Basci, F., & Kocak, T. (2004). Statistically partitioned, low power TCAM. In 2nd Annual IEEE Northeast Workshop on Circuits and Systems, Montreal, Canada. (pp. 129 - 132). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/NEWCAS.2004.1359039

Statistically Partitioned, Low power TCAM

Faysal Basci and Taskin Kocak

Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL, 32816-2450

Abstract— **Network search engines based on Ternary CAMs are widely used in routers. However, due to parallel search nature of TCAMs power consumption becomes a critical issue. In this work we propose an architecture that partitions the lookup table into multiple TCAM portions based on individual TCAM cell status and achieves up to 30% power reduction.**

Key-words: CAM, ternary CAM, low power, IP lookup, routing table

I. Introduction

Content addressable memory (CAM) provides access by data rather than by memory address. CAM's has higher advantage over other memory search algorithms, such as look-aside tag buffers, binary or tree based searches. However, this performance advantage comes with a price of higher silicon area, and higher power consumption. Today, commercial CAM chips or embedded CAMs are being utilized in lots of different applications including pattern recognition, neural networks, encryption, firewalls, switches and routers. In this paper, we are interested in the ones for networking applications. CAMs are generally used in packet forwarding lookup tables in the routers [1]. It is used to extract and process the address information from incoming packets: compare the destination address of the packet with the stored data and if a match occurs, associated routing information is given to the forwarding circuit. Despite their performance advantages, CAMs have serious power consumption problems. In [1], it is reported that a 500K-entry lookup table for an IPv6 network processor, formed by CAM chips will consume up to 133 W, which is around 3 W per chip. Moreover, in [6] it is reported that a 64K-word by 40-bit CAM chip consumes 5.2 W.

In this paper, we address the power consumption issues in ternary CAMs (TCAMs) used in IP forwarding tables and propose an approach which reduces the expected power consumption. In section II, an introduction to fundamentals of CAMs is presented, section III discusses usage of TCAMs in IP forwarding, and section IV presents our statistical partitioning approach. In section V, power consumption formulation of statistically partitioned TCAM is presented, and section VI discusses the experimental results. Finally, in section VIII, we present the conclusion.

II. CAM Basics

Fig. 1 shows a basic CAM architecture [2]. This is an n-bit k-word CAM. Data stored in CAM is searched by applying the reference word to bit lines, which run vertically, through bit line drivers. Any search word presented is searched in parallel. Due to this, the search is very fast, however this also implies that for each search operation all of the cells are utilized, resulting in excessive power consumption. There

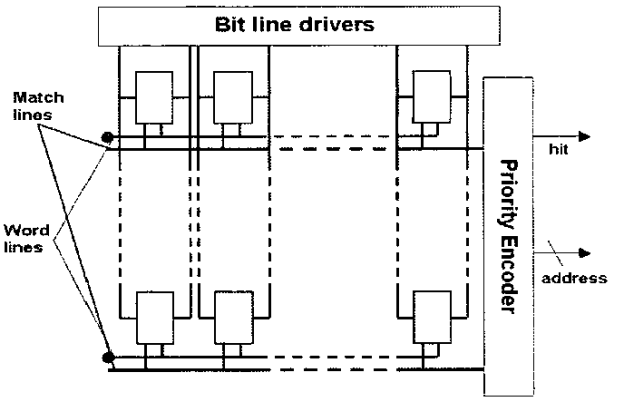

Fig. 1. CAM architecture. Data to be searched (or stored) is fed through bit line drivers, address is the encoded address of the matched data. Hit is a control signal indicating if a match found. Word lines are used when writing data to the cells.

are two classes of CAMs, binary CAMs and ternary CAMs. A binary CAM cell can store either a ***0*** or a ***1***. TCAMs, on the other hand, has the capability to store a "don't care (***x***)". To store an ***x*** we need an extra bit. This can be achieved by simply combining two binary CAM cells [3] or a more elaborate CAM cell design is also possible as in [7]. In TCAM cells the stored data is encoded to represent ***0***, ***1*** and ***x***. CAM architectures utilizes wire-anding; before a search operation is performed, the matchline is precharged and search lines are discharged. During a search operation, if a cell matches the stored data with the one on the search line it will do nothing, however when a mismatch is detected, the cell will pull down the match line to low. So, even if one bit mismatches, a mismatch will be issued. On the other hand, if a don't care is stored in a cell then the search data will be discarded by that cell. In a way it will behave like a matching cell.

CAMs (or TCAMs) consume power mainly in 3 parts: matchline (and searchline) precharging (pre-discharging), comparison, and clock and control signaling. Most of the power is consumed during pre-charging operation [5]. Therefore most of the CAM designs try to minimize pre-charging events as in [4] and [8]. Some approaches try to reduce the voltage swing across search and matchlines as in [9] whereas some approaches use system level optimizations as in [10].

0-7803-8322-2/04/$20.00 ©2004 IEEE.

Other system level approaches utilizes the application specific properties, as in the case of IP forwarding engines. Next section discusses the usage of TCAMs in IP forwarding.

III. TCAMs IN IP FORWARDING

Nowadays, IP lookups are based on classless interdomain routing (CIDR) scheme. After the adoption of CIDR in 1993, IP routes have been characterized by a routing prefix and the prefix length. In CIDR scheme, route lookups aim at the longest prefix match (LPM).

As TCAM provide a don't care storage capability, it is a favorable lookup hardware; in that, in IP lookup engines, when an entry is stored in a TCAM, depending on its prefix, some of its rightmost bits will be stored as *x*. For example in IPv4, for a 24 bit prefix, last 8 bits will be *x*. Moreover, because of *x* storage capability, routing entries belong to different prefix sets can be placed on the same chip. Generally routing table entries stored in TCAMs are ordered according to their prefixes. For example, the highest prefix set lies at the top of the entries (or lowest addresses), and the lowest prefix set lies at the bottom [12]. When multiple matches occur, a priority encoder chooses the longest matching prefix, which, in this example, is the match with lowest address.

IV. STATISTICAL PARTITIONING

When we look at the prefix distribution in the core routers we see that they all have a similar characteristic [11]. A great portion of the entries are accumulated at prefix set 24. If we just check this prefix set, we can achieve around 50% hit ratio provided that we have a random traffic pattern. However, as our aim is to find the longest matching prefix, we should also look at the higher prefixes. Indeed, prefixes higher then 24 are considerably rare. We exploit this fact and propose to partition the routing lookup table as shown in Fig. 2. The routing lookup table is divided into two parts, TCAM1 and TCAM2, respectively. In a lookup operation, first TCAM1 will be searched and if a mismatch occurs, then TCAM2 will be searched.

There is a buffer between TCAM1 and TCAM2 which is used to store the current search word. This enables pipelining the search operation. In the case of a mismatch in TCAM1, TCAM2 uses the value stored in the buffer to do a search. This way, TCAM1 can accept a new search word each time. Although pipelining ensure a sustained average throughput of one result per clock cycle, the average latency of the search operation will be higher. However, considering the overall latency of packet processing and queuing delays in network nodes, this latency will not be significant.

Beside the distribution of prefixes we should also take power consumption and average latency into account when partitioning the table. When we have all of these figures we can deduce an optimal partitioning.

V. POWER CONSUMPTION FORMULATION OF CAMs

Power consumption in a TCAM cell can be written as follows:

$$P_{TCAM} = P_{STC} + P_{CLK} + P_{MS} + P_{MT} + P_X \quad (1)$$

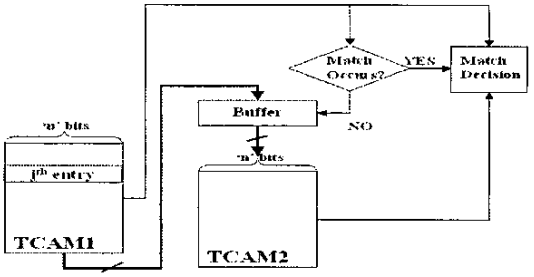

Fig. 2. Partitioned TCAM

where P_{STC}, P_{CLK} represents the static power consumption and power dissipation due the clock circuitry, and P_{MS}, P_{MT}, P_X represent the average dynamic power consumption in the case of a mismatch, match and don't care, respectively. Please note that latter three also include the power consumption due to matchline and searchline switchings. Here we are concerned about the dynamic power consumption caused by search operation. This is the dominant power drain, even under very low load. In fact, for networking applications the TCAM generally works with full load.

Let r_i represent the set of entries associated with a prefix ***i***, and prefix sets stored in TCAM1 be $\{r_m, \ldots r_n\}$ where m < n, and prefix sets stored in TCAM2 be $\{r_1, r_2, \ldots r_{m-1}\}$. The number of entries in set r_i can be represented by N_i, where 1< i <n (n=32 for IPv4). The number of don't care cells in a prefix ***i*** is ***n-i***. Then power consumption for a comparison operation of a word that belongs to the set r_i is:

$$P^i_{COMP_j} = P_{MT} * Mt^i_j + P_{MS} * Ms^i_j + P_X * (n - i) \quad (2)$$

where Mt^i_j represents the number of matching cells and Ms^i_j represents the number of mismatching cells in stored word ***j***. We can write the power consumptions for each TCAM portion as:

$$P_{TCAM1} = \sum_{i=m}^{n} [N_i * P_X * (n - i) + P_{MT} \sum_{j=1}^{N_i} Mt^i_j + P_{MS} \sum_{j=1}^{N_i} Ms^i_j] \quad (3)$$

$$P_{TCAM2} = \sum_{i=1}^{m-1} [N_i * P_X * (n - i) + P_{MT} \sum_{j=1}^{N_i} Mt^i_j + P_{MS} \sum_{j=1}^{N_i} Ms^i_j] \quad (4)$$

Let ρ_1 represent the fraction of entries that resides in TCAM1, then the total power consumption for a search operation can be written as:

$$P_{TOTAL} = P_{TCAM1} + (1 - \rho_o * \rho_1) * P_{TCAM2} \quad (5)$$

where ρ_o represents the probability that the search word has a match in the whole table.

VI. Experimental results

The value of $\boldsymbol{m}$ depends on the routing table entries and traffic pattern, so it will be different for each lookup table. To calculate the expected power consumption for different values of $\boldsymbol{m}$, for a typical lookup table, we have done calculations for a few networks including Telstra and Reach [11]. We have implemented an example TCAM circuit in TSMC $0.18\mu m$ CMOS technology and we run it at 100 MHz. P_{MT}, P_{MS} and P_X are obtained from Cadence Spectre simulations. In these simulations, number of TCAM cells varied and in each case TCAM block is tested with different matching and mismatching bit patterns. Finally, all of the obtained results are averaged for matching, miss-matching and don't care cells. The results are as follows:

$$P_{MT} = 397\ nW, P_{MS} = 515\ nW, P_X = 336\ nW \quad (6)$$

Fig. 3, shows the expected dynamic power consumption in

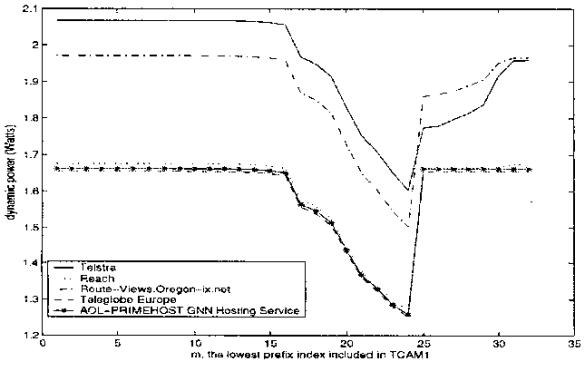

Fig. 3. Power consumption in search operation for partitioned CAM with varying $\boldsymbol{m}$

search operations for 5 different networks, based on the above values. It can be seen that for all of the networks the minimum power consumption is achieved for the value of $\boldsymbol{m}$=24. Power savings gained by doing that are between 20 to 24%. We can also use expected power - latency square product (PLSP) as another metric. If we represent search latency in terms of clock cycles with $\boldsymbol{L}$, expected value of the latency can be shown to be equal to the following :

$$E\{L\} = 2 - \rho_0 \quad (7)$$

Here we assume that, the system utilizing the results of the TCAM have the ability to handle two results per clock cycle, otherwise, there will be a stall stage and average latency will be equal to 2, independent of the hit rate.

Fig. 4 shows the plot of PLSP for the same network set. It can be seen that the until $\boldsymbol{m}$=25 the PLSP increases very slowly, and then it shows a sharp increase. When we calculate average latency for the case where $\boldsymbol{m}$=24, we see that it varies from 1.36 to 1.45 clock cycles. The same experiment is run for

TABLE I

Expected power savings and average latency when partitioning TCAM into three

Network	m1, m2	Power saving(%)	Avg. latency (cc)
Telstra	25, 24	30.50	1.6754
Reach	24, 21	28.96	1.6871
Route-V.Oregon	24, 21	27.90	1.6321
Teleglobe Europe	24, 21	29.20	1.6932
AOL-PR. GNN	24, 21	29.12	1.6916

the case of 3 partitions. This time, we have 3 TCAM parts, with **m1**, **m1** as partitioning point. The obtained minimum power values are tabulated in Table I. From the table it can be seen that with some added latency, power consumption can be reduced by up to 30%. Experiments we conducted show that further partitioning does not increase power savings considerably.

In the experiments, it is assumed that the number of cells that return a match is equal to the number of cells returning a mismatch. However, to exemplify the power consumption figure for different miss rates a simulation is done for Telstra network. The result is shown in Figure 5. As can be seen the optimal value of $\boldsymbol{m}$ is not affected, however the power consumption figures are scaled up or down with miss rate.

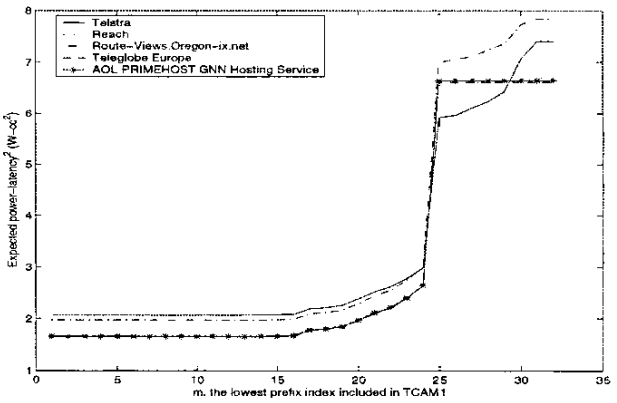

Fig. 4. Expected $power - latency^2$ product (PLSP) with varying $\boldsymbol{m}$

VII. Discussion

When the TCAM is partitioned, It can be claimed that except for the first partition the other(s) can be implemented using less number of bits. For example in the case of two partitions, if the the prefix down to 24 is stored in the first partition, first TCAM would be 32 bits wide and second one need to be only 23 bits wide. However, that would limit the ability to reconfigure the partitioning. On the other hand, software control of partitioning without compromising from speed or power reduction complicates design considerably. Giving full software control over partitioning requires that each and every word in the TCAM system to have an input selection mechanism, which will increase the amount of silicon area needed and will increase the power consumption. However,

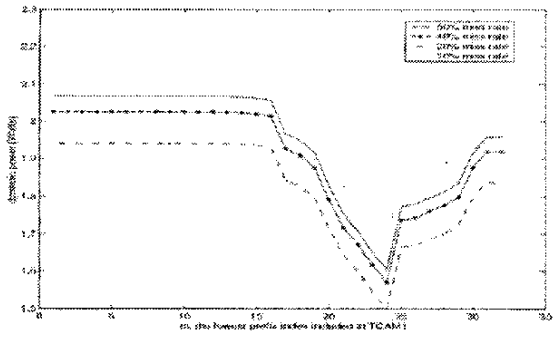

Fig. 5. Power consumption in search operation for partitioned CAM with varying m for Telstra Network with different miss rates for individual cells

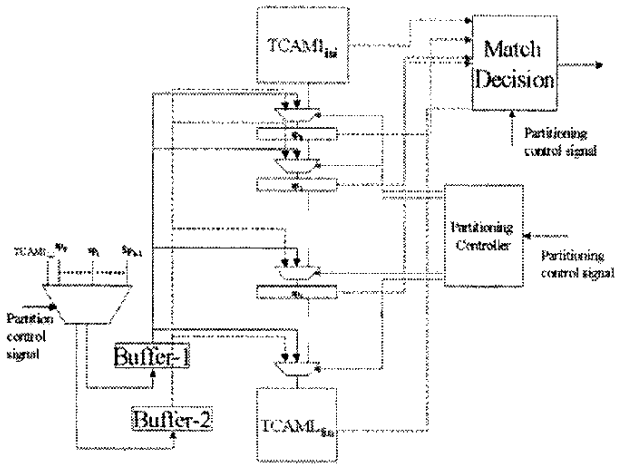

Fig. 6. Architecture allowing partial software control for partitioning

the prefix sets with a lot of entries are centered between 20 to 25 range. As can be seen from the results obtained, for the case of two partitions, partitioning is generally most beneficial at prefix 24 and for the case of three partitions, the partitioning again occurs in 20 to 25 range. A compromise would be to make only the part of TCAM, where the partitioning takes place, reconfigurable. Furthermore, instead of providing an input selection mechanism to each word, a group of word can use the same input selection mechanism. This way, the speed and power penalty could be minimized. Figure 6 shows an architecture allowing the partial software control for the case of three partitions. In this architecture, $TCAM1_{ini}$ represents the fixed part of the first partition, and $TCAML_{fin}$ is the fixed part of the last partition. Configurable sub-partitions $\{sp_0, \ldots sp_{k-1}\}$ have a multiplexer at their input, and it is controlled by the partitioning controller. Among all of the multiplexers either two (three-partitions) or one (two-partitions) of them will get their input from buffers and from that point on, that input will be propagated down to either the next partitioning point or $TCAM2_{fin}$ for three and two partitions respectively. For example, for the case of two partitions, if the sub partition i gets input from buffer-1 all of the sub partitions between $TCAM1_{ini}$ and sp_i will have the same input, that means TCAM1 contains all the sub partitions up to sp_i. And every other sub partition below sp_i and $TCAML_{fin}$ will constitute the last partition. Another multiplexer is connected at the input of the buffer, which selects one of the outputs of the sub partitions. If sp_i is the partitioning point then the multiplexer will choose the output of sp_{i-1}. This architecture, actually, allows us to use less number of bits in fixed portion the last partition. This way added power consumption due to selection mechanism can be compensated.

VIII. Conclusion

We have presented a partitioning scheme, which utilizes statistical distribution of prefixes and individual power consumption of cells in the cases of match, mismatch and don't cares. We showed that indeed the partitioning helps reducing the power consumption in IP lookup applications. Partitioning into two and three, reduce power consumption considerably, whereas further partitioning shows little improvement. We also represented an architecture which allows software control over partitioning. This architecture can be made more flexible with compromising power and silicon area.

References

[1] White Paper. EZCHIP Technologies. "Ipv6 to Ipv4 is not merely 50 more." (web: http://www.ezchip.com/html/tech IPv6.html)

[2] K. J. Schultz, "Content-addressable memory core cells: A survey." *Integration, the VLSI Journal 23*, Page(s): 171-188, 1997

[3] R. Sergio, R. Chavez, "Encoding don't cares in static and dynamic content-addressable memories." *IEEE Transactions on Circuits and Systems-II : Analog and Digital Signal Processing* Vol. 39, No.8, August 1992

[4] G. Thirugnanam, N. Vijaykrishnan, M.J. Irwin, "A novel low power CAM design." *14th Annual IEEE International ASIC/SOC Conference Proceedings*, Page(s): 198 -202, 12-15 Sept. 2001

[5] H.Y. Liang Hsiao, D.H. Wang, C.W. Jen. "Power modeling and low-power design of content addressable memories " *ISCAS, The 2001 IEEE International Symposium on Circuits and Systems*, Volume: 4 , Page(s): 926 -929, 6-9 May 2001

[6] F. Hafai, K.J. Schultz, G.F.R. Gibson, A.G. Bluschke, D.E. Somppi, "Fully parallel 30-MHz, 2.5-Mb CAM." *IEEE Journal of Solid-State Circuits*, Volume: 33 Issue: 11, Page(s): 1690 -1696, Nov. 1998

[7] I. Arsovski, T. Chandler, A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme" *IEEE Journal of Solid-State Circuits*, Volume: 38 Issue: 1, Page(s): 155 -158, Jan. 2003

[8] T. Chadwick, T. Gordon, R. Nadkarni, J. Rowland, "An ASIC-embedded content addressable memory with power-saving and design for test features." *IEEE Conference on Custom Integrated Circuits*, Page(s): 183 -186, 6-9 May 2001

[9] H. Miyatake, M. Tanaka, Y. Mori, "A design for high-speed low-power CMOS fully parallel content addressable memory macros." *IEEE Journal of Solid- State Circuits*, Volume: 36 Issue: 6, Page(s): 956 -968, June 2001

[10] C.S. Lin, J.C. Chang, B.D. Liu, "A low-power precomputation-based fully parallel content-addressable memory." *IEEE Journal of Solid-State Circuits*, Volume: 38 Issue: 4, Page(s): 654 -662, April 2003

[11] BGP table statistics, (web: http://bgp.potaroo.net/), September 22, 2003

[12] D.Shah, P.Gupta, "Fast updating algorithms for TCAM" *IEEE Micro*, Volume: 21 Issue: 1, Page(s): 36 -47, Jan.-Feb. 2001