International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 20 May 2017. Link :
http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-3

Title: Design Efficient 4-Bit Error Correction & Detection For 16-Bit Transformation

Volume o6, Issue 03, Pages: 185 – 191.

Paper Authors

K.V.V.P.Parvathi Devi, G.Gowtham.
Dept of ECE A.K.R.G College of Engineering& Technology, Nallajerla, A.P . India.
.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN EFFICIENT 4-BIT ERROR CORRECTION & DETECTION FOR 16-BIT TRANSFORMATION

K.V.V.P.Parvathi Devi
M.Tech., E.C.E Department
A.K.R.G College of Engineering& Technology
Nallajerla, A.P . India

G.Gowtham
Assistant Professor , E.C.E Department
A.K.R.G College of Engineering & Technology
Nallajerla, A.P . India

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 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.

Keywords- Coding, parallel filters, Hamming code,Sos checks, soft errors.

I. INTRODUCTION

Fault-tolerance is defined as the ability to produce correct results even in the presence of faults. Fault-tolerance is not a replacement of fault-prevention approach but a complement to it. Research activities in the area of fault-tolerant design have increased recently due to the following factors which have had a majorimpact on the design of these systems. I- Advances in Very Large Scale Integration (VLSI) technology have resulted in complex chips. This complexity could make the IC's susceptible to a diverse variety of failures and could lead to a decrease in reliability. II- Lower cost of extremely complex components and devices have made it

economical to introduce redundancy into the system.

III- Testing of complex components and systems are time-consuming and expensive, moreover there is a shortage of test equipment and experts.

IV- There is an ever-increasing demand for high reliability systems to undertake safety-critical applications, despite the fact that there is an upper limit to the reliability levels that can be achieved, using the fault-prevention approach.

V- In many applications, the system down-time needs to be minimized or even eliminated to improve the availability of the system. Basic to the design and implementation of fault-tolerant computing systems are consideration of the following three factors. Firstly, it is necessary to identify the basic principles which underlie all fault tolerant systems. Principles that can be applied at all levels in a system. Secondly, the measures and mechanisms to support and implement techniques based on these principles must be investigated. Thirdly, a framework is required to support a well structured approach to fault-tolerance in order to ensure that the additional complexity introduced by the fault tolerance techniques does not reduce rather than increase the reliability of the system.

Principles of Fault – Tolerance

To prevent faults leading to system failures five phases should be identified.

1] Error detection The presence of a fault in a system can produce an error which can cause a failure in the system. In order to tolerate a fault in a system generally its effects must first be detected, therefore an error detection mechanism should be deployed.

II) Reconfiguration If a fault is detected and a permanent failure located, the system should be able to reconfigure its components to replace the failed component or to isolate it from the rest of the system.

III) Retry In many cases a second attempt at an operation may be successful. This is particularly true in case of a transient fault. Thus a retry mechanism should be available in the system to handle these cases.

IV) Reset An error may cause too much damage to the system such that retry can not be successful and recovery may not be possible. In this case the system needs to be reset or restarted and therefore the design should provide these facilities.

IV) Fault treatment and continued service

After detection and (if necessary) reconfiguration, the effects of errors must be eliminated, and retry or reset should be used to check if the component can be used again (e.g. if failure caused by transient error). If possible the failed component should be replaced, repaired, and then put back to service. It is possible to have a fault-tolerant design with different permutations of this procedure. However to have an effective independent system all five phases are required. Having identified the principles of fault-tolerant design, now their implementation in hardware systems must be considered. The question of *how* fault-tolerance can be implemented in a system will be addressed in the next sections. But the questions of where fault-tolerance is

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

actually required and how muchis necessary, which concerns the reliability requirement, will be discussed in chapter five.

II. ECC-BASED PROTECTION OF PARALLEL FILTERS

The impulse response $h[n]$ completely defines a discrete time filter that performs the following operation on the incoming signal $x[n]$:

$$y[n] = \sum_{l=0}^{\infty} x[n-l] \cdot h[l].$$

The impulse response can be infinite or be nonzero for a finite number of samples. In the first case, the filter is an infinite impulse-response (IIR) filter, and in the second, the filter is a finite impulse-response (FIR) filter. In both cases, the filtering operation is linear such that

$$y_1[n] + y_2[n] = \sum_{l=0}^{\infty} (x_1[n-l] + x_2[n-l]) \cdot h[l]$$

This property can be exploited in the case of parallel filters that operate on different incoming signals, as shown on Fig. 1. In this case, four filters with the same response process the incoming signals $x1[n]$, $x2[n]$, $x3[n]$, and $x4[n]$ to produce four outputs $y1[n]$, $y2[n]$, $y3[n]$, and $y4[n]$. To detect and correct errors, each filter can be viewed as a bit in an ECC, and redundant filters can be added to form parity check bits . This is also illustrated in Fig. 1, where three redundant filters are used to form the parity check bits of a classical single error correction Hamming code . Those correspond to the outputs $z1[n]$, $z2[n]$, and $z3[n]$. Errors can be detected by checking if

$$z_1[n] = y_1[n] + y_2[n] + y_3[n]$$
$$z_2[n] = y_1[n] + y_2[n] + y_4[n]$$
$$z_3[n] = y_1[n] + y_3[n] + y_4[n].$$

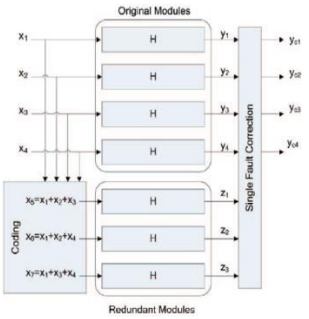
When some of those checks fail, an error is detected. The error can be corrected based on which specific checks failed. For example, an error on filter $y1$ will cause errors on the checks of $z1$, $z2$, and $z3$. Similarly, errors on the other filters will cause errors on a different group of zi. Therefore, as with the traditional ECCs, the error can be located. To correct the error, the failing output is reconstructed from the correct outputs. For example, when an error on $y1$ is detected, it can be corrected by making

$$y_{c1}[n] = z_1[n] - y_2[n] - y_3[n].$$

This ECC-based scheme reduces the protection overhead compared with the use of TMR. Table I summarizes the number of redundant filters needed for different parallel filter configurations. It can be observed that the number grows with the logarithm in base two on the number of filters. Therefore, the cost is

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

much smaller than TMR, in which the number of filters is tripled. The cost reductions were confirmed by some case study implementations. In this ECC-based scheme, the coding of the redundant filters is based on simple additions that replace the XOR binary operations in traditional ECCs. However, since both the inputs and outputs of the filters are sequences of numbers, a more general coding can be used. This type of coding has been explored for linear time-invariant systems (see, for example, but not for parallel filters. In those works, the processing of the linear system is modified to incorporate error detection and correction mechanisms. This is different from the approach proposed in this brief, where inputs are encoded but the processing of the filters is not modified. In the following, the use of a coding scheme for parallel filters in which the redundant filters are constructed as linear combinations of the original filters with arbitrary coefficients is explored.



ECC-based scheme for four filters and a Hamming code

FIG. 1 EXISTED SYSTEM

TABLE I
NUMBER OF REDUNDANT FILTERS IN THE ECC-BASED APPROACH

Number of parallel filters	Number of redundant filters
4	3
8	4
16	5
32	6

III.PROPOSED SYSTEM

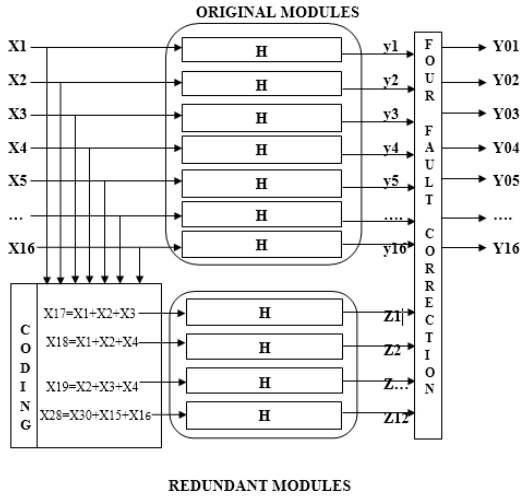


FIG. 2 PROPOSED SYSTEM

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly sought after in high-availability or life-critical systems.In

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

parallel FFTS, the sum of squares techniques is combined to detect the errors and correct them. So this can be done by using the equivalent of simple parity bit for all the FFTS. In this when an error is found, the old of the parity FFT is used to correct the errors.

Now, this will be explained in a proposed technique for sixteen parallel FFTS. Here to the input a redundant FFT is added that as sum of two original FFTS. Sum of squares also added to check the error. IP error is found then correction can be done by recomputing the FFT in error using old of parity FFT (x) and rest of FFT outputs.

So in this way also we can detect the errors. There is a technique to detect errors (i.e.) we combine the sum of squares and ECC approach then we can detect the errors. Same process is applied to this technique to check the errors.

IV. RESULTS

The output waveform is shown in below figure for proposed system

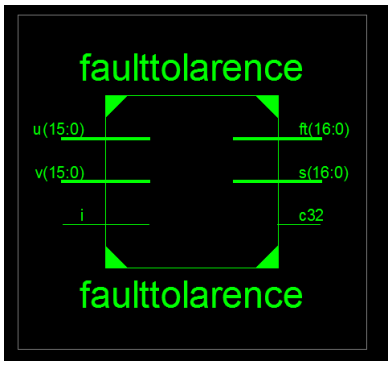


FIG. 3 RTL SCHEMATIC

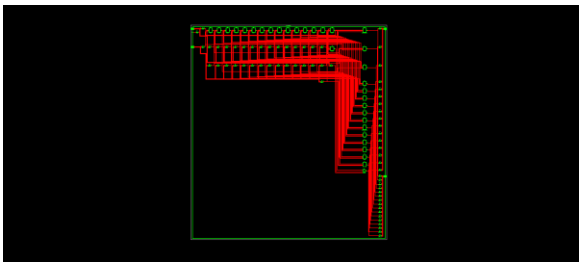


FIG. 4 TECHNOLOGY SCHEMATIC

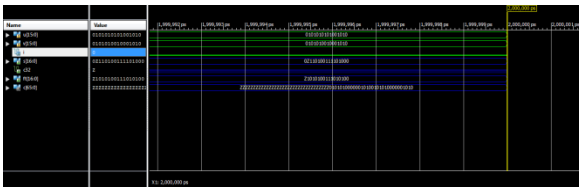


FIG. 5 OUTPUT WAVEFORM

V. 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 outputs to detect and correct errors. The scheme can be used for parallel filters that have the same response and process different input signals. The technique provides larger benefits when the number of parallel filters is large. The proposed architecture can be easily used to implement high order FIR filters e.g. 20-tap with different coefficients wordlength without suffering from large architecture reconstruction and with low hardware complexity needed for the design. The proposed scheme can also be applied to the IIR filters. The future work is to perform a VLSI implementation of pulse

shaping FIR filter for ultra wideband communications.

VI. REFERENCES

1) M. I. Soliman, G. Y. Abozaid, "FPGA implementation and performance evaluation of a high throughput crypto coprocessor," Journal of Parallel and Distributed Computing, Vol. 71 (8), pp.1075-1084, Aug. 2011.

2) V. K. Pachghare, Cryptography and information security, E. E. Ed., PHI Learning, New Delhi, 2009.

3) M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 19

4) X. Zhang, K. K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12 (9), pp. 957-967, Sep. 2004.

5) M. Jridi and A. AlFalou, "A VLSI implementation of a new simultaneous images compression and encryption method," 2010 IEEE International Conference on Imaging Systems and Techniques (IST), pp.75-79, July 2010.

6) Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, and Cheng-Wen Wu, "A High- Throughput Low-Cost AES Processor," IEEE Communications Magazine, Vol.41 (12), pp.86-91, Dec. 2003.

7) L. Ali, I. Aris, F. S. Hossain and N. Roy, "Design of an ultra-high speed AES processor for next generation IT security," Computers and Electrical Engineering, Vol.37 (6), pp.1160-1170, Nov. 2011.

8) K.H. Chang, Y.C. Chen, C. C. Hsieh, C. W. Huang and C. J. Chang, "Embedded a Low Area 32-bit AES for Image Encryption/Decryption Application," IEEE International Symposium on Circuits and Systems, pp. 1922-1925, May 2009.

9) J. M. G. Criado, M. A. V. Rodriguez, J. M. S. Perez, J. A. G. Pulido, "A new methodology to implement the AES algorithm using partial and dynamic reconfiguration," Integration, the VLSI Journal, Vol.43(1), pp. 72-80, Jan. 2010.

10) J. V. Dyken, J. G. Delgado-Frias, "FPGA schemes for minimizing the powerthroughput trade-off in executing the Advanced Encryption Standard algorithm," Journal of Systems Architecture, Vol.56(2–3), pp. 116-123, Mar. 2010.

11) I. Hammad, K. E. Sankary and E. E. Masry, "High-Speed AES EncryptorWith Efficient Merging Techniques," IEEE Embedded Systems Letters, Vol.2 (3), pp.6771, Sept. 2010.

12) N. Ahmad, R. Hasan, W. M. Jubadi, "Design of AES S-Box using combinational logic optimization," IEEE Symposium on Industrial Electronics & Applications, pp.696-699, Oct. 2010.

13) N. Ahmad, S. M. R. Hasan, "Low-power compact composite field AES S-Box/Inv S- Box design in 65 nm CMOS using Novel XOR Gate," Integration, the VLSI Journal, Article in Press.

14)Edwin NC Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic," Custom R&D Engineer Taxco Enterprise Pvt. Ltd.

15) Jarvinen, K., Tommiska, M., and Skytta, "A Fully Pipelined Memoryless 17.8 Gbps AES-128 Encryptor". Proc. ACM/SIGDA 11th ACM Int. Symposium on FieldProgrammable Gate Arrays, FPGA 2003, Monterey, CA, USA, February 2003, pp. 207–215.

16) KimmoJärvinen, MattiTommiska and JormaSkyttä, "Comparative Survey of High Performance Cryptographic Algorithm Implementations on FPGAs", IEE Proceedings - Information Security, vol. 152, no. 1, Oct. 2005, pp. 3-12.

17) A. Rudra, P.K. Dubey, C.S. Jutla, V. Kumar, J.R. Rao, and P. Rohatgi. "Efficient Rijndael Encryption Implementation with Composite Field Arithmetic", Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), pages 175–188, May 2001.

18) Tim Good and Mohammed Benaissa, "Very Small FPGA Application-Specific Instruction Processor for AES", IEEE Transactions on Circuit and Systems-I, Vol. 53, No. 7, July 2006.

19) Data Encryption Standard (DES), FIPS PUB (46-3), Oct. 25, 1999, Federal Information Processing Standard 46-3