

A Method to Design Single Error Correction Codes With Subset of Critical Bits

Vinjamuri Shiva

Btech: Ramanandatirtha Engineering College
Mtech: Vasthsalya Institute of Science and Technology (VLSI)

P. Mahender (Guide)

Btech: JJ Institute of Information Technology
Mtech: Vasthsalya Institute of Science and Technology (VLSI)

Abstract—Single error correction (SEC) codes are widely used to protect data stored in memories and registers. In some applications, such as networking, a few control bits are added to the data to facilitate their processing. For example, flags to mark the start or the end of a packet are widely used. Therefore, it is important to have SEC codes that protect both the data and the associated control bits. It is attractive for these codes to provide fast decoding of the control bits, as these are used to determine the processing of the data and are commonly on the critical timing path. In this brief, a method to extend SEC codes to support a few additional control bits is presented. The derived codes support fast decoding of the additional control bits and are therefore suitable for networking applications.

Index Terms—Error correction codes, high-speed networking, memory, single error correction (SEC).

I. INTRODUCTION

NETWORKING applications require high-speed processing of data and thus rely on complex integrated circuits

[1]. In routers and switches, packets typically enter the device through one port, are processed, and are then sent to one or more output ports. During this processing, data are stored and moved through the device [2].

Reliability is a key requirement for networking equipment such as core routers [3]. Therefore, the stored data must be protected to detect and correct errors. This is commonly done using error-correcting codes (ECCs) [4]. For memories and registers, single error correction (SEC) codes that can correct 1-bit errors are commonly used [5], [6].

One problem that occurs when protecting the data in networking applications is that, to facilitate its processing, a few control bits are added to each data block. For example, flags to mark the start of a packet (SOP), the end of a packet (EOP), or an error (ERR) are commonly used [7]. These flags are used to determine the processing of the data, and the associated control logic is commonly on the critical timing path. To access the control bits, if they are protected with an ECC, they must first be decoded. This decoding adds delay and may limit the overall

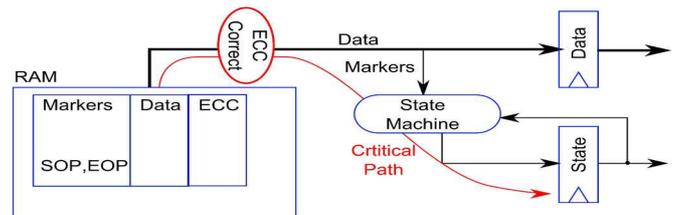


Fig. 1. Typical packet data storage in a networking application.

Then, a SEC code can protect a data block using 8 parity check bits, and another SEC code can protect the 3 control bits using 3 parity check bits. This option provides independent decoding of data and control bits which reduces the delay but requires additional parity check bits. Another option is to use a single ECC to protect both the data and control bits. Protecting $128 + 3$ bits requires only 8 parity check bits, thus saving 3 bits compared to the use of separate ECCs. However, in this case, the decoding of the control bits is more complex and incurs more delay.

In this brief, a method to extend a SEC code to also protect a few additional control bits is proposed. In the resulting codes, the control bits can be decoded using a subset of the parity check bits. This reduces the decoding delay and makes them suitable for networking applications. To evaluate the method, several codes have been constructed and implemented. They are then compared with existing solutions in terms of decoding delay and area.

The rest of this brief is organized as follows. In Section II, the problem of control bit decoding in networking applications is described. In Section III, the proposed method to construct the codes to support fast decoding of the control bits is presented. The proposed scheme is evaluated for some relevant examples in Section IV. Finally, the conclusion and some ideas for future work are presented in Section V.

II. DATA PROTECTION IN NETWORKING APPLICATIONS

Modern networking equipment supports data rates that range from 10 to 400 Gbit/s, and terabit rates are expected in the near future [8]. The clock frequencies used in current ASICs are typically in the range of 300 MHz to 1 GHz, and the clock frequencies in FPGAs are typically lower (under 400 MHz). To support these high data rates, on-chip packet data buses are wide, with typical widths between 64 and 2048 bits [9], [10].

TABLE IV
ASIC CIRCUIT AREA (μm^2) FOR 7 ADDITIONAL CONTROL BITS

	Minimum weight SEC code	Proposed SEC code
Encoder (64+7 all bits)	264.7	266.0
Decoder (64+7 all bits)	607.5	581.2
Encoder (128+7 all bits)	501.7	488.7
Decoder (128+7 all bits)	1081.3	1084.5
Encoder (256+7 all bits)	956.0	937.9
Decoder (256+7 all bits)	1892.9	1947.1

TABLE V
ASIC CIRCUIT DELAY (NS) FOR 7 ADDITIONAL CONTROL BITS

	Minimum weight SEC code	Proposed SEC code
Encoder (64+7, all bits)	0.54	0.54
Decoder (64+7, control bits)	0.67	0.60
Decoder (64+7, data bits)	0.72	0.80
Encoder (128+7, all bits)	0.67	0.67
Decoder (128+7, control bits)	0.81	0.72
Decoder (128+7, data bits)	0.89	1.02
Encoder (256+7, all bits)	0.86	0.87
Decoder (256+7, control bits)	0.92	0.83
Decoder (256+7, data bits)	0.99	1.34

control bits as that is the main design goal. For the encoders, the tool was configured to minimize delay on all bits. In all cases, identical synthesis constraints were applied to both the proposed codes and the minimum-weight codes. The circuit area and delay have been evaluated.

The results for the case of three additional control bits are shown in Tables II and III. The tables also show the results for the minimum-weight SEC codes. In this case, the reduction of the decoding delay of the control bits is in the range of 12%–18%. This shows the potential of the proposed scheme to reduce the critical path. The circuit area is similar to that of the minimum-weight SEC codes, in some cases slightly lower and in some slightly higher.

The proposed codes do have an impact on the decoding delay for the data bits. For the decoders, the added delay on data bits is significant for most word sizes. However, as discussed in the introduction, the major design goal is to reduce the decoding delay of the control bits as these typically determine the critical timing path.

The results for the case of seven control bits are shown in Tables IV and V. The proposed codes require a circuit area for both the encoder and the decoder similar to that of the minimum-weight codes. In terms of delay, decoding of the data bits is slower. On the other hand, the proposed codes are able to reduce the decoding delay of the control bits by approximately 9%–11%. This reduction is smaller than that for the three control bits case. This is expected as the number of parity bits (p_{cd}) used to decode the control bit increases (from three to four) and so does the decoder complexity. Therefore, the benefits of the proposed scheme decrease as the number of control bits increases.

In summary, the proposed method can be used to reduce the decoding delay of the control bits, especially when the number of control bits is small.

V. CONCLUSION AND FUTURE WORK

In this brief, a method to construct SEC codes that can protect a block of data and some additional control bits has been presented. The derived codes are designed to enable fast decoding of the control bits. The derived codes have the same number of parity check bits as existing SEC codes and therefore do not require additional cost in terms of memory or registers. To evaluate the benefits of the proposed scheme, several codes have been implemented and compared with minimum-weight SEC codes.

The proposed codes are useful in applications, where a few control bits are added to each data block and the control bits have to be decoded with low delay. This is the case on some networking circuits. The scheme can also be useful in other applications where the critical delay affects some specific bits such as in some finite-state machines. Another example is arithmetic circuits where the critical path is commonly on the least significant bits. Therefore, reducing the delay on those bits can increase the overall circuit speed. The use of the proposed scheme for those applications beyond networking is an interesting topic for future work. It may be possible to apply the idea of modifying the matrix of the code to enable fast decoding of a few bits to more advanced ECCs that can correct multiple bit errors. Finally, the scheme can also be extended to support more control bits by using one or two additional parity check bits. This would provide a solution to achieve fast decoding without using two separate codes for data and control bits.

REFERENCES

- [1] P. Bosshart *et al.*, "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN," in *Proc. SIGCOMM*, 2013, pp. 99–110.
- [2] J. W. Lockwood *et al.*, "NetFPGA—An open platform for gigabit-rate network switching and routing," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ.*, Jun. 2007, pp. 160–161.
- [3] A. L. Silburt, A. Evans, I. Perryman, S.-J. Wen, and D. Alexandrescu, "Design for soft error resiliency in Internet core routers," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 6, pp. 3551–3555, Dec. 2009.
- [4] E. Fujiwara, *Code Design for Dependable Systems: Theory and Practical Application*. Hoboken, NJ, USA: Wiley, 2006.
- [5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 124–134, Mar. 1984.
- [6] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalized parity-check matrices for SEC-DED codes with fixed parity," in *Proc. IEEE On-Line Test. Symp.*, 2011, pp. 198–20.
- [7] Ten Gigabit Ethernet Medium Access Controller, OpenCores. [Online]. Available: <http://opencores.org/project/ethmac>
- [8] P. Zabinski, B. Gilbert, and E. Daniel, "Coming challenges with terabit-per-second data communication," *IEEE Circuits Syst. Mag.*, vol. 13, no. 3, pp. 10–20, 3rd Quart. 2013.
- [9] UltraScale Architecture Integrated Block for 100 G Ethernet v.14. LigCOREIP Product Guide. PG165, Xilinx, San Jose, CA, USA. Jan. 22, 2015.
- [10] OpenSilicon Interlaken ASIC IP Core. [Online]. Available: www.opensilicon.com/open-silicon-ips/interlaken-controller-ip/
- [11] P. Reviriego, S. Pontarelli, J. A. Maestro, and M. Ottavi, "A method to construct low delay single error correction (SEC) codes for protecting data bits only," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 3, pp. 479–483, Mar. 2013.
- [12] J. E. Stine *et al.*, "FreePDK: An open-source variation-aware design kit," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ.*, Jun. 2007, pp. 173–174.