Implementation Field Programmable Gate Array using Modified Distributed Arithmetic & Finite Impulse Response by Filter Reconfigurable

Mr. N.Md. Mohasinul Huq[1] Subbamma Kalingiri[2]
[1]Associate Professor [2]PG-Scholar
[1,2]Department of Electronic & Communication Engineering
[1,2]SVR Engineering College Nandyal

Abstract— In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce/enhance certain aspects of that signal. A digital filter usually contains of an ADC to sample the input signal, it is continued by a microprocessor and the peripheral components such as memory to store data and filter coefficients etc. In some of the high performance applications, instead of using a general purpose microprocessor, FPGA or ASIC can be used for expediting operations such as filtering. One type of digital filter is FIR filter which is stable and gives linear phase response. For an Nth order FIR filter, the generation of each output sample takes N+1 MAC operations. Memory based structures are well-suited for many DSP algorithms, which includes multiplication with a set of coefficients which remains fixed. For this purpose, Distributed Arithmetic architecture is used in FIR filter. Distributed arithmetic is one way to implement convolution without the multiplier unit, where the MAC operations can be replaced by a series of LUT access and summations. LUT are the kind of logic that used in DRAM based FPGAs. It is mainly used in the applications like Software Defined Radio (SDR), Digital up/down converters, Multi-Channel filters where the coefficients are changed during the run time. Hence LUT's are needed to be reconfigurable. This project achieves high-throughput by implementing the reconfigurable FIR filter using modified Distributed Arithmetic (DA) based approaches. For this implementation of reconfigurable FIR filter RAM based LUT's are used, and the implementation of such LUT's remains costlier. Thus a shared LUT design is proposed for reconfigurable FIR filter. Requirement of separate registers are eliminated by sharing the registers for different bit slices. Thus the DRAM based FIR filter reduces the number of bit slices. The proposed design is implemented in Xilinx Virtex-5 FPGA device (XC5VSX95T-1FF1136).

Key words: Modified Distributed Arithmetic (DA), Finite Impulse Response (FIR) Filter, Reconfigurable Implementation, Field Programmable Gate Array (FPGA), Look Up Table (LUT)

I. INTRODUCTION

Digital filters are typically used to modify or alter the attributes of a signal in the time or frequency domain. It performs mathematical operations on a sampled or discrete time signal to reduce or enhance certain aspects of that signal. The most common digital filter is the linear timeinvariant (LTI) filter. An LTI interacts with its input signal through a process called linear convolution. LTI digital filters are generally classified as being finite impulse response (i.e., FIR), or infinite impulse response (i.e., IIR). As the name implies, an FIR filter consists of a finite number of sample values, reducing the above convolution sum to a finite sum per output sample instant. An IIR filter, however, requires that an infinite sum be performed. The motivation for studying digital filters is found in their growing popularity as a primary DSP operation. Digital filters are rapidly replacing classic analog filters, which were implemented using RLC components and operational amplifiers. Analog filters were mathematically modeled using ordinary differential equations of Laplace transforms. They were analyzed in the time or s (also known as Laplace) domain. Analog prototypes are now only used in IIR design, while FIR are typically designed using direct computer specifications and algorithms [1]. An FIR with constant coefficients is an LTI digital filter. It is a stable filter. It gives linear phase response. It can be seen to consist of a collection of a "tapped delay line," adders, and multipliers. One of the operands presented to each multiplier is an FIR coefficient, often referred to as a "tap weight" for obvious reasons. Historically, the FIR filter is also known by the name "transversal filter," suggesting its "tapped delay line" structure. A perfectly linear-phase filter has a group delay that is constant over a range of frequencies. The symmetry properties intrinsic to a linear-phase FIR can also be used to reduce the necessary number of multipliers L. Recently, with the advent of software defined radio (SDR) technology, finite impulse response (FIR) filter research has been focused on reconfigurable realizations.

Distributed arithmetic is one way to implement convolution with multiplier less unit, where the MAC operations are replaced by a series of LUT access and summations. Distributed Arithmetic is a different approach for implementing digital filters [3]. The basic idea is to replace all multiplications and additions by a table and a shifter accumulator. Basically each look up table is a bunch of single bit memory cells storing individual bit values in each of the cells. Distributed Arithmetic provides cost-effective and area-time efficient computing structures. The DA implementation of an FIR filter is particularly attractive for low-order cases due to LUT address space limitations. The outputs of a collection of low-order filters can be added together to define the output of a high-order FIR. To accelerate a DA filter, unrolled loops can be used. The input is applied sample by sample (one word at a time), in a bitparallel form. In this case, for each bit of input a separate table is required. While the table size varies (input bit width equals number of filter taps), the contents of the tables are the same.

II. RELATED WORK

A. Existing System

The existing structure for block FIR filter is [based on the recurrence relation of shown in Fig. 6 for the block size L = 4. It consists of one coefficient selection unit (CSU), one register unit (RU), M number of inner product units (IPUs), and one pipeline adder unit (PAU). The CSU stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using N ROM LUTs, such that filter coefficients of any particular channel filter are obtained in one clock cycle, where N is the filter length. The RU receives xk during the kth cycle and produces L rows of S0k in parallel. L rows of S0kare transmitted to M IPUs of the proposed structure. The M IPUs also receive M short-weight vectors from the CSU such that during the kth cycle, the (m + 1)th IPU receives the weight vector cM−m−1 from the CSU and L rows of S0k form the RU. Each IPU performs matrixvector product of S0k with the short-weight vector cm, and computes a block of L partial filter outputs (rmk). Therefore, each IPU performs L inner-product computations of L rows of S0k with a common weight vector cm. The structure of the (m+1)th IPU is shown in Fig. 7(b). It consists of L number of L-point inner-product cells (IPCs). The (l+1)th IPC receives the (l+1)th row of S0k and the coefficient vector cm, and computes a partial result of inner product r (kL −l), for $0 \leq l \leq L-1$. Internal structure of (l + 1)th IPC for L = 4 is shown in Fig. 8(a). All the M IPUs work in parallel and produce M blocks of result (rm k).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, where the duration of each cycle is T = TM + TA + TFA log2 L, TM is one multiplier delay, TA is one adder delay, and TFA is one full-adder delay.

B. Drawbacks

- It provides only block performances
- High delay
- Occupies high area

C. Proposed Method

There are several applications where the coefficients of FIR filters remain fixed, while in some other applications, like SDR channelizer that requires separate FIR filters of different specifications to extract one of the desired narrowband channels from the wideband RF front end. These FIR filters need to be implemented in a RFIR structure to support multi standard wireless communication. In this section, we present a structure of block FIR filter for such reconfigurable applications. In this section, we discuss the implementation of block FIR filter for fixed filters as well using MCM scheme.

III. IMPLEMENTATION

A. Proposed Structure

The proposed structure for block FIR filter is shown in Figure for the block size L = 4. It consists of one coefficient selection unit (CSU), one register unit (RU), M number of inner product units (IPUs), and one pipeline adder unit (PAU). The CSU stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using N ROM LUTs, such that filter coefficients of any particular channel filter are obtained in one clock cycle, where N is the filter length. The RU receives xk during the kth cycle and produces L rows of S0 k in parallel. L rows of S0 k are transmitted to M IPUs of the proposed structure. The M IPUs also receive M short-weight vectors from the CSU such that during the kth cycle, the (m + 1)th IPU receives the weight vector cM−m−1 from the CSU and L rows of S0 k form the RU. Each IPU performs matrixvector product of S0 k with the short-weight vector cm, and computes a block of L partial filter outputs (rm k). Therefore, each IPU performs L inner-product computations of L rows of S0 k with a common weight vector cm. In each cycle, the proposed structure receives a block of L inputs and produces a block of L filter outputs, where the duration of each cycle is T = TM + TA + TFA log2 L, TM is one multiplier delay, TA is one adder delay, and TFA is one full-adder delay

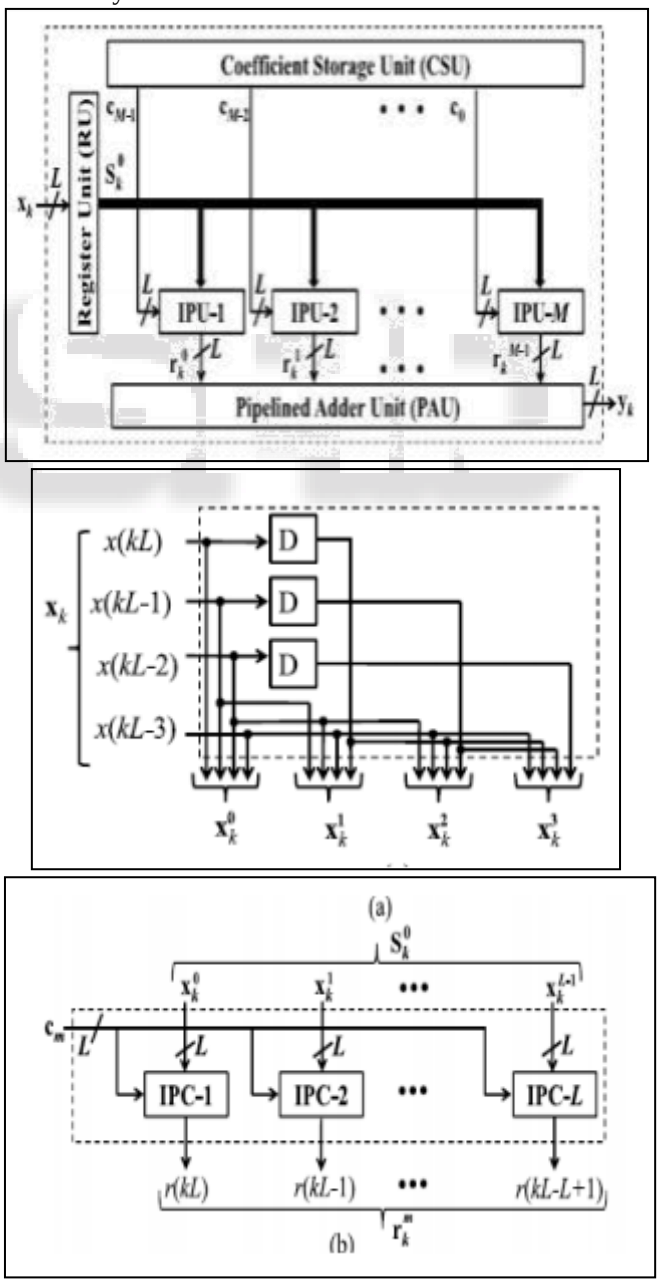

Fig. 1: proposed structures (a) Structure of RU (b) Structure of (m+1) IPU

B. MCM-Based Implementation of Fixed-Coefficient FIR Filter

We discuss the derivation of MCM units for transpose form block FIR filter, and the design of proposed structure for fixed filters. For fixed-coefficient implementation, the CSU is no longer required, since the structure is to be tailored for only one given filter. Similarly, IPUs are not required. The multiplications are required to be mapped to the MCM units for a low-complexity realization. In the following, we show that the proposed formulation for MCM-based implementation of block FIR filter makes use of the symmetry in input matrix S0k to perform horizontal and vertical common sub expression elimination [17] and to minimize the number of shift-add operations in the MCM blocks.

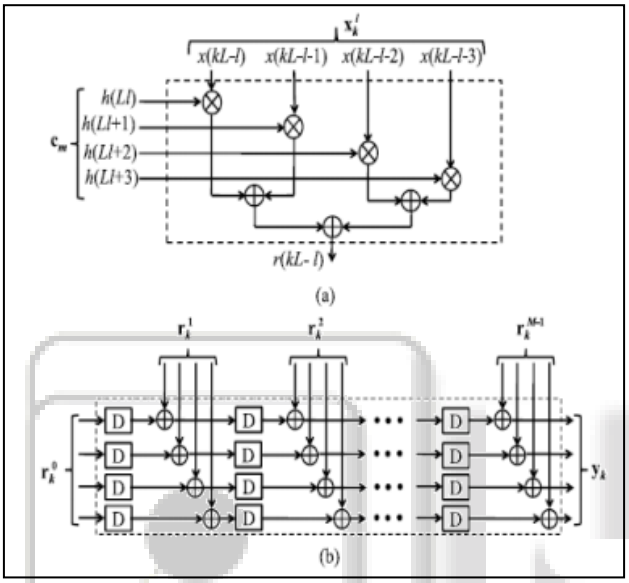

Fig. 2:

is no longer required, since the structure is to be tailored for only one given filter. The recurrence relation can be expressed as

Y(z)= z−1··z−1(z−1rM−1 +rM−2 +rM−3)··+r1+r0.

Where R =S0 k ·C

Similarly, IPUs are not required. The multiplications are required to be mapped to the MCM units for a low complexity realization. In the following, we show that the proposed formulation for MCM based implementation of block FIR filter makes use of the symmetry in input matrixS0k to perform horizontal and vertical common sub expression elimination and to minimize the number of shift-add operations in the MCM blocks.

C. Hardware and Time Complexities

The proposed structure for reconfigurable application consists of one CSU, one RU, M IPUs, and one PAU. The CSU consists of N ROM units of P words each, where P is the number of FIR filters to be implemented by the proposed reconfigurable structure. We have excluded complexity of CSU in the performance comparison, since it is common in all the RFIR structures. Each IPU is comprised of L IP cells, where each IP cell involves L multipliers and (L−1) adders. The RU involves (L − 1) registers of B-bit width. The PAU involves (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. Therefore, the proposed structure involves LN multipliers, L(N − 1) adders, and [B(N − 1) + B_(N − L)] (flip flops) FFs; and processes L samples in every cycle where the Duration of cycle period T = [TM + TA + TFA(log2L)].

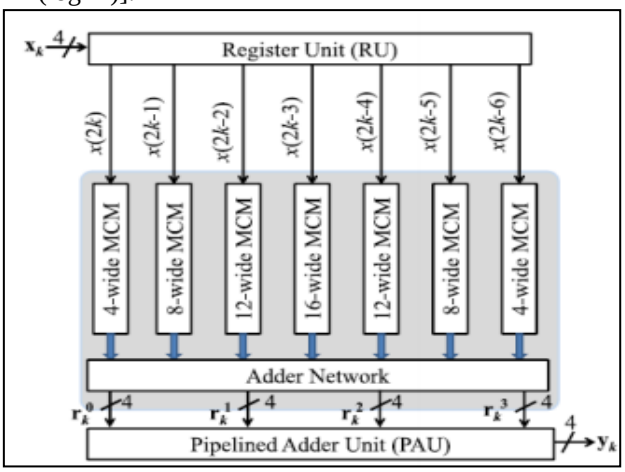

Fig. 3:

D. Advantages

- Block and higher order N processing.
- Less area requirement
- Low delay

IV. EXPERIMENTAL WORK

The results presented establish a clear area advantage of PSM over prior algorithms For Typical filter parameters with comparable maximum clock rates. In addition, the Industrial Relevance of the transposed FIR with multiplier block architecture and the RSG algorithm has been established through comparison with filters implemented using the Distributive Arithmetic Technique. In this chapter we presented a programmable multiplier less technique, based on the add and shift method and common sub expression elimination for low area, low Power and high speed implementations of FIR filters. We validated our techniques on Spartan-III devices where we observed significant area and power reductions over traditional Distributed Arithmetic based techniques and multiplier less technique.

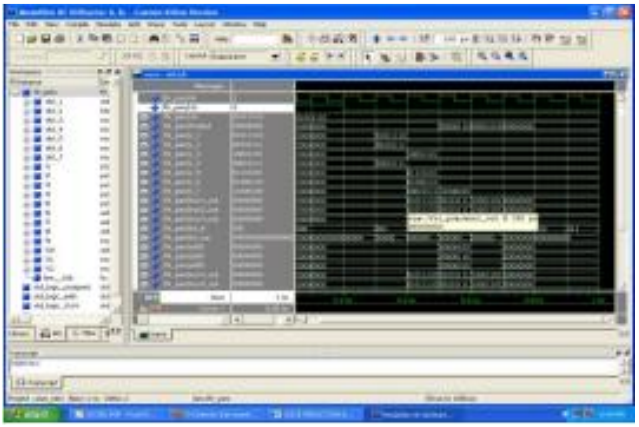

Fig. 4: Screen shot showing the simulation of PSM with filter

Methods	MSG	PSM
Slices	32	20
LUT	41	14
Power	0.389	0.268

Table 1: Comparison table of Power and Area of MSG and PSM

A. Comparison of Slices and LUT

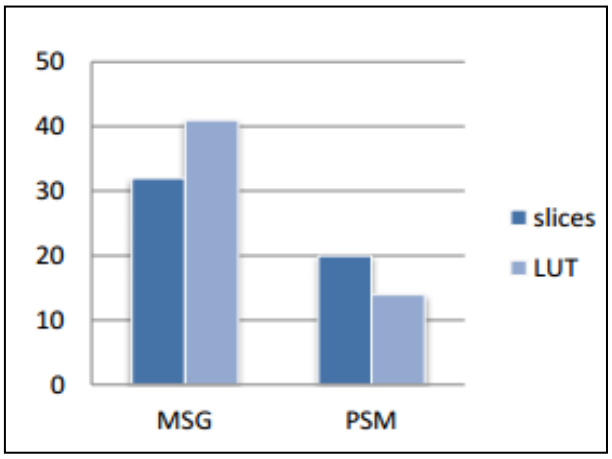

Fig. 5: Performance comparison of MSG and PSM.

B. Comparison of Power

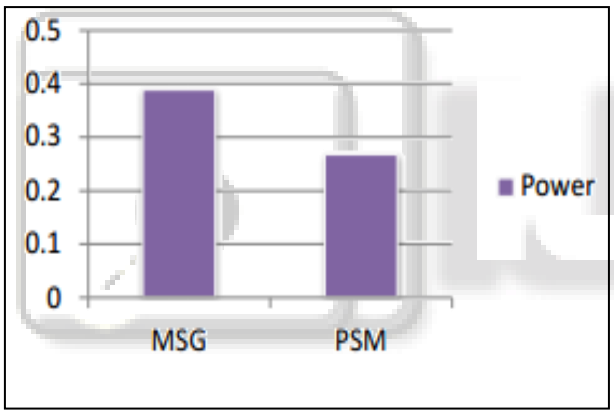

Fig. 6: Comparison of Power in MSG and PSM Methods

V. CONCLUSION

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for areadelay 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 subexpression 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 and less EPS 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 and less EPS than the proposed structure. Application-specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involve 42% less ADP and 40% less EPS than the best available FIR filter structure of [10] for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves 13% less ADP and 12.8% less EPS than that of the existing direct-from block FIR structure of [13].

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.
[10] S. Y. Park and P. K. Meher, "Efficient FPG and ASIC realizations of a DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 7, pp. 511–515, Jul. 2014.
[11] P. K. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.
[12] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul. 2008.
[13] P. K. Meher, "New approach to look-up-table design and memorybased realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592–603, Mar. 2010.

