Implementation of High -speed pipeline FIR filter using icarus

K.Virija*[1], Padakanti srilatha*[2], Kandakatla Shruthi reddy*[3]

[1]Assistant Professor, Department of Electronics and Communication Engineering, Bhoj Reddy Engineering College for Women, Hyderabad.

[2,3]U.G. Scholars, Department of Electronics and Communication Engineering, Bhoj Reddy Engineering College for Women, Hyderabad.

ABSTRACT:

Due to the increased demand of implementation of sophisticated DSP algorithms, low-cost design, I.e., low area and low power cost, are needed to make hand held devices small with good performance. Generally, dedicated multipliers are expensive in terms of chip area and it has also been observed that efficient hardware implementation of a FIR filter requires a relatively large number of multipliers. In the realization of Fir filters, it is evident that almost every structure utilizes Multipliers. which directly impacts the silicon area and power consumption. This report focuses on the development of a novel high-performance FIR Filter that requires an efficient multiplier unit to multiply coefficients with input samples, and hence, significantly reduces the computational complexity. In order to achieve this goal, our work is oriented towards the different multipliers unit which can be computationally efficient for the FIR filters.

KEYWORDS: DSP algorithm, FIR filter, multipliers, power consumption.

I.INTRODUCTION

A rapid pace of development in communication technology makes the communication devices out of date soon after their engineering. To go with such pace communication system, require insertion of latest technology. The upgraded modern device should be able to communicate with legacy of devices as well. The Software Defined Radio (SDR) helps one to add new functionality without much hardware changes, even after technological update. The ideal Software Radio, as defined by Wireless Innovation Forum refers to the complete software control of the entire system. This means that analogue conversion should take place only at antenna, ensuring the support for a wide frequency range. These kinds of software radios will be obviously future proof as the whole radio system will be dependent on programmability, leading to the same hardware behaving differently at different instances. SDR is fast becoming a crucial element of wireless technology. Many of the traditional methods of implementing transmitters and receivers are predicted to be replaced by SDR technology. The advantages offered by it including adaptability, reconfigurability, and multifunctionality encompassing modes of operation, radio frequency bands, air interfaces, and waveforms. A lot of research is going on for improving the architecture and the computational efficiency of SDR

systems. Computationally the most important part of an SDR receiver is the channelizer as it operates at high sampling rate. Channelization in SDR receivers involves the extraction of multiple narrowband channels from a signal using several bandpass filters called channel filters. Low power and high-speed FIR filters are required in the channelize. The key functional units in a digital filter are delay, adder, and multiplier of which the hardware complexity is dominated by multipliers.

Digital filters remove noise and interference from the original signal and are used for modification of various attributes of the signal. The most computationally intensive part of the wideband receiver of a software defined radio (SDR) is the intermediate frequency (IF) processing block. Digital filtering is the main task in IF processing. Digital filters are more accurate, more versatile and highly stable, thereby a preferred technique over its analogue counterpart. The principal objective of this exploration is to present a methodology for an upgraded framework of a FIR digital filter from software level to the hardware level that includes the selection of design method, structure and cost-effective hardware utilization.

A digital filter is a system that performs mathematical operations on a sampled, digitized signal to reduce or enhance certain features of the processed signal. Digital filter scheme consists of a prefilter or anti-aliasing filter to perform filtering of an input signal using a Band pass filter. We use an arithmetic scheme, known as floating- point representation to encode the filter coefficients. This thesis presents a method to implement FIR filters for SDR receivers using various adders and multipliers.

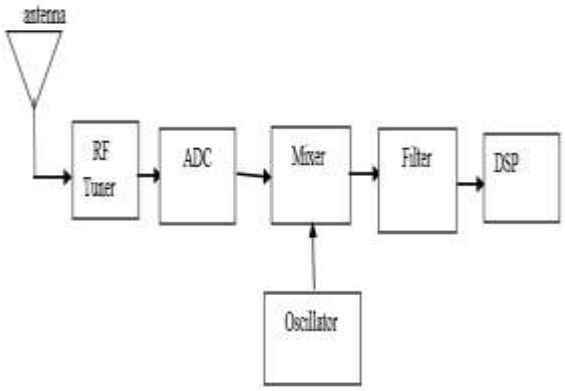

Figure:1. 1 SDR Receiver

An interface is needed between the analog signal and the digital filter, this interface is known as analog-to-digital converter (ADC). After the process of sampling and converting, a digital signal is ready for further processing using an appropriate digital signal processor. The output signal that is digitized is usually changed back into analog form using digital-to-analog converter (DAC).

II.RELATED WORK

In Paper Anubhuti Mittal, Ashutosh Nandi, Disha Yadav presented this paper “Comparative study of 16-order FIR filter design using different multiplication Techniques” to study a comparation between various adders and multipliers such as to optimize filter area, delay and power. Different multiplication techniques such as Vedic multiplier, add and shift method and Wallace tree (WT) multiplier are used for the multiplication of filter coefficient with

filter input. Various adders such as ripple carry adder, Kogi Stone adder, Brent Kung adder, Ladner Fischer adder and Han Carlson adder are analysed for optimum performance. To reduce the complexity of filter, coefficients were represented in canonical signed digit representation as it is more efficient than traditional binary representation. They designed finite impulse-response (FIR) filter in MATLAB using equal ripple method and the same filter was synthesized on Xilinx Spartan 3E XC3S500E target field-programmable gate array device using Very High-Speed Integrated Circuit Hardware Description Language (VHDL) subsequently the total on-chip power was calculated in Vivado2014.4. The comparison of simulation results of all the filters showed that FIR filter with WT multiplier is the best optimized filter. The FIR filter designed is of order 16, with density factor 20 using equal ripple method. They compared various multipliers and adder combination to implement filter in Xilinx, which was designed in MATLAB and concluded that FIR filter designed with WT multiplier has performance advantages among all the proposed filters. The combination proposed and compared were FIR with Vedic multiplier (VM) and ripple carry adder (RCA), FIR with VM and Brett Kung (BK), FIR with RCA and Barrel Shifter (BS), FIR with Kogi Stone (KS) and BS, FIR with BK and BS, FIR with Han Carlson (HC) and BS, FIR with Ladner Fischer (LF) and BS, FIR with WT and RCA.

The Aneela Pathan, Tayab D Memon, SharmeenKeerio, Imtiaz Hussain Kalwar have presented comparative analysis of Booth and Wallace Tree multiplier architectures using Altera small commercial FPGA device, in their paper “FPGA Based performance analysis of multiplier policies for FIR filter”. Comparison is done with respect to resources consumed and maximum frequency achieved for different multiplier bit width. In this work, two algorithms are implemented on small commercial FPGA devices provided by Altera. They have chosen two architectures i.e., cyclone and Stratix, area-performance was evaluated for these two algorithms. For varying bit widths i.e., 6,8,10 sizes of multiplicand and multipliers area performance were compared. They observed that maximum Performance (FMAX) of Cyclone and Stratix is consistent for Wallace tree multiplier, but when it is compared with Booth multiplier FMAX is decreasing by increasing the number of bits (i.e., multiplier bit-width) in both families. They concluded that performance of Wallace tree is much better than booth.

Evangelos Kyritsis, KiamalPekmestzi in their paper “Hardware Efficient Fast FIR Filter Based on Karatsuba Algorithm” proposed an efficient implementation of a programmable Finite Impulse Response (FIR) filter based on the use of the Karatsuba Multiplication Algorithm (KMA). Here an FIR filter circuit, a parallel, Modified Booth (MB) preen coded, Carry-Save (CS) Wallace tree multiplier have been used as building block. They have compared the proposed architecture with that of conventional architecture for speed, area, power.

The Karatsuba algorithm is applied to the FIR filter equation, a parallel architecture is obtained giving speed and area savings. In this paper, they proposed an architecture based on a modified formulation of KMA. In the proposed implementation carry-save (CS) arithmetic is used in order to decrease the critical path and to speed-up the calculations. For comparison between conventional and Karatsuba FIR Filters, both of them were designed for taps =16,32 and bits 2N. They used modalism for functional simulation, then the designs were implemented on Faraday 90nm (FSD0A_A_GENERIC_CORE_TT1V25C) technology using Synopsys tools. The synthesis has been made with Design Compiler. The mean power consumption has been obtained using Prime Power. They concluded that the proposed Karatsuba design showed an improved performance, a smaller circuit area and lower power consumption, compared with the Conventional transposed FIR filter.

Ila Sharma, Anil Kumar, Girish Kumar Singh, Heung-No Lee proposed a hybrid method-based design of multiplier less two-channel filter bank has been proposed with a given stopband attenuation (As) and roll-off factor. Windowing technique has been used for filter design. The implemented filter is synthesized using target field programmable gate arrays XC3S500E-4-FG320 on Xilinx Spartan 3E starter board. The performances of designed prototype filter are compared with the earlier published works in terms of reconstruction error, amplitude distortion, slices, flip-flops, four-input lookup tables and adders. The synthesis results demonstrate that the significant reduction in hardware is achieved in term of adder gain. For filter order, N = 32, and word length 12, the adder gain achieved in canonical sign digit (CSD).and factorized canonical sign digit (FCSD)is 41.77 and 43.07%, respectively, while for N = 30, it is 35.44% in CSD and 36.70% in FCSD. In paper Arunadevi Jawahar, P. Pushpa Latha in their paper “Implementation of high-order FIR digital filtering for software defined radio receivers” proposed an FIR filter for SDR receiver application. They utilized a transpose form structure. Wallace tree multiplier and kogge stone adder were used to implement the filter. A comparison was made between existing and proposed work. They achieved a minimum delay.

III.PROPOSED SYSTEM

Digital Filter design Process can be summarized into the following steps. The design of a digital filter involves following five steps.

• Filter specification: This may include stating the type of filter, for example lowpass filter, the desired amplitude and/or phase responses and the tolerances, the sampling frequency, the word length of the input data.

• Filter coefficient calculation: The coefficient of a transfer function H(z) is determined in which some which will satisfy the given specification. The choice of coefficient calculation method will be influenced by several factors. The most important of which are the critical requirements i.e., specification. The window, optimal and frequency sampling method are the most commonly used.

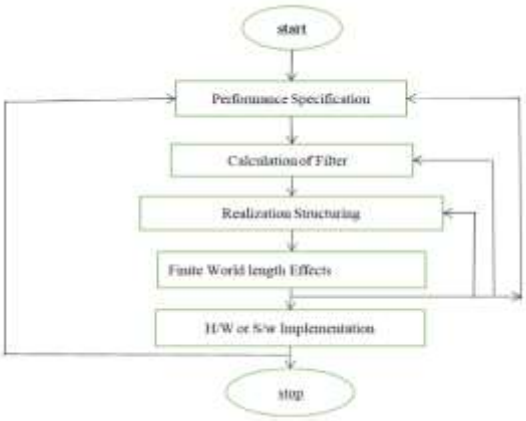

Figure 3.1 Summary of design stage for digital filter

• Realization: This involves converting the transfer function into a suitable filter network or structure.

• Analysis of finite word length effects: The effect of quantizing the filter coefficients and input data as well as the effect of carrying out the filtering Start Performance specification Calculation of filter coefficients Realization structuring Finite world length effects analysis H/W or S/W implementation Stop operation using fixed word length on the filter performance is analysed here.

• Implementation: This involves producing the software code and/or hardware and performing the actual filtering.

IV.RESULTS

Figure 4.1 Simulation

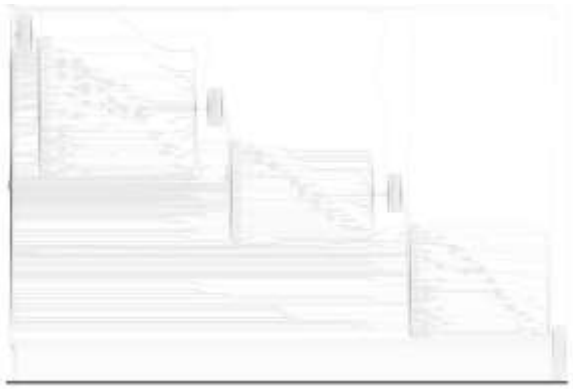

Figure 4.2 Synthesis (Block Diagram)

4.1 Statistics

Finite Impulse Response Filter:

4.1.1 Before reintegrating:

Number of wires: 357

Number of wire bits: 632

Number of public wires: 25

Number of public wire bits: 300

Number of memories: 0

Number of memory bits: 0

Number of processes: 0

Number of cells: 444

$_ANDNOT_ 52

$_AND_ 16

$_AOI3_ 26

$_DFF_P_ 64

$_NAND_ 2

$_NOR_ 44

$_NOT_ 13

$_OAI3_ 23

$_ORNOT_ 46

$_OR_ 11

$_XNOR_ 85

$_XOR_ 62

4.1.2 After integrating:

Number of wires: 248

Number of wire bits: 523

Number of public wires: 25

Number of public wire bits: 300

Number of memories: 0

Number of memory bits: 0

Number of processes: 0

Number of cells: 335

sky130_fd_sc_hd__a211oi_1 1

sky130_fd_sc_hd__a21bo_1 1

sky130_fd_sc_hd__a21boi_0 1

sky130_fd_sc_hd__a21o_1 1

sky130_fd_sc_hd__a21oi_1 15

sky130_fd_sc_hd__a311o_1 1

sky130_fd_sc_hd__a31oi_1 5

sky130_fd_sc_hd__and2_0 5

sky130_fd_sc_hd__and3_1 4

sky130_fd_sc_hd__buf_1 1

sky130_fd_sc_hd__xor3_1 1

Chip area for module (fir): 3135.507200

V.CONCLUSION & FUTURE WORK

The FIR filters are widely used in digital signal processing and can be implemented using programmable digital processors. But in the realization of large order filters the speed, cost, and flexibility are affected because of complex computations. So, the implementation of FIR filters on FPGAs is the need of the day because FPGAs can give enhanced speed. This is due to the fact that the hardware implementation of a lot of multipliers can be done on FPGA which are limited in case of programmable digital processors. In this thesis, a thirty-one-order band-pass FIR filter is implemented in Virtex6 XC6VLX760 FPGA. The Direct form and Transpose structure of these filters are implemented. In both the forms of structures, N Adder and N+1 multipliers are used to realize the N order Band pass filter. The designed filters can work for real time Receivers or processing of any digital signal. Here in this work number representation was discussed. Efficient floating point arithmetic unit with high precision was developed using Verilog code on Xilinx ISE 14.7 and comparison were made. In this thesis Five different multipliers Array multiplier, Wallace tree multiplier, Vedic multiplier, Modified Booth multiplier and constant multiplier were implemented in floating point format.

This presented work has successfully developed an efficient data path unit for an FIR filter processor using different types of multiplier unit in floating point

representation and implemented it for 31-order using Direct form and Transpose form structures in Xilinx Virtex-6 FPGA. The analysis shows that using a Vedic multiplier in the filter structure will be efficient way to reduce area compared to other multipliers.

Optimizing a hardware structure is a trade off between various design constraints, such as performance, resource utilization, power consumption, and precision. This depends of course on the application too. Here in this thesis two structures (direct form and transposed form) were used and a comparison was drawn between the window techniques for the receiver. There are other structures like poly phase, lattice structures which can also be studied in this context. Another future work could be a re-configurable filter implementation which are high in demand.

VI.REFERENCES

[1] Tan, Li “Digital Signal Processing: Fundamentals and Applications” Edition 2007.

[2] Mitra, S. K., “Digital Signal Processing” 3rd Edition, Tata Mc. Graw Hill Publications.

[3] Ifeachor, E.C., Jervis, B.W., “Digital Signal Processing”,2nd Edition, Low Price Edition 2007.

[4] Akanksha Kant, Shobha Sharma, “APPLICATIONS OF VEDIC MULTIPLIERDESIGNS” IEEE 2015.

[5] Joseph Cavanagh,” Computer Arithmetic and Verilog HDL Fundamentals”2010 by Taylor & Francis Group, LLC.

[6] Mi Lu,” Arithmetic and Logic in Computer Systems”2004 by John Wiley & Sons, Inc.

[7] Proakis, J.G., Manolakis, D.G., “Digital Signal Processing” 3rd Edition, PHI publication 2004.

[8] Anubhuti Mittal, Ashutosh Nandi, Disha Yadav presented this paper “Comparative study of 16-order FIR filter design using different multiplication Techniques” The Institution of Engineering and Technology 2017

[9] Aneela Pathan, Tayab D Memon, SharmeenKeerio, Imtiaz Hussain Kalwar, “FPGA Based performance analysis of multiplier policies for FIR filter” The Institution of Engineering and Technology 2016

[10] Evangelos Kyritsis, KiamalPekmestzi in their paper “Hardware Efficient Fast FIR Filter Based on Karatsuba Algorithm”2016 IEEE

[11] Ila Sharma, Anil Kumar, Girish Kumar Singh, Heung-No Lee,” Design of multiplier less prototype filter for two-channel filter bank using hybrid method in FCSD space”. The Institution of Engineering and Technology 2016

[12] Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal, “High Speed Signed Multiplier for Digital Signal Processing Applications” source: 978-1-4673-1318-6/12/$31.00 ©2012 IEEE.

[13] David Villeger, Vojin G Oklobdzija, “Evaluation of Booth Encoding Techniques

for Parallel Multiplier Implementation" source: Volume 29, Issue 23, 11 November 1993, ISSN 0013-5194.

[14] R. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits & Syst. II, vol. 43, Oct. 1996.

[15] S. D. Pezaris, "A 40-ns 17-Bit by 17-Bit Array Multiplier", IEEE Trans. On Computers, pp. 442-447, Abr. 1971.

[16] Ron S. Waters, Earl E. Swartz lander, "A Reduced Complexity Wallace Multiplier Reduction," IEEE TRANSACTIONS ON COMPUTERS, VOL. 59, NO. 8,pp. 1134 – 1137, AUGUST 2010