Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 8s (2024)

Optimizing Integrated Circuit Applications with Digital IIR Filtering Architectures

Bujjibabu P[1*], Kamaraju M[2], Babulu K[1*]

[1]Assistant Professor, Department of ECE, Aditya University.

[2]Professor, Department of ECE, GEC, Gudlavalleru, Krishna, AP, India.

[3]Professor, Department of ECE, UCE, JNTU-GV, Vizianagaram, AP, India.

[*]Research Scholar, Department of ECE, UCE, JNTU-GV, Vizianagaram, AP, India.

*Corresponding author: Email: bujjibabu_penumuchi@adityauniversity.in

Article History:

Received: *20-04-2024*

Revised: *10-06-2024*

Accepted: *24-06-2024*

Abstract

In this research, a high-performance IIR filter for low-power DSP applications is implemented and realized using a MATLAB system generator. There are numerous ways to implement IIR filters, including the Direct Form-I(DF-I), Direct Form-II(DF-II), Cascade, and Parallel Forms. All of these structures offer a selection area for a suitable realization that will decrease power consumption and increase the speed of digital filters. This particular work uses a Xilinx FPGA device to implement a 5th order Low pass IIR filter in DF-I, DF-II, and Cascade Form (DF-I/DF-II). Cascade (Direct Form-I) realization is the best way to build higher order IIR filters where power is a primary limitation, while cascade (Direct form-II), corresponding power analysis is also conducted, the technique is optimal if the area is a main constraint.

Keywords: MATLAB System Generator, IIR filter, Direct Form-I, Direct form-II, Cascade (Direct Form-I/ Direct form-II, FPGA.

1. Introduction

For all high-performance systems, lowering power consumption is crucial because it's ideal to maximize runtime with the least amount of space, weight, and battery life needed. Because of this, low-power technology has taken center stage in today's electronic world which would be the last phase of any application to be availed by the end user, irrespective of the field he or she is working.

DSP processors are used in the majority of applications, including those in telecommunication, medicine, speech processing, the military, image processing, and other fields, due to their dependability, high accuracy, and configuration flexibility. The power consumption of the device must be taken into consideration as a key design restriction for the implementation of high-performance DSP processors. To construct an advanced DSP, we are focusing on the processor's internal components. Digital filters are a key component of the architecture of DSP processors. One of the main factors contributing to DSP's enormous popularity is its phenomenal performance. The filter removes the unwanted components from the signal. Unwanted factors cause signal noise, making it impossible for the system to function effectively and provide accurate system information. Therefore, the filter's primary purpose is to filter out this noisy signal and deliver accurate system information. Two categories of filters exist based on the type of signal being processed. Those are analog and digital filters, respectively. Digital filters [4] have drawn a lot of interest in recent years.

Because digital filters offer improved features not found in any previous filter technologies, the strategy of digital sifters is a crucial matter here. It has been shown that there has been a significant increase in efficiency for computations involving the same level of signal of interest in less time. Based on their impulse response, filters be able to be viewed as two classes: Finite-Impulse-Response (FIR), and Infinite- Impulse-Response (IIR) filters. The main objective is to implement a high-performance IIR filter for DSP applications.

- Study about Digital filters
- Study realization architectures for implementing Digital filters (IIR filters)
- Generate filter transfer function based on general basic specifications
- Realize the generated filter function using the MATLAB system generator
- Compare performance reports (like power, area, delay) of all realization structures and finally conclude which realization architecture is best for implementing high-performance IIR filters based on design constraints.

Finite Impulse Response or FIR filters:

Since around is no feedback in the FIR filter, the impulse response is finite. It is a given that the impulse response will be finite if there is no feedback system. The phrase "finite impulse response" is hence also known as a "non-recursive filter".

$$y(n) = \sum_{i=0}^{N} b_i x(n-i)$$

In this above equation output y(n) is a function of past and present inputs.

Infinite Impulse Response or IIR filter:

Because the IIR filter incorporates feedback, the desired response is infinite, hence the name "Infinite Impulse Response" (IIR). The output should theoretically continue to ring endlessly when the input is an impulse. IIR-filter is similarly branded as a "recursive filter" as a result.

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

The output y[n] in the equation above is a function of previous output as well as previous and current input.

When compared to IIR filters, FIR filters provide a more consistent response, but they also have significant drawbacks. FIR filters are less effective than IIR filters because they suffer from several drawbacks. To attain a specific filter response characteristic, they need more memory calculation, and their latency is also longer than IIR filters. Additionally, the use of FIR filters to perform some replies is not practicable.

2. IIR Filter Architectures

IIR digital filters have the transfer function of the form [1][3][4].

$$H(Z) = \frac{\sum_{k=0}^{M} b_k\, z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Here $a_{k,}b_k$ are the filter coefficients

There are numerous ways to device IIR filters, including DF-I, DF-II, Cascade, Parallel forms, etc. All of the realized structures give the option to choose the best realization for lowering power consumption and accelerating the performance of digital filters.

Direct Form-I(DF-I): IIR-filters realization [5] of this form starts with this expression;

$$Y[n] = \sum_{k=0}^{N} b[k]X[n-k] + \sum_{k=1}^{N} a[k]y[n-k]$$

The IIR filter's non-recursive and recursive parts are discussed in turn in the expression's first portion. These two components are individually considered and realized in IIR filter DF-I.

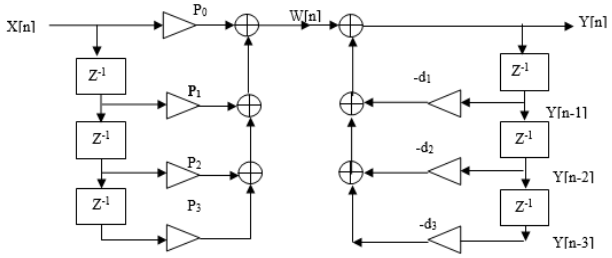

Fig. 1: Realization of IIR in DF-I style

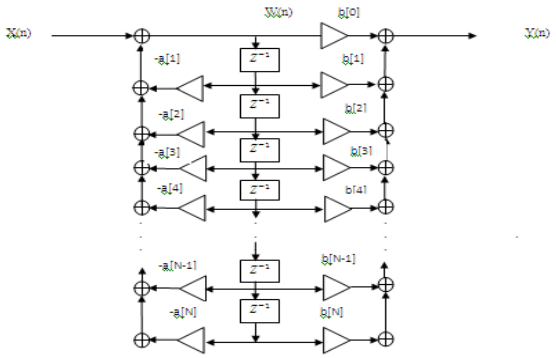

Fig. 2: Syntax of IIR in DF-II style

Direct Form-II(DF-II): Direct Form-II [5] realization structure reduced the number of delay lines to the minimum, in this realization structure the number of delay elements is equal to the order of the filter i.e. N, Direct canonic structure uses (2N+1) multiplications and 2N additions. Here recursive and non-recursive parts of the IIR filter are not required to realize separately.

$$Y[n] = \sum_{k=0}^{N} b[k]X[n-k] + \sum_{k=1}^{N} a[k]y[n-k]$$

Cascade Form: The direct form structures (DF-I, DF-II, or a combination of both) that are used to realize the sub-transfer functions or sub-system functions are connected in a cascade or series to

form the cascade-form construction. The provided transfer- function H(z) is realized in cascade form as the product of several second-order or first-order sections, as illustrated below.

$$H(z) = \frac{Y(z)}{X(z)} = \prod_{i=1}^{k} H_i(z)$$

X(z) → $H_1(z)$ → $H_2(z)$ → … → $H_k(z)$ ….

$$H(Z) = H_1(Z)\, H_2(Z)$$

Where

$H_1(z) = \frac{1 + b_{k1} + b_{k2}}{1 + a_{k1} + a_{k2}}$ and $H_2(z) = \frac{1 + b_{m1} + b_{m2}}{1 + a_{m1} + a_{m2}}$

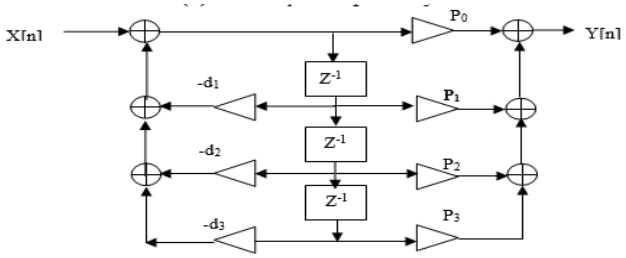

Fig. 3: Realization of IIR in DF-II style in cascade form

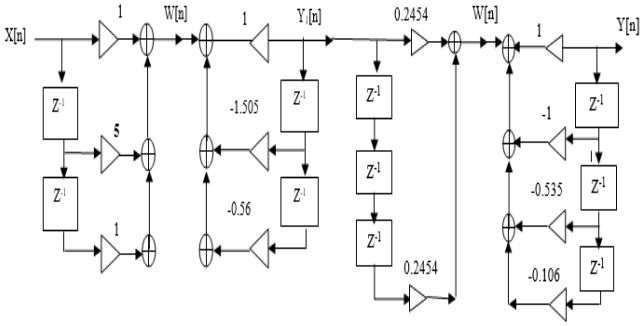

Fig. 4: Realization of IIR in Cascade form

Cascade form realization is used to gadget high-order filters which are realized by using either DF-I/DF-II, depending on user constraints. Hence the unwanted factor generates the noise in the signal so the system could not work properly and doesn't give accurate information. So, the main function of the filter is to filter this noise signal and provide accurate information about the system would be achieved.

3. DESIGN DESCRIPTION OF THE ARCHITECTURES.

The basic architectures are realized by taking a sample set of specifications like., Sampling frequency Fs=6 Khz, Pass- Band frequency Fp=2 Khz, Stop-band frequency Fstop=2.5 Khz, Pass-band attenuation Ap=1, Stopband attenuation Astop=22, and the Order=5. To realize the filter architecture from the specifications, one should follow the steps:

Step 1: Generate filter coefficients by using the MATLAB filter designer tool.

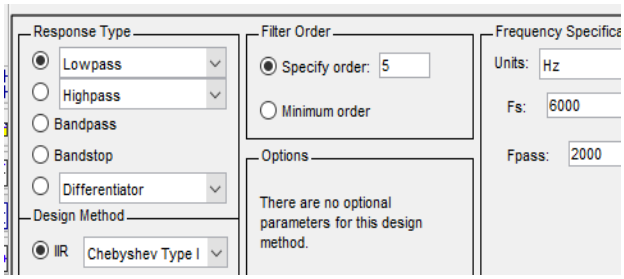

Fig. 5: supplying the specs to the filter designer

Step 2: Export filter coefficients to MATLAB workspace and generate transfer function.

$$\left[H(Z)=\frac{0.2454+1.227Z^{-1}+2.454Z^{-2}+2.454Z^{-3}+1.227Z^{-4}+0.2454Z^{-5}}{1+2.312Z^{-1}+2.531Z^{-2}+1.485Z^{-3}+0.4631Z^{-4}+0.0609Z^{-5}}\right]$$

Step3: Realizing the architectures from the transfer function in DF-I, DF-II, and Cascade forms using filter designer/Xilinx System Generator

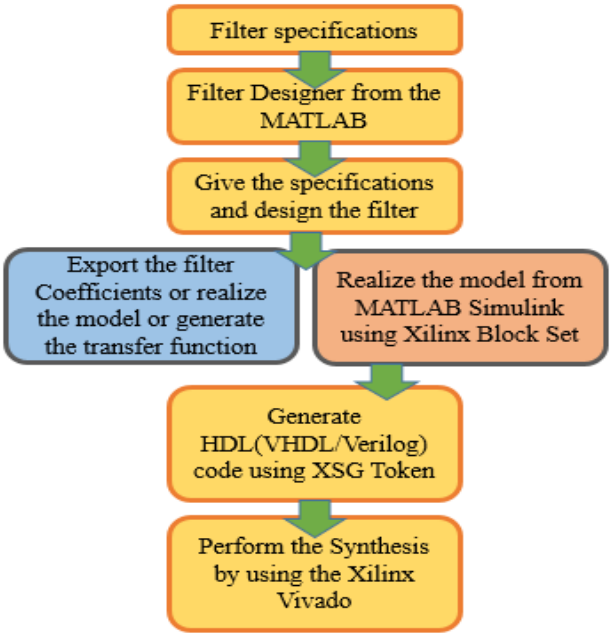

Fig. 6: Filter design flow using MATLAB Simulink and the Xilinx Vivado.

Step 4: HDL code generation using a system generator as per the FPGA board of interest.

Step5: Generate a Synthesis report to verify the design as per the requirement

Step6: Perform timing and power analysis for the realized architectures by taking the different constraints into consideration

Step 7: Comparison of power/area reports and identify the superior architecture for the given specifications.

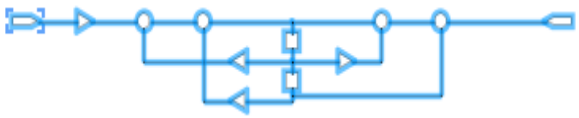

Fig. 7: Sample architecture realized by using the filter designer.

The interactive MATLAB filter designer tool can be used to create filters and validate their functionality. While the hardware requirements for the chosen model from the valid architecture cannot be remarked upon, it is necessary to use a VLSI-configured platform to convert the architecture in terms of hardware to estimate and then to obtain the netlist before creating an IC for the chosen model.

4. Implementation of a filter architecture

The Xilinx Block set is one of the interactive Toolboxes from the MATLAB Tool. Next to the functional verification, the model be re-framed or re-designed in the Simulink in which every identity is thoroughly configured with Xilinx Vivado to describe its hardware. The set of blocks that are used for architecture must be interfaced between Gateway-In and Gateway-Out. An additional set of blocks could be exported into the workspace from basic toolboxes. Any block, if outside, would be functioning as a normal Simulink block which never is converted as a netlist. Here the System Generator token plays a key role while extracting the netlist from the working model or the sub-system and is shown in Figure 8.

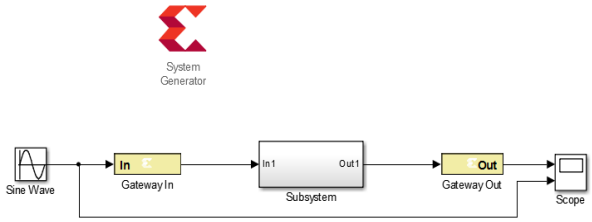

Fig. 8: Implementation of a filter as a sub-system

Subsystem Design:

A. Direct Form-I

The design of a 5th-order filter in DF-I using subsystem is represented in the following equation and is shown in Figure 9.

$$\left[H(Z)=\frac{0.2454+1.227Z^{-1}+2.454Z^{-2}+2.454Z^{-3}+1.227Z^{-4}+0.2454Z^{-5}}{1+2.312Z^{-1}+2.531Z^{-2}+1.485Z^{-3}+0.4631Z^{-4}+0.0609Z^{-5}}\right]$$

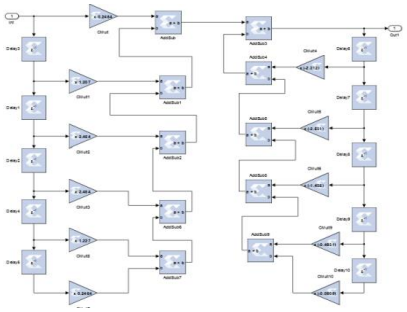

Fig. 9: Filter (DF-I) as a sub-system internal structure

B. Direct Form-II

The design of a 5th-order filter in DF-II using subsystem is represented in the following equation and is shown in Figure 10.

$$\left[H(Z)=\frac{0.2454+1.227Z^{-1}+2.454Z^{-2}+2.454Z^{-3}+1.227Z^{-4}+0.2454Z^{-5}}{1+2.312Z^{-1}+2.531Z^{-2}+1.485Z^{-3}+0.4631Z^{-4}+0.0609Z^{-5}}\right]$$

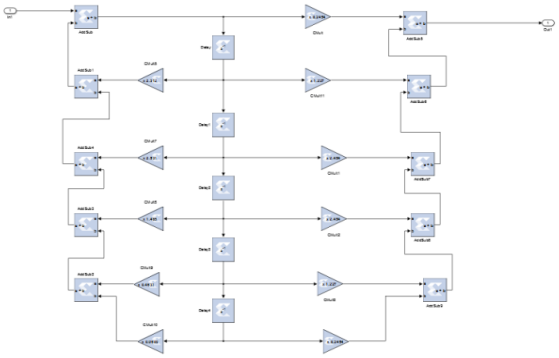

Fig. 10: Filter (DF-II) as a sub-system internal structure

C. Cascade form realization:

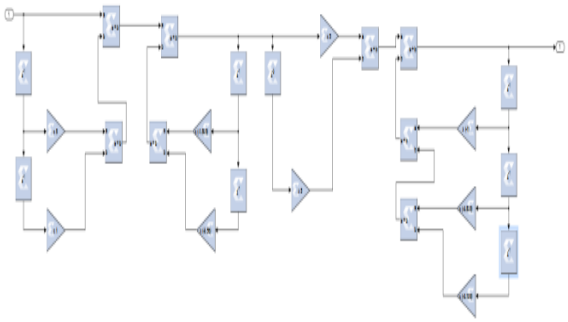

Fig. 11: Filter (Cascade form in DF-I) internal structure

The design of a 5th-order filter in cascade form of realization is represented in the following equation and is shown in Figure 11.

D. Cascade form realization (Direct Form-II)

The design of a 5th-order filter in cascade form using Direct Form-II realization is represented in the following equation and is shown in Figure 12.

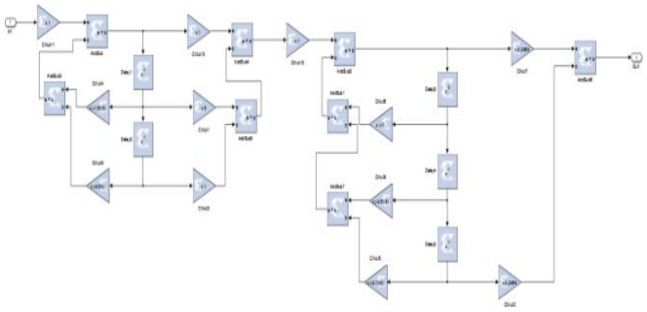

Fig. 12: Filter (Cascade form in DF-II) internal structure

5. RESULTS AND VALIDATIONS

The set of performance metrics is always w.r.t the type of the FPGA Board selected and the technology that is being utilized- which was invoked immediately after activating the token.

I. Power Performance Comparison at different frequencies:

A. Direct Form-I:

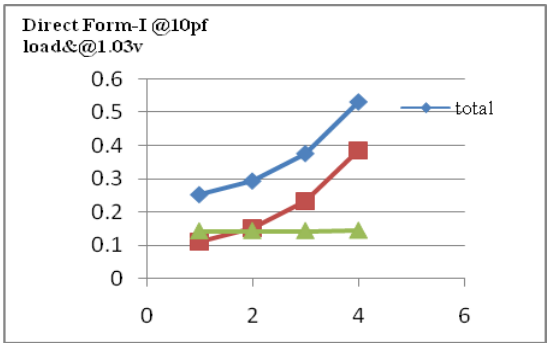

Fig. 13: Power Performance w.r.t frequency(DF-I)

Here, the IC operating frequency is obsolete at 4KHz only.

B. Direct Form-II:

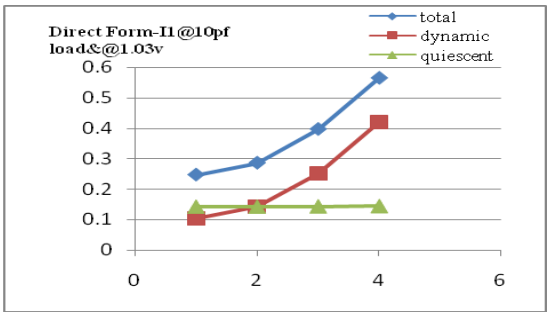

Fig. 14: Power Performance w.r.t frequency(DF-II)

Here, the IC operating frequency is obsolete at more than 4KHz only.

C. Cascade form (in DF-I):

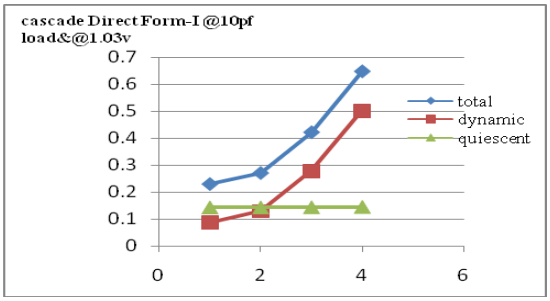

Fig. 15: Power Performance w.r.t frequency(in DF-I)

Here, the IC operating frequency is obsolete at more than 4KHz only and the total power consumption is more than the previous two architectures, under the same operating conditions.

. D. Cascade form (in DF-II)

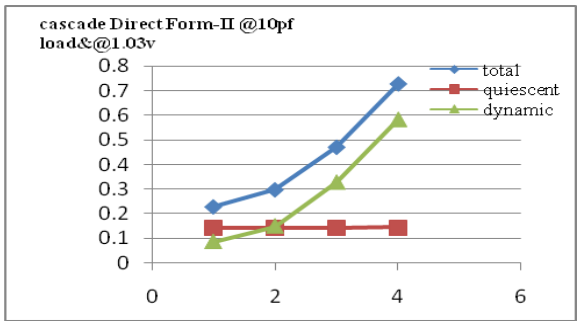

Fig. 16: Power Performance w.r.t frequency (in DF-II)

Here, the IC operating frequency is obsolete at also more than 4KHz, and the total power consumption is more than the previous three architectures, under the same operating conditions, and is said to be not a good choice.

DELAY Response of different Realization structures

Type of Architecture	Delay
Direct Form -I	61.74ns
Direct Form -II	50.353ns
CASCADE (Direct Form-I)	56.723ns
CASCADE (Direct Form-II)	55.36ns

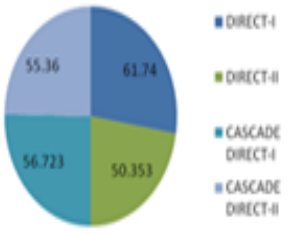

Fig. 17: Delay Performance of different architectures

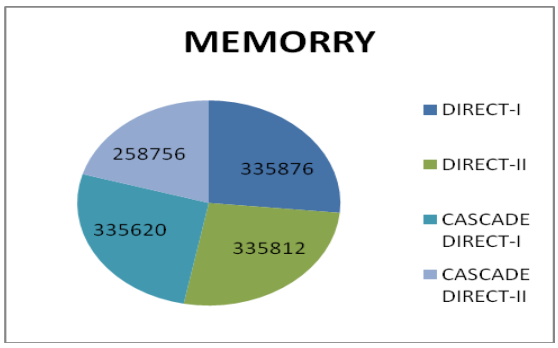

Fig. 18: Area performance of different architectures

Table I: Power Performance of different architectures.

FREQUENCY	DIRECT-I	DIRECT-II	CASCADE (Direct-I)	CASCADE (Direct-II)
100MHz	0.254	0.247	0.228	0.229
200MHz	0.295	0.286	0.271	0.298
500MHz	0.377	0.396	0.421	0.471
1000MHz	0.532	0.565	0.648	0.728

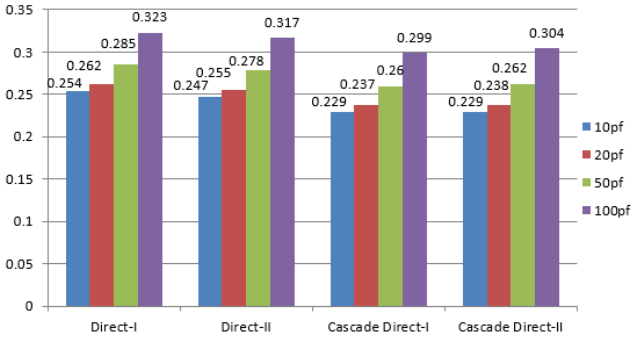

Fig. 19: Power Performance of different architectures.

6. Conclusion

Efficient algorithms are realized to implement/verify high-order filters concerning power consumption, area, and delay constraints. In conclusion, the realization of digital IIR filter architectures offers a powerful and versatile solution for various signal-processing tasks. Digital IIR filters can be implemented on various platforms ranging from general-purpose processors (GPUs, DSPs) to dedicated hardware (FPGA, ASIC). By understanding the characteristics, design principles, and implementation considerations of IIR filters, engineers can effectively leverage them to meet the requirements of diverse applications in the fields of telecommunications, audio processing, biomedical signal analysis, and control systems Direct Form-I realization technique is best suitable to implement high order filter when power is a major constraint. On the other hand, the direct form-II realization technique is best suitable for implementing high order filter when area and speed a major constraint.

References

[1] Implementation and simulation of IIR digital filters in FPGA using Matlab system generator IEEE 2014.

[2] IIR filters using Xilinx System Generator for FPGA Implementation Vol. 2, Issue 5, September-October 2012, pp.303-307, IJERA.

[3] Direct form-based pipelined IIR filter realization P. Boonyanant; S. Tantaratana IEEE. APCCAS 1998. 1998 IEEE Asia-Pacific Conference on Circuits and Systems.

[4] Microelectronics and Integrating Systems. Proceedings (Cat. No.98EX242) Pages: 85 - 88,

[5] Some comparisons between fir and air digital filters Published in: The Bell System Technical Journal (Volume: 53, Issue: 2, Feb. 1974) Page(s): 305 - 331 April 2014

[6] Comparison of two methods for the realization of multiplier-less elliptic IIR filters, IEEE April 2015.

[7] Comparison of IIR filter structure complexities using multiplier blocks, IEEE 06 August 2002

[8] A.G. Dempster and M.D.Macleod, "Multiplier blocks and complexity Of IIR structures", Electron. Lett., vol. 30, pp.1841-1842, 1994.

[9] M.D. Lutovacand L.D. Milic, "Design of computationally efficient elliptic IIR filters with a Reduced number of shift-and-add operations in multipliers IEEE Trans. Signal Processing, vol.45, pp.2422-2430 Oct.1997.

[10] A.G. Dempster, "Digital filter design for low-complexity implementation", Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., June 1995.

[11] Prokis J.G., Manolakis D. G., Digital Signal Processing., 3rd Edition, PHI publication 2004.

[12] Prokis J.G., Manolakis D. G., Digital Signal Processing., 3rd Edition, PHI publication 2004.

[13] K.Babulu. M.Kamaraju., P.Bujjibabu, K.Pradeep ”Design and implementation of H/W efficient Multiplier: Reversible logic gate approach” presented at the IEEE ICCSP 2015 conference DOI: 978-1-4799-8081-9/15/$31.00 © 2015.

[14] P.Bujjibabu, V.Satyanarayana, G.Jyothirmai, GNPK Mahalakshmi. E ”Low Power VLSI: High-End Design Techniques” presented at the International Journal of Scientific & Engineering Research, volume 4, issue 8, July 2013 ISSN 2229-5518.

Author’s biography:

Bujjibabu Penumutchi working as an Assistant Professor in the department of ECE at Aditya University (ADU), Surampalem while also pursuing an external Ph.D. from JNTUV, Gurajada Vizianagaram. He received his Master’s (M. Tech-VLSID) from the Adity Engineering College (JNTUK Kakinada) after graduating from the AITAM-Tekkali (JNTUH Hyderabad). His primary areas of competence are low-power VLSI design, digital IIR filter topologies design, logic realization, and optimization.

Corresponding author e-mail:

bujjibabu_penumuchi@adityauniversity.in

Dr. Kamaraju Maddu teaches Electronics and Communication Engineering at the Gudlavalleru Engineering College in Gudlavalleru. In addition to having over 155 academic journal and conference articles to his credit, he has extensive research experience in low-power VLSI design, Embedded Systems, Microprocessors, Microcontrollers, and Digital System Design. He is the Mentor right now (AS & A). He has directed numerous productive knowledge transfer programs and was the principal investigator for numerous funded internal and external grant applications. Several financed research initiatives also included him as a co-investigator.

E-mail: profmkr@gmail.com

Dr Babulu Kaparapu, Professor in the ECE department at the Gurajada-Vizianagaram University College of Engineering. Currently serving as both Director of Admissions and Director of Research & Development (I/C). Dr. Babulu Kaparapu earned his B.E. in ECE at Andhra University in Visakhapatnam, his M. Tech. in EI at REC Warangal, and his Ph.D. in VLSI & ESD at JNTUA in Anantapur. Dr. Babulu was able to publish his research contributions in numerous national and international Journals thanks to his instrument knowledge in the field of electronics. He has published more than 80 papers in national and international journals, made a substantial contribution to conferences, and participated in knowledge-sharing sessions. His main research interests are embedded system design, low-power VLSI design, and memory design.

E-mail: kapbbl@gmail.com