International Journal of Research Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 01
January 2018

A Method to Design Single Error Correction Codes for Fast Decoding

SHAIK.SALMA[1] P.NAGARAJU[2]

M-Tech, Dept of ECE,kakinada Institute of Engineering and technology, korangi.

Assoc Prof,Dept of ECE,kakinada Institute of Engineering and technology, korangi.

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 theassociated 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.

Keywords: - Error correction codes, reed-solomon codes, DRAM, soft errors.

1. INTRODUCTION

Networking applications require high-speed processing of data and thus rely on complex integrated circuits. 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. Reliability is a key requirement for networking equipment such as core routers. Therefore, the stored data must be protected to detect and correct errors. This is commonly done using error-correcting codes (ECCs). For memories and registers, single error correction (SEC) codes that can correct 1-bit errors are commonly used. 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. 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 frequency. One option is to protect the data and the control bits as different data blocks using

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 01
January 2018

separate ECCs. For example, let us assume 128-bit data blocks with 3 control bits. 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.

Fig 1: Parity check matrix for a minimum-weight SEC code that protects 128 data bits.

Packet data must frequently be stored in RAMs, e.g., in FIFOs for adapting processing rates. When storing packet data, it is necessary to delineate the packet boundaries. In the absolute simplest case, each segment on the bus can be delineated with a single EOP marker. The next valid segment is then assumed to be the start of the following packet. In practice, designers also use a SOP marker to explicitly mark the start of packets. There are also many cases in packet processing where a packet is in error and it must be dropped. To mark such error packets, an additional control signal (ERR) may be required.

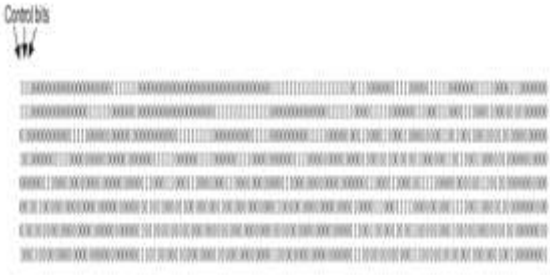

Fig 2: Parity check matrix for a minimum-weight SEC code that protects 128 data and 3 control bits.

As mentioned in the introduction, from an error protection perspective, it is attractive to store the data and the markers in a single wide memory, as shown in Fig. 1. In this way, relatively fewer ECC bits are required. The problem with this approach is when the data are read out. Typically, the markers feed into a state machine that controls the reading of the subsequent data. For example, the state machine may need to read out a single packet (up to an EOP), or it may need to read out a fixed number of bytes of data (e.g., deficit round robin scheduler). The critical timing path then consists of the ECC correction logic, followed by the state machine logic, as shown in red. With a traditional Hamming SEC code, as the data bus increases in width, the number of layers of logic required to decode the syndrome and perform correction also increases. Circuit designers frequently observe critical timing on the signal paths related to the correction of the markers which feed downstream state machines. For this reason, special ECC codes which can provide a fast decode of the small number of marker bits are extremely attractive.

2. SIMULATION IMPLEMENTATION

Simulation:

For starting simulation of a design, the software or the tool need to be kept in simulation mode. In order to initiate a simulation, go to Simulate > Start simulation. This will open a new simulation window as shown in below figure.

The window of start simulation consists of many tabs including a list of design tabs that list the available designs for simulation. VHDL and Verilog tabs to specify language

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 01
January 2018

specific options. A library tab to include any additional libraries. Timing and other options in the remaining two tabs. We only need to look on the design tab for the purpose of functional simulation.

VERILOG

Equipment portrayal dialects, for example, Verilog contrast from programming dialects on the grounds that they incorporate methods for depicting the proliferation of time and flag conditions (affectability). There are two task administrators, a blocking task (=), and a non-blocking (<=) task. The non-blocking task permits planners to portray a state-machine upgrade without expecting to pronounce and utilize transitory capacity variables (in any broad programming dialect we have to characterize some provisional storage rooms for the operands to be worked on along these lines; those are impermanent capacity variables). Since these ideas are a piece of Verilog's dialect semantics, architects could rapidly compose portrayals of expansive circuits in a generally minimal and succinct structure. At the season of Verilog's presentation (1984), Verilog spoke to a huge efficiency change for circuit originators who were at that point utilizing graphical schematic capture software and uniquely composed programming projects to report and mimic electronic circuits. The originators of Verilog needed a dialect with grammar like the C programming dialect, which was at that point generally utilized as a part of designing programming advancement. Verilog is case-delicate, has a fundamental preprocessor (however less advanced than that of ANSI C/C++), and proportional control stream watchwords (if/else, for, while, case, and so on.), and perfect administrator priority. Syntactic contrasts incorporate variable affirmation (Verilog requires bit-widths on net/regtypes[clarification needed]), boundary of procedural squares (start/end rather than wavy props {}), and numerous other minor contrasts.

3. SIMULATION RESULTS

Fig:-3 Schematic output

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 01
January 2018

Fig:-4 RTL Schematic

4. CONCLUSION

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.

5. 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 terabitper-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/

Authors Profile

I salma was born in kanigiri ,prakasam district ,andhrapradesh on june 15, 1992 . I graduated from the Kakinada Institute Of Engineering And Technology Enginering college (JNTU) kakinada . Presently I am studying M.Tech inkakinada Institute of Engineering and technology, korangi.

Mr. P.NAGARAJU was born DRAKSHRAMAM, AP, on MAY 01 1982.

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 01
January 2018

He graduated from the Jawaharlal Nehru Technological University, Hyderabad, Post-graduated from the Jawaharlal Nehru Technological University, Kakinada, and Pursuing Ph.D. from JNTUK. Presently He is working as an Asst Prof in Kakinada Institute of Engineering & Technology, Korangi. So far he is having 12 Years of Teaching Experience in various reputed engineering colleges. His special fields of interest included VLSI-Signal Processing, Embedded Systems, Digital Signal Processing & communication Systems.