

Implementation of CMOS Logic Circuits with Perfect Fault Detection Using Preservative Reversible Gates

Sajjad Parvin and Mustafa Altun

Dept. Electronic and Communication Engineering, Istanbul Technical University, Istanbul, Turkey
sajjadparvin.stu93@gmail.com, altunmus@itu.edu.tr

Abstract—In reversible circuits, a fault as a change in logic value at a circuit node always alters an output logic value, so observability of faults at the output is 100%. In other words, reversible circuits are latent-fault-free. Our motivation is to incorporate this unique feature of reversible circuits to design CMOS circuits having perfect or 100% Concurrent Error Detection (CED). For this purpose we propose a new, fault preservative, and reversible gate library called Even Target - Mixed Polarity Multiple Control Toffoli (ET-MPMCT). By using ET-MPCT, we ensure that the parity, even or odd, is preserved at all levels including the output level unless there is a faulty node.

Our design strategy has two steps for a reversible function: 1) implement the reversible functions with the ET-MPMCT library; and 2) apply reversible-to-CMOS gate conversion. In case an irreversible function needs to be synthesized then its reversible form is used followed by the two design steps. As a result, we have come up with a CMOS circuit having 100% CED. The performance of our approach is compared with other CED schemes in the literature in terms of area, detection rate, and power consumption. Simulations are done with Cadence Genus tool using TSMC 0.18 μm technology. Clearly, results are in favor of our proposed technique.

I. INTRODUCTION

Research on CED is mainly driven by the need of reliability-critical applications where the data integrity is of a crucial importance such as aerospace applications, online IoT maintenance, etc. Hence, any fault at any intermittent node must be detected at the output. However, utilizing conventional irreversible logic to achieve 100% concurrent fault detectability is far fetched due to the existence of latent-faults in circuits. This is due to the fact that the conventional CMOS logic do possess “don’t care” conditions which led to existence of latent-faults in the network, which are defined as faults that are dormant and might not cause a problem for current operation but it might be destructive for next operation [1].

To demonstrate the effect of latent fault in a conventional CMOS circuit consider an occurrence of a switching fault in a 2-input AND gate; causing an input node value to have a transition of $0 \rightarrow 1$ or $1 \rightarrow 0$. Only 50% of the times the fault occurred at the input will propagate to the output. The detection rate worsens as the number of inputs for the AND gate increases; in the 3-input AND gate only 25% of the times a switching fault propagate to the output. This low detection rate can be problematic for some applications using conventional irreversible CMOS circuits.

On contrary, reversible logic has 100% fault observability at the output and no fault can be masked at the output. This is

because reversible gates have bijective input-to-output relation, they have no “don’t care” condition. Hence any switching fault in a circuit node will be propagated to the output. Therefore, reversible circuits do not have latent switching faults. Our goal is to utilize the 100% observability of fault at the output and by constructing circuit entirely out of parity preservative gates, 100% fault detection is achieved. To demonstrate practicality of our scheme, we convert our reversible parity preservative circuit to its CMOS counterpart with perfect online fault-detection capability.

To make an irreversible logic circuit more robust toward fault, many approaches have been studied. One conventional approach is the use of Double Modular Redundancy or Triple Modular Redundancy (TMR). However, these approaches have huge area overhead. To tackle the DMR and TMR area overhead, several code based CED techniques have been proposed in the literature such as weight-based codes [2], Bose-Lin codes [3], etc. To further minimize the area for CED, a logic implication based fault detection schemes is proposed [4]. This technique yields a good detection rate with low area overhead, but like previously mentioned techniques, it can not detect all faults in the circuit.

Fault detection schemes in realm of reversible logic is investigated in [5]. It is shown that there are several reversible parity preservative gates. Though these gates are fault-free in theory, they are so complex that their CMOS implementation might neither be fault-free nor cost effective. Also in [5], a synthesis technique based on parity preservative gate is proposed using Fredkin gates. It is shown, Fredkin Synthesized (FS) fault-free CMOS circuit implementation results in huge area overhead. To compensate area overhead of FS circuits with near optimal fault detection rate, Single Parity (SP) and Hamming distance coding schemes are proposed using Toffoli gates in [5].

In this work, by exploiting reversible logic, we propose a new cost effective Even-Odd Preservative (EOP) gate library. By definition an EOP gate preserves evenness/oddness of applied inputs in terms of the number of 1’s at the output. Also we propose a conversion technique which converts a given synthesized reversible circuit to a fault preservative one using our EOP gate library. By utilizing this approach by almost doubling the size of any given circuit we can achieve 100% error detection concurrently. It is worth to mention that the idea described in this paper is similar to the SP scheme in [5], but CMOS realization of the described Single Parity scheme is not parity preservative. In this work we have analyzed the

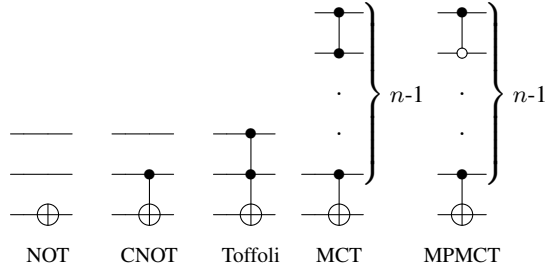


Fig. 1. Circuit representations of all the gates in the MPMCT gate library.

effectiveness of the SP scheme and proposes a new gate library upon the aforementioned scheme.

This paper is organized as follow. Section II provides background on reversible logic and reversible gates. Section III describes our proposed EOP gate library and the conversion technique. Section IV describes our procedure to convert a reversible logic to its CMOS fault-free counterpart. Section V presents the simulation methodology of our proposed technique and other CED techniques in the literature. Section VI concludes the paper including our future aim and direction.

II. PRELEMINARIES

Unlike irreversible functions where input and output have injective or surjective relation, reversible functions' input and output have bijective relation. Bijective mapping of input and output in reversible logic allows it to deduce input values from output. A reversible function can be realized by a **reversible circuit** consisting of reversible gates. The generalized and conventional reversible gate library which we use in this study, is called **Mixed Polarity Multiple Control Toffoli (MPMCT)** gate library. Definition of gates are as follow, with corresponding symbols given in Figure 1 where symbols \bullet , \circ and \oplus denote positive control, negative control and Toffoli target line, respectively.

- **NOT**: a 1-bit gate performing NOT operation.
- **CNOT**: a 2-bit gate performing 1 bit NOT operation on its target bit iff its control bit is 1.
- **Toffoli**: a 3-bit gate performing 1 bit NOT operation on its target bit iff its control bits are both 1.
- **Multiple Control Toffoli**: an n -bit gate, $n = 1, 2, 3, 4, \dots$, performing 1 bit NOT operation on its target bit iff all of its control bits are 1.
- **Mixed Polarity Multiple Control Toffoli**: an n -bit gate, $n = 1, 2, 3, 4, \dots$, performing 1 bit NOT operation on its target bit iff all of its positive control bits are 1 and all of its negative control bits are 0.

III. PROPOSED EOP PRESERVATIVE GATES AND CONVERSION TECHNIQUE

A. Proposed EOP Gates

By definition a reversible gate is **parity preservative** iff a reversible gate with inputs I_1, I_2, \dots, I_n and outputs O_1, O_2, \dots, O_n , satisfy $I_1 \oplus I_2 \oplus \dots \oplus I_n = O_1 \oplus O_2 \oplus \dots \oplus O_n$ where \oplus represents an XOR logic operation. Moreover, in reversible logic we have two distinct family of parity preservative gates. First type of gates when it is used to synthesize a circuit

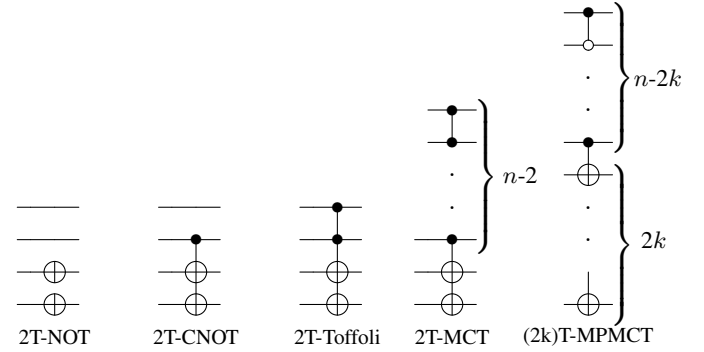


Fig. 2. Circuit representations of all the gates in the ET-MPMCT gate library.

utilizing solely those gates such as Fredkin gate [5], you must add as many as possible garbage bits to achieve conservative function. In a sense conservativity is achieved by having equal number of 1's at both input and output rows of a function. The second type is Even-Odd Preservative (EOP) function. This means that it suffices to add one garbage bit and make each rows of function's truth table even or odd in terms of number of 1's in each input and output row of truth table. It is obvious that the latter type parity preservative functions have superiority, since they require less garbage bits and as a result less costly gates to synthesize a given function.

To achieve EOP and less costly EOP gates in comparison to the literature parity preservative gates, we've modified the MPMCT gate library to achieve ET-MPMCT. In this approach by a set of control signals, we control even number of targets of a Toffoli gate at the same stage, as shown in Figure 2. This extra target bits guaranty that upon applying any input, having even or odd number of 1 to any gate in this library, the Evenness or Oddness of the computed 1's at the output will be the same as input. In a general sense any Toffoli gate with even number of target bit can serve as EOP gate. Hence by exploiting this feature we can achieve 100% CED.

B. Conversion of a MPMCT synthesized circuit to a EOP Circuit Using our Proposed Gate Library

The following lemmas are from [5]. Lemma 1 demonstrates the reason behind reversible circuits being latent-fault-free.

Lemma 1. *A switching fault ($0 \rightarrow 1$ or $1 \rightarrow 0$ transition) in a node of a reversible circuit always results in a change/transition at the output value.*

And since reversible logic is latent-fault-free, we would like to utilize them alongside with parity preservative gates to achieve perfect fault detection using parity preservative gates.

Lemma 2. *Consider a reversible circuit consisting of only preservative gates. For this circuit, 100% fault detection is possible for a switching fault occur in a node of the circuit.*

Now, that we have proven that reversible circuits are latent-fault-free and our gates are EOP, we can go through our proposed conversion technique. Our proposed scheme for synthesis using our ET-MPMCT gate library is straight forward. Steps of synthesis a function to an EOP gates synthesized circuit can be described as follow. For any given function, first

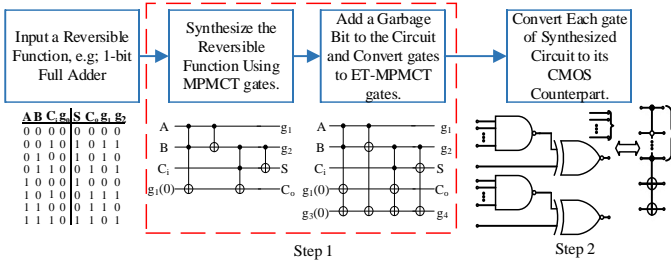


Fig. 3. Design flow of how to convert a function to a perfect fault tolerant circuit.

we need to check whether it is a reversible or not. If it is irreversible then garbage bits are added to the function to satisfy one-to-one matching constraint of reversible logic. Then we synthesize the function with conventional reversible synthesis technique such as exact, TBS, etc [6]. Up to here, We have the synthesized circuit using solely MPMCT gates which is not a parity preservative gate library. To make it parity preservative we add an extra garbage bit to the synthesized circuit. Then for each gate in our circuit we substitute it with its counterpart gate in our ET-MPMCT gate library which its second target bit is mapped to this newly added line. Eventually, by applying this procedure to all gates, EOP condition is satisfied for the whole circuit. Furthermore, we utilize 1-bit full adder as an example to demonstrate our approach.

Example 1. According to design flow shown in Figure 3, first we define a reversible function that we want to synthesize with perfect fault detection. In this example we will use reversible 1-bit full adder. Next we synthesize our reversible function using MPMCT gates shown in Figure 3. Afterwards we need to convert the MPMCT synthesized circuit to an EOP circuit. By adding an extra garbage bit to the circuit with a value of 0 or 1 where, in this case we assign 0 to this newly added garbage bit to our MPMCT synthesized circuit. Then we proceed through the circuit stage by stage and replace each MPMCT gate with its counterpart from our ET-MPMCT gate library. In the first stage we have a Toffoli gate, where we can substitute it with 2T-Toffoli gate in our ET-MPMCT gate library with its first target left on its original bit line and place the second target bit on the newly added garbage bit. This procedure is applied to the rest of the circuit's stages. The resultant EOP circuit is shown in step 1 of our design flow in Figure 3.

IV. CMOS IMPLEMENTATION OF PROPOSED EOP REVERSIBLE GATES

In this section we convert the produced circuits based on our synthesis technique using ET-MPMCT gate library to its CMOS realization. The goal is to achieve 100% concurrent error detectability using CMOS gates.

In conventional CMOS gates such as 2-input NAND gate, for three input values, output is mapped to value 1. Hence upon occurrence of fault at input node, the fault might not be carried to the output and get masked at the output. In other words, even though network is faulty, the NAND gate might produce a correct result. This means that the gate is not aware of the fault at its input. This case holds true for NOR, OR and AND gates as well. On the other hand, XOR, XNOR and Inverter gates distinguish the occurrence of fault at their inputs.

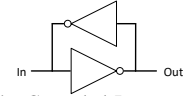


Fig. 4. Cascaded Inverter design.

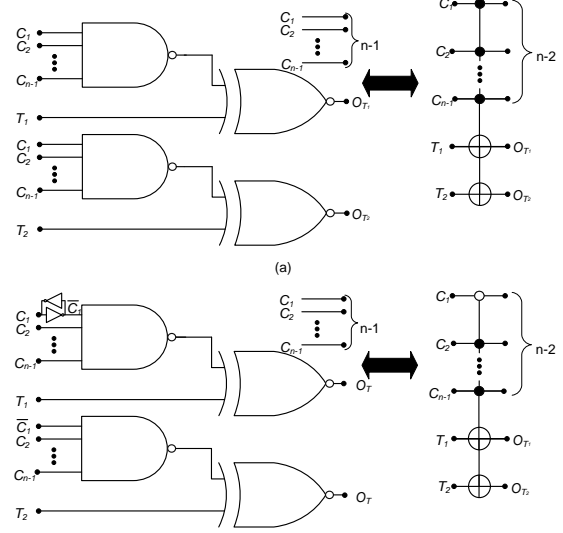


Fig. 5. CMOS implementation of our proposed ET-MPMCT gate library (a) ET-MPMCT gate CMOS realization with all positive control signal (b) ET-MPMCT gate CMOS realization with both negative and control signal.

But the problem is, these CMOS fault aware gates do not form a set of universal gate library. In order to achieve fault aware CMOS realization using reversible gates, we utilize NAND, XNOR, XOR and Inverter to satisfy the EOP condition of our reversible gates in CMOS realm. In particular, XOR gate is just used to implement (2k)T-CNOT gate and for the rest of the gates in our proposed gate library, we utilize XNOR gate based implementations. Moreover, instead of conventional Inverter gate we utilize a cascaded Inverter gate which is shown in Figure 4 where feedback Inverter must have bigger size to drive the input of other Inverter upon occurrence of fault, to hinder occurrence of “don't care” condition at the input of NAND gate in case we have negative input reversible gates. CMOS implementation of our ET-MPMCT gate library is shown in Figure 5. It should be noted that utilizing one NAND gate to control both XNOR gates is hazardous because in case of occurrence of a fault at the output of the NAND gate, results in masking of fault at the output side of our gate. Consequently, we need to utilize two NAND gate to achieve internally fault preservative CMOS circuit.

V. EXPERIMENTAL RESULTS

For our proposed gates we have proposed a conversion technique to convert any given reversible circuit synthesized using MPMCT gate library to an EOP circuit. We have demonstrated our results both on irreversible benchmarks and reversible benchmarks. For irreversible benchmarks we have used 1-bit Full Adder, 1-bit Subtractor and 2-bit Multiplier benchmarks and reversible benchmarks are chosen from [7] and [8]. We have compared the performance of our technique with the other approaches in the literature in terms of CMOS cost, power cost and fault-detection rate. For CMOS cost and power analysis we used Cadence Genus tool using TSMC

TABLE I

A COMPARISON AMONG CED TECHNIQUES WITH OUR PROPOSED APPROACH IN TERMS OF AREA AND POWER CONSUMPTION.

Benchmark	DMR area (μm^2)	[3] area (μm^2)	[4] area (μm^2)	[5] SP area (μm^2)	[5] FS area (μm^2)	Our Method area (μm^2)	DMR power (nW)	[3] power (nW)	[4] power (nW)	[5] SP power (nW)	[5] FS power (nW)	Our Method power (nW)
1-bit FA	1498	660	792	357	6307	558	316	139	174	195	1410	214
1-bit Sub.	1498	375	792	335	5336	765	316	77	174	464	1098	379
ex-1	536	660	312	161	4116	362	105	133	58	80	941	94
3_17	536	803	334	464	6325	743	328	163	180	224	1410	296
Peres Gate	606	642	369	178	2659	278	124	123	68	88	586	97
Miller Gate	1106	660	619	464	1978	615	197	142	108	190	454	218
4_49	3568	1766	1828	1088	19249	1840	726	346	399	576	3917	795
4b15g	4352	1784	2242	1231	11069	1936	666	370	366	592	2489	829
aj-e11	3318	1712	1724	1124	11733	1487	524	324	288	426	2387	618
hwb4	4280	1873	2206	1034	5033	1692	684	397	376	775	998	884
2-bit Mult.	964	963	570	2658	10273	3410	200	198	110	1673	2064	1874
Average	2024	1082	1072	827	7643	1244	381	219	209	480	1614	573

TABLE II

A COMPARISON AMONG DETECTION RATE (D.R.) OF CED TECHNIQUES IN LITERATURE AND OUR PROPOSED TECHNIQUE.

Benchmark	DMR	[3]	[4]	[5] SP	[5] FS	Our Method
1-bit FA	43.10%	65.42%	55.54%	82.10%	100%	100%
1-bit Sub.	57.55%	34.28%	50.20%	83.26%	100%	100%
ex-1	46.46%	78.58%	66.76%	82.10%	100%	100%
3_17	69.51%	64.21%	58.67%	83.41%	100%	100%
Peres Gate	49.91%	82.02%	61.03%	80.40%	100%	100%
Miller Gate	67.59%	66.71%	65.35%	88.91%	100%	100%
4_49	45.08%	59.44%	66.91%	83.41%	100%	100%
4b15g	42.68%	43.91%	51.40%	83.50%	100%	100%
aj-e11	41.19%	48.17%	55.25%	83.02%	100%	100%
hwb4	52.44%	53.23%	48.83%	81.23%	100%	100%
2-bit Mult.	40.54%	50.28%	41.30%	84.65%	100%	100%
Average	50.55%	58.75%	56.48%	83.27%	100%	100%

0.18 μm technology. And analysis of detection rate is done using Monte-Carlo simulation, where we inject a switching fault randomly in any intermediate nodes and check the output. Detection Rate (D.R.) is reported as faults that are observable and detectable by the used scheme at the output over total number of trials.

The results are compared with other CED methods in the literature utilizing both reversible and irreversible logics. For irreversible logic CED techniques, we have compared our results with DMR, Bose-Lin codes [3] and error detection based on logic implication [4]. Das and Touba in [3] used a SIS script [9] to satisfy synthesizing the condition for their proposed technique. Hence, we use SIS tool to synthesize all the irreversible CED techniques for a fair comparison.

For reversible CED approaches, we utilized Single Parity (SP) scheme and Fredkin Synthesized (FS) circuit approach discussed in [5] for comparison. For the SP scheme in [5], odd number of faults at the output are detectable. For our approach and FS circuits all the faults can be detected since their synthesis technique utilize solely parity preservative gates. Furthermore, the results for power cost (nW) and CMOS area (μm^2) are shown in Table I and the results of fault detection analysis is shown in Table II.

Based on the results in Table I and II our detection rate is perfect as FS approach discussed in [5] but with much less area cost and power consumption. This is due to the fact that the CMOS fault-free implementation of Fredkin gate is costly in comparison to our gate library. Also it worth to mention that in some cases DMR has more area overhead than our approach. This is because XOR/XNOR gate is not preferable by SIS tool since XOR/XNOR has worse performance in terms of delay, power and area. Hence, in some cases area of DMR might be more than our approach. The other reason can be the fact that our benchmark are small and if we've used benchmark with

bigger size, DMR would have yield better area in comparison to our approach but with much lower detection rate.

To sum up, in applications that doesn't require perfect detection rate and small area overhead, Alves' technique *et al.* [4] yields good result with roughly area cost which with just 10% of main circuit cost. One can achieve above 50% detection rate which can be really great for huge benchmarks. But if perfect CED rate is of an interest then our approach yields decent area cost and power cost.

VI. CONCLUSION

In this study, we have proposed a low cost universal library of EOP gates and a simple technique to covert any reversible MPMCT synthesized circuit to an EOP one. Also to demonstrate the practicality of our gates, we convert them to their CMOS realization, and then compare the results with other CED approaches in the literature such as DMR, Bose-Lin code and fully Fredkin synthesized circuits, etc. Clearly our approach shows a good potential in terms of fault-tolerance rate, area cost and power cost.

REFERENCES

- [1] M. J. Iaconi, "Optimal control of latent fault accumulation," in *[1989] The Nineteenth International Symposium on Fault-Tolerant Computing. Digest of Papers*, June 1989, pp. 382–388.
- [2] D. Das and N. A. Touba, "Weight-based codes and their application to concurrent error detection of multilevel circuits," in *Proceedings 17th IEEE VLSI Test Symposium (Cat. No. PR00146)*, April 1999, pp. 370–376.
- [3] —, "Synthesis of circuits with low-cost concurrent error detection based on bose-lin codes," in *Proceedings. 16th IEEE VLSI Test Symposium (Cat. No. 98TB100231)*, April 1998, pp. 309–315.
- [4] N. Alves, A. Buben, K. Nepal, J. Dworak, and R. I. Bahar, "A cost effective approach for online error detection using invariant relationships," *IEEE TCAD*, vol. 29, no. 5, pp. 788–801, May 2010.
- [5] M. Altun, S. Parvin, and M. H. Cilasun, "Exploiting reversible computing for latent-fault-free error detecting/correcting cmos circuits," *IEEE Access*, vol. 6, pp. 74 475–74 484, 2018.
- [6] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: An open source toolkit for the design of reversible circuits," in *Reversible Computation 2011*, ser. Lecture Notes in Computer Science, vol. 7165, 2012, pp. 64–76.
- [7] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Int'l Symp. on Multi-Valued Logic*, 2008, pp. 220–225, RevLib is available at <http://www.revlib.org>.
- [8] D. Maslov, G. W. Dueck, and N. Scott, "Reversible Logic Synthesis Benchmarks Page," 2005, <http://webhome.cs.uvic.ca/dmaslov>.
- [9] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Sis: A system for sequential circuit synthesis," EECs Department, University of California, Berkeley, Tech. Rep. UCB/ERL M92/41, 1992.