A Low-Power Content-Addressable Memory (CAM) Using Pipelined Hierarchical Search Scheme

Kostas Pagiamtzis, *Student Member, IEEE,* and Ali Sheikholeslami, *Senior Member, IEEE*

***Abstract*—This paper presents two techniques to reduce power consumption in content-addressable memories (CAMs). The first technique is to pipeline the search operation by breaking the match-lines into several segments. Since most stored words fail to match in their first segments, the search operation is discontinued for subsequent segments, hence reducing power. The second technique is to broadcast small-swing search data on less capacitive global search-lines, and only amplify this signal to full swing on a shorter local search-line. As few match-line segments are active, few local search-lines will be enabled, again saving power. We have employed the proposed schemes in a 1024 $\times$ 144-bit ternary CAM in 1.8-V 0.18-μm CMOS, illustrating an overall power reduction of 60% compared to a nonpipelined, nonhierarchical architecture. The ternary CAM achieves a 7-ns search cycle time at 2.89fJ/bit/search.**

***Index Terms*—Associative memory, content-addressable memory (CAM), hardware lookup, hierarchical search-lines, high speed, low power, neural network, pattern matching, pipelined hierarchical search scheme, pipelined match-lines, string matching.**

I. Introduction

CONTENT-ADDRESSABLE memories (CAMs) compare search data against a table of stored data and return the address of the matching data. This CAM search function operates much faster than its counterpart in software, and thus CAMs are replacing software in search intensive applications such as address lookup in Internet routers, data compression, and database acceleration [1], [2].

One of the key design challenges of today's high-capacity CAMs is reducing power consumption. The high power consumption of CAMs is due to the parallel nature of the CAM search operation in which a large amount of circuitry is active on every cycle. Since the power consumption of CAMs is proportional to the CAM memory size, CAM power consumption is increasing as applications require larger CAM sizes.

Two main components of power consumption in CAM are the match-line power consumption and the search-line power consumption. Previous efforts in reducing CAM power consumption have focused on reducing match-line power by directly reducing the voltage swing on the match-lines [3], or by using current-based techniques to indirectly reduce the match-line voltage swing [4], [5]. The selective precharge technique reduces match-line power consumption by breaking the search into two segments and observing that the second segment is rarely activated [6]. Despite the recent progress, power consumption in CAM is still high in comparison with RAM of similar size. This paper proposes reducing match-line power by pipelining the match-lines and reducing search-line power by using a hierarchical search-line scheme. The combined effect of these techniques is a power reduction of 60% over the conventional architecture. In the remainder of this section, we review the basic operation of CAM and provide an overview of our proposed CAM architecture. Section II provides details of the pipelined match-line technique and Section III provides details of the hierarchical search-line technique. Section IV describes a testchip architecture that implements the proposed architecture. Section V presents simulation results of the testchip functionality and performance which are confirmed by preliminary measurements of the fabricated chip. Finally, we discuss the implications and tradeoffs of the proposed architecture in Section VI and conclude in Section VII.

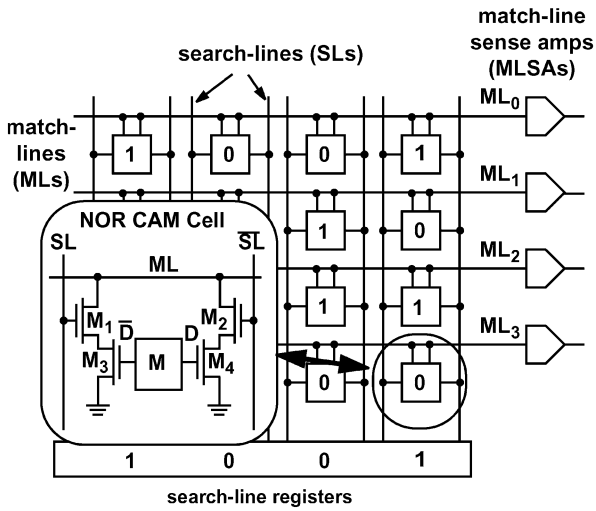

Fig. 1. Simplified conventional (nonpipelined nonhierarchical) CAM architecture. Four match-lines are shown (corresponding to four entries in the CAM) with four bits per match-line.

Fig. 1 shows a block diagram of a simplified conventional (nonpipelined nonhierarchical) CAM architecture. The CAM has four horizontal words (CAM entries) and four bits per entry. The CAM compares the search data (i.e., 1001 in the figure) to all the entries in the CAM, and identifies the words that match. The CAM search operation begins with reseting all the vertical search-lines (SLs) to ground and reseting the horizontal match-lines (MLs) to V_{DD}. The precharge of the SLs prevents wasting of direct-path current during the subsequent precharge

Manuscript received December 22, 2003; revised March 5, 2004. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: pagiamt@eecg.utoronto.ca).

Digital Object Identifier 10.1109/JSSC.2004.831433

of the MLs. The precharge of the MLs to $V_{\rm DD}$ puts them all temporarily in the match state. When the search operation is complete, the match-lines that remain in the match state will identify the words that match the search data.

After the ML precharge, the search-line registers drive the search data onto the differential search-lines $(\overline{\rm SL})$. Then each CAM cell compares its stored bit against the corresponding search bit on the SLs. The inset of Fig. 1 illustrates the schematic of a NOR-based CAM cell. The cell consists of a memory cell, M, which is a six-transistor SRAM cell in this work, and four compare transistors arranged in two pulldown paths between ML and ground. In the cell, a mismatch (or miss for short) between SL and D results in a series path from the ML to ground. On the other hand, a match between SL and D results in no path from ML to ground. The pulldown paths of the individual CAM cells combine on the ML like a dynamic NOR to form either a path to ground (in the case of any miss in the word) or no path to ground (in the case of a full match). In other words, any single miss in any of the cells of a word creates a path to ground that discharges the ML (indicating a miss). Conversely, if all bits of a word match, then the ML remains precharged high (indicating a match). In the example of Fig. 1, the search data, 1001, matches the uppermost word in the array. Hence, its associated ML $({\rm ML}_0)$ remains high indicating a match, while all the other match-lines discharge to ground, indicating misses. The match-line sense amplifiers (MLSAs) are used to distinguish a match from a miss, often using a threshold voltage as the reference.

As mentioned earlier, the two main sources of power consumption are the highly capacitive MLs and the highly capacitive SLs. The ML capacitance consists of the diffusion capacitance of the CAM cells (pulldown transistors) and the ML wire capacitance. This ML capacitance is precharged and discharged in every cycle, since almost all entries miss in the CAM for typical applications. The MLSA power also adds to the ML power consumption and must be included in the overall power consumption of the ML. The SL capacitance is due to the gate capacitance of the compare transistors in the CAM cells (pulldown transistors) and the SL wire capacitance. In conventional search schemes, the SLs are precharged low to ensure there is no path to ground from any ML during ML precharge and thus there is no direct-path current through ${\rm M}_1/{\rm M}_3$ or ${\rm M}_2/{\rm M}_4$ of the CAM cell. After the ML precharge, the SLs are activated according to the search data. As a result, either SL or $\overline{\rm SL}$ transitions twice in one clock cycle, while the other one remains at ground. Similarly, every ML transitions twice in every clock cycle. It is this frequent transition of the SLs and MLs that are the main source of power consumption in CAMs. With our proposed architecture and search scheme, we address this power consumption problem by effectively reducing the amount of capacitance that transitions on each cycle.

Fig. 2 depicts our proposed CAM architecture with pipelined match-lines and hierarchical search-lines [7]. The match-lines are divided into two pipeline segments, each segment with its own MLSA to decide if there is a match and its own pipeline flip-flop to store the outcome of the match operation for the segment. Segmenting the match-lines saves power because most words will miss in the first segment, eliminating the need to activate the subsequent segments.

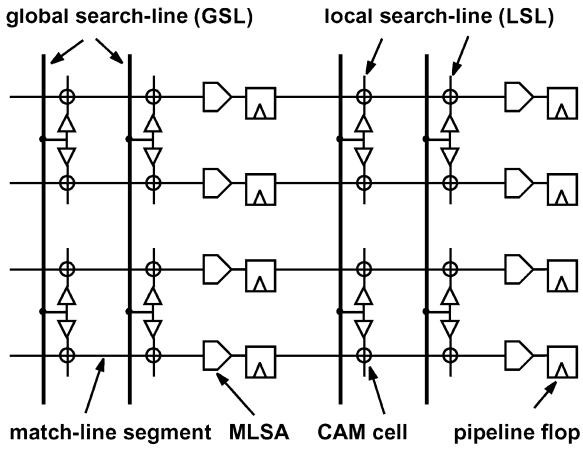

Fig. 2. Simplified block diagram of the proposed CAM with pipelined match-lines and hierarchical search-lines. The match-lines are divided into two segments. The search-lines are divided into GSL (low swing) and LSL (full swing).

To save search-line power, one would ideally use low-swing signaling on the SLs. For example, one could limit the SL swing to $(0{\rm V}, {\rm V}_{tn} + \Delta{\rm V})$, where V_{tn} is the threshold voltage of an NMOS transistor and ΔV is a small incremental voltage above the threshold. This would reduce the SL energy consumption to $C_{\rm SL}(V_{tn}+\Delta V)^2$ ($C_{\rm SL}$ is the SL capacitance) from the original $C_{\rm SL}V_{\rm DD}^2$ when $V_{\rm DD}$ is applied to SL. The price of this reduction in power is a major reduction in match-line speed, caused by a lower gate over-drive voltage on the compare transistors. To mitigate this problem, we propose the two-level search-line hierarchy of Fig. 2. The global search-lines (GSLs) drive the full height of the CAM and feed into the local search-lines (LSLs) which drive only a subset of CAM cells (only a single cell in this simplified figure). Low-swing signals drive the GSLs and local low-swing receivers amplify the GSL signal to the corresponding LSL. However, only LSL receivers feeding an active ML segment are enabled. LSLs feeding an inactive ML segment are disabled. In a large CAM block, many match-line segments are inactive leading to savings in search-line power.

II. Pipelined Match-Lines

Fig. 3 compares the nonpipelined ML architecture and the proposed pipelined ML architecture. Both the single-stage architecture and the pipeline architecture use a current-based MLSA [4]. In this ML sensing scheme, the match-line is precharged low and a current is forced into the ML. MLs in the miss state discharge the current to ground and thus there is little increase in the ML voltage. MLs in the match state collect charge and the ML voltage increases. The NMOS transistor ${\rm M}_{\rm sense}$ turns on only for MLs in the match state, which in turn flips the state of the ensuing half-latch, indicating a match.

The ML is divided into five ML segments, each evaluated sequentially in a pipeline fashion. The left-most segment has 8 bits while the other four segments have 34 bits each, for a total of 144 bits (a typical word width used for IPv6 address lookup). The MLSA current source that provides the $I_{\rm ML}$ current is divided among the five segments in proportion to the number of bits in each segment. This is to guarantee identical speed in all ML segments and to allow a fair comparison with the nonpipelined

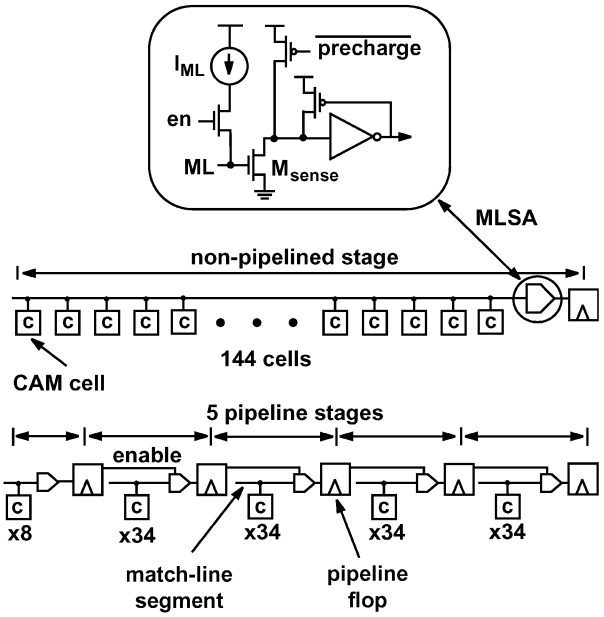

Fig. 3. Pipelined match-line architecture compared to the nonpipelined architecture. The match-lines are broken into five match-line segments. Each segment uses a current-based MLSA [4]. A segment is activated only if all previous segments have been matched.

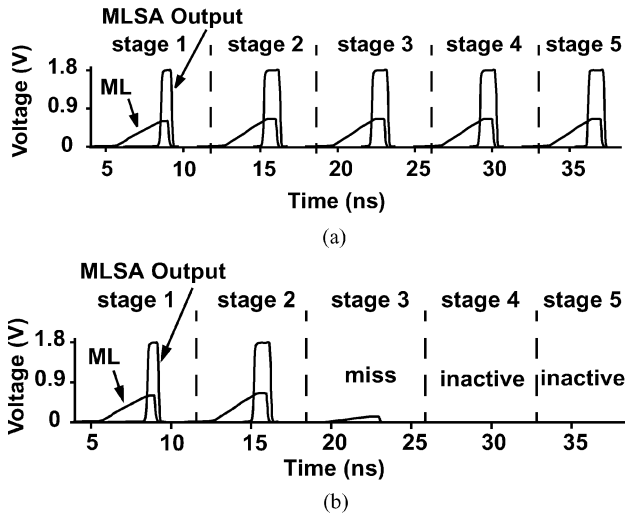

Fig. 4. Simulated waveforms in the pipelined match-line architecture for (a) the full-match case consisting of a match in every stage, and (b) a miss case where the third stage results in a miss and turns off the subsequent stages.

architecture. The pipelined ML operates from left to right, with each ML segment acting as an enable signal for the MLSA of the subsequent segment. Hence, only words that match a segment proceed with the search in their subsequent segments. Words that fail to match a segment do not search for their subsequent segments and hence consume no power.

Fig. 4 depicts simulated signal waveforms of the ML segments in the pipelined ML scheme. Fig. 4(a) shows a full match as indicated by the rising ML in every segment along with the corresponding full-rail output of the MLSA. Fig. 4(b) shows an example of a word that misses in the third stage as indicated by the lack of an MLSA output pulse. In this example, the ML

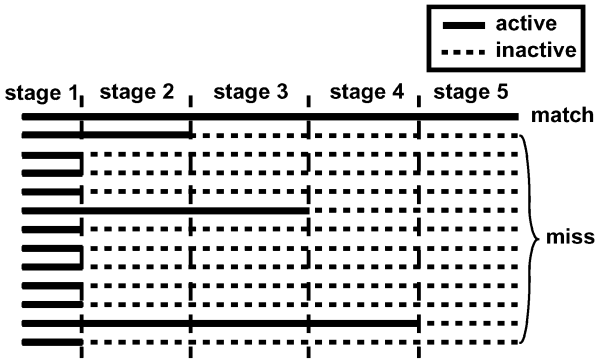

Fig. 5. Example of the ML activity for a typical search operation. The thick lines represent active ML segments consuming power and the dotted lines represent inactive ML segments that consume no power.

sensing circuitry of the fourth and fifth segments are not activated, hence saving power.

There is a power overhead incurred by the addition of flip-flops in each ML segment as well as by the repetition of circuitry in the MLSAs that cannot be divided among the segments, such as the half-latch. However, as we will see in Section V, this power overhead is more than compensated by the power savings due to shutting off match-line segments. Also, there is an area overhead due to the additional circuitry. Section VI discusses the impact of these overheads in more detail.

The key benefit of this architecture is that it exploits the same effect as the selective precharge scheme [6] to reduce match-line power consumption. In typical CAM applications, such as router address look-up, only one or two words match and all other words miss. Furthermore, most words will miss on the first few bits. This fact is exploited in power reduction by allocating only 8 bits to the first segment so that the majority of words will miss in this segment. The subsequent segments are larger (34 bits) to minimize the amount of duplicated sensing and pipeline circuitry. This larger segment size has little impact on the match-line power consumption. Fig. 5 illustrates the ML activity of this architecture for a typical search operation. The dotted lines indicate the large proportion of ML segments that are inactive and thus saving power.

III. Hierarchical Search-Lines

Having pipelined the match-lines, the significant portion of the power is now consumed by the highly capacitive search-lines. We address this problem by observing how the SLs are activated in the pipelined ML architecture. As the match signals traverse the pipeline stages from left to right, fewer ML segments survive the matching test and hence fewer ML segments will be activated. However, the SLs must be activated for the entire array at every stage of the pipeline, since the search-lines must reach the surviving match-line segments. This excessive power consumption is curtailed in our design by breaking the search-lines into global and local search-lines (GSLs and LSLs), with the GSLs using low-swing signaling and the LSLs using full-swing signaling but with reduced capacitance. Also, by a GSL not directly serving every single CAM cell on a search-line, the GSL capacitance is further reduced, resulting in extra power savings.

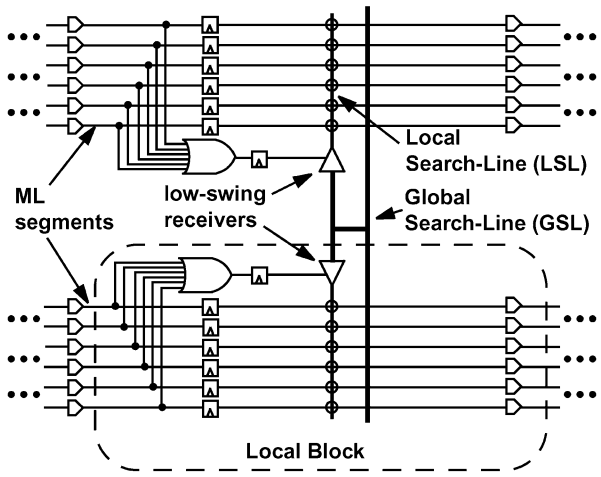

Fig. 6. Simplified schematic of the hierarchical search-line architecture. The SLs are divided into GSL (low swing) and LSL (full swing). The match results of the previous ML segments are ORed to decide whether to activate the low-swing receivers that drive the LSL.

A simplified block diagram illustrating this concept is shown in Fig. 6. Two local blocks are shown in detail in the diagram, with each local block spanning the height of 64 match-lines. In our 1024×144 bit architecture, each GSL feeds 16 LSLs. The search data are broadcast on the GSLs using low-swing signaling. As we will see later in this section, low-swing receivers on the GSLs translate a low-swing voltage of $V_{\mathrm{DDLOW}} = 0.45$ V to a rail-to-rail signal of 1.8 V on the LSLs. Each receiver is gated by a separate enable signal which is generated by ORing the match results of the previous local block. Thus, a low-swing receiver is enabled and the corresponding LSL is driven only when at least one incoming match-line is active. In most cases, no incoming match-line of a local block is active. As a result, the receiver of that local block is not clocked, hence keeping the corresponding LSL inactive, in turn saving power.

Fig. 7 shows a more detailed schematic of the hierarchical search-line circuitry. The global clock latches on the negative edge the logical OR of all the match results of the previous block. This OR result determines whether the LSLs of the current block should be activated. The negative-edge triggered receiver clock is generated by NANDing the global clock with the latched OR signal. There is a half-clock cycle timing margin between the latching of the OR signal and the arrival of the positive edge of the global clock at the NAND gate. This timing margin is sufficient to prevent any glitches on the receiver clock. To prevent an early arrival of the low-swing GSL signal at the low-swing receiver, an inverter chain delays the latching of the SL input. The clocking scheme of this architecture allows for incorporation of the hierarchical search-line scheme into a self-timed CAM design [8].

To keep the power consumption of the GSL to a minimum, small devices are used in the low-swing drivers. Although this slows down driving the GSL, a full pipeline cycle is dedicated to the low-swing driver. As this extra cycle operates concurrently with the ML pipeline, it does not contribute to the overall system latency. Fig. 8 shows a complete schematic of the hierarchical search-line scheme including a full pipeline stage with two local blocks.

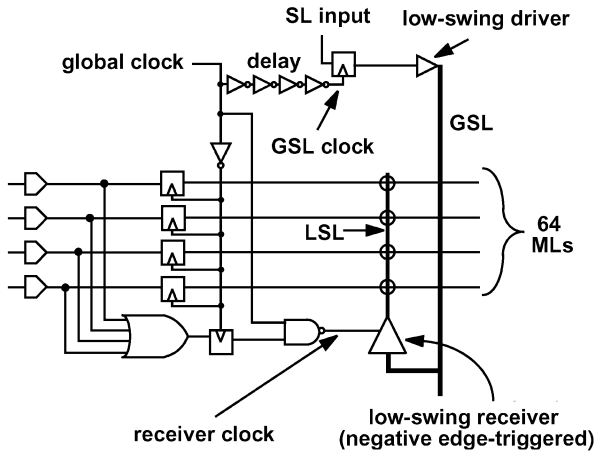

Fig. 7. Circuit operation of hierarchal search-line scheme. The global clock synchronizes operation by generating the GSL clock and the gated receiver clock.

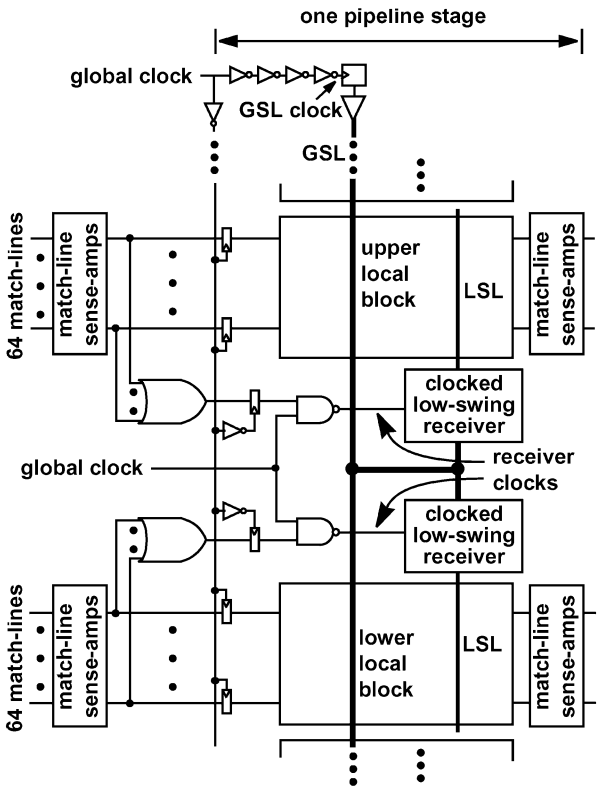

Fig. 8. Complete schematic of the hierarchical search-line architecture. Low-swing global search-lines drive the full height of the CAM macro. Full-swing local search-lines span the height of a local block. The clocked low-swing receivers are enabled only when the previous local block has at least one match, saving power.

Fig. 9 shows the schematic for a low-swing receiver [9] in which V_{DDLOW} is an externally generated supply voltage. The receiver is an edge triggered sense amplifier that compares the low-swing GSL (feeding into the PMOS pair on the left) with the reference inputs (feeding into the PMOS pair on the right). The right-most PMOS gate is connected to V_{DDLOW} and the other gate is connected to ground, effectively generating a total output current corresponding to an input voltage of $V_{\mathrm{DDLOW}}/2$. Although this relationship is approximate, there is sufficient

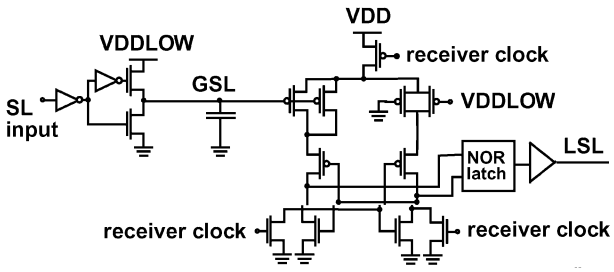

Fig. 9. Schematic of low-swing receiver [9]. The receiver amplifies the GSL on the negative edge of the receiver clock. The receiver's output is captured by the NOR latch and driven rail-to-rail on the LSL. When the receiver is not clocked, the latch holds the previous data on the LSL and thus consumes no power.

voltage margin to ensure correct operation. The benefit of generating the reference in this manner is the elimination of the requirement of explicitly generating a voltage of $V_{\mathrm{DDLOW}}/2$. The receiver resets when the clock is high and samples the input on the negative edge. The sense amplifier output nodes R and S are quiescent low and one of R or S pulse high on a data transition, feeding the following NOR latch. The combination of the receiver and the NOR-latch behave as a negative edge-triggered flip-flop. This flip-flop feeds the buffer that drives a rail-to-rail output on the LSL.

To see how the hierarchical search-line architecture saves power over the nonhierarchical architecture, we examine the average energy consumption on each search-line transition (up and down) in the nonhierarchical architecture:

$$E_{\mathrm{conv.}} = C_{\mathrm{SL}} V_{\mathrm{DD}}^2 \tag{1}$$

where C_{SL} is the capacitance of a search-line and V_{DD} is the supply voltage. In the hierarchical scheme, the corresponding power consumption is

$$E_{\mathrm{hier.}} = \underbrace{C_{\mathrm{GSL}} V_{\mathrm{DDLOW}}^2}_{\mathrm{global}} + \underbrace{\alpha N C_{\mathrm{LSL}} V_{\mathrm{DD}}^2}_{\mathrm{local}} + E_{\mathrm{overhead}} \tag{2}$$

where C_{GSL} is the capacitance of a global search-line, and C_{LSL} is the capacitance a local search-line. N represents the number of LSLs per GSL and α is the activity factor of a local block. In our implementation, $C_{\mathrm{GSL}} \approx C_{\mathrm{SL}}/6$ (discussed later in Section VI). E_{overhead} is due to the extra OR gates, clocking circuitry, and low-swing transmitter and receiver circuitry. For $N = 16$ (as implemented in our design) and a typical $\alpha = 20\%$, $E_{\mathrm{hier.}} = 0.37\, E_{\mathrm{conv.}}$, indicating a 63% search-line power reduction.

IV. Testchip Architecture

The proposed pipelined match-line and hierarchical search-line architecture has been targeted for a 1024×144-bit ternary CAM macro, although due to limited silicon area, we have designed a 256×144-bit testchip for fabrication. The testchip is implemented in a 1.8-V 0.18-μm CMOS process without resorting to special devices. Fig. 10 shows the overall organization of the testchip. The 144-bit MLs are segmented such that the first (left-most) segment has 8 bits and the subsequent segments each have 34 bits. The SLs are divided into 64-entry local blocks. Only the second and third segments (in grey) implement the hierarchical search-lines. The fourth and fifth segments use nonhierarchical search-lines. This organization allows for direct comparison of the power consumption of the hierarchical versus the nonhierarchical search-lines.

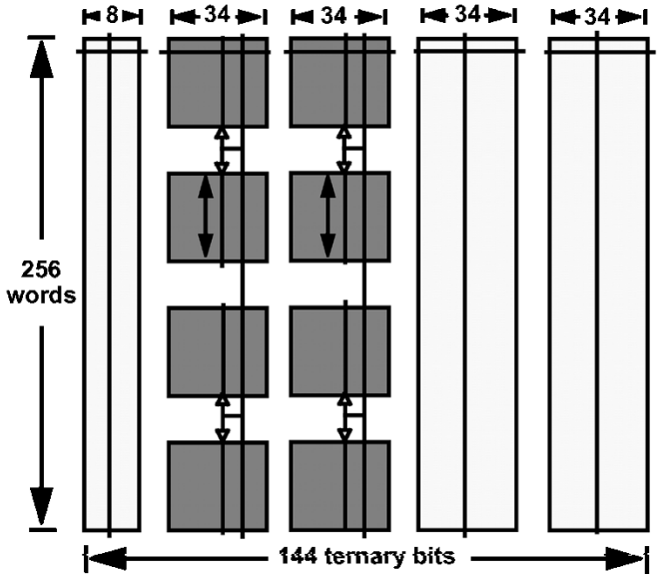

Fig. 10. Testchip architecture designed for a 1.8 V 0.18 μm CMOS process. The MLs are divided into 5 segments; the first segment has 8 bits and the subsequent segments have 34 bits. To allow for direct comparison of power consumption, the second and third segments (in grey) implement the hierarchical search-line architecture, and the fourth and fifth segments implement the nonhierarchical search-line architecture.

TABLE I
Chip Characteristics

Technology	0.18 µm, 6-metal
Organization	256×144 ternary bits
Chip Area	2.3 mm × 2.1 mm
Supply Voltage	1.8 V
Cycle Time	7 ns (142 MSearch/sec)

Table I summarizes the features of the testchip, with a photomicrograph of the fabricated die shown in Fig. 11. Indicated on the photo are an ML segment, a GSL, two LSLs, a local block, and low-swing receivers. The peripheral test circuitry consists of registers for shifting in the input data and shifting out the match results.

V. Simulation and Measurement Results

This section presents simulation results of the functionality and power consumption of the proposed CAM architecture along with preliminary measurement results of the fabricated testchip. The simulation results in this section include the effect of parasitics extracted from layout. Timing is determined by simulating all the subcircuits in HSPICE, and power consumptions are determined by simulating the complete CAM macro netlist in Nanosim [10], a transistor-level simulator capable of handling large netlists. In determining typical power consumption, we have assumed that there is only one matching location per search in the CAM that is populated with uniformly distributed random data. The worst case power consumption would occur when there are a large number of matches per cycle. This is too rare to be considered in most CAM applications and thus is not simulated or measured in this work.

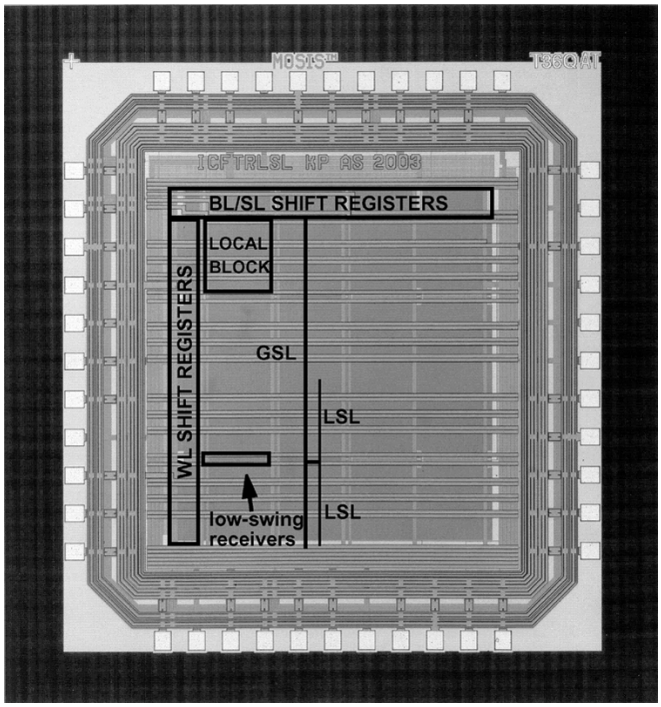

Fig. 11. Die photomicrograph of the testchip implemented in a 1.8-V 0.18-μm CMOS process.

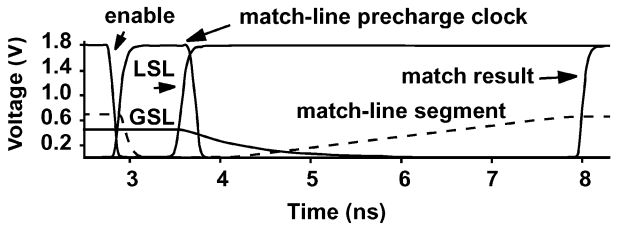

Fig. 12. Signal waveforms for a full search cycle on a 34-bit segment. The cycle time is dominated by the charging of the match-line segments.

Fig. 12 displays the simulated signal waveforms for a single-cycle search operation in a 34-bit segment of the CAM testchip. The cycle begins by clocking the low-swing receiver (negative-edge triggered) to sense the current cycle's GSL data and by clocking the GSL flip-flop (top of Fig. 8) to activate the next cycle's GSL data. After the receiver drives the LSL, the match-line sensing circuitry is activated. The match-line sensing activation is controlled independently for testing purposes; however, in a production system, a replica LSL could be used to activate the match-line sensing. The match-line sensing clock initiates charging of the match-lines by their respective current sources. A replica match-line, programmed to hit on every cycle, generates the shut-off signal for the current sources. Only match-lines that have a match will pass the threshold voltage of the NMOS transistor before shut-off, tripping the sense amplifier and indicating a match [4]. This scheme achieves a search cycle time of 7 ns.

Fig. 13 compares the energy consumption of the proposed architecture with the nonpipelined nonhierarchical architecture in units of fJ/bit/search, a common metric [11] for comparison of

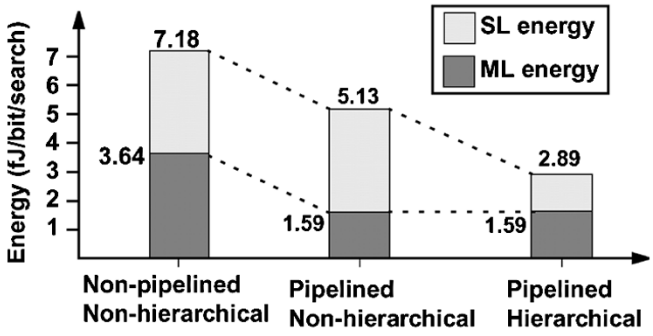

Fig. 13. CAM search cycle energy consumption for the nonpipelined nonhierarchical architecture, pipelined nonhierarchical architecture, and pipelined hierarchical architecture. The energy consumptions are simulated for 1024 $\times$ 144 blocks in a 1.8-V 0.18-μm CMOS process.

CAM energy consumption. The energy consumptions are simulated for 1024 $\times$ 144 blocks in a 1.8-V 0.18-μm CMOS process with uniformly distributed data in the CAM. The power consumption of the nonpipelined nonhierarchical CAM architecture is 7.18 fJ/bit/search consisting of 3.64 fJ/bit/search ML energy consumption and 3.54 fJ/bit/search SL energy consumption. The addition of pipelined match-lines reduces the overall energy consumption to 5.13 fJ/bit/search, consisting of an ML energy consumption of 1.59 fJ/bit/search (a 56% reduction) and no change in SL energy. Since most ML segments miss in the first 8-bit stage (rarely activating the subsequent 136 bits), one expects the ML energy consumption to be reduced by about $(136/144)$, or about 95%, compared to the nonpipelined architecture. The simulation results, however, indicate a 56% ML energy reduction. The difference is due to the overhead of clocking the ML flip-flops and the repeated circuitry in the MLSAs of each ML segment. Conventional master–slave flip-flops with pass-transistor-based latches were used in our implementation [12]. Replacing these flip-flops, with low-power flip-flops or pulsed latches could further reduce the ML energy consumption.

Having pipelined the match-lines, the search-line activity dominates the overall power consumption, as evident in the second bar of Fig. 13. Adding hierarchy to the search-lines reduces the energy consumption further to 2.89 fJ/bit/search, consisting of SL energy consumption of 1.3 fJ/bit/search (a 63% reduction) and an ML energy consumption of 1.59 fJ/bit/search. Overall, the combination of pipelined match-lines and hierarchical search-lines reduces power consumption by 60%, from 7.18 fJ/bit/search to 2.89 fJ/bit/search.

As noted earlier, the hierarchical and nonhierarchical SLs have separate power supplies that are indepedent of other circuitry. These dedicated power supplies are used in the preliminary measurement results that follow to measure the power consumption of the two SL schemes. To measure the nonhierarchal SL energy, we toggle the SLs on the testchip for a number of transitions and determine the average SL energy to be 3.94 fJ/bit/search. We simulate the extracted netlist of the testchip with the same input vectors and obtain a simulated value of 3.54 fJ/bit/search, which is about 10% below the measured value. Unlike the nonhierarchical SLs, the energy of the hierarchical SLs is a function of the number active local blocks. To measure the energy per block, we create a set of test

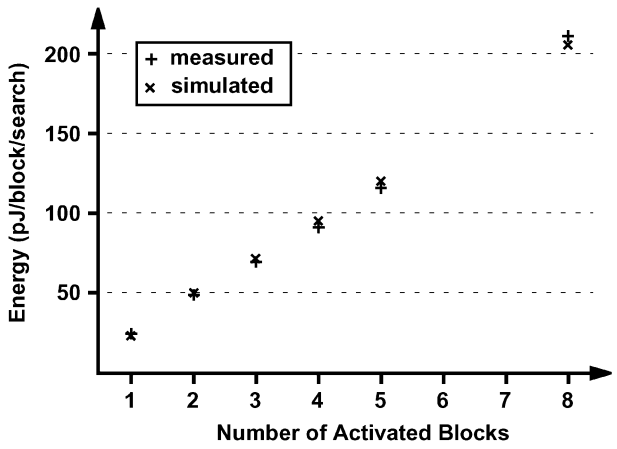

Fig. 14. Measured energy dissipation for hierarchical search-lines versus the number of activated blocks.

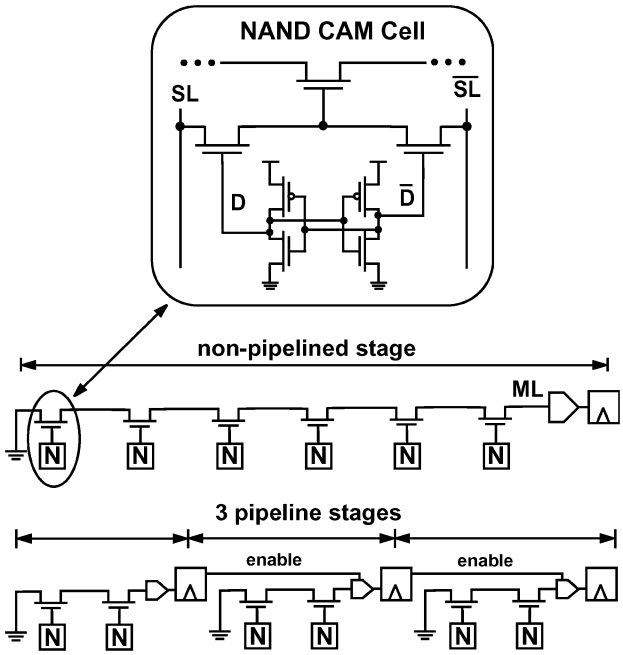

Fig. 15. A 6-bit word in a NAND-based CAM ML is pipelined in three stages, for illustration purposes. The overall time latency of the pipelined ML is expected to be similar to that of the nonpipelined ML.

vectors that activate a subset of local blocks. Fig. 14 compares the measured versus simulated energy per block as we increase the number of active blocks from 1 to 8. The measured energy values are within 5% of the simulated values.

VI. Discussion

A. Match-Line Segment Sizes and Local Search-Line Length

We explore means of reducing power consumption by referring to (2). Assuming the design is using a fixed V_{DDLOW} (0.45 V in this design) and a fixed V_{DD} (1.8 V), the power consumption can be minimized by an optimum combination of $C_{\mathrm{GSL}}, C_{\mathrm{LSL}}, N$, and α. A small N, for example, reduces E_{overhead} but at the same time increases α. Our simulation results indicate a minimum power consumption is achieved when N is between 8–32. By choosing $N = 16$, we reduce the area overhead to 6%. This area overhead is caused by the number of low-swing receivers which is proportional to N (the area overhead is negligible for the additional OR gates—implemented as dynamic NORs–and clocking circuitry). Second, by choosing to route the LSLs in Metal 2 and the GSLs in Metal 4, we reduce C_{GSL} (without affecting any other parameter in (2)).

B. Further Reduction in Power

The pipelined hierarchical search scheme presented in this paper is flexible in that it does not depend on a specific MLSA and ML sensing scheme. Furthermore, other techniques such as selective precharge [6], or bank selection schemes [13], [14] can be applied to further reduce the power consumption.

C. Pipelined Match-Line for NAND-Based CAM Architecture

In this paper, we describe the pipelined match-line scheme for a NOR-based CAM architecture in which the cells are connected in parallel (dynamic NOR fashion) to form a match-line. This scheme is also suitable for dynamic NAND-based CAM architecture [8] as described below.

In NAND architecture, the CAM cells are connected in series, similar to the pull-down network in multi-input NAND gate. The series combination of the cells allows a natural breakdown for pipeline stages, where the overall search latency of a pipelined match-line is approximately equal to the search latency of a nonpipelined match-line. To illustrate, Fig. 15 shows an example of a NAND-based architecture in which a 6-bit match operation is pipelined in three stages of two bits each. Assuming all stages have the same 2-bit propagation latency, the overall latency of the pipelined match-line is 6 bits, which is equal to the latency of the original nonpipelined match-line. Note that this is not the case in a NOR-based architecture, as the overall search time is increased by the number of stages in the pipeline. The advantage of pipelining match-lines in a NAND CAM is that while it saves power, it preserves the overall search time.

VII. Conclusion

To save power, we have explored adding pipelining to match-lines and adding hierarchy to search-lines in an otherwise nonpipelined, nonhierarchical CAM. The power savings of the pipelined match-lines is a result of activating only a small portion of the ML segments. Similarly, the power savings of the hierarchical search-lines is a result of activating only a small portion of the LSLs. Pipelining match-lines saves 56% power compared to nonpipelined match-lines. Adding hierarchy to search-lines saves 63% power compared to nonhierarchical search-lines. The combination of the two techniques reduces overall power consumption by 60%.

Acknowledgment

The authors thank the anonymous reviewers for their excellent comments on the first draft of this paper, I. Arsovski and M. van Ierssel for insightful discussions on this work, and B. Stevenson and M. Jarosz of the Canadian Microelectronics Corporation (CMC) for assistance in testing. The authors also

gratefully acknowledge chip fabrication and CAD tool access provided by the Canadian Microelectronics Corporation.

REFERENCES

[1] T.-B. Pei and C. Zukowski, "Putting routing tables in silicon," *IEEE Network Mag.*, vol. 6, pp. 42–50, Jan. 1992.

[2] L. Chisvin and R. J. Duckworth, "Content-addressable and associative memory: Alternatives to the ubiquitous RAM," *IEEE Computer*, vol. 22, pp. 51–64, July 1989.

[3] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE J. Solid-State Circuits*, vol. 36, pp. 956–968, June 2001.

[4] 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 J. Solid-State Circuits*, vol. 38, pp. 155–158, Jan. 2003.

[5] I. Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content-addressable memories," *IEEE J. Solid-State Circuits*, vol. 38, pp. 1958–1966, Nov. 2003.

[6] C. A. Zukowski and S.-Y. Wang, "Use of selective precharge for low-power content-addressable memories," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 1997, pp. 1788–1791.

[7] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2003, pp. 383–386.

[8] F. Shafai, K. J. Schultz, G. F. R. Gibson, A. G. Bluschke, and D. E. Somppi, "Fully parallel 30-MHz, 2.5-Mb CAM," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1690–1696, Nov. 1998.

[9] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: Effectiveness and robustness," in *IEEE Trans. VLSI Syst.*, vol. 8, June 2000, pp. 264–272.

[10] *Nanosim Reference Guide*, Synopsys, Mar. 2002.

[11] I. Y.-L. Hsiao, D.-H. Wang, and C.-W. Jen, "Power modeling and low-power design of content addressable memories," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 2001, pp. 926–929.

[12] J. M. Rabaey, A. Chandrakasan, and B. Nikoliæ, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[13] F. Zane, G. Narlikar, and A. Basu, "CoolCAMs: Power-efficient TCAM's for forwarding engines," in *Proc. IEEE INFOCOM*, vol. 1, 2003, pp. 42–52.

[14] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, "200 MHz/200 MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2003, pp. 387–390.

Kostas Pagiamtzis (S'98) received the B.A.Sc. degree (with honors) in computer engineering from the Division of Engineering Science in 1999 and the M.A.Sc. degree in the Department of Electrical and Computer Engineering in 2001, both at the University of Toronto, Toronto, ON, Canada. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Toronto, while holding a Natural Sciences and Engineering Research Council of Canada (NSERC) PGS B postgraduate scholarship.

He spent 16 months as a student intern with the Memory Development Group, Nortel Networks, in 1998. He held a summer internship in 1999 at SiberCore Technologies, Ottawa, Canada. His primary research interest is in architecture and circuit design for content-addressable memories. He is also interested in the design of emerging nonvolatile memories including ferroelectric and magnetoresistive memories, and architectures for digital communication applications such as error control coding and networking.

Mr. Pagiamtzis received the Teaching Assistant of the Year Award in 2002 and in 2003 by popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto.

Ali Sheikholeslami (S'98–M'99–SM'02) received the B.Sc. degree from Shiraz University, Shiraz, Iran, in 1990 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1994 and 1999, respectively, all in electrical and computer engineering.

In 1999, he joined the the Department of Electrical and Computer Engineering, University of Toronto, where he is currently an Assistant Professor and holds the L. Lau Junior Chair in electrical and computer engineering. His research interests are in the areas of analog and digital integrated circuits, high-speed signaling, VLSI memory design (including SRAM, DRAM, and CAMs), and ferroelectric memories. He has collaborated with industry on various VLSI design projects in the past few years, including work with Nortel, Canada, in 1994, with Mosaid, Canada, since 1996, and with Fujitsu, Japan, since 1998. He is currently supervising three active research groups in the areas of ferroelectric memories, CAMs, and high-speed signaling. He has coauthored several journal and conference papers, and received two U.S. patents on CAMs in 1998 and 1999.

Dr. Sheikholeslami received the Best Professor of the Year Award in 2000 and 2002 by the popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto. He has served on the Memory Subcommittee of the IEEE International Solid-State Circuits Conference (ISSCC) since 2001, and on the Technology Directions Subcommittee of the same conference since 2002. He presented a tutorial on ferroelectric memory design at the ISSCC 2002.