An Efficient Implementation of Programmable IIR Filter for FPGA

L. Babitha, U. Somanaidu, CH. Poojitha, K. Niharika, V. Mahesh, and Vallabhuni Vijay

1 Introduction

The system which has an impulse response for a limitless amount of time is observed as infinite impulse response (IIR) filters. It is one of the filters which is digital which consists of adders, delay, MCU, SoC, and processors [1]. Presently, these filters are widely being used for diverse applications including communications which includes messaging, signal processing of the video and audio, etc. The digital signal processing systems which are modern have a wide number of usages for these filters [1]. When compared to the FIR filters, these filters have fast computation speed, and the complexity of these filters is also less. These filters work in real time and are faster compared to the others. In IIR filter, the output depends on past inputs, present inputs, and the previous outputs.

The transfer function can be written as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{P} b_i z^{-i}}{\sum_{j=0}^{Q} a_j z^{-j}} \tag{1}$$

It is unstable and consists of both poles and zeroes. It depends on the previous filter, and it is called recursive filter. It does not have a linear phase character. IIR filters are designed using analog filters such as Butterworth filter, Chebyshev filter, and also the elliptical filter [2].

L. Babitha · U. Somanaidu · CH. Poojitha · K. Niharika · V. Mahesh · V. Vijay (✉)
Institute of Aeronautical Engineering, Dundigal, Hyderabad 500043, India
e-mail: v.vijay@iare.ac.in

L. Babitha
e-mail: babitha.l@iare.ac.in

U. Somanaidu
e-mail: u.somanaidu@iare.ac.in

J. K. Mandal et al. (eds.), *Innovations in Signal Processing and Embedded Systems*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1669-4_10

In order to design IIR filter, we use methods such as impulse invariant method, matched Z-transform, and bilinear transformation. These methods are used to design low-pass IIR filters. In order to design high-order IIR filter, filter stages of all pass filter, the delays, and the masking filter methods.

The frequency components of the input signals are allowed by the all pass filter which does not have any attenuation, and the all pass filter will also provide different frequency of predictive phase shifts of the input signal. All pass filters are also named as delay equalizers or phase correctors [3]. The amplitude of an all pass filter is unity. Phase changes from 0 to 360°. Applications such as communications, when the transmission lines send the signals from one point to another point they undergo phase change. The all pass filters will be utilized to compensate the phase change.

All pass filters are generally used for altering the phase response of IIR filter without effecting magnitude response. The IIR filters which are of all pass contain the properties which are beneficial which includes the number of multiplications which will be reduced, the characteristic of the phase in pass band which is highly linear and the group delay which will be cut down [3].

For example, we can get the IIR filter which is of high pass from the low-pass IIR filter [4–11].

$$H_l p(z) = \frac{1 * (E_0(z) + E_1(z))}{2} \tag{2}$$

$$H_h p(z) = \frac{1 * (E_0(z) - E_1(z))}{2} \tag{3}$$

To implement the IIR filter based on all pass filter, it is designed in MATLAB [2]. In this paper, the implementation of is done by using a very high-speed hardware description language simply known as VHDL. For FPGA, the description for the application specific digital structure can be done by using VHDL [1]. It has very vast library packages. This will be used in every step of the process. The processing of the data for the mathematical equations is very effective using the VHDL language and also for calculating the frequency response and for the processing of numbers. The searching for the coefficients and filtering the characteristics of the filter becomes faster and easier using this. When comparing to the other tools which are used for design, the VHDL provides a lot of features, the implementation will be featured in the FPGA, and the optimization will be more enhancing [2]. There are many advantages to the VHDL language. Some of them are the portability which allows the description of the device to be used on other tools, the flexibility which allows the description of the code of complex logic, it is independent of any device which means that the code can be run on any device irrespective of the operating system, and also it is very time efficient.

1.1 Introduction to FPGA

The IIR filters have been used widely and have many applications for their flexibility, and the cost of the filter and speed is variably suppressed when it comes to high-order filters [2]. So FPGA is implemented to achieve the results that we want to get. The field programmable gate array which can also be referred to as FPGA. This is one of the several styles of design of the VLSI technology. These contain a huge number of the logic gates and the interconnects which are programmable. The major applications of these are in wireless communication, medicine, electronic devices, etc. For the applications which are of the type of low volume, they provide prototyping which is considerably very fast and also a very effective designing of the chip. These provide the sampling rate which is variably higher than those of the older versions of the DSP chips. In this, the device's configuration can also be altered according to our use. In this, the duration of the design is shorter which gives an advantage [1].

The FPGA will result in many advantages to the filter and also makes the implementation easier and is mostly used to lessen the overall cost of the filter. In this, the resources for the internal logic are various the configurations for applications which give high performance are magnificent. They type of the filter which is IIR filter taken in the FPGA will be concatenated. The IIR filter which is implemented using FPGA will give high throughput with efficient utilization. In this, FPGA a set of coefficients will be searched for limitations of the frequency response which will be used as base [12]. The filter will be concatenated VHDL or Verilog is used to describe the model of the coefficients which are built-in which is supported by the FPGA. In several applications which include communications which require high-speed FPGA is the only resolve for the IIR filter [12].

2 Proposed Work

IIR filter is also developed by frequency response techniques. It is used to design arbitrary-band filter and narrow band filter with different specifications. Each of the delay will be changed by "m" delays as it is the basic principle of the frequency masking [3].

It is retrieved by connecting the periodic model filter and masking filter. FRM approach is mostly applicable for modeling the narrow band and arbitrary-band filters featuring with sharp filters with tiny word length effects [1]. Series connection of periodic model filter $G(n^m)$ and masking filter($S(n)$) [12].

$$K(n) = G(n^m) \cdot S(n) \tag{4}$$

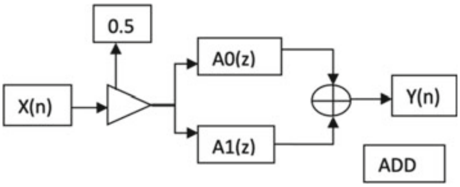

Fig. 1 All pass implementations of low-pass IIR filters

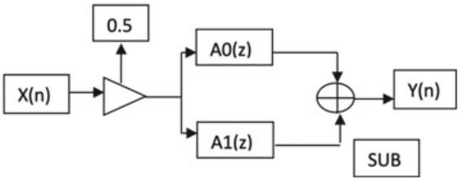

Fig. 2 All pass implementations of high-pass IIR filters

The images produced by the $G(n^m)$ are eliminated by masking filter. $G(n^m)$ is obtained by $G(n)$. This $G(n)$ is obtained from all pass functions where

$$G(n) = Go(n) + G1(n)/2 \tag{5}$$

$$Gc(n) = Go(n) - G1(n)/2 \tag{6}$$

As the above equations state the low-pass filter model for the $G(n)$ and the other equation states the high-pass complementary model filter for the $Gc(n)$.

From the above equations, $G(n)$ is the function of the all pass filter and $Gc(n)$ is complementary all pass function, and diagrams will be same as Figs. 1 and 2. The overall filter of the transition band will be N number of times smaller when compared to the model filter. $S(n)$ will be half band FIR filter which is a masked filter which helps in masking [13] (Fig. 3).

It is the magnitude response of the arbitrary bandwidth IIR filter.

The complementary pair should satisfy the property is

$$|G(n) + Gc(n)| = 1 \tag{7}$$

The steps to design are

a. Obtain the half band filter design using MATLAB default code (buffer). Half band IIR filter is used to achieve fastness in computation, consuming of less power and miniaturization.
b. Find and display the coefficients of all pass branches.
c. Define all pass branches and display the frequency response using freqz

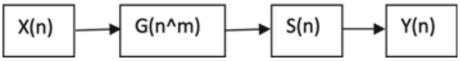

Fig. 3 Block diagram for masking approach

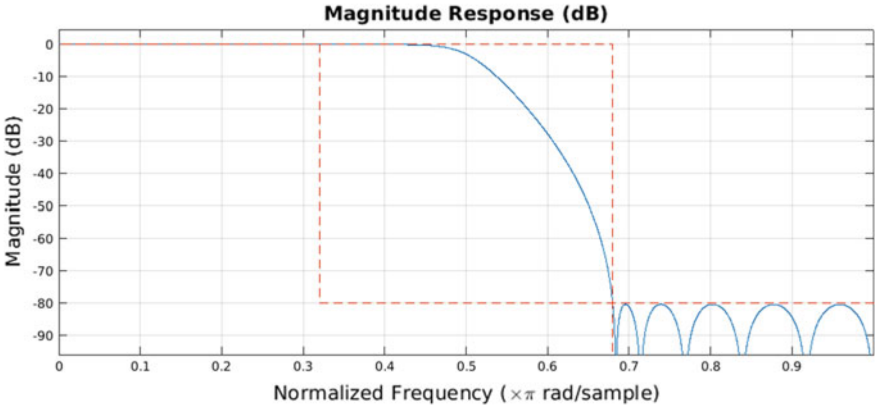

Fig. 4 IIR filter magnitude response

d. Form and compute frequency response of the equations G (n) and $Gc(n)$.

e. Replace the delays with m and form masked FIR filters ($Fo(n)$ and $F1(n)$) which are used to remove the unwanted spectra.

f. Compute frequency response of

$$H(n) = G(n^m) * Fo(n) + Gc(n^m) * F1(n) \tag{8}$$

When in contrast with the FIR filters, the IIR filters have minimum complexity, and the effectiveness of the filtering will be greater. The usage of the FPGA system is limited and is less as its throughput is reduced and has more sensitivity due to rounding of the coefficient. Due to the path length which is critical, the speed of the filter is limited to a certain amount (Fig. 4).

The algorithm of the IIR filter is visualized by the synchronous data flow which consists of coefficient multipliers, delays of the registers, and the circles which are of the multipliers and adders. It consists of 4 multiplier units but only the adder and the multiplier have the crucial path [13]. The clock frequency which is maximum is very much larger in the other signal flow graphs compared to the stages of all pass, and the pipelining is maximum for the signal flow graph (Fig. 5).

In order to reduce the critical path, the multiplication to coefficient Ci in the above diagram should have minimized delay. We need to find out the Ci values. Signal flow graphs optimization techniques like retiming, pipelining, and folding are not able to minimize the delay [14]. The integration of the filter which used the multiplied delay method, and the masked filter is a complex optimization method which cannot reduce the delay [1]. By replacing the multipliers which are of hardware to a different address, we can improve the functionalities of the IIR filter in FPGA and also make it run a lot faster.

In order to limit the problem, a Verilog code is programmed based on signal flow graph of 5th order low-pass filter. Through this, we can easily implement the IIR filter in FPGA for a better pace. In order to limit the problem, a Verilog code is

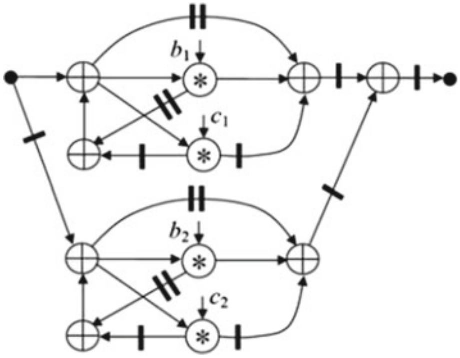

Fig. 5 SDF for low-pass filter of order $N = 5$

programmed based on signal flow graph of 5th order low-pass filter. Through this, we can easily implement the IIR filter in FPGA for a better pace. Pipelining is one of the techniques which helps in reducing the crucial path between the input and output. In this process, it computes the time carried by the crucial path, time carried by the adders and the multipliers. The path time must be greater than the addition of twice of adders and multipliers. Based upon that latches are introduced to reduce the delay.

3 Simulation Result

Figure 6 shows the schematic view of the filter which consists of the inputs and the outputs and the reset and the clock. This is the RTL design acquired from VHDL simulator. In this filter, the number of inputs and outputs are the same. In this filter, the inputs are of 31, and the outputs are of 31. Additional to the input and the output, the filter consists of clock and reset (Fig. 7).

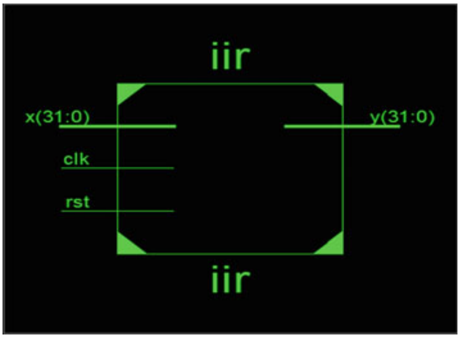

Fig. 6 Schematic view of IIR filter

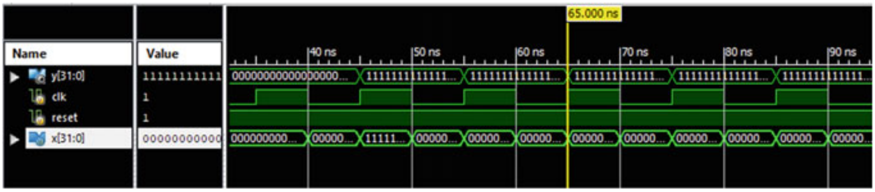

Fig. 7 Waveforms of IIR filter

4 Comparative Analysis

The table states about the filter attribute which are created based on the performance of proposed method and existing methods. Existing method is implementation of IIR using all pass branches and masked approaches. Proposed method is implementation of IIR based on SDF and pipelining. It states about the attributes regarding the filter which are created based on the performance of filter. In the above table, we have computation rate, critical path, and the complexity of the IIR filter. As we can see from the table, the computation rate increases by a variably high amount which makes the working of the filter faster, and the functionality of the filter will be very high and also the computations which are very complex can be done much faster and easier, and the consumption of the power decreases which makes the device to work longer with no interruption and the main important feature, and the complexity of the filter is reduced by a great amount which is the main purpose of this paper (Table 1).

5 Conclusion

In this paper, it is displayed that the critical path is recovered. An IIR filter is programmed for efficient implementation in FPGA. The high-quality IIR filters can be derived from all pass sections and masking methods. We can gain the characteristics such as less sensitivity of the coefficients, minimal consumption of power, and faster speed of computation when masking approach is performed. The problem of mapping data flow graphs (DFG) of infinite impulse response (IIR) filtering algorithms into application specific structure is considered. Methods of optimization of DFGs are considered for the purpose of finding IIR filter structures with the high throughput and hardware utilization. Optimization method is proposed which takes into account of the structural properties of FPGA, minimize its hardware volume, and provide the designing pipelined structures with high-clock frequency.

Table 1 Analysis of methods

Methods	Critical path	Rate of operation	Complexity
Implementation of IIR using all pass branches and masked approaches	Critical path is increased. It is caused due to the high delay	Due to increase in critical path, the speed factor is effected	In this method, we need to select the no. of stages, u or m factor. It is not suitable to implement IIR filter beyond order of 10
Implementation of programmable IIR based on SDF and pipelining	Critical path is reduced by using the pipelining	The rate of operation is increased due to decrease of maximum delay between the input and output	There is no need to choose the no. of stages and m factor to construct IIR filter, so there is less complexity

References

1. A. Sergiyenko, A. Serhienko, VHDL generation of optimized IIR filters, in *IEEE 2nd Ukraine Conference on Electrical and Computer Engineering*, UKRCON, pp. 1171–1174 (2019)
2. S.K. Mitra, K. Hirano, Digital all-pass networks, in IEEE Trans. Circuits Syst. **5**, CAS-21, pp. 688–700 (1974)
3. D. Datta, H.S. Dutta, High performance IIR filters implementation on FPGA. J. Electr. Syst. Inf. Technol. **8**(1), 1–9 (2021)
4. N.K. Gugilla, C. S. Dudha, Synthesis, in *Designing with Xilinx FPGA using Vivado*, ed. by S. Churiwala (Springer, 2017), pp. 97–110
5. V.I. Markova, J. Yli-Kaakinen, T. Saramaeki, An algorithm for the design of multipliers IIR filters as a parallel connection of two all pass filters, in *IEEE Asia Pacific Conference on Circuits and Systems*, *APPCCAS* (2006), pp. 744–747
6. R. Kaur, S.M. Patterh, S.J. Dillon, Real coded genetic algorithm for design of IIR digital filter with conflicting objectives. Math. In-Sci. **8**(5), 2635–2644 (2014)
7. C.S. Pittala, J. Sravana, G. Ajitha, P. Saritha, K. Mohammad, V. Vijay, S.C. Venkateswarlu, R.V. Rajeev, Novel methodology to validate DUTs using single access structure, in *5th International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech 2021)*, Kolkata, India, 24–25 Sept 2021
8. V. Vallabhuni, C. Kancharapu, T.S. Jaideep, D.R.K. Koushik, B.S. Venumadhav, V. Rajeev Ratna, Design of optimum multiplexer in quantum-dot cellular automata, in *International Conference on Innovative Computing, Intelligent Communication and Smart Electrical systems (ICSES-2021)*, Chennai, India, 24–25 Sept 2021
9. V. Vallabhuni, C.V. Sai Kumar Reddy, P. Chandra Shaker, V. Rajeev Ratna, M. Saritha, M. Lavanya, S. China Venkateswarlu, M. Sreevani, ECG performance validation using operational transconductance amplifier with bias current. Int. J. Syst. Assur. Eng. Manag. ISSN: 0975-6809
10. S. Swathi, S. Sushma, C. Devi Supraja, V. Bindusree, L. Babitha, V. Vallabhun, A hierarchical image matting model for blood vessel segmentation in retinal images. Int. J. Syst. Assur. Eng. Manag. (2021). ISSN: 0975-6809
11. R.R. Vallabhuni, M. Saritha, S. Chikkapally, V. Vijay, C.S. Pittala, S. Shaik, Universal Shift register designed at low supply voltages in 15 nm CNTFET using multiplexer, in *Advanced Techniques for IoT Applications. EAIT 2021*. Lecture Notes in Networks and Systems, ed. by J.K. D. De Mandal, vol. 292 (Springer, Singapore, 2022). https://doi.org/10.1007/978-981-16-4435-1_58
12. M. Ljiljana, *Multirate Filtering For Digital Signal Processing Matlab Applications Book* (University of Belgrade, Serbia, 2009), pp. 228–239

13. M. Ljiljana, *Multitrate Filtering For Digital Signal Processing Matlab Applications Book University of Belgrade*, Vol. 8 (2009), pp. 259–271
14. M. Ljiljana, *Multirate Filtering For Digital Signal Processing Matlab Applications Book* (University of Belgrade, Serbia, 2009), pp. 136–168