Enhancing Integrated Circuit Reliability: Self-Repairing Hybrid Adder with Fault Localization

Shirisha[1], Saritha Kunamalla[2], Gangone Swathi[3]

[1,2,3]Assistant Professor, Department of ECE, Malla Reddy Engineering College and Management Sciences, Hyderabad, Telangana.

Abstract

With the ongoing process shrink of integrated circuits and the resulting increase in integration density, the reliability of integrated circuits deployed in the field has become a significant concern. This is especially critical for systems with high stakes, such as server-class computers and embedded systems, where concurrent error detection is of paramount importance. Addition, being the most fundamental arithmetic operation, plays a central role in various Very Large-Scale Integration (VLSI) designs. The reliability of adders is a crucial component in ensuring the overall reliability of these systems. While several techniques have been proposed for concurrent error detection in adders, they primarily focus on detecting errors resulting from single faults. Unfortunately, they do not guarantee the detection of erroneous outputs caused by multiple faults, potentially jeopardizing concurrent error detectability. If a second fault occurs before the detection of the first fault, the system's error detection capability can be compromised. Therefore, it is essential to detect the first fault before a second one occurs to ensure the reliability of adders. This research introduces a novel approach: a self-repairing hybrid adder with built-in fault localization. It combines the advantages of a ripple carry adder and a carry-select adder to mitigate delays and reduce area overhead, while also providing self-repairing properties. This enables the system to not only detect faults but also precisely locate them, facilitating effective self-repairing mechanisms.

Keywords: Adder, Hybrid Carry Select Adder, Self-Repairing Properties, Fault Localization, Integrated Circuit Reliability, Vivado ISE, Concurrent Error Detection.

1. Introduction

The design of an error-detectable hybrid carry-select adder with self-repairing properties is an important project in the field of digital circuit design. The goal of this project is to create a high-performance adder that can detect and correct errors and maintain its functionality even in the presence of faults. A hybrid carry select adder (HCSA) is a type of digital circuit that performs addition of two binary numbers. It consists of a combination of two different types of adders, namely the ripple carry adder (RCA) and the carry select adder (CSA). The RCA is used to generate a sum bit and a carry bit, while the CSA is used to select the carry bit based on the input operands. In order to ensure reliable operation of an HCSA, it is important to include error detection and correction mechanisms. One approach to achieving this is to design an error detectable HCSA with self-repairing properties. The design of this circuit involves the incorporation of error detection circuitry, which checks for errors in the output of the adder. This can be done using parity checking or other techniques. In addition, the HCSA must have self-repairing properties to correct any errors that are detected. The design of an error detectable HCSA with self-repairing properties requires careful consideration of the circuit architecture and the error detection and correction techniques that are used. By incorporate in these features into the design, the resulting circuit can provide reliable and accurate addition of binary numbers, even in the presence of faults or errors. Overall, this project design is implemented using a combination of hardware description language (HDL) and simulation tools. The design is verified using simulations and tested on an FPGA platform to evaluate its performance.

The key objectives of this project include Designing a hybrid carry-select adder that can detect and correct errors in the carry signals. Implementing the error-detection and self-repairing circuitry using redundant hardware. Evaluating the performance of the adder in terms of speed, area, and power consumption. Overall, the proposed design of an error-detectable hybrid carry-select adder with self-repairing properties has the potential to provide significant improvements in the reliability and performance of digital circuits, particularly in applications that require high-speed and fault- tolerant computation.

2. Literate Survey

Akbar, Muhammad Ali, Bo Wang, and Amine Bermak,).et,al (2020)[1], designed adder that reduces the transistor count by 115% to 76.76% as compared to the existing self-checking carry-select adders. Moreover, their design can detect and localize multiple faults. The fault-recovery was achieved by using the hot-standby approach in which the faulty module was replaced by a functioning module at runtime. Cheng, Fu-Teng, Yu-Cherng Hung, and Chiou-Kou Tung),et,al (2020)l[2], proposed circuit will automatically detect the error due to single-fault occurred in chip internal. Thus, the reliability of the circuit operation was improved. After the related EDA software simulations, the error self-checking capability and two-way encoding/decoding functions were successfully verified. After simulations, the code transformations were with self-checking and bit-expandable capabilities. Their circuit not only improves the operation speed but also reduces the duplicate logic hardware. The concept of carry-select adder (CSA) and add one circuit were extended in the study to reduce the signal time delay. Zandevakili, Hamed, and Ali Mahani).et,al (2020),[3], initiated a new reconfigurable application specific integrated circuit (ASIC) structure with real-time fault tolerance capability which repairs itself in two steps with a minimum delay. It includes some reconfigurable basic cells with self-repair capabilities which were used to implement the first recovery step. In the second step, the fault recovery process is done by replacing the faulty basic cell using a new reconfigurable. Liu, Lizheng, et al (2020),[4], considered an autonomous error-tolerant architecture for convolutional neural networks. An error-tolerant synapse was designed to discover the errors, an active evolution scheme was designed to handle unrecoverable errors and implement network reconfiguration. Their design was implemented on FPGA, and the experimental results showed that their architecture can realize effective error tolerance for convolutional neural network and had fast error recovery ability under the premise of ensuring the same recognition accuracy. Musala, Sarada, et al (2022)[5], intendend two new concurrent error repairable carry select adders (CSLAs). The proposed concurrent error repairable CSLA I detected and repaired the faults in the circuit. To give the exact location of the fault, CSLA II is proposed. The proposed concurrent error repairable CSLA II gives the bit position of the fault along with the detection and repairing of faults. It takes less delay, area and power compared to that of the proposed CSLA.Srinivasulu, Avireni, Jitendra Kumar Saini, and Renu Kumawat et,al (2021)[6], designed 12 transistors based, full adder circuit (12T-FAC) using Carbon Nanotube Field Effect Transistor (CNFET) technology . The proposed design based on CNFET provides high fault resistance towards transient, permanent faults and worked with least power, delay and power-delay product (PDP). Later, features like fault detection and correction circuit have added in 12TFAC. The final version of full adder circuit capable of correcting errors have used in designing applications like multipliers.Valinataj, Mojtaba and Axel Jantsch et,al (2020)[7], designed self-checking hierarchical multipliers with multiple error detection capability up to the size of 64 × 64 bits in such a way that the low-cost and highspeed designs were achieved with high multiple error detection probabilities. Experimental results based on analysis and simulation show that the proposed 32 × 32 and 64 × 64 multipliers based on each of Dadda or Braum structures as highspeed parallel and array multipliers, respectively, Su, Fei, Chunsheng Liu, and Haralampos-G. Stratigopoulos et,al (2023)[8],Covered the state-of-the-art in research and development of dependability and testability solutions for AI hardware including digital or analog implementations of Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs), used in accelerators and neuromorphic designs. Trends, challenges and perspectives were also discussed.Afzaal, Umar, et al (2022)[9], proposed SYFR, an evolutionary method for automated synthesis of increased fault salience digital circuits suitable for fine grained use. Test results for synthesis of up to 60 input circuits with SYFR are reported. SYFR can be repeatedly applied to a circuit to obtain various design trade-offs between fault-resilience and implementation costs. Their tolerance to faults is workload-aware. In addition, a novel population seeding mechanism to reduce the design space is introduced and experimentally validated.Bin Talib, Ghashmi H., and Aiman H. El-Maleh,et,al(2021)[10], proposed different fault tolerant carry look-ahead adder designs against single-bit soft errors based on double modular. redundancy DMR and hybrid fault-tolerant schemes. In DMR based design, they combined a partial hardware redundancy scheme with a protected element to achieve full soft error masking, while in the hybrid design, they employed a partial hardware redundancy combined with a parity prediction scheme to improve fault tolerance capability of the adder while reducing area overhead. Their results showed that the proposed design schemes take precedence over other schemes in terms of failure rate, area overhead and delay overhead. Tilak Raju, D., and Y. Srinivasa Rao et,al(2022)[11], demonstrated that the approximate multiplier (AM) may be the key to improving hardware efficiency and speeding up multiplication operations. Their comprehensive literature on the entire development history and processes of AM findings, error analysis, and applications was missed in one location. As a result, their article outlines the history and advancements of AM architectural design and prospective study topics for future advancements. Their thorough study also discussed the methods researchers utilized to enhance AM design and provided an edge over other mentioned AM.

Jha, Rahul Kumar, Nishi Pandey, and Abhishek Agwekar et,al (2022)[12], proposed a new CRC based code-based algorithm has been proposed, which does not use any response change registers (LFSR). The Channel coding was often installed to obtain adequate reception quality on a portable communication communication transceiver to combat channel deterioration due to interference between signals, multi-channel scattering, and hot noise caused by electrical circuits. High speed and high-speed computer coding and decoder software can be useful in the field of communication. Rudposhti, Maytham Allahi, and Mojtaba Valinataj et,al (2021)[13], initiated some Square-RooT (SQRT) Carry Select Adder (CSLA) architectures including a high speed design, a design with the lowest area compared to previous CSLAs, and two hybrid designs. In addition, two hybrid CSLAs was proposed by exploiting the benefits of both proposed CSLA architectures.Bin Talib, Ghashmi H., and Aiman H. El-Maleh et,al(2021)[14], proposed different fault tolerant carry look-ahead adder designs against single-bit soft errors based on double modular redundancy DMR and hybrid fault-tolerant schemes. In DMR based design, they combined a partial hardware redundancy scheme with a protected Clement to achieve full soft error masking, while in the hybrid design, they employed a partial hardware redundancy combined with a parity prediction scheme to improve fault tolerance capability of the adder while reducing area overhead. They used two different circuits for merging the partial hardware redundancy into the carry generation logic and to achieve higher fault masking rate and lower area overhead in comparison with existing approaches. Simulation result showed that the proposed design schemes take precedence over other schemes in terms of failure rate, area overhead and delay overhead. Bin Talib, Ghashmi H., Aiman H. El-Maleh, and Sadiq M. Sait et,al (2018)[15], demonstrated that various fault tolerant designs required to attain highly reliable adders were depicted. In addition to the review of different fault tolerant designs of carry look-ahead and carry-select adders, their paper also gave their comparative assessment in terms of fault tolerance, area overhead, and weaknesses. The described fault tolerant design was classified based on their ability to detect or correct faults and the employed redundancy scheme.

3. Proposed Mythology

The time required for CSeA to compute the lowest bits is more than the required time for RCA. This additional delay is caused by the MUX. Therefore, if a simple RCA for initial bits is employed, the design will be more efficient in terms of hardware and time-delay. The complexity will also be reduced with the use of RCA as the beginning block. This is why in the proposed HA design, the least significant bits are computed using RCA, while a single RCA-based CSeA is used for computing the higher bits, as shown in Figure1.

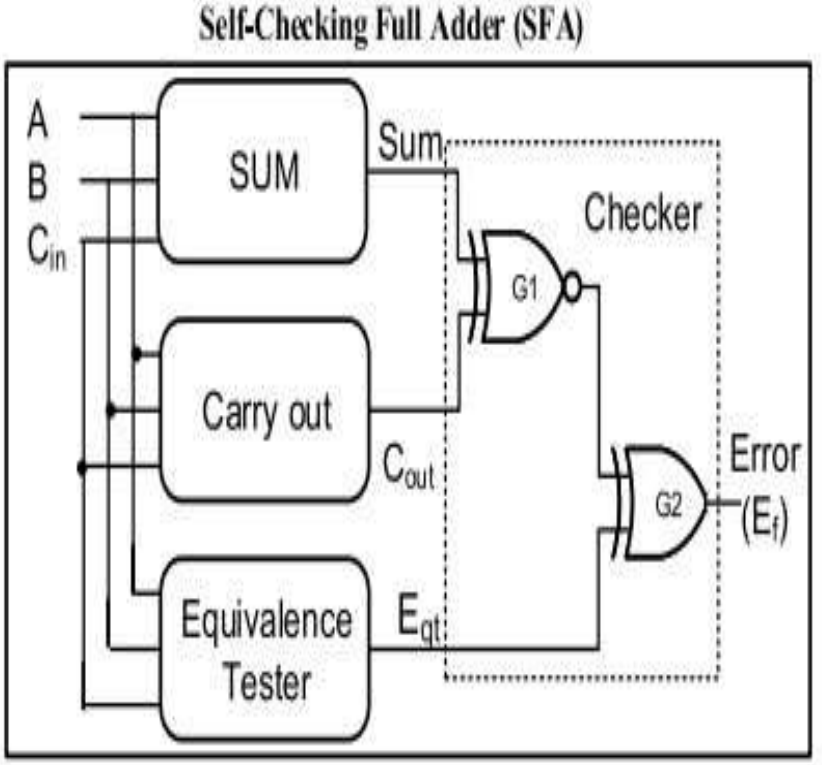


Figure 1. Block Diagram of Self Checking Full Adder

In addition to this, the proposed HA design follows the square root topology because a linear CSeA design has similar time delay as that of simple RCA.Therefore, sub-linear delay approach has been considered to balance the delay path by diving the adder in to blocks where the size of the block increases linearly from m, m+1, . . . , m+l. Fig 4.1b Block diagram of Carry select adder(CSeA).It should be noted that RCA-Block (RBL) is the fundamental building block of RCA, whereas, the CSeA constitutes of two fundamental blocks that is the initial block (INL) and the Adder Block (ABL), as shown in Figure 2.

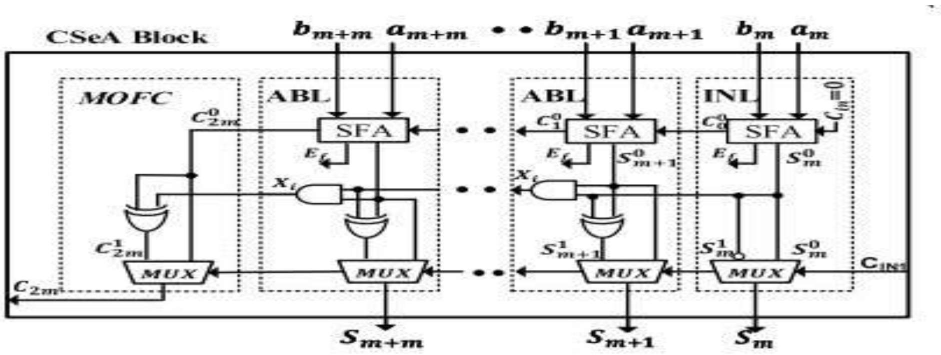


Figure 2. Block diagram of Carry select adder (CSeA)

The reason for having two fundamental blocks for constructing CSeA is because of the basic principle of single RCA based. CSeA design which states that: Except for the least significant bit which are always complement to each other, the Sum bit computed for complement value of initial Cin will also be complement to each other if all the lower Sum bits are equal to logic 1. Initial block (INL) is therefore responsible for generating the least significant Sum bit by taking the complement of the Sum bits generated at initial Cin = 0. All the other Sum bits will be generated by using the Adder Block (ABL) in which the AND gate is used to determine the status of the previous SUM bits computed for Cin = 0 while the XOR gate generates the corresponding SUM bit for Cin = 1 by considering the status of the previous SUM bits. The number of ABL used for designing CSeA block is equal to the (size_of _the_CSeA_block − 1). In Figure 3, the partial Sum and Cout bit is represented by S respectively, where j indicates the initial Cin and i indicates the bit number. The fault is indicated by the error signal Ef.The final Cout will be generated by using the Module of Final Cout (MOFC). The Cout generated by the MOFC after each CSeA block will be treated as an actual Cin for the next CSeA Block. Whereas, the Cout of RCA block is used as an actual Cin for the first CSeA block.

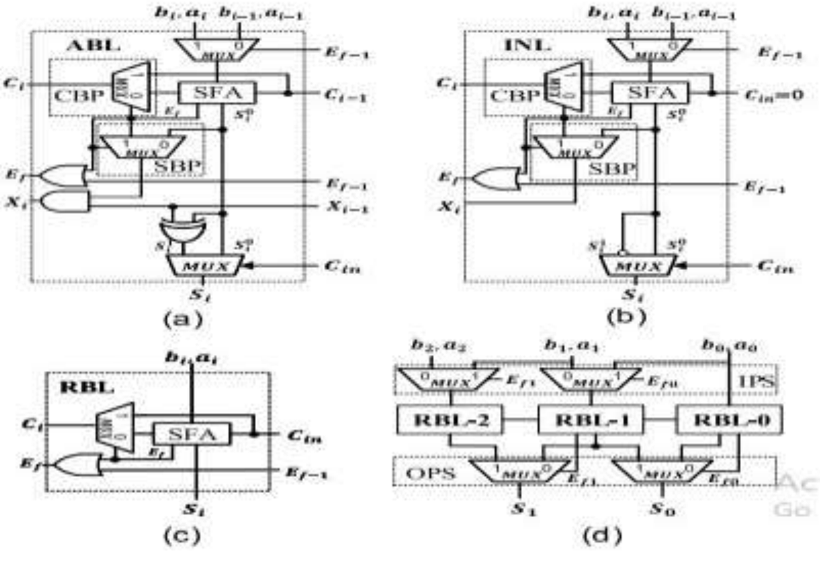


Figure 3. Fundamental blocks for constructing the self-repairing HA using (a) ABL (b) INL (c) RBL for CSeA and RCA block along with the shifting approach (d) for input and Sum-out in case of fault detection.

3.1 Fault detection and localization

Fault localization is achieved by using the approach of self-checking, independent of the propagated carry. In [11], a self-checking full adder was presented which can detect a fault based on its internal functionality and is independent of the propagated carry. The relationship between input and output bits of full adder was utilized for self-checking. Consider a full adder with inputs A, B, Cin, and the outputs Sum, Cout, as shown in Figure 3. The fault will not be indicated until.

Property 1 remains valid for that full adder: It can be observed from Fig.3, that the self-checking full adder can be designed with the expense of an extra Equivalence Tester (Eqt) bit, which is required to indicate the relationship of the input bits.

$$\begin{aligned}
&\mathit{if}\ (A == B == C_{in})\\
&\mathit{Then}\ (Sum \oplus C_{out} == 0)\\
&\mathit{Else}\ (Sum \oplus C_{out} == 1)
\end{aligned}$$

It can be observed from Fig. 3, that the self-checking full adder can be designed with the expense of an extra Equivalence Tester (Eqt) bit, which is required to indicate the relationship of the input bits. The Eqt will be equal to 1 if all input bits are equal and vice versa. Hence, the following three equations from Eq. (1) to (3) need to be implemented for designing a self-checking and fault localized adder. Since the goal of this design is to reduce the area overhead without compromising the reliability, Equations Eq. (1) to (3) which are used for designing a self-checking and fault localized full-adder need to be implemented with minimum transistor count.

$$\begin{aligned}
Sum &= \overline{A \oplus B \odot C_{in}}\\
C_{out} &= \overline{(A \odot C_{in}) \cdot \overline{A} + (A \oplus C_{in}) \cdot \overline{B}}\\
Equ.Tester(E_{qt}) &= \overline{(A \oplus B) + (A \oplus C_{in})}
\end{aligned}$$

A high speed and area efficient full adder design is found in [20]. However, this approach cannot be adopted completely because of the logic sharing between Sum and Cout, due to which the probability of common mode failure increases. Therefore, the equation and transistor level implementation of Cout has only been adopted from their design. The final implementation of Eq. (1) to (3) using pass transistor-based approach is shown in Figure 2.

3.2 Self-repairing approach

A hot-standby approach has been adopted for fault recovery. In this approach, if the fault is detected in any of the full adders, the generated error signal will shift the input bits such that the faulty adder will not be used for computation. The main challenge in doing this shift operation is the carry chain which is linked between each consecutive full adder, and the Xi bit which is indicating the status of all previous Sum bits in each CSeA block, as shown in Fig. 1(b). The problem of carry chain has been resolved by making Cout to be dependent on error signal Ef of the SFA. In case of fault, the Cout (i.e. Ci) will be equal to Cin (i.e. Ci−1).

$$C_{out} = \begin{cases} C_{out}, & \text{if } E_f == 0 \\ C_{in}, & \text{otherwise} \end{cases}$$

Since, Xi indicates the status of all previous Sum bits computed for initial Cin = 0, if any previous Sum bit is equal to 0 then Xi will be 0, else it will be 1.The Sum bit of each ABL is dependent on the previous value of Xi , therefore in case of fault detection the value of Xi should not be updated for the next ABL block. In order to achieve this, the Error signal (Ef) has been used to replace the SUM bit in case of fault detected, because Xi is produce through an AND gate and if the current Sum bit value is set to logic 1 then the previous value of Xi (i.e. Xi−1) will be propagated.

$$X_i = \begin{cases} X_i, & \text{if } E_f == 0 \\ X_{i-1}, & \text{otherwise} \end{cases}$$

Note that Xi is only propagated to the ABLs present in each CSeA block along with the next MOFC block, and it will not be propagated to the next CSeA block because each CSeA block is independent of the previous block. In order to accommodate all these changes, the fundamental blocks for extending the CSeA to an n-bit CSeA that is ABL and INL in Fig. 1(b), has been modified, as shown in Fig. 3(a) and (b), respectively. Since the carry-chain exists in RCA block as well, the fundamental block of RCA (i.e. RBL) has also been modified, as shown in Fig. 3(c). However, the OR gate present in the modified RBL is not applicable for the first full adder of RCA because of the absence of any previous error signal. The final SUM bit generated by the adder also needs to be shifted Note that Xi is only propagated to the ABLs present in each CSeA block along with the next MOFC block, and it will not be propagated to the next CSeA block because each CSeA block is independent of the previous block. In order to accommodate all these changes, the fundamental blocks for extending the CSeA to an n-bit CSeA that is ABL and INL in Fig. 1(b), has been modified, as shown in Fig. 3(a) and (b), respectively. Since the carry-chain exists in RCA block as well, the fundamental block of RCA (i.e. RBL) has also been modified, as shown in Fig. 3(c). However, the OR gate present in the modified RBL is not applicable for the first full adder of RCA because of the absence of any previous error signal. The final SUM bit generated by the adder also needs to be shifted.

4. Results and Discussion

The simulation results will done by using in Vivado ISE.The timing, power and synthesis reports listed below.

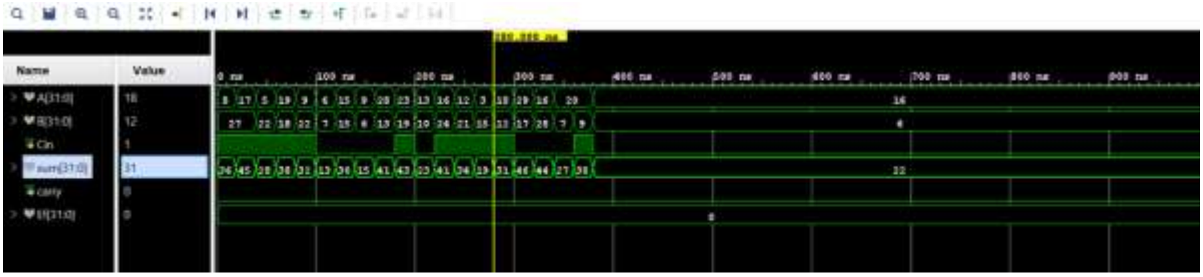
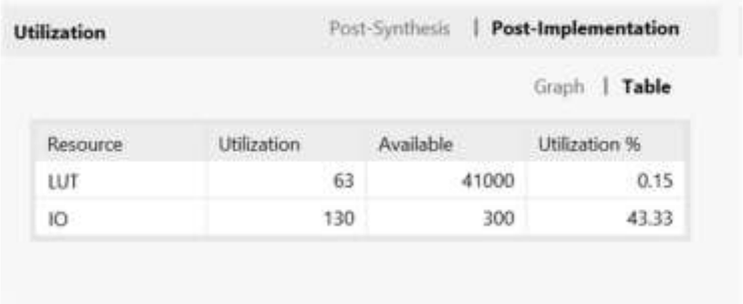
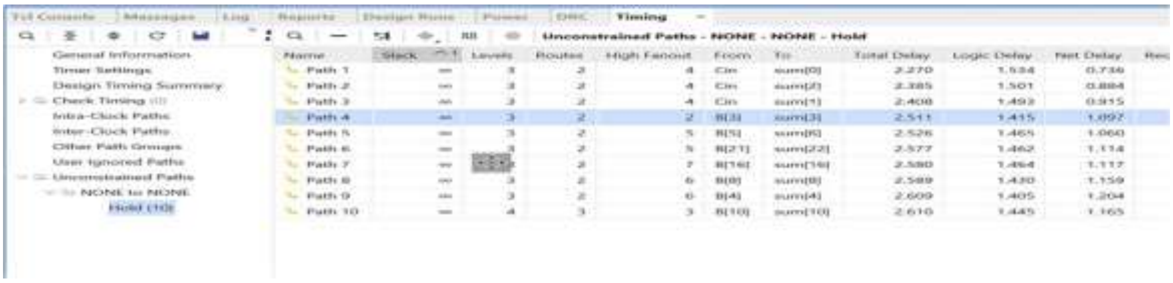


Figure 4. Simulation Output



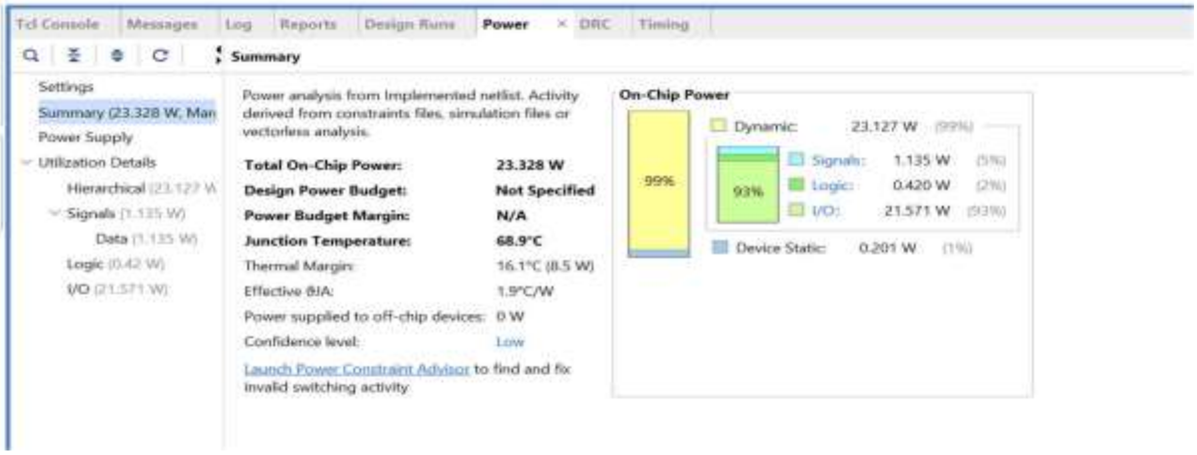
Resource	Utilization	Available	Utilization %
LUT	63	41000	0.15
IO	130	300	43.33

Figure 5. Area summary.



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	3	2	4	Cin	sum[0]	2.270	1.534	0.736
Path 2	∞	3	2	4	Cin	sum[2]	2.385	1.501	0.884
Path 3	∞	3	2	4	Cin	sum[1]	2.408	1.493	0.915
Path 4	∞	3	2	2	B[3]	sum[3]	2.511	1.415	1.097
Path 5	∞	3	2	5	B[5]	sum[6]	2.526	1.465	1.060
Path 6	∞	3	2	5	B[21]	sum[22]	2.577	1.462	1.114
Path 7	∞	[illegible]	2	7	B[16]	sum[16]	2.580	1.464	1.117
Path 8	∞	3	2	6	B[8]	sum[8]	2.589	1.430	1.159
Path 9	∞	3	2	6	B[4]	sum[4]	2.609	1.405	1.204
Path 10	∞	4	3	3	B[10]	sum[10]	2.610	1.445	1.165

Figure 6. Delay summary.



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**23.328 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**68.9°C**
Thermal Margin:	16.1°C (8.5 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

Dynamic:	23.127 W	(99%)
Signals:	1.135 W	(5%)
Logic:	0.420 W	(2%)
I/O:	21.571 W	(93%)
Device Static:	0.201 W	(1%)

Figure 7. Power summary.

Table 1. Comparison Table

Method	Area	Power(W)	Delay(ns)
Existing Method	82	26.8	3.521
Proposed Method	63	23.28	2.609

5. Conclusion

A self-checking and repairing HA design has been presented with reduced area overhead and increased coverage as compared to the previously presented design approaches. The HA design follows the architecture of single RCA based CSeA with the only difference of initial bits, which has been computed using RCA. A run-time self-repairing approach has been adopted by using hot-standby topology. The proposed design can be extended easily to any size by using fundamental block design presented in the paper. The proposed design with self-checking has been compared with the previously reported self-checking CSeA in terms of area and fault coverage. Moreover, due to the distributed self-checking mechanism, the proposed approach can detect and localized multiple faults with the condition that a single module should have single fault at a time.

References

[1]. N. Mehdizadeh, M. Shokrolah-Shirazi, and S. G. Miremadi, ‘‘Analyzing fault effects in the 32-bit OpenRISC 1200 microprocessor,’’ in Proc. 3rd Int. Conf. Availability, Rel. Secur., 2008, pp. 648–652.

[2]. A. Meixner, M. E. Bauer, and D. Sorin, ‘‘Argus: Low-cost, comprehensive error detection in simple cores,’’ in Proc. 40th Annu. IEEE/ACM Int. Symp. Microarchit, Dec. 2007, pp. 210–222.

[3]. H. Baig and J.-A. Lee, ‘‘An island style routing compatible fault-tolernat FPGA architecture with self-repairing capabilities,’’ in Proc. Field Program. Technol. (FPT), 2012, pp. 301–304.

[4]. P. E. Dodd and L. W. Massengill, ‘‘Basic mechanisms and modeling of single-event upset in digital microelectronics,’’ IEEE Trans. Nucl. Sci., vol. 50, no. 3, pp. 583–602, Jun. 2003

[5]. J. E. Smith and P. Lam, ‘‘A theory of totally self-checking system design,’’ IEEE Trans. Comput., vol. C-32, no. 9, pp. 831–844, Sep. 1983.

[6]. Penchalaiah, Usthulamuri, and VG Siva Kumar. "Design and Implementation of Low Power and Area Efficient Architecture for High Performance ALU." Parallel Processing Letters 32.01n02 (2022): 2150017.

[7]. S. V. G. Kumar, M. Vadivel, U. Penchalaiah, P. Ganesan and T. Somassoundaram, "Real Time Embedded System for Automobile Automation," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-6, doi: 10.1109/ICSCAN.2019.8878820.

[8]. A. G. Ganek and T. A. Corbi, ‘‘The dawning of the autonomic computing era,’’ IBM Syst. J., vol. 42, no. 1, pp. 5–18, 2003.

[9]. T. Koal, D. Schiet, and H. T. Vierhaus, ‘‘A concept for logic self repair,’’ in Proc. 12th Euromicro Conf. Digit. Syst. Design, Architectures, Methods Tools, Aug. 2009, pp. 621–624.

[10]. V. Ocheretnij, D. Marienfeld, E. S. Sogomonyan, and M. Gossel, ‘‘Self checking code-disjoint carry-select adder with low area overhead by use of add1-circuits,’’ in Proc. 10th IEEE Int. On-Line Test. Symp., Jul. 2004, pp. 31–36.

[11]. D. P. Vasudevan, P. K. Lala, and J. P. Parkerson, ‘‘Self-checking carry-select adder design based on two-rail encoding,’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 12, pp. 2696–2705, Dec. 2007.

[12]. M. A. Akbar and J.-A. Lee, ‘‘Comments on ‘self-checking carry-select adder design based on two-rail encoding,’’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 7, pp. 2212–2214, Jul. 2014.

[13]. M. A. Akbar and J.-A. Lee, ‘‘Self-checking carry select adder with fault localization,’’ in Proc. Euromicro Conf. Digit. Syst. Design (DSD), Sep. 2013, pp. 863–869.

[14]. N. Kito and N. Takagi, ‘‘Concurrent error detectable carry select adder with easy testability,’’ IEEE Trans. Comput., vol. 68, no. 7, pp. 1105–1110, Jul. 2019.

[15]. A. Majumdar, S. Nayyar, and J. S. Sengar, ‘‘Fault tolerant ALU system,’’ in Proc. Int. Conf. Comput. Sci. (ICCS), Sep. 2012, pp. 255–260.