

Research Article

Design of Low Power and Area Efficient New Reconfigurable FIR Filter using PSM and Shift and Add Method

¹G.O. Jijina, ²V. Ranganathan and ³R. Kalavathy

¹Department of ECE, St. Peters University, Avadi, Chennai-54, India

²Department of ECE, Sree Sastha Institute of Engineering and Technology, Chennai-123, India

³Department of ECE, Aarupadai Veedu Institute of Technology, Paiyanoor-603 104, India

Abstract: This study presents an architectural approach to the design of Low power and area efficient reconfigurable Finite Impulse Response (FIR) filter. FIR digital filters are used in DSP by the virtue of its, linear phase, fewer finite precision error, stability and efficient implementation. The proposed architectures implemented by using carry save adder, it offer Low power and area reductions and compared to the best existing reconfigurable FIR filter implementations in the literature and the proposed architectures have been implemented and tested on Spartan-3 xc3s200-5p4208 Field-Programmable Gate Array (FPGA) and synthesized.

Keywords: Channelizer, CSA, FIR filter, high speed filter, reconfigurability

INTRODUCTION

The explosive growth in mobile computing and portable multimedia applications has increased the demand for low power Digital Signal Processing (DSP) systems. One of the most widely used operations performed in DSP is Finite Impulse Response (FIR) filtering. The input-output relationship of the Linear Time Invariant (LTI) FIR filter can be expressed as the following Eq. (1):

$$y(n) = \sum_{k=0}^{N-1} c_k x(n-k) \quad (1)$$

where, N represents the length of FIR filter, c_k the coefficient and x the input data at time instant. In many applications, in order to achieve high spectral containment and/or noise attenuation, FIR filters with fairly large number of taps are necessary. Many previous efforts for reducing power consumption of FIR filter generally focus on the optimization of the filter coefficients while maintaining a fixed filter order (Samueli, 1989). In those approaches, FIR filter structures are simplified to add and shift operations and minimizing the number of additions/subtractions is one of the main goals of the research. However, one of the drawbacks encountered in those approaches is that once the filter architecture is decided, the coefficients cannot be changed; therefore, those techniques are not applicable to the FIR filter with programmable coefficients. The Multiplier Control Signal Decision (MCS) window scheme is introduced in Lee *et al.* (2011) for dynamically changes the filter order. The

Amplitude Detector (AD) and Control signal Generator (CG) circuits are used to detect and eliminate the consecutive number of low amplitude input data samples to provide efficient filter performance for dynamic power consumption. To reduce the complexity and increase the speed of operation of reconfigurable FIR filter, pipelined architectures are provided in Alex and Selvakumar (2013). A High speed PSM Multiplication technique by using common sub expression elimination algorithm is provided in Geethalakshmi and Jayamathi (2013). This multiplication results in area reduction and low complexity for multiplication operation. In Our proposed new reconfigurable FIR filter, PSM multiplier with BCSE algorithm and shift and add method is used to multiplication computation. The LUT in PSM is used to store the programmable coefficient value as coded format. In addition to, the fast carry save adder (Kishore and Rao, 2012) is incorporated to addition part of proposed new reconfigurable FIR filter (Chaplot and Paliwal, 2014) which further reduces the delay for filter operations.

ARCHITECTURE OF PSM

The PSM architecture presented in this section incorporates reconfigurability into BCSE. The PSM has a pre-analysis part in which the filter coefficients are analyzed using the BCSE algorithm in Geethalakshmi and Jayamathi (2013). Thus, the redundant computations (additions) are eliminated using the BCSS and the resulting coefficients in a coded format are stored in the LUT. The coding format is explained in

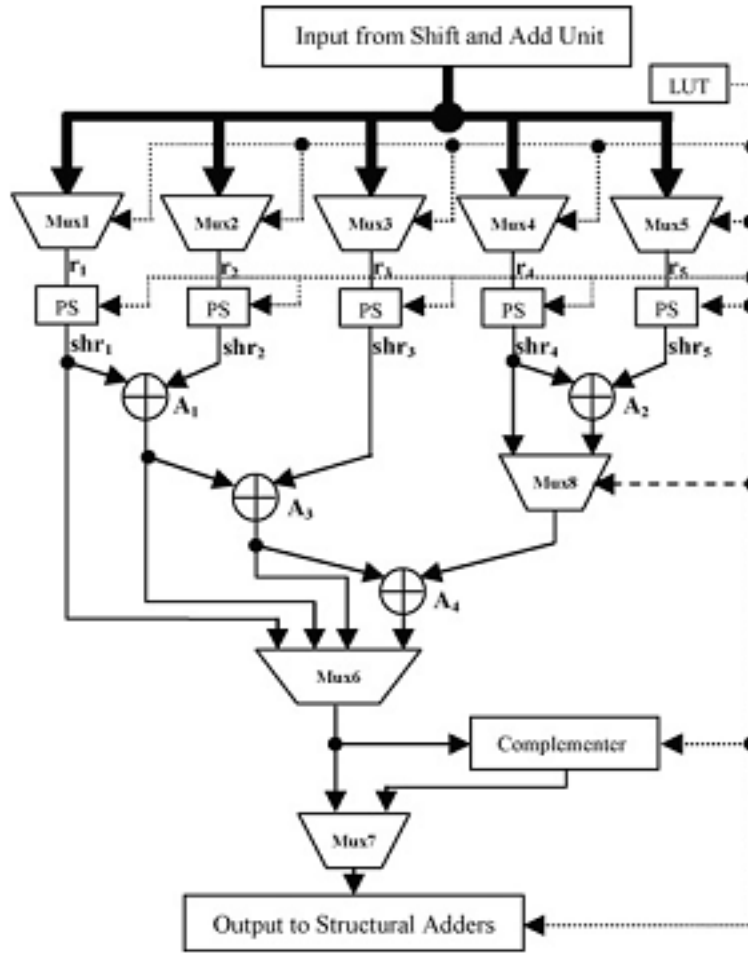


Fig. 1: Architecture of PE for PSM

the latter part of this section. The number of multiplexer units required can be obtained from the filter coefficients after the application of BCSE (Geethalakshmi and Jayamathi, 2013). The number of multiplexers is selected after considering the number of non-zero operands (BCSs and unpaired bits) in each of the coefficients after the application of the BCSE algorithm. The number of multiplexers will be corresponding to the number of non-zero operands for the worst-case coefficient (worst-case coefficient being defined as coefficient that has the maximum number of non-zero operands).

The architecture of PE for PSM (Geethalakshmi and Jayamathi, 2013) is shown in Fig. 1. The coefficient word length is fixed as 16 bits. We have done the statistical analysis for various filters with coefficient precision of 16 bits and different filter lengths (20, 50, 80, 120, 200, 400 and 800 taps, respectively) and it was found that the maximum number of non-zero operands is 5 for any coefficient. The analysis was done for filters with different pass band (ω_p) and stop band (ω_s) frequency specifications

given by 1) $\omega_p = 0.1\pi$, $\omega_s = 0.12\pi$; 2) $\omega_p = 0.15\pi$, $\omega_s = 0.25\pi$; 3) $\omega_p = 0.2\pi$, $\omega_s = 0.22\pi$; and 4) $\omega_p = 0.2\pi$, $\omega_s = 0.3\pi$, respectively. Based on our statistical analysis, we have fixed the number of multiplexers as 5 (same as the number of non-zero operands). The LUT consists of two rows of 18 bits for each coefficient of the form SDDDDXXDDDDXXMMMML and DDDDXDDDDXXDDDDXX, where "S" represents the sign bit, "DDDD" represents the shift values from 20 to 2-15 and "XX" represents the input "x" or the BCSs obtained from the shift and add unit. In the coded format, XX = "01" represents "x," "10" represents $x+2-1x$, "11" represents $x+2-2x$ and "00" represents $x+2-1x+2-2x$, respectively. Thus, the two rows can store up to five operands which is the worst case number of operands for a 16-bit coefficient. In most of the practical coefficients, the number of operands is less than the worst case number of operands, 5. In that case "MMMML" can be used to avoid unnecessary additions. The values "MMMM" will be given as select signal to the Mux6 and "L" to Mux8. "MMMML" indicates the presence of five operands (Samueli, 1989).

ARCHITECTURE OF PROPOSED RECONFIGURABLE FIR FILTER

The proposed coefficient representation technique uses signed digit to represent each sub-coefficients (Peiro *et al.*, 2002). In conventional coefficient partitioning method, the main coefficient may be assumed signed value, but the sub-coefficients are not signed.

For m-bit word length sub-coefficients case, their values are in $0 \sim 2^{m-1}$ range (Park and Roy, 2008). Eight partial products are calculated by pre-computer block using shift/sum operation and distributed to each tap's PE block. In practical implementation just four $\times 1$, $\times 3$, $\times 5$, $\times 7$ partial products are implemented in pre-computer block and other products ($\times 2$, $\times 4$, $\times 6$, $\times 8$) are composed by simple hardware shift operation of above four partial products inside PE block. The required four sub-coefficients to compose the desired coefficient are selected by four 8:1 multiplexers, (Mitola, 2000) which are controlled by MUX control block. This block uses h_i, j bits of each sub coefficient to control the selection bits of multiplexers. Note that it is need to eight partial products ($\times 1$, $\times 3$, $\times 5$, $\times 7$, $\times 9$, $\times 11$, $\times 13$ and $\times 15$ in practical implementation) and four 16:1 multiplexers in

conventional reconfigurable FIR filter architecture. The selected four partial products in PE block, after hardware shift operation are combined by add/sub operation while controlled by Add/Sub control block. This block uses the sign bit of each sub-coefficient and control the add/sub block. To implement the multiplication by zero for each sub coefficient, the multiplexer blocks are followed by AND gates, which is controlled by MUX control block (Mahesh and Vinod, 2008, 2010). Three full add/sub blocks are used to combine the partial products of sub coefficients. Implementing reconfigurable filter is as small as a single multiplier.

Pre-computer: The multiplications, $1 \times, 3 \times, 5 \times, 7 \times, 9 \times, 11 \times, 13 \times, 15 \times$, performed by the pre computer are simply implement Adder, Fig. 2 shows the whole pre computer structure (Lee *et al.*, 2011).

The proposed method of Add and Shift method will implement into the direct form fir filter for multiplier part shown in Fig. 2. The Processing Element in Fig. 2 is replaced by proposed processing element with less number of multiplexers and shifters unit is shown in Fig. 3.

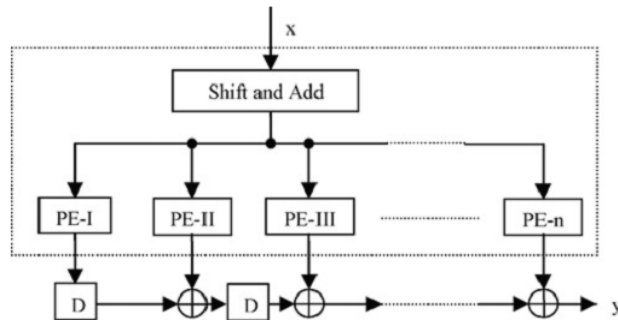


Fig. 2: Transposed direct form of an FIR filter

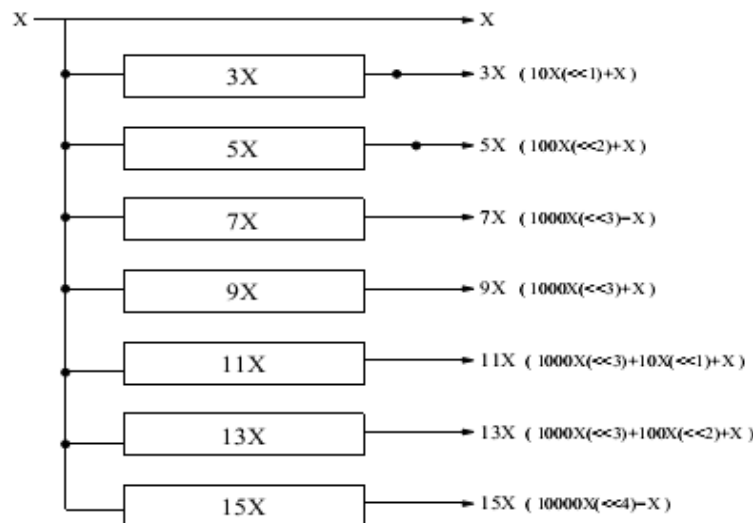


Fig. 3: The pre-computer structure

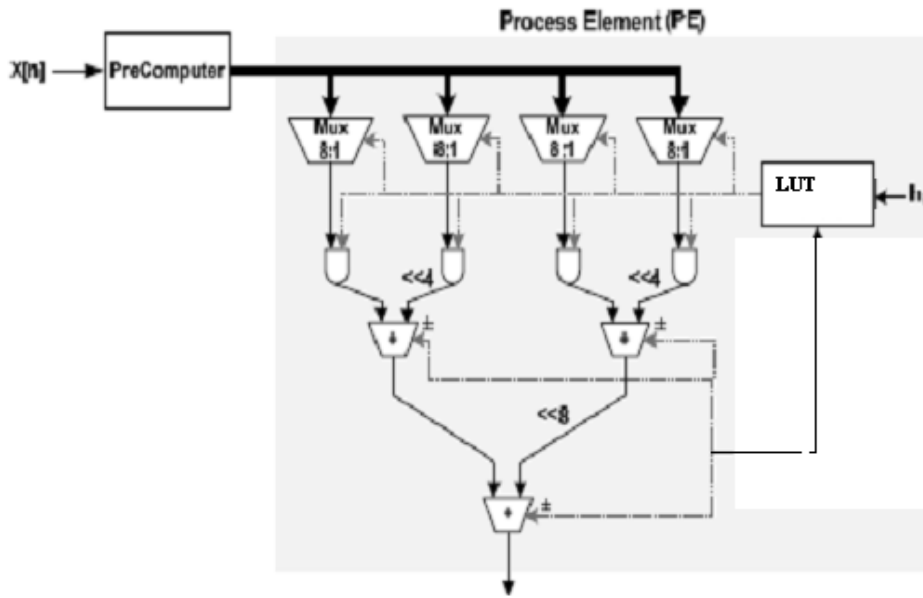


Fig. 4: Programmable element for proposed reconfigurable FIR filters

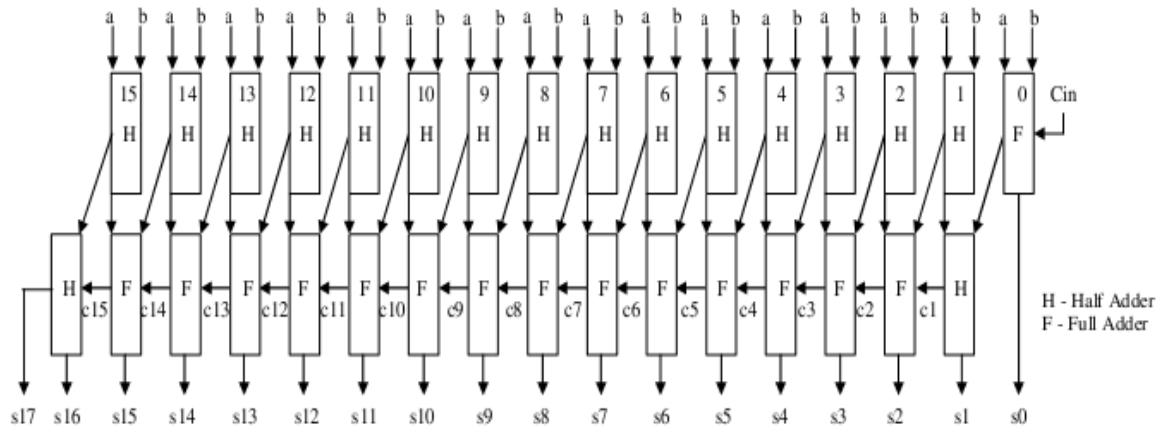


Fig. 5: Sixteen-bit CSA

Carry save adder: Basically, (Ramkumar *et al.*, 2010) carry save adder is used to compute sum of three or more n-bit binary numbers. Carry save adder is same as a full adder. But as shown in Fig. 4 (Alaoui, 2011), here we are computing sum of two 16-bit binary numbers, so we take 16 half adders at first stage instead of using 16 full adders. Therefore, carry save unit consists of 16 half adders, each of which computes single sum and carry bit based only on the corresponding bits of the two input numbers.

SIMULATION RESULTS

The results presented establish a clear area advantage of Proposed FIR architecture over prior architecture For Typical filter parameters with comparable Low power and Low area. The Proposed architecture achieved high clock frequency compared to

direct form architecture, (Anandan and Yogaanath, 2014; Jijina and Ranganathan, 2014) we validated our techniques on Spartan-III devices where we observed significant area and power. The comparisons for PSM and proposed Add and Shift method, in terms of LUT, Slices and Power are analyzed and tabulated in Table 1.

Table 1 shows that proposed shift and add method consumes less number of LUT and low power than PSM method. The proposed shift and add method of multiplication is incorporated to reconfigurable FIR filter. The programmable element in new reconfigurable FIR filter consists of less number of multiplexer and shifter units as shown in Fig. 5. The Simulation result for new reconfigurable FIR filter is shown in Fig. 6. The pictorial representations of comparisons for PSM and shift and add method performances are represented in Fig. 7.

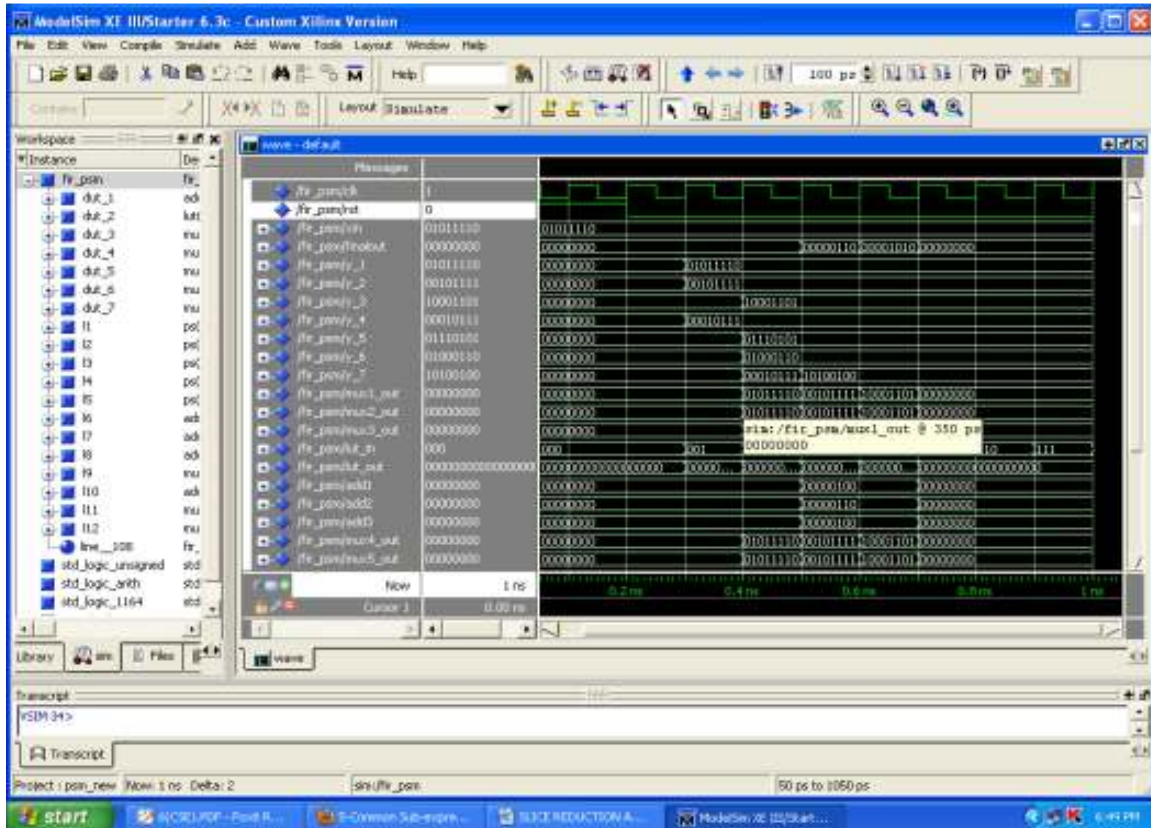


Fig. 6: Simulation result of proposed reconfigurable FIR filter

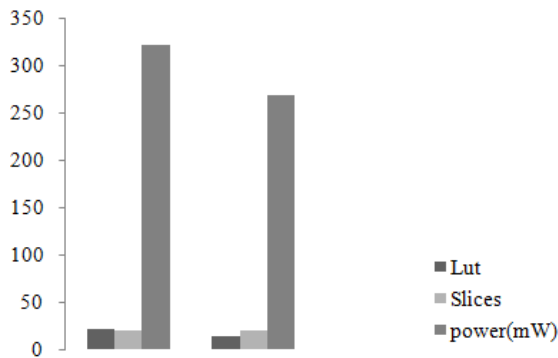


Fig. 7: Performance comparison of PSM and proposed add and shift

Table 1: Comparison of delay, clock frequency and power of direct form and proposed add and shift

Methods	PSM	Proposed shift and add
LUT	21	14
Slices	20	20
Power (mW)	321	235

CONCLUSION

We have proposed new approaches namely, novel based shift and add method for reconfigurable FIR filter, for implementing for low power and low area. The Proposed architecture provides the flexibility of

changing the filter coefficient word lengths dynamically. We have implemented the architectures on Spartan-III XC3S200-5PQ-208 FPGA and synthesized. The proposed reconfigurable FIR filter architecture gave Low area and low power when compared to the best existing reconfigurable FIR filter architecture. In future, the proposed reconfigurable FIR filter will be implemented in signal and image processing applications.

REFERENCES

Alaoui, C., 2011. Design and simulation of a modified architecture of carry save adder. *Int. J. Eng.*, 5(1): 102-113.

Alex, S. and J. Selvakumar, 2013. FPGA implementation of novel reconfigurable pipelined architectures for low complexity FIR filters. *Int. J. Appl. Res. Stud.*, 2(3).

Anandan, K. and N.S. Yogananth, 2014. VLSI implementation of reconfigurable low power FIR filter architecture. *Int. J. Innov. Res. Comput. Commun. Eng.*, 2(1).

Chaplot, R. and A. Paliwal, 2014. Low power and high speed reconfigurable FIR filter based on a novel window technique for system on chip. *Int. J. Digit. Appl. Contemp. Res.*, 2(6).

- Geethalakshmi, M. and A. Jayamathi, 2013. Reduction of multiplier in FIR filter using common sub expression elimination algorithm. *Int. J. Eng. Sci. Innov. Technol.*, 2(3), ISSN: 2319-5967.
- Jijina, G.O. and V. Ranganathan, 2014. Low power architecture for reconfigurable fir filter for SDR. *World Appl. Sci. J.*, 32(2): 164-168.
- Kishore, S.R.C. and K.V.R. Rao, 2012. Implementation of carry save adder in FPGA. *Int. J. Eng. Adv. Technol.*, 1(6), ISSN: 2249-8958.
- Lee, S.J., J.W. Choi, S.W. Kim and J. Park, 2011. A reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption. *IEEE T. VLSI Syst.*, 19(12): 2221-2228.
- Mahesh, R. and A.P. Vinod, 2008. Coefficient decimation approach for realizing reconfigurable finite impulse response filters. *Proceeding of IEEE International Symposium on Circuits System*, pp: 81-84.
- Mahesh, R. and A.P. Vinod, 2010. New reconfigurable architectures for implementing FIR filters with low complexity. *IEEE T. Comput. Aid. D.*, 29(2): 275-288.
- Mitola, J., 2000. *Software Radio Architecture: Object-oriented Approaches to Wireless Systems Engineering*. John Wiley and Sons, Chichester, New York.
- Park, J. and K. Roy, 2008. A low complexity reconfigurable DCT architecture to trade off image quality for power consumption. *J. Signal Process. Syst.*, 53(3): 399-410.
- Peiro, M.M., E.I. Boemo and L. Wanhammar, 2002. Design of high-speed multiplier less filters using a no recursive signed common sub expression algorithm. *IEEE T. Circuits-II*, 49(3): 196-203.
- Ramkumar, B., H.M. Kittur and P.M. Kannan, 2010. ASIC implementation of modified faster carry save adder. *Eur. J. Sci. Res.*, 42(1): 53-58.
- Samueli, H., 1989. An improved search algorithm for the design of multiplier less FIR filter with powers-of-two coefficients. *IEEE T. Circuits Syst.*, 36(7): 1044-1047.