. Binary CAM

CAM can be operated with NORCAM cells and the NAND CAM cells. In Binary CAM only '0' and '1' can be stored. Mostly we use the NOR CAM cells because it is faster than the NAND CAM cells. NAND CAM cells have less power efficiency. NOR CAM cells have the short circuit problem but it can be reduced by precharging the matchlines. In the below section the NOR CAM and NAND CAM cells are explained.

2.1.1. NOR Cell

The NOR cell shown in the Fig. 2 implements the comparison between the complementary stored bit, D (and $\overline{D}$), and the complementary search data on the complementary Searchline, SL (and $\overline{SL}$), using four comparison transistors, M1 through M4. These transistors implement the pulldown path of a dynamic XNOR logic gate with inputs SL and D. Each pair of transistors M1/M3 and M2/M4 forms a pulldown path from the Matchline, ML, such that a mismatch of SL and D activates least one of the pulldown paths. Connecting ML to ground. A match of SL and D disables both pulldown paths, disconnecting ML from ground. The NOR nature of this cell becomes clear when multiple cells are connected in parallel to form a CAM word by shorting the ML of each cell to the ML of adjacent cells. The pulldown paths connect in parallel resembling the pulldown path of a CMOS NOR logic gate. There is a match condition on a given ML only if every individual cell in the word has a match.

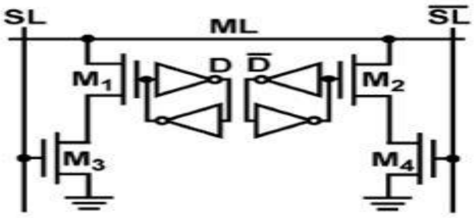


Fig. 2: NOR CAM cell

2.1. 2 NAND Cell

The NAND cell shown in the Fig. 3 implements the comparison between the stored bit, D, and corresponding search data on the corresponding Searchlines, (SL, $\overline{SL}$), using the three comparison transistors $M_{1,}$ M_D and M_d. Take an example to illustrate the bit-comparison operation of a NAND cell through an example. Consider the case of a match when SL=1 and D=1. Pass transistor M_D is ON and passes the logic "1" on the SL to node B. Node B is the bit-match node which is logic "1" if there is a match in the cell. The logic "1" on node B turns ON transistor.

Note that M_1 is also turned ON in the other match case when SL=0 and D=0. In this case, the transistor M_d passes logic high to raise node B. The remaining cases SL ≠ D, where, result in a miss condition, and accordingly node B is logic "0" and the transistor M_1 is OFF. Node B is a pass-transistor implementation of the XNOR function SL $\odot$ D. The NAND nature of this cell becomes clear when multiple NAND cells are serially connected. In this case, the ML_n and ML_{n+1} nodes are joined to form a word. A serial nMOS chain of all the M_i transistors resembles the pulldown path of a CMOS NAND logic gate. A match condition for the entire word occurs only if every cell in a word is in the match condition.

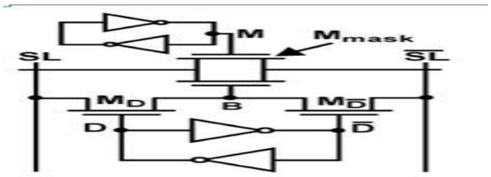


Fig. 3: NAND CAM cell operation

2.2 TCAM

The NOR and NAND cells that have been presented are binary CAM cells. Such cells store either logic “0” or logic “1”. Ternary cells, in addition, store an “X” value. The “X” value represents both “0” and “1”, allowing a wildcard operation. Wildcard operation means that an “X” value stored in a cell causes a match regardless of the input bit. As discussed earlier, this is a feature used in packet forwarding in Internet routers.

A ternary symbol can be encoded into two bits according to Table 1. Representation of these two bits can be done as D and $\overline{D}$. Note that although the D and $\overline{D}$ are not necessarily complementary, complementary notation maintained for the consistency with the binary CAM cell. Since two bits can represent 4 possible states, but ternary storage requires only three states, so disallow the state where D and $\overline{D}$ are both zero. To store a ternary value in a NOR cell, needed to add a second SRAM cell, as shown in Fig. 3. One bit, D, connects to the left Pulldown path and the other bit, and $\overline{D}$, connects to the right pulldown path, making the Pulldown paths independently controlled. Storing an “X” by setting both D and equal to logic “1”, which disables both pulldown paths and forces the cell to match regardless in the inputs. Storing a logic “1” by setting D=1 and $\overline{D}$ =1 and store a logic “0” by setting D=0 and $\overline{D}$=1. In addition to storing an “X”, the cell allows searching for an “X” by setting both SL and $\overline{SL}$ to logic “0”. This is an external don’t care that forces a match of a bit regardless of the stored bit. Although storing an “X” is possible only in ternary CAMs, an external “X” symbol possible in both binary and ternary CAMs. In cases where ternary operation is needed but only binary CAMs are available, it is possible to emulate ternary operation using two binary cells per ternary symbol.

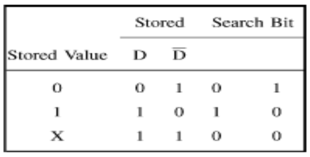
2.2.1 NOR CAM cell

As a modification to the ternary NOR cell of Fig. 2, implementing the pulldown transistors using pMOS devices and complementing the logic levels of the Searchlines and matchlines accordingly. Using pMOS transistors (instead of nMOS transistors) for the comparison circuitry allows for a more compact layout, due to reducing the number of spacing of p-diffusions to n-diffusions in the cell. In addition to increased density, the smaller area of the cell reduces wiring capacitance and therefore reduces power consumption. The trade-off that results from using minimum-size pMOS transistors, rather than minimum-size nMOS transistors, is that the pull down path will have a higher equivalent resistance, slowing down the search operation.

2.2.2 NAND CAM Cell

A NAND cell can be modified for ternary storage by adding storage for a mask bit at node M, as depicted in Table 1. When storing an “X”, set this mask bit to “1”. This forces transistor M_{mask} ON, regardless of the value of D, ensuring that the cell always matches. In addition to storing an “X”, the cell allows searching for an “X” by setting both SL and $\overline{SL}$ to logic “1”. Fig. 3 lists the stored encoding and search-bit encoding for the ternary NAND cell. Further minor modifications to CAM cells include mixing parts of the NAND and NOR cells, using dynamic-threshold techniques in silicon-on-insulator (SOI) processes, and alternating the logic level of the pulldown path to ground in the NOR cell. Currently, the NOR cell and the NAND cell are the prevalent core cells for providing storage and comparison circuitry in CMOS CAMs.

Table 1: TCAM CELL



	Stored		Search Bit	
Stored Value	D	$\overline{D}$		
0	0	1	0	1
1	1	0	1	0
X	1	1	0	0

3. PROPOSED METHOD

The Ternary CAM have the facility to store and search with the 0, 1 and don’t care state. This will give additional features of data forwarding and packet classification. To reduce the power consumption in Ternary CAM, an additional external circuit called precharge Controller can be used.

Traditional NOR TCAM is shown in the Fig. 4. Precharge all the MLs (ML_1 to MLm) when Pre signal is low and evaluates hit/miss of the search data when Pre goes high. Only the matching ML holds the charge to denote hit and remaining m-1; MLs drain their charge to denote miss. It is unnecessary to charge the mismatching MLs to full swing level, since those MLs have to be discharged in the evaluation phase. Unlike conventional CAM architectures, which during the Precharge phase merely charge the MLs, this CAM also predicts the MLs that would mismatch early in the Precharge phase. NOR and NAND CAM cells are the basic building elements of CAM architecture. NOR CAM is faster and less power efficient than NAND CAM. CAM constructed with NAND cells has charge sharing problems, and thus, they are not preferred by designers [1]. The proposed CAM is built with NOR CAM cell, and it creates a path between the nodes ML and ML_p when any of the bits in the ML mismatches

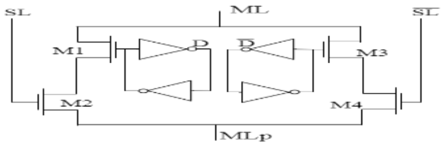


Fig. 4: Circuit of Ternary CAM

The above mentioned circuit is the Ternary CAM. Its construction is such that it can store the “0”, “1” and don’t care, here don’t care means irrespective of search data asserting the Matchline as 1. And it can be used to search with the “0”, “1” and don’t care, here don’t care means irrespective of the stored data asserting the Matchline as 1. To store the don’t care condition both D and $\overline{D}$ signals loaded with the logic 1. As D and $\overline{D}$ is 1, it gives 0 to the gates of M_1 and M_3. Gates of M_1 and M_3 at logic 0 means these two nMOS never turn ON, then there exists no path to discharge the Matchline. As the Matchline charge is 1, it is treated as Match. To search with the don’t care both SL and $\overline{SL}$ encoded with 0. As these two signals given to the gate of M_2 and M_4, these two nMOS never turn ON. And these two turned OFF nMOS creating no path from Matchline to the ground irrespective of the value at D and $\overline{D}$.

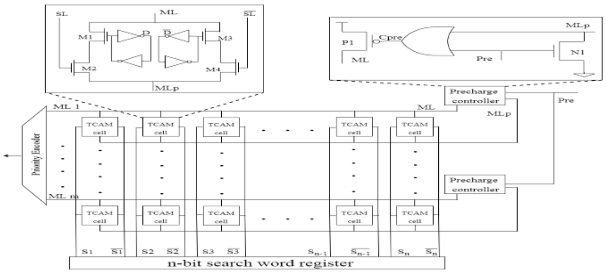


Fig. 5: Proposed system architecture

The Fig. 5 shows the main function of the Precharge controller which is to lively altering the Precharge time, so that mismatching ML won’t Precharges to full level. Regular Precharge signal which having the fixed Precharge and evaluation times is Pre and to the signal which is used for altering the Precharge time of MLs is C_{pre}. It will work such that if the search word misses then MLp directly connected with the ML. If a mismatch happens then MLp gets the charge of ML through the XOR circuit. If ML_p reaches enough charge to turn-on the OR gate then it stops further charging of ML. It will make the system’s power consumption reduced by approximately 74% of normal power consumption. The charge on the Match Line being used to evaluate the match or miss, so if the Match Line swing reduced in the case of miss, it will increase system speed. In the evaluation phase to make sure the mismatch occurred, the ML charge discharged through the MOS N1 using Pre. If a match occurs, then ML will be disconnected from ML_p and in this case Pre same as $C_{pre.}$ Hence ML will be charged full swing level in match case only

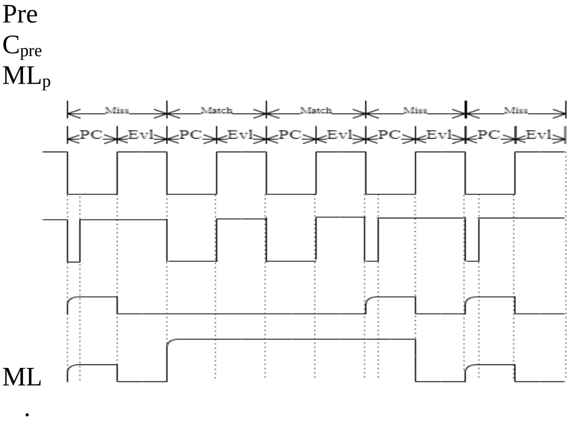


Fig. 6: Functionality of Precharge controller

The functionality of a Precharge controller for all possible subsequent cases is presented in Fig. 6. Initially, the nodes ML and ML_P are assumed to be without charge. When ML mismatches, C_{Pre} goes high to terminate charging of the ML as soon as the node MLP gets sufficient charge to drive OR gate high. C_{Pre} replicates Pre in the case of match to charge ML to full swing. In the case of miss followed by match, ML carries the full swing level of previous state even though C_{Pre} turns off Precharge MOS P1 early.

The Fig. 6 is the Ternary CAM with the Precharge controller which will reduces the power consumption in the Ternary CAM using the additional external circuit called Precharged controller. This will reduce the power consumption in CAM by reducing the voltage swing of the Match Line and Switching activity of the circuit

.

4. RESULTS

The main reasons for the power consumption are due to the charging and discharging of the matchlines. In the TCAM we are taking voltage source as Vdd=1V. The power in the consumption can be controlled by using the precharge controller. The Cpre and precharge signal is same when match case found and the matchlines are fully charged. In the case of the mismatch the matchlines are not been fully charged by the way the power reduction is obtained in the TCAM as shown in the Fig. 7.

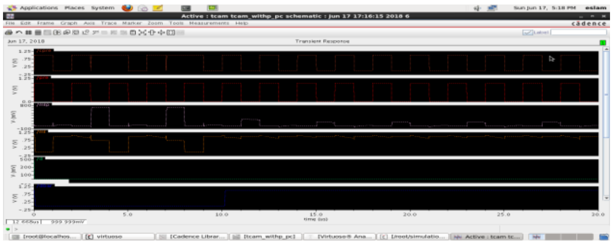

Applications Places System
Sun Jun 17, 5:18 PM eslam
Active : tcam tcam_withp_pc schematic : Jun 17 17:16:15 2018 6
File Edit Frame Graph Axis Trace Marker Zoom Tools Measurements Help
cadence
Jun 17, 2018
Transient Response
V (V)
V (mV)
0
5.0
10.0
15.0
20.0
25.0
time (us)
virtuoso
[Cadence Librar...
[tcam_withp_pc]
[Virtuoso® Ana...
Active : tcam tc...

Fig. 7: Simulation of TCAM.

The below Table 2. Shows the comparison between the Binary CAM and TCAM with the precharge controller and without precharge controller.

Table 2: Comparisons

Type of CAM	Power Consumption
Binary CAM with PC	9.68 n Watts
Binary CAM without PC	138.9 μ Watts
TCAM with PC	0.124 μ Watts
TCAM without PC	78.8 μ Watts

5. CONCLUSION

The power consumption in the TCAM is mainly due to the charging and discharging of the match Line. This charging and discharging time of the Matchline effects speed of the CAM also. Here with the help of Precharge Controller the voltage swing of the matchline is controlled. By reducing the voltage swing of matchline, the power consumption in TCAM reduces and speed of the CAM also increases.

6. REFERENCES

[1] Mohammed Zackriya V and Harish M. Kittur, “Content Addressable Memory—Early Predict and Terminate Precharge of Match-Line”: IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 25, NO. 1, JANUARY 2017.

[2] K. Pagiamtzis and A. Sheikholeslami, “Content-addressable memory (CAM) circuits and architectures: A tutorial and survey,” IEEE J. Solid-State Circuits, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[3] S. M. Jalaleddine, “Associative memories and processors: The exact match paradigm,” J. King Saud Univ.-Comput. Inf. Sci., vol. 11, pp. 45–67, Mar. 2013. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S1319157899800032

[4] A. Bremler-Barr and D. Hendler, “Space-efficient TCAM-based classification using gray coding,” IEEE Trans. Comput., vol. 61, no. 1, pp. 18–30, Jan. 2012.

[5] Y.-C. Shin, R. Sridhar, V. Demjanenko, P. W. Palumbo, and S. N. Srihari, “A special-purpose content addressable memory chip for real-time image processing,” IEEE J. Solid-State Circuits, vol. 27, no. 5, pp. 737–744, May 1992.

[6] S. K. Maurya and L. T. Clark, “A dynamic longest prefix matching content addressable memory for IP routing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 6, pp. 963–972, Jun. 2011.