



An Efficient VLSI Architecture of Fixed and Reconfigurable FIR based on Booth Multiplier

Ms. Kanaka¹, Ilayaraja M.E²

Meenakshi Ramaswamy Engg College¹

Assistant Professor, Meenakshi Ramaswamy Engg College²

Abstract: The possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, a flow graph for transpose form block FIR filter with optimized register complexity is derived. A generalized block formulation is presented for transpose form FIR filter. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP than the proposed structure by using Booth Multipliers. Application specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves less ADP and less EPS than the best available FIR filter structure proposed for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves less ADP than that of the existing direct-form blocks FIR structure. This Proposed System Implemented using Verilog HDL and Simulated by Modelsim 6.4 c and Synthesized by Xilinx tool. The proposed system implemented in FPGA Spartan 3 XC3S 200 TQ-144.

Index Terms: Transpose form, ADP, EPS, Ripple carry adder, Carry save adder, VLSI, FIR, Block processing.

I. INTRODUCTION

Finite Impulse Response (FIR) digital filter is widely used in several digital signal processing applications, such as speech processing, loud speaker equalization, echo cancellation, adaptive noise cancellation, and various communication applications, including software defined radio (SDR) and so on. Many of these applications require FIR filters of large order to meet the stringent frequency specifications.

This feature has been utilized to reduce the complexity of realization of multiplications. Several designs have been suggested by various researchers for efficient realization of FIR filters (having fixed coefficients) using distributed arithmetic (DA) and multiple constant multiplication (MCM) methods. DA-based designs use lookup tables (LUTs) to store pre computed results to reduce the computational complexity. The MCM method reduces the number of additions required for the realization of multiplications by common sub expression sharing, when a given input is multiplied with a set of constants. They can easily be designed to be "linear phase" (and usually are). Put simply, linear-phase filters delay the input signal, but don't distort its phase. They are simple to implement.

On most DSP microprocessors, the FIR calculation can be done by looping a single instruction. They are suited to multi-rate applications. By multi-rate, we mean either "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will be discarded (so the feedback will be incorporated into the filter). They have desirable numeric properties.

In practice, all DSP filters must be implemented using "finite-precision" arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters have no feedback, so they can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic. They can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.) This is an important consideration when using fixed-point DSP's, because it makes the implementation much simpler.



II. RELATED WORK

Pramod Kumar Meher (2006) proposed the structure that involves significantly less memory and less area delay complexity compared with the existing DA based structures for circular convolution. Besides, it is shown that the proposed systolic designs for circular convolution can be used for computation of linear convolution as well. Basant Kumar Mohanty and Pramod Kumar Meher (2015) explore the possibility of realization of block FIR filter in transpose form configuration for area delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Yu Pan and Pramod Kumar Meher(2014)proposed the resource minimization problem in the scheduling of adder tree operations for the MCM block, and presented a mixed integer programming (MIP) based algorithm for more efficient MCM based implementation of FIR filters. Experimental result shows that up to 15% reduction of area and 11.6% reduction of power (with an average of 8.46% and 5.96% respectively) can be achieved on the top of already optimized adder/subtractor network of the MCM block. Abbas Amira, Pramod Kumar Meherand Shrutisagar Chandrasekaran (2008) presented the design optimization of one and two dimensional fully pipelined computing structures for efficient implementation of finite impulse response (FIR) filter to obtain effective area, delay and power by using systolic decomposition of inner product computation based on distributed arithmetic (DA). The systolic decomposition scheme is found to offer a flexible choice of the address length of the lookup tables (LUT) for DA based computation to decide on suitable area time trade off. It is observed that by using smaller address lengths for DA based computing units, it is possible to reduce the memory size, but on the other hand that leads to increase of adder complexity and the latency. Realization of FIR filter in transpose form configuration for efficient area and delay realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, they have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. They have derived general multiplier based architecture for the proposed transpose form block filter for reconfigurable applications. In the existing method, the implementation of direct form structure has less area delay product (ADP) and less energy per sample (EPS) for the short length filters. But for medium or large length filters, it has high ADP and high EPS In the FIR filter structure, the ripple carry adders are used to add the partial inner products. The well known adder architecture, Ripple Carry Adder is composed of cascaded full adders for n bit adder. It is constructed by cascading full adder blocks in series. The carry out of one stage is fed directly to the carry in of the next stage bit parallel adder it requires n full adders.

III. PROPOSED SYSTEM

The possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, we have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. We have derived a general multiplier-based architecture for the proposed transpose form block filter for reconfigurable applications. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation direct-form FIR structure has less ADP and less EPS than the proposed structure.

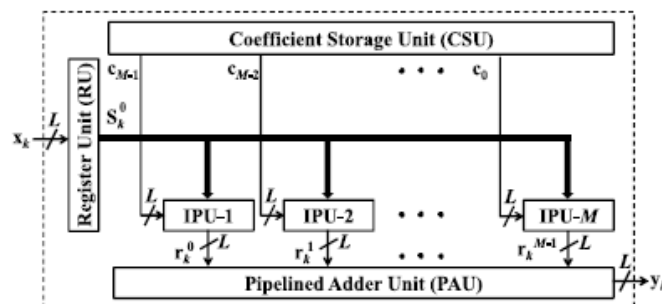


Fig 1: Proposed structure for block FIR filter

REGISTER UNIT:

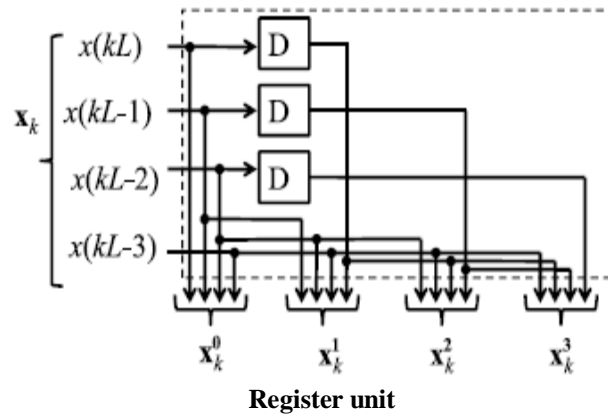
The RU receives x_k during the k^{th} cycle and produces L rows of S_0^k in parallel. It's designed by number of Flip Flop based on Input bit length.



**International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering**

ISO 3297:2007 Certified

Vol. 5, Issue 6, June 2017

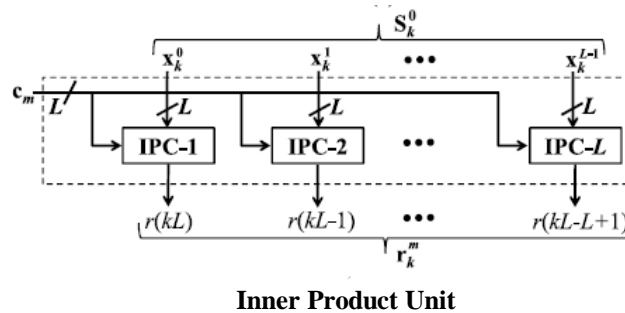


CO-EFFICIENT STORAGE UNIT (CSU):

The CSU stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using N ROM LUTs, such that

IPU

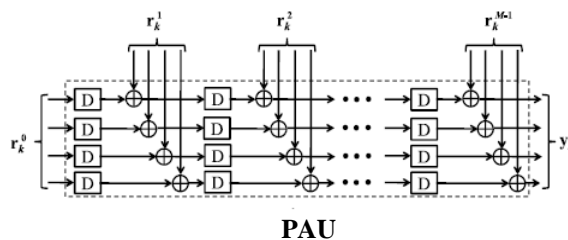
L rows of S_0 are transmitted to M IPU's of the proposed structure. The M IPU's also receive M short-weight vectors from the CSU such that during the k^{th} cycle, the $(m + 1)^{th}$ IPU receives the weight vector C_{M-m-1} from the CSU and L rows of S_0 form the RU. Each IPU performs matrix-vector product of S_k^0 with the short-weight vector c_m , and computes a block of L partial filter outputs (r_k^m). Therefore, each IPU performs L inner-product computations of L rows of S_0 with a common weight vector c_m . The structure of the $(m+1)^{th}$ IPU is shown in below Fig. It consists of L number of L-point inner-product cells (IPCs). All the M IPU's work in parallel and produce M blocks of result (r_k^m). These partial inner products are added in the PAU to obtain a block of L filter outputs. In each cycle, the proposed structure receives a block of L inputs and produces a block of L filter outputs.



Inner Product Unit

PIPELINED ADDER UNIT (PAU)

The PAU involves $L(M-1)$ adders and the same number of registers, where each register has a width of $(B+ B')$, B, and B' respectively, being the bit width of input sample and filter coefficients. The PAU is Designed by number of Adder and Number of Register.



PAU

BOOTH MULTIPLIERS

Booth's algorithm examines adjacent pairs of bits of the N-bit multiplier Y in signed two's complement representation, including an implicit bit below the least significant bit, $y_{-1} = 0$. For each bit y_i , for i running from 0 to $N - 1$, the bits y_i



**International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering**

ISO 3297:2007 Certified

Vol. 5, Issue 6, June 2017

and y_{i-1} are considered. Where these two bits are equal, the product accumulator P is left unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2^i is added to P; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from P. The final value of P is the signed product. The representations of the multiplicand and product are not specified; typically, these are both also in two's complement representation, like the multiplier, but any number system that supports addition and subtraction will work as well.

Radix-4 Booth algorithm scans strings of 3 bits with the algorithm given below:

1. Extending the sign bit 1 position if require, to ensure that n is even only.
2. Append a 0 to the right side of the least significant bit of the multiplier.
3. According to the value of each vector, Partial Product will be 0, +Y, -Y, +2Y, -2Y.

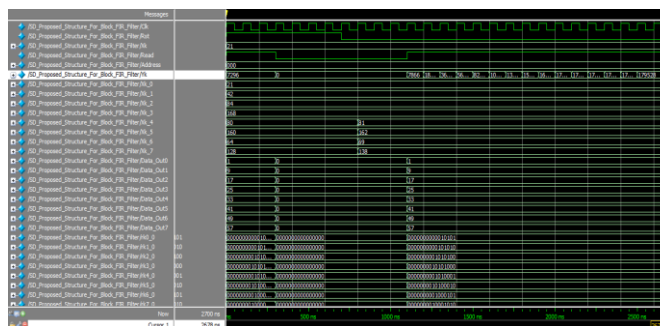
The negative values of y are considered by taking the 2's complement to the Booth recode the multiplier term, we have to consider the bits in groups of three, in a way that each group overlaps with the previous group by one bit. Grouping starts from the LSB and the first group only uses 2 bits of the multiplier.

groups	Partial products
000	0
001	1*multiplicand
010	1*multiplicand
011	2*multiplicand
100	-2*multiplicand
101	-1*multiplicand
110	-1*multiplicand
111	0

By using encoding table we can make encoding for Booth Multiplier.

IV. RESULT ANALYSIS

PROPOSED STRUCTURE FOR BLOCK FIR FILTER

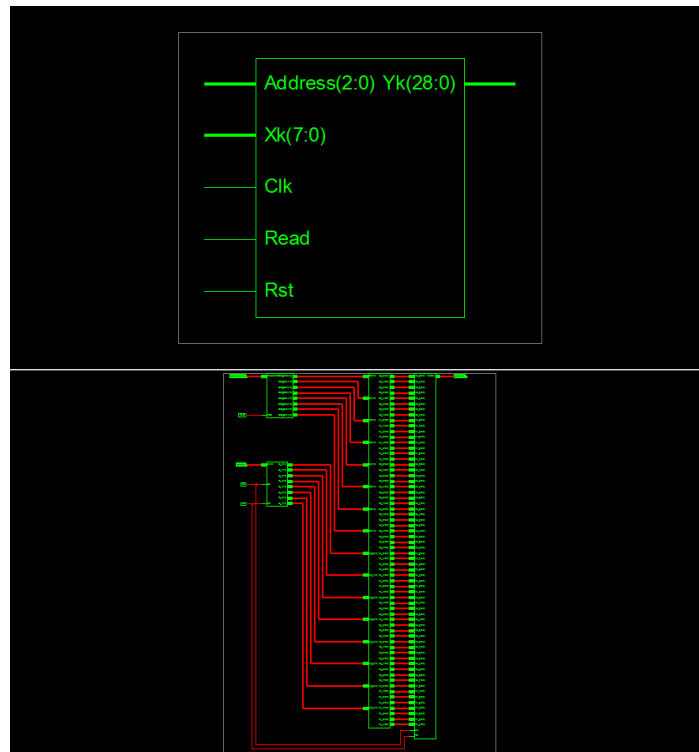


AREA AND DELAY COMPARISON

Method Name	Area in Number of LUT			Delay		
	LUT	Gate Count	Slices	Delay	Gate or Logic Delay	Path or Route Delay
Proposed FIR Filter	3214	111061	1259	22.500ns	13.259ns	9.241ns
					58.9% logic	41.1% route
Modified FIR Filter	4752	35204	789	37.680ns	16.383ns	21.297ns
					43.5%	56.5%



RTL SCHEMATIC VIEW



V. CONCLUSION

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for area delay efficient realization of both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications. We have presented a scheme to identify the MCM blocks for horizontal and vertical sub-expression elimination in the proposed block FIR filter for fixed coefficients to reduce the computational complexity. Performance comparison shows that the proposed structure involves significantly less ADP than the existing block direct-form structure for medium or large filter lengths while for the short-length filters, the existing block direct-form structure has less ADP than the proposed structure.

REFERENCES

- [1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
- [2] T. Hentschel and G. Fettweis, "Software radio receivers," in *CDMA Techniques for Third Generation Mobile Systems*. Dordrecht, The Netherlands: Kluwer, 1999, pp. 257–283.
- [3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, "A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals]," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 10, pp. 681–694, Oct. 1995.
- [4] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in *Proc. IEEE Southeastcon*, Apr. 1993, p. 1–6.
- [5] J. Mitola, *Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering*. New York, NY, USA: Wiley, 2000.
- [6] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1669–1675, Jul. 2006.
- [7] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," *IEEE J. Solid State Circuits*, vol. 39, no. 2, pp. 348–357, Feb. 2004.
- [8] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [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.