INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 3 (2016) PAGES 1077-1081
RECEIVED : 05/06/2016. PUBLISHED : 21/06/2016
June 21, 2016

EFFICIENT ARCHITECTURE OF EFFICIENT CODING SCHEME FOR FAULT TOLER PARALLEL FILTERS

[1]M.Shanmuga Priya, [2]Dr.S.Sathish,
[1]PG Scholar, Dept of Embedded System Technologies, Jayalakshmi Institute Of Technology, Thoppur,
[2]Asst prof, Dept of ECE, Jayalakshmi Institute Of Technology, Thoppur.

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 are 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 paper, 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 an FPGA implementation.

Keywords – Complexity, FPGA, Filter.

1. INTRODUCTION

Electronic circuits are increasingly present in automotive, medical, and space applications where reliability is critical. In those applications, the circuits have to provide some degree of fault tolerance.To add redundancy, a general technique known as triple modular redundancy (TMR) can be used. The TMR, which triplicates the design and adds voting logic to correct errors, is commonly used. However, it more than triples the area and power of the circuit, something that may not be acceptable in some applications. When the circuit to be protected has algorithmic or structural properties, a better option can be to exploit those properties to implement fault tolerance. Digital filters are one of the most commonly used signal processing circuits and several techniques have been proposed to protect them from errors. Most of them have focused on finite-impulse response (FIR) filters. Digital filters are widely used in signal processing and communication systems. 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. 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.

2. LITERATURE SURVEY

FPGA-based designs are more susceptible to single-event up-sets (SEUs) compared to ASIC designs, since SEUs in configuration bits of FPGAs result in permanent errors in the mapped de-sign. Moreover, the number of sensitive configuration bits is two orders of magnitude more than user bits in

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 3 (2016) PAGES 1077-1081
RECEIVED : 05/06/2016. PUBLISHED : 21/06/2016
June 21, 2016

typical FPGA-based circuits. In this paper, we present a high-reliable low-cost mitigation technique which can significantly improve the availability of designs mapped into FPGAs. Experimental results show that ,using this technique, the availability of an FPGA mapped design can be increases to more than 99%. 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-basederror-detection techniques.

3. PROPOSED SYSTEM

Hamming's development [Ham] is a very direct construction of a code that permits correcting single-bit errors. He assumes that the data to be transmitted consists of a certain number of information bits u, and he adds to these a number of check bits p such that if a block is received that has at most one bit in error, then p identifies the bit that is in error (which may be one of the check bits). Let k be the number of information bits, and m the number of check bits used. Because the m check bits must check themselves as well as the information bits, the value of p, interpreted as an integer, must range from 0 to which is distinct values. Because m bits can distinguish cases, we must have (1) This is known as the Hamming rule. Because p indexes the bit (if any) that is in error, the least significant bit of p must be 1 if the erroneous bit is in an odd position, and 0 if it is in an even position or if there is no error. A simple way to achieve this is to let the least significant bit of p, p0, be an even parity check on the odd positions of the block, and to put p0 in an odd position. The receiver then checks the parity of the odd positions (including that of p0). If the result is 1, an error has occurred in an odd position, and if the result is 0, either no error occurred or an error occurred in an even position. . This satisfies the condition that p should be the index of the erroneous bit, or be 0 if no error occurred. Similarly, let the next from least significant bit of p, p1, be an even parity check of positions 2, 3, 6, 7, 10, 11, … (in binary, 10, 11, 110, 111, 1010, 1011, …), and put p1 in one of these positions. Those positions have a 1 in their second from least significant binary position number.

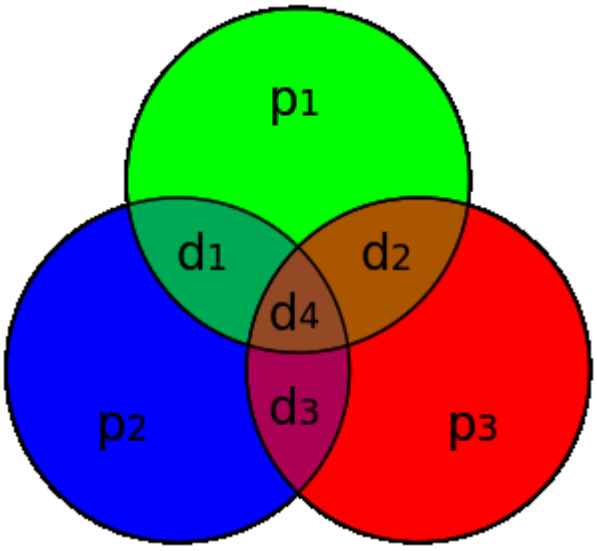

Fig.1. Graphical depiction of the four data bits and three parity bits

Hamming studied the existing coding schemes, including two-of-five, and generalized their concepts. To start with, he developed a nomenclature to describe the system, including the number of data bits and error-correction bits in a block. For instance, parity includes a single bit for any data word, so assuming ASCII words with seven bits, Hamming described this as an (8,7) code, with eight bits in total, of which seven are data. The repetition example would be (3,1), following the same logic. The code rate is the second number divided by the first, for our repetition example, 1/3.Hamming also noticed the problems with flipping two or more bits, and described this as the "distance" (it is now called the Hamming distance, after him). Parity has a distance of 2, so one bit flip can be detected, but not corrected and any two bit flips will be invisible. The (3,1) repetition has a distance of 3, as three bits need to be flipped in the same triple to obtain another code word with no visible errors.

4. SOFTWARE REQUIRMENTS

ModelSim is a useful tool that allows you to stimulate the inputs of your modules and view both outputs and internal signals. It allows you to do both behavioural and timing simulation, however, this document will focus on behavioural simulation. Keep in mind that these simulations are based on models and thus the results are only as accurate as the constituent models. ModelSim /VHDL, ModelSim /VLOG, ModelSim /LNL, and ModelSim /PLUS are produced by Model Technology™ Incorporated. Unauthorized copying, duplication, or other reproduction is prohibited without the written consent of Model Technology. The information in this manual is subject to change without notice and does not represent a commitment on the part of Model Technology. The program described in this manual is furnished under a license agreement and may not be used or copied except in accordance with the terms of the agreement. The online documentation provided with this product may be printed by the end-user. The number of copies that may be printed is limited to the number of licenses purchased. ModelSim is a registered trademark of Model Technology Incorporated. Model Technology is a trademark of Mentor Graphics Corporation. PostScript is a registered trademark of Adobe Systems Incorporated. UNIX is a registered trademark of AT&T in the USA and other countries. FLEXlm is a trademark of Globetrotter Software, Inc. IBM, AT, and PC are registered trademarks, AIX and RISC System/6000 are trademarks of International Business Machines Corporation. Windows, Microsoft, and MS-DOS are registered trademarks of Microsoft Corporation. OSF/Motif is a trademark of the Open Software Foundation, Inc. in the USA and other countries.

5. RESULT ANALYSIS

The dramatic increase in the logic density of silicon chips has made it possible to implement digital systems with multimillion gates on a single chip. The complexity of such systems makes it impractical to use traditional design descriptions (e.g., logic schematics) to provide a complete and accurate description of a design. Currently, all complex digital designs are expressed using a hardware description language (HDL). An HDL, unlike traditional programming languages such as C or C++, can describe functions that are inherently parallel. A major advantage of an HDL is that it provides a better and more concise documentation of a design than gate-level schematics. Two very popular HDLs are VHDL and VERILOG. In this text we use VHDL. VHDL is an acronym for VHSIC hardware description language; VHSIC in turn is an acronym for very high speed integrated circuit. The development of VHDL was funded by the U.S. Department of Defense (DoD) in the early 1980s. The syntax of VHDL is similar to that of programming language ADA; however, it has some significant differences from ADA. We present the important concepts of VHDL, especially the ones that are used in digital circuit design. . VHDL can provide unambiguous representation of a design at

different levels of abstraction as shown. Modern CAD (computer-aided design) tools can generate gate-level implementation of a design from its VHDL description. A behavioral VHDL

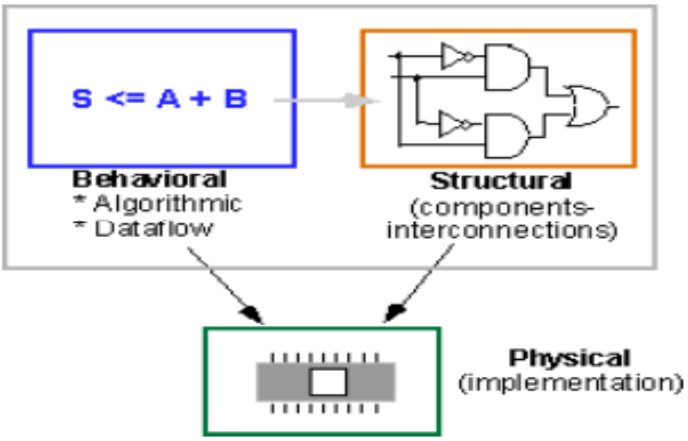

Fig.2. Behavioral, Structural and Physical

description of a circuit describes the function of the circuit in terms of its inputs using the types of statements used in a high-level programming language. The objective is to describe the correct operation of a circuit to be designed without being concerned with redundant details. This description does not specify how the function is actually implemented; thus the same description may result in several implementations of a circuit.

CONCLUSION

This brief has presented a new scheme to protect parallel filters that are commonly found in modern signal processing circuits. The approach is based on applying ECCs to the parallel filters outpus to detect and correct errors. The scheme can be used for parallel filters that have the same response and process different input signals. A case study has also been discussed to show the effectiveness of the scheme in terms of error correction and also of circuit overheads. The technique provides larger benefits when the number of parallel filters is large. 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.

REFERENCES

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

[2] 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.

[3] 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.

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

[5] Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword- setection processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 291–304, Feb. 2010.

[6] 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.

[7] 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.