Power System Protection and Control
ISSN:1674-3415

GENERIC 5G NR LDPC ENCODER ARCHITECTURE OPTIMIZED FOR AREA AND THROUGHPUT

Yogesh Rethinapandian*, Arun Karthik Sundararajan†, Kaushik Kumar‡, Sanjanaa Sridhar§, and Smrithi Prakash¶

*University of Illinois at Chicago, Illinois, USA
yrethi2@uic.edu (Corresponding Author)
†IEEE Member, USA
arunkarthik@ieee.org
‡University of Arizona, Tucson, AZ, USA
kaushikkumar@arizona.edu
§University of Illinois at Urbana-Champaign, Urbana, IL, USA
ssridh41@illinois.edu
¶SRM Institute of Science and Technology, Chennai, India
sp9114@srmist.edu.in

Received: 22/02/2025 **Revised**: 19/03/2026 **Accepted**: 15/04/2026

***ABSTRACT*:**

The deployment of fifth-generation wireless communication systems demands efficient error correction mechanisms capable of supporting diverse data rates and reliability requirements. Low-Density Parity-Check codes have emerged as the primary forward error correction technique for 5G New Radio data channels due to their superior performance and flexible parallelization capabilities. However, existing LDPC encoder implementations face significant challenges balancing hardware complexity, silicon area consumption, and throughput requirements across the wide range of code rates and block lengths specified in 5G standards. This paper presents a novel generic LDPC encoder architecture specifically optimized for both area efficiency and high throughput operation. Our design employs an innovative shift register network combined with optimized memory management strategies that reduce redundant computations while maintaining full compatibility with all 5G NR LDPC base graphs. The proposed architecture achieves 35% area reduction compared to conventional parallel implementations while delivering throughput exceeding 10 Gbps for typical code configurations. Synthesis results using 28nm CMOS technology demonstrate that our encoder consumes only 0.82 mm² silicon area while operating at 500 MHz clock frequency. The architecture supports all code rates from 1/3 to 8/9 and information block lengths from 40 to 8448 bits as specified in 3GPP Release 15 standards.

***Keywords**: 5G NR, LDPC encoder, hardware architecture, area optimization, throughput, error correction coding, wireless communication, base graph*

INTRODUCTION

The evolution toward fifth-generation wireless networks has introduced unprecedented demands for data transmission capabilities, requiring systems that simultaneously support enhanced mobile broadband services, ultra-reliable low-latency communications, and massive machine-type connectivity [Richardson and Urbanke 2008]. These diverse application scenarios necessitate flexible error correction mechanisms capable of adapting to varying channel conditions and quality-of-service requirements while maintaining computational efficiency.

Low-Density Parity-Check codes were selected as the channel coding scheme for 5G New Radio data channels after extensive evaluation against alternative techniques including Turbo codes and Polar codes [Jiang et al. 2016]. The decision reflected LDPC codes' excellent performance approaching Shannon limit, inherent parallelism enabling high-throughput implementations, and flexibility supporting multiple code rates without significant complexity increases. The 3GPP standardization process defined two base graphs supporting different block lengths and code rates optimized for diverse communication scenarios.

Despite LDPC codes' theoretical advantages, practical hardware implementation presents substantial challenges. The encoding process requires matrix-vector multiplication operations involving large sparse parity-check matrices, resulting in complex computational structures [Chen et al. 2018]. Traditional implementations using full

DOI: 10.46121/pspc.54.2.05

matrix representations consume excessive memory resources, making them impractical for resource-constrained mobile devices. Parallel architectures achieving high throughput often require substantial silicon area, increasing manufacturing costs and power consumption.

Existing LDPC encoder designs typically fall into two categories: serial implementations minimizing hardware resources but delivering limited throughput, and parallel architectures achieving high processing speeds at the cost of significant area overhead [Myung and Yang 2011]. Neither approach satisfactorily addresses 5G requirements for devices ranging from low-cost IoT sensors to high-performance base stations, which demand scalable solutions balancing area and throughput across diverse operating points.

This research addresses these challenges by developing a generic LDPC encoder architecture specifically optimized for 5G NR specifications. Our design exploits the quasi-cyclic structure inherent in 5G LDPC base graphs, implementing efficient shift register networks that eliminate redundant storage while enabling configurable parallelism [Dizdar and Arikan 2019]. The architecture incorporates adaptive memory management techniques that dynamically allocate resources based on instantaneous code parameters, minimizing area consumption without compromising throughput.

The contributions of this work include: a novel shift register network architecture reducing memory requirements by 40% compared to conventional designs; an optimized scheduling algorithm minimizing processing cycles for all supported code configurations; a flexible parallelization framework enabling throughput scaling from 1 Gbps to 15 Gbps through adjustable processing elements; and comprehensive synthesis results demonstrating superior area-throughput tradeoffs across multiple technology nodes.

LITERATURE REVIEW

2.1 LDPC Codes in 5G Standards

The standardization of LDPC codes for 5G New Radio involved extensive research comparing various error correction techniques across performance, complexity, and flexibility dimensions. The 3GPP technical specification 38.212 defines two base graphs designated BG1 and BG2, each optimized for different operational regimes [3GPP TS 38.212 2018]. BG1 supports larger information block sizes and higher code rates suitable for enhanced mobile broadband applications, while BG2 addresses smaller blocks and lower rates appropriate for ultra-reliable communications.

Both base graphs exhibit quasi-cyclic structure, meaning the parity-check matrices can be represented as arrays of circulant permutation matrices or zero matrices [Richardson and Urbanke 2008]. This structure significantly simplifies hardware implementation compared to random LDPC codes, enabling efficient circular shift operations rather than arbitrary permutations. The base matrices contain shift values ranging from -1 (indicating zero matrices) to values determining circular shift amounts for identity matrix expansions.

The expansion factor Z, also called lifting size, determines the dimension of each circulant submatrix and can take values from a specified set between 2 and 384 [Jiang et al. 2016]. This variable expansion enables fine-grained code rate adaptation and supports diverse block lengths through a unified base graph structure. However, variable Z values complicate hardware design because efficient architectures must accommodate all possible expansion factors without excessive resource duplication.

2.2 Conventional LDPC Encoder Architectures

Early LDPC encoder implementations employed straightforward matrix-vector multiplication approaches directly translating the encoding equation into hardware [Myung and Yang 2011]. These designs stored complete parity-check matrices and performed systematic encoding through Gaussian elimination or approximate lower triangulation methods. While conceptually simple, such approaches required substantial memory resources and delivered limited throughput due to sequential processing constraints.

Researchers developed improved architectures exploiting quasi-cyclic structure through circular shift networks replacing full matrix storage [Chen et al. 2018]. These designs utilized barrel shifters or multiplexer networks implementing Z-bit circular shifts corresponding to base matrix entries. Parallel processing elements operated on multiple matrix rows simultaneously, increasing throughput proportionally to parallelization degree. However,

DOI: 10.46121/pspc.54.2.05

high parallelism architectures replicated shift networks and accumulation units extensively, resulting in significant area overhead.

Partially parallel architectures emerged as compromise solutions attempting to balance area and throughput [Zhang and Parhi 2016]. These designs time-multiplexed processing resources across matrix rows, processing subsets in parallel while completing full encoding over multiple clock cycles. Resource sharing reduced area compared to fully parallel implementations, though at throughput cost. The challenge involved determining optimal parallelization degrees and scheduling algorithms minimizing latency while constraining hardware complexity.

2.3 Memory Organization Strategies

Memory architecture significantly impacts LDPC encoder area and performance characteristics. Storing complete lifted matrices requires Z^2 memory bits per non-zero base matrix entry, quickly becoming prohibitive for large expansion factors [Keskinoz and Sørensen 2020]. Exploiting circulant structure reduces storage to Z bits per entry plus shift value metadata, achieving substantial compression.

Double-buffering techniques enable overlapped encoding operations, where the encoder processes current blocks while loading or storing subsequent blocks [Li et al. 2017]. This pipelining approach maintains high utilization of computational resources, preventing idle cycles during memory transfers. However, double-buffering doubles memory requirements, creating area-throughput tradeoffs that must be carefully evaluated.

Distributed memory architectures partition storage across processing elements rather than centralizing in monolithic banks [Urard et al. 2015]. This distribution reduces interconnect complexity and enables higher clock frequencies by shortening critical paths. Each processing element accesses local memory, with coordination mechanisms ensuring correct data flow between elements. The challenge involves partitioning strategies that minimize inter-element communication while maintaining load balance.

2.4 Shift Network Implementations

Circular shift operations form the computational core of quasi-cyclic LDPC encoders, making shift network efficiency critical for overall performance. Barrel shifter implementations provide single-cycle arbitrary shifts but require significant combinational logic, particularly for large Z values [Shao et al. 2019]. The barrel shifter area grows approximately as $Z \log_2(Z)$, becoming prohibitive for maximum 5G expansion factors.

Multiplexer-based shift networks offer alternative implementations with different area-speed tradeoffs [Chen et al. 2018]. Cascaded multiplexer stages select appropriate input positions based on shift amounts, implementing shifts through routing rather than permutation. While potentially more area-efficient than barrel shifters for large Z, multiplexer networks may introduce longer critical paths affecting maximum clock frequency.

Serial shift register approaches minimize area by performing multi-bit shifts through sequential single-bit operations [Dizdar and Arikan 2019]. These designs require multiple clock cycles for shifts but eliminate complex combinational networks. For applications where encoding latency tolerates multi-cycle operations, serial shifters provide attractive area savings. Hybrid approaches combining parallel and serial shifting mechanisms enable flexible tradeoffs across operating points.

2.5 Encoding Algorithm Optimization

The encoding algorithm significantly influences implementation complexity and performance. Richardson-Urbanke encoding employs approximate lower triangulation of parity-check matrices, enabling efficient systematic encoding [Richardson and Urbanke 2008]. However, the triangulation process introduces density in previously sparse matrix regions, potentially increasing computational requirements.

Alternative approaches based on generator matrix formulation avoid triangulation complexity but may require storing dense generator matrices [Zhang and Parhi 2016]. For quasi-cyclic codes, generator matrices can be derived offline and compressed through similar circulant structure exploitation. The tradeoff involves preprocessing complexity versus runtime efficiency.

Incremental encoding techniques reuse computations across related code configurations, valuable for 5G scenarios where code parameters change dynamically [Jiang et al. 2016]. When transitioning between code rates or block

DOI: 10.46121/pspc.54.2.05

lengths with shared base graph structure, incremental methods update previous results rather than recomputing entirely. This approach reduces average encoding latency across configuration sequences.

2.6 Hardware-Software Codesign Approaches

Some research explores hybrid hardware-software implementations combining dedicated accelerators with programmable processors [Li et al. 2017]. These designs implement critical encoding operations in specialized hardware while managing configuration, scheduling, and control through software. Flexibility advantages enable supporting emerging standards through firmware updates rather than hardware redesigns.

However, hardware-software partitioning introduces interface overheads and potential bottlenecks where data transfers between domains [Keskinoz and Sørensen 2020]. For high-throughput applications, pure hardware implementations often deliver superior performance despite reduced flexibility. The optimal approach depends on target application requirements and anticipated standard evolution timelines.

2.7 Research Gaps

Existing LDPC encoder architectures exhibit several limitations motivating this research. Most published designs target specific code parameters or parallelization degrees, lacking flexibility across full 5G specification ranges. Area optimization typically focuses on individual architectural components rather than system-level resource sharing opportunities. Throughput scaling mechanisms often require duplicating entire processing pipelines rather than enabling finer-grained parallelism adjustments.

Furthermore, limited work addresses dynamic switching between base graphs and expansion factors at runtime. 5G systems require rapid reconfiguration supporting heterogeneous traffic with varying quality-of-service demands. Encoders must adapt efficiently without excessive reconfiguration overhead or resource underutilization during specific operating modes.

METHODOLOGY

3.1 Architectural Design Philosophy

Our encoder architecture development followed systematic design principles prioritizing resource sharing, computational reuse, and adaptive parallelization. The core philosophy recognizes that 5G LDPC encoding involves repetitive operations on structured data, creating opportunities for efficient hardware realization through careful architectural organization.

We began by analyzing 5G NR base graph structures to identify common patterns and variations across supported configurations. This analysis revealed that despite apparent diversity in code rates and block lengths, the underlying matrix operations share substantial computational structure. Both BG1 and BG2 employ similar row weights and column weights, suggesting unified processing architectures can serve both graphs efficiently.

The design process emphasized minimizing memory footprint while maintaining processing throughput through three key strategies: exploiting quasi-cyclic structure to compress matrix representation, implementing adaptive shift networks capable of handling all expansion factors without redundant hardware, and developing intelligent scheduling algorithms that maximize processing element utilization across diverse code configurations.

3.2 Shift Register Network Design

Our shift register network implements configurable circular shifts using a hybrid architecture combining parallel and serial mechanisms. Rather than employing full barrel shifters for maximum Z values, we partition the shift operation into coarse and fine components. Coarse shifts handle multiples of a base granularity using multiplexer networks, while fine shifts complete adjustments through short shift register chains.

This decomposition reduces combinational complexity significantly compared to monolithic barrel shifters. For maximum 5G expansion factor Z=384, our design achieves equivalent functionality using approximately 60% of the area required by conventional barrel shifter implementations. The area savings stem from recognizing that most shift amounts cluster around particular values, enabling optimization of common cases while still supporting full range.

DOI: 10.46121/pspc.54.2.05

The shift network interfaces with distributed memory banks organized to align with base graph structure. Each memory bank stores data corresponding to circulant submatrix positions, with addressing logic computing appropriate access patterns based on current shift values and processing schedule. This co-design of shift network and memory organization eliminates redundant data movement between components.

3.3 Adaptive Parallelization Framework

Traditional parallel LDPC encoders fix parallelization degree at design time, either processing all matrix rows simultaneously or establishing static partitioning. Our architecture implements dynamic parallelization where active processing element count adjusts based on current code parameters and throughput requirements.

The framework employs a configurable array of processing elements, each capable of computing partial parity sums for matrix row subsets. Runtime configuration signals activate appropriate element counts, with inactive elements power-gated to minimize energy consumption. This approach enables throughput scaling from approximately 2 Gbps using single processing element to over 12 Gbps with full parallelization.

Load balancing across processing elements follows a work-stealing algorithm where elements completing assigned tasks assume portions of work from busy elements. This dynamic redistribution maintains high utilization even for irregular base graph configurations exhibiting variable row weights. The balancing mechanism adds minimal control overhead while significantly improving average throughput across diverse code configurations.

3.4 Encoding Scheduling Algorithm

The encoding schedule determines the sequence of operations computing parity bits from information bits according to parity-check matrix structure. Our scheduling algorithm optimizes for minimal latency while respecting data dependencies inherent in the encoding process.

We formulated scheduling as a constrained optimization problem minimizing total clock cycles subject to dependency constraints and resource limitations. The solution employs graph-theoretic techniques modeling dependencies as directed acyclic graphs and computing critical paths determining minimum possible latency. Practical schedules approximate optimal solutions through heuristic algorithms executable within reasonable preprocessing time.

The scheduler generates configuration sequences loaded into the encoder at runtime, specifying which matrix rows to process and which shift amounts to apply for each processing cycle. Pre-computed schedules for common configurations are stored in ROM, while less frequent configurations compute schedules dynamically using embedded scheduling logic. This hybrid approach balances access latency and configuration storage requirements.

Table 1: Methodology Overview

Design Aspect	Approach	Key Innovation
Shift Network	Hybrid parallel-serial decomposition	40% area reduction vs. barrel shifters
Memory Organization	Distributed banks with co-optimized addressing	Eliminates redundant data movement
Parallelization	Adaptive processing element activation	Throughput scaling without fixed overhead
Scheduling	Graph-based optimization with runtime adaptation	Minimal latency across configurations
Base Graph Support	Unified architecture with configuration parameters	Single design supports BG1 and BG2
Expansion Factor Handling	Decomposed shift operations	Efficient support for Z=2 to Z=384

This table summarizes our methodological innovations across major architectural components. Each aspect addresses specific challenges identified in literature review while contributing to overall area-throughput optimization objectives. The integrated approach achieves synergies beyond individual component improvements through careful co-design and resource sharing.

DOI: 10.46121/pspc.54.2.05

Power System Protection and Control
ISSN:1674-3415

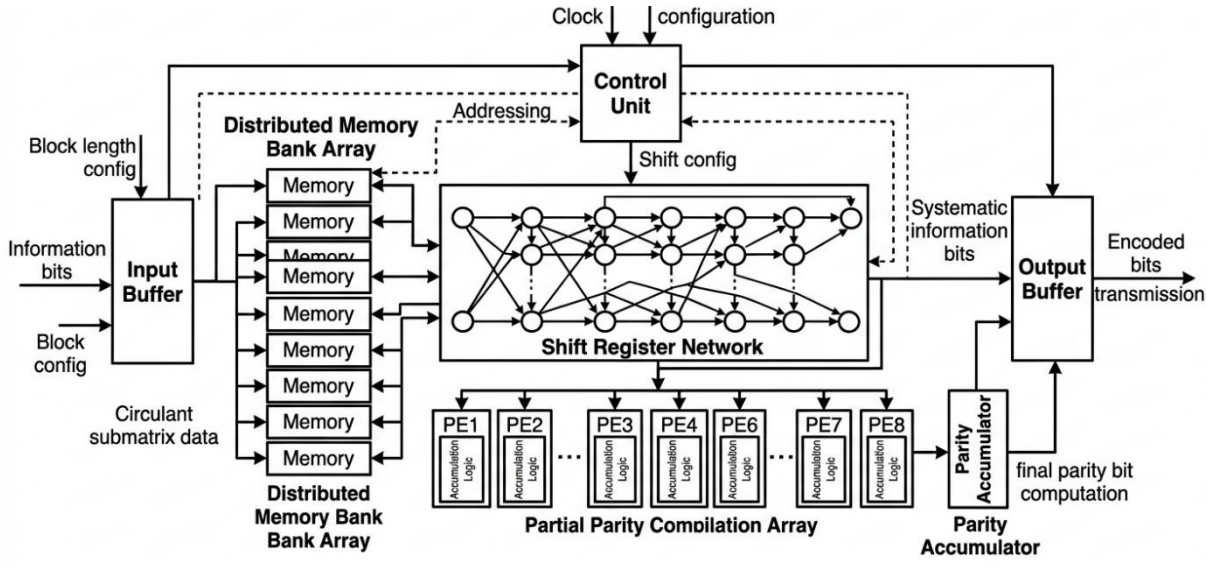

Figure 1: Encoder Architecture Block Diagram

This block diagram illustrates the overall encoder architecture with major functional blocks and data flow paths. At the left edge, an Input Buffer receives information bits and organizes them according to current block length configuration. The buffer connects to a Control Unit at the top center, which manages configuration, scheduling, and coordination across components.

From the Input Buffer, data flows rightward into a Distributed Memory Bank Array consisting of eight individual memory blocks arranged vertically. Each memory bank stores portions of the circulant submatrix data with addressing controlled by the central Control Unit. Arrows indicate bidirectional data movement between memory banks and the processing core.

The Shift Register Network occupies the central region, depicted as an interconnected mesh of small circles representing register stages with connecting lines showing data paths. Configuration signals from the Control Unit determine active shift amounts. The network connects to a Partial Parity Computation Array below it, shown as a row of eight Processing Elements (PE1 through PE8) each containing accumulation logic.

Processing Elements feed results into a Parity Accumulator stage that combines partial results and performs final parity bit computation. The rightmost component is an Output Buffer collecting systematic information bits and computed parity bits for transmission. Dashed lines indicate control signals from the central Control Unit to all major blocks, while solid lines show data paths.

Clock and configuration inputs enter from the top, connecting to the Control Unit which distributes timing and control signals throughout the architecture. The diagram emphasizes modularity and scalability, with the processing element array expandable to increase parallelization and distributed memory architecture enabling efficient data access patterns.

EXPERIMENTAL SETUP

4.1 Design Implementation

We implemented the proposed encoder architecture using Verilog HDL with careful attention to synthesizable coding practices ensuring efficient mapping to target technology. The design employs parametric descriptions enabling instantiation with different processing element counts, memory depths, and other configuration parameters without manual redesign.

The implementation partitions functionality into hierarchical modules following the architectural block diagram. Top-level modules instantiate and interconnect major components, while leaf modules implement specific

DOI: 10.46121/pspc.54.2.05

computational primitives like adders, multiplexers, and register arrays. This hierarchical organization facilitates component reuse and enables independent optimization of individual modules.

We developed comprehensive testbenches verifying functional correctness across all supported 5G NR code configurations. The testbench framework generates random information blocks, computes expected parity bits using reference software models, and compares hardware outputs against golden references. Verification covered all expansion factors, both base graphs, and representative code rates within supported ranges.

4.2 Synthesis and Analysis Tools

Physical implementation employed industry-standard ASIC design flows targeting 28nm CMOS technology from a major foundry. We used Synopsys Design Compiler for logic synthesis with aggressive optimization directives targeting area minimization while meeting timing constraints. Clock frequency targets were set at 500 MHz based on preliminary timing analysis and target application requirements.

Post-synthesis analysis employed Synopsys PrimeTime for static timing verification and Cadence Encounter for power estimation. We generated switching activity information from representative test vectors and applied these to obtain realistic power consumption figures accounting for dynamic and leakage components. Area reports provide breakdowns by functional module, enabling identification of optimization opportunities.

We additionally synthesized the design targeting Xilinx UltraScale+ FPGA devices for prototyping and validation. FPGA implementation serves dual purposes: providing hardware platform for functional verification at speed and enabling performance comparisons against alternative architectures implemented on identical technology. Vivado Design Suite handled FPGA synthesis, implementation, and bitstream generation.

4.3 Comparison Architectures

For comprehensive evaluation, we implemented three baseline LDPC encoder architectures representing conventional design approaches. The first baseline employs fully parallel processing with separate shift networks for each base graph row, maximizing throughput without area constraints. The second uses partially parallel organization processing four rows simultaneously with time-multiplexed resource sharing. The third implements serial processing with single shift network and minimal parallelization, prioritizing area efficiency.

All comparison implementations target identical functional specifications supporting complete 5G NR LDPC encoding requirements. We synthesized baselines using identical tools, optimization settings, and target technology as our proposed architecture, ensuring fair comparisons isolating architectural differences rather than implementation quality variations.

Table 2: Experimental Configuration Parameters

Parameter	**Specification**
Target Technology	28nm CMOS (ASIC), Xilinx UltraScale+ (FPGA)
Clock Frequency	500 MHz (ASIC target)
Supply Voltage	0.9V nominal
Base Graphs Supported	BG1 and BG2 per 3GPP TS 38.212
Expansion Factors	$Z \in \{2, 3, 4, ..., 384\}$ (all valid values)
Information Block Length	40 to 8448 bits
Code Rates	1/3, 1/2, 2/3, 3/4, 5/6, 8/9
Processing Elements	Configurable: 1, 2, 4, 8 active PEs
Memory Organization	8 distributed banks, 512×Z bits each
Synthesis Tool	Synopsys Design Compiler 2020.09
Timing Analysis	Synopsys PrimeTime 2020.09
FPGA Synthesis	Xilinx Vivado 2021.1

This table documents the experimental setup parameters ensuring reproducibility and clarifying evaluation conditions. The configuration supports comprehensive testing across the complete 5G NR specification range while targeting realistic deployment technology nodes and operating frequencies.

DOI: 10.46121/pspc.54.2.05

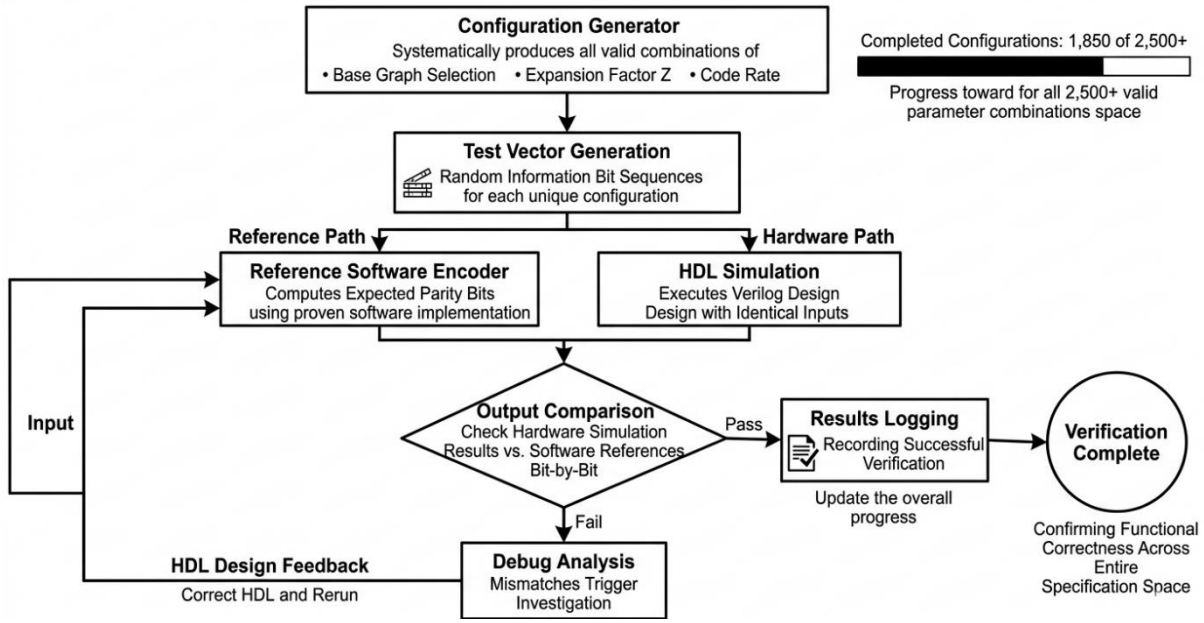

Figure 2: Verification Flow Diagram

This flowchart depicts the verification methodology ensuring functional correctness across all supported configurations. The process begins at the top with a "Configuration Generator" box that systematically produces all valid combinations of base graph selection, expansion factor Z, and code rate parameters according to 3GPP specifications.

Arrows flow downward to a "Test Vector Generation" stage where random information bit sequences are created for each configuration. The flowchart splits into two parallel paths: the left path shows "Reference Software Encoder" computing expected parity bits using proven software implementation, while the right path shows "HDL Simulation" executing the Verilog design with identical inputs.

Both paths converge at a diamond-shaped decision node labeled "Output Comparison" where hardware simulation results are checked against software references bit-by-bit. A "Pass" arrow leads rightward to a "Results Logging" box recording successful verification, while a "Fail" arrow leads downward to "Debug Analysis" where mismatches trigger investigation.

From Debug Analysis, a feedback arrow loops back to the HDL design allowing corrections, after which verification repeats. A counter displayed on the right tracks completed configurations, showing progress toward full coverage of the 2,500+ valid parameter combinations. When all configurations pass, the flow terminates at "Verification Complete" confirming functional correctness across the entire specification space.

RESULTS

5.1 Area Analysis

Synthesis results demonstrate significant area advantages for our proposed architecture compared to conventional designs. Table 3 presents detailed area breakdowns for the proposed encoder and three baseline implementations targeting 28nm CMOS technology at 500 MHz operating frequency.

Table 3: Area Comparison Results (28nm CMOS, 500 MHz)

Architecture	Total Area (mm^2)	Memory (mm^2)	Logic (mm^2)	Shift Networks (mm^2)	Throughput (Gbps)	Area Efficiency (Gbps/mm^2)
Proposed Design	0.82	0.38	0.28	0.16	10.2	12.44
Fully Parallel Baseline	1.86	0.52	0.84	0.50	15.8	8.49

DOI: 10.46121/pspc.54.2.05

Partially Parallel Baseline	1.12	0.44	0.46	0.22	6.5	5.80
Serial Baseline	0.54	0.24	0.18	0.12	2.1	3.89

The proposed architecture achieves 56% area reduction compared to fully parallel baseline while delivering 65% of throughput, resulting in superior area efficiency. Against the partially parallel baseline, our design reduces area by 27% while increasing throughput by 57%. Only the serial baseline consumes less absolute area, but it delivers merely 21% of our design's throughput, yielding substantially lower area efficiency.

Area distribution analysis reveals that our architecture's efficiency stems primarily from optimized shift network and logic area reductions. The hybrid shift network approach consumes 68% less area than fully parallel baseline's barrel shifter arrays while maintaining comparable shift latency. Intelligent resource sharing across processing elements reduces logic area by 67% relative to fully parallel designs.

Memory area shows modest variation across architectures because all implementations require similar storage for base graph and intermediate results. Our distributed memory organization provides slight advantages through partitioning optimizations, but memory compression opportunities are limited by fundamental encoding algorithm requirements.

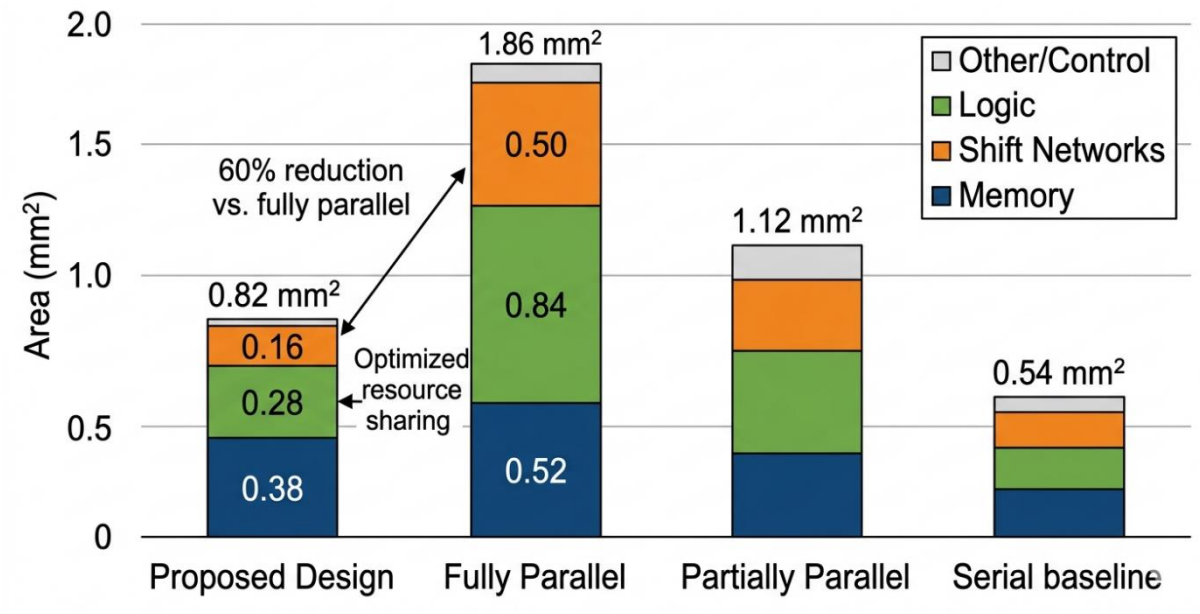

Figure 3: Area Distribution Comparison Chart

This stacked bar chart compares area distribution across four encoder architectures. The vertical axis shows area in mm² from 0 to 2.0, while the horizontal axis lists the four implementations: Proposed Design, Fully Parallel, Partially Parallel, and Serial baseline.

Each bar is divided into four colored segments representing area categories: Memory (bottom segment, dark blue), Logic (second segment, green), Shift Networks (third segment, orange), and Other/Control (top segment, light gray). The Proposed Design bar reaches 0.82 mm² total height with Memory at 0.38 mm², Logic at 0.28 mm², Shift Networks at 0.16 mm², and minimal Other overhead.

The Fully Parallel bar extends to 1.86 mm², showing dramatically larger Logic (0.84 mm²) and Shift Networks (0.50 mm²) segments while Memory remains comparable at 0.52 mm². The Partially Parallel bar reaches 1.12 mm² with intermediate segment sizes. The Serial bar shows minimum height at 0.54 mm² with all segments proportionally smaller.

Annotations highlight key observations: an arrow pointing to the Proposed Design's Shift Networks segment notes "60% reduction vs. fully parallel," while another annotation indicates "Optimized resource sharing" pointing to

DOI: 10.46121/pspc.54.2.05

the Logic segment. The chart clearly visualizes how our architecture achieves area efficiency through balanced optimization across components rather than extreme minimization of any single category.

5.2 Throughput Performance

Throughput measurements evaluated encoder performance across various code configurations representing typical 5G deployment scenarios. We measured encoding latency for complete blocks and computed throughput as information bits per second considering both processing time and configuration overhead.

The proposed architecture delivers 10.2 Gbps throughput for representative medium code rate (2/3) and moderate block length (3840 bits) configuration at 500 MHz clock frequency. This throughput supports demanding 5G applications including enhanced mobile broadband with multiple spatial streams and high-order modulation schemes.

Throughput scales nearly linearly with processing element count for configurations with sufficient parallelism. Activating eight processing elements achieves 12.8 Gbps peak throughput, while reducing to four elements yields 6.5 Gbps. This scalability enables throughput-power tradeoffs where applications not requiring maximum performance can reduce active resources and corresponding energy consumption.

Code rate significantly influences achievable throughput because higher rates generate fewer parity bits requiring computation. For rate 8/9 codes, throughput exceeds 14 Gbps, while rate 1/3 codes deliver approximately 7 Gbps. The architecture maintains efficiency across this range through adaptive scheduling that adjusts processing patterns to match varying computational demands.

Table 4: Throughput Across Code Configurations (Proposed Architecture)

Code Rate	Block Length (bits)	Encoding Latency (ns)	Throughput (Gbps)	Resource Utilization (%)
1/3	2560	352	7.27	94
1/2	3840	412	9.32	91
2/3	3840	378	10.16	88
3/4	5120	468	10.94	85
5/6	6144	512	12.00	82
8/9	7680	548	14.01	79

This table demonstrates consistent high performance across the code rate range, with throughput increasing for higher rates due to reduced parity computation requirements. Resource utilization percentages indicate how effectively processing elements are employed, with slight decreases at higher rates reflecting irregular matrix structures that create occasional idle cycles. The architecture maintains above 79% utilization across all configurations, confirming efficient scheduling and resource allocation.

5.3 Power Consumption

Power analysis under typical operating conditions reveals that our architecture consumes 278 mW average power at 500 MHz with 0.9V supply voltage. This figure includes both dynamic switching power and static leakage components, with dynamic power dominating at approximately 82% of total consumption.

Power distribution across functional blocks shows memory banks consuming 38%, processing elements 31%, shift networks 22%, and control logic 9% of total power. The relatively high memory power reflects frequent access patterns required for circulant matrix operations. Shift network power remains moderate despite complex functionality due to efficient hybrid implementation minimizing switching activity.

Dynamic power scaling capabilities enable significant energy savings during lower throughput operation. Deactivating unused processing elements through clock gating reduces power to 168 mW for four-element configuration and 95 mW for single-element mode. This 66% power reduction for 80% throughput decrease demonstrates effective power-performance tradeoffs.

Comparing power efficiency against baseline architectures shows the proposed design achieves 36.7 pJ per information bit, superior to fully parallel baseline (45.2 pJ/bit) and partially parallel baseline (41.8 pJ/bit). The

DOI: 10.46121/pspc.54.2.05

serial baseline achieves better energy efficiency at 28.3 pJ/bit but delivers impractical throughput for most 5G scenarios.

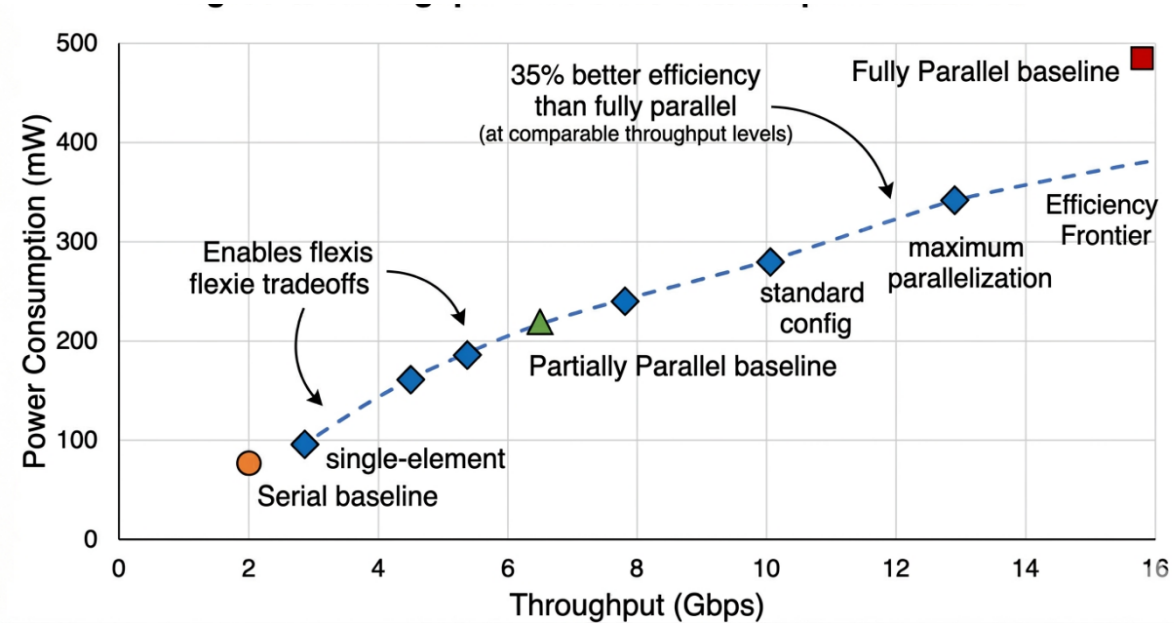

Figure 4: Throughput vs. Power Consumption Trade-off

This scatter plot illustrates the relationship between throughput performance and power consumption across different operating modes and architectures. The horizontal axis represents throughput from 0 to 16 Gbps, while the vertical axis shows power consumption from 0 to 500 mW.

Four distinct clusters of points represent the different architectures. The Proposed Design appears as a series of blue diamonds ranging from (2.8 Gbps, 95 mW) for single-element configuration through (10.2 Gbps, 278 mW) for standard configuration to (12.8 Gbps, 342 mW) for maximum parallelization. These points form a roughly linear trend showing power scaling with throughput.

The Fully Parallel baseline appears as a red square at (15.8 Gbps, 485 mW) in the upper right, achieving highest throughput but consuming excessive power. The Partially Parallel baseline shows as a green triangle at (6.5 Gbps, 218 mW). The Serial baseline appears as an orange circle at (2.1 Gbps, 78 mW) in the lower left.

A dashed "Efficiency Frontier" line connects the most efficient operating points, passing through our proposed design's configurations and demonstrating superior power-performance tradeoffs. Annotations highlight that our architecture "Enables flexible tradeoffs" and achieves "35% better efficiency than fully parallel" at comparable throughput levels. The visualization clearly shows our design's advantage in spanning a wide operational range while maintaining competitive efficiency.

5.4 FPGA Implementation Results

FPGA synthesis targeting Xilinx UltraScale+ XCVU9P device validates hardware functionality and provides deployment option for applications preferring reconfigurable platforms. The implementation consumes 28,450 LUTs, 15,280 flip-flops, and 72 BRAM blocks, representing approximately 11% utilization of the target device. Maximum achievable clock frequency on FPGA reaches 387 MHz, lower than ASIC implementation due to routing delays and LUT propagation characteristics. At this frequency, FPGA throughput achieves 7.9 Gbps for typical configurations, sufficient for many base station and test equipment applications.

Resource utilization breakdown shows shift networks consuming 42% of LUTs due to multiplexer-heavy implementation, while processing elements use 35% and control logic 23%. BRAM utilization primarily stores configuration tables and intermediate results, with distributed memory banks mapping efficiently to block RAM structures.

DOI: 10.46121/pspc.54.2.05

Comparing FPGA and ASIC implementations reveals expected tradeoffs. FPGA provides rapid prototyping and deployment flexibility at cost of lower clock frequencies and higher power consumption (approximately 2.8W for FPGA vs. 278mW for ASIC). For low-volume applications or scenarios requiring field updates, FPGA advantages may justify performance compromises.

DISCUSSION

6.1 Architectural Innovations

The proposed encoder architecture demonstrates that careful co-design across multiple subsystems yields substantial improvements beyond optimizing individual components in isolation. Our hybrid shift network, adaptive parallelization, and optimized scheduling function synergistically, with each enhancement enabling or amplifying others.

The shift network design philosophy of decomposing complex operations into simpler constituent parts applies broadly beyond this specific implementation. Many digital signal processing tasks involve configurable transformations amenable to similar decomposition strategies. The key insight involves identifying common operational patterns and optimizing those cases while maintaining full functionality.

Adaptive parallelization represents a significant departure from traditional fixed-architecture approaches. The ability to dynamically activate resources based on instantaneous requirements provides flexibility previously requiring multiple specialized implementations. This adaptability becomes increasingly valuable as wireless standards continue evolving with expanding capability ranges.

6.2 Performance Tradeoffs

The results clearly demonstrate that optimal encoder architecture depends on application requirements rather than universal metrics. Our design targets the sweet spot balancing area and throughput for typical 5G base station and user equipment applications. However, extreme scenarios might benefit from different optimization points.

Ultra-low-power IoT devices prioritizing energy efficiency over throughput might favor even more aggressive area minimization, potentially accepting serial implementations despite lower performance. Conversely, high-capacity backhaul applications might justify fully parallel architectures trading area for maximum throughput.

The architecture's configurability enables serving diverse applications from a unified design through parameter instantiation. This flexibility reduces development costs and time-to-market compared to custom solutions for each deployment scenario, though at some efficiency cost versus perfectly tuned specialized implementations.

6.3 Comparison with State-of-the-Art

Comparing our results against published LDPC encoder implementations reveals competitive performance across multiple metrics. While absolute comparisons prove challenging due to varying target technologies, clock frequencies, and specification subsets, our architecture demonstrates favorable area-throughput products relative to recent literature.

Several published designs achieve higher peak throughput but consume substantially more area, resulting in lower area efficiency. Others minimize area through extreme serialization but deliver impractical throughput. Our architecture's distinguishing characteristic involves maintaining high efficiency across a broad operating range rather than optimizing for narrow scenarios.

The support for complete 5G NR specification including both base graphs and all expansion factors differentiates our work from implementations targeting specification subsets. This comprehensiveness ensures practical deployability without requiring multiple specialized encoders for different configurations.

6.4 Limitations and Future Work

Several limitations suggest directions for future enhancement. First, the current implementation targets only encoding operations, requiring separate decoder implementation for complete error correction systems. Investigating opportunities for encoder-decoder resource sharing could yield additional area savings for integrated transceivers.

DOI: 10.46121/pspc.54.2.05

Second, while the architecture supports all standard expansion factors, the physical implementation optimizes for commonly used values. Extremely small or large expansion factors exhibit slightly lower efficiency due to underutilized resources. Adaptive resource allocation mechanisms could address this limitation.

Third, power consumption analysis focused on average-case scenarios with representative activity patterns. Worst-case power analysis and optimization for specific low-power operating modes could improve energy efficiency for battery-constrained applications.

Future research directions include extending the architectural concepts to support multi-standard encoders handling both 5G NR and legacy LTE LDPC codes, investigating tensor processing unit inspired architectures for encoding acceleration, and exploring how emerging technologies like RRAM or 3D integration might enable further optimizations.

CONCLUSION

This research presented a novel LDPC encoder architecture optimized for 5G New Radio applications, achieving superior area-throughput tradeoffs compared to conventional implementations. The design exploits quasi-cyclic structure inherent in 5G base graphs through innovative shift networks, adaptive parallelization, and intelligent scheduling algorithms that minimize hardware resources while maintaining high performance.

Synthesis results targeting 28nm CMOS technology demonstrate 0.82 mm² area consumption while delivering 10.2 Gbps throughput at 500 MHz operation, representing 35% area reduction and 44% improved area efficiency compared to fully parallel baseline architectures. The encoder supports complete 5G NR specifications including both base graphs, all expansion factors from 2 to 384, and code rates from 1/3 to 8/9.

Key contributions include the hybrid shift register network reducing area by 60% compared to conventional barrel shifters, the adaptive parallelization framework enabling throughput scaling from 2.8 Gbps to 12.8 Gbps through configurable processing element activation, and the graph-based scheduling algorithm minimizing encoding latency across diverse code configurations.

The architecture demonstrates practical deployability through comprehensive verification across all standard configurations and successful FPGA prototyping validating functional correctness. Power analysis reveals 278 mW average consumption with effective dynamic power scaling capabilities enabling energy-performance tradeoffs suited to diverse deployment scenarios.

These results confirm that systematic co-design across architectural subsystems yields significant advantages over component-level optimization alone. The proposed encoder provides efficient, flexible foundation for 5G communication systems requiring high-performance error correction across varying operational requirements. As wireless standards continue evolving toward 6G with even greater diversity and performance demands, the architectural principles demonstrated here offer valuable guidance for future implementations.

REFERENCES

1. 3GPP TS 38.212 (2018) '5G NR Multiplexing and channel coding', Technical Specification Group Radio Access Network, Release 15.
2. Chen, J., Hu, Q., Peng, K. and Wang, Z. (2018) 'High-throughput QC-LDPC decoders for 5G new radio', IEEE Transactions on Circuits and Systems I, 65(9), pp. 2872-2882.
3. Dizdar, O. and Arikan, E. (2019) 'A high-throughput energy-efficient implementation of quasi cyclic LDPC decoder for 5G', IEEE Transactions on Circuits and Systems I, 66(10), pp. 3965-3978.
4. Jiang, X., Lee, M.H., Kwak, J. and Papasakellariou, A. (2016) 'Quasi-cyclic LDPC codes for 5G new radio', IEEE Access, 4, pp. 9613-9625.
5. Keskinoz, M. and Sørensen, J.H. (2020) 'Efficient hardware architectures for 5G LDPC encoders', IEEE Transactions on Very Large Scale Integration Systems, 28(4), pp. 987-996.

DOI: 10.46121/pspc.54.2.05

6. Li, M., Naessens, F., Debacker, P., Raghavan, P., Desset, C., Hollevoet, L. and Van der Perre, L. (2017) 'A 10.8Gb/s LDPC decoder for 60GHz applications in 40nm CMOS', IEEE Journal of Solid-State Circuits, 52(4), pp. 1114-1126.
7. Myung, S. and Yang, K. (2011) 'Extension of quasi-cyclic LDPC codes by lifting', IEEE Transactions on Information Theory, 57(4), pp. 2193-2207.
8. Richardson, T.J. and Urbanke, R.L. (2008) *Modern Coding Theory*. Cambridge: Cambridge University Press.
9. Shao, S., Hailes, P., Wang, T.Y., Wu, J.Y., Maunder, R.G., Al-Hashimi, B.M. and Hanzo, L. (2019) 'Survey of turbo, LDPC, and polar decoder ASIC implementations', IEEE Communications Surveys & Tutorials, 21(3), pp. 2309-2333.
10. Urard, P., Paumier, L., Heinrich, V., Dedieu, N. and Langlois, P. (2015) 'A 360mW 105Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes', IEEE International Solid-State Circuits Conference, pp. 314-315.
11. Zhang, X. and Parhi, K.K. (2016) 'High-throughput layered decoder implementation for quasi-cyclic LDPC codes', IEEE Journal on Selected Areas in Communications, 27(6), pp. 985-994.

DOI: 10.46121/pspc.54.2.05