ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

Hybrid RAM-TCAM mapping for high performance processor

[1]M.Manimaraboopathy, [2]Dr.G.A Sathish Kumar, [3]S.Natarajan & *[1]Dr.M.Anto Bennet
[1]Department of Electronics and Communication Engineering, VelTech, Avadi, Chennai, Tamilnadu, India
[2]Department of Electronics and Communication Engineering, Sri Venkateswara Engineering College Sriperumpudhur, Chennai, Tamilnadu, India
[3] Department of Electrical and Electronics Engineering, VelTech, Avadi, Chennai, Tamilnadu, India
***Corresponding author: E-Mail: bennetmab@gmail.com**

ABSTRACT

An efficient hardware solution to perform table lookup is the content addressable memory (CAM). A CAM can be used as a co-processor for the network processing unit to offload the table lookup tasks. Besides the networking equipment, CAMs are also attractive for other key applications such as translation look-aside buffers in virtual memory systems. Ternary content addressable memories (TCAMs) are hardware-based parallel lookup tables with bit-level masking capability. A special logic unit, named Multiple Match Resolver (MMR), is required to resolve the best candidate if more than one words indicate a "match". In the early development of TCAM, the capacity was small, with only a few hundred to several thousand words. The design of MMR was relatively easy, and could be realized using static digital logics. They are attractive for applications such as packet forwarding and classification in network routers. Despite the attractive features of TCAMs, high power consumption is one of the most critical challenges faced by TCAM designers. This work proposes circuit techniques for reducing TCAM power consumption. The main contribution of this work is divided in two parts: (i) reduction in match line (ML) sensing energy, and (ii) static-power reduction techniques. The ML sensing energy is reduced by employing (i) positive-feedback ML sense amplifiers (MLSAs), (ii) low-capacitance comparison logic, and (iii) low-power ML-segmentation techniques. Here this technique was used in CAM memory which is combined with RAM memory to enhance the performance level while searching data from memory. The focus of this work is not on the TCAM memory cell design, but rather, it is on the low-power circuit techniques for multiple match resolution and detection in TCAM. Both digital techniques and mixed-signal techniques are presented and analyzed in details.

KEY WORDS: Ternary Content-Addressable Memory (TCAM), Content Addressable Memory (CAM), charge-injection match detection circuit (CIMDC), Random Access Memory, (RAM), Match Line (ML).

1. INTRODUCTION

Ternary Content-Addressable Memory (TCAM) devices are increasingly used for performing high-speed packet classification. A TCAM consists of an associative memory that compares a search key in parallel against all entries. TCAMs may suffer from error events that cause ternary cells to change their value to any symbol in the ternary alphabet {"0","1","*"}. Due to their parallel access feature, standard error detection schemes are not directly applicable to TCAMs; an additional difficulty is posed by the special semantic of the "*" symbol. This paper introduces PEDS, a novel parallel error detection scheme that locates the erroneous entries in a TCAM device. PEDS is based on applying an error-detecting code to each TCAM entry and utilizing the parallel capabilities of the TCAM by simultaneously checking the correctness of multiple TCAM entries. A key feature of PEDS is that the number of TCAM lookup operations required to locate all errors depends on the number of symbols per entry in a manner that is typically orders of magnitude smaller than the number of TCAM entries. For large TCAM devices, a specific instance of PEDS requires only 200 lookups for 100-symbol entries, while a naive approach may need hundreds of thousands of look ups. PEDS allows flexible and dynamic selection of trade off points, space complexity, and number of lookups. Ternary Content-Addressable Memory (TCAM), error-correcting codes, error-detecting codes, error detection, packet classification. Content Addressable Memory (CAM) is a special type of computer memory used in certain very high speed searching applications. Memory map is a structure of data that indicates how memory is laid out. Memory maps can have a different meaning in different parts of the operating system**.** A logical source register is renamed using its identifier to obtain the current mapping. This is performed faster and more efficiently in terms of energy with a RAM structure.

The RAM is directly indexed by a source register, whereas this register is compared against all current mappings in the CAM. A Content Addressable Memory (CAM) is an application specific memory that allows its entire contents to be searched within a single clock cycle. Binary CAM performs exact-match searches, while 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 of ternary data requires considerable layout area. DRAM based main memory significantly increases the power and cost budget of a computer system, new memory technologies such as Phase-change RAM (PRAM), Ferroelectric RAM (FRAM), and Magnetic RAM (MRAM) have been proposed to replace the DRAM. Among these memories, PRAM is the most promising candidate for large scale main memory because of its high density and low power consumption. In previous researches, a hybrid main memory approach of DRAM and PRAM is

adopted to make up for the latency and endurance limits of PRAM. On the other hand, large amount of a main memory is used for page cache to hide disk access latency. Many page caching algorithms such as LRU, LIRS, and CLOCK are developed and show good performance, but these are only consider the main memory with uniform access latency and unlimited endurance. They cannot be directly adapted to the hybrid main memory architecture with PRAM [8-12].We propose a new page caching algorithm for the hybrid main memory. It is designed to overcome the long latency and low endurance of PRAM. On the basis of the LRU replacement algorithm, we propose page monitoring and migration schemes to keep read-bound access pages to PRAM.

2. EXPERIMENTAL SECTION

A proposed technique is implemented with new form of CAM architecture to reduce the existing draw backs. In searching the data in CAM memory, first check the parity bit for input data and content in memory. Then we count number of ones in parity bit data to form as an index for fetching data. Then this was given to RAM memory to renaming or update the data. Due to this architecture, can get correct fetched data and also can get speed process of fetching data by new CAM architecture. A RAM is an acronym for Random Access Memory**,** a type of computer memory that can be accessed randomly that is any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of memory found in computers and other devices such as printers. Common usage, the term *RAM* is synonymous with main memory, the memory available to programs. For example, a computer with 8MB RAM has approximately 8 million bytes of memory that programs can use. In contrast, ROM refers to special memory used to store programs that boot the computer and perform diagnostics. Most personal computers have a small amount of ROM (a few thousand bytes). In fact, both types of memory (ROM and RAM) allow Random Access Memory. To be precise therefore, RAM should be referred to as read/write and ROM as read operation only.

A parity-bit is introduced to boost the search speed of the parallel CAM. Concurrently, a power-gated sense amplifier is proposed to improve the performance of the CAM comparison in terms of power and robustness. It also reduces the peak turn-on current at the beginning of each search cycle. A Parity Bit Based CAM. The parity bit based CAM design is consisting of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word and ML. This additional parity bit in theory reduces the sensing delay and boosts the driving strength of the 1-mismatch case (which is the worst case) by half. The CAMs can be used in Asynchronous Transfer Mode (ATM) switching network components as a translation table. Since ATM networks are connection-oriented virtual circuits need to be set up across them prior to any data transfer. CAM can act as an address translator in an ATM switch and perform the VPI/VCI translation very quickly. During the translation process, the CAM takes incoming VPI/VCI values in ATM cell headers and generates addresses that access data in an external RAM

2.1. Initialize memory: A CAM word (l-bit) is implemented by connecting (l) CAM cells in parallel. All the cells in a CAM word share an ML but they have separate SLs. The ML is connected to a ML sense amplifier (MLSA), which determines if the corresponding word matches with the search key. A conventional MLSA pre -charges the ML to VDD and places the search key on the SLs. During this operation, the ML remains at VDD only if the cell-level comparisons of all the bits result in "match". In other words, even a single-bit "mismatch" can create a discharge path for ML indicating a word (l bit) "mismatch". A CAM array (n x l) is implemented by connecting (n) CAM words sharing the same set of SLs. When the search key (l-bit) is written on SLs, it is compared with all the (n) words in parallel. The main components of an (n x l) CAM array including the SL drivers and a search key register. In this stage, we first made the CAM memory architecture which gives condition to ML sensing line to detect power level. Here we give total content information of data address. A Psychology, memory is the process in which information is encoded, stored, and retrieved. Encoding allows information that is from the outside world to reach our senses in the forms of physical stimulation. In this first stage we must change the information so that we may put the memory into the encoding process. Storage is the second memory stage or process. This entails that we maintain information over periods of time.

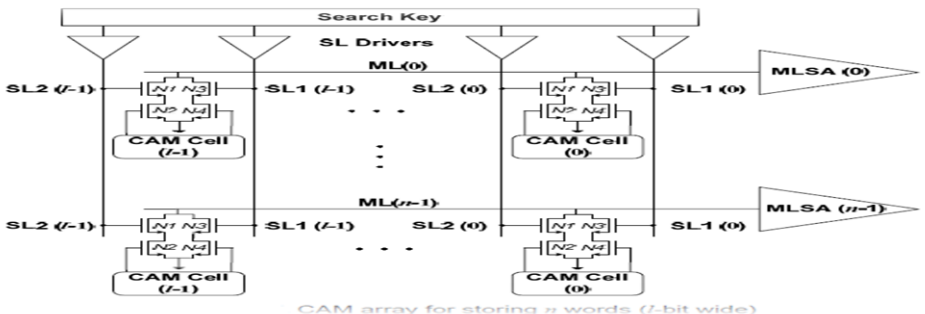

Fig.1.Initialize memory

Finally the third process is the retrieval of information that we have stored. We must locate it and return it to our consciousness. Some retrieval attempts may be effortless due to the type of information. All MLs are pre charged to VDD prior to every Search operation. If can be show fig 1 entries match with the search key, MLs remain at VDD, and the subsequent Search operation does not consume energy to pre-charge them. Unfortunately in most applications, the majority of the table entries do not match with the search key. As a consequence, almost all MLs are discharged to GND in every search operation. The switching activity of the SLs depends on the statistics of the search key. Assuming random data, each bit in the search key has equal probability of being '0' or '1'. Thus, almost half of the SLs switch in every Search operation. Each ML (SL) is highly capacitive because it is shared by all cells in a row (column). Therefore, owing to their large capacitances and high switching activities, the MLs and SLs consume a significant portion of the total TCAM power.

2.2. Extract counting of 1'S: The extract number of "1" in the stored words are counted and kept in the Counting bits segment. The data input for searching also made the same process. This information was first compare and finally search was done for matched no. of ones. In order to understand various trade-offs in CAM design, it is useful to write a general expression for the Search operation. When a search key is applied to a CAM-based table, it is compared bit-by-bit with all the table entries. If all the bits of a table entry match with the respective bits of the search key, it indicates a "match".

2.3. Extract parity bit: Extract parity bit for the stored words and keeps it in parity bit segment. Comparing and checking are done as same as in counting bit formation. RAM parity checking is the storing of a redundant parity bit representing the parity (odd or even) of a small amount of computer data (typically one byte) stored in Random Access Memory, and the subsequent comparison of the stored and the computed parity to detect whether data occurred. The parity bit was originally stored in additional individual memory chips with the introduction of plug-in DIMM, SIMM, etc. Modules they became available in non-parity and parity (with an extra bit per byte, storing 9 bits for every 8 bits of actual data) versions. Here SL (l-1) and BL (l-1) are 'lath' bits (or in this case, the most significant bits) of the search key and the table entry, respectively. Notice that the logical function is essentially the XNOR logic

$[SL(l-1) \oplus BL(l-1)] = '1'$ …… (1)

if $SL(l-1) = BL(l-1) = '1'$ or $SL(l-1) = BL(l-1) = '0'$.Therefore, a "match" (M = '1') will occur when $[SL(l-1) \oplus BL(l-1)] = '1'$ for all the bits: 0 to (l-1).

Taking the complements of both sides, equation can be rewritten as

$MM = M = [SL(l-1) \oplus BL(l-1)] + [SL(l-2) \oplus BL(l-2)] \ldots [SL(0) \oplus BL(0)]$

$[SL(l-1) \oplus BL(l-1)] = [SL(l-1) \cdot BL(l-1)] + [SL(l-1) \cdot BL(l-1)]$ v …… (2)

Hence, the "mismatch" (MM = '1') will occur when at least one bit of the search key fails to match the corresponding bit of a table entry. The logical function $[SL(l-1) \oplus BL(l-1)]$ is essentially the XOR logic.

2.4. Detect ML sensing line: In this stage, extract the matching line in the register and also gives controlling for each ML sensing lines. These ML sensing lines consists of controlling block at each output. This compare power level and gives command to power supply.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 VDD. Finally, the search key bit and its complementary value are placed on SL1 and SL1c, respectively. If the search key bit is identical to the stored value (SL1=BL1, SL1c=BL1c), both ML-to-GND pull-down paths remain 'OFF', and the ML remains at VDD indicating a "match". Otherwise, 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". 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. 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". Similarly for SL1 = '0', ML remains at '1' detecting "match".

2.5. Update page: Page Mapping is updated the searched address and page information in RAM memory and shown in fig.2 Page Mapping Structure. A large-capacity TCAM chip is expensive partially due to the large cell area. A smaller TCAM cell can reduce the cost of a TCAM chip by improving the layout density. The 6T dynamic cell is relatively smaller but it requires a specialized embedded DRAM process. Hence, the static cells are more attractive due to their compatibility with the standard logic process. Two TCAM cells that is more area-efficient than the conventional 16T static TCAM cell.

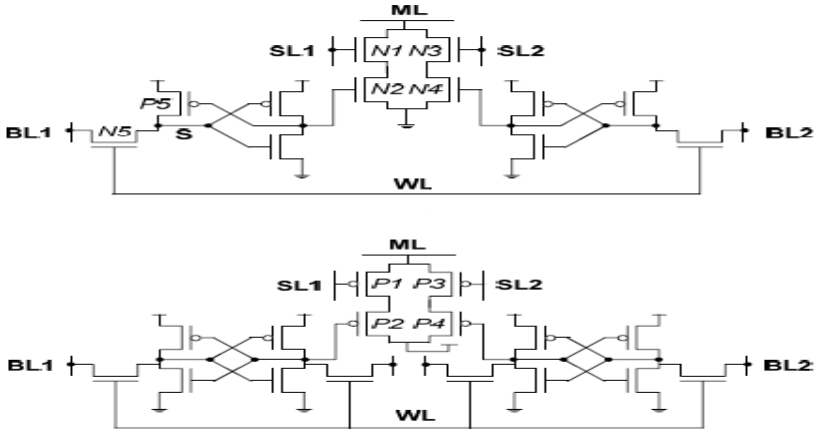

Fig.2.Page mapping structure

A 12T static TCAM cell reduces area by eliminating two access transistors and two driver transistors. It maintains a '0' state at node 'S' by satisfying the following two conditions: (i) BLs are discharged to ground, and (ii) the N5 leakage is higher than the P5 leakage. The second condition is fulfilled under all process and temperature variations by keeping the WLs at a non-zero voltage (VWL ≈ 200mV). This condition increases the BL leakages by 2-3 orders of magnitude. Therefore, this cell is not appropriate for low-power TCAMs. Moreover, this cell is not suitable for the READ operation, which is required for chip verification. The layout of this cell is more compact than that of the conventional 16T cell because it has an equal number of PMOS and NMOS transistors.

In order to minimize the TCAM cell area, the transistors and interconnects must be laid-out at the minimum distance specified by the design rules. Although such a dense layout is area-efficient, it leads to high inter-wire capacitance. The parasitic capacitances of BLs and WLs are not critical because READ or WRITE operations are performed only during the table updates, maintenance, and testing. During the SEARCH operation, most of the power is consumed in switching SLs and MLs. Hence, they should be routed such that their parasitic capacitances are minimized. The inter-wire capacitances of SLs and MLs can be reduced by placing them equally apart from the other parallel lines. Further reduction in the line capacitance can be achieved by minimizing the wire-widths of SLs and MLs. However, the lines should be wide enough to avoid problems such as electro migration and poor signal integrity under the worst-case operating conditions.

2.6. Design method:

Match line sense amplifiers: Most low-power MLSAs strive to minimize the ML voltage swing. A illustrates the conventional MLSA described. Initially, all the MLs are pre-charged to VDD, and the search key is written on the SLs. If a TCAM word is identical to the search key, the ML remains at VDD. Otherwise, it discharges to GND through mismatching cells. In order to avoid a short-circuit current; the SLs are switched to GND during the pre-charge phase. Hence, most of the SLs switch in every search operation, causing high power consumption. This scheme has the ML connected to GND during the pre-charge phase, so the SLs can remain at their previous values. Thus, the average SL switching activity can be reduced approximately by half. This scheme achieves further power reduction by lowering the ML voltage swing. The ML sensing is initiated by charging up the ML using a constant current source. Since a matching ML does not have a current discharge path, it charges at a faster rate than a mismatching ML. When the matching ML charges to the NMOS threshold voltage (Vth), its MLSO changes from '0' to '1'. A dummy ML emulating the "match" condition generates an MLOFF signal to end the ML sensing. Another MLSA that reduces the ML voltage swing using charge redistribution. This scheme also has the MLs connected to GND during the pre-charge phase. The ML sensing begins with fast pre-charging of the MLs using a Fast Pre signal. Transistors N1 and N2 restrict the ML voltage swing to (VREF– Vth). After the Fast Pre pulse, the MLs are left floating. Under the "mismatch" condition, the ML voltage drops below (VREF – Vth), and transistors N1 and N2 turn on. Transistor N2 equalizes the voltages of nodes ML and SP by redistributing charge at the two nodes. A small current source (IREF) feeds the node SP to compensate for ML leakages. The voltage VREF can be varied to trade off power consumption with the speed of operation. This method can reduce the ML voltage swing to even below Vth. If a show fig 3 the fast pre-charging of mismatching MLs causes short circuit power dissipation. A charge-injection match detection circuit (CIMDC) eliminates the short circuit power. CIMDC uses an injection capacitor (CINJ) for each ML. Typically; CINJ is sized 3-4 times smaller than CML. Initially, all the injection capacitors are pre-charged to VDD and all the MLs are discharged to ground. At evaluation, charge is injected from CINJ to CML using the Charge in signal.

Under the "match" condition, the voltage of CML rises to a voltage determined by the ratio of CINJ and CML. Under the "mismatch" condition, ML is discharged to ground. An offset sense amplifier differentiates

between the "match" and "mismatch" conditions. Although the charge injection scheme reduces the ML swing to very small voltages (~ 300mV), it suffers from lower noise margin and area penalty due to CINJ. The delay and energy consumption of the above ML sensing schemes for different word sizes when they are simulated in 0.18µm CMOS technology. Global masking (GM) also alters the delay and energy by changing the ML capacitance.

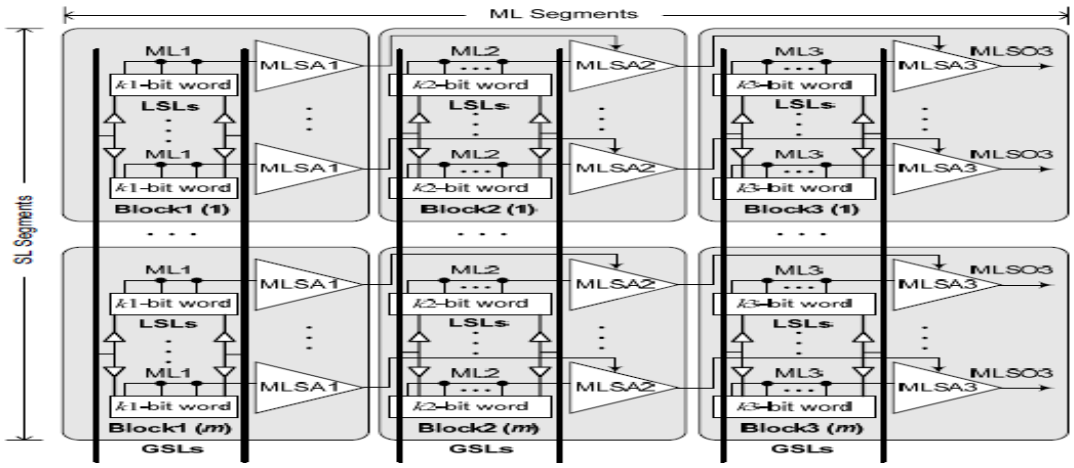

Fig.3.Match line segmentation

So far, it has been assumed that all the bits of a word share the same ML. The power consumption of ML sensing can be significantly reduced by segmenting MLs. One of the most popular ML-segmentation techniques is selective pre charge. A conventional TCAM performs a search operation in one step for all the bits. The selective recharge scheme divides the search operation into multiple stages. A illustrates the most common implementation of this scheme using two stages: Pre-Search and Main-Search. The Pre-Search stage performs the search operation on the first segment. If this results in "match", the Main-Search stage also performs the search operation on the second segment. This scheme can achieve significant power savings if the Pre-Search stage causes "mismatch" in most of the words. For small values of k, the energy consumed by Pre-Search stage is small. However, k should be large enough to cause "mismatch" in most of the words. The optimal value of k for minimum average energy depends on the statistics of the incoming data (search key). For example, a selective recharge TCAM designed for networking applications with l = 144 and k = 36 can save up to 75% of the ML power, where l is the total number of bits per word. A recent design further extends the original selective pre charge scheme by dividing each ML into five segments, which also enables the use of hierarchical SLs as explained in the next subsection.

2.7. Search line drivers: A significant portion of the TCAM power is also consumed by SL drivers in switching highly capacitive SLs. The SL switching activity depends on the incoming data statistics. For random data, almost half of the SLs are switched in fig 4 every search operation. Thus, a significant amount of power can be saved by reducing the voltage swing of SLs.It can be smaller SL voltage swing reduces the ION/IOFF ratio of the ML pull-down paths. Therefore, most TCAM designs do not reduce the SL voltage swing. Some recently published designs extend the selective pre charge idea to SLs by dividing them into a two-level hierarchy of global SLs (GSLs) and local SLs.These hierarchical SLs are implemented along with the ML segments described. For example, a group of 64 ML1-words can be defined as Block1, a corresponding group of ML2-words can be defined as Block2, and so on. During a search operation, GSLs broadcast the search key throughout the TCAM, but LSLs within a block are activated only when there is at least one "match" in the previous block. The LSLs of Block2 (m) will be activated only when at least one "match" is found in Block1 (m).

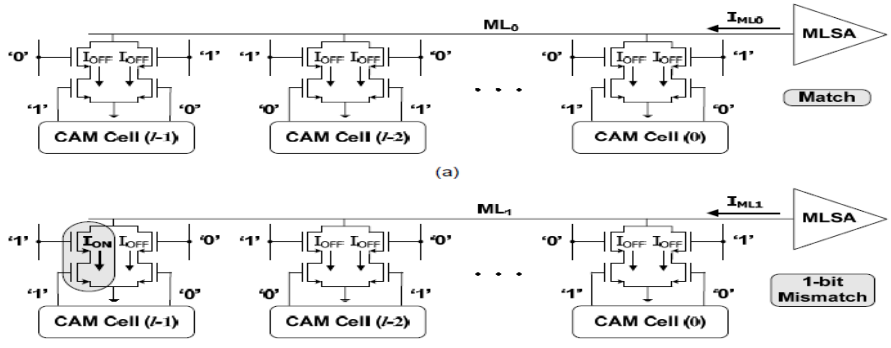

Fig.4.Search line drivers

In every search operation, only a few words match with the search key. Thus, most blocks will not contain even a single "match", and this scheme will save power by keeping the LSLs of these blocks inactive. The ION/IOFF ratio of the ML pull down paths is maintained by having a rail-to-rail voltage swing (1.8V) in LSLs. The power consumption is reduced by having a smaller voltage swing (0.45V) in GSLs. The lows wing GSL signals are converted to the full-swing LSL signals using low-swing amplifiers. This scheme reduces the SL power consumption by 60%. However, the power reduction comes at the expense of area overhead due to wide OR-gates (64-input), low-swing amplifiers, and other control circuits, which are embedded in the TCAM array. This scheme requires two separate power supply pins and an on-chip distribution network to support the low-swing GSL-drivers and the full-swing LSL-drivers. Since the area consumed by the power supply distribution network is not negligible, this scheme further reduces the effective on-chip area available for the core TCAM array. This scheme can be implemented only if the MLs are divided into multiple segments and the incoming data is searched sequentially. This constraint also degrades the search speed.

2.8. Low- capacitance comparision logic: A significant portion of the TCAM power is consumed in switching highly capacitive MLs. There are two main sources of the ML capacitance: (i) interconnect capacitance of the metal used for ML routing, and (ii) drain capacitances of the comparison logic transistors. The ML interconnect capacitance mainly consists of (i) ML-to substrate capacitance, and (ii) the coupling capacitance between MLs and other parallel lines such as WLs, GND and VDD buses.The ML-to-substrate capacitance can be show fig 6.3 reduced by choosing a high-level metal (such as M4) with minimum width (as specified by the design rules) for routing MLs. Similarly, the ML coupling capacitance can be minimized by (i) routing MLs and other parallel lines in different metals, and (ii) placing MLs equally apart from the other parallel lines. Drain capacitances of the comparison logic transistors also contribute to the ML capacitance. A significant reduction in the ML capacitance can be achieved by employing minimum size transistors in the comparison logic circuits. If secondary effects are ignored, the drain capacitance, ION and IOFF are directly proportional to the channel width. As a consequence, the speed and robustness of the ML sensing is not affected by the channel width. Therefore, the minimum size transistors reduce the search energy without degrading the ION/IOFF ratio. In sub-100nm CMOS technologies, the channel width can be slightly larger than the minimum size specified by the design rules to avoid excessive process variations and secondary effects (such as normal and reverse narrow channel effects).

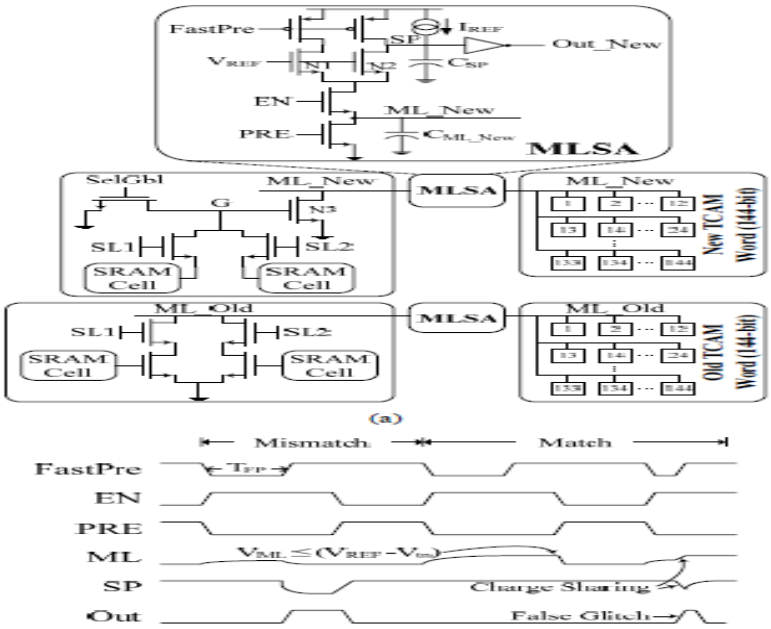

Fig.5.Comparison logic

A larger IML1/IML0 ratio has two main advantages. First, it makes the ML sensing less sensitive to process variations and operating conditions. For example, if an ML with one-bit mismatch (ML1) is receiving a larger current than the dummy ML due to the mismatch in their current sources, ML1 may be detected as a "match" before the output of the dummy MLSA turns 'OFF' the current sources. This problem may also be caused by a threshold voltage mismatch between the dummy MLSA and the MLSA connected to an ML1. A larger margin between IML1 and IML0 can cope with larger process variations. Thus, the same dummy ML can be used for a larger block of TCAM words increasing the layout density of the TCAM chip. Secondly, a larger IML1/IML0 ratio allows the implementation of wide TCAMs because IML0 is proportional to the word size (l), and a larger value of l diminishes the difference between IML1 and IML0. We analyzed the proposed and conventional comparison logic

circuits by implementing them in two 145-bit wide TCAM words. A charge redistribution MLSA is used for ML sensing whose timing signal. All the control signals are common to both MLSAs. Initially, the MLs are discharged to ground using PRE. The search operation is initiated by the rising edge of EN, and the falling edges of Fast Pre and PRE. The ML voltage swing is restricted by the NMOS transistors (N1 and N2) whose gates are connected to a reference voltage (VREF). The Fast Pre pulse pre charges the MLs to a voltage near (VREF – Vth). The evaluation begins with the rising edge of the Fast Pre signal. Under the match condition, the ML does not have a pull down path, and its node SP remains at VDD.Under the mismatch condition, the node SP is pulled down to GND through N2 and ML discharge path. A small current source (IREF) at the node SP compensates for ML leakages. In our design, IREF has been set to one-fifth of ION.

3. RESULTS

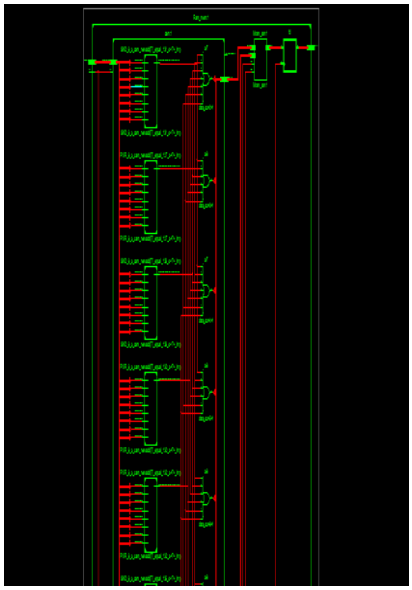

Fig.6.RTL schematic diagram

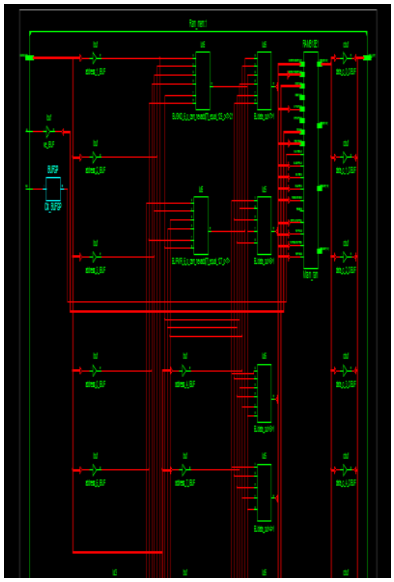

Fig.7.Technology schematic diagram

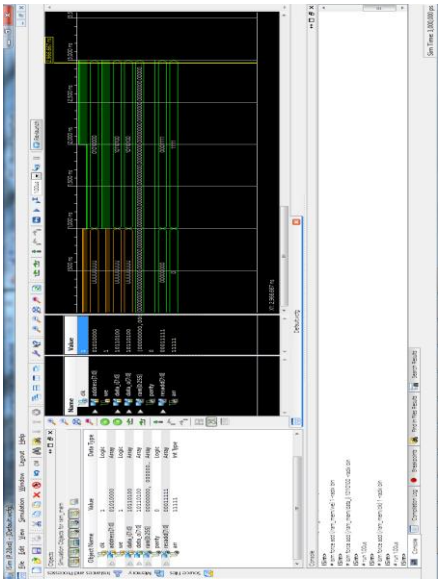

Fig.8.Output timing diagram

The I Sim GUI opens and loads the design. The simulator time remains at 0 ns until you specify a run time. For comparison purposes, you can browse to the completed folder for a completed version of the simulate_isim.bat batch file.RAM speed automatically increase. They associate the input (compared) with their memory contents in one clock cycle. They are configurable in multiple formats of width and depth of search data that allows searches to be conducted in parallel. We can add new entries into their table to learn what they don't know before. It is designed to overcome the long latency and low endurance. The power consumption is 77.2% and searching time period with in second shown in fig.(6-8).

4. CONCLUSION

Implement hybrid combination of RAM with CAM architecture. In this method, we change the architecture of CAM design to speed up the process. Experimental result shows better performance result than traditional CAM architecture. We presented a renaming mechanism consisting of a RAM table and a low-complexity CAM table, as a hybrid design that took the best of both approaches. Experimental results showed that a two-way hybrid approach achieved small performance slowdowns (about 2% and 1% for integer and floating-point benchmarks, respectively) with respect to a four-way CAM-based renaming mechanism that was able to recover in one clock cycle. These small slowdowns were accompanied by a drastic reduction of the original associative searches carried out in the CAM-based approach to only 8% and 3%. Hybrid designs reduced the dynamic energy by 16% and 12% with respect to the original CAM consumption, closing the dynamic energy consumption gap between CAM and RAM approaches.

REFERENCES

Akhbarizadeh J, Nourani M, and Cantrell C.D, Prefix segregation scheme for a TCAM-based IP forwarding engine, IEEE Micro, 25(4), 2005, 48–63

Casagrande G, Warn Varnas A, Stephan Y, and Folegot T, Genesis of the coupling of internal wave modes in the Strait of Messina, J.Mar.Syst., 78(1), 2009, S191–S204.

Cheng K, Wei C.H, and Jiang S.Y, Static divided word matching line for low-power content addressable memory design, in Proc. Int. Symp. Circuits and Syst., 2004.

Clark R, and Schimmel D.E, Scalable pattern matching for high speed networks, in Proc. IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM), Napa, CA, 2004, 249–257.

Dr. AntoBennet M, Manimaraboopathy M, Maragathavalli P, Dinesh Kumar T R, Low Complexity Multiplier For Gf(2m) Based All One Polynomial, Middle-East Journal of Scientific Research, 21(11), 2014, 2064-2071.

Dr. AntoBennet M, Power Optimization Techniques for sequential elements using pulse triggered flipflops, International Journal of Computer & Modern Technology , 1(1), 2015, 29-40.

Dr. AntoBennet M, Sankar Babu G, Suresh R, Mohammed Sulaiman S, Sheriff M, Janakiraman G, Natarajan S, Design & Testing of Tcam Faults Using T_H Algorithm, Middle-East Journal of Scientific Research, 23(08), 2015, 1921-1929.

Efthymiou A, and Garside J.D, A CAM with mixed serial-parallel comparison for use in low energy caches, IEEE Trans.Very Large Scale Integr.(VLSI) Syst., 12(3), 2004, 325–329.

Giunchiglia F, and Shvaiko P, Semantic matching, in The Knowledge Engineering Rev.Cambridge, U.K, Cambridge Univ. Press, 2004.

Hershey P, and Silio C, Surmounting data overflow problems in the collection of information for high speed communications systems, in Proc. IEEE Systems Conf., Montreal, QC, Canada, 2008.

Huang T, Chang S.W, Liu W.Y, and Huang W, A energy-efficient TCAM with novel low power schemes, in Proc. IEEE Int. Symp. VLSI Design, Autom. Test (VLSI-DAT), 2007.

Pagiamtzis K, and Sheikholeslami A, A low power content-addressable memory (CAM) using pipelined hierarchical search scheme, IEEE J. Solid-State Circuits, 39(9), 2004, 1512–1519.

Pagiamtzis K, and Sheikholeslami A, Content-addressable memory (CAM) circuits and architectures:A tutorial and survey, IEEE J. Solid- State Circuits, 41(3), 2006, 712–727.

Shvaiko P and Euzenat J, A survey of schema-based matching approaches, J. Data Semantics, IV, 2005, 146–171.

Wang J, Wang C.C, and Yeh C.W, High-speed and low-power design techniques for TCAM macros, IEEE J.Solid-State Circuits, 43(2), 2008, 530–540.