Implementation of Fault Tolerant FIR Filter for Digital Communication Systems

Jyotishma Bharti
Department of Electronic and Communication Engineering
Integral University, Lucknow

Tarana Afrin Chandel
Associate Professor
Department of Electronic and Communication Engineering
Integral University, Lucknow

Abstract

A realistic communication system is not free from noise. So the transmission of information through may rather be corrupted by noise in the channel. Therefore it is necessary for every communication systems to have suitable means to recognize and correct those errors in the information received over communication channels. There are various types of filters are used by Digital signal processing (DSP) applications. In which digital parallel FIR filters are very widely used in numerous application. Over the years, many implementation techniques of digital FIR filter for DSP application has exploit the various practical difficulties such as low speed, high delay and above of all fault tolerance. Due to the VLSI complexity scaling, there are many complex systems that embed with many filters. The filters operations in those complex systems are usually parallel. As filters is the unit that comes in any type of communication system ranging from simple voice data to complex real time data conversation. So it is then mandatory to implement some technique that shows the fault tolerance achieved in parallel filters. In this paper we are implementing the FIR Filter with 6-bit Fault tolerant using BCH codes. The complete design has been developed by VHDL and synthesize and simulated by XILINX ISE Tool.

Keywords- Error Correction Codes (ECC), Digital Signal Processing (DSP), Finite Impulse Response (FIR) Parallel FIR, Very Large Scale Integration (VLSI)

I. INTRODUCTION

The demand for high performance and low power DSP is getting exponentially higher due to the bombardment of multimedia application such as automotive, medical and space applications where reliability is critical. And in those specific applications, the electronic circuits should have to provide some degree of fault tolerance. Although there are various other techniques that can be used to protect a circuit from errors. These errors can be removed ranging from modifications in the manufacturing process of the circuits for reducing the number of errors by adding redundancy bits at the logic or system level in order to ensure that these errors can do not affect the system functionality.

As in this paper we are more focusing on Filter processing so it is mandatory to emphasize on the application area of Filters. The filters are basic unit of any type of communication. In the basic communication system the transmission of information/raw data is not always be free from unwanted information bits or Noise. Thus it is very necessary for any communication systems to must have appropriate means for the finding and correction of errors in the information received over any communication channels. This paper deals with the study of the use of the various Error corrections coding scheme into Digital FIR Filter design.

II. ERROR CORRECTING CODES – AN OVERVIEW

During the Digital communication system the information always travels through a medium. And the transformation of the information through this channel could always not be free from noise. Also we know that in digitally encoded data, there is a series of symbols denoted generally in the terms of 0s and 1s. Suppose that we want to transmit the information that "There is no class on Monday". This information can be defined by 01101111, say, and transmitted over a channel where some unwanted data i.e. "noise" may be introduced. Noise means simply errors. The aim of an error-correcting code is to lengthen the message in such a way that the original message can be recovered even if errors are present here. By error we mean that there is unwanted change in data which is required to be correct for efficient and reliable reception of data.
The following diagram represents the communications channel.

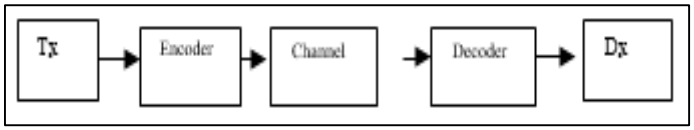


Fig. 1: Basic Communication System

An error is the change or the mismatching takes place between the data sent by transmitter and the data received by the receiver e.g. 10101010 sent by sender 10101011 received by receiver. Here is an error of 1 bit. Error control refers to mechanisms to detect and correct errors that occur in the transmission of frame. There are assorted techniques available for error control e.g. Error detection, Positive acknowledgement Retransmission after time-out, Negative acknowledgement and retransmission. These mechanisms are also referred as Automatic Repeat Request (ARQ). The base of all error detection and correction is the inclusion of redundant information.

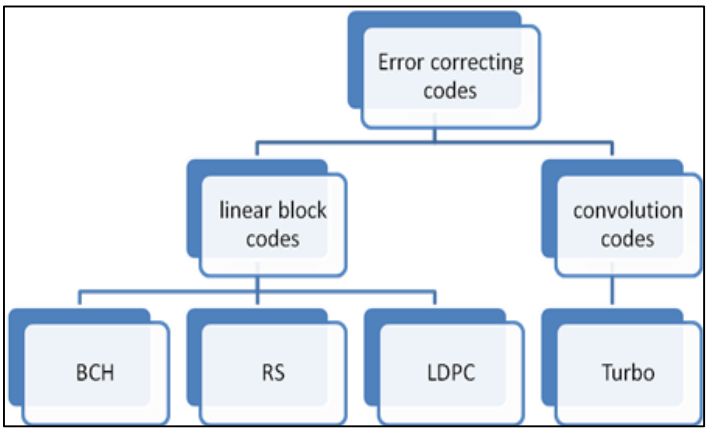


Fig. 2: Classification of Error Correcting Codes

III. Proposed Block Diagram

In this work we are generating a Self-error detector and corrector for FIR Filter. We are integrating two different topologies into a one solution. The proposed block diagram is shown in figure 3.

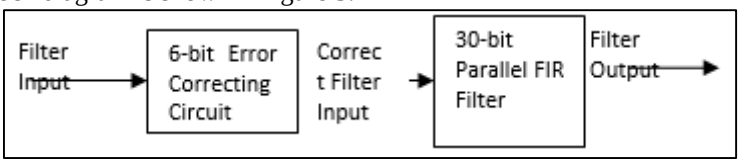


Fig. 3: Proposed Block Diagram

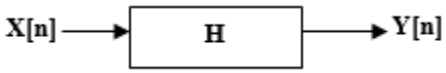
The figure 3 shows the proposed schematic for our research work. The filter input first propagates through the error detection and correction circuit. Then the output from the error correcting detecting circuit will finally go to the parallel FIR filter. The basics of these FIR filter and Error detecting correcting circuit will be discussed in coming sections.

IV. Parallel Filter

A discrete time filter follows the following equation:

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

As stated in equation 1, x[n] is the input signal, y[n] is the output, and h[l] is the impulse response of the filter. The FILTERS has been categorized by this impulse response. For a FIR FILTER the response h[l] should be nonzero, only for a finite number of samples. Otherwise the filter is an infinite impulse response (IIR) filter. There are several structures to implement both FIR and IIR filters.



We can also cascade n number of filters in a single communication systems.

V. Error Correction Codes

In our work we had been gone through various Error detection and correction circuits. But we found BCH codes is the best suited for our purpose. The some property of BCH codes is going to be discussed in this section.
The BCH codes use to be defined by the code size *n* and the number of errors to be corrected *t.*
Block length: n = 2m - 1
Number of information bits: k ≥ n-m*t
Minimum distance: dmin≥2t + 1.

The generator polynomial of the code is specified in terms of its roots over the Galois field GF (2^m). Let's take α as a primitive element in GF (2^m).The generator polynomial g(x) of the code is the lowest degree polynomial over GF (2). Let $m_i(x)$ be the minimum polynomials of α_i then generator polynomial G(x) can be computed by:

$$G(x) = LCM [m_1(x), m_3(x) \ldots m_{2t}(x)] \quad (2)$$

In this work n=63, k=30 and t=6 is considered. Hence the generator Polynomial with, a, a2... a4 as the roots is obtained by multiplying the following minimal polynomials:

$m_1(x) = 1+x+x^4$

$m_3(x) = 1+x+x^2+x^3+x^4$

.

.

.

Substituting m1(x) and m3(x) in equation (2) generator polynomial is obtained.

$G(x) = LCM \{m_1(x), m_3(x)\}$

$G(x) = \{(1+x+x^4)(1+x+x^2+x^3+x^4)\}$

$$G(x) = 1+x^4+x^6+x^7+x^8 \quad (3)$$

To build BCH codes over GF (2^6), we need to find out the elements of GF (2^6) generated by $p(x) = 1+x+x^6$ is given in Table below

GALOIS FIELD GF(2^6) WITH $\rho(\alpha) = 1 + \alpha + \alpha^6 = 0$

0	0	(0 0 0 0 0 0)	α^{15}	$\alpha^3 + \alpha^5$	(0 0 0 1 0 1)
1	1	(1 0 0 0 0 0)	α^{16}	$1+\alpha + \alpha^4$	(1 1 0 0 1 0)
α	α		α^{17}	$\alpha+\alpha^2 + \alpha^5$	(0 1 1 0 0 1)
α^2	α^2	(0 1 0 0 0 0)	α^{18}	$1+\alpha+\alpha^2+\alpha^3$	(1 1 1 1 0 0)
α^3	α^3	(0 0 1 0 0 0)	α^{19}	$\alpha+\alpha^2+\alpha^3+\alpha^4$	(0 1 1 1 1 0)
α^4	α^4	(0 0 0 1 0 0)	α^{20}	$\alpha^2+\alpha^3+\alpha^4+\alpha^5$	(0 0 1 1 1 1)
α^5	α^5	(0 0 0 0 0 1)	α^{21}	$1+\alpha + \alpha^3+\alpha^4+\alpha^5$	(1 1 0 1 1 1)
α^6	$1+\alpha$	(1 1 0 0 0 0)	α^{22}	$1 + \alpha^2 + \alpha^4+\alpha^5$	(1 0 1 0 1 1)
α^7	$\alpha+\alpha^2$	(0 1 1 0 0 0)	α^{23}	$1 + \alpha^3 + \alpha^5$	(1 0 0 1 0 1)
α^8	$\alpha^2+\alpha^3$	(0 0 1 1 0 0)	α^{24}	$1 + \alpha^4$	(1 0 0 0 1 0)
α^9	$\alpha^3+\alpha^4$	(0 0 0 1 1 0)	α^{25}	$\alpha + \alpha^5$	(0 1 0 0 0 1)
α^{10}	$\alpha^4+\alpha^5$	(0 0 0 0 1 1)	α^{26}	$1+\alpha+\alpha^2$	(1 1 1 0 0 0)
α^{11}	$1+\alpha$	(1 1 0 0 0 1)	α^{27}	$\alpha+\alpha^2+\alpha^3$	(0 1 1 1 0 0)
α^{12}	$1 + \alpha^2$	(1 0 1 0 0 0)	α^{28}	$\alpha^2+\alpha^3+\alpha^4$	(0 0 1 1 1 0)
α^{13}	$\alpha + \alpha^3$	(0 1 0 1 0 0)	α^{29}	$\alpha^3+\alpha^4 + \alpha^5$	(0 0 0 1 1 1)
α^{14}	$\alpha^2 + \alpha^4$	(0 0 1 0 1 0)	α^{30}	$1+\alpha + \alpha^4+\alpha^5$	(1 1 0 0 1 1)

Table 1: The elements of GF (2^6) generated by $p(x) = 1+x+x^6$

In BCH, the code words are formed by adding the remainder after division of message polynomial with generator polynomial. All code words are the multiples of generator polynomial. At the encoding side, the generator polynomials are not usually split as it will demand more hardware and control circuitry. The polynomial is used as such for encoding. For BCH circuit there are two blocks has been used named As Encoder and Decoder. Encoder always starts with Parallel to serial register with 33 bit LFSR.

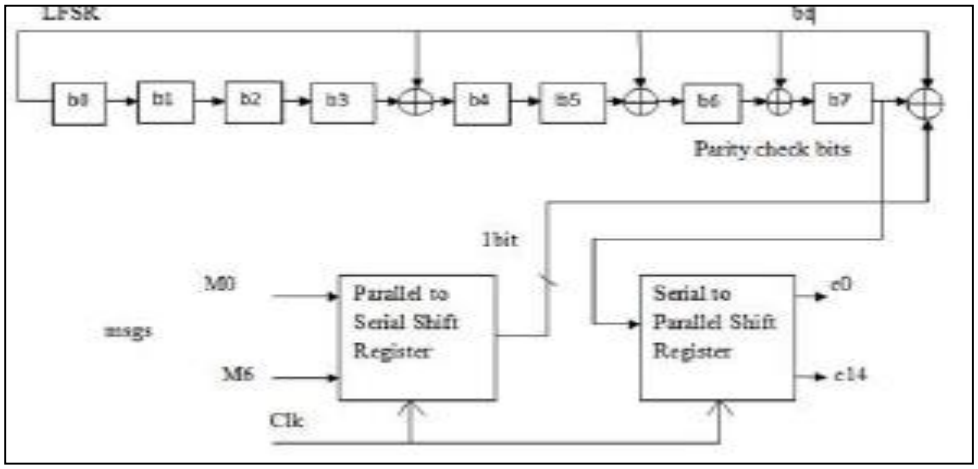


Fig. 4: BCH Encoder

The (63,30) BCH Encoder is always implemented with a Linear Feedback Shift Register (LFSR). BCH code word are encoded as follows.

$$C(x) = x^{n-k} * M(x) + b(x) \quad (4)$$

Where, Message bits $M(x) = M_0 + M_1x + ... + M_{k-1}x^{k-1}$
Code word $C(x)$ is $c_0 + c_1x + ... + c_{n-1}x^{n-1}$
Remainder is $b(x) = b_0 + b_1x + ... + b_{m-1}x^{m-1}$ where c_i, b_i are the subsets of Galois field. Figure 4 shows block diagram of (63,30) BCH Encoder module. The 30 message bits (M0, M1….M29) are applied to the parallel to serial shift register. Using these message bits, parity bits are computed and then sent to serial to parallel shift register. These parity bits are appended to original message bits to obtain 63 bit encoded data. This entire encoding process requires 63 clock cycles.

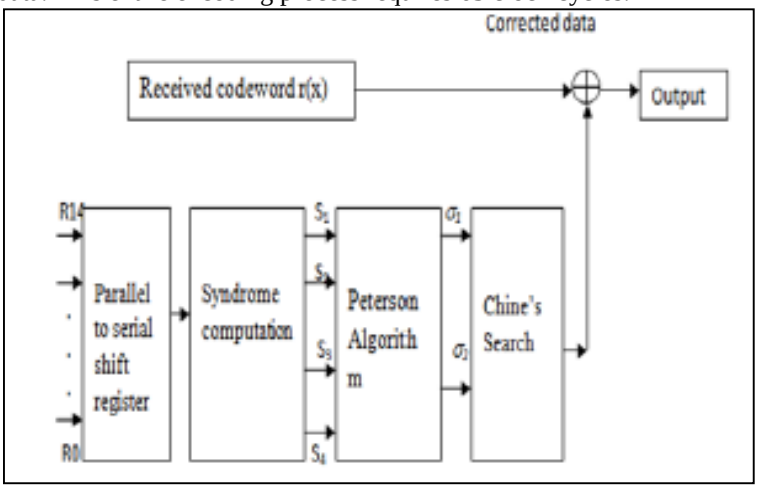


Fig. 5: BCH Decoder

Figure 5 shows the block diagram of BCH decoder. The decoding algorithm for BCH codes consists of three major steps.

- Calculate the syndrome value Si , i=1,2,….,2t from the received word r(x).
- Determine the error location polynomial s*(x)*
- Find the roots of s*(x)*and then correct the errors

At the decoder side the received 63 bit data will propagates through the parallel to serial shift register, then obtained serial output will be used as a input to compute syndrome s(x). For error free transmission s(x) = 0. Otherwise, transmitted message will be in error. This is called error detection process. There are several types of error correction process i.e. Peterson and Zierler algorithm, Chine's search algorithm.

Peterson's algorithm accepts syndrome s(x) as a input and computes error locator polynomial σ(x). For finding the error locator polynomial using the formula σ1= S1, σ2= $(S_3+S_1^3)*(S_1-1)$. This polynomial can be further used to find the location of the errors. Using chine's search algorithm error location is determined. This process includes searching the unique roots of the error locator polynomial. The input for the chine's search is σ(x) and it returns roots of error locator polynomial which corresponds to the error positions.

VI. Synthesis & Simulation Results

This section shows the various results obtained from the XILINX ISE Tool. The figure 6 & 7 shows the RTL view of our designed circuit i.e. Fault tolerant FIR Filter. As shown in the figure there are two inputs named as A, and error. The input A denotes the original message to be filter through the FIR filtration. While error signal are for stimulating the various number of errors.

Fig. 6: RTL for Fault Tolerant FIR

The figure 6 shows only the input output ports which can be used as a physical interface for real world implementation. On the other hand figure 7 shows the internal structure of the Fault tolerant FIR filter. In this two separate block has been connected together to achieve the desired output.

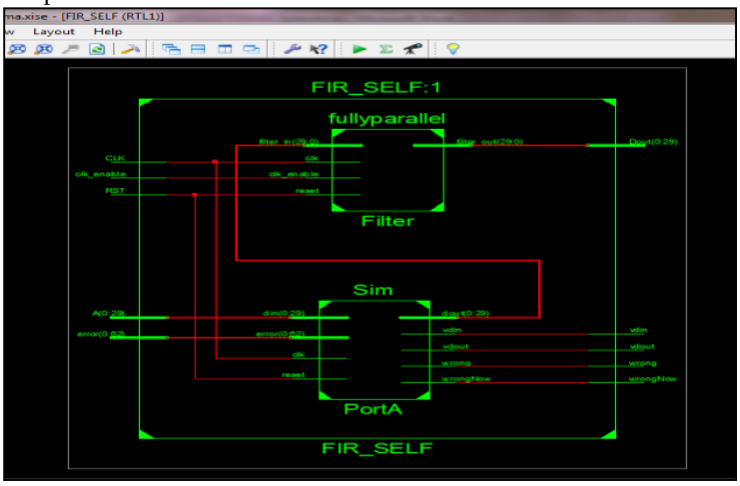


Fig. 7: Internal RTL

The figures 8, 9 & 10 shows the cumulative result for the fault tolerant fir filter with different input conditions.

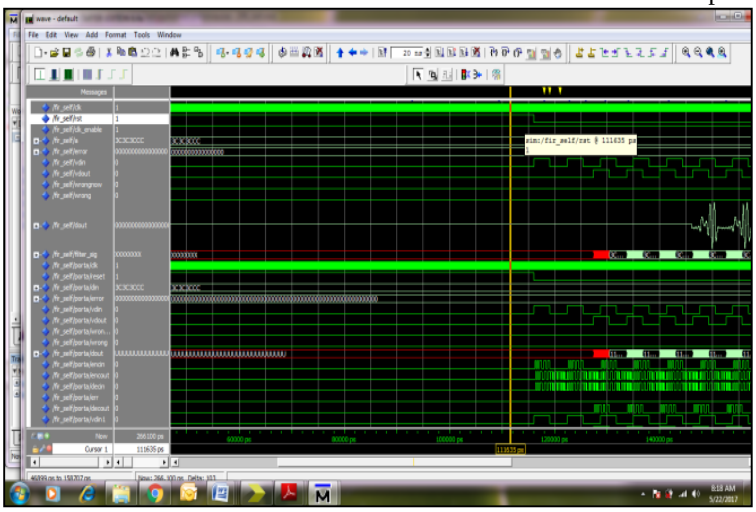


Fig. 8: Simulation Result with initial Conditions

The figure 8 shows the simulation results with initial conditions i.e. RST='1'. For initialization of all the registers, signals into its original states/values.

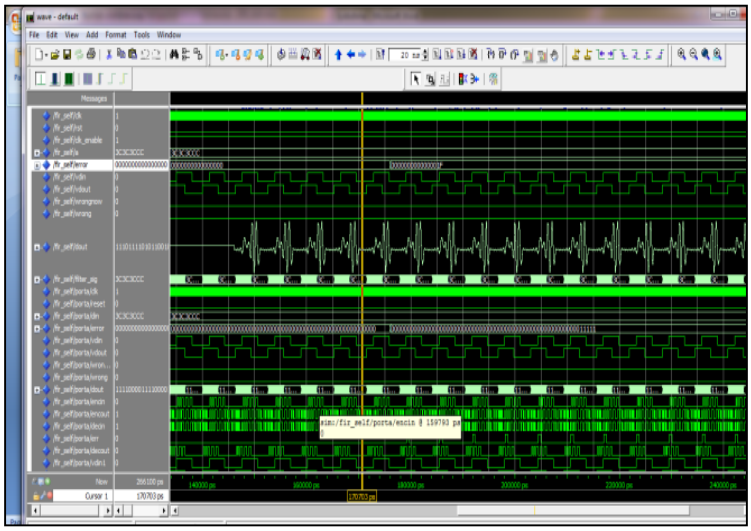


Fig. 9: Simulation results with 4 bit error correction

The figure 9 & 10 shows the result with stimulation of 4 bit and 6 bit error values, respectively, through error signal. And we can easily observe that there is not a slight change in the output waveform.

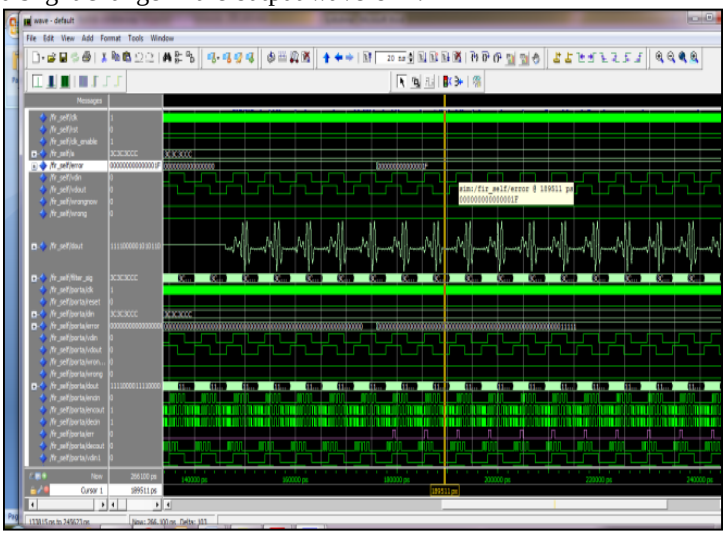


Fig. 10: Simulation results with 6 bit error correction

Fig. 11: Simulation results with >6 bit error

The figure 11 shows the simulation result with more than 6 bit error stimulation through error signal. And also there is some changes happened in the waveform. In other words we can see there are some extra lobes had been developed.

VII. CONCLUSION

We had successfully implemented the Fault tolerant FIR Filter. Our design has provide the adequate result with less than and equal to 6. Means our fir filter can retain their original input signals if by any means there has been error generated. The complete design has been designed and implemented by Xilinx ISE tool. The Synthesis has been done by XILINX Synthesis Tool and the simulation has been carried out by Xilinx ISIM and Modelsim.

REFERENCES

[1] P. P. Vaidyanathan, Multirate Systems and Filter Banks, Englewood Cliffs, N.J., USA:Prentice Hall, 1993.
[2] A. Sibille, C. Oestges and A. Zanella, MIMO: From Theory to Implementation, New York, NY, USA: Academic, 2010.
[3] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and ElectroMagnetic Disturbances, New York, NY, USA: Springer Verlag, 2010.
[4] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.
[5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop.vol. 28, no. 2, pp. 124–134, Mar. 1984.

[6] 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.
[7] T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in Proc. Norchip Conf., 2004, pp. 75–78.
[8] Y.-H. Huang, "High-efficiency soft-error tolerant digital signal processing using finegrain sub word-detection processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 291–304, Feb. 2010.
[9] Yu-Chi Tsao and Ken Choi, 2012. "Area- Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm" IEEE transactions on very large scale integration (vlsi) systems, vol. 20, no. 2, feb. 2012.
[10] Yu-Chi Tsao and Ken Choi, 2011. "Hardware- Efficient Parallel FIR Digital Filter Structures For Symmetric Convolutions "978-1-4244-9474- 3/11/$26.00 ©2011 IEEE.
[11] Lavina Magdalene Mary "Area Efficient Parallel Fir Digital Filter Structures Based On Fast Fir Algorithm" Vol. 3, Issue 1, January –February 2013, pp.2042-2046
[12] D. A. Parker and K. K. Parhi, 1997. "Lowarea/ power parallel FIR digital filter implementations," J. VLSI Signal Process.Syst., vol. 17, no. 1, pp. 75–92, 1997.
[13] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley, 1999.
[14] J. G. Chung and K. K. Parhi, 2002. "Frequency spectrum- based low-area low-power parallel FIR filter design," EURASIP J. Appl. Signal Process., vol. 2002, no. 9, pp. 444–453, 2002.
[15] Alfredo Rosado-Muñoz and Manuel Bataller- Mompeán,"FPGA Implementation of an Adaptive Filter Robust to Impulsive Noise: Two Approaches,"vol.58, no.3, Mar.2011.
[16] Z.-J. Mou and P. Duhamel, 1991. "Short-length FIR filters and their use in fast non recursive filtering," IEEE Trans. Signal Process., vol. 39, no.6, pp. 1322–1332, Jun. 1991.
[17] C. Cheng and K. K. Parhi, 2004. "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," IEEE Trans.