

Design and Implementation of 4 Bit Multiplier Using Fault Tolerant Hybrid Full Adder

C. Kalamani, V. Abishek Karthick, S. Anitha, K. Kavin Kumar

Abstract—The fault tolerant system plays a crucial role in the critical applications which are being used in the present scenario. A fault may change the functionality of circuits. Aim of this paper is to design multiplier using fault tolerant hybrid full adder. Fault tolerant hybrid full adder is designed to check and repair any fault in the circuit using self-checking circuit and the self-repairing circuit. Further, the use of conventional logic circuits may result in more area, delay as well as power consumption. In order to reduce these parameters of the circuit, GDI (Gate Diffusion Input) techniques with less number of transistors are used compared to conventional full adder circuit. This reduces the area, delay and power consumption. The proposed method solves the major problems occurring in the most crucial and critical applications.

Keywords—Gate diffusion input, hybrid full adder, self-checking, fault tolerant.

I. INTRODUCTION

ADDITION is the most fundamental arithmetic operation performed in many VLSI systems such as microprocessors and application specific Digital Signal Processing (DSP) architecture. All complex arithmetic circuits consist of full adder to execute arithmetic operation such as multiplication, division etc. [1], [2]. These full adders are the nucleus of any system. Many applications contain full adder and it is one of the important parts of circuit.

Faults may lead to failure of the system. So it is necessary to eliminate the fault. A fault may be due to processing, missing materials such as hot carrier effects and metal migration. Faults of this type require the defective part to be replaced. Faults of this type require the defective part to be replaced. The second type of fault is a transient fault where the circuit produces incorrect outputs due to the phenomena such as power supply variation, lightning, α -particles and electromagnetic interference. If the circuit is sequential, the transient fault may cause latch errors which may in turn lead to system failure. A source of transient fault is α -particles present in cosmic rays, which is basis of non-permanent errors in semiconductor circuits [3]. Fault tolerant system is incorporated in applications such as space application, defense surveillance, medical supervisory system and other safety related services. Overall system performance can be effect by the presence of fault in a circuit [4].

Ms. C. Kalamani is with the ECE Department, Dr. Mahalingam College of Engineering and Technology, Coimbatore, Tamilnadu, India (e-mail: kalamec18@gmail.com).

V. Abishek Karthick, S. Anitha and K. Kavin Kumar are with ECE Department, Dr.Mahalingam College of Engineering and Technology, Coimbatore, Tamilnadu, India.

In most of the systems, full adders are encountered in the critical path. The design criteria for full adder are usually multi fold. System complexity depends on transistor count in the ALU. Many fault tolerant full adders were presented in the past. Self-checking full adder can detect the single fault at a time. It is complex and incompatible with its self-checking memory systems. But the usage of conventional full adder in the circuit results in larger delay, area and power consumption [4]. Therefore, in order to overcome this, a fault tolerant full adder is presented. A technique called as GDI is used to optimize the full adder circuit. In addition to the fault detection and fault repair, it can offer a reduced amount of delay and power.

II. EXISTING METHODS

The GDI based sum cell [5] provides a full adder circuit design with only a less number of transistors when compared with the conventional full adders and hybrid full adder.

The fault-tolerant adder design [4] could check for both the single and double faults that occur in a circuit and also it could repair both faults simultaneously without any error in detection and correction.

Since GDI based full adder has utilized less number of transistors, it occupies lesser area and low delay and power consumption than the conventional full adder [6], [7].

The fault-tolerant adder design has the ability to check the single faults in a circuit and when double faults occur, even though it uses only less number of transistors for the whole circuit design, it cannot repair any of the faults. The fault tolerant hybrid full adder design with GDI techniques is to check single and double error. The proposed method is implemented using fault tolerant hybrid full adder design. They provide better performance than the existing methods.

III. PROPOSED METHOD

The method of fault tolerant full adder uses GDI technique [8] to design the full adder circuit. This is used to design 4 bit multiplier.

A. GDI Based Self-Checking Full Adder Design

Self-checking full adder uses the GDI based sum cell instead of the conventional full adder and existing methods [5], [8], [9].

The output expressions for the sum and carry outputs of the full adder is shown in (1) and (2):

$$\text{Sum} = A \oplus B \oplus C_{in} \quad (1)$$

619

minimum hardware resulting in minimum delay and power consumption.

IV. RESULTS AND DISCUSSION

The existing and proposed full adder, multiplier using proposed full adder were designed using Cadence Virtuoso 64 tool of 180 nm technology. The simulation results were obtained. The conventional full adder circuit is designed using Cadence Virtuoso tool and the schematic diagram is given in Fig. 3. The output waveform of the conventional full adder

circuit is shown in Fig. 4.

The full adder circuit has been designed using GDI technique and the schematic diagram of the GDI based full adder is shown in Fig. 5. Waveform of GDI based full adder is shown in Fig. 6. Fig. 7 gives the schematic diagram of the self-checking full adder designed using conventional full adder. Fig. 8 gives the output of the self-checking full adder designed using the conventional full adder. In this figure, we could observe that the output of the Fc and Fs are zero since there is no fault in carry and sum respectively.

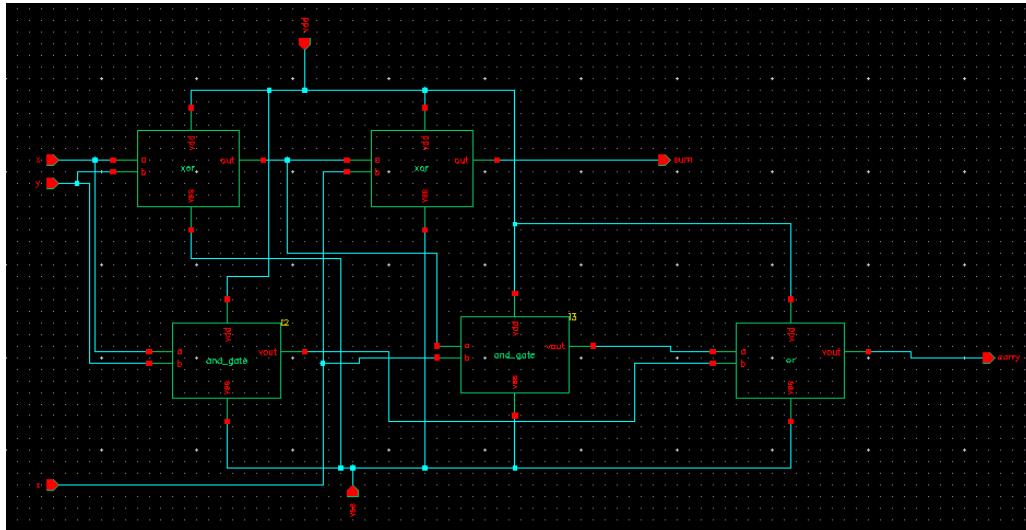


Fig. 3 Schematic diagram of full adder

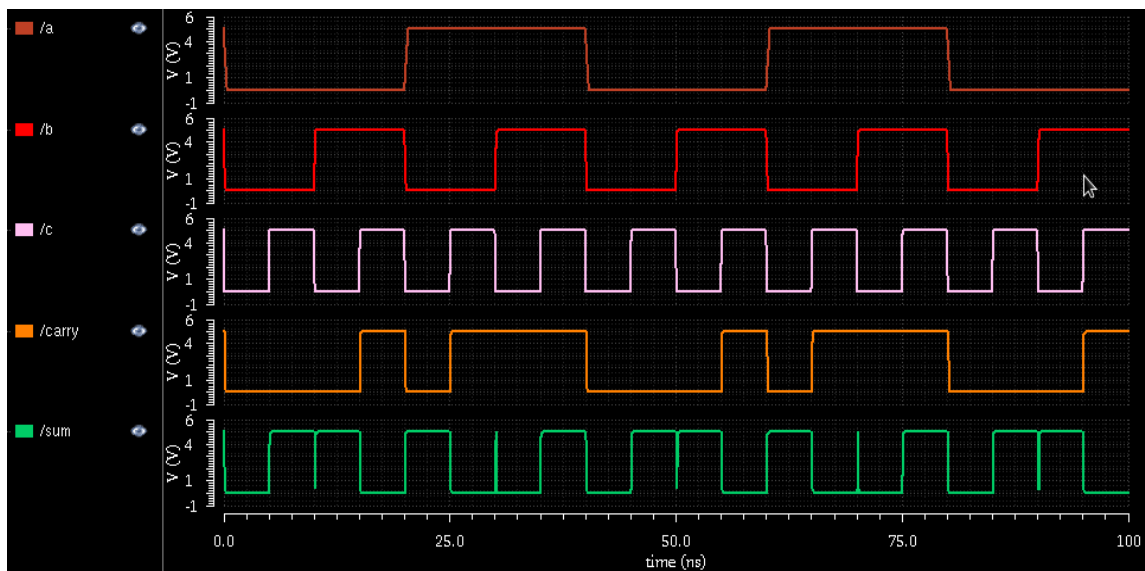


Fig. 4 Waveform of Full adder

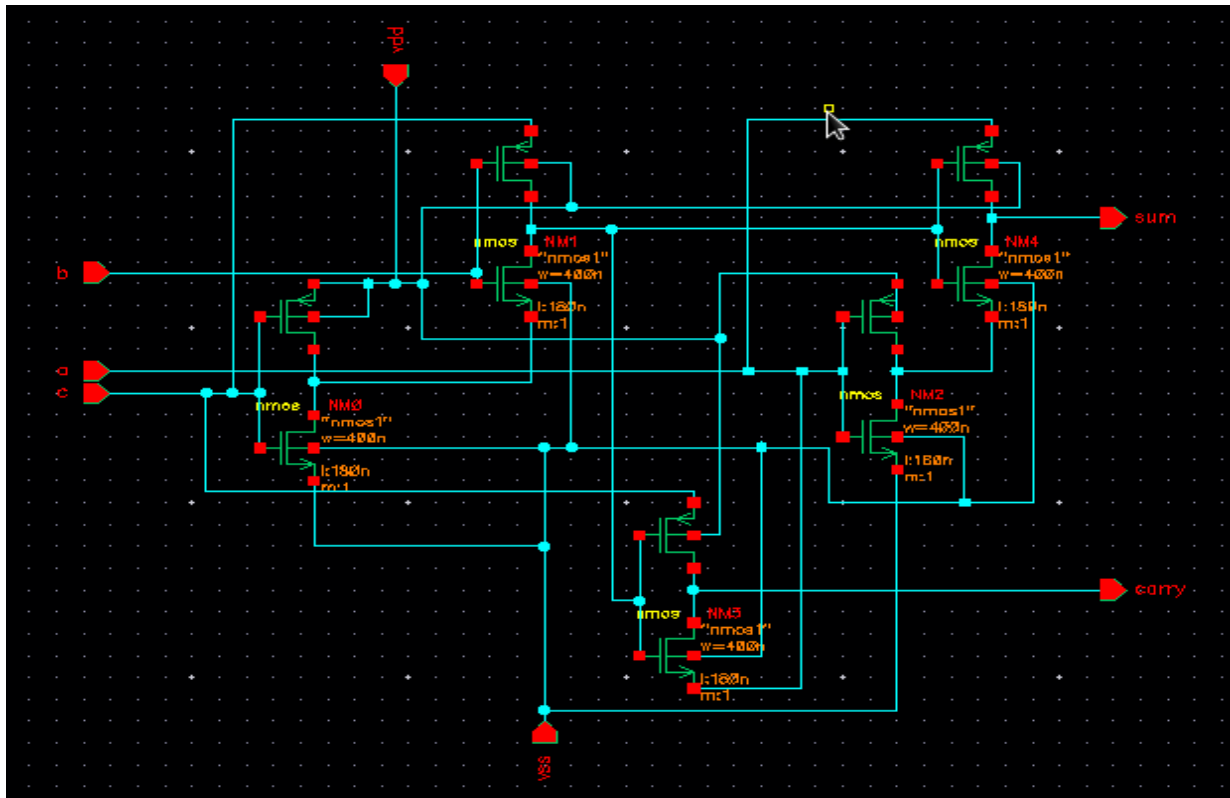


Fig. 5 Schematic diagram of GDI based full adder

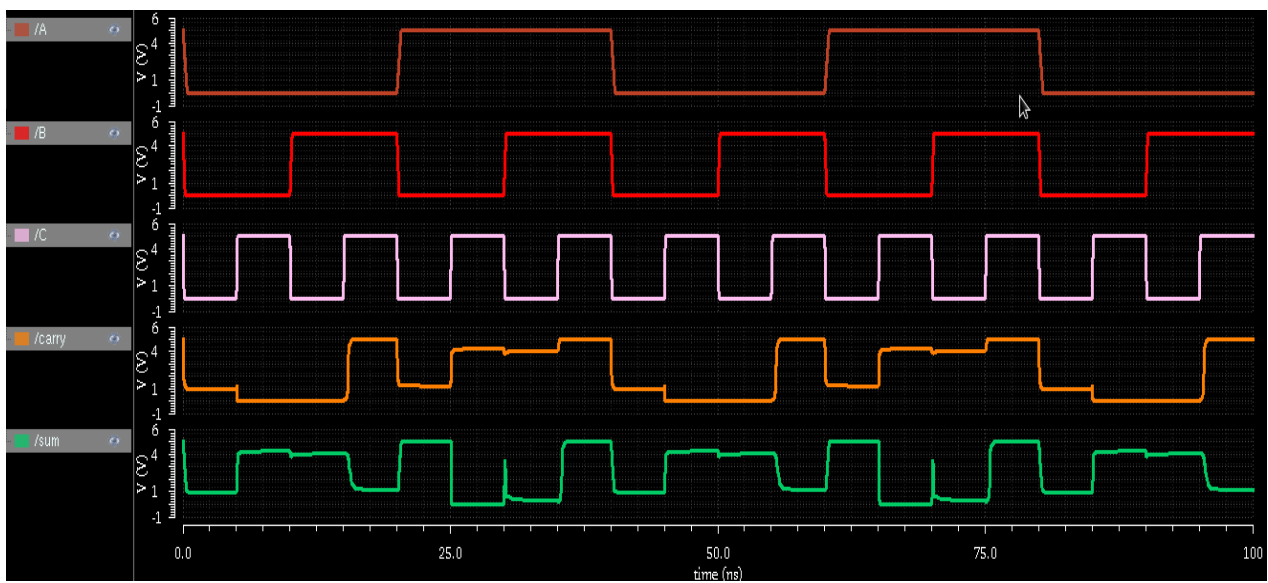


Fig. 6 Waveform of GDI based full adder

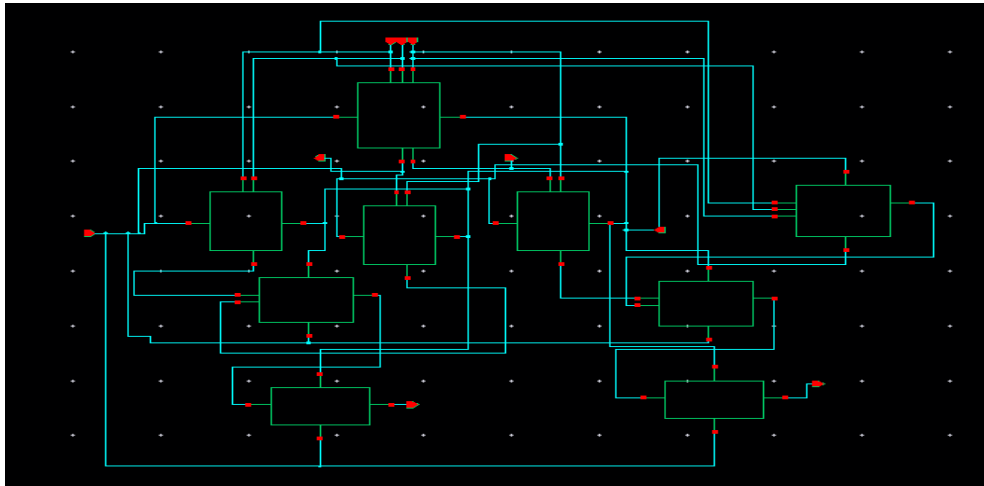


Fig. 7 Schematic diagram of Self-checking full adder

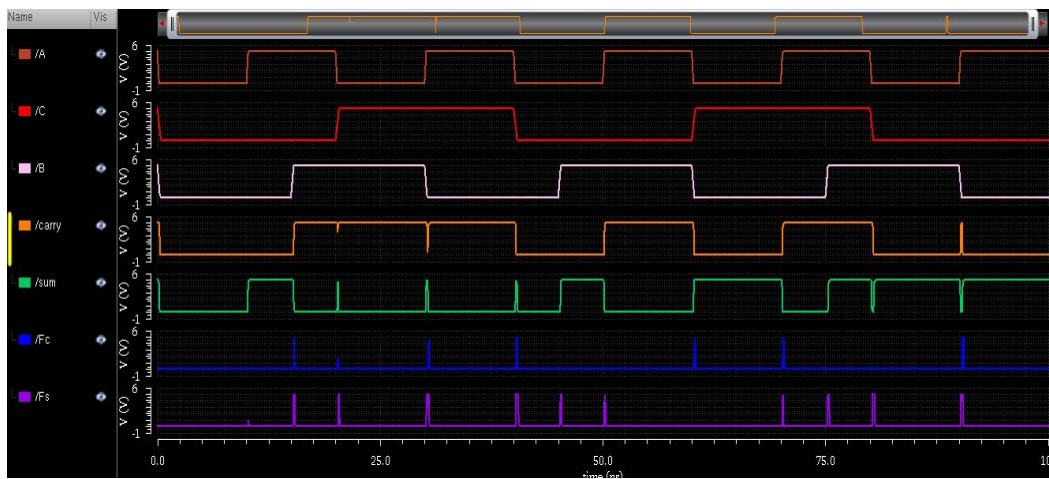


Fig. 8 Output of Self-checking full adder

The schematic diagram and waveform of the self-checking full adder designed using the GDI technique is shown in Figs. 9 and 10 respectively.

The self-repairing full adder is designed using the self-checking full adder and multiplexer and the schematic diagram of this self-repairing full adder is given in Fig. 11. Fig. 12 gives the output of self-repairing full adder. The output fault has been repaired using the self-repairing circuit.

The self-checking full adder designed using the conventional full adder used in the previous self-repairing design is replaced by the self-checking full adder designed using GDI technique. The schematic diagram of this self-repairing full adder is given in Fig. 13.

The output of the self-repairing full adder using GDI is shown in Fig. 14. A multiplier is implemented using the existing method of self-repairing full adder as in [10] which is shown in Fig. 15.

The array multiplier is designed using self-repairing full adder. Fig. 16 gives the schematic diagram of the multiplier designed using the proposed self-repairing full adder.

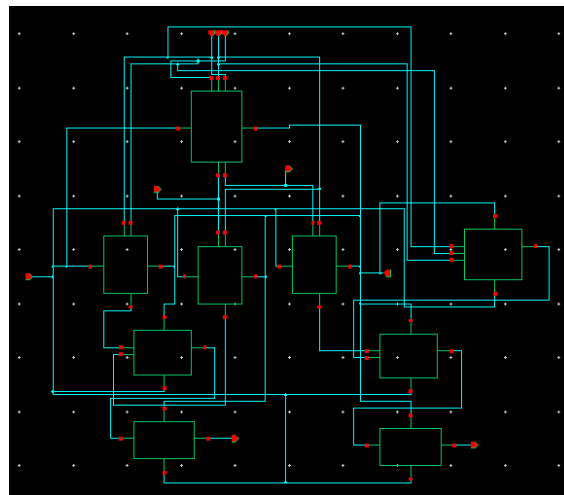


Fig. 9 Schematic diagram of Self-checking full adder using GDI

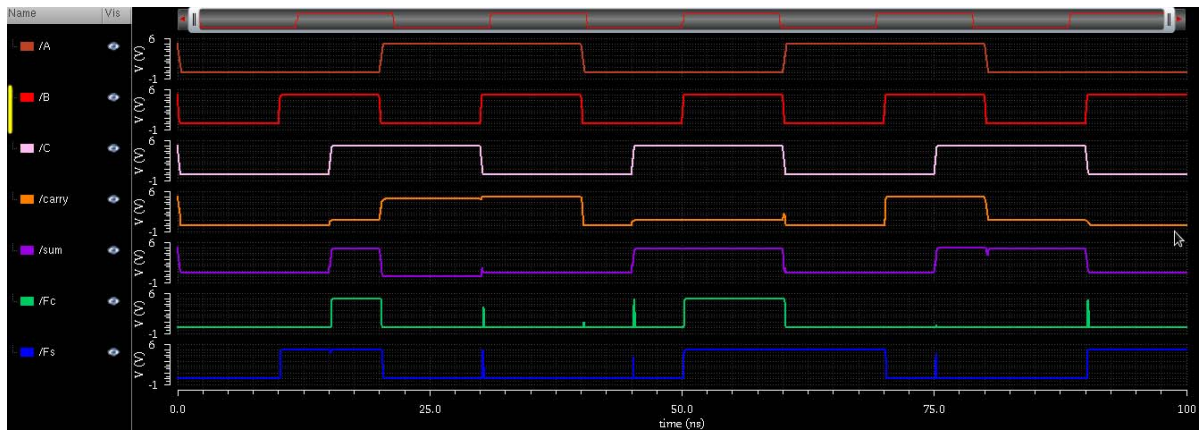


Fig.10 Output of Self-checking full adder using GDI

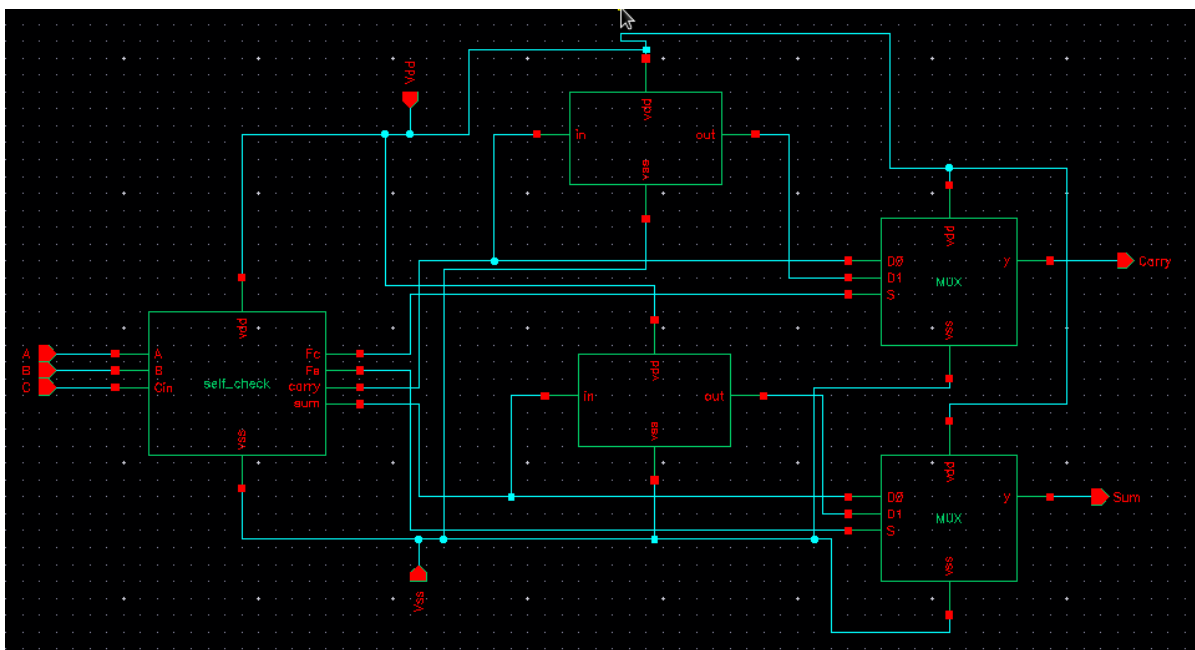


Fig.11 Schematic diagram of Self-repairing full adder

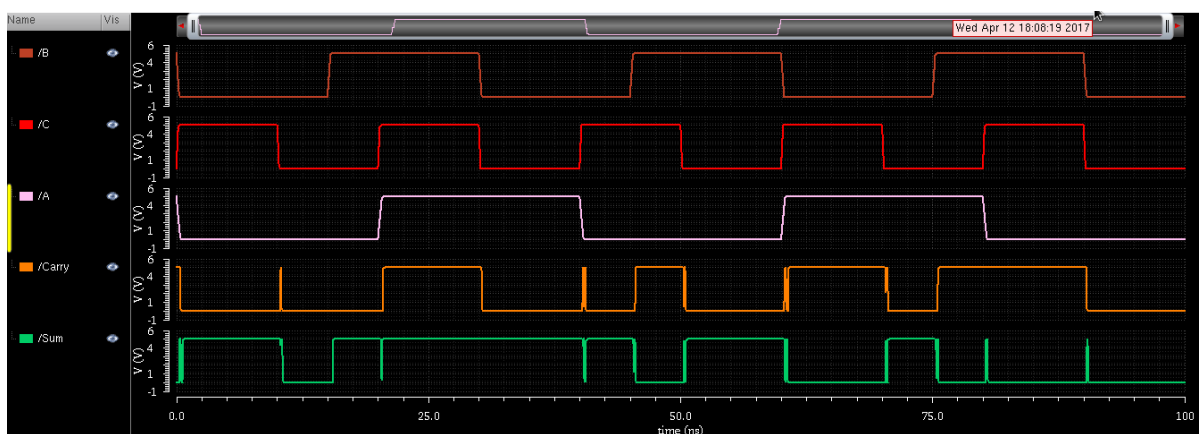


Fig.12 Output of Self-repairing full adder

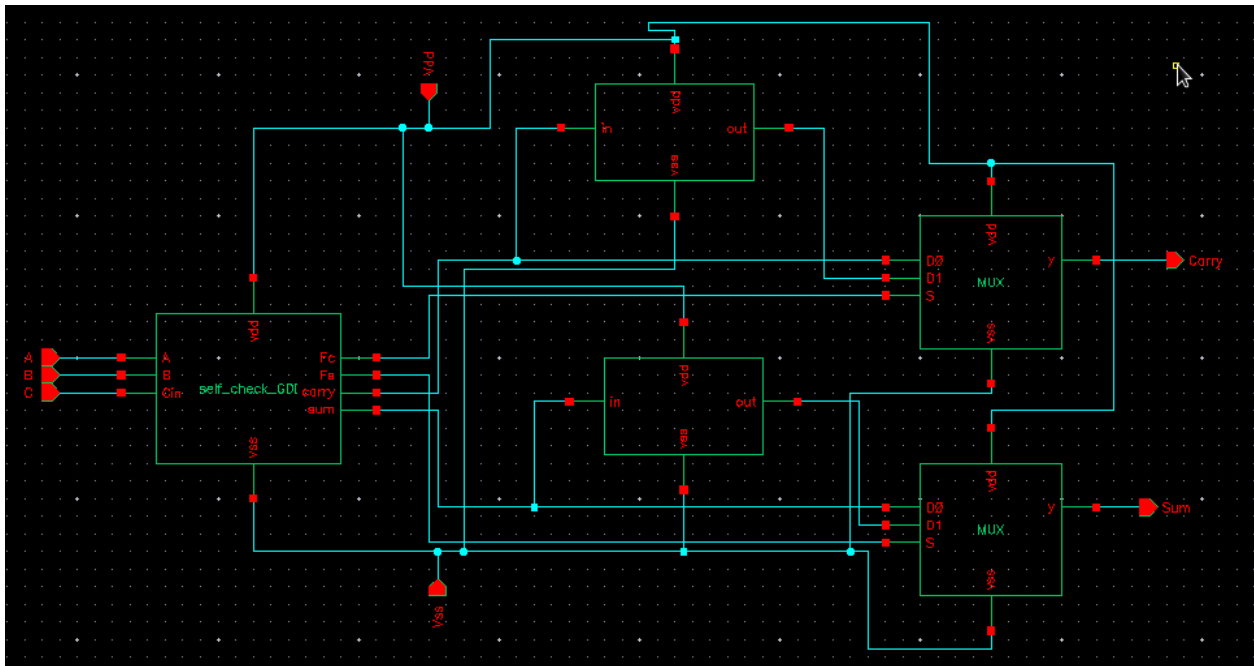


Fig. 13 Schematic diagram of Self-repairing full adder using GDI

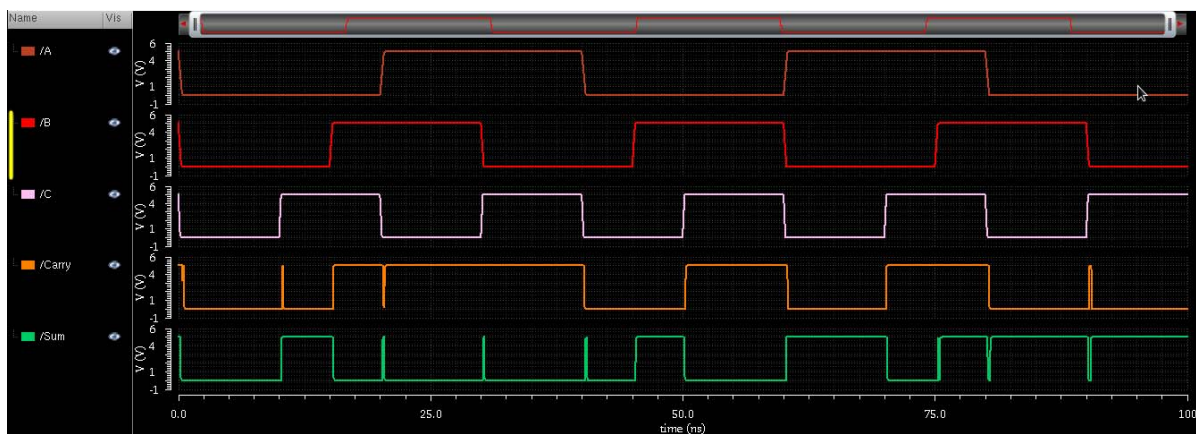


Fig. 14 Self-repairing full adder (GDI) –Output

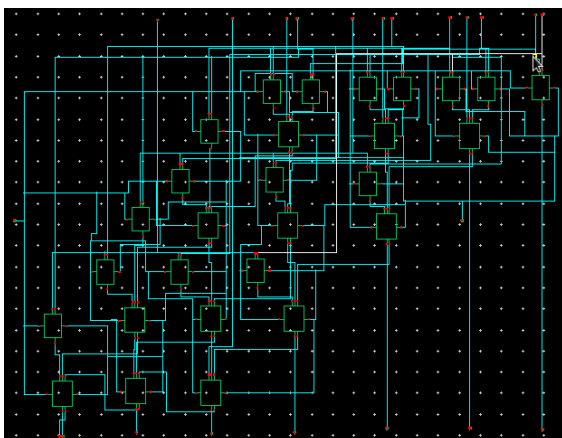


Fig. 15 Schematic diagram of Multiplier

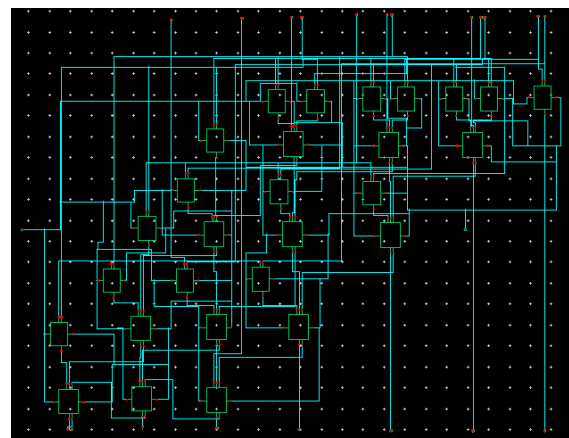


Fig. 16 Schematic diagram of Multiplier with GDI

A detailed comparison on the power, delay and power delay product of all designed circuit are given Table I.

TABLE I
COMPARISON OF POWER, DELAY AND PDP

Name	Power (mw)	Delay (ps)	Power delay product (pJ)
Self-check	1269.84	186.28	236.54
Self-check (GDI)	1247.6	22.79	28.43
Self-repair	1274.9	324.9	414.22
Self-repair (GDI)	1274.7	299.5	381.77
Multiplier	972.59	327.99	318.99
Multiplier (GDI)	659.65	311.80	205.67

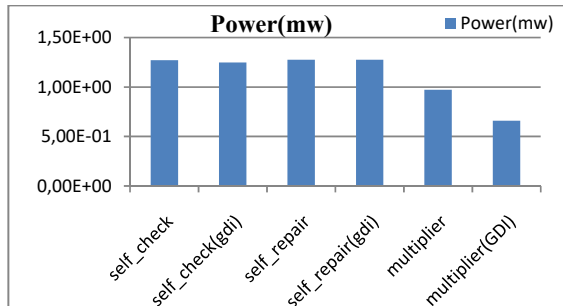


Fig. 17 Comparison chart for power

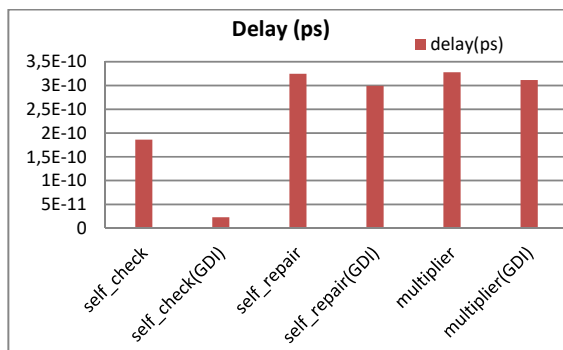


Fig. 18 Comparison chart for delay

From the comparisons made in Tables I, II and Figs. 18, 19 it has been observed that the self-repairing full adder using GDI technique provides less delay and power consumption along with less number of transistors when applied in a multiplier circuit and provides fault-free output.

V.CONCLUSION

A technique of self-checking and self-repairing full adder using GDI technique has been proposed. The design can detect and repair both single and double faults at a time using less number of transistors providing lesser delay and less power consumption. The comparison results of the designs are found better than the previous existing methods. The design is extendable up to a desirable level. A 4-bit fault tolerant multiplier is also implemented using the design. It works efficiently when cascaded and can handle the faults successfully.

TABLE II
COMPARISON OF NUMBER OF TRANSISTORS USED

Name	List of logic gates	Transistor count
Self-checking full adder	Full adder-1	26
	XNOR-5	60
	NOT-2	4
	Functional unit-1	28
		Total=118
Self-checking full adder (GDI)	Full adder (GDI)-1	10
	XNOR-5	60
	NOT-2	4
	Functional unit-1	28
		Total=102
Self-repairing full adder	Self-check adder-1	118
	NOT-2	4
	MUX-2	20
		Total=142
Self-repairing full adder (GDI)	Self-check adder (GDI)-1	102
	NOT-2	4
	MUX-2	20
		Total=126
Multiplier	Self-repair adder-12	1704
	AND-16	96
		Total=1800
Multiplier (GDI)	Self-repair adder (GDI)-12	1512
	AND-16	96
		Total=1608

REFERENCES

- [1] U. Ko, T. Balsara, and W. Lee, "Low-power design techniques for high-performance CMOS adders", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 3, no. 2, pp. 327-333, Jun. 1995.
- [2] Abdelmonaem Ayachi, Belgacem Hamdi, "A Fault-Tolerant Full Adder in Double Pass CMOS Transistor", International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, Vol: 10, No: 1, 2016, pp: 36-40.
- [3] Hungse cha, Elizabeth M. Rudnick, Gwan S. choi, Janak H. Patel and Ravishankar K. Iyer "A fast and accurate gate-level transient fault simulation environment", Proceedings of the International Symposium on Fault-Tolerant Computing, pp. 310-319, June 1993.
- [4] Pankaj Kumar, Rajender Kumar Sharma, "Real-time fault tolerant full adder design for critical applications", Engineering Science and Technology, an International Journal 19, 2016, pp: 1465-1472.
- [5] Rajkumar Sarma and Veerati Raju, "Design and performance analysis of hybrid adders for high speed arithmetic circuit" International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3, June 2012, pp: 21-32.
- [6] M. Shoba, R. Nakkeeran, "GDI based full adders for energy efficient arithmetic applications", Eng. Sci. Technol. Int. J. 19 (01), 2016, pp: 485-496
- [7] K. Navi, V. Foroutan, M. RahimiAzghad, M. Maeen, Ebrahimpour, M. Kaveh, O. Kavehei, "A novel low-power full-adder cell with new technique in designing logical gates based on static CMOS inverter," Microelectron J. 40 (10), 2009, pp:1441-1448.
- [8] Sudeshna Sarkar, Monika Jain, Arpita Saha, AmitRathi, "Gate Diffusion Input: A technique for fast digital circuits", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Volume 4, Issue 2, Ver. IV, Mar.-Apr. 2014, pp: 49-53.
- [9] Pankaj Kumar, Rajender Kumar Sharma, "Low voltage high performance hybrid full adder", Engineering Science and Technology, an International Journal 19, 2016, pp: 559-565.
- [10] Sumit Vaidya and Deepak Dandekar, "Delay-Power performance comparison of multipliers in VLSI circuit design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010, pp.47-56.