Design and implementation of Pulse-Shaping FIR Interpolation filter using BCSE Algorithm

L. P. Usha Sri
4th Sem, M.Tech (VLSI and Embedded Design),ECE
CMRIT
Bangalore, India

Naveen Kumar G. N.
Assistant Professor,
CMRIT
Bangalore, India

***Abstract*— This brief proposes an optimization technique for the implementation of Root Raised Cosine FIR Interpolation filter. It utilizes the application of previously existing Window Technique in calculation of coefficients necessary for the filter. This reduces the number of taps in the filter thereby reducing the number of multipliers needed for the implementation. After selecting the necessary coefficients, it utilizes the previously existing 2-Bit Binary Common Sub Expression Elimination Algorithm to further reduce the number of multipliers needed for implementing Constant Multiplications. This drastic reduction in number of multiplications leads to more compact design with less area and therefore less power when compared to previous architectures. This technique has succeeded in reducing the area considerably and is considered more appropriate for designing area efficient Root Raised Cosine FIR interpolation Filters.**

Keywords*— *Finite impulse response (FIR) interpolation filter, Binary Common Sub Expression Elimination Algorithm, Window technique, square root raised cosine.

I. INTRODUCTION

In today's world, the key requirement in designing any communication or information bearing device is its compactness. People are in constant look out for efficient and robust designs which are compact in nature. In the telecommunication industry, filter plays a major role. The advantages of Finite Impulse Response (FIR) interpolation filter are far greater than those of Infinite Impulse Response (IIR) filter. Hence there is a constant need for efficient and small designs of FIR Interpolation filter.

A. Interpolation

Interpolation is done when there is a need for changing from one sampling rate to another.it is also called as Upsampling.it is a process of inserting zero-valued samples between original samples to increase the sampling rate. This is also called as zero stuffing. Interpolation is a process of up sampling and filtering of the signal. Because of up sampling, unwanted spectral components will be added to the signal. These undesired components are removed through filtering. Distortions may arise due to up sampling. Filtering the up sampled signal will remove distortions. [10] For example, consider speech processing systems. Speech parameters are computed at lower sampling rates for low bit storage or processing. For constructing a synthetic speech signal, the speech parameters are required to be at higher sampling rates. In these cases, digital interpolation process comes into the picture.

B. Square Root Raised Cosine for Pulse Shaping

Quantization of amplitude of a signal leads to conversion of continuous amplitude of a signal to the discrete steps. This sudden change of amplitude from one value to other, leads to high frequency spectral components in the frequency domain of the signal. The signal is transmitted on a limited bandwidth. In such cases, adjacent symbols may interfere with each other. This leads to Inter Symbol Interference (ISI). Due to ISI, the reconstructed signal at the receiver may not be same as the transmitted signal. Thus, the need for pulse shaping arises. Raised Cosine technique is a method for pulse shaping. It reduces the Inter Symbol Interference. In this technique, non-zero portion of frequency spectrum is a Cosine function which is Raised up above the Frequency axis. This technique satisfies Nyquist criterion of Sampling rate by removing the spectral noise at integral multiples of sampling rate. The transfer function of Receiver side and Transmitter side combined together will form a Raised Cosine filter. When considered only on Transmitter side, the square root of transfer function is considered. This is called Root Raised Cosine filter. According to [13], Transfer function of Root Raised Cosine Filter is given in equation (1)

$$\begin{aligned} h(t) &= (1/\sqrt{Ts})\,(1-\beta+4\beta/\pi), && t=0 \\ &= (\beta/\sqrt{2}Ts)\,[\,(1+2/\pi)\,\mathrm{Sin}\,(\pi/4\beta)+(1-2/\pi)\,\mathrm{Cos}\,(\pi/4\beta)], && t=\pm Ts/4\beta \\ &= (1/\sqrt{Ts})\,[(\mathrm{Sin}\,(\pi t/Ts\,(1-\beta))+4\beta t/Ts\,\mathrm{Cos}\,(\pi t/Ts\,(1+\beta)))/\,(\pi t/Ts\,(1-(4\beta t/Ts)^2))], && \text{otherwise} \end{aligned}$$

Where β –roll off factor, Ts – reciprocal of symbol rate (1)

C. Finite Impulse Response (FIR) Filters

FIR filters are a class of digital filters. They are also called as non-recursive digital filters since they do not have recursive (feedback) part in their design and is evident from equation (2). Although FIR filters can also be designed through the use of recursive algorithms.

Basic equation of FIR filter is

$$Z[n] = \sum_{j=0}^{M} aj\,x[n-j]$$

$$Z[n] = a_0\,x[n] + a_1\,x[n-1] + a_2\,x[n-2]\ldots\ldots\ldots a_M\,x[n-M]$$

M^{th} order filter has M+1 term on the right hand side (2)

According to [1], the symmetries of the filter coefficients can be utilized thus reducing the number of computations and hence area and power. [2] proposed a very efficient technique for obtaining the desired output of a filter by altering the precision and tap length of the filter. They provided an

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

efficient reconfigurable FIR filter with 32-taps.every filter design involves the use of adders and multipliers. [3] provided the lower bounds for a constant multiplication problem. Implementation of multipliers in hardware involves a lot of complexity in terms of area and power. Effective way to reduce the multiplier use is through the algorithm of common sub expression elimination [4]. From [4], it can be known that coefficient multiplication is easier if it is done in binary form. Dong Shi and Ya jun yu [5] proposed that FIR filters can be designed with optimum discrete components. This algorithm is used to design FIR filters with low complexity. According to [7], the order of the filter can be changed dynamically depending on the filter coefficients and input data. Reconfigurable architectures are possible by the use of algorithm proposed in [7]. [9] proposed that new reconfigurable architectures are possible by the use of common sub expression elimination algorithm. [14] proposed a two-step optimization technique for designing a reconfigurable architecture for multi standard digital up converter.[14] also provided a basic Binary Common Sub expression based algorithm for implementing a constant multiplier.

This brief proposes a technique of applying the rectangular Window algorithm[11] for carefully selecting the coefficients in designing an FIR filter which is based on BCSE algorithm [4],[9]. This brief also does a comparison for various input bit lengths in terms of gate count and delay.

II. ISSUES IN DESIGNING FIR FILTER USING BCSE ALGORITHM AND ITS SOLUTION

A. *Issues in designing FIR filters using BCSE Algorithm*

BCSE algorithm involves minimum usage of multipliers. It just involves shifting and adding operation. In this paper, the filter being designed is the programmable channel select filter at the baseband level. Implementation of FIR filter involves proper selection of coefficients (a_0, a_1,……a_n) and multiplying it with input and its delayed version. The filter design involves multipliers which usually are bulky and power consuming designs. This drawback of involving multipliers in the design led to development of many techniques and algorithms that gives the same result without the involvement of multipliers. Few of the techniques are canonical signed digit (CSD) [2], 3-bit BCSE Algorithm [4], 2-bit BCSE Algorithm [9][14]. Using the 2-bit BCSE algorithm, number of multiplications per unit sample has been reduced. The additions required to filter a unit input sample has also been reduced. This leads to reduction of hardware used in its implementation.

As proposed in [14], in BCSE algorithm, a data with n-bit word length can form 2^n–(n+1), binary common sub expressions among themselves [6]. Choosing an optimum length for BCS will result in efficient utilization of hardware. The computation time is also reduced. The computation time is measured in terms of Logic Depth. Logic Depth is defined as the number of stages of calculation of partial products in a multiplication. Partial products are obtained by repeating the shifting and adding operations of input bits. Maximum operating frequency of the filter is defined in terms of computation time.

Taps in a filter plays a major role in determining the performance of the filter. For an M-tap filter, the number of adders required are (M/2)*X. ‘X’ being the number of adders for a single constant multiplication operation. If there is a reduction of ‘K’ adders for a single constant multiplication, total number of (N/2)*K adders are reduced from the original filter [14]. Thus, using BCSE Algorithm, hardware can be effectively reduced in implementing an FIR filter which uses Root Raised Cosine technique for pulse shaping.

B. *Steps involved in the proposed design*

Step1. Get the input.

Step2. Get the coefficients after performing the rectangular window technique [11] to the standard root raised cosine FIR filter and according to the size of the input bits.

Step3. Perform the shifting and adding operation according to 2-bit BCSE algorithm [14].

Step4. Store the result in the output unit

Step5. Go to step 3 if the coefficients are not finished.

III. PROPOSED ARCHITECTURE

The major modules involved in the design are ‘data generator’ block, ‘coefficient generator and interpolation unit’ and ‘coefficient selector and accumulation unit’.

The proposed block diagram for the FIR interpolation filter based on BCSE is shown in Fig.1.

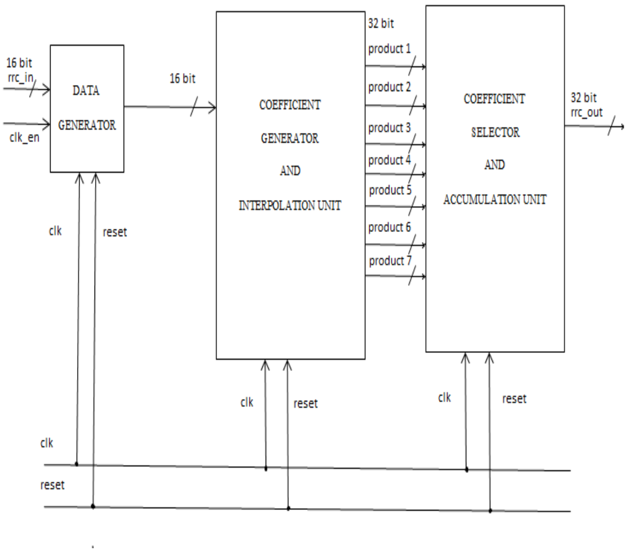


Fig.1. Proposed architecture of RRC FIR interpolation filter

A. *Data generator block:* If multiple standards are used then data generator block consists of a mux with ‘reset’ as its control signal for enabling or disabling it. Since only one frequency is used, mux has not been employed. Inputs are ‘rrc_in’, which is of 16 bit length, ‘clk’ clock signal,’clk_en’ clock enable bit and the ‘reset’, which is the universal reset bit. It is chosen to be active low in nature.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

B. *Coefficient generator and interpolation unit:* this block provides the coefficients necessary to generate the output. Number of taps of the filter depends on the coefficients selected. Initially, all the coefficients required for a 16 bit input are generated. Out of the generated 49 taps, optimum length of rectangular window is chosen to be 7, centered around the center frequency. All the other coefficients are neglected as their amplitudes are less than 10% of maximum amplitude at center frequency. The coefficients are generated with the interpolation factor of 4, roll off factor of 0.22 with 6 bits used per sample. These values are chosen to be optimum for the frequency around 70MHz. These coefficients can be viewed in Fig.2 This drastically reduces the taps which determine the number of adders and delay elements used. Interpolation is also done before selecting the coefficients. These coefficients, once selected, are multiplied with the input bits using 2-bit BCSE algorithm. Its implementation is as shown in Fig.2.Logic depth in 2-bit BCSE based filter is 4. For a 2-bit BCSE based filter [14], the propagation delay(D) is given by (3)

$$D = (4*T_{adder}) + T_{accumulation\ \&interpolation} \qquad (3)$$

Multiplication operation for a 16 bit data is given in (4)

$$Y = x + 2^{-1}x + 2^{-2}x + 2^{-3}x + \ldots\ldots\ldots\ldots.2^{-15}x; \qquad (4)$$

The 2-bit BCSE algorithm [14] is given by (5) and (6)

$$Z = x + 2^{-1}x; \qquad (5)$$

$$Y = Z + 2^{-2}Z + 2^{-4}Z + \ldots\ldots\ldots\ldots.2^{-14}Z; \qquad (6)$$

[0 0], [0 1], [1 0] does not require any adder for implementation as they have one non zero bit. The value of [1 1] can be obtained simply by adding [1 0] and the left shifted version of [0 1]. Hence it needs only one adder for the implementation. Constant Multiplier is the unit where 2-bit BCSE algorithm comes into the picture. It's basic unit is described in Fig.3. 'x' describes the input bit, 'h' represents the bit in the coefficient word. The left shifted version is obtained by a unit delay element. The number of basic units in a multiplier block depends on number of bits in the coefficient and the input.

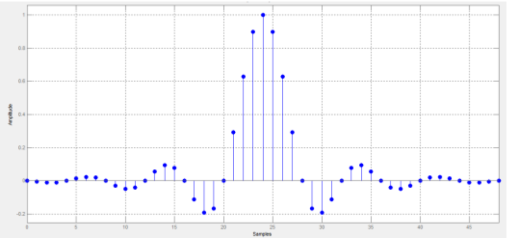


Fig.2.Coefficients of a RRC-FIR filter

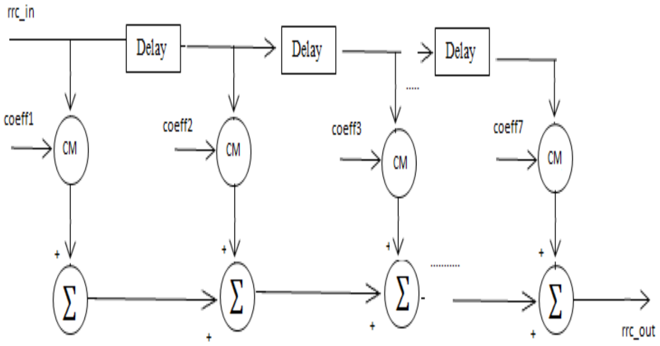


Fig.3. Architecture for implementing shift and add operation

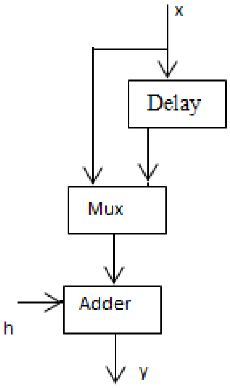


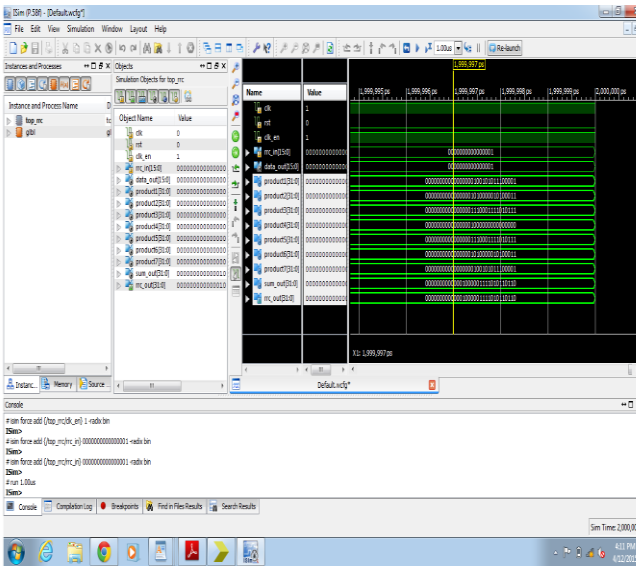
Fig.4. Implementation of basic unit in Constant Multiplier

C. *Coefficient selector and accumulation unit:* The interpolated data has to be accumulated into a single output. It is same as the final summation done to get the FIR filter response. This block accomplishes this task. It solely consists of string of adders. Because of using the BCSE algorithm, the number of adders used is less that required previously.

IV. RESULTS AND DISCUSSION

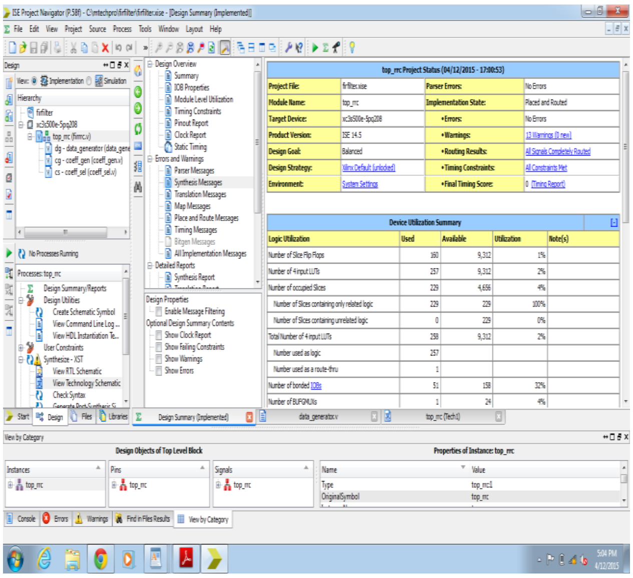
The proposed design is implemented on XC3S500E field programmable gate array (FPGA) device using Xilinx ISE14.5 EDA tool. The coefficients are calculated by the use of MATLAB R2013a version simulation tool. Snapshot of the result obtained for 16 bit input data and 16 bit coefficients has been shown in Fig.5. Design Summary of the implementation is shown in Fig.6. Comparisons have been made for varying coefficient lengths from 8 bits to 32 bits. The results can be viewed in Table I. Area can be estimated by observing the Gate count. Comparisons have been done with the existing algorithms in Table II.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)



```
# isim force add {/top_rrc/clk_en} 1 -radix bin
ISim>
# isim force add {/top_rrc/rrc_in} 0000000000000001 -radix bin
ISim>
# isim force add {/top_rrc/rrc_in} 0000000000000001 -radix bin
ISim>
# run 1.00us
ISim>
```

Fig. 5. Simulated output for 16-bit data



top_rrc Project Status (04/12/2015 - 17:00:53)			
Project File:	firfilter.xise	Parser Errors:	No Errors
Module Name:	top_rrc	Implementation State:	Placed and Routed
Target Device:	xc3s500e-5pq208	•Errors:	No Errors
Product Version:	ISE 14.5	•Warnings:	13 Warnings (0 new)
Design Goal:	Balanced	•Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	All Constraints Met
Environment:	System Settings	•Final Timing Score:	0 (Timing Report)

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	160	9,312	1%	
Number of 4 input LUTs	257	9,312	2%	
Number of occupied Slices	229	4,656	4%	
Number of Slices containing only related logic	229	229	100%	
Number of Slices containing unrelated logic	0	229	0%	
Total Number of 4 input LUTs	258	9,312	2%	
Number used as logic	257			
Number used as a route-thru	1			
Number of bonded IOBs	51	158	32%	
Number of BUFGMUXs	1	24	4%	

Properties of Instance: top_rrc

Name	Value
Type	top_rrc1
OriginalSymbol	top_rrc

Fig. 6. Design Summary for 16- bit data

TABLE I
COMPARISON RESULTS OF VARIOUS WORD LENGTHS

Word length	8 - bit	12 - bit	16 – bit	32 - bit
Gate count	7731	10524	24591	129188
Delay	12.675ns	12.789ns	13.599ns	20.165ns

TABLE II
COMPARISON RESULTS ON FPGA PLATFORM

References	Method used	Filter length	Max. frequency (MHz)	Gate count
XC3S500e (Proposed)	2-bit BCSE	7 Tap	73.53	24,591
[14] XC3S400N	2-bit BCSE	25/37/49 Tap	69.74	29,545
[14] XCV3000	2-bit BCSE	25/37/49 Tap	58.9	29,425
[14] XCV2000E	2-bit BCSE	25/37/49 Tap	49.5	30,564
[14] XC2VP4	2-bit BCSE	25/37/49 Tap	83.4	29,470
[15] XC2VP4	MAC Based	64 Tap	59.7	-
[4] XCV3000	3-bit BCSE	20 Tap	37.2	22,956

From the Tables, it is observed that the proposed algorithm works well for smaller bit counts. However, there is a tradeoff between area and maximum frequency. The proposed design consumes less area than those proposed in [4], [15], [14].

V. CONCLUSION

The brief addresses various issues in designing an efficient Root Raised Cosine FIR Interpolation filter with low complexity. This brief also offers an optimization technique for the already existing algorithm to make the desired filter more efficient by reducing area and hence complexity. Comparisons of the proposed technique with other available FIR architectures implemented on FPGA shows the advantages of the proposed architecture in terms of speed and area.

REFERENCES

[1] Y.Son, K.Roo, and Y.Kim, “1:4 interpolation FIR filter,” *IEEE Electron,Lett*.,vol. 40,no. 25,pp.1570-1572,Dec.2004.

[2] K.H. Chen and T.D.Chieueh, “A low power digit based reconfigurable FIR filter,” *IEEE trans. Circuits Syst. II,Exp. Briefs*, vol.53,no.8,pp.617-621,Aug.2006.

[3] O. Gustafsson, “Lower bounds for constant multiplication problems,” *IEEE Trans. Circuits Sys II, Exp. Briefs*, vol. 54,no.11,pp.974-978, Nov. 2007.

[4] R.Mahesh and A.P.Vinod, “A new common sub expression elimination algorithm for realizing low-complexity higher order digital filters,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol 27,no. 2, pp.217-229,Feb. 2008.

[5] D.Shi and Y.J.Yu, “Design of linear phase FIR filters with high probability of achieving minimum number of adders,” *IEEE trans. Circuits Syst.I,Reg. Papers*, vol. 58, no.1, pp.126-136, Jan. 2011.

[6] M.Thenmozhi, N.Krithika, “ Analysis of Efficient Architectures for FIR filters using Common Subexpression Elimination Algorithm”, International journal of Scientific and Technology Research, vol. 1, Issue 4, ISSN 2277-8616, May 2012.

[7] S.J.Lee, J.W. Choi, S.W.Kim, and J.Park, “A Reconfigurable FIR filter Architecture to trade off filter performance for dynamic power consumption,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 19,no. 12,pp. 2221-2228,Dec. 2011.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

[8] S.F. Hsiao, J.H. Zhang Jian, and M.C. Chen, " Low cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/accumulation," *IEEE trans. Circuits Syst. II,Exp. Briefs,* vol. 60, no. 5, pp. 287-291, May 2013.

[9] R.Mahesh and A.P.Vinod, " New reconfigurable architectures for implementing FIR filters with low complexity," *IEEE Trans. Comput. – Aided Design Integr. Circuits Syst.,* vol.29, no.2, pp.275-288, Feb. 2010.

[10] Ronald W. Schafer and Lawrence R. Rabiner, "A digital signal processing approach to interpolation," *IEEE Trans.,* vol. 61, pp. 692-702, June 1973.

[11] Elena Punskaya, "Design of FIR filters,"-www.sigproc.eng.cam.ac.uk/~op205 page 68-134.

[12] Michael Zoltowski, "Equations for the raised cosine and square root raised cosine shapes," Square root raised cosine-equations.pdf

[13] www.wikipedia.org/wiki/Root-Raised-Cosine_filter

[14] Indranil Hatai, Indrajit chakrabarti, Swapna banerjee, " An Efficient VLSI Architecture of a reconfigurable pulse shaping FIR interpolation filter for multistandard DUC," *IEEE* Trans., Very large Scale Integration(VLSI) Systems, pp.247-251, May 2014.

[15] J. Chandran, R.Kaluri, J.Singh, V.Owall and R. Velijanovski, "Xilinx vertex II Pro Implementation of a reconfigurable UMTS Digital channel filter," in Proc. IEEE Workshop Electron. Des., Test and Appl., Jan 2014,pp. 77-82.

[16] Digital filter design, www.Mikroelectronika.com

[17] www.wikipedia.com/FIRfilters

(This work is licensed under a Creative Commons Attribution 4.0 International License.)