OPTIMIZED FIR FILTER DESIGN FOR EFFICIENT FIXED AND RECONFIGURABLE SYSTEM IMPLEMENTATIONS

Alakurthy Shiva Kumar, Department of ECE, Eklavya University, MP.
DR. Ram Dulare Nirala, HOD of ECE, Eklavya University, MP.

Abstract

This paper proposes the possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, a flow graph for transpose form block FIR filter with optimized register complexity is derived. A generalized block formulation is presented for transpose form FIR filter. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP and less EPS than the proposed structure. Application specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves less ADP and less EPS than the best available FIR filter structure proposed for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves less ADP and less EPS than that of the existing direct-form blocks FIR structure. This Proposed System Implemented using Verilog HDL and Simulated by Modelsim 6.4 c and Synthesized by Xilinx tool. The proposed system implemented in FPGA Spartan 3 XC3S 200 TQ-144.

Keywords: Register unit (RU), Co-efficient Storage Unit (CSU), Inner Product Unit (IPU), Inner Cell Unit (ICU), Pipelined Adder Unit (PAU), Registers, Flip Flops and VLSI.

1. Introduction

The implementation of the proposed FIR filter system in Verilog HDL and its subsequent deployment on the Spartan 3 XC3S 200 TQ-144 FPGA involves a systematic and multi-step process. This implementation overview provides a high-level explanation of the key steps involved in translating the design from a conceptual model to a functional hardware system. The implementation journey begins with the translation of the design specifications and requirements into a hardware description language (HDL), specifically Verilog HDL in this case. The Verilog HDL implementation encompasses the modeling of various components of the FIR filter system, including processing blocks, memory units, control logic, and interfaces. Each module is designed to capture the behavior and functionality of its corresponding hardware component, ensuring a comprehensive representation of the entire system. Once the Verilog HDL design is complete, the next step is simulation using tools such as ModelSim 6.4c. Simulation serves as a crucial validation step, allowing designers to verify the correctness and functionality of the FIR filter system under different input conditions and scenarios. Through simulation, designers can detect and debug errors, refine the design, and ensure that it meets the specified performance requirements before proceeding to hardware implementation. Following successful simulation, the Verilog HDL design undergoes synthesis using Xilinx synthesis tools. During synthesis, the high-level Verilog code is translated into a gate-level netlist, which represents the logical structure and interconnections of the design at the

register-transfer level. Synthesis optimizations are applied to improve area utilization, timing performance, and power consumption, ensuring an efficient and optimized implementation of the FIR filter system. After synthesis, the synthesized netlist is ready for FPGA implementation targeting the Spartan 3 XC3S 200 TQ-144 FPGA device. FPGA implementation involves place-and-route, where the synthesized netlist is mapped onto the physical resources of the FPGA, and timing closure is achieved to meet timing constraints. I/O pin assignment and bitstream generation prepare the design for programming onto the FPGA, configuring it to operate as the FIR filter system. Once the FPGA is programmed, the implemented FIR filter system undergoes thorough verification and testing to validate its functionality and performance. Test vectors are applied, and the output responses are compared against expected results to ensure correctness and accuracy. Iterative refinement may be performed to fine-tune the design, address any issues or constraints, and optimize performance further, ensuring that the implemented system meets the desired specifications and requirements. In the implementation of the FIR filter system involves a structured and systematic approach, starting from design specification and Verilog HDL modeling, through simulation, synthesis, FPGA implementation, and verification. Each step in the implementation process plays a critical role in translating the design from a conceptual model to a functional hardware system, ultimately enabling its deployment in real-world applications for signal processing tasks.

2. LITERATURE SURVEY

1. A. P. Vinod; this paper presents a method to implement FIR filters for SDR receivers using minimum number of adders. We use an arithmetic scheme, known as pseudo floating-point (PFP) representation to encode the filter coefficients. By employing a span reduction technique, we show that the filter coefficients can be coded using considerably fewer bits than conventional 24-bit and 16-bit fixed-point filters. Simulation results show that the magnitude responses of the filters coded in PFP meet the attenuation requirements of wireless communication standard specifications. The proposed method offers average reductions of the number of adders and the number of full adders needed for the coefficient multipliers over conventional FIR filter implementation methods.

2. J. Park, W. Jeong; this paper presents a programmable digital finite impulse response (FIR) filter for high-performance and low-power applications. The architecture is based on a computation sharing multiplier (CSHM) which specifically targets computation re-use in vector-scalar products and can be effectively used in the low complexity programmable FIR filter design. Efficient circuit-level techniques, namely a new carry-select adder and conditional capture flip-flop (CCFF), are also used to further improve power and performance.

3. D.Bhavani; with the development of Software Defined Radio (SDR) technology FIR filters have been concentrated on reconfigurable implementation. The filter coefficients in reconfigurable filters change dynamically in run time. In biomedical applications like Electro Cardio Gram (ECG) the coefficients of FIR filters remain fixed. So there is need to implement a Reconfigurable and Fixed FIR filter structure to support multi-standard communications. The need for reducing the area while increasing the computational speed is still felt. This paper addresses the problem of developing a high-speed and area efficient VLSI architectures for FIR filter in Fixed and Reconfigurable applications. The proposed architecture is compared with that of RFIR and FFIR filters using Ripple Carry Adder (RCA) and Carry Select Adder (CSLA) in Pipelined Adder Unit in terms of Area Delay Product(ADP).

ISSN: 2278-4632
Vol-14, Issue-4, No.02, April: 2024

3. Implementation:

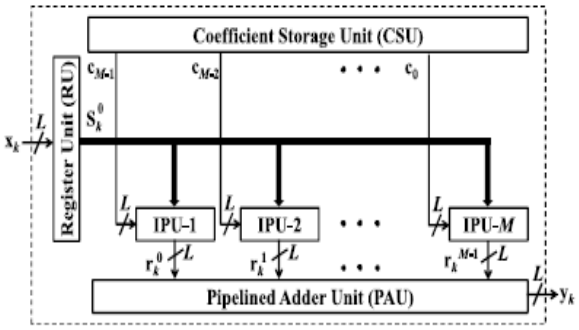

Fig: Proposed structure for block FIR filter

We explore the possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, we have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. We have derived a general multiplier-based architecture for the proposed transpose form block filter for reconfigurable applications. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP and less EPS than the proposed structure.

4.Related Work: The brief introduction of different modules used in this project is discussed below:

4.1 REGISTER UNIT (RU):

The RU receives $\mathbf{x}_k$ during the k^{th} cycle and produces L rows of $\mathbf{S0}_k$ in parallel. It's designed by number of Flip Flop based on Input bit length.

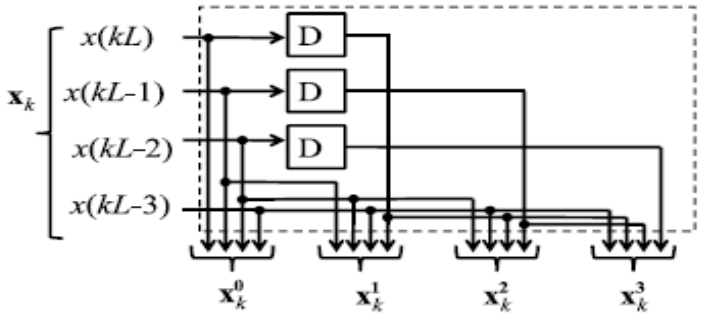

Fig. Proposed RU(Register Unit)

4.2 CO-EFFICIENT STORAGE UNIT (CSU):

The CSU stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using N ROM LUTs, such that filter coefficients of any particular channel filter are obtained in one clock cycle, where N is the filter length.

ISSN: 2278-4632
Vol-14, Issue-4, No.02, April: 2024

4.3 IPU:

L rows of **S** 0kare transmitted to M IPUs of the proposed structure. The M IPUs also receive M short-weight vectors from the CSU such that during the k^{th} cycle, the $(m + 1)^{th}$ IPU receives the weight vector $\mathbf{C}_{M-m-1}$ from the CSU and L rows of **S**0k form the RU. Each IPU performs matrix-vector product of $\mathbf{S}_k^0$ with the short-weight vector **c**m, and computes a block of
L partial filter outputs ($\mathbf{r}_k^m$). Therefore, each IPU performs L inner-product computations of L rows of **S**0k with a common weight vector **c**m. The structure of the $(m+1)^{th}$ IPU is shown in bellow Fig. It consists of L number of L-point inner-product cells (IPCs). All the M IPUs work in parallel and produce M blocks of result ($\mathbf{r}^m_k$). These partial inner products are added in the PAU to obtain a block of L filter outputs. In each cycle, the proposed structure receives a block of L inputs and produces a block of L filter outputs.

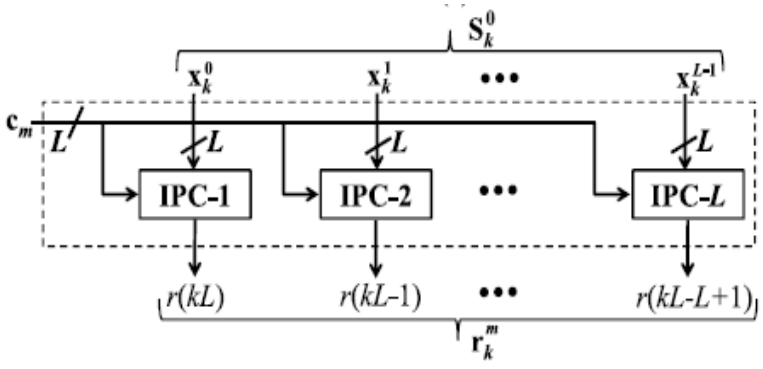

Fig. Proposed IPC Inner Product Cell

4.4 ICU:

The $(l+1)^{th}$ IPC receives the $(l+1)^{th}$ row of $\mathbf{S}_0^k$ and the coefficient vector **c**m, and computes a partial result of inner product r (kL −l), for $0 \le l \le L - 1$. Internal structure of (l + 1)th IPC for L = 4 is shown in bellow Fig.

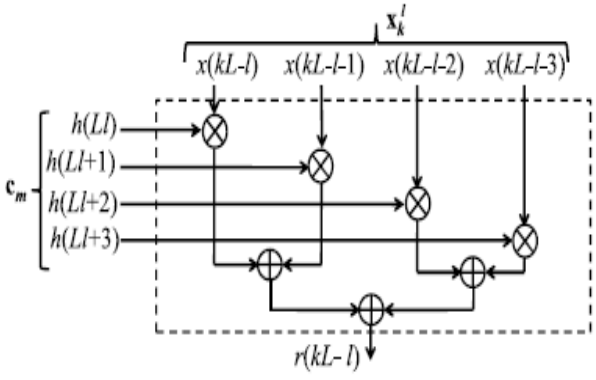

Fig. Proposed ICU Inner Computation Unit

4.5 PIPELINED ADDER UNIT (PAU):

The PAU involves L(M−1) adders and the same number of registers, where each register has a width of (B+ B'), B, and B' respectively, being the bit width of input sample and filter coefficients. The PAU is Designed by number of Adder and Number of Register.

ISSN: 2278-4632
Vol-14, Issue-4, No.02, April: 2024

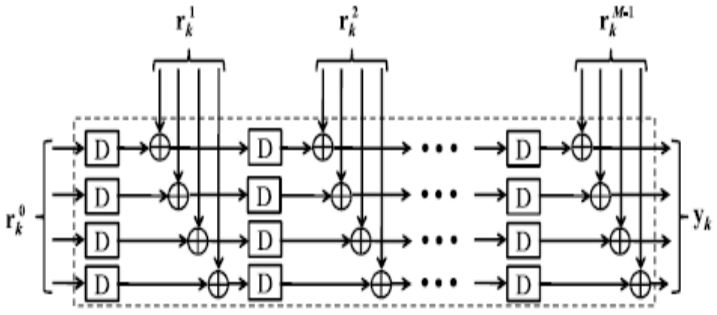

Fig. Proposed Pipelined Adder Unit

4.6 FLIP FLOP and REGISTERS

Flip-flop is a 1-bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flops. Such a group of flip-flops is known as a Register. The n-bit register will consist of n number of flip-flop and it is capable of storing an n-bit word.

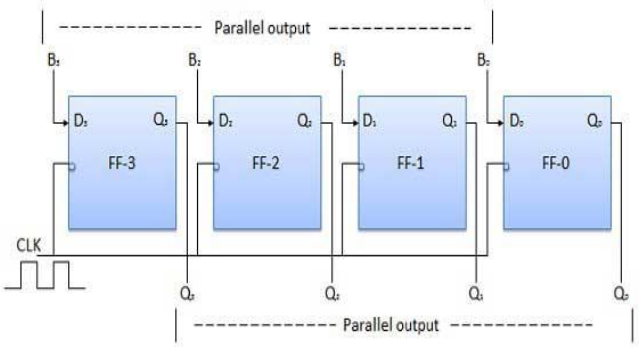

Fig. Registers by Flip Flop

4.7 VEDIC MULTIPLIER:

The 8-bit multiplier is designed using four 4x4 Vedic multipliers which employ Urdhva Tiryagbhyam sutra and RCA technique for partial product addition. The output of these Vedic multipliers is added by modifying the logic levels of ripple carry adder. Block diagram of the proposed 8x8 multiplier is illustrated in bellow figure

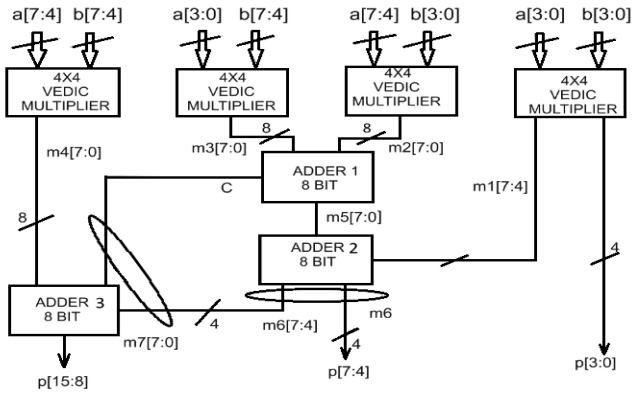

Fig. Vedic Multiplier Design 8x8

ISSN: 2278-4632
Vol-14, Issue-4, No.02, April: 2024

5. Results:

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,123	15,360	7%	
Number of 4 input LUTs	4,268	15,360	27%	
Logic Distribution				
Number of occupied Slices	2,350	7,680	30%	
Number of Slices containing only related logic	2,350	2,350	100%	
Number of Slices containing unrelated logic	0	2,350	0%	
Total Number of 4 input LUTs	4,453	15,360	28%	
Number used as logic	4,268			
Number used as a route-thru	185			
Number of bonded IOBs	43	173	24%	
IOB Flip Flops	14			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	41,538			
Additional JTAG gate count for IOBs	2,064			

Fig. Proposed FIR Filter

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,104	15,360	7%	
Number of 4 input LUTs	3,487	15,360	22%	
Logic Distribution				
Number of occupied Slices	2,033	7,680	26%	
Number of Slices containing only related logic	2,033	2,033	100%	
Number of Slices containing unrelated logic	0	2,033	0%	
Total Number of 4 input LUTs	3,681	15,360	23%	
Number used as logic	3,487			
Number used as a route-thru	194			
Number of bonded IOBs	43	173	24%	
IOB Flip Flops	14			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	43,196			
Additional JTAG gate count for IOBs	2,064			

Fig. Modified FIR Filter

METHOD NAME	AREA IN NUMBER OF LUT			MEMORY IN Kilobytes	DELAY		
FILTER NAME	LUT	GATE COUNT	SLICES	SIZE	DELAY	GATE OR LOGIC DELAY	PATH OR ROUTE DELAY
PROPOSED FIR FILTER (WALLACE MULTIPLIER)	3487	43196	1104	319976 kilobytes	40.862 ns	20.564ns	20.298 ns
MODIFIED FIR FILTER (VEDIC MULTIPLIER)	4268	41538	1123	266344 kilobytes	40.894 ns	18.178 ns	22.716 ns

Fig: AREA DELAY COMPARISON

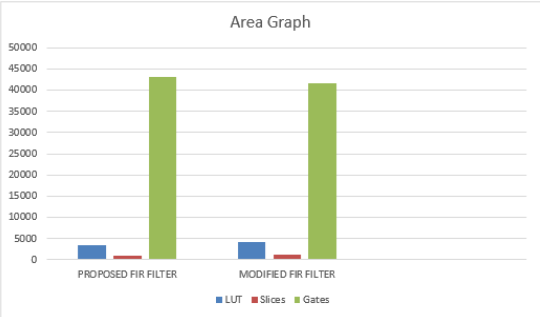

Fig: AREA GRAPH

ISSN: 2278-4632
Vol-14, Issue-4, No.02, April: 2024

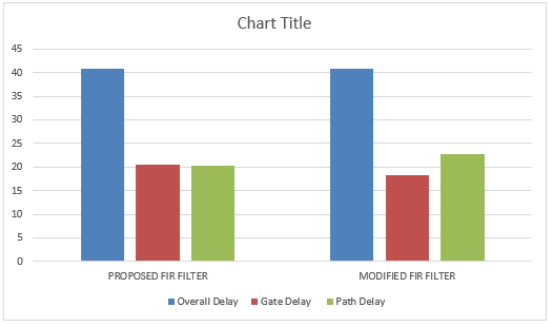

Fig: DELAY GRAPH

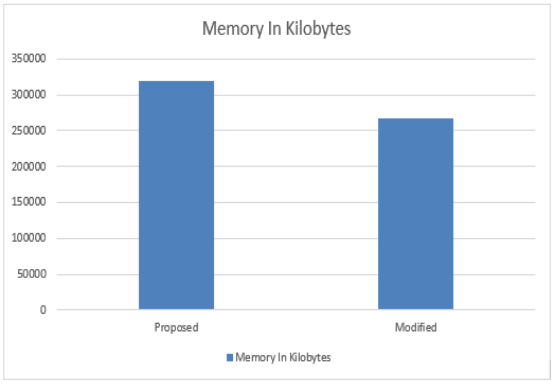

Fig: MEMORY BYTES

6. ACKNOWLEDGEMENT

We would like to thank all the authors of different research papers referred during writing this paper. It was very knowledge gaining and helpful for the further research to be done in future.

7. CONCLUSION:

This is completely eco-friendly project multipurpose and portable. As the cooling un are of small size , silent contains no liquids or gases, have no moving parts and have a long life. The coefficient of performance of this refrigerator is much smaller than that of a conventional compressor -type refrigerator when the required cooling capacity is high. This paper deals with hybrid power generation along with refrigeration system.This project can also monitor the set and inside temperature of refrigerator on LCD module.

REFERENCES

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[2] T. Hentschel and G. Fettweis, “Software radio receivers,” in CDMA Techniques for Third Generation Mobile Systems. Dordrecht, The Netherlands: Kluwer, 1999, pp. 257–283.

[3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, “A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals],” IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 10, pp. 681–694, Oct. 1995.

[4] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in Proc. IEEE Southeastcon, Apr. 1993, p. 1–6.
[5] J. Mitola, Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York, NY, USA: Wiley, 2000.
[6] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," IEEE Trans. Wireless Commun., vol. 7, no. 5, pp. 1669–1675, Jul. 2006.
[7] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," IEEE J. Solid State Circuits, vol. 39, no. 2, pp. 348–357, Feb. 2004.
[8] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 617–621, Aug. 2006.
[9] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 2, pp. 275–288, Feb. 2010.
[10] Basant Kumar Mohanty, Senior Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications IEEE transactions on very large scale integration (vlsi) systems, vol. 24, no. 2, February 2016.
[11] Basant Kumar Mohanty, Senior Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications IEEE transactions on very large scale integration (vlsi) systems, vol. 24, no. 2, February 2016.
[12] Indranil Hatai, Indrajit Chakrabarti, Member, IEEE, and Swapna Banerjee, Senior Member, IEEE An Efficient Constant Multiplier Architecture Based on Vertical-Horizontal Binary Common Sub-expression Elimination Algorithm for Reconfigurable FIR Filter Synthesis IEEE transactions on circuits and systems I: Regular Papers, VOL. 62, NO. 4, APRIL 2015
[13] S.Y. Park and P.K. Meher (Jun 2013) ―Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic,‖ IEEE Trans .Circuits Syst. II, Express Br. 60346–350.
[14] Basant Kumar Mohanty and Pramod Kumar 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, february 2016.
[15] Indranil Hatai, Indrajit Chakrabarti and Swapna Banerjee, "An efficient constant multiplier architecture Based on vertical-horizontal binary common Sub-expression elimination algorithm for Reconfigurable fir filter synthesis," IEEE transactions on circuits and systems—i: regular papers, vol. 62, no. 4, April 2015.

