

RESEARCH

Open Access



High performance IIR filter implementation on FPGA

Debarshi Datta^{1*} and Himadri Sekhar Dutta²

*Correspondence:

debarshidatta7@gmail.com

¹ Department of Electronics and Communication Engineering, Brainware Group of Institutions, Barasat, Kolkata, India

Full list of author information is available at the end of the article

Abstract

This paper presents an improved design of reconfigurable infinite impulse response (IIR) filter that can be widely used in real-time applications. The proposed IIR design is realized by parallel-pipeline-based finite impulse response (FIR) filter. The FIR filters have excellent characteristics such as high stability, linear phase response and fewer finite precision errors. Hence, FIR-based IIR design is more attractive and selective in signal processing. In addition, the other two modern techniques such as look-ahead and two-level pipeline IIR filter designs are also discussed. All the said designs have been described in hardware description language and tested on Xilinx Virtex-5 field programmable gate array board. The implementation results show that the proposed FIR-based IIR design yields better performance in terms of hardware utilization, higher operating speed and lower power consumption compared to conventional IIR filter.

Keywords: FIR, IIR, High speed, Parallel-pipeline, FPGA

Introduction

The basic function of a filter is to remove unwanted portion of a given signal or suppress noise and to excerpt suitable parts of that signal for further processing [1]. Filters may be analog or digital types. The advantages of the digital filters have small size, high efficiency and the possibilities of a fast reconfiguration. Digital filters are more accuracy and compared with the analogical circuits. Digital filters are typically two types; one finite impulse response (FIR) and another infinite impulse response (IIR). Generally, FIR filters are stable, linear in phase, fewer finite precision errors and efficiently implemented [2]. But its computational complexity is several times larger than IIR filter. Additionally, IIR filter requires fewer coefficients and less storage space. Furthermore, IIR filters offer better magnitude responses and it closer to analog models. Hence, the realization of IIR filter requires less number of multiplier units as comparable with FIR filter. Therefore, IIR filters are widely used in high speed systems. The frequency characteristic of a digital IIR filter is a distinct version like analog type filter frequency response [3].

Implementation of IIR filter on field programmable gate array (FPGA) provides high throughput with effective utilization of hardware, flexibility and reaching high speed of calculating accuracy [4]. In this context, implementation of digital IIR architecture on FPGA is very much useful to achieve high performance system [5].

During last few years, various types of digital filters have been discussed on FPGA environment to measure the area, power and speed. Recently, Seshadri et al. [6] have described fast FIR, and IIR filters which are realized by look-ahead arithmetic technique. The designs have successfully validated in FPGA device and achieved a speed of 220 MHz. Again, Yuan et al. [7] have discussed a new approach of FIR filter design using high-accuracy non-scaled stochastic adder so that the design has improved computation accuracy with less hardware resources. Similarly, De Cacqueray- Valmenier et al. [8] have explained polyphase realization of IIR filters for implementing in wireless communication systems. This paper specially focused on the Discrete Fourier Transform (DFT) filter banks which are implemented by IIR filters instead of FIR filters. Again, S. Islam et al. [9] proposed gate level IIR architecture is implemented on FPGA board and analysed its performance using impulse response. Similarly, A. Paul et al. [10] developed the digital FIR filters in direct form and transposed form. The said filters are implemented on FPGA board and discussed the synthesis parameters. N. Wong et al. [11] proposed a new design of IIR structure using vector fitting algorithm. The design has achieved fast convergence and accurate IIR approximation. D. S. Sidhu et al. [12] discussed IIR filter design using hybrid gravitational search algorithm (HGSA) to minimize magnitude response error, linear phase response error with matching the stability criterion. B. Singh et al. [13] proposed digital IIR filter design using DE hybridized and pattern search technique. DE method is used for global search technique, and pattern search is needed for local search technique. Again, Y. Yu et al. [14] have discussed cooperative evolutionary genetic algorithm for IIR filter design to optimize the magnitude and phase response using minimum filter order. Wang et al. [15] discussed local search operator which enhanced multi-objective evolutionary algorithm (LS-MOEA) for designing of IIR filter with multiple objectives. Again, Kaur et al. [16] developed a real coded genetic algorithm (RCGA) for IIR filter design. Similarly, A. Sergiyenko et al. [17] discussed digital filters realization on FPGA device.

This paper focuses on design and implementation of IIR filters on FPGA device. This paper briefs three types of IIR filters. They are lossy integrator-based look-ahead IIR filter, two-level parallel-pipeline IIR filter and proposed FIR-based IIR filter. All the designs have been coded and simulate in Xilinx FPGA [18]. The FPGA implementations of the IIR architectures are easier to debug with optimum hardware resources. The proposed FIR-based IIR design allows to reduce computation time and to optimize the available hardware resources in the FPGA platform.

This work is organized as follows: “[Proposed IIR filter implementation](#)” section describes various IIR filters realization; then “[FPGA implementation](#)” section is dedicated to FPGA implementation and finally, conclusion is made in final section.

Proposed IIR filter implementation

The IIR filter can operate with specific number of filter order. Basically, digital IIR filter consists of feedback topology, and it is capable of realizing both zeros and poles of a system transfer function, whereas the FIR filters are all-zero filter.

For implementation of an N th order IIR digital filter, architecture requires $(2N+1)$ number of coefficients, $(2N+1)$ number of multipliers, $(2N)$ number of two-input

adders and N number of registers. The linear difference equation for this IIR filter is as follows [2]:

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

Thus, the output $y[n]$ of a digital IIR filter depends upon present input and previous output. Naturally, IIR filter requires very few number of multiplication blocks in comparison with FIR filter design and so that, the realization of IIR filters have improved in real-time systems.

IIR filter can be realized into four methods, explicitly, direct structure, canonic structure, cascade structure and parallel structure [2]. The key factors that mark the select of a digital IIR filter for a particular realization are being computational complexity, memory storage and finite word-length conditions. However, the cascade and parallel realizations of IIR filters are more robust than the direct and canonic realization filters, and they have better frequency characteristics which are closer to the desired responses. The following sub-sections describe three different techniques of IIR filter designs and they are look-ahead IIR filter, two level parallel-pipeline IIR filter and FIR-based IIR filter.

Look-ahead IIR filter

The realization of pipelining 1st order digital IIR filter topology with look-ahead technique is widely used in high speed systems [19]. Pipelining is a method in which multiple numbers of instructions are overlapped to make a high throughput. The fundamental idea of look-ahead pipelining is to add-cancel poles and zeroes to the transfer function such that the denominator coefficients are become zero. Pipelining technique is basically introduced a register or latch (D or Z^{-1}) between every sub-unit so that it can reduce the critical path delay. Therefore, it rises the clock speed or sample speed. However, the pipelining technique increases the system complexity with number of loops in pipelining stages. The pipeline-based 1st order digital IIR filter is always stable only if original IIR filter stable.

Consider the transfer function of 1st order IIR filter is as follows [2]:

$$H(z) = \frac{z^{-1}}{1 - az^{-1}} \quad (2)$$

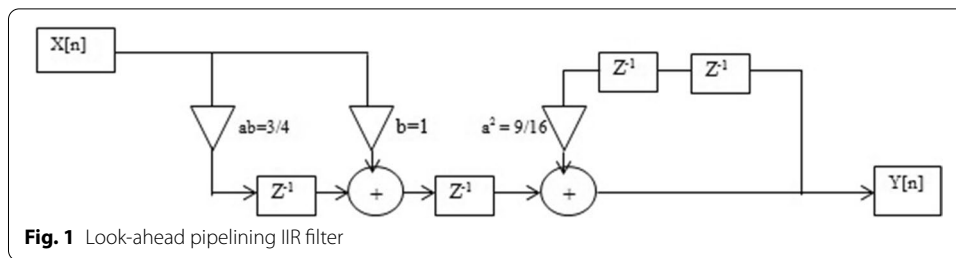
where the coefficient $|a| \leq 1$, for a stable system. The above transfer function has only one pole placed at $z=a$. Then the difference equation can be formed as [2]

$$y(n+1) = ay(n) + x(n) \quad (3)$$

Then,

$$y[n+2] = ay[n+1] + x[n+1] = a^2y[n] + ax[n] + x[n+1] \quad (4)$$

Figure 1 shows realization of look-ahead pipelining IIR filter with coefficient $a = \frac{3}{4}$.



Two-level parallel-pipeline IIR filter

The pipelining procedure can be used for improving the critical path computation which is limited by communication and at the terminal point pipelined technique does not increase the sample speed significantly. At this point parallel processing combined with pipelining is to improve the sample speed. In parallel processing technique, several outputs are taken at one clock period and as a result, the actual sample speed is improved substantially. Moreover, both parallel and pipelining techniques are dual each other, if one calculation has pipelined method then other should be accomplished by parallel approach. These two processes feat concurrency accessible in the computation for various methods. In pipelining system, individual set is computed in interleaved mode while in parallel processing, computations are performed in identical hardware process. Therefore, parallel-pipeline IIR filter is to attain high sample rate by a factor $L \times M$, where L represents the block processing levels and M signifies pipelining stages [20]. Then sample speed is given by

$$T_{\text{sample}} = \frac{T_{\text{clk}}}{LM} \tag{5}$$

Consider the design is realized by 2-pipeline ($M=2$) and 2-parallel ($L=2$) stages. The given filter order is one, and only one loop update operation is needed. In parallel processing system, each delay component is denoted as a block level, and two times sample period is the clock period of this block level. Therefore, the modified loop equation $y(n+2)$ should be updated on input $x(n)$ and output $y(n)$ sequence. The Eq. (4) splits into two parts into even and odd sequences [21].

Let, even sequence $n=2k$ and odd sequence $n=2k-1$.

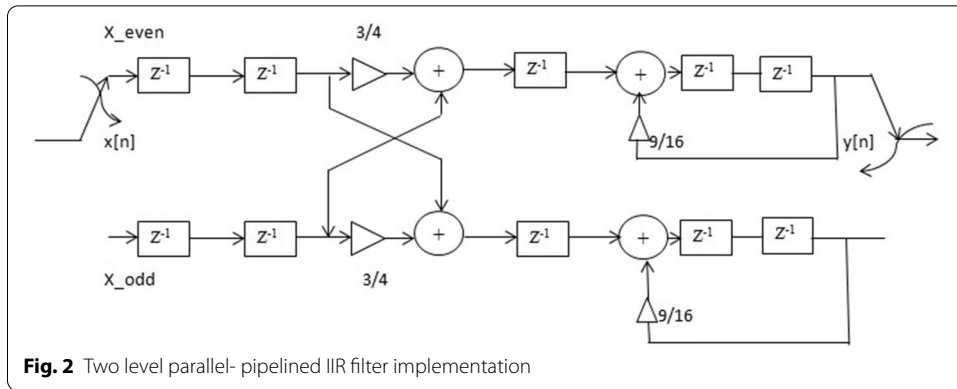
So that even equation is

$$y[n+2] = y[2k+2] = a^2y[2k] + ax[2k] + x[2k+1] \tag{6}$$

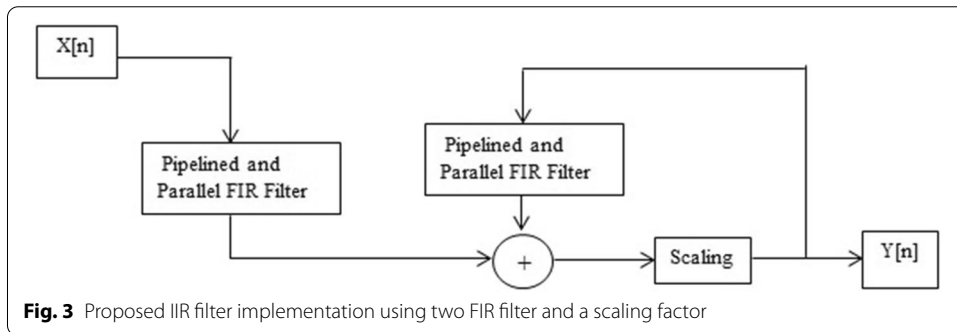
And the odd equation is

$$y[n+2] = y[2k+1] = a^2y[2k-1] + ax[2k-1] + x[2k] \tag{7}$$

The above two Eqs. (6) and (7) are the basis of the parallel-pipeline IIR filter implementation. Using these equations, the two level parallel-pipelined IIR architecture is realized and shown in Fig. 2. This structure is made of two non-recursive parts with coefficient of $\frac{3}{4}$, two unit delays of two recursive parts with coefficient of $\frac{9}{16}$.



The original system has pole at $z = a$, whereas in parallel the pole location is at $z = a^2$. This is closer to origin, since $|a^2| \leq |a|$ (since $|a| \leq 1$). This movement of pole is to develop the strength of the architecture to the round off noise.



Proposed FIR-based IIR filter

The above two systems suffer from some limitations due to inherent properties of IIR filter architecture. High throughput IIR filter can be achieved using parallel-pipeline FIR filters with a scaling factor, as shown in Fig. 3. Here, combined fine grain parallel-pipeline 3-tap two FIR filters make a high speed IIR filter architecture. This approach can significantly improves the performance of the IIR filter.

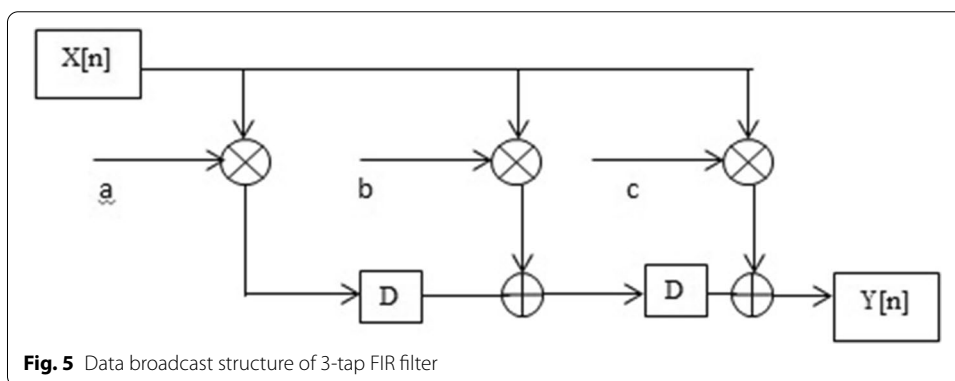
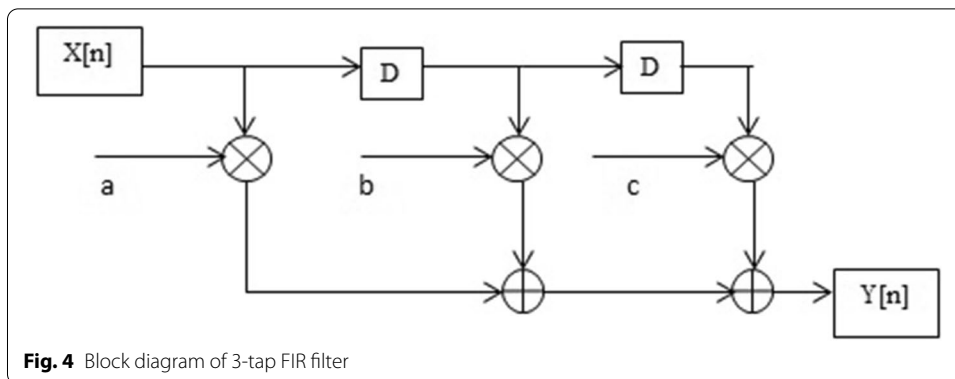
Figure 3 uses FIR filter which is described in the following method. Consider the 3rd order FIR filter equation is [22]

$$y[n] = ax[n] + bx[n - 1] + cx[n - 2] \tag{8}$$

The structure implementation of this 3rd order FIR filter is shown in Fig. 4.

The lowest time required for getting new sample or critical delay which is calculated from one multiplication and two addition operations. If T_M and T_A are the time taken for multiplication and addition operations, respectively, then sample period (T_{sample}) is [22]

$$T_{sample} \geq T_M + 2T_A \tag{9}$$



or sample frequency (f_{sample}) is given by

$$f_{\text{sample}} \leq \frac{1}{T_M + 2T_A} \tag{10}$$

In this structure, the data are broadcast to all the multipliers simultaneously. Hence, the critical path can reduce substantially. Therefore, critical path delay is calculated as $T_M + T_A$. This procedure increases the sample frequency. The realization of this filter is shown in Fig. 5.

The proposed design includes combined fine-grain, data-broadcast and parallel-pipeline processing of 3rd order FIR filter, shown in Fig. 6. The FIR filter structure has 3 parallel inputs and uses 2-pipelining delay for reducing the critical path. Fine grain pipelining in parallel filter can further reduce the critical path. In fine grain process, the multiplier unit is broken into two smaller units (m_1, m_2), and a latch is placed between the two multiplier units to achieve the high clock speed. Therefore, this combination process reduces the sample period by [22]

$$T_{\text{sample}} = \frac{1}{6}(T_M + T_A) \tag{11}$$

The pipelining and parallel processing combine techniques are used for lower power consumption. The pipeline technique decreases the capacitance which is charged or discharged in one clock cycle whereas parallel technique is increased the clock cycle for

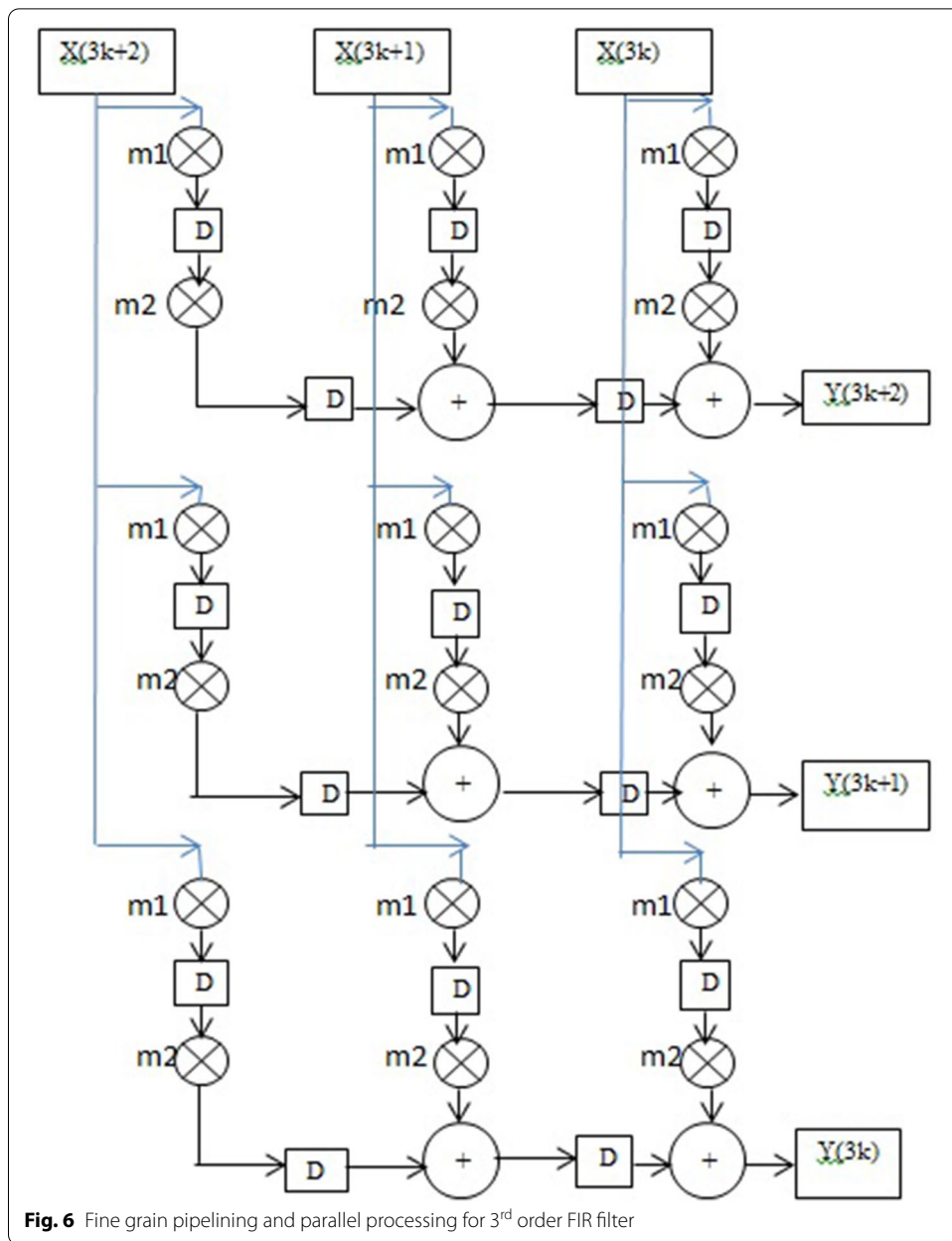


Fig. 6 Fine grain pipelining and parallel processing for 3rd order FIR filter

charging or discharging the capacitance. The reduction of power consumption is due to the clock lines as compare with a pipeline system which needs to be operated using a high speed clock for same throughput or sample speed.

FPGA implementation

All the given architectures are being described HDL, and XILINX ISE Design Suite 14.7 is used for synthesis. Virtex-5 XC5VLX50T FPGA board (Speed Grade -3) is the target device for implementation designed architectures [23]. The maximum clock frequency is limited to 300 MHz. However, the operating speed of digital filter decreases with increasing the filter

Table 1 Device utilization summary of different IIR filters

Synthesis parameter	Look-ahead IIR filter	Parallel IIR filter	Proposed FIR-based IIR filter
Number slice registers	180	192	172
Number of slice LUT	145	135	120
Number of IOBs	31	45	35
BRAMs	8	10	8
DSP48Es	18	20	16
Power (W)	0.150	0.180	0.120
Speed (MHz)	172.280	245.306	285.105

Table 2 Comparison of Seshadri et al. [6] architecture and proposed FIR-based IIR design

Architecture	Slice registers	Power (W)	Speed (MHz)
R. Seshadri et al. [6]	230	0.140	220
Proposed FIR-based IIR design	172	0.120	285.105

word length. The proposed IIR filter has been synthesized, and then generated bit streams have been downloaded on FPGA device. The hardware utilization summary such as slices, LUTs, IOBs, and maximum working frequency can be obtained in the compilation report. Table 1 shows the results of synthesis report of different types of IIR filters. The results exhibit that the proposed FIR-based IIR filter implementation requires less slice registers. The FIR-based IIR method reduces power and enhances the maximum operating speed.

A comparison result is discussed in Table 2 with key parameters like Slice Registers, power and operating speed. The proposed design method reduces area by approximately 33.72% which leads to power optimization and enhances the maximum frequency. Hence, the proposed design is more energy efficient as compare to other architecture. As a result, the implementation of proposed FIR-based IIR architecture is more attractive and has improved in real-time signal processing.

Conclusion

This paper describes design and implementation of reconfigurable IIR filters. The development of the different types of IIR filters has been successfully implemented in FPGA for analysing the performance. The results obtained in the IIR filter tests, related to response times are very satisfactory, emphasizing the processing levels of the reconfigurable architectures, and overall higher performance is obtained using pipeline and parallel technology. The proposed FIR-based IIR filter has increased the sampling speed, and also at same time, it reduces power consumption. Comparison results show that the proposed FIR-based IIR filter has achieved maximum operating speed of 285.105 MHz with optimum power and area. The proposed solutions can be developed in real time signal processing systems.

Abbreviations

IIR: infinite impulse response; FIR: finite Impulse Response; HDL: hardware description language; FPGA: field programmable gate array; DFT: discrete Fourier transform; HGSA: hybrid gravitational search algorithm; LS-MOEA: local search operator enhanced multi-objective evolutionary algorithm; RCGA: real coded genetic algorithm.

Acknowledgements

The authors would like to thank the Department of Electronics and Communication, Brainware Group of Institutions and Kalyani Govt. Engineering College for providing the ISE design suite 14.7 and Virtex-5 FPGA board.

Authors' contributions

All the authors listed on the title page have contributed significantly to the work, have read the manuscript, attest to the validity and legitimacy of the data and its interpretation and agree to its submission. The first author has prepared proposed architecture, tables, figures, results and body of the manuscript. The second author or co-author has describes abstract, literature survey discussion and conclusion sections. Both authors read and approved the final manuscript.

Funding

There has been no significant financial support for this work.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests associated with this work.

Author details

¹ Department of Electronics and Communication Engineering, Brainware Group of Institutions, Barasat, Kolkata, India.

² Department of Electronics and Communication Engineering, Kalyani Government Engineering College, Nadia, India.

Received: 11 September 2019 Accepted: 3 December 2020

Published online: 06 January 2021

References

- Shenoi B (2006) A (2000) Introduction to digital signal processing and filter design. Wiley-Inter-science, Hoboken
- Proakis GJ, Manolakis GD (2013) Digital signal processing: principles, algorithms and applications. Pearson Education, Singapore
- Meddins B (2000) Introduction to digital signal processing. Newnes, Oxford
- Woods R, McAllister J, Yi Y, Lightbody G (2008) FPGA-based implementation of signal processing systems. Wiley, Chichester
- Xilinx white paper WP330 (2009) Infinite Impulse Response Filter Structures in Xilinx FPGAs
- Seshadri R, Ramakrishnan S (2019) FPGA implementation of fast digital FIR and IIR filters. Wiley, Chichester. <https://doi.org/10.1002/cpe.5246>
- Yuan B, Wang Y (2016) High-accuracy FIR filter design using stochastic computing. In: IEEE Computer Society annual symposium on VLSI (ISVLSI); Pittsburgh, PA. <https://doi.org/10.1109/ISVLSI.2016.63>
- De Cacqueray-Valmenier M, Coskun A, Kale I (2016) The use of almost linear phase IIR filters in DFT modulated filter banks for communication systems. In: 2016 proceedings of the international workshop on signal processing advances in wireless communications, Edinburgh, UK. <https://doi.org/10.1109/SPAWC.2016.7536754>
- Islam S, Sarker R, Saha S (2012) Design of a programmable digital IIR filter based on FPGA. In: International conference on informatics, electronics and vision, Dhaka, Bangladesh
- Paul A, Khan ZT, Poddar PP, Hasan M, Ahmed T (2015) Reconfigurable architecture design of FIR and IIR in FPGA. In: International conference on signal processing and integrated networks (SPIN), IEEE Xplore, India. <https://doi.org/10.1109/SPIN.2015.7095408>
- Wong N, Lei UC (2008) IIR approximation of FIR filters via discrete-time vector fitting. IEEE Trans Signal Process 56(3):1296–1302
- Sidhu SD, Dhillon SJ, Kaur D (2015) Design of digital IIR filter with conflicting objectives using hybrid gravitational search algorithm. Math Problems Eng 2015, 282809. <https://doi.org/10.1155/2015/282809>
- Singh B, Dhillon SJ, Brar SY (2013) A hybrid differential evolution method for the design of IIR digital filter. ACEEE Int J Signal Image Process 4(1):1–10
- Yu Y, Xinjie Y (2007) Cooperative co-evolutionary genetic algorithm for digital IIR filter design. IEEE Trans Ind Electron 54(3):1311–1318
- Wang Y, Li B, Chen Y (2011) Digital IIR filter design using multi-objective optimization evolutionary algorithm. Appl Soft Comput 11(2):1851–1857
- Kaur R, Patterh SM, Dhillon SJ (2014) Real coded genetic algorithm for design of IIR digital filter with conflicting objectives. Appl Math Inf Sci 8(5):2635–2644
- Sergiyenko A, Serhienko A (2018) Digital filter design using VHDL. In: Fifth international conference "high performance computing" HPC-UA
- Brown S, Vranesic Z (2008) Fundamentals of digital logic with VHDL design, 2nd edn. McGraw-Hill Higher Education, New York
- Meyer-Baese U (2007) Digital signal processing with field programmable gate arrays. Springer, Berlin
- Parhi KK (1999) VLSI digital signal processing systems. Wiley, New York
- Virtex-5 FPGA Data Sheet: DC and Switching Characteristics DS202 (v5.5) June 17, 2016

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.