JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Architecture based IIR Filter Design and Performance evaluation for DSP APP

Korangi. Ramya Lakshmi Sai Samhitha
ECE Department
ISTS Women Engineering College
East Gonagudem, Andhra Pradesh

Mr. M.V.D Pavan Kumar, Ast. professor
ECE Department
ISTS Women Engineering College,
East Gonagudem, Andhra Pradesh

***Abstract*—**

A relatively recent innovation in the domain of digital filtering has been the introduction of signal processing applications with effective power utilization. No scope towards the effective and efficient architecture realization with existing part of literature, and hence investigation is made and is confined with cascade form of filter design. In this project, architectures are realized based on common set of specifications to arrive at the high-performance recursive filter for low power applications using Xilinx System Generator. Infinite Impulse Response (IIR) filters can be realized in many forms those are Direct form-I, Direct form-II, Cascade, parallel form. All these structures provide a space for selection of an appropriate architecture for reduction of power consumption and improvement in speed of digital filters. In this particular work, a 5th order low pass IIR filter is realized in Direct form-I, Direct form-II, Cascade form (Direct form-I/ Direct form-II) as an example of the methodology in a Xilinx FPGA device. Also corresponding power analysis was performed. As per power and timing analysis reports at high frequency range example at 100MHz Direct-II form consumes 0.565W and Direct-I form consumes 0.532W, at high loads Direct-I form consumes 0.317W, Direct-II form consumes 0.323W. Cascade form is mostly used to implement high order filters. When a filter is implemented by cascade form direct-I form consumes 0.648W, direct-II form consumes 0.728W.from these results i finally concluded that cascade (Direct form-I) realization is the best technique to implement higher order IIR filters when power is a main constraint, cascade (Direct form-II) technique is best with area and speed as constraints.

***Index Terms*— IIR filter, Direct form, Cascade form, Xilinx System generator.**

I. INTRODUCTION

NEED FOR LOW POWER: All high-performance systems need to reduction of power dissipation and power consumption [11] [12] since it is desirable to maximize the runtime with minimum requirements in size, weight of batteries and battery life. That's why the technology with low power consumption has become a major subject in today's electronic world.

Now a day's most of the applications, like medicine, military, speech processing, telecommunication, image processing etc. All are implemented by using DSP processors because of its great accuracy, reliability, and flexibility in configuration. To build a sophisticated DSPs for that we are concentrating on the internal elements of processor, In DSP processors architecture is one of the important elements is digital filters. An important design constraint for implementation of high-performance DSP processors is power consumption of the device and must be addressed. In fact, the prodigious performance of the filter is one of the key reasons that DSP has become very popular. Filter eliminates the unwanted frequency response in signal. The unwanted factor generates the noise in signal so the system could not work properly and does not give accurate information. So, the main function of the filter is to filtering this noise signal and could provide accurate information of the system. Based on the types of processing signal, filters are classified into two types. Those are digital filters and analog filters.

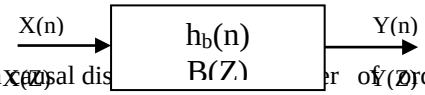
In recent years digital filers [4] have received a great deal of attention. Here digital filters design is an important discussion because digital filters offers better features which are not present in any other filter technologies. It is observed that a huge amount of efficiency improvement within less time for computation at the same level of signal of intrest.

II. DIGITAL FILTER

A digital filter is an important component of a digital signal processing system, which is widely used in signal filtration, detection and prediction. In recent years Digital filters have received a great deal of attention. Here digital filters design is an important discussion because digital filters offer better features that have not present in any other filter technologies. We can do things by utilizing digital filters (unlimited flexibility) that may not be possible by the analog world.

Types of Digital Filters:
Finite impulse response or FIR filter
Infinite impulse response or IIR filter
FIR filters:

The impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. Therefore, the term "finite impulse response" is also called as "non recursive filter". In which the impulse response h(t) does become exactly zero at times t > T for some finite T, thus being of finite duration.



For a causal discrete-time FIR filter of order *N*, each value of the output sequence is a weighted sum of the most recent input values**:**

$$\mathbf{Y[n] = b_0\, x[n]+b_1\, x[n-1]+.....+b_N x[n-N]}$$
$$= \sum_{i=0}^{N} b_i . x[n-i]$$

Where**:**

- x[n] is the input signal,
- y[n]is the output signal,
- N is the filter order; an Nth-order filter has (N+1) terms on the right-hand side
- b_i is the value of the impulse response at the *i* instant for 0<i<N of an Nth-order FIR filter. If the filter is a direct form FIR filter, then bi is also a coefficient of the filter.

This computation is also known as discrete convolution. These terms are commonly referred to as *taps*, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a 5th order/6-tap filter, for instance.

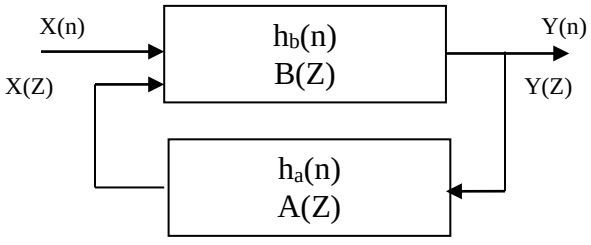
The impulse response of the filter as defined is nonzero over a finite duration. Including zeros, the impulse response is the infinite sequence**:**

$$h(n) = \sum_{i=0}^{N} b_i \bullet \delta[n-i] = \begin{cases} b_n & 0 \le n \le N \\ 0 & otherwise \end{cases}$$

If an FIR filter is non-causal, the range of nonzero values in its impulse response can start before $n = 0$, with the defining formula appropriately generalized.

IIR filters:

The impulse response is infinite because there is feedback in the IIR that's why it is called as Infinite Impulse Response" (IIR), so when you input an impulse, the output theoretically rings indefinitely. Therefore, the term "finite impulse response" is also called as "recursive filter".



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

In this current equation output y(n) is a function of past output and past and present input FIR filters are given more stable response compared to IIR filters

Truth tables: 2-4 DECODER

TYPES OF IIR FILTERS

The modern design methodology for linear continuous-time filters is called network synthesis. Some important filter families designed in this way are:

- Chebyshev filter
- Butterworth filter
- Bessel filter
- Elliptic filter

- **Chebyshev filter:**

Chebyshev Filters are analogue or digital filters having a steeper roll-off and more pass band ripple (type I) or stop band ripple (type II) than Butterworth filters. Chebyshev filters have the property that they minimize the error between the idealized and the actual filter characteristic over the range of the filter, but with ripples in the pass band. This type of filter is named after Pafnuty Chebyshev because its mathematical characteristics are derived from Chebyshev polynomials.

Because of the pass band ripple inherent in Chebyshev filters, the ones that have a smoother response in the pass band but a more irregular response in the stopband are preferred for some applications.

➢ **Type I Chebyshev filter:** Type I Chebyshev Filters are the most common types of Chebyshev filters. The gain (or amplitude) response, as a function of angular frequency of the nth-order low pass filter is equal to the absolute value of the transfer function evaluated at:

$$G_n(\omega) = \left|H_n(j\omega)\right| = \frac{1}{\sqrt{1+\varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

where ε is the ripple factor, ω_0 is the cut-off frequency and T_n is a Chebyshev polynomial of the nth order.

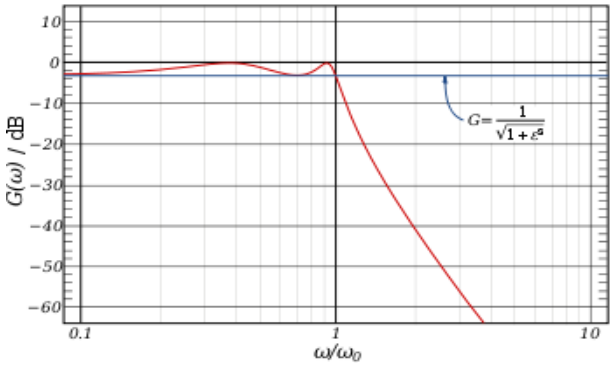


Fig 1: Frequency Response of Chebyshev type-1 Filter

➢ **Type II Chebyshev filters:**

Also known as inverse Chebyshev filters, the Type II Chebyshev filter type is less common because it does not roll off as fast as Type I, and requires more components. It has no ripple in the passband, but does have equal ripple in the stopband. The gain is:

$$G_n(\omega, \omega_0) = \frac{1}{\sqrt{1+\frac{1}{\varepsilon^2 T_n^2(\omega_0/\omega)}}}$$

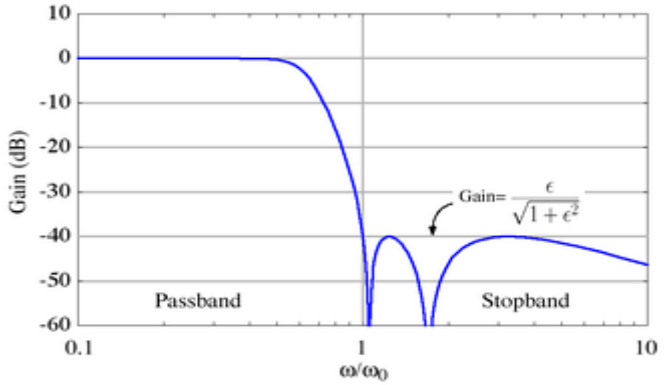


Fig 2: Frequency Response of Chebyshev Type II Filter

➢ **Comparison with other linear filters:**

The following illustration shows the Chebyshev filters next to other common filter types obtained with the same number of coefficients:

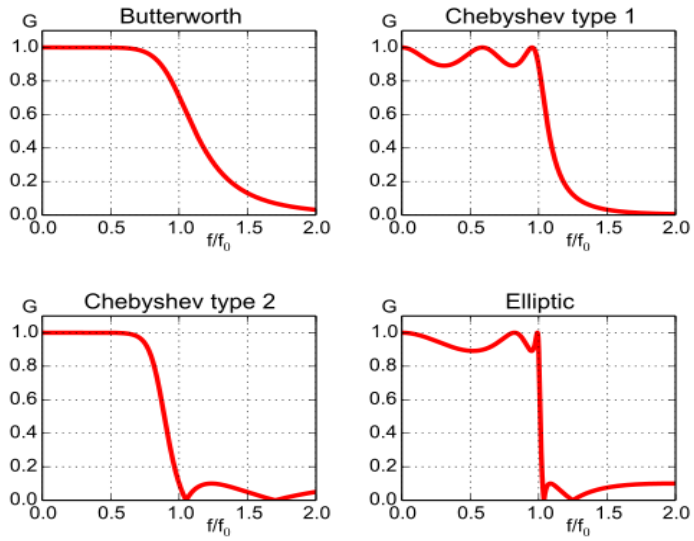


Fig 3: Various Illustrations

Chebyshev filters are sharper than the Butterworth Filter; they are not as sharp as the elliptic one, but they show fewer ripples over the bandwidth.

- **Butterworth filter:**

The Butterworth Filter is a type of signal processing filter designed to have as flat a frequency response as possible in the pass band. It is also referred to as a maximally flat magnitude filter.

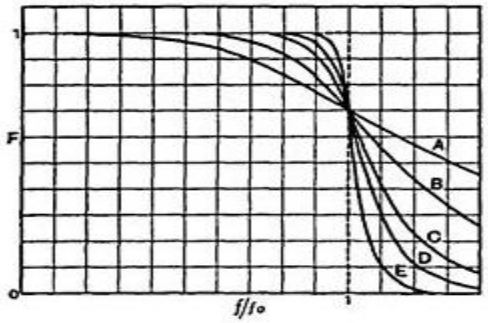


Fig 4: Frequency Response of Butterworth Filter

Butterworth had a reputation for solving "impossible" mathematical problems. At the time, filter design required a considerable amount of designer experience due to limitations of the theory then in use. The filter was not in common use for over 30 years after its publication. Butterworth stated that:

"An ideal electrical filter should not only completely reject the unwanted frequencies but should also have uniform sensitivity for the wanted frequencies".

Such an ideal filter cannot be achieved but Butterworth showed that successively closer approximations were obtained with increasing numbers of filter elements of the right values. At the time, filters generated substantial ripple in the passband, and the choice of component values was highly interactive. Butterworth showed that a low pass filter could be designed whose cut-off frequency was normalized to 1 radian per second and whose frequency response (gain) was

$$G(\omega) = \frac{1}{\sqrt{1+\omega^{2n}}}$$

where ω is the angular frequency in radians per second and *n* is the number of poles in the filter—equal to the number of reactive elements in a passive filter. If ω = 1, the amplitude response of this type of filter in the passband is $1/\sqrt{2} \approx 0.707$, which is half power or −3 dB.

The frequency response of the Butterworth filter is maximally flat (i.e. has no ripples) in the passband and rolls off towards zero in the stop band. Compared with a Chebyshev Type I/Type II filter or an elliptic filter, the Butterworth filter has a slower roll-off, and thus will require a higher order to implement a particular stopband specification, but Butterworth filters have a more linear phase response in the pass-band than Chebyshev Type I/Type II and elliptic filters can achieve.

There are four techniques used to design IIR filters:

- The Bilinear transformation method.
- The Impulse Invariant method
- Matched z-transform
- Approximate Derivative method

Bilinear transformation:

Bilinear transformation method for analogue-to-digital filters conversion. The *bilinear transformation* is a mathematical mapping of variables. In digital filtering, it is a standard method of mapping the *s* or analogue plane into the *z* or digital plane. It transforms analogue filters, designed using classical filter design techniques, into their discrete equivalents. The bilinear transformation maps the *s*-plane into the *z*-plane by $H(z) = H(s)\big|_{s=2f_s\frac{z-1}{z+1}}$

This transformation maps the $j\Omega$ axis (from $\Omega = -\infty$ to $+\infty$) repeatedly around the unit circle (e^{jw}, from $\omega = -\pi$ to π) b

$$\omega = 2\tan^{-1}\left(\frac{\Omega}{2f_s}\right)$$

Summary of analogue to digital transformation

Technique	Mapping	+/-
Bilinear Transformation	$H(z) = H(s)\Big\|_{s=\frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right]}$	+ Preserves shape of filter - Aliasing
Impulse invariant method	$H(s) = \sum_{k=1}^{N}\frac{c_k}{s-p_k}$ then $H(z) = \sum_{k=1}^{N}\frac{c_k}{1-e^{p_k T}z^{-1}}$	+ No aliasing - Restricted pole location, shape distortion
Matched z-transform	$H(s) = \frac{c_k\prod_{k=1}^{M}(s-z_k)}{\prod_{k=1}^{N}(s-p_k)}$ then $H(z) = \frac{\prod_{k=1}^{M}(1-e^{z_k T}z^{-1})}{\prod_{k=1}^{N}(1-e^{p_k T}z^{-1})}$	+ No aliasing - Shape distortion
Approximate Derivative	$H(z) = H(s)\Big\|_{s=\frac{1-z^{-1}}{T}}$	+ Directly maps pole and zero locations, - Aliasing

III.IIR Filter Architecture Realization:

DIRECTFORM-I:

Direct form-I realization of IIR filters 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 first part of the expression refers to non-recursive part and later refers to recursive part of IIR filter. In IIR filter direct-I realization, these two parts are separately considered and realized. Directform-I realization of IIR filters starts with this expression:

As non-recursive and recursive part of IIR filter are separately realized, it doesn't matter which of them will be used first in filtering process.

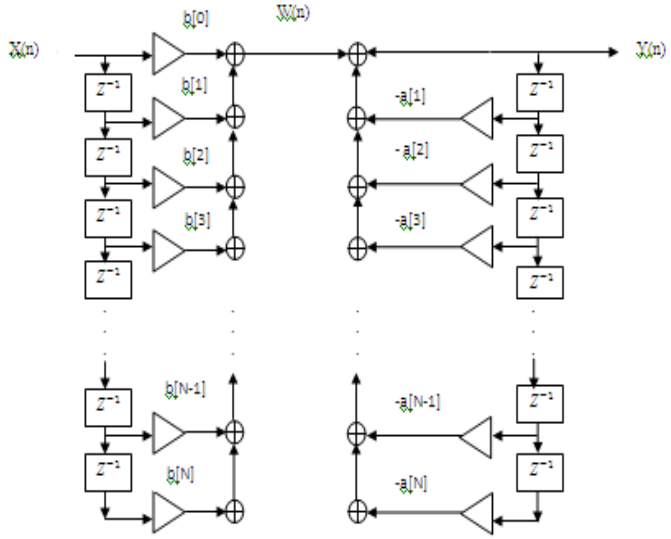


Fig 5: IIR filter direct form-I realization

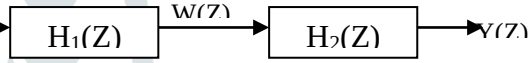
Direct form-I realization requires in total of 2N delay lines, (2N+1) multiplications and 2N additions.

As seen from above figures direct-I form realization requires a total of 2N delay lines, (2N+1) multiplications and 2N additions, Where N is the order of the filter.

For example:
Consider a 3-order equation
We can implement a H(Z) as a cascade of two filter functions

$$H(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

Where



$$H_1(z) = \frac{W(z)}{X(z)} = P(z) = p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}$$

$$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{D(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

A cascade of the two structures realizing a $H_1(Z)$ and $H_2(Z)$ leads to the realization of H(Z) is shown below:

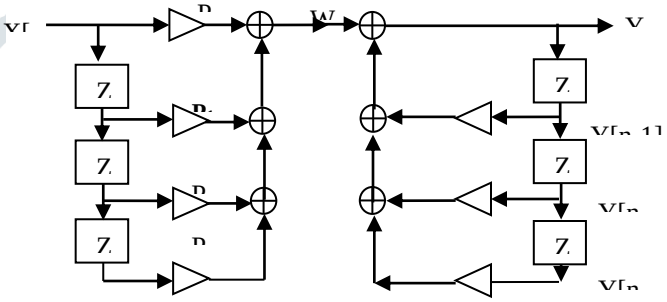


Fig 6: H(z) function realization

DIRECTFORM-II

Direct form-II realization structure reduced the number of delay lines to the minimum, in this realization structure the number of delay elements are equal to order of the filter i.e. N, Direct canonic structure uses (2N+1) multiplications and 2N additions

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

Here Recursive and non-recursive parts of the IIR filter are not realized separately

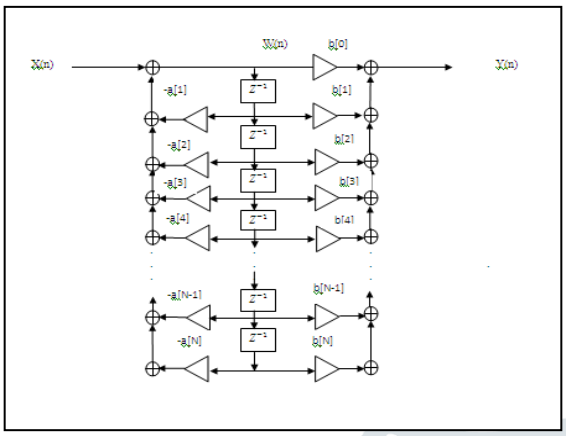


Fig 6 : IIR filter direct form-II realization

For example:
Consider a 3-order equation

$$H(z)=\frac{P(z)}{D(z)}=\frac{p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}}{1+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}}$$

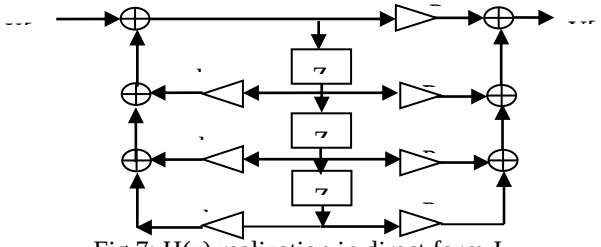


Fig 7: H(z) realization in direct form-I

CASCADE FORM REALIZATION:

The cascade form structure is nothing but a cascade or series interconnection of the sub transfer functions or sub system functions which are realized by using the direct form structures (either direct form-I or direct form-II or combination of both), In cascade form realization, the given transfer function H (z) is expressed as a product of a number of second order or first order sections as shown below

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

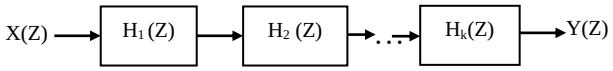


Fig 8: Block diagram representation of H(z)

$$H_1(Z)=\frac{1+b_{k1}+b_{k2}}{1+a_{k1}+a_{k2}}$$

$$H_2(Z)=\frac{1+b_{m1}+b_{m2}}{1+a_{m1}+a_{m2}}$$

Cascade realization (Direct form-I/Direct form-II):

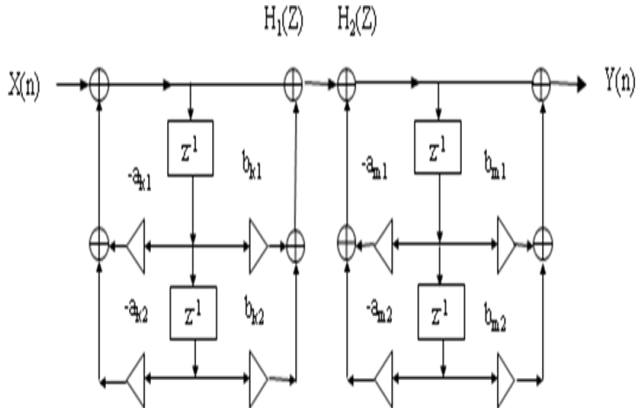


Fig 9: Cascade Architecture in Direct Form-II

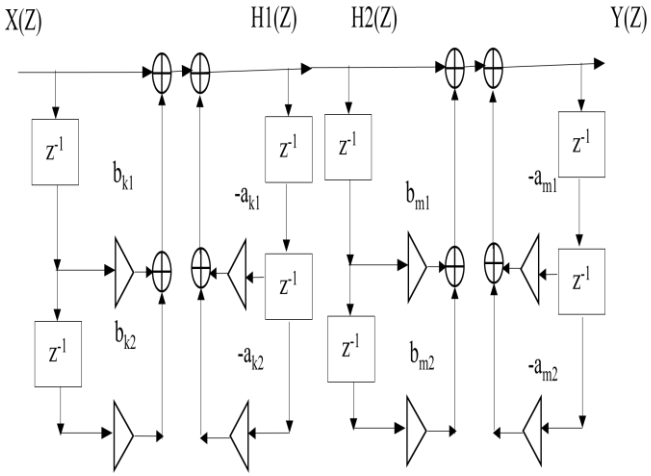


Fig 10: Cascade Architecture in Direct form-I

Cascade form realization is used to implement high order filters which are realized by using either direct from-I / direct form-II depending on user constraints.

IV.IIR FILTER DESIGN AND IMPLEMENTATION

STEP1: Generate filter coefficients by using MATLAB FDA tool.

Filter Specifications:
Sampling frequency Fs=6 Khz
Pass Band frequency Fp=2 Khz
Stop band frequency Fstop=2.5 Khz
Pass band attenuation Ap=1
Stop band attenuation Astop=22
Order=5

STEP2: Generated Transfer function for above filter specifications using MATLAB syntax "fit"

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

STEP3: Realizing transfers function in Direct Form-I, Direct Form-II, Cascade Forms using XILINX System Generator.

STEP4: HDL Code generation using system generator

STEP5: Generate synthesis report to verify filter constraints such as area, delay and power

STEP6: Perform timing and power analysis for the realization techniques.

STEP7: Comparison of Delay and Power reports.

IMPLEMENTATION FLOW CHART:

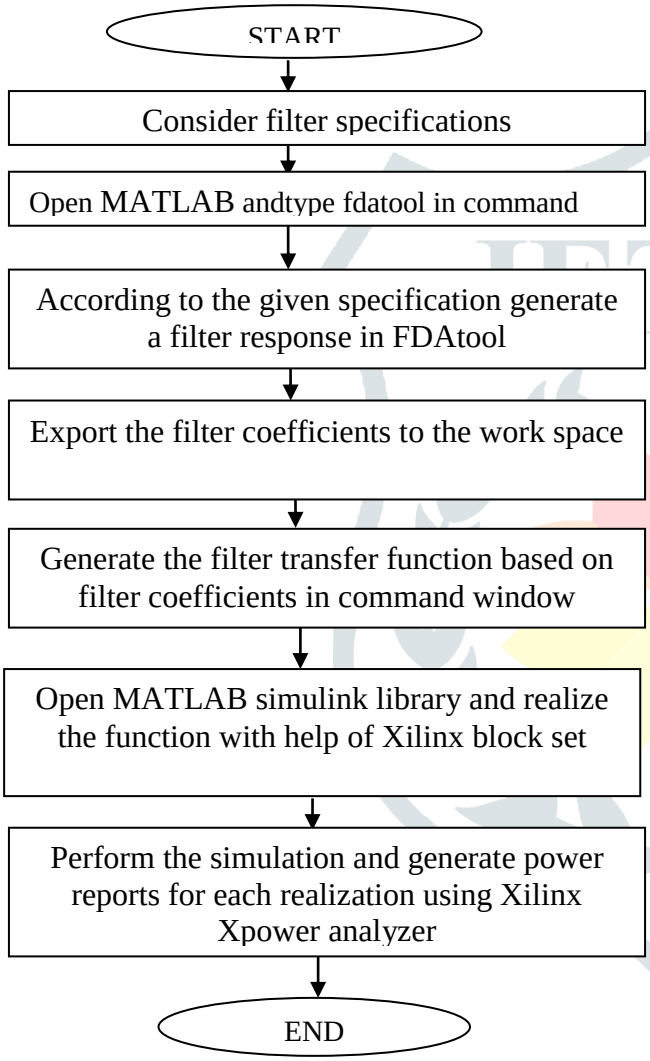


Fig 11: Flow chart

V. IMPLEMENTATION OF AN IIR FILTER

The following architecture is proposed for the implementing the designed filter using the basic blocks available in the Xilinx block set of MATLAB system generator like adders, multipliers and delays.

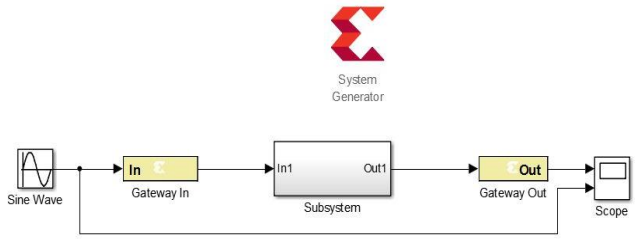


Fig 12: Implementation of an IIR filter in system generator

Subsystem design:

Direct form-I

Design of a 5th order filter in direct form-I using subsystem.

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

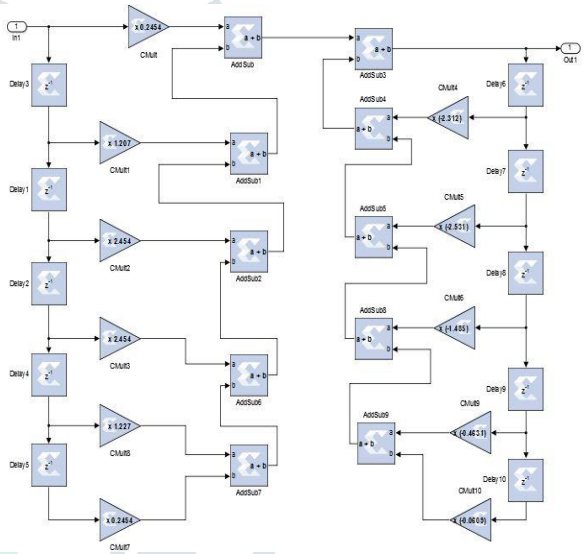


Fig 13: Realization of a filter in Direct form-I

Direct form-II

Design of a 5th order filter in direct form-II using subsystem.

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

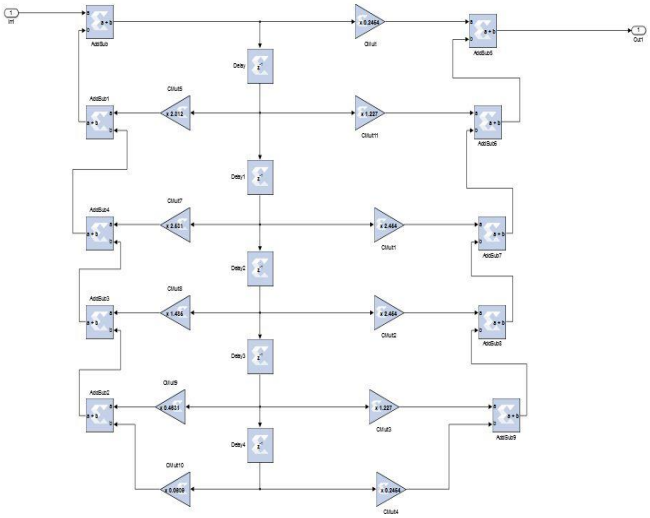


Fig 14: Realization of a filter in Direct form-II

Cascade form realization (Direct form-I)

Design of a 5th order filter in cascade form using Direct form-I realization.

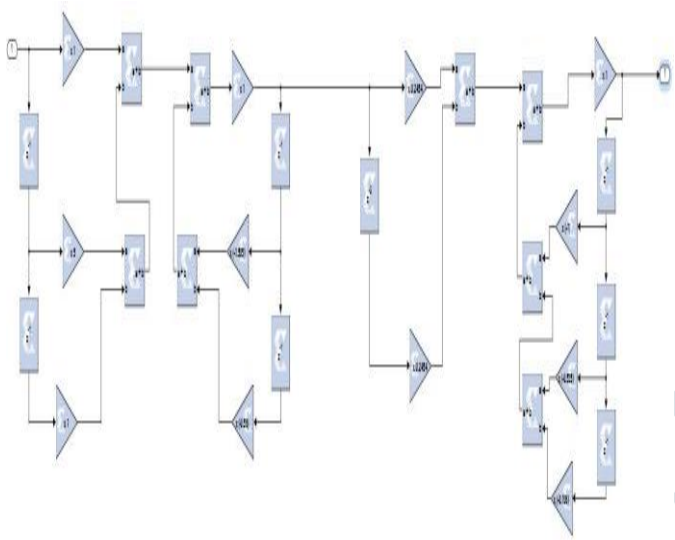


Fig 15: Realization of a filter in Cascade form (Direct form-I)

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

Cascade form realization (Direct form-II)

Design of a 5th order filter in cascade form using Direct form-I realization.

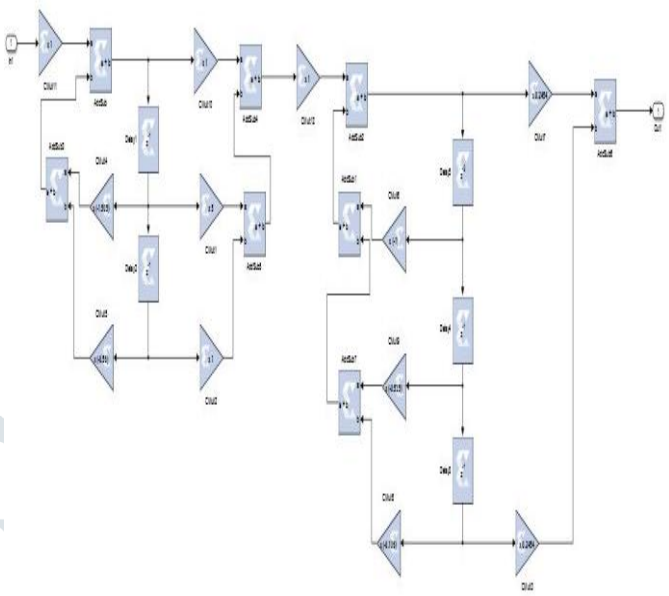


Fig 16: Realization of a filter in Cascade form (Direct form-II)

VI RESULTS

DELAY response of different realization structures

REALIZATION	DELAY
DIRECT-I	61.74ns
DIRECT-II	50.353ns
CASCADE DIRECT-I	56.723ns
CASCADE DIRECT-II	55.36ns

POWER response of different realization structures

Fig 17:Memory and delay of realization structures

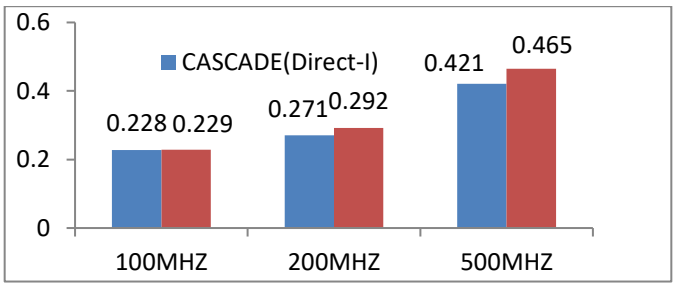


Fig 18:Power report of realization structures at different frequency ranges

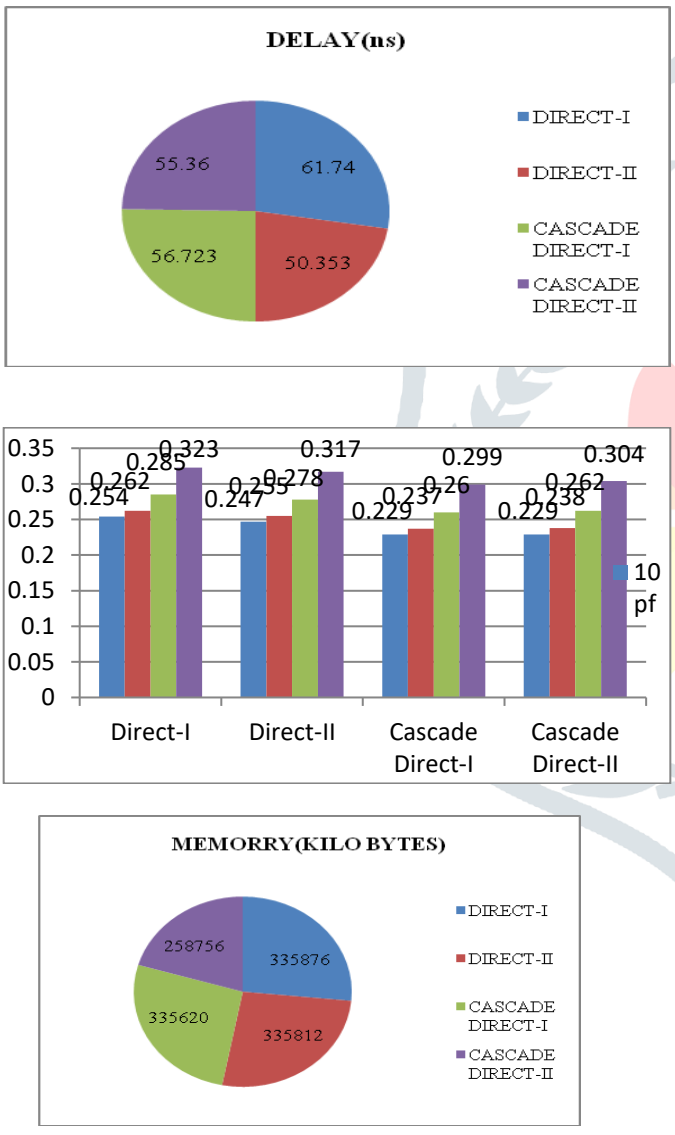


Fig 19: Power report of realization structures at different loads

VII. CONCLUSION

We have so many realization architectures to implement IIR filter like Direct Form-I, Direct Form-II, Cascade (Direct form-I/ Direct form-II), parallel form. All these structures provide a space for selection of an appropriate architecture for reduction of power consumption and improvement in speed of digital filters.

Efficient algorithms are realized to implement high order filter with respect to power consumption, delay, area. Direct form-I technique is best suitable to implement high order filter when power is major constraint.

On the other hand, Direct form-II technique is best suitable to implement high order filter when speed, area is major constraint.

ACKNOWLEDGMENT

We would like to thank all the authors in the references for providing great knowledge and helpful advices when ever required.

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]Bhattacharyya.A;Sharma.P;Murali.N;Murty.S.AV.S."Development of FPGA based IIR Filter implementation of 2-degree of Freedom PID controller," India Conference (INDICON), 2011 Annual IEEE , vol., no., pp.1,8, 16-18 Dec. 2011

[4] Direct form-based pipelined IIR filter realizationP. Boonyanant; S. TantaratanaIEEE. APCCAS 1998. 1998 IEEE Asia-Pacific Conference on Circuits and Systems. Microelectronics and Integrating Systems. Proceedings (Cat. No.98EX242)Pages: 85 - 88,

[5] Some comparisons between fir and iir digitalfilters 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 realization of multiplier less elliptic IIR filters, IEEE April2015.

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

[11] 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.

[12] 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.