Low Power Fault Free Coding Design for Cam Interface

K.Suresh Kumar, Y.Rajasree Rao, K.Manjunathachari

***Abstract:** In the design modeling of a CAM interface, the controlling and access operation defines the performance of memory interfacing. In a CAM application, data stored in the Memory units are mapped to a given query input and an output is developed as a match signal to which as decision is made. in the process of CAM operation, to obtain a faster matching and low power consumption, a new search approach and pattern alignment logic is defined. To improve the storage capacity of a CAM unit, a multi page interface is proposed. To the defined unit a new fault tolerance approach is integrated for a reliable, low power and fast processing CAM application.*

***Index Terms:** CAM unit, low power, fast search, high volume, fault tolerant.*

I. INTRODUCTION

With the diversity in data accessing, and emerging new technologies, new mode of data access are evolving. With this increment, the computation probability of data validation for data exchange or security concern is increasing. In a real time interface to govern data exchange router units are interfaced. Wherein to give a security monitoring firewalls, antivirus etc., are used. these application operate on a permanently defined memory content to which a input data is matched to obtain a validation of given data. this operation is performed with the support of a content based addressable memory (CAM) unit. The CAM units are used as a database storage where a pre defined data are trained and on the requirement accessed to give a result. As the access probability are becoming dynamic with new technologies such as heterogonous network, cloud computing, distributed servers etc., the enhancement in performance reliability and efficiency of CAM interfacing is needed. To improve the performance of CAM interfacing, various developments were made in past. [1], [2] uses finite automata or hashing methods. A simple discrete analyzer or a simple CAM interface has the advantage of performance however the cost of the area implemented is large [3],[4]. The simplest and usual addressing approach overrides the higher area constraint, but reduces the operational performance. A method to increase the design performance by characteristic contrasts of regular expressions applied to minimize search time is outlined in [5-8]. Each character is represented as a single wire resulting in a specific volume of the memory. This approach increases the character input and reduces the gate counts. For binary expression form a binary deciding diagram (BDD) method has been raised to more than one addresses representing a tree core based structure. This minimizes the area of implantation to a low count compared to exiting CAM interface. The CAM access is however needed to be focused for fault tolerance to minimize the error effect. In [9,10] an objective of fault tolerance in memory interface is presented. However, the process of fault tolerance on the interface of memory access is CAM addressing not defined. In recent years, IDS has been proposed to map with different applications for security concern. [11] uses the map to view the rest of the address and the recording in the memory. An FPGA implementation of the bit split string has been introduced in [12]. In [13] method for Connecting the AC State Machines states are suggested to reduce the memory size. Also, to specify the size of the memory, label transition table and a CAM based lookup table is introduced in [14]. In [15, 16] a hash-based pattern that matches the co-processor, using the memory to buffer list of strings and state transitions is outlined. Modifying the pattern algorithm suggested AC algorithm for observing multiple characters at once. Also, the content of the CAM used to match the string is widely used in different applications even under the content shift on the memory.

These developments were focused with the objective of faster mapping approach. however, with the emergence of new technologies and new devices, the constraint in resource demands for new solution. In this paper a new architecture with the objective of faster, low power, high volume and fault free condition is proposed. to outline the suggested approach this paper is defined in 6 sections. Wherein section 2 outlined the conational CAM interface architecture with the proposed architecture. Section 3 outlines the functional detail of the proposed architecture. Section 4 outline the simulation results and section 5 conclude the presented approach.

II. CAM INTERFACE ARCHITECTURE

CAM unit are developed as an interface unit in accessing data from a pre allocated memory location . where in the match of content, a data is retrieved. The conventional operation of CAM unit is illustrated in figure 1.

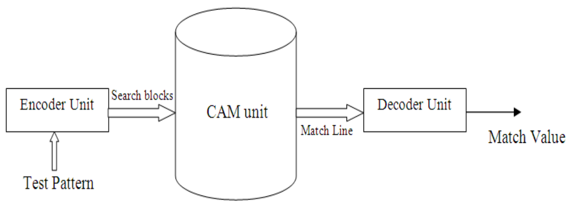


Figure 1: Interface unit for CAM unit

Manuscript published on 28 February 2019.
***Correspondence Author(s)**
K Suresh Kumar, Department of Electronics and Communication Engineering, SSJ Engineering College, Hyderabad, India.
Y. Rajasree Rao, Department of Electronics and Communication Engineering, St.Peter's Engineering College, Hyderabad, India.
K. Manjunathachari, Department of Electronics and Communication Engineering, GITAM University, Hyderabad, India.

The unit is operated by a encoder and a decoder unit, where, the operation architecture of the encoder and the decoder unit is illustrated in figure 2.

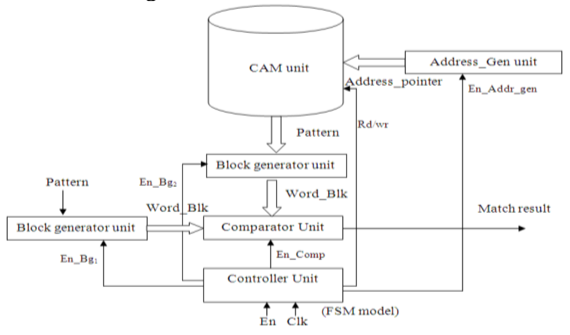


Figure 2: conventional architecture for CAM interface unit [4]

The conventional interface unit, is observed to process the operation with a block generator unit, where the given pattern is transformed to a word block and passed to a comparator unit for match generation. The controller unit generate a FSM search operation of each search stage to make a decision. Wherein the CAM interface is observed to be operationally well designed, following constraint were observed with this design. The Redundant patterns in the memory unit are making the CAM operation slower. The stored data at the memory unit have High transition patterns consume a large power in storage and operation. In the storage of high volume data a linear memory element for high volume storage is constraint in operation, and leads to slower operation. The storage error due to memory fault and Fault conditions in memory unit is resulting in wrong match output. To overcome the stated issues a new CAM interface architecture [20,21] is proposed. The modified architecture of the CAM interface unit is presented in Figure 3.

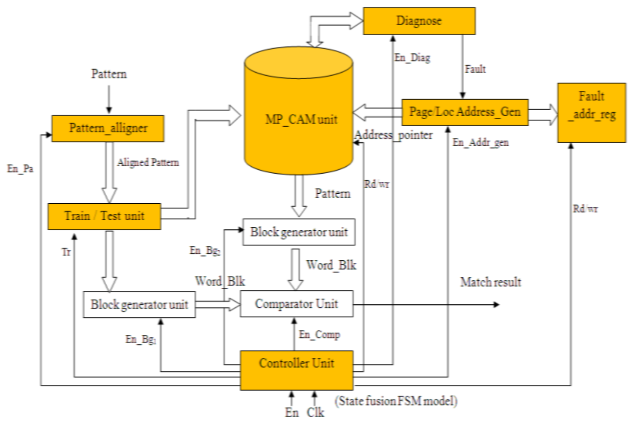


Figure 3: Proposed CAM interfacing architecture

The modified units of the proposed architecture are highlighted in the figure shown. The operation outlined of the proposed architecture is presented in in following section.

III. FUNCTIONAL OUTLINE

The function operation of the proposed architecture operates in 3 working modes- Diagnosis, train and test mode. In the diagnosis mode, self-test operation is made. The memory is filled with ‘0’ and ‘1’ to test stuck-0 or stuck-1 fault. Fault address are stored in Fault_address register. In train mode patterns are buffered onto the memory unit of CAM. The operation of the self test operation is illustrated in Figure 4.

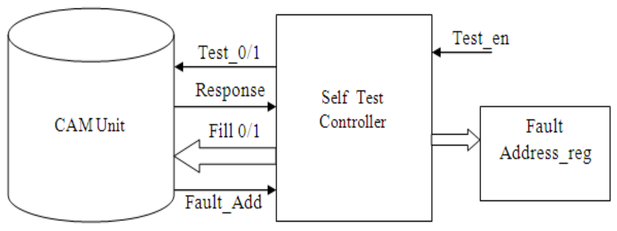


Figure 4: Fault testing operation in the CAM interface

The train signal is set high and patterns are given as input. Patterns are passed through aligner unit to give a shift alignment to reduce pattern transitions. Realigned patterns are buffered into addressed location. The address location are validated with fault_address register to fill data. A redundancy of stored pattern is evaluated using hamming distance and new state transitions in controller unit is defined outlined in [18].

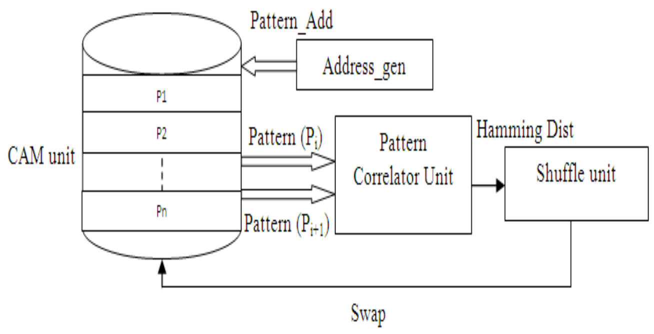


Figure 5: pattern aligned unit for minimal match overhand

On the test mode, a test pattern is given as input. The pattern is realigned and divided to blocks of 4 bit word. The word is passed to a comparator unit. The controller enable the reading of memory content by enabling address_gen unit. The controller operates on a FSM mode where for the pattern testing a state flow using the conventional FSM flow and the proposed flow is illustrated in figure below. For a test pattern $abcd$ and $bbda$ the state transition is observed as,

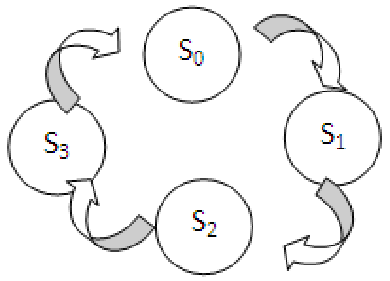


Figure 6: State transition flow in pattern matching using conventional approach

For a given pattern sequence is observed as,

$$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_0$$
$$a \rightarrow b \rightarrow c \rightarrow d$$

For the pattern $bbda$ again 4 state transitions are made,

$$S_0 \to S_1 \to S_2 \to S_3 \to S_0$$
$$b \to b \to d \to a$$

The total state transient observed is 8. Wherein it is observed that, a common pattern can move on to the same stage defined as,

$$\begin{tikzcd} a \arrow[r] & b \arrow[r] \arrow[rd, dashed] & c \arrow[r] & d \\ b \arrow[r] & b \arrow[u, dashed] \arrow[r] & d \arrow[r] & a \end{tikzcd}$$

This will result in a new state transition as,

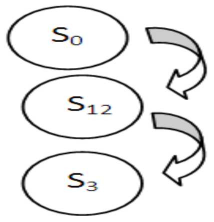


Figure 7: The realigned pattern match in proposed system

The number of transition observed is minimized by 3 from 4 stage and a total stage is then observed to be 7.The new FSM controller is given in the controller unit and the pattern are passed on matching based on the selected state transition. To read the pattern, the address_gen generate a page address and location address to the address pointer. The address is validated from fault_address register and memory data is passed to the comparator A match signal is buffered and passed to match decoder. To Conserve the power consumption in the CAM operation a aligner unit is proposed [19]. The aligner minimizes the transition pattern by linear shift operation illustrated in figure below.

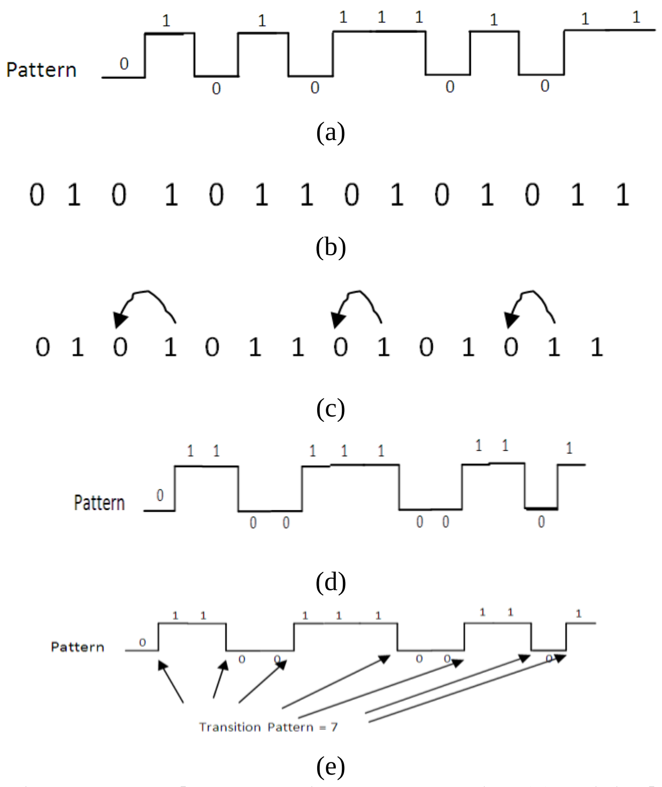


Figure 8: Reorder pattern in CAM operation (a) Original pattern, (b) Binary pattern, (c) Realignment operation, (d) Realigned Pattern, (e) Transition count

The power dissipation defined by,

$$P_d = \sum_{line} C_m V_{DD}^2 N_{tr} \qquad (1)$$

Where N_{tr} are the number of tansiton observed, the total power is conserved based on the minimikzation of transiton. The aligned patterns are stored and used for matching with use less power in logical transitions. To enhance the volume of data accessing in this memory interface, a multi page memory unit is defined. The interface unit is defined with a multi page accessing. The accessing utterance architecture is presented in figure below.

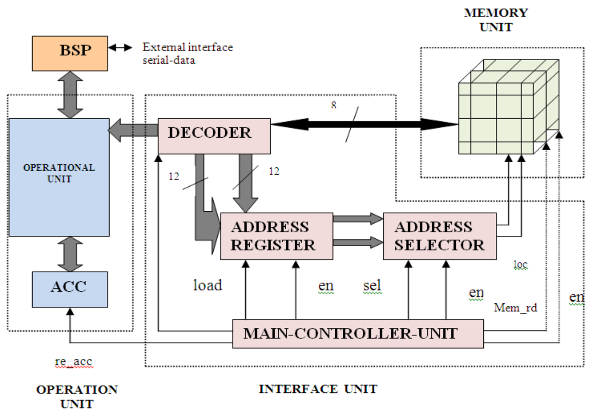


Figure 9: Proposed system architecture for CAM interface in Multi page memory

The integrated aligner unit gives the advantage of low power consumption. The correlative pattern fusion in state machine leads to faster search operation. The new address_gen unit [22] gives page and location addressing resulting in volumetric data buffer in a lower dimension. The integrated fault diagnosis unit, results in reliable operation in CAM interfacing.

IV. SIMULATION RESULTS

To evaluate the performance of the proposed approach a timing result using a HDL definition for the encoder, decoder and memory unit is developed based on VHDL coding, and simulated over Aldec's Active HDL for timing operational validation. To obtain the realization logic over a targeted PLD, Xilinx FPGA device of Virtex family is considered, and the implementation details gives the realization requirement for PLD implementation. The obtained observations are illustrated in below figures.

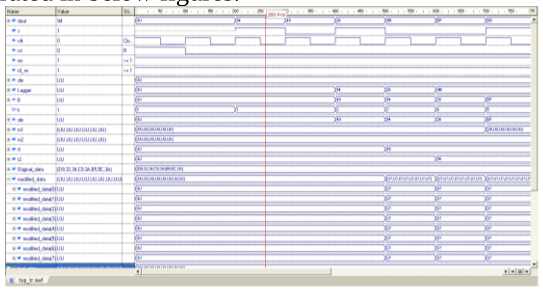


Figure 10. Simulation plot showing the observations made for the developed Sequencing algorithm

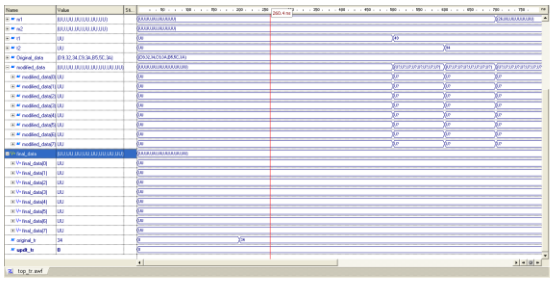


Figure 11.Figure illustrating the original data the modified data regenerated after coding

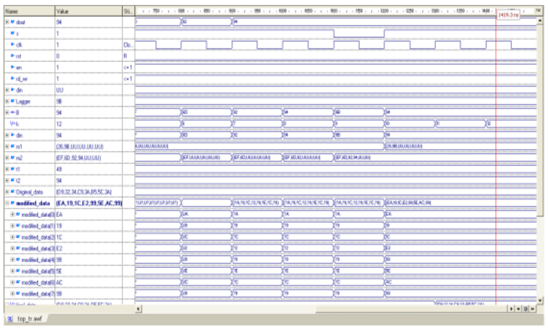


Figure 12 figure illustrating the bus , Sequencing and input data line for the designed encoder and decoder unit

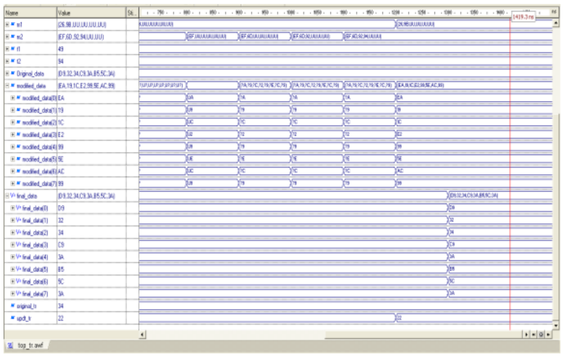
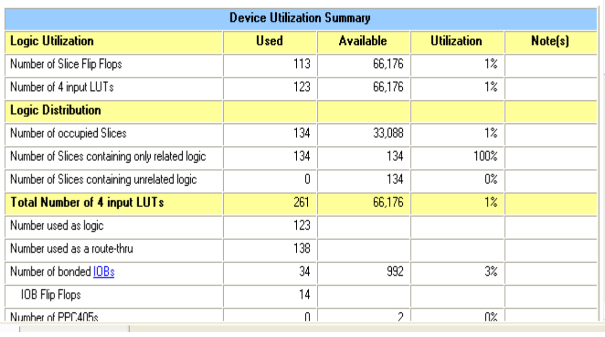


Figure 13. Illustrating the content of the memory buffered data for regeneration and the reconstructed data from it.

The implementation of the proposed work onto the Xilinx device is carried out, the observed implementation summery is illustrated below.



Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	113	66,176	1%	
Number of 4 input LUTs	123	66,176	1%	
Logic Distribution				
Number of occupied Slices	134	33,088	1%	
Number of Slices containing only related logic	134	134	100%	
Number of Slices containing unrelated logic	0	134	0%	
Total Number of 4 input LUTs	261	66,176	1%	
Number used as logic	123			
Number used as a route-thru	138			
Number of bonded IOBs	34	992	3%	
IOB Flip Flops	14			
Number of PPC405s	0	2	0%	

Figure 14: Implementation summery of the synthesized proposed architecture

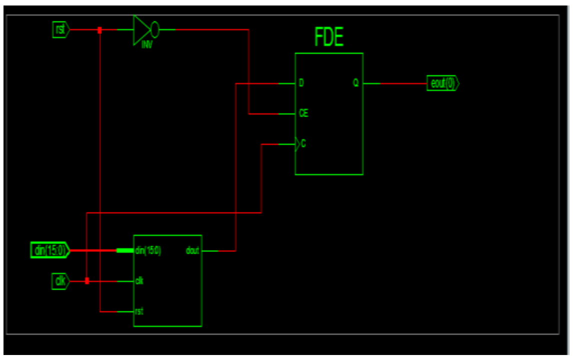


Figure 15: RTL visualization of the Implemented system

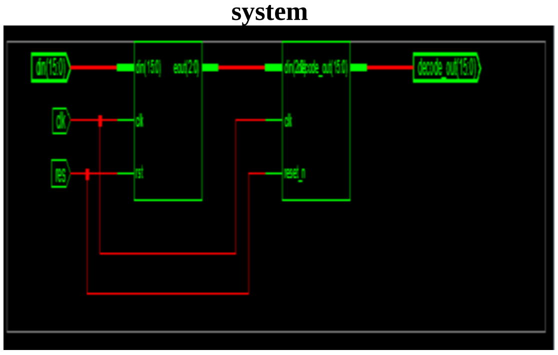
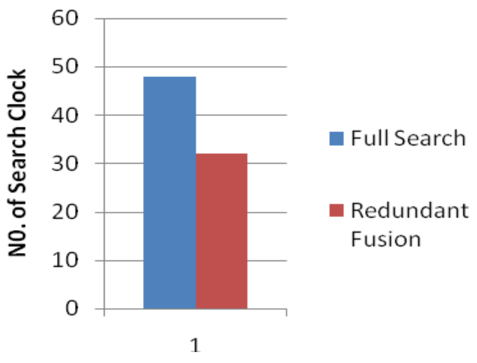


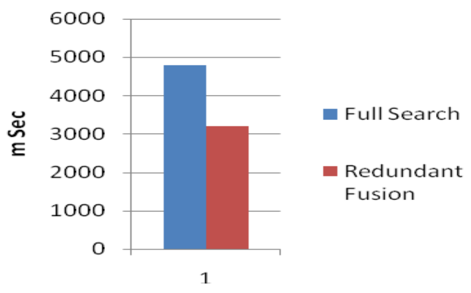
Figure 16: data flow in the Implemented design

A comparative analysis of this implementation is outlined in below tables. For a 2KB, (2048 x 16 bit) memory unit, testing on 2 test patterns under a 100msec clock cycle.

a) Pattern Search Speed comparison



(a)



(b)

Figure 17: (a) Number of search clocks for pattern matching, (b) Time taken for search (mSec)

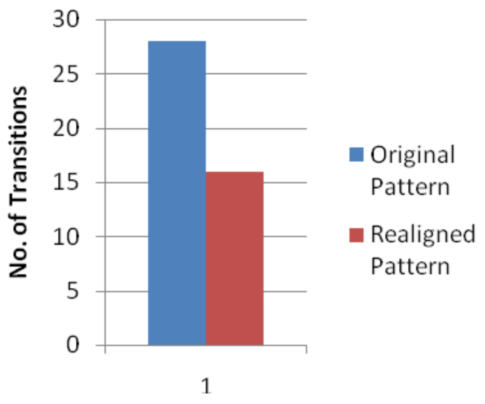
The observation for the method is outlined in the table 1 below.

Table 1: Observation of search clock and processing time for full search and redundant fusion method

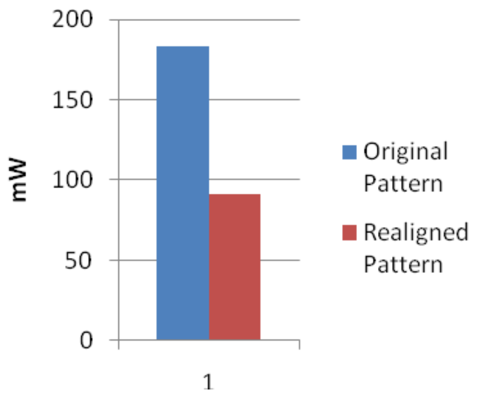
	No. of search clocks	Time taken (mSec)
Full search method	48	4800
Redundant fusion method	32	3200

Search speed is improved by 1600mSec ie. 16 clock cycles.

b) Power comparison



(a)



(b)

Figure 18: (a) Number of binary transitions, (b) power consumed

Table 2: Observation of number of binary transitions and the power consumed for the original and realigned pattern

	No. of Transitions	Power consumed
Original pattern	28	183mW
Realign pattern	16	91mW

Power consumption is reduced by, 92mW

c) Overhead comparison

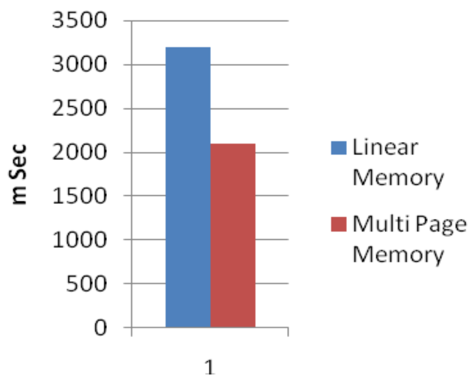
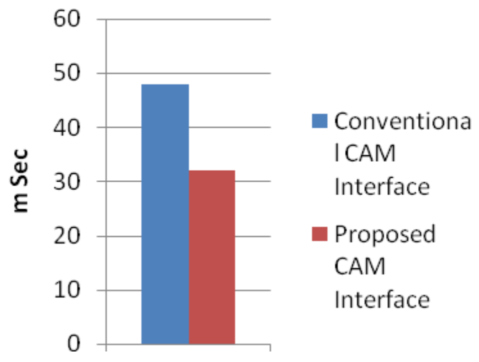


Figure 19: Read/Write delay

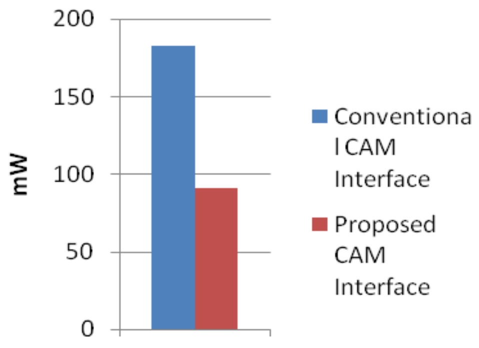
Table 3: Observation of memory capacity and time delay in read/write operation (mSec) for the linear and multipage memory

	Memory capacity	Read/write time(mSec)
Linear memory unit	6KB	3200
Multi page memory unit	6KB (2x3 page)	2100

d) Overall observation



(a)



(b)

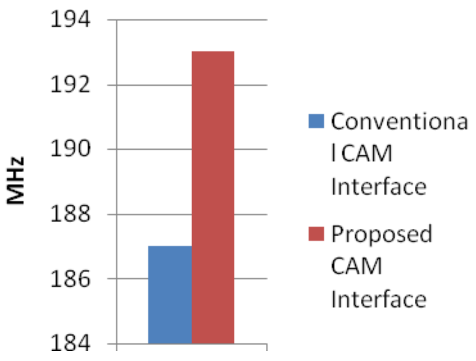
Figure 20: (a) Search Speed, (b) Power consumed

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

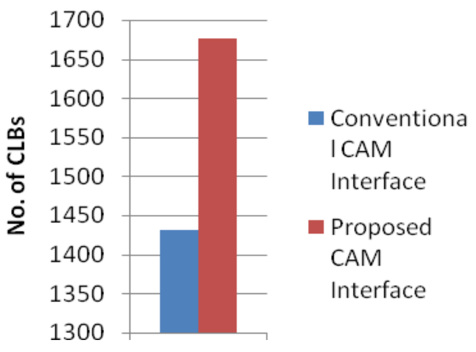
Table 4: Observation of search speed, power consumed and Data capacity for the developed CAM over conventional CAM interface

	Speed (mSec)	Power (mW)	Data Capacity
Conventional CAM interface	48	183	M
Proposed CAM interface	32	91	M×P

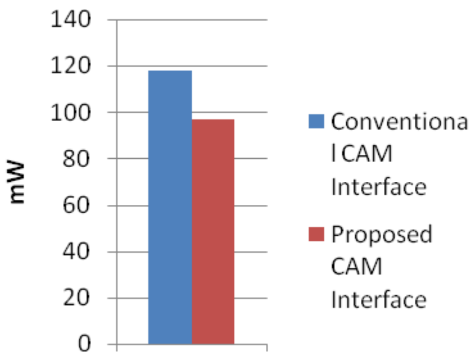
e) Implementation observation over Xilinx virtex2P (2vpx70ff1704-6)



(a)



(b)



(c)

Figure 21: Targeted FPGA parameter (a) Processing Speed (MHz), (b) No. of CLBs in implementation (c) Power consumption (mW)

Table 5: Observation of Device speed, Area coverage, power consumed and the Number of I/Os for the developed CAM interface over the conventional design

	Device speed (MHz)	Area (CLB's)	Power (mW)	I/O's
Conventional CAM interface	187	1432	118	33
Proposed CAM interface	193	1676	97	33

V. Conclusion

This paper outlined a new interface architecture for CAM operation giving the advantage of Higher speed of searching operation, Low power consumption in storage and processing. A high volume storage interface, with controlling in page memory interface and a fault tolerant system for CAM interfacing, where the trained patterns are preserved for fault free condition. The CAM interface is given with the provisioning of self testing in both stuck 0 and stuck 1 operation and the pattern aligner unit save the transition hence minimizing the matching overhead and power consumption. In the CAM interfacing it is needed that, the access is to be correct and faster, this objective id developed here by the integration of a fault free design of CAM accessing in CAM interface by developing an new interface architecture for memory accessing using fault control address register and FSM merge logic in pattern reading for CAM application.

REFERENCES

1. A. V. Aho and M. J. Corasick, "Efficient string matching: An AID to bibliographic search," Commun. ACM, vol. 18, no. 6, pp. 333–340, 1975.
2. Derek Pao1, Wei Lin2,1, and Bin Liu, "Pipelined Architecture for Multi-String Matching", IEEE Computer Architecture Letters Vol. 7, 2008.
3. M. Alicherry, M. Muthuprasanna, and V. Kumar, "High speed address matching for network IDS/IPS," in Proc. IEEE Int. Conf. Netw. Protocols (ICNP), 2006, pp. 187–196.
4. B. Brodie, R. Cytron, and D. Taylor, "A scalable architecture for high-throughput regular-expression address matching," in Proc. 33rd Int. Symp. Comput. Arch. (ISCA), 2006, pp. 191–122.
5. Z. K. Baker and V. K. Prasanna, "High-throughput linked-address matching for intrusion detection systems," in Proc. Symp. Arch. For Netw. Commun. Syst. (ANCS), Oct. 2005, pp. 193–202.
6. Y. H. Cho and W. H. Mangione-Smith, "A address matching co-processor for network security," in Proc. 42ndIEEE/ACM Des. Autom. Conf., Anaheim, CA, Jun. 13–17, 2005, pp. 234–239
7. Y. H. Cho and W. H. Mangione-Smith, "Fast reconfiguring deep packet filter for 1+Gigabyte Network," in Proc. 13th Ann. IEEE Symp. Field Program. Custom Comput. Mach. (FCCM), 2005, pp. 215–224
8. C. R. Clark and D. E. Schimmel, "Scalable address matching on high speed networks," in Proc. 12th Ann. IEEE Symp. Field Program. Custom Comput. Mach. (FCCM), 2004, pp. 249–257.
9. Ali, M., Welzl, M., Hessler, S. and Hellebrand, S. (2007) 'An efficient fault tolerant mechanism to deal with permanent and transient failures in a network on chip', *Int. J. High Performance Systems Architecture*, Vol. 1, No. 2, pp.113–123
10. Dou, W., Miao, S. and Li, Y. (2015) 'Fault-tolerant parallel computing for DEM data blocks with layered dependent relationships based on redundancy mechanism', *Int. J. High Performance Computing and Networking*, Vol. 8, No. 4, pp.337–344.

11. S. Dharmapurikar and J. Lockwood, "Fast and scalable address matching for content filtering," in Proc. Symp. Arch. for Netw. Commun. Syst. (ANCS), Oct. 2005, pp. 183–192.
12. E. C. Oh and P. D. Franzon, ''Design considerations and benefits of three-dimensional ternary content addressable memory,'' in Proc. IEEE Custom Integr. Circuits Conf., 2007,pp. 591–594.
13. D. Bhattacharya, A. Bhoj, and N. Jha,''Design of efficient content addressable memories in high-performance FinFET technology,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 5,pp. 963–967, May 2015.
14. A. McAuley and P. Francis, ''Fast routingtable lookup using CAMs,'' in Proc. IEEE12th Annu. Joint Conf. IEEE Comput.Commun. Soc., Netw.: Found. Future,1993, vol. 3, pp. 1382–1391.
15. F. Yu, R. Katz, and T. Lakshman,''Gigabit rate packet pattern-matchingusing TCAM,'' in Proc. 12th IEEEInt. Conf. Netw. Protocols, Oct. 2004,pp. 174–183.
16. K. Eshraghian, ''Memristormos contentaddressable memory (MCAM): Hybridarchitecture for future high performancesearch engines,'' IEEE Trans. Very LargeScale Integr. (VLSI) Syst., vol. 19, no. 8,pp. 1407–1417, Aug. 2011.
17. Q. Guo, X. Guo, Y. Bai, and E. Ipek,''A resistive TCAM accelerator fordata-intensive computing,'' in Proc. 44thAnnu. IEEE/ACM Int. Symp. Microarchitect.,2011, pp. 339–350.
18. K.Suresh Kumar, Y.Rajasree Rao, K. Manjunathachari, "Fast Map-Addressing for Content Addressable Memories using Register Reconfiguration", 5th IEEE International Conference on Communication and Signal, 2016.
19. K.Suresh Kumar, Y.Rajasree Rao, K. Manjunathachari, "Low-Power Correlative Register Sequencing for Content Addressable Memory", International Journal of Engineering Studies (IJES) Vol 8, 2016.
20. K.Suresh Kumar, Y.Rajasree Rao, K. Manjunathachari, "A Fault Tolerance Approach For Multi-Page Content Addressable Memory", 4th IEEE International Conference on Innovations in Information, Embedded And Communication Systems(ICIIECS-2017), March 2017.
21. K.Suresh Kumar, Y.Rajasree Rao, K. Manjunathachari, " Robust Fault Tolerance In Content Addressable Memory Interface", IOSR Journal of VLSI And Signal Processing (IOSR-JVSP), Volume 7, Issue 3, Ver. I, May 2017.
22. K.Suresh Kumar, Y.Rajasree Rao, K. Manjunathachari, "Address Mapping in Content Addressable Memory Interface With a Low Power Approach", International Journal of Engineering Research and Development(IJERD), Volume 13, Issue 12, Ver I, December, 2017