International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Synthesis and Implementation of Adder and Subtractor Using RPLA

Om. Pradeep[1], K. Anil Kumar[2], A.Soujanya[3]

[1]Department of Electronics and Communications, Student in Chalapathi Institute of Engineering and Technology, Guntur, Andhra Pradesh, India.
e-mail: pradeep@gmail.com

[2]Department of Electronics and Communications, Working as an Associate Professor in Chalapathi Institute of Engineering and Technology, Guntur, Andhra Pradesh, India. e-mail: kothuru.anilkumar@gmail.com

[3]Department of Electrical and Electronics, Working as an Assistant Professor in Chalapathi Institute of Engineering and Technology,Guntur, Andhra Pradesh, India. e-mail: : p.soujanya90@gmail.com

ABSTRACT

Reversible logic gates are very much in demand for the future computing technologies as they are known to produce Zero power dissipation.. Reversible logic circuits are of interests to power minimization having applications in low power CMOS design, optical information processing, DNA computing, bioinformatics, quantum computing and nanotechnology. In recent years, reversible logic has emerged as a promising computing paradigm having application in low power CMOS and optical computing. The classical set of gates such as AND, OR, and EXOR are not reversible. In this paper, the authors have proposed the synthesis of combinational circuits using RPLA. Furthermore, the application is shown by implementing the full adder and full subtractor functions through it and the performance parameters are compared with the previous work.

Keywords: Reversible logic; CMOS; Nanotechnology; RPLA

1. INTRODUCTION

This section provides an effective background of reversible logic with its definition and the motivation behind it. A. Definitions Researchers like Landauer have shown that for irreversible logic computations, each bit of information lost generateskTln2 joules of heat energy, where k is Boltzmann's constant and T the absolute temperature at which computation is performed [1]. Bennett showed that kTln2 energy dissipation would not occur, if a computation is carried out in a reversible way [2], since the amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Furthermore, voltage-coded logic signals have energy of Esig = ½CV2 and this energy gets dissipated whenever switching occurs in conventional (irreversible) logic implemented in modern CMOS technology. It has been shown that reversible logic helps in saving this energy using charge recovery process [10]. Reversible circuits are those circuits that do not lose information. Reversible computation in a system can be performed only when the system comprises of reversible gates. These circuits can generate unique output vector from each input vector, and vice versa, that is, there is a one-to-one mapping between input and output vectors. Thus, an NXN reversible gate can be represented as Iv= (I1, I2, I3, I4 …IN) Ov= (O1, O2, O3 ….ON). Where Iv and Ov represent the input and output vectors respectively. Classical logic gates are irreversible since input vector states cannot be uniquely reconstructed from the output vector states. There are a number of existing reversible gates such as Fredkin gate [3, 4, 5], Toffoli Gate (TG) [3, 4] and Feynman gate [6]. B. Proposed Contribution and Motivation behind the Work The reversible logic operations do not erase (lose) information and dissipate very less heat. Thus, reversible logic is likely to be in demand in high speed power aware circuits. Reversible circuits are of high interest in low-power CMOS design [7], optical computing [8], quantum computing [9] and nanotechnology [10]. The most prominent application of reversible logic lies in quantum computers. A quantum computer can be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; each gate performs an elementary unitary operation on one, two or more two–state quantum systems called quits. Each quit represents an elementary unit of information corresponding to the classical bit values 0 and 1. Any unitary operation is reversible, hence quantum networks effecting elementary arithmetic operations such as addition [12] multiplication and exponentiation cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible).Thus, Quantum Arithmetic must be built from reversible logic components [10]. Furthermore, programmable logic arrays (PLAs) have a number of medical and industrial applications, such as ultrasonic flaw detection. The reasons stem from the fact that PLAs are considerably faster than high end DSPs. They provide the cost effective solution to the exponentially increasing needs of industrial electronics. Thus, seeing the benefits of programmable logic array and reversible logic in industrial electronics and applications, the authors propose the reversible programmable logic array (RPLA) designed using reversible gates. In order to demonstrate the proposed architecture of RPLA, a 3 input RPLA which can perform any 28 functions using the combination of 8 min terms (23) is also designed [9].

I. ***BASIC REVERSIBLE GATES:*** There are a number of existing reversible gates in literature. Fredkin [3, 4, 5] and Feynman gates [3] are used to construct the reversible PLA.

A brief description of the gates is given below [1]. A. Fredkin Gate Fredkin gate [3, 4, 5], is a (3*3) conservative reversible gate originally introduced by Petri [4, 5]. It is called 3*3 gates because it has three inputs and three outputs. The input triple(x1, x2, x3) associates with its output triple (y1, y2, y3) as follows:

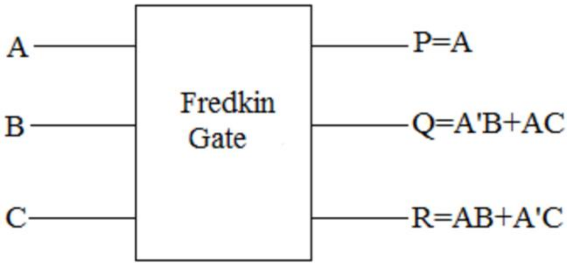


Fig 1: Fredkin gate

Figure 1show the implementation of the Fredkin gate as OR and AND functions respectively. (a) (b) Fig. 1. Fredkin gate as (a) OR function (b) AND function

Toffoli Gate Toffoli [3, 4] gives constructions for an arbitrary reversible or irreversible function in terms of a certain gate library. However, his method makes use of a large number of temporary storage channels, i.e. input-output wire-pairs other than those on which the function is computed (also known as ancillary bits). Sasao and Kinoshita show that any conservative function (f (x) is conservative if x and f (x) always contain the same number of 1s in their binary expansions) has an implementation with only three temporary storage channels using a certain fixed library of conservative gates, although no explicit construction[5] .

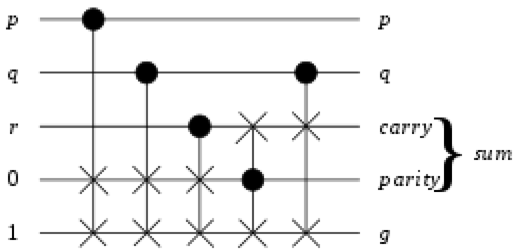


Fig. 2. Quantum representation

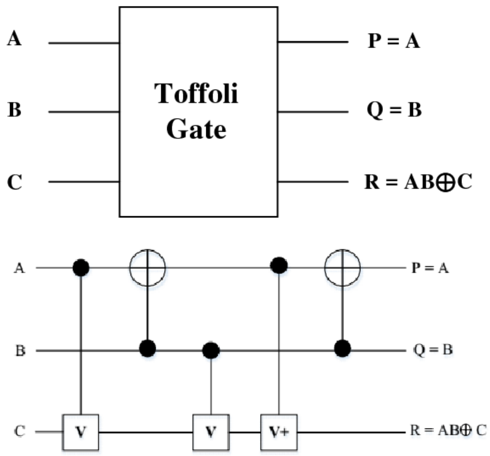


Fig. 3: Toffoli Gate and its Quantum Implementation

III. PROPOSED REVERSIBLE PROGRAMMABLE LOGIC ARRAY

In this paper, the authors have proposed the architecture of reversible PLA called RPLA [10]. The architecture of reversible PLA is shown in Figure 4. The RPLA consists of reversible AND array designed from reversible Toffolli gate and Feynman gate, in which the Feynman gates are used to avoid the problem of fan-out and for generating the complement of the inputs. The reversible AND array realizes the selected product terms of the input variables. The reversible OR array designed from reversible Fredkin gates is used to generate various possible functions of the product terms (outputs of reversible AND array).

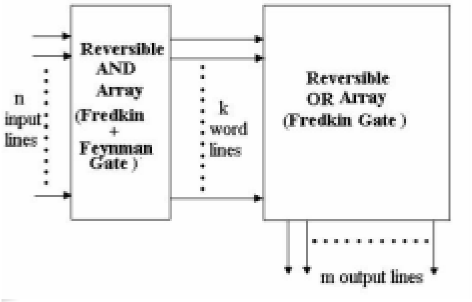

n input lines
Reversible AND Array (Fredkin + Feynman Gate)
k word lines
Reversible OR Array (Fredkin Gate)
m output lines

Fig. 4. Proposed Reversible PLA (RPLA) Architecture

IV. DESIGN OF THREE INPUT REVERSIBLE

PLA In order to demonstrate the actual design of the proposed reversible PLA (RPLA), a 3 input RPLA is shown in Figure 4 In Figure 4, the AND functions required to realize the AND array are implemented using reversible Fredkin gates. The application of Fredkin gate as an AND and OR gate has already been discussed in the previous section. In the AND array, the complement of the inputs are required and moreover fan-out is not allowed in reversible logic, thus Feynman gates are used to complement and replicate the signals when required. The designed 3 input reversible AND array will generate 8 product terms as outputs, which are combined using the reversible OR array designed using the Fredkin gate, to generate the required output functions.[8] A. Applications of the designed 3 Input RPLA The designed 3 input RPLA is used to implement the 1 bit full adder and 1-bit subtractor. The 1-bit full adder as shown in Figure 5 is implemented using the 3 input RPLA by generating the product terms in the full adder truth table through the AND array, and then appropriately combining the product terms through the reversible OR array to finally generate the required SUM and CARRY output functions. Similarly, the 1- bit subtractor as shown in Figure 8 is implemented to generate the Difference and Borrow output functions. Figure 5 shows the full adder using reversible PLA and its quantum depth is shown in fig. 6. Figure 7 shows the quantum representation of fig. 5. Reversible full subtractor is analyzed in fig. 5 and fig. 6.

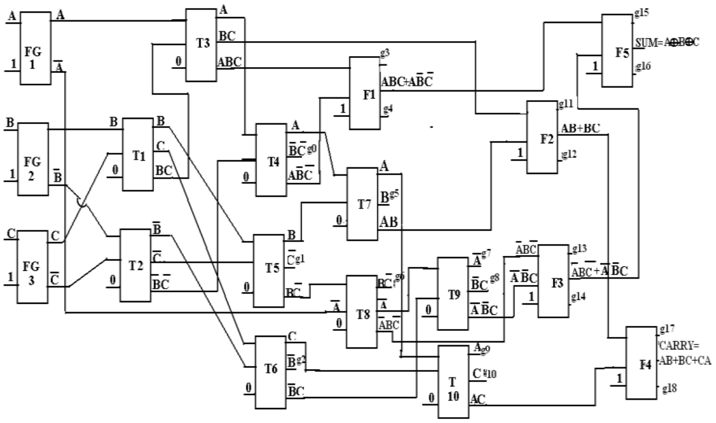


Fig 5 : Design of full adder using RPLA.

In this design, A, B and C are the inputs of full adder, sum and carry are the corresponding outputs. The functioning of CNOT, CCNOT and FREDKIN gates used in the circuit resembles same as the outputs of the adder and multiplier circuits Therefore, for the CNOT gate the inputs given are A and 1 then the outputs are A and Ā. Similarly the remaining inputs B,C are given to the respective two CNOT gates as B,1 and C,1 then the outputs are B, $\overline{B}$ and C , $\overline{C}$. These outputs are again given as inputs to TOFFOLI gates. For the T1 TOFFOLI gate the inputs are B, C (i.e. corresponding output of third CNOT gate) and 0,then the outputs are B,C and BC. Similarly the TOFFOLI gates outputs will be obtained in the same manner as shown in the fig 3.8.Again the outputs of TOFFOLI gates are given as inputs to different FREDKIN gates. For the F1 FREDKIN Gate the inputs are ABC, $\overline{ABC}$ and 1, then the outputs are g, ABC + $\overline{ABC}$ and g where g is the garbage output i.e, the unused output. Similarly, the FREDKIN gate outputs will be obtained in the same manner. The SUM and CARRY outputs are obtained from F4 and F5 FREDKIN gates.

V REVERSIBLE FULL SUBTRACTOR- DESCRIPTION

Similarly, the one-bit subtractor as shown in Figure 3.9 is implemented to generate the difference and borrow output functions. The full subtractor using reversible PLA requires 3 CNOT gates and 10 TOFFOLI gates and 5 FREDKIN gates.

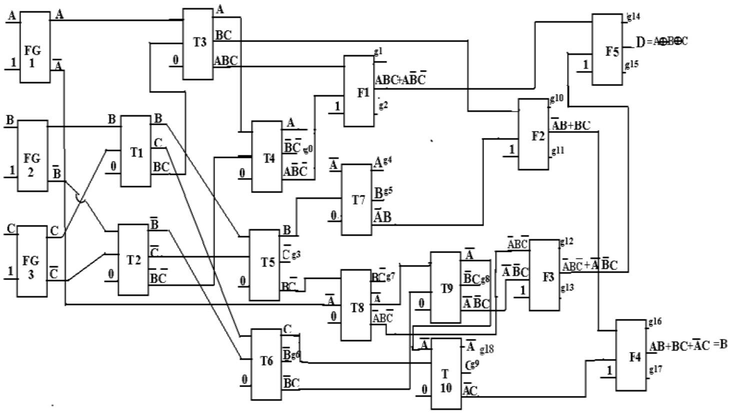


Fig 6: Design of reversible full subtractor using RPLA.

The Reversible full subtractor circuit has A, B and C are the inputs, difference and barrow are the corresponding outputs. The functioning of CNOT, CCNOT and FREDKIN gates used in the circuit resembles same as the outputs of the adder and multiplier circuits. Therefore, for the CNOT gate the inputs given are A and 1 then the outputs are A and Ā. Similarly the remaining inputs B,C are given to the respective two CNOT gates as B,1 and C,1 then the outputs are B, $\overline{B}$ and C , $\overline{C}$. These outputs are again given as inputs to TOFFOLI gates. For the T1 TOFFOLI gate the inputs are B, C (i.e. corresponding output of third CNOT gate) and 0,then the outputs are B,C and BC. Similarly the TOFFOLI gates outputs will be obtained in the same manner as shown in the fig 3.8. Again the outputs of TOFFOLI gates are given as inputs to different FREDKIN gates. For the F1 FREDKIN Gate the inputs are ABC, $\overline{ABC}$ and 1,then the outputs are g, ABC + $\overline{ABC}$ and g where g is the garbage output i.e, the unused output. Similarly, the FREDKIN gate outputs will be obtained in the same manner. The difference and borrow outputs are obtained from F4 and F5 FREDKIN gates. The gates will produce different outputs by which some of them are not used in the concerned outputs; those outputs are called as garbage outputs. The numbers of garbage outputs are 19 in which these are reduced when compared to existing rpla full subtractor. The difference between the Full adder and Full subtractor circuits is only the CARRY and BORROW outputs and the SUM is same as the output of DIFFERENCE.

IV.RESULTS AND DISCUSSION

Performance Parameters

In the previous chapters, it was discussed about the realization of reversible adders and subtractor using RPLA and is compared with the previous designs in terms of different quantum parameters. The performance parameters like gate count, quantum cost, and number of garbage outputs are obtained for the implemented circuits shown in figures 3.8 and 3.9 are compared with the existing Design. It was observed that, gate count is the main reduced factor for the proposed design because as it comprises of TOFFOLI gates. The performance calculations are tabulated in Table 1.1 and 1.2.

Table 1.1: Comparison of Performance Parameters for reversible full adder using RPLA

Quantum Parameters	**Existing Design[10]**	**Implemented Design**
Gate Count	70	18
Quantum Cost	170	78
Garbage Outputs	74	19
Constant Inputs	70	18

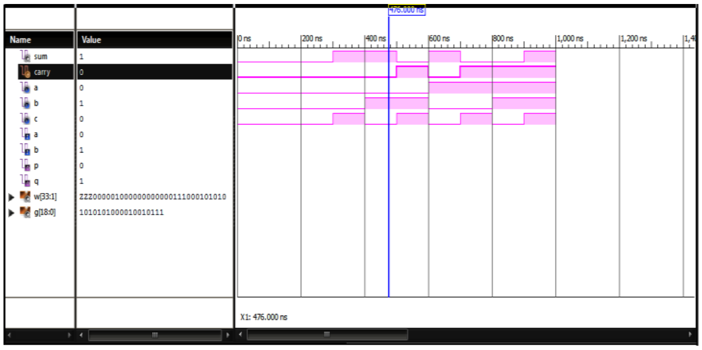
Table 1.2: Comparison of Performance Parameters for reversible full subtractor using RPLA

Quantum Parameters	**Existing Design[10]**	**Implemented Design**
Gate Count	70	16
Quantum Cost	170	78
Garbage Outputs	74	17
Constant Inputs	70	16

The proposed reversible full adder and full subtractor are implemented on Xilinx verilog HDL.

V.SIMULATION RESULTS

Simulation results of reversible full adder and reversible full subtractor. Fig. 7 and 8 show the simulation results of the realized circuits.

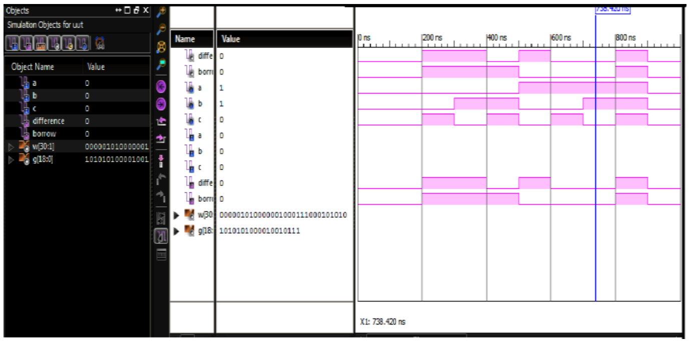


Name	Value
sum	1
carry	0
a	0
b	1
c	0
a	0
b	1
p	0
q	1
w[33:1]	ZZZ000001000000000000111000101010
g[18:0]	1010101000010010111

X1: 476.000 ns

Fig 7: Simulation result for Reversible full adder.

The resultant simulated values of the inputs and outputs given to the circuit are those values can be observed under **value** column, as shown and there by changing the values to the inputs the corresponding outputs are observed by simulation.



Objects

Simulation Objects for uut

Object Name	Value
a	0
b	0
c	0
difference	0
borrow	0
w[30:1]	000001010000001
g[18:0]	101010100001001

Name	Value
diffe	0
borro	0
a	1
b	1
c	0
a	0
b	0
c	0
diffe	0
borro	0
w[30:	000001010000001000111000101010
g[18:	1010101000010010111

X1: 738.420 ns

Fig 8: Simulation result for Reversible full subtractor.

It can be observed from these two simulated waveforms is that good response analysis for the detection of single missing gate faults(SMGFs). Testing for faults is also performed through software program, signatures are generated for all possible faults and are compared with good signature (signature of fault-free circuit) to count the number of faults detected.

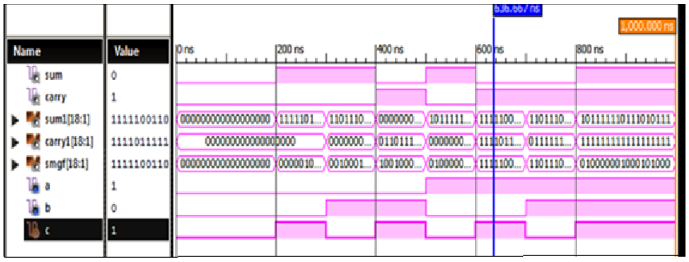


Fig: 6.3 Simulation result for reversible full adder with SMGFs.

Figures 6.3 and 6.4 shows the simulation of the testing reversible circuits of full adder and full subtractor, in which SMGF1 [18:1] indicates the tested pattern of sequence generated, sum[18:1] and carry[18:1] indicates good signature.

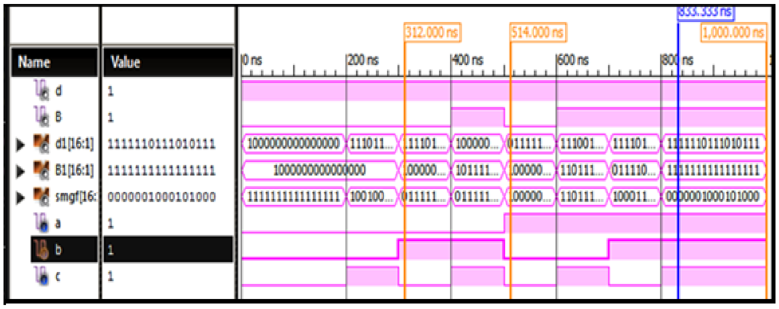


Fig 9: Simulation result for reversible full subtractor with SMGFs.

Fig 9 gives the(i.e. signature of fault-free circuit) of full subtractor and fig 10 gives the smgf1[18:1] are signatures of faulty full subtractor circuit ,in which SMGF1 [18:1] indicates the tested pattern sequence generated. The difference d1[18:1] and borrow b1[18:1] indicates comparison results of the circuit with and without faults.

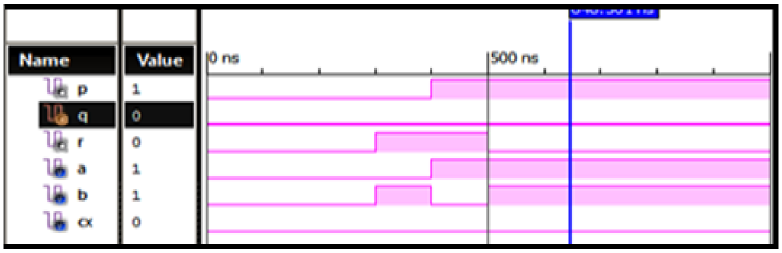


Fig 10: Simulation results for CNOT Augmented gate without any faults

For testing to be done, firstly the respective gates used in the circuits are augmented and the corresponding simulated outputs are shown in the fig 11 i.e., for CNOT gate, the augmented CNOT output without any stuck at and missing gate faults.

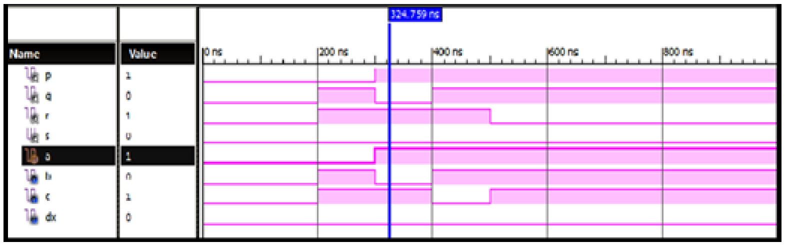


Fig 11: Simulation result for Augmented CCNOT gate.

Fig 6.6 shows the corresponding simulated outputs are for CCNOT gate, i.e, the augmented CCNOT output for detecting the stuck at and missing gate faults.

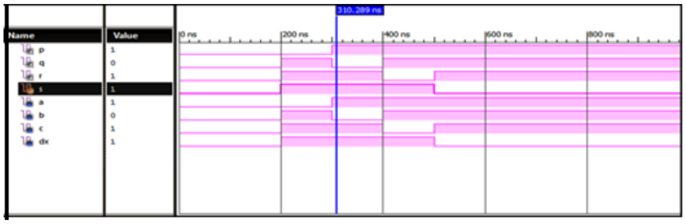


Fig 12: Simulation result for Augmented FREDKIN gate

Fig 12 shows the corresponding simulated outputs are for FREDKIN gate, i.e., the augmented output for detecting the stuck at and missing gate faults
For the basic gates in the realized circuits all the stuck-at and some of the missing-gate faults are tested with 100% fault coverage by applying the universal test vector. The universal test set is directly found without the need for running ATPG. The results with and without faults are shown in figures below.

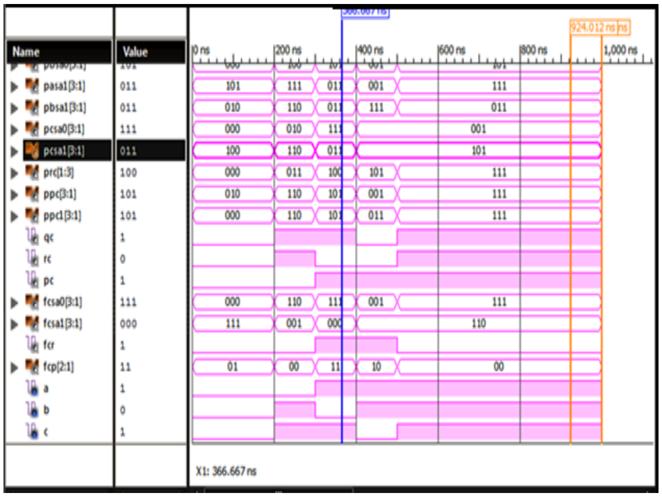


Fig 13: Simulation result for comparison of all faults in CNOT gate by using Universal test vector

Fig 13 shows the corresponding simulated outputs are for tested CNOT gate with UTV, i.e., the comparison of augmented CNOT outputs without faults and faulty Fredkin gate outputs there by showing all detected stuck-at faults and missing gate faults by using UTV(UNIVERSAL TEST VECTOR) algorithm. Thus, in the same manner testing is applied to the gates and then circuit behavior is observed by fault coverage analysis by applying a procedure to detect fault coverage i.e Fault coverage % = No .of faults detected/ Total no .of faults .i.e shown in the Table 1.2.

REFERENCES

1. R. Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development, 5, pp. 183 -191, 1961.

2. C.H. Bennett, “Logical Reversibility of Computation”, IBM J. Research and Development, pp. 525-532, November 1973.

3. E. Fredkin, T Toffoli, “Conservative Logic”, International Journal of Theory. Physics, 21(1982), pp.219 -253.

4. T. Toffoli., “Reversible Computing”, Tech memo MIT/LCS/TM -151, MIT Lab for Computer Science (1980).

5. Alberto LEPORATI, Claudio ZANDRON, Giancarlo MAURI," Simulating the Fredkin Gate with Energy{Based P Systems", Journal of Universal Computer Science, Volume 10,Issue 5,pp 600 -619.

6. Himanshu Thapliyal and M.B Srinivas, “Novel Reversible “TSG” Gate and Its Application for Designing Reversible Carry Look Ahead Adder and Other Adder Architectures”, Tenth Asia-Pacific Computer Systems Architecture Conference (ACSAC05), Singapore, October- 26, 2005, pp 805-817.

7. GSchrom, “Ultra Low Power CMOS Technology”, PhD Thesis, Technischen Universidad Wien, June 1998.

8. E. Knil, R. Laflamme, and G.J Milburn, “ A Scheme for Efficient Quantum Computation with Linear Optics”, Nature, pp 46 -52, Jan 2001.

9. M. P. Frank, “Introduction to reversible computing: motivation, progress, and challenges”, In Proceedings of the 2nd Conference on Computing Frontiers, pages 385–390, 2005.

10. Himanshu Thapliayal and Hamid R.Arabnia”center for vlsi and Embedded System Technologies,RPLA Using Fredkin and Feynman Gates For industrial Electronics and Applications” In the Proceedings of the 3rd conference on computing Frontiers -pages 250-255,2001.

11. M. Nielsen and I. Chaung, “Quantum Computation and Quantum Information”,Cambridge University Press, 2000. R.C. Merkle, “Two Types of Mechanical Reversible Logic”, Nanotechnology,4:114 -131,1993.

12. Vlatko Vedral, Adriano Bareno and Artur Ekert, “Quantum Networks for Elementary Arithmetic Operations”, arXiv:quant-ph/9511018 v1, nov 1995.
13. D. Maslov, G.W. Dueck, and D. M. Miller, “Techniques for the synthesis of reversible toffoli networks,” ACM Trans. on Design Automation of Electronic Systems, vol. 12, no. 4, 2007.
14. D. Maslov, G.W. Dueck, and D.M. Miller, “Toffoli network synthesis with templates,” IEEE Trans. on CAD, vol. 24, no. 6, 2005, pp. 807-817.
15. D.Y. Feinstein, M.A. Thornton, and D.M. Miller, “Partially redundant logic detection using symbolic equivalence checking in reversible and irreversible logic circuits,” in Design, Automation and Test in Europe, 2008, pp. 1378–1381.
16. Dey, Bappaditya, et al. "A novel design gate based low-cost configurable ro puf using reversible logic." 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2019.