Efficient Coding Schemes for Fault-Tolerant Parallel Filters: A Survey

Kavita Goswami[1], Prof. Kamna Mishra[2]

[1]Mtech Scholar, [2]Research Guide

Department of Electronics and Communication, LNCT, Bhopal

Abstract - As the complexity of communications and signal processing systems increases, so does the number of blocks or elements that they have. In many cases, some of those elements operate in parallel, performing the same processing on different signals. A typical example of those elements is digital filters. The increase in complexity also poses reliability challenges and creates the need for fault-tolerant implementations. A review on scheme based on error correction coding has been presented to protect parallel filters. In that scheme, each filter is treated as a bit, and redundant filters that act as parity check bits are introduced to detect and correct errors. In this brief, the idea of applying coding techniques to protect parallel filters is addressed in a more general way. In particular, it is shown that the fact that filter inputs and outputs are not bits but numbers enables a more efficient protection. This reduces the protection overhead and makes the number of redundant filters independent of the number of parallel filters. The present literature review is described. Finally, both the effectiveness in protecting against errors and the cost are evaluated for a field-programmable gate array implementation in prior work.

Index Terms—coding, parallel filters, soft errors.

I. INTRODUCTION

Traditionally, the problem of computational fault-tolerance has been solved through modular redundancy. In this technique, several identical copies of the system operate in parallel using the same data, and their outputs are compared with voter circuitry. If no errors have occurred, all outputs will agree exactly. Otherwise, if an error has occurred, the faulty module can be easily identified and the correct output determined. Modular redundancy is a general technique and can be applied to any computational task. Unfortunately, it does not take advantage of the structure of a problem and requires a large amount of hardware overhead relative to the protection afforded.

A more efficient method of protecting computation is to use an arithmetic code and tailor the redundancy to the specific operation being performed. Arithmetic codes are essentially error-correcting codes whose error detecting and correcting properties are preserved during computation. Arithmetic codes offer performance and redundancy advantages similar to existing error-correcting codes used to protect communication channels. Unfortunately, arithmetic codes exist for only a limited number of operations.

This review addresses the general problem of designing an arithmetic code to protect a given computation. A practical arithmetic code must satisfy three requirements:

- It must contain useful redundancy.
- It must be easily encoded and decoded.
- Its inherent redundancy must be preserved throughout computation.

The first two requirements are shared by error-correcting codes for protecting data transmission, while the third requirement is unique to arithmetic codes. This host of requirements makes designing practical arithmetic codes an extremely difficult problem.

A. Fault Tolerance

Fault tolerance is the property that enables a system to continue with its correct operation even in the presence of faults (errors), and it is generally implemented by error detection and subsequent system recovery. Fault tolerance has been a subject of research for a long time, and significant amount of work has been produced over the years to provide fault tolerance, systems are usually designed such that some redundancy is included. The common types of redundancy used are information, hardware, and time redundancy.

Error-detecting and error-correcting codes provide fault tolerance while using information redundancy, i.e. the data includes additional information (check bits) that can verify the correctness of the data before it is used (error-detection), or even correct erroneous data bits (error-correction). Different error-detecting and error-correcting codes have been proposed including parity codes, cyclic codes, arithmetic codes etc. . The major disadvantage of error-detecting and error-correcting codes is that they are limited to errors that occur during transfer of data (system bus) or errors in memory.

B. Fault Tolerance in Real-Time Systems

Since RTSs, like any other computer system manufactured in the latest semiconductor technologies, are susceptible to soft errors, it is important to employ fault tolerance in RTSs as well. However, fault tolerance usually introduces a time overhead which negatively impacts RTSs. The time overhead due to usage of fault tolerance may increase the AET, and it may lead to a missed deadline. Therefore, special consideration should be taken when employing fault tolerance in RTSs. when errors occur with some probability, there is a probability the deadlines might be missed. Therefore, when optimizing the usage of fault tolerance in RTSs, another important aspect to consider is optimization of fault tolerance such that the probability to meet the deadlines is maximized, which is needed for hard RTSs, or at least the probability to meet the deadlines is higher than a given reliability requirement, which may be needed for soft RTSs.

II. SYSTEM MODEL

A. Architecture of error detection

The architecture shown in Figure 2.1 consists of two processing nodes (processors), a shared memory and a Compare & Control Unit (CCU) connected through a shared bus. In such architecture RRC is performed as follows. Each job is duplicated and concurrently executed on both processing nodes. At a given time (a checkpoint request), the execution of the job is interrupted and a checkpoint is taken at each node. The checkpoint includes sufficient information such that the job can be resumed from that particular point. We consider a checkpoint to be represented as the state (status) of a processing node. Once the states of both nodes are obtained, each processing node sends its state to the CCU. The CCU compares the states from both processing nodes. If the states match, i.e. no errors are detected; the CCU stores one of the states in memory and signals to the processing nodes to continue with the execution of the job. If the states do not match, i.e. an error is detected; the CCU loads the most recently saved state from memory and sends it to both processing nodes forcing them to roll-back the execution of the job.

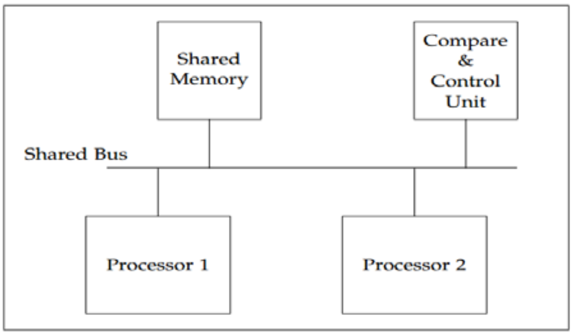

Figure 2.1: Error detection system model.

B. Fault Model and Fault Assumptions

For the fault model, we consider that soft errors (faults) that occur in the processing nodes cause erroneous outcome of the undergoing computation, i.e. bit-flips in the result produced after some computation. The fault model considers the occurrence of soft errors as an independent event. This means that the occurrence of a soft error does not depend on previous soft errors that have occurred. Further, the fault model considers that the probability Pt that no errors occur in a processing node within an interval of length T is given. This model is not limited to the number of faults that can occur within a time interval, which is an assumption that has been used in other research studies.

While soft errors can occur in any part of a computer system, i.e. memories, communication controllers, buses, etc , we address soft errors that occur only in the processing nodes, and we assume that errors occurring elsewhere in the system are handled with conventional techniques for fault tolerance, e.g. error-correction codes (ECC) for handling soft errors that occur in memory. Further, we assume that each soft error provides a unique erroneous outcome. By using this assumption, if two soft errors occur, one in each processing node, the states of both processing nodes will differ due to that each soft error has caused a different erroneous outcome. Figure 2.2 shows the model of fault tolerance parallel filter.

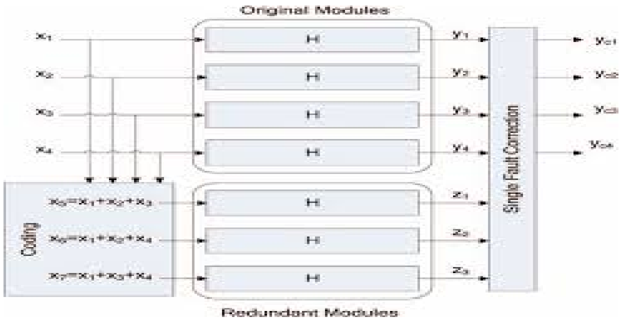

Figure 2.2 fault tolerance in parallel filter.

III. RELATED WORK

Z. Gao, P. Reviriego, Z. Xu, X. Su, J. Wang and J. A. Maestro,[1] As the complexity of communications and signal processing systems increases, so does the number of blocks or elements that they have. In many cases, some of those elements operate in parallel, performing the same processing on different signals. A typical example of those elements is digital filters. The increase in complexity also poses reliability challenges and creates the need for fault-tolerant implementations. A scheme based on error correction coding has been recently proposed to protect parallel filters. In that scheme, each filter is treated as a bit, and redundant filters that act as parity check bits are introduced to detect and correct errors. In this brief, the idea of applying coding techniques to protect parallel filters is addressed in a more general way. In particular, it is shown that the fact that filter inputs and outputs are not bits but numbers enables a more efficient protection. This reduces the protection overhead and makes the number of redundant filters independent of the number of parallel filters. The proposed scheme is first described and then illustrated with two case studies. Finally, both the effectiveness in protecting against errors and the cost are evaluated for a field-programmable gate array implementation.

Z. Gao et al. Digital filters are widely used in signal processing and communication systems. In some cases, the reliability of those systems is critical, and fault tolerant filter implementations are needed. Over the years, many techniques that exploit the filters' structure and properties to achieve fault tolerance have been proposed. As technology scales, it enables more complex systems that incorporate many filters. In those complex systems, it is common that some of the filters operate in parallel, for example, by applying the same filter to different input signals. Recently, a simple technique that exploits the presence of parallel filters to achieve fault tolerance has been presented. In this brief, that idea is generalized to show that parallel filters can be protected using error correction codes (ECCs) in which each filter is the equivalent of a bit in a traditional ECC. This new scheme allows more efficient protection when the number of parallel filters is large. The technique is evaluated using a case study of parallel finite impulse response filters showing the effectiveness in terms of protection and implementation cost.

K. B. Muralidharan, G. S. Kumar and M. Bhasi,[3] One of the major challenges in performing incremental computations on parallel distributed stream processing systems is in the implementation of a mechanism for passing state values across successive runs. One approach is to enhance the granularity from record-at-a-time processing to processing at micro-batch level. A contrasting approach is to follow the record-at-a-time semantics and ensure scalability by means of distributed state management. Both approaches, however, require observing high degree of fault tolerance. In this work, we study the problem of process state management against non-terminating data stream workloads for low-latency computing using the micro-batch stream processing approach. We attempt to examine methods that could yield optimum levels of state retentions with high degree of fault tolerance for typical processing workloads and propose a three-pronged approach to harness the demand.

N. Madhuri, S. R. Doradla and M. Surya kalavathi,[4] Hybrid Filter configuration consists of an active and a passive filter connected in parallel. It is employed to suppress source current harmonics, compensates reactive power and moreover to improve the power quality. Fault tolerant method is the ability to maintain system performance and stable environment in the event of component failures. Recently, fault tolerant shunt active power filter was proposed using redundancy method to ensure reliability of active power filters. However in this work, we propose a fault tolerant hybrid filter configuration to offer the reliability to guarantee dependability. The proposed work is compared with the fault tolerant shunt active power filter. Simulation results are obtained using MATLAB/SIMULINK for fault tolerant hybrid filter and demonstrate superior results in comparison with fault tolerant shunt active filter.

J. Park, J. Park and S. Bhunia,[5] Increasing process variations coupled with aggressive scaling of cell area and operating voltage in the quest of higher density and lower power have greatly affected the reliability of on-chip memory. Error correction code (ECC) has been traditionally used inside memory to provide uniform protection to all bits in a code word. They suffer from either adequate protection against multibit failures or large overhead due to encoding/decoding logic and parity bits. To address this issue, we present a variable data-length ECC (VL-ECC) for the embedded memory devices of digital signal processors, in which the data length of ECC can be dynamically reconfigured to preferentially protect the relatively more important bits. In the proposed VL-ECC, when the number of failures exceeds the error correction capability, the data length of ECC is reduced to focus on the relatively more important higher order data bit parts, thereby minimizing system quality degradation due to bit failures. When the proposed VL-ECC is applied to the embedded memory devices of an H.264 processor, average peak signal-to-noise-ratio improvements of up to 5.12 dB are achieved compared with the conventional ECC under supply voltage of 800 mV or lower. With the fast Fourier transform processor, signal-to-quantization noise ratio is improved by up to 5.2 dB.

A. Asuvaran and S. Senthilkumar,[6] For perfect communication during data transmission between transmitter and receiver the reliability is important factor, sometimes reliability is missed due to appearance of errors. ECC provides reliable data delivery over unreliable communication channel. Error correction codes (ECCs) are used to shield memories from soft errors and stuck-at defects. Single error correction (SEC) codes that can correct 1-bit error per word are a typical choice for memory protection. In some cases, SEC codes are extended to conjointly offer double error detection and are referred to as SEC-DED codes. Repair proficiencies are usually used for defects, whereas error correction codes are used for soft errors. Recently, some proposals are created to use error correction codes to handle with defects. In this work tend to correct two single bit errors within the combination of one soft error and one stuck-at defect and to reduce delay of Error Correction Codes.

P. Reviriego, S. Pontarelli, C. J. Bleakley and J. A. Maestro,[7] In modern signal processing circuits, it is common to find several filters operating in parallel. Proposed is an area efficient technique to detect and correct single errors occurring in pairs of parallel filters that have either the same input data or the same impulse response. The technique uses a primary implementation comprised of two independent filters and a redundant implementation that shares input data between both filters so as to detect and correct errors.The area cost of the proposed scheme is shown to be slightly more than double that of the unprotected filter, whereas the conventional triple modular redundancy solution requires an area three times that of the unprotected filter.

IV. PROBLEM STATEMENT

In the previous work the efficient coding schemes for fault - tolerant parallel filters are the filters that has been used in the filter bank for the communication channel in the protection status. The data’s communicated through this channel length have been arrived from the associated architecture In many cases, the filters perform the same processing on different incoming signals as there is a tendency to use multiple-input-multiple- output systems .This parallel operation can be exploited for fault tolerance. In fact, reliability is a major challenge for electronic systems. In particular, soft errors are an important issue, and many techniques have been proposed over the years to mitigate them. Some of these techniques modify the low-level design and implementation of the integrated circuits to prevent soft errors from occurring. Other techniques work at a higher abstraction level by adding redundancy that can detect and correct errors. The proposed method some error correcting code can be designed and can be expended with new low bit or higher bit architecture and concluded with parameters like area, delay and power can be determined.

V. CONCLUSION

Technology scaling has enabled us to keep pace with the power, performance, area and functionality requirements of electronic circuits. Along with the advantages, it has also given challenges due to increased leakage current, reliability failures, etc. Reliability failures include systematic failures due to the aging effects of silicon structures caused by Negative Bias Temperature Instability (NBTI) and Hot Carrier Injection (HCI) and random failures due to atmospheric particle strikes, called as soft errors. The contribution of particle strike induced failures to the overall device fail rate is more than ten times than that due to a hard reliability fail. This highlights the importance of the requirement of soft error mitigation in safety critical systems. in this present review different methods for error detection and correction and various work done on the field of parallel filter has been reviewed.

REFERENCES

[1] Z. Gao, P. Reviriego, Z. Xu, X. Su, J. Wang and J. A. Maestro, "Efficient Coding Schemes for Fault-Tolerant Parallel Filters," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 7, pp. 666-670, July 2015.

[2] Z. Gao et al., "Fault Tolerant Parallel Filters Based on Error Correction Codes," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 2, pp. 384-387, Feb. 2015.

[3] K. B. Muralidharan, G. S. Kumar and M. Bhasi, "Fault tolerant state management for high-volume low-latency data stream workloads," 2014 International Conference on Data Science & Engineering (ICDSE), Kochi, 2014, pp. 24-27.

[4] N. Madhuri, S. R. Doradla and M. Surya kalavathi, "Fault tolerant control of Hybrid power filter," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 235-239.

[5] J. Park, J. Park and S. Bhunia, "VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 2, pp. 120-124, Feb. 2014.

[6] A. Asuvaran and S. Senthilkumar, "Low delay error correction codes to correct stuck-at defects and soft errors," 2014 International Conference on Advances in Engineering and Technology (ICAET), Nagapattinam, 2014, pp. 1-7.

[7] P. Reviriego, S. Pontarelli, C. J. Bleakley and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," in Electronics Letters, vol. 48, no. 20, pp. 1258-1260, September 27 2012.

[8] P. P. Vaidyanathan, Multirate Systems and Filter Banks, Englewood Cliffs, N.J., USA: Prentice Hall, 1993.

[9] A. Sibille, C. Oestges and A. Zanella, MIMO: From Theory to Implementation, New York, NY, USA: Academic, 2010.

[10] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances, New York, NY, USA: Springer Verlag, 2010.

[11] M. Nicolaidis, “Design for soft error mitigation,” IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405-418, Sep. 2005.

[12] 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.

[13] A. Reddy and P. Banarjee “Algorithm-based fault detection for sig¬nal processing applications,” IEEE Trans. Comput., vol. 39, no. 10, pp. 1304-1308, Oct. 1990.

[14] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, “Totally fault toler¬ant RNS based FIR filters,” in Proc. IEEEIOLTS, 2008, pp. 192-194.

[15] Z. Gao, W. Yang, X. Chen, M. Zhao and J. Wang, “Fault missing rate analysis of the arithmetic residue codes based

fault-tolerant FIR filter design," in Proc. IEEEIOLTS, 2012, pp. 130-133.

[16] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," IEEE Trans. Very Large Scale Integr. Syst., vol. 14, no. 4, pp. 336-348, Apr. 2006.