Modeling TCAM Power for Next Generation Network Devices

Banit Agrawal Timothy Sherwood

Computer Science Department
University of California, Santa Barbara
{banit,sherwood}@cs.ucsb.edu

Abstract

Applications in Computer Networks often require high throughput access to large data structures for lookup and classification. Many advanced algorithms exist to speed these search primitives on network processors, general purpose machines, and even custom ASICs. However, supporting these applications with standard memories requires very careful analysis of access patterns, and achieving worst case performance can be quite difficult and complex. A simple solution is often possible if a Ternary CAM is used to perform a fully parallel search across the entire data set. Unfortunately, this parallelism means that large portions of the chip are switching during each cycle, causing large amounts of power to be consumed. While researchers have begun to explore new ways of managing the power consumption, quantifying design alternatives is difficult due to a lack of available models. In this paper we examine the structure inside a modern TCAM and present a simple, yet accurate, power model. We present techniques to estimate the dynamic power consumption of a large TCAM. We validate the model using industrial TCAM datasheets and prior published works. We present an extensive analysis of the model by varying various architectural parameters. We also describe how new network algorithms have the potential to address the growing problem of power management in next-generation network devices.

Keywords: *CAM, Ternary CAM, TCAM, SRAM, Power, Modeling, Router, Network algorithms.*

1 Introduction

Network systems are continually called upon to support different applications in the network at line speed. Whether it is a classic problem such as IP-lookup, or an emerging domain such as worm detection, many network algorithms require the ability to index and search large amounts of state with incredibly high throughput. For example, in the case of IP-lookup, the state is a routing table, while for worm detection the state may be a set of patterns to match. While there is a great deal of work on advanced algorithms to speed the search through this state with traditional memories, the complexities of implementation motivate some to seek a memory design that directly supports the search primitives. Content Addressable Memories provide the required search capabilities with a minimum of additional cost and complexity.

Content Addressable Memories (CAMs), and specifically Ternary CAMs (TCAMs), are memories that are mostly used in networking devices. CAMs provide read and write such as a normal memory, but additionally support *search* which will find the index of any matching data in the *entire* memory. A TCAM in particular can include wildcard bits which will match both one and zero. These wildcards can be used on both the access operations of the memory (indicating some bits of the search are "don't cares") or can be stored with the data itself (indicating some bits of the data should not be used for determining a match). The fully parallel search provided by TCAM eases the implementation of many complex operations such as routing table lookup. Because the TCAM searches every location in memory at once, the ordering of the elements in the TCAM is less important and large indexing structures can often times be entirely avoided. This parallel search directly implements the requirements of some applications (such as IP-lookup), and can serve as the building block of more complex searching schemes [24]. TCAM is also used in other high-speed networking applications such as packet classification [10, 13, 24], access list control, pattern matching for intrusion detection [30]. TCAM are also being used with network processors as a coprocessor to complement the network processors in several applications such as packet classification and routing lookup. While there are many advantages, a fully parallel search of memory does not come for free. In the power constrained situations that most high performance routers find themselves, these searches can, if unoptimized, easily consume tens or hundreds of Watts. To give as an example of growing power concerns, Cisco Systems provides a 600-watt redundant AC power system, which can support up to four 150 watt external network devices such as routers [25]. Now TCAMs are being increasingly used

as an integral component into the next-generation routers and next-generation network search engines (NSEs) [21]. Some of the previous work have addressed the issue of power consideration of network processors [5, 14]. But they do not account for the TCAM power consumption in their framework. In order for the network community to begin to address these problems, a simple yet accurate TCAM power model is required.

The traditional computer architecture community has been well served by the adoption of Cacti [29], eCacti [15], and other architecture level memory models. These models help designers evaluate various on-chip cache designs and have led to a variety of different research endeavors that seek to make tradeoffs across the traditional boundaries of architecture and circuits. We discuss about some of these related works in Section 2. In this paper we present a power model for TCAM with the intent of clarifying and encapsulating many of the most important aspects. While there are many different TCAM designs that have been proposed [1, 18, 12, 7, 4], most modern designs share a similar nand based cell design. We provide a description of TCAM operation as it relates to power modeling in Section 3. Building on this understanding of the TCAM internals, we abstract away the aspects of lesser importance, and converge on a simple to use model based on simple but effective approximations of the line capacitances and switching activities in Section 4.

One of our main contributions is a model that is more accurate and useful than a simple Watts/bit approximation. If the architecture and networking communities have to develop new algorithms and TCAM power management schemes, we must embrace a model that exposes the *opportunities for improvement*. This means that we must consider the effects of different length TCAM entries on power consumption, the banking of TCAMs to reduce word and match line capacitance, and the effect of the priority encoder. In Section 5, we present our model and show that it matches closely with recently published circuit data for a variety of configurations.

2 Related Work

Although TCAM is very useful in high-speed networking applications, most network designers worry about the power dissipation/consumption in TCAM. A number of techniques have been proposed to reduce the power consumption in TCAM [17, 20, 22] by enabling search in only a subset of the TCAM. In CoolCAM [17], the authors assume that the power consumption is proportional to the number of rows. They provide a set of clever algorithms to search less number of rows, which reduces the total power consumption for search operation. Although we find their assumptions to be a good approximation for making relative estimations, an absolute quantitative figure for power saving in terms of Joules or Watts would be better. In EaseCAM [20], a page based scheme is used to reduce the power consumption and they base their savings on the CAM implementation inside Cacti (which is different than a TCAM). Besides rows pruning to reduce the power consumption, there is also some work which proposes to reduce the number of bits in comparison [11]. Therefore, we need a power model for TCAM which can also take column bits as the input parameters to estimate the dynamic power.

There are some other general power modeling tools such as Orion [27], Wattch [3], Cacti [23], which model content addressable memory (CAM) power consumption in different ways. Hsiao et al. [9] provide a power model for CAM that accounts for evaluation power, input transition power and clock power. These models do not account for TCAM cell and the transistor sizing of various transistors in TCAM cell. We have found no previous work that provides a publicly available power modeling tool for TCAM. As an imminent requirement, we provide a publicly available modeling tool for TCAM, which can take high-level architectural parameters as input and provides the dynamic power as output.

Our model is scalable with respect to CMOS technology because it takes CMOS feature size and TCAM cell layout parameters as input parameters. Hence, we can estimate the dynamic power of any TCAM design by taking care of the TCAM cell used. Our model also takes architectural parameters as inputs such as number of rows, number of search bits, and number of banks. This will help in exploring new network algorithms that have the potential to address the growing problem of power management in next-generation network devices.

3 TCAM Structures

To ground our modeling technique, it is worth describing the internal structure of of a modern TCAM design, along with the various design options and parameters. In the following subsection we describe the read, write, and search operations of a high speed TCAM, and describe the primary channels of power consumption to motivate our model.

3.1 TCAM Architecture

Before we present the details of our model, it is worth reviewing the important structures in a TCAM, particularly as they relate to power consumption. In particular we will concentrate primarily on the aspects relating to search as this dominates the overall system power on an active device. The fundamental operations of a TCAM are

• *write* – updates the entries in a row of TCAM cells

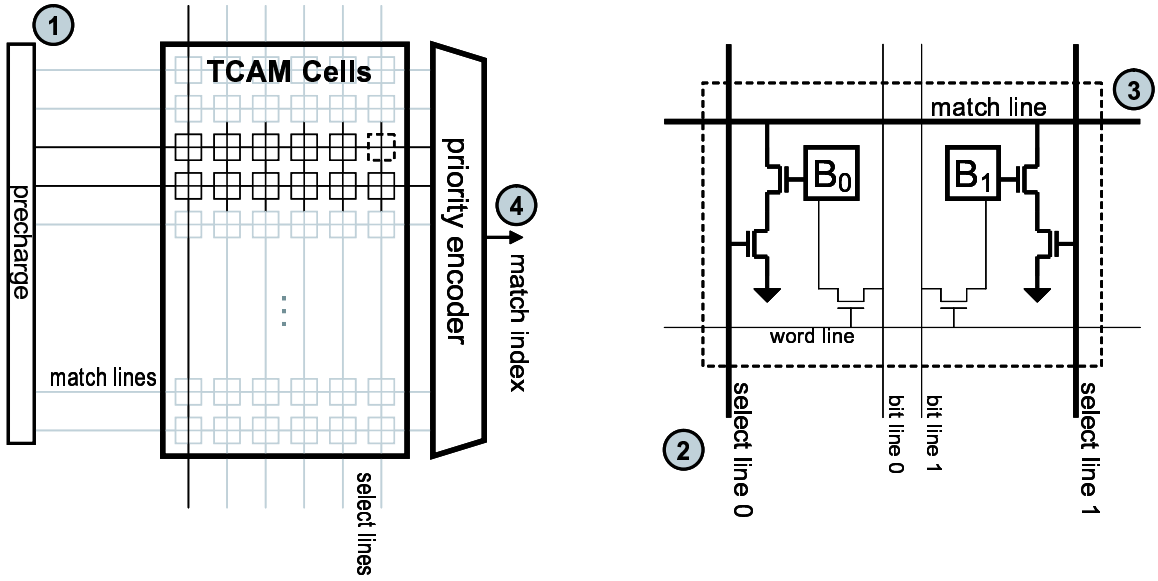

Figure 1: A conventional TCAM architecture along with TCAM cell structure is shown above. The major components of TCAM architecture are precharge circuit, TCAM cells array, and priority encoder as shown on the left hand side. On the right hand side, a typical TCAM cell structure with *matchlines* and *searchlines* is shown. The bit B_0 and B_1 can be static or dynamic storage cells.

• *read* – reads the contents of a row of TCAM cells
• *search* – finds a match across all TCAM rows

Figure 1 shows the generic design of a NAND based Ternary CAM. At a high level (pictured on the left) the TCAM has 3 major components: the decoder/precharge unit, the actual array of TCAM cells, and the priority encoder. The decoder is needed for random access addressing during a read or a write, while the precharge unit is needed to precharge the match lines before a search. During a search, all of the bits in the TCAM cells are compared against the bits driven on the select line. After the search is complete, the rows which match will have their match line set high, and all others will be set low. Because there is the potential for multiple matches on a given search, especially when searches and data include "don't care" bits, a final step is needed to pick the one of R bits which is the highest priority match. Typically this is implemented by keeping the TCAM in partially sorted order by priority, and using a priority encoder to select the first row that matches.

Central to the power of Ternary CAM over a traditional CAM is the ability to include "don't care" bits in both the search and in the data. This wildcard matching enabled by TCAM is a natural fit to many applications including the matching prefix problem from IP-lookup [17], sorting [19], or range queries for packet classification [13, 24]. A "don't care" bit in the search pattern requests that a particular column of bits not be taken into consideration for determining which rows are the match. This can be used to handle data of variable length or find all entries with a common prefix. The ternary part of TCAM comes from the fact that the cell itself can encode a "don't care" bit, and thus can be in one of three states: 'match 0', 'match 1', and 'match both'. Typically this is stored in the TCAM as two bits as shown in the right half of Figure 1.

The biggest benefit of using a TCAM is that the comparison happens directly in the cells, which means that to understand the power consumed by the comparison operation we must understand the internals of a TCAM cell. The right side of Figure 1 shows the design of a TCAM cell with the actual bit storage of the two states bits abstracted away as B_0 and B_1. The bold lines show the paths relevant to the search operation. A logical zero is stored in the TCAM cell when $B_0 = 1$ and $B_1 = 0$, while a logical one is stored when $B_0 = 0$ and $B_1 = 1$. The position of the '1' in either B_0 or B_1 determines where the select line comparison should occur. A logical "don't care" is stored by insuring that *no* comparison is done for this bit which is achieved with $B_0 = 0$ and $B_1 = 0$.

The cell B_0 and B_1 can be either static TCAM cells (4 CMOS Transistors) or can be dynamic TCAM cells (1 CMOS Transisor). Later we show that our model calculates the dynamic power by taking TCAM cell layout into consideration. Hence, we are able to measure the dy-

namic power of any TCAM design by using the TCAM cell layout parameters.

3.2 Search Operation Energy

Once the TCAM cells have been set to one of the three legal states, a search can be done. The four steps shown in Figure 1 are as follows. (1) Before the search occurs, all of the select lines are set low to insure that the match line is insulated from ground. The match lines are then precharged, meaning that the line is set high but then disconnected from power so that the value of the line is essentially stored in the capacitance of the wire. (2) Once precharged, the select lines are driven to force the comparison to take place. To search for a logical one $Select_0$ is driven high, while a logical zero match can be found by driving $Select_1$ high. To do a don't care search, neither of the $Select$ lines for a given bit are used. If there is a mismatch, then the select line will be high, and the bit will be high resulting in a path from the match line to ground. Thus, the charge stored on the match line will remain intact *only* if there are no mis-matches (3). In this way the match line is acting as a very long NAND gate, combining the local match results in each cell to effect the status of the match line. Once the proper match lines have drained, there may still be multiple entries that match the query. Arbitration between these matches is often performed by taking the *first* match in a specified order, letting the position of the entry enforce the priority (4). In the case of IP-lookup this corresponds nicely to longest prefix matching as long as the entries are inserted into the TCAM in partially sorted order. The priority encoder performs this function across all of the match lines in the design.

The power consumption in the design comes primarily from the combined effect of step 1 and 3. Every match lines in the system is filled with charge and then dumped to ground on every access. The total capacitance of these lines, combined with the operating voltage of the match lines, determine to first order the power consumption of the system. The other significant components are the toggling of the select lines (2), and the priority encoder switching (4). In this paper we precisely quantify the access energy from each of these parts so as to provide an easy to use, general purpose, TCAM power model.

4 Modeling of TCAM

In the previous section we described at a high level the internal TCAM architecture. It is now time to explain the power modeling approach we have taken, and derive the most important parameters. While there is some low level circuits discussion, our end result will be a simple to use and intuitive model that has been validated against real hardware designs for the architecture and networking communities to build from. We specifically target search operation as this is the most crucial and most power-consuming operation in networking applications such as routing lookup, classification, and string matching for intrusion detection. Given that all of these applications are performance driven, and that the TCAM is likely to be accessed with very high frequency, we restrict our discussion to a dynamic power model as dynamic power will dominate static power in this domain.

Of course the power consumption of any hardware component is mainly dependent on the voltage supply (V_{dd}), equivalent capacitance(C_{eq}) and the operating frequency (f). Most of this power consumption in TCAM is due to charging and discharging of various control lines. The operating frequency decides how fast these lines are charged/discharged. To find the frequency of TCAM, we need to find the access time of various components in TCAM which we describe in Section 4.4. The worst-case dynamic power for a hardware component with equivalent capacitance (C_{eq}) and operating frequency (f) is given in equation 1.

$$dyn\text{-}power \quad = \quad C_{eq} \quad * \quad V_{dd}^2 \quad * \quad f \qquad (1)$$

The worst case power consumption will be the same as the average case if on every cycle the line is toggled, switching from '0' to '1' or '1' to '0'. While an estimate based on this idea will provide a bound on the energy consumption, most real designs do not toggle every line every cycle. If the capacitance of a hardware component does not charge or discharge every cycle, then we need to find out the total switching activity over a period of time to calculate the actual dynamic power. The revised equation for dynamic power is shown in equation 2.

$$dyn\text{-}power \quad = \quad C_{eq} \quad * \quad V_{dd}^2 \quad * \quad Activity \quad * \quad f$$

$$Activity \; = \; average \;\; switching \;\; factor \;\; (\leq \; 1) \qquad (2)$$

We have already identified the hardware components that are responsible for the bulk of the power consumption in a search operation in Section 3. We model each of these hardware components separately and calculate the equivalent capacitance. While at first glance, the switching activity may seem dependent on the actual set of IP address traces, 5-tuple packet classification rules, or input patterns for matching, in reality it is highly independent of all these factors as we will describe. Initially, we provide the worst-case power consumption by assuming $Activity$ is 100%, but we then relax this by examining a more realistic worst case operating behavior.

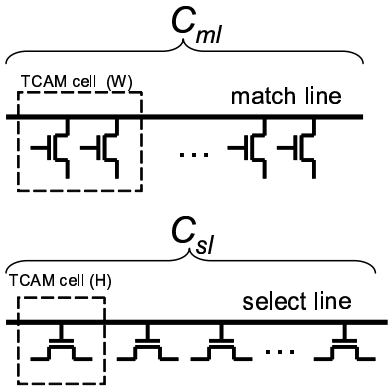

Figure 2: The capacitive loading effect is shown in the above figure for matchlines and searchlines. C_{ml} gets some load capacitance from the drain capacitance of comparison transistors, whereas C_{sl} gets some capacitive loading effect from the gate capacitance of the comparison NMOS transistor.

In the next subsections we explain the power modeling of the Match Lines, the Search Lines, and the Priority Encoder and the delay modeling of TCAM.

4.1 Match line power

The largest amount of energy dissipated by TCAM comes from the match line dissipation. Each and every access (which could be once a cycle), all of the match lines are charged and then all but a few are discharged leaving only the matches. The amount of energy required to perform this operation is a direct function of the total capacitance of all the match lines, as that determines the amount of charge required to precharge the line to the desired voltage. If we look back to Figure 2, we can see that the match lines run the entire length of a row, and that hanging off of the line are many transistors each of which is a potential path to ground (if a mis-match occurs). If we remove all of the components not directly connected to the match line, we end up with the top half of Figure 2.

The capacitance of the match line is a function of both the length of the wire, and the number of transistors which source the line. For each cell in a row, there are two such transistors, one for each select line. Assuming the width of the TCAM is known, we can calculate the capacitance of a match line over l bits as shown in Equation 3 (Figure 4.1).

Equation 3 (Figure 4.1) will estimate the capacitance of a single match line. We calculate the capacitances of the components ($C_{metal-wire-per-cell}$, $C_{drain-compareT}$, $C_{nandinv-gate}$, $C_{precharge}$) using TCAM cell layout and Cacti parameters for a particular CMOS technology. The value of $C_{metal-wire-per-cell}$ depends on the TCAM cell width and metal wire capacitance (C_{metal}). For $0.18\ um$ CMOS technology and a TCAM cell design of height $4.05\ um$ and width $4.33\ um$, we find the values of $C_{metal-wire-per-cell}$, $C_{drain-compareT}$, $C_{nandinv-gate}$, and $C_{precharge}$ to be $1.39fF$, $0.9fF$, $2.4fF$, and $8.9fF$ respectively. The last step is that we need to account for the fact that every cycle, *many* match lines are discharging. This is easy enough to handle as we can just scale Equation 3 by a factor equal to the number of rows. Note that in the case of worst case power estimation, it will prove quite accurate as on each cycle every line is toggling from '0' to '1' back to '0'.

While we make some simplifications in our model, in Section 5 we extract the height and width from the layout of several published TCAM cells and compare our power estimates with their measured results.

4.2 Select line power

While the largest amount of power comes from the match lines (as we will show in Section 5), the match lines are not the only large capacitances that need to be charged every cycle. With each access the select lines must be charged according to the search query (described in Section 3). As the TCAM scales, the size of the select lines grow, and the amount of energy to drive these lines can quickly add up. The select lines are different from the match lines in a couple of respects that prevent them from dominating. First, as can be seen in Figures 1 and 2, there is only one transistor per cell that interfaces with this line, and it is connected to the gate. We need to take care of the capacitive loading effect from the gate capacitance of NMOS compare transistor while calculating the equivalent capacitance for the search line.

For r number of rows, we can calculate the search line capacitance using the equation 4 as shown in Figure 4.1. For equation 4, we calculate the values of capacitances ($C_{metal-wire-per-cell}$, $C_{gate-compareT}$, C_{driver}) using TCAM cell layout and cacti parameters for a particular CMOS technology. The value of $C_{metal-wire-per-cell}$ depends on the TCAM cell height, and metal wire capacitance (C_{metal}). For $0.18\ um$ CMOS technology and a TCAM cell design of height $4.05\ um$ and width $4.33\ um$, we find the values of $C_{metal-wire-per-cell}$, $C_{gate-compareT}$, and C_{driver} to be $1.31fF$, $0.103fF$, and $18.31fF$ respectively. One important thing to notice is that *not every line switches every cycle*. To search for a one or a zero in the TCAM cells, only one of the select lines is driven high. To do a "don't care" bit, neither line is driven. The case where $Select_0$ and $Select_1$ are both driven high should never happen. This takes a factor of 2 off of worst case activity factor right away (from 100% down to 50%). While we might be tempted to consider the possibility that sometimes we may search for the

$$C_{matchline}\ (\ in\ \ fF\ ) = (C_{metal-wire-per-cell} + 2 * C_{drain-compareT})\ *\ l\ \ + C_{nandinv-gate} + C_{precharge} \tag{3}$$

where, $C_{metal-wire-per-cell}$ is the metal wire capacitance per TCAM cell
$C_{drain-compareT}$ is the drain capacitance of the NMOS compare transistor
$C_{nandinv-gate}$ is the gate capacitance of the nand gate and inverter
$C_{precharge}$ is the capacitance of the precharge circuit

$$C_{searchline}\ (\ in\ \ fF\ ) = (C_{metal-wire-per-cell} + C_{gate-compareT})\ *\ r\ \ + C_{driver} \tag{4}$$

where, $C_{metal-wire-per-cell}$ is the metal wire capacitance per TCAM cell
$C_{gate-compareT}$ is the gate capacitance of the NMOS compare transistor
C_{driver} is the capacitance of the driver circuit

Figure 3: Matchline and Searchline capacitance equations by taking into account the capacitance of metal wires and its loading effect.

same bit at the same position in two contiguous cycles, in the TCAM architecture described in Section 3, this cannot happen. We must bring the select lines back to zero before precharging the match lines to prevent a direct path from V_{dd} to ground (from the driver, through the matchline, through a matching select bit, and to ground). It is conceivable that any future design of TCAM, that does not require the select lines to return to zero, can take advantage of the fact that some bits are less likely to switch than others, but we have estimated from ip-traces that this would likely only reduce the activity factor from 50% to 46%.

4.3 Priority Encoder power

The priority encoder also consumes some power and is dependent on the number of rows. The dynamic power consumption in the priority encoder is independent of the number of bits. We take the result from [6, 28] to estimate the energy consumption of a N $\times$ 1 priority arbiter, where N is the number of rows. We use a 256-bit priority encoder from [6] and apply 2-level lookahead to design a higher-bit priority encoder. We calculate the number of the 256-bit priority encoders and the primitive gates required for a higher-bit priority encoder. The power consumption of the primitive gates is negligible compared to a 256-bit priority encoder. We calculate the total power consumption of a higher bit priority encoder by multiplying the number of the 256-bit priority encoders required and the energy consumption of one 256-bit priority encoder [6].

4.4 Delay Modeling

The maximum operating frequency for any TCAM structure is mainly dependent upon the time of accessing precharge circuit, precharging matchlines, driving searchlines and priority encoding. The access time for any circuit is decided by the time constant of the circuit which is a product of equivalent resistance (R_{eq}) and equivalent capacitance (C_{eq}). We already calculated the values of equivalent capacitance of these components in earlier sections and now we calculate the equivalent resistance of each of these components. While we use Horowitz's approximation approach (described in [29]) to calculate the access time of precharge circuit, matchline and searchline, we use result from [6] to find the access time of priority encoder. Usually, TCAM search operation is pipelined with priority encoder in a different pipeline stage. Hence, the *critical-path* TCAM delay for larger TCAMs is mostly dictated by the access time of matchline and searchline.

While this delay model can provide us with the maximum operating frequency of the TCAM structure, the TCAM circuit can be run at a lower speed to minimize the power consumption. Hence, we can have different power consumption for different operating frequency. We only show the maximum frequency calculation to show how much frequency target can be met and we limit our discussion to energy consumption per access for the rest of our paper.

5 Evaluation

Building on the work done in Cacti [29], and the capacitive models developed in Section 4, we have created power estimator for TCAM that is parameterizable and simple to use for either relative or absolute comparisons. In this section we evaluate our model against several physical implementations, and describe the scaling of the various components in the TCAM. For convenience, our model has been coded into a simple tool which is made publicly available at `http://www.cs.ucsb.edu/~arch/`.

While, thus far, we have described the power modeling of TCAM at a theoretical level, our goal is to enable fair comparisons not only between different TCAM de-

signs, but even between TCAM, Memory, and even logic. To calculate such parameters as the capacitance of a wire per unit length in real physical units ($fF/\mu m$) we need to use the characteristics of existing process technology and VLSI implementations. We collected these parameters from existing tools, VLSI layouts, and published results. In particular we base our parameters on the following:

- *Cacti tool [29]* - Cacti provides a number of low-level circuit parameters for 0.80 μm CMOS technology. We scale the parameters appropriately for 0.18 μm CMOS technology. Some of these parameters include the capacitance of metal wires, gate capacitance per unit area etc. To verify that the scaled values are not significantly different than current technology we checked them against a published literature. For example, the wire capacitance (C_{metal}) from [8] is 0.25 $fF/\mu m$, which is within 10% of our estimate.

- *TCAM layout* - While Cacti is a useful start, and will ensure fair comparisons between Cacti results and our model, it is limited in it's usefulness because it assumes a 6-transistor static SRAM cell, not a much larger TCAM cell. We collect the remaining required values from a published static TCAM cell layout [1]. The parameters extracted from this design include TCAM cell width, cell height, and the width of the transistors used in the comparison operation.

As network algorithm and architecture designers prefer to concentrate on the high-level architectural parameters to optimize their design, our model takes all the high-level architectural parameters such as *number-of-rows*, *column-bits*, and *CMOS technology* as the input parameters and provides the worst-case dynamic energy consumption. While these parameters are most directly applicable to an architect, TCAM cell design is not a solved problem and it is still progressing. Because of this there can be variations in the height and width of a cell and we leave these as optional parameters (the default height and width are extracted from [1]). For example, a Dynamic TCAM [4, 12] increases the effective number of bits per area because it uses only 1-transistor dynamic cell to store a bit. When our model is given the height and width of dynamic TCAM cell, we can estimate the power accurately even though the bit storage is completely different. We now demonstrate that our power model matches closely with published implementations and describe the effect of architectural parameters on the dynamic power of TCAM in a search operation.

5.1 Validation of our model

To check our design we take four physical implementations, two from industry datasheets, and two from circuits conferences.

SibreCore Technologies, a producer of TCAMs, states in a white paper [26] that SibreCore's SCT2000 consumes less than 1.7 W/Mb without any active power management and at a frequency of 66 MHz. We used our model to estimate the power consumption of a 1Mb TCAM in 0.18 μm CMOS technology with 2.5 V supply voltage which we believe is close to their design technology. Our model predicted a power of 1.85 W/Mb, a percent difference of less than 8% from the published results. Given that we do not have precise technology or layout parameters, we believe this to be a close fit.

Analog Bits, a TCAM vendor, markets a 512×144 TCAM, which runs at 800 MHz and consumes 0.5 mA/MHz current [2]. This TCAM is available for TSMC CL013LV/LVOD process. We use the same process features to find the power consumption for this TCAM configuration using our model and we find that it consumes $0.53\ mA/MHz$ current, which is with in 6% of the published results. Using our delay model, we find that using four subbanks this TCAM configuration can run at maximum of about 840 MHz.

We also validate our model with two results from circuit conferences. Noda et.al. [4] finds that the power dissipation for a conventional 4.5 Mb static TCAM without any power management features is approximately 7.0 W assuming a supply voltage of 1.5V and operating frequency of 143 MHz. We feed these high-level architectural parameters to our model and we find that the power dissipation is 6.4 W, which is again close. To verify the feasibility of operating frequency, we find the access time for this TCAM configuration, which gives a maximum operating frequency of 188 MHz using 16 subbanks.

Noda. et.al. [4] also provides the power dissipation of a dynamic TCAM (2 W) and TCAM cell features. We use these dynamic TCAM cell features to measure the power dissipation of the dynamic TCAM and it is found to be 2.712 W.

We also compare the relative contributions of each hardware component ($matchline$, $searchline$, etc) from [4] and we find that the results are accurate with in 10%.

5.2 Energy Breakdown

Figure 4 shows the breakdown of the energy consumed for a single access of the TCAM for a variety of different configurations. All of the configurations are in $0.18\mu m$ at 1.8V. The first configuration has 32k entries, each of length 36, and requires a total of 16.7 nJ for search access. The majority of the power, as expected, is being

consumed by the match line, but there is still a noticeable impact from both the search line and priority encoder. As a point of comparison, an SRAM of similar size would require approximately 1.9 nJ for a simple read operation. While an SRAM read will happen much faster, if more than 8.7 memory accesses are required on average to perform the same job, a TCAM could actually be a lower power system.

The second configuration shows the effect of doubling the size of an entry (from 36 to 72) and the third configuration shows the effect of doubling the *number* of entries. Surprisingly, adding more entries into a single bank of TCAM costs more per bit in terms of power than extending the size of an entry. We can see that between these two configurations, the match line power is roughly equivalent, but because of the longer search lines the search line power grows significantly. Doubling the number of entries also requires more energy in the priority encoder to select among twice as many possibilities.

The final configuration is the Dynamic TCAM which is a 64k x 36 configuration and is comparable to the third configuration. The reduced size of the device results in shorter match and select lines, although the sum of the capacitive loading from the transistors is unchanged due to the fact that an equal number of comparisons still need to be made. Instead the transistors are just packed onto shorter lines. The impact of the reduced line size decreases both the select and match line power but leaves the priority encoder unchanged.

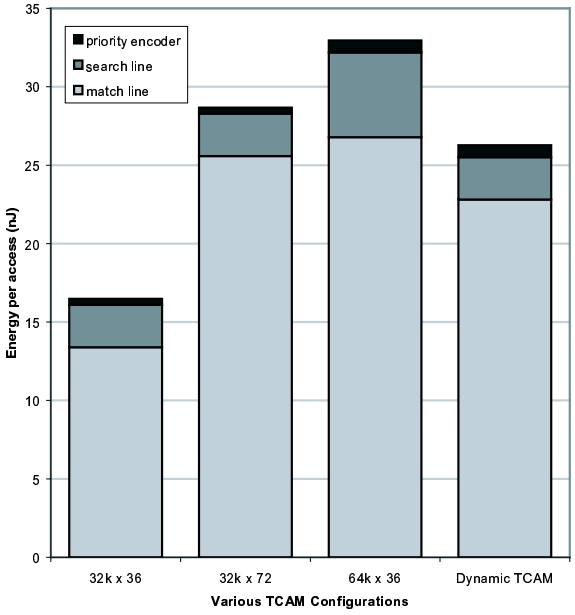

Figure 4: Breakdown of the energy consumption per access from the priority encoder, search line, and match line for a variety of TCAM configurations. The Dynamic TCAM is 64k x 36, with 1-T cells and its cell layout is used in power estimation.

5.3 Effect of Parameters

The results from Section 5.2 introduced an interesting phenomenon. For a fixed size TCAM (in terms of number of cells) a larger entry size (column bits) can sometime lead to *less* energy per access than a tall and narrow TCAM bank. To explore this idea further, we present Figure 5 which shows the variation in power per access for varying number of columns.

Two different sizes are shown, both for 0.18 μm technology with a fixed total size. On the x-axis, we show the effect of scaling the size of the entries, but because the total size in bits is fixed, the number of entries will be reduced as the entry size increases. While the energy per access is quite flat across a wide range on entry sizes, it climbs sharply below 32. Many vendors choose to ship a device that has a long maximum entry size (for example 72 or 144 entries), that is then configurable to smaller entries as needed.

The two configurations discussed assume that there is no sub-banking and no dynamic power management to reduce the energy per access on a large TCAM. If instead the TCAM was broken into banks, each with 4k entries, this could potentially reduce the power significantly. The problem is knowing *exactly* which bank your data resides in *without* searching through them. The dashed line in Figure 5 shows exactly this. There is a significant reduction in power, but the energy per access scales linearly with the length of an entry because there will be less and less possible banks across which we can divide the data. These banked energy numbers are not meant to be directly achievable, but rather serve as a guide to algorithm designers seeking to trade of smart ways of controlling TCAM banks with the underlying dynamics of the hardware.

The final results that we present are in Figure 6 which shows the effect of technology and voltage scaling on the total power for TCAM search. As the feature size drops, and the voltages are scaled, the power for searching through a TCAM has dropped significantly. A 100k entry TCAM in 0.4μm technology requires more than a factor of 10 times more joules/access than a comparable design in 90nm.

5.4 Potential Extensions and Uses

Our model can also be easily extended for various power optimization techniques with some extra work. For example, low voltage swing using current sensing scheme for matchline reduces the TCAM power consumption [1]. We can multiply our already calculated equivalent capac-

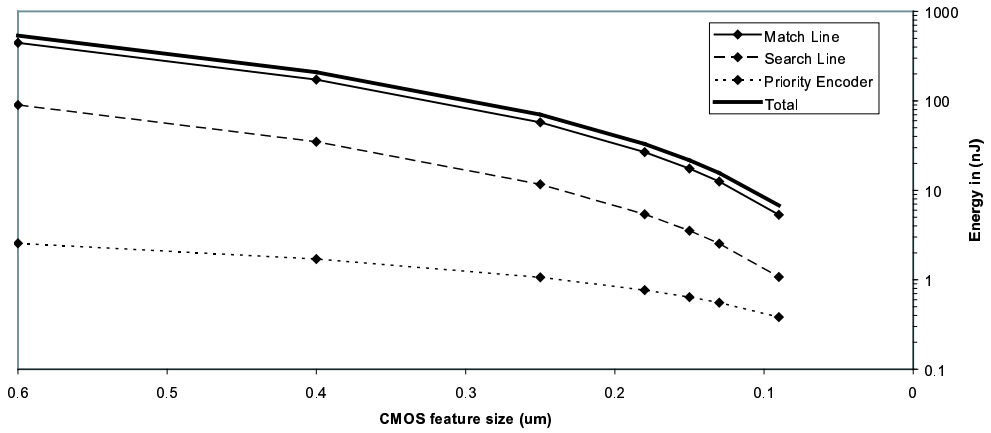

Figure 6: The scaling of energy consumption per access for a fixed sized TCAM as feature size and voltage are scaled back with time. The y-axis is in log scale.

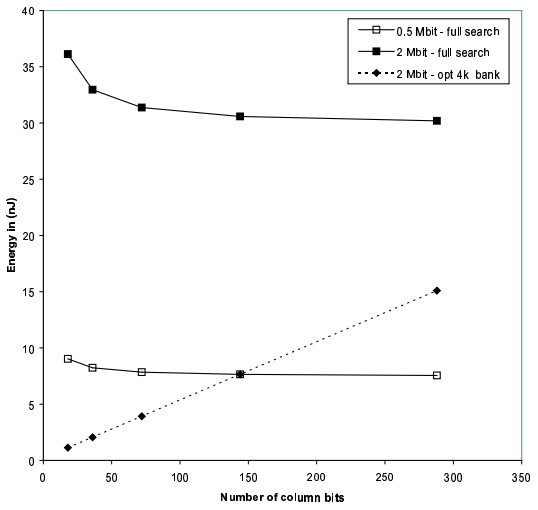

Figure 5: The scaling of energy per access with the length of the TCAM row (entry). For a monolithic design, the energy per access actually drops as the length is increased slightly, but this will come that expense of long latencies. An optimally banked design scales almost linearly with the size of an entry.

itance with $(\delta V)^2$, where δV is the voltage swing. We also need to find the equivalent capacitance of sense amplifier circuit to find the total power consumption. Similarly, with little effort, our model can also be extended for other power optimization techniques such as *low power dual matchline* [16].

Our model can quantify the power savings in a TCAM for new network algorithms and TCAM power management schemes because it takes the high-level architectural parameters as input. Hence, researchers can explore new algorithms and schemes to prune the search space of TCAM, either by searching less number of rows, searching less number of bits, or shutting off some banks to reduce the TCAM power consumption.

We also modify Cacti [23] to get the power consumption of SRAM data array instead of total cache power. We find that for 32k entries with 36 column bits, the energy to perform a search in TCAM is only a factor of 8 times more expensive than a single access to SRAM. We believe that enabling direct comparisons between SRAM and TCAM, these models will open the door for researchers to explore and combine them in novel and interesting ways.

6 Conclusions

In high-speed networking applications, TCAM has been used as one of the principal components due to its ability to perform fully associative ternary search. This ability can be exploited to perform an wide range of operations, and new applications are still being discovered and implemented. To provide a fair comparison against past techniques when power is concerned, there is a need for an accurate TCAM power model that can be directly compared against comparable SRAM, cache, and logic models.

In this paper we have shown that such a model can be built through the calculation of match and select line capacitances by considering both the wire length and loading of these lines. Our model can factor in changes in voltage, operating frequency, number of entries, length of entries, and even circuit level parameters such as cell height and width. We describe how TCAMs scale with

these parameters, and validate our model against several physical designs. Our model can also be easily extended to find dynamic power consumption based on some specific networking traces.

We show that the energy to search a TCAM is not significantly more than the energy consumed by several SRAM accesses to a memory of comparable size. This sort of comparison will provide a foundation on which to do hybrid SRAM/TCAM algorithms research. We believe our model will enable researchers to find and exploit tradeoffs at the algorithm and architecture level, and will enable realistic energy estimations to be made across a wide range of TCAM based applications and designs.

Acknowledgments

We would like to thank the anonymous reviewers for providing useful comments. This work was funded in part by NSF Career Grant CCF-0448654.

References

[1] I. Arsovski, T. Chandler, and 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*, 38:155–158, January 2003.

[2] Analog Bits. High Speed Ternary CAM Datasheet, 2004.

[3] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*, pages 83–94, New York, NY, USA, 2000.

[4] H. Noda et.al. A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture. *IEEE Journal of Solid-State Circuits*, 40(1):245–253, January 2005.

[5] Mark A. Franklin and Tilman Wolf. Power Considerations in Network Processor Design. In *Proc. of Network Processor Workshop in conjunction with Ninth International Symposium on High Performance Computer Architecture (HPCA-9)*, pages 10–22, Anaheim, CA, February 2003.

[6] W. W. Fung and M. Sachdev. High Performance Priority Encoder for Content Addressable Memories. In *Micronet R&D Annual Workshop*, 2004.

[7] B. Gamache, Z. Pfeffer, and S. P Khatri. A fast ternary CAM design for IP networking applications. In *12th International Conference on Computer Communications and Networks (IC3N-03)*, Dallas, TX, October 2003.

[8] M. Horowitz, R. Ho, and K. Mai. The future of wires, 1999.

[9] I.Y.L. Hsiao, D.H. Wang, and C.W. Jen. Power modeling and low-power design of content addressable memories. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 4, pages 926–929, 2001.

[10] M. Kounavis, A. Kumar, H.M. Vin, R. Yavatkar, and A. Campbell. Directions in Packet Classification for Network Processors. In *Proc. of Network Processor Workshop in conjunction with Ninth International Symposium on High Performance Computer Architecture (HPCA-9)*, pages 10–22, Anaheim, CA, February 2003.

[11] Xudong Li, Zhen Liu, Wei Li, and Bin Liu. SCP-TCAM: A Power-Efficient Search Engine for fast IP Lookup. In *ISBN Proceedings*, 2004.

[12] V. Lines, A. Ahmed, P. Ma, S. Ma, R. McKenzie, H. Kim, and C. Mar. 66MHz 2.3M Ternary Dynamic Content Addressable Memory. In *8th IEEE International Workshop on Memory Technology, Design, and Testing (MTDT 2000)*, San Jose, CA, 2000.

[13] Huan Liu. Efficient Mapping of Range Classifier into Ternary-CAM. In *10th Symposium on High Performance Interconnects HOT Interconnects (HotI'02)*, Stanford, CA, August 2002.

[14] Y. Luo, J. Yang, L. Bhuyan, and L. Zhao. NePSim: A Network Processor Simulator with Power Evaluation Framework. *IEEE Micro Special Issue on Network Processors for Future High-End Systems and Applications*, Sep/Oct 2004.

[15] M. Mamidipaka and Nikil Dutt. eCACTI: An Enhanced Power Model for On-chip Caches. Technical Report CECS TR-04-28, September 2004.

[16] N. Mohan and M. Sachdev. Low power dual matchline ternary content addressable memory. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 633–636, May 2004.

[17] G. J. Narlikar, A. Basu, and F. Zane. CoolCAMs: Power-Efficient TCAMs for Forwarding Engines. In *IEEE INFOCOM: The Conference on Computer Communications*, 2003.

[18] K. Pagiamtziz and A. Sheikholeslami. A Low-Power Content-Addressable Memory (CAM) Using Pipelined Heirarchical Search Engine. *IEEE Journal of Solid-State Circuits*, 2004.

[19] R. Panigrahy and S. Sharma. Sorting and Searching using Ternary CAMs. *IEEE Micro*, 23(1):44–53, 2003.

[20] V. C. Ravikumar, R.N. Mahapatra, and L. Bhuyan. EaseCAM: An Energy and Storage Efficient TCAM-Based Router Architecture for IP Lookup. *IEEE Trans. Comput.*, 54(5):521–533, 2005.

[21] Cypress Semiconductors. Ayama(tm) 10000a network search engine, 2004.

[22] S. Sharma and R. Panigrahy. Reducing TCAM Power Consumption and Increasing Throughput. In *10th Symposium on High Performance Interconnects HOT Interconnects (HotI'02)*, Stanford, CA, August 2002.

[23] P. Shivakumar and N. P. Jouppi. Cacti 3.0: An Integrated Cache Timing, Power and Area Model. Technical Report Western Research Lab (WRL) Research Report, 2001/2.

[24] E. Spitznagel, D. Taylor, and J. Turner. Packet Classification Using Extended TCAMs. In *11th IEEE International conference on network protocols (ICNP)*, 2003.

[25] Cisco Systems. 600-Watt Redundant AC Power System, 1998.

[26] SibreCore Technologies. Classification and Forwarding Coprocessors come of age - The myths and Realities of High Performance Ternary CAMs (TCAMs) , March 2002.

[27] H.S. Wang, X.P. Zhu, L.S. Peh, and S. Malik. Orion: A power-performance simulator for interconnection networks. In *In Proc. International Symposium on Microarchitecture*, pages 294–305. IEEE Press, Nov 2002.

[28] J. Wang and C. Huang. High-Speed and Low-Power CMOS Priority Encoders. *IEEE Journal of Solid-State Circuits*, 35(10):1511–1514, October 2000.

[29] S. Wilton and N. Jouppi. An enhanced access and cycle time model for on-chip caches. Technical Report 93/5, DEC Western Research Lab, 1994.

[30] F. Yu, R. H. Katz, and T. V. Lakshman. Gigabit Rate Packet Pattern-Matching Using TCAM. In *11th IEEE International conference on network protocols (ICNP)*, Berlin, Germany, 2004.