jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2019, VOL. 65, NO.2 , PP. 313-318
Manuscript received September 16, 2018; revised April, 2019. DOI: 10.24425/ijet.2019.126316

A New Approach of an Error Detecting and Correcting Circuit by Arithmetic Logic Blocks

S. Kavitha, Fazida Hanim Hashim, and Noorfazila Kamal

***Abstract*—This paper proposes a unique method of an error detection and correction (EDAC) circuit, carried out using arithmetic logic blocks. The modified logic blocks circuit and its auxiliary components are designed with Boolean and block reduction technique, which reduced one logic gate per block. The reduced logic circuits were simulated and designed using MATLAB Simulink, DSCH 2 CAD, and Microwind CAD tools. The modified, 2:1 multiplexer, demultiplexer, comparator, 1-bit adder, ALU, and error correction and detection circuit were simulated using MATLAB and Microwind. The EDAC circuit operates at a speed of 454.676 MHz and a slew rate of -2.00 which indicates excellence in high speed and low-area.**

***Keywords*—EDAC, ALU, speed, block reduction, power, slew rate**

I. Introduction

IN the field of information era, the error detection and correction systems are identifying roots of errors. Early communication channels were extremely uncertain and extremely noisy where input signals transmit from one end to receiving end [1]. The error detection and correction circuit have to encode the data information in redundant formats known as error codes. Coded inputs in either binary or hexadecimal are supposed as errors if the codes do not match in a operation [2]. The common magnetic core technology has to be less predictable than the early integrated circuit (IC) memories. So, the IC design/designer were quick to incorporate single error correction and double error detection (SEC/DED) codes into their design. The error codes are protected the storage and memory processor to data transfers in a data processing system. The decoding of the corrupted signal and encoding the corrected signal is indispensable due to common codes not utilized in arithmetic operations [3].

One way to protect the arithmetic computation against fault induced errors is to use comparison of the two results which describes a specific fault and error detection. Another approach is to use triplication with two out of three voting on the three results that describes a single fault masking and error correction [4]. The new approach of arithmetic computations designs is adopted to identify the duplication way for a single fault and error detection. The decoding logic is reproduced along with the Arithmetic Logic Unit (ALU), to establish that a single fault in the decoder doesn’t go unexplored. The encoder stands a significant element whose failure will contribute to undetected faults. Such an arrangement, pointed out to as a self-checking, leads to error detection by the checker associated with the memory subsystem or later when the erroneous stored value is adopted as an input to the ALU. An undetected mismatch would demand at least two faults in separate blocks. Arithmetic error detecting codes are represented in terms of the arithmetic weights of detectable errors and assign us to represent arithmetic operations on coded operands directly. It also has two classes of codes called product codes and residue codes [5].

The new technique of circuit arrangement is carrying out the arithmetic circuit for error detection and correction in this investigation. Several circuit design approaches were carried out in this investigation for further circuit optimization in terms of speed or propagation delay, power dissipation, and area. For data transfer operation in communication, the per-bit error probability of around 10 to the power of 10 is acceptable [6]. However, at a rate of many millions of arithmetic operations per-second, such an error probability in computations can lead to several bit-errors per second. Thus, this investigation examines the key method that can improve the robustness and accuracy of the arithmetic circuit [7]. The main objective of this paper is to improve the logic blocks for detecting and correcting errors at the basic arithmetic operation level. The four critical factors in circuit design such as speed, propagation delay, power, slew rate, and several transistors will be optimized without giving in on accuracy.

II. DESIGN AND IMPLEMENTATION

The proposed error detection and correction (EDAC) circuit comprises an ALU unit and a comparator circuit. A register circuit is designed employing a D-flip-flop and it is carry out a serial-in-serial out process. The register A has been not an input system which is memory/original channel. The ALU circuits consist of two input AND gate, OR gate, 2:1 mux and a full adder circuit. At the last stage, the comparator circuit establishes the design of establishing the bit error. This investigation provides the design method for error detection and correction by applying arithmetic logic unit (ALU) circuits from previous investigations [8] [2]. In most of these arguments, the demands of low power consumption, lower occupying area and high bit throughput. The existing researches primarily concentrate on the circuit-level or transistor-level design, it can extend to reduce the power dissipation in digital integrated circuits [3]. The different type power consumption and various design considered for new design, which new design strategies will be brought in to reduce the power dissipation.

A. 2-to-1 Multiplexer Circuit

The design of 2:1 multiplexer is used various technique. The Proposed circuit used PTL logic that logic 0 would connect A to the output while a logic value of 1 would connect 1 to the

Authors are with Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia (e-mail: kavithapari@gmail.com).

output. The design of higher number multiplexer circuit, the selector pins is corresponding to [$\log_2$ (N-1)], where n is the number of inputs. According to look up table, the higher order multiplexer circuit is produced to output according to K-Map method, which identify all the cases when Y is equal to 1. A clear awareness of this 2-to-1 multiplexer would need 2 AND gates, an EX-OR gate, and a NOT gate. The 2:1 multiplexer or data selector allows one and only one of the 2^n sources which are to be logically connected to a common destination f at a time. Here the source Ii, where i = 1,2,….2^n is transferred to the output if the input address is i. The address signal is the binary integer in which control the selection of the desired source.

$$f = \bigvee_{i=0}^{2^n-1} I_{i+1} \cdot \underbrace{S_1^{i1} S_2^{i2} \dots\dots S_n^{in}}_{Address} \qquad (1)$$

Where $S_j^{ij} = \begin{cases} S_j, \text{if } I_j = 0 \\ S_j, \text{if } I_j = 1 \end{cases}$

The block level design considers conventional circuits. The designed circuits are carried out in design system, simulate and justified by MATLAB Simulink. The multiplexer 2 to 1 circuit that implements the function of depends upon the selection input. multiplexer circuit design method is reducing the number of transistor in design and reduced the size of integrated circuit. This circuit has two input selectors and one output. There are four possible bit string patterns which can be derived into the input selector such as 00, 01, 10 and 11.

The conventional CMOS 2:1 multiplexer has 2 AND gate, 1 NOT gate and 1 OR gate which gives the result Y = A$\bar{S}$+BS. The conventional circuit CMOS circuit used 19 transistor used in TTL configuration. This paper rearranges the conventional 2:1 multiplexer blocks, which are reduced the 3 NOT gates (6 transistors) by PTL logic So the new approach block design method is reduced 6 transistors in 2:1 multiplexer circuits. The new approach block design method gives superior results than the conventional circuit that will be discussed further in the result and discussion section.

B. *Demultiplexers*

Demultiplexers are commonly characterized as 1–to–2^n devices. The design of demultiplexer circuit has one input and 2^n output lines. The proposed block diagram requires only 1–to–4 demultiplexer. The designer added an enable–high to the demultiplexer for getting sensible output. When selection input is disable (logical 0) and all outputs are 0. When selection input is enable , the selected input gets the output, X. Remember, that X can have a value of either 0 or 1. The conventional Demux circuit was designed with enable logic. The 1:4 Demux circuit designed with enable input used 4-input AND gate which is utilized more transistors in the TTL method. The modified Demux circuit is designed without the enable input due to the data input is a chosen value, which reduces at least 8 transistors in TTL configuration. The design of the demultiplexer circuit achieves the AND logic gate and NOT logic gate to understand the demultiplexing function. Finally, the Boolean expression of the demultiplexer is given in equation (2).

The de-multiplexer outputs

$$f_i = S_1^{i1}, S_2^{i2} \dots\dots\dots S_n^{in} \qquad (2)$$

Where $S_j^{ij} = \begin{cases} S_j, \text{if } I_j = 0 \\ S_j, \text{if } I_j = 1 \end{cases}$

C. *Comparator Circuit*

The comparator circuit is a function, which o two inputs have to equal each other [10]. The designer has to implement the EX-NOR logic gate and the AND logic gate to design this comparator circuit. The logic gate model is modified form the existing logic model and implemented into TTL configuration structure.

The conventional comparator gate is consisting of 7 logic gates. Our error correction circuits is being to be used only A=B decision circuit. According to the Boolean method, the conventional circuits have modified and used only 3 logic blocks. So the modified comparator circuit reduced 3 blocks which is almost reduced to 6 transistors by PTL configuration design. According to VLSI design method, the modified is reduced occupying area which gives improved speed and low power consumption

D. *Full adder*

The conventional full adder circuit used two EX-OR gate for sum operation and 3 AND gate and 1 OR gate used for carry circuits. The existing literature is used a bit slice technique which reduced the one EXOR logic blocks. This paper also used the pass transistor technique and reduced, sum part of (AÅB) directly fed into the carry circuit.

E. *1 bit ALU circui*

The conventional ALU circuit consists of AND gate, OR gate, Full adder and MUX circuit. This paper illustrates the main ideas relating to the arithmetic error correcting codes by bi-residue technique, which represents a triple number. Encoding for the class of residue codes is similar to single residue codes, except two residues must be computed. This three-channel computation strategy can be generalized to n channels to permits the tolerance of more faults [11]. However, the cost overhead of a higher degree of replication becomes prohibitive. Table I shows the arithmetic operations that are obtainable by controlling the value of Y with the two selection inputs S1 and S0. If the inputs from B are ignored and we insert all 0s at the Y inputs, the output sum becomes G = A + 0 + Cin.

TABLE I
FUNCTION TABLE FOR ARITHMETIC CIRCUIT

Select	Select	Input	G = (A1Y1Cin)	G = (A1Y1Cin)
S1	S0	Y	Cin = 0	Cin = 1
0	0	All 0s	A	A+1
0	1	B	A+B	A+B+1
1	0	B’	A + B’	A+B’+1
1	1	All 1s	A – 1	G = A

According to table I, The logic operations of arithmetic circuit manipulate the bits of the operands by treating each bit in a register as a binary variable. There are four commonly used logic operations such as AND, OR, XOR and NOT.

TABLE II
FUNCTION OF ALU

	S_0	Output	Operation
0	0	G = A^ B	AND
0	1	G = A OR B	OR
1	0	G = A XOR B	XOR
1	1	G = A'	NOT

The logic circuit can be combined with the arithmetic circuit to produce an ALU. The configuration for one stage of the ALU is shown in Fig 1. The outputs of the arithmetic and logic circuits in each stage are applied to a 2-to-1 multiplexer with selection variable S2. When S2 = 0, the arithmetic output is selected and when S2 = 1, the logic output is selected. This diagram shows just one typical stage of the ALU, the circuit must be repeated n times for an n-bit ALU. The ALU logic has designed is not as simple as it could be and has a fairly high number of logic levels, contributing to propagation delay in the circuit. With the use of logic simplification software, we can simplify this logic and reduce the delay. For example, it is quite easy to simplify the logic for a single stage of the ALU.

The ALU consists of AND gate, OR gate 2:1 multiplexer and a full adder circuit. In this arrangement, full adder AND, OR gate results are important which is to be considered by selection input. According to design the logic inputs are positively enabled. The output of the ALU circuit shown in Figure 6 which is clearly indicating there is no delay power dissipation and logic transition effect.

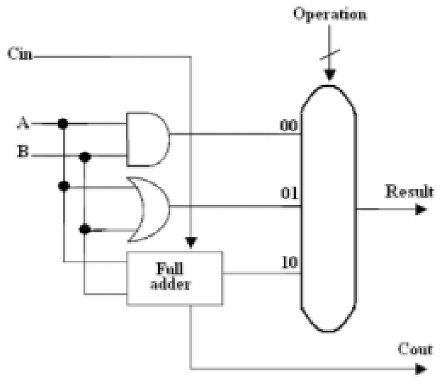

Fig. 1. ALU Circuit

F. Correction Circuit

A single parity bit can indicate that there is an error in a certain group of bits. In order to correct a detected error, more information is required because the position of the bit in error must be identified before it can be corrected [12]. More than one parity bit must be included in a group of bits to be able to correct a detected error. In a 7-bit code, there are seven possible single bit errors. In this case, three parity bits can not only detect an error but can specify the position of the bit in error. The Hamming code provides for single error correction. The following table covers the construction of a 7-bit Hamming code for single error correction.

TABLE III
BIT POSITION TABLE FOR A 7 BIT ERROR CORRECTION CODE

Bit Designation	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Bit Position	1	2	3	4	5	6	7
Binary Position	001	010	011	100	101	110	111
Data Bits (Dn)	x	x	x	x	x	x	x
Parity Bits (Pn)	y	y	y	y	y	y	y

G. Detection Circuit

The various existing error correction and detection methods are used a parity bit. An error bit contains either an even or odd number of 1s. [13]. An even parity bit makes the total number of 1s even and odd parity bit makes the total number of odd. The proper system has a error either in evern parity or odd parity, but not both. A parity error checker provides for the detection of a single bit error, which is very unlikely but cannot check for two errors in one group. For instance, we assume that the circuit transmits the BCD code is 0101. Parity can be used with any number of bits. The total code transmitted including the even parity bit is 0, 0101BCD Code [13]. Then let assume that an error occurs in the third bit from the left (the 1 becomes a 0). Even Parity Bit 0, 0001Bit Error.

H. Correction and Detection Circuit

Normally, there are error detection and correction circuits. Our proposed method has combined the error detection and correction as a simple way and make it in integrated circuit form. The correction and detection circuits are designed, by implementing the basic components. The modified ALU correction and detection circuits are shown in Fig 2. According to CMOS logic design, the proposed correction and detection circuit has been designed in terms of TTL configuration and block level configuration. The block level configuration would lead to reduce the logic cell in successive manner and give a logical effort of the block diagram. The TTL configuration would be reduced number of transistor and give a bit transition effort in the logic design [14]. This proposed circuit is analyzed in both ways and the output value simulated and measured. After that, we have generated it in one circuit and simulate the circuit. For the correction and detection circuit, we have to design by using arithmetic logic unit (ALU) circuit method such as bit slice method. Then the design is simulated with binary keys such as 000, 001, 010, 011, 100, 101, 110, and 111 to see the simulation performance. The design of the correction and detection circuit with the method of ALU CMOS are shown in the Fig. 2. This circuit is implemented from the block of the multiplexer circuit, demultiplexer circuit, and comparator circuit and designed in MATLAB.

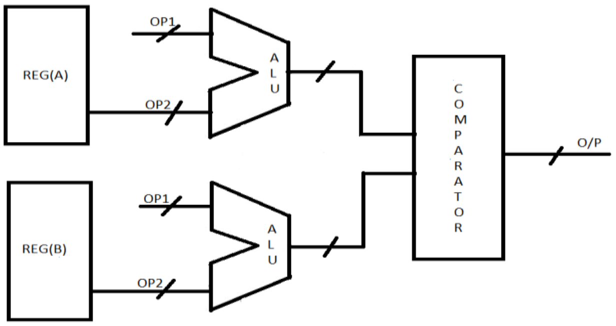

Fig. 2. Proposed Error Detection and correction circuit

III. III. RESULTS AND DISCUSSION

The proper selection of test vectors is crucial and the accuracy of the simulation can tell much about the circuit. The proposed error correction circuit simulated by MATLAB, DSCH2 and Micro wind. The circuit is simulated by MATLAB Simulink and verified by timing diagram.

A Conventional multiplexer circuit output depends upon the selection input. The three inputs are fed as clocking input into the function $A\overline{S} + BS$ which is clearly mentioned in the class. The clocking inputs are normalized as a logic one. The output is also displayed without delay which clearly indicates that the circuit doesn't have any fault. The output graph clearly is shown in the Fig 3(a) and 3(b)

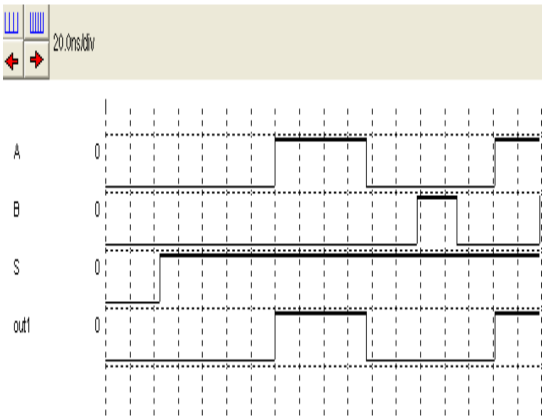

Fig. 3a. Simulink simulation

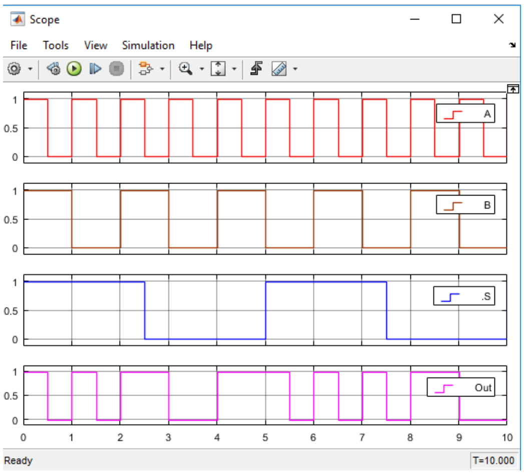

Fig. 3b. Timing diagram

The modified 2:1 multiplexer circuits undergone the simulation by MATLAB Simulink tools is shown in Fig 3(a). The input parameter has given logically 1 which is identified as amplitude values. The output parameters are measured in the output waveforms the change of time is 1.003ms. The average speed of individual 2:1 multiplexer is shown 900.824 MHz. The slew rate of 2:1 multiplexer circuit is much reduced within stability condition.

The modified demultiplexer circuit simulation result is shown in Fig. 4. According to modification, the demultiplexer circuit is given a better result, which shown in figure 4. This paper the logic patterns are to be magnified so the demultiplexer circuits used. According to the initial circuit diagram the 4:1 demultiplexer circuit used. The output of the demultiplexer circuit depends upon the data input and selection input. . The design gives a better operating frequency 539.344 MHz, RMS value for the 1 full cycle is 0.4364 which is clearly shown in Fig 4(a) and 4(b), The simulation result is verified with the timing diagram of the layout.

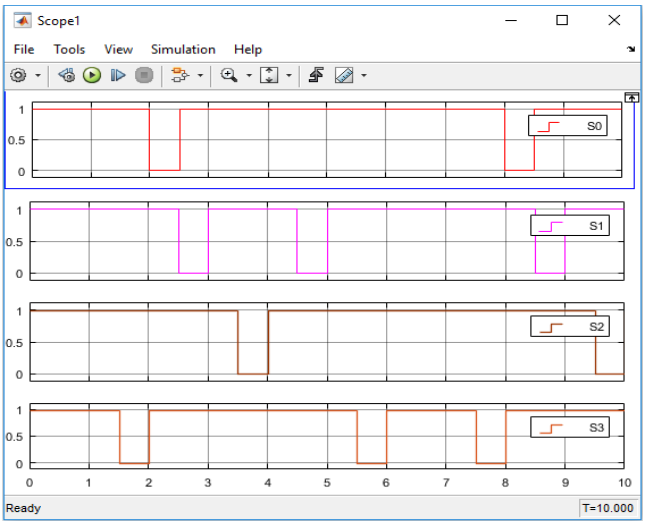

Fig. 4a. Matlab Simulation

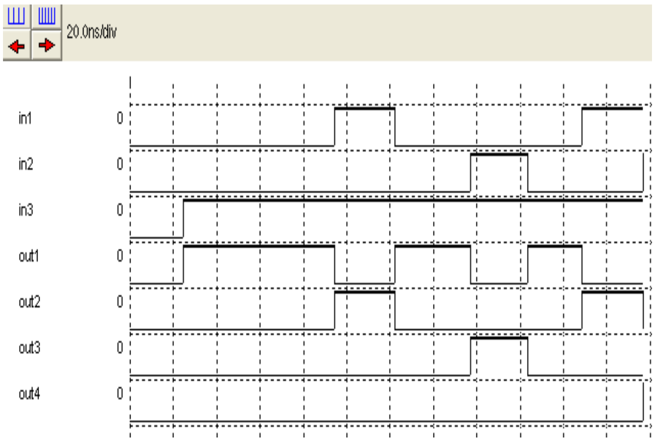

Fig. 4b. Timing diagram

The two inputs are compared, and it gives the complement output in the output terminal. The timing diagram of the comparator circuit is shown in Fig 5(b). It is clearly indicated that there is no propagation delay in the comparator circuit. This vertical placement of layout is used to calculate the total number of transistor in the circuit. The comparator circuit used only A=B magnitude, when A=1, B=1, and A=0, B=0 the output shown as logically zero. when A=1, B=0 and B=1, A=0 the output shown as logically 1. As an error detection and correction circuits, the comparison output always zero which is indicating there is no error sending bit pattern are data signals which are clearly indicating in Fig 5(b). The modified comparator circuit has 4 components, which gives the change of time/cycle of 1.003ms., which is shown in Fig 5(a) The frequency of the comparator circuit is 888.856 due to the fact that electron mobility has equal distribution in the output node. The simulated results verified with DSCH2 timing diagram, which is shown in Fig 5(b)

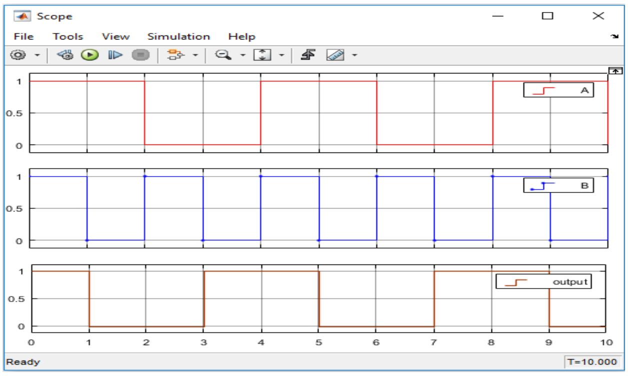

Fig 5(a) Comparator circuit

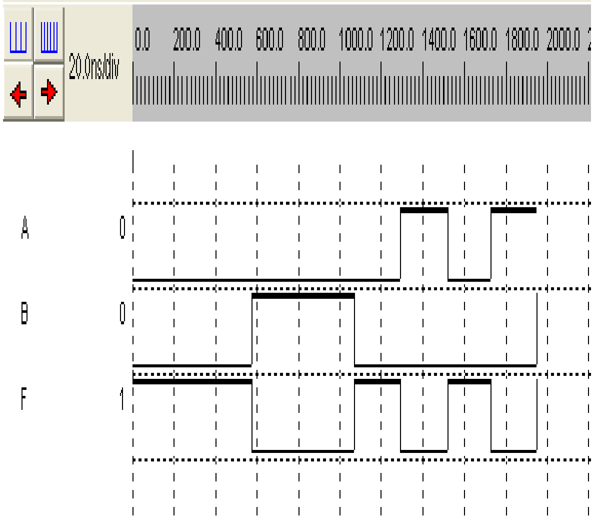

Fig 5(b) Timing Diagram

The EDAC circuit is simulated and imposed into the analyses. The change of time /cycle is calculated, which is equal to ΔT 2.036ms. The speed of the frequency circuits is approximately 455MHz. The ECC circuit layouts are imposed by LVS layout characteristic. The LVS had given power dissipation for each and every circuit, propagation delay and area. According to simulation itself, the Shannon theorem-based circuit gives better performance than CPL, CMOS, and Mixed Shannon method. LVS has extended to parameter analysis data throughput, latency, and EPI. The Shannon based circuit gives better or high throughput, low latency, and high EPI than other circuits due to the regular arrangement of circuit transistor. This Shannon based circuit has less critical path than other circuits.

The ECC circuit was simulated in MATLAB Simulink. The ECC circuit consists of registers A and B. These registers have at least 1-word (8-bit) number in the system. The inputs are fed into Serial in the serial out method. The output of the register fed into ALU circuits. The ALU circuit analyses the results depends upon the selection input. The ALU must have the same selection and correction code. The above-mentioned proposed circuit is to detect the error from the register and correct the error by ALU selection values. The output of the two ALU values is compared with the digital comparator circuit. If A=B, the digital error has detected and corrected by the system. This system may be extended as IEEE 32 bit format analysis. The output of ALU results are compared.

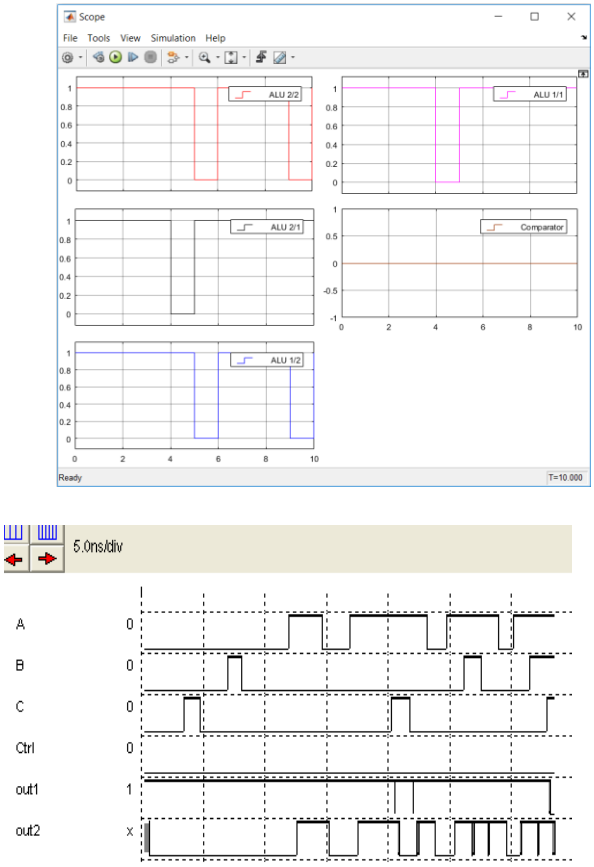

Fig 6 EDAC circuit

TABLE IV
MATLAB SIMULINK RESULTS OF EDAC CIRCUIT AND ITS AXILLARY CIRCUITS

Parameter	2:1 Mux	DeMux	Comparator	Full Adder	ALU	EDAC
Time(ΔT)(ms)	1.033	1.854	1.003	1.467	2.589	2.036
Amplitude	0.9056	0.9488	0.937	0.976	0.8801	0.937
1/ ΔT(MHz)	900.824	539.344	888.856	698.305	386.250	454.676
ΔY/ΔT(/ks)	823.042	511.727	698.256	550.466	339.939	340.666
RMS	0.7466	04364	0.7137	0.6901	0.7559	0.7237
Rise time(ms)	79.200	396.00	126.720	396.00	396.00	396.00
Fall time(ms)	80.82	398	128.820	397.98	39782	398.3
Slew rate(/s)	-10.000	-2	-7.00	-2.00	-2.00	2.00

An EDAC circuit layouts are simulated by Microwind VLSI CAD tools. The Shannon adder based EDAC circuit gives lower power dissipation and less occupying area than other adder based EDAC circuit due to Shannon based circuit. The propagation delays have increased due to 2 more transistors added in the circuit for regulating the critical path. The Shannon based error correction and detection circuit gives high throughput than other adder-based circuits, which is clearly indicated in Table V.

TABLE V
SIMULATION RESULT FOR POWER DISSIPATION, DELAY, AREA, EPI, THROUGHPUT AND LATENCY FOR PROPOSED CORRECT AND DETECT ERROR CIRCUIT (90NM)

ALU /Adder	Power (mW)	Delay (ps)	Area μm^2	Throuhput GHz	Latency N	EPI Watts/IPS)
Mixed Shannon	0.189	743	0.0024	0.709	4.229	0.04
CPL	0.199	254	0.0025	1.086	2.762	0.04
Shannon	0.186	653	0.0024	1.727	4.124	0.04
CMOS	0.246	37	0.0036	1.46	2.054	0.04

IV. CONCLUSION

The designed circuits and its layout simulation results are presented. A proposed EDAC and modified ALU, 2:1 Mux, Demux and Comparator circuit exhibits superior achievement as compared to the conventional EDAC circuit in terms of power dissipation, propagation delay, and speed. The proposed EDAC circuit demonstrates better performance than other an existing and modified EDAC cells designed with CPL, CMOS, Shannon and Mixed Shannon in terms of power dissipation, propagation delay, a number of transistors, area, throughput, latency and EPI of the circuits. The EDAC circuits were simulated in Simulink and results were verified with LVS. The paper describes the design of Arithmetic Circuit has basic logic gates, ALU circuits, a Multiplexer circuit, Demultiplexer circuit and Comparator circuit are performed by utilizing DSCH 2 and MATLAB Simulink CAD tool. The all modified circuit is checked and justify by layout simulator and confirm with the timing diagram of a circuit by the Microwind 3 program.

REFERENCES

[1] Joseph Crop Evgeni Krimer, Nariman Moezzi-Madani , Robert Pawlowski , Thomas Ruggeri 1 , Patrick Chiang 1 and Mattan Erez Error Detection and Recovery Techniques for Variation-Aware CMOS Computing: A Comprehensive Review. J. Low Power Electron. Appl. 2011, 1, pp 334-356.

[2] Jani Mäkipää , Matthew J. Turnquist , Erkka Laulainen and Lauri Koskinen Timing-Error Detection Design Considerations in Subthreshold: An 8-bit Microprocessor in 65 nm CMOS *LPEA* 2012, *2*(2), pp 180-196.

[3] Varadan Savulimedu Veeravalli Fault tolerance for arithmetic and logic unit IEEE South eastcon 2009 Pages: 329 – 334.

[4] S. H. Mozafari, M. Fazeli, S. Hessabi, S. G. Miremadi, "A Low Cost circuit level fault detection technique to Full Adder design", *Electronics Circuits and Systems (ICECS) 2011 18th IEEE International Conference on*, 2011, pp. 446-450.

[5] Ryan C. Bickham, , Daniel B. Limbrick, William H. Robinson, Bharat L. Bhuva, " An Analysis of Error Detection Techniques for Arithmetic Logic Units (ALUs) "RADECS 2010, IEEE Proceedings.

[6] Shinjini Yadav and A. K. Singh Implementation of 32-Bit Fault Tolerance ALU using VHDL for Checking bit up to 6 Bits. International Journal of Research and Development in Applied Science and Engineering, Volume 10, Issue 2, July 2016, pp 1-7.

[7] C. Senthilpari, G. Ramanamurthy, P. Velrajkumar An Efficient EPI and Energy Consumption of 32 bit ALU Using Shannon Theorem Based Adder Approach WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, Issue 7, Volume 10, July 2011, pp231-238.

[8] J. Crop, R. Pawlowski and P. Chiang, "Regaining throughput using completion detection for error-resilient, near-threshold logic," *DAC Design Automation Conference 2012*, San Francisco, CA, 2012, pp. 974-979.

[9] C Senthilpari, AK Singh, K Diwakar "Design of a low-power, high performance, 8× 8 bit multiplier using a Shannon-based adder cell" Microelectronics Journal 39 (5), pp 812-821.

[10] Abhishek Rai, B Ananda Venkatesan M.Tech Scholar, "Analysis and Design of High Speed Low Power Comparator in ADC" © 2014 IJEDR | Volume 2, Issue 1 | ISSN: 2321-9939.

[11] M. R. Kakoee, I. Loi and L. Benini, "Variation-Tolerant Architecture for Ultra Low Power Shared-L1 Processor Clusters," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 12, pp. 927-931, Dec. 2012.

[12] C. H. Chen, D. Blaauw, D. Sylvester and Z. Zhang, "Design and Evaluation of Confidence-Driven Error-Resilient Systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 8, pp. 1727-1737, Aug. 2014.

[13] C.Senthilpari K.Diwakar KumarMunusamy J. SheelaFrancisca Layout parameter analysis in Shannon expansion theorem based on 32 bit adder circuit Engineering Science and Technology, an International Journal Volume 20, Issue 1, February 2017, Pages 35-40.

[14] C Senthilpari "A Low-power and High-performance Radix-4 Multiplier Design Using a Modified Pass-transistor Logic Technique" IETE Journal of Research 57 (2), 149-155.

[15] A. Rahimi, L. Benini and R. K. Gupta, "Variability Mitigation in Nanometer CMOS Integrated Systems: A Survey of Techniques From Circuits to Software," in *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1410-1448, July 2016.