

HIGH PERFORMANCE FIR FILTER BASED ON PIPELINED PARALLEL PREFIX ADDER FOR SIGNAL PROCESSING APPLICATIONS

Basavoju Harish¹, M.S.S. Rukmini² and K. Sivani³

¹Department of Electronics and Communication Engineering, Vignan's Lara Institute of Technology and Science, India

²Department of Electronics and Communication Engineering, Vignan's Foundation for Science, Technology and Research, India

³Department of Electronics and Instrumentation Engineering, Kakatiya Institute of Technology and Science, India

Abstract

Multiple adders and multipliers make up a complex digital signal processing system (DSP). The efficient design of adders and multipliers improves the DSP system performance. In this paper, modified 4-Tap digital Finite Impulse Response (FIR) filter is built using the Pipelined Brent-Kung adder (PBKA) and a Vedic multiplier. The top-level module (FIR filter) is created by writing PBKA and PBKA-based Vedic multiplier Verilog code. The results of Pipelined Brent Kung adder-based FIR filter are compared with BKA and KSA-based 4-Tap digital FIR filter. According to the synthesis results, the PBKA-based FIR filter operates 57% faster than the BKA-based FIR filter. In terms of Power Delay Product (PDP) PBKA based FIR filter is 22% efficient than BKA based FIR filter. Xilinx 14.7 ISE software is used for simulation, while Virtex-7 FPGA is used for synthesis.

Keywords:

Brent Kung Adder, FIR Filter, Kogge Stone Adder, Parallel Prefix Adder, Pipelined Brent Kung Adder, Vedic Multiplier, Virtex 7 FPGA

1. INTRODUCTION

A finite Impulse Response Filter is a filter that measures impulse response over a finite time. Signal, image, audio processing, and other fields are where the majority of its applications may be found [1]. The inherent properties of the FIR filter make it ideal for generating capable firm filters. These features include linear phase and unrestricted stability, effortlessness of construction, lack of run-over oscillations, higher computing, and the possibility to design filters with coefficients smaller than one. Approximation of the ideal filter is the design process of the FIR filter. The primary methods for designing these filters are windowing techniques. The primary goal of an FIR filter is to remove undesirable noise and distortion while preserving valuable signals [2]. The essential aspects of the FIR filter are Preprocessing, anti-aliasing, etc. which makes it useful in signal processing applications. The FIR filter is the best option for constructing a noise-free filter because it does not require any bit truncation or rounding. For practical image processing applications, the FIR filter is an excellent option that can be implemented in either software or hardware [3].

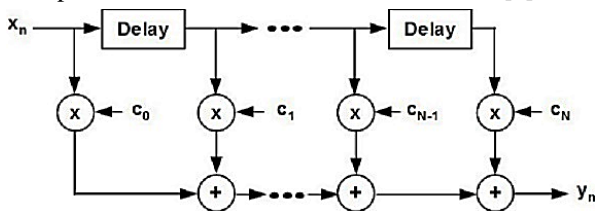


Fig.1. Structure of FIR filter

The Eq.1 represents FIR filter whose structure is shown in Fig.1.

$$Y(n) = \sum_{k=0}^{N-1} h(k) X(N-k) \quad (1)$$

where, $Y(n)$ = filter output; X = input; h = filter coefficients; N = number of coefficients. N and $(N+1)$ adders and multipliers respectively are used to make a Digital FIR filter.

The Fig.1 shows an FIR Filter construction with $(N+1)$ multipliers, N adders, and a series of delay elements. FPGA implementation of Multiplier takes up more LUTs (Lookup tables) of space, and multiplier design is a complex operation that requires a lot of resources, resulting in a high implementation cost [4]. Transfer function computation is the fundamental operation to design a filter, which completes the filter response. The insertion of delay elements in the filter structure results in pipelining. Pipelining improves the system speed but the design occupies more area. For every clock cycle, several inputs are processed, resulting in numerous outputs (parallel processing), increasing hardware complexity once again [5]. The FIR filter gives convolution of the sample response as well as the input. As a result, multiplication in an FIR filter is one of the most powerful operations. The speed of the DSP processor depends upon the design of adder and multiplier blocks [6]. An FIR filter with a good adder and multiplier can outperform the competition. A 4-Tap FIR filter is built in this study employing high-speed adders like the Pipelined Brent Kung adder, Brent Kung adder, and the Kogge Stone adder. The designs were written in Verilog HDL and simulated with Xilinx ISE 14.7 before being implemented on a Virtex 7 FPGA. According to obtained simulation results, the proposed FIR filter with PBKA was found efficient in terms of speed and area.

The rest of the paper is structured as follows: The parallel prefix adders are discussed in section 2, 3 and 4. Modified Brent Kung adder (ie., Pipelined Brent Kung adder) is discussed in section 5. and the multipliers utilized in digital FIR filter construction are discussed in Section 6. Experimental results are exhibited in section 7 and section 8 discusses the conclusion of the work.

2. PARALLEL PREFIX ADDERS

Parallel prefix adders are also called as High-speed adders. Parallel prefix adders are utilized to offer the most efficient output. The parallel prefix addition is divided into three stages: pre-processing, carry creation, and final sum calculation [7].

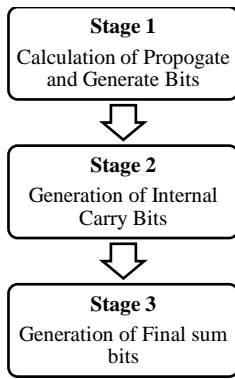


Fig.2. Stages in Parallel prefix adder

Parallel Prefix trees combine this produce and propagate signals in a separate method to provide a required carry input for each piece of addition. The Fig.3 and Fig.4 depict two operators, the black and gray cell, as well as their logic circuits, which are employed in parallel prefix tree architectures. The Black cell generates:

$$\text{Generate, } G = G_i + P_i \cdot G_j \quad (2)$$

$$\text{Propagate, } P = P_i \cdot P_j \quad (3)$$

The gray cell generates: $\text{Generate, } G = G_i + P_i \cdot G_j \quad (4)$

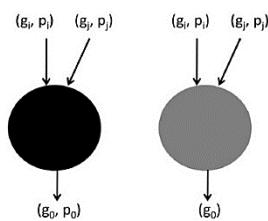


Fig.3. Black and Gray cells

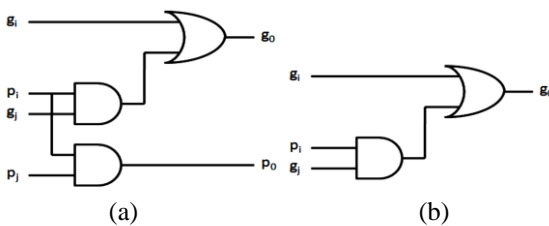


Fig.4. Logic circuits of (a) Black and (b) Gray cells

3. KOGGE STONE ADDER

One of the parallel prefix types of carry look-ahead adders is the KSA. Peter M. Kogge and Harold S. Stone identified this adder as having higher performance in VLSI implementations. The Fig.5 depicts the KSA structure. The KSA is a low-fan-out wide-area fast adder in which each block generates P and G bits as defined by Eq.(2) - Eq.(4) [7].

4. BRENT KUNG ADDER

Richard Peirce Brent and Hsiang Te Kung (1982) introduced Adder, whose architecture is represented in Fig.6. When compared to the KS Adder, it operates faster, has less wiring

complexity, and takes up less space. Prefixes for 2-bit groups are computed using the BK Adder. These prefix trees are used to calculate the other prefix for four-bit groups, and they've also been used to calculate the eight-bit group, and so on. This architecture number of stages is expressed as $\log_2 N$. Eleven gray cells and fourteen black cells are used in the sixteen-bit BK Adder, but fifteen grey cells and 36 black cells are used in the sixteen-bit BK Adder. As a result, BK Adder has a simple architecture and takes up less space than KS Adder. When compared to the KS Adder, the BK Adder runs at a higher frequency [7] [8].

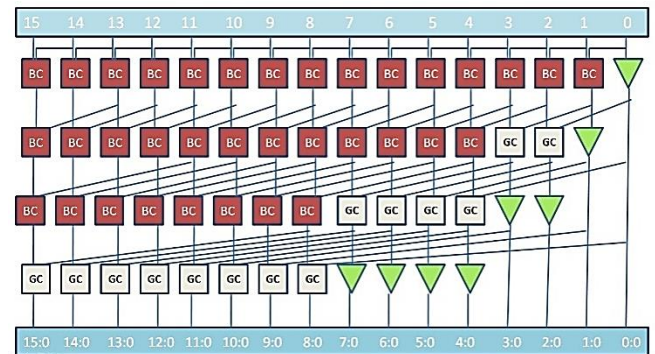


Fig.5. 16-bit Kogge Stone Adder structure

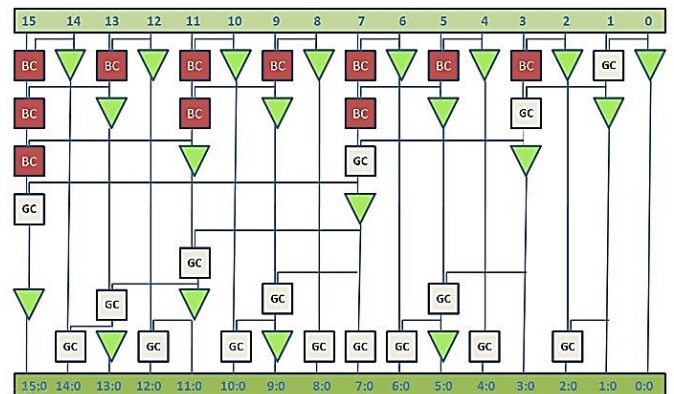


Fig.6. 16-bit Brent Kung adder structure

5. PIPELINED BRENT KUNG ADDER

A 4-stage pipelined architecture is incorporated in sixteen-bit Brent Kung Adder shown in Fig.7. The sixteen-bit pipelined BK Adder utilizes Gray Cells and Black Cells and an additional four-stage pipeline registers. This architecture is complex due to pipelined registers and consumes more area compared to BK Adder. The pipelined BK Adder works as a high-speed Adder. In Fig.10, the numbers of pipelined stages are four which means that the clock speed is four times that of a non-pipelined equivalent circuit. In a circuit without pipelining, there are no circuit components that enable synchronization of the computations. All computations are generated without any sequential components inserted. For pipelined circuits, the reciprocal of the maximum delay establishes the highest possible pipeline clock rate. If the total complex logic is broken into smaller, simpler stages that require synchronization components, then the maximum frequency can be further higher [8]-[10].

6. VEDIC MULTIPLIER

The most essential features of multiplier speed are the buildup of partial-product terms and the multiplication procedure. The partial products are obtained by AND operation between input operands. Adders are commonly used to combine partial products. As a result, the multiplier output can be balanced with speed using an additive process, which may be accomplished by familiarizing the compressor with the additive process so that it can better implement entities like speed and area. To speed up the multiplication process, the Vedic mathematical approach to multiplication is critical [11]-[15].

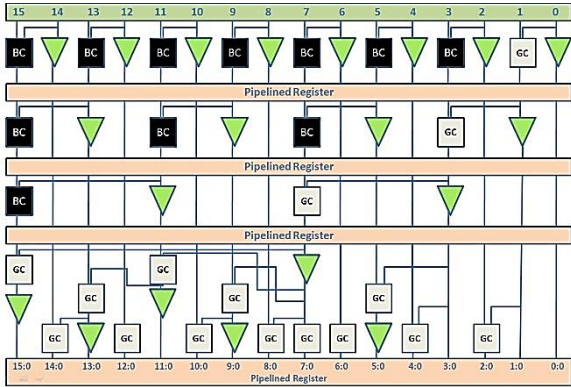


Fig.7. 16-bit Pipelined Brent Kung adder structure

We used the Urdhva tiryagbhyam approach to make rapid multipliers. The Fig.8 shows the 16x16 VM with the Pipelined BK adder.

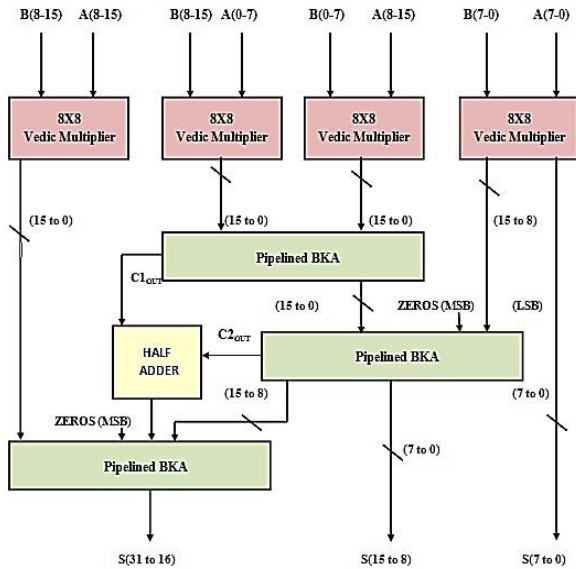


Fig.8. Block diagram of Vedic multiplier using Pipelined BK Adder

7. SIMULATION RESULTS

As part of the filter design, the processing element, which will include an adder, a delay unit, and a multiplier block, will be built. As many times as the demand dictates, the processing element (PE) block will be called or utilized. The 4-Tap FIR filter is built with pipelined BKA, a pipelined BKA-based vedic multiplier, and

a D flipflop as a delay element. The findings are compared to those of a 4-Tap FIR filter built with BKA and KSA. According to the literature, Pipelined BKA operates at a higher speed than other parallel prefix adders, hence a Pipelined BKA-based vedic multiplier is built and employed in FIR filter design [16]-[20].

The Fig.9 depicts the simulation findings. The synthesis results are compared to a 4-Tap FIR filter built with BKA and KSA. The Table.1 shows the outcomes of the synthesis. The Fig.10 depicts the suggested design RTL schematic. The Fig.11 – Fig.14 illustrate comparisons of area, delay, power dissipation, and PDP.



Fig.9. Simulation results of FIR filter using Pipelined BKA adder

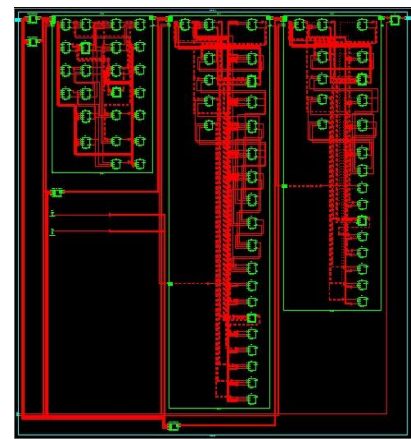


Fig.10. RTL schematic of FIR filter using Pipelined BKA

Table.1. Synthesis results of modified FIR filter

Method	Delay (ns)	Area (LUT)	Power dissipation (mW)	PDP (nJ)
KSA based FIR filter	2.29	250	307	705
BKA based FIR filter	1.57	177	336	528
Pipelined BKA based FIR filter (Proposed)	1.0	193	412	412

8. CONCLUSION

The work focuses on implementing the FIR filter using Pipelined BKA, BKA, and KSA parallel prefix adders. The designs are described using Verilog HDL and implemented on Virtex 7 FPGA. From the synthesis results, it is observed that the FIR filter designed using Pipelined BKA is operating 57% faster

than the FIR filter designed using BKA. The PDP of the FIR filter designed using Pipelined BKA is 22% more efficient than the PDP of the BKA-based FIR filter. Hence from the synthesis results, it is concluded that the FIR filter designed using Pipelined BKA is efficient in terms of delay and PDP and hence it is used in signal processing applications.

REFERENCES

- [1] J.F. Sayed, B.H. Hasan and F. Arifin, "Design and Evaluation of a FIR Filter Using Hybrid Adders and Vedic Multipliers", *Proceedings of International Conference on Robotics, Electrical and Signal Processing Techniques*, pp. 748-752, 2021.
- [1] P.R. Sreesh and L.S. Kumar, "Performance Analysis of Fixed Point FIR Filter Architectures", *Proceedings of 3rd International Conference on Advances in Electronics, Computers and Communications*, pp. 1-6, 2020.
- [2] S. Janwadkar and R. Dhavse, "Strategic Reduction of Area and Power in FIR Filter Architecture for ECG Signal Acquisition", *Proceedings of IEEE International Conference on Counil*, pp. 1-7, 2020.
- [3] U. Maddipati, S. Ahemedali and K.N.J. Priya, "Comparative Analysis of 16-Tap FIR Filter Design using Different Adders", *Proceedings of International Conference on Computing, Communication and Networking Technologies*, pp. 1-4, 2020.
- [4] S. Akash and N. Radha, "An Efficient Implementation of FIR Filter Using High Speed Adders for Signal Processing Applications", *Proceedings of International Conference on Inventive Research in Computing Applications*, pp. 1047-1051, 2020.
- [5] R.V. Arjun and K. Gayathree, "Implementation of Optimized Digital Filter using Sklansky Adder and Kogge Stone Adder", *Proceedings of International Conference on Advanced Computing and Communication Systems*, pp. 661-664, 2020.
- [6] B. Harish and K. Sivani, "Design and Performance Comparison among Various types of Adder Topologies", *Proceedings of International Conference on Computing Methodologies and Communication*, pp. 725-730, 2020.
- [7] B. Harish and K. Sivani, "Design of High Speed Efficient Parallel Prefix Brent-Kung Adder Architecture", *Journal of Green Engineering*, Vol. 10, No. 7, pp. 3508-3519, 2020.
- [8] B. Harish and K. Sivani, "Implementation of Frequency Efficient Multiplexer based CORDIC on FPGA", *International Journal of Advanced Science and Technology*, Vol. 29, No. 5, pp. 753-762, 2020.
- [9] B. Harish and K. Sivani, "Design of MAC Unit for Digital Filters in Signal Processing and Communication", *International Journal of Speech Technology*, Vol. 23, No. 1, pp.1-5, 2021.
- [10] B. Harish and K. Sivani, "Ultra High Speed Full Adder for Biomedical Applications", *International Journal of Reconfigurable and Embedded Systems*, Vol. 10, No. 1, pp. 1-25, 2021.
- [11] B. Harish and K. Sivani, "Performance Comparison of Various CMOS Full Adders", *Proceedings of International Conference on Energy, Communication, Data Analytics and Soft Computing*, pp. 3789-3792, 2017.
- [12] A.A. Wahba and H.A. Fahmy, "Area Efficient and Fast Combined Binary/Decimal Floating Point Fused Multiply Add Unit", *IEEE Transactions on Computers*, Vol. 66, No. 2, pp. 226-239, 2021.
- [13] A. Simson and S. Deepak, "Design and Implementation of High Speed Hybrid Carry Select Adder", *Proceedings of International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies*, pp. 1-6, 2021.
- [14] S. Daphni and K.V. Grace, "Design an Area Efficient Kogge Stone Adder using Pass Transistor Logic", *Proceedings of International Conference on Intelligent Communication Technologies and Virtual Mobile Networks*, pp. 614-618, 2021.
- [15] G. Thakur, H. Sohal and S. Jain, "FPGA-Based Parallel Prefix Speculative Adder for Fast Computation Application", *Proceedings of International Conference on Parallel, Distributed and Grid Computing*, pp. 206-210, 2020.
- [16] A. Kumar, A.A. Shetty and R. Pinto, "Design and Implementation of 64-bit Parallel Prefix Adder", *Proceedings of IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics*, pp. 159-164, 2020.
- [17] Y. Devi Ykuntam, K. Pavani and K. Saladi, "Design and Analysis of High Speed Wallace Tree Multiplier using Parallel Prefix Adders for VLSI Circuit Designs", *Proceedings of International Conference on Computing, Communication and Networking Technologies*, pp. 1-6, 2020.
- [18] G. Thakur, H. Sohal and S. Jain, "Design and Analysis of High-Speed Parallel Prefix Adder for Digital Circuit Design Applications", *Proceedings of International Conference on Computational Performance Evaluation*, pp. 95-100, 2020.
- [19] R. Akhil and B.A. Goud, "Delay and Area Analysis of Hardware Implementation of FFT using FPGA", *Proceedings of IEEE International Conference on Electronics, Computing and Communication Technologies*, pp. 1-6, 2020.