

A high-speed fault-tolerant vedic FIR filter architecture on FPGA using reversible logic for ML accelerator applications

Guguloth Erna¹, Vadtya Saidulu², Chinta Manjusha³, GVK Murthy⁴, Vijaya Bhaskar Chalampalem⁵,
Puthalapattu Muni Sekhar⁶

¹Department of Electronics and Communication Engineering, PACE Institute of Technology and Sciences, Ongole, India

²Department of Electronics and Communication Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

³Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India

⁴Department of Electrical and Electronics Engineering, PACE Institute of Technology and Sciences, Ongole, India

⁵Department of Electronics and Communication Engineering, Siddharth Institute of Engineering and Technology, Puttur, India

⁶Department of Electrical and Electronics Engineering, Siddharth Institute of Engineering and Technology, Puttur, India

Article Info

Article history:

Received Dec 27, 2025

Revised Jul 10, 2026

Accepted Jul 18, 2026

Keywords:

Error correction codes

Fault tolerant

Finite impulse response filter

Reversible logic

Vedic multiplier

ABSTRACT

Digital signal processing circuits for biomedical monitoring, wireless communication, and machine-learning hardware require high speed, low power consumption, and fault tolerance. This work proposes a high-speed, power-efficient, and fault-tolerant finite impulse response (FIR) filter architecture by integrating Vedic mathematics, reversible logic, and error-correction coding into an FPGA-based framework for machine-learning accelerators. A 16-tap FIR filter is implemented on a Xilinx Virtex-5 FPGA using the Urdhva-Tiryakbhayam Vedic multiplication technique with reversible Peres and controlled-NOT (CNOT) gates and a reversible look-ahead carry adder (LACA), enabling parallel computation with reduced information loss. A Hamming-code-based error correction code (ECC) module is incorporated to detect and correct single-bit soft errors without the hardware overhead associated with triple modular redundancy (TMR). Synthesis results demonstrate a propagation delay of 30.932 ns and power consumption of 2.645 mW, achieving a 5.57% latency improvement over an multiple constant multiplication (MCM)-based FIR baseline while providing single-bit fault correction. The proposed architecture demonstrates its suitability as a replacement for multiply-accumulate (MAC) units in convolutional neural network accelerators, particularly for power-constrained and fault-sensitive edge artificial intelligence applications.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vijaya Bhaskar Chalampalem

Department of Electronics and Communication Engineering

Siddharth Institute of Engineering and Technology

Narayanavanam Road, Puttur, India

Email: vijayas4u@gmail.com

1. INTRODUCTION

Digital signal processing (DSP) is widely used in healthcare monitoring, communication systems, and artificial intelligence, where filters improve signal quality by removing noise prior to analysis [1]. Finite impulse response (FIR) filters are preferred due to their stability and linear-phase response, making them suitable for biomedical and multimedia applications [2]. Field-programmable gate array (FPGA)-based implementations are commonly used to meet real-time processing requirements; however, conventional irreversible logic designs suffer from information loss during switching, leading to increased power

consumption and thermal overhead. In addition, soft errors may propagate during repeated computations, degrading accuracy in machine-learning and convolution-based systems [3].

Although techniques such as triple modular redundancy (TMR) improve fault tolerance, they incur nearly three times hardware overhead, limiting their suitability for resource-constrained FPGA platforms [4]. To overcome these limitations, recent approaches have explored Vedic arithmetic for fast multiplication and reversible logic for low-power computation, offering a promising direction for FIR filter architectures that achieve reduced delay and power consumption while maintaining computational reliability.

The novelty of this work lies in the first reported integration of Urdhva-Tiryakbhayam (UT)-based Vedic multiplication, quantum-inspired reversible logic (Peres and controlled-NOT (CNOT) gates), and Hamming-code-based error correction code (ECC) within a unified FIR filter architecture implemented on a Xilinx Virtex-5 FPGA for machine-learning accelerators. While prior works have independently explored Vedic multipliers for speed [5], reversible adders for power efficiency [6], [7], and ECC-based fault tolerance [8], no existing FPGA implementation combines all three or evaluates their combined impact on delay, power, area, and reliability. The proposed design achieves a 5.57% reduction in latency and 1.42% reduction in power compared to the multiple constant multiplication (MCM)-FIR baseline, while also enabling single-bit soft-error correction. It implements a 16-tap FIR filter using UT-based Vedic multiplication with reversible Peres and CNOT gates to reduce information loss, integrated with a Hamming ECC module for error detection and correction across parallel outputs. FPGA synthesis on Xilinx Virtex-5 reports a 30.932 ns propagation delay and 2.645 mW power consumption, confirming improved performance over the baseline. The architecture is also suitable as a replacement for conventional multiply-accumulate (MAC) units in convolutional neural network (CNN) accelerators. The remainder of this paper is organized as follows: section 2 reviews related work, section 3 presents the proposed architecture, section 4 presents the results and discussion, and section 5 concludes the work.

2. LITERATURE SURVEY

Considerable research has focused on improving FIR filter performance in terms of speed, area, and power through arithmetic optimization, distributed arithmetic, and parallel multiplier designs. However, many existing approaches either increase hardware overhead due to redundancy or lack integrated soft-error correction, particularly in FPGA implementations. Although Vedic multipliers have been explored for faster computation and reversible adders for power efficiency, the combined use of reversible logic with ECC-based protection in parallel FIR architectures remains largely unexplored.

Hatai *et al.* [9] proposed a high-order FIR filter requiring additional interconnections among constant-multiplication and accumulation units, achieving only a significant improvement in delay-area-power product due to interconnect overhead. A signed FIR filter using Radix-4 partial-product reduction was presented in [5], where horizontal and vertical multiplication accelerates accumulation. Kumar *et al.* [10] proposed a FIR filter using Radix-2 signed multiplication for electrooculography (EOG) signal compression, employing 4:2 and 3:2 compressors to reduce hardware resources by about 50%. However, the design focuses on area efficiency and does not incorporate fault tolerance or soft-error correction.

Mageshwari *et al.* [6] implemented a reversible Vedic multiplier in Cadence 180 nm CMOS, achieving reduced propagation delay through quantum-inspired adders and parallel processing. A critical review of existing work shows that FIR filter designs typically optimize a single metric speed (Vedic or distributed arithmetic), area (MCM or CSE), power (reversible logic), or reliability (TMR or ECC) rather than addressing all four simultaneously. TMR-based methods [11] provide fault tolerance but incur nearly threefold hardware overhead, limiting their use in resource-constrained FPGAs. Reversible-logic designs [6], [7] reduce power without supporting error correction, while ECC-based approaches [1], [8] improve reliability but do not leverage Vedic arithmetic for speed. No prior FPGA implementation integrates reversible Vedic multiplication, quantum-inspired logic, and Hamming-code ECC within a single FIR architecture evaluated for machine-learning accelerators, forming the key motivation and novelty of this work.

3. PROPOSED METHOD FOR VEDIC FIR FILTER USING REVERSIBLE LOGIC GATES

The proposed Vedic FIR filter is designed using low-power reversible logic to avoid information loss during computation. Multiplication is performed using the UT Vedic method, which enables parallel generation of partial products for reduced latency. The multiplier integrates reversible Peres and CNOT gates to minimize switching energy [12]. The architecture follows a four-stage pipeline: 16-bit inputs are processed using a 16×16 Vedic multiplier that decomposes operations into four parallel 8×8 sub-multiplications; partial products are accumulated using a reversible look-ahead carry adder (LACA) to reduce carry propagation delay; four parallel FIR channels (X1–X4) execute simultaneous filtering and forward outputs (OUT1–

OUT4); and finally, a Hamming-code-based ECC module generates parity bits and uses syndrome computation $S=yH^T$ to detect and correct single-bit errors. The design is implemented in VHDL, simulated in Xilinx ISE, and synthesized on a Virtex-5 XC5VLX110T FPGA, with timing analysis from post-synthesis reports and power estimation using the XPower Analyzer tool.

In conventional irreversible logic, each bit erasure dissipates energy of approximately $kT \cdot \ln 2$, where k is Boltzmann's constant and T is temperature. Bennett's principle of reversible computing ensures lossless computation through bijective input-output mapping [12]. A reversible circuit is characterized by an $n \times n$ structure, evaluated using metrics such as number of gates (NG), fundamental expense (FE), constant inputs (CI), garbage outputs (GO), and total reversible logic cost (TRLIC), where lower TRLIC indicates improved hardware efficiency and reduced power consumption [7].

3.1. Lossless logic gates

3.1.1. Controlled-NOT gate

The CNOT gate operates on qubits and follows the principles of linearity and quantum superposition. It takes two inputs, X and Y , and produces outputs P and Q with low quantum cost. Its reversible nature allows computation in both forward and reverse directions as shown in the Figure 1, unlike irreversible gates that lose information during switching [13]. The output relation is such that when $X=0$, $Q=Y$, and when $X=1$, $Q=\neg Y$; equivalently, $Q=X \oplus Y$.

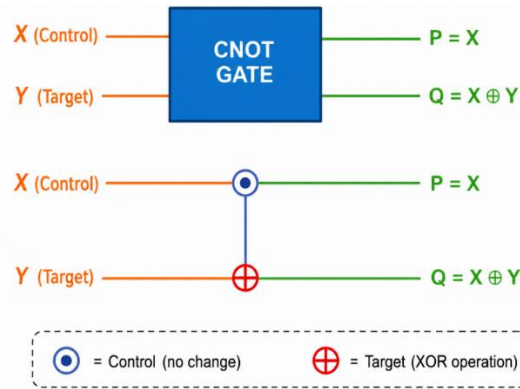


Figure 1. Reversible CNOT gate

3.1.2. Peres gate

The Peres gate is a 3×3 reversible logic gate with inputs X , Y , and Z and outputs P , Q , and R as shown in Figure 2. Its sum operation is based on XOR functionality, and it can support ripple carry addition when used in modified configurations. It has low quantum cost and reduces GO, thereby improving power efficiency [7]. The selection of Peres and CNOT gates as primary reversible elements is based on their low quantum cost and functional completeness. The CNOT gate (quantum cost 1) performs XOR operations required for partial-product accumulation, while the Peres gate (quantum cost 4) implements full-adder functionality within a single reversible block, making it an efficient 3×3 reversible adder. Other gates such as Toffoli and Fredkin (both quantum cost 5) were not used due to higher complexity and increased GO. This combination minimizes TRLIC while preserving bijective mapping, contributing to the low power consumption observed in synthesis results.

3.2. Vedic FIR filters fault tolerant multiplication using urdhva tiryakbhayam algorithm

The proposed FIR filter employs a 16×16 Vedic multiplier to perform multiplication between input samples and filter coefficients. The use of UT Vedic multiplication enables low-latency, low-power operation, making it suitable for real-time applications [14]. The architecture integrates fault-tolerant reversible logic, including adders and carry-save structures, to support FIR operations efficiently. The UT-based Vedic method performs vertical and horizontal multiplication to generate partial products in parallel, reducing computation delay compared to conventional approaches.

Unlike traditional FIR implementations that rely on sequential accumulation or methods such as MCM and signed-digit arithmetic, the proposed design reduces scheduling overhead by generating all partial products simultaneously. These are then accumulated using reversible adders to minimize information loss.

Although 16-tap FIR filters typically incur higher computation latency due to multiple accumulation stages, the parallel nature of the Vedic approach significantly reduces execution time. By integrating this method into a reversible, fault-tolerant architecture, the design improves processing speed while mitigating data-path delay and fan-out issues in conventional FIR implementations [15].

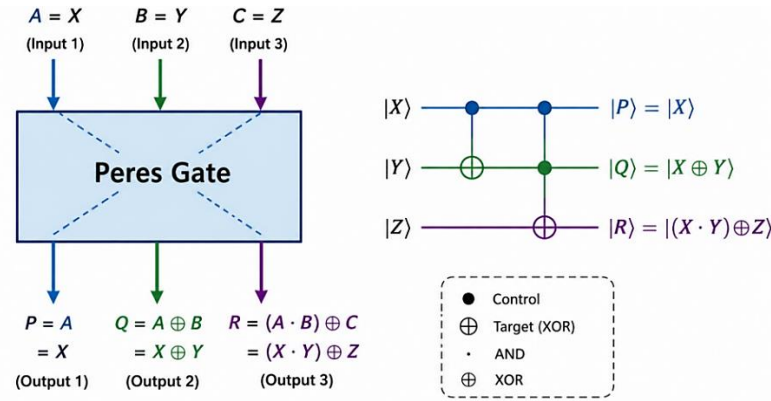


Figure 2. Logic diagram for Peres gate

3.3. Reversible look-ahead carry adder

To illustrate UT multiplication, consider a 3-digit multiplicand (326) and multiplier (738), producing a 32-bit output word after unsigned multiplication. In the first step, the least significant partial product is obtained by vertically multiplying the least significant digits of the operands. In the second step, remaining partial products are generated and added along with the carry from the previous step by multiplying corresponding higher-order digits. The resulting sum and carry form intermediate outputs that are propagated to the next stage. In the final step, all partial products are combined to obtain the complete product. Compared to array and Radix-2 multipliers, the UT-based approach reduces computational delay by generating partial products in parallel, making it more efficient for FIR filter implementations [16]. The Figure 3. Describes the Process for 4×4 bit scripture Vedic multiplier using UT algorithms.

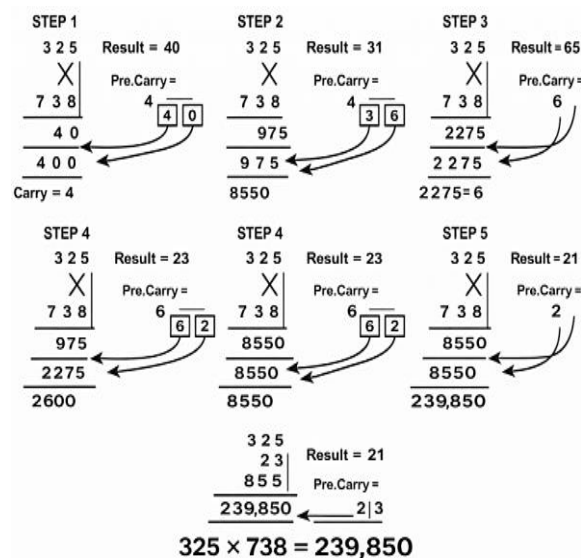


Figure 3. Process for 4×4 bit scripture Vedic multiplier using UT algorithms

Propagation delay is a major limitation in direct and transposed FIR filter architectures that use conventional adders and multipliers. In such designs, internal products are accumulated sequentially, increasing overall latency. In the proposed approach, reversible logic gates are used to perform parallel

accumulation of partial products, improving computational efficiency [16]. The architecture combines Controlled-NOT and Peres gates with a carry look-ahead adder structure, enabling faster execution compared to serial addition methods. Unlike ripple-carry adders, the carry look-ahead adder reduces delay by computing carry signals in parallel across all bits, thereby eliminating sequential carry propagation and improving overall processing speed.

$$\text{CarryPropagate}(P_i) = X_i \oplus Y_i \quad (1)$$

$$\text{CarryGenerate}(P_i) = (X_i \& Y_i) \oplus (X_i \oplus Y_i) \text{Cin} \quad (2)$$

A new expression for input carry for (n+1) stages as (3):

$$C_1 = G_0 + P_0 \text{Cin} \quad (3)$$

3.4. Error correction and codes

In an ECC scheme, n-k parity bits are added to a block of k information bits to form an n-bit encoded word [17]. These parity bits are generated using XOR combinations of the original data bits, enabling both error detection and correction. For example, a Hamming code maps four information bits into a seven-bit codeword by introducing three parity bits. In this case, data bits I₁, I₂, I₃, and I₄ are used to compute parity bits P₁, P₂, and P₃ using predefined XOR relationships [18].

$$P_1 = I_1 \oplus I_2 \oplus I_3 \quad (4)$$

$$P_2 = I_1 \oplus I_2 \oplus I_4 \quad (5)$$

$$P_3 = I_1 \oplus I_3 \oplus I_4 \quad (6)$$

If one of the bits is incorrect, the information and parity verify bits remain unchanged and can be retrieved at a next time. Recomposing the identity check bits and comparing the outcomes with the kept values achieves this. An error on d₁ will result in mistakes on all three parity checks in the example under consideration; a mistake on I₂ will only affect p₁ and p₂; a mistake in I₃ will affect p₁ and p₃; and a mistake on I₄ will affect p₂ and p₃. As a result, the incorrect data bit can be found and fixed. The creating (Generator matrix), which consists of identity and parity matrix G and parity check H matrices are frequently used to formulate this. For instance the Hamming code, which is:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Here, G is the generator matrix, which consists of unit matrix along with parity check bits.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Here, H is the combination of the transpose of the parity and identity matrix of the generator matrix. Calculating $y = x * G$ is used for encoding, while calculating $y * H$. Module two addition (XOR) and multiplication serve as the foundation for the operator * is used for error detection. The vector S, often called the syndrome, is used to locate the incorrect bit. Table 1 represents the values of S that correspond to the erroneous (Faulty) position. If a mistake is found in Table 1, it can be fixed by just flipping the bit. By creating a involve of verify filters Z_j, this ECC approach can be applied to the cascaded filters under consideration [19].

The new method is based on the ECC. It takes a block of K bits and adds n-k parity check bits to create a block of n bits. XOR combines the K data bits to generate the parity check bits. Errors can be found and fixed by carefully planning those combinations. Just three redundant filters are needed to achieve correction. The method can be applied to parallel filters that process various input signals while having the same response. High-level synthesis is used in the parallel filter's fault-tolerant error correcting code. To lessen the noise in the input data, use the Hamming code as per Table 1. The inputs to this propped Hamming

code filter architecture are x_1 , x_2 , x_3 , and x_4 , while the corrective outputs are yc_1 , yc_2 , yc_3 , and yc_4 . Verifies filters for 4 filters (y_1 , y_2 , y_3 , and y_4) and Hamming code.

Where, x_1 , x_2 , and x_3 is the binary input of FIR filter and z_1 is the output of redundant (Repetitive) module for finding the unnecessary data bit location [20].

Where, y_1 , y_2 , and y_3 is the output of original filter modules as shown in Figure 4 and Z_1 , Z_2 , and Z_3 is redundant code word output:

$$Z_1[n] = W_1[n] + W_2[n] + W_3[n] \quad (7)$$

$$Z_2[n] = W_1[n] + W_2[n] + W_4[n] \quad (8)$$

$$Z_3[n] = W_1[n] + W_3[n] + W_4[n] \quad (9)$$

For instance, an error in filter y_1 will result in faults in Z_1 , Z_2 , and Z_3 checks. Problems in a different group of Z_i will also result from issues on the other filters. Consequently, the error can be found and fixed, just like with conventional ECCs. Figure 4 illustrates Hamming code-based fault-tolerant FIR filter.

Table 1. Error location of hamming codes

Syndrome-1	Syndrome-2	Syndrome-3	Location of the faulty digit	Operation
0	0	0	No error	None
1	1	1	I_1	Better(Correct) I_1
1	1	0	I_2	I_2
1	0	1	I_3	I_3
0	1	1	I_4	I_4
1	0	0	I_1	P_1
0	1	0	I_2	P_2
0	0	1	I_3	P_3

Figure 4 shows the analysis of the error bits using (8)-(15). Figure 5 depicts the ASM charts delivered, showing how to detect and correct errors using syndromes and how redundant digits are used to recognize the faulty location and make the system soft-error-free. During Vedic FIR filter fault-tolerant which consists of X_1 , X_2 , and X_n is the input data, G is the generator matrix, which consists of a unit matrix, and the output of the FIR filter.

$$S = Y \cdot H^T \quad (10)$$

Here, Y is the represent the position of error while transmission so:

$$Y = X^E \quad (11)$$

Substitute Y in the S :

$$S = [X^E]H^T \quad (12)$$

(Therefore $XH^T = 0$):

$$S = EH^T \quad (13)$$

Where S is the single error bit detect.

Where, S is the syndrome detector, if $S=000$ no errors then $S!=000$ then errors to be find the fault location as per the 7 to 9 equas and final syndromes were detect and correct in FIR filter position. Here $yc_1=Z_1$, $yc_2=Z_2$, and $yc_3=Z_3$.

Filter correction is performed by using the remaining data to recreate the incorrect outputs and verify them. For instance, if an error is found on y_1 , it can be fixed by producing.

$$y_{c1[n]} = z_1[n] - y_2[n] - y_3[n] \quad (14)$$

Equans 10 tells about the rectifies output yc_1 bits on $yc_1[n]$.

Errors in the remaining data outputs can be corrected using the same equations. In this situation,

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}$$

Here in the hamming matrix reflected the minus one in identity matrix instsed of 1, so it is not error in the negative 1's then syndromes $S=000$.

To find mistakes, define the check shape (Matrix) and compute $S=y H^T$. The filter in error is then also detected using the vector syndrome. In this instance, a nonzero value in vector s equals 1 in conventional Hamming coding [21]. If $Z1!=W1$, we find the faulty point where it happened until we get 0. The comparisons may yield a few variations by cause new and verify filter design have distinct finite-precision effects. Quantifying effects will determine those variations. As a result, the threshold must be used for comparisons, with values below it categorized as 0. This confirmation that minor mistakes might not be fixed. Most of the time, this won't be a problem because little mistakes are acceptable.

The proposed formulation shows that the scheme can be extended to any linear block code and any number of parallel filters. As the number of filters (k) increases, the efficiency of the method improves; for example, only four redundant filters are required to correct a single fault when $k=11$. The encoding and decoding operations introduce negligible impact on the overall complexity of the 16-tap FIR filter, as they involve only basic arithmetic operations such as addition, subtraction, and comparison.

However, error propagation within the encoding and decoding logic must be considered. Since multiple parity outputs (z_i) may share intermediate computations, a fault in a shared adder can affect multiple outputs (e.g., z_1 and z_2), leading to incorrect correction decisions. To avoid this, each z_i should be computed independently to eliminate logic sharing, ensuring that errors affect only the corresponding parity outputs while the main data outputs remain unaffected. Similarly, a fault in syndrome computation will impact only a single bit if independent logic allocation is maintained [22].

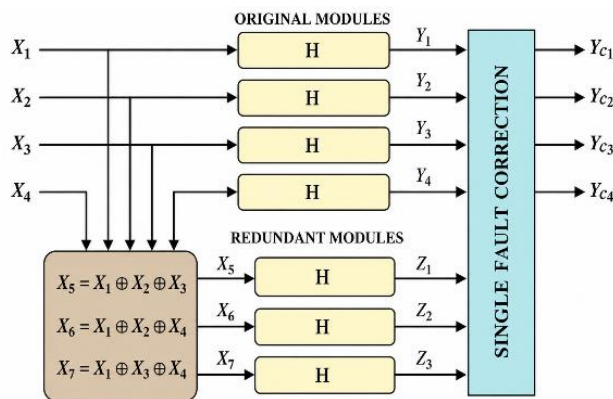


Figure 4. Hamming code-based fault-tolerant FIR filter

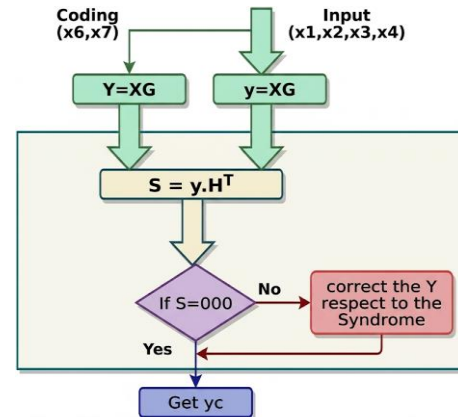


Figure 5. ASM for the FIR filter tolerant

4. RESULTS AND DISCUSSION

4.1. Register transfer logic of vedic fault tolerant fir filter

Figure 6 shows the register transfer level (RTL) schematic of the 16-tap FIR architecture implemented using coefficient-based filter stages. The fault-tolerant synthesis is optimized at the gate level to reduce netlist complexity and improve reliability through streamlined internal logic. The design minimizes unnecessary logic utilization by optimizing control signals and reducing redundant operations. Although the architecture introduces redundant modules for fault detection and correction, these are selectively used to mitigate soft errors and improve system reliability while maintaining efficient hardware utilization.

Figure 7 shows the simulation results for a reversible Vedic multiplier-based fault-tolerant FIR filter. X_1 , X_2 , X_3 , and X_4 are the 8-bit inputs of the 4-parallel FR filters. OUT_1 , OUT_2 , OUT_3 , and OUT_4 are the corrected filter outputs obtained for each input sample.

As observed from the synthesis results, the Vedic reversible FIR filter reports a propagation delay of 30.932 ns and a power consumption of 2.645 mW, representing a 5.57% latency improvement and a 1.42% power reduction relative to the MCM-FIR baseline (Table 2). However, the look-up tables (LUTs) count increases from 1,678 (MCM) to 2,096 (Vedic), and the slice count rises from 1,120 to 1,307. A critical analysis of these increases is provided below.

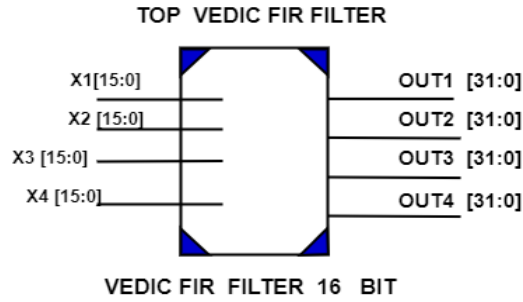


Figure 6. RTL representation for Vedic FIR filter

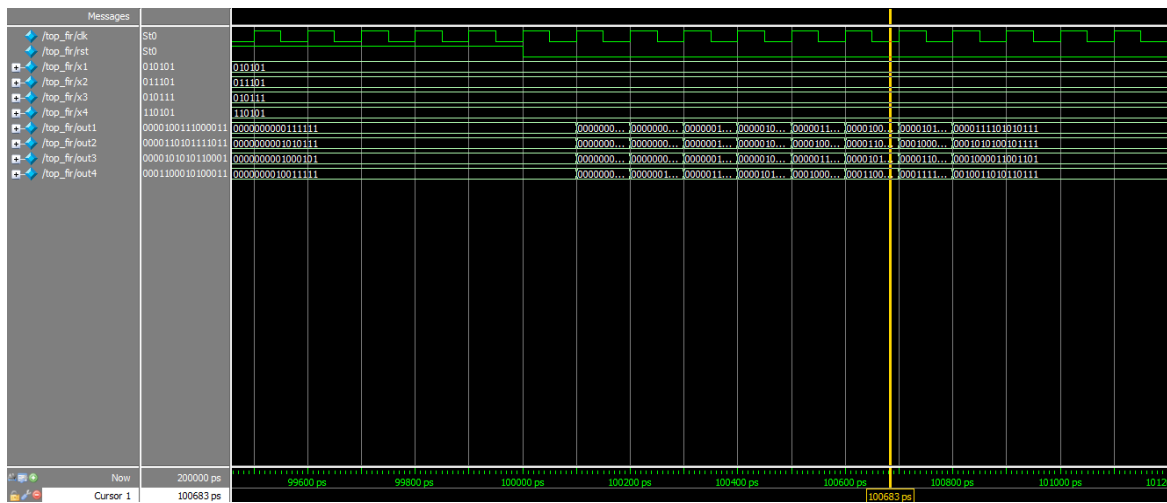


Figure 7. Simulation results of reversible Vedic multiplier based ECC FIR

Table 2. Comparison results of MCM and VEDIC FIR filters

Parameters	Fir_MCM	Fir_Vedic
Delay (ns)	32.759	30.932
Flipflops	384	384
LUT	1678	2096
Slices	1120	1307
Power (W)	2.683	2.645

The increase in LUT and slice utilization is attributed to three main architectural factors. First, reversible logic gates such as Peres and CNOT, used in the UT multiplier, require an $n \times n$ bijective mapping to preserve reversibility, which introduces CI and GO that must be implemented in hardware, thereby increasing LUT and routing usage. Second, the Hamming-code-based ECC module adds dedicated parity generation and syndrome detection logic across all parallel outputs, including XOR-based encoder trees and correction circuitry, which further contributes to LUT overhead. Third, the parallel filter bank replicates the FIR datapath across four channels (X1–X4) to enable concurrent processing, significantly increasing the combinational logic compared to a single-channel MCM-FIR architecture. Figure 8 describes the FPGA synthesis hardware resource.

From a design-efficiency perspective, the increase in LUTs and slices is a deliberate architectural trade-off rather than a limitation. The proposed design achieves simultaneous reductions in propagation delay and power consumption the two most critical metrics for real-time FPGA-based DSP systems while introducing fault tolerance not present in the MCM baseline. The power–area efficiency remains favorable, as a 1.42% reduction in power is achieved at the cost of a 24.9% increase in LUT usage, justified by the added capability of ECC-based single-bit error correction across parallel outputs. In safety-critical applications such as biomedical processing and CNN inference, real-time error detection and correction is more valuable than marginal area savings. Moreover, the Xilinx Virtex-5 FPGA provides sufficient logic

resources, and the overall utilization remains within device limits, confirming that the area overhead does not pose a practical constraint.

The proposed Vedic FIR fault-tolerant architecture is integrated into CNNs at the hardware kernel level rather than at the software or algorithmic level. In standard CNN accelerators, the MAC unit serves as the core computational block for convolution operations between input feature maps and kernel weights. In this work, the conventional MAC unit is replaced by a UT-based reversible multiplier implemented using Peres and CNOT gates. This enables kernel-weight multiplication with zero information loss, reduced switching power, and embedded ECC-based soft-error correction on accumulated outputs.

As shown in Figure 9 and described in (15), each convolution operation involving input activations and weight tensors is executed using the proposed fault-tolerant Vedic multiplier block. This hardware-level replacement enables real-time detection and correction of soft errors commonly observed in high-frequency CNN computations, particularly in radiation-prone or thermally stressed environments. The proposed architecture thus serves as a drop-in replacement for conventional MAC units in FPGA-based CNN accelerators without requiring any modification to CNN weights, network topology, or the software inference pipeline. To demonstrate the practical relevance of the proposed architecture for machine-learning workloads, an edge-AI scenario is considered where a 1-D CNN is deployed on an FPGA for real-time ECG arrhythmia classification. In this system, each convolutional layer performs repeated MAC operations on 16-bit inputs, and the proposed Vedic FIR-based fault-tolerant multiplier replaces the conventional MAC unit. With a propagation delay of 30.932 ns per multiplication, corresponding to an approximate throughput of 32.3 million operations per second, the design satisfies the real-time requirements of ECG benchmarks operating at 250–500 Hz with 256–512 sample windows. In addition, the integrated Hamming ECC mitigates single-bit soft errors, which have been reported to cause up to 15% misclassification in unprotected CNN accelerators under fault conditions, by correcting them transparently during inference. This makes the proposed architecture suitable as a reliable compute primitive for safety-critical edge-AI applications such as wearable health monitoring, autonomous sensing, and radiation-prone aerospace inference systems. The (15) tells about the weighted input and input layer.

$$Z = \sum_{i=1}^n W_i X_i + b \quad (15)$$

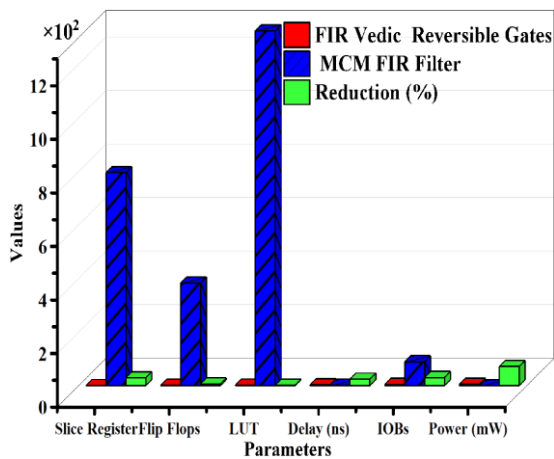


Figure 8. FPGA synthesis hardware resource

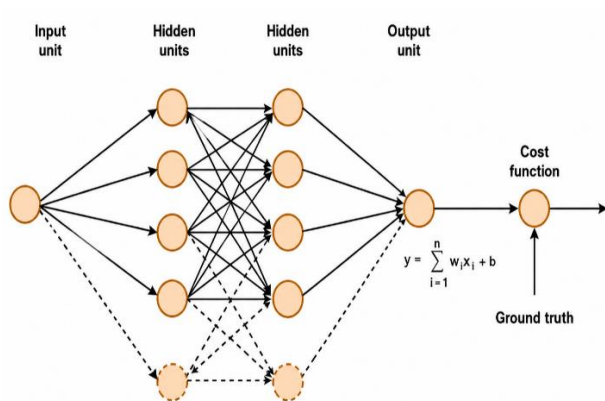


Figure 9. Vedic fault tolerant integrated into CNN

Table 3 presents a comparative analysis of the proposed architecture against six state-of-the-art FPGA-based Vedic and FIR filter implementations. Mittal *et al.* [21] implemented a design on a Spartan-E FPGA with 2,501 LUTs, 20.12 ns delay, and 7.63 mW power, showing lower delay but significantly higher power consumption than the proposed method. Farrukh *et al.* [23] reported 1,200 LUTs on a Zynq-706 platform, but with higher delay (55.34 ns) and power (8.63 mW), indicating lower speed efficiency. Zhou *et al.* [24] achieved 1,000 area units and 10.21 ns delay in a 32-bit design, but with 7.87 mW power and no fault-tolerance capability. Gao *et al.* [8] proposed a fault-tolerant ECC-based parallel filter on a Virtex-7 FPGA with low delay (3.96 ns) and compact area (299 units), but the 64-bit architecture limits direct comparison with the 16-bit proposed design. Sumalatha *et al.* [25] and Sridevi and Sathiyaraj [26], both implemented on Virtex-5 platforms, report lower area usage (397 and 421 units respectively), but without

reversible logic or ECC-based fault tolerance. Importantly, none of the surveyed architectures in Table 3 integrate reversible Vedic multiplication, quantum-inspired gate logic, and Hamming ECC within a single FPGA-mapped FIR filter. The higher LUT count of the proposed design (2,096) is therefore a direct consequence of this combined functionality, enabling zero-information-loss computation and single-bit soft-error correction, features not simultaneously present in any competing work.

Table 3. An outline of the FPGA implementation as it stands currently [8], [21], [23]–[26]

Authors and years	FPGA details	Bit-width	Area	Delay (ns)	Power (mW)
Mittal <i>et al.</i> [21], 2017	Spartan E XC3S500E	16-Taps	2501	20.12	7.63
Farrukh <i>et al.</i> [23], 2020	Zynq-706 FPGA	16	1200	55.34	8.63
Zhou <i>et al.</i> [24], 2021	Spartan 6 XC7VX990T	32	1000	10.21	7.87
Gao <i>et al.</i> [8], 2015	FPGA Virtex-7	64	299	3.96	5.05
Sumalatha <i>et al.</i> [25], 2019	FPGA Virtex-5	32	397	4.12	4.01
Sridevi and Sathiya [26], 2025	FPGA Virtex-5	16	421	4.67	34.74
Proposed investigation	FPGA Virtex-5	16	2096	30.932	2.645

Regarding hardware validation, this work performs design-level verification through FPGA synthesis and post-synthesis static timing analysis on the Xilinx Virtex-5 platform using the Xilinx ISE Design Suite. FPGA synthesis is a widely accepted hardware-level validation approach, as the synthesized netlist is mapped to configurable logic blocks (CLBs), LUTs, flip-flops, and routing resources, with timing and power metrics extracted from post-synthesis reports rather than purely behavioral simulation. The RTL schematic (Figure 6) and FPGA resource utilization report (Figure 8) confirm correct structural mapping of the reversible FIR filter datapath onto the Virtex-5 architecture. However, hardware-in-the-loop (HIL) testing, including board-level deployment, oscilloscope-based timing verification, and real-signal fault injection, was not performed due to equipment constraints. This remains a limitation, and physical validation using platforms such as Xilinx Nexys/ZedBoard is planned as future work, as stated in the conclusion.

5. CONCLUSION

This work presents a fault-tolerant and power-efficient Vedic FIR filter architecture based on reversible logic computing. The integration of UT multiplication with a reversible LACA eliminates information loss during computation, while a Hamming-code-based ECC improves reliability by detecting and correcting single-bit faults in parallel outputs. FPGA implementation on a Xilinx Virtex-5 device achieves a power consumption of 2.645 mW with a 5.57% reduction in delay compared to conventional MCM-based FIR filters.

Despite these advantages, certain limitations remain. The design has been validated only on the Xilinx Virtex-5 FPGA, and performance metrics such as LUT usage, area, and timing may vary on newer FPGA families or ASIC implementations due to technology-dependent optimizations. The ECC scheme provides only single-bit error correction and double-bit error detection (SECDDED), making it unsuitable for multi-bit upsets in radiation-prone environments, where stronger codes such as BCH or Reed–Solomon would be required with additional hardware overhead. In addition, the architecture is fixed at a 16-tap, 16-bit configuration, and its scalability to higher orders and wider word lengths has not been evaluated. Furthermore, while the design targets CNN accelerator integration at the MAC level, full system-level evaluation in terms of throughput, energy per inference, and accuracy has not been performed.

Future work will focus on extending the design to higher-order FIR filters and wider precision, exploring ASIC implementations for low-power internet of thing and wearable applications, integrating multi-bit ECC schemes for radiation-hardened systems, and performing full system-level benchmarking in CNN accelerators under real workloads. Formal verification of the reversible logic design will also be pursued to ensure correctness under all fault scenarios.

FUNDING INFORMATION

The authors declare that no funding was received for this work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Guguloth Erna	✓	✓	✓		✓	✓			✓	✓			✓	
Vadtya Saidulu		✓				✓		✓		✓		✓		
Chinta Manjusha	✓		✓	✓			✓			✓	✓		✓	
GVK Murthy					✓	✓		✓		✓	✓			
Vijaya Bhaskar	✓		✓	✓	✓			✓	✓	✓			✓	
Chalampalem									✓	✓				
Puthalapattu Muni Sekhar	✓				✓		✓		✓		✓			

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

- [1] C. Savalam and V. N. T. Alapati, "Soft error detection and correction for parallel digital filters using Hamming code," *International Journal of System Assurance Engineering and Management*, Feb. 2023, doi: 10.1007/s13198-023-01876-6.
- [2] V. B. Chalampalem and M. Pidugu, "An efficient high performance reconfigurable canonical sign digit architecture for software defined radio," *International Journal of Reconfigurable and Embedded Systems*, vol. 13, no. 1, pp. 25–32, Mar. 2024, doi: 10.11591/ijres.v13.i1.pp25-32.
- [3] A. Hazarika, N. Choudhury, and S. Poddar, "Approximate Vedic Multiplier Architecture for Efficient CNN Acceleration on Embedded Devices," in *Proceedings - 2024 IEEE 48th Annual Computers, Software, and Applications Conference, COMPSAC 2024*, 2024, pp. 473–482. doi: 10.1109/COMPSAC61105.2024.00071.
- [4] S. Y. Park and P. K. Meher, "Low-power, high-throughput, and low-area adaptive fir filter based on distributed arithmetic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 6, pp. 346–350, 2013, doi: 10.1109/TCSII.2013.2251968.
- [5] N. M. Priya, C. B. Thangammal, R. Seshasayanan, and S. Radley, "High performance fir filter based on vedic mathematics," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 3, pp. 829–835, Apr. 2023, doi: 10.1007/s13198-023-01899-z.
- [6] N. Mageshwari, P. Sakthivel, and R. Seetharaman, "Design and synthesis of reversible Vedic multiplier using cadence 180 nm technology for low-power high-speed applications," *Scientific Reports*, vol. 15, no. 1, May 2025, doi: 10.1038/s41598-025-04002-7.
- [7] S. F. Naz and A. P. Shah, "Reversible Gates: A Paradigm Shift in Computing," *IEEE Open Journal of Circuits and Systems*, vol. 4, pp. 241–257, 2023, doi: 10.1109/OJCS.2023.3305557.
- [8] Z. Gao *et al.*, "Fault tolerant parallel filters based on error correction codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp. 384–387, Feb. 2015, doi: 10.1109/TVLSI.2014.2308322.
- [9] I. Hatai, I. Chakrabarti, and S. 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, pp. 1071–1080, 2015, doi: 10.1109/TCSI.2015.2388838.
- [10] G. K. Kumar, N. R. Chinnapurapu, and K. Srinivas, "Power and area efficient FIR filter using Radix- 2r multiplier for de-noise the electrooculography (EOG) signal," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-73514-5.
- [11] Y. Hu and C. Wang, "FTI-TMR: A fault tolerance and isolation algorithm for interconnected multicore systems," *Engineering*, vol. 7, pp. 1–23, 2026, doi: 10.3390/eng7080389.
- [12] C. Kalamani, R. Murugasami, S. Usha, and S. Saravanakumar, "Design of encoder and decoder using reversible logic gates," *Measurement: Sensors*, vol. 31, pp. 1–12, Feb. 2024, doi: 10.1016/j.measen.2023.100989.
- [13] K. Inayat, F. Bin Muslim, J. Iqbal, S. A. H. Mohsan, H. K. Alkahtani, and S. M. Mostafa, "Power-Intent Systolic Array Using Modified Parallel Multiplier for Machine Learning Acceleration," *Sensors*, vol. 23, no. 9, p. 4297, Apr. 2023, doi: 10.3390/s23094297.
- [14] Z. Aizaz and K. Khare, "Energy efficient approximate booth multipliers using compact error compensation circuit for mitigation of truncation error," *International Journal of Circuit Theory and Applications*, vol. 50, no. 6, pp. 2252–2270, Feb. 2022, doi: 10.1002/cta.3252.
- [15] B. Y. Galadima, G. S. M. Galadanci, A. Tijjani, and M. Ibrahim, "A review on reversible logic gates," *Bayero Journal of Pure and Applied Sciences*, vol. 12, no. 1, pp. 242–250, Apr. 2020, doi: 10.4314/bajopas.v12i1.38s.
- [16] M. Swathi and B. Rudra, "Implementation of Reversible Logic Gates with Quantum Gates," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, Jan. 2021, pp. 1557–1563. doi: 10.1109/CCWC51732.2021.9376060.
- [17] N. De Carvalho, L. Nogueira, M. Anderson, D. Freitas, J. Da Silveira, and C. Marcon, "Improving Error Correction Efficacy by

- Applying Nesting Encoding Approach,” in *2025 38th SBC/SBMicro/IEEE Symposium on Integrated Circuits and Systems Design (SBCCI)*, Manaus, Brazil, 2025, pp. 1-5, doi: 10.1109/SBCCI66862.2025.11218665.
- [18] V. Priyanka, N. S. Reddy, G. Jeevana, and M. A. Arab, “Design of Arithmetic Logic Unit using Reversible Logic Gates,” in *2024 2nd World Conference on Communication and Computing, WCONF 2024*, 2024, pp. 1–6. doi: 10.1109/WCONF61366.2024.10692088.
- [19] J. Zhang, Z. Chen, M. Ma, R. Jiang, H. Li, and W. Wang, “High-Performance ECC Scalar Multiplication Architecture Based on Comb Method and Low-Latency Window Recoding Algorithm,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 32, no. 2, pp. 382–395, Feb. 2024, doi: 10.1109/TVLSI.2023.3321772.
- [20] D. Vaithiyanathan, B. P. James, and K. Mariammal, “Comparative Study of Single MAC FIR Filter Architectures with Different Multiplication Techniques,” in *2023 2nd International Conference on Electrical, Electronics, Information and Communication Technologies, ICEEICT 2023*, Apr. 2023, pp. 01–10. doi: 10.1109/ICEEICT56924.2023.10157620.
- [21] A. Mittal, A. Nandi, and D. Yadav, “Comparative study of 16-order FIR filter design using different multiplication techniques,” *IET Circuits, Devices and Systems*, vol. 11, no. 3, pp. 196–200, Mar. 2017, doi: 10.1049/iet-cds.2016.0146.
- [22] J. Chen, C. H. Chang, J. Ding, R. Qiao, and M. Faust, “Tap Delay-and-Accumulate Cost Aware Coefficient Synthesis Algorithm for the Design of Area-Power Efficient FIR Filters,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 712–722, Feb. 2018, doi: 10.1109/TCSI.2017.2725916.
- [23] F. U. D. Farrukh *et al.*, “Power efficient tiny yolo cnn using reduced hardware resources based on booth multiplier and wallace tree adders,” *IEEE Open Journal of Circuits and Systems*, vol. 1, pp. 76–87, 2020, doi: 10.1109/OJCS.2020.3007334.
- [24] B. Zhou, G. Wang, G. Jie, Q. Liu, and Z. Wang, “A High-Speed Floating-Point Multiply-Accumulator Based on FPGAs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 10, pp. 1782–1789, Oct. 2021, doi: 10.1109/TVLSI.2021.3105268.
- [25] M. Sumalatha, P. V. Naganjaneyulu, and K. S. Prasad, “Low power and low area VLSI implementation of vedic design FIR filter for ECG signal de-noising,” *Microprocessors and Microsystems*, vol. 71, p. 102883, Nov. 2019, doi: 10.1016/j.micpro.2019.102883.
- [26] A. Sridevi and A. Sathiya, “Integration in CNN and FIR filters for improved computational efficiency in signal processing,” *Ain Shams Engineering Journal*, vol. 16, no. 1, p. 103201, Jan. 2025, doi: 10.1016/j.asej.2024.103201.

BIOGRAPHIES OF AUTHORS



Dr. Guguloth Erna    is with the Department of Electronics and Communication Engineering at PACE Institute of Technology and Sciences (Autonomous), Ongole, Andhra Pradesh, India. He is involved in teaching and research in the areas of digital signal processing, VLSI design, and fault-tolerant hardware systems. He can be contacted at email: ernamist@gmail.com.



Vadtya Saidulu    is with the Department of Electronics and Communication Engineering at Mahatma Gandhi Institute of Technology, JNTUH, University, Hyderabad in the areas of antenna and wave propagation, VLSI design, and fault-tolerant hardware systems. He can be contacted at email: vsaidulu_ece@mgit.ac.in.



Chinta Manjusha    is with the Department of Electronics and Communication Engineering at Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India. She is involved in teaching and research in the areas of digital signal processing, computer network and image processing in VLSI design, and fault-tolerant hardware systems. She can be contacted at email: cmanjusha@kluniversity.in.



Dr. GVK Murthy    is currently working as a principal in the Department of Electrical and Electronics Engineering at PACE Institute of Technology and Sciences (Autonomous), Ongole, Andhra Pradesh, India. He can be contacted at email: murthy260@gmail.com.



Dr. Vijaya Bhaskar Chalampalem    is currently in the Department of Electronics and Communication Engineering, Siddharth Institute of Engineering and Technology, Puttur, India. Currently his research interest is developing low power and area efficient high performance architectures for future wireless communications. He can be contacted at email: vijayas4u@gmail.com.



Dr. Puthalapattu Muni Sekhar    is currently in the Department of Electrical and Electronics Engineering, Nodal Officer, Siddharth Institute of Engineering and Technology, Puttur, India. He can be contacted at email: manamuni13@gmail.com.