

## Comparative Power Analysis of Precomputation Based Content Addressable Memory

M. Arun and A. Krishnan

Department of ECE, KSR College of Engineering, Tiruchengode 637215, India

---

**Abstract: Problem statement:** In this study we discuss about Signature Detection Technique (SDT) used in Network Intrusion Detection System (NIDS). Design of SDT using Content Addressable Memory (CAM) is discussed. **Approach:** Two novel architectures, XOR and ones count based Pre computation CAM architectures are proposed and implemented in third party back end tool with 0.18  $\mu\text{m}$  technology, power consumptions are compared. **Results:** Proposed architecture consumes 90% less power added with 5.2% increment in speed. **Conclusion:** Power reduction was achieved by reducing the number of bit comparisons of pre computation technique.

**Key words:** Network Intrusion Detection System (NIDS), Content Addressable Memory (CAM), bloom filter, Signature Detection Technique (SDT), pre computation, implementation results, hardware techniques, Internet packets

---

### INTRODUCTION

Some content strings of Internet packet payload, also known as “signatures,” imply network intrusion attempts. Signature based Network Intrusion Detection System (NIDS) collects these signatures and scans the payload of the Internet packets for them in order to identify, deter and contain such malicious behaviors. A scalable and fast solution is needed to accommodate the largest signature set today and to sustain the real time processing of the high-speed network. This is very challenging especially for today’s high-speed networks with line speeds of 10 GBPS and beyond. Software based NIDSs (Sendil and Nagarajan, 2009; Yusof *et al.*, 2010; Mohammed *et al.*, 2010) are not scalable to high-speeds. Hardware NIDSs have gained a lot of attention recently due to the intrinsic speed advantage over software systems. Two enhanced novel architectures, XOR based Pre Computation CAM (XPCAM) and Ones Count Pre Computation CAM (OCCAM) are studied and implemented in third party back end tool with 0.18  $\mu\text{m}$  technology. The contributions of this study are as follows:

- Detailed Hardware architectures XPCAM and OCCAM are presented. Instead of working with device or technology, we concentrate on the pre computation techniques to improve the speed and reduce the power consumption

- Proposed architectures are implemented using third party back end tool with 0.18  $\mu\text{m}$  technology and power consumptions are compared

The rest of the study is organized as follows. Related work summarizes the related work on hardware-based Signature Detection Techniques. The problem of CAM based pre computation technique using ones count and XOR respectively are explained. Implementation results of the proposed architectures are presented. Finally, power consumptions of both methods are compared and analysed.

**Related work:** Conventional RAM is used in Signature detection techniques with the cost higher power consumption due to more number of inputs and computations. Number of inputs is reduced by neglecting the address lines in CAM architectures. In the past decade, much research on energy reduction has focused on the circuit and technology domains ((Pagiamtzis and Sheikholeslami, 2006) provides a comprehensive survey on CAM designs from circuit to architectural levels). Several works on reducing CAM power consumption have focused on reducing match-line power (Miyatake *et al.*, 2001; Arsovski *et al.*, 2003). The reduction in number of comparisons may help to improve system performance and conserve system resources such as network bandwidth, memory capacity and disk space. Instead of working in device level, our approach concentrates on pre computation techniques. CAM is a critical device for applications involving communication networks, Local Area

---

**Corresponding Author:** M. Arun, Department of ECE, KSR College of Engineering, Tiruchengode 637215, India

Network bridges/switches, databases, lookup tables and tag directories, due to its high-speed data search capability (Yang and Mareboyana, 2009). Figure 1 shows the memory organization of the Pre Computation based (OCCAM) architecture proposed by (Pagiamtzis and Sheikholeslami, 2006) which consists of data memory, parameter memory and parameter extractor, where  $k \ll n$ . To reduce massive comparison operations for data searches, the operation is divided into two parts. In the first part, the parameter extractor extracts a parameter from the input data, which is then compared to parameters stored in parallel in the parameter memory. If no match is returned in the first part, it means that the input data mismatch the data related to the stored parameter. Otherwise, the data related to those stored parameters have to be compared in the second part. It should be noted that although the first part must access the entire parameter memory, the parameter memory is far smaller than that of the CAM. Moreover, since comparisons made in the first part have already filtered out the unmatched data, the second part only needs to compare the data that match from the first part. The OCCAM exploits this characteristic to reduce the comparison operations, thereby saving power. Therefore, the parameter extractor of the OCCAM is critical, because it determines the number of comparison operations in the second part. As we stated previously, the parameter extractor plays a significant role since this circuit determines the number of comparison operations required in the second part. Therefore, the design goal of the parameter extractor is to filter out as many unmatched data as possible to minimize the required number of comparison operations in the second part. The ones-count function was adapted to perform parameter extraction in (Lin *et al.*, 2003). For ones count approach, with an  $n$  bit data length, there are  $n+1$  type of one's count (from 0 ones to  $n$  ones count). Further, it is necessary to add an extra type of one's count to indicate the availability of stored data. Therefore, the minimal bit length of the parameter is equal to  $\log(n+2)$ . The parameter extractor for the ones-count approach is implemented with full adders as shown in Fig. 2. We used a 14-bit example to illustrate the ones-count PB-CAM system and discuss the disadvantages by mathematical analysis. For a 16-bit length input data, all the input data contain  $2^{16}$  numbers and the number of input data related to the same parameter for ones count approach is  ${}_{16}C_r$ , where  $r$  is a type of one's-count (from 0 to 15 one's-counts). Then we can compute the average probability that the parameter occurs. The average probability can be determined by:

$$\text{Average probability} = {}_{16}C_r / 2^{16} \quad (1)$$

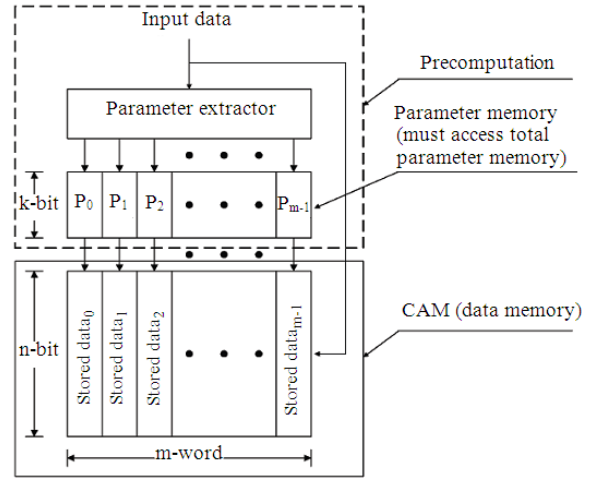


Fig. 1: PB-CAM Architecture

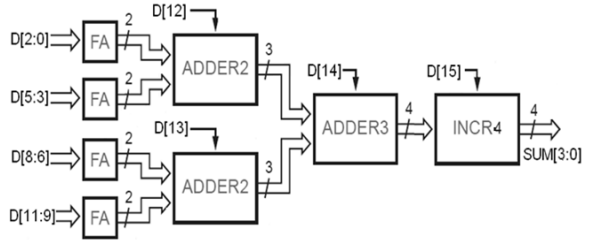


Fig. 2: One's count parameter extractor

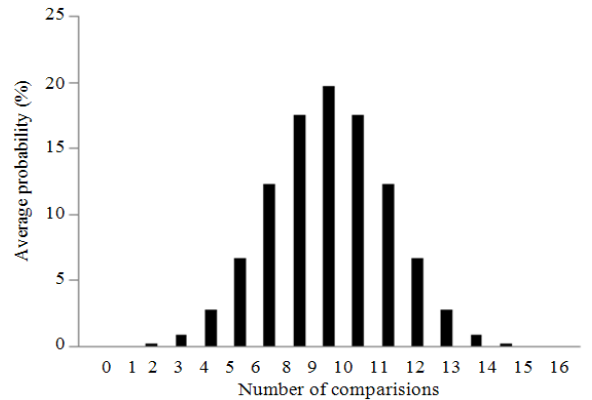


Fig. 3: Gaussian distribution

Figure 3 shows the number of data related to the same parameter and their average probabilities for the input data that is 16-bit in length. Note that with conventional CAMs, the comparison circuit must compare all stored data, whereas with OCCAMs, a large amount of unmatched data can be initially filtered

out, reducing comparison operations for minimum power consumption in some cases. However, the average probabilities of some parameters, such as 0, 1, 2, 3, 13, 14, 15 and 16 are less than 1%. In Fig. 3, we can see that parameters with over 2000 comparison operations range between 4 and 12. However, the summation of the average probabilities for these parameters is close to 92%. Although the number of comparison operations required for OCCAM s is fewer than that of conventional CAMs, OCCAM s fail to reduce the number of comparison operations in the second part when the parameter value is between 5 and 9, thereby consuming a large amount of power. As can be seen in Fig. 3, random input patterns for the ones-count approach demonstrate the Gaussian distribution characteristic. Note that the Gaussian distribution will limit any further reduction of the comparison operations in OCCAM.

**MATERIALS AND METHODS**

**XPCAM:** The key idea behind our method is to reduce the number of comparison operations by eliminating the Gaussian distribution. For a 16-bit input data, if we can distribute the input data uniformly over the parameters, then the number of input data related to each parameter would be  $2^{14}/16= 4096$  and the maximum number of required comparison operations would be  $2^{14}/16= 4096$  for each case in the second part of the comparison process. Compared with the ones-count approach, this approach can reduce comparison operations by a minimum of 3912 and a maximum of 8774 (i.e., for parameter value is from 4-12) for 92% of the cases. Based on these observations, we propose a new parameter extractor called Block-XOR, which is shown in Fig. 4 to achieve the previous requirement. In our approach, we first partition the input data bit into several blocks, from which an output bit is computed using XOR logic operation for each of these blocks. The output bits are then combined to become the input parameter for the second part of the comparison process. To compare with the ones-count approach, we set the bit length of the parameter to  $\log n$ , where  $n$  is the bit length of the input data. Therefore, the number of blocks is  $n/\log n$  in our approach. Accordingly, all the blocks contain 4 inputs as shown in the upper part of Fig. 4. We added a multiplexer to select the correct parameter. The selected signal is defined as:

$$S = A_3A_2A_1A_0 \tag{2}$$

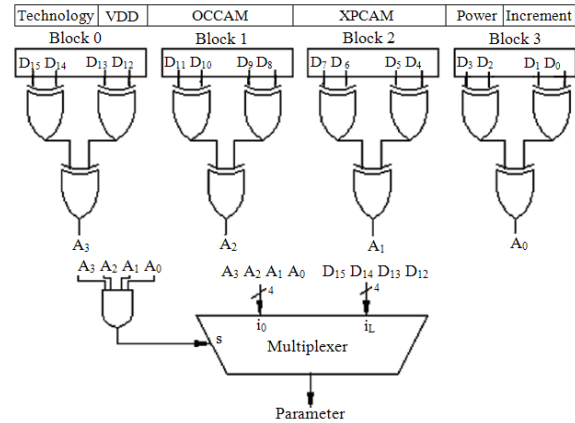


Fig. 4: XOR based Pre computation block

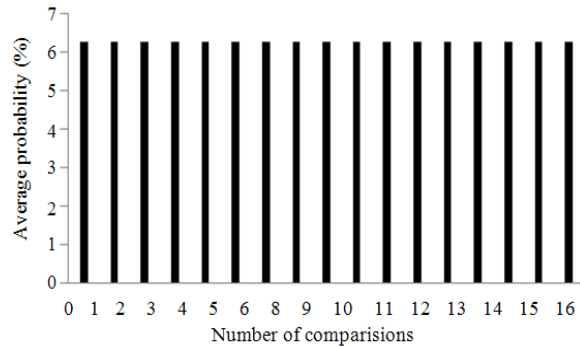


Fig. 5: Uniform distribution

The concept of Block-XOR approach is to uniformly distribute the parameter over the input data. By the rule of product, the number of input data that results in the same parameter is  $8 \times 8 \times 8 \times 8 = 4096$ . Consequently, the average probability can be determined as  $4096/2^{16} \times 100\% = 6.25\%$ . Obviously, the concept of Block-XOR approach can reduce the comparison operations, hence minimize power consumption. Figure 4 shows the number of input data that result in the same parameter for the proposed XPCAM. As can be seen from Fig. 3 and 5, in most cases, the proposed XPCAM required far fewer comparison operations than the ones-count approach for parameter values between 4 and 12.

**RESULTS AND DISCUSSION**

OCCAM and XPCAM, both are implemented in a third party back end tool and power consumptions are compared for different technology. Table 1 shows that XPCAM consumes more than 90% less power in all technology added with 5.2% increment in speed. Due to the reduction of bit comparisons in XPCAM, results were achieved.

Table 1: Implementation results

Technology ( $\mu\text{m}$ )	OCCAM				XPCAM				Power saving (%)	Increment in speed (%)
	VDD (V)	I (mA)	Power (mW)	Speed (Hz)	I (mA)	Power (mW)	Speed (Hz)			
0.8	5.0	2.1	36.482	474	2.6	4.057	500	88.88	5.2	
0.6	5.0	2.6	46.242	474	2.8	3.537	500	92.35	5.2	
0.35	3.5	1.8	23.211	474	1.9	1.409	500	93.93	5.2	
0.25	2.8	1.8	18.493	474	0.9	0.418	500	97.74	5.2	
0.18	3.3	0.9	5.083	474	0.8	0.161	500	96.83	5.2	

**CONCLUSION**

In this study, we presented the implementation of low power signature detection technique based on CAM with ones complement and XOR based pre computation techniques. Implementation results shows that XPCAM consumes less power with high speed than OCCAM by reducing the number of bit comparison operations. This study focuses on detection of known attacks as signatures contained in a single packet. As future work, we would like to extend the approach of this study to detect unknown attacks using hardware techniques.

**REFERENCES**

Arsovski, I., T. Chandler and A. Sheikholeslami, 2003. A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme. *IEEE J. Solid-State Circ.*, 38: 155-158. DOI: 10.1109/JSSC.2002.806264

Lin, C.S., J.C. Chang and B.D. Liu, 2003. A low-power precomputation-based fully parallel content-addressable memory. *IEEE J. Solid-State Circ.*, 38: 622-654. DOI: 10.1109/JSSC.2003.809515

Miyatake, H., M. Tanaka and Y. Mori, 2001. A design for high-speed low-power CMOS fully parallel content-addressable memory macros. *IEEE J. Solid-State Circ.*, 6: 956-968. DOI: 10.1109/4.924858

Mohammed, M., T. Bouchentouf, A. Benazzi and M. Azizi, 2010. Amelioration of attack classifications for evaluating and testing intrusion detection system. *J. Comput. Sci.*, 6: 716-722. DOI: 10.3844/jcssp.2010.716.722

Pagiamtzis, K. and A. Sheikholeslami, 2006. Content-Addressable Memory (CAM) circuits and architectures: A tutorial and survey. *IEEE J. Solid-State Circ.*, 41: 712-727. DOI: 10.1109/JSSC.2005.864128

Sendil, M.S. and N. Nagarajan, 2009. Analyzing the peer to peer traffic aggregation using an optimized method. *J. Comput. Sci.*, 5: 738-744. DOI: 10.3844/jcssp.2009.738.744

Yang, B. and M. Mareboyana, 2009. Progressive content-sensitive data retrieval in sensor networks. *J. Comput. Sci.*, 5: 529-535. DOI: 10.3844/jcssp.2009.529.535

Yusof, M.M., M. Omar and Z. Shukur, 2010. A disruption-tolerant model for building a mobile application using web service. *J. Comput. Sci.*, 6: 1430-1437. DOI: 10.3844/jcssp.2010.1430.1437