An Extensive Review On Fault Tolerant Parallel Filter Computation

Priyanka Patel[1], Prof. Tarun Verma

[1]*Research scholar,* [2]*Guide*

Department of Electronics and Communication, LNCT, Bhopal

Abstract – The demand for processing power is increasing steadily. In many application fields, there can never be enough computing power. Simulations in the field of engineering, like virtual crash tests, or in the field of bioinformatics, as protein folding, are examples for applications that require enormous computing power. But even consumer PCs continue to demand for more and more computing power. Moore's Law, predicting that the performance of microprocessors doubles about every 18 months, has proven to be true in the past, and will most likely stay true for the near future. One contributing factor to this performance increase is technological improvements. However, the direct influence of technology on computing performance is limited. Architectural improvements are another main source for sustained performance improvements. 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

Keywords – Digital Filter ,Error correction code, Fault Tolerant ,Soft Error .

I. INTRODUCTION

The role of electronics in every branch of science has become pivotal. Integrated circuits, digital and analog, are used extensively in applications ranging from medical to home-automation. System reliability has been a major concern since the dawn of the electronic digital computer age . As the scale of integration increased from small/medium to large and to today's very large scale, the reliability per basic function has continued its dramatic improvement. Due to the demand for enhanced functionality, the complexity of contemporary computers, measured in terms of basic functions, rose almost as fast as the improvement in the reliability of the basic component. Secondly, our dependence on computing systems has grown so great that it becomes impossible to return to less sophisticated mechanisms. Previously, reliable computing has been limited to military, industrial, aerospace, and communications applications in which the consequence of computer failure had significant economic impact and/or loss of life. Today even commercial applications require high reliability as we move towards a cashless/automated life-style. Reliability is of critical importance in situations where a computer malfunction could have catastrophic results. Reliability is used to describe systems in which it is not feasible to repair (as in computers on board satellites) or in which the computer is serving a critical function and cannot be lost even for the duration of a replacement (as in flight control computers on an aircraft) or in which the repair is prohibitively expensive. Systems that tolerate failures have been of interest since the 1940's when computational engines were constructed from relays.

High reliability is needed in many signal processing applications to ensure continuous operation and to check the integrity of results. High reliability is needed in life critical applications, such as aircraft guidance systems or in medical equipment, where failures can jeopardize human lives, or in remote applications, such as satellites or underwater acoustic monitors, where repair is impossible or prohibitively expensive. Robustness is also needed in systems that must operate in hazardous environments, such as military equipment, or in spacecraft that must be protected against radiation. In all of these applications there is a high cost of failure, and reliability is of great importance.

The complexity of signal processing algorithms has been steadily increasing due to the availability of special purpose, high-speed signal processors. Many algorithms that were once too computationally intensive are now implemented in real time by multiprocessor systems. In these systems, the large amount of hardware increases the likelihood of a failure occurring, and makes reliable operation difficult. It is impossible to guarantee that components of a system will never fail. Instead, failures should be anticipated, and systems designed to tolerate failures gracefully. This design methodology is known as fault-tolerant computing and it received considerable attention by early computer designers because of the unreliability of existing components after the development of integrated circuits, which were several orders of magnitude more reliable, fault-tolerance became a secondary issue. Attention was focused on developing faster, more complex circuits, and as a result, circuit densities grew exponentially. In many areas, however, semiconductor reliability has not kept pace with the level

of integration, and fault-tolerance is becoming a major issue again.

A component is said to have failed when it does not compute the correct output, given its input. A failure is caused by a physical fault, and the manifestation of a failure is errors within the system. In a fault-tolerant system, the basic idea

Is to tolerate internal errors, and keep them from reaching and corrupting the output. This process is known as error masking.

The highest and most desirable level of fault-tolerance is known as concurrent error masking. In this technique, errors are masked during operation and the system continues to function with no visible degradation in performance. In general, concurrent error masking usually requires the following steps:

A. Fault Detection - determine that the output is invalid and that a fault has occurred within the system.

B. Fault Location - determine which system component failed.

C. Fault Correction - determine the correct output.

Concurrent error masking is difficult and expensive to achieve, and basically requires the entire system to be checked for errors at each time step. Often, lower levels of protection are acceptable and a few erroneous outputs may be tolerated. In these instances, less expensive techniques which check only a part of the system at each time step are adequate. In other applications, there is a high cost for computing erroneous results and a lower cost for delaying the result. Is these applications, concurrent fault detection coupled with off-line fault location/correction may be a viable alternative.

A diagnosis test is characterized by its diagnostic resolution, defined as the ability to get closer to the fault. When a failure is indicated by a pass/fail type of test in a system that is operating in the field, a diagnostic test is applied. We have employed different types of diagnosis mechanisms for the ALU. We found that the best diagnosis mechanism for the 64-bit ALU is designing the Boolean, addition, subtraction and shifting operations with thirty two reconfigurable 2- bit ALU chunks and designing the multiplier separately. We had to split the multiplier into identical bit slices of Booth encoders, Booth selectors, full adders, half adders and carry propagation adders. It was easy to reconfigure the multiplier once it was split into identical bit slices if a fault is detected and a permanent failure located, the system may be able to reconfigure its components to replace the failed component or to isolate it from the rest of the system.

Digital Filters

Digital filters are discrete-time systems. A discrete-time system is essentially an algorithm for converting an input sequence into an output sequence. The input signal x(n) is transformed by the system into a signal y(n), and is expressed by the general relationship between x(n) and y(n) as follows:

$$y(n) = H[x(n)] \ldots\ldots\ldots\ldots (1)$$

The symbol H denotes the transformation performed by the system on x (n) to produce y (n). Figure 2 graphically illustrates the mathematical relationship of Equation 1

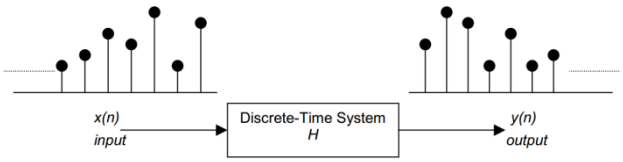

Figure: 1 Block diagram representation of a discrete-time system.

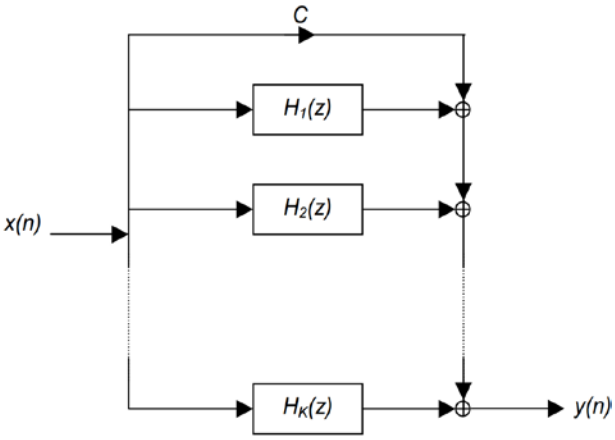

Figure:2 Parallel Form Structure of IIR Digital Filter.

II. LITERATURE SURVEY

Z. Gao et al, 2015 [1] 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.

Z. Gao, W. Yang, X. Chen, M. Zhao and J. Wang,2012 [2] Relative to the Triple Modular Redundancy (TMR) scheme, the arithmetic residue codes based fault-tolerant DSP design consumes much less resources. However, the price for the low resource consumption is the fault missing problem. The basic tradeoff is that, smaller modulus used for the fault checking consumes fewer resources, but the fault missing rate is higher. The relationship between the value of modulus and the fault missing rate is analyzed theoretically in this research work for fault-tolerant FIR filter design, and the results are verified by FPGA implemented fault injections.

P. Reviriego, C. J. Bleakley and J. A. Maestro,2011 [3] In this brief, an efficient technique for implementation of soft-error-tolerant finite impulse response (FIR) filters is presented. The proposed technique uses two implementations of the basic filter with different structures operating in parallel. A soft error occurring in either filter causes the outputs of the filters to differ, or mismatch, for at least one sample. The filters are specifically designed so that, when a soft error occurs, they produce distinct error patterns at the filter output. An error detection circuit monitors the basic filter outputs and identifies any mismatches. An error correction circuit determines which filter is in error based on the mismatch pattern and selects the error-free filter result as the output of the overall error-protected system. This technique is referred to as structural dual modular redundancy (DMR) since it enhances traditional DMR to provide error correction, as well as error detection, by means of filter modules with different structures. The proposed technique has been implemented and evaluated. The system achieves a soft error correction rate of close to 100% for isolated single soft errors and has a logic complexity significantly less than that of conventional triple modular redundancy.

Y. H. Huang, 2010 [4] the soft error problem in digital circuits is becoming increasingly important as the IC fabrication technology progresses from the deep submicrometer scale to the nanometer scale. This research work proposes a subword-detection processing (SDP) technique and a fine-grain soft-error-tolerance (FGSET) architecture to improve the performance of the digital signal processing circuit. In the SDP technique, the logic masking property of the soft error in the combinational circuit is utilized to mask the single-event upset (SEU) caused by disturbing particles in the inactive area. To further improve the performance, the masked portion of the datapath can be used as the estimation redundancy in the algorithmic softerror-tolerance (ASET) technique. This technique is called subword-detection and redundant processing (SDRP). In the FGSET architecture, the soft error in each processing element (fine grain) can be recovered by the arithmetic datapath-level ASET technique. Analysis of the fast Fourier transform processor example shows that the proposed FGSET architecture can improve the performance of the coarse-grain SET (CGSET) by 8.5 dB. The low-cost SDP technique (1.03x) yields a noise reduction of 5.3 dB over the CGSET approach (1.40x), while the efficient SDRP I (1.57x) and SDRP II (1.88x) techniques outperform the CGSET approach by 24.5 and 30.5 dB, respectively.

S. Pontarelli, G. C. Cardarilli, M. Re and A. Salsano,2008 [5] In this research work, the design of a finite impulse response (FIR) filter with fault tolerant capabilities based on the residue number system is analyzed. Differently from other approaches that use RNS, the filter implementation is fault tolerant not only with respect to a fault inside the RNS moduli, but also in the reverse converter. An architecture allowing fault masking in the overall RNS FIR filter is presented. It avoids the use of a trivial triple modular redundancy (TMR) to protect the blocks that performs the final stages of the RNS based FIR computation.

Byonghyo Shim and N. R. Shanbhag,2006[6] In this exploration, Authors present energy-efficient soft error-tolerant techniques for digital signal processing (DSP) systems. The proposed technique, referred to as algorithmic soft error-tolerance (ASET), employs low-complexity estimators of a main DSP block to achieve reliable operation in the presence of soft errors. Three distinct ASET techniques - spatial, temporal and spatio-temporal- are presented. For frequency selective finite-impulse response (FIR) filtering, it is shown that the proposed techniques provide robustness in the presence of soft error rates of up to P/sub er/=10/sup -2/ and P/sub er/=10/sup -3/ in a single-event upset scenario. The power dissipation of the proposed techniques ranges from 1.1 X to 1.7 X (spatial ASET) and 1.05 X to 1.17 X (spatio-temporal and temporal ASET) when the desired signal-to-noise ratio SNR/sub des/=25 dB. In comparison, the power dissipation of the commonly employed triple modular redundancy technique is 2.9 X.

M. Nicolaidis, 2005[7] In nanometric technologies, circuits are increasingly sensitive to various kinds of perturbations. Soft errors, a concern for space applications in the past, became a reliability issue at ground level. Alpha particles and atmospheric neutrons induce single-event upsets (SEU), affecting memory cells, latches, and flip-flops, and single-event transients (SET), initiated in the combinational logic and captured by the latches and flip-flops associated to the outputs of this logic. To face this challenge, a designer must dispose a variety of soft error mitigation schemes adapted to various circuit structures, design architectures, and design constraints. In this research work, Authors describe various SEU and SET mitigation schemes that could help the designer meet her or his goals.

A. L. N. Reddy and P. Banerjee, 1990 [8] the increasing demands for high-performance signal processing along with the availability of inexpensive high-performance processors have results in numerous proposals for special-purpose array processors for signal processing applications. A functional-level concurrent error-detection scheme is presented for such VLSI signal processing architectures as those proposed for the FFT and QR factorization. Some basic properties involved in such computations are used to check the correctness of the computed output values. This fault-detection scheme is shown to be applicable to a class of problems rather than a particular problem, unlike the earlier algorithm-based error-detection techniques. The effects of roundoff/truncation errors due to finite-precision arithmetic are evaluated. It is shown that the error coverage is high with large word sizes

III. PROBLEM IDENTIFICATION

The new technique is based on the use of the ECCs. A simple ECC takes a block of k bits and produces a block of n bits by adding n — k parity check bits. The parity check bits are XOR combinations of the k data bits. By properly designing those combinations it is possible to detect and correct errors [1]. the proposed scheme can also be applied to the IIR filters. Future work will consider the evaluation of the benefits of the proposed technique for IIR filters. The extension of the scheme to parallel filters that have the same input and different impulse responses is also a topic for future work. The proposed scheme can also be combined with the reduced precision replica approach presented in [3] to reduce the overhead required for protection. This will be of interest when the number of parallel filters is small as the cost of the proposed scheme is larger in that case. Another interesting topic to continue this brief is to explore the use of more powerful multibit ECCs, such as Bose-Chaudhuri-Hocquenghem codes, to correct errors on multiple filters.

IV. CONCLUSION

Real-time systems possess stringent requirements for fault tolerance because faulty hardware could jeopardize human life. For such systems, checkers are employed so that incorrect data never leaves the faulty module and recovery time from faults is minimal. We have investigated information, hardware and time redundancy. After analyzing the hardware, the delay and the power overheads we have decided to use time redundancy as our fault tolerance method for the IIR digital filter.

REFERENCES

[1] 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.

[2] 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. IEEE IOLTS, Jun. 2012, pp. 130-133.

[3] P. Reviriego, C. J. Bleakley, and J. A. Maestro, “Strutural DMR: A technique for implementation of soft-error-tolerant FIR filters,” IEEE Trans. Circuits Syst., Exp. Briefs, vol. 58, no. 8, pp. 512-516, Aug. 2011.

[4] Y.-H. Huang, “High-efficiency soft-error-tolerant digital signal process¬ing using fine-grain subword-detection processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 291-304, Feb. 2010.

[5] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, “Totally fault tolerant RNS based FIR filters,” in Proc. IEEE IOLTS, Jul. 2008, pp. 192-194.

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

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

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

[9] T. Hitana and A. K. Deb, “Bridging concurrent and non-concurrent error detection in FIR filters,” in Proc. Norchip Conf., 2004, pp. 75-78. [9]

[10] P. P. Vaidyanathan. Multirate Systems and Filter Banks. Upper Saddle River, NJ, USA: Prentice-Hall, 1993. [10] A. Sibille, C. Oestges, and A. Zanella, MIMO: From Theory to Imple-mentation. San Francisco, CA, USA: Academic Press, 2010.

[11] P. Reviriego, S. Pontarelli, C. Bleakley, and J. A. Maestro, “Area efficient concurrent error detection and correction for paral¬lel filters,” IET Electron. Lett., vol. 48, no. 20, pp. 1258-1260, Sep. 2012.

[12] A. V. Oppenheim and R. W. Schafer, Discrete Time Signal Processing. Upper Saddle River, NJ, USA: Prentice-Hall 1999.

[13] S. Lin and D. J. Costello, Error Control Coding, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall. 2004.

[14] R. W. Hamming, “Error correcting and error detecting codes,” Bell Syst. Tech. J., vol. 29, pp. 147-160, Apr. 1950