ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

A Review-Architecture for Matching of Data Encoded with Hard Systematic Error Correcting Code Using FPGA for Low Complexity and Low Latency

Priyanka A. Wathodkar, Radhika Harne

M. Tech Student, Dept. of Electronics and Tele-Communication, Government College of Engineering, Amravati, India

Assistant Professor, Dept. of Electronics and Tele-Communication, Government College of Engineering Amravati, India

ABSTRACT:To reduce latency and complexity, a new architecture for matching of data protected with an error correcting code (ECC) is presented in brief. The proposed architecture parallelizesthe comparison of the data and that of the parity information based on the fact that the codeword of an ECC is usuallyrepresented in a systematic form consisting of the raw data and the parityinformation generated by encoding. In order to furtherreduce the latency and complexity, a new butterfly-formedweight accumulator (BWA) architecture is proposed for the efficient computation ofthe Hamming distance. Grounded on the BWA, the proposed architectureexamines whether the incoming data matches the stored data if a certainnumber of erroneous bits are corrected.

KEYWORDS: Data comparison, error correcting codes (ECCs), Hamming distance, systematic codes, tag matching

I. INTRODUCTION

Data comparison is widely used in computing systems to perform many operations such as the tag matching in a cache memory and the virtual-to-physical address translation in a translation lookaside buffer (TLB). Because of such prevalence, it is important to implement the comparison circuit with low hardware complexity. Besides, the data comparison usually resides in the critical path of the components that are devised to increase the system performance, e.g., caches and TLBs, whose outputs determine the flow of the succeeding operations in a pipeline. The circuit, therefore, must be designed to have as low latency as possible, or the components will be disqualified from serving as accelerators and the overall performance of the whole system would be severely deteriorated. As recent computers employ error-correcting codes (ECCs) to protect data and improve reliability [2]–[6], complicated decoding procedure, which must precede the data comparison, elongates the critical path and exacerbates the complexity overhead. Thus, it becomes much harder to meet the above design constraints. Despite the need for sophisticated designs as described, the works that cope with the problem are not widely known in the literature since it has been usually treated within industries for their products. Recently, however, [7] triggered the attraction of more and more attentions from the academic field. The most recent solution for the matching problem is the direct compare method [7], which encodes the incoming data and then compares it with the retrieved data that has been encoded as well. Therefore, the method eliminates the complex decoding from the critical path. In performing the comparison, the method does not examine whether the retrieved data is exactly the same as the incoming data.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

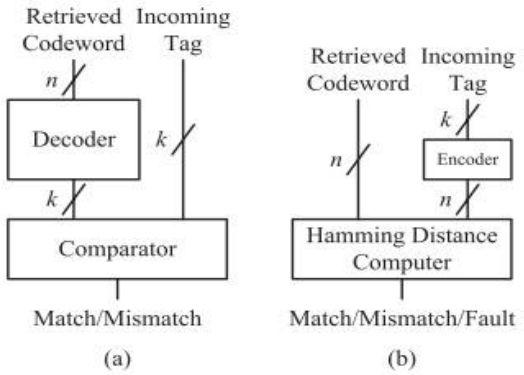

Fig. 1. (a) Decode-and-compare architecture and (b) encode-and-compare architecture.

Instead, it checks if the retrieved data resides in the error correctable range of the codeword corresponding to the incoming data. As the checking necessitates an additional circuit to compute the Hamming distance, i.e., the number of different bits between the two codewords, the saturate adder (SA) was presented in [7] as a basic building block for calculating the Hamming distance. However, [7] did not consider an important fact that may improve the effectiveness further, a practical ECC codeword is usually represented in a systematic form in which the data and parity parts are completely separated from each other [8]. In addition, as the SA always forces its output not to be greater than the number of detectable errors by more than one, it contributes to the increase of the entire circuit complexity. In this brief, we renovate the SA-based direct compare architecture to reduce the latency and hardware complexity by resolving the aforementioned drawbacks. More specifically, we consider the characteristics of systematic codes in designing the proposed architecture and propose a low-complexity processing element that computes the Hamming distance faster. Therefore, the latency and the hardware complexity are decreased considerably even compared with the SA based architecture.

II. RELATED WORK

A Promising Solution for Comparison of Data Coded With Error-Correcting Codes

Data comparison circuit usually resides in the critical path of components that are intended to increase the system performance. So these circuits should be designed to have very low latency and complexity as possible to increase the overall performance of the system. A promising solution for comparison of data which is protected by Error-Correcting Codes (ECC) is presented in this brief to reduce latency and complexity. A practical ECC codeword is usually represented in systematic form in which data part and parity part are completely separated. This property of systematic codeword is used for proposing a new architecture for data comparison. In addition, a new butterfly-formed weight accumulator (BWA) is introduced for the efficient computation of the Hamming distance to further reduce the latency and complexity. This BWA uses a modified half adder (HA) circuit and an XOR design to further reduce area and latency. The implementation result also shows drastic reduction in area and latency using this proposed method.

Improved Architecture for Direct Comparison of Data Encoded With Hard Systematic Error Correcting Codes

In current scenario, there are situations in a computing system where incoming information needs to be compared with a piece of stored data to locate the matching entry, e.g., cache tag array lookup and translation look-aside buffermatching .In this paper, a new architecture to reduce complexity and latency for matching the data protected with an error-correcting code (ECC).It is based on the fact that the codeword of an ECC generated by encoding is usually represented in a systematic form, and it consists of the raw data and the parity information. The proposed architecture parallelizes the comparison of the data and that of the parity information. To further reduce the latency and complexity, in addition, a modified butterfly formed weight accumulator (BWA) is proposed for the efficient computation of the Hamming distance. The proposed architecture examines whether the incoming data matches the stored data if a certain number of erroneous bits are correct.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Review on different error correction codes

Data transfer between two or multiple nodes is a common but critical operation in many applications of communication network. When the data transfer through the communication network some error bit and noise is added with data. To obtain the original data at the receiver side Error Correction Codes are used. Communication performance is improved by enabling the transmitted signal to better withstand the effects of channel disabilities such as noise, interference and fading which occur during transmission as well as during comparison. In short the Error Correction is the detection of errors and reconstruction of the original, error free data. There are errors correcting codes to detect as well as correct errors. In this paper the detail review of error correction code techniques is done. The main aim of this paper is to study the different error correcting codes techniques available for the error correction during the transformation of the data from the communication channel in details

A Novel Architecture for Matching the Data Protected With an Error-Correcting Code (ECC)

In this paper we presented a novel architecture for matching the data protected with an Error-Correcting Code (ECC) which proposed to reduce latency and complexity where Data comparison is widely used in computing system to perform so many operations. Where incoming information is needs to be compared with a piece of stored data to locate the matching entry. If both incoming bits and stored bits are matching m eans there is no error if mismatched means some type of error will occur like random error or burst error. To detect and correct the error here error correcting codes are used. To further reduce the latency and complexity, in addition, a new butterfly-formed weight accumulator (BWA) is proposed for the efficient computation of the Hamming distance and also demonstrates LDPC coding and decoding for Error Correcting Codes further more examines whether the incoming data matches the stored data if a certain number of burst errors are corrected.

III. PROPOSED ALGORITHM

A. Datapath Design for Systematic Codes

In the SA-based architecture [6], the comparison of two codewords is invoked after the incoming tag is encoded. Therefore, the critical path consists of a series of the encoding and then-bit comparison as shown in Fig. 3(a). However, [6] did not consider the fact that, in practice, the ECC codeword is of a systematic form in which the data and parity parts are completely separated as shown in Fig. 4. As the data part of a systematic codeword is exactly the same as the incoming tag field, it is immediately available for comparison while theparity part becomes available only after the encoding is completed. Grounded on this fact, the comparison of thek-bit tags can be started before the remaining (n–k)-bit comparison of the parity bits. In the proposed architecture, therefore, the encoding process to generate the parity bits from the incoming tag is performed in parallel with the tag comparison, reducing the overall latency as shown in Fig. 3(b).

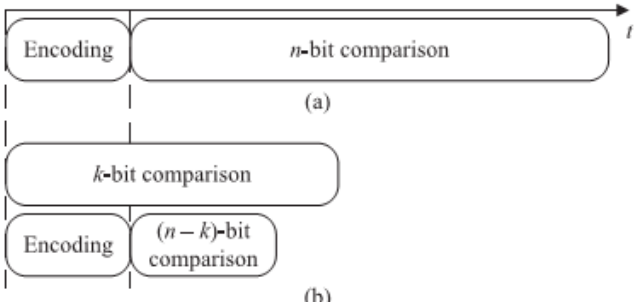

Fig. 3. Timing diagram of the tag match in (a) direct compare method [6] and (b) proposed architecture.

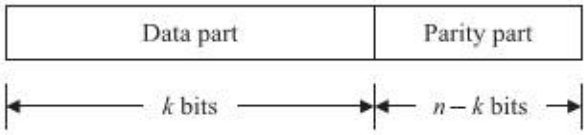

Fig. 4. Systematic representation of an ECC codeword.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

B. Architecture for Computing the Hamming Distance

The proposed architecture grounded on the data path design is shown in Fig. 5. It contains multiple butterfly-formed weight accumulators (BWAs) proposed to improve the latency and complexity of the Hamming distance computation. The basic function of the BWA is to count the number of 1's among its input bits. It consists of multiple stages of HAs as shown in Fig. 6(a), where each output bit of a HA is associated with a weight. The HAs in a stage are connected in a butterfly form so as to accumulate the carry bits and the sum bits of the upper stage separately. In other words, both inputs of a HA in a stage, except the first stage, are either carry bits or sum bits computed in the upper stage. This connection method leads to a property that if an output bit of a HA is set, the number of 1's among the bits in the paths reaching the HA is equal to the weight of the output bit. In Fig. 6(a), for example, if the carry bit of the gray-colored HA is set, the number of 1's among the associated input bits, i.e., A, B, C, and D, is 2. At the last stage of Fig. 6(a), the number of 1's among the input bits, d, can be calculated as

$$d = 8\mathrm{I} + 4(\mathrm{J} + \mathrm{K} + \mathrm{M}) + 2(\mathrm{L} + \mathrm{N} + \mathrm{O}) + \mathrm{P}. \qquad (2)$$

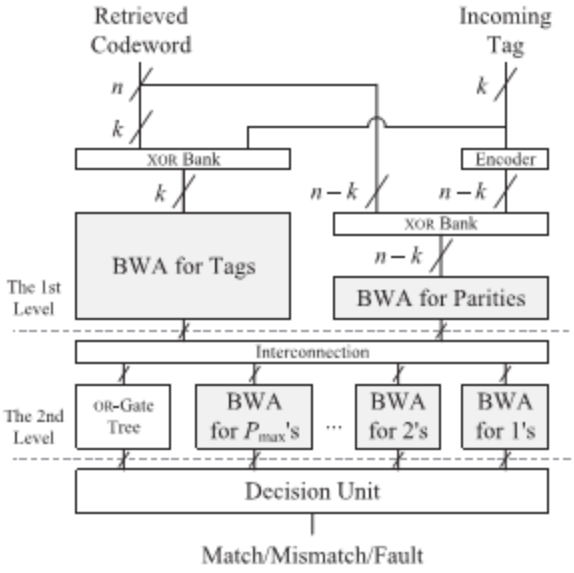

Fig. 5. Proposed architecture optimized for systematic codewords.

Note that sum-bit lines are dotted for visibility.

Since what we need is not the precise Hamming distance but the range it belongs to, it is possible to simplify the circuit. When rmax=1, for example, two or more than two 1's among the input bits can be regarded as the same case that falls in the fourth range. In that case, we can replace several HAs with a simple OR-gate tree as shown in Fig. 6(b). This is an advantage over the SA that resorts to the compulsory saturation expressed in (1)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

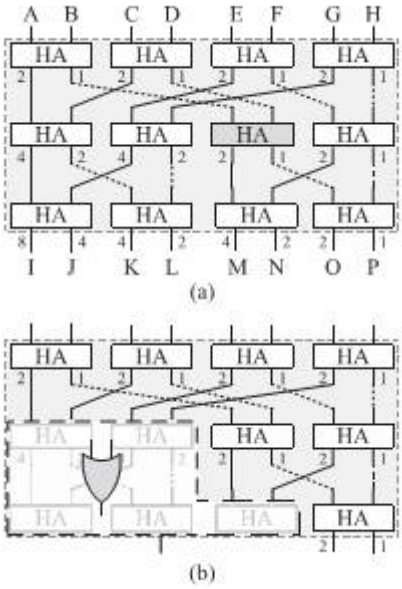

Fig. 6. Proposed BWA. (a) General structure and (b) new structure revised for the matching of ECC-protected data..
Note that in Fig. 6, there is no overlap between any pair of two carry-bit lines or any pair of two sum-bit lines. As the overlaps exist only between carry-bit lines and sum-bit lines, it is not hard to resolve overlaps in the contemporary technology that provides multiple routing layers no matter how many bits a BWA takes. We now explain the overall architecture in more detail. Each XOR stage in Fig. 5 generates the bitwise difference vector for either data bits or parity bits, and the following processing elements count the number of 1's in the vector, i.e., the Hamming distance. Each BWA at the first level is in the revised form shown in Fig. 6(b), and generates an output from theOR-gate tree and several weight bits from the HA trees. In the interconnection, such outputs are fed into their associated processing elements at the second level. The output of the OR-gate tree is connected to the subsequent OR-gate tree at the second level, and the remaining weight bits are connected to the second level BWAs according to their weights. More precisely, the bits of weight ware connected to the BWA responsible forw-weight inputs. Each BWA at the second level is associated with a weight of a power of two that is less than or equal to Pmax, where Pmaxis the largest power of two that is not greater than rmax+1. As the weight bits associated with the fourth range are all ORed in the revised BWAs, there is no need to deal with the powers of two that are larger than Pmax.

For example, let us consider a simple (8, 4) single-error correction double-error detection code. The corresponding first and second level circuits are shown in Fig. 7. Note that the encoder and XOR banks are not drawn in Fig. 7 for the sake of simplicity. Since rmax=2, Pmax=2 and there are only two BWAs dealing with weights 2 and 1 at the second level. As the bits of weight 4 fall in the fourth range, they are ORed. The remaining bits associated with weight 2 or 1 are connected to their corresponding BWAs. Note that the interconnection induces no hardware complexity, since it can be achieved by a bunch of hard wiring.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

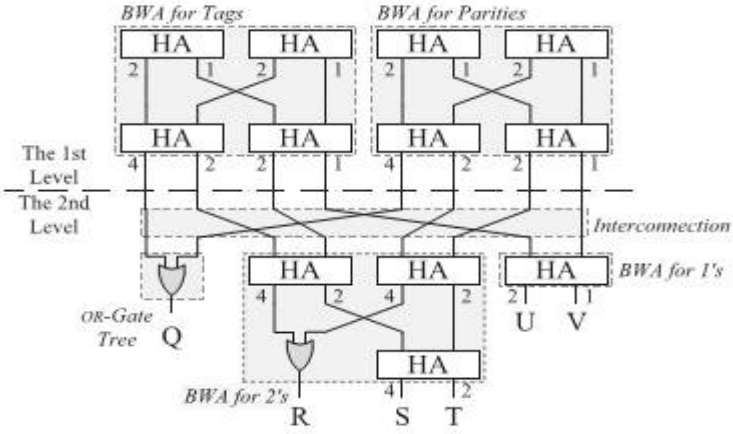

Fig. 7. First and second level circuits for a (8, 4) code.

Taking the outputs of the preceding circuits, the decision unit finally determines if the incoming tag matches the retrieved codeword by considering the four ranges of the Hamming distance. The decision unit is in fact a combinational logic of which functionality is specified by a truth table that takes the outputs of the preceding circuits as inputs.

IV. CONCLUSION AND FUTURE WORK

To reduce the latency and hardware complexity, a new architecture has been presented for matching the data protected with an ECC. The proposed architecture examines whether the incoming data matches the stored data if a certain number of erroneous bits are corrected. To reduce the latency, the comparison of the data is parallelized with the encoding process that generates the parity information. The parallel operations are enabled based on the fact that the systematic codeword has separate fields for the data and parity. In addition, an efficient processing architecture has been presented to further minimize the latency and complexity.

REFERENCES

1. Byeong Yong Kong, Jihyuck Jo, HyewonJeong, Mina Hwang , Soyoung Cha, Bongjin Kim, and In- Cheol Park "Low-Complexity Low-Latency Architecture for Matching Of Data Encoded With Hard Systematic Error-Correcting Codes" IEEE transactions on (vlsi) systems, vol. 22, July 2014
2. J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen, S. Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel xeon processor 7100 series," IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 846–852, Apr. 2007.
3. J. D. Warnock, Y.-H. Chan, S. M. Carey, H. Wen, P. J. Meaney, G. Gerwig, H. H. Smith, Y. H. Chan, J. Davis, P. Bunce, A. Pelella, D. Rodko, P. Patel, T. Strach, D. Malone, F. Malgioglio, J. Neves, D. L. Rude, and W. V. Huott "Circuit and physical design implementation of themicroprocessor chip for the zEnterprisesystem," IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 151–163, Jan. 2012.
4. H. Ando, Y. Yoshida, A. Inoue, I. Sugiyama, T. Asakawa, K. Morita, T. Muta, and T. Motokurumada, S. Okada, H. Yamashita, and Y. Satsukawa, "A 1.3 GHz fifth generation SPARC64 microprocessor," in IEEE ISSCC. Dig. Tech. Papers, Feb. 2003, pp. 246–247.
5. M. Tremblay and S Chaudhry, " A third-generation 65nm 16-core 32-thread plus 32-scout-thread CMT SPARC processor," in ISSCC. Dig. Tech. Papers, Feb. 2008, pp. 82–83.
6. AMD Inc. (2010). Famil 10h AMD Opteron Processor Product Data Sheet, Sunnyvale, CA, USA [Online]. Available: http://support.amd.com/us/Processor_TechDocs/40036.pdf
7. W. Wu, D. Somasekhar, and S.-L. Lu, "Direct compare of information coded with error-correcting codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 11, pp. 2147–2151, Nov. 2012.
8. S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
9. Y. Lee, H. Yoo, and I.-C. Park, "6.4Gb/s multi-threaded BCH encoder and decoder for multi-channel SSD controllers," in ISSCC Dig. Tech. Papers, 2012, pp. 426–427

BIOGRAPHY

Priyanka Ashokrao Wathodkar is a PG student, department of electronics and telecommunication , government college of engineering Amravati. She received bachelor degree from Santgadge baba Amravati University, MS, INDIA