RESEARCH ARTICLE OPEN ACCESS

An Efficient Reconfigurable Content Addressable Memory

Saswathy Sekharan
(Department of Electronics and Communication, Nehru Institute of Engineering and Technology, Coimbatore-641105,)

Abstract
This paper introduces an efficient reconfigurable Content Addressable memory (CAMs) which is a hardware search engine that are much faster than other algorithmic approaches for search intensive applications. Content Addressable Memories are composed of conventional semiconductor memory (usually SRAM) with added comparison circuitry that enables a search operation to complete in a single clock cycle. To understand more about Content Addressable Memory, it helps to contrast it with RAM. A RAM is an integrated circuit that stores data temporarily. In CAM, the user supplies the data and gets back the address.In this paper we introduce a temporary memory called Cache. The cache-CAM (C-CAM) saves 80% power over a conventional CAM. Compared with existing software search engines proposed hardware search engine can do multiple searches at a time with more flexibility.

Index Terms—Content Addressable Memory (CAM), VLSI, Introduction to VHDL, Hardware, RTL modelling.

I. INTRODUCTION

High search intensive applications with added comparison circuitry that completes the search operation in a single clock cycle. Most memory devices store and retrieve data by addressing specific memory locations [1]. As a result, this path often becomes the limiting factor for systems that rely on fast memory accesses. The time required to find an item stored in memory can be reduced considerably if the item can be identified for access by its content rather than by its address[9]. A memory that is accessed in this way is called content-addressable memory or CAM[5]. CAM provides a performance advantage over other memory search algorithms, such as binary or tree-based searches or look-aside tag buffers, by comparing the desired information against the entire list of pre-stored entries simultaneously, often resulting in an order-of-magnitude reduction in the search time. CAMs are hardware search engines that are much faster than algorithmic approaches for search-intensive applications. CAMs are composed of conventional semiconductor memory (usually SRAM) with added comparison circuitry that enable a search operation to complete in a single clock cycle[2]**.**

II. PROPOSED SYSTEM

Content Addressable Memory (CAM) is a special type of computer memory used in certain very high speed searching applications. CAM is designed such that the user supplies a data word and the CAM searches its entire memory to see if that data word is stored anywhere in it. The CAM is composed of a conventional semiconductor memory generally SRAM (Static RAM) with an added comparison oriented circuitry that enables a search operation to complete in a single clock cycle.

CAM is used in applications where search time is very critical and must be very short.
Some of the typical applications of the CAM includes the applications in network devices like routers and switches packet forwarding and packet classification in the internet routers etc. The power dissipation and cost are the drawback of Content Addressable Memory. So we introduce a additional memory called Cache in order to prevent these drawbacks up to an extent.

1. Block Diagram

1.1. CAM

Figure 1 CAM is a hardware search engine that is much faster than algorithmic approaches for search intensive applications. Since CAM is an outgrowth of Random Access Memory (RAM) technology, in order to understand CAM, it helps to contrast it with RAM. A RAM is an integrated circuit that stores data temporarily.

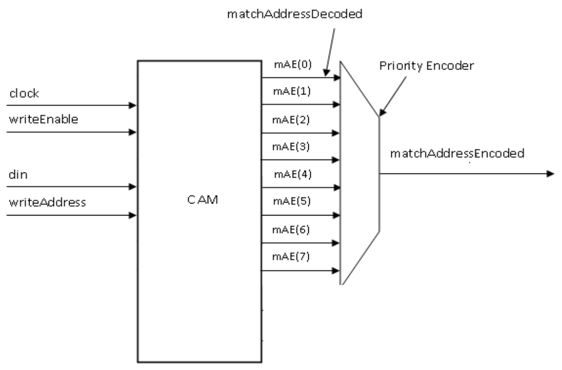


Fig.1:.Simple CAM

Data is stored in a RAM at a particular location, called an address. In a RAM, the user supplies the address, and gets back the data. The number of address line limits the depth of a memory using RAM, but the width of the memory can be extended as far as desired. With CAM, the user supplies the data and gets back the address. The CAM searches through the memory in one clock cycle and returns the address where the data is found. The CAM can be preloaded at device startup and also be rewritten during device operation. Because the CAM does not need address lines to find data, the depth of a memory system using CAM can be extended as far as desired, but the width is limited by the physical size of the memory.

Fig2: Content Addessable Memory

1.2.POWER USAGE IN CAM

The search operation in the CAM is explained in the section operating modes. The search operation done in this way will consume more power. During a mismatch the pre-charged match line will get grounded (i.e. it will become ‘0’). So there will be a current flow from vcc to ground for each mismatch which will add to the power consumption. And since the mismatch to match ratio is so high the power consumption is extremely high.

Different methods are adopted to reduce power consumption like using block cam, cache cam etc…

Block cam

Here the entire CAM is divided into several block CAM’s. And in address’s a few more bits are added as a header for all address to identify each block. While searching, the controller selects the blocks to be searched based on some criteria.

Cache cam

Here the most frequently used contents are kept in a small cache, along with the address. Since the size of cache is small the power consumption due to mismatch will be low

1.3. BLOCK DIAGRAM OF CACHE CAM

By using a small cache along with the CAM, we avoid accessing the larger and higher power CAM. For a cache hit rate of 90%, the cache-CAM (C-CAM) saves 80% power over a conventional CAM, for a cost of 15% increase in silicon area. Even at a low hit rate of 50%, a power savings of 40% is achieved.

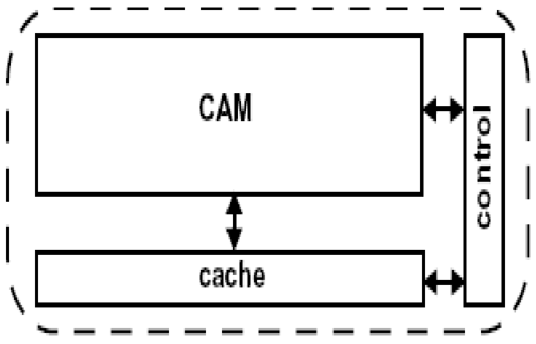


Fig 3: Simplified block diagram of a cache-cam

Fig 3 represents a simplified view of the C-CAM, where the cache holds frequently accessed CAM data. This caching concept is similar to that of caching in processor systems. Unlike processor cache systems where the main purpose of the cache is to increase performance, the purpose of caching CAM is to reduce power consumption. In C-CAM, every search operation that results in a match from the smaller cache saves power by avoiding a search in the larger and higher-power CAM.

The power savings in a C-CAM implementation depends heavily on the trade-off between the cache size and the cache hit rate. To illustrate this trade-off, Fig. 4 plots the power consumption of a 32k C-CAM system versus the cache size. The plot shows the portion of power consumed by the cache, the portion consumed by the CAM, and the total power. The reported power is normalized to the power dissipation of a conventional 32k CAM. As expected, the cache power consumption increases linearly with the cache size. The CAM power, on the other hand, decreases with the cache size, due to the increased cache hit rate and thus less frequent access to the CAM

The total C-CAM power is at a minimum at about the 4k cache size, and increases for larger cache sizes as the cache hit rate levels off. Replacement policy is one of the factors that determine the cache hit rate. Thus a high-performance policy would reduce the CAM power in Fig. 5 by avoiding misses that search the full CAM. The cost of this reduction in CAM power is the power consumption of the circuitry implementing the cache replacement policy. Therefore we desire a cache-replacement that generates a high cache hit rate while consuming little power.

Cache -CAM (C-CAM) that can potentially reduce power consumption between 40-80% depending on the cache hit rate. cache holds frequently accessed CAM data. This caching concept is similar to that of caching in processor systems. Unlike processor cache systems where the main purpose of the cache is to increase performance , the purpose of caching CAM is to reduce power consumption. In C-CAM, every search operation that results in a match from the smaller cache saves power by avoiding a search in the larger and higher-power CAM.By using a small cache along with the CAM, we avoid accessing the larger and higher power CAM. For a cache hit rate of 90%, the cache-CAM (C-CAM) saves 80% power over a conventional CAM, for a cost of 15% increase in silicon area. Even at a low hit rate of 50%, a power savings of 40% is achieved.

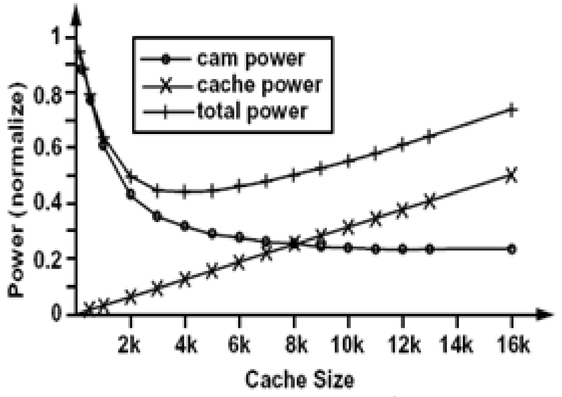


Fig. 4: Power vs. Cache Size for a cache

1.4. CACHE REPLACEMENT POLICIES

The cache replacement policy decides which cache entry should be removed when a new entry is inserted into a full cache. The replacement policy and characteristics of the input data stream together determine the resulting cache hit rate for a given cache size. A low-complexity replacement policy may consume little power; however, it results in a low cache hit rate leading to high overall system power dissipation. On the other hand, a high-complexity replacement policy may consume so much power that it overwhelms the savings resulting from the high cache hit rate.

To explore this spectrum of replacement policies, consider three cache replacement policies: sequential replacement, random replacement, and least-recently used (LRU) replacement. The sequential replacement policy simply increments an address counter sequentially to point to the location of the new entry. This can be implemented using a shift register. This implementation consumes little power but results in a low hit rate. The random replacement policy chooses a random cache location for any new cache entry. The implementation of the random replacement policy consists of four pseudo-random number (PN) sequence generators and a decoder. This implementation consumes slightly more power than the sequential policy but generally results in an increased hit rate. Finally, the LRU policy tracks the relative time when each cache location receives a hit and replaces the location that is least recently used. This policy is implemented in the elementary form and has the highest complexity and power consumption compared to the other two replacement policies, but offers a moderately higher cache hit rate than the random policy.

After the cache is initially filled, the cache replacement policies are active only when there is a write to the cache. Since a cache replacement occurs on every cache miss, the portion of cycles where a cache write occurs is equal to the cache miss rate. The LRU Policy actually consumes power even during cache search cycles (i.e. including cycles that do not have a cache write) since LRU counters are updated even on cache searches. Thus, for the LRU policy, there is an extra power consumption that occurs on every search in addition to the power consumption per write. However, this power consumption is small compared to the other sources of overhead that are present on every cycle

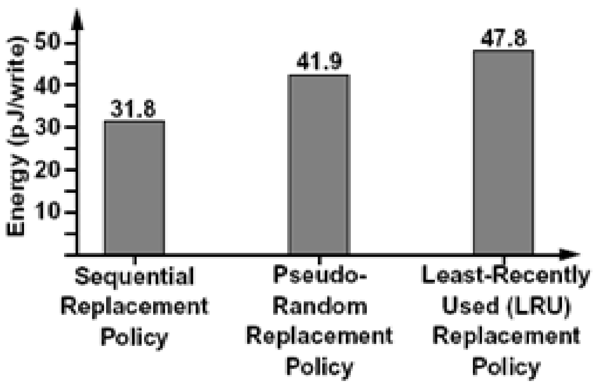


Fig.5: Simulated energy consumption for the three cache replacement policies.

III. BLOCK DIAGRAM OF MAIN CAM

The working of Cache cam is similar to the working of CAM. As in CAM the Cache supports two modes of operation

- *Read*
- *Write*

READ:

This mode is selected for searching the data in the cam. Here CAM is associated with the cache. In read mode the con_rw will be '1' at the same time con_enable and clock will be high. Cam controller is responsible for providing the necessary signals.

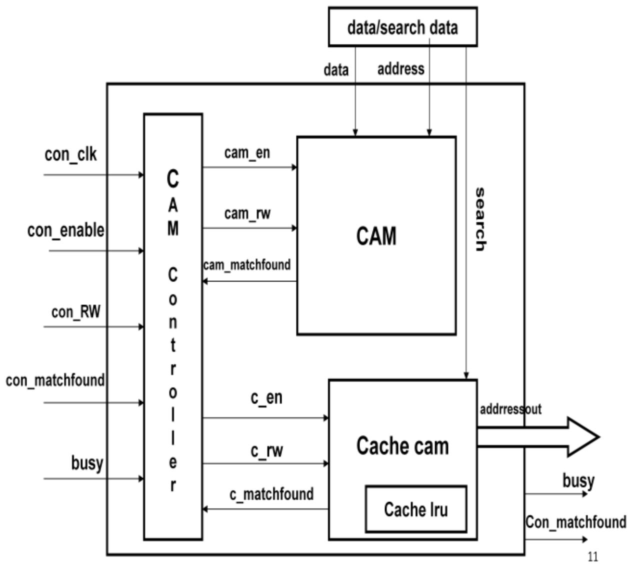


Fig.6:1 Block diagram of main CAM

Depending upon these signals the cam and cache will work. When the con_rw is high the C_rw and C_en become high and it searches in the cache. Output will be addressout. At this time cam will be disable. While searching if there is no data in Cache, it will directly search in the CAM and address will be out and matchfound signal will be high. Busy signal show that it is in the search mode.Cache is used as with cache lru which helps to update the recently used data and deletes the unused data. By using the cache it will reduce the power. The size of the cache will be half of CAM size.

WRITE:

This mode is selected for writing the data in the memory. In write mode cache enable will be zero and cam enable will be '1'. In this mode cam controller will give the necessary signals and it will write directly in the CAM. On comparing with the simple cam, cache cam needs three clock cycle for searching and giving out the address stage pipeline operation of the cache CAM.

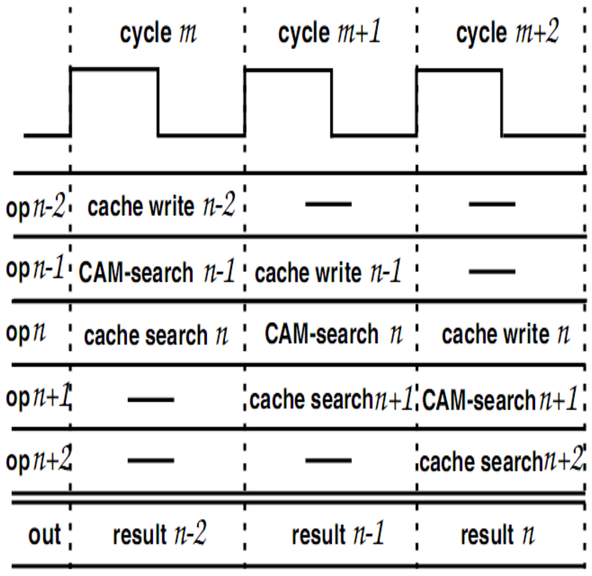


Fig. 6.2 Timing diagram illustrating the three-

IV. HARDWARE IMPLEMENTATION

The Spartan-3E Starter Kit board highlights the unique features of the Spartan-3E FPGA family and provides a convenient development board for embedded processing applications. The board highlights features such as:

Spartan-3E FPGA specific features

a. Parallel NOR Flash configuration
b. MultiBoot FPGA configuration from Parallel NOR Flash PROM
c. SPI serial Flash configuration
d. Embedded development
e. MicroBlaze™ 32-bit embedded RISC processor
f. PicoBlaze™ 8-bit embedded controller
g. DDR memory interfaces

Fig.7: Spartan-3E Starter Kit board

50MHZ ON-BOARD OSCILLATORS

The board includes a 50 MHz oscillator with a 40% to 60% output duty cycle. The oscillator is accurate to ±2500 Hz or ±50 ppm.Clocks can be supplied off-board via an SMA-style connector. Alternatively, the FPGA can generate clock signals or other high-speed signals on the SMA-style connector and optionally install a separate 8-pin DIP-style clock oscillator in the supplied socket.

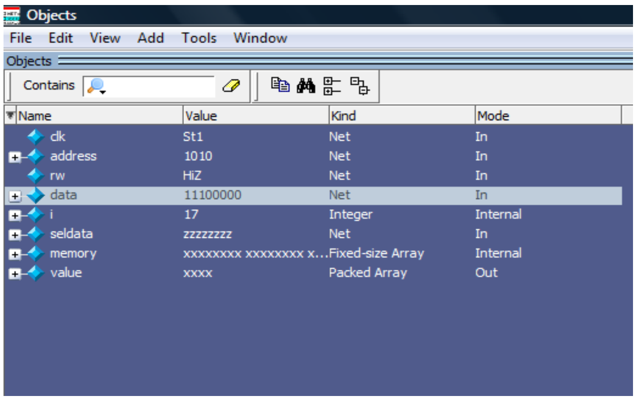
V. EXPERIMENTAL SIMULATION RESULTS

CAM is used to get the address location where a specific data is saved. It is high speed searching engine. We can obtain the output within a clock pulse.

Memory locations with specific address locations are being created. All the process are carrried out at the positive edge of clock pulse.This isdone using the‘always’ operator. When rw=1, the writing process are taken place.To these specific address locations datas are being saved using read-write symbol.

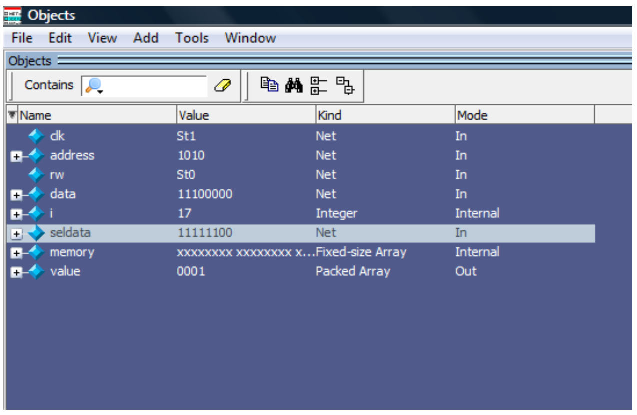
After the writing process is over,we go for reading/searching.Reset the read write symbol to zero,ie rw=0 .The data to be searched is given as seldata. For loop is used for this searching process.The loop is initiated from first address location to the final address location using variable i.if the data in the ‘i’th address location matches with the seldata,the value of i is being displayed as output.

A.Simulation of CAM



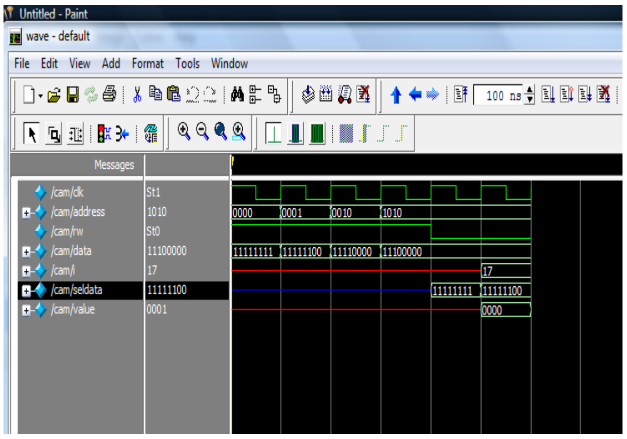
Name	Value	Kind	Mode
clk	St1	Net	In
address	1010	Net	In
rw	HiZ	Net	In
data	11100000	Net	In
i	17	Integer	Internal
seldata	zzzzzzzz	Net	In
memory	xxxxxxxx xxxxxxxx x...	Fixed-size Array	Internal
value	xxxx	Packed Array	Out

Fig 8.1. Simultion –input to cam



Name	Value	Kind	Mode
clk	St1	Net	In
address	1010	Net	In
rw	St0	Net	In
data	11100000	Net	In
i	17	Integer	Internal
seldata	11111100	Net	In
memory	xxxxxxxx xxxxxxxx x...	Fixed-size Array	Internal
value	0001	Packed Array	Out

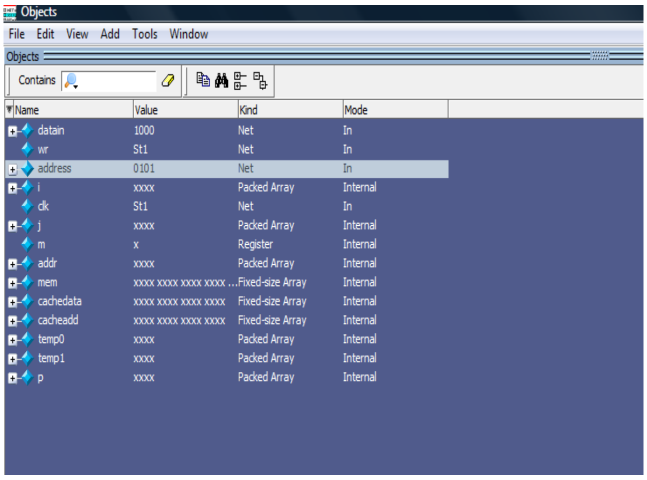
Fig 8.2. Simulation output of cam



Messages	
/cam/clk	St1
/cam/address	1010
/cam/rw	St0
/cam/data	11100000
/cam/i	17
/cam/seldata	11111100
/cam/value	0001

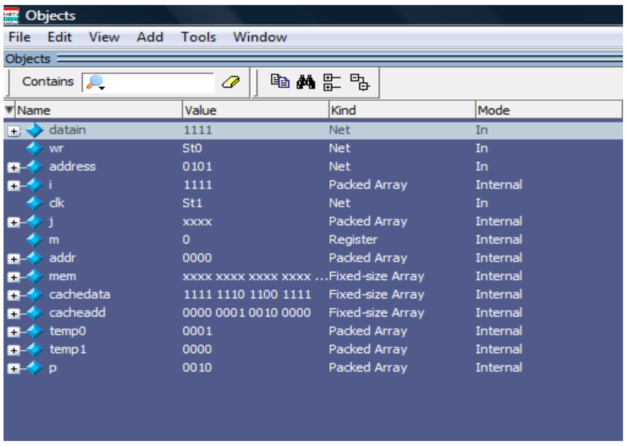
Fig 8.3 Simulation waveform of cam

B.Simulation of CACHE CAM



Name	Value	Kind	Mode
datain	1000	Net	In
wr	St1	Net	In
address	0101	Net	In
i	xxxx	Packed Array	Internal
clk	St1	Net	In
j	xxxx	Packed Array	Internal
m	x	Register	Internal
addr	xxxx	Packed Array	Internal
mem	xxxx xxxx xxxx xxxx ...	Fixed-size Array	Internal
cachedata	xxxx xxxx xxxx xxxx	Fixed-size Array	Internal
cacheadd	xxxx xxxx xxxx xxxx	Fixed-size Array	Internal
temp0	xxxx	Packed Array	Internal
temp1	xxxx	Packed Array	Internal
p	xxxx	Packed Array	Internal

Fig.9.1. Simulation input of CACHE CAM



Name	Value	Kind	Mode
datain	1111	Net	In
wr	St0	Net	In
address	0101	Net	In
i	1111	Packed Array	Internal
clk	St1	Net	In
j	xxxx	Packed Array	Internal
m	0	Register	Internal
addr	0000	Packed Array	Internal
mem	xxxx xxxx xxxx xxxx ...	Fixed-size Array	Internal
cachedata	1111 1110 1100 1111	Fixed-size Array	Internal
cacheadd	0000 0001 0010 0000	Fixed-size Array	Internal
temp0	0001	Packed Array	Internal
temp1	0000	Packed Array	Internal
p	0010	Packed Array	Internal

Fig.9.12.Simulation output of CACHE CAM

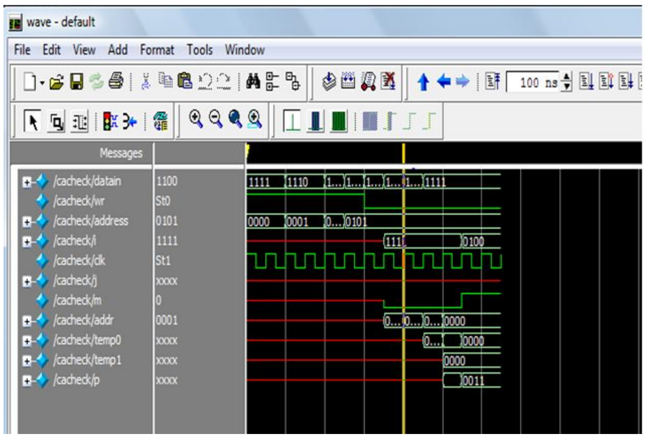


Fig.9.3. Simulation waveformt of CACHE CAM

VI. ADVANTAGES

(i) They associate the input (compared) with their memory contents in one clock cycle.
(ii) They are configurable in multiple formats of width and depth of search data that allows searches to be conducted in parallel.
(iii) CAM can be cascaded to increase the size of lookup tables that they can store.
(iv) We can add new entries into their table to learn what they don't know before.
(v) They are one of the appropriate solutions for higher speeds

VII. APPLICATIONS

CAM can be used to accelerate any applications ranging from local-are networks, database management, file-storage management, pattern recognition, artificial intelligence, fully associative and processor-specific cache memories, and disk cache memories. Although CAM has many applications, it is particularly well suited to perform any kind of search operations. Following is a discussion of some of these applications.

VIII. CONCLUSION

For any application that requires a fast memory search, CAM can provide a solution. There are many manufacturers that provide discrete CAM components, such as Music Semiconductor, Kawasaki LSI, Netlogic Microsystems Inc and Lara Technology. As CAM becomes more popular and system-on-a-chip design becomes more common, intellectual property and embedded versions of CAM such as that offered in the most advanced

PLDs have also become available. In the coming months, CAM will enjoy increased popularity, particularly with the growing number of communications applications that can utilize its capabilities.

This paper proposes C-CAM which uses caching to save power in CAM. Through simulation and early test chip measurements, we show that a cache added to a 32k CAM can save as much as 80% of power consumption for hit rates of 90%. Even for hit rates as low as 50% the power savings are still 40%.

REFERENCES

[1] Kostas Pagiamtzis and Ali Sheikholeslami Department of Electrical and Computer Engineering University of Toronto, Canada W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
[2] Anargyros Krikelis, Charles C. Weems (editors) (1997) Associative Processing and Processors,IEEE Computer Science Press.ISBN 0-8186-7661-2
[3] Hannum et al.. (2004). System and method for resetting and initializing a fully associative array to a known state at power on or through machine specific state. U.S. Patent Rob92] Ian N. Robinson. Pattern-addressable memory. *IEEE Micro*, *12(3):20 30*, June 1992
[4] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.
[5] [Sch96] Kenneth James Schultz. *CAM-Based Circuits for ATM Switching Networks*. PhD thesis, University of Toronto, 1996.C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
[6] Kostas Pagiamtzis, Student Member, IEEE, and Ali Sheikholeslami, Senior Member, IEEE
[7] [7] .[CD89] Lawrence Chisvin and R. James Duckworth. Content-addressable and associative memory: Alternatives to the ubiquitous RAM. *IEEE Computer*, 22(7):51 64, July 1989.
[8] [RSBD01] Miguel Ruiz-Sanchez, Ernst W. Biersack, and Walid Dabbous. A survey and taxonomy of IP lookup algorithms. *IEEE Network*, 15(2):8 23, March-April 2001.
[9] [PZ92] Tong-Bi Pei and Charles Zukowski. Putting routing tables in silicon. *IEEE Network Magazine*, 6(1):42 50, January 1992.
[10] Tetsu Nagamatsu et al., A 1.9ns BiCMOS CAM Macro with Double Match Line Architecture, IEEE 1991 Custom Integrated Circuits Conference, p. 14.3.1ff