Pipelined Match-Lines and Hierarchical Search-Lines for Low-Power Content-Addressable Memories

Kostas Pagiamtzis and Ali Sheikholeslami
Department of Electrical and Computer Engineering
University of Toronto, Canada
{pagiamt,ali}@eecg.toronto.edu

***Abstract*— This paper presents a pipelined match-line and a hierarchical search-line architecture to reduce power in content-addressable memories (CAM). The overall power reduction is 60%, with 29% contributed by the pipelined match-lines and 31% contributed by the hierarchical search-lines. This proposed architecture is employed in the design of a 1024 × 144 bit ternary CAM, achieving 7 ns search cycle time at 2.89 fJ/bit/search in a 0.18 µm CMOS process.**

I. Introduction

Content-addressable memories (CAMs) compare in parallel input search data to all contents of memory and return the address of the data. Search-intensive tasks that benefit from CAMs include the address lookup function in Internet routers, data compression, database acceleration, and neural networks [1], [2]. With these applications demanding increasing CAM capacity and higher speed, the main challenge in CAM design remains that of reducing power.

There are two major sources of dynamic power consumption in CAM: match-line power and search-line power. Previous research has mostly focused on power reduction on the match-lines [3]–[6] while little attention has been given to power reduction on the search-lines. Match-line sensing techniques, for example, focus on minimizing the signal swing on the match-lines by reducing the precharge and discharge levels [3] or by resorting to current-based techniques [4], [5]. Selective precharge reduces match-line power dissipation by breaking the search into two segments [6]. Despite the recent progress, power dissipation in CAM is still high compared to RAM of similar size.

This paper presents two techniques for power reduction in CAM: pipelined match-lines and hierarchical search-lines. These techniques are conceptually depicted in Fig. 1 in which match-lines are segmented and the search-lines are divided into a two-level hierarchy. The horizontal match-lines are partitioned into five segments by flip-flops. This partitioning reduces the capacitance on each match-line segment, and therefore reduces search cycle time. Also, segmenting reduces power since most words fail to match in the first or second segments of the search word and, hence, will not require activating subsequent segments. The vertical search-lines, which broadcast the input data to the CAM cells, are divided into a two-level hierarchy with global search-lines (GSLs) running the full height of the CAM and feeding the local search-lines (LSLs) which span the height of a local block. These techniques reduce power consumption by 60% compared to a conventional architecture. In the remainder of this paper, we demonstrate how power is reduced. We summarize the energy dissipation results in Fig. 2 for a 1024 × 144 bit ternary CAM macro in a 0.18 µm CMOS process.

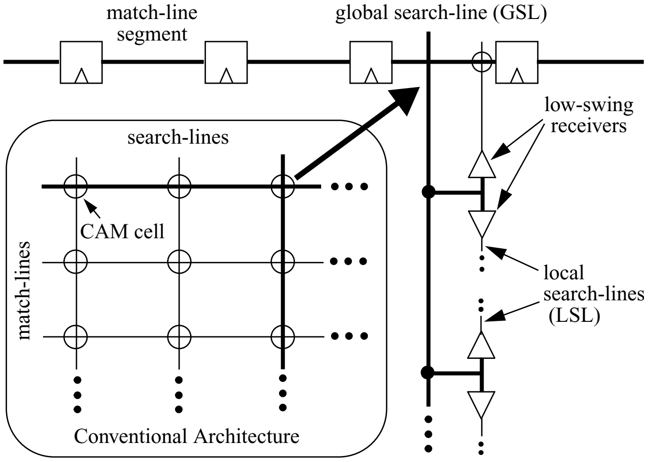

Fig. 1. CAM architecture with pipelined match-lines and hierarchical search-lines.

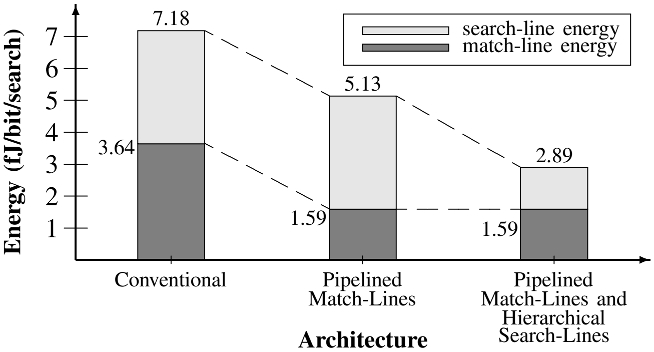

Fig. 2. CAM search cycle energy dissipation for the conventional architecture, pipelined match-lines, and pipelined match-lines with hierarchical search-lines.

II. Pipelined Match-Lines

Fig. 3 shows the pipelined match-line architecture. Pipeline flip-flops break a match-line into five segments, each segment with its own match-line sense amplifier. The left-most segment has 8 bits while the other four segments have 34 bits each,

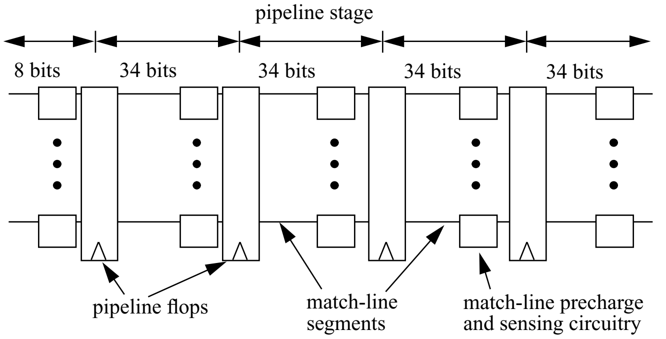

Fig. 3. Pipelined match-line architecture. The match-lines are broken into five match-line segments. A segment is activated only if there is a match in the previous segment.

for a total of 144 bits (a typical word width used for IPv6 address look-up). From left to right, each match-line segment is used as an enable signal for the match-line sense amplifier of the following segment of the same match-line. Hence, only words that match a segment are searched for a match in their following segments. Words that fail to match a segment are not searched for their following segments and hence consume no power.

This architecture has two key benefits. First, a CAM with pipelined match-lines scales to arbitrary word widths while maintaining a small search cycle time. The second benefit of this architecture is that it exploits the same effect as the selective precharge [6] scheme 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.

The pipelined match-line architecture reduces the power consumption on the match-lines by 56% compared to the conventional match-line architecture. As a result of pipelining, the search-line activity dominates the overall power consumption, which is evident in the second bar of Fig. 2. In the next section, we describe a novel architecture for reducing search-line power consumption.

III. Hierarchical Search-Lines

As the match signals traverse the pipeline stages from left to right, fewer match-line segments survive the matching test and hence fewer match-line segments will be activated. However, the search-lines 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.

A general block diagram illustrating this concept is shown in Fig. 4. 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. Low-swing receivers on the GSLs translate the low-swing voltage of $V_{DDLOW} = 0.45\,\mathrm{V}$ to a rail-to-rail signal of $1.8\,\mathrm{V}$ on the LSLs. V_{DDLOW} is an externally generated supply voltage. As indicated in the diagram, 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 which keeps the corresponding LSL inactive, therefore saving power.

We use a simple clocking scheme for this architecture that allows incorporation into a self-timed design [7]. The global clock, shown at the top of Fig. 4, is used to derive the GSL

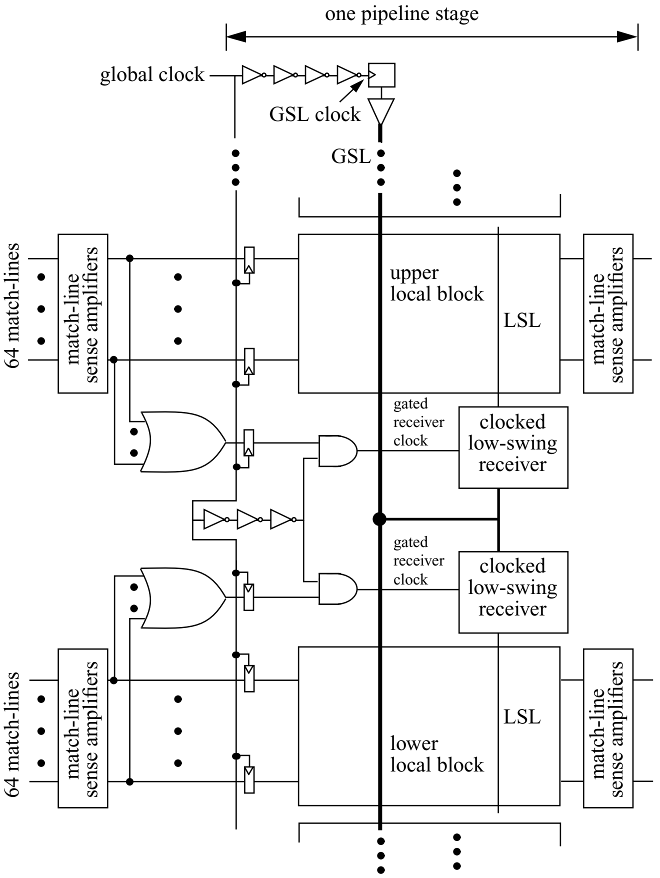
one pipeline stage
global clock
GSL clock
GSL
64 match-lines
match-line sense amplifiers
upper local block
LSL
match-line sense amplifiers
gated receiver clock
clocked low-swing receiver
gated receiver clock
clocked low-swing receiver
64 match-lines
match-line sense amplifiers
LSL
lower local block
match-line sense amplifiers

Fig. 4. 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.

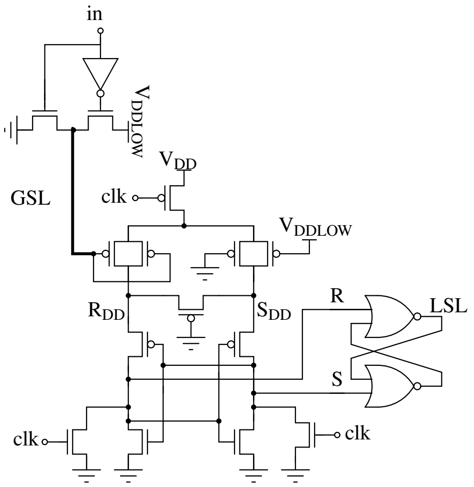

Fig. 5. Low-swing receiver.

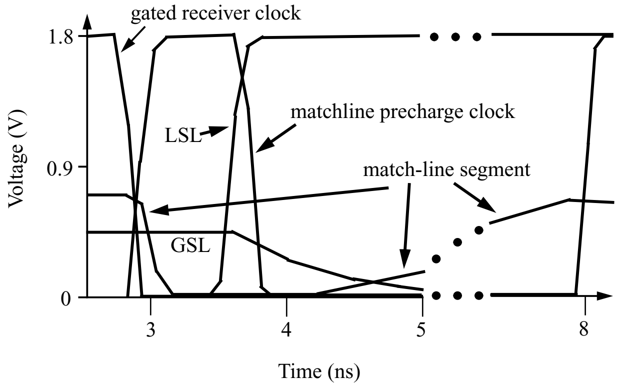

Fig. 6. Signal waveforms for a full search operation. The cycle time is dominated by the charging of the match-line segments. A portion of the cycle time, showing the rising match-line, has been removed to simplify the figure.

clock and the receiver clock. This scheme adds a pipeline cycle to the search-line path, allowing a full clock period for the low-swing data to traverse the GSLs and thus removing the low-swing signals from the critical path. The GSL clock is delayed to prevent the low-swing data from causing a hold timing violation at the low-swing receiver. The global clock, inverted and delayed, is fed into the low-swing receivers through the ANDs that gate the clock. We delay the receiver clock to ensure that the clock gating signal feeding the ANDs from the flip-flop is stable when the receiver clock arrives at the AND gate. For the purpose of the testchip this delay is implemented by inverters, however, in a timing-critical system the inverters can be replaced by a replica of the clock-to-output path of the flip-flop.

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

$$E_{conv.} = C_{SL} V_{DD}^2 \ , \qquad (1)$$

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

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

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, $C_{GSL} \approx C_{SL}/6$ (discussed later in Section V). $E_{overhead}$ 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 $\alpha = 20\%$, $E_{hier.} = 0.37 E_{conv.}$, indicating a 63% power reduction.

Fig. 5 shows the schematic for a low-swing receiver [8]. The receiver is an edge triggered sense amplifier flip-flop 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 $V_{DDLOW}/2$. This eliminates the requirement of explicitly generating a voltage of $V_{DDLOW}/2$. 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.

IV. Simulation Results

The proposed architecture has been targeted for a 1024 × 144 bit ternary CAM macro, although due to fabrication resource constraints, we have designed a 256 × 144 bit testchip for fabrication. The simulation results presented in this section include the effect of parasitics extracted from layout. All subcircuits were simulated in HSPICE for timing verification, and power consumptions were measured by simulating the complete CAM macro netlist in Nanosim [9], a transistor-level simulator capable of handling large netlists.

Fig. 6 displays the simulated signal waveforms for a single-cycle search operation in a 34-bit segment of the CAM, which uses an SRAM core cell and NOR-based match-line architecture. The cycle begins by clocking the low-swing receiver (negative triggered) to sense the current cycle's GSL data and by clocking the GSL flip-flop (top of Fig. 4) to activate the next cycle's GSL data. The receiver drives the corresponding LSL, and when the LSL's traversal is completed, the match-line sensing circuitry is activated. We control the match-line sensing activation 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. The voltage on the match-lines rise in a race to reach the threshold voltage of an NMOS transistor [4]. The match-line with the fastest rate of rise wins the race. This scheme achieves a search cycle time of 7 ns.

V. Discussion

In this section, we explore means of reducing power consumption by referring to equation (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_{GSL}, C_{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 to 32. By choosing $N = 16$, we have also been able to reduce the area overhead to 6%. This area overhead is caused by the number of low-swing receivers (also proportional to N). Second, by choosing to route the LSLs in metal2 and the GSLs in metal4, we reduce C_{GSL} (without affecting any other parameter in equation(2)).

Table I summarizes the features of the testchip, with the layout shown in Fig. 7. Indicated on the plot are a match-line segment, a GSL, two LSLs, a local block, and a low-swing receiver block. The peripheral test circuitry consists of flip-flops for shifting the input and output data.

VI. Conclusion

This paper presents a CAM architecture with pipelined match-lines and hierarchical search-lines that reduces power consumption. We have compared the energy dissipation per search operation in units of fJ/bit/search for three architectures. The conventional architecture dissipates 7.18 fJ/bit/search. Incorporating pipelined match-lines (with conventional search-lines) reduces the match-line energy dissipation by 56% resulting in an overall energy dissipation of 5.13 fJ/bit/search. Finally, hierarchical search-lines in concert with pipelined match-lines, reduce search-line energy dissipation by 60% with an overall energy dissipation of 2.89 fJ/bit/search.

Acknowledgment

The authors thank Igor Arsovski for insightful discussions on this work. Also, the authors gratefully acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] T.-B. Pei and C. Zukowski, "Putting routing tables in silicon," *IEEE Network Magazine*, vol. 6, no. 1, pp. 42–50, January 1992.

[2] L. Chisvin and R. J. Duckworth, "Content-addressable and associative memory: Alternatives to the ubiquitous RAM," *IEEE Computer*, vol. 22, no. 7, pp. 51–64, July 1989.

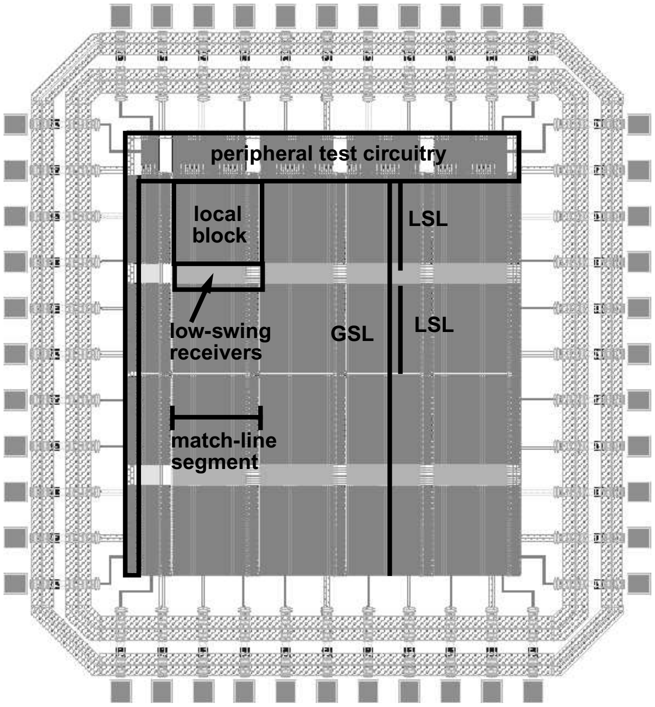

Fig. 7. Layout plot of the testchip.

TABLE I

Chip characteristics (Simulation results).

Technology	0.18 µm, 6-metal
Organization	256 × 144 ternary bits
Chip Size	2.3 mm × 2.1 mm
Supply Voltage	1.8 V
Cycle Time	7 ns (142 MSearch/sec)
Power Consumption	94 mW @ 142 MSearch/sec

[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, no. 6, 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, no. 1, pp. 155–158, January 2003.

[5] I. Arsovski and A. Sheikholeslami, "A current-saving match-line sensing scheme for content-addressable memories," in *IEEE International Solid-State Circuits Conference Technical Digest*, 2003, pp. 304–305,494.

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

[7] 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, no. 11, pp. 1690–1696, November 1998.

[8] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: Effectiveness and robustness," *IEEE Trans. VLSI Syst.*, vol. 8, no. 3, pp. 264–272, June 2000.

[9] *NanoSim Reference Guide*, Synopsys, March 2002.