ISLAMIC UNIVERSITY OF TECHNOLOGY, (IUT)

Content-Addressable Memory (CAM) circuits and Performance comparison of positive feedback matchline sensing circuits

By

Nowab Reza MD Ashif (092446)

Asif Newaz Khan (092450)

Miraz Ahmed (092452)

A Dissertation

Submitted in Partial Fulfillment of the Requirement for the Bachelor of **Science in Electrical and Electronic Engineering**

Academic Year:2012-2013

Department of Electrical and Electronic Engineering.

Islamic University of Technology (IUT) A Subsidiary Organ of OIC.

Dhaka, Bangladesh.

A Dissertation on,

Content-Addressable Memory (CAM) circuits and Performance comparison of positive feedback matchline sensing circuits

Submitted By

-------------------------- ------------------------- -----------------------

Nowab Reza MD Ashif Asif Newaz Khan Miraz Ahmed

Approved By

____

Dr. Md. Shahid Ullah

Professor & Head of the

Department of EEE, IUT.

____

Syed Iftekhar Ali

Thesis Supervisor

Assistant Professor

Department of EEE, IUT

Abstract

CAM is used to implement lookup-table function in a single clock cycle using dedicated comparison circuitry. CAMs are especially popular in network routers for packet forwarding and classification, but they are also beneficial in a variety of other applications that require high-speed table lookup. The main CAM design challenge is to reduce power consumption associated with the large amount of parallel active circuitry, without sacrificing speed or memory density. Ternary content addressable memories (TCAMs) are hardware based parallel lookup tables with bit-level masking capability. Ternary content-addressable memories (TCAMs) can perform high-speed and deterministic table lookups. However, its main drawback is high power consumption. A significant portion of the TCAM power is consumed by matchline sense amplifiers (MLSAs) for match detection. This work compares circuit techniques for reducing TCAM power consumption. The ML sensing energy is reduced by employing positive-feedback ML sense amplifiers (MLSAs).

Table of Contents

Table of Figures

List of Tables

Chapter 1

Introduction to CAM

1.1 Introduction

Recently there has been a phenomenal increase in the number of internet users and use of real-time applications which have resulted in a demand for very high speed networks. The internet contains routers and switches, which process data packets and forwards them towards their destinations. An IP packet consists of a header and a payload. The header contains information such as source address and destination address etc. A switch transfers an incoming packet to an output port based on the header of the packet. A router on the other hand is a bit complex; it forwards an incoming packet after routing its path from source to destination [4]. Each router maintains a routing table and packets are forwarded based on the information in this routing table.

One of the most time consuming process during packet forwarding is table lookup [4]. Software methods of table lookup such as radix trees are relatively slow and they do not scale very well with the table size. Under good conditions, a hash function can perform the lookup in one memory access. However, its worst-case search time, which depends on the table size and the hashing function, can be considerably worse than the tree searches [4]. Therefore, many of the table lookup tasks at different network layers that were originally implemented in software are now being replaced by hardware solutions to meet the performance requirements. An efficient hardware solution to perform table lookup is the content addressable memory (CAM) [4].

CAM is an outgrowth of random access memory (RAM) [4]. In addition to the conventional READ and WRITE operations, CAMs also support SEARCH operations. A CAM stores a number of data words and compares a search key with all the stored entries in parallel. If a match is found, the corresponding memory location is retrieved. In the presence of multiple matches, a priority encoder (PE) resolves the highest priority match [1]. An example of a simplified routing table is displayed in Table 1 [1]. All four entries in the table are 5-bit words, with the don't care bit, "X", matching both a 0 and a 1 in that position. Because of the "X" bits, the first three entries in the Table represent a range of input addresses, i.e., entry 1 maps all addresses in the range 10100 to 10111 to port A. The router searches this table for the destination address of each incoming packet, and selects the appropriate output port. For example, if the router receives a packet with the destination address 10100, the packet is forwarded to port A. In the case of the incoming address 01101, the address lookup matches both entry 2 and entry 3 in the table. Entry 2 is selected since it has the fewest "X" bits, or, alternatively, it has the longest prefix, indicating that it is the most direct route to the destination [1]. This lookup method is called longest-prefix matching. [1]

Entry no.	Address (binary)	Output Port
1	101XX	A
2	0110X	B
3	011XX	C
4	10011	D

Table 1 CAM-based implementation of the routing table [1]

Fig.1 illustrates how a CAM accomplishes address lookup by implementing the routing table shown in Table 1 [1]. On the left of Fig. 1, the packet destination-address of 01101 is the input to the CAM. As in the table, two locations match, with the (priority) encoder choosing the upper entry and generating the match location 01, which corresponds to the most-direct route. This match location is the input address to a RAM that contains a list of output ports, as depicted in Fig. 1. A RAM read operation outputs the port designation; port B, to which the incoming packet is forwarded. We can view the match location output of the CAM as a pointer that retrieves the associated word from the RAM. In the particular case of packet forwarding the associated word is the designation of the output port. This CAM/RAM system is a complete implementation of an address lookup engine for packet forwarding [1].

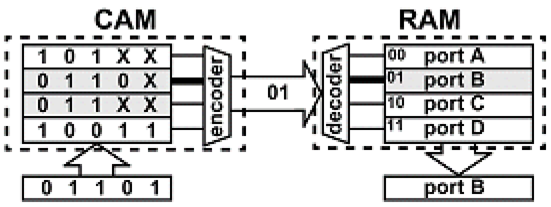

Figure 1 CAM-based implementation of the routing table [1]

CAM-based table lookup is very fast due to the parallel nature of the SEARCH operation [4]. However, the speed of a CAM comes at the cost of increased silicon area and power consumption, two design parameters that designers struggle to reduce [1]. As CAM applications grow, demanding larger CAM sizes, the power problem is further exacerbated. Reducing power consumption, without sacrificing speed or area, is the main thread of recent research in large capacity CAMs [1].

1.2 CAM Basics

A small model is shown in Fig. 2. The figure shows a CAM consisting of 4 words, with each word containing 3 bits. There is a match-line corresponding to each word (ML_0, ML_1 etc) feeding into matchline sense amplifiers (MLSAs), and there is a differential searchline pair corresponding to each bit of the search word ($SL_0, \overline{sL_0}$, SL_1, $\overline{sL_1}$ etc.) [1]. A CAM search operation begins with loading the search-data word into the search-data registers followed by precharging all matchlines high, putting them all temporarily in the match state. Next, the searchline drivers broadcast the search word onto the searchlines, and each CAM cell compares its stored bit against the bit on its corresponding searchlines. Matchlines on which all bits match remain in the precharged-high state. Matchlines that have at least one bit that misses, discharge to ground. The MLSA then detects whether its matchline has a matching condition or miss condition. Finally, the encoder maps the matchline of the matching location to its encoded address [1].

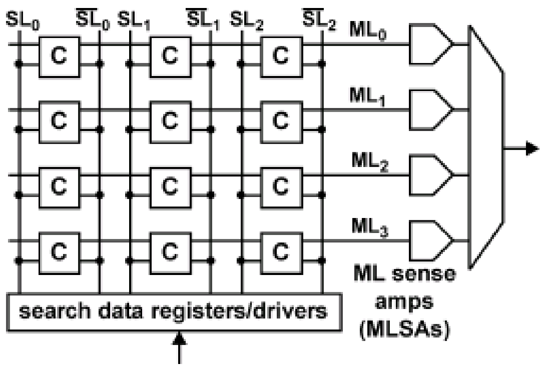

Figure 2 Simple CAM model with 4 words having 3 bits each [1]

1.3 Binary CAM Cells

A CAM cell basically has two functions: bit storage and bit comparison. There are two types of binary CAM cells: Fig. 3(a) shows a NOR type CAM cell, Fig. 3(b) shows a NAND type CAM cell [1]. The bit storage in both cases is an SRAM cell where cross-coupled inverters implement the bit storage nodes D and $\bar{D}$. For simplicity the nMOS access transistors and bitlines which are used to read and write the SRAM have not been shown in Fig. 3.

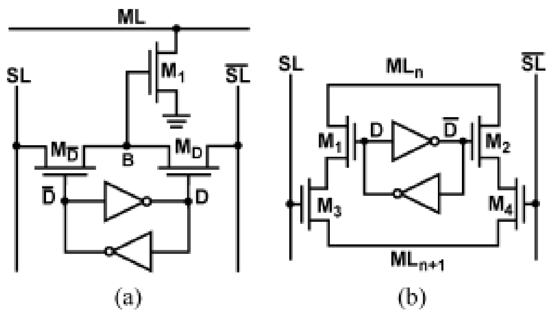

Figure 3 CAM core cells variations: (a) NAND type CAM (b) NOR type CAM [1]

1.3.1 <u>NOR Cell</u>

The NOR cell implements the comparison between the complementary stored bit, D (and D), and the complementary search data on the complementary searchline, SL (and SL), using four comparison transistors, which are all typically minimum-size to maintain high cell density. [1] Each pair of transistors M1/M3 or M2/M4 will form a pull down path for the matchline, ML, when SL and D do not match. A match of SL and D disables the pull down paths and thus disconnects the matchline, ML from ground. The NOR nature of this cell becomes clear when multiple cells are connected in parallel to form a CAM word by shorting the ML of each cell to the ML of adjacent cells [1]. The pull down paths connect in parallel resembling the pull down path of a CMOS NOR logic gate.

1.3.2 <u>NAND Cell</u>

The NAND cell performs the comparison operation between SL and D using three comparison transistors M_1, M_D and $\bar{M}_D$, which are typically minimum size to maintain high cell density. [1] When there is a match between SL and D, and both SL and D are 1 then M_D turns on passing logic 1 to node B if both SL and D are logic 0 then $\bar{M}_D$ passes logic level 1 to node B indicating a match. The NAND nature of this cell becomes clear $\bar{M}_D$ when multiple NAND cells are serially connected. [1]

1.4 <u>Ternary CAM Cells</u>

A 16-T TCAM cell is shown in Fig. 4. TCAM cells are similar to binary CAM cells except that TCAM cells have two SRAMs whereas Binary CAM cells have only one SRAM. The advantage of using TCAMs is that masking can be done, i.e. it can store ternary values. Masking is done in this type of circuits by turning off both ML to GND paths by storing $V_x = V_y = 0$ (Local Masking) or by putting $SL_1 = SL_2 = 0$ (Global masking). [4]

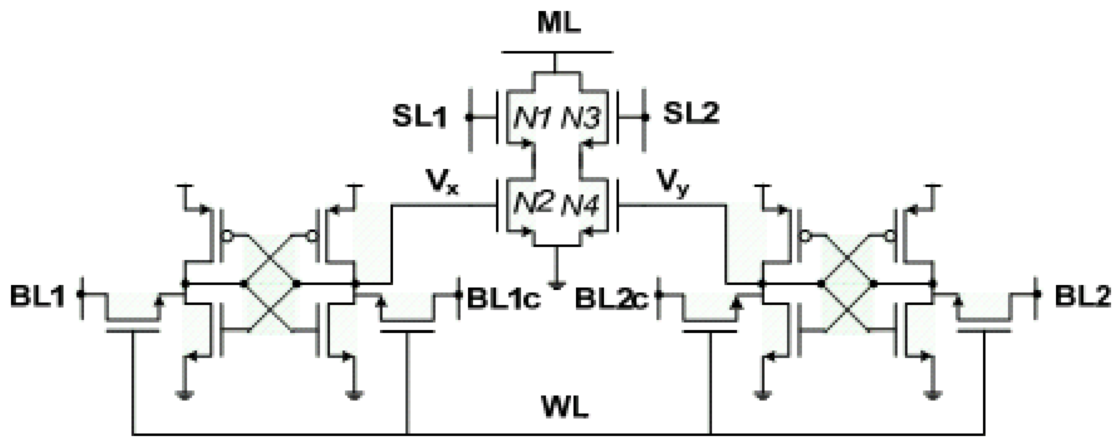

Figure 4 16-T Static TCAM cell [4]

1.5 <u>Write Operation</u>

The WRITE operation is performed by placing the data on the bit lines (BLs) and enabling the word line (WL). This turns on the access transistors (N6-N7), and the internal nodes of the cross-coupled inverters are written by the BL data. Fig. 5 shows the WRITE operation when ‘0’ is being written to a cell which originally stored ‘1’. Originally, V_x = ‘1’ and V_y = ‘0’, P1 and N9 were ‘ON’, and P2 and N8 were ‘OFF’. When WL is enabled (WL = ‘1’), access transistors (N6-N7) conduct resulting in BL currents I_0 and I_1 (shown by dashed arrows in Figure 4). These transient currents form voltage dividers (P1-N6 and N7-N9). If these transient currents can pull one of the nodes (V_x and V_y) to the inverter threshold voltage, the other node will flip due to the feedback action of the cross-coupled inverters. [4] The same procedure is followed for TCAM cells.

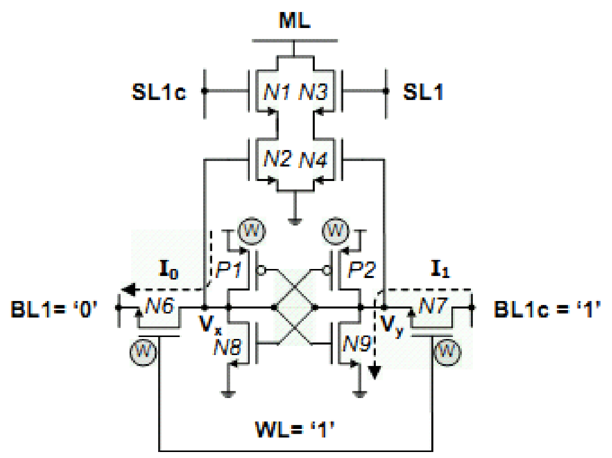

Figure 5 WRITE operation in a conventional 10T binary CAM cell [4]

1.6 Search Operation

The conventional SEARCH operation is performed in three steps. [4] First, search lines (SLs) SL1 and SL1c are reset to GND. Second, ML is pre-charged to VDD. Finally, the search key bit and its complementary value are placed on SL1 and SL1c, respectively. If the search key bit is identical to the stored value (SL1=BL1, SL1c=BL1c), both ML-to-GND pull-down paths remain 'OFF', and the ML remains at VDD indicating a "match". Otherwise one of the pull-down paths conducts and discharges the ML to GND indicating a "mismatch". Resetting SL1 and SL1c to GND during the ML pre-charge phase ensures that both pull-down paths are 'OFF', and hence do not conflict with the ML pre-charging. Fig. 6 shows the SEARCH operation when 0 is stored in the cell (Vx = 0 and Vy = 1). For SL1 = 1 (SL1c = 0), ML is discharged to 0 detecting "mismatch" as shown in Figure 6(a). Similarly for SL1 = 0, ML remains at 1 detecting "match". The same procedure is followed for TCAM cells.

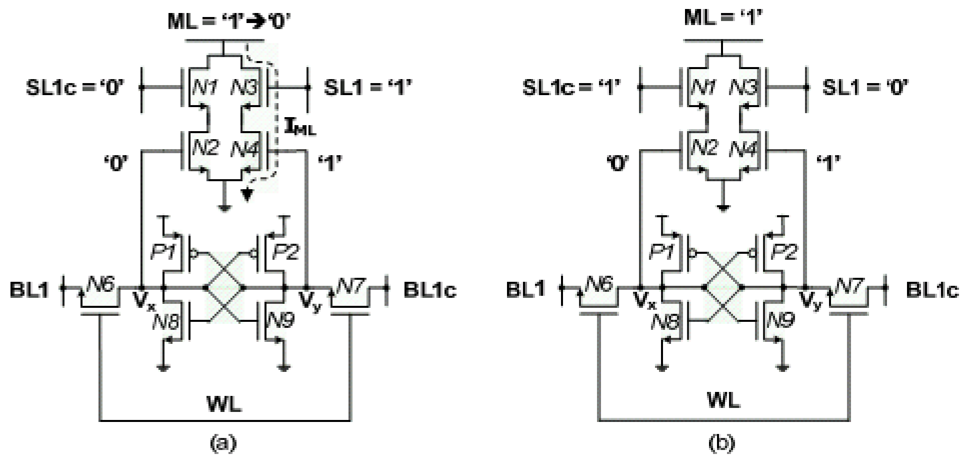

Figure 6 SEARCH operation in a 10T binary CAM cell for (a) "mismatch", and (b) "match" [4]

Chapter 2
Methods of reducing power consumption

As discussed at the beginning of chapter one that CAM based table lookup is very fast due to its parallel nature but the speed comes with the sacrifice of increased silicon area and power consumption. So basically the main concern is to reduce power consumption and keep the silicon area as small as possible. In order to reduce power consumption various matchline sensing schemes, searchline driving approaches and CAM architectures have been developed. This chapter describes some of this energy saving scheme.

2.1 Matchline Sensing Schemes:

The matchline is one of the two key structures in a CAM where power consumption is the highest. In this section some matchline sensing schemes have been discussed later a few search line driving approaches and CAM architectures have been discussed.

2.1.1 Conventional (Precharge-High) Matchline Sensing:

In this matchline sensing scheme the matchline is first precharged to V_{DD} and then evaluation is carried out by allowing the pull down transistors to pull down the matchlines to GND in the case of a miss, or leave the matchline high in the case of a match. Fig. 7(a) shows, in schematic form, an implementation of this matchline-sensing scheme. Fig. 7(b) shows the signal timing which is divided into three phases: SL precharge, ML precharge, and ML evaluation. [1] The operation begins by asserting slpre to precharge the searchlines low, disabling all the pull down paths. With the pull down paths disconnected, the operation continues by asserting $\overline{\text{mlpre}}$ to precharge the matchline high. Once the match-line is high, both slpre and $\overline{\text{mlpre}}$ are de-asserted. Then the ML evaluation phase begins by placing the search word on the searchlines. If there is at least one single-bit miss on the matchline, a path (or multiple paths) to ground will discharge the matchline, ML, indicating a miss for the entire word, which is output on the MLSA sense-output node, called MLSO. If all bits on the matchline match, thematchline will remain high indicating a match for the entire word. [1]

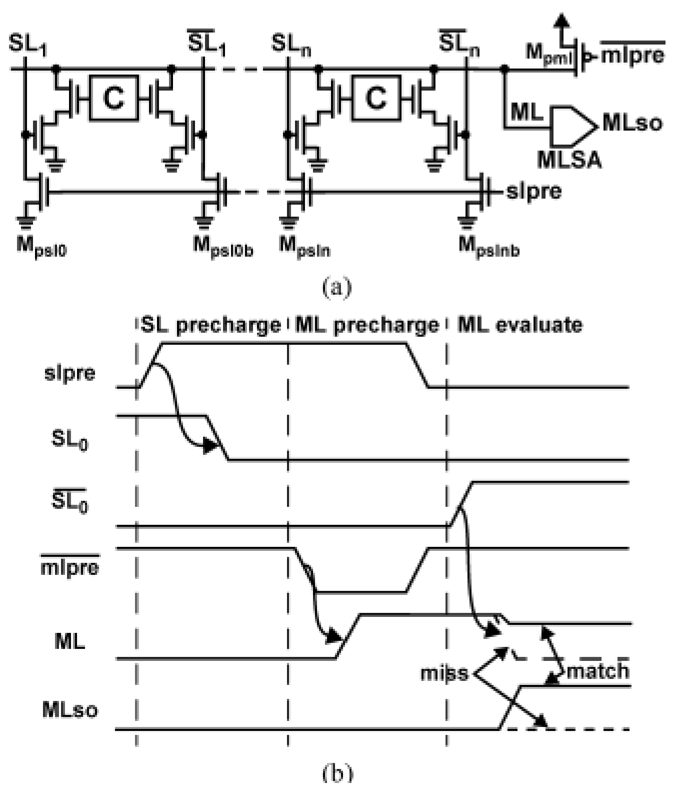

Figure 7 (a) conventional matchline sensing scheme, and (b) timing diagram [1]

Using the simple matchline model, we can find the time required to precharge and evaluate the matchline. [1] The time to precharge the matchline (which we define as the 10% to 90% risetime of ML) through the precharge device M_{pml} of Fig. 7(a) is given by

$$t_{MLpre} = 2.2T_{MLpre} \tag{1}$$

$$= 2.2R_{EQpre}C_{ML} \tag{2}$$

here R_{EQpre} is the equivalent resistance of the precharge transistor.

The time to evaluate the matchline depends on the matchline capacitance and the matchline pulldown resistance. [1] The worst case matchline pulldown resistance occurs when only a single bit misses, activating only a single pulldown path. Referring to Fig. 8, for a single miss,m=1 , and the time for the evaluation,which we define as the time for the matchline to fall to 50% of the precharge voltage, is given by

$$t_{ML-eval} = 0.69 T_{ML-eval} \quad (3)$$

$$= 0.69 R_{ML} C_{ML} \quad (4)$$

Since, typically, minimum-sized devices are used in the cell the pulldown resistance is in the range of 5–10 kΩ in a 0.18- µm CMOS technology. [1] The capacitance,C_{ML} , depends on the number of bits on the matchline, but can be as high as a few hundred fF in a typical 0.18-µm CMOS technology. [1]

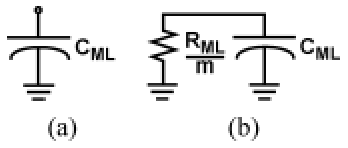

Figure 8 Matchline circuit model for (a) the match state and (b) the miss state. [1]

The dynamic power consumed by a single matchline that misses is due to the rising edge during precharge and the falling edge during evaluation, and is given by the equation

$$P_{miss} = C_{ML} V_{DD}^{2} f \quad (5)$$

here f is the frequency of search operations. In the case of a match, the power consumption associated with a single matchline depends on the previous state of the matchline; however,since typically there is only a small number of matches we can neglect this power consumption. [1] Accordingly, the overall matchline power consumption of a CAM block with w matchlines is

$$P_{ML} = w P_{miss} \quad (6)$$

$$= w C_{ML} V_{DD}^{2} f \quad (7)$$

2.1.2 <u>Low-Swing Schemes</u>

One method of reducing the ML power consumption, and potentially increasing its speed is to reduce the ML voltage swing. [1] The reduction of power consumption is linearly proportional to the reduction of the voltage swing. The resulting power equation is-

$$P_{ML} = w C_{ML} V_{DD} V_{MLswing} f \quad (8)$$

where $V_{MLswing}$ is the voltage swing of the ML. The main challenge addressed by various low-swing implementations is using a low-swing voltage without resorting to an externally generated reference voltage. [1]

Fig. 9 is a simplified schematic of the matchline sensing scheme, which saves power by reducing swing. The matchline swing is reduced from the full supply swing to V_{LOW} =300 mV. Precharge to 300 mV is accomplished by associating a tank capacitor,C_{tank}, with every matchline. [1] The recharge operation (assertion of pre signal) charges the tank capacitor to V_{DD} and then uses charge sharing (enabled by eval signal) to dump the charge onto the matchline. The matchline precharge voltage is given by

$$V_{MLpre} = V_{DD} \frac{C_{tank}}{C_{ML} + C_{tank}} \qquad (9)$$

and is set to be equal to 300 mV in the design. [1] In the case of a miss, the matchline discharges through the CAM cell(s) to ground, whereas in the case of match, the matchline remains at the precharge level. Matchline evaluation uses a sense amplifier that employs transistor ratios to generate a reference level of about $V_{LOW}/2$=150 mV. [1] This sense amplifier is shown in generic form in the figure.

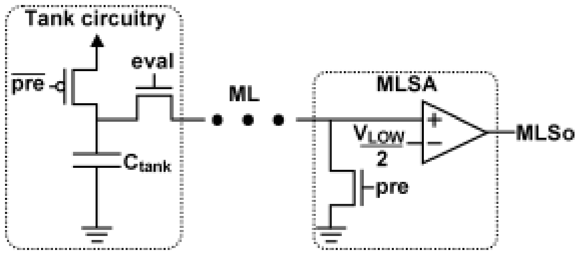

Figure 9 Low-swing matchline sensing scheme. [1]

2.1.3 Selective Precharge Scheme:

Selective precharge allocate power to matchlines nonuniformly. It performs a match operation on the first few bits of a word before activating the search of the remaining bits . For example, in a 144-bit word, selective precharge initially searches only the first 3 bits and then searches the remaining 141 bits only for words that matched in the first 3 bits. [1] Assuming a uniform random data distribution, the initial 3-bit search should allow only 1\2words to survive to the second stage saving about 88% of the matchline power. [1] In practice, there are two sources of overhead that limit the power saving. [1] First, to maintain speed, the initial match implementation may draw a higher power per bit than the search operation on the remaining bits. Second, an application may have a data distribution that is not uniform, and, in the worst-case scenario, the initial match bits are identical among all words in the CAM, eliminating any power saving. [1]

Fig. 10 is a simplified schematic of an example of selective precharge scheme. The example uses the first bit for the initial search and the remaining (n-1) bits for the remaining search. To maintain speed, the implementation modifies the precharge part of the precharge-high scheme. The ML is precharged through the transistor M_1, which is controlled by the NAND CAM cell and turned on only if there is a match in the first CAM bit. The remaining cells are NOR cells. Note that the ML of the NOR cells must be pre-discharged to ground to maintain correct operation in the case that the previous search left the matchline high due to a match. Thus, one implementation of selective precharge is to use this mixed NAND/NOR matchline structure. Selective precharge is perhaps the most common method used to save power on matchlines since it is both simple to implement and can reduce power by a large amount in many CAM applications.

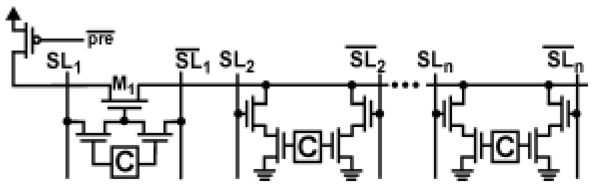

Figure 10 Sample implementation of the selective-precharge matchline technique. [1]

2.1.4 Pipelining Scheme

In selective precharge, the matchline is divided into two segments. More generally, an implementation may divide the matchline into any number of segments, where a match in a given segment results in a search operation in the next segment but a miss terminates the match operation for that word. A design that uses multiple matchline segments in a pipelined fashion is the pipelined matchlines scheme. [1] Fig11(a) shows a simplified schematic of a conventional NOR matchline structure where all cells are connected in parallel. Fig. 11(b) shows the same set of cells as in Fig. 11(a), but with the matchline broken into four matchline segments that are serially evaluated. If any stage misses, the subsequent stages are shut off, resulting in power saving. [1]

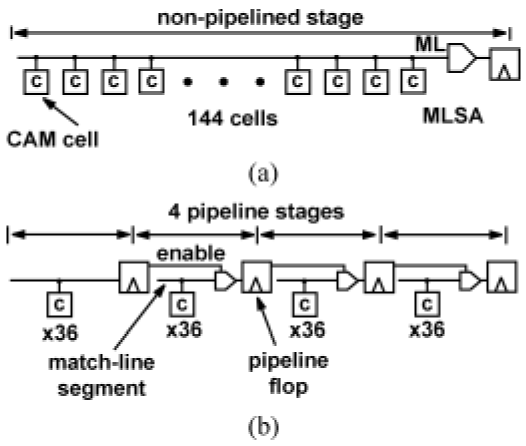

Figure 11 Pipelined matchlines reduce power by shutting down after a miss in a stage. [1]

The drawbacks of this scheme are the increased latency and the area overhead due to the pipeline stages. [1] By itself, a pipelined matchline scheme is not as compelling as basic selective precharge; however, pipelining enables the use of hierarchical searchlines, thus saving power. [1]

Another approach is to segment the matchline so that each individual bit forms a segment . [1] Thus, selective precharge operates on a bit-by-bit basis. In this design, the CAM cell is modified so that the match evaluation ripples through each CAM cell. If at any cell there is a miss, the subsequent cells do not activate, as there is no need for a comparison operation. The drawback of this scheme is the extra circuitry required at each cell to gate the comparisonwith the result from the previous cell. [1]

2.1.5 <u>Current-Saving Scheme:</u>

Current-saving scheme is the modified version of current-race scheme which we will be discussed in chapter three. Current-race scheme uses the same current on each matchline, regardless of whether it has a match or a miss. [1] The key improvement of the current-saving scheme is to allocate a different amount of current for a match than for a miss. In the current-saving scheme,matches are allocated a larger current and misses are allocated a lower current. Since almost every matchline has a miss, overall the scheme saves power. [1]

The main difference from the current-race scheme is the addition of the current-control block. [1] This block is the mechanism by which a different amount of current is allocated, based on a match or a miss. The input to this current-control block is the matchline voltage,V_{ML} , and the output is a control voltage that determines the current,I_{ML} , which charges the matchline. The current-control block provides positive feedback since higher V_{ML} results in higher I_{ML} , which, in turn, results in higher V_{ML} .

Recall that matchlines in the match state are purely capacitive [see Fig. 8(a)], whereas matchlines in the miss state have both capacitance and resistance [see Fig. 8(b)]. In this scheme, the matchline is precharged low, just as in the current-race scheme. In the current-race scheme, the current source provides a constant amount of current regardless of the state of the matchline. In the current-saving scheme, the amount of current is initially the same for all matchlines, but the current control reduces the current provided to matchlines that miss (have a resistance to ground), but maintains the current to matchlines that match(with no resistance to ground). The current-control block increases the current as the voltage on the ML rises and the voltage on the ML rises faster for large ML resistance. Recall that the ML resistance depends on the number of bits that miss. Since the amount of current decreases with the number of misses, it follows that the power dissipated on the matchline also depends on the number of bits that miss. This scheme saves over 50% of matchline power compared to the current-race scheme.

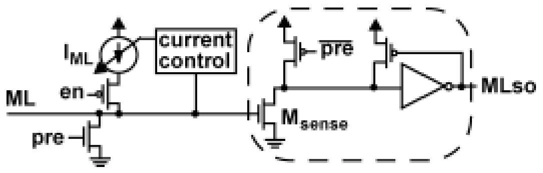

Figure 12 Current-saving matchline-sensing scheme. [1]

2.2 <u>Searchline Driving Approaches:</u>

Searchline power consumption depends partly on the matchline scheme. In this section, we look at the power consumption of the searchlines for three cases: matchline precharge high, matchline precharge low and pipelined matchlines with hierarchical searchlines.

2.2.1 <u>Conventional approach:</u>

The conventional approach to driving the searchlines applies to matchline schemes that precharge the matchlines high. In this approach, during the search cycle, the searchlines are driven by a cascade of inverters first to their precharge level and then to their data value. The searchline power consumption depends on the searchline capacitance, which consists of wire capacitance and one transistor gate capacitance per row. The equation for the dynamic power consumption of the searchlines is

$$P_{SL} = 2n * \frac{1}{2} * C_{SL} * V_{DD}^{2} * f \qquad (10)$$

$$= nC_{SL} * V_{DD}^{2} * f \qquad (11)$$

Where C_{SL} is the total capacitance of a single searchline n, is the total number of searchline pairs (and 2n is the total number of searchlines), and V_{DD} is the power supply voltage. To this power, we must add the power consumption of the drivers.

2.2.2 <u>Eliminating Searchline precharge:</u>

We can reduce searchline power by eliminating the SL precharge phase depicted in Fig.7b. Eliminating the SL precharge phase reduces the toggling of the searchlines, thus reducing power. As discussed earlier, matchline-sensing schemes that precharge the matchline low eliminate the need for SL precharge. These schemes directly activate the searchlines with their search data without going through an SL precharge phase. [1] The equation for the reduced power in this case is

$$P_{SL} = \frac{1}{2} * n * C_{SL} * V_{DD}^{2} * f \tag{12}$$

This equation shows that matchline-sensing schemes that precharge the matchlines low also save power on the searchlines. In fact, in these precharge-low schemes, the reduction in searchline power can be as large as, or even larger than, the reduction in matchline power.

2.2.3 <u>Hierarchical Searchlines:</u>

Another method of saving searchline power is to shut off some searchlines (when possible) by using the hierarchical searchline scheme. [1] Hierarchical searchlines are built on top of pipelined matchlines. The basic idea of hierarchical searchlines is to exploit the fact that few matchlines survive the first segment of the pipelined matchlines. With the conventional searchline approach, even though only a small number of matchlines survive the first segment, all searchlines are still driven. Instead of this, the hierarchical searchline scheme divides the searchlines into a two-level hierarchy of global searchlines (GSLs) and local searchlines (LSLs). [1] Fig. 13 shows a simplified hierarchical searchline scheme, where the matchlines are pipelined into two segments, and the searchlines are divided into four LSLs per GSL. The overall power consumption on the searchlines is

$$P_{SL} = \left(\underbrace{C_{GSL} * V_{DD}^{2}}_{GSL} + \underbrace{\alpha C_{LSL} * V_{DD}^{2}}_{LSL} \right) f \tag{13}$$

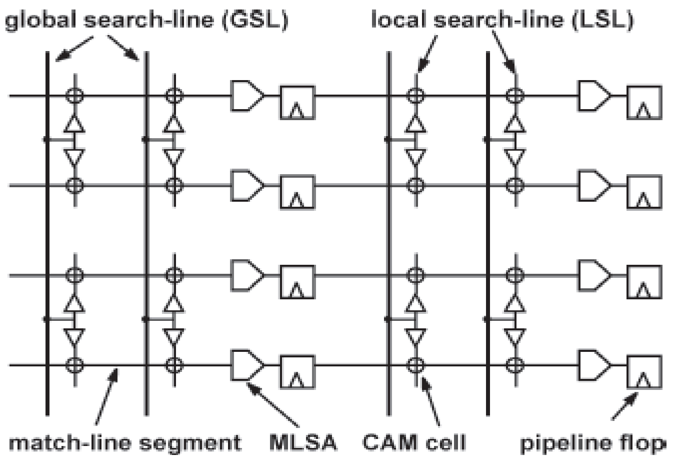

Figure 13: Hierarchical searchline structure [1]

where C_{GSL} is the GSL capacitance, C_{LSL} is the LSL capacitance (of all LSLs connected to a GSL) and α is the activity rate of the LSLs. C_{GSL} primarily consists of wiring capacitance, whereas C_{LSL} consists of wiring capacitance and the gate capacitance of the SL inputs of the CAM cells. The factor α, which can be as low as 25% in some cases, is determined by the search data and the data stored in the CAM. [1] We see from (13) that α determines how much power is saved on the LSLs, but the cost of this savings is the power dissipated by the GSLs. Thus, the power dissipated by the GSLs must be sufficiently small so that overall searchline power is lower than that using the conventional approach. If wiring capacitance is small compared to the parasitic transistor capacitance, then the scheme saves power. [1] However, as transistor dimensions scale down, it is expected that wiring capacitance will increase relative to transistor parasitic capacitance. [1] In the situation where wiring capacitance is comparable or larger than the parasitic transistor capacitance, C_{GSL} and C_{LSL} will be similar in size, resulting in no power savings. [1] In this case, small-swing signaling on the GSLs can reduce the power of the GSLs compared to that of the full-swing LSLs. [1] This results in the modified searchline power of

$$P_{SL} = 2n\left(C_{GSL} * V_{LOW}^{2} + \alpha C_{LSL} * V_{DD}^{2}\right) f \qquad (14)$$

Where V_{LOW} is the low-swing voltage on the GSLs (assuming an externally available power supply V_{LOW})

This scheme requires an amplifier to convert the low-swing GSL signal to the full-swing signals on the LSLs. Fortunately; there are only a small number of these amplifiers per searchline, so that the area and power overhead of this extra circuitry is small. [1]

2.3 <u>Power Saving CAM Architectures:</u>

This section discusses about mainly two power saving architectures for large-capacity CAMs. First, a straightforward architectural technique that saves power is bank-selection, where only a portion of the CAM is active on any given cycle. The main purpose of the bank selection scheme was to reduce area but it has been used to save power instead. [1] The basic concept is that bank selection divides the CAM into smaller portions called banks. Fig. 14 provides a block diagram of a simplified bank-selection scheme.

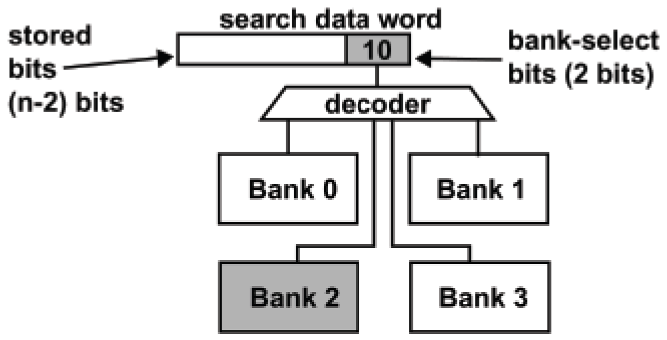

Figure 14: Simplified diagram of bank selection scheme. [1]

Two extra data bits, called bank-select bits, partition the CAM into four blocks. When storing data, the bank-select bits determine into which of the four blocks to store the data. When searching data, the bank-select bits determine which one of the four blocks to activate and search. The decoder accomplishes the selection by providing enable signals to each block. For example in the figure, the bank-select bits are 10 which selects bank 2. Since, only one of four blocks is active at any time, only 1/4 of the comparison circuitry is necessary compared to the case with no bank selection, thus saving area. [1] Instead of saving area, recent bank selection schemes aim to reduce power. [1] Bank selection reduces overall power consumption in proportion to the number of blocks. The major drawback of bank selection is the problem of bank overflow. Since, in a CAM, there are many more input combinations than storage locations, the storage of a bank can quickly overflow. [1] The overflow condition requires extra circuitry and forces multiple banks to be activated at once, decreasing the savings in power. To avoid overflow, an external mechanism can balance the data in the banks by periodically re-partitioning the banks.
The second architectural technique for saving power, which applies only to binary CAM, is pre-computation.

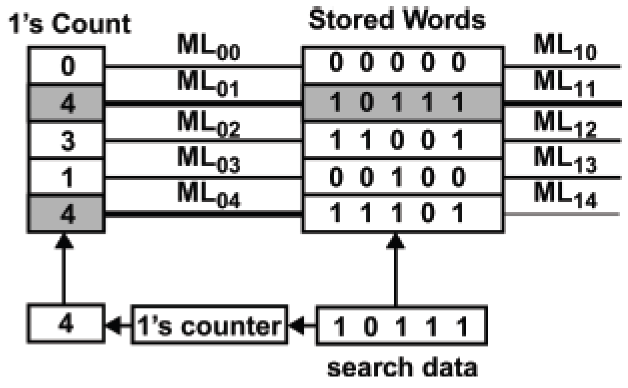

Figure 15 Conceptual View of Pre-Computation –based CAM. [1]

Pre-computation stores some extra information along with each word that is used in the search operation to save power. These extra bits are derived from the stored word, and used in an initial search before searching the main word. If this initial search fails, then the CAM aborts the subsequent search, thus saving power. This is the same mechanism that we have already seen used in the selective-precharge technique; however, pre-computation has a second purpose, which is to save CAM cell area. [1] Fig. 15 shows a conceptual view of the operation of a pre-computation-based CAM (PB-CAM). The extra information holds the number of ones in the stored word. For example, in Fig. 15, when searching for the data word, 10111, the pre-computation circuit counts the number of ones (which is four in this case). The number four is compared on the left-hand side to the stored 1's counts. Only matchlines ML01and ML_{04} match, since only they have a 1's count of four. In the stored-words stage in Fig. 15, only two comparisons actively consume power and only matchline ML_{11} results in a match. The compelling part of PB-CAM results from exploiting the 1's count to simplify the comparison circuitry in the second stage.

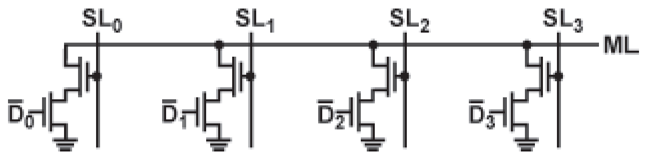

Figure 16: Schematic View of matchline structure of pre-computation based CAM. [1]

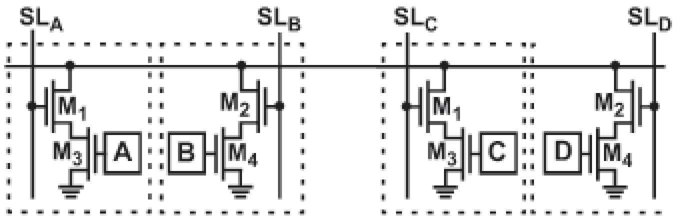

Figure 17 Two ternary CAM cells are viewed as four independent half cells in the dense encoding scheme. [1]

Chapter 3

Simulation of some matchline sensing schemes

A significant portion of the TCAM power is consumed by MLSAs for match detection; for these reason a few matchline sensing schemes have been simulated and compared. This chapter discusses about the operation of current race scheme and a few modifications of the current race scheme. The simulated results (voltage and current waveforms) for all the schemes are provided in the chapter. All the simulations have been done with 0.18-μm CMOS technology using HSPICE.

3.1 Current race scheme:

As shown in fig. 18 the matchlines (ML) are reset to ground and the match-sense nodes of the sense amplifiers are precharged to V_{DD} by asserting MLRST. During the precharging of the MLs, the new search data is applied to the SLs. Next, MLEN is activated; the current sources attached to each ML begin charging the MLs. Since a path to ground is created by any unmasked mismatch between the search and stored data, a ML with no mismatch experiences a charge-up at a higher rate compared to any ML with at least one mismatch.

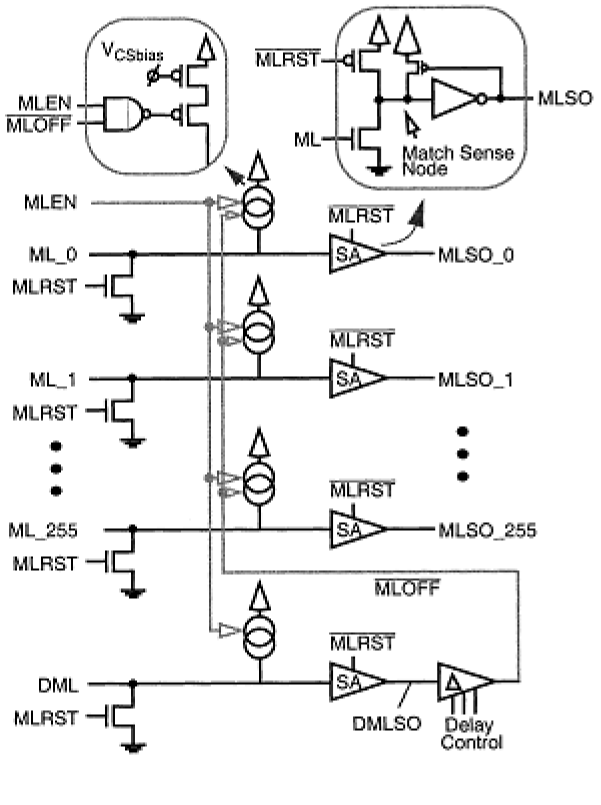

Figure 18 Current-race ML sensing scheme. [3]

A dummy matchline (DML) is also included which is identical to a ML with no mismatch. [3] During the ML charge-up process, all MLs are timed in a race against the DML, to reach the threshold voltage of their corresponding sense amplifier. Once the DML charges up beyond the threshold of the sense amplifier, its match-sense node is discharged, and the MLOFF signal is sent to all MLs to disable their current sources and to latch the match results. All MLs that have reached the threshold of the sense amplifier detect a match and all others sense amplifiers detect a mismatch.

This sensing scheme saves power firstly by cutting the current when DML reaches the threshold voltage of the ML sense amplifier thus reducing the voltage swing. This reduces the ML power dissipation. Secondly, by pre-charging the MLs to ground, the SLs do not need to be reset between consecutive searches; hence SL switching activity is minimized and thereby reducing SL power dissipation by a factor of two. [3] Fig. 19 shows the simulated voltage waveforms for the current race scheme.

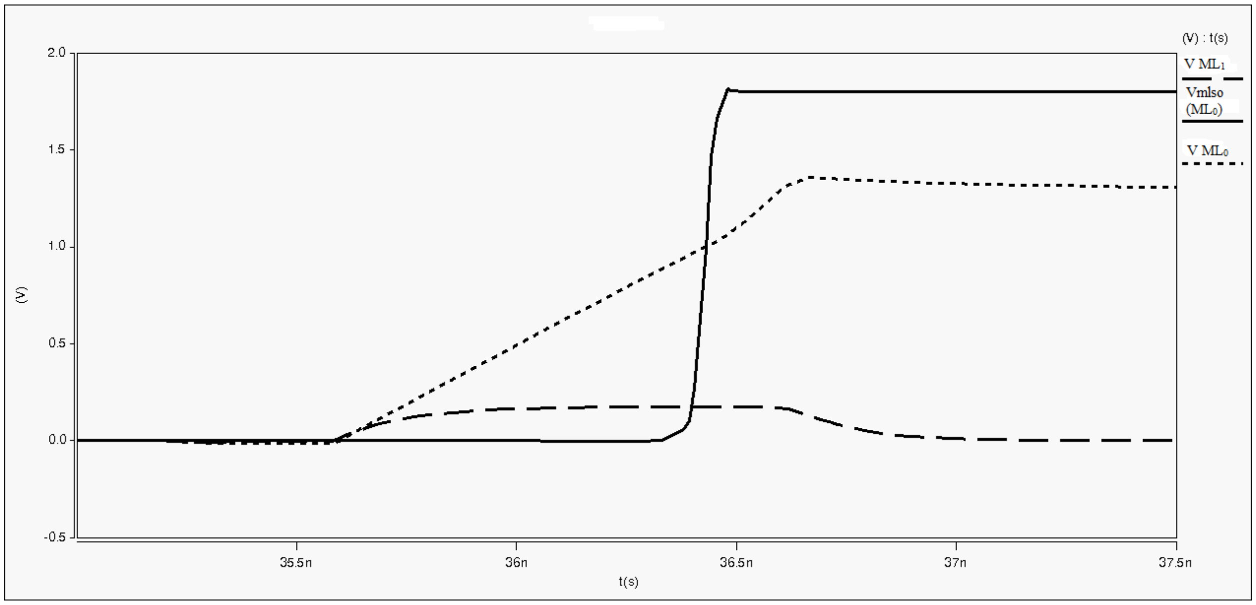

Figure 19 Voltage waveform for ML0, Vmlso (ML0) and ML1 for current race scheme

3.2 <u>Resistive shielding scheme:</u>

Fig. 20 shows the MLSA with resistive shielding. It uses an nMOS transistor (N3) to decouple the ML and its MLSA. The N3 channel resistance shields the sensing point (SP) in Fig. 20 from the highly capacitive ML. [2] This way, the current source can be sized down to save power without sacrificing the sensing speed. Due to the body effect and the decreasing gate-to-source voltage, the N3 channel resistance increases when the ML voltage is rising up. [2] The N3 channel resistance depends on the number of mismatch bits. For instance, ML_0 would be rising faster than $ML_{1,}$ which implies that N3 has higher channel resistance in case of ML_0 to shield SP. Since less current flows to ML, SP node charges faster to reach the threshold voltage of N1. [2] Faster sensing of the dummy word also reduces energy consumption because the ML current sources are shut down sooner. Energy and delay can be further reduced by decreasing $V_{res.}$ Since the rising ML voltage is increasing the current to MLSA sensing point, the resistive shielding scheme provides the same effect as a positive-feedback circuit does. Fig. 21 shows the voltage waveforms for ML_0 and ML_1 for a 16 word 32-bit TCAM when it is simulated with the resistive shielding scheme in 0.18-µm CMOS technology. Fig. 22 shows the current waveform in the sensing point (I_{SP}) for ML_0 and ML_1, it can be seen that I_{SP} for ML_0 is higher than that for ML_1 indicating that the positive feedback of the circuit is functional. I_{SP} for greater number of mismatches ML_K will be much lower than ML_1.

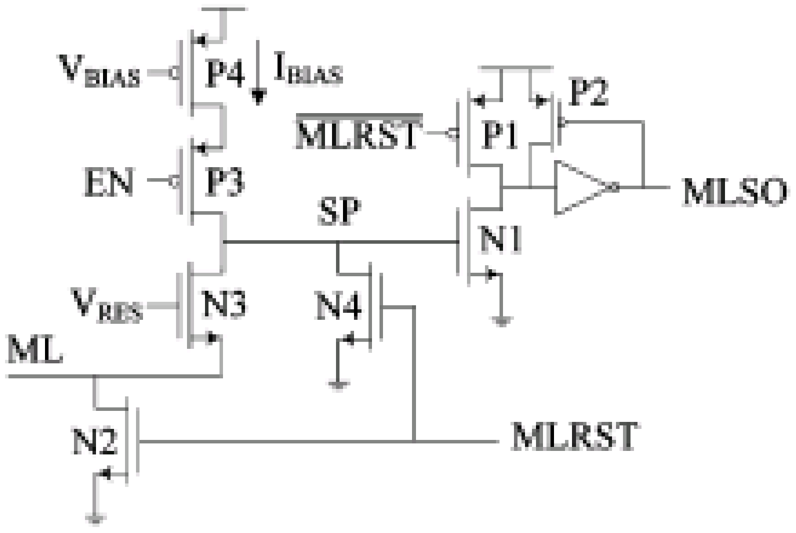

Figure 20 Resistive shielding scheme [2]

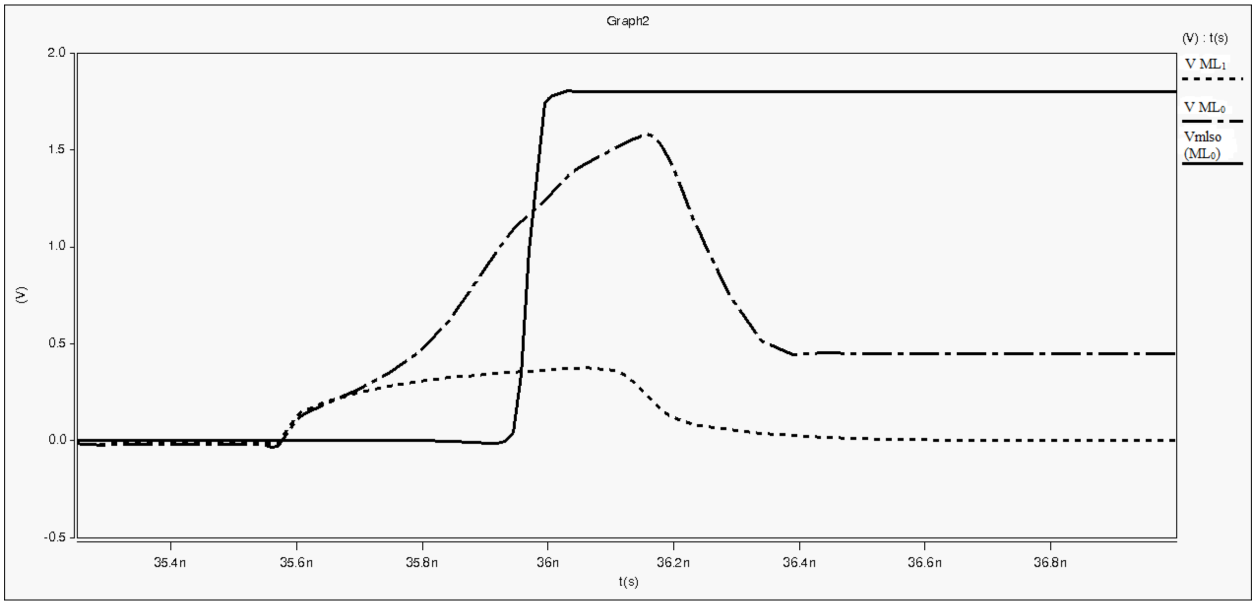

Figure 21 Voltage waveform for ML0, Vmlso (ML0) and ML1 for resistive shieldiing scheme

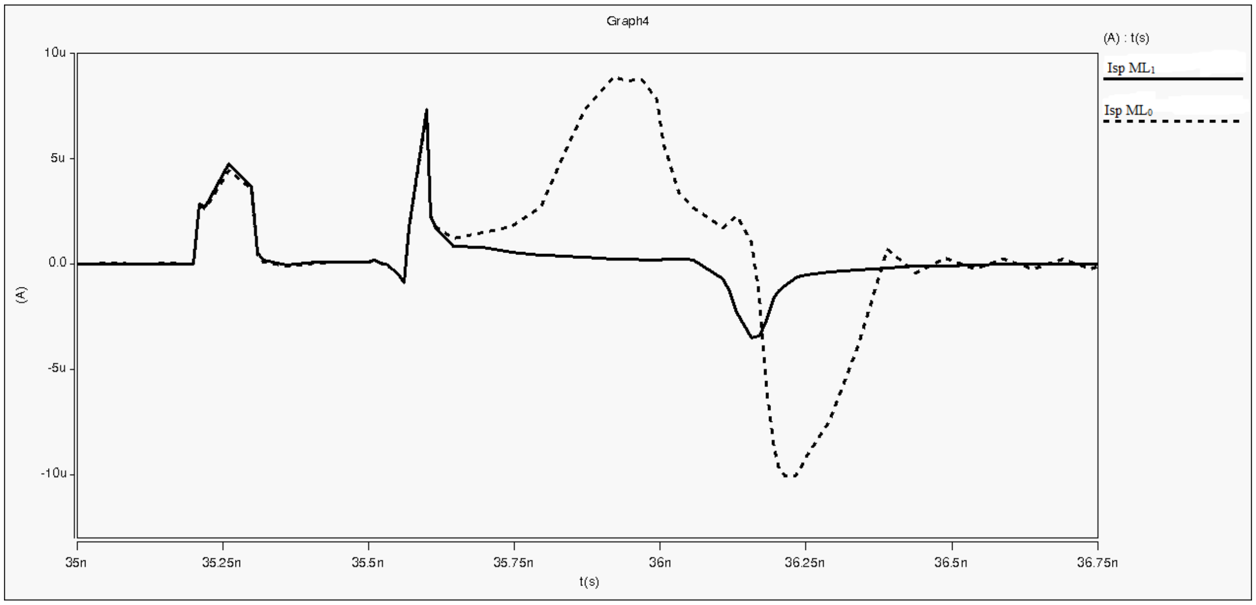

Figure 22 Current waveform for ML0 and ML1 for resistive shielding scheme

3.3 Active feedback scheme:

Fig. 23 shows the MLSA with active feedback. Transistor N3 operates as a constant current source (I_{FB}) to bias the feedback circuit. [2] Initially, all MLs receive the same current from the current sources (I_{Bias}). [2] As ML_0 charges at a faster rate than ML_k, its P6 source-to-gate voltage becomes smaller than that of $ML_{k.}$ In order to keep the current through P6 constant (I_{FB}), a reduction in $V_{GS\ P6}$ is compensated by an increase in P6 source-to-drain voltage. [2] Since the source terminal of P6 is close to V_{DD} a larger $V_{SD\ P6}$ results in a smaller $V_{CS.}$ Thus the faster charging of ML_0 makes its V_{CS} smaller than that of $ML_{k.}$ As a consequence ML_0 receives higher current (I_{BIAS}) and charges more rapidly than $ML_{k.}$ [2] This action continues until DML reaches the MLSA threshold voltage and switches DMLSO and hence turns off the current sources. Fig. 24 shows the voltage waveforms for ML_0 and ML_1 for a 16 word 32-bit TCAM when it is simulated with the active feedback scheme in 0.18-μm CMOS technology. Fig. 25 shows the current waveform in the matchline (I_{ml}) for ML_0 and ML_1, it can be seen that I_{SP} for ML_0 is higher than that for ML_1 indicating that the positive feedback of the circuit is functional. I_{SP} for greater number of mismatches ML_K will be much lower than $ML_{1.}$

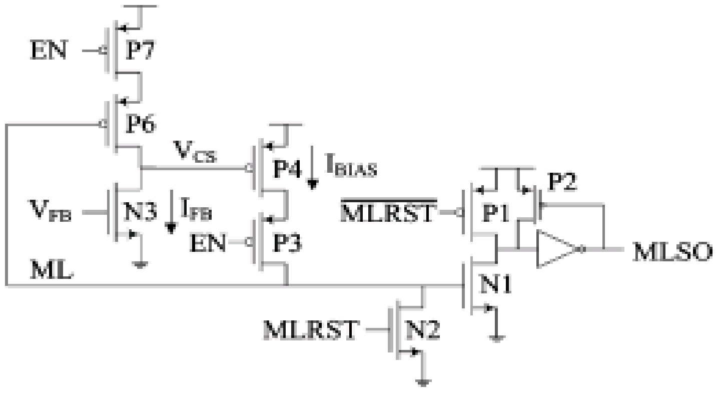

Figure 23 Active feedback scheme [2]

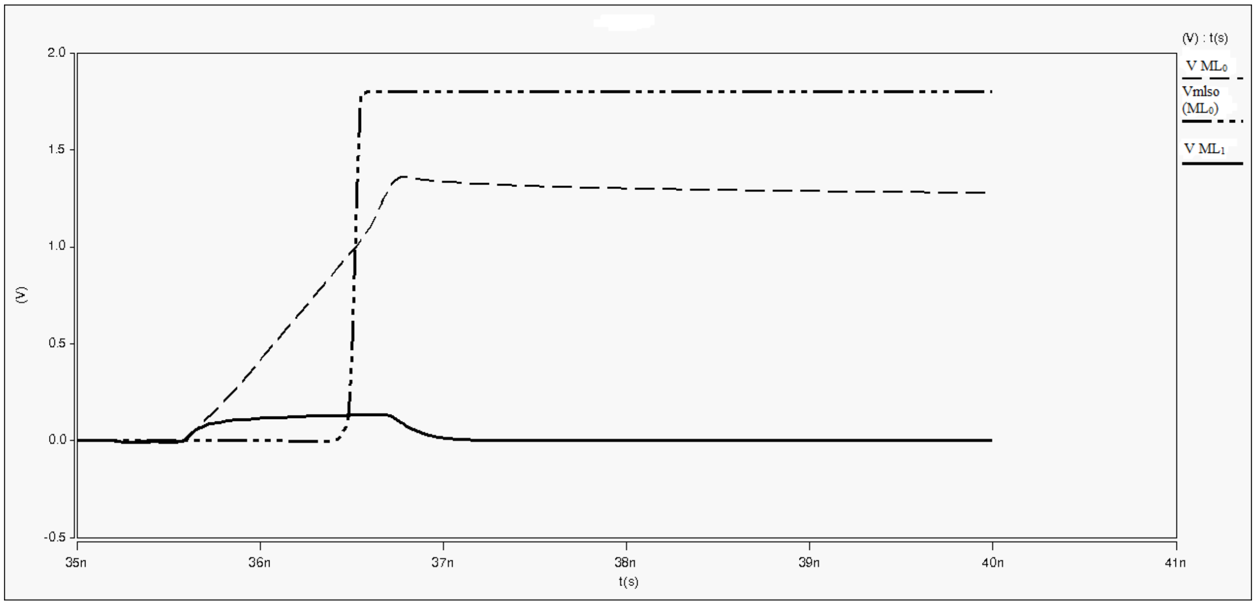

Figure 24 Voltage waveform for ML0, Vmlso (ML0) and ML1 for active feedback scheme

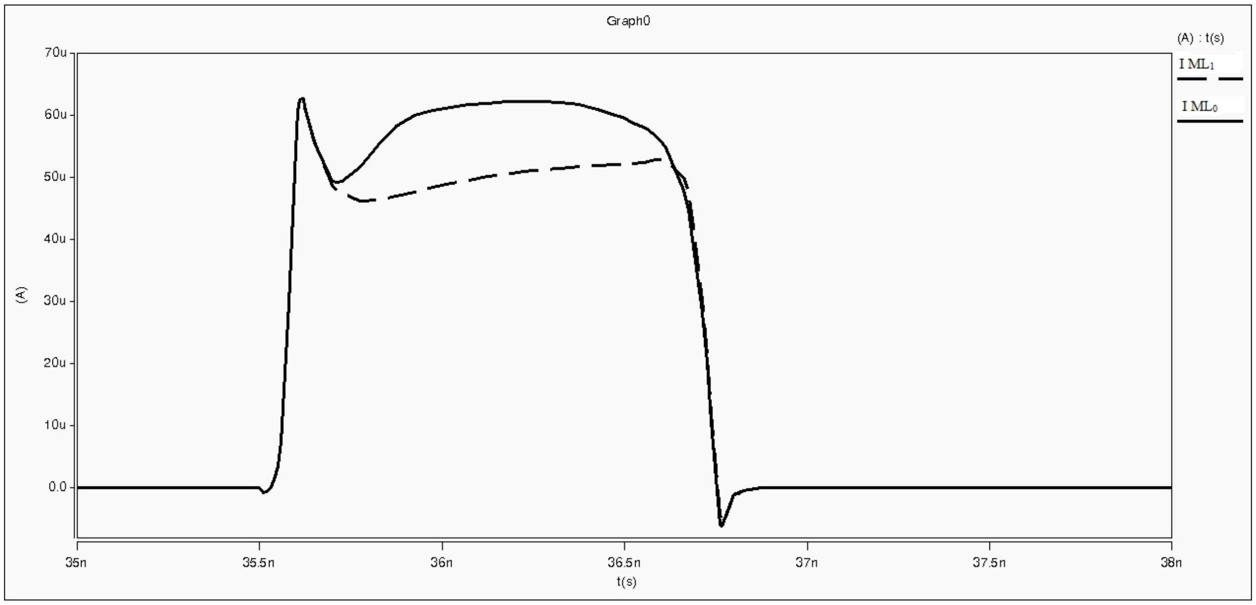

Figure 25 Current waveform for ML0 and ML1 for active feedback scheme

3.4 Results:

The simulated search times and power consumption for all the above simulated schemes are tabulated in table 4.

Name of the scheme	Number of transistors per MLSA	Voltage Margin/mV	Search time/ns	Peak Power/µW	Energy consumption/fJ
Current Race	6	814.6	0.877	2.5177	2.528
Resistive Shielding	8	814.62	0.418	2.2357	1.1253
Active feedback	9	851.32	0.964	3.7646	3.507

Table 2 Simulation Results

The voltage margin for the three simulated matchline sensing schemes were kept constant and they were compared on the basis of search time, peak power, energy consumption and number of transistors per MLSA. Considering Fig 23 and 26 we can see that current flowing to the sensing point for ML_0 is far greater for Resistive Shielding Scheme than the Active Feedback Scheme, which shows that the feedback works better in the Resistive Shielding Scheme. From table 4 it can also be concluded that Resistive Shielding Scheme is the most efficient and fastest among the three simulated schemes.

4. Reference

[1] Kostas Pagiamtzis, Student Member, IEEE, and Ali Sheikholeslami, Senior Member, IEEE "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 3, MARCH 2006

[2] Nitin Mohan, Member, IEEE, Wilson Fung, Member, IEEE , Derek Wright, Student Member, IEEE, and Manoj Sachdev, Senior Member, IEEE "A Low-Power Ternary CAM With Positive-Feedback Match-Line Sense Amplifiers" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 56, NO. 3, MARCH 2009

[3] Gor Arsovski, Student Member, IEEE, Trevis Chandler, Student Member, IEEE, and Ali Sheikholeslami, Senior Member, IEEE "A Ternary Content-Addressable Memory (TCAM) Based on 4T Static Storage and Including a Current-Race Sensing Scheme" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 1, JANUARY 2003

[4] Nitin Mohan, Member, IEEE "Low-Power High-Performance Ternary Content Addressable Memory Circuits" A thesis presented to the University of Waterloo in the fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Electrical and Computer Engineering, 2006