



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 12, Issue 11, November 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)

# An Efficient Vedic Multiplier Design using Novel Compressor and Encode

Miss.S.Mekala<sup>1</sup>, Mrs.P.Kokila<sup>2</sup>

PG Scholar (M.E-VLSI DESIGN), Nandha Engineering College (Autonomous), Erode, India<sup>1</sup>

Associate Professor, Department of ECE, Nandha Engineering College (Autonomous), Erode, India<sup>2</sup>

**ABSTRACT:** The principal considerations for crafting and implementing a Digital Signal Processing (DSP) Processor Involve the optimization of the area and reduction of power consumption. The Finite Impulse Response Filter serves as the key component in this endeavor, comprising three fundamental modules: adder blocks, flip flops, and multiplier blocks. The performance of the FIR Filter is heavily influenced by the multiplier, which is typically the slowest module. This paper proposes the use of two distinct multipliers - the Array multiplier and the Booth Multiplier - to be incorporated into the Finite Impulse Response Filter, ultimately comparing their performance based on various parameters. Moreover, this project introduces techniques aimed at accelerating processor speed through Vedic mathematics, comparable to the principles used in array and booth multiplication. As the number of bits increases, the gate delay and area demonstrate slower growth rates compared to other multipliers. In light of this, the project presents a methodology for modifying the architecture of the probabilistic multiplier, utilizing existing techniques to effectively enhance processor speed. Employing Verilog HDL and Xilinx 14.2 ISE tools, the proposed filters are successfully designed and implemented, resulting in notable improvements in terms of area and delay. With significantly lower power consumption, reduced delay, and efficient operational frequency, the booth multiplier stands out as the ideal choice for designing the FIR Filter in low voltage and low power VLSI applications.

**KEYWORDS:** FIR filter is heavily influenced by multiplier, vedic, Digital signal processing ,flip flop, array and Booth multiplier.

## I. INTRODUCTION

Finite impulse response (FIR) filters are a crucial component in digital signal processing (DSP) applications, with one of the primary types being infinite impulse response (IIR) filters[1],[3] and [6]. With their high performance, FIR filters are utilized in various digital communication and video processing systems. Depending on the specific application, the FIR filter may need to operate at high sample rates or function as a low-power circuit at moderate sample rates. Unlike IIR filters with internal feedback, the impulse response of an FIR filter settles to zero within a finite number of sample intervals. For an FIR filter of Nth-order, the impulse response lasts for N+1 samples before decaying to zero. This forms the basis of any DSP system and is used in a wide range of applications, including video and image processing, and wireless communications. However, in certain applications such as video processing, the FIR filter should operate at high frequencies. On the other hand, in applications like cellular telephony, the FIR filter must be low-power and operate at moderate frequencies. To improve the overall throughput or reduce power consumption of the original FIR filter, parallel processing, also known as block processing[1],[10] [13] is used. Traditionally, this involves replicating the hardware units present in the original filter.

## II. BACKGROUNDS

### A. FIR Filter

A Finite Impulse Response (FIR) filter in signal processing refers to a filter with a finite duration impulse response, which reaches zero within a limited time frame. This is in contrast to infinite impulse response (IIR) filters, which have internal feedback and can respond indefinitely[7]. The impulse response of an Nth-order discrete-time FIR filter remains active for only N + 1 samples before eventually converging to zero. It should be noted that FIR filters exist in both discrete-time and continuous-time domains.

*B. Types of Multipliers*

There are various types of multipliers, including binary multipliers used in digital electronics such as computers, to multiply two binary numbers. These circuits are constructed using binary adders[8]. There are different computer arithmetic techniques used to implement digital multipliers[9][10], most of which involve calculating a series of partial products that are then summed together. This process is similar to traditional long multiplication used for base-10 integers, but adapted for the binary numeral system. The Array multiplier is an efficient way to organize a combinational multiplier. It is also possible to pipeline Array multipliers to reduce the clock period, although this may increase latency.

**III. EXSISTING SYSTEM**

After extensive contemplation on the inadequacies of conventional transports in System on Chips (SoCs), their creators have reached a crossroads where they intersect With PC architecture designers, who are constantly seeking novel and versatile methods for constructing chips. The expansive reach and development of the Internet has not only captivated the attention of PC architects ,but also SoC engineers, prompting them to adopt packet-based switching networks for future communication infrastructure in SoCs[1],[7] and [13]. Undoubtedly, the actual reason for the success and versatility of the Internet lies in its well-defined protocol stack, with the intention of separating communication from computation.

**IV. PROPOSED SYSTEM**

The use of traditional FIR filters is based on sequential multiplication, resulting in greater arithmetic complexity as the filter order increases due to the summation of products. Numerous approaches have been devised to address this challenge and minimize the arithmetic complexity of FIR filtering. However, existing traditional FIR filters incorporate a larger number of multipliers, causing a delay that needs to be minimized for optimum filter design. To achieve this, it is crucial to reduce the number of multipliers and increase the number of adders, as multipliers are slower and occupy more space compared to adders. By implementing symmetrical properties, a parallel structure can be devised for the filter. Furthermore, incorporating FFA enables the conversion of multipliers into adders and subtracters, further optimizing the filter design.

*A. 4-Bit Multiplier*

The execution of the 4x4 Vedic multiplier in binary is facilitated by the utilization of VERILOG code[1] and[13]. In order to reduce the latency of the 4x4 multiplier in fig.1, a design methodology involving nine full adders and a specialized 4-bit adder has been utilized.

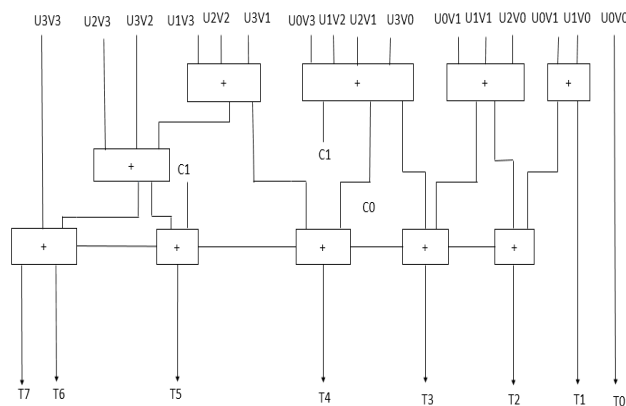


Fig.1.Architecture of 4\*4 Multiplier

**B. Architecture of 4\*4 Multiplier**

In figure A, B, C, D represent four individual inputs shows in Fig.2. C0 and C1 indicate the least significant bit (LSB) and most significant bit (MSB) of the carry outputs, respectively, While Sum represents the overall sum of the four inputs.

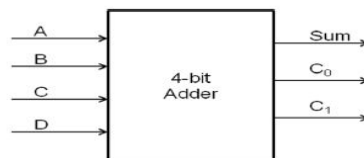


Fig.2. Architecture of 4\*4 special Adder

Below are the Boolean expressions for these values:  $Sum = p \oplus q \oplus r \oplus s$ ,  $c_0 = ((\text{NOT } q) \text{ AND } s) \text{ OR } (r \text{ AND } (\text{NOT } s)) \text{ OR } (q \text{ AND } (\text{NOT } r))$ ,  $c_1 = p \text{ AND } q \text{ AND } r \text{ AND } s$ . Utilizing the UrdhvaTiryakbhyam method, also known as the vertically and crosswise method, provides an alternative approach for multiplying complex numbers. Vertically refers to the straightforward multiplication performed above, while crosswise denotes the diagonal multiplication and subsequent summation. This technique simplifies multi-bit multiplication, breaking it down into single-bit multiplication and addition. As a result, all partial products are generated in one step, greatly minimizing carry propagation from the least significant bit (LSB) to the most significant bit (MSB) during the addition process. This sutra can be implemented from either the right-hand side or the left-hand side, offering a more efficient solution.

**C. Vedic Multiplier**

In this section a baseline Vedic multiplication, Vedic mathematics is an ancient mathematics that has a unique system of computation based on simple rule and basic principles with which mathematical problem can be solved. It deals with many modern mathematical terms[4],[10] including arithmetic, trigonometry, geometry (plane, co-ordinate), quadratic equations, factorization and even calculus. The beauty of Vedic mathematics lies in the fact that it reduces the complex looking calculations in conventional mathematics to a very simple one.

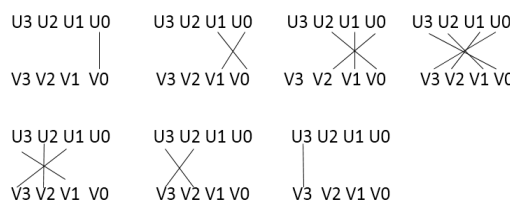


Fig.3.Line diagram

Let us take an example of two four bit binary numbers a and b by applying vertically and crosswise method to it, as shown in figure 3.

$$\begin{aligned}
 p_0 &= a_0b_0 \text{ ----- (I)} \\
 s_1p_1 &= a_1b_0 + a_0b_1 \text{ ----- (II)} \\
 s_2p_2 &= s_1 + a_2b_0 + a_1b_1 + a_0b_2 \text{ ----- (III)} \\
 s_3p_3 &= s_2 + a_3b_0 + a_2b_1 + a_1b_2 + a_0b_3 \text{ ----- (IV)} \\
 s_4p_4 &= s_3 + a_3b_1 + a_2b_2 + a_1b_3 \text{ ----- (V)} \\
 s_5p_5 &= s_4 + a_3b_2 + a_2b_3 \text{ ----- (VI)} \\
 s_6p_6 &= s_5 + a_3b_3 \text{ ----- (VII)}
 \end{aligned}$$

From equation no. (I) to (VII), the final result can be obtained as  $s_6p_6p_5p_4p_3p_2p_1p_0$ .

### C. Compressor

Compressor is a type of adder circuit which reduces the operands of the multiplier. In this type of multiplication 4 or more bits are added simultaneously so as to increase the speed. It replaces the other adder circuits like half adder and full adder. Generally XOR/XNOR gates[4],[5],[13] and [14] multiplexers are used for its design in fig.4.

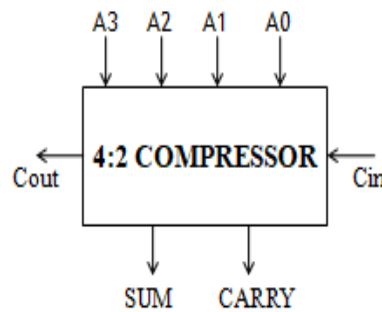


Fig.4.4\*2 Compressor

$$A4+A3+A2+A1+Cin=SUM+2(CARRY + Cout)$$

Lower significant compressor bit the input Cin is the carry bit generated and Cout is the carry output of the compressor circuit.

## V. HARDWARE AND SOFTWARE USED

### A. VERILOG

In the realm of electronics, a hardware description language (HDL) serves as a tailor-made computer language utilized to program the intricate structure, design, and operations of electronic circuits, specifically those involving digital logic. The objective of employing a hardware description language is to provide a definitive and formal representation of an electronic circuit, simplifying automated analysis, simulation, and virtual testing of the aforementioned circuit[6],[10] and [12]. Furthermore, it empowers the translation of an HDL program into a lower-level specification of physical electronic components, akin to the assortment of masks implemented in the development of an integrated circuit. In essence, a hardware description language bears a certain similarity to commonly used programming languages, such as C, as it encompasses textual explanations comprising expressions, statements, and control structures. However, one crucial differentiation between standard programming languages and HDLs is the explicit integration of the concept of time in the latter.

### B. Design using HDL

The utilization of HDL has resulted in significant improvements in efficiency, making it the cornerstone of contemporary digital circuit design. Typically, designs start with a set of requirements or a high-level architectural diagram. Flowchart applications or state diagram editors are often used to prototype control and decision structures. The process of writing a HDL description heavily relies on the circuit's nature and the personal coding style of the designer. Acting as a "capture language," HDL initially begins with a high-level algorithmic description, such as a C++ mathematical model. To automate the creation of repetitive circuit structures in the HDL language, designers often turn to scripting languages like Perl. Special text editors also provide features such as automatic indentation, syntax-dependent coloration, and macro-based expansion of entity/architecture/signal declarations. The HDL code then goes through a code review or auditing process. Automated checkers are employed to prepare the HDL description for synthesis. These checkers flag any deviations from standardized code guidelines, identify potential ambiguities in code constructs before they cause misinterpretation, and check for common logical coding errors such as dangling ports or shorted outputs.

### C. Spartan-3

Crafted specifically to cater to the demands of high volume and cost-sensitive consumer electronic applications, the Spartan-3 family of Field-Programmable Gate Arrays boasts a formidable lineup of eight members, offering a wide range of densities from 50,000 to five million system gates. Continuously improving on the success of its predecessor, the Spartan-IIIE family, this powerful family boosts logic resources, internal RAM capacity, total



number of I/Os, and overall performance, while also enhancing clock management functions. These developments, coupled with state-of-the-art process technology, enable greater functionality and bandwidth per dollar, setting new industry standards for programmable logic. With exceptional affordability, Spartan-3 FPGAs are ideally suited for various consumer electronics applications, including broadband access, home networking, display/projection, and digital television equipment. Providing a superior alternative to mask-programmed ASICs,[7],[9] and [13] FPGAs eliminate the high initial cost, lengthy development cycles, and rigidity associated with traditional ASICs. Moreover, Their Programmability allows for field upgrades without any Hardware replacement, something that is impossible with ASICs.

D. Architectural overview

The structure of the Spartan-3 family encompasses five crucial programmable components. Primarily, the Configurable Logic Blocks (CLBs) house RAM-based Look-Up Tables (LUTs), facilitating the integration of logical and storage elements that can function as flip-flops or latches. These CLBs can be customized to execute various logical functions and store data. Secondly, the Input/output Blocks (IOBs) control the flow of information between the I/O pins and the internal logic of the device. Each IOB supports both bidirectional data flow and 3-state operations. A wide range of twenty-six signal standards is available, including eight advanced differential standards. Additionally, Double Data-Rate (DDR) registers are incorporated, while the Digitally Controlled Impedance (DCI) feature simplifies board layouts by offering automatic on-chip terminations. Thirdly, the Block RAM serves as a storage unit in the form of 18-K bit dual-port blocks. Fourthly, the Multiplier blocks accept two 18-bit binary inputs and calculate their product. Lastly, the Digital Clock Manager (DCM) blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals. The DCMs are strategically positioned at the ends of the outer block RAM columns. The comprehensive network of traces and switches in the Spartan-3 family architecture interconnects all five functional elements, facilitating the transmission of signals among them. Each functional element is equipped with a switch matrix, enabling multiple connections to the routing. In terms of physical arrangement, a ring of IOBs surrounds a regular array of CLBs. The XC3S50 device has a single column of block RAM integrated into the array, while the XC3S200 to XC3S2000 devices have two columns, and the XC3S4000 and XC3S5000 devices have four columns. Each column consists of several 18-Kbit RAM blocks, with each block dedicated to a multiplier.

VI. RESULTS

The simulated results of the suggested system are available through Xilinx ISE 14.2, providing a comprehensive synthesis report. The data table displays Fig.5 the various parameter utilized in the calculation of both the current and proposed systems, utilizing the Spartan-3E processor.

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	11,440	0%	
Number of Slice LUTs	83	5,720	1%	
Number used as logic	83	5,720	1%	
Number using O6 output only	69			
Number using O5 output only	0			
Number using O5 and O6	14			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	33	1,430	2%	
Number of MUXC's used	0	2,860	0%	
Number of LUT Flip Flop pairs used	83			
Number with an unused Flip Flop	83	83	100%	
Number with an unused LUT	0	83	0%	
Number of fully used LUT-FF pairs	0	83	0%	
Number of slice register sites lost to control set restrictions	0	11,440	0%	
Number of bonded IOBs	32	102	31%	
Number of RAMB16W1ERS	0	32	0%	

Fig.5.Proposed synthesis report

A. Proposed RDL schematic

Upon completion of the synthesis process, the RTL schematic Fig.7. is automatically generated, incorporating the desired functionality .The schematic presents a clear overview of the routing between the various cells.

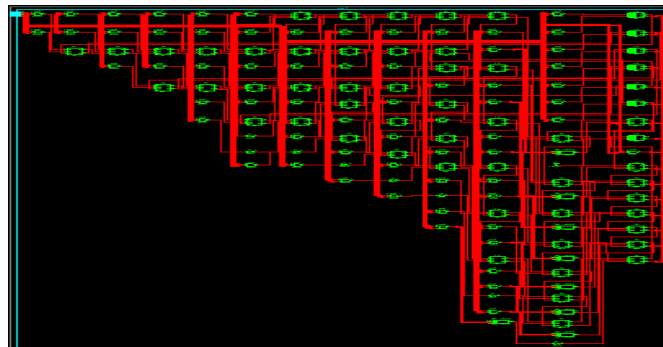


Fig.6.RTL Schematic Block

**B. Simulation output**

The simulation output Fig.7 can execute two binary values as input A,B and output is X. The input A is 00000010,another input B is 00000010 the output X value will be 0000000000000100.

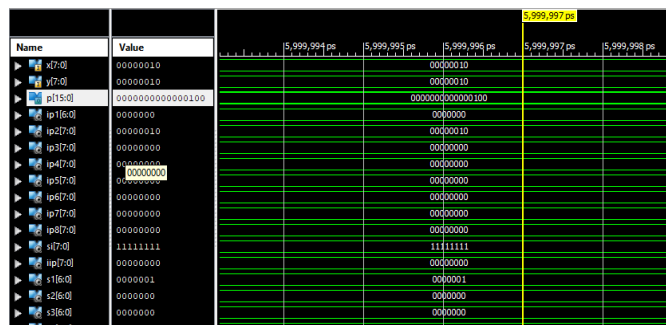


Fig.7.Simulation output

**C. Power result**

The proposed system power and delay simulation can expressed as following diagram fig.8.

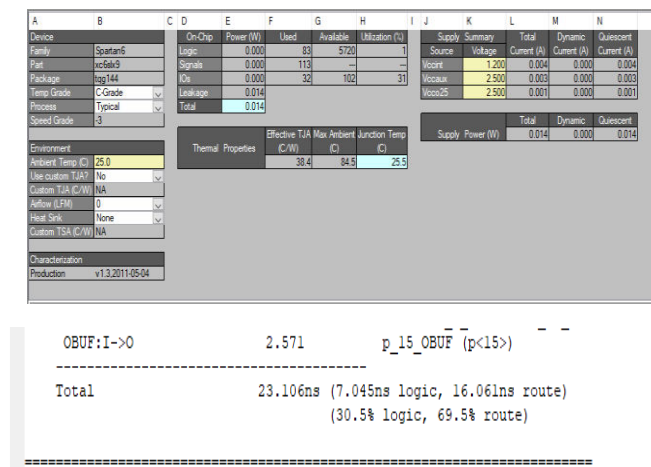


Fig.8.Power &delay analysis



COMPARISON

After simulating & generating the results, then comparing the terms of existing [1] and proposed vedic multiplier in Xilinx ISE as shown in below table-1. The comparison of slice registers, LUTs, IOBs, etc... are synthesized in design summary.

In this, it compares the reference paper of booth multiplier and vedic multiplier It calculates the delay & power.

S.No	Terms	Existing system	Proposed system
1.	No.of slice LUTs	114	83
2.	No.of Occupied slice	60	33
3.	No.of IOBs	38	32
4.	Delay(ns)	29.198	23.106
5.	Power(w)	0.052	0.014
6.	Power Delay Product (PDP)	1.51	0.323

Table.1.Comparison of existing and proposed system.

VII. CONCLUSION

Ultimately, a Vedic multiplier has been used in the design of a Finite Impulse Response (FIR) filter. By using this multiplier, the processor's speed is boosted due to the huge reduction in design complexity. By using a probabilistic method, the FIR filter processor has been able to shorten the time needed for its multiplication operation. High-speed multiplication processes must be executed efficiently in order for general processors and digital signal processors (DSPs) to function. The sutra method from the ancient Vedas is used in this proposed multiplier to perform basic mathematical computations[2],[4] and [13]. The globe can attain improved performance and quality for new technology devices by adopting these ancient Vedic sutras. The suggested block diagram shows a decrease in terms of area, time delay, and power consumption for the FIR filter when utilizing the proposed multiplier.

VIII. FUTURE WORKS

The same Vedic multiplication will be carried out in the future with activity-based switching of the encoding multipliers and multiplicand; in the worst scenario, the inputs' Vedic encoding will be altered for the multiplication.

ACKNOWLEDGMENT

The success of a work depends on the team work and co-operation of various people involved either directly or indirectly. I express my thanks to our beloved Chairman of Sri Nandha Educational Trust Thiru.V.Shanmugan, and our beloved Secretary, Thiru.S.Thirumoorthi, of Nandha Educational Institutions for providing us all the basic amenities to complete the course successfully. I wish to express my deep sense of gratitude and thanks to our beloved Principal Dr.N.Rengarajan, for the excellent facilities and continual support provided during the course study and project work. I am grateful to Dr.C.N.Marimuthu, Dean (R & D) and Dr.S.Kavitha, Dean (ECE) for providing guidance to complete our project work in shape. I am very indebted to Project guide Mrs.P.Kokila, Assistant Professor (ECE) project Coordinator Mr.G.Prabhakaran, Assistant Professor (ECE), for their valuable guidance, effective suggestions and kind advice to complete my project. I thank all the Teaching and Non-Teaching staff members of Electronics and Communication Engineering Department for their valuable suggestions in my project work.



#### REFERENCES

- [1] Jaeha Kung and Youngjoo Lee, "Simplified Compressor and Encoder Designs for Low-Cost Approximate Radix-4 Booth Multiplier" IEEE Transactions on circuits and system, 2023.
- [2] H. Pei, X. Yi, H. Zhou, and Y. He, "Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 68, no. 1, pp. 461–465, Jan. 2021.
- [3] M. Bakhshalipour, P. Lotfi-Kamran, A. Mazloumi, F. Samandi, M. Naderan, M. Modarressi, and H. Sarbazi-Azad, "Fast Data Delivery for Many-Core Processors," IEEE Transactions on Computers (TC), 2018.
- [4] E. Viegas, A. O. Santin, A. Franc, a, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, "Towards An Energy-Efficient Anomaly-Based Intrusion Detection Engine for Embedded Systems," IEEE Transactions on Computers (TC), 2017.
- [5] M. Alioto, Enabling the Internet of Things: From Integrated Circuits to Integrated Systems. Cham, Switzerland: Springer International, 2017.
- [6] Andrea Bonetti , Adam Teman , Philippe Flatresse and Andreas Burg "Multipliers -Driven perturbation of coefficients for low power operation in reconfiguration FIR filters ,"IEEE Trans.2017.
- [7] M. Salehi, A. Ejlali, and B. M. Al-Hashimi, "Two-Phase Low-Energy N-Modular Redundancy for Hard Real-Time Multi-Core Systems," IEEE Transactions on Parallel and Distributed Systems (TPDS), 2016.
- [8] Y.Huan et al., "A 101.4 GOPS/W reconfigurable and scalable controlcentric embedded processor for domain-specific applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 63, no. 12, pp. 2245–2256, Dec. 2016.
- [9] A.Ejlali, B. M. Al-Hashimi, and P. Eles, "Low-Energy Standby-Sparing for Hard Real-Time Systems," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 2012.
- [10] R. Melhem, D. Mosse, and E. Elnozahy, "The Interplay of Power Management and Fault Recovery in Real-Time Systems," IEEE Transactions on Computers (TC), 2004.
- [11] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," IEEE J. Solid-State Circuits, vol. 39, no. 2, pp. 388–391, Feb. 2004.
- [12] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 497–510, May 2004.
- [13] D. Zhu, R. Melhem, and B. R. Childers, "Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multiprocessor Real-Time Systems," IEEE Transactions on Parallel and Distributed Systems (TPDS), 2003.
- [14] L.Wang and N. R. Shanbhag, "Low-power filtering via adaptive errorcancellation," IEEE Trans. Signal Process., vol. 51, no. 2, pp. 575–583, Feb. 2003.



SJIF Scientific Journal Impact Factor

Impact Factor: 8.423



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details