Multiplierless FIR Filter Design using Canonical Signed Digit and its implementation on FPGA

By
PRANABESH BHATTACHARJEE
Registration No. 137281 of 2016-2017
Exam Roll No. M6VLS19016

Under the Guidance of
Prof. P. VENKATESWARAN

Thesis submitted in partial fulfilment of the requirements
for the award of the Degree of Master of Technology
M.Tech. in VLSI Design and Microelectronics Technology

Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata-700032.

May 2019

DECLARATION OF ORIGNALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this Thesis contains literature survey and original work by the undersigned candidate, as part of his Master of Technology, M.Tech. course in VLSI Design and Microelectronics Technology.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name of the Candidate : Pranabesh Bhattacharjee

Exam Roll No. : M6VLS19016

Thesis Title : Studies on Multiplierless FIR Filter Design using Canonical Signed Digit and its implementation on FPGA

Signature of the Candidate : ----------

Date:

FACULTY OF ENGINEERING AND TECHNOLOGYJADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

I hereby recommend that this Thesis prepared by **Pranabesh Bhattacharjee** entitled **Studies on Multiplierless FIR Filter Design using Canonical Signed Digit and its implementation on FPGA** under my supervision be accepted for partial fulfillment of the requirements for the award of the Degree of Master of Technology, M.Tech. in VLSI Design and Microelectronics Technology.

……..

Dr. P. Venkateswaran
Professor
Dept. of Electronics and Tele-Communication Engineering
Jadavpur University, Kolkata- 700032

……

Dr. Sheli Sinha Chaudhuri
Professor and Head
Department of Electronics and
Telecommunication Engineering,
Jadavpur University, Kolkata- 700032

……..

Prof. Dr. Chiranjib Bhattacharjee
Dean
Faculty of Engineering and Technology
Jadavpur University, Kolkata- 700032

FACULTY OF ENGINEERING AND TECHNOLOGYJADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL*

The foregoing Thesis is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the Degree for which it has been submitted. It is understand that by this approval, the undersigned don't necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the Thesis only for the purpose for which it is submitted.

Committee on ……

Final examination for

Evaluation of the thesis of:

PRANABESH BHATTACHARJEE

Exam Roll No: **M6VLS19016** ……

…….

Signature of Examiners

* Only in case the thesis is approved.

ACKNOWLEDGEMENT

I express my deep sense of gratitude and indebtedness to my Thesis Supervisor, Prof. Palaniandavar Venkateswaran, Department of Electronics & Tele-Communication Engineering, Jadavpur University, Kolkata, for providing me the opportunity to carry out the Thesis work. I am grateful to him for the valuable insights and suggestions that he gave me throughout my M. Tech. Project.

Further, I express my special thanks to Dr. Supriya Dhabal, Assistant Professor, Dept. of Electronics & Communication Engineering, Netaji Subhash Engineering College, Kolkata, for his constant and effortful guidance by providing necessary information whenever required.

I would like to thank our course coordinator of M.Tech. course in VLSI design and Microelectronics Technology, Prof. Subir Kumar Sarkar for his help during the entire duration of the course. I also thank, Prof. Sheli Sinha Chaudhuri, Head, Department of Electronics & Tele- Communication Engineering, Jadavpur University for her compassionate approach. I would like to express my sincere gratitude to all the teaching and non-teaching staffs of the department for providing necessary support.

Last but not the least, this work would not have been possible without the love, support and encouragement of my family, classmates and near and dear ones and I all of them.

Computer Networking Lab (T-3-16A)
ETCE Dept., Jadavpur University
Kolkata - 700 032. West Bengal.

May 2019

.....
PRANABESH BHATTACHARJEE

ABSTRACT

The Finite Impulse Response (FIR) filter is a digital filter widely used in Digital Signal Processing applications in various fields like imaging, instrumentation, communications, etc. Programmable Digital Signal Processors (PDSPs) can be used in implementing the FIR filter. However for realizing a large-order filter, many complex computations are needed which affects the performance of the common digital signal processors in terms of speed, cost, flexibility, etc. Field Programmable Gate Array (FPGA) has become an extremely cost effective means of implementing computationally intensive digital signal processing algorithms to improve overall system performance. The FPGA implementation of FIR filter, utilizing the dedicated hardware resources can effectively achieve Application Specific Integrated Circuit - (ASIC) like performance while reducing development time cost. In this Thesis, FIR filter is implemented on FPGA focusing less area, power consumption. We have considered a Canonical Signed Digit - CSD technique for the multiplierless filter design. This approach gives a better performance than the common filter structures in terms of speed of operation, cost, and power consumption in real-time. The FIR filter is designed using MATLAB FDA tool and then simulated with the help of Quartus II synthesis environment and implemented on Altera Cyclone IV FPGA board. Simulation result shows that the proposed method provide more optimized design in terms of area and power in comparison to the conventional multiplier based tapped delay FIR filter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Preamble

Filter is an important topic in the field of signal processing. Filtering is done basically to block unwanted part of a signal while allowing the required portion to pass through. In the field of signal processing applications there is a tremendous importance of electrical filters, there are basically two kinds of filter- (a) analog, and (b) digital filter. Both the filters are used to pass the input signal samples within a certain frequency range and block rest of the frequency samples [1-3]. Analog filters are made of electrical components (resistor, capacitor, inductor, op-amps etc.) on the other side the digital filters consists of delay, adder/subtractor and multipliers [1-3]. depending upon the application they are used. Few of the applications of these electronic filters are like audio signal processing, image processing, video processing, noise elimination etc.

Digital filter further classified into two types- (a) Finite Impulse Response (FIR) (b) Infinite Impulse Response (IIR) filter [4-5]. An FIR filter have finite impulse response with no feedback path on the other side an IIR filter have infinite number impulse response consists of feedback path in each stage. FIR filters are very popular in the field of Digital Signal Processing (DSP) not only for its simplicity, but also less hardware requirement, less power consumption, linear phase characteristics [6].

Field Programmable Gate Array (FPGA) is a based digital filter design [7] is a very popular research topic now-a-days for its high throughput and flexibility compare to any high end DSP chip [8]. It does not have any fixed hardware structure and not constraint by any fixed instruction set. It can be program according to the application, supports parallel processing and have ability to process large data in a few clock cycle [9].

1.2 Literature Survey

A large number of electronic gadgets come into market every year. Digital filter is the most common component for all of them. These filters are used to process several signals like audio, video, image etc.. Today's competitive market demands the products to be reliable as well as compact and cost efficient. To meet these goals, production cost (hardware cost) needs to be reduced. Many design approaches and platforms are introduced to implement digital filters keeping these factors in mind. One of the most popular platforms to implement a digital filter is FPGA. In [10], [11] the FIR filter implementation method on a XILINX Spartan FPGA board has been discussed. In [12] an ALU based universal FIR filter has been proposed and demonstrated. This filter can be implemented just by programming the instructions in the ROM with identical hardware architecture. MATLAB FDA tool based 16 order constant coefficients FIR filter design is shown in [13] which can be implemented to Quartus II FPGA hardware. XILINX Spartan 2 FPGA board based implementation process with detail steps are shown in [14]. In [15] a distributed algorithm based higher order FIR filter design method and FPGA implementation process discussed and it is also shown that the design the filter speed is higher and the resource occupation is fewer. In [16] an asynchronous FIR Filter architecture was presented, along with an asynchronous analog to digital converter. They designed FIR Filter architecture using the micro-pipeline asynchronous style. They successfully implemented for the first time on a commercial FPGA board. A specific library has also been designed for this purpose. It allows the synthesis of asynchronous primitive blocks (the control path in this case) on the synchronous FPGA. Simulation results of the FIR Filter after the place and route validate the implementation. This work is still going on, in order to optimize the implementation. Basic structure and hardware characteristics of the FIR digital filter and design method of the FIR filter are discussed in [17] on the basis of the FIR filter structure. They proposed a structure which is based on FPGA, draws the coefficient by MATLAB and adopts the pipeline to implement the FIR digital filter. They also presented the introduction of the overall framework of the FIR digital filter adopting the finite state machine as well as the principle of each module of the design. The design is implemented by use of the verilog hardware description

language and each module is verified and simulated by Quartus 8.0 and Modelsim-Altera. An FPGA based filter for multipurpose application has been designed [18]. In [19], principle of distributed algorithms is presented and applied to the ECG signal FIR filter design methods, combined with Altera's Cyclone series chip made to achieve FIR digital filter design Distributed algorithms can greatly reduce the size of the hardware circuitry to improve the circuit speed of execution also useful for FPGA implementation. A very simple method to design and implement an FIR filter using MATLAB FDA tool is shown in [20]. This technique is used in this project also, where the VHDL code for a specific design generated automatically using MATLAB. The 'power of two' conversion of a floating point number is demonstrated in [21] which are used to reduce the hardware cost. In [23] multiplier representation using Canonical Signed Digit (CSD) is presented, which reduces the number of non-zero bits in the fixed point coefficients and helps to reduce the numbers of utilized hardware [24-29]. Application of CSD technique for digital filter design is demonstrated in [30]. FIR filter design related more many work has been done to trade off the design utilization[31], partial self-reconfigurable adaptive design method [32], develop some new computer program for designing optimum linear phase filter[33], high speed [34], and higher order [35] design approach are demonstrated by the authors on their respective work.

1.3 Thesis Motivation

The package density of electronic devices is increasing day by day and hence the circuit complexity of gadgets like smartphones are also increasing rapidly. The motivation behind the present research work is to fulfill the optimized design criteria of digital filter design. The target of this research work is to implement the FIR filter design on FPGA hardware with the following criteria:

(a) Less hardware utilization
(b) Less power consumption
(c) Less cell delay

In order to achieve those goals a CSD based filter design approach is proposed in this work over the conversion multiplier based method. In this thesis the proposed design method has been discussed. Moreover, the proposed design method is utilizing minimum resources of an FPGA chip compare to conventional design without hampering the filter performance.

1.4 Organization of the Thesis

The thesis is divided into 4 chapters and its outline is described as given below:

***CHAPTER 1*:** It is an introductory chapter. In this chapter it is investigated that how the present work is relevant to other modern contemporary research works all around the globe. A brief literature review is also presented in this chapter. Finally, the motivation and objective of the present work is discussed in this chapter

***CHAPTER 2*:** This chapter consists of a detail description about different structures and properties of FIR filters, CSD computation method and its application for FIR filter design. MATLAB Filter Design and Analysis (FDA) tool FIR design and Hardware Description Language (HDL) code generation process in details also covered in this chapter.

***CHAPTER 3*:** The details about FPGA technology and step by step design implementation process of a MATLAB generated filter code covered in this chapter. Here all the simulation and synthesis results are shown; analysis and comparison of the experimental results have been discussed also shown to prove that the proposed design method is justified.

***CHAPTER 4*:** It is the concluding chapter. Here the findings of complete work are summarized. Conclusion of this work is discussed in this chapter. The relevance of the thesis work is once again analyzed.

Chapter 2

DESIGN OF FIR FILTER USING CSD

2.1 Introduction

FIR filter is a very popular name in the field of DSP [5]. This filter has a finite number of impulse responses with no feedback. The present output value of this filter depends on the present and the past input only, that's why these filters are also called non-recursive filter [6]. FIR filters are used for applications like linear phase responses like high-quality audio systems, biomedical signal processing, spectrum analysis, digital image processing, and pattern recognition. The basic components of an FIR filter are delay block (basically D-FF), multiplier block, and adder block. If the order of the filter increases, the utilization of the number of components also increases proportionally. Canonical Signed Digit (CSD) [23] method is a unique computation method which is basically used to reduce the number of non-zero bits. That means less calculation is required without affecting the final output in case of multiplication. This technique has been applied in this thesis to design and implement an optimized FIR filter. The basic properties and structure of an FIR filter along with the details of the CSD computation method with respect to the filter design has been discussed in the following sections.

2.2 FIR Filter Properties

Finite Impulse Response (FIR) filter is a common kind of digital filter. A digital filter generally uses a digital processor to execute numerical calculations of a digital signal. That processor may be a general purpose computer such as a PC, or a dedicated Digital Signal Processor (DSP) chip. There are several advantages of

The following list gives some of the main advantages of digital over analog filters [1-5].

a) A digital filter is programmable, i.e. its task is determined by a program stored in the processor's memory. This means without affecting the circuitry a digital filter can be reconfiguring, unlike an analog filter.
b) Digital filters are easily designable, testable and implementable on a general

purpose computer or workstation.

c) Properties of an analog filter are subjected to drift and are dependent on temperature, (particularly for those circuits composed of active components). Digital filters do not suffer from these difficulties, and so these are very stable with respect to time and temperature.

d) Unlike their analog equivalents, digital filters can handle low-frequency signals perfectly. As the speed of digital signal processing technology continues to increase, those digital filters are being applied for high-frequency signals also in the field of RF (radio frequency) domain, in the past which was the exclusively reserve for analog technology.

e) Digital filters can be multipurpose because of their ability to process signals in a variety of ways, which includes its capability to adapt to changes in the features of the signal.

The output response of any digital filter is basically convolution of input sequences and filter coefficients in the frequency domain. This output equation of a filter is expressed as [1-2]

$$Y(z) = X(z) * H(z) \tag{2.1}$$

where H(z), X(z), Y(z) are the z-transforms of time domain impulse response, input samples, and output samples respectively. H(z) is basically the transfer function of the filter and h(n) is the n^{th} filter coefficient approximated according to the desired response.

$$H(z) = \sum_{n=0}^{N} h(n) z^{-n} \tag{2.2}$$

Eq. (2.2) represents the expression for H(z) which is the summation of the z-transform of individual impulse responses in time domain "h(n)" for an 'N' order filter. A digital filter usually implemented by delays, multipliers, and adders. Those are the basic building blocks of an FIR filter [6]. All these above mentioned blocks/modules and signal flow diagram are shown in Figure 2.1.

Finite impulse response filters are a kind of digital filters that have a finite impulse response. FIR filters operate only on present and past input values.

$$h(n) = \begin{cases} 0, & n \geq n_1 \\ 0, & n \geq n_2 \end{cases} \quad \text{Where } n_1 \text{ and } n_2 \text{ lies between } -\infty \text{ to } +\infty \tag{2.3}$$

where $h(n)$ denotes the impulse response of the digital filter, '*n'* is the discrete time index and n_1, n_2 are constants. A difference equation is the discrete time equivalent of a continuous time differential equation.

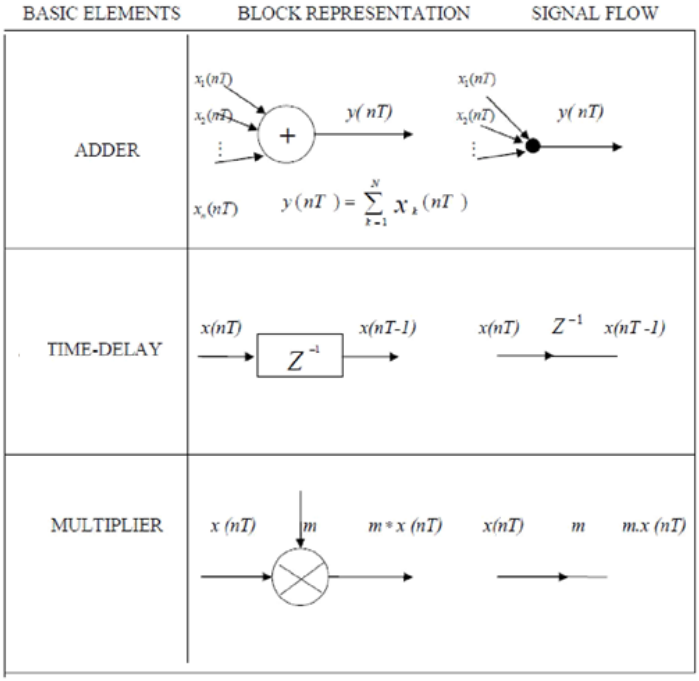

Figure 2.1 Block representation & Signal flow of basic filter elements

The general difference equation for an FIR digital filter [1-6] is,

$$y(n) = \sum_{k=0}^{n-1} h_k x(n-k) \tag{2.4}$$

where y(n) is the filter output at discrete time instance n, h_k is the k_{th} feed-forward tap or filter coefficient, and x(n-k) is the input of the filter is delayed by k samples. The Σ denotes summation from k = 0 to k = n-1 where 'n' is the order of the filter. In a FIR filter, if there is a single impulse response is present at the input and all succeeding inputs are zero, then after a finite time the output of that filter becomes zero. That's why FIR filters are finite. Eq. (2.4) describes the nature of an FIR filter only works with present and past inputs. So the FIR filter is also known as non-recursive filters. The FIR digital filter is always stable, because it has no feedback and linear phase response, because of its good features such as only zeros, linear phase, stability and design flexibility, the FIR filter is widely employed in Digital Signal Processing (DSP).

2.3 FIR Filter Structure

The computational algorithm implementing the equation of an FIR filter can be represented using a block diagram and it is done using the basic building blocks shown in Figure 2.1. The way of presenting the difference equations in the form of a block diagram and signal flow diagram is the simple way to write an algorithm, which can be implemented in the digital computer. In the below sections individually discussed the direct form structure, transpose form structure, cascade structure and then lattice structure. If a digital filter has the number of delays equal to the order of the transfer function said to be canonic. Otherwise, it is called non-canonic structure.

2.3.1 Direct-Form Structure

In direct-form structures [6] for the digital filter, the real filter coefficients appear as multipliers in the block diagram representation. Considering X(z) is the filter input and Y(z) is the filter output, then the transfer function H(z) is given as,

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^{N} a_i z^{-i} \tag{2.4}$$

There are four Direct-form structures, which are different realizations of Eq. (2.3). The first direct structure only is presented here and is as shown in Figure 2.1. This structure is also called Tapped Delay Line (TDL) structure because each multiplier blocks are tapped by a delay line

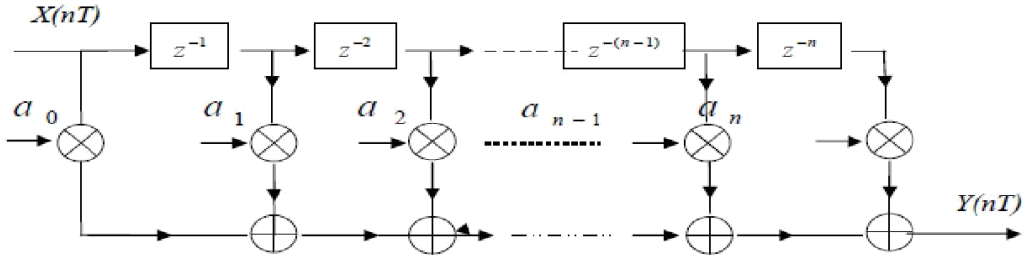

Figure 2.2 Direct-Form of FIR Filter

The signal flow diagram of Figure 2.2 is as shown below in Figure 2.3

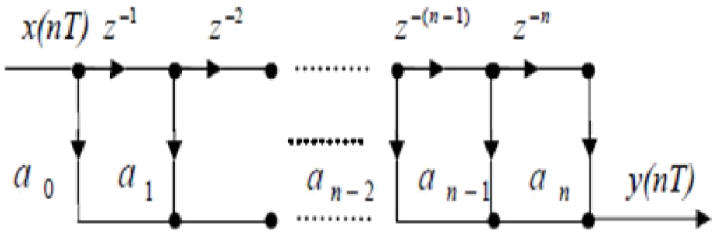

Figure 2.3 Direct-Form signal flow diagram

The 1-D structure is also called canonical because it possesses n-time delay elements. As seen from the signal flow diagram the above representation requires 'n' delay elements, 'n + 1' multipliers, and 'n' adders to implement in the digital computer. The above structure suffers extreme coefficient sensitivity as the value of grows large. That is a small change in a coefficient for a large value of n causes large changes in the zeros of H(z).

2.3.2 Transpose-Form Structure

In transpose form structure [6] basically flow-graph-reversal theorem. In this case, the position of the input and output changes and the direction of signal flow also become in reverse direction, the comparison of direct form and transpose form are shown in Figure 2.4 and 2.5.

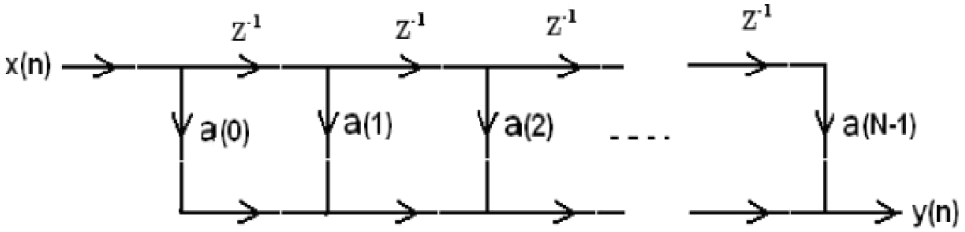

Figure 2.4 Signal flow diagram of direct-form

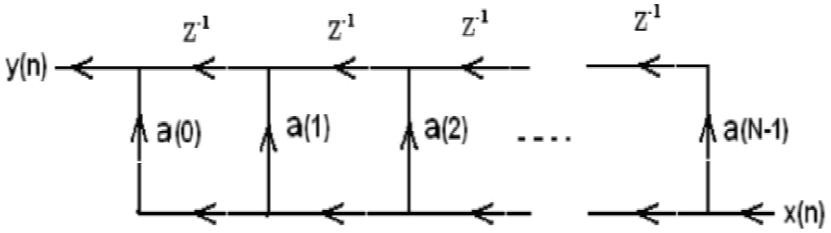

Figure 2.5 Signal flow diagram of Transpose-form

The direction of signal flows is shown using arrows. Comparing Figure 2.4 and 2.5 it can be easily observed that all the direction of the signal flow changes including the

I/O position.

2.3.3 Cascade Structure

This structure generally used to define a short length FIR filter [6]or to connect two or more individual short length filters. The z-transform of a long FIR filter can be factored into a cascade of short-length filters.

$$b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_m z^{-m} = b_0 (1 - z_1 z^{-1})(1 - z_2 z^{-2}) .... (1 - z_m z^{m}) \quad (2.5)$$

where the z_i are the zeroes of the polynomial.

The overall filter implementation can be done using a cascade structure as shown in Figure 2.6.

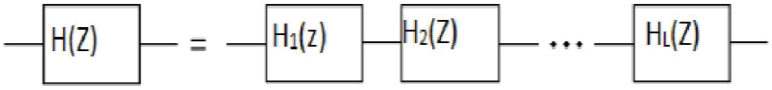

Figure 2.6 Cascaded Structures

This is occasionally done in FIR filter implementation when one or more of the short length filters can be implemented efficiently.

2.3.4 Lattice Structure

FIR filters sometimes designed using lattice structure [6] in case of adaptive filtering, digital speech processing, etc. In case of finite word length filter, lattice structure gives good robustness. FIR filter design using a lattice structure is shown in Figure 2.7.

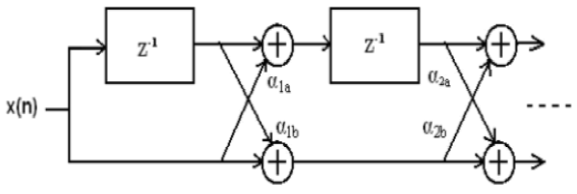

Figure 2.7 Lattice Structure

2.4 Comparison of Various Structures

There are several design processes and structures of FIR filter are shown in the above

sections, among those structures the direct-form realizations or tapped-delay structure is the simplest model. However, the other models offer some distinct advantages. For this work, the tapped delay structure is used for simplicity.

2.5 Canonical Signed Digit

The idea of invention Canonical Signed Digit (CSD) [23] is mainly focusing to minimize the non-zero terms from a binary representation without changing its decimal value. It is first invented by Kung and Leiserson in the year of 1978. It is a technique to encoding a signed-digit representation, not only 0 and 1, but also -1. In signed digit format each digit of the number has a sign associated with it.

As an example 30 can be represented in binary 00011110, this number can be expressed in another way like (32-2) or in binary (001000-10), which has a reduced non zero terms. In the case of CSD, the signed digit (-1) term is replaced by $\bar{1}$ so in CSD it will be represented as 0010001̄0. The number of non-zero terms reduced by 2 units. Using CSD maximum 50% reduction of non-zero bits are possible that means if a 2's complement number having 'n' bits then in CSD it mostly maximum up to n/2.

2.5.1 Computation Method in CSD

The conversion of a signed binary into a CSD is explained below in details [23], it consists of some easy steps.

1) 1st need to check the number is in (+) or (-) if it contains a (-) sign then it needs to convert 2's complement first, for (+) value there is no such conversion needed only weighted binary representation is enough
2) After completion of the 1st stage then next it needs to check from LSB to MSB of the signed binary number there must not be any consecutive any non-zero terms. i.e. ci*ci-l = 0
3) If there are more than one non-zero elements in a row then both 1 will be converted to zeros and a binary 1 will add to the next bit, if there is already a 1 then need to follow the same method, the 1 will add with the preceding bit.
4) At the end forcefully a -1 need to add at the rightmost digit.

Example:

Considering a random decimal number 287, which is 1 0001 1111 in binary representation (256 + 16 + 8 + 4 + 2 + 1) =1 0001 1111.

Step1: In this step starting from LSB If there are any consecutive two non-zero bits then will be converted to zeros and a binary 1 will add to the next bit. That means the number will be 100100000.

Step2: At the next step a -1 value will be added forcefully at the LSB from where the consecutive 1 bits counting starts. So, the final representation will be $100100001\bar{}$, which is 256 + 32 – 1= 287.

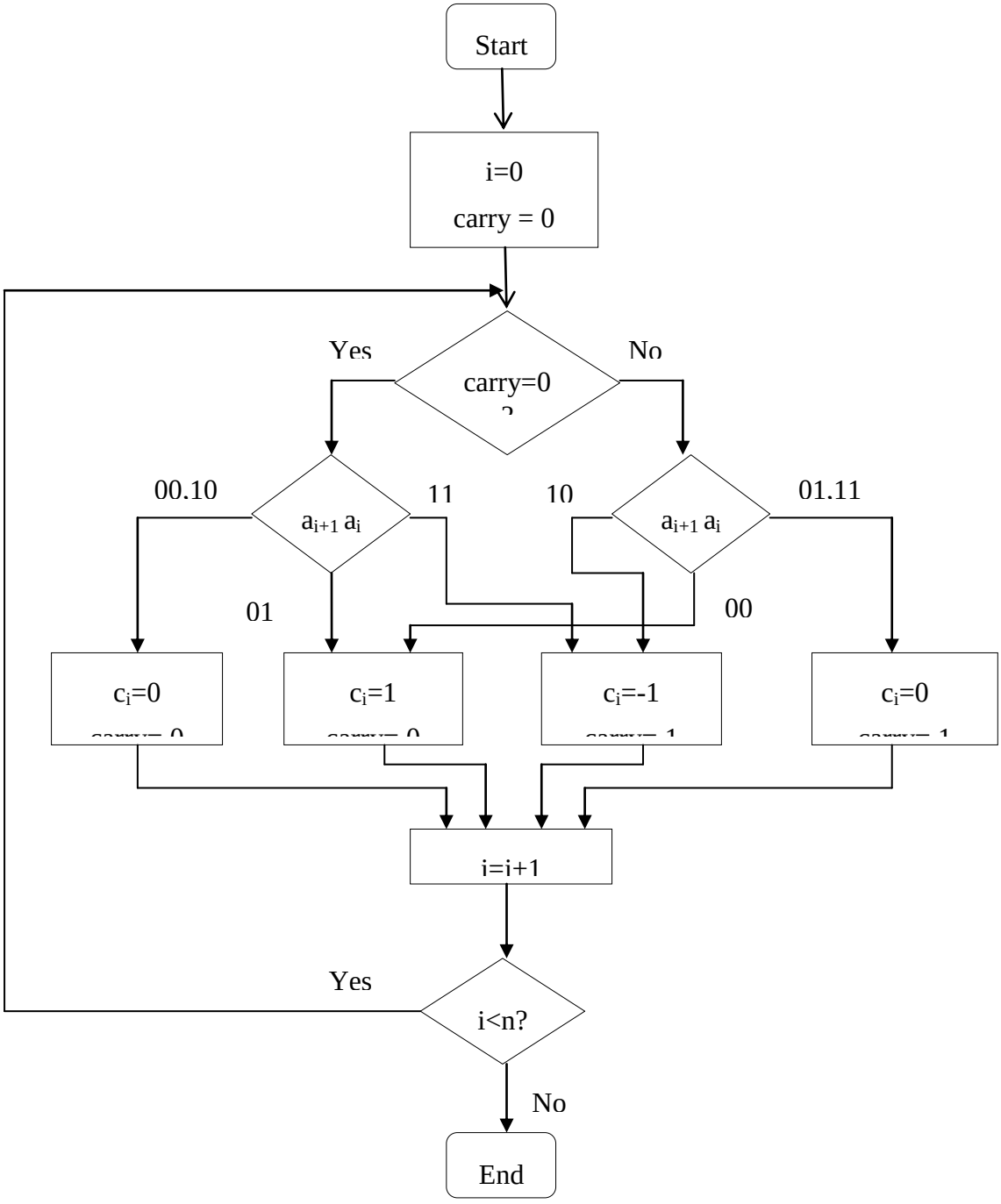

Figure 2.8 Flowchart for conversion of 2's complement into CSD

From the above example, it is clearly visible that the non-zero bits are decreases 6 to 3, which is ½ of the original binary expression and without any hampering the decimal value. This technique is very much useful to reduce the complexity of calculation. In Table 2.1 some basic comparison of 2's complement number with CSD is given to show the reduction of non-zero bits.

In the preceding section, this above method is used to design an FIR filter. The main application is done for the quantized coefficients of the FIR filter. The MATLAB generated coefficients are generally has a huge fraction length, that values are truncated up to 16 digit fraction length for converting the value in fixed point. The detail process is mentioned in below section.

Table 2.1: 2's complement vs CSD comparison

Decimal Number	2'Complement	CSD Representation
4	100	100
3	011	$10\bar{1}$
2	010	010
1	001	001
0	000	000
-1	111	$00\bar{1}$
-2	110	$0\bar{1}0$
-3	101	$\bar{1}01$
-4	100	$\bar{1}00$

2.6 Filter Design Using MATLAB FDA Tool

MATLAB is a well-known tool in all engineering domain and a MathWorks product. The Filter Design and Analysis Tool (FDA Tool) provides a very good user interface to the designer for designing and analyzing filters easily [20]. FDA tool enables a designer to design any kind of digital filters by just providing the desired filter

specifications. FDA tool also offers tools for investigating filter properties, such as magnitude and phase response and pole-zero values.

The following GUI is the user interface of the FDA tool. The upper half in the GUI displays the information of the provided filter specifications and responses of the designed filter. In the left side of that region, displays filter properties, such as filter order, structure, number of sections are used, stability of the filter, etc. It also provides access to the filter manager for working with multiple filters. In the upper right, the filter display region shows various filter responses, such as magnitude response, group delay and filter coefficients.

The lower part of the GUI is the user interactive portion for filter designer. The Design Panel is in the lower where need to provide the desired filter specifications. A view of FDA tool user interface is shown in Figure 2.8.

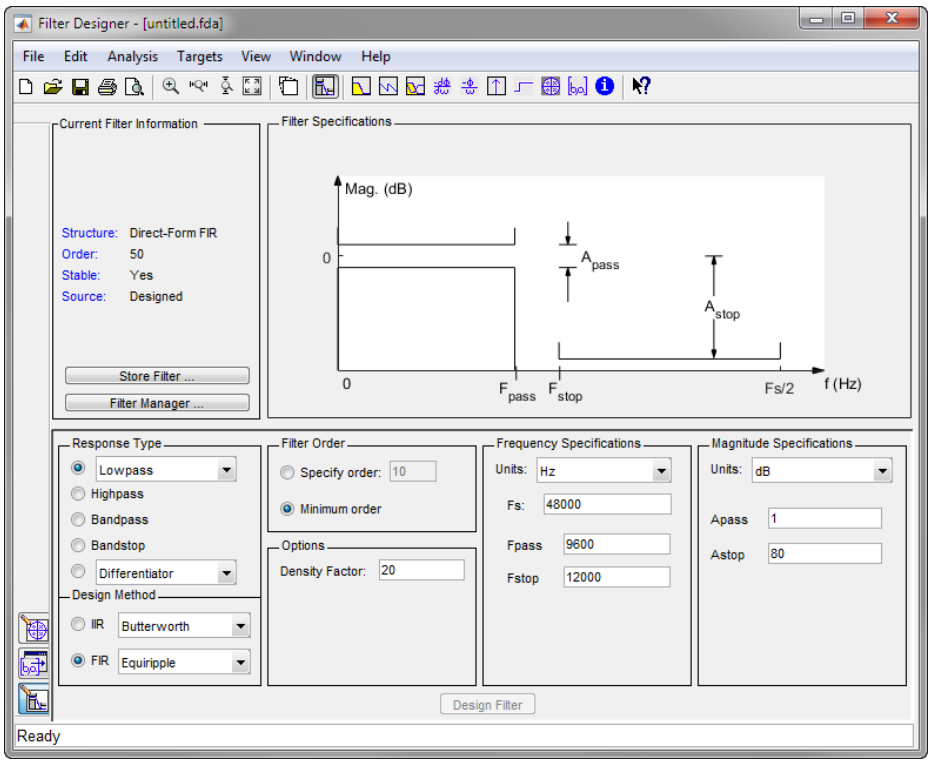

Figure 2.9 MATLAB FDA tool user interface

2.7 Realization of an FIR Filter

The realization of an FIR filter can be achieved by [20] following design procedure:

a) Choosing a filter structure

b) Quantize using fixed-point and floating-point arithmetic.

c) Choosing number representation format, e.g. signed magnitude, two's

complement

d) Choosing between serial and parallel processing
e) Generate the software code to design a hardware circuit, which one performs the actual filtering.
f) Verifying the simulation that the resulting design meets given performance specifications.

2.8 Design Steps

The detailed step by step [20] design procedures of the desired FIR filter are as follows:

a) "Lowpass" is chosen from the dropdown menu under "Response Type" and "Equiripple" under FIR Design Method. In general, when anyone wants to change the design method or response type, the filter parameters and filter display region update automatically.
b) Then the filer order has been specified by entering 5, 10, 20, and 30 (4 different times significantly).
c) There is a density factor option in the FIR equiripple filter which used control the density of the frequency grid. Increasing this value design a filter which more closely approximates of an ideal equiripple filter, but in that case more time is required as the number of computation increases. This value can be kept at 20 by default.
d) Normalized (0 to 1) has been selected in the drop down menu of frequency specifications area.
e) Wpass and Wstop values are given 0.4 and 0.6 in the frequency specifications area has been chosen.
f) In the magnitude specifications area the Wpass and Wstop having positive weights, one per band, those values are used during optimization of the FIR equiripple filter these values are kept 1 in this design.
g) After setting the design specifications, the Design Filter button has been clicked at the bottom of the GUI to design the filter. The magnitude response of the filter is displayed in the Filter Analysis area as shown in Figure 2.9.
h) After getting the magnitude response the filter is quantized to get the nearest rounding off value (truncated value) of the magnitude responses as shown in

Figure 2.10. The filter coefficients are in floating point format (real value), those real values are not synthesizable into FPGA. It needs to convert corresponding fixed point format, this tool has this opportunity to this by quantizing the filter using the fixed-point algorithm and generate truncated coefficients, as shown in Figure 2.10.

i) After generating all the required coefficients, using the target button the respective VHDL code is generated using fixed point value as shown in Figure 2.11. First time using multiplier then second time using CSD. This two VHDL code simulated and synthesized result compared in the next chapter.

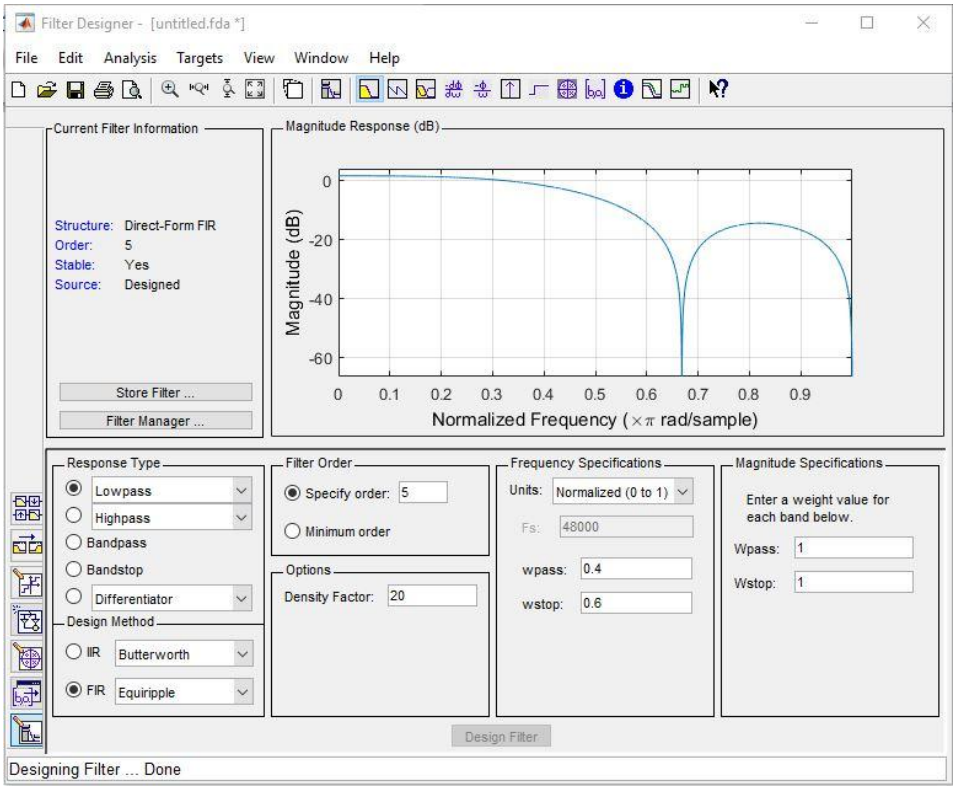

Figure 2.10 MATLAB design of low pass FIR filter using FDAtool box window

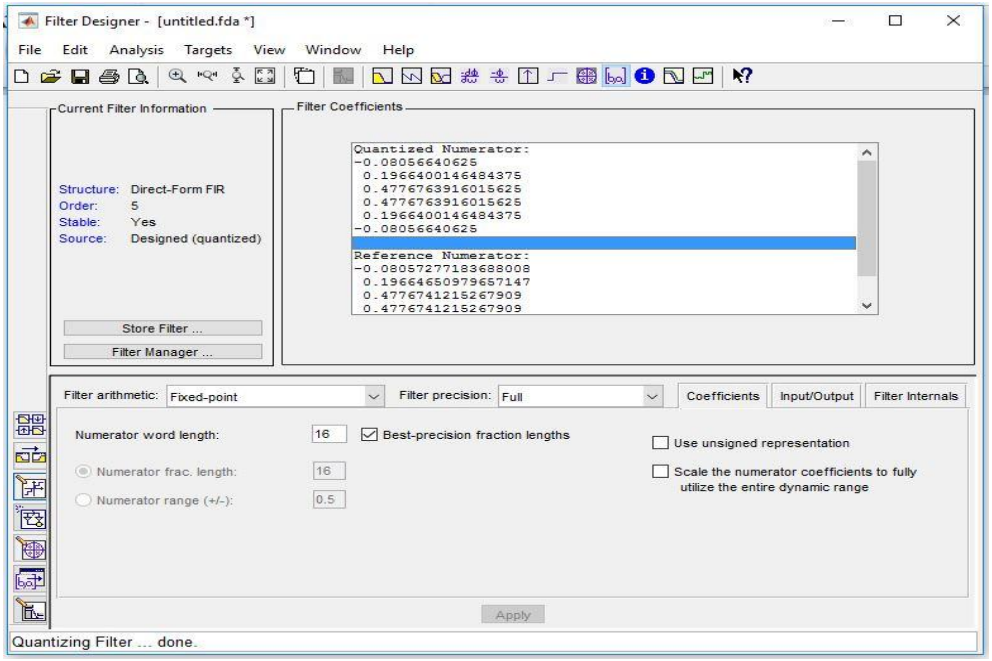

Figure 2.11 Quantized and normally obtained filter coefficients

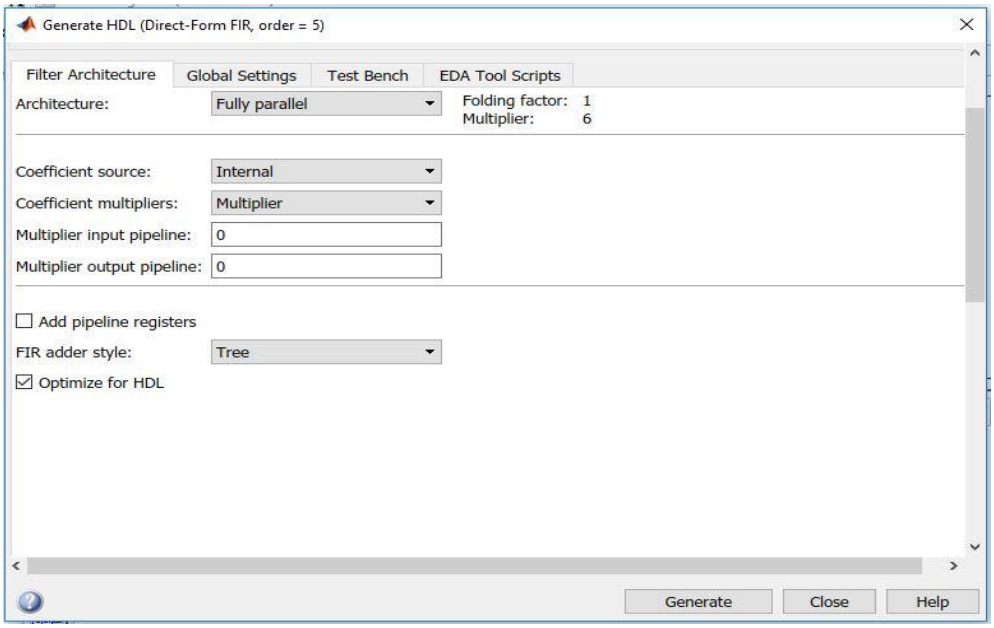

Figure 2.12 MATLAB FDA tool HDL code generation window

2.9 Restriction and Assumption

There are certain assumptions and restrictions in this implementation which are as follows:

a) The input and output are in the digital (signed binary) form.

b) Input considered 16 bit signed binary and output considered 32 bit signed binary

MATLAB generated coefficients quantized for best precision 16 digit fraction length

has been taken, that means after point (.) there are 16 digits which are truncated value of a big fraction length. This floating point value is converted into 16 bit signed binary using the following process,

a) At first, this floating point value multiplied with 2^{16} to convert it an equivalent fixed point number (rounding off value considered).
b) If the value is positive then it normally converted to binary, but if it is negative then 2's complement value is considered at the time of conversion in CSD.

For a 5th order low pass FIR filter are designed using MATLAB FDA tool and the respective fixed point, binary (2's complement) and CSD converted bits are also shown in Table 2.2.

Table 2.2: Coefficients representation in different formats for a 5th order low pass FIR filter

Quantized Filter Coefficients	**Fixed Point Representation (Using 16-bit fraction length)**	**Corresponding 16 bit signed binary(2's complement) digit**	**Corresponding 17-bit CSD digit**
-0.080566406	-5280	1110101101100000	$0000\bar{1}0\bar{1}00\bar{1}0\bar{1}00000$
0.196640014648438	12887	0011001001010111	$00\bar{1}01\,00\bar{1}0\bar{1}0\bar{1}0\bar{1}00\bar{1}$
0.477676391601563	31305	0111101001001001	$0100\bar{1}01001001001$
0.477676391601563	31305	0111101001001001	$0100\bar{1}01001001001$
0.196640014648438	12887	0011001001010111	$00\bar{1}01\,00\bar{1}0\bar{1}0\bar{1}0\bar{1}00\bar{1}$
-0.080566406	-5280	1110101101100000	$0000\bar{1}0\bar{1}00\bar{1}0\bar{1}00000$

2.10 Summary

The detail method of MATLAB FDA tool based FIR filter design using CSD is discussed elaborately in this chapter. The detail features, structures of an FIR filter and design methods are discussed in the subsections. The CSD technique in detail with its advantages and how this computation method is used to design an optimized FIR filter is also shown above, which justifies the objective of this thesis.

Chapter 3

FPGA Implementation of FIR Filter

3.1 Introduction

Over the past few decades Field Programmable Gate Array (FPGA) based FIR filter design become very popular in the field of digital signal processing because of design process simplicity, flexibility, and a huge choice of hardware options. The main advantage of using FPGA over DSP is an FPGA does not have a fixed hardware structure; it can be programed according to the choice of the designer. Although the number of logic cells are fixed but the interconnections among them are defined by user [11].

In this chapter the FPGA implementation process of the designed low pass FIR filter is described and the simulation and synthesis report is compared for different order filter in details for both multiplier and CSD based design. In this thesis work Quartus II synthesis tool from Altera is used and the design implementation done choosing the Cyclone IV (EP4CE15F17C8) FPGA IC. The detail steps of implementation and result comparison is done in the subsequent chapters.

3.2 FPGA HDL Based Design Process

To design the multiplierless FIR filter on FPGA, Very High-Speed Integrated Circuit Hardware Description Language (VHDL) is used as the designing language. The Hardware Description Languages (HDL) is used to define the behavior and structure of system (digital and mixed signal system) [8].

Advantages of using HDL to design FPGAs:

a) Top-down approach: HDLs are used to make complex designs. The top-down approach to system policy supported by HDLs is advantageous for large projects for which it requires many engineers working together.

b) Functional simulation in the design flow: One can verify the behavior or functionality of the design early in the design flow at the time of HDL simulation.
c) Synthesis of HDL code to gates: One can synthesize the hardware description language to a design implemented with logic gates, it works automatically after synthesis. This step decreases project time by excluding the need to define each gate manually.
d) Early testing of different design implementations: HDLs allows one to check different implementations of the design early in the design flow. Then the synthesis tool can be used to perform the logic synthesis and optimization into gates.

3.2.1 Basics of VHDL

VHDL stands for Very High Speed Integrated Circuits (VHSIC) Hardware Description Language (HDL) [8]. This coding language is used to describe digital electronic systems. It was first invented at United States Government's VHSIC program in 1980 and was accepted as a standard for describing the function and structure of Integrated Circuits (ICs). Soon after, it was advanced and adopted as a standard by the Institute of Electrical and Electronic Engineers (IEEE) in the US (IEEE-1076-1987) and in other countries. The VHDL continues to evolve, although new standards have been developed (VHDL-93, 2008) most commercial VHDL tool use 1076-1987 version of VHDL. VHDL enables the designer to:

a) Define the design structure and the behavior of the circuit and systems, and to specify how the sub-systems are interconnected.
b) Define the function of the design similar to the C language.
c) Simulate and verify the design before fabrication, so that the designer get a chance to quickly compare the substitute approach and test for exactness without the delay and extra expense of multiple prototyping.

VHDL is a general purpose, C-like programming language with additions to model concurrent and sequential both execution flows. VHDL allows four styles of programming- (a) structural, (b) data flow, (c) behavioral, and (d) mixed programming.
The first one, structural, is the most commonly used structure of the integrated circuit and

its interconnections. The data flow model allows the designer to define the total system using logical computation. The behavioral model permits the designer to test concepts quickly, where the designer can state the high-level function of the design without taking much care of how it will be done structurally. This can be applicable for quick design and very attractive for medium speed and low-volume applications, where the designer needs not to be an expert. Finally the mixed is a combination of the any before mentioned design process in a single design.

3.3 Simulation and Synthesis Tools

To simulate and implementation of the the multiplier less FIR filter on FPGA, the following tools are used

a) Altera-Modelsim for simulation and Altera Quartus II (for HDL synthesis).
b) The hardware implementation is done using Altera Cyclone IV

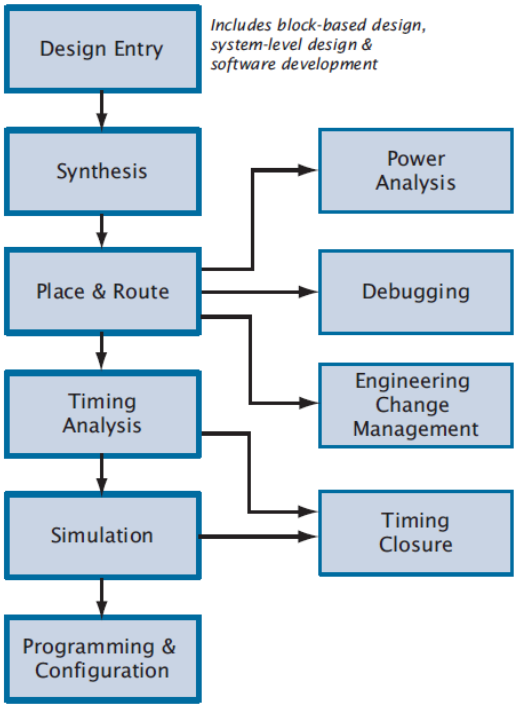

Figure 3.1 Altera Quartus tool design flow

3.3.1 Altera Quartus II

The Altera Quartus II front end design environment available for any digital or System on

Chip (SoC) design. This tool offers all the features to assist in FPGA and CPLD based system design. Quartus II automatically identify certain types of HDL (VHDL/Verilog) code for analysis & synthesis.

The Quartus II tool uses inference because Altera megafunctions are optimized for Altera FPGA/CPLD devices. For some architecture-specific features, such as RAM and DSP blocks, the designer must use Altera megafunctions. The Quartus II tool maps the following logic to megafunctions during HDL synthesis:

Counters, adders/ subtractors, multipliers, multiply-accumulators and multiply-adders, RAM and shift resistors.

3.4 FPGA: An Overview

An FPGA is a fully reconfigurable logic chip[10]. The hardware chip consists of a series of logic gates, unlike the traditional gate arrays. In the traditional array, all those gates are specified and already connected internally at the manufacturing stage. The FPGA is programmable as the choice of the designer anytime which is an advantage of to develop or change the prototype model to be implemented physically.

3.4.1 Programmable Logic

Programmable logic can be well-defined as a device with configurable logic gates and flip-flops are linked together by programmable interconnects. The memory cells are used control and define the logic functions and how the different logic functions are interconnected.

There are some major programmable logic architectures available in the market. Each architecture typically has different vendor-specific sub-variants. The major categories are [10]:

a) Simple Programmable Logic Devices (SPLD)
b) Complex Programmable Logic Devices (CPLD)
c) Field Programmable Gate Arrays (FPGA)

3.4.2 FPGA - Field Programmable Gate Array

The FPGA is advancing rapidly and very popular hardware of the future of computing.

Already development has shown that it can massively reduce the price of specialized system development and it can compete on a variety of attributes with the top range commercially available microprocessors. Its initial role in rapid system prototyping is still important but in recent times it has grown in importance as a platform for implementing complete solutions. There are four main kinds of FPGAs commercially available in the market like hierarchical PLD, symmetrical array, row-based, and sea-of-gates (Figure 3.9). In all of these FPGAs, the interconnections and way of programming varies.

The basic architecture of FPGA consists of a 2D array of logic blocks and flip- flops which gives the opportunity to the developer to configure each of the logic blocks, the inputs/outputs, and the interconnection between blocks. FPGA families differ from each other by the physical means for implementing user programmability, the arrangement of interconnection, and basic functionality of logic blocks [10].

At present, there are mainly four technologies are available in FPGA: static RAM cells, anti-fuse, EEPROM, and EEPROM. As per the application the features are chosen.

a) Static RAM technology**:** The static RAM FPGA is constructed by pass transistors, transmission gates (TG), and multiplexers that are organized by SRAM cells. The main advantage of this technology is that it allows fast in-circuit reconfiguration. The size of the chip is a biggest disadvantage of RAM technology.

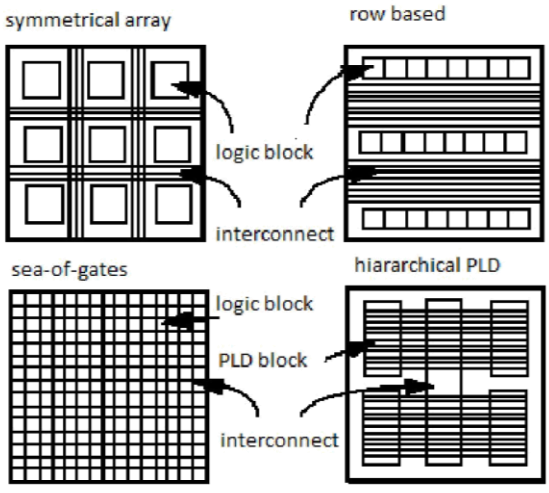

Figure 3.2 Classes of FPGAs

b) Anti-Fuse Technology**:** It is cheaper than the RAM technology; this device is one time programmable device, which is the main disadvantage of this technology.

The term anti-fuse means it works only when the fuse connection breaks, it normally retains high-impedance state and it can be programmed into low impedance state or "fused" state.

c) EPROM / EEPROM Technology**:** Both the technologies are reprogrammable (Electrically programmable). In case of EPROM, UV light is required to erase any program but is case of EEPROM it is electrically erasable.

In Table 3.1 some of the commercially available FPGAs are listed from different vendors.

Table 3.1: Commercial FPGA Technology

Company name	**Architecture**	**Logic block type**	**Programming technology**
Actel	Row-based	Multiplexer-based	Anti-fuse
Altera	Hierarchical-PLD	PLD block	EPROM
Atmel	Symmetrical array	Multiplexer-based	Anti-fuse
Quick Logic	Symmetrical array	Look-up Table	Static RAM

FPGA is an integrated circuit that can be programmed after manufacturing to use as any digital circuit determined by the designer. The key difference between FPGA and Application Specific Integrated Circuit (ASIC) is that the ASIC can only be used for a certain application. The major building blocks in modern FPGAs are made by Configurable Logic Blocks (CLBs), Interconnections, Input / Output Blocks (IOBs) and embedded blocks such as DSP blocks [9]. An overview of the architecture of modern FPGAs is shown in Figure 3.1.

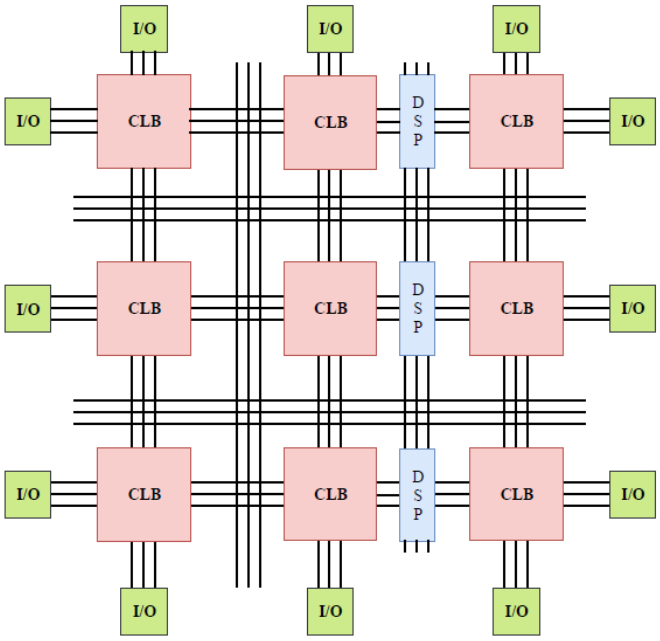

Figure 3.3 An overview of a typical FPGA architecture

The red boxes I the above figure indicate a number of CLB blocks distributed over an FPGA. The green boxes indicate input/output blocks of an FPGA. The blue boxes indicate DSP blocks. Vertically and horizontally distributed lines indicate interconnections between the blocks.

3.4.3 Configurable Logic Block

The Configurable Logic Block (CLB) [9] is used to implement any kind of digital logic functions and mathematical operations of any complex digital operations. Inside each CLB, there are so-called Configurable Logic Elements (CLEs) which in turn consists of Look-Up Table (LUT), D-flip flops and 2-to-1 multiplexers as illustrated in Figure 3.13 [5]. The purpose of a LUT is to implement mathematical operations such as addition. The LUT box and its three-inputs are shown in Figure 3.3.

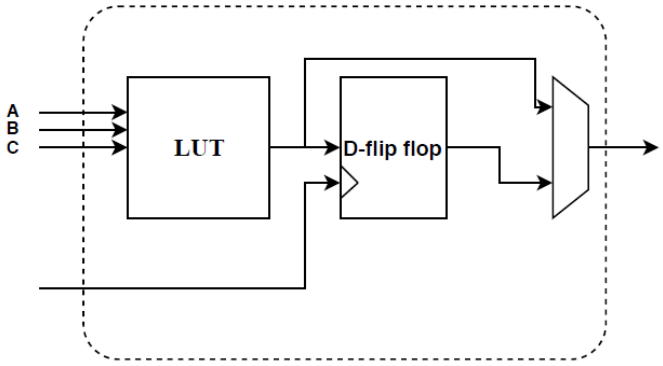

Figure 3.4 Basic internal structure of a typical CLB

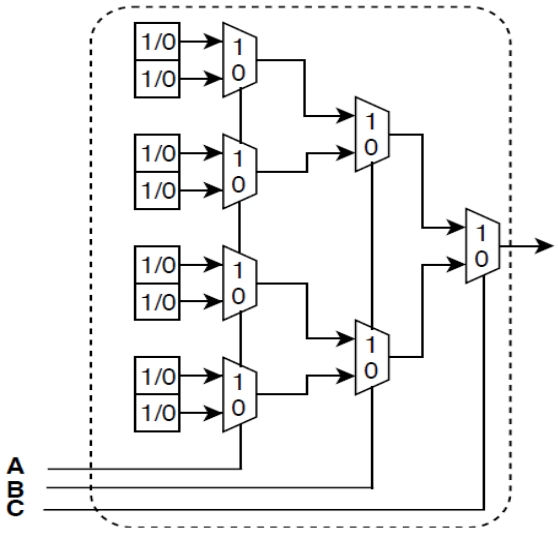

Figure 3.5 Basic internal structure of a LUT

The core of the FPGA consists of thousands of CLB copies connected with each other's. The LUT is made of a series of cascaded multiplexers where the LUT inputs are used as the select lines and the inputs to multiplexers is a 1-bit SRAM memory that is set to either 0 or 1. The internal structure of a LUT is shown in Figure 3.5. The three-inputs of a LUT control the selection of the 1/0 input bit of the multiplexers.

3.4.4 Interconnections

As shown in Figure 3.6, a big part of the FPGA's architecture is utilized by vertically and horizontally distributed interconnections [9]. These interconnections consist of switch boxes and connection boxes in order to get the desired connection. Each CLB is

surrounded by a group of connection boxes that in turn are connected to each other via a group of switch boxes, as shown in Figure 3.6. These connection boxes are used to connect different CLBs to each other and furthermore provide a connection to both I/O blocks and embedded blocks [9]. In Figure 3.6 SB refers to Switch Boxes and CB refers to Connection Boxes.

The placement of interconnections shows an important role to determine the flexibility and efficiency of the FPGA. Therefore, manufacturers nowadays offer flexible construction of interconnections including sets of both short and long wires in order to efficiently perform any digital circuit.

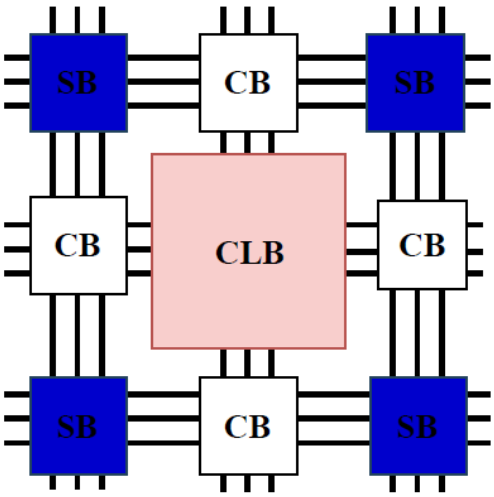

Figure 3.6 Simplified networks of interconnections

3.4.5 Input Output Block (IOB)

Input / Output blocks are placed all around FPGA's edges [9, 10], as shown in Figure 3.3 where the IOBs function as a communication interface between the internal signals and the external pins.

3.4.6 DSP Block

Although a Configurable Logic Block is capable of performing arithmetic operations and storing data, it would be too slow and utilizes a huge amount of resources because a CLB is designed to be relatively general in order to perform different types of functions. As the demand for the FPGA is increasing, modern FPGAs offer embedded blocks to perform

specific functions, some of which are memory blocks and DSP blocks. The introduction of these embedded blocks clearly reduces area utilization, power consumption and as a consequence results in better performance. Although the internal structure of DSP blocks and its related details are different depending upon manufacturer and device version, the overall architectures of most DSP blocks look alike. A DSP block consists of an adder and multiplier followed by an accumulator. The aim of introducing DSP blocks is to enhance the performance of these arithmetic operations. Also, there are specific connections in each DSP block that are used to connect multiple DSP blocks to each other which is used to implement an efficient FIR filter [10].

In Figure 3.7, the total FPGA design flow is shown from the design entry to hardware programming file generation.

3.5 Altera Cyclone IV FPGA Kit

The Altera Cyclone IV Trainer Kit (VPTB– 23), made by VI Microsystems uses an EP4CE15F17C8 FPGA. It has the following feature, such as-

a) EP4CE15F17C8 Altera Cyclone IV FPGA features:
   15408 Logic elements, 504Kbs of Embedded Memory, 56 Embedded Multipliers, 4Nos of PLL.
b) 16 Nos of digital slide switches for input.
c) 16 Nos of discrete LEDs for digital output.
d) 16× 2 LCD display.
e) One Limit Switch for giving manual clock.
f) FPGA configuration through.
   On-board USB blaster to configure FPGA/On board Flash Prom EP4CS (programmable through USB Blaster)
g) 2Nos of 20pin header with 30 IO lines
h) On board programmable traffic light controller, PLL oscillator (1MHz - 100MHz) using jumpers.
i) 4 Nos of 7 segments LED display
j) One relay and Buzzer provided.
k) Stepper & DC motor driver provided on board (Motor Optional)

l) 4x4 matrix key provided

m) SPI Based Analog to Digital Converter

Resolution: 12 bits; Number of Channel: 2 Channels; Input Range: 0 to 5V; Speed: 1 Msps

n) SPI Based Digital to Analog converter

Resolution: 12 bits; Number of Channel: 2 Channels; Output Range: 0 to 5V; Speed: 8.5 µS settling time

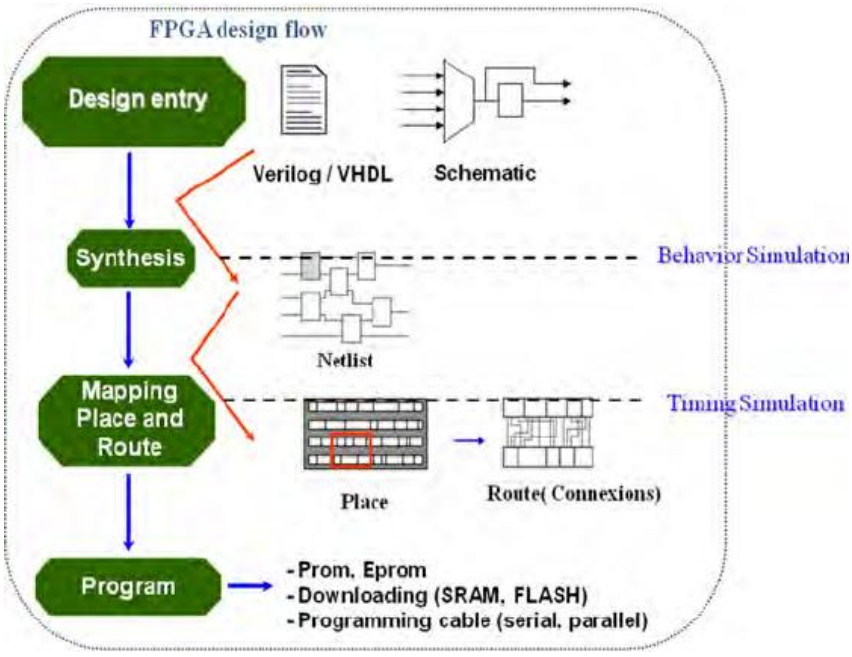

Figure 3.7 A typical FPGA design flow

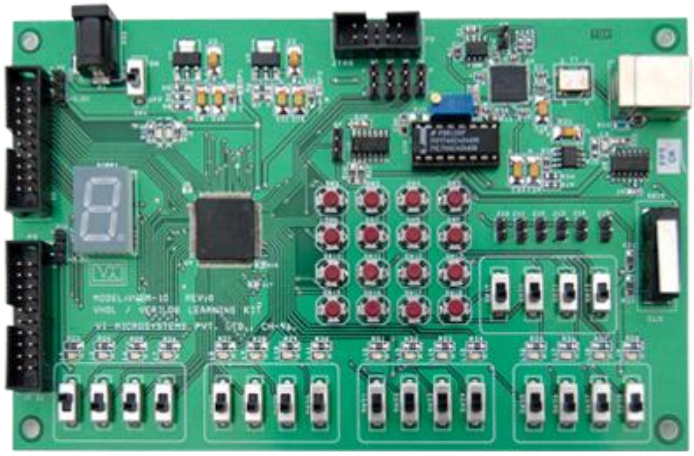

Figure 3.8 VPTB – 23, Altera Cyclone IV Trainer Kit

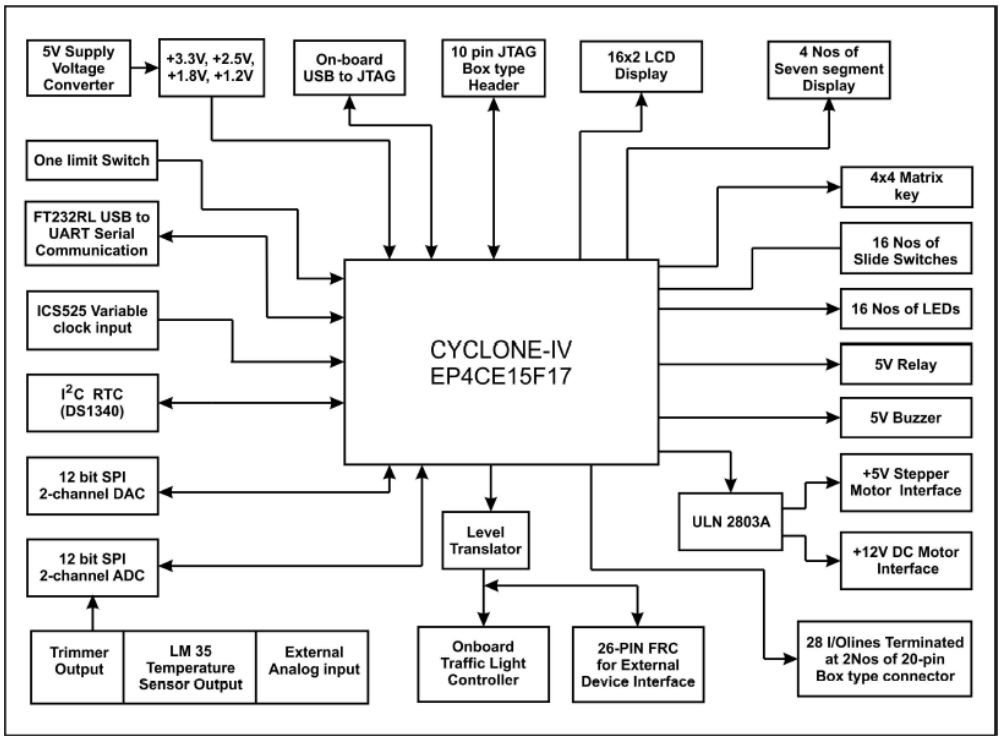

Figure 3.9 Cyclone IV IC Block Diagram

3.6 FPGA Implementation of Designed Methodologies

In the previous chapter for different order low pass FIR filter designed using the settings mentioned on the section 2.7 by using MATLAB FDA tool and the VHDL code has been generated for both multiplier and CSD based design.

Step 1: The MATLAB generated VHDL codes first compiled and simulated using Quartus II inbuilt Altera-Modelsim tool to verify the functionality matching for both the design (Multiplier and CDS) for same order.

Step 2: After verification the synthesis process has been done of those codes using Altera Quartus II by selecting the particular FPGA board (i.e. Altera Cyclone IV EP4CE15F17C8) and all the synthesis reports are observed for each of the design.

Then a comparison for the same design, order with different computation methodology (multiplier and CSD) has been compared to check the effects of those different computation strategies to fulfill the target of this thesis.

3.7 Simulation Results

In this section, the VHDL codes generated via MATLAB are simulated using Altera-

Modelsim to check whether the codes are working properly and yielding the desired result for both multiplier and CSD based designs. Here same input is used in both the cases and all the other parameters like clock frequency, simulation time, duty cycle have been kept as same.

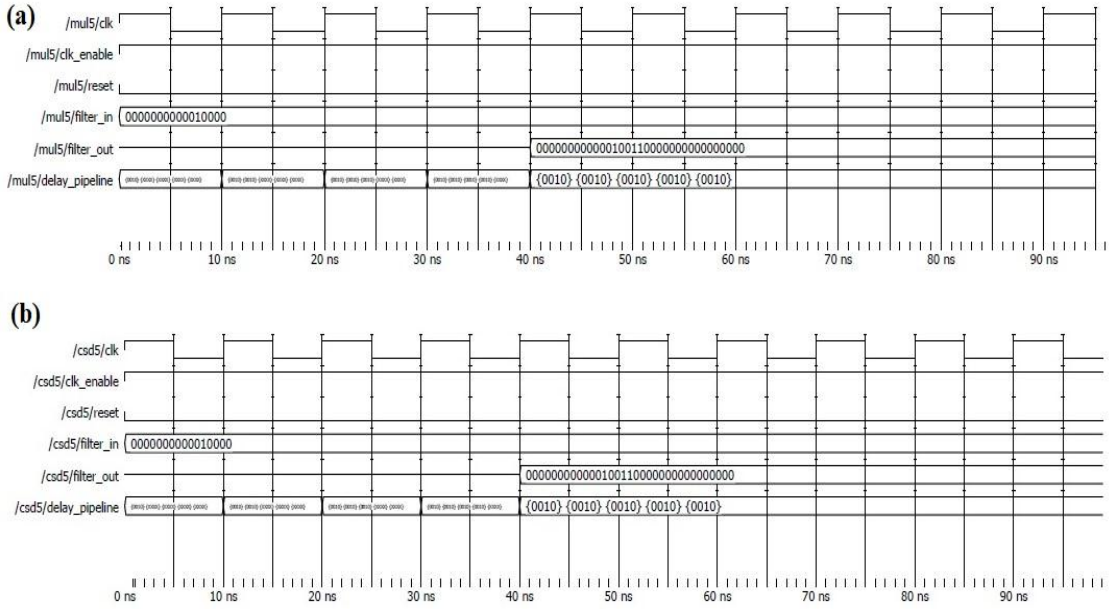

Figure 3.10 Simulation diagram of 5th order low pass FIR filter using (a) multiplier, (b) CSD

A small decimal number "16" is taken as input; the 16 bit signed binary representation of this number is "0000000000010000". Clock frequency taken 10ns, the duty cycle has been kept 50% (standard) and the simulation done for 100ns as shown in Figure 3.10. The output of that 5th order low pass filter is "1245184" in binary "00000000000010011000000000000000" (the output is taken as signed 32-bit binary format). The output value is the same for both the design cases (multiplier and CSD). So, it is clear that there is no violation of the desired outcome between those two different computation methods.

In Table 3.2 the quantized (truncated values, for best precision 16 bit fraction length has been taken) filter coefficients of a 5th order low pass FIR filter generated on MATLAB, are noted and the respective fixed point, binary and CSD converted bits are also shown on that table.

3.8 Synthesis Report Using Multiplier Method

After simulation and verification check then codes are synthesized using Quartus II for the Altera Cyclone IV hardware (IC no. EP4CE15F17C8). The detailed synthesis report for MATLAB generated VHDL code for 5th order low pass FIR filter multiplier based design is given below.

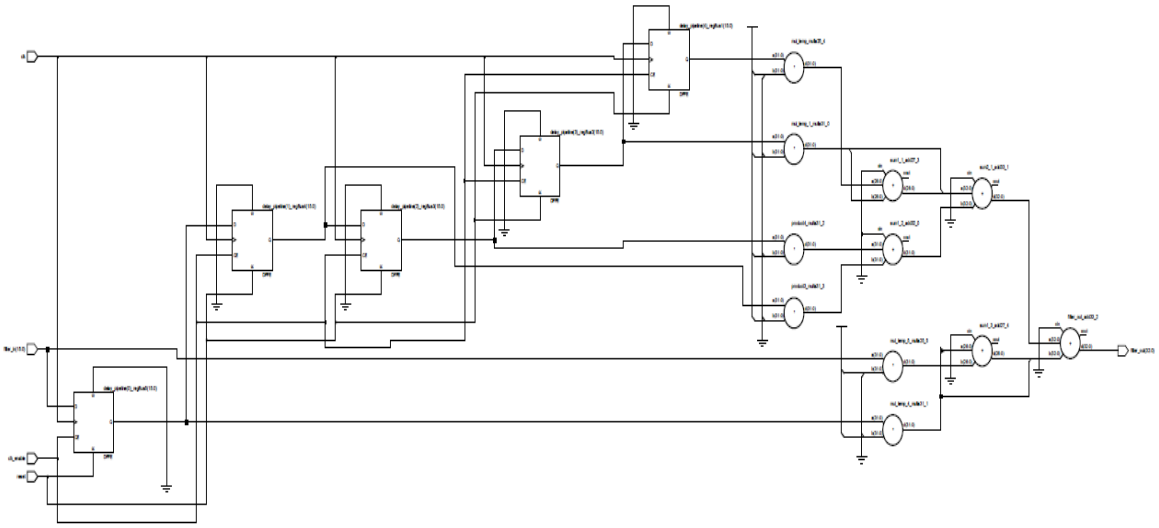

Figure 3.11 RTL schematic of 5th order low pass FIR filter using multiplier based design

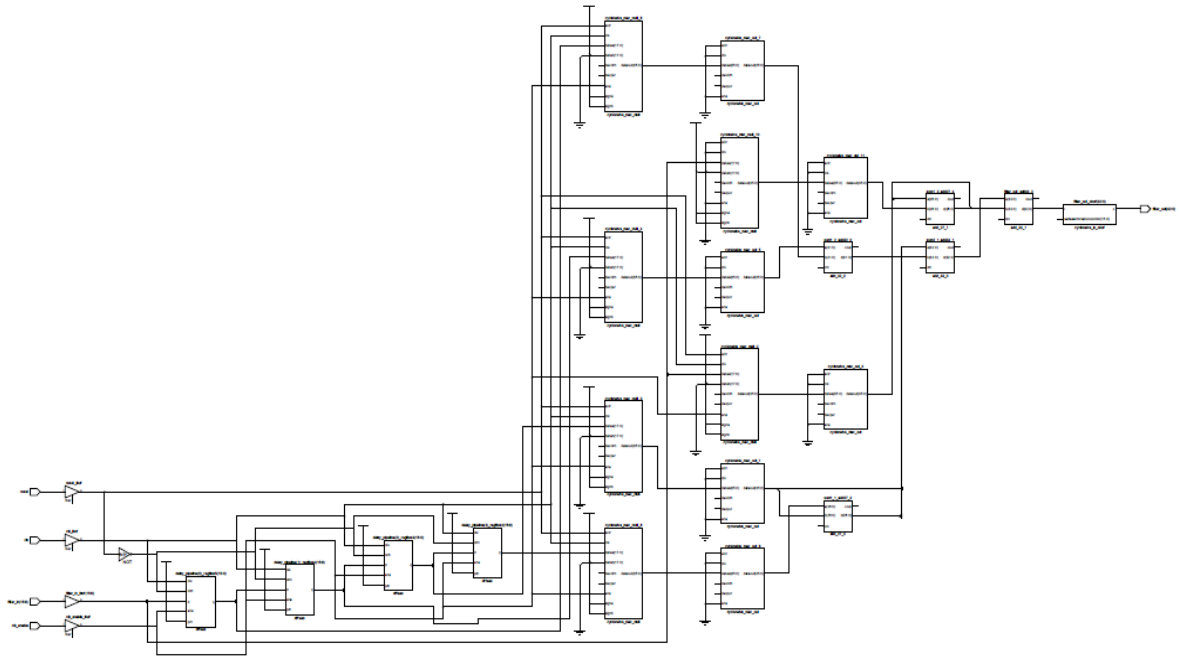

Figure 3.12 Technology schematic of 5th order low pass FIR filter using multiplier based design

In Figure 3.11 the RTL view of multiplier based 5^{th} order low pass filter is shown. The register-transfer level (RTL) view is a design abstraction in a digital circuit design, which basically shows the logical construction view and their interconnections as per the theoretical description. Here all the blocks are clearly visible which commonly use to draw a filter structure.

In Figure 3.12 the technology map viewer of the FPGA design is shown. The technology map viewer shows the practical internal connection inside an FPGA (such as device logic cells and I/O ports) of the design. The structure is basically constructed using LUT, accumulator, IO blocks and other registers. In Table 3.3 the hardware utilization of the selected FPGA board is shown.

Table 3.2: Device Utilization for Altera Cyclone IV EP4CE15F17C8 (Using multiplier)

Resource	**Used**	**Available**	**Utilization(%)**
IOs	52	166	31.33
LUTs	152	15408	0.99
Registers	64	15408	0.42
Memory Bits	0	516096	0.00
DSP block 9-bit elements	12	112	0.00
Total logic elements	808	15408	5
Total combinational functions	797	15408	5
Dedicated logic registers	80	15408	<1

3.9 Synthesis Report Using CSD Method

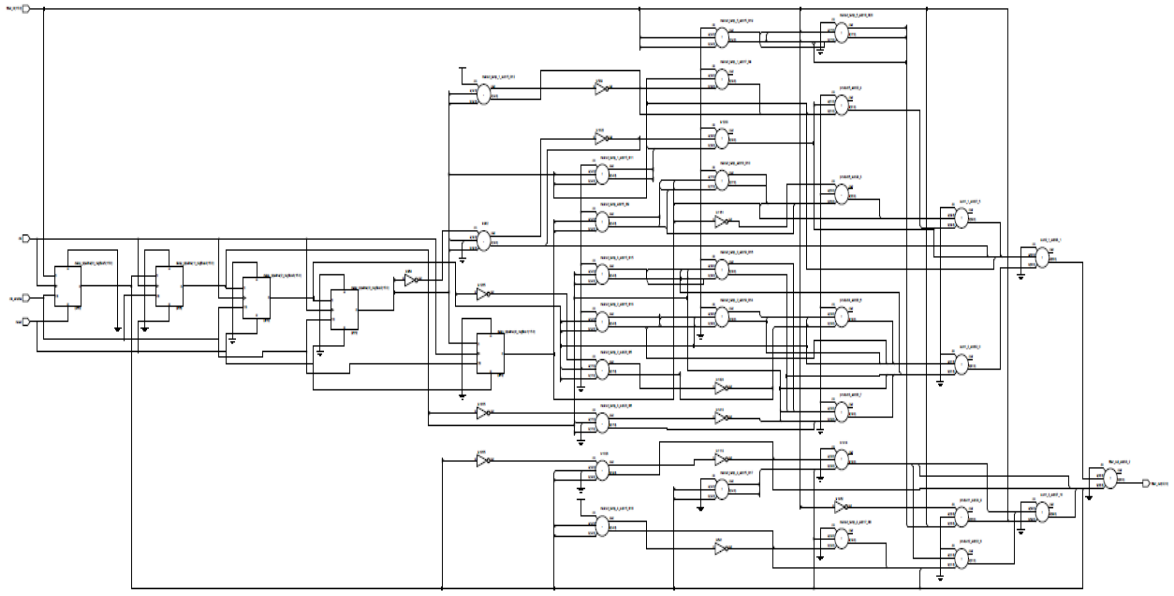

Figure 3.13 RTL schematic of 5th order low pass FIR filter using CSD based design

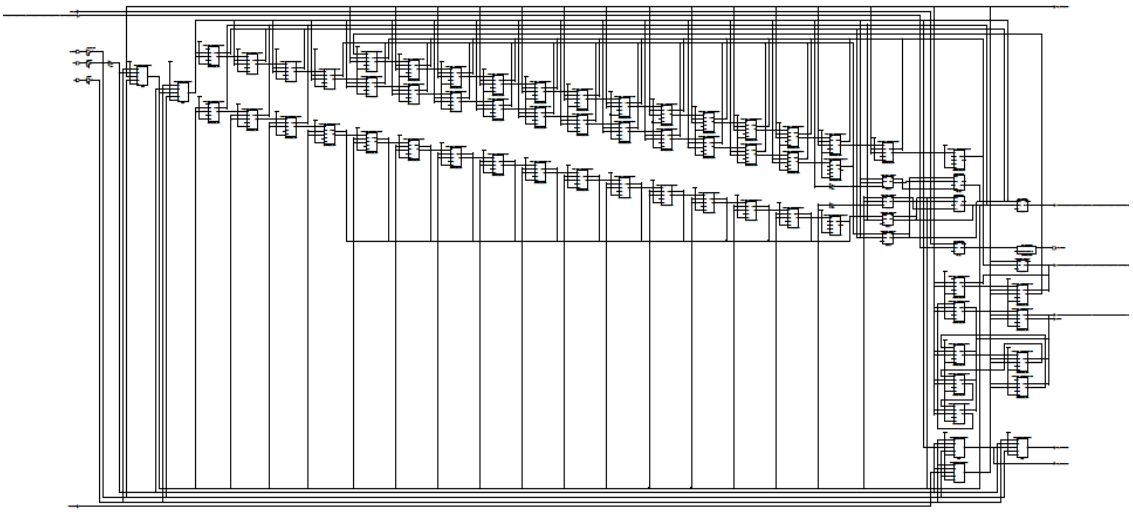

Figure 3.14 Technology schematic of 5th order low pass FIR filter using CSD based design

For 5th order FIR filter CSD based design, the design reports are shown in Figures 3.13-3.14 and Table 3.4.

From the RTL view and technology map view of CSD it can observe that the hardware structure is complex comparing to multip0lier based design. But, in the actual report the hardware consumption reported is lesser than the multiplier as shown in Table

3.6. The power consumption and cell delay also lesser than the multiplier based design which is shown on Table 3.5 and discussed in the below section

Table 3.3: Device Utilization for Altera Cyclone IV EP4CE15F17C8 (Using CSD)

Resource	Used	Available	Utilization (%)
IOs	52	166	31.33
LUTs	636	15408	4.13
Registers	80	15408	0.52
Memory Bits	0	516096	0.00
DSP block 9-bit elements	0	112	0.00
Total logic elements	866	15408	6
Total combinational functions	831	15408	5
Dedicated logic registers	80	15408	<1

3.10 Synthesis Report Comparison and Discussion

The comparison of the detail synthesis report of multiplier and CSD based filter design has been presented in Table 3.5 and Table 3.6. As evidenced by the values in the Table 3.5, it can be inferred that power consumption in most of the cases of CSD is low as compared to the case of multiplier. On the other side, in case of timing report the cell delay also decreases in case of CSD, which fulfills the thesis objective.

Table 3.4: Comparison of power, time Utilization between multiplier and CSD based filer

Filter Order	Power Utilization (mW)		Time Delay (%)	
	Multiplier	CSD	Multiplier	CSD
5	88.09	86.48	76.65(cell),24.35(net)	33.5(cell),66.5(net)
10	87.61	85.56	65.58(cell),34.42(net)	41.23(cell),58.77(net)
20	76.99	79.01	65.58(cell),34.42(net)	41.23(cell),58.77(net)
30	85.57	84.81	65.58(cell),34.42(net)	49.15(cell),50.85(net)

In case of time delay, there are two terms in the result in the synthesis report those terms are

Table 3.5: Comparison of area utilization for different order FIR filter

Device Used	Filter order	Percentage Utilization for multiplier based design (%)	Percentage Utilization for CSD based design (%)
IOs	5	31.33	31.33
	10	30.72	30.72
	20	30.72	30.72
	30	30.72	30.72
LUTs	5	0.99	4.13
	10	1.13	4.11
	20	1.84	5.82
	30	2.84	8.96
Registers	5	0.42	0.52
	10	0.93	1.04
	20	1.87	1.97
	30	3.01	3.12
DSP block 9-bit elements	5	10.71	0
	10	10.71	0
	20	17.86	0
	30	28.57	0
Total logic elements	5	5	6
	10	5	6
	20	9	8
	30	13	13
Total combinational functions	5	5	5
	10	5	5
	20	8	7
	30	12	11
Dedicated logic registers	5	<1	<1
	10	1	1
	20	2	2
	30	3	3

a) Cell Delay- It is basically the quantity of time delay from input to its output of a logic gate in a path. In broad way total time to get the final value from a logic circuit.

b) Net Delay- It is the amount of propagation time delay from the output of a cell to the input of the next cell. Parasitic capacitance of the circuit individual circuits itself and the interconnection path between the two cells is mostly responsible for this delay.

In Table 3.6, the area utilization report for different order (5, 10, 20, and 30) filter with same design parameters are shown for both the multiplier and CSD based design. The corresponding magnitude responses are shown in Figure 3.15. It can be easily justified from the below table that no DSP blocks are used in case of CSD based design so, the total percentage of area utilization decreases as compared to multiplier based design.

If the fraction length is taken lesser than the current experimental fraction length than the area utilization also reduces further but it can affect the perfection of operation.

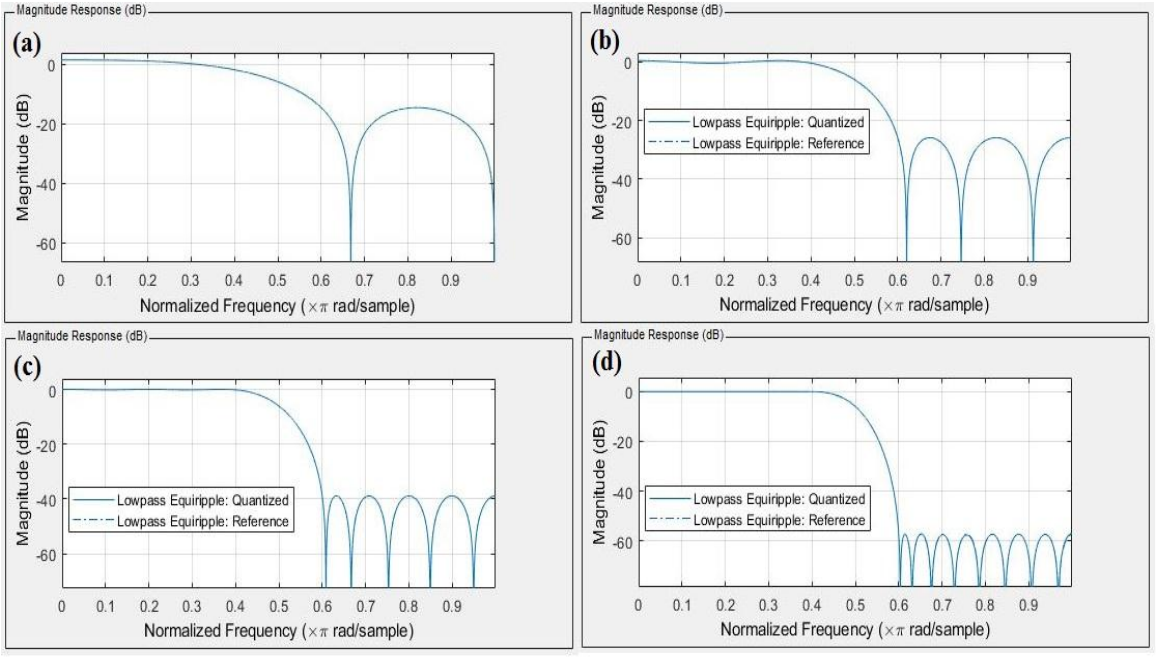

Figure 3.15 Normalized magnitude responses for different order low pass FIR filters (a) 5, (b) 10, (c) 20, (d) 30.Stop band 0.4π and pass band 0.6π

This table made for very small order filter for a large design more significant changes can

be noticed and from the Figure 3.15, it can easily be noticed that if the filter order increases then the responses will improve much better. In practical design, very higher order filters are used for signal processing purpose which gives better accuracy and perfection but it takes a large area, consumes much power and delay time also increases as shown in Table 3.5 and 3.6. More hardware means more making cost. According to the "No Free Lunch" theorem [37], there is no single option to optimize all things using any single algorithm, every strategy has some pros and cons. The CSD based design is to meet some of the design goals but not fulfill all the desired goals. This algorithm is basically focused on reducing non-zero terms in filter coefficients which mean less computation, less computation time, less dynamic power dissipation, less hardware requirement etc. There can be further many theorems can be applied to fulfill the further goals discussed in the future work section.

3.10 Summary

In this chapter detail about VHDL design flow, FPGA technology and the step by step implementation process of MATLAB designed low pass FIR filter is shown in this chapter. The simulation result of a 5^{th} order low pass FIR filter discussed in details. The synthesis results for a 5^{th} order low pass FIR filter is shown and the report comparison between CSD and multiplier based design of different order low pass filters have been elaborately discussed in this chapter.

Chapter 4

Conclusion and Future Scope of Work

4.1 Conclusion

FIR filters are widely used in various signal processing applications and it can be implemented using programmable digital processors [3]. In case of realization of large order FIR filters the speed, cost, and flexibility are affected because of complex computations. Due to these issues FPGA is a good choice for the programmer over DSP hardware for digital filter design. The major drawbacks of conventional multiplier based FIR filter design methodology is to implement an FIR filter that takes large area, power consumption and time delay are also an issue [19]. In this work CSD based bit reduction technique has been introduced which used to design an optimized FIR filter. It helps to overcome those difficulties of conventional multiplier based method.

In this thesis, low-pass FIR filters of different orders are implemented on EP4CE15F17C8 Altera Cyclone IV FPGA using conventional multiplier and proposed CSD based technique. The experimental results for both the cases have been compared and the final outcome meets our expectations.

4.2 Future Scope of Work

The primary objective of this work is to optimize the filter design such that it can be implemented by utilizing less area, power and delay time without hampering the quality of the original response.

In this project work only FIR filter implementation is presented. The same idea can be applied to design an IIR filter also. To reduce the numbers of structural adder and multiplier block adders further, few well known techniques can be applied over CSD like, Horizontal Common Sub-expression Elimination, Vertical Common Sub-expression Elimination are some of the techniques which minimize the non–zero bits further [39-40]. Any nature-inspired optimization algorithm can be used to design a filter with

comparatively better response as well as reduce the non-zero coefficient bits. Some of those popular optimization algorithms such as flower pollination [41], cuckoo search [42], artificial bee colony [44], cosine modulated [43], swarm [45] optimization algorithms can be used for the application of filter design.

The finding of the work has been accepted for publication as per details given below:
P. Bhattacharjee, P. Venkateswaran et al, "Design of Band-pass FIR Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA", IEEE Region 10 Symposium, TENSYMP 2019, IEEE Kolkata Section, Kolkata, June 7 – 9, 2019.

Gmail **Pranabesh Bhattacharjee <pranabesh.dream@gmail.com>**

TENSYMP 2019 notification for paper 148

TENSYMP 2019 <tensymp2019@easychair.org> Tue, Apr 9, 2019 at 1:48 PM
To: Pranabesh Bhattacharjee <pranabesh.dream@gmail.com>

Dear Pranabesh Bhattacharjee,

Congratulations!!!

We are pleased to notify that your paper with number 148 titled “Design of Band-pass Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA ” is PROVISIONALLY ACCEPTED for presentation in IEEE TENSYMP 2019. The presentation schedule will be indicated later on the conference website.

Please do as follows for submitting your final camera-ready manuscript within 30.04.2019:

1. Prepare the final version of your paper incorporating all the reviewer’s suggestions and format the file strictly following the IEEE guidelines with PDFExpress. (The detailed steps for using PDFExpress will be given in the conference website shortly.)
2. The response to the reviewers’ comments of your paper is to be sent as a separate word/pdf file at tpc.tensymp2019@gmail.com
3. The instructions for e-copyright form will be available in the conference website. Authors are requested to submit the e-copyright form as instructed.
3. Please note that this acceptance is provisional. To publish the paper in the conference proceedings, registration of one author per paper is mandatory.
4. To get your paper published in IEEEXplore digital library, the paper must be successfully presented at the conference. “IEEE reserves the right to exclude a submission from distribution after the conference, including exclusion from IEEE Xplore, if the submission does not meet IEEE standards for scope and or quality.”
5. IEEE CROSSCHECK report will be sent to your email.
Please note that the overall similarity index (crosscheck report) excluding bibliography and quotations of the final version of your paper should be below the limit of 30 % and similarity with each of the other papers should not exceed 10%.

Thanks and Regards,
TPC, IEEE TENSYMP 2019

This is an auto-generated email - please do not reply to this mail.

---------- REVIEW 1 ----------
PAPER: 148
TITLE: Design of Band-pass Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA
AUTHORS: Arya Sikder, Supriya Dhabal, Pranabesh Bhattacharjee and Palaniandavar Venkateswaran

Overall evaluation: 1 (weak accept)

---------- Overall evaluation ----------
This paper presents an algorithm for Type-1 Band-Pass filter using an improved Sine Cosine Algorithm (ISCA)and compares the results with SCA and Parks-McClellan algorithm.
1.In lieu of the flow chart , the pseudo code of ISCA would be sufficient.
2. The statement “The algorithm ……..eventually reaching the global optimum” is misleading. There is no non-parametric meta-heuristic algorithm which provides necessary as well as sufficient conditions for the convergence to global optimum.
3. In the concluding paragraph , it is claimed that the ISCA provides faster convergence. But no supporting evidence is provided by the authors. The authors must revise the paper to include supporting evidence for the convergence of ISCA vs SCA [cf. Mirjalili's original paper Fig 13.].
4. Relatively free from syntax and grammatical errors. Well written.

---------- REVIEW 2 ----------
PAPER: 148
TITLE: Design of Band-pass Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA
AUTHORS: Arya Sikder, Supriya Dhabal, Pranabesh Bhattacharjee and Palaniandavar Venkateswaran

Overall evaluation: 1 (weak accept)

---------- Overall evaluation ----------
In this paper, the authors have proposed an improved version of sine-cosine algorithm for design of Band-pass filter and its implementation using FPGA.

The authors should state the motivation behind the improvements proposed.

Some portions of the paper are copied from the following papers:

[1] Supriya Dhabal, Niloy Chakraborty, Adrika Mukherjee, Jayeta Biswas. "Design of higher order FIR low pass filter using Cuckoo search algorithm", 2016 International Conference on Communication and Signal Processing (ICCSP), 2016.

[2] Seyedali Mirjalili. "SCA: A Sine Cosine Algorithm for solving optimization problems", Knowledge-Based Systems, 2016.

---------- REVIEW 3 ----------
PAPER: 148
TITLE: Design of Band-pass Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA
AUTHORS: Arya Sikder, Supriya Dhabal, Pranabesh Bhattacharjee and Palaniandavar Venkateswaran

Overall evaluation: 1 (weak accept)

---------- Overall evaluation ----------
I have the following concerns.

1) Position update method is not explained properly. How can one be assured of the system's convergence?
2) What is the impact of the parameter 'C1' on the system? It has not been described well.
3) The relation between the proposed algorithm and the type-I band pass filter has not been described properly.

---------- REVIEW 4 ----------
PAPER: 148
TITLE: Design of Band-pass Filter using Improved Sine Cosine Algorithm and its Implementation on FPGA
AUTHORS: Arya Sikder, Supriya Dhabal, Pranabesh Bhattacharjee and Palaniandavar Venkateswaran

Overall evaluation: -1 (weak reject)

---------- Overall evaluation ----------
Please note that the following paper has got strong resemblance with the present contribution even though it was not cited –
Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. Knowledge-Based Systems, 165, 374-406.

Without this to me the technical contribution appears inadequate.

REFERENCES

[1] L. Tang and J. Jiang, “Digital Signal Processing: Fundamentals and Applications”, Academic Press, 2007.

[2] S. K. Mitra and Y. Kuo, “Digital Signal Processing: a computer-based approach”, 2nd Edition, Tata Mc. Graw Hill, 2006.

[3] E. C. Ifeachor and B. W. Jervis, “Digital Signal Processing: a practical approach”, 2nd Edition, Pearson, 2007.

[4] J. G. Proakis and D. G. Manolakis, “Digital Signal Processing: Principles, Algorithms, and Edition”, 3rd Edition, PHI publication, 2004.

[5] A. Antoniou, “Digital Filter”, 3rd Edition, Tata Mc. Graw Hill publications, 2001.

[6] D. L. Jones, “FIR Filter Structures”, Version 1.2, 2004.

[7] C. J. Chow, S. Mohanakrishnan, and J. B. Evans, “FPGA implementation of digital filters” , in Proc. of Icspat, Vol. 93 ,p. 1, 1993.

[8] J. F. Wakerly, “Digital Design”, 3rd Edition, Pearson Education, 2016.

[9] U. Farooq, Z. Marrakchi, and H. Mehrez, “FPGA architectures: An overview”, Tree-based Heterogeneous FPGA Architectures, Springer, pp. 7–48, 2012.

[10] A. Ahmed, “FIR Filter Features on FPGA”, Bachelor thesis, Linköping University, Department of Electrical Engineering, Elektroteknik, 2018.

[11] U. Meyer-Baese and U Meyer-Baese, “Digital signal processing with field programmable gate arrays”, Springer, Vol. 2, 2004.

[12] A. W. Ruan, Y. B. Liao, P. Li, and X. J. Li, “An ALU-Based Universal Architecture for FIR Filters”, in Proc. of IEEE, International Conference on Communications, Circuits and Systems, pp. 1070 – 1073, 2009.

[13] X. Jiang and Y. Bao, "FIR filter design based on FPGA", in Proc. of IEEE, International Conference on Computer Application and System Modeling, Vol. 13, p. 621, 2010.

[14] F. Nekoei, Y.S. Kavian, and O. Strobel, “Some Schemes Of Realization Digital FIR Filter On FPGA For Communication Application”, in Proc. of IEEE, 20th

International Crimean Conference on Microwave and Telecommunication Technology, pp. 616 – 619, 2010.

[15] J. Li, M. Zhao, and X. Wang, “Design and Simulation of 60-order Filter Based on FPGA” , in Proc. of IEEE, International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 113 - 115, Oct 2011.

[16] T. Beyrouthy and L. Fesquet, “An event-driven FIR filter: Design and implementation”, in Proc. of IEEE, 22nd IEEE International Symposium on Rapid System Prototyping (RSP), pp. 59 -65, 2011.

[17] L. Jieshan and H. Shizhen, “An Design of the 16-order FIR Digital Filter Based on FPGA”, in Proc. of IEEE, International Conference on Information Science and Engineering (ICISE), pp. 489-492, 2009.

[18] W. Hu, G. Zhang, and W. Li, “Self Programmable Multipurpose Digital Filter Design Based On FPGA”, in Proc. of IEEE, International Conference on Internet Technology and Applications, pp. 1-5, 2011.

[19] N. Li, S.Y. Hou, S.Y. Cui, "Application of Distributed FIR filter based on FPGA in the analyzing of ECG signal", in Proc. of IEEE, International Conference on Intelligent System Design and Engineering Application, Vol. 1, pp. 335-338, 2010.

[20] W. Yunlong, W. Shihu, and J. Rendong, “An Extreme Simple Method for Digital FIR Filter Design”, in Proc. of IEEE, Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Vol. 1, pp. 410-413, 2011.

[21] C.L. Chen, K.Y. Khoo, and A.N. Willson, "A simplified signed powers-of-two conversion for multiplierless adaptive filters", in Proc. of IEEE, Int'l. Symp. Circuits, Sys., Vol. 2, pp. 364-367, May 1996.

[22] S. J. Chapman, “Matlab Programming for Engineers”, Thomson learning, 3rd Edition, 2005.

[23] B. R. Vishwanath and T. S. Theerthesha, “Multiplier Using Canonical Signed Digit Code”, International Journal for Research in Applied Science & Engineering Technology, Vol. 3, No. V, pp. 415-420, May 2015.

[24] J. Kang and J. Gaudiot, "A simple high-speed multiplier design," IEEE Trans. Comput., Vol. 55, No. 10, pp. 1253-1258, Oct. 2006.

[25] A. Efthymiou, W. Suntiamorntut, J. Garside, and L.E.M. Brackenbury, "An asynchronous, iterative implementation of the original Booth multiplication algorithm", in Proc. of IEEE, 10th IEEE Int'l. Symp. Asynchronous Circuits, and Syst., Crete, Greece, pp. 207-215, 2004

[26] G. K. Ma and F. J. Taylor, "Multiplier policies for digital signal processing", IEEEASSP Mag., Vol. 7, No. 1, pp. 6-20, Jan. 1990.

[27] S. M. Kim, J. G. Chung, and K. K. Parhi, "Design of low error CSD fixed-width multiplier", in Proc. of IEEE, Int'l. Symp. Circuits, Syst., Vol. 1, pp. 1-69 - 1-72, May 2002.

[28] B. Parhami, "Computer Arithmetic: Algorithms and Hardware Designs", Oxford Press, London, 1999.

[29] K. K. Parhi, "A Systematic Approach for Design of Digit-serial Signal Processing Architectures", IEEE Transactions on Circuits and Systems, Vol. 38, No. 4, pp. 358-375, 1991.

[30] R. M. Hewlitt and E. S. Swartzlantler, "Canonical signed digit representation for FIR digital filters", IEEE Workshop on SiGNAL PROCESSING SYSTEMS. SiPS 2000, pp. 416-426. IEEE, 2000.

[31] Ed. Rocha, "Implementation trade-offs of digital FIR filters," Military Embedded System, open system publishing, 2007.

[32] C.S. Choi and H. Lee, "A Partial Self-Reconfigurable Adaptive FIR Filter System", IEEE Workshop on signal processing systems, pp. 204-209, 2007.

[33] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters", IEEE Trans. Audio Electroacoust., Vol. AU-21, pp. 506 -526, 1973 .

[34] Y. Li, C. Peng, D. Yu, and X. Zhang, "The implementation methods of high speed FIR filter on FPGA", In: Solid-State and Integrated-Circuit technology, 9th International Conference on. IEEE, pp. 2216–2219, 2008.

[35] J. Selvakumar, B. V. Bhaskar, and S. Narendran. "FPGA based efficient fast FIR algorithm for higher order digital FIR filter", In: Electronic System Design (ISED), pp. 43-47, 2012.

[36] F. F. Daitx, V.S. Rosa, E. Costa, P. Flores, and S. Bampi, "VHDL generation of optimized FIR filters", in Proc. of IEEE, In 2008 2nd International Conference on signals, Circuits and Systems, pp. 1-5, 2008.

[37] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", IEEE transactions on evolutionary computation, Vol. 1, No. 1, pp.67-82. 1997

[38] A. P. Vinod and E. M. Lai, "FIR filter implementation by efficient sharing of horizontal and vertical common sub-expressions", Electronics Letters, Vol. 39, No. 2, pp. 251-253, Jan. 2003.

[39] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters", IEEE transactions on computer-aided design of integrated circuits and systems, Vol. 27, No. 2, pp.217-229. 2008.

[40] R. I. Hartley, "Sub-expression sharing in filters using canonic signed digit multipliers", IEEE Trans. On Circuits and Systems II, Vol. 43, No.10, pp. 677-688, Oct. 1996.

[41] S. Dhabal and P. Venkateswaran, "An improved global-best-driven flower pollination algorithm for optimal design of two-dimensional FIR filter" Soft Computing, Springer, pp.1-18, 2018.

[42] S. Dhabal and P. Venkateswaran, "An Improved Global-Best-Guided Cuckoo Search Algorithm for Multiplierless Design of Two-Dimensional IIR Filters", Circuits, Systems, and Signal Processing, Springer, Vol. 38, No. 2, pp.805-826. 2019.

[43] S. Dhabal and P. Venkateswaran, "Efficient Cosine Modulated Filter Bank Using Multiplierless Masking Filter and Representation of Prototype Filter Coefficients Using CSD", International Journal of Image, Graphics & Signal Processing, Vol.4, No. 10, Sep. 2012.

[44] S. Dhabal and P. Venkateswaran, "A novel accelerated artificial bee colony algorithm for optimal design of two dimensional FIR filter", Multidimensional Systems and Signal Processing, Springer, Vol. 28, No.2, pp.471-493, Apr. 2017.

[45] S. Dhabal and S. Sengupta, "Efficient design of high pass FIR filter using quantum-behaved particle swarm optimization with weighted mean best position", Third

International Conference on Computer, Communication, Control and Information Technology (C3IT), In Proc. of, IEEE, pp. 1-6, 2015.