FIR Filter Design For Reconstruction Of Stable And Dynamic Applications

[1]L.Ramesh, [2]P.Kishan

[1] *Sree* Dattha *Institute* of *Engineering* and Science, Hyderabad
[2] *Nova College* Of *Engineering* & Technology, Hyderabad

***Abstract* - Transpose-form finite impulse response (FIR) structures are inherently pipelined and support multiple constant multiplication (MCM) results in significant saving of computation. However, transpose-form configuration does not directly support the block processing unlike direct-form configuration.**

In this paper, we explore the possibility of realization of block FIR filter in transpose- from configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on detail computational analysis of transpose-form configuration of FIR filter we have derived a flow graph for transpose-from block FIR filter with optimized register complexity.

A generalized block formulation is presented for transpose-from 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 MCM scheme is also presented for the block implementation of fixed FIR filters.

Performance comparison shows that the proposed structure involves significantly less area-delay product (ADP) and less energy per sample (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 FIR structure has less ADP and less EPS than the proposed structure.

ASIC 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 structure proposed 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. Based on these findings, we present a scheme for the selection of direct-form and transpose-form configuration based on the filter lengths and block-length for obtaining area delay and energy efficient block FIR structures.

Keywords: **Multiple Constant Multiplication, FIR Filter, ADP and EPS.**

I. INTRODUCTION

Digital filters are a very important part of DSP. In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. As mentioned in the introduction, filters have two uses: signal *separation* and signal *restoration*. Signal separation is needed when a signal has been contaminated with interference, noise, or other signals. For example, imagine a device for measuring the electrical activity of a baby's heart (EKG) while still in the womb. The raw signal will likely be corrupted by the breathing and heartbeat of the mother. A filter might be used to separate these signals so that they can be individually analyzed.

Signal restoration is used when a signal has been distorted in some way. For example, an audio recording made with poor equipment may be filtered to better represent the sound as it actually occurred. Another example is the deblurring of an image acquired with an improperly focused lens, or a shaky camera. These problems can be attacked with either analog or digital filters. Which is better? Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. Digital filters, in comparison, are vastly superior in the level of performance that can be achieved. For example, a low-pass digital filter presented in Chapter 16 has a gain of 1 +/- 0.0002 from DC to 1000 hertz, and a gain of less than 0.0002 for frequencies above 1001 hertz. The entire transition occurs within only 1 hertz. Don't expect this from an op amp circuit! Digital filters can achieve *thousands* of times better performance than analog filters. This makes a dramatic difference in how filtering problems are approached. With analog filters, the emphasis is on handling limitations of the electronics, such as the

accuracy and stability of the resistors and capacitors. In comparison, digital filters are so good that the performance of the filter is frequently ignored. The emphasis shifts to the limitations of the *signals*, and the *theoretical* issues regarding their processing.

It is common in DSP to say that a filter's input and output signals are in the *time domain*. This is because signals are usually created by sampling at regular intervals of *time*. But this is not the only way sampling can take place. The second most common way of sampling is at equal

intervals in *space*. For example, imagine taking simultaneous readings from an array of strain sensors mounted at one centimeter increments along the length of an aircraft wing. Many other domains are possible; however, time and space are by far the most common. When you see the term *time domain* in DSP, remember that it may actually refer to samples taken over time, or it may be a general reference to any domain that the samples are taken in.

II. LITERATURE SURVEY

In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. This is in contrast to the other major type of electronic filter, the analog filter, which is an electronic circuit operating on continuous-time analog signals.

A digital filter system usually consists of an analog-to-digital converter to sample the input signal, followed by a microprocessor and some peripheral components such as memory to store data and filter coefficients etc. Finally a digital-to-analog converter to complete the output stage. Program Instructions (software) running on the microprocessor implement the digital filter by performing the necessary mathematical operations on the numbers received from the ADC. In some high performance applications, an FPGA or ASIC is used instead of a general purpose microprocessor, or a specialized DSP with specific paralleled architecture for expediting operations such as filtering. Digital filters may be more expensive than an equivalent analog filter due to their increased complexity, but they make practical many designs that are impractical or impossible as analog filters. When used in the context of real-time analog systems, digital filters sometimes have problematic latency (the difference in time between the input and the response) due to the associated analog-to-digital and digital-to-analog conversions and anti- aliasing filters, or due to other delays in their implementation.

III. FUNDAMENTALS AND ALGORITHMS:

a) DIGITAL FILTER:

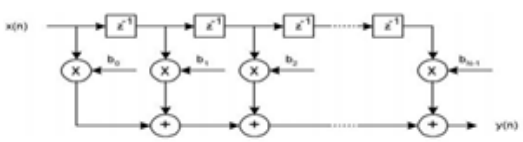
A digital filter is a system that performs mathematical operations on a sampled, digitized signal to reduce or enhance certain features of the processed signal. Digital filter scheme consists of a prefilter or anti-aliasing filter to perform filtering of an input signal using a low pass filter. This is required to restrict the bandwidth of a signal to satisfy the sampling theorem. An interface is

needed between the analog signal and the digital filter, this interface is known as analog-to- digital converter (ADC). After the process of sampling and converting, a digital signal is ready for further processing using an appropriate digital signal processor. The output signal that is digitized is usually changed back into analog form using digital-to-analog converter (DAC). The digital filtering process is shown in Figure 1 (Alam & Gustafsson, 2014). Digital filter is a major topic in the field of digital signal processing (DSP). Over the past few years the field of DSP has become so popular both technologically and theoretically. The major reason for its success in the industry is due to the use of the low cost and development of software and hardware. Applications of DSP are mainly the algorithms that are implemented either in software using interactive software like MATLAB or a processor. In high-bandwidth applications FPGA, ASIC or a specialized digital signal processor are used for expediting operations of filtering. Digital filters are preferably used because they eliminate several problems associated with analog filters. There are two fundamental types of digital filters: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) (Tian, Li, & Li, 2013).

b) FIR FILTER:

FIR filters also known as non-recursive digital filters have a finite impulse response because after a finite time the response of FIR filter settles to zero. Block diagram of FIR filter is shown in Figure 2. The basic

structure of FIR filter consists of adders, multipliers and delay elements as shown in Figure 3. The difference equation of nth order digital filter (FIR) can be represented as:



$$y(n) = \sum_{(k=0)}^{(N-1)} h(n)\,x(n-k) = \sum_{(k=0)}^{(N-1)} b_k\,x(n-k)$$

The transfer function $H(z)$ is given as:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^{N} a_i z^{-i}$$

where X(z) is the filter's input and Y(z) is the filter's output. In realization, a given transfer function is used to convert into a suitable filter structure (Xu, Yin, Qin, & Zou, 2013).

The main advantages of the FIR filter design over their IIR equivalents are the following:

(1) FIR filters with exactly linear phase can easily be designed.

(2) There exist computationally efficient realizations for implementing FIR filters.

(3) FIR filters realized non-recursively are inherently stable and free of limit cycle oscillations when implemented on a finite-word length digital system.

(4) Excellent design methods are available for various kinds of FIR filters with arbitrary specifications. (5) Design and noise issues are less complex than IIR filter (Rabiner, Kaiser, Herrmann, & Dolan, 1974).

c) DESIGN STAGES OF DIGITAL FILTERS:

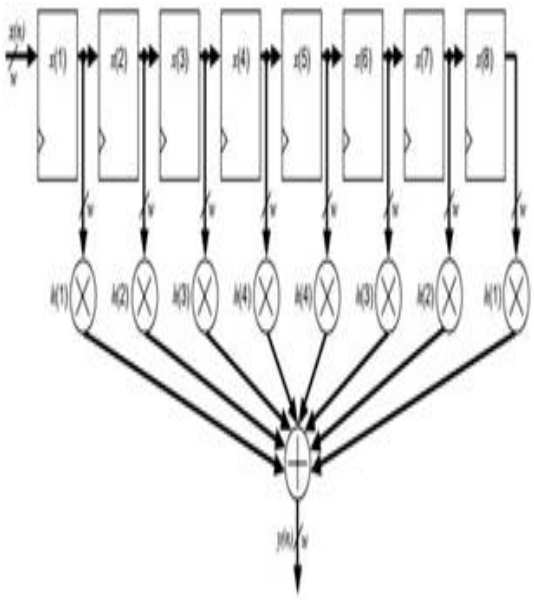
Design of a digital filter involves the following five steps:

(1) Filter specification
(2) Filter coefficient calculation
(3) Realization
(4) Analysis of finite word length effect and
(5) Implementation

In first stage the required specifications of the FIR filter are defined. Whereas, in second stage window method is selected because it offers a simple and flexible way of calculating the FIR filter coefficient; due to its well-defined equations.

IV. FIR FILTER ARHITECTURES

The finite impulse response (FIR) filter is used in many digital signal processing (DSP) systems to perform signal preconditioning, antialiasing, band selection, decimation/interpolation, low-pass filtering, and video convolution functions. Only a limited selection of off-the-shelf FIR filter circuits is available; these circuits often limit system performance. Therefore, programmable logic devices (PLDs) are an ideal choice for implementing FIR filters. Altera FLEX devices, including the FLEX 10K and FLEX 8000 families, are flexible, high- performance devices that can easily implement FIR filters. For example, you can use a FLEX device for one or more critical filtering functions in a DSP microprocessor-based application, freeing the DSP processor to perform the lower-bit-rate, algorithmically complex operations. A DSP microprocessor can implement an 8-tap FIR filter at 5 million samples per second (MSPS), while an off-the-shelf FIR filter circuit can deliver 30 MSPS. In contrast, FLEX devices can implement the same filter at over 100 MSPS. This application note describes how to map the mathematical operations of the FIR filter into the FLEX architecture and compares this implementation to a hard-wired design. Implementation details— including performance/device resource tradeoffs through serialization, pipelining, and precision—are also discussed.

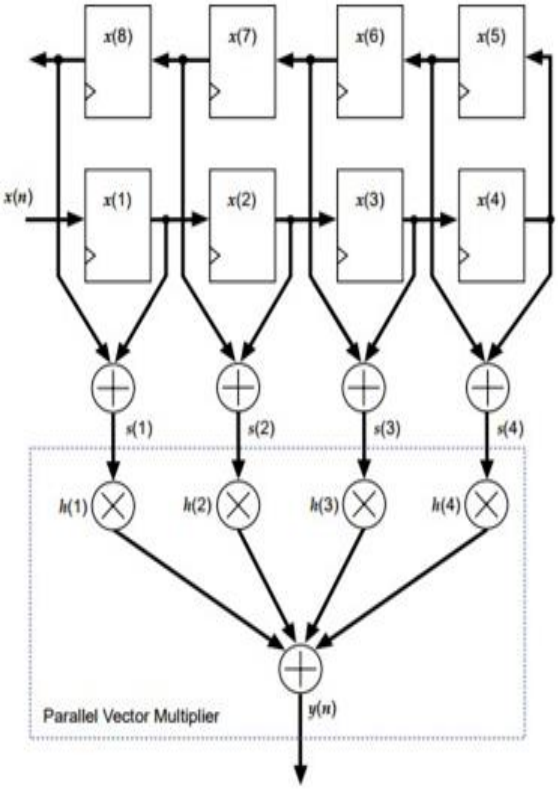


a) CONVENTIONAL FIR STRUCTURES:

The output of each register is called a tap and is represented by x(n), where n is the tap number. Each tap is multiplied by a coefficient h(n) and then all the products are summed. The equation for this filter is:

$$y(n) = \sum_{n=1}^{8} x(n)h(n)$$

For a linear phase response FIR filter, the coefficients are symmetric around the center values. This symmetry allows the symmetric taps to be added together before they are multiplied by the coefficients. See Figure 2. Taking advantage of the symmetry lowers the number of multiplies from eight to four, which reduces the circuitry required to implement the filter.



b) RECONFIGURABLE ARCHITECTURE:

Reconfigurable FIR filter architecture The architecture of block FIR filter for Reconfigurable applications is shown in the Fig.(a) for block size L=4. The main blocks are one Register Unit (RU), one Coefficient Storage Unit (CSU), one Pipeline Adder Unit (PAU), and M number of Inner Product Units (IPUs).

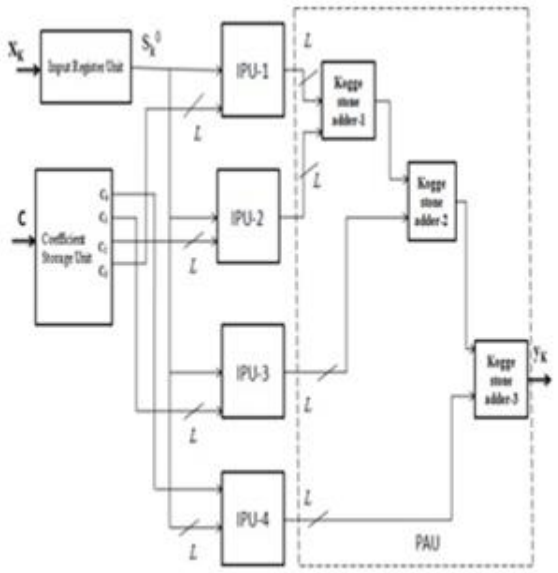


Fig.(a). Block FIR filter for reconfigurable applications.

The Coefficient Storage Unit (CSU) is used to store the coefficients of all the filters. These coefficients are used in the reconfigurable applications. It has N Read Only Memory (ROM) Lookup Tables (LUTs) where N is the length of the filter (N=ML). The Register Unit (RU) is used for storing the input samples is shown in Fig.2. It contains (L-1) registers. During the K th cycle, the register unit accepts input sample XK and computes L rows of SK 0 in parallel. The outputs from the RU are given as inputs to M Inner Product Units.

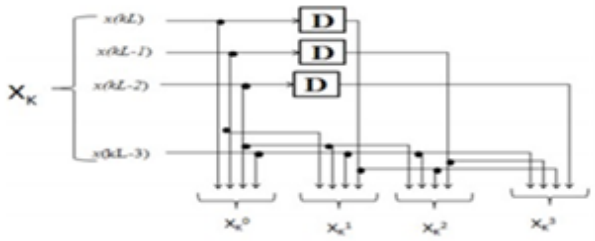


Fig.(b).Internal structure of Register Unit (RU) for block size L=4

The Inner Product Unit (IPU) is used to perform a multiplication operation of SK 0 with the small weight vector cm is shown in Fig.(b). The M Inner Products Units accepts L rows of SK 0 from the RU and M small weight vectors from the CSU. Each Inner Product Unit contains L number of Inner Product Cells (IPCs) which performs L inner product computations of L rows of SK 0 with coefficient vector cm and produces a block of L number of partial inner products. All the four IPUs work simultaneously and M blocks of a result are obtained.

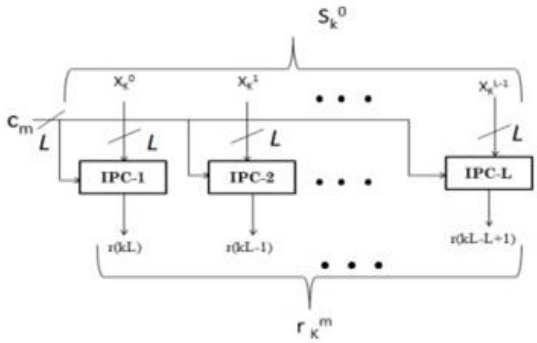


Fig.(b). Internal structure of (m+1)th IPU

The internal structure of (l +1) th IPC is shown in Fig.(c) The Inner Product Cell (IPC) accepts (l+1)th row of Sk 0 and small weight vector cm and produce a partial result of inner product r(kL
−l), for 0 ≤ l ≤ L – 1. Each IPC consists of L multipliers and (L-1) number of adders.

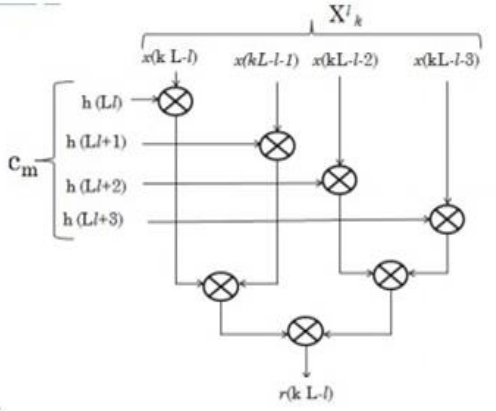
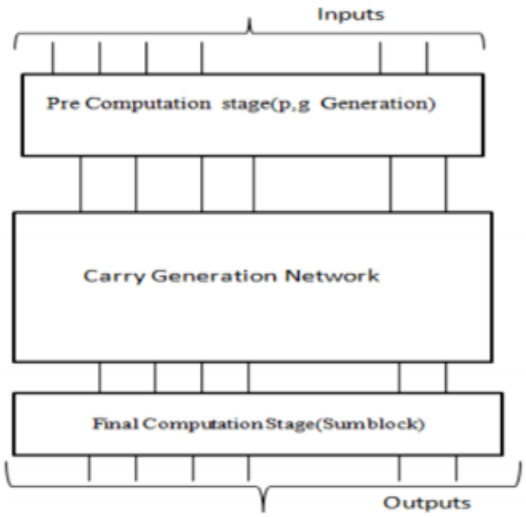


Fig.(c). Internal structure of (l + 1)th IPC

The Pipelined Adder Unit (PAU) receives partial products from all the M IPUs. Array of Kogge Stone Adder is used in PAU to add all the partial products is shown in Fig.5. KSA is one of the Carry Tree Adders or Parallel Prefix Adders. Kogge Stone Adders gains more importance among all the adders because of its high performance.

KSA can be implemented in 3 stages, namely Pre-Computation Stage, Carry generation network and final computation stage. Generate and Propagate signals are computed in Pre-Computation Stage, corresponding to each pair of input bits A and B. The second stage compute carries corresponding to each bit. Execution of these operations is performed in parallel form, and they are partitioned into smaller pieces. Group generate and propagate bits which are computed in the first stage are used as intermediate signals in carry generation network. The final computation stage is common for all the adders of this family which gives the summation of input bits.



Fig(d). Different Stages in Kogge Stone Adder

c) FIXED FIR FILTER ARCHITECTURE:

The architecture of block FIR filter for fixed application is shown in the Fig.6. For Fixed FIR filter implementation, the CSU is not necessary here filter coefficients are fixed. Similarly,

IPUs are not used because multiplication operation is performed with Multiple Constant Multiplication (MCM) units to reduce the huge complexity of the architecture.

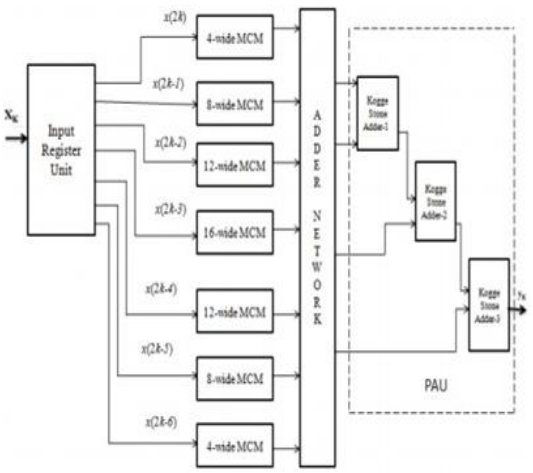


Fig.(e). Fixed FIR filter using MCM.

The MCM based method is more efficient when a given input variable is multiplied with more number of fixed constants using shift and add method is shown in Fig.7. It can be implemented by using adders/sutractors and shifters. Initially, the constants are expressed in binary form. Then for every non-zero digits in the binary format of the constant, based on its digit position the input variable is shifted and adds up the shifted variable to obtain a result. MCM is employed in many applications like error correcting codes, frequency

multiplication and Multiple Input and Multiple Output systems (MIMO).

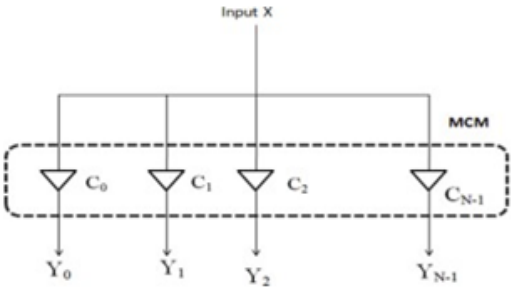
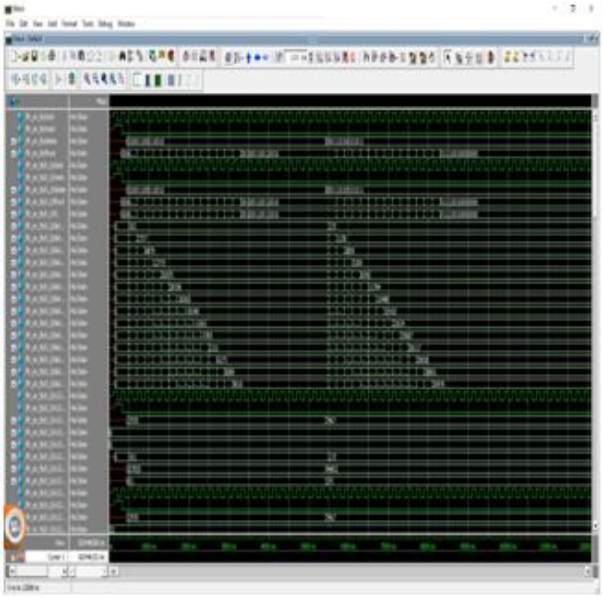


fig.(f). FBlock Diagram of MCM.

MCM based technique for a Fixed FIR filter with block size L=4, make utilize of symmetry in input matrix SK 0 to execute vertical and horizontal common subexpression elimination and to reduce the number of shift and add functions in the MCM units. MCM can be employed in both vertical and horizontal order of the coefficient matrix. The MCM based method consists of six input samples similar to six MCM blocks. All MCM blocks compute the required product terms using shift and add method. The outputs of all MCM blocks are given to the adder network to produce the inner product terms. In the Pipelined Adder Unit (PAU) array of KSA is used to add inner product values and produce a block of the filter output.

V. RESULTS



VI. CONCLUSION

FIR filters are extensively used in wired, wireless communications, video, audio processing and handheld devices are preferred because of their stability and linear phase properties. This paper presents a novel design methodology for an optimized FIR digital filters from software level to the hardware level.

The main goal is to encompass all the fields that are used in the efficient hardware realization of filters i.e. design method, selection of structure and the algorithm to reduce the arithmetic complexity of FIR filtering.

Theoretical and experimental result suggests that the Kaiser window gives the minimum mainlobe width i.e. 0.11719 and a sharp cutoff which means this window has less transition width, and the study showed that the direct-form structure approach is simpler, more robust to withstand the quantization errors, low cost and offers better performance than other common structures.

Proposed optimized filter implementation using an appropriate quantization scheme results in reducing arithmetic complexity, area and hardware resources. Comparison revealed that the optimized filter implementation is requiring 42% less hardware resources than the normal filter implementation.

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] J. Mitola, Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York, NY, USA: Wiley, 2000.

[3] 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.

[4] P. K. Meher, "New approach to look-up-table design and memory based realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592–603, Mar. 2010.

[5] 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.

[6] 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.

[7] 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.

[8] B. K. Mohanty and P. K. Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," IEEE Trans. Signal Process.,vol. 61, no. 4, pp. 921–932, Feb. 2013.

[9] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finiteimpulse response filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 1,pp. 120–133, Jan. 2014.

[10] S. Y. Park and P. K. Meher, "Efficient FPGA 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.