

CONTENT ADDRESSABLE MEMORY BASED ON SPARSH CLUSTERED NETWORK ALGORITHM

¹ K. Ratna, ² N. Kranthi Reddy, ³ A. Triveni

^{1,2,3}Department of Electronics & Communication Engineering, Aurora's Scientific, Technological and Research Academy, Hyderabad, India, 500 005

Abstract- Most memory devices store and retrieve data by addressing specific memory locations. As a result, this path often becomes the limiting factor for systems that rely on fast memory accesses. The time required to find an item stored in memory can be reduced considerably if the item can be identified for access by its content rather than by its address. A memory that is accessed in this way is called content-addressable memory or CAM. Content Addressable Memory provides a performance advantage over other memory search algorithms, such as binary or tree-based searches or look-aside tag buffers, by comparing the desired information against the entire list of pre-stored entries simultaneously, often resulting in an order-of-magnitude reduction in the search time. Different Content Addressable Memory algorithms are available today, each of them having its own advantages as well as disadvantages. Here study on Content Addressable Memory architecture and algorithm is done. The introduction of Sparse Clustered Network (SCN) had a great impact on Content Addressable Memory designs. A Content Addressable Memory based on Sparse Clustered Networks has been proposed. The dynamic energy consumption of the proposed design is significantly lower compared with that of other conventional low-power Content Addressable Memory design.

Keywords- Content Addressable Memory (CAM), Sparse Clustered Network (SCN), Associative Memory; Message Error Rate (MER)

I. Introduction

A Content Addressable Memory is a type of memory that can be accessed using its contents rather than an explicit address. In order to access a particular entry in such memories, a search data word is compared against previously stored entries in parallel to find a match. Each stored entry is associated with a tag that is used in the comparison process. Once a search data word is applied to the input of a Content Addressable Memory, the matching data word is retrieved within a single clock cycle if it exists. This prominent feature makes CAM a promising candidate for applications where frequent and fast look-up operations are required, such as in translation look-aside buffers (TLBs), database accelerators, image processing, parametric curve extraction, Hough transformation, Huffman coding/decoding and image coding. Although dynamic CMOS circuit techniques can result in low-power and low-cost CAM's, these designs can suffer from low noise margins, charge sharing, and other problems not to be energy efficient when scaled. Thus a new family of associative memories based on SCNs has been recently introduced and implemented using field-programmable gate arrays (FPGAs). Such memories make it possible to store many short messages instead of few long ones as in the conventional Hopfield networks with significantly lower level of computational complexity. Furthermore, a significant improvement is achieved in terms of the number of information bits stored per memory bit (efficiency). A low-power CAM employing a new algorithm for associatively between

input tag and the Corresponding address of the output data which is based on sparse clustered network using binary connections that on-average eliminates most of the parallel comparisons performed during a search, is compared against various other content addressable memories. It was found that the dynamic energy consumption of the proposed design is significantly lower compared with that of a conventional low-power Content Addressable Memory design. Given an input tag, the proposed architecture computes a few possibilities for the location of the matched tag and performs the comparison on them to locate a single valid match.

II. Literature Review

Yamagata T et al (1992) proposed a 288-kb (8K words X 36 b) fully parallel content addressable memory (CAM) LSI using a compact dynamic Content Addressable Memory cell with a stacked-capacitor structure and a novel hierarchical priority encoder [1]. The proposed CAM cell is shown in Fig 2.1. It consists of five NMOS transistors and two stacked capacitors. Four of the transistors (M_{s0} , M_{s1} , M_{w0} , and M_{w1}) are to store and access data, and one (M_d) is used as a diode to isolate current paths during match operations. Charges are stored on stacked capacitors (C_{s0} , C_{s1}) and the M_{s0} and M_{s1} gates. The opposite electrodes of the C_{s0} and C_{s1} are connected to a cell plate voltage V_{cp} , which is equal to half V_{cc} (V_{cc} : supply voltage). Two bit lines are supplied with data in write and search operations. The word line (WL) allows write access to each cell in a word. The match line (ML)

passes through a word to perform a logical AND of the results of each cell's comparison. The ML is also used to read cell data. The storage capacitor (C_{so}) is stacked on the gate of the M_{s0} and M_{w0} . The gate electrodes of the M_{s0} and M_{w0} are fabricated with the first poly-Si layer [1]. The stacked capacitor is composed of the second poly-Si layer, the insulator film, and the third poly-Si layer. The second poly-Si layer (storage node) is also used to connect the drain (or source) of the M_{w0} to the gate of the M_{s0} . Similarly, the storage capacitor (C_{s1}) is formed on the gate of the M_{s1} and M_{w1} .

A write operation is performed by activating a select WL and then driving the bit lines according to the write data. A read operation is accomplished by discharging the bit lines and then driving a selected ML to a high level. A read operation is accomplished by discharging the bit lines and then driving a selected ML to a high level. A match operation is achieved by precharging both the bit lines and the match lines to a high level, and then loading the bit lines with search data. When match occurs at several words in a search operation (multiple response), the CAM outputs the address of the matched word with the highest priority. A priority encoder (PE) circuit is utilized for multiple-response resolution and match address generation [1]. As a bit capacity of Content Addressable Memory's becomes larger, the number of words increases rather than the bit length of words. Thus in a high density Content Addressable Memory chip, the configuration is such that the cell array is divided into several blocks. This creates a serious problem concerning the layout of the priority encoder. When the priority encoder is incorporated in each block, the silicon area occupied by the priority encoder and the power dissipation of the priority encoder are increased in proportion to the number of divided blocks. So in order to overcome this, novel hierarchical priority encoder architecture suitable for high-density CAM was proposed.

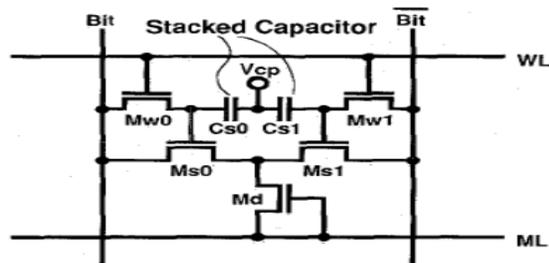


Fig 2.1: CAM cell structure with a stacked capacitor

In the novel CAM cells, since a stacked capacitor is adopted as the storage capacitor, a storage capacitance of 90ff is attained in a 66 μ m Content Addressable Memory cell using a 0.8 μ m CMOS process [1]. This is sufficient for the high soft-error immunity and provides stable performance of the operations mentioned above.

Furthermore, it suggests the possibility of achieving a more compact Content Addressable Memory cell by device scaling. The novel hierarchical priority encoder reduces the circuit area and power dissipation. A typical search cycle time of 150ns and a maximum power dissipation of 1.1W have been obtained using circuit simulation.

Schultz K et al (1996) developed a CAM based on bank selection architecture. In the bank-selection architecture the CAM array is divided into B equally partitioned banks that are activated based on the value of added bits of length $\log_2(B)$ to the search data word. These extra bits are decoded to determine, which banks must be selected. The basic concept is that bank selection divides the CAM into subsets called banks. The bank-selection scheme partitions the Content Addressable Memory and shuts off unneeded banks. Two extra data bits, called bank-select bits, partition the Content Addressable Memory into four blocks. When storing data, the bank-select bits determine into which of the four blocks to store the data. When searching data, the bank-select bits determine which one of the four blocks to activate and search. The decoder accomplishes the selection by providing enable signals to each block [2]. In the original preclassification schemes, this architecture was used to reduce area by sharing the comparison circuitry between blocks. Although the blocks are physically separate, they can be arranged such that words from different blocks are adjacent. Thus only one of four blocks is active at any time, only 1/4 of the comparison circuitry is necessary compared to the case with no bank selection, thus saving area. Bank selection reduces overall power consumption in proportion to the number of blocks. Thus, using four blocks ideally reduces the power consumption by 75% compared to a Content Addressable Memory without bank selection. The major drawback of bank selection is the problem of bank overflow. Since, in a Content Addressable Memory, there are many more input combinations than storage locations, the storage of a bank can quickly overflow. For example, a Content Addressable Memory with 72-bit words (and an additional bank-select bit) and 32K entries divided into two banks with 16K entries.

While each bank has 16K locations, there are actually 2 possible entries per bank. Thus, it can often occur that there are more entries than can fit in the assigned bank [11]. This overflow condition requires extra circuitry and forces multiple banks to be activated at once, decreasing the savings in power. To avoid overflow, an external mechanism can balance the data in the banks by periodically re-partitioning the banks.

Murkowski C et al (1997) developed a CAM aimed at reduced energy consumption. Energy reduction of Content Addressable Memory employing circuit-level techniques are mostly based on the following strategies:

- 1) Reducing the SL energy consumption by disabling the precharge process of SLs when not necessary and
- 2) Reducing the ML precharging, for example, by segmenting the ML, selectively precharging the first few segments and then propagating the precharge process if and only if those segments match [9].

This segmentation strategy increases the delay as the number of segments is increased. A hybrid-type CAM integrates the low-power feature of NAND type with the high performance NOR is type while similar to selective precharging method, the ML segmented into two portions. The high-speed CAM designed in 32-nm CMOS achieves the cycle time of 290 Ps using a swapped Content Addressable Memory cell that reduces the search delay while requiring a larger Content Addressable Memory cell (11-transistors) than a conventional Content Addressable Memory cell [9 transistors(9T)] used in SCN-Content Addressable Memory [3]. A high-performance AND-type match-line scheme is proposed in, where multiple fan-in AND gates are used for low switching activity along with segmented style match-line evaluation to reduce the energy consumption.

Lin C et al (2003) developed an architectural technique for saving power, which applies only to binary CAM, is pre-computation. Pre-computation stores some extra information along with each word that is used in the search operation to save power. These extra bits are derived from the stored word, and used in an initial search before searching the main word. If this initial search fails, then the Content Addressable Memory aborts the subsequent search, thus saving power. The extra information holds the number of ones in the stored word. For example when searching for the data word, 10111, the pre-computation circuit counts the number of one's (which is four in this case). Hence PB-CAM is also known as 1's count [9]. First, it counts the number of ones in an input and then compares the result with that of the entries using an additional CAM circuit that has the number of Ones in the CAM data previously stored. This activates a few MLs and deactivates the others. In the second stage, a modified Content Addressable Memory hierarchy is used, which has reduced complexity, and has only one pull-down path instead of two compared with the conventional design. The modified architecture only considers 0 mismatches instead of full comparison since the 1s have already been compared. The number of comparisons can be reduced to $M \times [\log(N+2)] + (M \times N)/(N+1)$ bits, where M is the number of entries in the Content Addressable Memory and N is the number of bits per entry [3]. In the proposed design, we demonstrate how it is possible to reduce the number of comparisons to only N bits. Furthermore, in PB-CAM, the increase of the tag length affects the energy consumption, the delay, and also complicates the precomputation stage.

Pagiamtzis K et al (2004) developed a Content Addressable Memory which is a combination of pipelined architecture and hierarchical architecture. The power can be saved by power adding pipelining to match-lines and adding hierarchy to search-lines in an otherwise non pipelined, non hierarchical CAM. The power savings of the pipelined match-lines is a result of activating only a small portion of the match line segments. Similarly, the power savings of the hierarchical search-lines is a result of activating only a small portion of the local search lines. Pipelining match- lines saves 56% power compared to non pipelined match-lines. Adding hierarchy to search-lines saves 63% power compared to nonhierarchical search-lines. The combination of the two techniques reduces overall power consumption by 60% [4].

The match line is divided into five match line segments, each evaluated sequentially in a pipeline fashion. The left-most segment has 8 bits while the other four segments have 34 bits each, for a total of 144 bits (a typical word width used for IPv6 address lookup).The match line segment array (MLSA) current source that provides the current is divided among the five segments in proportion to the number of bits in each segment. This is to guarantee identical speed in all match line segments and to allow a fair comparison with the non pipelined architecture. The pipelined match line operates from left to right, with each match line segment acting as an enable signal for the match line segment array of the subsequent segment. Hence, only words that match a segment proceed with the search in their subsequent segments [4-9]. Words that fail to match a segment do not search for their subsequent segments and hence consume no power.

Having pipelined the match-lines, the significant portion of the power is now consumed by the highly capacitive search lines. This problem is solved by observing how the search lines are activated in the pipelined match line architecture. As the match signals traverse the pipeline stages from left to right, fewer match lines segments survive the matching test and hence fewer match line segments will be activated. However, the search lines must be activated for the entire array at every stage of the pipeline, since the search-lines must reach the surviving match-line segments. This excessive power consumption is curtailed in our design by breaking the search-lines into global and local search-lines (GSLs and LSLs), with the global search-lines using low-swing signaling and the local search-lines using full-swing signaling but with reduced capacitance. Also, by global search-lines not directly serving every single Content Addressable Memory cell on a search line, the global search-lines capacitance is further reduced, resulting in extra power savings. Consecutive sub search words are different. However, the cycle time is drastically increased when the search- data patterns are correlated. For example, if we have correlations in the first 8 bits of the

stored data, the cycle time is increased to 1.359 ns, which is 5.2 times that of the non-correlated scenario [7]. In the proposed design, the cycle time is independent of the correlation between the input patterns. Furthermore, the asynchronous architecture is more susceptible to process variations compared with its synchronous counterparts. The concept for saving power is to change the encoding of stored data in the CAM cell. By storing the entry 101XX in a 5-bit ternary Content Addressable Memory, the stored word will match the four search data words 10100, 10101, 10110, and 10111. Correspondingly, it can be viewed that a stored word in a ternary Content Addressable Memory as actually storing multiple words. This can be extended to multiple entries. Thus, for example, the three entries 00011, 001XX, 01000 will match any search word in the range 00011 to 01000 [7]. Efficiently mapping a set of ranges to stored ternary Content Addressable Memory words allows for reduction of the size of the Content Addressable Memory required for an application. Reducing the number of entries also has the effect of reducing power consumption, since power consumption in Content Addressable Memory is proportional to the array size. To allow for a more dense encoding than conventional ternary Content Addressable Memory, propose changing the encoding of the Content Addressable Memory resulting in a smaller average number of entries. For the application of network routing, their scheme in comparison to conventional ternary encoding reduces the number of entries by almost 50% in some cases.

A group of two ternary cells is called a local match line in this scheme. In conventional ternary Content Addressable Memory encoding, there are three possible states (0, 1, X) for each ternary Content Addressable Memory cell and, correspondingly, nine states in total for two ternary cells. But only some of the possible combinations are available. For example, the combinations of 0, 2 and 1, 3 are available, but the combinations of 0, 3 and 0, 2, 3 are not available. Since there are four storage cells for two ternary cells, there are actually 16 possible states. This architecture is modified to accommodate the dense encoding.

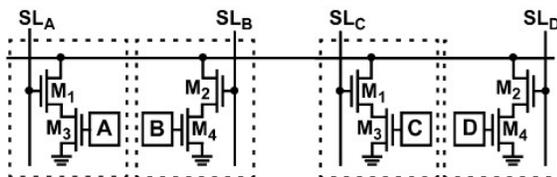


Fig 2.2: Two ternary CAM cells viewed as four independent half cells in the dense encoding system

The dense encoding scheme makes use of these previously unused states to make possible the storage of any combination. The match line architecture is modified

to accommodate the dense encoding. The match line is divided into local match lines (LMLs) and global match lines (GMLs). The global match lines are divided into sets of local match lines that are made up of four ternary half-cells. A global match line operates the same way as the conventional NOR does match line, where a path to ground indicate a miss and no path to ground indicate a match. The logic levels on the local match lines, however, are inverted so that a path to ground indicates a match, while no path to ground indicates a miss.

To see how this works, it is assumed that a local match line stores the numbers 0, 2, and 3, which accordingly means that ABCD as per Table 2.1. In the case where the search word is 0, 2, or 3 (corresponding to search lines set to 0001, 0100, and 1000, respectively), there is a path to ground indicating a match. In the case of search for a 1 (corresponding to search lines set to 0010) there is no path to ground indicating a miss. The GMLs are used to construct longer words that are made up of multiple local match lines

The proposed algorithm lowers the error rate by an order of magnitude for our sample network with 60% deleted contents [8]. Data storage and retrieval methodologies in associative memories are different from the widely-known indexed memories in which the data is written and accessed using explicit addresses. In associative memories, only associations between parts of data patterns are stored in a way that data can later be accessed by presenting a partial input pattern. Classical associative memories implemented with Hopfield Neural Networks (HNN) store input patterns (messages) in a network consisting of nodes and connections between them, where the index of each data bit corresponds to that of a node. The decoding algorithm for Hopfield Neural Networks retrieves messages from partial inputs using the stored integer weights. A drawback of Hopfield Neural Networks is that the number of nodes in the network must be equal to the length of each message which, due to a fixed available memory capacity, limits the diversity of the stored messages the number of different messages the network can store.

Table 2.1: Possible Encoding For Two Ternary Cells

Dense Encoding

Entry Data	Stored Values (ABCD)
-	0000
0	0001
1	0010
2	0100
3	1000
0,1	0011
2,3	1100
0,2	0101
1,3	1010
1,2	0110
0,3	1001
0,1,2	0111
1,2,3	1110
0,2,3	1101
0,1,3	1011
0,1,2,3	1111

Furthermore, the network is not efficient in terms of memory usage since the ratio between the number of bits stored to that used approaches zero as the network grows to increase its storage capacity. These drawbacks were addressed in Sparse Clustered Networks (SCNs) where the interconnections are binary and the nodes are clustered in such a way that a significantly larger diversity and memory efficiency are achieved compared to those of an Hopfield Neural Networks.

A drawback of the conventional SCNs is that due to the use of binary-weighted interconnections, deleting or updating messages causes the removal of the shared connections used by other stored messages [8]. The lost connections result in a significant increase in the Message Error Rate (MER) the ratio between the average numbers of correctly retrieved messages to the total number of partial input messages selected from the previously stored ones. Some of the lost connections can be recovered due to the error-correcting capability of Sparse Clustered Networks for which the degree or recovery depends on the selection of hardware architecture and the application. So propose an algorithm based on a variation of the conventional Sparse Clustered Networks algorithm that incorporates Multiple-Valued (MV) interconnections in

this algorithm (MV-SCN), deletions and updates of messages are achieved with a significant improvement in the Message Error Rate compared to that of its binary-weighted counterpart. Two architectures are recommended for use in the related applications: The first one optimized for aggressively updated, low Message Error Rate applications, and the second, suitable for applications requiring a low hardware complexity and low Message Error Rate but more relaxed deletion or updating requirements. A trade-off analysis is then followed to elaborate the benefits of architecture that depend on the application requirements. Both of the recommended architectures can be tuned for a desired Message Error Rate with the cost of hardware resources.

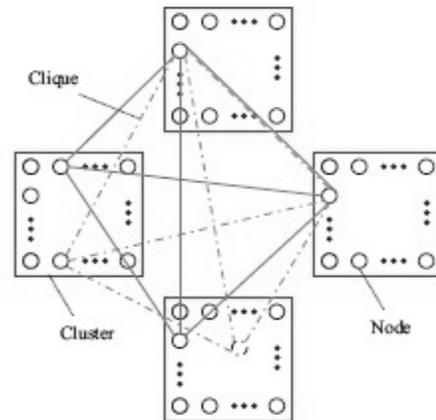


Fig 2.3: Graphical representation of SCN with four clusters Storing two messages

Comparisons on them to locate a single valid match. TSMC 65-nm CMOS technology was used for simulation purposes. Following a selection of design parameters, such as the number of content-addressable memory entries, the energy consumption and the search delay of the proposed design are 8%, and 26% of that of the conventional NAND architecture, respectively, with a 10% area overhead.

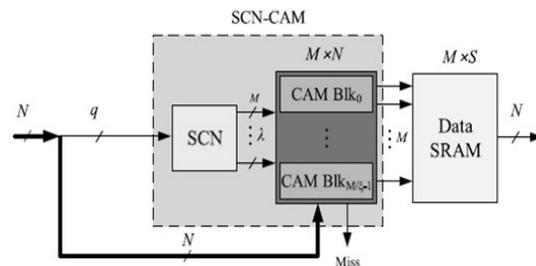


Fig 2.4: Block diagram of SCN-CAM

The sparse clustered network based content-addressable memory consists of a sparse clustered network based classifier coupled to a content-addressable memory array. The content-addressable memory array is

divided into several equally sized sub-blocks, which can be activated independently. For a previously trained network and given an input tag, the classifier only uses a small portion of the tag and predicts very few sub-blocks of the content-addressable memory to be activated. Once the sub-blocks are activated, the tag is compared against the few entries in them while keeping the rest deactivated and thus lowers the dynamic energy dissipation. As shown in Fig 2.4, this architecture consists of a sparse clustered network based classifier, which is connected to a special-purpose content- addressable memory array. The sparse clustered network based classifier is at first trained with the association between the tags and the address of the data to be later retrieved.

The proposed CAM array is based on a typical architecture, but is divided into several sub-blocks that can be compare-enabled independently. Therefore, it is also possible to train the network with the association between the tag and each content-addressable memory sub-block if the number of desired sub-blocks is known. Once an input tag is presented to the sparse clustered network based classifier, it predicts which content- addressable memory content-addressable memory sub- block(s) need to be compare-enabled and thus saves the dynamic power by disabling the rest. Disabling a content-addressable memory sub-block avoids charging its highly capacitive SLs, while applying the search data, and also turns the precharge path off for the MLs.

The SCN-CAM uses only a portion of the actual tag to create or recover the association with the corresponding output. The operation of the content-addressable memory, on average, allows this reduction in the tag recalled from the memory. The decoding process [9] is divided into four steps

- 1) An input tag is reduced in length to q bits and divided into c equally sized partitions. The q bits can be selected within the tag bits in such way to reduce the correlation.
- 2) Local Decoding (LD): A single neuron per cluster in $p1$ is activated using a direct binary-to-integer mapping from the tag portion to the index of the neuron to be activated.
- 3) Global Decoding (GD): GD determines which neuron(s) in $P2$ must be activated based on the results from LD and the stored connection values. If there exists at least one active connection from each cluster in $P2$ toward a neuron in $P2$, that neuron is activated.
- 4) If more than one neuron is activated in $P2$, then, the same number of word comparisons is required to detect the correct match. A single activated neuron means no further comparisons are required.

Next section deals with the sparse clustered network based content addressable memory architecture. In order to exploit the prominent feature of the sparse clustered network associative memory in the classification of the search data, a conventional content- addressable memory array is divided into sufficient number of compare-enabled sub-blocks such that:

- 1) The number of sub blocks are not too many to expand the layout and to complicate the interconnections and
- 2) The number of sub-blocks should not be too few to be able to exploit to energy-saving opportunity with the sparse clustered network based classifier.

Consequently, the neurons in $P2$ are grouped and O Red to construct the compare-enable signal(s) for the content-addressable memory array. Even the conventional content- addressable memory arrays are divided into multiple sub-blocks since long bit lines and SLs can slow down the read, write, and search operations due to the presence of drain, gate, and wire capacitances. The total number of sub-blocks can be selected depending on the silicon-area availability since each sub-block will slightly increase the silicon area. If the input data word is not uniformly distributed, more sub-blocks will be activated during a search consuming higher amounts of energy while them accuracy of the final output is not affected. Therefore, a false-negative output is never generated. However, since the full length of the tag is not used in sparse clustered network based content addressable memory; it is possible to select the reduced- length tag bits depending on the application and according to a pattern to reduce the tag correlation. A complete circuit for sparse clustered network based content addressable memory was implemented and simulated using HSPICE and TSMC 65-nm CMOS technology [9]. The energy consumption of sparse clustered network based content addressable memory depends on various design parameters, such as q , c , and the effect of non uniform input distributions, the energy The total energy consumption is divided into the energy consumption in the sparse clustered network based classifier (ESCN), and the content-addressable memory sub-blocks (ECAM). The SCN-based classifier's contribution to the energy consumption includes decoders, EDec, SRAMs (accessed, ESRAMacc, and idle, ESRAMidle), and the logical gates to perform the GD and to generate the compare enable signals for the content-addressable memory array. Also it includes the content-addressable memory portion of the energy model consists of match (ECAM match) and mismatch (ECAM mismatch) portions, and Static energy consumption of the idle content-addressable memorys (ECAMstat) due to the presence of leakage current occurring in advanced technologies of match (ECAM match) and mismatch (ECAM mismatch) portions. As the value of q is increased, the energy consumption is

decreased as well since the number of comparisons is reduced but up to a point until the energy consumption of the sparse clustered network based classifier itself would dominate that of the content-addressable memory array. Therefore, the energy consumption of the sparse clustered network based classifier is not dependent on the original tag length, and rather on the number of entries in the content-addressable memory array.

Algorithm:

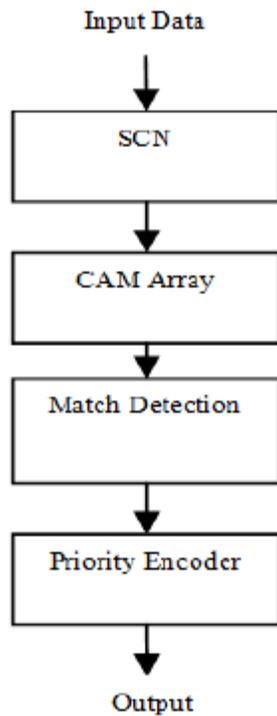


Fig 2.5 shows the Design flow of system.

Once the input is given to SCN classifier, it predicts which CAM block has to be selected from the array. The classifier generates the compare- enable signals to array such that it avoids unnecessary search operations thus reducing the dynamic power

III. Conclusion

This paper reviews implementation of a low power CAM based on sparse clustered network. In addition other CAM architectures and a comparative study on the same with latest proposed architecture are also included here. As a result of this analysis it came to conclusion that the proposed content- addressable memory architecture with sparse clustered network is more advantageous and reliable one as compared to other architecture sparse clustered network content addressable memory employs a novel associatively mechanism based on recently developed family of associative memories based on sparse cluster network. Sparse clustered network addressable memory is suitable for low- power

applications, where frequent and parallel look-up operations are required. Sparse clustered network content-addressable memory employs a sparse clustered network based classifier, which is connected to several independently compare-enabled are enabled once a tag is presented to the sparse clustered network based classifier. By using independent nodes in the output part of sparse clustered network content-addressable memory’s training network, simple and fast updates can be achieved without retraining the network entirely. With optimized lengths of the reduced- length tags, sparse clustered network content-addressable memory eliminates most of the comparison operations given a uniform distribution of the reduced-length inputs. Depending on the application, non uniform inputs may result in higher power consumptions, but does not affect the accuracy of the final result. In other words, a few false-positives may be generated by the sparse clustered network based classifier, which are then filtered by the enabled content-addressable memory sub- blocks.

References

- [1] Yamagata T, Mihara M, Hamamoto T, Murai Y, Kobayashi T, A 288-kb Fully Parallel Content Addressable Memory Using a Stacked-Capacitor Cell Structure, December 1992.
- [2] Schultz K and Gulak P, Fully parallel integrated CAM/RAM using preclassification to enable large capacities, J. Solid-State Circuits, May 1996; 5, 689–699.
- [3] Zukowski C and Wang S Y, Use of selective precharge for low power on the match lines of content-addressable memories, Aug. 1997; 64– 68.
- [4] Lin C S, Chang J C, and Liu B D, A low power precomputation based fully parallel content-addressable memory, J. Solid-State Circuits, April 2003; 4,. 654–662.
- [5] Pagiantzis K and Sheikholeslami K, Pipelined match-lines and hierarchical search-lines for low power content-addressable memories, Sep. 2003; 39, 1512–1519.
- [6] Ruan S J, Wu C Y, and Hsieh J Y, Low power design of precomputation-based content addressable memory, March 2008.
- [7] Onizawa N, Matsunaga S, Gaudet V C, and Hanyu T, High throughput low-energy content addressable memory based on self-timed overlapped search mechanism, May 2012; 41–48.
- [8] Jarollahi H, Onizawa N, Hanyu T and Gross W J, Associative Memories Based on MultipleValued Sparse Clustered Networks, July 2014.

- [9] Jarollahi H, Gripon V, Onizawa N, and Gross W J, Algorithm and Architecture for a Low-Power Content-Addressable Memory Based on Sparse Clustered Networks, April 2014.
- [10] Gripon V and Berrou C, Sparse neural networks with large learning diversity, 2011.
- [11] Pagiamtzis K and Sheikholeslami A, ContentAddressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey, J Solid State Electronics M arch, 2006.
- [12] Peng M and Azgomi S, Content-Addressable memory (CAM) and its network applications, 1996.