ISSN 2320-9984

Volume 5 Issue 2 | June 2017

ISSN: 2320-9984 (Online)

International Journal of Modern Engineering & Management Research

Website: www.ijmemr.org

A Review on Various Techniques for FIR Filter Design

Soniya Gogya
M.Tech. Scholar
Takshshila Institute of Engineering & Technology
Jabalpur, (M.P.) India
Email:soniya.gogya25@gmail.com

Prof. Mohit Khilwani
Assistant Processor
Department of Electronics and Communication Engg.
Takshshila Institute of Engineering & Technology
Jabalpur (M.P.) [INDIA]
Email:mohitkhilwani@takshshila.org

Abstract—*Since a decade Digital Signal Processing has gained in popularity in the area for communication. Popularity for digital signal processing has risen is due to the lots for advantages it possesses, such as high noise immunity, low cost, high speed, and flexibility. In recent times, lots for innovative methods are being developed to make communication devices as small and portable as possible. Hence significant hardware implementation plays a key role in making communication devices more portable. FIR filters are the basic components used to manufacture communication devices. The hardware implementation for the RNS filter can be done using Look-Up-Table (LUT) in the block RAM's in FPGA's for the Xilinx Virtex FPGA's. The RAMs generated by the core generator are used for this purpose. The implementation was done on an example filter, to compare it with other implementations.*

1. INTRODUCTION

Signal processing can be broadly classified into two types Analog and Digital Signal Processing.

In recent years, Digital Signal Processing (DSP) has gained a lot for popularity. DSP plays a very significant role in today's world. Some for the applications for DSP are in the areas for communication, medicine, and entertainment. In the early stages, the use for digital signals in communication devices was expensive. It has only after the invention for the microprocessor and the development for Integrated Circuits that the utilization for digital signal processing in communication was practical and feasible.

The wide growth in the utilization for DSP was due to the lots for advantages it possesses like high noise immunity, low cost, high speed, and flexibility. The additional advantages for using equipment with digital signal processes are stability and consistency. Digital filters are required in most communication equipment. To design the filter to meet the required specifications, lots for methods like Butterworth filter design[l], Chebyshev filter design [1], or Parks-McClellan filter design [1] can be used based on the type for filter required. To reduce the hardware utilization for implementing the filter, lots for methods have been developed for the representation for the filter coefficients.

2. BACKGROUND

2.1 Classification for Digital Filters:

Digital filters are characterized by their impulse response, their transfer function or by difference equations. Digital Filters can be classified into two groups based on the type for impulse response they have, infinite impulse response (IIR) or finite impulse response (FIR). The impulse response for a filter is the response for the filter when the input signal is an impulse signal.

2.1.1 Finite Impulse Response

Filters (FIR Filter): Digital Filters [2] with finite impulse response are called Finite Impulse Response Filters[3]. FIR filters are very broadly used in communication equipment. The broad usage for FIR filters in communication is due to linear phase FIR filters are easily determined. By making the filter coefficients symmetric, linear phase can be determined in FIR filters, i.e. the first and the last coefficients are the same and so forth. Linear phase filters are very important, especially in devices, which handle signals carrying information in the phase. Nonlinear phase distortion can cause the information to be lost, making the signal useless. When FIR filters with symmetric coefficients are used to implement the filter, a significant savings in hardware is also determined, as only half for the filter coefficients have to be implemented.

A characteristic for FIR filter is that the impulse response for a FIR filter is the same as the filter coefficients. FIR filters do not have poles, they only have zeros. Hence the response for a FIR filter is only dependent on current and previous inputs and not on the output for the filter. Since a finite number for bits must be used to represent the input, output and the coefficients for any digital filter, FIR filters can be designed with sufficient word length to guarantee that no rounding or truncation will be done in the multiplication for the filter input by the coefficients. This should be compared to an IIR filter, where we must always do rounding when the output is multiplied by filter coefficients in order to prevent the word length from growing without bound. It is the fact that only the input is multiplied by the filter coefficients in an FIR filter that allows us to design FIR digital filters without error in the arithmetic operations. This also makes the application for RNS arithmetic particularly attractive for FIR filters. The time response representation for FIR filter also called the

difference equation and is given by

$$y_k = \sum_{i=0}^{n-1} a_i x_{k-1} \qquad \text{eq1.1}$$

$$H(z)= \frac{Y(z)}{X(z)} = \sum_{i=0}^{n-1} a_i x_{k-1} \qquad \text{eq1.2}$$

3. RESIDUE NUMBER SYSTEM

The Residue Number System [19] was founded by the Chinese scholar Sun Tzu in the first Century AD, who stated the Chinese Remainder Theorem. It was in the year 1734 that Euler provided the profor for the Chinese Remainder Theorem and introduced the concept for the 'Modulo' operation. The residue number system has been used for a long time to implement FID digital filters. The modulus operation is the basic operation that is used in the conversion for a number from Binary to Residue Number System. The Chinese Remainder Theory is used for converting back to binary from Residue Number System.

The Modulo operation in the modulus operation, the result is reset to the least value after the maximum value has been reached, i.e. the results for the modulus operation are the remainder left after division by the chosen modulus and these reminders are called 'residues' and they repeat. The modulus operation [18] can be seen as in a clock where the maximum number that is reached is twelve, and beyond which the clock starts over at one and repeats itself.

Consider a binary integer 'a' and positive modulus 'm', then we have

$$b=a(\text{mod } m) \qquad \text{eq3.1}$$

Here, 'b' is the remainder determined after 'a' is divided by 'm'. The divisor 'm' is called the modulus. The integer 'b' is called the residue and is not unique as more than one remainder can be determined during the division. The possible residues for a modulo 'm' operation are 0 to m-1.

For example, we can have 2 = 16(mod 7 but, we can also have

$$9 = 16(\text{mod}7)$$

Two integers that have the same residue are said to be equivalent. If two integers 'a' and 'b' are equivalent, then:

we can express them as, b(mod m) = a (mod m)

$$b = a(\text{mod } m)$$

The main advantage with the 'mod' operation is that modular addition subtraction and multiplication is carry free. The Residue Number System is based on the Modulus Operation. In RNS [17], a number is represented by a set for Moduli, i.e. the residue representation for an integer 'x' is a set for residues $\{r_1,r_2,\ldots\ldots r_n\}$, determined by performing the 'mod' operation on the integer 'x' using the corresponding set for Moduli $\{m_1,m_2\ldots\ldots\ldots m_n\}$. The integer r_i is also defined by the set for equations :

$$x=q_i m_i+r_i \text{ where } i=1,2,\ldots\ldots\ldots n$$

The integer q_i is such that $0 <= r_i < m_i$ Hence we can conclude that q_i is the quotient determined from x/m_i, also denoted by $[x/m_i]$ and r_i is the remainder determined from x/m_i or as the modulus operation performed on 'x'. The modulus for a number is not unique hence, we define the integer r_i as the least positive remainder determined by the division for x/m_i and is called the residue for x modulo m_i, [18] also written as x mod m_i or as

The residue representation for a number is unique as the least positive remainder for a number when divided by any number is unique. But the converse for this statement is not true.

For example, in a three Moduli system with $m_1=3$, $m_2= 4$ and $m_3= 5$ the residue representation for both 11 and 71 is {2, 3, 1}. This ambiguity occurs only in those numbers that meet the conditions stated by the theorem given below. In [18] it states that, two integers x and x' have the same residue representation for Moduli m_1, m_2...... m_n if and only if (x - x') is an integer multiple for the least common multiple for the Moduli, denoted by M. Hence, to have a unique mapping from the Residue representation, the integer 'x' must have a range for M (for positive number we would have x lie between 0 and M-I, however we require both positive and negative numbers. In this case the range for x is different, depending on whether M is even or odd. For even M, $-,M/2 \leq x <,M/2$ and for odd $\{M-(M-1)/2\} \leq x \leq \{M-(M-1)/2\}$.

Since the Residue Number System is based on the modulo operation, all the advantages present in the modulo arithmetic are also present in Residue arithmetic. Hence in the residue number system, addition, subtraction and multiplication are carry free, i.e. the result for the arithmetic operation is independent for the neighbouring digits. In multiplication there are no partial products, hence parallel operations can be carried out without having to wait for the results from adjacent bits.

Another factor to be considered in the residue number system is the Dynamic Range. Dynamic Range for the RNS is the product for the residue Moduli. Dynamic Range is the total number for residues that can be uniquely represented. From the dynamic range for the RNS, the number for bits that will be representing the number is determined. It is given as bits = $\frac{log_{10}(Product\ for\ Moduli)}{log_{10}(2)}$

The Moduli {3, 5, 7} corresponds to

$$\text{bits} = \frac{log_{10}(3x5x7)}{log_{10}(2)}$$

$$\text{bits} = \frac{log_{10}(105)}{log_{10}(2)}$$

$$\text{bits}=6.7142$$

The dynamic range must be considered because when the Residue Number System is used to implement the filter coefficients, the number for bits required to represent the filter coefficients are calculated.

3.2 RNS based FIR filter

Researchers have discussed about the optimized RNS based FIR filter model [4, 5]. It is defined by a set for relatively prime integers called the Moduli. The Moduli set are denoted as $\{m_1, m_2, ...m_n\}$ where $m_1,m_2,............m_n$ is the modulus. Each integer can be represented as a set for smaller integers called the residues. The residue-set is denoted as $\{r_1,r_2,..........r_n\}$ where r^i is i^{th} the residue. The residue is defined as the least positive remainder when X is divided by the Moduli m_i [5]. This relation can be notationally written based on the congruence and the equation is given below

$$\text{Mod } m_i = r_i \text{................(2)}$$

The same congruence can be written in an alternative notation as:

$$= r_i \text{..................(3)}$$

The RNS is capable for uniquely representing all integers X that lie in its dynamic range. The dynamic range is determined by the Moduli-set

$\{m_1,m_2 ,..,m_n \}$ and denoted as M where:

$$M = \prod_{i=1}^{n} mi \text{ ....(4)}$$

The RNS provides unique representation for all integers in the range between 0 and M-1. If the integer X is greater than M-1, the RNS representation repeats itself. Therefore, more than one integer might have the same residue representation. It is important to emphasize that the Moduli have to be relatively prime to be able to exploit the full dynamic range M.

The below figure 3.1 shows the general structure for RNS based FIR filter. X(t) and Y (t) are the input and output for this figure. The forward and reverse conversion is based on the special Moduli set and the New Chinese Remainder Theorem (NCRT) and the three FIR filter blocks are used here to speed up the processes.

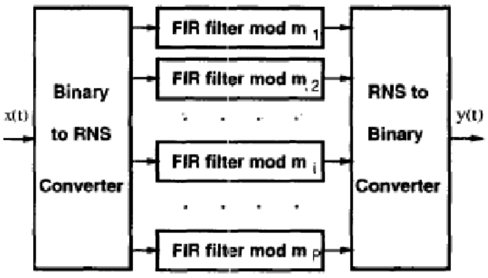

Figure 1: RNS based FIR filter

3.2.1 Choice for Moduli:

The choice for Moduli should satisfy the following conditions. They should be relatively prime. The Moduli should be as small as possible so that operations modulo require minimum computational time. The Moduli should imply simple weighted to RNS and RNS to weighted conversions as well as simple RNS arithmetic. The Moduli set should be for the forms 2^k+1, 2^k-1 and 2^k for simple conversions and simple arithmetic in RNS system. The product for the Moduli should be large enough in order to implement the desired dynamic range.

REFERENCES:

[1] S. Pontarelli, G. C. Cardarilli, M. Re, A. Salsano, “Optimized Implementation for RNS FIR Filters Based on FPGAs”, J Sign Process Syst. 67:201–212, DOI 10.1007/ s11265-010-0537, 2014

[2] S. Mirzaei, A. Hosangadi and R. Kastner “FPGA Implementation for High Speed FIR Filters Using Add and Shift Method” IEEE Explore, ISSN :1063-6404, p. no.432 – 437, 2014

[3] D Zivaljevi and N. Stamenkovic, and V. Stojanovi, present “FIR Filter Implementation Based on the RNS with Diminished-1 Encoded Channel”, Springer Journal Signal Processing System, Vol 7, 4 (2013), p. no.697-699 , 2013 Figure 3.1: RNS based FIR filter

[4] B. Basuvula and Mr. B. N. Srinivasa, “FPGA IJCSET, vol. 1, pp. 91- 94, 2012

[5] R. Conway, J. Nelson, “Improved RNS FIR filter Architecture”, IEEE Transaction on Circuits and Systems, vol.51, no.1, pp.26-28, January 2004.

[6] S. Pontraelli, G. C. Cardarilli, Marco Re, A. Salsano, “Optimized implementation for RNS FIR filters based on FPGAs", Springer Journal Signal Processing System, vol. 67, no.3 ,pp.201-212, September 2012.

[7] P. V. A. Mohan, “RNS to Binary converter for a Three Moduli set {2n+1-1,2n, 2n-1}”, IEEE Transaction on Circuits and Systems, vol.54, no.9, pp.775-779, September 2007.

[8] E. Grosswald, “Topics from the theory for Numbers”, Collier Macmillan Ltd., New York, 1996.

[9] A. Omondi and B. Premkumar, “Residue Number System: Theory and implementation”, Imperial College Press, , (ISBN 978-1-86094-866-4), 2007

[10] J.L. Beuchat, “Some Modular Adder and Multipliers for Area Programmable Gate Arrays”, Proceedings for the IEEE 17th International Symposium on Parallel and Distributed Processing, Washington DC, USA, pp-190.2, April 2003.

[11] S. Timarchi, K. Navi, “Improved Modulo 2n +1 Adder Design”, World Academy for Science, Engineering and Technology, 2001.

[12] G. Ignatius, M. Krishna Varma U, N. S. Krishna, P.V. P. Sudheesh, Extended Kalman Filter based Estimation for Fast Fading MIMO Channels, IEEE, ISSN :1063-6404, p. no.247-251, 2013

[13] A. G. Dempster and M.D. Macleod, "Comparison affixed-point FIR digital filter design techniques, " Trans. IEEE Circuits and Systems, vol. 44, pp. 591-593, July 1997.

[14] K.H.Tan, W.F. Leong, S. Kadam, M.A Soderstrand, L.G Johnson, "Public-domain MATLAB prograln to generate highly optimized VHDL for FPGA implementation, " IEEE ISCAS '01, vol. 4, pp.514 -517, May 2001.

[15] W. F. Leong, “Optimizing FIR filters coefficients using CSD representation and DM technique”, M S Thesis, Oklahoma State Univ., OK, May 2002.

[16] K. Howe Tan, "Optilnizing FIR filters coefficients using CSD representation andDMtechnique " , MS Thesis, Oklahoma State Univ., OK, May 2002.

[17] M. A. Soderstrand, W. K. Jenkins, G. A. Jullien and F. A. Taylor, "Residue Number System Arithmetic: Modern Applications in Digital Signal Processing, IEEE Press, pp. 591-593, 1988.

[18] N. S. Szabo and R.T. Tanaka, “Residue Arithmetic and its Applications to Computer Technology”, ISSN 0824707079, 2nd Edition, McGraw-Hill, 1997

[19] P.V. Ananda Mohan, “ Residue Number Systems: Algorithms and Architectures”, Kluwer Academic Publishers, *ISSN NO* :0925-9864, 2009

[20] FPGA general information, http:// vv'"\v\v.andraka.conl/\vhatisan.htn

[21] J.V. Oldarea; R. C. Dorf, “Area-Programmable Gate Arrays: Configurable Logic for rapid prototyping and implementation for digital systems”, John Wiley & Sons Inc., 1995.

[22] FPGA general information, http://www.vcc.com/fpga.html

[23] Xilinx official web page, http://xilinx.com

[24] Xilinx Virtex Application Notes, http://direct.xllinx.com/xaap/xaap130.pdf.

[25] R.K Mallik , K.V.Rangarao, "On the impulse response for a discrete-time linear IIR system" accepted and to appear in Digital Signal Processing Journal, December 2002.

[26] V. Pasham, A. Miller and K. Chapman, Transposed Form FIR Filters, Xilinx Application Note, XAP219 ver. 1.1, http://www.xilinx.coln/xapp/xapp219.pdf.

[27] K. Kaluri, W. F. Leong, K. H. Tan, L.G. Johnson, M.A. Soderstrand, “ Comparison for RNS and optimized FIR digital filters in Xilinx FPGA's”, Proc. 44th IEEE MWSCAS '01, vol. 1, pp. 438 -441,2001.

[28] K Kaluri, W. F. Leong, K.H. Tan, L.G. Johnson, M. A. Soderstrand, "FPGA hardware implementation for an RNS FIR digital filter”, 35th Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1340 -1344, 2001