Fault protection of quasi-delay-insensitive pipeline models using efficient coding for asynchronous network-on-chip

Renu Siddagangappa, Nayana Dunthur Krishnagowda, Deepthi Tumkur Srinivas Murthy
School of Electrical and Computer Engineering, REVA University, Bengaluru, India

Article Info

Article history:

Received Oct 25, 2023
Revised Nov 18, 2023
Accepted Nov 30, 2023

Keywords:

Coding approach
Fault detection
Fault-tolerant
Pipeline model
Quasi-dealy-insensitive

ABSTRACT

One promising approach for creating the chip-level connection of multiprocessing system on chip (MPSoC) is asynchronous logic. However, asynchronous systems are susceptible to errors. In this manuscript, the efficient fault-tolerant (FT) quasi-delay-insensitive (QDI) pipeline modules are designed using a delay-insensitive redundant check (DIRC) coding mechanism. The DIRC coding approach can tolerate single and multi-bit transient faults (TFs) in QDI-pipeline modules. The 4-phase 1-of-n coding approach incorporates DIRC-based QDI pipeline stages to strengthen the asynchronous links against TFs. The DIRC-based QDI pipeline stages are further used as asynchronous links in asynchronous network-on-chip ((NoC) for fault-free communication. The performance metrics like chip area, delay, and power parameters are evaluated in detail against different data widths for both basic unprotected and DIRC-based QDI pipeline modules. The DIRC-based QDI-pipeline module with 1-of-4 code uses only <2% chip area with a delay of 7.4 ns and power of 117 mW on Artix-7 chip for data width 128. The code rate of the proposed work decreased by 33.33% for both 1-of-2 and 1-of-4 codes in DIRC-based QDI-pipeline modules.

Corresponding Author:

Renu Siddagangappa
School of Electrical and Computer Engineering, REVA University
Bengaluru, India
Email: renusiddagangappa@gmail.com

1. INTRODUCTION

Asynchronous circuits have received emphasis from system-on-chip (SoC) designers as clock distribution on a large chip grows increasingly challenging. Several synchronous islands communicate asynchronously with one another over asynchronous communication networks in a global asynchronous-local synchronous (GALS) system. Due to its numerous benefits, the asynchronous network-on-chip (NoC) paradigm has consequently evolved as an alternative method for communicating with big SoC designs [1]–[2]. The asynchronous-based NoCs offer several benefits over equivalent synchronous solutions. On the other hand, the links and routers that make up on-chip networks should be tolerant of various defect types as semiconductor innovation grows and on-chip networks grow [3]. However, asynchronous designs can have several drawbacks. Designing asynchronous circuits is more complicated and involved than designing synchronous circuits because asynchronous circuits must operate without errors and avoid deadlock during handshaking. Due to the handshaking procedure, asynchronous circuits could need a more prominent design area. Asynchronous circuit design and development are additionally made more challenging by the absence of developing computer-aided design (CAD) tools [4]–[5]. In particular, asynchronous circuits and quasi-delay-insensitive (QDI) circuits are appealing for their ability to tolerate such fluctuation. They are achieving popularity in the SoC sector for this compassion, which may be converted into chip area reduction, timing closure improvement, and power-saving. The QDI circuits connect logic gates without a clock signal to sequence and function properly, even when gate delays occur [6]–[7]. Despite abundant research on fault-tolerant scheduled logic systems, asynchronous circuits have received little attention. Practical strategies for synchronous systems are rendered useless or inefficient because the lack of clock inputs renders a defective asynchronous circuit susceptible to issues unlikely to occur in a timed logic. New techniques must be investigated for fault tolerance in asynchronous circuits [8]–[10].

The existing QDI-based asynchronous NoC modules and their coding mechanism with faults tolerant features are discussed with their performance realization. Pontes *et al*. [11] present the converters based on the 2-phase protocol for asynchronous 1-of-n codes with 3-dimensional (3D) features. The four-phase to two-phase and vice versa conversion is achieved concerning signal transition graph (STG). The long link latency and throughput are discussed for both conversions. The work obtains the 8.77 mW of power for 4-phase NoC architectures. Vivet *et al*. [12] discuss the scalable NoC architecture with fault-tolerable (FT) asynchronous links with 3D features. The asynchronous links are used to tolerate the faults in 3D NoC. The NoC obtains a throughput of 7.8 Gbps with 55.7 mW power using 65-nm complementary metal oxide semiconductor (CMOS) technology. Ouyang *et al*. [13] explain the asynchronous FT router architecture in NoC. The work discusses the handshaking mechanism with the GALS NoC and QDI delay module. The work analyzes the average latency against injection rate for NoC designs. The asynchronous FT router obtains the 9.94 mW power during implementation. Zhang *et al*. [14] describe the deadlock detection and recovery in the QDI NoC caused by permanent faults. The work discusses the asynchronous protocols and pipeline fault module. The fault-caused deadlock and management strategies for QDI NoCs are discussed in detail. The fault-link isolation and recovery are discussed with STG approaches. The work analyzes the power, latency, and throughput against fault numbers. The researches [15]–[17] discuss the secured circuits testing with FT features. The circuits are designed using QDI and wave dynamic differential logic (WDDL) approaches. The FT is achieved using the triple mode redundancy (TMR) approach. The dynamic behavior of TMR-based WDDL and QDI concerns different fault resistances. The fault injection attacking approaches for resistive bridging faults are analyzed.

Ho *et al*. [18] present the asynchronous logic-based QDI sense-amplifier half-buffer (SAHB) method for NoC routers. The SAHB offers less transistor switching and dynamic power consumption by adding QDI in an asynchronous NoC router. The process-voltage-temperature (PVT) variations are very high in SAHB-based QDI circuits. The work obtains the 5.69 ns and 258 MHz throughput latency using 65 nm CMOS technology. Thonnart *et al*. [19] discuss the asynchronous NoC backbone-based industrial system-on-chip (SoC) system for latency analysis. The asynchronous NoC backbone has a bridge and GALS structure followed by QDI routers and links to construct the qualcomm-based SoC system. The asynchronous NoC links improve the throughput by 5% and reduce the latency by 23% with a reduction of 78% power. Nachiar *et al*. [20] explain the heterogeneous NoC module with FT capacity. The synchronous and asynchronous router with a 4-phase handshaking protocol mechanism is discussed in detail. The work analyzes the latency and throughput against uniform traffic for different virtual channels (VCs). Moreira and Giaconi [21] describe advanced SoCs' QDI interconnection mechanism. The chronos link automation is discussed and compared with conventional SoC links to realize the latency and total wire counts. Rashid *et al*. [22] explain the FT-based NoC router for heterogeneous computing architectures. The FT-based router computation and detection are analyzed in detail. The work discusses the latency and reliability analysis against different traffic. The FT-based NoC router obtains more area overhead of 26.6% and 28% power than the baseline router architecture.

Siddagangappa and Nayana [23] present a detailed review of asynchronous NoC architectures with the FT mechanism. The work discusses current asynchronous NoC working operations, their features, and performance comparison. The work also discusses the challenges in synchronous NoCs and suggests possible solutions. Bhat *et al*. [24] present the hybrid combination of clock and power networks on different combinational circuits. The work realizes different universal gates using a CMOS oscillator with an inverter mechanism to minimize power utilization, and it is better than traditional combinational circuits. Ramesh and Abed [25] describe the multi-level cache module for many-core chip processors. The first-level and last-level are shared with a common cache in each bus line to improve the performance in bus-based networks. The work analyzes the memory access time, throughput and fault realization. Al-Musawi *et al*. [26] explain the chua-chaotic system (CCS) with an artificial neural network (ANN) mechanism for image encryption and decryption on the hardware platform. The ANN uses a tangent sigmoid to activate each neuron layer. The work analyzes the entropy, histogram and area utilization on the chip. Kavitha *et al*. [27] present the optical ring (OR) NoC wavelength routing architecture. The OR-NoC uses ring topology, which is better than other optical network topologies. The single wavelength is reused for multiple communication on a single waveguide. Nandalal and Bhakthavatchalu [28] discuss the hardware encryption architecture with programmable features for a blockchain-based security system. The work uses an elliptic curve digital signature and secure hash algorithm (SHA) -256 algorithm to enhance the security of blockchain systems.

The efficient fault-tolerant-based QDI pipelined module using delay-insensitive (DI) redundant check (DIRC) codes is discussed in this manuscript. The contribution of the significant work is listed as follows: The DIRC coding approach is used to protect the 4-phase 1-of-n QDI pipeline modules from transient faults (TFs). The DIRC coding mechanism can tolerate all the single-bit and multi-bit TFs in QDI pipeline links. The existing DI or QDI links can transition to the DIRC coding scheme without compromising their inherent timing robustness. With modest and appropriate hardware and performance overhead, the DIRC pipeline can be built adaptable to suit the varied fault-tolerance requirements in real-time systems. The DIRC-based QDI links are used as asynchronous links in asynchronous NoCs for fault-free communication. The organisation of the manuscript is as follows. The DIRC working operation followed by the QDI pipelined module is explained in section 2. The results of the QDI pipelined module using DIRC codes concerning the chip area, latency, and power and code rate are realised in detail in section 3. Lastly, it concludes the overall work with a futuristic scope in section 4.

2. QDI PIPELINE MODEL USING DIRC APPROACH

The fault-tolerant coding mechanism uses two coding types: systematic and unordered. The DI codes are examples of both systematic and unordered codes. The unordered codes of the same length must contain other code words and ensure that valid data words are separated. These unordered codes need a DI mechanism to detect and correct faults or errors. The systematic codes contain data and check fields in a fault-to-tolerant coding mechanism. The data field has original data information, and the check field is obtained from data information and is used to recover the original data bits when a fault occurs. The implementation of the systematic codes is illustrated in Figure 1. It contains sender and receiver units and is separable. The sender unit has a check generator and sends the data and check word information to the receiver. The receiver has an error corrector followed by a check generator. The receiver detects and corrects the errors in the error corrector using the DI approach and generates the original data information.

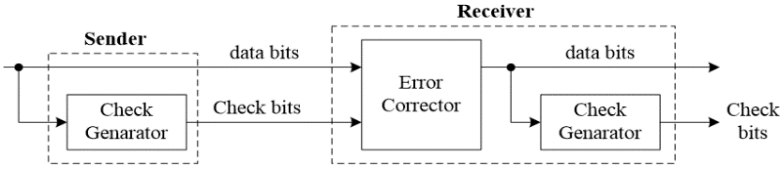


Figure 1. Implementation of systematic codes

2.1. Single DIRC stage

Consider A = (A_0, A_1 … A_{CN-1}) has a data vector and contains a payload as CN 1-of-n code. The check generation operation produces the check word (C), and it is obtained from A. Each A_i and C contains 1-of-n codes. The final DIRC code word has (A_0, A_1 … A_{CN-1}, C) containing the (CN+1) 1-of-n codes ($CN \geq 2$). The systematic DIRC code contains a sequence of code words followed by check bits. Where CN is a number of 1-of-n data words in the DIRC code, this work considers 1-of-2 and 1-of-4 codes for the DIRC coding mechanism. The 1-of-2 and 1-of-4 codes can detect and correct one and 2-bit errors, respectively. The hardware architecture of the single DIRC stage is illustrated in Figure 2.

The single DIRC stage contains three 1-of-n adders, three error filters (EFs) completion detector (CD), and an acknowledge (ACK) generator. The first two 1-of-n adders have data word information, and the last 1-of-n adder is used for check word generation. The code words (A_0, A_1) and check word (C) are inputs to the single DIRC stage and obtain code word output (A_0'', A_1'') and check word (C') as outputs. The TF is applied to the DIRC stage, especially the A_0 input. The error filters correct the fault and produce the original data words. The CD and ACK generator produces the acknowledge signal for the previous DIRC stage. The detailed explanation of each unit is explained as follows. The architecture of the 1-of-n adder is illustrated in Figure 3. The 1-of-2 and 1-of-4 adders are shown in Figures 3(a) and 3(b), respectively. The 1-of-2 adder is constructed using four C-elements and two OR gates. It contains two sets of inputs (a, b), has two codes (a_0, a_1) or (b_0, b_1), and produces (S_0, S_1) as output. The 1-of-4 adder is constructed using sixteen C-elements and four OR gates. It contains four sets of inputs (a, b). Each set contains four codes (a_0, a_1, a_2, a_3) or (b_0, b_1, b_2, b_3) and produces (S_0, S_1, S_2, S_3) as outputs. The 1-of-n adder (2 or 4) outputs are considered as (A_0', A_1'), which contains 1-of-n codes and is used further for error-filtering operations.

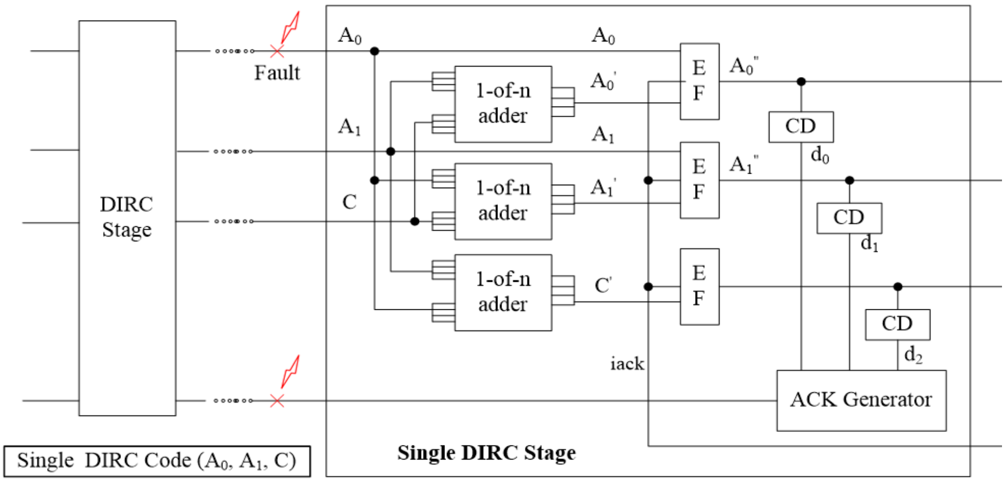
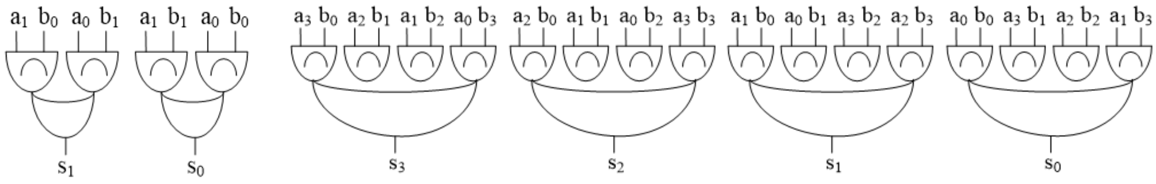


Figure 2. Hardware architecture of the single DIRC stage



(a) (b)

Figure 3. The architecture of 1-of-n adder in terms of (a) 1-of-2 codes and (b) 1-of-4 codes for DIRC stage

The check word (C) generation using the DIRC coding approach is illustrated in Figure 4. The check word for 1-of-2 and 1-of-4 codes (CN = 2) are shown in Figures 4(a) and 4(b), respectively. The check words are generated on the sender side. The DIRC coding mechanism check generation is defined as (C) = ($A_0 + A_1 + \dots + A_{CN-1}$). If any check words are faulty during the transmission process, the correction of the check word is regenerated at the receiver side using an error filtering process. The error filters are used to produce the fault-free output. The receiver receives the 1-bit faulty data from A_0, which is altered by fault. The error filter (EF) for 1-of-n codes is illustrated in Figure 5. The input A_0 of 1-of-n code uses the C-elements with "&" logical operation to construct the EFs. The C-element produces the output 0 or 1 when both the inputs are 0 or 1. The EF for each code word is ($A_0'' = A_0$ & A_0').

A_0	A_1	C
01	01	01
01	10	10
10	01	10
10	10	01

(a)

A_0	A_1	C	A_0	A_1	C	A_0	A_1	C	A_0	A_1	C
0001	0001	0001	0010	0001	0010	0100	0001	0100	1000	0001	1000
	0010	0010		0010	0100		0010	1000		0010	0001
	0100	0100		0100	1000		0100	0001		0100	0010
	1000	1000		1000	0001		1000	0010		1000	0100

(b)

Figure 4. Check word (C) generation using the DIRC coding approach (a) 1-of-2 codes and (b) 1-of-4 codes

The final EF output for 1-of-n code is A''' = (A_0'', A_1'' ... A_{CN-1}''). The 1-bit TF is filtered at the receiver side by error filtering. These filtered and checked word outputs are further used to generate the acknowledge signal (iack). The DIRC Stage submodules are illustrated in Figure 6. Figures 6(a) and 6(b) show the completion detector and ACK generator. The CD receives the filtered and check word outputs (A_0'', A_1'' and C') and performs the simple OR operation for corresponding data bits to generate the CD outputs (d_0, d_1, and d_2). These CD outputs generate ACK signals (ack_0, ack_1, and ack_2) using C-elements.

The obtained ACK signals (ack$_0$, ack$_1$, and ack$_2$) perform the NAND operation to generate the ACK output (iack). This iack signal is used further in the previous DIRC stage for the code word and checks word operations at the sender side.

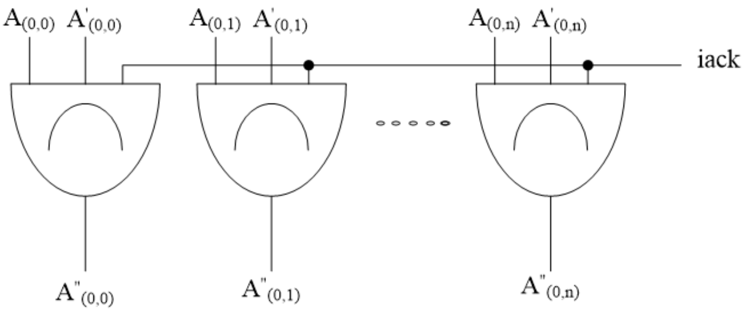
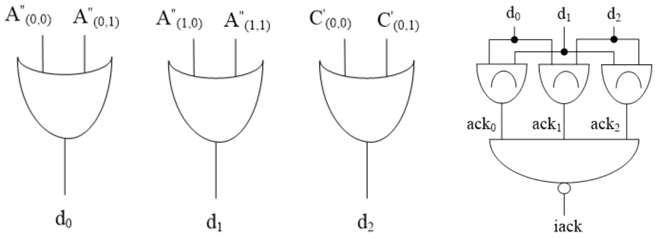


Figure 5. EF for 1-of-n codes



(a) (b)

Figure 6. DIRC stage submodules including (a) completion detection (CD) and (b) ACK generator

2.2. QDI Pipeline module using DIRC

The fault-tolerant is easily achieved in current 1-of-n QDI pipeline modules (4-phase) with minor alteration using the DIRC coding mechanism. The architecture of the QDI Pipeline module with a sequence of stages is illustrated in Figure 7. The primary pipeline contains N 1-of-n data channels [29]. So, divide all the N 1-of-n data channels into G groups (G ≥ 1). 'N' is the basic QDI Pipeline with 1-of-n channels, and 'G' is the number of DIRC channels grouped.

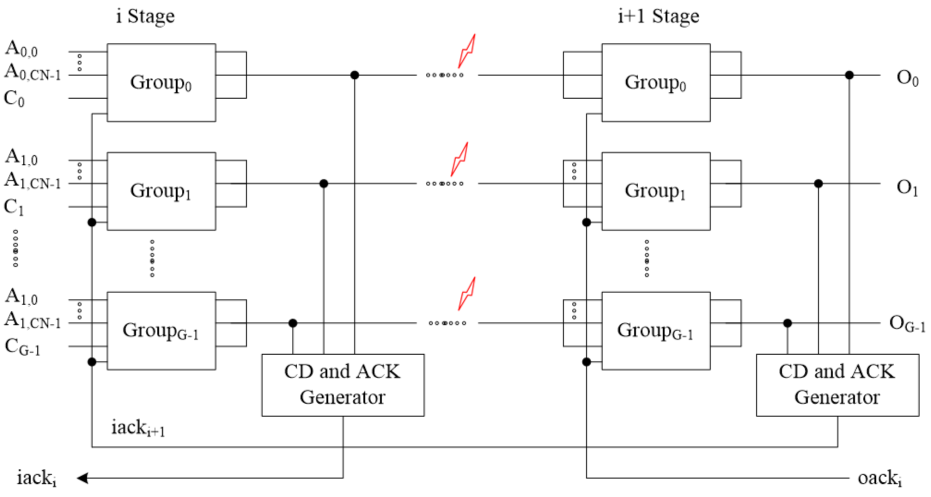


Figure 7. Architecture of the QDI pipeline module

The single group (G) is created using four 1-of-n DIRC channels (N = 4). The single group requires one check channel to create a single DIRC pipeline stage (G =1). Suppose four 1-of-n DIRC channels are divided into two groups (G = 2) in a DIRC pipeline stage, which requires an additional check channel for each group. The N 1-of-n DIRC channels are generally grouped into ‘G’ groups, each containing CN 1-of-n data channels. So, the fault-tolerant mechanism is achieved by adding one additional 1-of-n check word to each group of the DIRC pipeline stage. Each DIRC pipeline stage has one check word and CN code words. The (A_0, A_1, and C_{G-1}) and oack$_i$ are input to the QDI-pipeline stage. The (O_0, O_1, and O_{G-1}) and iack$_i$ are output in the QDI-pipeline stages, where ‘i’ describes the number of stages in the QDI pipeline using DIRC. These DIRC pipeline stages replace the primary QDI pipeline stages to enhance the fault tolerance in asynchronous NoC. The asynchronous links from NoCs are protected using DIRC-based sender and receiver units.

3. RESULTS AND DISCUSSION

The results of the fault-tolerant QDI pipeline module using the DIRC coding mechanism are presented in this section. The design modules are constructed using Verilog-HDL on the Xilinx ISE environment. The simulation results are carried out on the Modelsim simulator. The DRIC-based mechanism is constructed for 1-of-2 and 1-of-4 codes for the QDI-based pipelined module. The simulation results of the single DIRC stage with the fault-tolerant mechanism for 1-of-4 codes are depicted in Figure 8. The two 4-bit inputs (a and b), 4-bit fault, and 1-bit ACK (oack) signals are considered inputs. The 4-bit check words (a_o and b_o), code word (c_o), and 1-bit ACK (iack) signals are considered outputs. The inputs are given in random sequence, and fault-free outputs are generated in the DIRC stage.

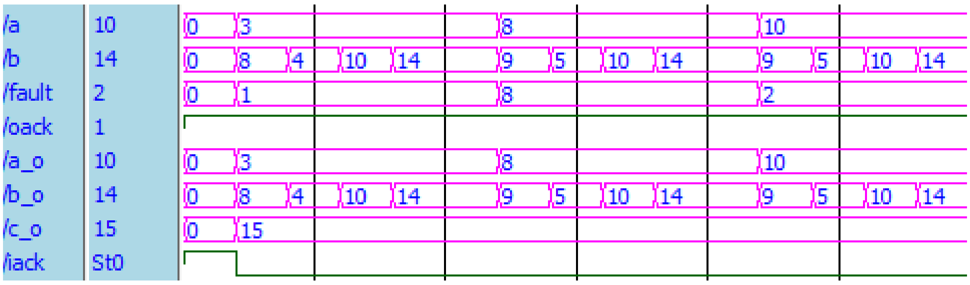


Figure 8. Simulation results of DIRC stage with fault-tolerant mechanism for 1-of-4 codes

This work uses the basic unprotected pipeline modules by Zhang *et al.* [24] for performance comparison. The area (slice LUTs (look-up-table)), combination delay, and power parameters are considered performance metrics for essential and QDI-based Pipeline Module (4-stage) realization. The performance analysis of the QDI-based Pipeline module (4-stage) using 1-of-2 and 1-of-4 codes are tabulated in Tables 1 and 2, respectively. The QDI pipeline with different data widths (N) is used to estimate the performance metrics (area, delay, and power) for both basic (unprotected) and DIRC pipelined modules.

Table 1. Performance analysis of the QDI based Pipeline module (4-stage) using 1-of-2 codes

N	G	Basic			DIRC (4-stage)		
		Area	Delay (ns)	Power (mW)	Area	Delay (ns)	Power (mW)
4	2	8	1.285	84	10	1.345	84
8	4	16	1.285	84	20	1.865	84
16	8	32	1.285	84	39	1.875	85
32	16	64	1.285	85	77	2.267	85
64	32	128	1.285	86	150	2.128	86
128	64	256	1.285	87	300	2.603	87

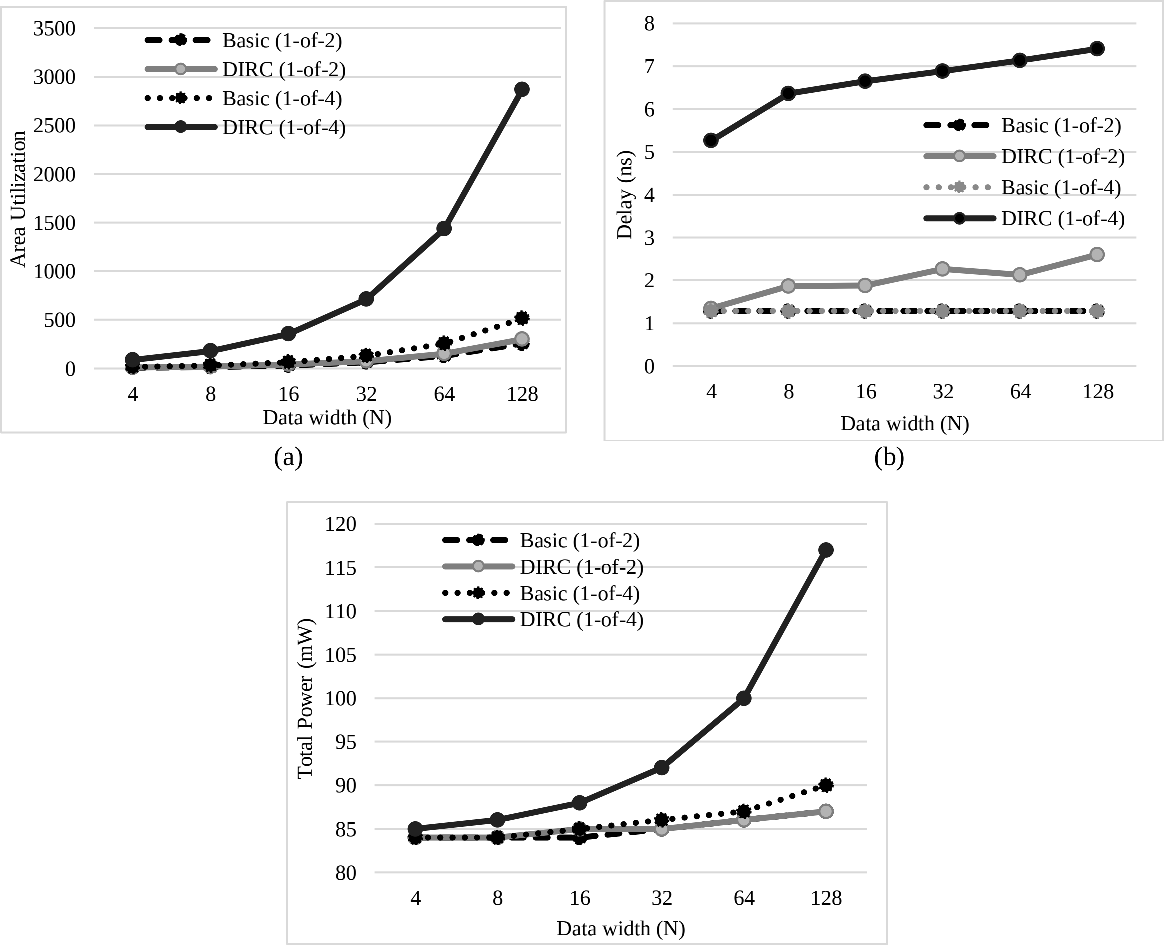
The basic (unprotected) pipelined module with 1-of-2 codes uses 256 slice LUTs with a delay of 1.285 ns by consuming a total power of 87 mW for 256 data width (N). Similarly, the DIRC pipelined module with 1-of-2 codes uses 300 slice LUTs with a delay of 2.603 ns by consuming a total power of 87 mW for 256 data width (N) Artix-7 FPGA. The basic (unprotected) pipelined module with 1-of-4 codes uses

512 slice LUTs with a delay of 1.285 ns by consuming a total power of 90 mW for 256 data width (N). Similarly, the DIRC pipelined module with 1-of-4 codes uses 2865 slice LUTs with a delay of 7.407 ns by consuming a total power of 117 mW for 256 data width (N) on Artix-7 FPGA.

Table 2. Performance analysis of the QDI based Pipeline module (4-stage) using 1-of-4 codes

N	G	Basic			DIRC (4-stage)		
		Area	Delay (ns)	Power (mW)	Area	Delay (ns)	Power (mW)
4	2	16	1.285	84	88	5.262	85
8	4	32	1.285	84	180	6.359	86
16	8	64	1.285	85	357	6.656	88
32	16	128	1.285	86	713	6.882	92
64	32	256	1.285	87	1434	7.132	100
128	64	512	1.285	90	2865	7.407	117

The graphical representation of the basic (unprotected) and DIRC pipeline module's performance against data widths is illustrated in Figure 9. The chip area in terms of slice and LUTs increases exponentially when the data width increases in both basic (unprotected) and DIRC pipeline modules for 1-of-2 and 1-of-4 codes, as shown in Figure 9(a). The combinational delay of 1.285 ns is the same for all data widths in basic (unprotected) pipelined modules with 1-of-2 and 1-of-4 codes. The combinational delay increases as data width increases in DIRC pipeline modules with 1-of-2 and 1-of-4 codes, are shown in Figure 9(b). The total power increases exponentially when the data width increases in both basic (unprotected) and DIRC pipeline modules for 1-of-2 and 1-of-4 codes, as shown in Figure 9(c).



(a) (b)

(c)

Figure 9. Performance realization of basic and QDI-based pipeline modules against data widths for: (a) area, (b) delay, and (c) total power

The computational performance of the fault-tolerant codes is realized using code rate. The code rate analysis of different fault-tolerable codes with proposed DIRC codes is tabulated in Table 3 and represented in Figure 10. The code rate of any fault-tolerant module is expressed using (1). The temporal redundancy delay-insensitive code (TRDIC) based asynchronous circuits are constructed to mitigate the single event issues [25]. The proposed DIRC codes decrease the code rate by 17.5% and 34% compared to TRDIC with 1-of-2 and 1-of-4 codes, respectively [30].

$$Code\ Rate\ (CR) = \frac{(\log_2 n)*CN}{n*(CN+1)} \tag{1}$$

The zero-sum codes are used for error correction in asynchronous global communication [31]. The proposed DIRC codes decrease the code rate of 13.15% and 23.25% compared to zero-sum codes with 1-of-2 and 1-of-4 codes, respectively [31]. The DI codes with single parity (SP) and hamming codes (HC) are used to improve fault resilience in asynchronous communication links [32]. The proposed DIRC codes decrease the code rate by 29.93% against SP with 1-of-2 codes and 9.34% against HC with 1-of-4 codes [32]. The fault-tolerant four-phase DI codes mitigate the TFs in asynchronous communication links [33]. The proposed DIRC codes decrease the code rate by 34% compared to DI codes in 1-of-2 and 1-of-4 codes [33]. The proposed DIRC codes decrease the code rate than different fault-tolerant methods and are suitable for use in QDI pipeline modules for error-free asynchronous communication.

Table 3. Code rate analysis of different fault-tolerable codes [30]–[33] with proposed work

Coding approach	QDI	Systematic	1-of-n codes	Code rate
TRDIC [30]	Yes	No	1-of-2	0.4
			1-of-4	0.5
Zero-Sum [31]	No	Yes	1-of-2	0.38
			1-of-4	0.43
DI with SP [32]	No	No	1-of-2	0.471
DI with HC [32]	No	No	1-of-4	0.364
DIC [33]	No	No	1-of-2	0.5
			1-of-4	0.5
This work	Yes	Yes	1-of-2	0.33
			1-of-4	0.33

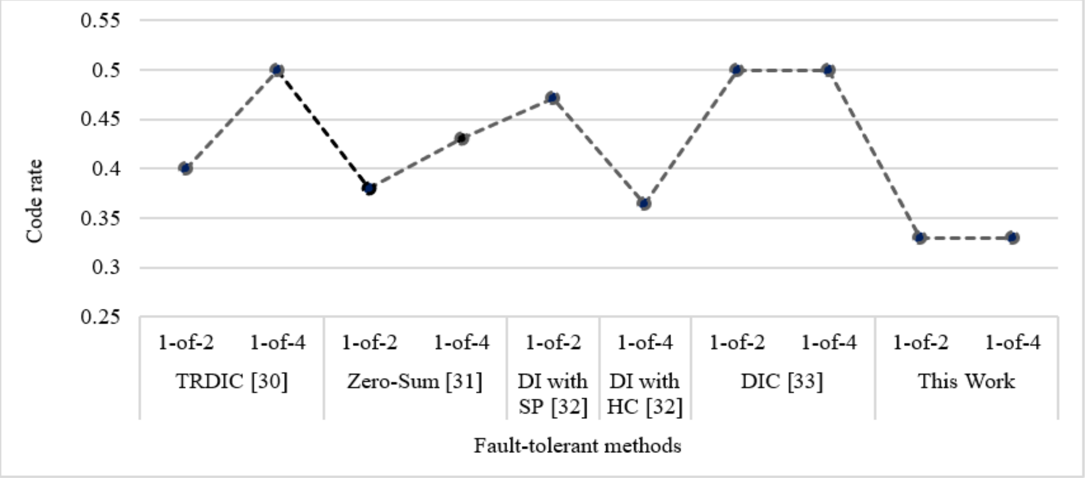


Figure 10. Code rate analysis representation of proposed work with current works [30]–[33]

4. CONCLUSION

The fault-tolerant QDI pipeline modules are designed using DIRC codes on the Artix-7 FPGA platform in this manuscript. The DIRC-based QDI pipeline modules can detect and correct single and multi-bit TFs. The DIRC stage is constructed of both 1-of-2 and 1-of-4 codes. The systematic, unordered DIRC code contains code and check words. Each DIRC stage has 1-of-n adders, error filters, completion detectors, and an acknowledge generator. This DIRC stage is incorporated in the QDI-pipeline module to tolerate the

TFs. The work is designed using verilog-HDL on the Xilinx environment and implemented on Artix-7 FPGA. The chip area, delay, and power parameters are discussed in detail concerning different data widths for 1-of-2 and 1-of-4-based QDI pipeline modules. The QDI-based pipeline module using 1-of-2 codes uses only 1% chip area, with a delay of 2.6 ns and power of 87 mW. Whereas the QDI-based pipeline module utilizes a chip area of 2% with a delay of 7.4 ns and power of 117 mW for 128-bit data width using 1-of-4 codes. The proposed work also compares the code rate with current fault-tolerant coding approaches with better reduction.

REFERENCES

[1] X. T. Tran, V. Beroulle, J. Durupt, C. Robach, and F. Bertrand, "Design-for-test of asynchronous networks-on-chip," in *2006 IEEE Design and Diagnostics of Electronic Circuits and systems*, 2006, vol. 2006, pp. 161–165, doi: 10.1109/DDECS.2006.1649605.

[2] S. Peng and R. Manohar, "Fault tolerant asynchronous adder through dynamic self-reconfiguration," in *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 2005, vol. 2005, pp. 171–178, doi: 10.1109/ICCD.2005.56.

[3] M. Imai and T. Yoneda, "Improving dependability and performance of fully asynchronous on-chip networks," in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, Apr. 2011, pp. 65–76, doi: 10.1109/ASYNC.2011.15.

[4] A. Alhussien, C. Wang, and N. Bagherzadeh, "A scalable delay insensitive asynchronous NoC with adaptive routing," in *ICT 2010: 2010 17th International Conference on Telecommunications*, 2010, pp. 995–1002, doi: 10.1109/ICTEL.2010.5478830.

[5] T. N. K. Jain, M. Ramakrishna, P. V. Gratz, A. Sprintson, and G. Choi, "Asynchronous bypass channels for multi-synchronous NoCs: A router microarchitecture, topology, and routing algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1663–1676, Nov. 2011, doi: 10.1109/TCAD.2011.2161190.

[6] W. J. Bainbridge and S. J. Salisbury, "Glitch sensitivity and defense of quasi delay-insensitive network-on-chip links," in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, May 2009, pp. 35–44, doi: 10.1109/ASYNC.2009.18.

[7] Y. Thonnart, P. Vivet, and F. Clermidy, "A fully-asynchronous low-power framework for GALS NoC integration," in *Proceedings -Design, Automation and Test in Europe, DATE*, Mar. 2010, pp. 33–38, doi: 10.1109/date.2010.5457239.

[8] S. B. Furber *et al.*, "Overview of the SpiNNaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013, doi: 10.1109/TC.2012.142.

[9] G. Zhang, W. Song, J. Garside, J. Navaridas, and Z. Wang, "An asynchronous SDM network-on-chip tolerating permanent faults," in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, May 2014, pp. 9–16, doi: 10.1109/ASYNC.2014.10.

[10] W. Song, G. Zhang, and J. Garside, "On-line detection of the deadlocks caused by permanently faulty links in quasi-delay insensitive networks on chip," in *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*, May 2014, pp. 211–216, doi: 10.1145/2591513.2591518.

[11] J. Pontes, P. Vivet, and Y. Thonnart, "Two-phase protocol converters for 3D asynchronous 1-of-n data links," in *20th Asia and South Pacific Design Automation Conference, ASP-DAC 2015*, Jan. 2015, pp. 154–159, doi: 10.1109/ASPDAC.2015.7058997.

[12] P. Vivet *et al.*, "A 4×4×2 homogeneous scalable 3D network-on-chip circuit with 326MFlit/s 0.66pJ/b robust and fault-tolerant asynchronous 3D links," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, Jan. 2016, vol. 59, pp. 146–147, doi: 10.1109/ISSCC.2016.7417949.

[13] Y. Ouyang, Q. Chen, X. Wang, X. Ouyang, H. Liang, and G. Du, "AFTER: asynchronous fault-tolerant router design in network-on-chip," *Journal of Circuits, Systems and Computers*, vol. 25, no. 6, p. 1650050, Jun. 2016, doi: 10.1142/S021812661650050X.

[14] G. Zhang, W. Song, J. Garside, J. Navaridas, and Z. Wang, "Handling physical-layer deadlock caused by permanent faults in quasi-delay-insensitive networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 11, pp. 3152–3165, Nov. 2017, doi: 10.1109/TVLSI.2017.2729081.

[15] G. Ait Abdelmalek, R. Ziani, and M. Laghrouche, "Testing and fault tolerance of secured circuits," *International Journal of Circuits, Systems and Signal Processing*, vol. 10, pp. 1–6, 2016.

[16] G. A. Abdelmalek, R. Ziani, and R. Mokdad, "Fault tolerance improvement of the secured circuits," *International Journal of Electrical, Electronics and Data Communication*, vol. 6, no. 9, pp. 50–52, 2018.

[17] G. A. Abdelmalek, R. Ziani, and R. Mokdad, "Security and fault tolerance evaluation of TMR-QDI circuits," *IET Information Security*, vol. 13, no. 3, pp. 213–222, May 2019, doi: 10.1049/iet-ifs.2018.5439.

[18] W. G. Ho, K. S. Chong, K. Z. L. Ne, B. H. Gwee, and J. S. Chang, "Asynchronous-logic QDI quad-rail sense-amplifier half-buffer approach for NoC router design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 196–200, Jan. 2018, doi: 10.1109/TVLSI.2017.2750171.

[19] Y. Thonnart, P. Vivet, S. Agarwal, and R. Chauhan, "Latency improvement of an industrial SoC system interconnect using an asynchronous NoC backbone," in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, May 2019, vol. 2019-May, pp. 46–47, doi: 10.1109/ASYNC.2019.00014.

[20] C. C. Nachiar, R. Poovendran, and D. Saraswathi, "Architectural exploration of heterogeneous NoC with fault tolerant capacity," in *2020 International Conference on System, Computation, Automation and Networking, ICSCAN 2020*, Jul. 2020, pp. 1–5, doi: 10.1109/ICSCAN49426.2020.9262453.

[21] M. T. Moreira and S. Giaconi, "Chronos link: a QDI interconnect for modern SoCs," in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, May 2020, vol. 2020-May, pp. 67–68, doi: 10.1109/ASYNC49171.2020.00018.

[22] M. Rashid *et al.*, "Fault-tolerant network-on-chip router architecture design for heterogeneous computing systems in the context of internet of things," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–20, Sep. 2020, doi: 10.3390/s20185355.

[23] R. Siddagangappa and D. K. Nayana, "Asynchronous NoC with fault tolerant mechanism: a comprehensive review," in *International Conference on Trends in Electrical, Electronics, Computer Engineering, TEECCON 2022*, May 2022, pp. 84–92, doi: 10.1109/TEECCON54414.2022.9854837.

[24] R. Bhat, M. R. Ansari, and R. Khanam, "Effect of integrated power and clock networks on combinational circuits," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 9, no. 3, pp. 242–248, Nov. 2020, doi: 10.11591/ijres.v9.i3.pp242-248.

[25] T. Ramesh and K. H. Abed, “An efficient multi-level cache system for geometrically interconnected many-core chip multiprocessor,” *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 11, no. 1, p. 93, Mar. 2022, doi: 10.11591/ijres.v11.i1.pp93-102.

[26] W. A. Al-Musawi, M. A. A. Al-Ibadi, and W. A. Wali, “Artificial intelligence techniques for encrypt images based on the chaotic system implemented on field-programmable gate array,” *IAES International Journal of Artificial Intelligence (IJAI)*, vol. 12, no. 1, pp. 347–356, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp347-356.

[27] T. Kavitha, G. Maheswaran, J. Maheswaran, and C. K. Pappa, “Optical network on chip: design of wavelength routed optical ring architecture,” *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 1, pp. 167–175, Feb. 2023, doi: 10.11591/eei.v12i1.4294.

[28] D. K. Nandalal and R. Bhakthavatchalu, “Design of programmable hardware security modules for enhancing blockchain based security framework,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 3178–3191, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3178-3191.

[29] G. Zhang, W. Song, J. D. Garside, J. Navaridas, and Z. Wang, “Transient fault tolerant QDI interconnects using redundant check code,” in *Proceedings - 16th Euromicro Conference on Digital System Design, DSD 2013*, Sep. 2013, pp. 3–10, doi: 10.1109/DSD.2013.11.

[30] J. Pontes, N. Calazans, and P. Vivet, “Adding temporal redundancy to delay insensitive codes to mitigate single event effects,” in *Proceedings - International Symposium on Asynchronous Circuits and Systems*, May 2012, pp. 142–149, doi: 10.1109/ASYNC.2012.26.

[31] M. Y. Agyekum and S. M. Nowick, “Error-correcting unordered codes and hardware support for robust asynchronous global communication,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 1, pp. 75–88, Jan. 2012, doi: 10.1109/TCAD.2011.2165070.

[32] J. Lechner, A. Steininger, and F. Huemer, “Methods for analysing and improving the fault resilience of delay-insensitive codes,” in *Proceedings of the 33rd IEEE International Conference on Computer Design, ICCD 2015*, Oct. 2015, pp. 519–526, doi: 10.1109/ICCD.2015.7357160.

[33] F. Huemer, J. Lechner, and A. Steininger, “A new coding scheme for fault tolerant 4-phase delay-insensitive codes,” in *Proceedings of the 34th IEEE International Conference on Computer Design, ICCD 2016*, Oct. 2016, pp. 392–395, doi: 10.1109/ICCD.2016.7753311.

BIOGRAPHIES OF AUTHORS

Renu Siddagangappa received a Bachelor’s degree in Electronics and Communication Engineering from Visvesvaraya Technological University, Belgaum, Karnataka, India, in 2011 and a Master of Technology degree in VLSI Design and Embedded Systems from Reva Institute of Technology and Management, Bangalore, India in 2014. She is pursuing a Ph.D. in the School of ECE at REVA University, Bangalore, and Karnataka, India. Her research interest includes VLSI design. She can be contacted at email: renusiddagangappa@gmail.com.

Nayana Dunthur Krishnagowda is working as a Professor School of ECE, REVA University, Bangalore, India. I obtained a Bachelor’s degree in Electronics and Communication Engineering from Mysore University, India, in 1994 and a Master of Technology degree in Digital Electronics and Advanced Communication from NITK, Surathkal, and Karnataka, India, in 1998. Ph.D. in Electronics and Communication Engineering, Jain University, Bangalore, Karnataka, India. I have more than 20 years of teaching experience. My research interest includes VLSI design, digital electronics, and signal processing. Published more than 20 research papers in reputed journals and published two Indian patents. She can be contacted at email: nayanadk@reva.edu.in.

Deepthi Tumkur Srinivas Murthy has obtained a doctoral degree in 2021 from VTU Belgaum. She has 14 years of teaching experience and one year of research experience. She has chaired many international and national conferences and published research journals. She works as an Associate Professor at the School of ECE, REVA University, and Bengaluru. Her area of interest is signal processing, biomedical signal processing, and image processing. She can be contacted at email: deepthimurthy@reva.edu.in.