



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 5, May 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118

Investigation of Suitable DSP Architecture for Efficient FPGA Implementation of FIR Filter

Shubham Suryavanshi¹, Satyarth Tiwari²

P.G. Student, Department of Electronics & Communication Engineering, Bhabha University, Bhopal,

Madhya Pradesh, India¹

Assistant Professor, Department of Electronics & Communication Engineering, Bhabha University, Bhopal,

Madhya Pradesh, India²

Abstract: In this paper, we have investigated pipeline and parallel processing architectures of finite impulse response (FIR) filter for efficient field programmable gate array (FPGA) implementation. Our simulation results show that parallel processing architecture is more efficient as compared to pipeline architecture. Further, it is shown that fast FIR architecture is most suitable as compared to conventional parallel processing architecture due to its hardware complexity reduction. Fast FIR filter architecture shows 25% improvement in area as compared to conventional parallel processing architecture for identical performance.

KEYWORDS: Field Programmable Gate Array (FPGA); MATLAB; System Generator; Finite impulse response (FIR) filter

I. INTRODUCTION

Digital Signal Processing (DSP) is used in many applications such as video processing, speech processing, image processing, biomedical signal processing and wireless communication. The implementation of advanced digital signal processing applications is challenging due to constraints on sampling rate, area and power consumption. Due to ever growing demand of portable applications such as Smart Phones, Tablets, PDA etc. implementation of Digital Signal Processing applications must be area and power efficient. Digital signal processing systems are realized using hardware as well as software approach. Hardware approach uses Digital Signal processors (DSPs) while software approaches make use of software's like MATLAB, C, C++ etc. These approaches are very generic and not suitable to implement portable applications. Field Programmable Gate Array (FPGA) is the best candidate to overcome the limitations of generic hardware and software approaches. In this paper an attempt is made to find suitable architecture for efficient implementation of FIR filter on FPGA. The goal of the paper is to maximize performance by increasing sampling rate, throughput and less resource for implementation of DSP system. Reconfigurable hardware (FPGA) offers a trade off between performance, cost, power, flexibility and design effort. Non-recurring engineering (NRE) [1]. Intensive digital signal processing algorithms to improve overall performance. This paper describes the implementation of various architectures of finite impulse response (FIR) filter on field programmable gate array (FPGA). Pipeline and parallel processing architectures are implemented and analysed to evaluate performance in terms of high throughput, resource utilization, maximum frequency of operation which is bounded by critical path. The rest of the paper is arranged as follows. Section II gives the background and evolution of Field Programmable Gate Arrays. The various FIR filter architecture and its implementation is described in Section III. Results of the implementation are described in Section IV. Finally conclusions of the work are given in Section V.

II. RELATED WORK

Field Programmable Gate Array (FPGAs) are parallel structures containing a uniform array of configurable logic blocks (CLBs), memory, DSP slices and some other elements. The main advantage to digital signal processing within an FPGA is the ability to adapt the implementation to match system requirements. This means in a multiple-channel or high-speed system, you can take advantage of the parallelism within the device to maximize performance [2]. Very high-speed I/O further reduces cost by maximizing data flow from input through the processing chain to final output.

At high throughput or increased data rates the Digital signal processors may not prove to be efficient to sample, process and output the information without any loss. This is due to the many shared resources, buses and even the core within

the processor. The FPGA, however, can dedicate resources to each of these functions. Digital signal processors are instruction based, not clock based. Typically, three to four instructions are required for any mathematical operation on a single sample. In contrast, the FPGA is clock based, so every clock cycle has the potential ability to perform a mathematical operation on the incoming data stream.

Optimized pipelined architecture for finite impulse response filter implemented on FPGA is presented in [3], [4]. Results show better performance. References [5]–[7] presents parallel structure implementation of filter on FPGA for area optimized results. Parallel structure based on Fast FIR architecture (FFA) and present proposed method for implementation. Comparative analysis of parallel filter and FFA filter in terms of number of multiplier and adder required for implementation is done in [8]. Above reference shows direct benefits of FPGA for digital signal processing in terms of functional specialization, platform reconfigurability and parallelism.

III. METHODOLOGY

We implemented and compared analysis of pipelined architecture, parallel processing and fast FIR architecture for finite impulse response filter. Architectures are analysed for high throughput, maximum frequency and resources utilization.

In high-speed signal processing applications, the direct form filter figure 1 is not desirable in that its critical path, corresponding to the maximum computation delay in generating an output, includes many computational elements contributing to the delay. Furthermore, this computation delay increases with the number of taps in the direct form filter. Pipelined architecture figure 2 with additional delay elements have been inserted in the filters to reduce both the computation delay and input capacitive load, but leads to a penalty in terms of an increased latency. Latency means the difference in the availability of the first output data in the pipelined system and the sequential system. Adding delay element after adder just enables D flip-flop of logic block in FPGA. So, this will not increase area for implementation. Parallel processing and pipelining techniques are duals each other: if a computation can be pipelined, it can also be processed in parallel. Parallel processing figure 3 increases the sampling rate by replicating hardware so that several inputs can be processed in parallel and several outputs can be produced at the same time. The effective sampling speed is increased by the level of parallelism. Replication of hardware leads to more resource utilization but with reduced latency and improved critical path over direct form FIR filter. For 3-level parallel filter $3N$ number of multiplication and $3(N-1)$ number of addition required.

Fast FIR algorithm (FFA) figure 4 used to produce reduced complexity parallel filtering structures. Consider for 3-level parallel filter $2N$ number of multiplication and $2N+4$ number of addition required. Referring previous structure it is clear that the reduced complexity structure provides a saving of approximately 33%.

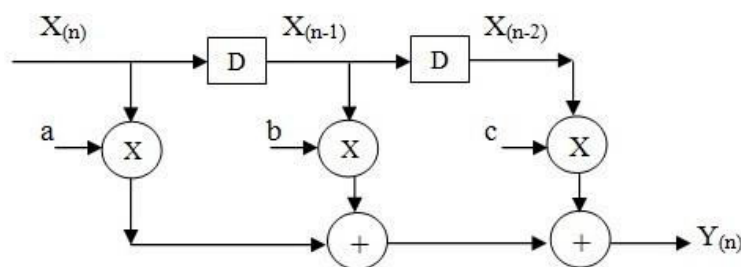


Fig1. Direct Form

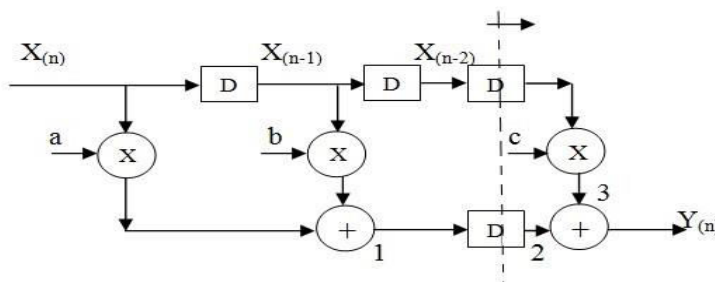


Fig2. Pipelined Structure

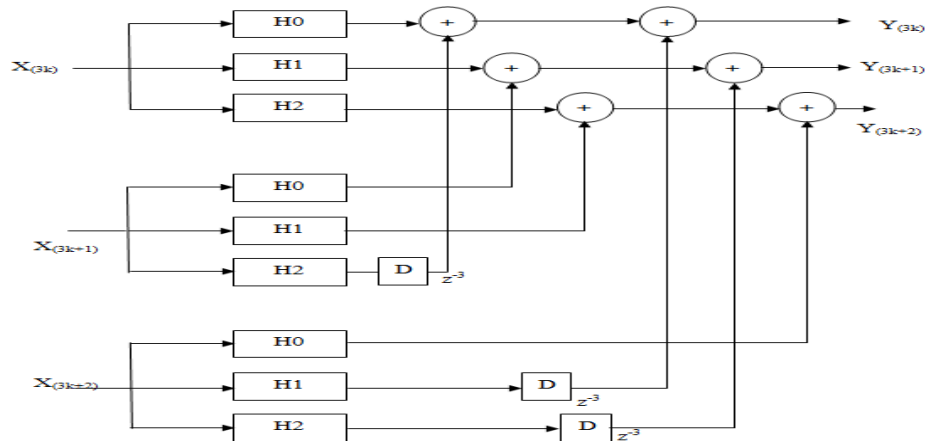


Fig 3.Parllel Structure

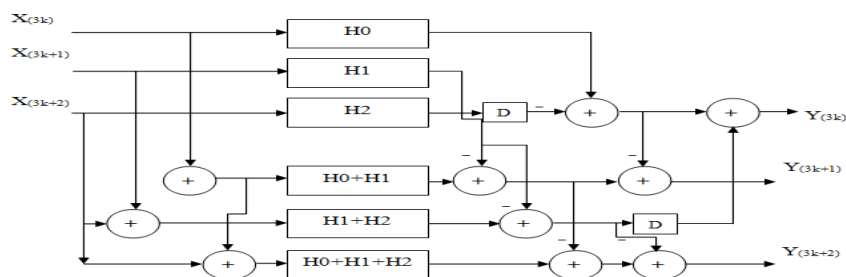


Fig 4. FFA Structure

A. Implementation tools

Filter model designed using MATLAB Simulink graphical tool and then using system generator Hardware descriptive language HDL code is generated for synthesis and simulation. Same models are created using Xilinx blocks in MATLAB as shown in above figures. Figure 5 shows design flow for FPGA implementation and analysis [9]. Xilinx Synthesis Tool XST used for synthesize design [10]. Filter Design and Analysis (FDA) Tool used to calculate filter coefficient, which are required for Matlab model. Reference [11] shows implementation using Xilinx system generator.

B. Filter specifications

We design model for 21-tap low pass FIR filter, with three elements delay block, adder block and multiplier. Here we assume latency of two for multiplier block. Detailed specification shown in Table I and filter coefficient given in Table II below: Table 1 Filter Specification

Parameters	Values
Sampling Frequency F_s	16KHz
Pass Band Frequency F_p	4KHz
Stop Band Frequency F_s	6KHz
Pass Band Ripple	1db
Stop Band Attenuation	80db

We considered 3-stage pipelined architecture for implementation. For parallel processing level-3 of parallelism is considered. With this assumption we proceed to result discussion.

Table II. Filter Coefficients

Filter coefficients	
$h(0), h(20) =$	0.0028574928916898844
$h(1), h(19) =$	-0.0080821234910925487
$h(2), h(18) =$	-0.00078127842533529514
$h(3), h(17) =$	0.018509955656029461
$h(4), h(16) =$	-0.020478899701979331
$h(5), h(15) =$	-0.017008709356672432
$h(6), h(14) =$	0.061898368116084498
$h(7), h(13) =$	-0.03457000192208106
$h(8), h(12) =$	-0.10644324942849505
$h(9), h(11) =$	0.28928019159138546
$h(10) =$	0.62589513309607037

III. RESULTS AND DISCUSSION

Result analysis for 21-tap low pass equiripple FIR filter, shown in Table III and figures below gives comparison between direct form, pipelined architecture and parallel structure. Resource utilization is better in pipelined structure as compared to parallel architecture. This due to introduction of delay element after adder blocks which enables D flip-flop in adder block.

Critical path which is of the important performance parameter of FIR filter is low for FFA architecture with reduced hardware complexity.

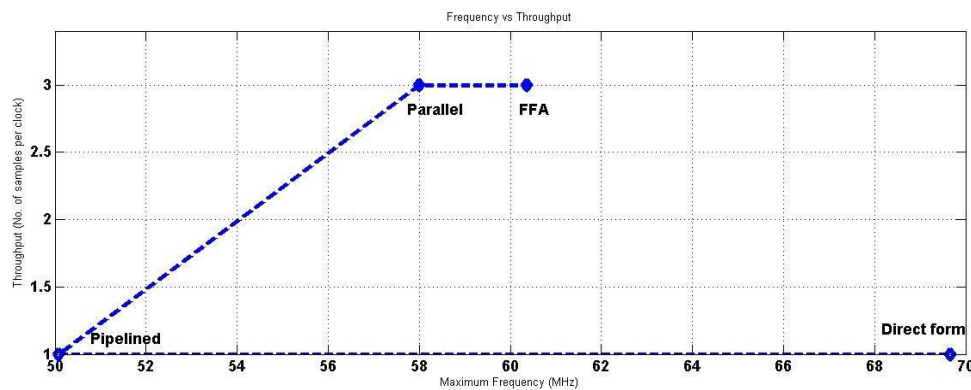


Fig 5. Frequency Vs. Throughput

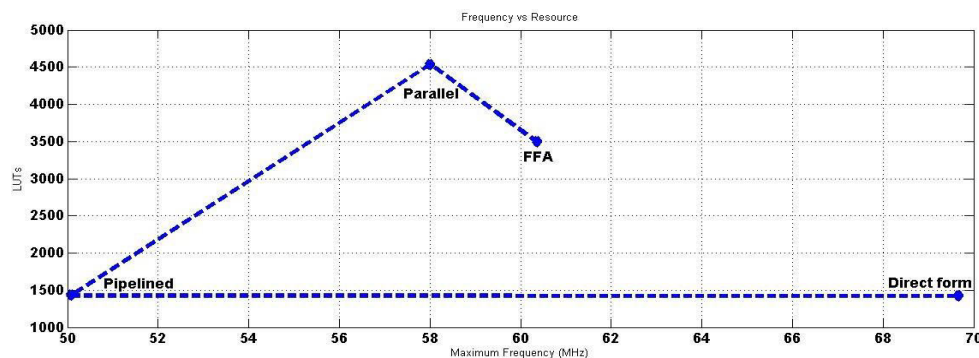


Fig 6. Frequency Vs. Resources

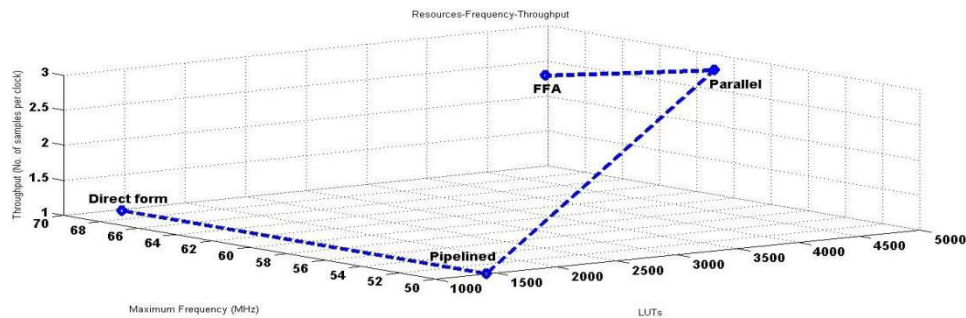


Fig. 7. Resources-Frequency-Throughput

V. CONCLUSION

Fast FIR architecture is well suitable for efficient FPGA implementation. Fast FIR filter architecture shows 25% improvement in area as compared to conventional parallel processing architecture and also shows 20% improvement in frequency of operation over pipelined architecture. FPGA implementation of Fast FIR filter for digital signal processing applications results in increased sample rate, reduced area and improved speed over conventional parallel and pipelined architectures.

Table III. Results.

	Parameters	Direct form	Pipelined(3-stage)	Parallel(level-3)	FFA(level-3)
Resources	No. of Flip-Flops	1245	1297	3765	2728
	No. of LUTs	1429	1433	4537	3497
Timing & Clock	Min. Period (nsec)	14.352	19.963	17.238	16.567
	Max. Frequency (MHz)	69.677	50.093	58.011	60.361
	Latency(cycles)	2	4	4	3
	Throughput(No. of samples/clock)	1 sample	1 sample	3 sample	3 sample
Power	Power (mW)	29	29	41	39

REFERENCES

- [1] Keshab K Parhi. VLSI digital signal processing systems: design and implementation. John Wiley & Sons, 2007.
- [2] Mahesh Kadam and Kishor Sawarkar. An overview of reconfigurable hardware for efficient implementation of dsp algorithms. IOSR Journal of Engineering, 4:34–43, 2014.
- [3] Shraddha S Borkar and Awani S Khobragade. Optimization of fir digital filter using low power mac, 2012.
- [4] Martin Kumm and Peter Zipf. High speed low complexity fpga-based fir filters using pipelined adder graphs. In Field-Programmable Technology (FPT), 2011 International Conference on, pages 1–4. IEEE, 2011.
- [5] Yu-Chi Tsao and Ken Choi. Area-efficient parallel fir digital filter structures for symmetric convolutions based on fast fir algorithm. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 20(2):366–371, 2012.
- [6] Chao Cheng and Keshab K Parhi. Further complexity reduction of parallel fir filters. In Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, pages 1835–1838. IEEE, 2005.
- [7] Chao Cheng and Keshab K Parhi. Low-cost parallel fir filter structures with 2-stage parallelism. Circuits and Systems I: Regular Papers, IEEE Transactions on, 54(2):280–290, 2007.
- [8] PN Krishnapriya and Arathy Iyer. Power and area efficient implementation for parallel fir filters using ffas and da.
- [9] Rashmi Nanda, Chia-Hsiang Yang, and Dejan Markovic. Dsp architecture optimization in matlab/simulink environment. In VLSI Circuits, 2008 IEEE Symposium on, pages 192–193. IEEE, 2008.
- [10] www.xilinx.com/support/documentation/sw-manuals/xilinx11/ise-c-fpga-design-flow-overview.htm.
- [11] V Elamran, Kalagarla Abhiram, Narreddi Bhanu Prakash Reddy, and Narreddi Bhanu. Fpga implementation of audio enhancement using xilinx system generator. Journal of Applied Sciences, 14(17):1972–1977, 2014.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details