International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 05, May 2016
www.ijates.com
ijates
ISSN 2348 - 7550

AREA EFFICIENT AND FAULT TOLERANT PARALLEL FIR FILTER BASED ON ECC

M. Anogeeth[1] , Mr. Chenthil[2]

[1] PG scholar, [2]Assistant Professor, Department of ECE,
Sasurie Academy of Engineering, Coimbatore, (India)

ABSTRACT

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. Many techniques exploit the filters structure and properties to achieve fault tolerance have been proposed. As technology scales, it enables more complex systems that incorporate on 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 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 area.

Keywords : DMC (Decimal Matrix Code) , Fast Carry Chain Adder, Parallel Filters.

I. 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. 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 filters.

The increasing development in VLSI technology ,the real time realization of FIR filters with less latency has become more important .The complexity of filter grows with the length of the filter ,several algorithms have been made to develop architectures for realization of FIR filter in ASIC and FPGA platforms and one of them is error robust design .The main portion of error tolerant –based FIR computation is parity blocks that stores the pre-computed values and can be read out easily which makes FIR computation well suited for FPGA realization .This technology represents a number of attractive features such as simplicity ,regularity and modularity.

1.1 Fault Tolerance

The property that enables a system to continue operating properly in event of failure of (one or more faults) within some of its components. There are different fault tolerance approaches to conventional circuits and DSP circuits .In some cases , the reliability of those systems is critical and fault tolerant filter implementation are needed. In the techniques implemented so far, the protection of single filter is considered. In recent times

researcher has projected, a simple technique having the existence of parallel filter to accomplish fault tolerance[3].It has been proposed to protect digital circuits in signal processing by single error detecting and error correcting codes[7].Filters can be protected using ECCs in which each filter is equivalent to a bit in traditional ECC and filters are arranged in parallel architecture.

1.2 Parallel Processing

A technique duplicating functional units to operate different task simultaneously i.e ,we can perform same processing for different signals.

1.2.1.Fir Filters

A Finite Impulse Response (FIR) digital filter is one whose impulse response is of finite duration. The general difference equation for a FIR digital filter is

$$y(n) = \Sigma x(n-1).\ H(l)$$

Where x(n) is the Input Signal

1.3 HAMMING CODE

A simple ECC takes a block of k bits and reduces a block of n bits by adding $n-k$ parity check bits. Hamming Code (7,4) is a linear error-correcting code that encodes four bits of data into seven bits by adding three parity bits

II. EXISTING WORK

2.1 Triple Modular Redundancy

Digital filters are one of the most commonly used signal processing circuits and several techniques have been proposed to protect them from errors .Filters used to detect and reduces bit errors.

Triple modular redundancy is a fault tolerant form in which three systems perform a process and result processed by majority voting system to produce a single output.If any one of the system fails it does not affect the whole system.

TMR triplicates the design and adds voting logic to correct errors.It triples the area and power of circuit, which may not be acceptable in many applications.

2.1.2 Drawbacks

- More area overhead system
- More power consumption
- Low speed architecture
- Circuits are more complex to design

2.2 Hamming Code Based Error Correction Codes

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 [1]. 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.

As an example, let us consider a simple Hamming code [11] with $k = 4$ and $n = 7$. In this case, the three parity check bits $p1$, $p2$, $p3$ are computed as a function of the data bits $d1$, $d2$, $d3$, $d4$ as follows:

$$p1 = d1 + d2 + d3$$

$$p2 = d1 + d2 + d4$$

$$p3 = d1 + d3 + d4.$$

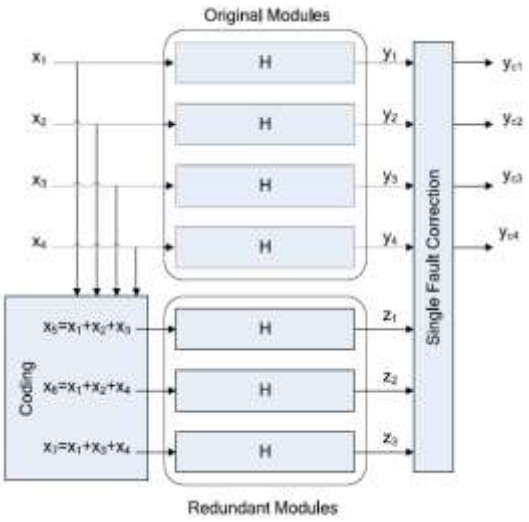

Fig 1 Existing Scheme For Four Filters And A Hamming Code

III. PROPOSED METHOD

3.1 Parallel Filters With Same Response

A discrete time filter implements the following equation:

$$y[n] = \sum_{l=0}^{\infty} x[n-l] \cdot h[l] \qquad (1)$$

where $x[n]$ is the input signal, $y[n]$ is the output, and $h[l]$ is the impulse response of the filter. When the response $h[l]$ is nonzero, only for a finite number of samples, the filter is known as a FIR filter, otherwise the filter is an infinite impulse response (IIR) filter. There are several structures to implement both FIR and IIR filters. In the following, a set of k parallel filters with the same response and different input signals are considered.

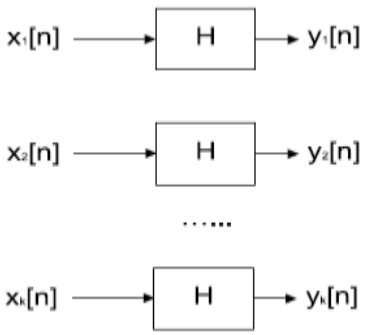

Fig 2 Parallel Filters

These parallel filters are illustrated in Fig 2. This kind of filter is found in some communication systems that use several channels in parallel. In data acquisition and processing applications is also common to filter several signals with the same response. An interesting property for these parallel filters is that the sum of any combination of the outputs *yi* [*n*] can also be obtained by adding the corresponding inputs *xi* [*n*] and filtering the resulting signal with the same filter *h*[*l*].

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

3.2 Fast Carry Chain Adder

Ripple-carry adders are implemented within a Virtex-5 FPGA device exploiting the generic configurable resources and the dedicated carry chain structures. The latter speed -up the critical propagation path and allow high-speed to be reached[12]. Xilinx Virtex-5 devices realize fast carry chains by using dedicated multiplexers (MUXCY) and routing resources that can be used to implement an *n*-bit efficient addition , where *A* and *B* are the addends and C_0 is the carry-in. The generic sum bits S_i ($0 \le i < n$) is calculated, as reported in (1), by XOR-ing the bit propagate $p_i = A_i \oplus B_i$ with the carry bit C_i computed as

$$Si=(Ai \oplus Bi) \oplus Ci = pi \oplus Ci \quad (2)$$

$$Ci=(Ai\text{-}1 \cdot Bi\text{-}1)+(Ai\text{-}1 \oplus Bi\text{-}1) \cdot Ci\text{-}1$$
$$= not(pi\text{-}1) \cdot Ai\text{-}1 + pi\text{-}1 \cdot Ci\text{-}1 \quad (3)$$

The advantage of this adder structure is evident when the delay of the dedicated carry chain is compared to the generic LUT fabric. For example, in the Virtex-5 family, the single bit sum generation stage (horizontal signal propagation through the LUT and the XORCY) is about 17 times slower than the single bit carry propagation stage (vertical propagation along the generic MUXCY). It is then clear that any attempt to improve performances of adder architectures using generic logic and routing resources would be ineffective.

3.3 Decimal Matrix Code

In the proposed DMC, first, the divide-symbol and arrange-matrix ideas are performed, i.e., the *N* -bit word is divided into *k* symbols of *m* bits ($N = k \times m$), and these symbols are arranged in a $k_1 \times k_2$ 2-D matrix ($k = k_1 \times k_2$, where the values of k_1 and $k2$ represent the numbers of rows and columns in the logical matrix ${}_4H$ are produced by performing decimal integer addition of selected symbols per row. Here, each symbol is regarded as a decimal integer. It should be noted that divide-symbol and arrange-matrix are implemented in logical instead of in physical.

To explain the proposed DMC scheme, we take a 32-bit word. The cells from D_0 to D_{31} are information bits. This 32-bit word has been divided into eight symbols of 4-bit. $k_1 = 2$ and $k_2 = 4$ havbeen chosen simultaneously. H_0–H_{19} are horizontal check bits; The horizontal redundant bits *H* can be obtained by decimal integer addition as

follows:

$H_4\ H_3\ H_2\ H_1\ H_0 = D_3\ D_2\ D_1\ D_0 + D_{11}\ D_{10}\ D_9\ D_8$

$H_9\ H_8\ H_7\ H_6\ H_5 = D_7\ D_6\ D_5\ D_4 + D_{15}\ D_{14} D_{13}\ D_{12}$

The input is given to the filter block by using fast carry chain adder and a multiplier in the encoding section. In the filter block the inputs are $a_0(3:0) - a_4(3:0)$,clock ,reset, x(3:0).The output is obtained from the filter block and is fed to decoding and comparator block. The decoding process is carried out and the output is 32 bits .The inputs and encoded values are compared and there is no error.

3.4 Advantages

- Easy to predict the error occurred bit
- Easy to built a checker matrix
- Easy to implement

IV. SIMULATION RESULTS

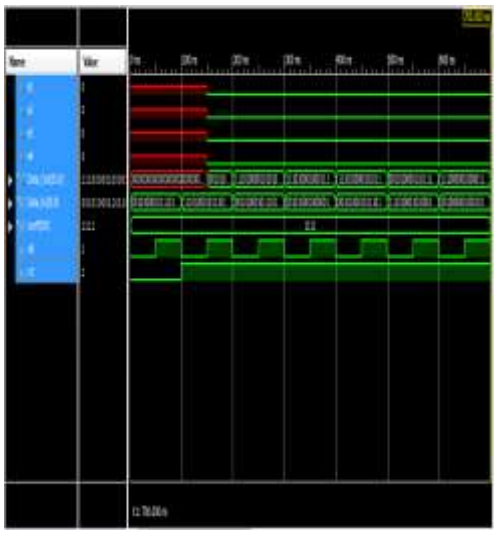

Fig 3 Simulation Result of FIR Filter

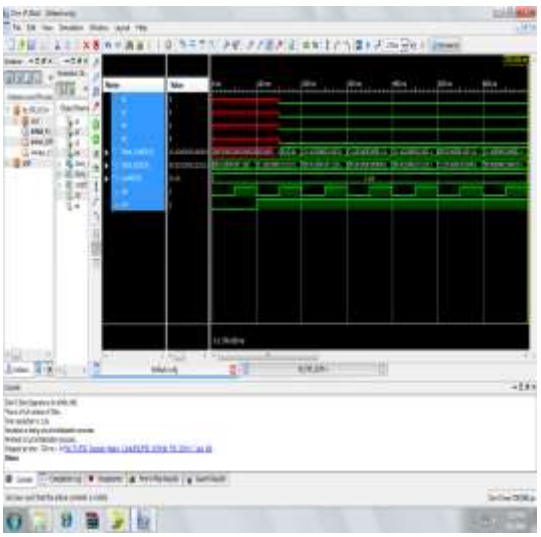

Fig 4 Simulation Result of DMC

4.1 Area Utilisation

Selected Device : 7v585tlffg1761-2l

Slice Logic Utilization:

Number of Slice Registers: 80 out of 728400 0%

Number of Slice LUTs: 56 out of 364200 0% Number used as Logic: 56 out of 364200 0%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 136

Number with an unused Flip Flop: 56 out of 136 Number with an unused LUT: 80 out of 136

Number of fully used LUT-FF pairs: 0 out of 136

Number of unique control sets: 1

IO Utilization:

Number of IOs: 50

Number of bonded IOBs: 50 out of 850 5%

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 1out of 32 3%

Number of DSP48E1s 25 out of 1260 1%

V. CONCLUSION AND FUTURE WORK

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 output 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 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 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.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their detailed comments and suggestions which helped to improve the quality of the paper.

REFERENCES

[1] Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao, Jing Wang, and Juan Antonio Maestro , " Fault Tolerant Parallel Filters Based on Error Correction Codes" ieee transactions on very large scale integration (vlsi) systems, vol. 23, no. 2, february 2015.

[2] T. Hitana and A. K. Deb, "Bridging Concurrent And Non-Concurrent Error Detection In Fir Filters," in proc. norchip conf., 2004, pp. 75–78.

[3] M. Nicolaidis, "Design For Soft Error Mitigation," ieee trans. device mater. rel., vol. 5, no. 3, pp. 405–418, sep. 2005.

[4] 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.

[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] 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

[7] A. Sibille, C. Oestges, and A. Zanella, Mimo: From Theory To Implementation. san francisco, ca, usa: academic press, 2010.

[8] 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.

[9] 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.

[10] p. p. vaidyanathan. Multirate Systems And Filter Banks. upper saddle river, nj, usa: prentice-hall, 2013.

[11] A Fast Carry Chain Adder For Virtex 5 Fpgas paolo zicari 1 and stefania perri2 Department Of Electornics, Computer Science And Systems University Of Calabria Italy