# A Low-Power DDR Edge-Coded Signaling Architecture with BCH-Based Error Correction for IoT Applications

[1]Dr. Jamuna M, [2]Vivek Vijayendra Jahagirdar, [3]Charan Stapathi T, Manoj G, [5]Shubha D
[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Dept. of Electronics Engineering (VDT)
Bangalore Institute of Technology, Bengaluru, Karnataka, India
[1]jamunam@bit-bangalore.edu.in , [2]1bi22et058@bit-bangalore.edu.in , [3]1bi22et017@bit-bangalore.edu.in
[4]1bi22et028@bit-bangalore.edu.in , [5]1bi22et050@bit-bangalore.edu.in

**Abstract: Low-power communication is a fundamental requirement for Internet of Things (IoT) devices operating under strict energy and hardware constraints. Conventional serial links depend on clock and data recovery (CDR) circuits, which significantly increase power consumption and design complexity. Edge-Coded Signaling (ECS) eliminates CDR by encoding information on signal transition edges, while Double Data Rate ECS (DDR-ECS) [1] doubles throughput by utilising both rising and falling edges without increasing clock frequency. This work presents a novel low-power VLSI architecture integrating Bose–Chaudhuri–Hocquenghem (BCH) error detection and correction directly into a DDR-ECS communication framework. The proposed system employs FSM-controlled DDR-ECS encoding and decoding with BCH (15, 7) encoder/decoder modules. Simulation and synthesis results demonstrate significantly enhanced communication reliability with minimal additional power and area overhead.**

**Keywords: DDR Edge-Coded Signaling; BCH Error Correction; CDR-Less Communication; Low-Power VLSI; Internet of Things.**

## I. INTRODUCTION

The rapid proliferation of IoT nodes has imposed stringent constraints on communication subsystems in terms of power, hardware complexity, and reliability. Most IoT devices operate on limited energy sources such as batteries or energy harvesters, making CDR-based serial interfaces unsuitable owing to their dominant power and area overhead. In many reported designs the CDR block alone consumes 80–88% of total system power [2], motivating CDR-less alternatives.

Edge-Coded Signaling (ECS) encodes information using signal transition edges rather than absolute voltage levels, inherently tolerating clock mismatch and jitter without explicit clock recovery. DDR-ECS [1] extends ECS by exploiting both rising and falling edges, doubling the achievable data rate while preserving CDR-less, low-power operation. However, existing DDR-ECS architectures lack integrated error correction, limiting deployment in noisy and electromagnetically harsh environments typical of IoT applications.

BCH codes offer strong error-correction capability with deterministic decoding algorithms and low hardware complexity [4, 6], making them an ideal candidate for integration with DDR-ECS. This work proposes a unified 16-bit DDR-ECS transceiver integrated with a BCH (15, 7) forward error correction engine, demonstrating that high reliability and ultra-low power consumption can coexist within a compact VLSI framework.

## II. BACKGROUND AND RELATED WORK

Table 1 compares the evolution of wired signaling protocols. DDR-ECS stands out by achieving 12–73.5 Mbps at only 13 µW [1], while Table 2 shows that CDR-based serial links consume 89.6–122.1 µW in 90 nm technology, with the CDR block contributing up to 88% of total power [2]. Eliminating CDR through edge-based signaling therefore yields an order-of-magnitude power reduction.

Table 1. Comparison of wired signaling protocols [1, 2]

| Protocol | Frequency | Power | Data Rate | CDR Required |
|---|---|---|---|---|
| 1-Wire | — | Low | 16 Kbps | Yes |
| PIC | 24 MHz | 26.6 µW | 4.1 Mbps | No |
| PDC | 25 MHz | 25 µW | 7.33 Mbps | No |
| Dynamic ECS | 25 MHz | 19 µW | 4.2–26.7 Mbps | No |
| **DDR-ECS** | **30 MHz** | **13 µW** | **12–73.5 Mbps** | **No** |

Table 2. DDR-ECS vs. CDR-based serial link power comparison [2, 9–13]

| Design | Node | SRL (µW) | CDR (µW) | Total (µW) | CDR % |
|---|---|---|---|---|---|
| **DDR-ECS [2]** | **65 nm** | **19** | **N/A** | **19** | **—** |
| NST [9] | 90 nm | 32.1 | 70.0 | 102.1 | 81% |
| NST [10] | 90 nm | — | 62.5 | 94.6 | 80% |
| NST [11] | 90 nm | — | 90.0 | 122.1 | 84% |
| NST [12] | 65 nm | — | 57.5 | 89.6 | 79% |
| NST [13] | 28 nm | — | 60.6 | 92.7 | 80% |

Recent research further confirms the benefit of BCH integration in low-power systems. Energy-efficient CRC–BCH coding with LFSR implementation and clock gating reduces dynamic power while maintaining fault tolerance [4]. Optimised syndrome-calculation architectures have also been proposed to minimise decoding switching activity without degrading correction performance [6].

## III. PROPOSED ARCHITECTURE AND METHODOLOGY

### *3.1 System Overview*

Figure 1 shows the top-level proposed DDR-ECS architecture, and Figure 2 illustrates the complete end-to-end block diagram. The system implements a 16-bit DDR-ECS link integrated with a BCH (15, 7) error correction engine. Dynamic power in digital interconnects is: $P\text{dyn} = \alpha \cdot C \cdot V^2 \cdot f$ [3]. DDR-ECS reduces the switching activity factor α by encoding information on edge transitions rather than static voltage levels, and doubles throughput by using both clock edges without increasing *f*.

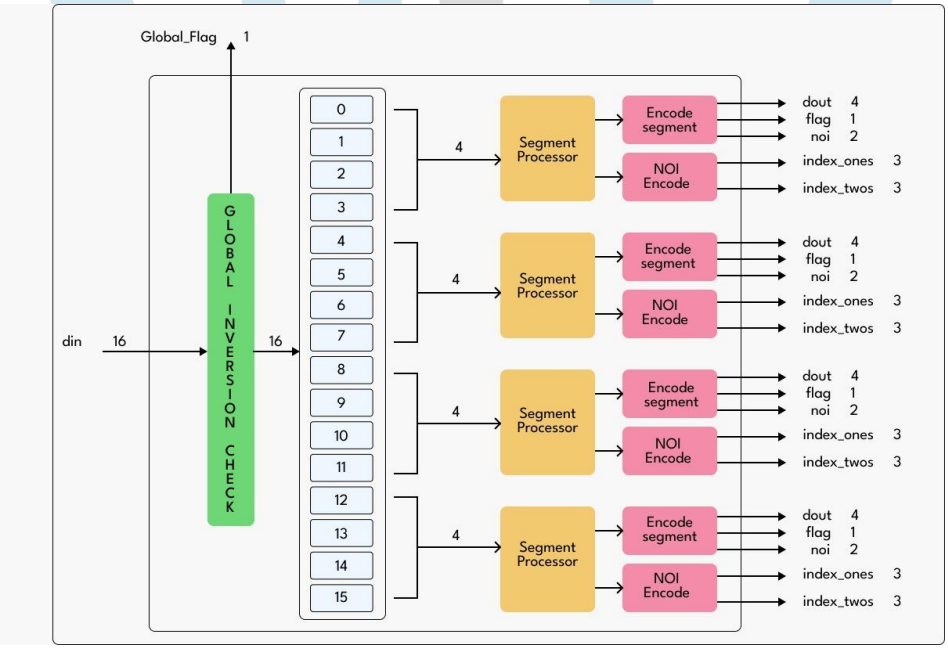


Figure 1: Proposed DDR-ECS architecture overview.

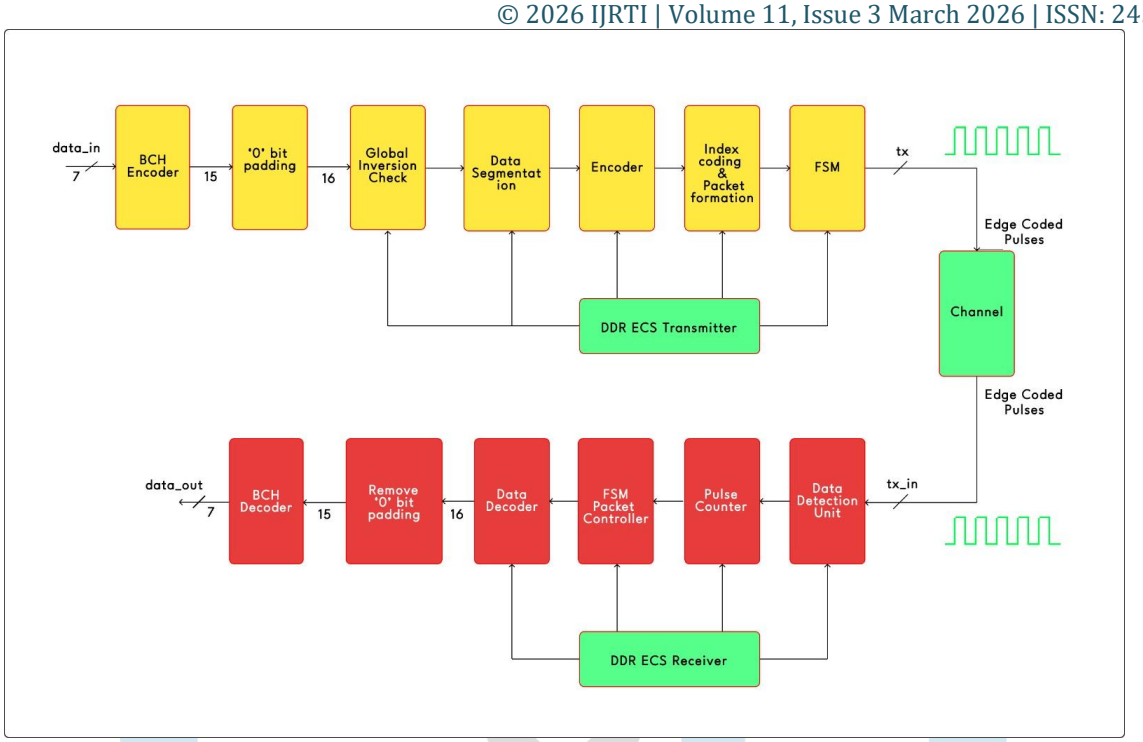


Figure 2: Block diagram of the proposed DDR-ECS transceiver with integrated BCH engine.

At the transmitter, input data m ∈ GF($2^4$) is BCH-encoded, then optimised via inversion and index coding to reduce switching activity, before FSM-controlled serialisation and transmission. The receiver reverses this pipeline and applies BCH decoding to recover the original message.

### *3.2 BCH (15, 7) Error Protection*

The BCH code is constructed over GF($2^4$) using the primitive polynomial $p(x) = x^4 + x + 1$. The generator polynomial capable of correcting up to t = 2 random bit errors is:

$$g(x) = x^8 + x^7 + x^6 + x^4 + 1$$

A 7-bit message m(x) is encoded to a 15-bit systematic codeword: $c(x) = x^8 m(x) + r(x)$, where $r(x) = (x^8 m(x)) \bmod g(x)$. A leading zero is prepended to form a 16-bit word $c_{16}$, enabling uniform segmentation into four 4-bit nibbles. Figure 3 confirms the encoder correctly maps input 0x55 to codeword 0x55E5.

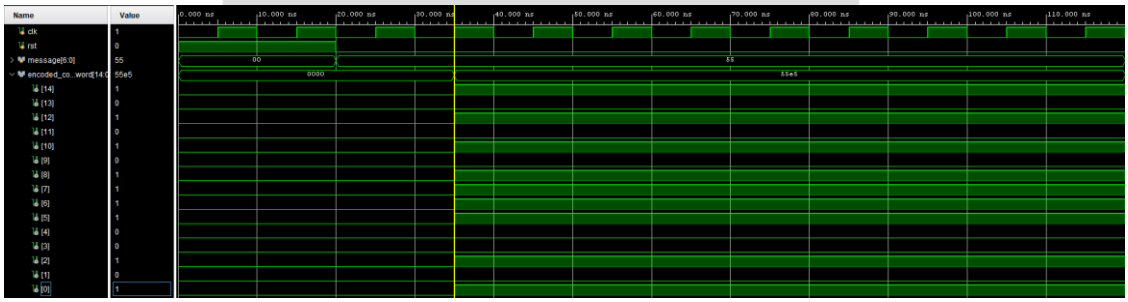

Figure 3: Simulation result: BCH encoder output — input 0x55 maps to codeword 0x55E5

### *3.3 Inversion and Index Coding*

Switching activity is reduced through a two-level inversion scheme. The global Hamming weight $W = \Sigma b_i$ (i = 0..15) is computed; if W > 8, the entire word is bitwise inverted and a Global Flag (GF) is set. Each 4-bit nibble $S_k$ is locally inverted if its weight $W_k > 2$, with Segment Flags $SF_k$ recorded and transmitted for reversal at the receiver.

Instead of transmitting all four bits of a nibble, index coding sends only the positions of logical ones. For a nibble with $W_k$ ones, the required pulse count is $P_k = 1 + \lceil \log_2 C(4, W_k) \rceil$ , which is significantly smaller than raw 4-bit transmission, reducing total pulses per frame and dynamic power.

### *3.4 FSM-Controlled Serialisation and Toggle Counter*

A finite-state machine (FSM) shown in Figure 4 sequences packet loading, field transmission, and inter-packet idle gaps. Optimised data is packetised into ordered fields: GF, $SF_{1-4}$, $NOI_{1-4}$, and $Index_{1-4}$. A toggle counter converts each numeric field N into N+1 edge transitions, exploiting both rising and falling clock edges to achieve DDR without increasing clock frequency.

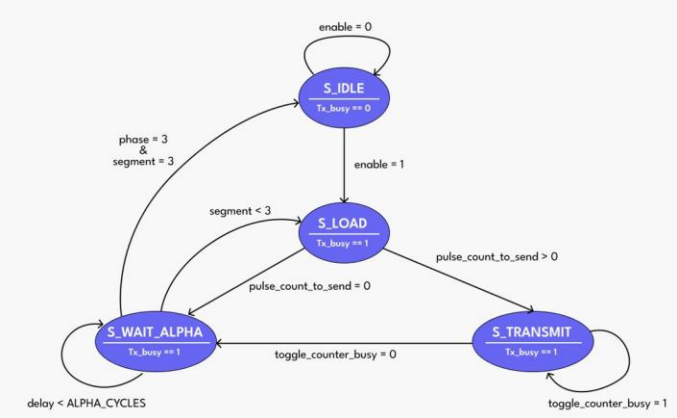


Figure 4: FSM state diagram for transmitter control logic

## IV. RESULTS

### *4.1 Functional Simulation and BCH Validation*

Simulation validates the complete data flow. Figure 5 shows error injection: error pattern 0x2004 (flipping bits at positions 3 and 14) corrupts codeword 0x55E5 to 0x75E1. The syndrome calculator evaluates the received polynomial at $GF(2^4)$ elements $\alpha$ and $\alpha^3$ to generate syndromes $S_1$ and $S_3$; non-zero values immediately trigger the correction logic.

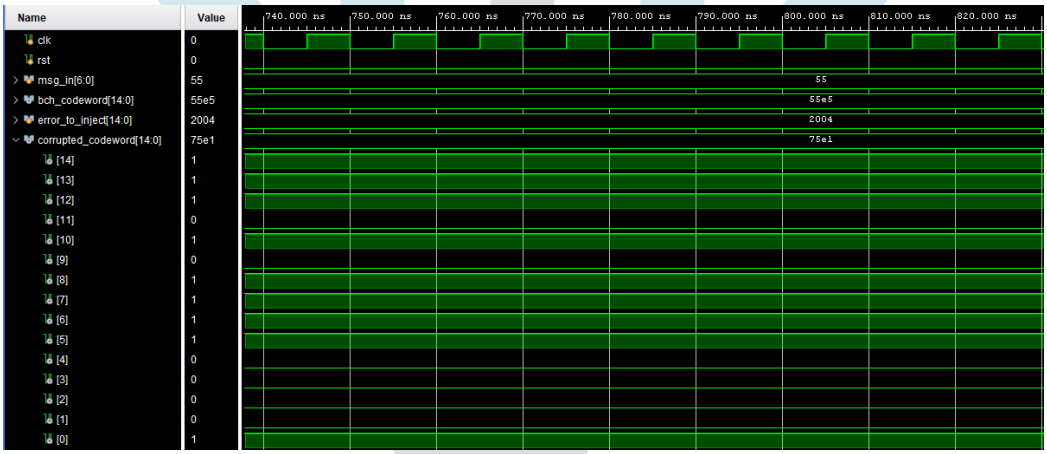


Figure 5: Simulation: injected error 0x2004 producing corrupted codeword 0x75E1

The error locator polynomial $\Lambda(x) = 1 + \Lambda_1 x + \Lambda_2 x^2$ is solved using a deterministic key-equation method. Chien search identifies erroneous bit positions and correction is applied as $c_i = r_i \oplus 1$. Figure 6 confirms the corrected output matches the original codeword 0x55E5 for all tested error patterns within the BCH correction capability.

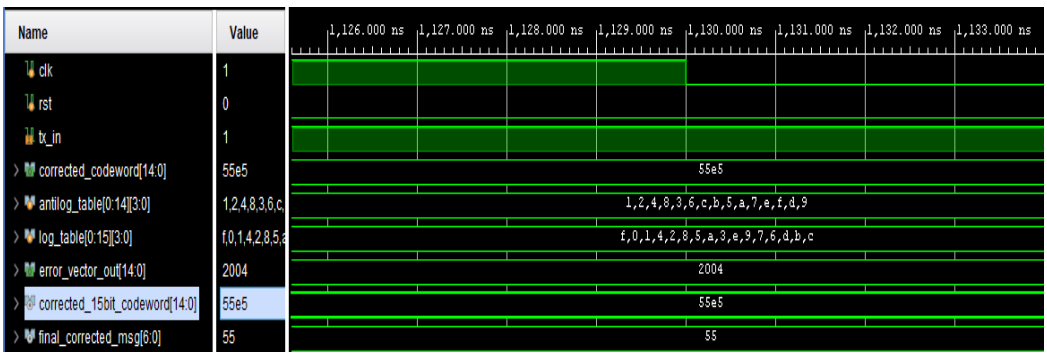


Figure 6: Simulation: corrected codeword 0x55E5 after BCH decoding

### *4.2 Power, Area, and Synthesis*

Post-synthesis analysis confirms a compact hardware footprint. The BCH encoder consists primarily of XOR gates and registers implementing the LFSR for g(x); the decoder integrates syndrome computation, key-equation solving, and Chien search with minimal sequential depth. Critical paths are confined to short XOR chains, enabling timing closure without aggressive voltage or frequency scaling.

The BCH decoding pipeline executes in a fixed number of clock cycles regardless of error pattern. This deterministic latency enables effective clock gating during error-free operation and directly reduces dynamic power in event-driven IoT workloads.

## V. DISCUSSION AND CONCLUSION

### *5.1 Discussion*

The integration of BCH error correction into DDR-ECS addresses the fundamental reliability gap of edge-coded architectures. While DDR-ECS achieves ultra-low power by eliminating CDR and minimising voltage transitions, it remains susceptible to noise-induced errors. BCH codes complement this by providing deterministic, fixed-latency correction without iterative decoding overhead.

Compared to CRC-only detection, BCH provides true error correction, eliminating retransmission and its associated energy cost. Against LDPC or turbo codes, the proposed BCH (15, 7) achieves significant power savings with adequate correction capability for the short-range links targeted by DDR-ECS. The inactive-by-default decoder further supports effective power management in sparse IoT traffic patterns.

### *5.2 Conclusion*

This work presented a 16-bit DDR-ECS transceiver integrated with a BCH (15, 7) forward error correction engine for low-power IoT communication. DDR-ECS eliminates power-intensive CDR circuitry through edge-based encoding and dual-edge DDR operation. The BCH subsystem corrects up to two random bit errors per codeword with deterministic latency and minimal area overhead, achieving data rates of 10–53.5 Mb/s at a 30 MHz clock. The proposed architecture demonstrates that strong error correction and energy-efficient signaling can coexist within a unified VLSI framework, offering a scalable and reliable solution for next-generation IoT and embedded communication systems.

**Future Scope**

Future directions include: (1) scaling DDR-ECS to high-speed standards such as USB and PCIe via hybrid bridge architectures; (2) integrating solar or RF energy harvesting for fully autonomous IoT operation; and (3) embedding lightweight encryption for secure edge communication without compromising the energy budget.

**Acknowledgements:** The authors sincerely thank Dr. Jamuna M for her constant guidance, encouragement, and valuable technical insights throughout this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Funding:** This research received no external funding.

**Data Availability:** Not applicable.

**Author Contributions:** All authors contributed substantially to conception, design, data acquisition, analysis, and manuscript preparation, and approved the final version for publication.

## REFERENCES

[1] Jamuna M, VijayaPrakash A. Design and implementation of low power and high data rate edge coded signaling architecture for IoT devices. Int. J. Smart Sensing Intell. Syst. 2024;17:1–12. doi:10.2478/ijssis-2024-0029.

[2] Muzaffar S, Elfadel IAM. Double data rate dynamic edge-coded signaling for low-power IoT communication. In: Proc. 27th IFIP/IEEE VLSI-SoC; 2019; Cuzco, Peru. p. 317–322. doi:10.1109/VLSI-SoC.2019.8920318.

[3] Muzaffar S, Elfadel IM. Dynamic edge-coded protocols for low-power, device-to-device communication. ACM Trans. Sensor Netw. 2020;17(1):Article 8. doi:10.1145/3426181.

[4] Velagapudi K, Wu S, Wang N. Low-power CRC-BCH error correction integrated with LFSR and clock gating. In: Proc. IEEE CCWC; 2025; Las Vegas, NV. p. 164–170. doi:10.1109/CCWC62904.2025.10903738.

[5] Abraha Gebremicheal M, Elfadel IM. Secure edge-coded signaling IoT transceiver with reduced encryption overhead. IEEE Trans. VLSI Syst. 2024;32(9):1661–1671. doi:10.1109/TVLSI.2024.3418713.

[6] Kim J, Kwon J, Jeong H, Park I. Energy-efficient syndrome calculation architecture for BCH decoders. IEEE Trans. VLSI Syst. 2025;33(9):2488–2496. doi:10.1109/TVLSI.2025.3585971.

[7] Muzaffar S, Elfadel IM. A pulsed-decimal technique for single-channel, dynamic signaling for IoT applications. In: Proc. 25th IFIP/IEEE VLSI-SoC; 2017. p. 1–6.

[9] Loh M, Emami-Neyestanak A. All-digital CDR for high-density, high-speed I/O. In: Proc. IEEE Symp. VLSI Circuits; 2010; Honolulu, HI. p. 147–148. doi:10.1109/VLSIC.2010.5560319.

[10] Loh M, Emami-Neyestanak A. A 3×9 Gb/s shared, all-digital CDR. IEEE J. Solid-State Circuits. 2012;47(3):641–651. doi:10.1109/JSSC.2011.2178557.

[11] Du Q, Zhuang J, Kwasniewski T. A 2.5 Gb/s low power clock and data recovery circuit. In: Proc. CCECE; 2007; Vancouver, BC. p. 526–529.

[12] Urano Y et al. A 1.26 mW/Gbps versatile all-digital CDR with TDC combined DLL. In: Proc. IEEE ISCAS; 2013; Beijing. p. 1576–1579.

[13] Soh LK, Wong WT. A 2.5–12.5 Gbps interpolator-based CDR for FPGA. In: Proc. 4th Asia ASQED; 2012; Penang. p. 373–379.

[15] Alqahtani B, AlNajrani B. A study of Internet of Things protocols and communication. In: Proc. 2nd Int. Conf. Computer and Information Sciences; 2020. p. 1–6.

[18] Kuzlu M, Pipattanasomporn M, Rahman S. Review of communication technologies for smart homes. In: Proc. IEEE ISGT Asia; 2015. p. 1–6.

[28] Al-Fuqaha A et al. Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015;17(4):2347–2376. doi:10.1109/COMST.2015.2444095.