Optimized Digital Filter through Pipeline Processing and Look-Ahead Operation

Ajeet Kumar Srivastava, Vishal Awasthi, Anand Kumar Gupta and Parul Awasthi
Department of Electronics & Communication Engineering, C S J M University, India
E-mail: ajeetkumar9@rediffmail.com

Abstract

The performance of a digital filter depends on the hardware platform and the computational structure. Pipeline processing is used to achieve high throughput and speed, but it can be challenging in feedback systems like IIR filters. FIR filters have advantages such as linear phase and stability but require higher orders and additional hardware demands, arithmetic operations, area usage, and power consumption. we proposes a design and implementation approach for non-pipelined and pipelined FIR filters that optimize performance while preserving the dynamics of the filters. The proposed approach is well-suited for practical applications where real-time implementation and low power consumption are critical factors. The proposed structure requires fewer adders than the traditional direct form filter and is more power-efficient than direct and transposed filters. The simulation results shows that the proposed structure consumed 19.62% less power in comparison to traditional direct form filter design. This structure has the potential to be an appealing alternative due to its ability to achieve both the potential for trade-offs and complexity reduction of parallel architecture.

Keywords: ***Finite Impulse Response (FIR), Pipelining and parallel Architecture, RestructralDesign, Optimization Techniques.***

1. Introduction

Digital filters are an important component of digital signal processing systems, used to extract meaningful information from raw input data. The performance of a digital filter depends not only on the capabilities of the hardware platform used, but also on the computational structure of the code. In order to achieve high throughput and speed, pipeline processing can be used to break down operations into smaller, quicker operations, with registers placed between levels to reduce critical path delay. However, pipeline processing can be challenging in feedback systems such as IIR filters, as the introduction of registers in a feedback loop can alter the loop delay and modify the transfer function. To overcome this challenge, computations must be redeveloped into a look-ahead filter form. FIR filters, on the other hand, offer advantages such as linear phase, stability, fewer finite precision errors, and efficient implementation, but require higher orders and result in additional hardware demands, arithmetic operations, area usage, and power consumption. As a result, minimizing these parameters is a key goal in digital filter design. In this paper, we present a design and implementation approach for non-pipelined and pipelined IIR and FIR filters that optimize performance while preserving the dynamics of the filters. Finite Impulse Response (FIR) filters are a type of digital filter widely used in digital signal processing applications. Unlike Infinite Impulse Response (IIR) filters, FIR filters have a finite impulse response and provide a linear phase response, making them ideal for applications that require precise phase information in the output signal.

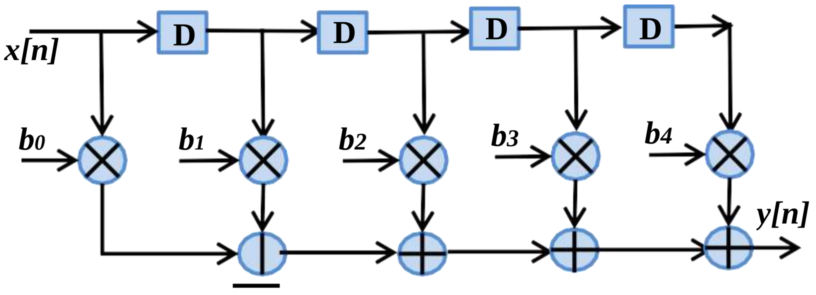


Figure 1 Structure of 5 taps FIR filter

The effectiveness of the proposed approach is demonstrated through experimental results, which show that the proposed method achieves significant performance improvement over traditional methods while maintaining a low computational cost. This makes the proposed approach well-suited for practical applications where real-time implementation and low power consumption are critical factors.

The remaining contains are organized as follows. In section 2, the related work is presented. Section 3, outlined the methodology to design the traditional FIR structure. The hybrid form of low pass FIR filter design is discussed in section 4. Then, in Section 5, results are presented where the proposed structure is compared with the traditional hybrid form FIR filter. Finally, the conclusions are given in section 6.

2. Related work

In the field of digital signal processing, high-performance FIR filters designed with various techniques have drawn attention. To increase the effectiveness and speed of FIR filters, numerous research work have been carried out in this area. Utilising hybrid FIR filter types, which combine the benefits of symmetric and asymmetric filters, is an alternative approach. To develop and implement a high performance FIR filter, various techniques might be used. Several of the popular techniques are discussed. Compared to symmetric filters, these filters offer a superior frequency response and need fewer coefficients [1]. Chao-Huang et al, [2] has suggested a novel design methodology for a cost-efficient FIR-Processor that can be easily implemented on SoPC systems. This methodology offers several advantages over conventional methods, such as flexibility in changing coefficient values and easy extension to an n-tap FIR filter. Additionally, it can work with high input signal frequencies, and many identical units can be integrated into one chip for parallel processing. Because of its effectiveness and simplicity, the Kaiser window and direct-form structure are suggested, and the optimised filter implementation uses 42% fewer hardware resources than a typical implementation. Overview of the FIR filter's power-efficient structure is

- Pipeline processing: This involves breaking up the FIR filter into smaller, sequential stages, each of which can be processed in parallel using dedicated hardware. Pipeline processing reduces the critical path delay and improves the filter's throughput.
- Look-Ahead Parallel FIR filter Structure: This involves implementing multiple FIR filters in parallel, each with a different set of coefficients. The input signal is split into multiple streams, and each stream is processed by a separate FIR filter. The output signals from each filter are combined to produce the final output signal.
- Time-multiplexed FIR filters: This involves implementing multiple FIR filters using a time-multiplexed approach. The input signal is sampled and held, and each sample is processed by a different FIR filter. The output signals from each filter are time-multiplexed to produce the final output signal.
- Distributed arithmetic: This is a method for implementing FIR filters using shift registers and lookup tables. The filter coefficients are represented in binary form and stored in lookup tables, and the filter is implemented using a series of shift and add operations. This method is well-suited for pipelining and parallel processing.

The choice of method depends on the specific requirements of the application, such as the desired filter performance, hardware resources, and power consumption. Each method has its advantages and disadvantages, and the designer must select the most appropriate method based on the specific application requirements. By using these methods, it is possible to implement high performance FIR Filters with look ahead Approach in an efficient and resource-friendly manner, making it ideal for digital signal processing applications.

3. Proposed an Efficient FIR Filter Structure

FIR filters are frequently used. The transfer function of an Nth-order FIR filter can be expressed as

$$\mathrm{H(z)} = \sum_{i=0}^{N} h_i z^{-i} \qquad \textbf{(1)}$$

The transfer function for a FIR filter given in (1) can be divided into subsections of M taps and recast, assuming that N + 1 is an integer multiple 2 of M, as shown in

$$\mathrm{H(z)} = \sum_{k=0}^{\frac{(N+1)}{M}-1} \left[\sum_{i=0}^{M-1} h_{Mk+\mathrm{i}}\, z^{-\mathrm{i}} \right] z^{-Mk} \qquad \textbf{(2)}$$

- Pipelining techniques: There are various pipelining techniques that can be used to optimize instruction execution time and increase program execution speed. These include instruction-level pipelining, data-level pipelining, and task-level pipelining. Each of these techniques involves breaking down a task or program into smaller subtasks that can be executed in parallel or in overlapping phases, thus reducing the overall execution time.
- Hardware acceleration: Hardware acceleration is a technique that involves using specialized hardware to accelerate the execution of certain tasks, such as filter coefficient multiplication. This can be achieved using application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), or digital signal processors (DSPs), among others.
- Parallel processing: Parallel processing involves breaking down a task or program into smaller subtasks that can be executed in parallel on multiple processing units or cores. This can significantly reduce the overall execution time of a program or task, and is often used in conjunction with pipelining and hardware acceleration techniques.
- Code optimization: Code optimization involves writing code that is optimized for performance and efficiency, by minimizing the number of instructions required to execute a task, minimizing the use of memory and other system resources, and using efficient algorithms and data structures. Code optimization can help to reduce the overall execution time of a program or task, and is often used in conjunction with other optimization techniques such as pipelining and hardware acceleration.In many DSP systems,

The final structure must then incorporate a matrix multiplication as a result of extra retiming [15]. The filter structure depicted in Figure 2 has equivalent numbers of multipliers, adders, and delay elements to any of the structures as long as N+1 is a multiple of M. However, similar to the traditional hybrid form FIR filters, the critical path is reduced compared to the direct form filter, and the number of wide delay elements is lower than that of the transposed form FIR filter. Additionally, instead of having multiple smaller MCM blocks as in the traditional hybrid form FIR filter, there is now one matrix MCM block. This not only enables the use of redundancies within a row or column of coefficients, but also between entire rows or columns [16].

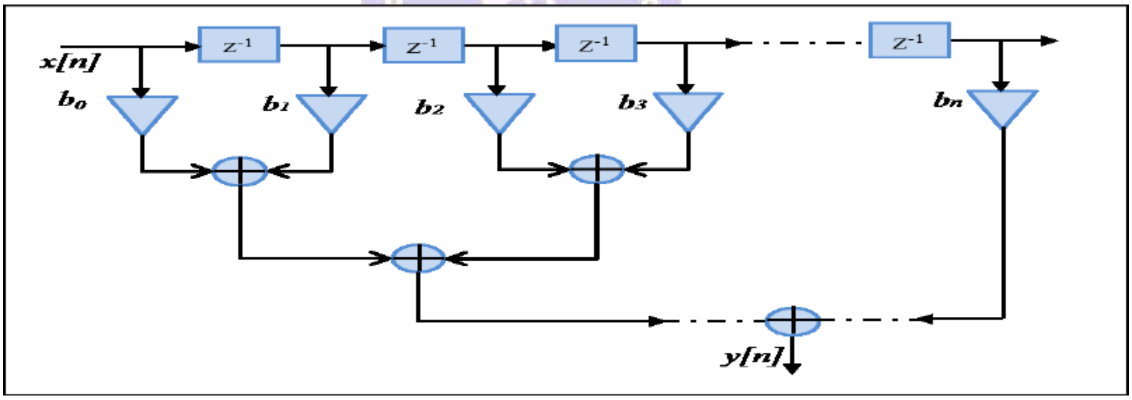
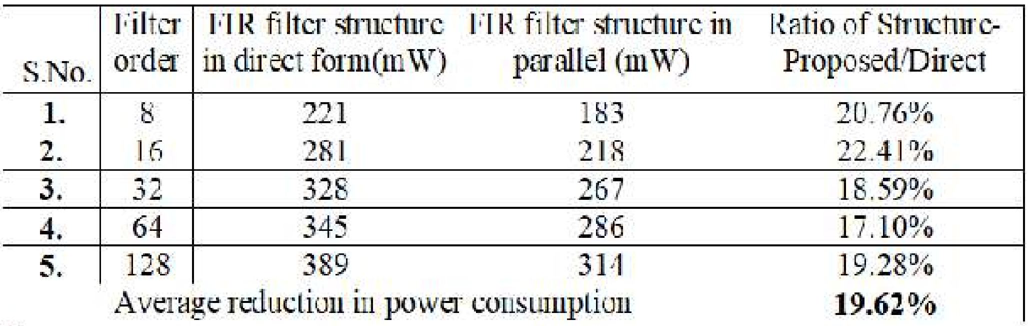


Figure 2 FIR filter structure in Parallel from

4. Results and Discussion

A FPGA in the Virtex 4 family is used to synthesise the updated structure that has thus been built, as shown in Table 1, by combining the suggested methods based filter operations. The results of a proposed low-pass FIR filter design based on parallel structuer is compared with other existing designs. The simulation waveforms of broadcast and non-broadcast low-pass filters are expected to be the same, and the same is expected for cutset-retime and feed forward for third-order broadcast and non-broadcast low-pass filters for the Kaiser window when k=1 or k=2. The proposed third-order low-pass FIR filter design shows improved speed and area compared to existing designs. Specifically, the proposed design uses feedforward techniques for the broadcast low pass filters, using a Kaiser window when k=1 or k=2. The synthesis results and comparison with existing designs are shown in Table 1. The proposed design has shown that a lower power consumption and lower delay compared to existing designs, indicating that it is faster and more power-efficient. Power is measured in mw.

Table 2 Comparative chart of power consumption



S.No.	Filter order	FIR filter structure in direct form(mW)	FIR filter structure in parallel (mW)	Ratio of Structure-Proposed/Direct
1.	8	221	183	20.76%
2.	16	281	218	22.41%
3.	32	328	267	18.59%
4.	64	345	286	17.10%
5.	128	389	314	19.28%
	Average reduction in power consumption			**19.62%**

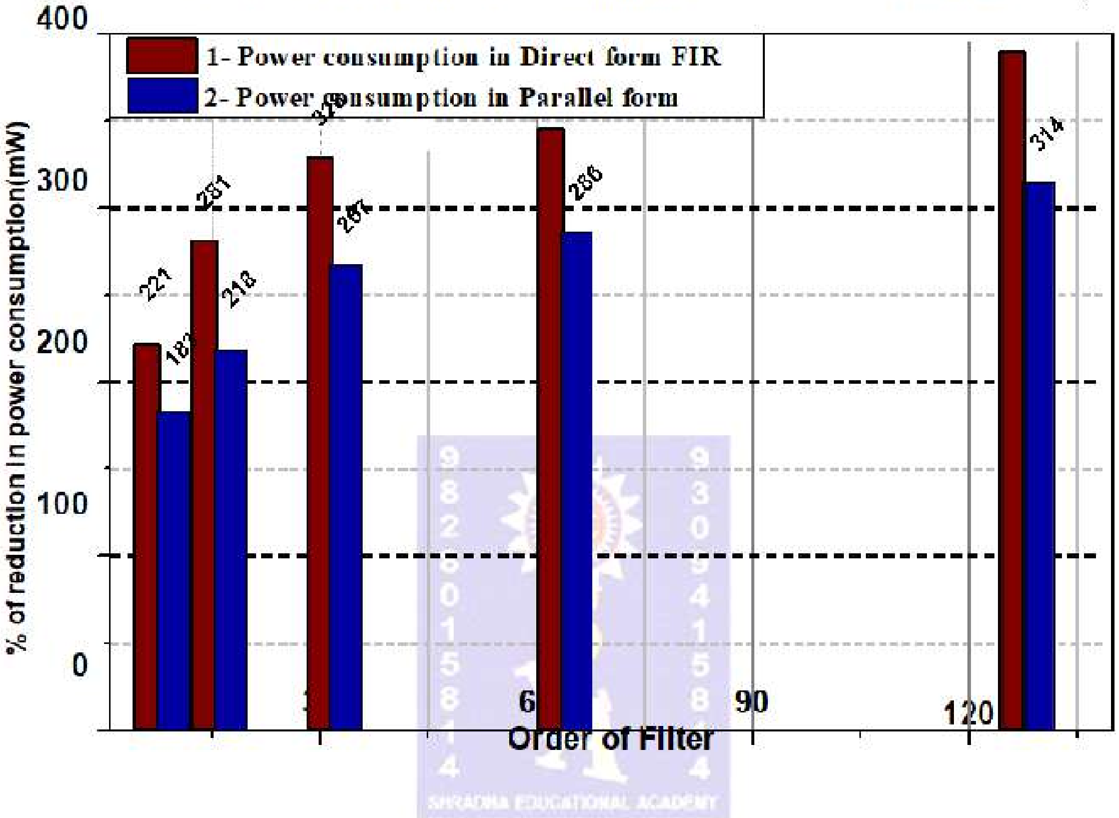


Fig.3.Average power consumption

Figure 3 compares the power consumption of the proposed flexible architectural filter structure to that of an existing filter structure. The suggested FIR filter's average power for various filter lengths is compared to existing filter topologies. The results are shown in Table 2. It is discovered that the power consumption for the suggested design has been reduced by 19.62% when compared to the conventional form.

5. Conclusion

In this work, a proposed FIR filter structure that supports pipeling and parallel processing was taken into consideration. High performance FIR filters are created using a special case of a parallel filter with shared delay components for the subfilters. As shown, the proposed method requires less adders than the traditional direct form filter. The suggested FIR filter design, which is more power-efficient than direct and transposed filters, allows for the trading of the critical path, the maximum fan-out for a node, and the amount of larger delay elements. Given that it achieves both parallel architecture's potential for trade-offs and complexity reduction, the proposed structure seems to hold promise as an appealing alternative. Further investigation must be conducted, nevertheless, at the circuit level. A combination like this was previously unimaginable. Additionally, the suggested FIR filter design can leverage MCM blocks to create implementations that are more effective and require fewer multipliers and adders.

6. References

[1] A.P. Vinod, C.H. Chang, P.K. Meher, A. Singla, Low power FIR filter realization using minimal difference coefficients: part I-complexity analysis, in: Proceedings of IEEE Asia Pacific Conference

on Circuits and Systems (APCCAS 2006), , pp. 1547–1550, 2006

[2] Chao-Huang Wei, Hsiang-Chieh Hsiao, Su-Wei Tsai "*FPGA Implementation of FIR Filter with smallest Processor*" 3rd International IEEE-NEWCAS, 2005

[3] Ryou, A. and Simon, J., "*Active cancellation of acoustical resonances with an FPGA FIR filter*". Review of Scientific Instruments, Vol. 88 (1), 2017.

[4] P. K. Meher, "*Hardware-efficient systolization of DA-based calculation of finite digital convolution*," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[5] B. K. Mohanty and P. K. Meher, "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, pp. 444-452, Feb. 2016

[6] R. Guo, L.S. DeBrunner, K. Johansson, Truncated MCM using pattern modification for FIR filter implementation, in: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2010), pp. 3881–3884, 2010.

[7] Wang Y, Li B, Chen Y, "Digital IIR filter design using multi-objective optimization evolutionary algorithm". Appl Soft Comput 11(2):1851–1857, 2011.

[8] P. K. Meher, "New approach to look-up-table design and memorybased realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592– 603, Mar. 2010.

[9] Abhijit Chandra, Sudipta Chattopadhyay, "*Design of hardware efficient FIR filter A review of the state-of-the-art approaches*", Engineering Science and Technology, an International Journal, Volume -1, Issue: 1, 2015.

[10] W.B. Ye, Y.J. Yu, "*Single-stage and cascade design of high order multiplierless linear phase FIR filters using genetic algorithm*" IEEE Transaction Circuits System I Regular Pap, 60 (11) pp. 2987-2997, 2013.

[11] M. Faust and C. H. Chang, "*Minimal logic depth adder tree optimization for multiple constant multiplication*", Proc. IEEE International Symp. Circuits System, pp. 457-460, 2010.

[12] Muhammad, K., Roy, K.: Reduced computational redundancy implementation of DSP algorithms using computation sharing vector scaling. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 10(3), pp. 292–300 (2002)

[13] K.-Y. Khoo, Z. Yu, and A. N. Willson, "*Design of optimal hybrid form FIR filter*," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), vol. 2, pp. 621–624, 2001.

[14] Lee, K.-Y., Lin, S.-T., Wang, T.-C.: Enhanced double via insertion using wire bending. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29(2), 2010.

[15] X. Lou, Y. J. Yu, and P. K. Meher, "*Fine-grained critical path analysis and optimization for area-time efficient realization of multiple constant multiplications*," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 3, pp. 863–872, 2015.

[16] Umadevi, S., Vigneswaran, T., Vinay, S.K., Seerengasamy, V.: A novel, less area computation sharing high speed multiplier architecture for FIR filter design. Res. J. Appl. Sci. Eng. Technol. 10(7), 816–823 2015.

[17] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and highperformance applications," IEEE J. Solid State Circuits, vol. 39, no. 2, pp348–357, Feb. 2004.