Study & Performance Comparison of Positive Feedback Match-Line (ML) Sensing Schemes for Low Power TCAM

بسم الله الرحمن الرحيم

Md. Istiaque Rahaman, 082433
Iftekharul Islam, 082435
Md. Mehedi Hassan Galib,082437

Syed Iftekhar Ali
Assistant Professor
Electrical and Electronic Engineering (EEE)
Islamic University of Technology (IUT)

A Thesis submitted to the Department of Electrical and Electronic Engineering (EEE)
in Partial Fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering (EEE)

Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Gazipur, Bangladesh
October, 2012

CERTIFICATE OF RESEARCH

This is to certify that the work presented in this thesis paper is the outcome of the research carried out by the candidates under the supervision of Syed Iftekhar Ali, Assistant Professor Electrical and Electronic Engineering (EEE) It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree or any judgment.

Authors

Md. Istiaque Rahaman **Iftekharul Islam** **Md. Mehedi Hassan Galib**

Signature of Supervisor

Syed Iftekhar Ali
Assistant Professor
Electrical and Electronic Engineering (EEE) Department

Signature of the Head of the Department

Prof. Dr. Md. Shahid Ullah
Head of the Department
Electrical and Electronic Engineering,
Islamic University of Technology (IUT).

Abstract

Ternary content addressable memories (TCAMs) are hardware-based parallel lookup tables with bit-level masking capability. They are attractive for applications such as packet forwarding and classification in network routers especially in internet applications. Despite the attractive features of TCAMs, high power consumption is one of the most critical drawbacks of TCAM. Hence reducing the power consumption without sacrificing the speed and voltage margin is the most difficult part in TCAM design. Among different match line sensing schemes, the use of positive feedback in the sense amplifiers is one of the best solutions to this problem. The main feather of this work is to perform comparison among different existing positive feedback based match line sensing schemes, i.e., mismatch dependent, active feedback and resistive feedback schemes using four performance parameters which are (i) search time (ii) voltage margin (iii) peak dynamic power and (iv)worst case energy consumption. All the schemes are simulated using 130nm, 1.2V CMOS logic. It is shown in this work that the energy saving is maximum (68.77%) in resistive feedback scheme compared to conventional CR-MLSA. Again, comparing among the positive feedback based schemes it is found that the resistive feedback provides with the best speed. Mismatch dependent scheme provides the best voltage margin and peak dynamic power. The worst case energy consumption is least in Active feedback scheme among all three positive feedback based scheme.

Acknowledgements

This work was supported by Electrical and Electronic Engineering Department of Islamic University of Technology, OIC, Bangladesh.

We would like to express my deep gratitude to our supervisor, Syed Iftekhar Ali, Assistant Professor of Electrical and Electronic Engineering Department of Islamic University of Technology for providing moral support, technical guidance and encouragement. He gave us full freedom to develop our research. At the same time, he was easily accessible whenever we needed his guidance and feedback. During the tough times (when we were having difficulties in simulation results), he provided us moral support and useful tips. Our association with him made our undergraduate research an enjoyable learning experience. We did not have enough ideas about VLSI circuit design, when we started our research and our supervisor extended his hand to cooperation to us by making us understand the basis of VLSI design. He pressurized us only in certain tasks where he felt that, things should have been done by us without his assistance and at the end of the day after finishing the task we gained self-confidence and containment. We are honored to do research under his supervision.

We would like to thank Prof. Shahid Ullah, Head of the department, EEE, IUT; Prof. Kazi Khairul Islam, Prof. Ashraful Hoque, Dr. Fakrul Islam, Niamul Quader, Moshiur Rahman Farazi for encouraging us to do research work on VLSI circuit design. We would also like to thank our friends to support us during our study and research.

Last, but definitely not the least, a very special thanks to our beloved parents who always supported us with their unending love and care during all ups and downs of our research. Their encouragement, moral support, faith and love were the source of my motivation, energy and enthusiasm.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

In the era of modern data communication and networking, massive increase of internet users throughout the world has given birth to the demand of high speed internet networks. The internet network comprises of routers and switches which processes the data packets and sends it to an appropriate recipient. A header, user data and a trailer makes a data packet. In the header field there are flag, address field and control field [1]. When independent networks and links are connected to create internetworks (network of networks) or a large network, the connecting device (called routers or switches) route or switch the packets to their final destination [1]. When a switch receives the packet, this destination address is examined; the routing table which contains destination address and corresponding forwarding output ports in a table is consulted to find the corresponding port through which the packet should be forwarded. This destination address in the header of a data packet remains the same during the entire journey of the packet and the routing tables are dynamic and are updated periodically.

Typically, optical fiber communication is used to transport the data from one router to another. Advances in optical fiber technologies, such as wavelength division multiplexing, have drastically increased the data transfer rates over optical fibers. To optimize the high speed data rate of optical fiber communication technology, routers and switches are needed to match the speed of optical fiber communication. One of the most time consuming task in a network switch or a router is to consult the routing lookup table and find the exact destination of the data packets. Hence, the existing technologies have been pushed to their limit to meet the higher data rate demand of today's bandwidth-hungry world.

The communication at the network layer is host-to-host (computer-to-computer); a computer somewhere in the world may need to communicate with another computer which is somewhere else in the world. Usually, computers communicate through the internet. The packets sent by the host computer may pass through numerous routers and switches before reaching the client computer. For this level of communication, we need a global addressing scheme; called Internet Protocol (IP) address in the network layer of the TCP/IP protocol suite. Each internet address is 32 bits in length; gives us a maximum of 2^{32} addresses. The addresses are referred to as IPv4 (IP version 4). But to accommodate more users in internet lead to need of more addresses. This need along with other concerns about IP layer, motivated a new design of IP layer called the new generation of IP or IPv6 (IP version 6). In this version the internet uses 128 bits addresses that give much greater flexibility in address allocation. The increasing number of network nodes supported by IPv6 significantly increases the capacity and word-size of the routing table used for packet forwarding compared to that of IPv4. This leads to an increased length of data packet and routing table. But the tradeoff is increased time consumption in finding the exact destination of the data packets from the routing lookup table [1].

Now any hardware or software search methods may be deployed for table lookup. But the software search methods such as radix trees are relatively slow. In some cases a hash function can perform the lookup in one memory access. Although the radix trees show better performance than the hashing function but both of their performance degrades drastically with the increase of routing table's dimension. 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). **A** content-addressable memory (CAM) compares input search data against a table of stored data, and returns the location where it finds a match. CAMs have a single clock cycle throughput making them faster than other hardware and software-based search systems. Although the CAM has many applications like parametric curve extraction, Hough transformation, Huffman coding/decoding, Lempel–Ziv compression, and image coding but the primary commercial application of CAMs today is to classify and forward Internet protocol (IP) packets in network routers. Hence the current CAM research is primarily driven by the networking applications, which require high capacity CAMs with low-power and high-speed operation.

1.1 CAM Basics

A CAM is a good choice for implementing lookup operation due to its fast search capability. However, the speed of a CAM comes at the expense of increased silicon area and power consumption, two design parameters that researchers are striving to reduce. Power consumption problem becomes more acute with the larger CAMs. Hence, reducing power consumption, without sacrificing speed or area, is the main challenge forward.

1.1.1 Conceptual Structure

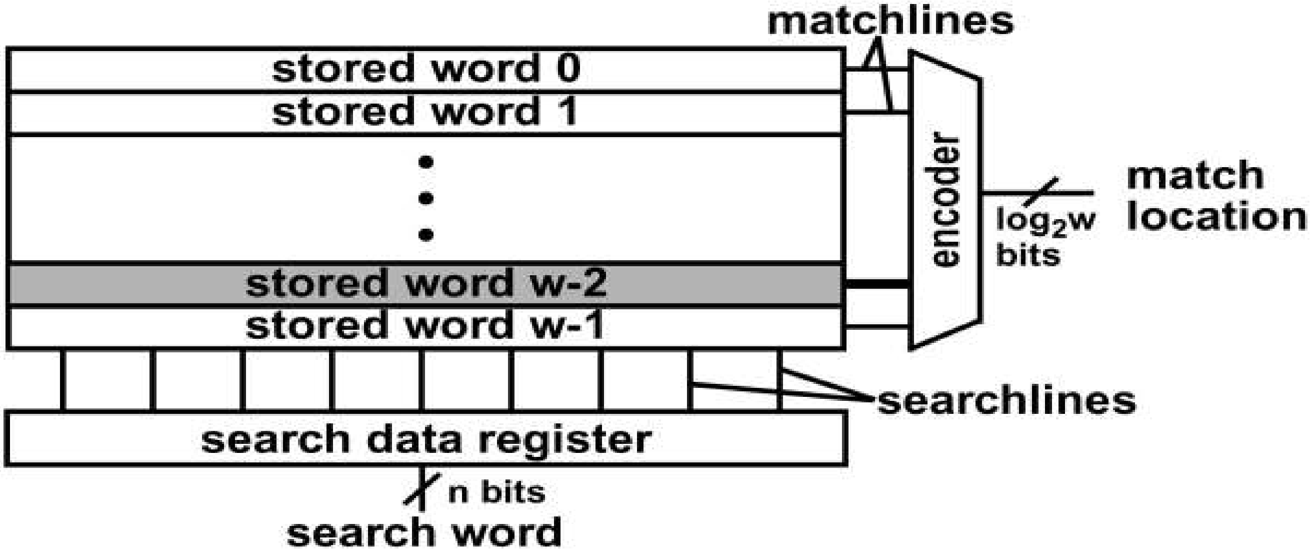


Fig: 1.1 Conceptual view of a content-addressable memory containing w words [2]

Figure 1.1 provides a conceptual view of a CAM with w number of words. The input to the system is the search word that is broadcasted onto the searchlines to the table of stored data. The number of bits in a CAM word varies between 36 to 144 bits. Each stored word has a matchline that indicates whether the search word and stored word are identical (the match case) or are different (a mismatch case). The matchlines are fed to an encoder which generates a binary *match location* corresponding to the matchline that is in the match state [2]. Usually among the stored data only one line is expected to be matched with the search data. A priority encoder is used in those cases where there is probability of more than one match. A priority encoder selects the highest priority matching location to map to the match result, with words in lower address locations receiving higher priority. So the overall function of a CAM is to provide a memory location in which the stored data matches with the search data after consulting the routing lookup table.

1.1.2 CAM basic Structure

The figure 1.2 shows a CAM consisting of 4 words, with each word containing 3 bits arranged horizontally (corresponding to 3 CAM cells). There is a matchline corresponding to each word (ML_0, ML_1, ML_2 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.). A CAM search operation begins with loading the search-data word into the search-data registers. Then search data is provided into the differential searchlines and each CAM core cell compares its stored bit against the bit on its corresponding searchlines. Matchlines on which all bits match become/remain 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 [2].

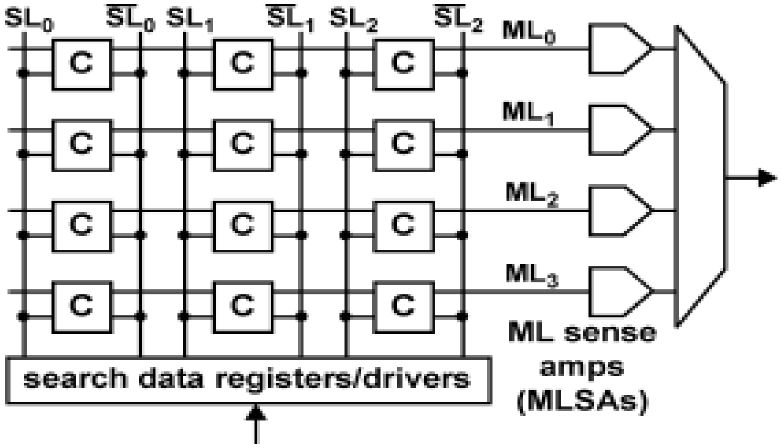


Fig: 1.2 Simple schematic of a model CAM with 4 words having 3 bits each [2].

A CAM cell serves two basic functions: bit storage (as in RAM) and bit comparison (unique to CAM). CAMs can be divided into two categories: (i) binary CAMs and (ii) ternary CAMs

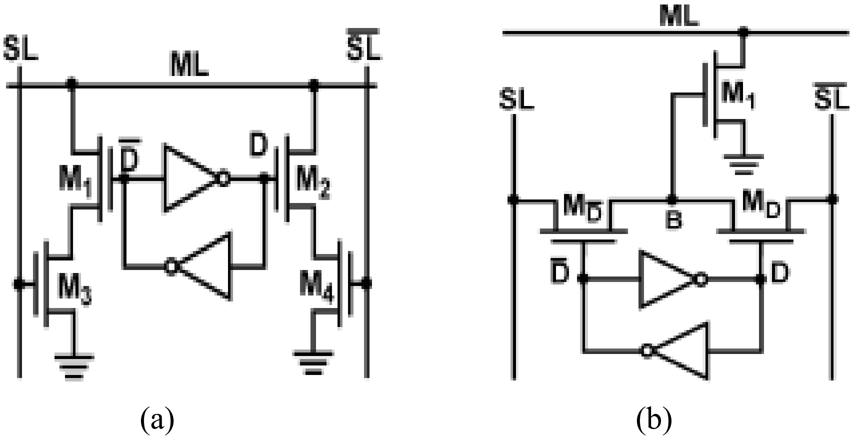
(TCAMs). A binary CAM can store and search binary words (made of ‘0’s and ‘1’s). Thus, binary CAMs are suitable for applications that require only exact-match searches. A more powerful and feature-rich TCAM can store and search ternary states (‘1’, ‘0’, and ‘X’). The state ‘X’, also called ‘mask’ or ‘don’t care’, allowing a wildcard operation. Wildcard operation means that an “X” value stored in a cell causes a match regardless of the input bit. In the next section, Section 1.2 the structure of binary CAM will be discussed.

1.2 The Binary CAM cell

We begin our survey by looking at the two common binary CAM cells, the NOR cell and the NAND cell, each of which can be used as the basic building block in Fig 1.3. Shows a NOR-type CAM cell and Fig.1.4 shows a NAND-type CAM cell. The bit storage in both cases is an SRAM (Static Read only Memory) cell where cross-coupled inverters are deployed and the bit-storage nodes are D and D_c. Although is some cases DRAM (Dynamic RAM) are used but most common data storage mechanism is the use of SRAM cells. The bit comparison, which is logically equivalent to an XOR of the stored bit and the search bit is implemented by NOR and the NAND cells.

1.2.1 NOR cell

According to Fig. 1.4a the NOR cell implements the comparison between the complementary stored bit, D (and D_c), and the complementary search data on the complementary searchline, SL (and SL), using four comparison transistors M_1, M_2, M_3, M_4. These transistors implement the pulldown path of a dynamic XNOR logic gate with inputs SL and D. Each pair of transistors form a pulldown path from the matchline, ML, such that a mismatch of SL and D activates least one of the pulldown paths, connecting ML to ground. A match of SL and D disables both pulldown paths, disconnecting ML from ground [2]. 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. The pulldown paths connect in parallel resembling the pulldown path of a CMOS NOR logic gate. There is a match condition on a given ML only if every individual cell in the word has a match. Fig. 1.3b shows a variant of the NOR cell. The NOR cell uses 9-transistors compared to previous 10-T NOR cell.



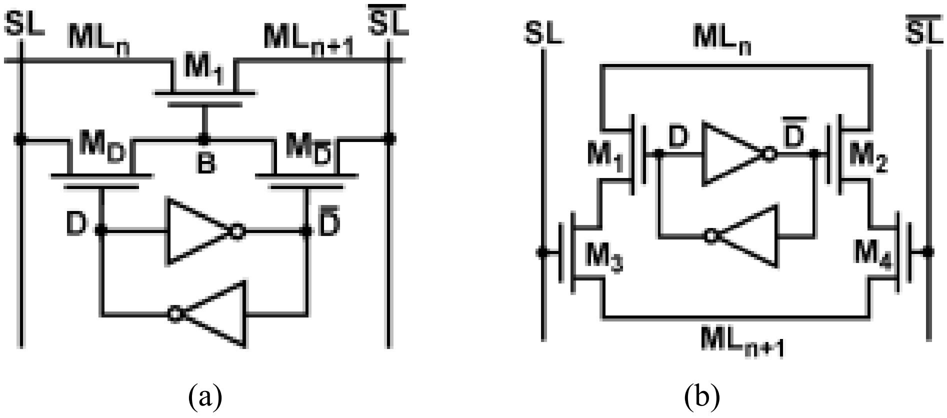
(a) (b)

Figure: 1.3 (a) 10-T NOR cell (b) 9-T NOR cell [2]

1.2.2 NAND Cell

The NAND cell implements the comparison between the stored bit, D, and corresponding search data on the corresponding searchlines, (SL, SL_c), using the three comparison transistors M_1, M_D and M_{Dc}. For example in 9-T NAND cell in Fig. 1.4a, Consider the case of a match when SL=1 and D=1. Pass transistor M_D is ON and passes the logic "1" on the SL to node B. The logic "1" on node B turns ON transistor M_1. M_1 is also turned ON in the other match case when SL=0 and D=0. In this case, the M_{Dc} transistor passes logic high to raise node B. The remaining cases, where SL & D are not same, result in a miss condition, and accordingly node B is logic "0" and the transistor is OFF. Node B is a pass-transistor implementation of the XNOR function [2]. The NAND nature of this cell becomes clear when multiple NAND cells are serially connected. A serial nMOS chain of all the transistors resembles the pulldown path of a CMOS NAND logic gate. A match condition for the entire word occurs only if every cell in a word is in the match condition. One variant of 9-T NAND cell (Fig. 1.4a) is 10-T NAND cell shown in Fig. 1.4b.

An important property of the NOR cell is that it provides a full rail voltage at the gates of all comparison transistors. On the other hand, a deficiency of the NAND cell is that it provides only a reduced logic "1" voltage at node B, which can reach only V_{DD}-V_{tn} when the searchlines are driven to V_{DD} (where V_{DD} is the supply voltage and V_{tn} is the nMOS threshold voltage).



(a) (b)

Figure: 1.3 (a) 10-T NAND cell (b) 10-T NAND cell [2]

The bit storage portion of a CAM cell is a standard 6T static RAM (SRAM) cell. Hence, this cell performs READ and WRITES operations similar to an SRAM cell. The following section will briefly describe the READ, WRITE and SEARCH operation of a 10-T NAND cell.

1.2.2.1 WRITE Operation

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 [1]. Figure 1.5 shows the WRITE operation when ‘0’ is being written to a cell which originally stored ‘1’. Originally, Vx = ‘1’ and Vy = ‘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 I0 and I1 (shown by dashed arrows in Figure 1.5). These transient currents form voltage dividers (P1-N6 and N7- N9). If these transient currents can pull one of the nodes (Vx and Vy) to the inverter threshold voltage, the other node will flip due to the feedback action of the cross-coupled inverters. If the inverter threshold voltage is $V_{DD}/2$, N7 needs to be much larger (≥10x) than N9 to pull Vy above this value because it is difficult to pass logic ‘1’ using an NMOS transistor. On the other hand, Vx can be pulled below this value by choosing same size P1 and N6 (shown by encircled W in Figure 1.5). Thus, the latter sizing is adopted almost universally.

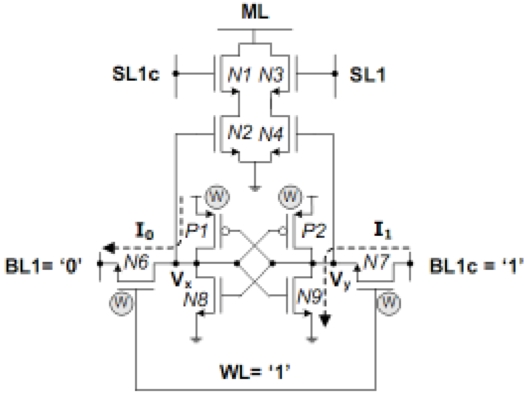


Figure1.5. WRITE Operation in 10-T Binary CAM (NAND cell) [1]

1.2.2.2 READ Operation

The READ operation is performed by pre-charging the BLs to VDD and enabling the WL. Figure 1.6 shows the READ operation, when '0' is stored (i.e. Vx = '0', Vy = '1'). Since the BL drivers are turned off during the READ operation, current IREAD discharges BL1 (through N6 and N8). BL1c remains at VDD because Vy = '1'. Therefore, a small differential voltage develops between BL1 and BL1c [1]. Since the BLs are shared among all the cells in a column, they are highly capacitive. The small voltage swing in the BLs reduces power consumption and the access time during the READ operation. As shown in Figure 1.6, the current IREAD raises the voltage Vx. Thus, the driver transistors (N8-N9) are sized such that Vx remains below the inverter threshold voltage, and hence the cell does not flip during the READ operation. Typically, the driver transistors (N8-N9) are sized 1.5 times wider than the access transistors (N6-N7).

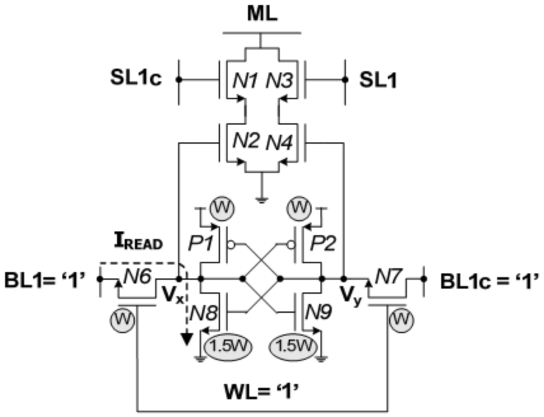
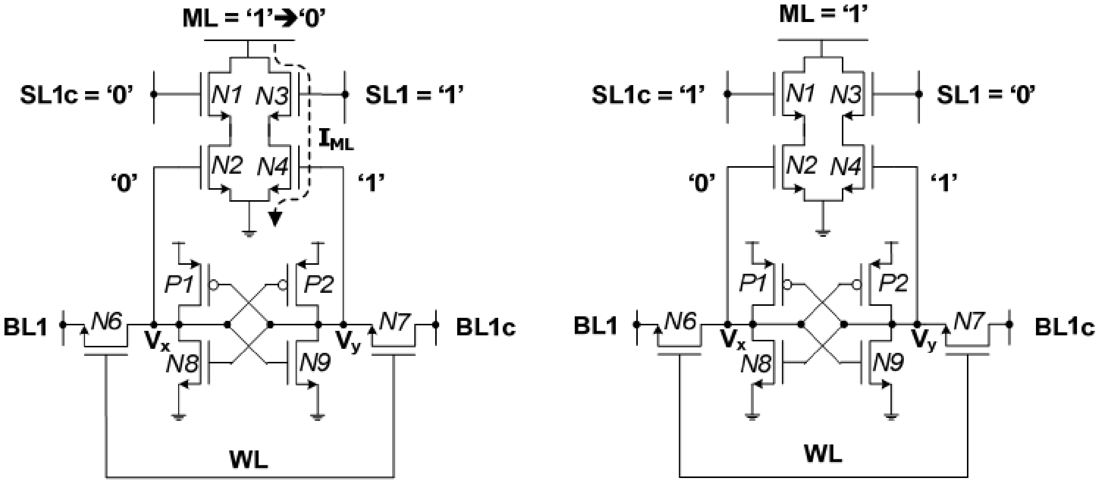


Figure1.6. READ Operation in 10-T Binary CAM (NAND cell) [1]

1.2.2.3 SEARCH Operation

The conventional SEARCH operation is performed in three steps. First, search lines (SLs) SL1 and SL1c are reset to GND. Second, ML is pre-charged to V_{DD}. 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 V_{DD} indicating a "match". Otherwise, if the search key bit is different from the stored value, one of the pull-down paths conducts and discharges the ML to GND indicating a "mismatch" [1]. 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. Figure 1.7 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 1.7(a). Similarly for SL1 = '0', ML remains at '1' detecting "match" as shown in Figure 1.7(b).



(a) (b)

Figure1.7: SEARCH Operation (a) mismatch (b) match [1]

1.3 The Ternary CAM cell (TCAM)

While the Binary CAM performs exact-match searches, a more powerful Ternary CAM (TCAM) allows pattern matching with the use of "don't cares." Don't cares act as wildcards during a search, and are particularly attractive for implementing longest-prefix-match searches in routing tables. Dynamic storage of ternary data requires refresh operation and an embedded DRAM process, while static storage (SRAM) of ternary data requires considerable layout area. The construction of TCAM by SRAM will be discussed in detail in the next chapter.

CHAPTER 2

2. The Ternary CAM structure, Matchline & Searchline sensing schemes

As stated earlier that, the basic difference between Binary CAM and TCAM is the later is capable of doing wildcard operation that is if “X” value is stored in the TCAM then it shows a match regardless of the input bits in the seachline. Here “X” means don’t care or mask condition and either “1” or “0” can be stored in the TAM cell. This sort of operation can be implemented using basic two types of TCAM cells namely NOR type TCAM cell and NAND type TCAM cell. Each TCAM cell is connected with a matchline (ML) and a searchline (SL, SL). This wildcard operation is extensively used in packet data forwarding in networking application. In this chapter we will first discuss about how this wild card operation is used in packet data forwarding using TCAM, basic TCAM structure will be discussed in detail and then different ML & SL sensing schemes will be discussed.

2.1 Basic searching operation

Address-lookup function is used in network routers for forwarding data packets from an incoming port to an outgoing port. The address-lookup function examines the destination address of the packet and selects the output port associated with that address. A basic routing table contains binary address and corresponding port to be selected.

Serial	Address(Binary)	Port
1.	101XX	A
2.	0110X	B
3.	011XX	C
4.	10011	D

Table 2.1: Sample Routing Table

There are four entries, each of them has 5-bits 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. This lookup method is called longest-prefix matching.

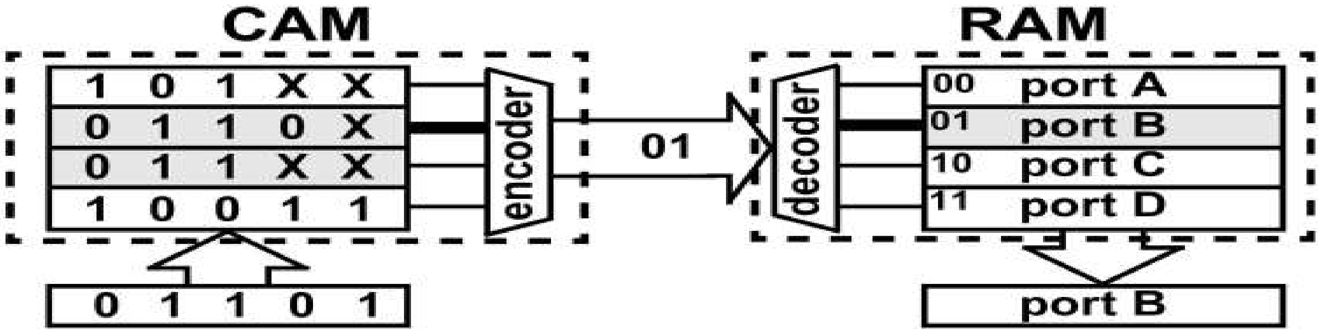


Fig:.2.1 TCAM Based implementation of routing table [2]

In figure 1.2 a CAM accomplishes address lookup by implementing the routing table shown in Table 1.1. On the left of Fig. 1.2, 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. 2.1 A RAM read operation outputs the port designation; port B, to which the incoming packet is forwarded.

2.2 TCAM structure

Generally a TCAM cell comprises of two SRAM cells and a comparator circuit. A single SRAM cell is constructed by two cross coupled inverter and 6 transistors are there in a single SRAM cell. The comparator circuit is made of 4 transistors. So, a TCAM cell is actually a 16-T structure shown. Match Line (ML), Search Line (SL) & Word Line are connected to a TCAM cell. In the following subsections the basic structure and operation of SRAM, Comparator circuit, NOR and NAND type TCAM cells, TCAM word and Array will be elaborated.

2.2.1 SRAM Structure

A SRAM is a single bit storage device made of two cross coupled inverters. In Figure 2.2, two n-MOS access transistors named M1, M2 are there to read/write a bit in a SRAM cell. When the word line is high then M1 and M2 are turned on and bit present in bit lines pass into SRAM. When the word lines are kept low then the bits written in the SRAM are stored in it. To read a data from the SRA word line is made high which in turn turns on M1 & M2 & bit present in SRAM pass into bit lines and thus they are read.

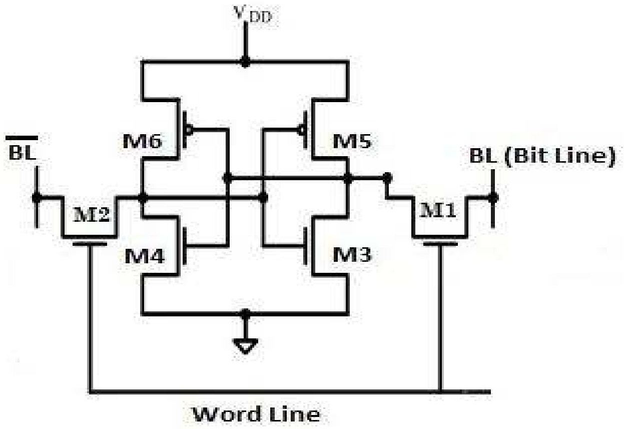
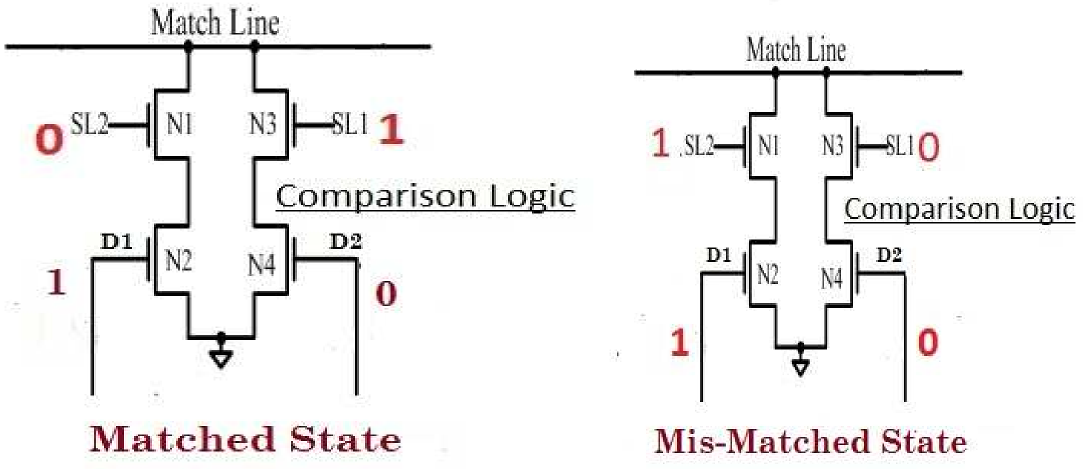


Figure 2.2 Single SRAM cell [3]

2.2.2 Comparator Circuit

A comparator circuit compares a search data with the stored data and provides with the result either match or mismatch.
If stored data & search data are same then the condition is called match condition and ML is floating (Fig. 2.3a). ML can be charged or discharged depending upon different schemes (SAs).
If the stored data & search data are not same then the condition is denoted as mismatch condition and ML grounded (Fig. 2.3b). ML can be charged or discharged depending upon different schemes (SAs).



(a) (b)

Figure 2.3: (a) matched state (b) mismatched state [2]

2.2.3 Ternary core cells for NOR- type CAM

A ternary symbol can be encoded into two bits according to Table 2.2. We represent these two bits as D and D_c. Note that although the D and D_c are not necessarily complementary, we maintain the complementary notation for consistency with the binary CAM cell. Since two bits can represent 4 possible states, but ternary storage requires only three states, we disallow the state where D and D_c are both zero. One bit, D, connects to the left pulldown path and the other bit, D_c, connects to the right pulldown path, making the pulldown paths independently controlled. We store an “X” by setting both D and D_c equal to logic “1”, which disables both pulldown paths and forces the cell to match regardless in the inputs. We store a logic “1” by setting D=1 and D_c = 0 and store a logic “0” by setting D= 0 and D_c= 1. In addition to storing an “X”, the cell allows searching for an “X” by setting both SL and to logic “0”. This is an external *don’t care* that forces a match of a bit regardless of the stored bit.

Value	D	D_c	SL	SL_c
0	0	1	0	1
1	1	0	1	0
X	1	1	0	0

Table 2.2 Ternary Encoding for NAND cells

To store a ternary value in a NOR cell, two SRAM cells are used, as shown in Fig. 2.4. Although storing an “X” is possible only in ternary CAMs, an external “X” symbol possible in both binary and ternary CAMs. In cases where ternary operation is needed but only binary CAMs are available, it is possible to emulate ternary operation using two binary cells per ternary symbol. A detailed circuit diagram of NOR type TCAM is shown in Fig. 2.5

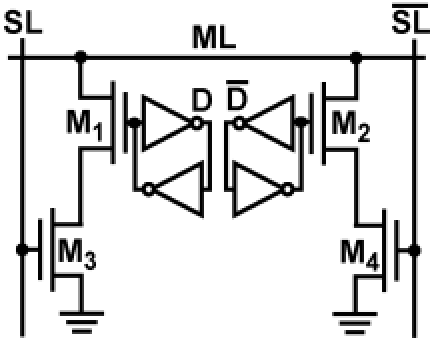


Figure 2.4 Ternary Core cells for NOR- type CAM [2]

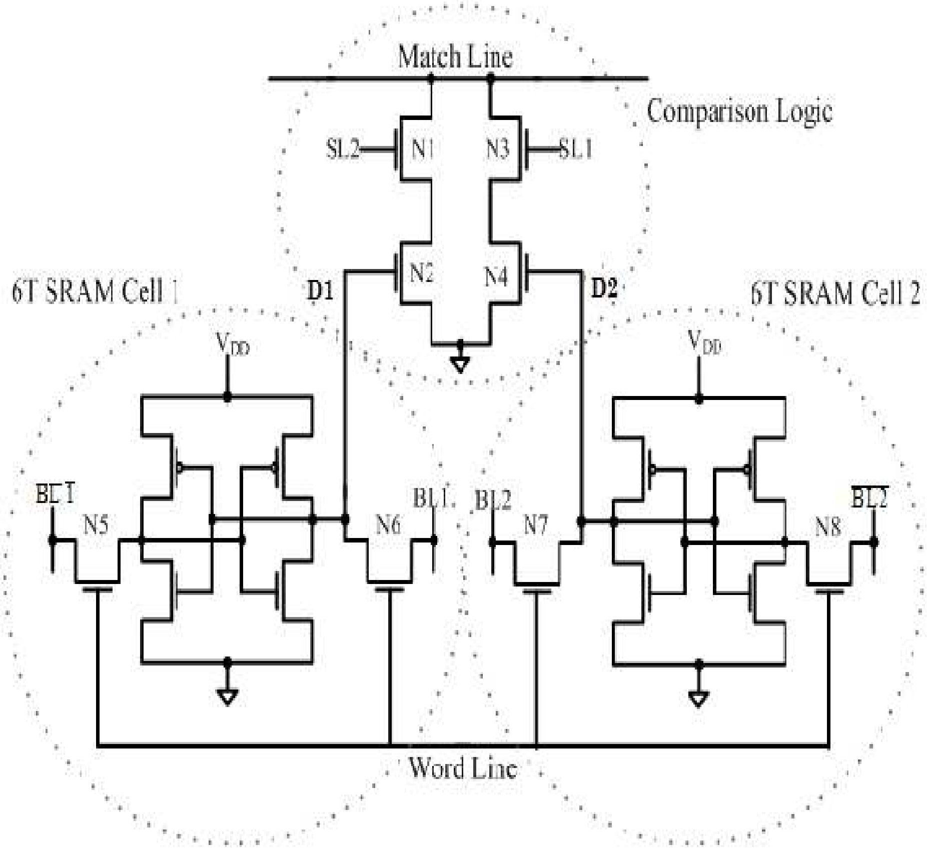


Figure 2.5 Ternary Core cells for NOR- type CAM (detailed circuit diagram) [3]

As a modification to the ternary NOR cell of Fig.2.4, is implementing the pulldown transistors using pMOS devices and complementing the logic levels of the searchlines and matchlines accordingly. Using pMOS transistors (instead of nMOS transistors) for the comparison circuitry allows for a more compact layout, due to reducing the number of spacings of p-diffusions to n-diffusions in the cell. In addition to increased density, the smaller area of the cell reduces wiring capacitance and therefore reduces power consumption. The tradeoff that results from using minimum-size pMOS transistors, rather than minimum-size nMOS transistors, is that the pulldown path will have a higher equivalent resistance, slowing down the search operation.

2.2.4 NAND cell

A NAND cell can be modified for ternary storage by adding storage for a mask bit at node M, as depicted in Fig. 2.6 [5] [6]. When storing an “X”, we set this mask bit to “1”. This forces transistor ON, regardless of the value of D, ensuring that the cell always matches. In addition to storing an “X”, the cell allows searching for an “X” by setting both SL and to logic “1”. Table 2.2 lists the stored encoding and search-bit encoding for the ternary NAND cell.

Value	D	D	SL	SL
0	0	0	0	1
1	1	0	1	0
X	0	1	1	1
X	1	1	1	1

Table 2.3 Ternary Encoding for NOR cells

Further minor modifications to CAM cells include mixing parts of the NAND and NOR cells, using dynamic-threshold techniques in silicon-on-insulator (SOI) processes, and alternating the logic level of the pulldown path to ground in the NOR cell [6]-[8]. Currently, the NOR cell and the NAND cell are the prevalent core cells for providing storage and comparison circuitry in CMOS CAMs.

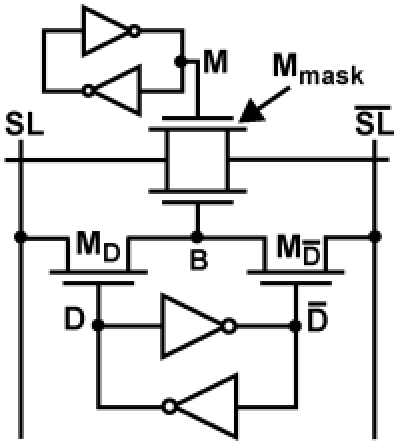


Figure 2.6 Ternary Core cells for NAND- type CAM [2] [6]

2.2.5 TCAM word

Several TCAM cells having a common match line (ML) & word line (WL) is called a TCAM word (Fig. 2.7). Search data bits are supplied in SLs. Sense Amplifier (SA) senses match or mismatch condition. MLSO is the output of SA and usually high for matched case. Basically the data packet’s address bits are stored in words and the searchline data is the desired client address in network applications.

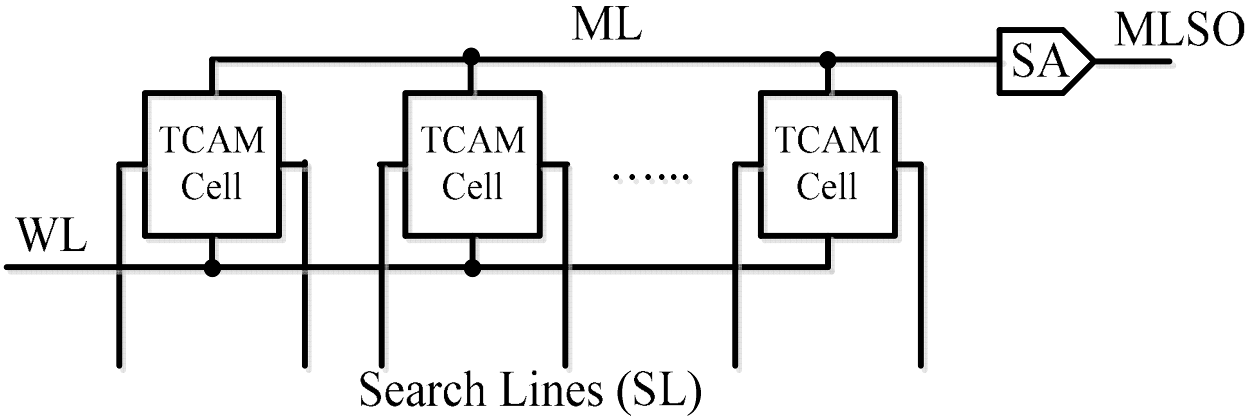


Figure 2.7 TCAM Word

2.2.6 TCAM Array

Several TCAM words having common search lines (SL) form a TCAM Array. The TCAM array is actually a lookup table of all the possible addresses. A desired address is feed to the search line and then some searching scheme is deployed to find a word which matches with the search data. For example, there are k numbers of words in the array having unique addresses stored in it. A search data is feed to it and it can be matched with any one of k number of words and to show a match the corresponding MLSO becomes high or low according to the matchline sensing scheme being used.

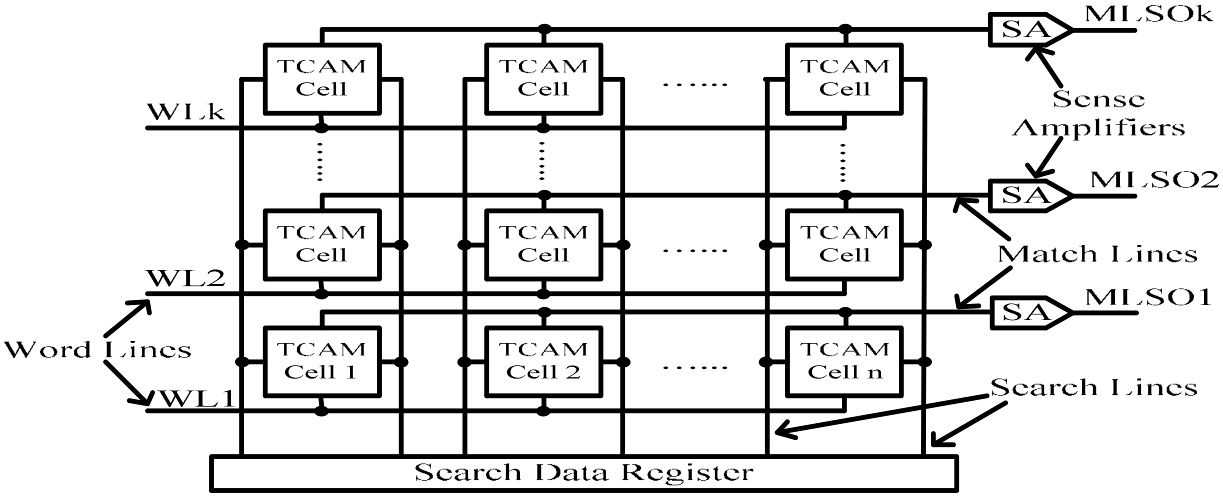


Figure2.8 TCAM array [3]

2.3 Matchline structures

There are two major structures in CAM. One of them is matchline structure and the other one is searchline structure. In the following subsection we shall demonstrate the NAND cell and NOR cell in constructing a CAM matchline.

2.3.1 NAND Matchline

A number of NAND cells are cascaded to form the matchline. The NAND cells cascaded together may be of binary of ternary type.

On the right of the figure, the precharge pMOS transistor, M_{pre}, sets the initial voltage of the matchline, ML, to the supply voltage, V_{DD}. Next, the evaluation nMOS transistor, M_{eval} , turns ON. In the case of a match, all nMOS transistors are ON, effectively creating a path to ground from the ML node, hence discharging ML to ground. In the case of a miss, at least one of the series nMOS transistors, through, is OFF, leaving the ML voltage high. A sense amplifier, MLSA, detects the difference between the match (low) voltage and the miss (high) voltage. Unlike the NOR matchline, the NAND matchline has a evaluation transistor marked as M_{eval} in Fig. 2.9 [2].

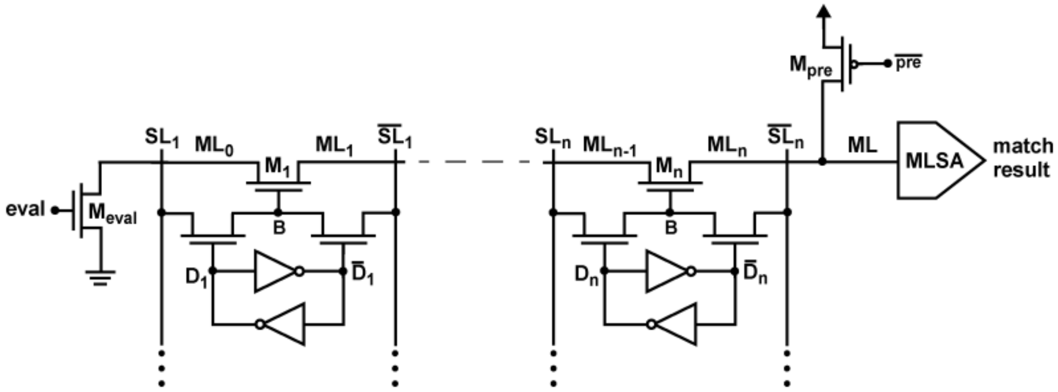


Figure 2.9: NAND matchline structure [2]

There is a potential charge-sharing problem in the NAND matchline. Charge sharing can occur between the ML node and the intermediate nodes. For example, in the case where all bits match except for the leftmost bit in Fig. 2.9, during evaluation there is charge sharing between the ML node and nodes through. This charge sharing may trigger a false match. To eliminate the charge sharing problem along with ML all the intermediate ML nodes are precharged high. This is accomplished by charging the searchlines and their complements V_{DD}. Then all the nMOS from M_1 to M_n are ON and prechages the intermediate nodes. When this precharge of the intermediate match nodes is complete, the searchlines are set to the data values corresponding to the incoming search word. This procedure eliminates charge sharing, since the intermediate match nodes and the ML node are initially shorted. NOR cells avoid this problem by applying maximum gate voltage to all CAM cell transistors when conducting. However, in case of NAND structure there is an increase in the power consumption due to the searchline precharge because usually the number of match is more than the number of mismatch in a TCAM array.

Two drawbacks of the NAND matchline are a quadratic delay dependence on the number of cells, and a low noise margin. The quadratic delay-dependence comes from the fact that adding a NAND cell to a NAND matchline adds both a series resistance due to the series nMOS transistor and a capacitance to ground due to the nMOS diffusion capacitance. These elements form an RC ladder structure whose overall time constant has a quadratic dependence on the number of NAND cells [9]. Most implementations limit the number of cells on a NAND matchline to 8 to 16 in order to limit the quadratic degradation in speed. The low noise margin is caused by the use of nMOS pass transistors for the comparison circuitry. Since the gate voltage of the NAND matchline transistors (M_1 through M_n) when conducting, in Fig.2.9, is V_{DD} - V_{tn} , the highest voltage that is passed on the matchline is , V_{DD} - $2V_{tn}$ (where is the threshold voltage of the nMOS transistor, augmented by the body effect). One implementation of a NAND-based CAM reclaims some noise margin by employing the bootstrap effect by reversing the polarity of the matchline precharge and evaluation [10] [11].
The NOR matchline structure will be discussed in the next section.

2.3.2 NOR Matchline

NOR cells are connected in parallel to form a NOR matchline (ML), shown in Fig. 2.10. The NOR cells can be of both binary and ternary type.
A typical NOR search cycle operates in three phases: searchline precharge, matchline precharge, and matchline evaluation. First, the searchlines are precharged low to disconnect the matchlines from ground by disabling the pulldown paths in each CAM cell. Second, with the pulldown paths disconnected, the transistor precharges the matchlines high. Finally, the searchlines are driven to the search word values, triggering the matchline evaluation phase. In the case of a match, the ML voltage, V_{ML}, stays high as there is no discharge path to ground. In the case of a miss, there is at least one path to ground that discharges the matchline. The matchline sense amplifier (MLSA) senses the voltage on ML, and generates a match result.

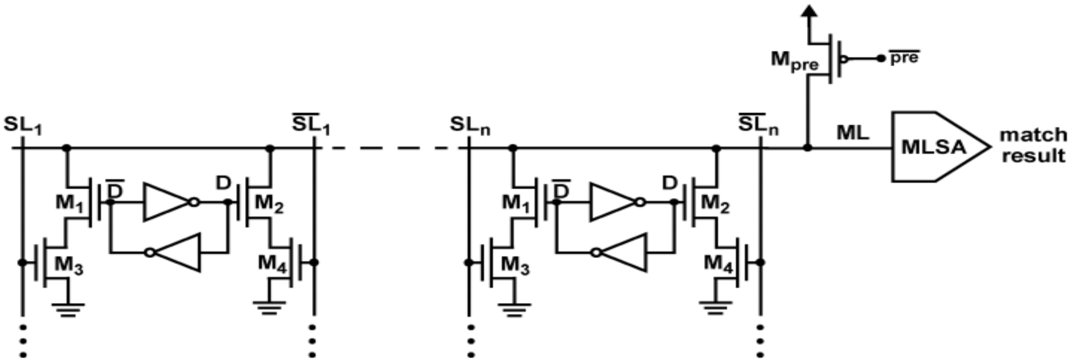


Figure 2.10: NOR Matchline structure [2]

As the NOR matchlines are most prominent in today's CAM architecture so there are several variations of this scheme for evaluating the matched state which will be discussed in the sections following it.

2.4 Matchline sensing scheme

In this section several matchline sensing schemes and their procedure to generate the match result will be discussed. First, the conventional precharge-high scheme will be discussed and other important ML sensing scheme will be followed.

2.4.1 Conventional (Precharge-High) Matchline Sensing

We review the basic operation of the conventional prechargehigh scheme and look at sensing speed, charge sharing, timing control and power consumption.

2.4.1.1 Basic Operation

The basic scheme for sensing the state of the NOR matchline is first to precharge high the matchline and then evaluate by allowing the NOR cells to pull down the matchlines in the case of a miss, or leave the matchline high in the case of a match. Fig. 2.11 shows, in schematic form, an implementation of this matchline-sensing scheme. Fig. 2.12 shows the signal timing which is divided into three phases: SL precharge, ML precharge, and ML evaluation. The operation begins by asserting slpre to precharge the searchlines low, disconnecting all the pull down paths in the NOR cells. With the pull down paths disconnected, the operation continues by asserting to precharge the matchline high. Once the matchline is high, both slpre and $\overline{\text{mlpre}}$ are de-asserted. The ML evaluate 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 entireword, which is output on the MLSA sense-output node, called MLso. If all bits on the matchline match, the matchline will remain high indicating a match for the entire word.

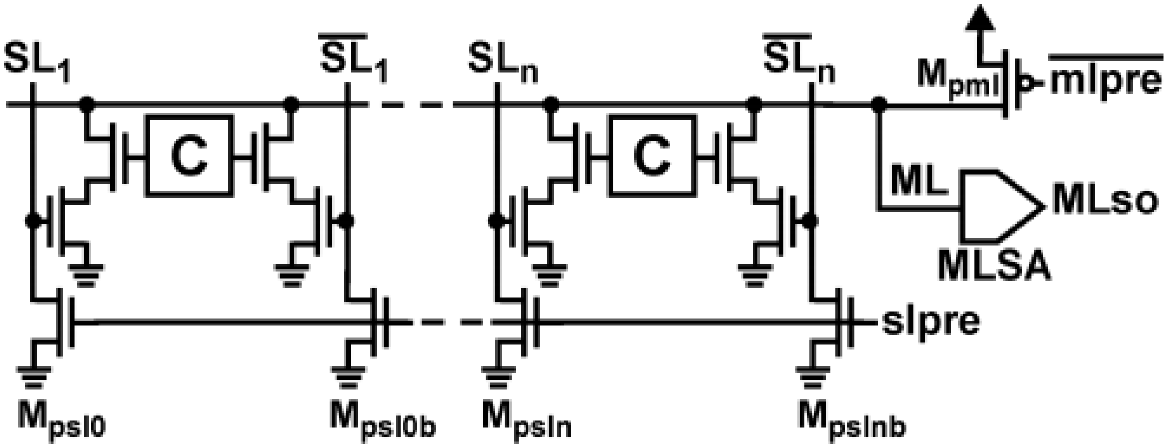


Figure 2.11: Schematic of conventional ML sensing scheme [2]

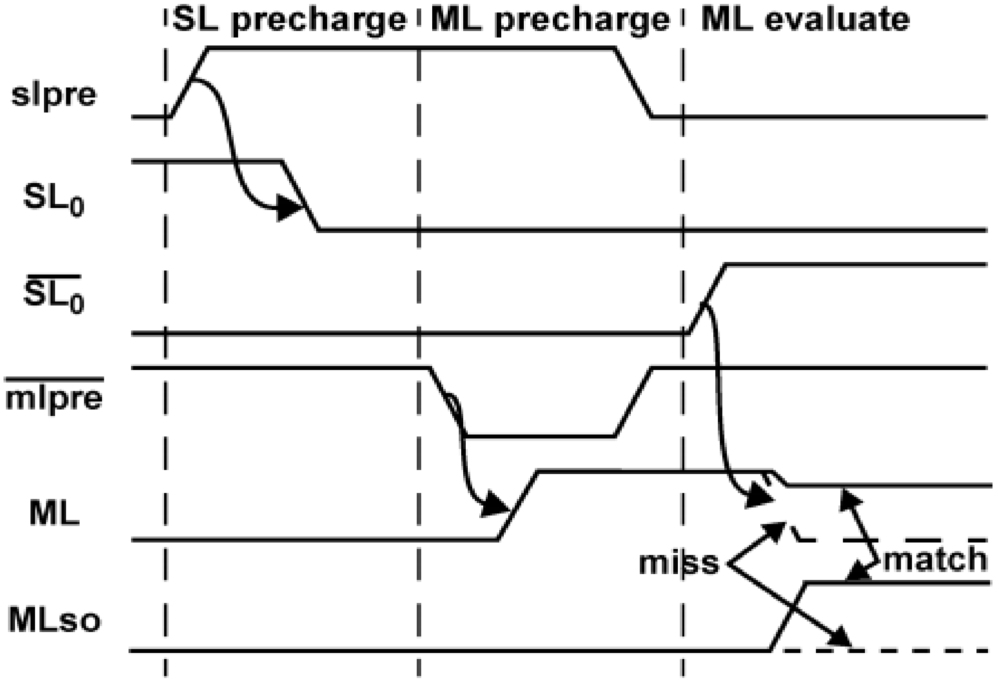


Figure 2.12: Timing diagram showing signal transitions in conventional ML sensing scheme [2]

2.4.1.2 Matchline Model

To facilitate the analysis of matchline sensing schemes, we desire a simple matchline circuit model. As shown in Fig. 2.13, the model for the match state of the matchline is a capacitor C_{ML} and the model for the miss state of the matchline is a capacitor, C_{ML} , in parallel with a pulldown resistor R_{ML}/m , where m is the number of bits that miss on the matchline. The matchline capacitance, C_{ML}, consists of the matchline wiring capacitance, the NOR-cell diffusion capacitance of transistors M_1 and M_2, the diffusion capacitance of precharge transistors, and the input capacitance of the MLSA (Figure 2.4). For misses, the equivalent matchline resistance R_{ML} varies with the number of bits, m , that miss in a word; however, for the purpose of analysis we use the worst-case (largest) resistance, which occurs when there is a 1-bit miss (that is m = 1).

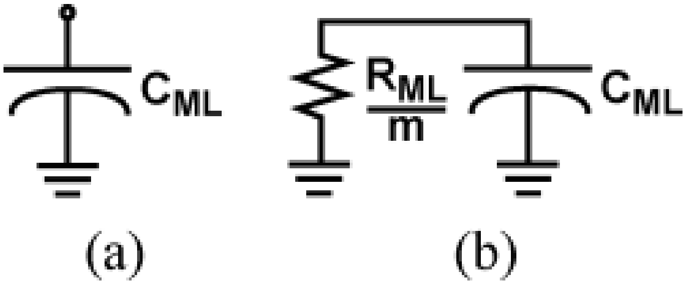


Figure 2.13: Matchline circuit model (a) Match (b) Mismatch [2]

2.4.1.3 Matchline Delay

Using the simple matchline model, we can find the time required to precharge and evaluate the matchline. The time to precharge the matchline (which we define as the 10% to 90% rise time of ML) through the precharge device is given by –

$t_{ML} = 2.2\ \tau_{MLpre}$ … … … (1)

$= 2.2\ R_{EQpre}\, C_{ML}$ … … … (2)

Where 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. The worstcase matchline pulldown resistance occurs when only a single bit misses, activating only a single pulldown path. 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

$t_{ML\text{-}eval} = 0.69\ \tau_{ML\text{-}eval}$ … … … (3)

$= 0.69\ R_{ML}\, C_{ML}$ … … … (4)

Since, typically, in CMOS minimum-sized devices are used in the cell, the pulldown resistance is in the range few kilo ohms. The capacitance, C_{ML}, depends on the number of bits on the matchline, but can be as high as a few hundred fF in CMOS technology.

2.4.1.4 Charge Sharing

There is a potential charge-sharing problem depending on whether the CAM storage bits D and D are connected to the top transistor or the bottom transistor in the pulldown path. Fig. 2.12 shows these two possible configurations of the NOR cell. In the configuration of Fig. 12(a), there is a charge-sharing problem between the matchline, ML, and nodes X_1 and X_2. Charge sharing occurs during matchline evaluation, which occurs immediately after the matchline precharge-high phase. During matchline precharge, SL and SL_c are both at ground. Once the precharge completes, one of the searchlines is activated, depending on the search data, causing either M_1 or M_2 to turn ON. This shares the charge at node X_1 or node X_2 with that of ML, causing the ML voltage, V_{ML}, to drop, even in the case of match, which may lead to a sensing error. To avoid this problem, the configuration of Fig. 2.14(b) is used. Here the stored bit is connected to the top transistors. Since the stored bit is constant during a search operation, charge sharing is eliminated.

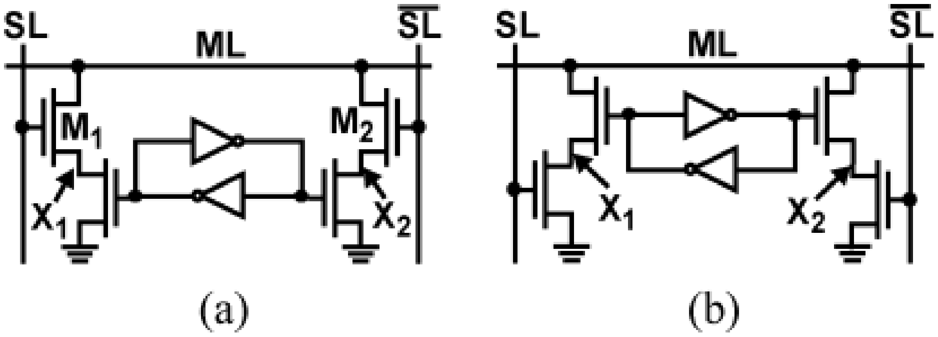


Figure 2.14 (a) charge sharing problem exists (b) charge sharing problem eliminated [2]

2.4.1.5 Power Consumption

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-

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

Where 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 are only a small number of matches we can neglect this power consumption. Accordingly, the overall matchline power consumption of a CAM block with matchlines is-

$P_{ML} = wP_{miss}$ … … … (6)

$= wC_{ML}V_{DD}^{2}f$…. ….. … (7)

2.4.1.6 Replica Control

A standard issue in CAM design, as in all memory design, is how to control the timing of clocked circuits. In the case of the precharge-high scheme, the signals that must be generated are slpre, $\overline{\text{mlpre}}$, and a clock for MLSA. Some authors suggests the use of a replica matchline to generate timing signals in a similar fashion to that used in replica wordlines and bitlines in traditional memory design. The replica matchline controls the length of precharge and evaluation phases during a CAM cycle minimizing the effect of process variation since the replica matchline loading tracks the variations in the rest of the array. Fig. 2.15 is a simplified block diagram showing how a replica matchline may be used to control the precharge and evaluate timing [10] [11]. A replica word is programmed so that its matchline is in the (slowest case) one-bit miss state on every cycle regardless of the input data word. The transition on the replica matchline, $ML_{REPLICA}$, is used to generate the shutoff signal which latches all the matchlines and ends the search cycle.

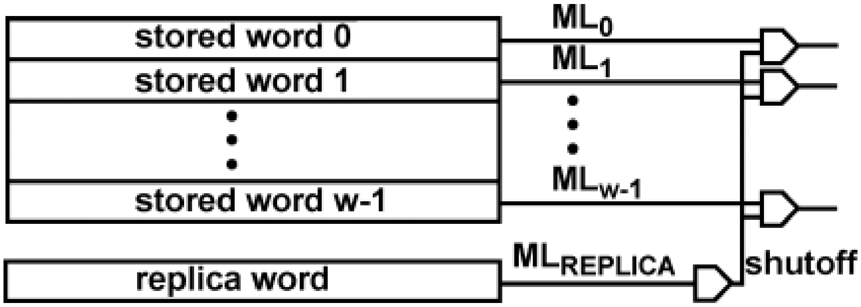


Figure 2.15: Replica matchline for timing control

However the main problem associated with the conventional (prechage high) scheme is the energy consumption. All the MLs are prechaged high first and then it remains high in case of match and discharged in case of mismatch. Usually, the number of match is very less compared to the number of mismatch. So a lot of energy is wasted during the detection of match or mismatch. To reduce this high power consumption different schemes have been propose and some of the prominent schemes will be discussed in the following sections.

2.4.2 Low-Swing Scheme

One method of reducing the ML power consumption, and potentially increasing its speed, is to reduce the ML voltage swing [12], [13]. The reduction of power consumption is linearly proportional to the reduction of the voltage swing, resulting in the modified power equation-

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

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.

Fig. 2.16 is a simplified schematic of the matchline sensing scheme of [12], 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. The precharge operation (assertion of pre signal) charges the tank capacitor to and then uses charge sharing (enabled by eval signal) to dump the charge onto the matchline. The matchline precharge voltage is given by-

$V_{MLpre} = V_{DD}\ C_{tank}/ (C_{tank}+C_{ML})$… … … (9)

and is set to be equal to 300 mV in the design [12]. 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 $V_{LOW}/2$ = 150mV. This sense amplifier is shown in generic form in the figure. A similar charge-sharing matchline scheme was also described in [41]. But the trade off here is the reduction of noise margin and area increment arising from the extra capacitor.

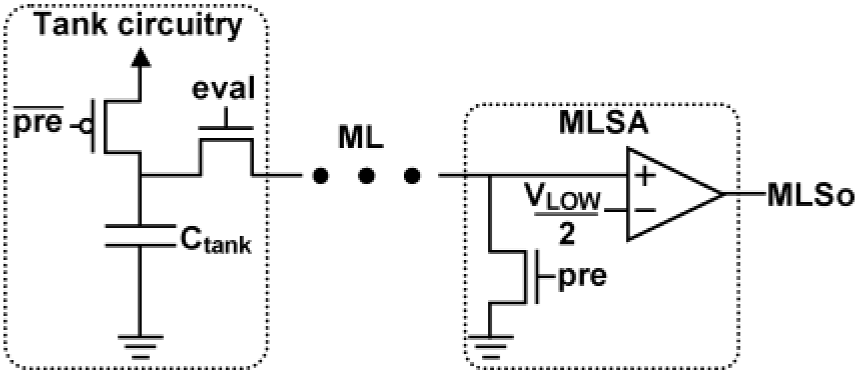


Figure 2.16 Low swing matchline sensing scheme [12]

2.4.3 Selective Prechage Scheme

In conventional matchline sensing scheme same power was allocated to the all the MLs and thus huge energy was wasted, regardless of the specific data pattern, and whether there is a match or a miss. We now examine three schemes that allocate power to matchlines nonuniformly.

The first technique, called selective precharge, performs a match operation on the first few bits of a word before activating the search of the remaining bits. For example, in a 32-bit word, selective precharge initially searches only the first 3 bits and then searches the remaining 29 bits only for words that matched in the first 3 bits. Assuming a uniform random data distribution, the initial 3-bit search should allow only 1/2 words to survive to the second stage saving about 88% of the matchline power. In practice, there are two sources of overhead that limit the power saving. 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.

Fig. 2.17 is a simplified schematic of an example of selective precharge similar to that presented in the original paper [43]. 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 [of Fig. 2.12]. 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 (circuitry not shown) 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 [14], [15]–[19] since it is both simple to implement and can reduce power by a large amount in many CAM applications.

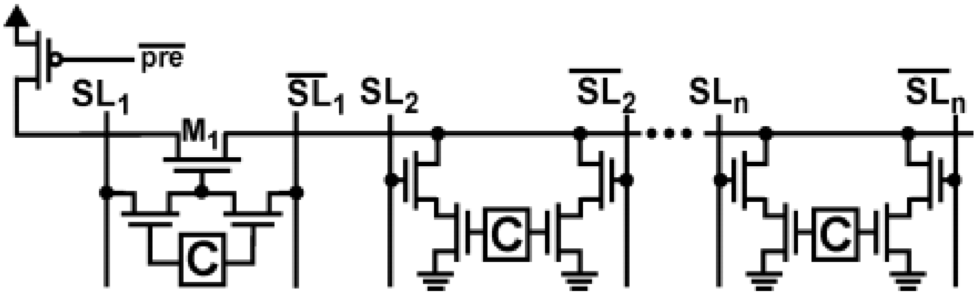


Figure 2.17 Sample implementation of the selective-precharge matchline technique [2] (mixed NAND/NOR structure).

2.4.4 Pipelining Matchline Sensing Scheme

While the selective precharge scheme divides ML into two segments, pipelining scheme divides ML into more segments and perform comparison serially segment by segment [20], [21]. But both of the schemes energy saving depends on the data storage pattern and in worst case scenario there may be no energy saving. Again to do some initial matching some additional circuitry is used here which gives rise to more complexity.

Fig. 2.18(a) shows a simplified schematic of a conventional NOR matchline structure where all cells are connected in parallel. Fig. 2.18(b) shows the same set of cells as in Fig. 2.16(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. The drawbacks of this scheme are the increased latency and the area overhead due to the pipeline stages. 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. Another approach is to segment the matchline so that each individual bit forms a segment [22]. 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 comparison with the result from the previous cell.

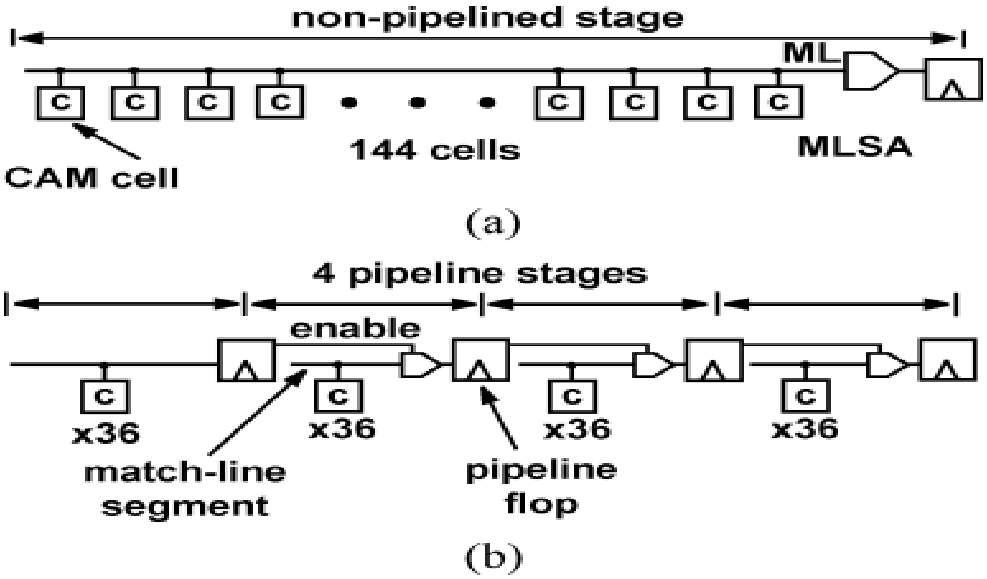


Figure 2.18 Pipelining Match line Sensing Scheme [22]

2.4.5 Current Race Scheme

The ML sensing schemes discussed so far suffers from different types of problem. For example: energy consumption is the most severe problem in conventional ML sensing scheme discussed in section 2.4.1, low voltage margin and increased silicon area consumption is the main problem in low- swing scheme discussed in section 2.4.2 and complexity and worst case energy consumption is the main problem in selective prechage an pipelining ML sensing scheme discussed in section 2.4.3 and 2.4.4 respectively. So far most popular energy reduction scheme is Current Race (CR) scheme shown in Fig 2.19

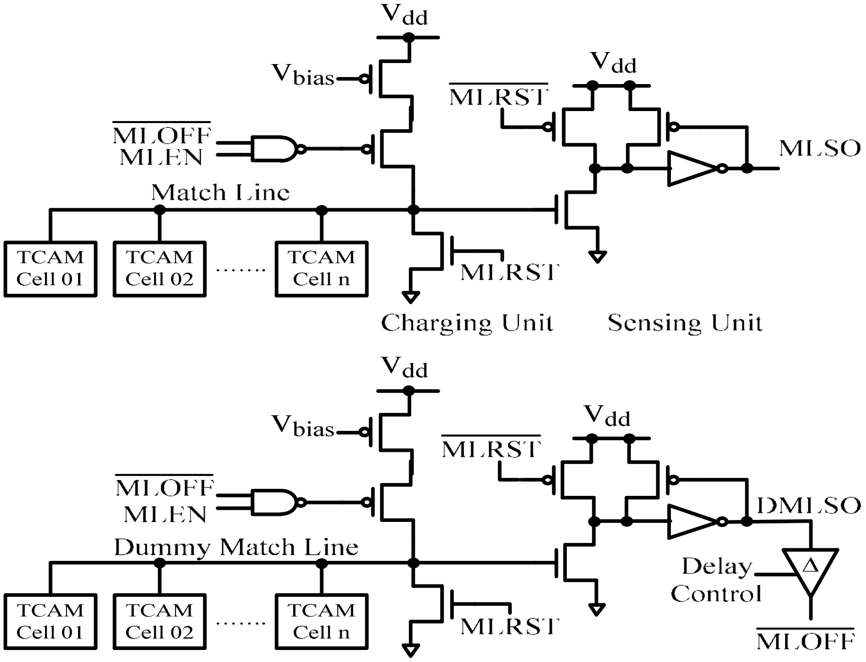


Figure 2.19 current Race scheme [3]

Unlike conventional ML sensing scheme MLs are pre-discharged to ground in CR secheme. MLEN signal initiates the search operation. During the search period MLs are charged towards high in case of match. SLs are not pre discharged to ground in this technique. This reduces the SL switching activity compared to the conventional scheme [3] and saves around 50% SL energy. For fully matched words the corresponding MLs get quickly charged to a threshold which causes the sensing unit to output high at MLSO. For mismatched words, MLs have discharging paths to ground, hence cannot be charged up to that threshold. So, outputs of the associated MLSAs remain low. A dummy word meaning a fully matched word is used to control the charging duration of MLs. As soon as the dummy word output becomes high further charging of all MLs is discontinued by the $\overline{\text{MLOFF}}$ signal. Both matched and mismatched MLs are given same current during the ML charging phase initially in CR scheme and current increases with increase in no of bits mismatching with the search data. It is evident from the simulation results that; current is the least in case of full match and its increasing with the number of bits mismatch. So, large amount of energy is wasted in large number of mismatched MLs in this scheme also.

To eliminate this large power consumption problem some positive feedback in MLSA techniques are used which will be discussed in the sections to follow.

2.4.6 Matchline sensing scheme with positive feedback in MLSA

The main problem in CR scheme was that, the current given to both matched and mismatched MLs was same hence energy consumption was in higher side. We can reduce the power consumption by giving less current to the mismatched MLs compared to the matched MLs. This is one by using positive feedback in MLSA. Some positive feedback based scheme will be discussed in the subsequent subsections to follow. All of these schemes are basically modified version of current race scheme.

2.4.6.1 Mismatch Dependent Matchline Sensing Scheme

A general architecture for the Mismatch Dependent (MD) Matchline Sensing scheme is shown in Fig. 2.20(a). The CAM array consists of a search-word register, which holds an n-bit search word, and m memory rows that store the CAM entries being searched. Also included in the array is a dummy row, which is designed to mimic a matched word.

Fig. 2.20(b) shows a detailed circuit diagram of a single row and the method by which comparison is performed.

To differentiate between a full match (ML0) and a mismatch, all MLs are charged by identical current sources causing their voltages to race toward a sense-voltage threshold. Assuming all current sources are identical and constant, the high-impedance ML0 will develop a voltage higher than that of an ML1 (ML with 1 bit mismatch), while the low-impedance MLn will stay close to GND. The sense circuitry detects this difference in voltage level, and differentiates between an ML0 and an MLn (n>0). Since the voltage level of an MLn (where n is large) stays close to GND over the entire cycle, it would be power efficient to cut the current supplied to it shortly after this is realized. To save current, a current-saving control (CSC) block on each ML (Fig. 2.20) is included to monitor the voltage development on an ML and accordingly reduce the charging current as it becomes evident with time if an ML is mismatched [23].

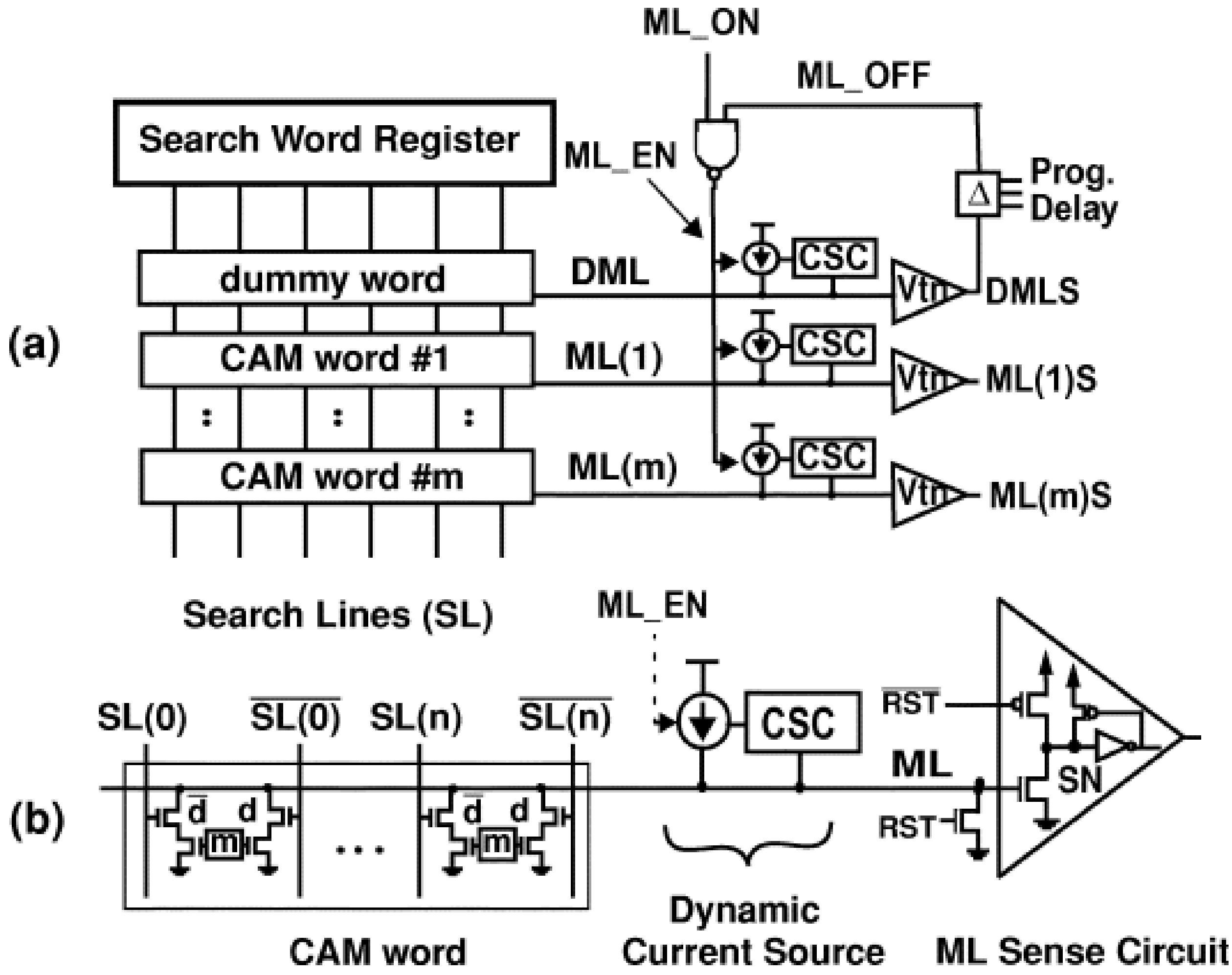


Figure 2.20 Mismatch Dependent Matchline Sensing Scheme [23]

The search operation is performed in two steps. Prior to a search, the RST signal resets all MLs to GND, precharges all sense nodes (SN in each of the ML sense circuits) to V_{DD}, and supplies the new search data on the SLs. Current sources attached to each ML are then enabled, via ML_EN, and start charging all MLs with identical currents. Fig. 2.20 shows how the voltage on an ML ramps depending on the number of mismatches. Since an ML0 has no path to GND, it ramps faster than any MLn (n>0). This current race continues until the voltage on an ML0 crosses the voltage threshold of the ML sense circuit, discharging the SN node through the nMOS device and signaling a match. At this point, all current sources are turned off with a “shut-off” signal from the dummy row, preventing even the closest mismatch (i.e., ML1) from reaching the sense threshold. To generate this signal, a dummy row is designed to always act as a match, independent of the search data. The dummy ML (DML) ramps past the sense threshold at the same time as an ML0 and signals a match with DMLS. By the time this shut-off signal reaches all the sense circuits, any ML0 has crossed the sense threshold and signaled a match, while any MLn (n>0) has stayed below the sense threshold and signaled a miss.

To account for process variations between different MLs, a programmable delay is placed in the path of this global shut-off signal. By shutting off the current supplied to all MLs, this scheme reduces the ML voltage swing, and by doing so decreases the ML power consumption. To achieve further power savings on the ML, the current sources of this scheme have been designed to dynamically allocate less current to MLs with more mismatches. Since the mismatch level of an ML is not known prior to sensing, a small amount of energy is spent for an initial assessment of the state of each ML. To do this, the dynamic current sources start by supplying small identical currents to all MLs. This current develops an ML voltage (V_{ML}) which indicates the probable state of each ML. For example, for a given current, an ML0 develops a higher voltage compared to an ML1, since an ML0 does not leak any charge to GND. On the other hand, an MLn (where n is large) sinks its charge to GND and remains close to GND (as seen in Fig. 2.21). This scheme uses this V_{ML} to allocate more current to probable matches (MLs with high V_{ML}), and cut current to large mismatches (MLs with V_{ML} close to GND). By allocating less current to MLs with a lower V_{ML}, the dynamic current source effectively allocates less power to mismatches, thus saving power. Fig. 2.22 shows the circuit-level implementation of this MD ML sensing scheme.

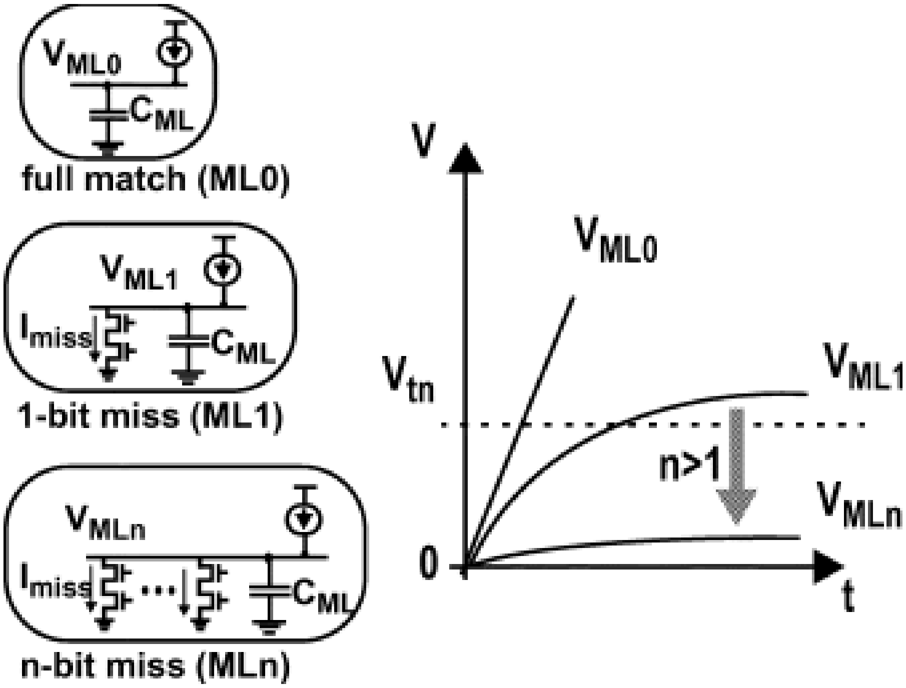


Figure 2.21 Voltage Variation in mismatch dependent ML sensing scheme [23]

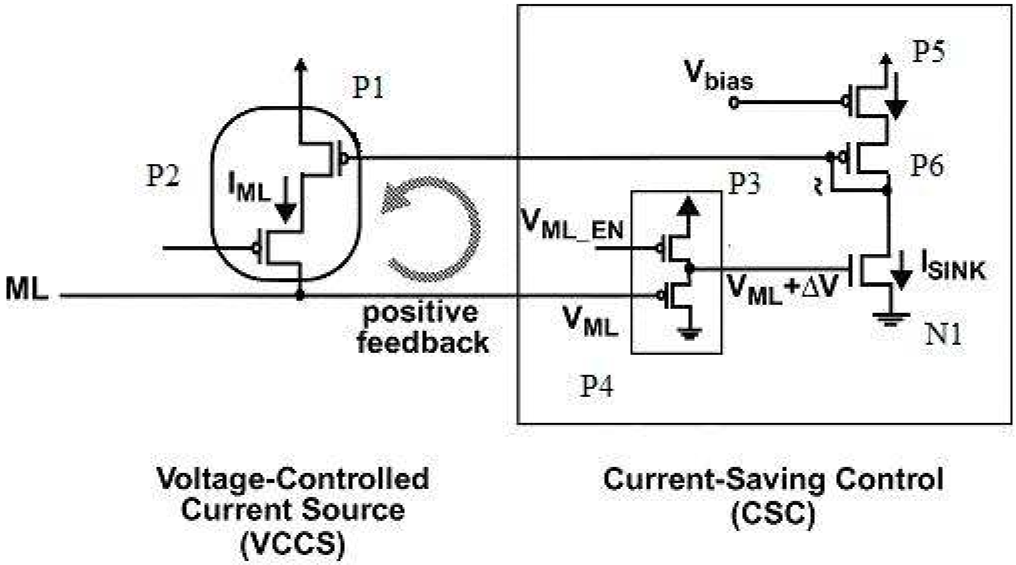


Figure 2.22 Circuit level implementation of mismatch dependent ML sensing scheme [23]

The I_{ML} current in Fig. 2.22 is the feedback current and this current is highest in case of full match and it decreases with the increase in number of mismatchs. Thus current given to the mismatched word is less compared to the matched word and energy consumption is reduced compared to the conventional scheme. The current variation of MD scheme (for 32bit word) is shown in Fig. 2.23. But suffers from the problem that, when idle, there is a dc path from V_{DD} to ground causing static power consumption.

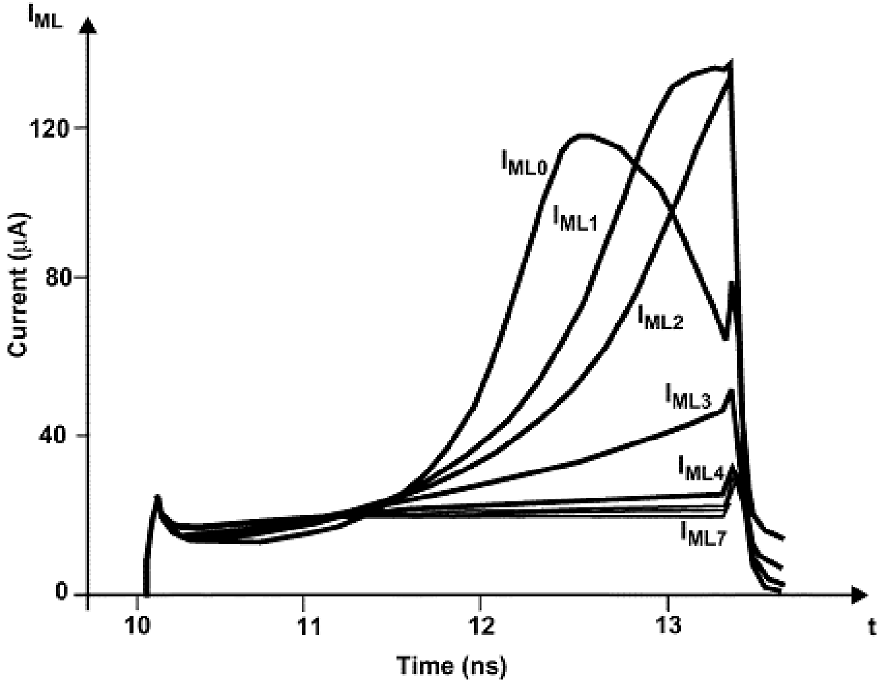


Figure 2.23 Current Variation in mismatch dependent ML sensing scheme [23]

2.4.6.2 Active Feedback Scheme

In order to reduce the energy consumption and delay without sacrificing the voltage margin, there is another positive feedback based ML sensing scheme called active feedback. The circuit diagram of this scheme is shown in Figure 2.24. Transistor N3 operates as a constant current source (I_{FB}) to bias the feedback circuit. The MLEN signal enables the MLSA by activating EN, I_{BIAS} and I_{FB} (Figure 2.24). Initially, all MLs receive the same current from the current sources (I_{BIAS}). As ML_0 charges at a faster rate than ML_k, its P6 source-to-gate voltage (V_{SG_p6}) becomes smaller than that of ML_k. In order to keep the current through P6 constant (I_{FB}), a reduction in V_{SG_p6} is compensated by an increase in the P6 source-to-drain voltage (V_{SD_p6}). Since the source terminal of P6 is at V_{DD} (P7 is acting as a switch), a larger V_{SD_p6} results in a smaller V_{CS}. Thus, the faster charging of ML_0 makes its V_{CS} (V_{CS0}) smaller than that of ML_k (V_{CSk}). As a consequence, ML_0 receives more current and charges more rapidly than ML_k. This positive feedback action continues until DML (emulating ML_0) reaches the MLSA threshold voltage and switches DMLSO ('0'→'1'). This transition flips $\overline{\text{MLOFF}}$ ('1'→'0'), which turns off all of the current sources by switching EN ('0'→'1') [1].

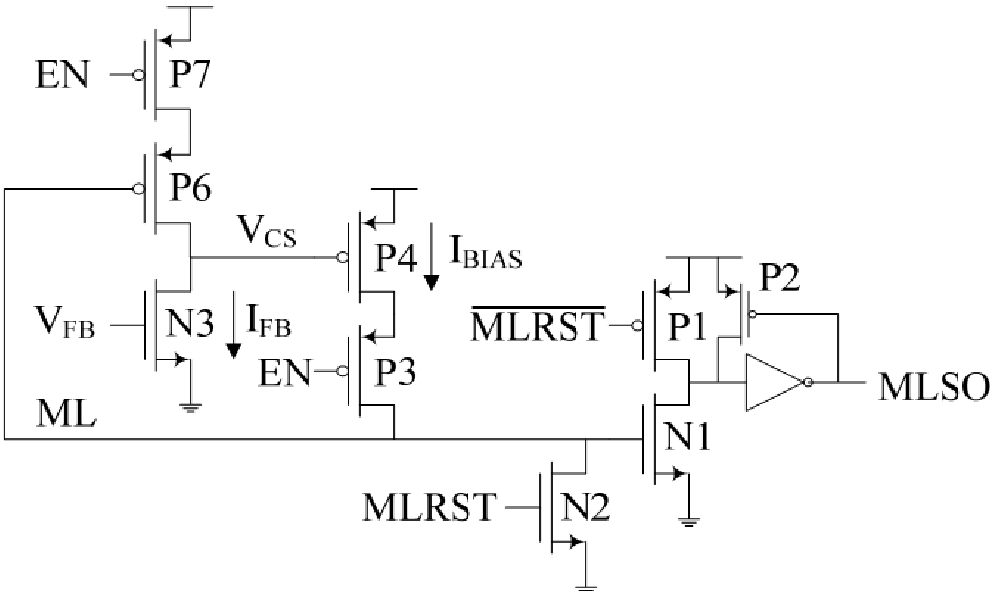


Figure 2.24 Circuit level implementation of Active Feedback ML sensing scheme [1]

The I_{BIAS} current in Fig. 2.24 is the feedback current and this current is highest in case of full match and it decreases with the increase in number of mismatchs. Thus current given to the mismatched word is less compared to the matched word and energy consumption is reduced compared to the conventional scheme. The current variation of MD scheme (for 32bit word) is shown in Fig. 2.25.

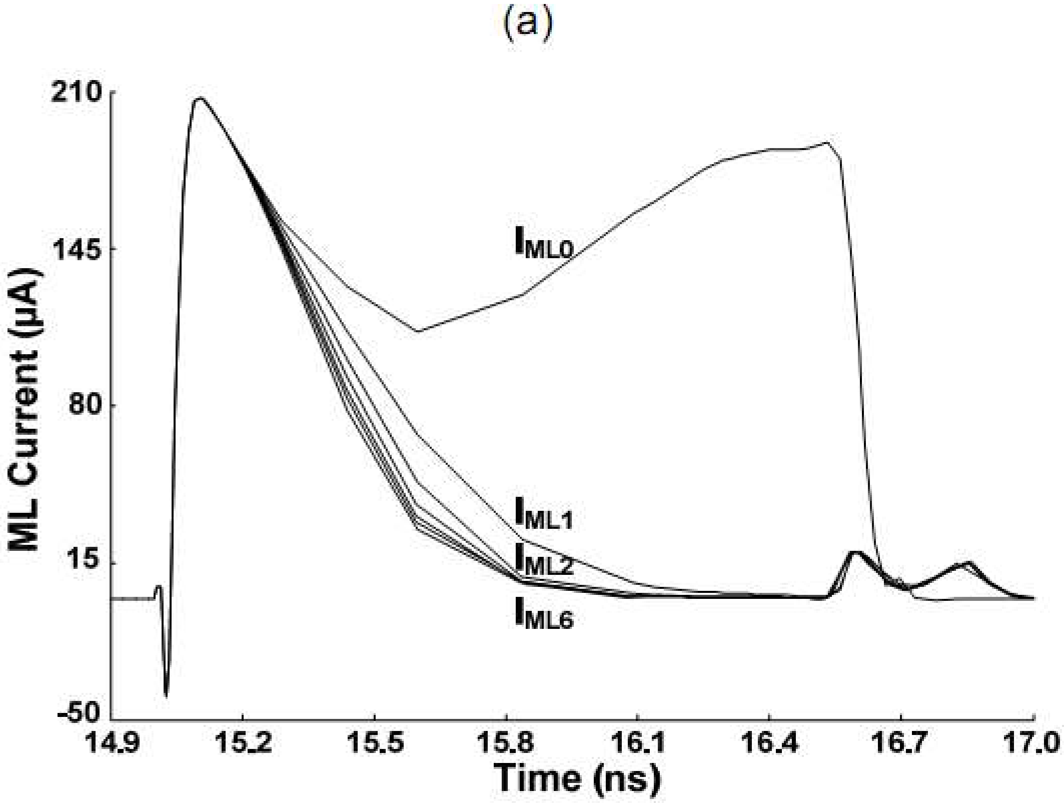


Figure 2.25 Current Variation in active feedback ML sensing scheme [1]

2.4.6.3 Resistive Feedback Scheme

Figure 2.23 shows resistive feedback [70] scheme. It uses an NMOS transistor (N3) in the triode region to decouple the ML and its MLSA. The N3 channel resistance shields the sensing point (SP in Figure 2.26) from the highly capacitive ML. This way the current source (I_{BIAS}) can be sized down to save power without sacrificing the sensing speed. It can be noticed that due to the body effect and the decreasing gate-to-source voltage (V_{GS_N3}), the N3 channel resistance increases when the ML voltage is rising up. Since the ML voltage rises faster as the value of *k* decreases, the increase in the N3 channel resistance is strongly affected by the number of mismatch bits (*k*). For instance, ML_0 would be rising faster than ML_1, which implies that the N3 of ML_0 has a higher resistance to shield the node SP. Since less current is now being diverted to the ML, the node SP charges much faster to reach the threshold voltage. Thus the increasing N3 resistance expedites the arrival of the corresponding MLSO ('0'→'1'). Faster sensing of the dummy word (emulating ML_0) also reduces energy consumption because the faster arrival of DMLSO (and hence $\overline{\text{MLOFF}}$) shuts-down the ML current sources sooner [24]. The charging current of an ML_k is less affected by the N3 resistance because it has a larger V_{GS_N3} and a weaker body effect than ML_0. In other words, the N3 channel resistance creates a level-shift between ML and SP. As the ML voltage increases, the amount of level-shift also increases rapidly, and SP rises to the MLSA threshold voltage more quickly. Therefore, the overall effect is similar to a positive feedback between ML and SP.

The energy and delay of the resistive-feedback MLSA can be further reduced by decreasing V_{RES}. Although the positive feedback action results in a large voltage margin between ML_0 and ML_1, a combination of small V_{RES} and large I_{BIAS} may reduce the voltage margin causing a false match for ML_1. For example, a reduction in V_{RES} increases the N3 channel resistance, which may not be able to divert enough current to ML_1 (and subsequently to GND) particularly if I_{BIAS} is large. As a consequence, the node SP of ML1 may exceed the MLSA threshold voltage indicating a false match (MLSO: '0'→'1').

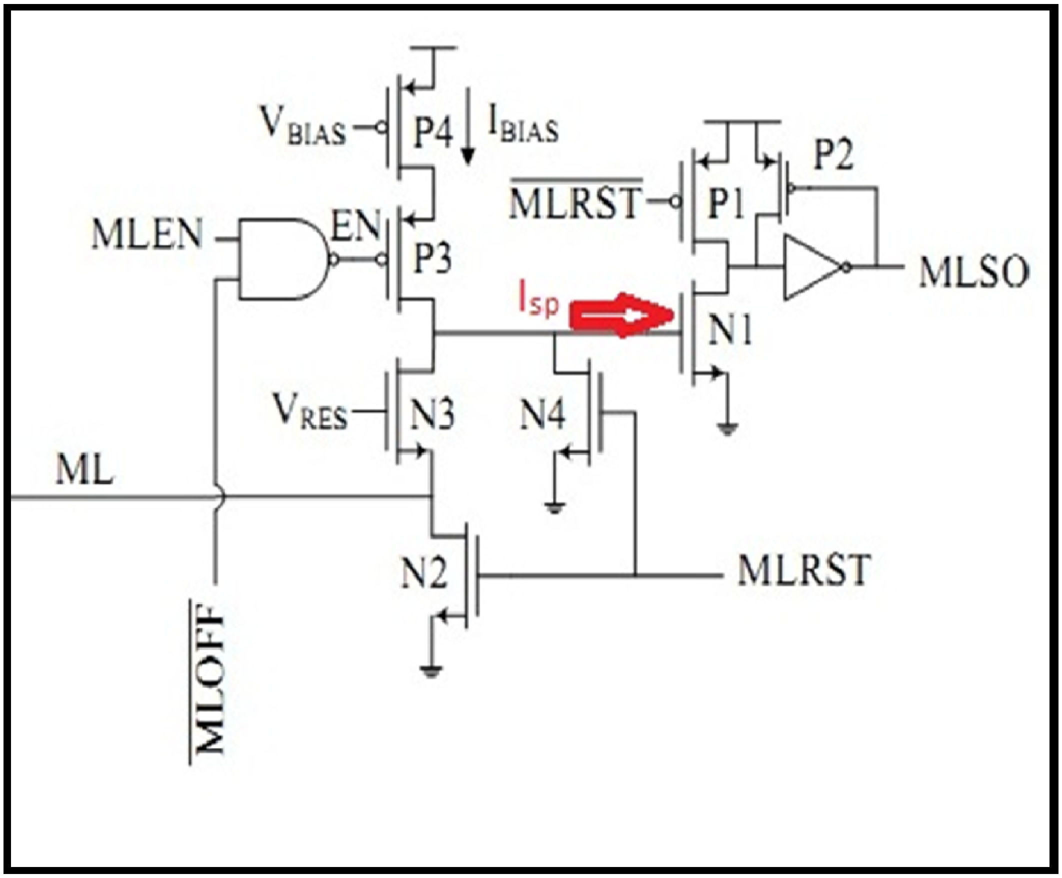


Figure 2.26 Circuit level implementation of Resistive Feedback ML sensing scheme [1] [24]

The I_{SP} current in Fig. 2.26 is the feedback current and this current is highest in case of full match and it decreases with the increase in number of mismatches. Thus current given to the mismatched word is less compared to the matched word and energy consumption is reduced compared to the conventional scheme.

2.5 Searchline sensing scheme

Searchline power consumption depends partly on the matchline scheme. In the following sections, we shall discuss about some prominent seachline sensing scheme such as Conventional Approach, Eliminating seachline prechage and hierarchical search line.

2.5.1 Conventional Approach

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 –

$$P_{SL} = nC_{SL}V_{DD}^{2}f\ldots\ \ldots\ \ldots\ (10)$$

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. As stated earlied that there are two searchlines per bit, which are precharged low and then charged to the appropriate (differential) search-data values. This results in two transitions per searchlines pair, or, equivalently, one transition per searchline. To this power, we must add the power consumption of the drivers. To maintain high speed, we drive the capacitance with drivers consisting of a cascade of inverters sized using exponentially increasing widths [55]. When the searchline drivers are sized to minimize delay, the drivers add an overhead of about 25% to.

2.5.2 Eliminating Searchline Prechage

We can save searchline power by eliminating the SL precharg phase depicted in Fig. 2.11. Eliminating the SL precharge phase reduces the toggling of the searchlines, thus reducing power. As discussed before, matchline-sensing schemes that precharge the matchline low eliminate the need for SL precharge, since enabling the pulldown path in the NOR cell does not interfere with matchline precharge. These schemes directly activate the searchlines with their search data without going through an SL precharge phase. Since, in the typical case, about 50% of the search data bits toggle from cycle to cycle, there is a 50% reduction in searchline power, compared to the precharge-high matchline-sensing schemes that have an SL precharge phase. The equation for the reduced power in this case is

$$P_{SL} = 0.5nC_{SL}V_{DD}^{2}f\ldots\ \ldots\ \ldots\ (11)$$

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.

2.5.3 Hierarchical Searchline

Another method of saving searchline power is to shut off some searchlines (when possible) by using the hierarchical searchline scheme [20], [21], [25], [26]. 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). Fig. 2.27 shows a simplified hierarchical searchline scheme, where the matchlines are pipelined into two segments, and the searchlines are divided into four LSLs per GSL. In the figure, each LSL feeds only a single matchline (for simplicity), but the number of matchlines per LSL can be 64 to 256. The GSLs are active every cycle, but the LSLs are active only when necessary. Activating LSLs is necessary when at least one of the matchlines fed by the LSL is active. In many cases, an LSL will have no active matchlines in a given cycle, hence there is no need to activate the LSL, saving power. Thus, the overall power consumption on the searchlines is-

$$P_{SL} = (C_{GSL} V_{DD}^2 + \alpha C_{LSL} V_{DD}^2)f \dots \dots \dots (12)$$

where is C_{GSL} 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. We see from (12) 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 [56], then the scheme saves power. However, as transistor dimensions scale down, it is expected that wiring capacitance will increase relative to transistor parasitic capacitance. 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. In this case, small-swing signaling on the GSLs can reduce the power of the GSLs compared to that of the full-swing LSLs [49], [50]. This results in the modified searchline power of-

$$P_{SL} = 2n(C_{GSL} V_{DD}^2 + \alpha C_{LSL} V_{DD}^2)f \dots \dots \dots (13)$$

Where V_{LOW} is the low-swing voltage on the GSLs (assuming an externally available power supply). This scheme requires an amplifier to convert the low-swing GSL signal to the full-swing signals on the LSLs. Fortunately, there is only a small number of these amplifiers per searchline, so that the area and power overhead of this extra circuitry is small.

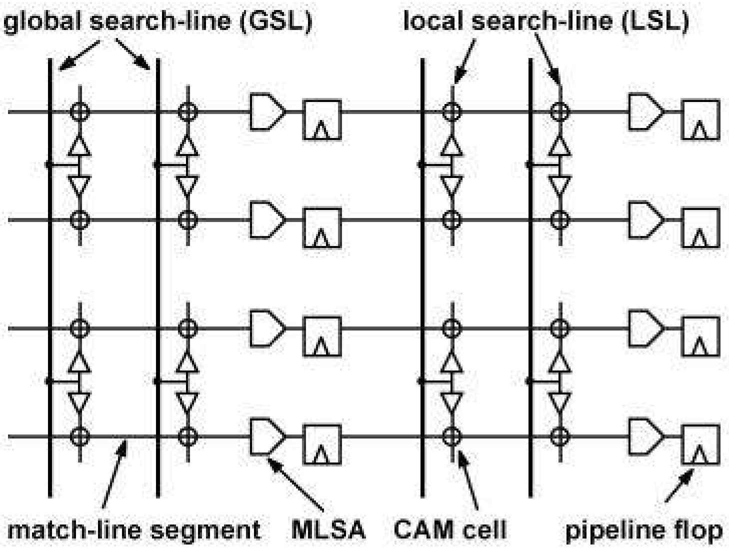


Figure 2.27 Hierarchical Searchline structure [26] [2]

In the next section we shall discuss about our simulation results an performance of different positive feedback based scheme and conventional CR-MLSA.

CHAPTER 3

3. Simulation Results & Performance Analysis

In our simulation we are using 130 nm & 1.2V CMOS technology for HSpice simulation. As mentioned in chapter 2, TCAM actually works in the form of array i.e. multiple words × multiple bits. In our simulation, we have used 32-words×32-bits TCAM arrays and a dummy word of same bit.

We have performed comparative analysis among positive feedback based Current Race (Mismatch dependent, Active feedback, Resistive feedback). We have used CR as a reference design as it is tunable in a wide range of speed and compare the several parameters with it. Although there are a lot number of parameters, we worked only on search time, voltage margin, peak dynamic power and worst case energy consumption. For the sake of fair comparison we have tuned CR in three different cases in such a way that the search time of CR is fully matched with the search time of Mismatch Dependent, Active Feedback and Resistive Feedback respectively and then compared the voltage margin, peak dynamic power and worst case energy among them.

3.1 Mismatch Dependent Scheme:

The circuit diagram of mismatch dependent sensing scheme is given in figure .It has two part, namely charging unit and sensing unit. We need to change the gate parameters of the MOSFETs in the charging unit to make the feedback working. The aspect ratio of the MOSFET of the charging unit is given below.

MOS name	Length (Lmin)	Width(Wmin)
P1	1	3
P2	1	3
P3	10	1
P4	1	18
P5	3/2	1
P6	3/2	1
N1	3/2	3/2

Table 3.1: Aspect ratio of gate parameters of MD scheme

The Vbias was .6v and Lmin and Wmin are minimum feature size (130nm)

In the following subsection we shall discuss about different performance parameters such as search time, voltage margin, peak dynamic power and worst case energy consumption for MD scheme simulated by the aspect ratio mentioned above.

3.1.1 Match line current variation in Mismatch dependent scheme

As stated in section 2.4.6.1 the full matched ML current will be highest and the current will be decreasing with the increase of number of bits mismatches in mismatch dependent scheme. Our simulation result shows exactly the same behavior shown in Fig. 3.1. This also shows that the feedback is working properly here.

In case of current race scheme, the scenario is quite opposite. ML current is lowest in case of match and ML current increases with the increment of number of mismatch (Fig. 3.2). As the number of mismatches is generally higher in real time searching so the power consumption is on higher side in CR scheme and to reduce the power consumption the CR based positive feedback schemes were developed. All the feedback based schemes show similar results in ML current variations compared to the ML current variation in CR scheme.

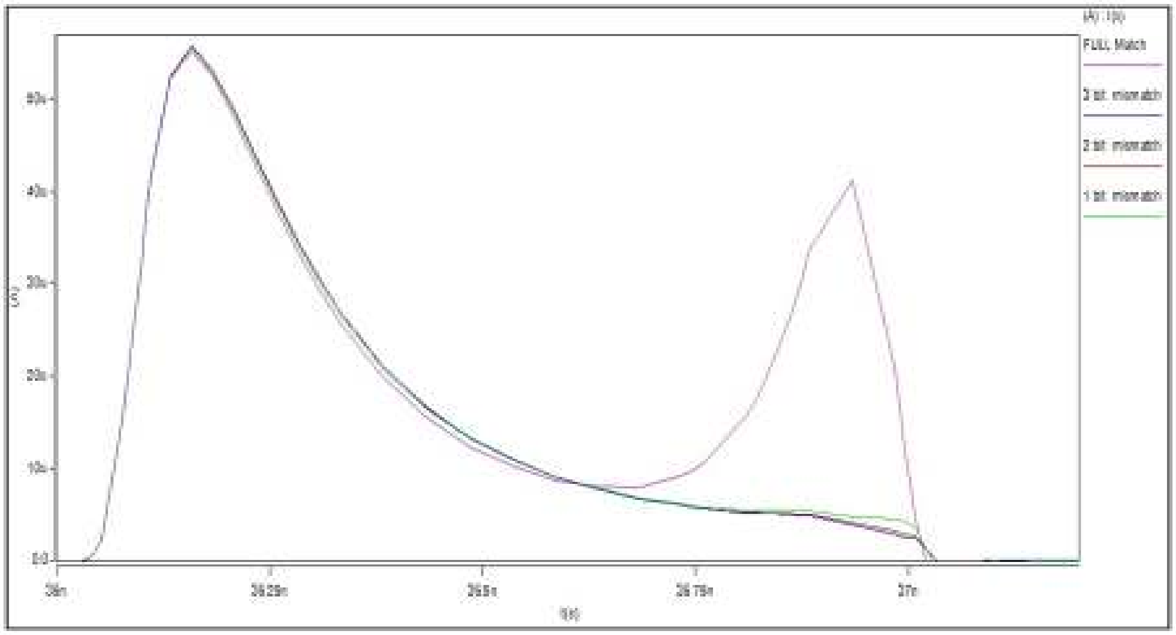


Figure 3.1 ML current variation in mismatch dependent scheme

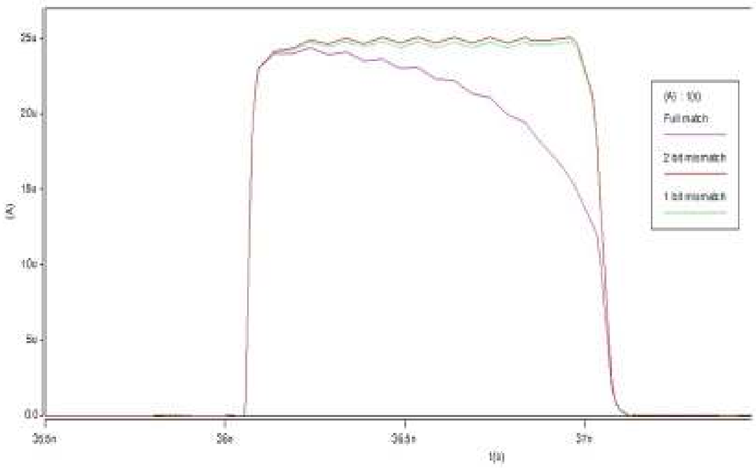


Figure 3.2 ML current variation in current Race scheme

3.1.2 Search time

Search time is defined as the time from 50% (0.6V) of MLEN to 50% (0.6V) of the final output of a matched ML (in case of 130 nm & 1.2 V logic) [3].Using the aspect ratio mention above and bias voltage V_{bias} = .6v ,we found search time 738.99 ps for the mismatch dependent scheme which is shown in the figure 3.1

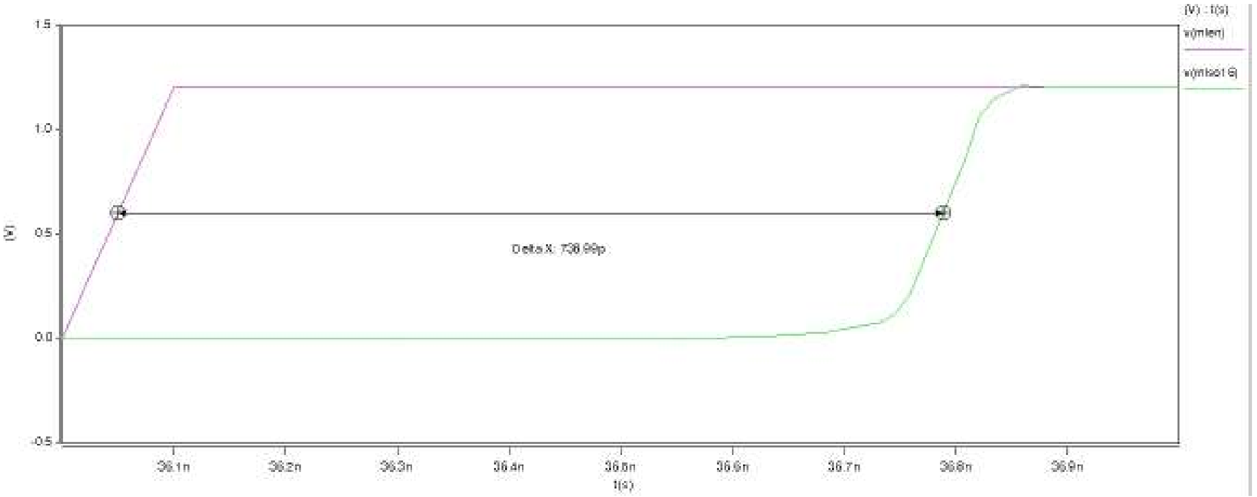


Figure 3.3: Search time for mismatch sensing scheme

For determining the search time we first took the time difference between 50% of MLEN voltage and the 50% of the match MLSO(match line sensing scheme) and in our case the full match was word number sixteen in our routing table (MLSO 16).

In order to make comparison between MD scheme and CR scheme we tuned the gate parameters of CR in such a way that both MD and CR gives same search time that is 738.99ps

The Aspect ratio of the MOSFETs used in CR scheme for equal search time with MD is given below
M9 L=Lmin W='12*Wmin/10'
M10 L=Lmin W='12*Wmin/10'

3.1.3 Voltage margin

Among all types of mismatches one bit mismatch causes maximum resistance in the ML pull-down path since there is only one path through which the ML can discharge. If there are multiple mismatches, multiple pull-down paths exist in parallel and hence the equivalent resistance of ML to ground path is lower which take less time for ML to discharge as the numbers of paths for discharging are increasing. Maximum resistance in the pull-down path means less charge leakage from ML to ground during match evaluation. Hence ML with 1-bit mismatch charges faster than MLs with more than one mismatch. So, 1-bit mismatch is the hardest to detect and it has the highest probability to be detected as a false match. So, there should be a distinct voltage gap between full match ML and 1 bit mismatch ML.

Voltage margin is defined as the difference between the sensing threshold of the sensing unit and the maximum voltage to which a 1-bit mismatched ML is charged [3]. It has been calculated using graphical method shown figure 3.2.This was the voltage margin for mismatch dependent sensing scheme

To find the voltage margin we first determined the crossing point of matched ML and MLSO. Then we determined the maximum voltage up to which the ML of 1 bit mismatched charged and the difference between them was the voltage margin.

In case of mismatch dependent scheme the voltage margin was 392.61 mV and equivalent speed CR has voltage margin 591.86mV which shown in figure 3.2 and figure 3.3 respectively

So from the above comparison we can say that CR has better voltage margin than same speed mismatch dependent sensing scheme

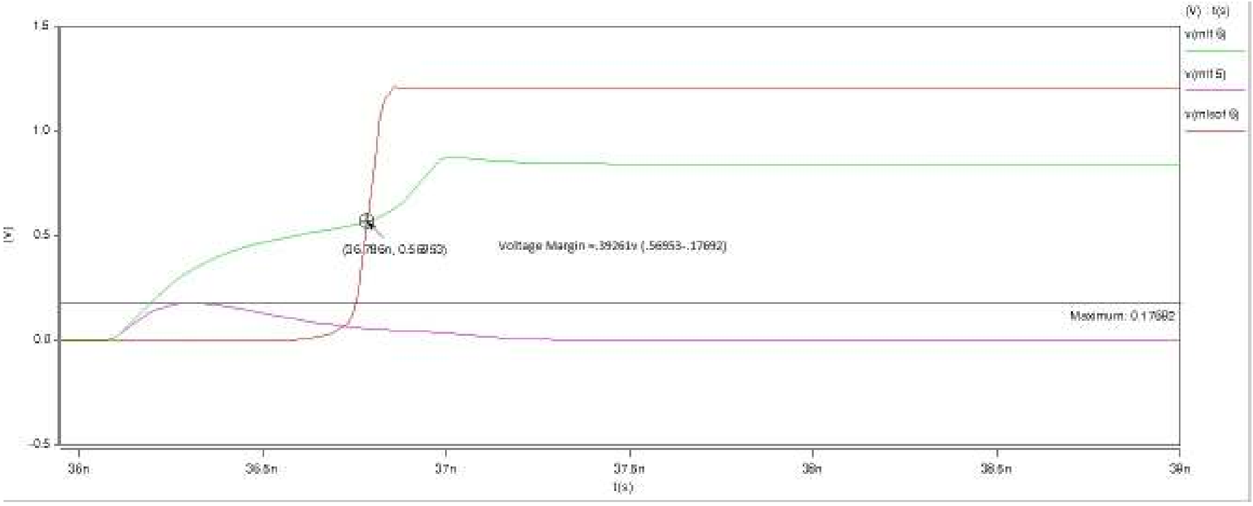


Figure 3.4 : Voltage margin of mismatch dependent sensing Scheme

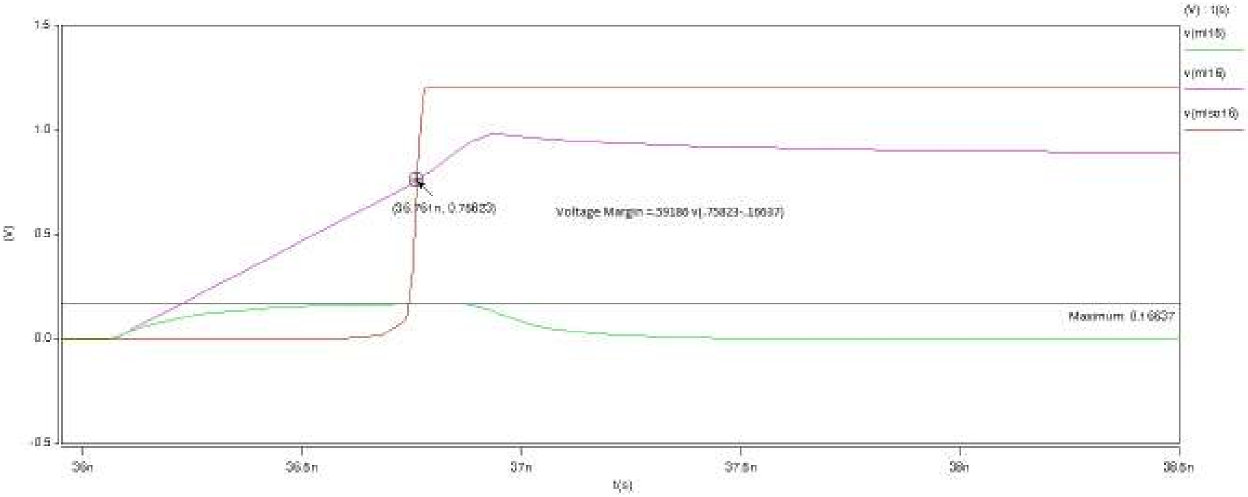


Figure 3.5: Voltage margin of CR of equivalent speed Mismatch dependent sensing Scheme

3.1.4: Peak Dynamic Power and Worst Case Energy Consumption

Peak power consumption with worst case data pattern in routing table is a critical TCAM performance criterion. Many energy saving techniques concentrate on reducing average power consumption but the peak power consumption increases. Increased peak power consumption means more power has to be allocated for the TCAM chip which will be useful only for a short duration but during rest of the search cycle most of that allocated power remains unutilized. So, lower peak power consumption means cheaper supply can be used or the extra power can be used for other components. The worst case routing table used in energy comparison has been used to obtain peak power consumptions of various schemes.

The energy consumption of a scheme is depending on different type of mismatch condition. So, assessing probabilities of different mismatch conditions may be difficult. So, we prefer to calculate on the lower boundary (worst case) of energy in our scheme. Fully matched words consume the highest energy among all types of words. In case of mismatched words energy per word decreases with number of mismatches for (Mismatch dependent, Active feedback, Resistive feedback) while in Current Race this increases with number of mismatches. So, 1-bit mismatch will cause maximum energy consumption for (Mismatch dependent, Active feedback, Resistive feedback) and minimum energy consumption per word if the mismatch is detected in CR. On the other hand the maximum energy consumption occurs in case of full mismatch or in this case a 32 bit mismatch. So we prepared two routing table for worst case energy consumption. For Mismatch dependent, Active feedback, Resistive feedback schemes we prepared the routing table with two full matched and 30 words with 1 bit mismatch to get worst case energy consumption and for CR we prepared the routing table with two full matched and 30 words with full mismatch.

The peak dynamic power and worst case energy for mismatch dependent sensing unit was shown in figure 3.4 and equivalent speed current race in figure 3.5.

Peak dynamic power is the maximum peak in the graph and worst case energy can be calculated from the area under the curve.

The peak dynamic power of Mismatch dependent scheme was 2.4513 mW and CR was 1.53 mW. So in this case equivalent speed CR is better than mismatch dependent scheme.

Again the worst case energy consumption of Mismatch dependent was 1074.85 fJ and CR was 1143.98 fJ. So if we move from the CR to MD we can save 6.04% of total energy.

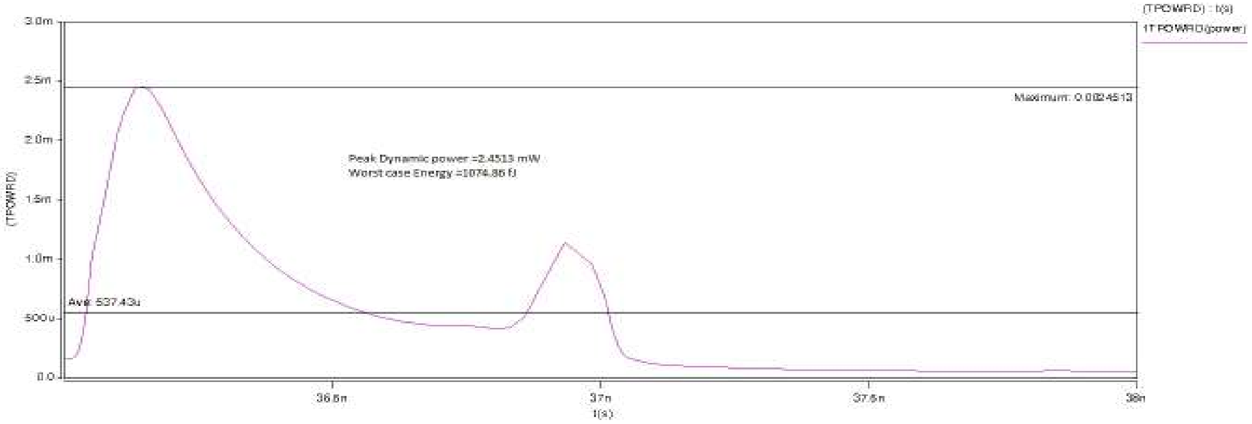


Figure 3.6: Peak Dynamic Power and Worst Case Energy for MD sensing Scheme

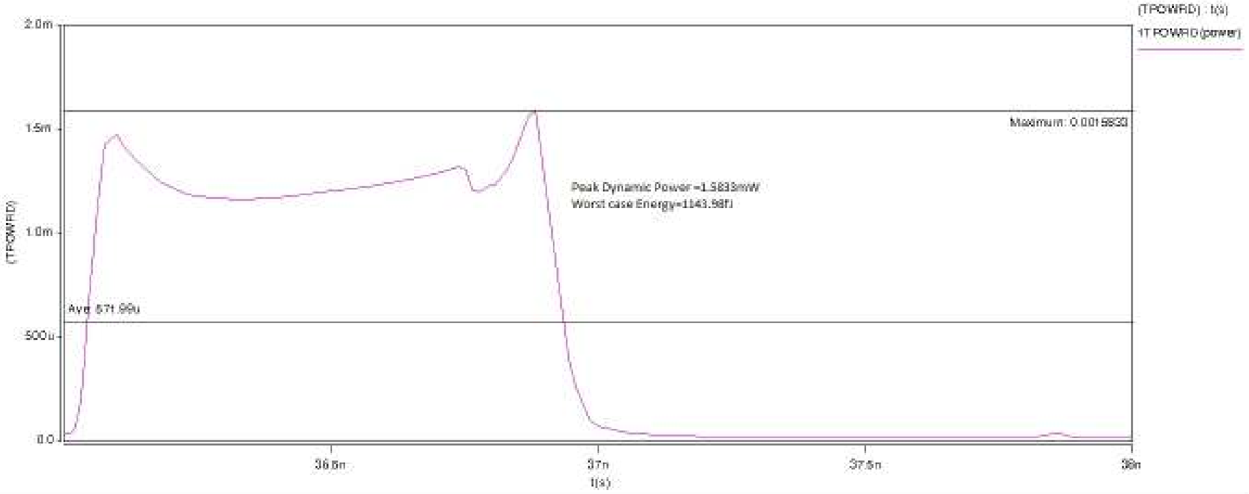


Figure 3.7: Peak Dynamic Power and Worst Case Energy for equivalent speed CR sensing Scheme

So the final comparison between the MD and equivalent speed CR is given below in table 3.2

Comparison Parameter	Mismatch Dependent scheme	Current Race of same speed
Search Time(pS)	738.99	707.47
Voltage Margin(mV)	392.61	591.86
Peak dynamic Power(mW)	2.4513	1.58
Worst case energy consumption(fJ)	1074.85	1143.98

Table 3.2 : Comparison between MD with equivalent speed CR

Although the Mismatch dependent has degraded performance in Voltage margin and Peak dynamic power but it will save 6.04% energy if we use MD instead of CR

3.2 Active Feedback Sensing Scheme:

In this scheme we have to change the aspect ratio of the MOSFET to make the feedback work properly. The gate parameters use in Active feedback is listed below

MOS name	Length (Lmin)	Width(Wmin)
P6	1	7/3
P5	7/3	7/3
N3	11/4	7/3
P4	1	8/3
P3	1	10/3
P1	1	10/3
P2	10/3	5/4
N1	1	10/3
N2	1	10/3

Table 3.3: Aspect ratio of gate parameters of AF scheme

VFB was tuned to 0.55 and Lmin and Wmin are minimum feature size (130nm)
In the following subsection we shall discuss about different performance parameters such as search time, voltage margin, peak dynamic power and worst case energy consumption for AF(Active Feedback) scheme simulated by the aspect ratio mentioned above.

3.2.1 Match line current variation in Active Feedback based scheme

As stated in section 2.4.6.2 the full matched ML current will be highest and the current will be decreasing with the increase of number of bits mismatches in mismatch dependent scheme. Our simulation result shows exactly the same behavior shown in Fig. 3.8. This also shows that the feedback is working properly here.

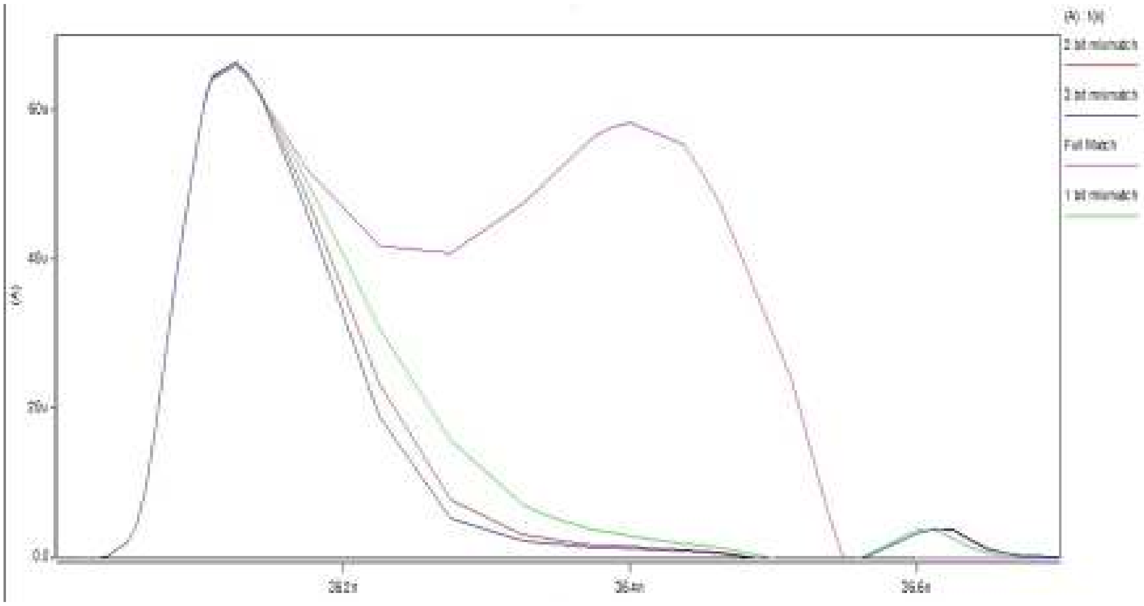


Figure 3.8 Current Variation in active feedback ML sensing scheme

3.2.2 Search time

For the above configuration we got the searching time 319.95ps which shown in the figure 3.6
To get the same speed we tuned CR to below configuration
M9 L=Lmin W='29*Wmin/10'
M10 L=Lmin W='29*Wmin/10'

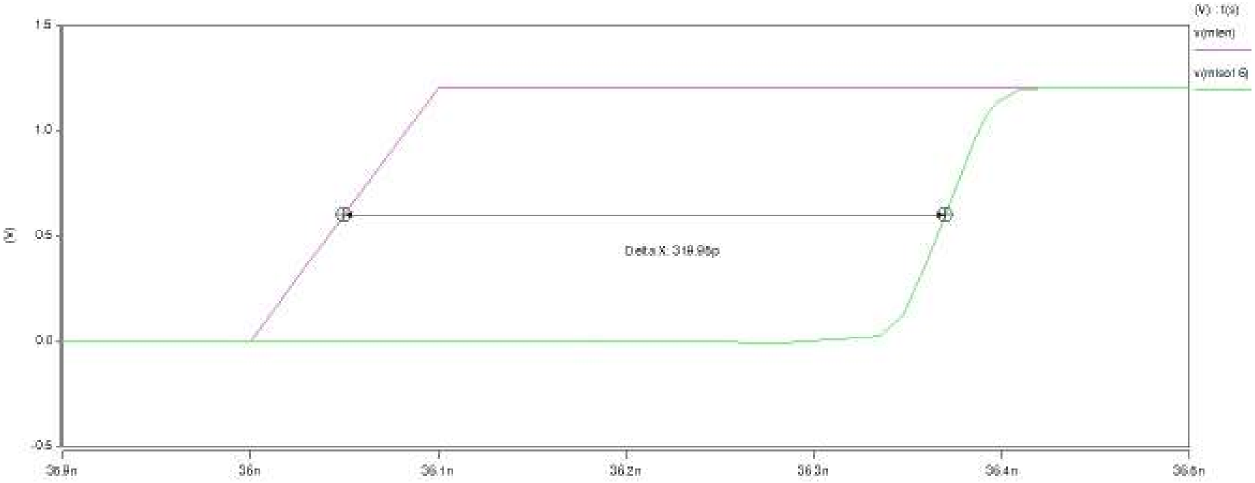


Figure 3.9: Search time for active feedback sensing scheme

3.2.3 Voltage margin

In case of active feedback scheme the voltage margin was 375.02 mV and equivalent speed CR has voltage margin 408.65mV which shown in figure 3.7 and in figure 3.8 respectively
So from the above comparison we can say that CR has better voltage margin than same speed active feedback sensing scheme

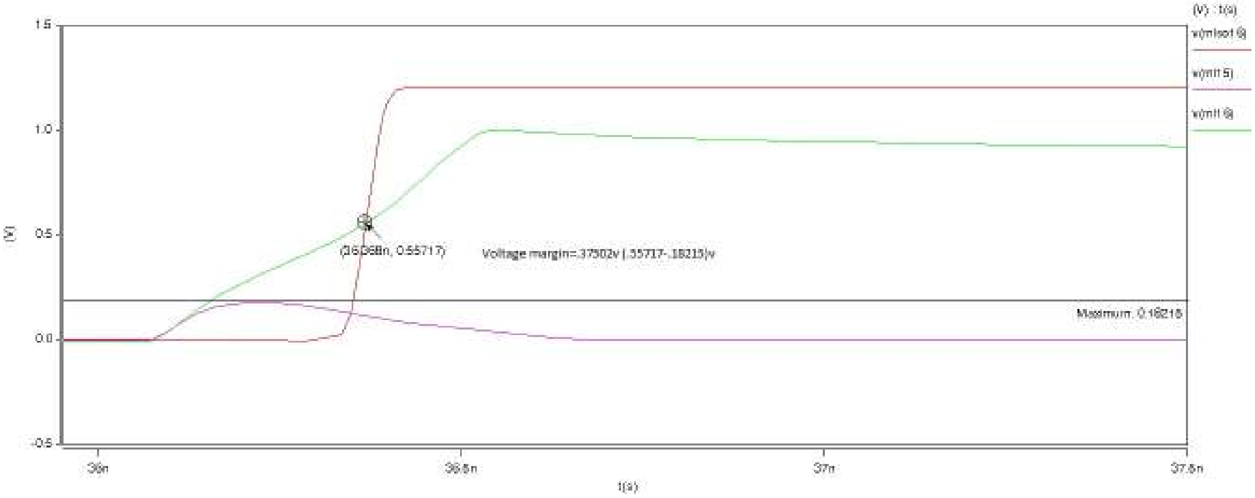


Figure 3.10 : Voltage margin of active feedback sensing Scheme

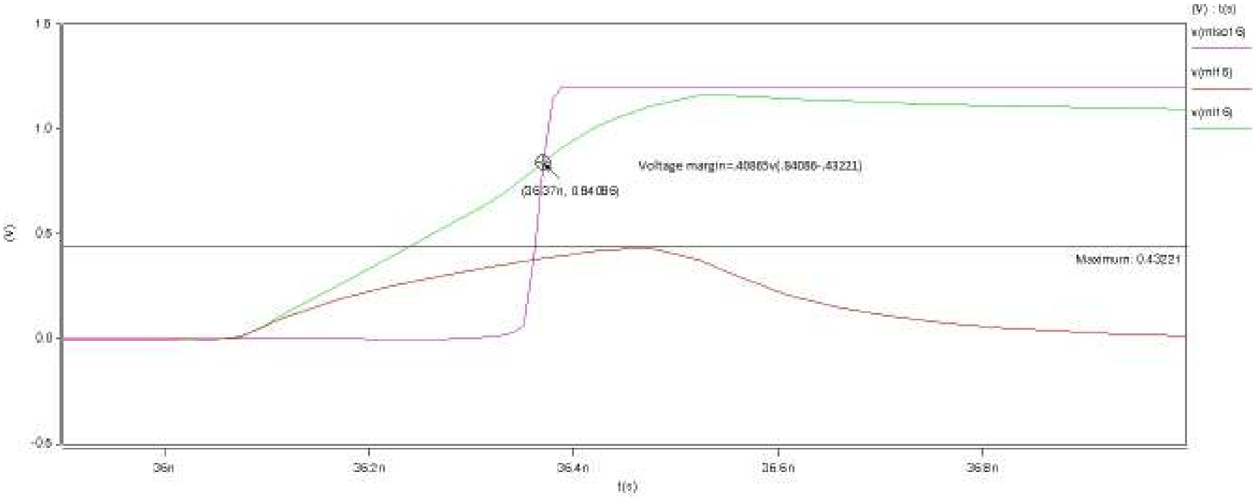


Figure 3.11: Voltage margin of CR of equivalent speed Active feedback sensing Scheme

3.2.4 Peak Dynamic Power and Worst Case Energy Consumption

The peak dynamic power and worst case energy for active feedback sensing unit was shown in figure 3.9 and equivalent speed current race in figure 3.10.

The peak dynamic power of active feedback scheme was 3.49 mW and CR was 3.189 mW. So in this case equivalent speed CR is better than active feedback scheme.

Again the worst case energy consumption of active feedback was 891.9 fJ and CR was 1370.4 fJ. So if we move from the CR to AF we can save 53.64% of total energy.

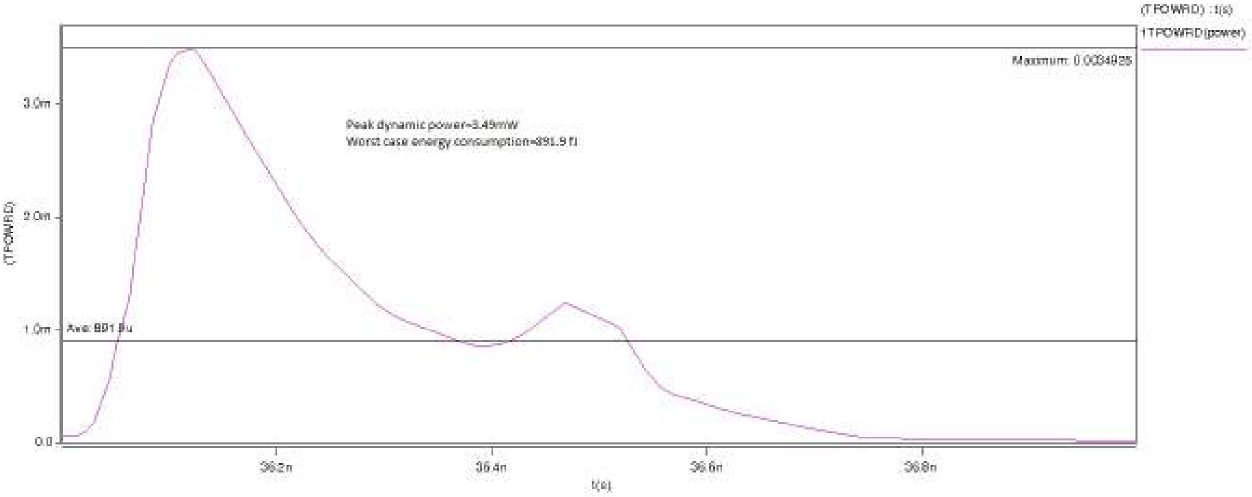


Figure 3.12: Peak Dynamic Power and Worst Case Energy for AF sensing Scheme

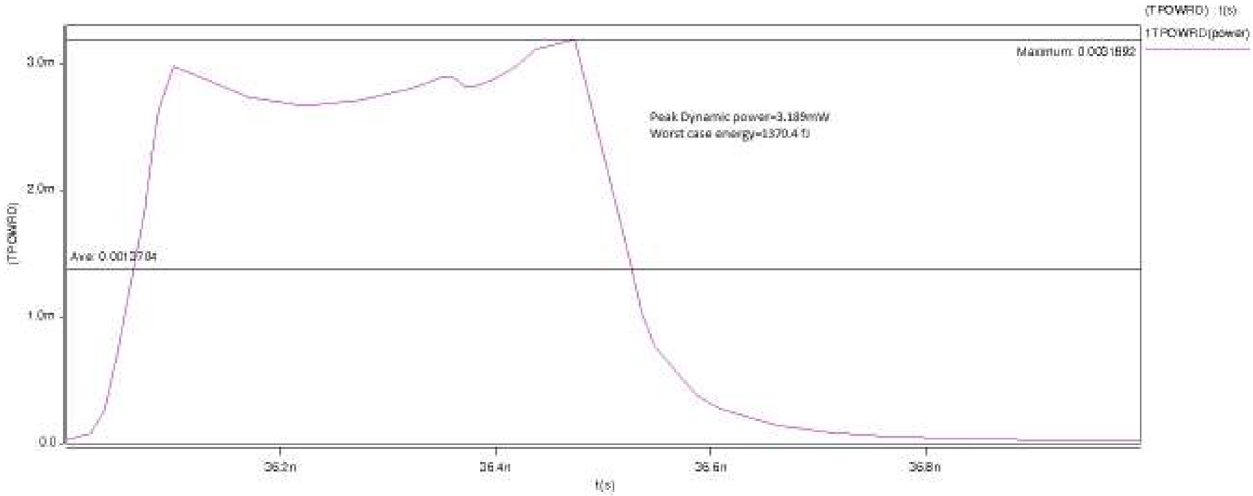


Figure 3.13: Peak Dynamic Power and Worst Case Energy for equivalent speed CR sensing Scheme

So the final comparison between the AF and equivalent speed CR is given below in table 3.4

Comparison Parameter	Active Feedback	Current Race of same speed
Search Time(pS)	319.95	325.56
Voltage Margin(mV)	375.02	408.65
Peak dynamic Power(mW)	3.49	3.189
Worst case energy consumption(fJ)	891.9	1370.4

Table 3.4: Comparison between AF with equivalent speed CR

Although the Active feedback has degraded performance in Voltage margin and Peak dynamic power but it will save 53.64% energy if we use AF instead of CR

3.3 Resistive Feedback

In this scheme we have to change the aspect ratio of the MOSFET to make the feedback work properly. The gate parameters use in Resistive feedback is listed below

MOS name	Length (Lmin)	Width(Wmin)
P4	1	10/3
P3	1	10/3
N3	1	10/3
N4	1	7/3
N2	1	10/3
P1	1	10/3
P2	10/3	11/9
N1	1	10/3

Table 3.5: Aspect ratio of gate parameters of RF scheme

VRES was tuned to 0.76, Vbias was tuned 0.50 and Lmin and Wmin are minimum feature size (130nm)

In the following subsection we shall discuss about different performance parameters such as search time, voltage margin, peak dynamic power and worst case energy consumption for RF(Resistive Feedback) scheme simulated by the aspect ratio mentioned above.

3.3.1 Match line current variation in Resistive Feedback scheme

As stated in section 2.4.6.3 the full matched ML current will be highest and the current will be decreasing with the increase of number of bits mismatches in mismatch dependent scheme. Our simulation result shows exactly the same behavior shown in Fig. 3.14. This also shows that the feedback is working properly here.

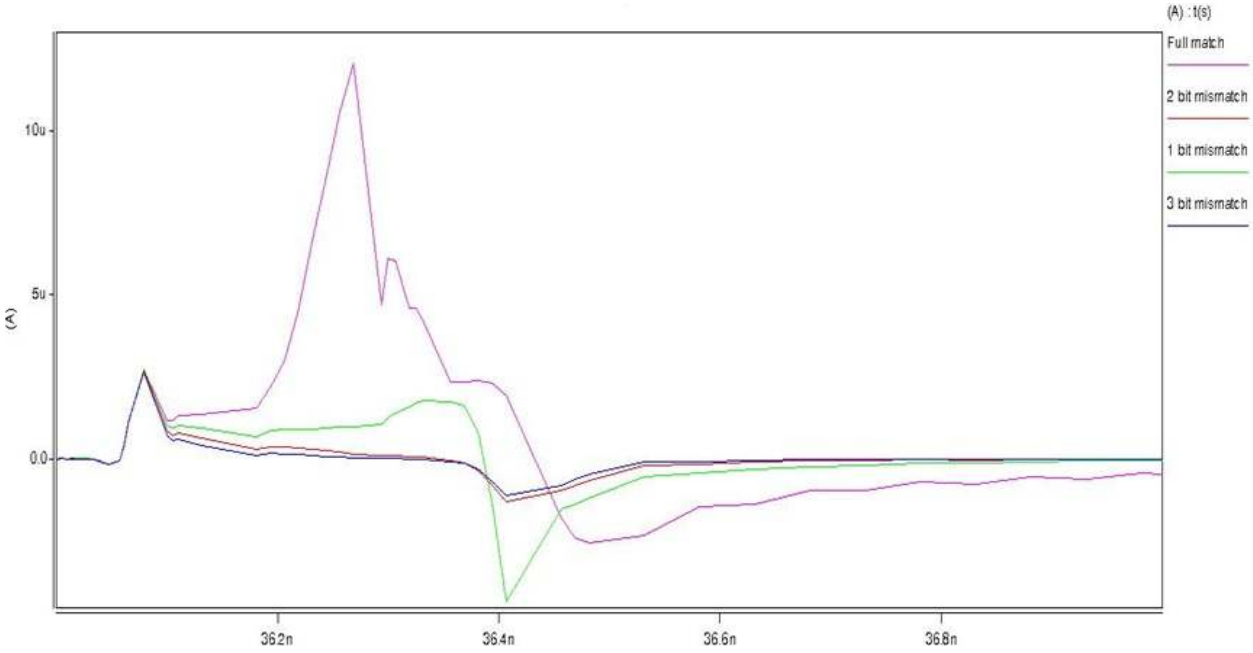


Figure 3.14 Current Variation in resistive feedback ML sensing scheme

3.3.2 Search time

For the above configuration we got the searching time 234.28ps which shown in the figure 3.11
To get the same speed we tuned CR to below configuration
M9 L=Lmin W='4*Wmin'
M10 L=Lmin W='4*Wmin'

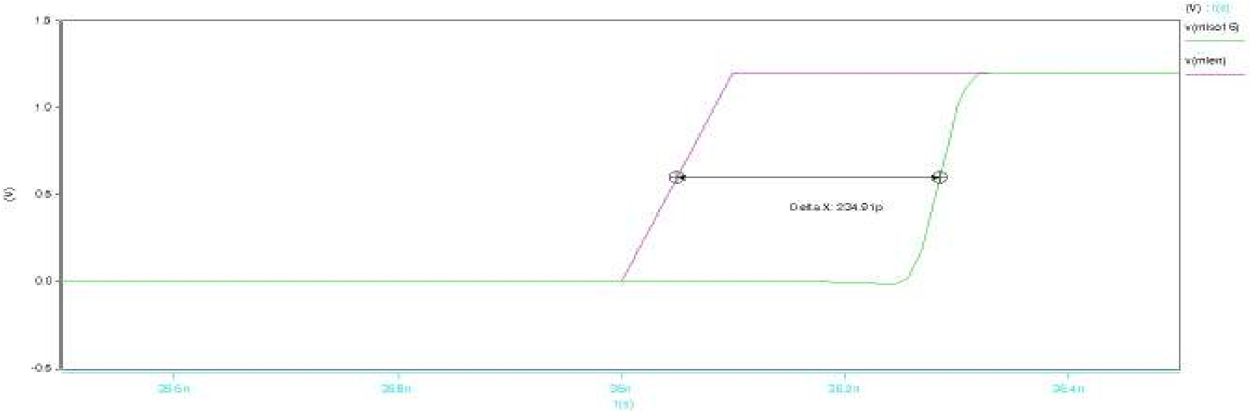


Figure 3.15: Search time for resistive feedback sensing scheme

3.3.3 Voltage margin

In case of resistive feedback scheme the voltage margin was 378.28 mV and equivalent speed CR has voltage margin 268.7mV which shown in figure 3.12 and in figure 3.13 respectively
So from the above comparison we can say that resistive feedback has better voltage margin than same speed CR sensing scheme

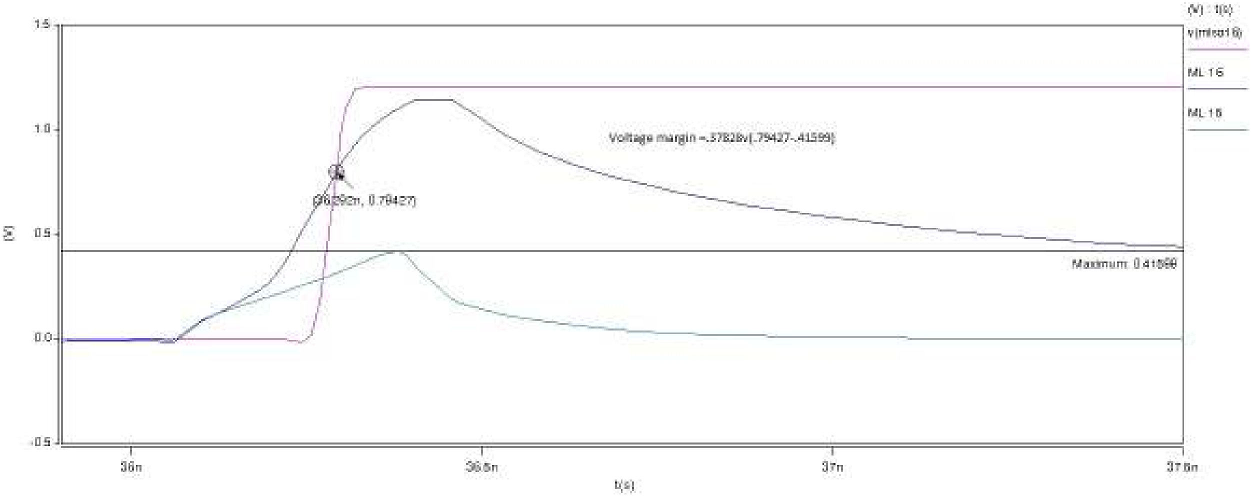


Figure 3.16: Voltage margin of resistive feedback sensing Scheme

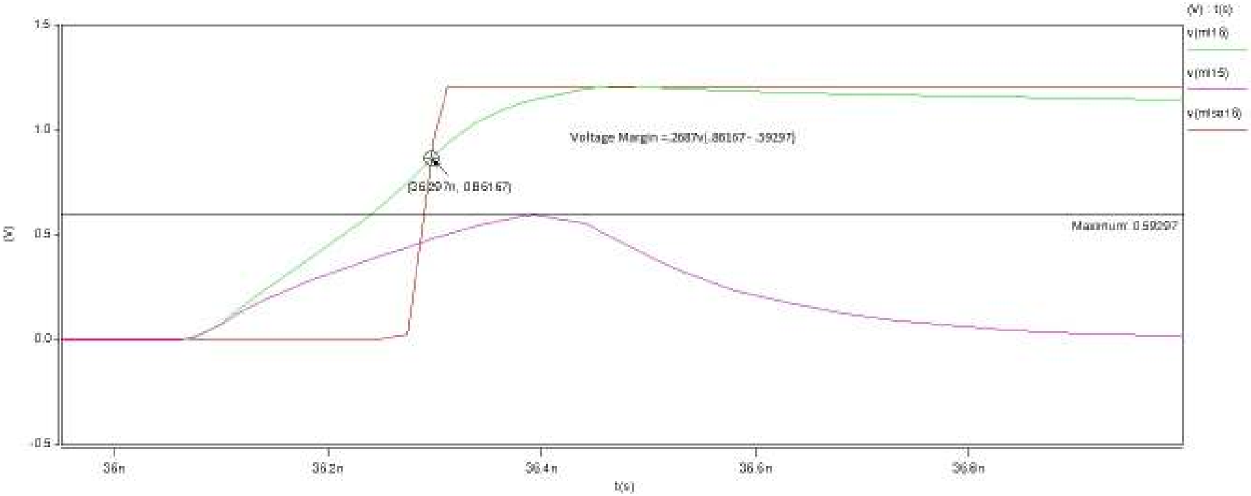


Figure 3.17: Voltage margin of CR of equivalent speed Resistive feedback sensing Scheme

3.3.4: Peak Dynamic Power and Worst Case Energy Consumption

The peak dynamic power and worst case energy for resistive feedback sensing unit was shown in figure 3.14 and equivalent speed current race in figure 3.15.

The peak dynamic power of resistive feedback scheme was 3.533 mW and CR was 4.34 mW. So in this case resistive feedback is better than equivalent speed CR scheme.

Again the worst case energy consumption of resistive feedback was 936.74 fJ and CR was 1581 fJ. So if we move from the CR to AF we can save 68.77% of total energy.

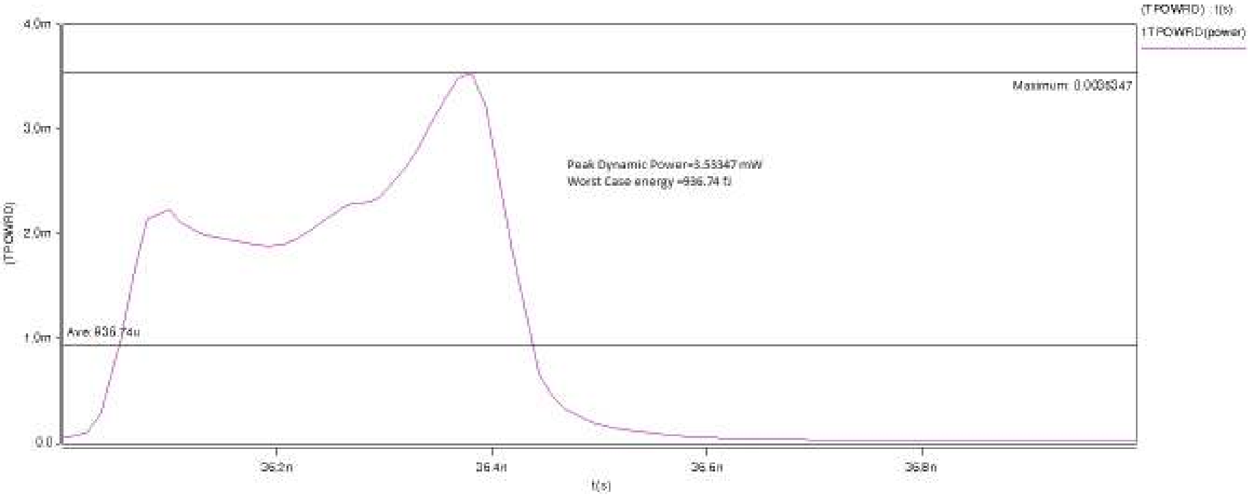


Figure 3.18: Peak Dynamic Power and Worst Case Energy for RF sensing Scheme

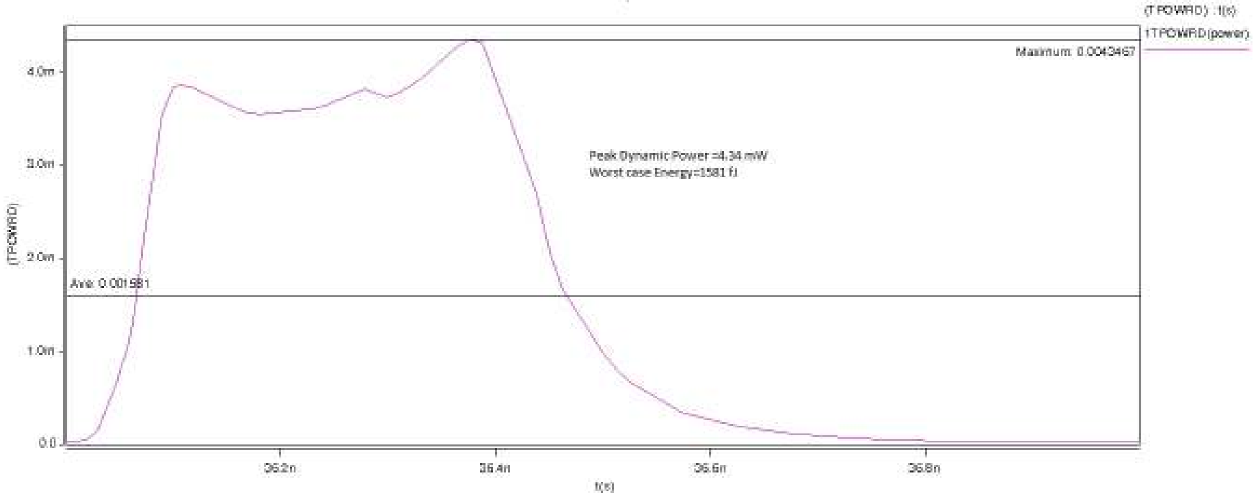


Figure 3.19: Peak Dynamic Power and Worst Case Energy for equivalent speed CR sensing Scheme

So the final comparison between the RF and equivalent speed CR is given below in table 3.6

Comparison Parameter	**Resistive Feedback**	**Current Race of same speed**
Search Time(pS)	234.91	238.36
Voltage Margin(mV)	378.28	268.7
Peak dynamic Power(mW)	3.533	4.3460
Worst case energy consumption(fJ)	936.74	1581.00

Table 3.6: Comparison between RF with equivalent speed CR

If we use RF instead of CR, then it not only improves performance in Voltage margin and Peak dynamic power but also saves 68.77% of energy.

3.4 Final Comparison

Comparison Parameter	**Mismatch** Dependent	**Active Feedback**	**Resistive Feedback**
Search Time(pS)	738.99	319.95	***234.91***
Voltage Margin(mV)	***392.61***	375.02	378.28
Peak dynamic Power(mW)	***2.4513***	3.49	3.5334
Worst case energy consumption(fJ)	1074.85	***891.9***	936.74

Table 3.7: Final Comparison

Among the three positive feedbacks based scheme, resistive feedback is superior in term of search speed but it has a little bit degraded property in voltage margin, peak dynamic power and worst case energy consumption

In term of voltage margin and peak dynamic power mismatch dependent is superior although it is worst in term of search speed and worst case energy consumption.

In term of worst case energy consumption active feedback is superior and it is medium in search speed and peak dynamic power but worst in case of voltage margin.

CHAPTER-4

4. Conclusion

TCAMs are gaining importance in high-speed lookup-intensive applications. However, the high power consumption of TCAMs is limiting their popularity and versatility. A significant portion of the TCAM power is consumed by MLSAs for match detection. We discussed three MLSAs that apply positive feedback for power reduction in ML sensing. Instead of providing the same current to all MLs, these MLSAs modulate the ML current source such that a larger current flows into the ML_0 (match) and a smaller current flows into the ML_K (mismatch). We tuned the gate parameters in such a way that the combination maximizes the speed, noise margin and minimizes the energy consumption of ML sensing. The above statement implies that the proposed MLSAs do not necessarily reduce the power consumption (Power = Energy x Frequency) because the increase in frequency offsets the reduction in energy. However, power reduction can be achieved if the frequency is not increased. Simulation results of the mismatch dependent MLSA (in 0.12µm CMOS technology) show about 6-7% reduction in energy over the conventional CR-MLSA. The active feedback scheme (in 0.12µm CMOS technology) show about 53-54% reduction in energy over the conventional CR-MLSA. But the energy reduction was best in resistive feedback scheme and there was about 68-69% reduction in energy over the conventional CR-MLSA. In addition, all three positive feedback based schemes improve the robustness of ML sensing by feeding less current to ML_1 and more current to ML_0.

In case of Mismatch dependant and active feedback scheme this energy saving comes in expense of reduced voltage margin and peak dynamic power. The worst case energy consumption is relatively less in these schemes compared to conventional CR-MLSA.

But the resistive feedback scheme shows no degradation of voltage margin and peak dynamic power compared to conventional CR-MLSA. Here also the worst case energy consumption is relatively less compared to conventional CR-MLSA.

We have found that among all three positive feedback based schemes resistive feedback provides with the best search time. The voltage margin and peak dynamic power is the best in case of Mismatch dependent scheme. Worst case energy consumption is least in Active feedback scheme among all three positive feedback based scheme.

So, our suggestion goes like this if the higher search speed is our main criteria then we must go for resistive feedback scheme. It the router is exposed to a noisy environment then we should go for the scheme which shows the best voltage margin which is the mismatch dependent scheme. If the energy consumption and heating of the device is of concern then we should opt for active feedback based scheme.

Future research can be carried out in understanding the search algorithms and applying that information to reduce the switching activity in SLs. In addition, innovative circuit techniques can be developed for the comparison logic to reduce the voltage swing and capacitance of SLs. Since large cell area is also a serious concern for large-capacity TCAMs, future research can also include the design of low-area TCAM cells that are compatible with the standard CMOS process. Nonvolatile TCAMs can also be explored if the process technology supports the integration of high-speed logic and non-volatile memory.

References

[1] Nitin Mohan, “Low-Power High-Performance Ternary Content Addressable Memory Circuits”, A PhD thesis presented to the University of Waterloo.

[2] K. Pagiamtzis and A. Sheikholeslami, “Content addressable memory (CAM) circuits and architectures: a tutorial and survey,” *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, March 2006.

[3] Syed Iftekhar Ali and M S Islam, “An Energy Efficient Design of High-Speed Ternary CAM Using Match-Line Segmentation and Resistive Feedback in Sense Amplifier”, JOURNAL OF COMPUTERS, VOL. 7, NO. 3, MARCH 2012, pp 567-577

[4] S. Choi, K. Sohn, M.-W. Lee, S. Kim, H.-M. Choi, D. Kim, U.-R. Cho, H.-G. Byun,Y.-S. Shin, and H.-J. Yoo, “A 0.7 fJ/bit/search, 2.2 ns search time hybrid type TCAM architecture,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2004, pp. 498–499.

[5] S. Choi, K. Sohn, and H.-J. Yoo, “A 0.7 fJ/bit/search, 2.2-ns search time hybrid-type TCAM architecture,” *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 254–260, Jan. 2005.

[6] H. Miyatake, M. Tanaka, and Y. Mori, “A design for high-speed lowpower CMOS fully parallel content-addressable memory macros,” *IEEE J. Solid-State Circuits*, vol. 36, no. 6, pp. 956–968, Jun. 2001.

[7] S. Liu, F. Wu, and J. B. Kuo, “A novel low-voltage content-addressable memory (CAM) cell with a fast tag-compare capability using partially depleted (PD) SOI CMOS dynamic-threshold (DTMOS) techniques,” *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 712–716, Apr. 2001.

[8] G. Thirugnanam, N. Vijaykrishnan, and M. J. Irwin, “A novel low power CAM design,” in *Proc. 14th Annu. IEEE ASIC/SOC Conf.*, 2001, pp. 198–202.

[9] G. Thirugnanam, N. Vijaykrishnan, and M. J. Irwin, “A novel low power CAM design,” in *Proc. 14th Annu. IEEE ASIC/SOC Conf.*, 2001, pp. 198–202.

[10] K. J. Schultz, F. Shafai, G. F. R. Gibson, A. G. Bluschke, and D. E. Somppi, “Fully parallel 25 MHz, 2.5-Mb CAM,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 1998, pp. 332–333.

[11] 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, Nov. 1998.

[12] 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. (CICC)*, 2003, pp. 387–390.

[13] M. M. Khellah and M. Elmasry, “Use of charge sharing to reduce energy consumption in wide fan-in gates,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 1998, pp. 9–12.

[14] A. Roth, D. Foss, R. McKenzie, and D. Perry, “Advanced ternary CAM circuits on 0.13 _m logic process technology,” in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2004, pp. 465–468.

[15] 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 Syst. (ISCAS)*, vol. 4, 2001, pp. 926–929.

[16] A. Efthymiou and J. D. Garside, “An adaptive serial-parallel CAM architecture for low-power cache blocks,” in *Proc. IEEE Int. Symp. Low Power Electronics and Design (ISLPED)*, 2002, pp. 136–141.

[17] , “A CAM with mixed serial-parallel comparison for use in low energy caches,” *IEEE Trans. VLSI Syst.*, vol. 12, no. 3, pp. 325–329, Mar. 2004.

[18] N. Mohan and M. Sachdev, “Low power dual matchline content addressable memory,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 2004, pp. 633–636.

[19] K.-H. Cheng, C.-H.Wei, and S.-Y. Jiang, “Static divided word matchline line for low-power content addressable memory design,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 2004, pp. 629–632.

[20] 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. (CICC)*, 2003, pp. 383–386.

[21] --- ---, “A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme,” *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.

[22] J. M. Hyjazie and C. Wang, “An approach for improving the speed of content addressable memories,” in *Proc. IEEE Int. Symp. Circuits Syst.(ISCAS)*, vol. 5, 2003, pp. 177–180.

[23] Igor Arsovski and Ali Sheikholeslami, “A Mismatch-Dependent Power Allocation Technique for Match-Line Sensing in Content-Addressable Memories”, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 11, NOVEMBER 2003, pp 1958-1962

[24] N. Mohan, W. Fung, D. Wright, and M. Sachdev, “A ternary CAM with positivefeedback match line sense amplifiers and a match-token priority encoding scheme,” submitted to *IEEE Journal of Solid-State Circuits*.

[25] H. Noda, K. Inoue, M. Kuroiwa, F. Igaue, K. Yamamoto, H. J. Mattausch, T. Koide, A. Amo, A. Hachisuka, S. Soeda, I. Hayashi, F. Morishita, K. Dosaka, K. Arimoto, K. Fujishima, K. Anami, and T. Yoshihara, “Acost-efficient high-performance dynamic TCAM with pipelined

hierarchical search and shift redudancy architecture," *IEEE J. Solid- State Circuits*, vol. 40, no. 1, pp. 245–253, Jan. 2005.

[26] H. Noda, K. Inoue, M.Kuroiwa, A. Amo, A. Hachisuka, H. J. Mattausch, T. Koide, S. Soeda, K. Dosaka, and K. Arimoto, "A 143 MHz 1.1W4.5 Mb dynamic TCAM with hierarchical searching and shift redundancy architecture," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2004, pp. 208–209.