An Extensive Survey of Literature on Fault-Tolerance Design for Matrix–Vector Multiplications

Priyanka Tripathi[1], Prof. Rishi Jha[2]

[1]M.Tech. Scholar, [2]Guide

Department of Electronics and Communication Engineering, NIIST, Bhopal

Abstract- Many GPUs and DSP applications demand high output and quick response, performance constraints that usually dictate distinctive architectures with high levels of concurrency. DSP designers would like the aptitude to govern and valuate complicated algorithms to extract the mandatory level of concurrency. Performance constraints may also be addressed by applying different technologies. Modification at the implementation level of style by the insertion of a brand new technology will usually build viable an existing marginal algorithmic rule or design. This examination work outlines an intensive survey of literature on economical fault-tolerance style methodology to realize high performance matrix vector multiplication (MVM). MVM is employed in many variant of applications as well as info retrieval, text classification, and image process. Often, the MVM operations are iterative or repetitive and consume an outsized % of runtime of applications.

Keywords- Fault-Tolerance Design, Parallel Matrix–Vector Multiplications, Error Correction Codes, VLSI, Parallel Processing.

I. INTRODUCTION

Today Communication Engineering has become the vital field of engineering. Over the past few decades the whole world has been enrolled by communication gadgets whether it is television, radio, ATM or any other communication system. All will be dreaded to think world beyond communication. Communication basically involves transfers and reception of information from one place to another place or from one point of time to another point of time. In doing so there may be situations that error may be encountered in the channel due to various factors like Electromagnetic Interferences, Cross talk and Bandwidth limitation etc. Reliability and Efficiency are the two important goals which are to be achieved by all the advancement in the field of communication theory. Reliability refers to the reception of the correct information that was sent through the channel and efficiency refers to the speed of transmission, transceiver‟s complexity and ease in sending and receiving the information. However in most of the situations reliability has to be given preference over efficiency but in some cases, one has to be compensated for another.

The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications. This work presents different multiplier architectures.

Minimizing power consumption for digital systems involves optimization at all levels of the design. This optimization includes the technology used to implement the digital circuits, the circuit style and topology, the architecture for implementing the circuits and at the highest level the algorithms that are being implemented. Digital multipliers are the most commonly used components in any digital circuit design. They are fast, reliable and efficient components that are utilized to implement any operation. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application.

The fault-tolerance system plays a major role in process control, transportation, electronic equipments, space, communications and many other areas that affect our lives. Due to increase in dependency and demand, the size and complexity of computer hardware and software system has grown up extremely. In many systems, the fault-tolerance is achieved by applying a set of analysis and design techniques to generate systems with dramatically improved dependability. In the early days of fault-tolerant computing, it was possible to evaluate specific hardware and software outcomes. The chips are used in embedded systems that contain complex, highly-integrated functions,

and hardware and software must be capable to cope up with a variety of standards subject to techno-economic constraints.

In this examination work an extensive survey of literature on efficient fault-tolerance design for integer parallel matrix–vector multiplications based on recent work in matrix multiplication and error correction codes.

II. FAULT TOLERANCE DESIGN

A fault-tolerance is the ability of a system to continue correct performance of its intended tasks after the occurrence of hardware and software faults. Fault tolerant system research covers a wide spectrum of applications namely embedded real-time systems, commercial transaction systems, transportation systems, and military/space systems, distribution and service systems, etc.. Fault tolerance approach in any system results in the improvement as far as the efficiency and performance is concerned. During last few decades many researchers have contributed in the development of fault tolerance techniques that explore the performance of computer systems which are prone to software or hardware failures. The idea of implementing fault tolerance in separate layers of an embedded system helps in managing the complexity of the derived architectural solutions. Integrating provisions for coping with both hardware and software faults can reduce the overlapping of fault tolerance techniques. The reliability of fault tolerant software presents special difficulties since all the errors present are design errors, that cannot be accurately modeled by the traditional models used for hardware. To tackle fault tolerance related issues, the following terminology are commonly used:

Faults- A fault, sometimes called a bug, is the identified or hypothesized cause of a software failure. Software faults can be classified as (i) design faults and (ii) operational faults according to the phases of creation.

Design faults- A design fault is a fault occurring in the software design and development process. Design faults can be recovered with fault removal approaches by revising the design documentation and source code.

Operational faults- Such faults occur during the lifetime of the system and are invariably due to physical causes, such as processor failures or disk crashes and software operation due to timing, race conditions, workload-related stress and other environmental conditions.

Error- An error is the part of the system state which is liable to lead to a failure. It is an intermediate stage in between faults and failures. Software faults are most often caused by design faults. Design faults occur when a designer, either misunderstands a specification or simply makes a mistake.

Failure- A failure mode is an identifiable weakness in the system design and manufacture. Failures can be classified into severity classes, e.g. critical, major, minor. A failure occurs when the user perceives that a software program is unable to deliver the expected service.

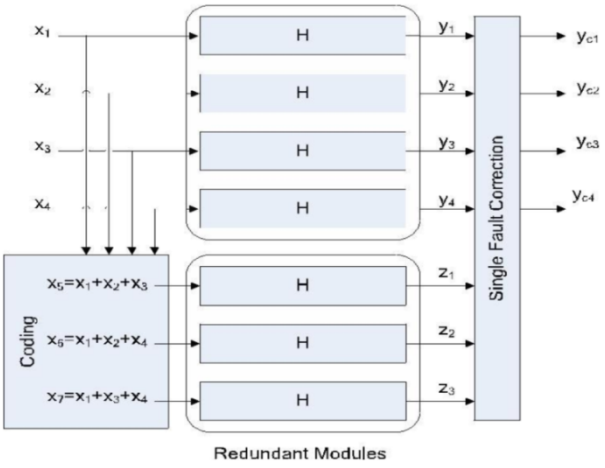

Fig. 2.1 Block representation of Fault tolerant FIR filter

The parallel filters used are designed to detect multiple errors and correct single error. The system consists of original module and a redundant module. The original module comprises of four FIR filters. The redundant module comprises of three FIR filters. In the original module the applied input gets convoluted by using its filter coefficients then it generates the convoluted output. The redundant module is the module used for achieving the reliable operation over the original module. The redundant module is the parity module which is used to generate the parity bits[2]. These parity bits are represented as z1, z2, z3 as shown in fig 2.1.

III. LITERATURE SURVEY

Z. Gao, Q. Jing, Y. Li, P. Reviriego and J. A. Maestro, [1] Parallel matrix processing is a typical operation in many systems, and in particular matrix-vector multiplication (MVM) is one of the most common operations in the modern digital signal processing and digital communication systems. This examination work reported a fault-tolerant design for integer parallel MVMs. The scheme combines ideas from error correction codes with the self-checking capability of MVM. Field-programmable gate array evaluation shows that the proposed scheme can significantly reduce the overheads

compared to the protection of each MVM on its own. Therefore, the proposed technique can be used to reduce the cost of providing fault tolerance in practical implementations.

I. Sayahi, M. Machhout and R. Tourki, [2] In diverse domains, the scientific applications requires severe computing algebra routines. The matrix multiplication presents an indispensable mathematical operation in many high performance fields. This work presents a new FPGA design and implementation for matrix vector multiplication. The design has been implemented with Xilinx System Generator. The results of FPGA implementation were compared with similar work on VIRTEX 4 platform. It demonstrates the efficiency of our work in term of resources utilization and speed up.

I. Zbierska, T. Talaśka and R. Długosz, [3] In this work present a flexible circuit that allows for a parallel and partially asynchronous multiplication of matrices. The circuit, realized in the CMOS technology, has been designed for the application, for example, in fast Kalman filtering in automotive active safety applications. In such filters encounter numerous operations performed on matrices that are time consuming. Additionally, the size of the processed matrices may be the subject of changing, depending on the input data and tasks realized by the system. Software implementation of Kalman filters supports the required flexibility in a natural way, however, this approach is not efficient from the point of view of the computational power. The proposed solution different sizes of the multiplied matrices. Additionally the power dissipation is well fitted to the realized task. The circuit realized in the CMOS 180 nm technology is able to multiply two 3×3 matrices with data rate of 20 - 100 MSps, depending on the signal resolution (in bits) and sizes of the input matrices.

S. S. Inala and P. Pushpalatha, [4] As technology will increase as quality of circuits additionally will increase that tends face reliableness challenges and creates want for fault-tolerant implementations. Signal processing and communication circuits are effected by soft errors. The complexity increases as protection against soft errors increases in many applications. There are distinct advancements in VLSI technology i.e., number of circuits called Fast Fourier Transforms (FFTs) increased on a small chip. In this case have to implement the fault tolerance in order to preserve the data efficiently. There are distinct techniques exist to achieve fault tolerance. The most used technique is algorithmic-primarily based fault tolerance (ABFT) techniques that attempt to use recursive properties to find and accurate errors. But in advanced systems it is not unusual that variety of the filters function in parallel, as an instance a filters having same response are connected in parallel and subjected to different inputs. ECC (error correction codes) is the one of the method to detect and correct errors in FFTs. Propose different multipliers in FFTs architecture for which multiplier is most suitable in whole design. In this proposed method each filter can be considered as a bit. This method allows most efficient protection when there is large number of parallel filters present. The method is evaluated employing a Array, changed booth, Wallace tree and Dadda multipliers showing the potency in terms of speed, low power consumption and space.

B. Kalra and J. B. Sharma, [5] Orthogonal frequency division multiple access is a technique in which single data stream is transmitted with small rate of subcarriers. OFDM system needs FFT processor for generation of subcarriers. A Radix-2 Based 32 bit and 64 Point Pipeline FFT Processor Using bit parallel multiplication process for OFDM system is presented in this work. Proposed architecture uses single path delay feedback FFT architecture. This is an ROM less architecture. In this architecture ROM is replaced by complex constant multiplier and all twiddle factors are multiplied using bit parallel multiplication module. Pipelining methodology provides good latency to the FFT processor. Propagation delay of proposed architecture is calculated by test bench waveform on Xilinx ISE simulator and it is equal to 630ns at 10ns clock period. Due to small propagation delay speed of the OFDM system gets increases and latency also gets improve.

A. Schöll, C. Braun, M. A. Kochte and H. Wunderlich, [6] Reported a fault tolerance approach for sparse matrix operations that detects and implicitly locates errors in the results for efficient local correction. This approach reduces the runtime overhead for fault tolerance and provides high error coverage. Existing algorithm-based fault tolerance approaches for sparse matrix operations detect and correct errors, but they often rely on expensive error localization steps. General checkpointing schemes can induce large recovery cost for high error rates. For sparse matrix-vector multiplications, experimental results show an average reduction in runtime overhead of 43.8%, while the error coverage is on average improved by 52.2% compared to related work. The practical applicability is demonstrated in a case study using the iterative Preconditioned Conjugate Gradient solver. When scaling the error rate by four orders of magnitude, the average runtime overhead increases only by 31.3% compared to low error rates.

K. Z. Woo, P. K. Chong and L. K. Chian, [7] Multiplications are often involved in Digital Signal Processing such as for digital filters and FFT (Fast Fourier Transform). It requires high speed multipliers and logic components (such as adders, subtractors, and shifters). Processing speed is always critical for Digital Signal Processing, and there are many attempts to reduce the processing latency that may cause performance issues on the end product. There has been a number of parallel multiplication approaches proposed to speed up computation. This examination work aims to design and implement several parallel multiplication approaches using Field Programmable Gate Array (FPGA). These approaches make use of the resources in FPGA to achieve fast multiplication. The parallel methods implemented and compared in this examination include partitioning of multiplicands for parallel multiplication, hybrid Look-up tables (LUT) parallel multiplication, and Wallace Tree Multiplication algorithm. Comparison is made based on the number of processing cycles and also the amount of resources used in the FPGA through simulation. The proposed designs utilized lesser processing cycles (5 cycles) for a single process by using the FPGA resources effectively.

Z. Gao et al., [8] 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.

Table 1: Summary of Literature Survey

SR NO	TITLE	AUTHOR	YEAR	APPROACH
1	An Efficient Fault-Tolerance Design for Integer Parallel Matrix–Vector Multiplications	Z. Gao, Q. Jing, Y. Li, P. Reviriego and J. A. Maestro	2018	Reported a fault-tolerant design for integer parallel MVMs. The scheme combines ideas from error correction codes with the self-checking capability of MVM
2	FPGA implementation of matrix-vector multiplication using Xilinx System Generator	I. Sayahi, M. Machhout and R. Tourki	2018	This work presents a new FPGA design and implementation for matrix vector multiplication.
3	Parallel matrix multiplication in 2-gain Kalman filter realized in hardware	I. Zbierska, T. Talaśka and R. Długosz	2017	In this examination a flexible circuit that allows for a parallel and partially asynchronous multiplication of matrices has presented
4	Relative performance of multipliers: A fault tolerance perspective for parallel FFTs,	S. S. Inala and P. Pushpalatha,	2017	Implement the fault tolerance in order to preserve the data efficiently. There are distinct techniques exist to achieve fault tolerance
5	Bit parallel multiplication based 64 Point Pipeline FFT Processor for OFDM application	B. Kalra and J. B. Sharma	2016	A Radix-2 Based 32 bit and 64 Point Pipeline FFT Processor Using bit parallel multiplication process for OFDM system is presented
6	Efficient Algorithm-Based Fault Tolerance for Sparse Matrix Operations	A. Schöll, C. Braun, M. A. Kochte and H. Wunderlich	2016	A fault tolerance approach for sparse matrix operations that detects and implicitly locates errors in the results for efficient local correction.
7	Implementation of parallel	K. Z. Woo, P. K. Chong	2015	This examination aims to design and

	multiplications on FPGA	and L. K. Chian		implement several parallel multiplication approaches using Field Programmable Gate Array (FPGA).
8	Fault Tolerant Parallel Filters Based on Error Correction Codes	Z. Gao et al	2015	A simple technique that exploits the presence of parallel filters to achieve fault tolerance has been presented

IV. PROBLEM STATEMENT

An important domain in computer design is parallel processing. Additionally there's hardware underutilization as a results of serial execution in parallel machine. Electronic circuits are progressively appearing in automotive, medical and space applications wherever reliability is essential. In those applications, the circuits got to give some extent of fault tolerance. A number of techniques will be utilized defend a circuit from errors. Those vary from modifications within the producing method of the circuits to scale back the amount of errors to adding redundancy at the logic or system level to confirm that errors don't have an effect on the system performance. Once the circuit to be protected has recursive or structural properties, a more robust choice will be to take advantage of those properties to implement fault tolerance. Digital filters are one in all the foremost normally used signal process circuits and a number of other techniques are planned protect them from errors. Parallel processing with error correction has provided fast processing with fault tolerance capabilities. But designing of hardware is always having challenging to keep in mind area and power with speed in terms of delay.

V. CONCLUSION

In this work presented an extensive survey of literature on fault tolerance design of parallel vector matrix multiplication. This examination work introduces techniques for high performance MVM using Error Correction Codes and real applications that take advantage of the work. The main contribution of this literature review is to investigate uniform design strategy that proves to work well parallel fault tolerance architecture. It is flexible in the sense that it can be applied on wide range of problem sizes, and that its performance can be controlled through a small set of tuning parameters. The main parameter considered are speed in terms of delay and area in terms resources utilized using FPGA.

REFERENCES

[1]. Z. Gao, Q. Jing, Y. Li, P. Reviriego and J. A. Maestro, "An Efficient Fault-Tolerance Design for Integer Parallel Matrix–Vector Multiplications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 1, pp. 211-215, Jan. 2018

[2]. I. Sayahi, M. Machhout and R. Tourki, "FPGA implementation of matrix-vector multiplication using Xilinx System Generator," 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, 2018, pp. 290-295

[3]. Zbierska, T. Talaśka and R. Długosz, "Parallel matrix multiplication in 2-gain Kalman filter realized in hardware," 2017 IEEE 30th International Conference on Microelectronics (MIEL), Nis, 2017, pp. 101-104

[4]. S. S. Inala and P. Pushpalatha, "Relative performance of multipliers: A fault tolerance perspective for parallel FFTs," 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, 2017, pp. 194-198

[5]. B. Kalra and J. B. Sharma, "Bit parallel multiplication based 64 Point Pipeline FFT Processor for OFDM application," 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016, pp. 1-5.

[6]. A. Schöll, C. Braun, M. A. Kochte and H. Wunderlich, "Efficient Algorithm-Based Fault Tolerance for Sparse Matrix Operations," 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, 2016, pp. 251-262

[7]. K. Z. Woo, P. K. Chong and L. K. Chian, "Implementation of parallel multiplications on FPGA," 2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, 2015, pp. 32-37

[8]. 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.

[9]. Q. Qi and C. Chakrabarti, “Parallel high throughput soft-output sphere decoder,” in Proc. IEEE Workshop Signal Process. Syst., Oct. 2010, pp. 174–179.

[10]. M. Wu, C. Dick, and J. R. Cavallaro, “Improving MIMO sphere detection through antenna detection order scheduling,” in Proc. SDR Forum, 2011, pp. 280–284.

[11]. G. Wang, M. Wu, Y. Sun, and J. R. Cavallaro, “A massively parallel implementation of QC-LDPC decoder on GPU,” in Proc. IEEE 9th Symp. Appl. Specific Process. (SASP), Jun. 2011, pp. 82–85.

[12]. G. Wang, H. Shen, B. Yin, M. Wu, Y. Sun, and J. R. Cavallaro, “Parallel nonbinary LDPC decoding on GPU,” in Proc. ASILOMAR Conf., Nov. 2012, pp. 1277–1281.