

Fully-Parallel Multi-Megabit Integrated CAM/RAM Design

Kenneth J. Schultz and P. Glenn Gulak

Dept. of Electrical and Computer Engineering, University of Toronto
Toronto, Ontario, M5S 1A4, Canada

Abstract: Previous implementations of large-capacity Content Addressable Memories (CAMs) have employed advanced fabrication techniques or serialized operation. This paper describes a more generally applicable fully-parallel solution based on circuit and architectural innovation. A "pre-classified" CAM is integrated into the same array as its target RAM, and both use the same core cells. Architecture and operation are described, as are two critical-path circuits: the match-line pull-down and the multiple match resolver. An 8 kb test chip is described, and simulation results for a 1 Mb configuration are presented.

1. Introduction

In most memories, data is accessed based on its location. Access based on data content is often more natural. Memories operating on this access principle are known as Content Addressable Memories (CAMs). Despite a long history and many potential applications, CAMs have yet to gain a significant market share, due to the disparity between CAM and RAM capacity.

This disparity is caused by a number of factors. The added circuit complexity required to implement content addressability dictates that CAM density lags RAM density. Also, the relatively small CAM market provides chip manufacturers little return on investment for applying state-of-the-art fabrication technology. And without adequate density at competitive prices, the market will not grow. A catch-22 ensues, from which the only escape is to develop architectural and circuit techniques that lead to a significant density improvement at an affordable cost.

The majority of large-capacity CAMs rely on advanced fabrication techniques [1] or serialized processing [2]. We seek a more generally applicable fully-parallel (single-

cycle) solution based on circuit and architectural innovation.

This paper reviews a novel architecture, proposed in [3], that enables CAMs to achieve RAM-like density, as well as enabling the efficient integration of CAM and target RAM into a physically contiguous and homogeneous array. Following that, we describe the circuit design required to realize such a memory, as well as the initial 8 kb test chip employing these techniques. A 0.8 μm BiCMOS ASIC process [4] is employed for all simulations, as well as for test chip implementation. Performance estimates for a 1 Mb memory are also presented.

2. Architecture

A. Background

The main barrier to large-capacity CAM development has been the architectural difficulty in implementing column decoding in a fully-parallel CAM. Fully-parallel CAMs are those in which all stored words, in all bit positions, are checked against the reference word simultaneously, so that each SEARCH can be performed in a single clock cycle. In a fully-parallel CAM, a match line connects all cells in a single physical row. The reference word is broadcast on the bit lines, and if there is a mismatch in a cell, the cell will pull down its match line. If all cells in a row match their respective bit lines, the match line remains high, and a match has been found. If all cells in a row belong to the same word, one is checking a word stored in memory versus a reference word, as desired. However, under this constraint, one cannot achieve high density with practical dimensions. If there are multiple words making up a single physical row, either all words in a given row must match for the match line to remain high (this is not the operation we want), or only one word at a time, in each row, is enabled to pull down the match line (requiring multiple sequential operations to complete the search [2], hence the operation is no longer fully-parallel).

To achieve a true fully-parallel search, one may use an aggressive process technology and essentially implement a

This work has been financially supported by Bell-Northern Research, the Natural Sciences and Engineering Research Council of Canada, North American Life Assurance Co., and a Walter Sumner Memorial Fellowship.

Design and fabrication support provided by Northem Telecom is greatly appreciated. Thanks to E. Domer, B. Hagglund, S. Komachuk, E. Munter, S. Nichols, and I. Perryman for their helpful comments.

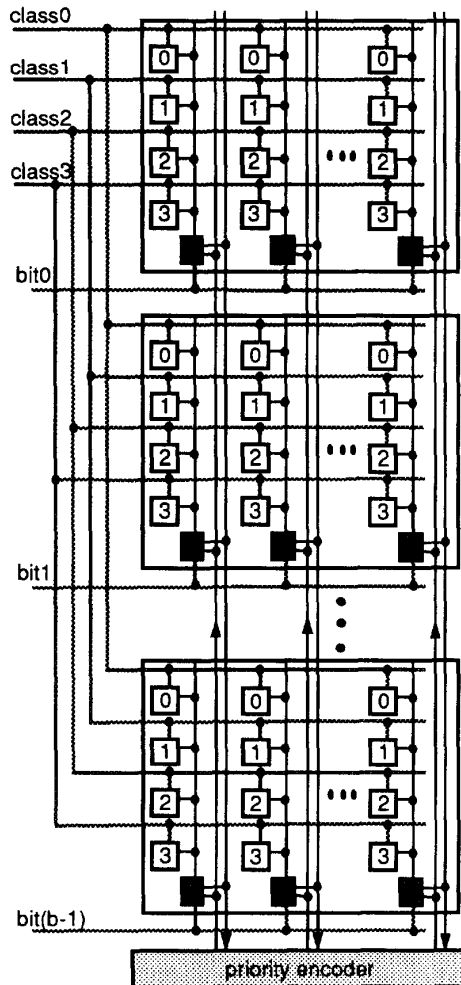


Figure 1: Pre-classified CAM block diagram.

large number of small narrow CAMs (with one word per row) on a single chip [1]. Although the column decode problem is addressed by the architecture and process technology, the density (for equivalent technologies) is actually worse than that of a standard full-parallel CAM, due to the area overhead of the required post-encoding circuits.

Alternatively, one may perform pre-classification prior to the fully-parallel operation to assign the data to one of C "classes" and thereby determine which of the C words sharing a comparator will have access to it. Motomura *et al.* [5] introduced this technique in a dictionary processor application.

B. Pre-classified CAM

Figure 1 shows a block diagram of a pre-classified CAM. The core of the CAM is composed of b separate RAM sub-arrays, one for each I/O bit. Each of these has C rows; based on the class read from a Pre-classification

RAM (PRAM), the same row number will be accessed simultaneously in each of the sub-arrays. The PRAM contents can be re-written during operation to provide a uniform distribution of data throughout the C classes, for improved memory efficiency.

During a SEARCH, the reference data is asserted onto the reference bit lines (running horizontally). Data from the selected class is read onto the sub-array bit lines (running vertically) and are compared to the reference data in the shaded boxes. Each of the y columns has access to an XOR gate (1-bit comparator) and match line pull-down circuit. The match lines (running vertically downward) are pulled low by any unmatched bit in their column. Match lines are processed in the priority encoder, where a single match is selected in the event of multiple matches.

During a READ, a single responder select line (running vertically upward) will be asserted, based on the previous search result. In each sub-array, the accessed class will be read onto the sub-array bit lines. In the column corresponding to the asserted responder select line, the sensed data will be connected, through the shaded box, to the horizontal reference bit lines, for reading out of the memory.

During a WRITE, a single row is selected in each sub-array by the pre-classification circuitry. In the column selected by the responder select line, write data will be asserted from the reference bit lines to the sub-array bit lines, and from there written into the selected cell.

C. Integrated CAM/IRAM

In most applications, the results of a CAM SEARCH are used to select a word in an external "target" RAM for subsequent READS and WRITES. Since RAM core cells are used for storage in a pre-classified CAM, target RAM cells can be integrated into the bit blocks together with the CAM cells. The resulting architecture is shown in Figure 2. RAM cells used to store CAM data are shaded, but are physically identical to the other cells. The priority encoder outputs are combined with r address bits to yield "decoded responder select lines" (DRSLs) which select among the RAM columns associated with the matched CAM column. Column access circuitry is represented by diamonds, and XOR gates by the shaded portion of the diamonds in CAM columns. We refer to the 2^r words associated with each CAM entry as a page. The physical dimensions of the memory are $y \times 2^r$ columns by $C \times b$ rows.

3. XOR design and SEARCH operation

While RAM-column data is sensed at the memory periphery only after DRSL selection during READS, each CAM column has a dedicated differential latch sense amplifier [6] that is activated during SEARCHes. Its outputs (sao and saon) drive a dynamic differential XOR gate, which is shown in Figure 3 along with the BiCMOS match line pull-down circuit.

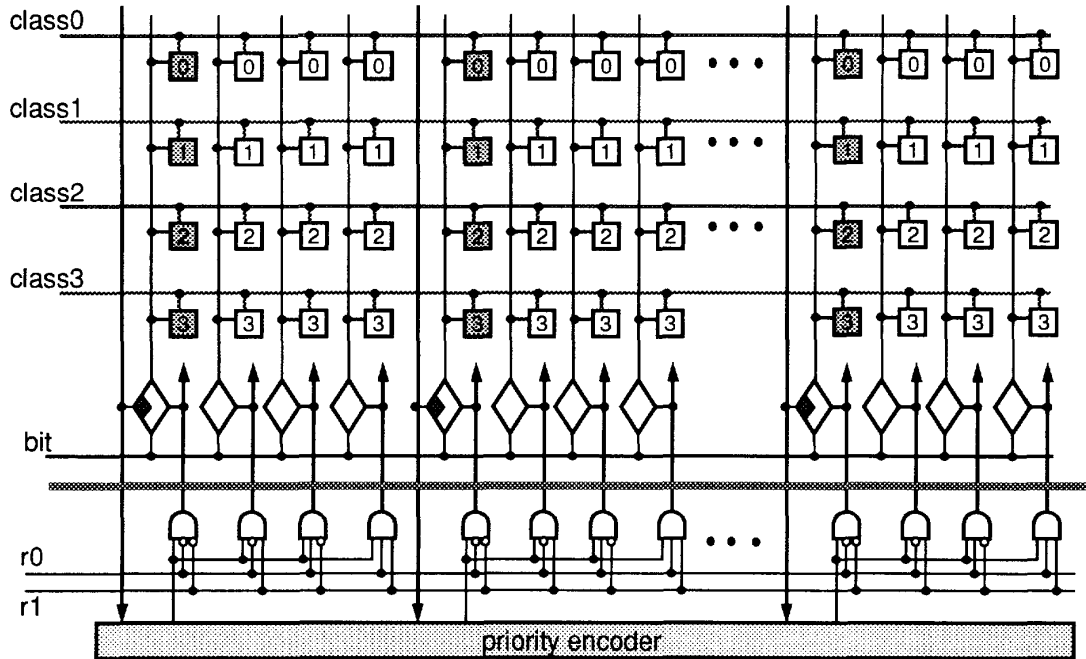


Figure 2: Integrated CAM/RAM architecture. The circuitry above the horizontal gray line represents a single sub-array; circuits below are shared by all sub-arrays. In this example, two additional bits of decoding are applied, resulting in a page size of $2^2 = 4$.

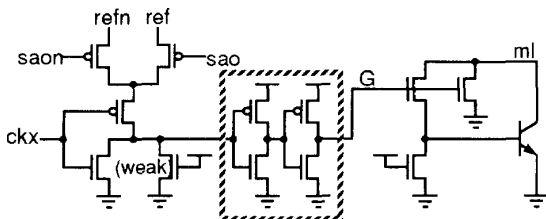


Figure 3: Shared XOR gate and BiCMOS match line pull-down. The extra buffering stages, enclosed by the dashed box, are explained in the text.

A full-rail match line signal transition is desirable, since it leads to faster priority encoder operation. A 5X NPN ($20 \times 0.8 \mu\text{m}$ emitter) provides full pull-down with small load sensitivity, as shown in Figure 4. The worst-case scenario was simulated — a single-bit mismatch, requiring a single circuit to pull down the match line. The rightmost curve, corresponding to a load of 1024 memory rows plus 32 pull-downs (31 inactive), confirms operation of this circuit for megabit capacities; the load sensitivity is 240 ps/pF. For this load, match line power dissipation is 8.0 mW per line at 40 MHz, or 1.0 W for a typical configuration. A $160 \mu\text{m}$ PMOS device is used for match line precharge.

Notice in Figure 3 that the reference inputs of the XOR (ref/refn) are diffusion loads. In the case of a mismatch ($\text{ref} \neq \text{sao}$), one of these inputs — driven by a 5X push-pull

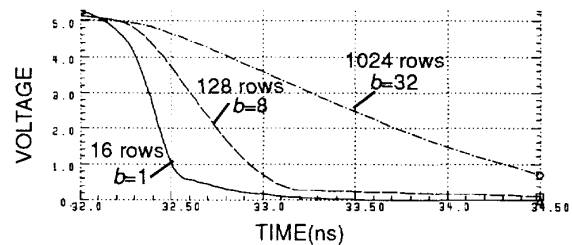


Figure 4: Simulated match line pull-down waveforms for various loads.

BiCMOS buffer (not shown) — is required to supply current to the pull-down circuit through two series PMOS transistors. In the worst case, all XOR gates connected to the ref/refn signals have a mismatch of the same polarity. To limit the dynamic load on these nodes, two extra stages of buffering are used in the pull-down gate, as shown in the box of Figure 3. The effect of this extra buffering is demonstrated in Figure 5, where the ref/refn signals must drive 128 XOR gates. In Figure 5(a), only a single data mismatch occurs, leading to minimal dynamic loading; inclusion of the inverters results in slightly slower operation. However, as shown in Figure 5(b), match line pulldown is not successfully achieved without the extra stages in the case of 128 mismatches of the same polarity.

Area requirements for the bit line precharge and DRSL-column access circuitry (the diamonds in Figure 2)

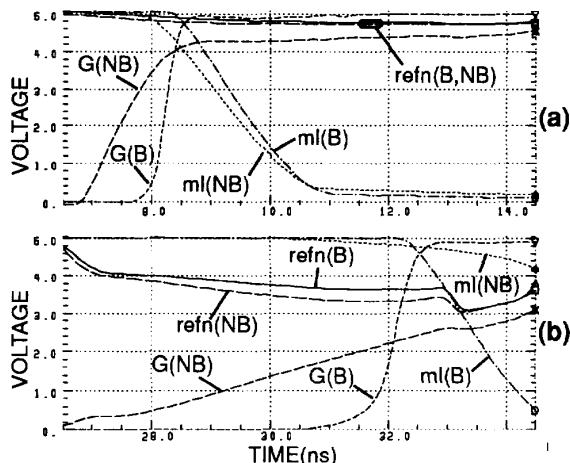


Figure 5: Simulated XOR performance, both with (B) and without (NB) the extra buffer stages outlined in Figure 3: (a) data mismatch in 1 of 128 gates, (b) data mismatch in all 128 gates.

are equivalent to 4.4 memory rows per I/O bit. The CAM sense amplifier, XOR, and match line pull-down occupy 4.1 rows when straddling 4 columns. Fewer rows are required for larger page sizes, since the layout can spread in the column dimension. The minimum allowable page size is 4 words ($r = 2$).

4. Multiple match resolution

The multiple-match resolver (MMR) is the critical path portion of the priority encoder. It selects from among multiple matches, giving priority to matches based on their physical location. To achieve high performance for a large number of inputs, the delay of the MMR must increase sub-linearly with fan-in. Any technique relying on bit-wise signal propagation or long pass-transistor networks is unacceptable. We employ a hierarchical method similar to a carry-look-ahead adder. The implementation of a 16-input MMR is shown in Figure 6. Figure 6(a) demonstrates the hierarchical connection of four 4-input sections; a 64-input section can be built up analogously, using the 16-input sections as the components in a third level of hierarchy. The critical path of such a 64-input MMR consists of only 6 gate delays.

The lowest-level gate is shown in Figure 6(b). The circuit efficiently shares transistors between outputs, and is derived from the NOR logic in [1]. We use NAND logic instead, with inverted match line inputs, resulting in an NMOS series path. This provides higher speed and allows up to 8 inputs. Single-stage inverter “sense amplifiers” are used on the full-rail match lines for simplicity and high speed.

ROM-like circuitry is used to derive the address output of the priority encoder from the MMR outputs.

5. Control circuitry and pipelined operation

Since the CAM/RAM contents are partitioned into classes, and *a priori* knowledge of precise data distribution among classes is impossible to obtain, some classes will fill before the memory, as a whole, is full. For improved memory efficiency, the control circuitry manages overflows into adjacent classes, providing 2-bit count storage for each class and allowing up to 3 adjacent overflows. Once a class has filled, subsequent WRITE requests to that class are directed to an adjacent class, as defined by the counter. If a SEARCH to the target class results in a miss, the number of additional classes denoted by the counter also must be SEARCHED before a true miss has occurred.

This method has small hardware overhead and efficient memory usage, but results in non-deterministic throughput, which may not be acceptable for real-time applications. In fact, all operations (SEARCH, READ, WRITE) are essentially instructions executed by the control circuitry, and they may require multiple clock cycles depending on count status. Alternative methods are described in [3].

Operation is timed by two clocks, as shown in Figure 7(a). A number of constraints must be satisfied:

- For maximum speed, the PRAM and CAM/RAM should operate in parallel; we don’t want two serial memory operations in the critical path; the CAM/RAM should not require PRAM output till the next cycle.
- The control circuitry must take the class from the PRAM (cycle j) and add the appropriate counter value before applying it to the CAM/RAM in cycle $j+1$.
- multiple SEARCHes to multiple classes, required as part of the same SEARCH operation due to overflows, must occur in consecutive clock cycles; therefore, cycle j class processing requires cycle j SEARCH results.
- SEARCH results are not available until some time after the CAM/RAM operation.

Thus, class processing takes place towards the end of cycle j and the result must be available to the CAM/RAM early the following cycle $j+1$. The PRAM must have been read earlier in cycle j . As a consequence, data associated with a SEARCH must be applied to the PRAM one cycle before it is applied to the CAM/RAM.

The result is pipelined operation, requiring the pipeline registers shown in Figure 7(b). Because of the pipeline, the CAM/RAM will be completing a previous SEARCH, READ or WRITE at the same time that the PRAM is READING a new class for a new SEARCH. The data being applied to the PRAM does not interfere with the data being SEARCHED for or written in the CAM/RAM. Pipe registers are also included for addresses, so that they may be applied at the same time as the data. This new address also does not interfere with ongoing CAM/RAM operation. Figure 7(a) shows a READ followed by a SEARCH to demonstrate the concurrence of old and new operations.

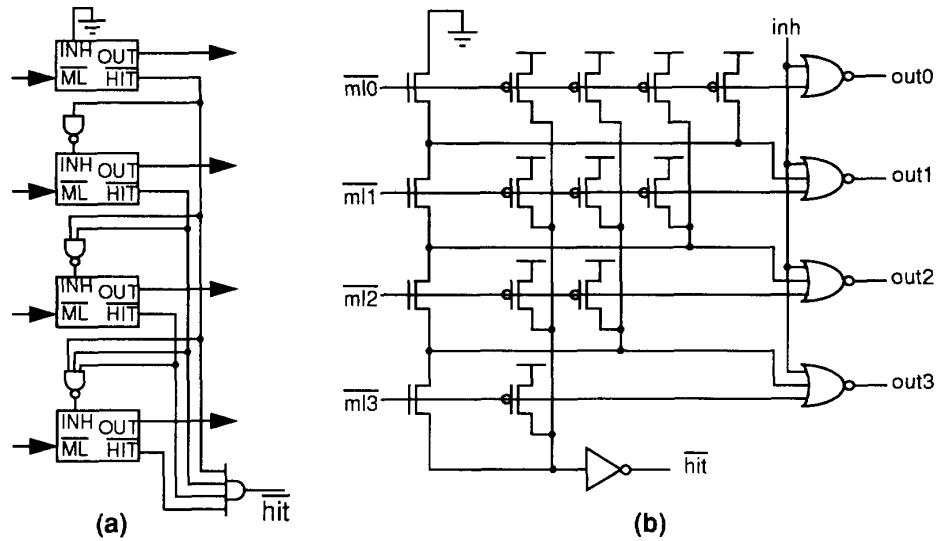


Figure 6: Priority encoder: (a) four 4-bit sections hierarchically connected, (b) 4-bit section.

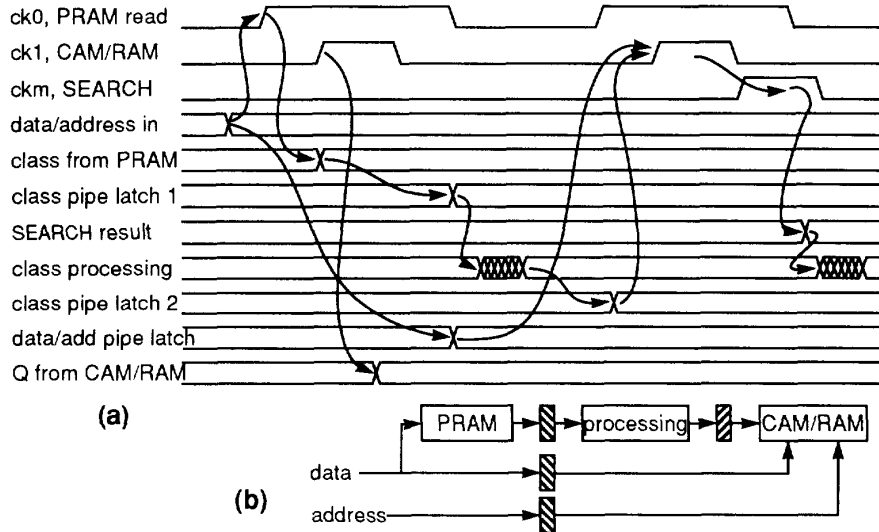


Figure 7: (a) Pipelined operation, READ followed by a new SEARCH; (b) location of pipeline registers.

This scheme allows maximum throughput: $o + n + 1$ cycles for a SEARCH with o overflows followed by n READS or WRITES.

6. Test chip

An 8 kb test chip was implemented in 0.8 μm technology [4] to verify the circuit design. A die photo is shown in Figure 8. The configuration has (using previously-defined variables) $b = 8, C = 16, y = 16, r = 2$. The 64×4 PRAM is located at the lower left; the 6 LSBs of

the input data are used to select 1 of 16 classes. The SRAM core cell size is $131.3 \mu\text{m}^2$ ($13.4 \times 9.8 \mu\text{m}$), and the die size is $3 \times 4 \text{ mm}$, including pads and frame.

Since the RAM arrays are the densest and most structurally complicated part of the chip, we expect the majority of fabrication faults to occur there. In TEST mode, the CAM/RAM can be operated as a series of 64×8 RAMs (one per class) and can be observed from the chip pins, as can the PRAM. This is a valuable test mechanism, even for production versions of chips using this architecture.

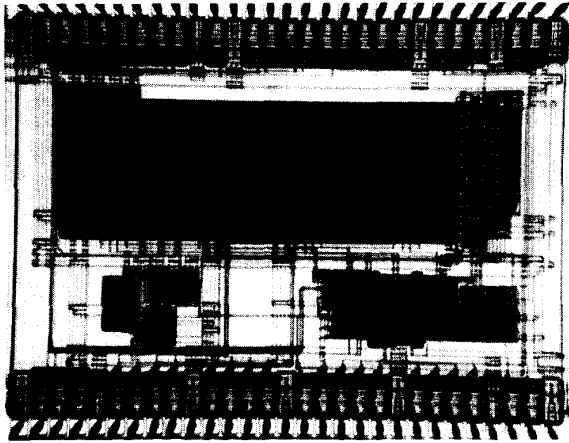


Figure 8: 8 kb CAM/DRAM test chip die photo.

7. Megabit performance estimate

The memory design is scalable up to megabit capacities, and has been successfully simulated with a netlist corresponding to a 1 Mb configuration; we present the results in this section. The configuration has $b = 32$, $C = 32$, $y = 128$, and $r = 3$. We propose a layout wherein I/O circuitry, class decoders, and $ref/refn$ buffers are located in a central spine, with 512 columns on either side. Maximum loading conditions roughly correspond to those analyzed in Section 3 of this paper. The 128-input MMR has three levels of hierarchy: an 8-input lowest level, and two higher levels with 4 inputs each, as described in Section 4.

Waveforms in Figure 9 show a successful SEARCH followed by a READ at 37 MHz. This verifies the critical path

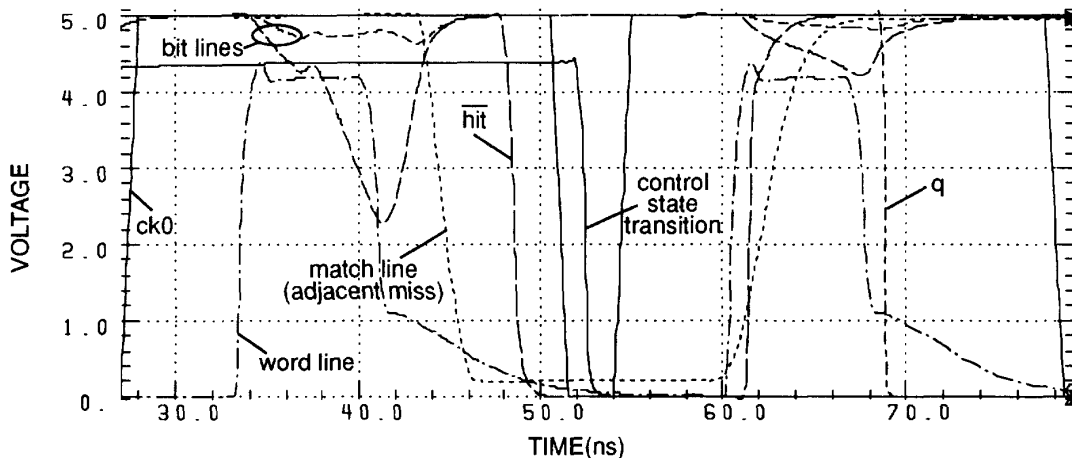


Figure 9: Simulation of a 1 Mb CAM/DRAM at 37 MHz: a successful SEARCH followed by a READ.

from the MMR through the class processing circuitry, and back to the CAM/DRAM wordline decoders.

We believe that the same circuit design, if implemented in state-of-the-art SRAM fabrication technology, would result in multi-megabit capacities and similar speeds.

8. Conclusions

Architectural and circuit innovation are required to implement CAMs with RAM-like densities and capacities. We have described the design of a pre-classified CAM, integrated with its target RAM, emphasizing two critical-path circuits: the match line pull-down and the multiple-match resolver. A speed of 37 MHz has been simulated for a 1 Mb configuration, and larger capacities are projected for state-of-the-art SRAM technology. An 8 kb test chip has been fabricated.

References

- [1] T. Yamagata *et al.*, "A 288-kbit Fully Parallel Content Addressable Memory Using Stacked Capacitor Cell Structure", *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1927-1933, December 1992.
- [2] G. J. Lipovski, "A Four Megabit Dynamic Systolic Associative Memory Chip", *Journal of VLSI Signal Processing*, vol. 4, pp. 37-51, 1992.
- [3] K. J. Schultz and P. G. Gulak, "Architecture for Multi-Megabit Integrated CAM/DRAM", *Proc. Canadian Conf. on VLSI*, pp. 6A1-6A7, 1993.
- [4] R. Hadaway *et al.*, "A Sub-Micron BiCMOS Technology for Telecommunications", *Proc. European Solid-State Device Research Conf.*, pp. 513-516, 1991.
- [5] M. Motomura *et al.*, "A 1.2 Million Transistor, 33-MHz, 20-b Dictionary Search Processor (DISP) ULSI with a 160-kb CAM", *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1158-1165, October 1990.
- [6] A. L. Silburt *et al.*, "A 180-MHz 0.8- μ m BiCMOS Modular Memory Family of DRAM and Multiport SRAM", *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 222-232, March 1993.