AN EFFICIENT VLSI DESIGN OF FIR FILTER IMPLEMENTATION BASED ON VARIOUS MULTIPLIERS: A REVIEW

Yatheesh KC
Department of ECE
JSS Academy of Technical Education
NOIDA, INDIA
yatheeshkc@jssaten.ac.in

Hritik Gupta
Department of ECE
JSS Academy of Technical Education
NOIDA, INDIA
guptahritk215@gmail.com

Sandeep Kumar Kushwaha
Department of ECE
JSS Academy of Technical Education
NOIDA, INDIA
sandeepsis2000@gmail.com

Prince Yadav
Department of ECE
JSS Academy of Technical Education
NOIDA, INDIA
princeyadav8333@gmail.com

Abstract *– Area optimization and power consumption reduction are the two most crucial considerations in the designing and implementation of any DSP processors. The FIR Filter serves as the essential building block for the designing and developing the DSP processors. The three fundamental modules that make up the FIR Filter are: Multiplier block, Adder block and Flip flops. The multiplier, the slowest block of all, has a significant impact on the FIR Filter's performance. Array multiplier, booth multiplier, and a combination technique have all been used to create the FIR Filter that has been proposed in this paper (Karatsuba and Urdhva Tiryagbhyam) and discovered that the combined method is more efficient than the compared algorithms because it has less delay, which speeds up binary multiplication.*

Key words: FIR, booth multiplier, Karatsuba and Urdhva Tiryagbhyam etc

I. INTRODUCTION

The Digital Signal Processing (DSP) is extensively used in the biomedical fields along with the disciplines of voice and image processing. For DSP applications as well as signal analysis and estimation, digital filters are practical building blocks .With the advent of VLSI-based technology, the number of processes needed for creating digital filters has gradually decreased, which has encouraged the creation of on-chip VLSI-oriented architecture for DSP applications. Digital filters based on their impulse response is divided into two groups: First is FIR filters and second is infinite impulse response (IIR) filters. Compared to IIR filters, FIR filters are easier to use, have better stability, and are guaranteed to have linear phase characteristics.

[1] describes many multiplication algorithms. The most effective mathematical techniques in terms of delay and area are Vedic ones. There is a wealth of material on Vedic mathematical techniques. The most popular approach for greater bit length multipliers is the Karatsuba algorithm. But each algorithm has advantages and pitfalls of its own. The Urdhva-Tiryagbhyam method from Vedic mathematics and the Karatsuba algorithm are combined in the suggested paradigm. The greatest aspects of both algorithms can be utilised effectively to reduce both latency and area by combining them.

The FIR filter used in this project was created using the multipliers stated above. The same's outcomes have been attained and scrutinised individually. The results have also been contrasted on various factors such as power consumption, device utilization area (LUT, Slices, gate count) and delay factors such as gate delay and path delay.

II. RELATED WORK

B.N Mohan Kumar [1] For adding the output of the PE in this study, the Carry Increment Adder (CIA) in the accumulator is used. The AM-CIA-FIR filter is the name of the suggested approach. According to the experiment's findings, the AM-CIA-FIR filter design uses 14.01% less FPGA resources than the PSA-FIR filter.

K. Anjali Rao. [2] They proposed 3-parallel finite impulse response (FIR) structures in this study by using fast FIR algorithms and polyphase coefficient symmetry (FFAs). The proposed FFA-based structure performs better than similar existing structures when compared to the quantity of multipliers needed.

Ratnala Vinay. [3] In order to identify the most energy-effective system, two MAC ways are—the Multiple Constant Multiplication (MCM) system and the Distributed Arithmetic (DA) system—are computed in this study utilizing simulation and synthesis tools. They discover that DA is a memory- and power-effective system that uses fewer memory units, reducing the complexity of the filter's design.

Nithish Kumar V. [4] In this research, researchers offer a modified Multiply and Accumulate (MAC) unit-based FIR filter implementation that is both energy and space economical. With the MAC unit created by both the standard adder and the modified carry select adder, the performance analysis of the suggested FIR filter is approximated. The suggested 5-tap and 9-tap FIR filter architecture was created using Verilog HDL and implemented using SAED 90nm CMOS technology. The Area Delay Product (ADP) of the suggested 5-tap and 9-tap filters improves by 18.26% and 13.94%, respectively, over the traditional technique, according to the findings of the ASIC synthesis. Similar improvements are made to the Power Delay Product (PDP) of 16.80% and 12.54%, respectively.

Shih-Yao Huang. [5] In this research, they investigate the development of a Finite Impulse Response filter for CSP that is both memory- and computation-efficient. With the help of 9-tap Radial and 11-tap Angular filters with a Kaiser windowing technique they obtained 38.6 dB of PSNR for frame interpolation. They use TSMC 40nm technology to create two VLSI circuits for evaluation. They also develop a 2-D CSP engine FPGA system, which runs at 80 megahertz and produces 1024 x 1024 resolution video.

R. Raja Sudharsan. [6] This paper discusses the design of an 8 bit, 8 tap FIR filter for noise filtering and a band pass filter for frequency stabilization. resulting in a 13% reduction in noise compared to the inherent noise in the analogue signal, takes 13.42 ns less time than other active algorithms, and feeds the output from the FPGA to the D/A converter to validate the original input signal.

Bashar S. Mohamad Ali. [7] This paper focus on how to use FIR filters in actual laboratory work to make it simple for students to understand the theory behind them. Utilizing the MATLAB Filter Design and Analysis Tool is part of the effort (FDAT). Using signal frequencies below 5KHz, a real-time FIR filter (with an order of 10) was created in the Arduino programme.

Wangqian Chen. [8] This study proposes an interpolation-based design methodology for centrosymmetric bandpass filters and narrow-band sparse FIR filters. By cascading a model filter and a masking filter, the design technique is implemented. The proposed cascaded realisation of FIR filters, according to simulation results, significantly improves sparsity compared to the traditional direct form or transposed direct form.

Abhyarthana Bisoyi. [9] The main goal of this paper was to use the idea of Vedic multipliers to implement the digital multipliers. Urdhva-Tiryakbyham algorithm of Vedic multiplication is used to perform the digital multiplication. It has been noted that 128 out of 232 bits are required for the 32-bit Vedic multiplier and traditional binary multiplier, resulting in a utilisation of 55% for both multipliers.

K. Deergha Rao. [10] Based on the URDHVA TIRYAKBHYAM (Vertically and cross wise) sutra of Indian Vedic mathematics, two potential architectures for a Vedic real multiplier are suggested in this paper. Additionally, an expression for the path delay of a N–N Vedic real multiplier with minimum path delay architecture is developed. L. According to the implementation results, the Vedic complex multiplier is more effective than the Booth complex multiplier, requiring less route delay and less power consumption, but using more power than the array complex multiplier.

G.Challa Ram. [11] In order to increase performance, a high-speed Vedic multiplier based on 16 sutras (algorithms) was designed, as described in this work.The effectiveness of the Vedic Urdhva Tiryagbhyam (vertical and crosswise) multiplication method, which differs from conventional multiplication, is discussed in this study. By comparing different factors of array multiplier and Vedic Multiplier it is cleared that Vedic multiplier has significantly lower delay. The latency can be decreased as the number of bits increases by utilising the Vedic multiplier rather than the Array multiplier.

Koyel Dey. [12] This research paper proposed a 8-bit digital multiplier which has high speed and consume less power. Multiplier is designed using Vedic algorithms and low power 16nm technology. According to simulation results, the suggested 8-bit Vedic multiplier, which uses 16nm technology, performs better than the other technologies since its Power Delay Product is significantly lower.

KK.Deergha Rao . [13] This study offers a low path delay solution for an IEEE 754 format 32 x 32 bit complex floating point multiplier employing four floating point real multipliers. Two architectures for a 24 × 24 bit Vedic real multiplier are proposed for mantissa multiplication in a 32 bit floating point real multiplier, both of which are based on the Indian Vedic mathematics sutra known as URDHVA TIRYAKBHYAM. Comparing the new architecture's path delay study to that of the traditional 24 x 24 real Vedic multiplier has revealed that the new architecture has a lower path delay than the traditional architecture.

III. DIFFERENT MULTIPLIER'S USED

- **Array Binary multiplier**

  It serves as a block for multiplication in which a collection of similar cells produces incomplete products, which are then simultaneously accumulated. When there is a requirement to reduce the number of computations, high performance machines use the parallel implementation. To create a M*N array multiplier, we need (M-1) n bit adders and M*N AND gates.

If we use ripple carry adders then delay in the adders is going to increase. This is because each full adder has to wait for the carry bit to be calculated from the previous full adder.

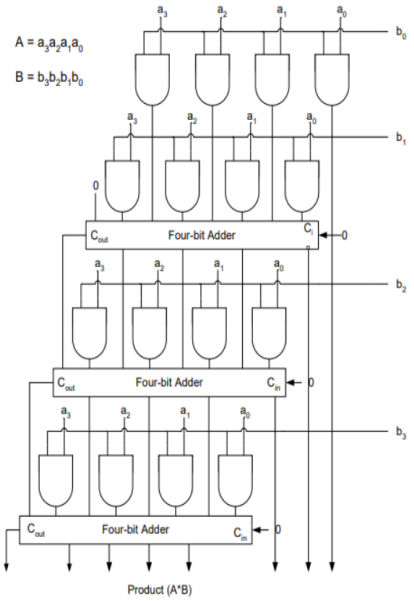


Fig 1 Array multiplier

- **Booth multiplier**

  Using a booth encoder to reduce the quantity of partial products produced during multiplication is one of the most effective and efficient ways to speed up the multiplier. A smaller number of adds must be made with the booth encoding approach than with the traditional multiplication rule. The carry looks ahead adder is the type of adder utilised here. It has been discovered that the net architecture created by the adder and multiplier results in higher speed and area optimization by using fewer partial products and using less energy.

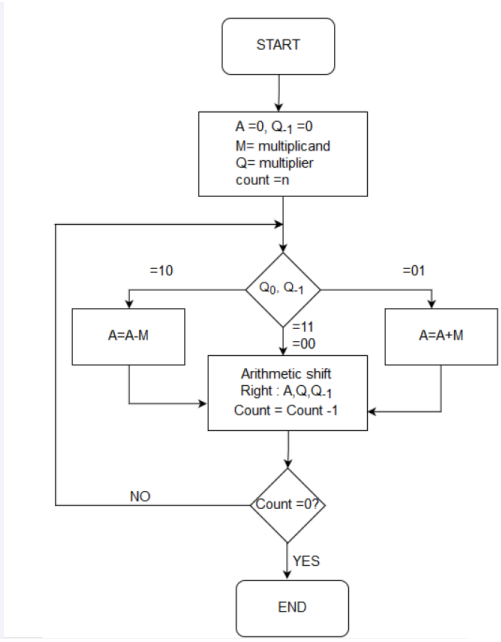


Fig 2. Flow chart multiplier

- **Karatsuba -Urdhva tirayagbhyam Binary multiplier**

An old Vedic multiplication technique is the Urdhva-Tiryagbhyam sutra. It is a universal formula that can be used in any multiplication situation. The formula, which is only one compound word long and says "Vertically and crosswise," is exceedingly brief. The Urdhva Tiryagbhyam method makes it possible to multiply data more quickly by reducing the number of steps needed.

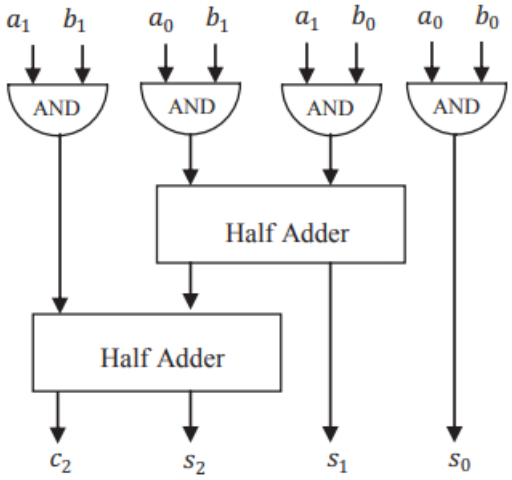


Fig 3. Block diagram of 2x 2-bit Vedic Multiplier

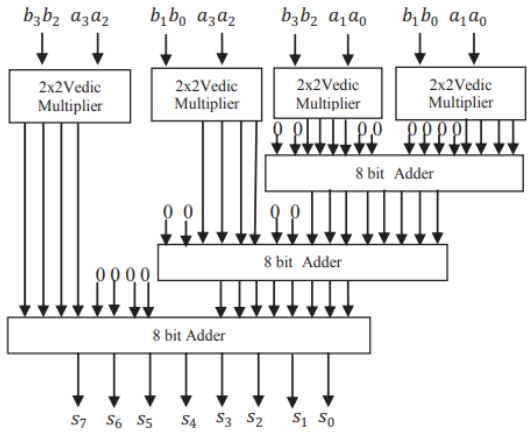


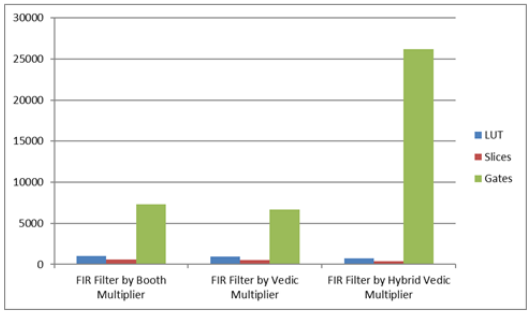
Fig 4. Architecture of 4x4 bit Vedic Multiplier

IV. CONCLUSION

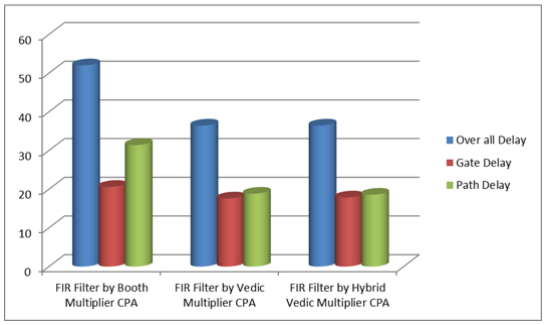
A comparative study is carried between the three is done and it has been found that the FIR Filter design using Karatsuba and Urdhva-Tiryagbhyam is more effective than the Array Multiplier and Booth Multipliers. The comparison has been made between multipliers in terms of power consumption, device utilization area (LUT, Slices, gate count) and delay factors such as gate delay and path delay. Comparison between design summaries obtained from the Xilinx software. Speed comparison is done on the basis of the timing report and synthesis report. This work demonstrates a highly successful combination of the Karatsuba and Urdhva-Tiryagbhyam algorithms to effectively limit the percentage increase in delay and area of a multiplier. By adopting pipelining techniques and efficient adders in place of ripple adders, the model can be more enhance in terms of delay.

METHOD NAME	AREA			DELAY		
Spartan 3XC 3S200 TQ144	LUT	SLICES	GATE COUNT	DELAY	GATE DELAY	PATH DELAY
FIR Filter by Booth Multiplier CPA	1042	571	7283	52.009ns	20.600ns	31.409ns
FIR Filter by Vedic Multiplier CPA	925	489	6702	36.441ns	17.605ns	18.835ns
FIR Filter by Hybrid Vedic Multiplier CPA	764	406	26192	36.442ns	17.867ns	18.575ns

Area Graph



Delay



V. REFERENCES

[1] B.N Mohan Kumar, H.G Rangaraju, "Array Multiplier and CIA based FIR Filter for DSP applications," *2021 Special Issue of First International Conference on Engineering, Science, and Technology (ICEST)*, 2021, pp. 52-59, doi: 10.47392/irjash.2021.020

[2] K. A. Rao, A. Kumar and N. Purohit, "Efficient Implementation for 3-Parallel Linear-Phase FIR Digital Odd Length Filters," *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, 2020, pp. 1-6, doi: 10.1109/CICT51604.2020.9312116.

[3] R. Vinay, T. S. V. S. Vijayakumar, L. M. Saini and B. Singh, "Power efficient FIR filter Architecture using Distributed Arithmetic Algorithm," *2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA)*, 2020, pp. 1-4, doi: 10.1109/ICMICA48462.2020.9242720.

[4] V. N. Kumar, K. R. Nalluri and G. Lakshminarayanan, "Design of area and power efficient digital FIR filter using modified MAC unit," *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 884-887, doi: 10.1109/ECS.2015.7125041.

[5] S. -Y. Huang, W. -C. Chen and C. -T. Huang, "FIR Filter Design and Implementation for Phase-Based Processing," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1748-1752, doi: 10.1109/ICASSP40776.2020.9054246.

[6] R. R. Sudharsan, "Synthesis of FIR Filter using ADC-DAC: A FPGA Implementation," *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*, 2019, pp. 1-3, doi: 10.1109/INCCES47820.2019.9167696.

[7] B. S. M. Ali, Z. K. Farej and A. M. Ibrahim, "Implementing FIR Filters using Arduino Due Platform for Educational Purposes," *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, 2019, pp. 49-54, doi: 10.1109/IICETA47481.2019.9012990.

[8] W. Chen, M. Huang and X. Lou, "Sparse FIR Filter Design Based on Interpolation Technique," *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631685.

[9] A. Bisoyi, M. Baral and M. K. Senapati, "Comparison of a 32-bit Vedic multiplier with a conventional binary multiplier," *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, pp. 1757-1760, doi: 10.1109/ICACCCT.2014.7019410.

[10] K. D. Rao, P. V. Muralikrishna and C. Gangadhar, "FPGA Implementation of 32 Bit Complex Floating Point Multiplier Using Vedic Real Multipliers with Minimum Path Delay," *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2018, pp. 1-6, doi: 10.1109/UPCON.2018.8597031.

[11] G. C. Ram, Y. R. Lakshmanna, D. S. Rani and K. B. Sindhuri, "Area efficient modified vedic multiplier," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016, pp. 1-5, doi: 10.1109/ICCPCT.2016.7530294.

[12] K. Dey and S. Chattopadhyay, "Design of high performance 8 bit binary multiplier using vedic multiplication algorithm with 16 nm technology," *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, 2017, pp. 1-5, doi: 10.1109/IEMENTECH.2017.8076956.

[13] K. D. Rao, C. Gangadhar and P. K. Korrai, "FPGA implementation of complex multiplier using minimum delay Vedic real multiplier architecture," *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, 2016, pp. 580-584, doi: 10.1109/UPCON.2016.7894719.