HARDWARE IMPLEMENTATION OF REVERSIBLE FAULT TOLERANT COMBINED ADDER/SUBTRACTOR

Hemant Gour[1], Nitesh Dodkey[2], Siddarth Singh Parihar[3]
Dept. of Electronics and Communication Engineering

Abstract: Programmable reversible logic circuit is design style for nanotechnology and quantum computing with minimum heat generation, quantum cost and garbage output. Reversible logic helps in reducing the heat dissipation by eliminating the information loss. In this paper, a 4-bit reversible combined adder and sub tractors using DKG gate is proposed. A performance analysis is also performed over the proposed designed to synthesize the proposed logic circuit.

Keywords—Reversible Gates, Reversible Full adders, Fault Tolerant, Combined adder/subtractor

I. INTRODUCTION

Irreversible logic circuits dissipate heat in the amount of kT ln2 Joule for every bit of information that is lost irrespective of their implementation technologies, where k is the Boltzmann constant and T is the operating temperature [1]. Information is lost when the circuit implements nonbijective functions. Therefore in irreversible logic circuit the input vector cannot be recovered from its output vectors. Reversible logic circuit by definition realizes only those functions having one-toone mapping between its input and output assignments. Hence in reversible circuits no information is lost. According to [2] zero energy dissipation would be possible only if the network consists of reversible gates. Thus reversibility will become an essential property in future circuit design. Fault detection can generally be done in three methods: parity preserving method, parity generation method and dual rail online error detection methods. Among these the most commonly used method if parity preservation. This method id implemented at gate level. Hence there is no extra cost in design and verification effort [17]. Parity check is one of the widely used mechanisms for detecting single level faults. Thus, parity preserving circuit design is important for the development of fault tolerant reversible systems in nanotechnology. Unidirectional error detecting codes like parity code, hamming code, Reed Solomon code, Berger code and Bose Lin code can be used to develop fault tolerant circuit. Parity code is the simplest and cheapest error detecting code [18] as it requires only one error – checking bit to append to the information bits. This error – check bit is computed in such a way that the number of 1's in information bits along with the parity bit is made odd or even. Hardware overhead is very less and computation speed is high [19]. But it can detect only single errors or all odd numbers of errors in the information bits. The modified version of parity code is Hamming code. But when compared to parity code it requires addition hardware overhead. This technique also detects single bit error and double errors and all uni directional errors. Reec Soloman is a polynomial based error detection code [20] providing error correction capability. But at the cost of increased area and speed overhead when compared to Hamming code and cannot detect all unidirectional errors. Berger code and Bose Lin code are all systematic and separable code. Bose Lin code has fixed number of check bits and is not dependent on number of information bits [21]. It can detect uni directional errors. On the other hand Berger code is optimal, in terms of number of check bits required for I information bits, among all separable codes the detect unidirectional errors [22]. No extra decoders are required to extract information bits, when needed for processing form the code word. They detect all multiple, unidirectional errors. It is least redundant code among all unidirectional error detection codes [23]. In this paper, we have used Berger code method to incorporate fault tolerance to the combined adder/subtractor unit. DKG gate is used to implement combined FA (full adder) and FS (full subtractor) circuit, and then this FA/FS circuit is used to form ripple carry adder/subtractor circuit. We have also synthesized circuit to calculate garbage output, ancilla inputs, quantum cost, gate count and comparative study is also performed.

II. PRELIMINARIES

Reversible logic imposes many design constraints that need to be either ensured or optimized for implementing any particular Boolean functions. Firstly, in reversible logic circuit the number of inputs must be equal to the number of outputs. Secondly, for each input pattern there must be a unique output pattern. Thirdly, each output will be used only once, that is, no fan out is allowed. Finally, the resulting circuit must be acyclic. Any reversible logic design should minimize the followings:

- Garbage Output: outputs that are not used as primary outputs are termed as garbage outputs.
- Constants: constants are the input lines that are either set to zero(0) or one (1) in the circuit's input side
- Gate Count: number of gates used to realize the system
- Hardware Complexity: refers to the number of basic gates (NOT, AND & EXOR gate) used to synthesize the given function

III. BERGER CODE FOR ARITHMETIC CIRCUITS

The Berger code is a separable code and unordered code, insensitive to propagation delay of individual bits in the code word. Berger codes are introduced by J,M, Berger in 1961 [24]. The codes are optimal systematic codes that detect all unidirectional errors [25]. Berger codes are formed by appending a special set of bits, call the check bits to each word information. The check bits to each word of information. The check bits are created based on number of 1's in the original information. A Berger code of length 'n' will have 'I' information bits and K check bits [22] where

K = [log2(I + 1)]---(1)

and

n = 1 + K --- (2)

Berger codes use fewest number of check bits of the available separable codes.

Berger check reversible circuit design

The Berger check circuit proposed in this paper is based on the following logic.

Let two n-bit numbers be X = X_n ----------X_2X_1 and Y = Y_n ………. Y_2 Y_1. Let the sum and internal carries be S = S_{n-1} …….. S_1S_0 and C = C_n …….. C_2C_1.

Where X_i, Y_i, S_i, C_i {0,1}.

Let N(X) be the numbers of 1's in binary representation of X that is N(X_i) = X_i.

Hence the Berger check prediction equation [26] is as follows:

N(X) + N(Y) + Cin = N(S) + N(C) + Cout ----------- (3a)

Where Cin is the carry input and Cout = C_n

The modified form if equation 3a is as shown below:

N(S) = N(X) + N(Y) + Cin – N(C) – Cout ----------- (3b)

Berger check logic can both detect unidirectional and multiple errors. If there are multiple errors in the circuit such as flipping of two bits of same type, then there will be inequality in equation 3a. Hence error is detected. But Berger check prediction algorithm fails when both 1 and 0 are flipped in data word [27]. In such case error code are not detected.

IV. ADDER/SUBTRACTOR

In this section the design of combined adder subtractor is discussed, The ripple carry adder/subtractor consists of DKG gate, Feynman Gate, BVF gate and Peres gate. These gates are shown in figure 1, 2, 3 & 4.

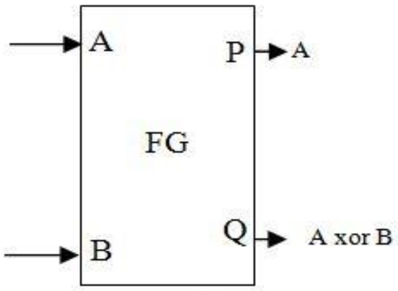

Figure 1: Feynman Gate (FG)

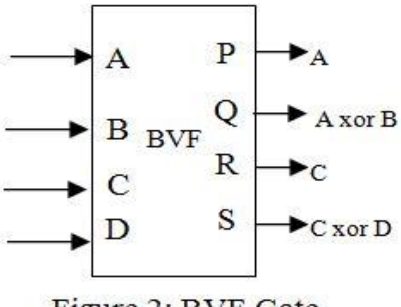

Figure 2: BVF Gate

The Feynman gate and the BVF gate are fan out gates as fan out is not allowed in reversible logic. When input B of Feynman gate is set to zero then the value of both outputs is equal to A. The quantum cost of Peres gate is 1. Similarly when input A and Input are connected together and then input B and Input D is set to zero then all outputs will be equal to zero. The qyantum cost of BVF gate is 2.

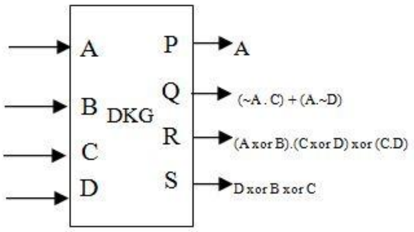

Figure 3: DKG gate

The DKG gate is used to perform full addition or full subtraction using input A, when A = 1 then R produces borrow and S produces difference, when A = 0 then R produces carry and S produces sum. P and Q outputs are garbage outputs in Ripple vary adder/Subtractor design. The quantum cost of DKG gate is 6.

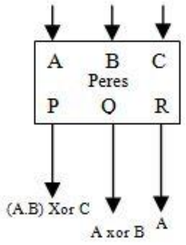

Figure 4: Peres Gate

The Peres gate is used to perform half adder, when the input C is set to zero, Q generates the sum and output P generates the carry. Now this half adder is used to implement full adder. So two Peres gates implements one full adder. Quantum cost of Peres gate is 4. Figure 5 shows the Peres gate half adder. The full adder circuit using Peres gate is shown in figure 6.

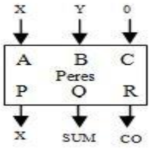

Figure 5: Half Adder using Peres Gate

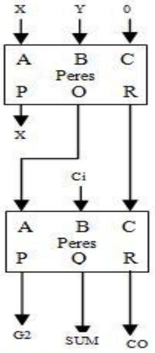

Figure 6: Full Adder using Peres Gate

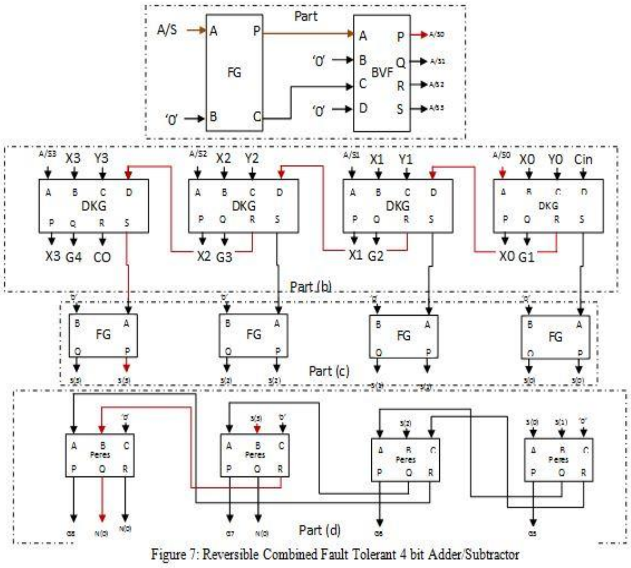

Figure 7: Reversible Combined Fault Tolerant 4 bit Adder/Subtractor

Reversible combined fault tolerant 4 bit adder/subtractor is shown in figure 7, this figure is divided into 4 parts namely: part (a), part (b), part (c) and part (d). Each part performs a distinguish task in the operation of addition and subtraction. Part (b) of figure 7 is the combined 4 bit ripple carry adder/subtractor, to perform the operation of addition the "A/S" input of the DKG gate must be set to zero and to perform the operation of subtraction, it must be set to one. Since fan-out is not allowed in reversible logic hence we cannot simply apply the single "A/S" selection signal to the four DKG gate. The solution to this problem is to design a fan-out circuit, the fan-out circuit is shown in part (a) of figure 7, the fan-out circuit consists of one Feynman gate (FG) and one BVF gate, these gate generates four copies (A/S0, A/S1, A/S2, A/S3) of the input "A/S" signal. The DKG gate generates four sum/difference signals (s(0), s(1), s(2), S(3)). The output of part (b) of figure 7 generates the sum/difference signals, these signals used by the fault tolerant signal to calculate the number of '1' in them. Now the sum/difference signals are needed to be used at two places namely: first the output of the circuit and secondly to the fault tolerant circuit. So fan-out circuit is again required, part (c) of figure 7 does the task using Feynman gate (FG), the "B" input of the FG gate is assigned to a constant '0' to produce two copies of the signal s(i). Four FG gates are employed to generate four copies of the signals. The fault tolerant circuit calculates the number of '1' in the sum/difference signal (s(0), s(1), s(2), S(3)), this is implemented using Peres gates. 1's counting unit is shown in figure 8, which consists of one full adder and two half adders.

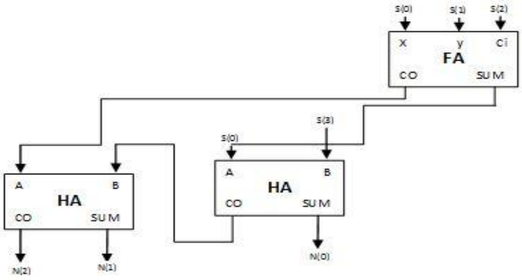

Figure 8: Berger check unit

The Berger check unit shown in figure 8 is implemented using Peres gates, the full adder (FA) is implemented using two Peres gates and the 2 half adder (HA) is implemented using two Peres gates as shown in part (d) of figure 7.

V. SYNTHESIS

This section discusses the synthesis of reversible circuit shown in figure 7. Ancilla inputs is the number of constant signals applied to the circuit to perform the task assigned to it. In figure 7, 3 ancilla inputs are used in part (a) of figure 7 to generate the copies of "A/S" signal, 4 ancilla inputs are used in fan-out circuit shown in part (c) of figure 7 for generating copies of sum/difference signal (s(0), s(1), s(2), S(3)), and 3 ancilla inputs are used in Berger check unit of part (d) of figure 7. So a total of 10 ancilla inputs are used in the proposed design.

Garbage output is the number of unused outputs generated during the operation, and the inputs regenerated are not garbage outputs. Four garbage outputs are generated by the Q outputs of DKG gates in ripple carry adder/subtractor namely: G1, G2, G3, G4, as shown in part (b) of figure 7. Four garbage outputs are generated by P outputs of Peres gates in Berger check circuit shown in part (d) of figure 7. So a total of 8 garbage outputs are generated by the proposed design.

Quantum cost is the number of primitive gates used in the design, the quantum cost of the design is calculated by adding the quantum cost of individual reversible gates. In this design 5 Feynman gates (FG) are used, the cost of each FG gate is 1 so total quantum cost for FG gate is 5, similarly 1 BVF gate will cost 2 quantum units, 4 DKG gate will cost 24 quantum units and 4 Peres gate will cost 16 quantum units, so the total quantum cost of the design is 47 units.

The delay units of the proposed design by adding the quantum cost of the critical path, the critical path of the proposed design is marked in red ink in figure 7. The total quantum cost of all these gates in 36, so the delay of the design is 36 units.

Gate count is the number of reversible gates, here in this design the gate count is 14.

Table 1 summarizes the synthesis of the proposed design and compares it with the other designs, as depicted in table 1 that proposed design is better than other designs available in literature. The design 1 and design 2 [28] is only of adder and same resources are being used by subtractor of the design. Both the designs [28] are fault tolerant.

Table 1: Design Synthesis Report

S. no	Parameter	Proposed Design	Design 1 adder [28]	Design 2 Adder [28]
1	Ancilla Inputs	10	9	12
2	Garbage outputs	8	2	8
3	Quantum Cost	47	54	72
4	Delay	36	46	66
5	Gate count	14	13	24

VI. CONCLUSION

In this paper design and implementation of 4 bit reversible fault tolerant combined adder/subtractor is discussed. The proposed design uses DKG gate to implement ripple carry adder/Subtractor and Berger check logic to implement fault tolerance for the circuit. The proposed is better than other designs available in literature in terms of number of garbage outputs, ancilla inputs, quantum cost and delay. In future this design can be modified to implement other fast adder/subtractor design like carry look ahead adder, carry chain adder, carry save adder and Kogge Stone adder.

REFERENCES

[1] R. Landauer, “Irreversibility and heat generation in the computing process”, IB J. Research and Development, vol. 5, pp. 183-191, 1961.

[2] C. H. Bennet, “Logical reversibility of computation”, IBM J. Research and Development, vol. 17, no. 6, pp. 525-532, 1973.

[3] M. S. Islam, M. R. Islam, M. R. Karim, A. A. Mahmud and H. M. H. Babu, “Variable block carry skip logic using reversible gates”, In Proc. of 10th International Symposium on Integrated Circuits, Devices & Systems, Singapore, pp 9-12, 8-10 September, 2004.

[4] M. S. Islam, M. R. Islam, M. R. Karim, A. A. M. and H. M. H. Babu, “Minimization of adder circuits and variable block carry skip logic using reversible gates”, In Proc. of 7th International Conference on Computer and Information Technology, Dhaka, Bangladesh, 26-28 December, 2004, pp. 378-383.

[5] M. S. Islam and M. R. Islam, " Minimization of reversible adder circuits" , Asian Journal of Information Technology, vol. 4, no. 12, pp. 1146-1151, 2005.

[6] M. S. Islam, M. M. Rahman, Z. Begum and M. Z. Hafiz, "Low cost quantum realization of reversible multiplier circuit", Information Technology Journal, vol. 8, no. 2, pp.208-213, 2009.

[7] B. Parhami , “Fault tolerant reversible circuits”, in Proceedings of 40th Asimolar Conf. Signals, Systems, and Computers, Pacific Grove, CA, pp. 1726-1729, October 2006.

[8] R. K. James, Shahana T. K., K. P. Jacob and S. Sasi, “Fault Tolerant Error Coding and Detection using Reversible Gates”, IEEE TENCON, pp. 1- 4, 2007.

[9] M. S. Islam, M. M. Rahman, Z. Begum, M. Z. Hafiz and A. A. Mahmud, “Synthesis of fault tolerant reversible logic circuits”, In Proc. IEEE International Conference on Testing and Diagnosis, Chengdu, China, 28-29 April, 2009.

[10] M. S. Islam and Z. Begum, "Reversible logic synthesis of fault tolerant carry skip BCD adder", Bangladesh Academy of Science Journal, vol. 32, no. 2,pp. 193-200, 2008.

[11] R. Feynman, “Quantum mechanical computers”, Optical News, vol. 11, 1985, pp. 11-20.

[12] E. Fredkin and T. Toffoli, “Conservative logic”, Intl. Journal of Theoretical Physics, pp. 219-253, 1982.

[13] T. Toffoli, “Reversible computing”, In Automata, Languages and Programming, Springer-Verlag, pp. 632-644, 1980.

[14] Peres, “Reversible logic and quantum computers”, Physical Review: A, vol. 32, no. 6, pp. 3266-3276, 1985.

[15] J. W. Bruce, M. A. Thornton, L. Shivakumaraiah, P.S. Kokate, X. Li, “Efficient adder circuits based on a conservative reversible logic gates”, In Proceedings of IEEE Computer Society Annual Symposium on VLSI, Pittsburg, PA, pp. 83-88, 2002.

[16] M. Haghparast and K. Navi, “A novel fault tolerant reversible gate for nanotechnology based systems”, Am. J. of App. Sci., vol. 5, no.5, pp. 519-523, 2008.

[17] Parhami, Behrooz. "Fault-tolerant reversible circuits." Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on. IEEE, 2006.

[18] Boudjit, M., M. Nicolaidis, and K. Torki. "Automatic generation algorithms, experiments and comparisons of self-checking PLA schemes using parity codes." Design Automation, 1993, with the European Event in ASIC Design. Proceedings.[4th] European Conference on. IEEE, 1993.

[19] Stojcev, M. K., and M. D. Krstic. "Parity error detection in embedded computer system." Telecommunications in Modern Satellite, Cable and Broadcasting Service, 2001. TELSIKS 2001. 5th International Conference on. Vol. 2. IEEE, 2001.

[20] Cardarilli, Gian-Carlo, et al. "Concurrent error detection in Reed–Solomon encoders and decoders." IEEE Transactions on very large scale integration (VLSI) systems 15.7 (2007): 842-846.

[21] Das, Debaleena, et al. "Low cost concurrent error detection based on module weight-based codes." On-Line Testing Workshop, 2000. Proceedings. 6th IEEE International. IEEE, 2000.

[22] Abdulhadi, A. H., and Ali H. Maamar. "Self Checking Register File Using Berger Code." 6th WSEAS Interntional Conference on Circuits, Systems, Electronics, Control, and Signal Processing. 2007.

[23] Kavousianos, Xrysovalantis, and Dimitris Nikolos. "Novel single and double output TSC Berger code checkers." VLSI Test Symposium, 1998. Proceedings. 16th IEEE. IEEE, 1998.

[24] Berger, Jay M. "A note on error detection codes for asymmetric channels." Information and Control 4.1 (1961): 68-73.

[25] Burg, Ariel, and Osnat Keren. "On the Robustness of Berger Codes against error injection attacks." European Cooperation in Science and Technology (2013).

[26] Lo, J-C., Suchai Thanawastien, and T. R. N. Rao. "Concurrent error detection in arithmetic and logical operations using Berger codes." Computer Arithmetic, 1989., Proceedings of 9th Symposium on. IEEE, 1989.

[27] Ossi, E. J., et al. "Soft-error mitigation at the

architecture-level using berger codes and instruction repetition." Proceedings of the IEEE Workshop on Silicon Errors in Logic–System Effects (SELSE'09). 2009.

[28] Jayashree, H. V., S. Ashwin, and V. K. Agrawal. "Berger check and fault tolerant reversible arithmetic component design." Electron Devices and Solid-State Circuits (EDSSC), 2015 IEEE International Conference on. IEEE, 2015.