Design of reversible circuits with high testability

H.M. Gaur✉ and A.K. Singh

A new method of designing reversible logic circuits which can be adopted by any synthesis technique to produce parity preserving reversible circuits based on Multiple Controlled Tofolli gates is proposed. The designed circuit using proposed methodology is easily testable by checking the input and output parity. A set of benchmark circuits and corresponding testable designs are implemented. The results under testable designs show an average reduction of 32% in operating cost as compared to prior work.

Introduction: Miniaturisation does not seem to be promising method in the development of compact electronic devices due to the breakdown of Dennard scaling [1]. Reversible logic is one of the foregrounds to meet the destiny of future electronics based on quantum computation. These circuits guarantee nearly energy free computation by preventing loss of information in the form of heat as in irreversible logic circuits [2, 3]. The researchers are at par with the latest innovations in this area to develop combinational and sequential logic circuit designs which are based on different gate libraries. A reversible gate library defines a set of basic reversible gates which are used individually or in combination to synthesise a reversible circuit in the manner they produce efficient results in terms on performance parameters. Multiple controlled Tofolli (MCT) often called as *k*-CNOT gates, multiple controlled Fredkin (MCF), NOT-CNOT-Tofolli (NCT) are widely used gate libraries [4, 5] and the parameters include gate cost, quantum cost, ancilla input and garbage output. These parameters are directly related to the power dissipation and the size which well defines the operating cost of any designing methodology [6]. Parity checking is found most favourable method of testing reversible logic circuits because they perform fully controllable and observable operations due to the ability of producing bijective functions. Several online testing methods base on parity checking have been proposed for their recognition by the detection of single/multiple bit faults as any type of fault occurrence will result in change of single/multiple values of bits on the wires of the circuit [6]. These methods can be categorised as designing using novel gates [7, 8] and designing using a modification of standard circuits or gates [6, 9, 10]. In this work, a new method of designing *k*-CNOT-based reversible circuit is proposed, which produces parity preserving circuits rather than converting a standard circuit by means of novel gates or any modification in order to incorporate testability feature. The testability of these circuits can be achieved more easily by checking the input and output parity without any algorithm formulation under single bit fault detection. The circuits produced are incorporated with high testability which can be adopted by any synthesis method to induce online testability features.

Proposed design methodology: *k*-CNOT gates are taken into the formation of a new design methodology to produce parity preserving reversible circuits. A *k*-CNOT gate has *k* number of control inputs and one target input *t* as shown in Fig. 1 to form a reversible gate of size of size $(k+1)\times(k+1)$. The gate passes all the inputs to its control directly to the output and the final output $f(t, k_n)$ can be given by (1). The gate with $k=0$ is called as a NOT gate whose output is complement of input *t*.

$$f(t, k_n) = (k_1 \cdot k_2, \ldots, k_n) \oplus t \tag{1}$$

The method of designing a reversible circuit is done in sets of two *k*-CNOT gates (*set*1, *set*2, …*setn*) as shown in Fig. 2. In each set, first *k*-CNOT gate is placed on the wires in accordance with the Boolean expression to be generated at the output. The second gate should be of the same size as that of first gate whose control inputs will be fixed as that of the first gate in a set. The target input may take any place on *n* wires of the circuit except the place assigned to the previous gate to avoid redundancy as shown in *set*2 and *setn*. The control input of second gate is placed such that it produces a necessary function for producing final output. If the control of second gate does not find any function to build, it can take a constant input wire to create required function. In case of a NOT gate, the second gate can take any place on the wires except that of previous gate as shown in *set*1. The output functions should be taken in the ESOP-based format which provides easy solution in the way of placement of gates on the wires. The circuit produced will be parity preserving showing its inbuilt testability feature which facilitates the construction of online testable circuits. Fault detection can be achieved easily by generating input–output parity of the circuit. For an *n* wire reversible circuit with input wires x_i and output wires y_i, the parity *P* can be generated by (2). Here, $\sum$ denotes EXOR sum rather than usual sum. For example, an rd32 benchmark has four input and output wires, the parity *P* can be given by (3)

$$P = \left(\sum_{i=1}^{n} x_i\right) \oplus \left(\sum_{i=1}^{n} y_i\right) \tag{2}$$

$$P = (x_1 \oplus x_2 \oplus x_3 \oplus x_4) \oplus (y_1 \oplus y_2 \oplus y_3 \oplus y_4) \tag{3}$$

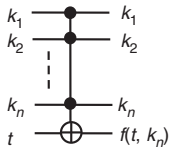

Fig. 1 *k-CNOT gate*

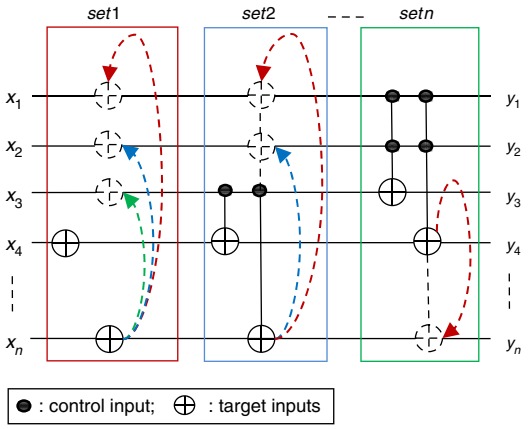

Fig. 2 *Proposed design methodology*

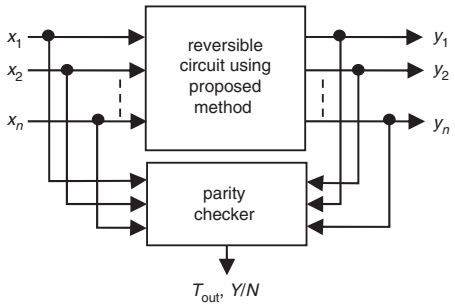

Fig. 3 *Fault detection*

Design for testability: The faults can be identified by checking the input and output parity by using a parity checker circuit as shown in Fig. 3. Here, $(x_1, x_2, \ldots, x_n)$ are the primary input wires to the circuit and $(y_1, y_2, \ldots, y_n)$ are the output wires of the reversible circuit constructed using the proposed method. The parity *P* will always result in logic value 0 for a parity preserving circuit. Hence, if any single bit fault occurs in the circuit, the output T_{out} of parity checker will flips from logic value 0 to 1.

In reversible circuits, parity checking can be achieved by simply EXORing all the inputs and outputs by means of 1-CNOT (CNOT) gates. These gates are placed from each wire to new wire before and after whole circuit. The input to the new wire added is assigned a logic value 0, so that the output will be logic value 0 for the correct operation of the circuit. If any bit fault occurred in the circuit, the value at the output of new wire will change to logic value 1 which detects its occurrence.

Algorithm: The proposed algorithm covers both the steps of designing and testability for reversible logic circuits. For a given input variables v_i and output Boolean function v_o in ESOP-based format, the algorithm will produce online testable reversible circuit C^T which provides the

location of gates on n wires of a circuit. The placement of k-CNOT gates (Ψ) is done in the set of two gates. For every set in the generation of output function, the gates are placed in the manner that it produces a necessary function to obtain exact output from all output function list (L). The control input of second gate will take the same place as of previous gate, but the target input will check for all wires except the wire assigned for previous gate. If the control of second gate did not find any option from the list, it will take another constant input wire (m) else it will connect to another fixed wire with constant input 0 which raise the *flag* bit and terminate. In each set, the control of second gate will meet this termination criterion (T). The register C^T and n will be updated at each cycle. After the construction of L, CNOT gates are added from each wire to a new wire with constant input 0 before and after the whole circuit (ϕ) to produce testable circuit.

Input: v_i and v_o
Output: C^T
Step 1: *Initialisation*
(a): *assign* $n = max(v_i, v_o)$
(b): *initialise* $C^T = 0$, L and *flag*
(c): *assign* input variables position on the wires
Step 2: *Output functions construction*
(a): *perform* operation Ψ and *update* C^T
(b): *repeat* 2(a) *until* T
(c): *update* n
if flag = 0
$n = n + m$
else
$n = n + m + 1$
Step 3: *repeat step* 2 *until* $|L| = 0$
Step 4: *perform* operation ϕ and *update* C^T
$n = n + 1$
Step 5: *exit*

Results and comparison: A set of benchmark circuits is designed using proposed method and implemented on RC-viewer tool [4]. The designs of four reversible benchmarks using the proposed method are depicted in Figs. 4*a–d*. The results based on their performance parameters are listed in Table 1, where n represents the number of wires, N symbolises gate count, QC is quantum cost and GO signifies garbage output.

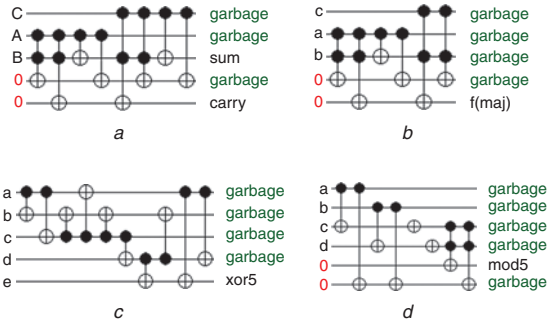

Fig. 4 *Designs using proposed methodology*
a rd32
b majority
c xor5
d mod5

Table 1: Performance parameters of proposed designs

Circuit	Proposed method			
	n	N	QC	GO
rd32	5	8	20	3
majority	5	6	20	4
xor5	5	10	10	4
alu	10	19	45	11
ham3	4	7	13	1
mod5	6	8	16	5
$n$th_prime3	4	8	12	1
permanent_2 × 2	8	6	30	3

The circuits obtained by proposed methodology are converted into their testable designs in accordance with the given Design for Testability (DFT) methodology under the detection of single bit faults. The results and comparison of testable circuits with that of previous work [6] on the basis of performance parameters are listed in Table 2. The previous method utilises the principle of modification of gates into their respective extended Tofolli gates (ETG) and several CNOT gates to preserve the parity of circuit on an extra wire which results in high operating cost. It can be seen that present work shows a reduction of 32% in operating cost as compared to previous work under design for testability.

Table 2: Comparison of proposed DFT with [10]

Circuit	Previous work [10]				Proposed DFT			
	n	N	QC	GO	n	N	QC	GO
rd32	5	20	45	3	6	16	28	3
majority	7	36	154	5	6	14	28	4
xor5	6	22	22	4	6	20	20	4
alu	12	58	82	10	13	38	64	11
ham3	4	21	25	0	5	14	20	1
mod5	6	25	29	4	7	18	26	5
$n$th_prime3	4	18	22	0	4	15	19	1
permanent_2 × 2	7	21	54	0	9	21	45	3
Average	7	32	62	4	8	22	36	5

Conclusion: A new approach of designing k-CNOT based reversible circuit is presented. The methodology produces parity preserving reversible circuits which are easily tested by checking the input and output parity by means of CNOT gates on an extra wire. The method shows that the circuits produced are incorporated with testability feature rather to put extra efforts in converting original circuit into their testable form. Hence, the method provides solutions to both the problem of designing and testability of reversible circuits which can be adopted by any synthesis algorithm to produce parity preserving circuits to incorporate testability features. The results depict the efficiency of the work in this area. The reduction in operating cost using the best synthesis algorithm will be the primary concern in these types of reversible logic design to meet the requirements of future electronics.

Submitted: *18 January 2016* E-first: *8 June 2016*
doi: 10.1049/el.2016.0161
One or more of the Figures in this Letter are available in colour online.
H.M. Gaur (*Department of Electronics and Communication Engineering, NIT Kurukshetra, Haryana, India*)
✉ E-mail: leoharimohan84@gmail.com
A.K. Singh (*Department of Computer Applications, NIT Kurukshetra, Haryana, India*)

References

1 Mack, C.: 'The multiple lives of Moore's law-cover story', *IEEE Spectr.*, 2015, **52**, (4)
2 Bennett, C.H.: 'Logical reversibility of computation', *IBM J. Res. Dev.*, 1973, **17**, (6), pp. 525–532
3 Landauer, R.: 'Irreversibility and heat generation in the computing process', *IBM J. Res. Dev.*, 1961, **15**, (3), pp. 183–191
4 Maslov, D., Dueck, G., and Scott, N.: 'RC viewer'. http://www.cs.uvic.ca/~dmaslov, accessed December 2015
5 Reversible Benchmarks: http://www.revlib.org, accessed December 2015
6 Gaur, H.M., Singh, A.K., and Ghanekar, U.: 'A review on online testability for reversible logic', *Proc. Comput. Sci.*, 2015, **70**, pp. 384–391
7 Vasudevan, D.P., Lala, P.K., Di, J., and Parkerson, J.P.: 'Reversible logic design with online testability', *IEEE Trans. Instrum. Meas.*, 2006, **55**, (2), pp. 406–414
8 Thapliyal, H., and Vinod, A.P.: 'Designing efficient online testable reversible adders with new reversible gate'. Proc. IEEE Int. Symp. Circuits and Systems, 2007, pp. 1085–1088
9 Noor Mahammad, S.K., and Veezhinathan, K.: 'Constructing online testable circuits using reversible logic', *IEEE Trans. Instrum. Meas.*, 2010, **59**, (1), pp. 101–109
10 Nayeem, N.M., and Rice, J.E.: 'Online testable approaches in reversible logic', *J. Electron. Test.*, 2013, **29**, (6), pp. 763–778