IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 30, 2019, accepted October 21, 2019, date of publication November 4, 2019,
date of current version November 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951279

Perfect Concurrent Fault Detection in CMOS Logic Circuits Using Parity Preservative Reversible Gates

SAJJAD PARVIN AND MUSTAFA ALTUN
Department of Electronics and Communication Engineering, Istanbul Technical University, 34469 Istanbul, Turkey
Corresponding author: Sajjad Parvin (sajadparvin.stu93@gmail.com)

This work was supported in part by the TUBITAK-1002 under Project 215E268, and in part by the Istanbul Technical University BAP under Project #41956.

ABSTRACT Reversible logic has 100% fault observability meaning that a fault in any circuit node propagates to the output stage. In other words, reversible circuits are latent-fault-free. Our motivation is to incorporate this unique feature of reversible logic to design CMOS circuits having perfect or 100% Concurrent Error Detection (CED) capability. For this purpose, we propose a new fault preservative reversible gate library called Even Target - Mixed Polarity Multiple Control Toffoli (ET-MPMCT). By using ET-MPMCT, we ensure that the evenness/oddness of applied 1's at input, is preserved at all levels of a circuit including output level unless there is a faulty node. A single fault always destroys the parity of input at the output. Our design strategy has two steps for a given function: 1) implement the function with our proposed reversible ET-MPMCT gate library; and 2) apply reversible-to-CMOS gate conversion. For the first step, we propose two approaches. For our first approach in step 1, we first need to have a reversible form of the given function if it is irreversible. Then, synthesize the reversible function using Mixed Polarity Multiple Control Toffoli (MPMCT) gate library by conventional reversible logic synthesis techniques. Finally, the synthesized circuit is converted to ET-MPMCT constructed circuit which is fault preservative. Our second approach, is an ESOP - Exclusive Sum of Products - based synthesis approach modified for our proposed fault preservative ET-MPMCT gate library. It does work with both irreversible and reversible functions. We synthesize our circuits with both approaches in step 1 and choose the circuit with lower number of reversible gates to be fed to step 2 of our design strategy. In second step of our design strategy, we convert our fault preservative reversible circuits into their CMOS counterparts. The performance of our designs is compared with other CED schemes in the literature in terms of area, power consumption, delay and detection rate. Simulations are done with Cadence Genus tool using TSMC 40nm technology. Clearly, results are in favor of our proposed techniques.

INDEX TERMS Reversible logic, fault preservation, latent faults, fault tolerant CMOS, concurrent error detection, QCA, quantum computing.

I. INTRODUCTION

Concurrent Error Detection (CED) techniques are a pre-requisite for reliability-critical applications where the data integrity is a must. CED techniques are incorporated in applications ranging from aerospace applications to online IoT maintenance. Hence, any fault at any intermittent node must not go undetected at the output, even if a faulty circuit produces the correct results at the output. However, utilizing conventional irreversible logic to achieve 100% concurrent fault detectibility is far fetched due to the existence of latent faults in circuits. Latent faults are defined as faults that might exist in a circuitry, but the circuit still produces the correct results at the output [1]. These faults can be hazardous for next operations, even though the correct results are obtained for the current operation. These faults are due to the fact that the conventional CMOS logic does possess "don't care" conditions. In other words, CMOS logic gates have very low fault-observability at their outputs and this leads to existence of latent faults in the network.

The associate editor coordinating the review of this manuscript and approving it for publication was Poki Chen.

To demonstrate the effect of latent fault in a conventional CMOS circuit consider a switching fault in a two-input AND gate causing an input node value to have a transition 0→1 or 1→0. Only 50% of the times the fault occurred at the input will propagate to the output. The detection rate decreases as the number of inputs for the AND gate increases; in the three-input AND gate only 25% of the times, a switching fault propagates to the output. This low detection rate is even worse for more complex circuits.

On contrary, reversible logic has 100% fault observability at the output. Hence no fault at an intermittent node is masked at the output. This is because reversible gates have bijective input-to-output relation; they have no ''don't care'' condition. Any switching fault in a circuit node propagates to the output. Therefore, reversible circuits do not have latent switching faults. Our goal is to utilize the 100% observability of faults at the output of reversible gates and implement circuits entirely out of parity preservative reversible gates to achieve 100% fault detection or 100% CED.

Several CED approaches have been proposed in literature. One conventional approach is the use of Double Modular Redundancy (DMR) [2]. However it can be shown, latent faults can disturb DMR scheme [3]. Consider a scenario where there exists a permanent latent fault in one of the replicas; this condition disrupts the detection mechanism of DMR. To get away with permanent latent fault issue in one of the replicas, one must use N-Modular Redundancy (NMR) or other similar techniques [2], [4], [5] which have significant area overhead. To compensate for huge area overhead of concurrent NMR scheme, several encoding schemes have been proposed in the literature such as weight-based codes [6], Berger codes [7], Bose-Lin codes [8], etc. Even though, they are effective codes for detecting faults, they detect those faults that are excited and observed at the output. This means that latent faults are ignored. Moreover some coding techniques require re-synthesis of the circuit [9] which might yield a descent fault-tolerance rate but area might be huge. To reduce area overhead of coding fault-tolerance schemes, online error detection using logic implication has been studied [10], [11]. In [10], even though the latent fault is ignored, the fault-tolerance rate is more or less the same as coding schemes with much lower cost. Since fault detection using logic implication scheme utilizes few number of AND gates to detect faults at intermittent nodes. It should be noted that, the aforementioned fault detection schemes do not fully solve the problem due to the existence of latent faults [12]. We show that our reversible circuit based solution with CMOS realization have superiority in both area and fault detection capability.

In reversible logic domain, two types of schemes are proposed to achieve a fault tolerant circuit: fault preservative gates and coding schemes. Using fault preservative gates to achieve fault detection capability, a circuit must be implemented solely out of these gates that are inherently fault preservative such as Khan gate and Islam gate [13]–[15]. These gates work perfectly in theory and they preserve the input parity at the output, but they are not area efficient in CMOS and do not guarantee perfect or 100% fault detection while realized in CMOS. Fredkin gate [16] is another parity preservative gate with 100% fault detection capability, but it is shown in [17] that this gate's CMOS implementation is not efficient at all. Moreover, the coding schemes in reversible realm is discussed in [17] and CED techniques using reversible logic are proposed such as single parity and hamming code. Although, these approaches increase the fault detection rate of a circuit to a good amount, perfect fault detection is not achievable [17].

In this work, by exploiting reversible logic, we propose a new cost effective Even-Odd Preservative (EOP) gate library, named ''ET-MPMCT'' - Even Target - Mixed Polarity Multiple Control Toffoli - that preserves evenness/oddness of applied input in terms of number of 1's in binary representation. We propose two approaches to achieve perfect CED based on our proposed ET-MPMCT gate library. In our first approach, for a given function, if it is irreversible, we convert it to a reversible form. Next, we synthesize the function using a synthesis algorithm with MPMCT gates. Next, we apply our proposed conversion approach to convert our synthesized reversible circuit with MPMCT gates to a fault preservative one using our ET-MPMCT gate library. By utilizing this approach by almost doubling the size of any given circuit we can achieve 100% concurrent error detection rate. Besides, in our second approach we propose an ESOP - Exclusive Sum of Products - synthesis technique based on our ET-MPMCT gate library. In comparison to our first approach, the latter one is more straightforward, since it does not require irreversible to reversible conversion, and it performs better for relatively large functions in terms of area cost. Finally, after obtaining synthesized reversible circuits, we compare the results of our approaches and choose a circuit with minimum number of reversible gates. And we replace each reversible gate with its CMOS counterpart to achieve a CMOS circuit satisfying 100% fault detection.

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 our proposed techniques plus describing our procedure to convert our fault preservative reversible circuits to their CMOS latent-fault-free counterpart. Section IV presents the simulation methodology and results of our proposed techniques and other CED techniques in the literature. Section V concludes the paper including our future aim and direction.

II. PRELEMINARIES

In this section, we provide necessary terms and definitions on reversible logic and reversible fault tolerance that we have used throughout this paper.

A. REVERSIBLE LOGIC

Consider n inputs. A gate is called **reversible** gate iff each of its 2^n input bit permutations is mapped bijectively to each of its 2^n output bit permutations. This requires exactly n outputs

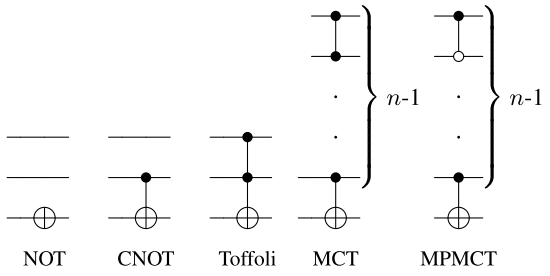

FIGURE 1. Circuit representations of all the gates in the MPMCT gate library.

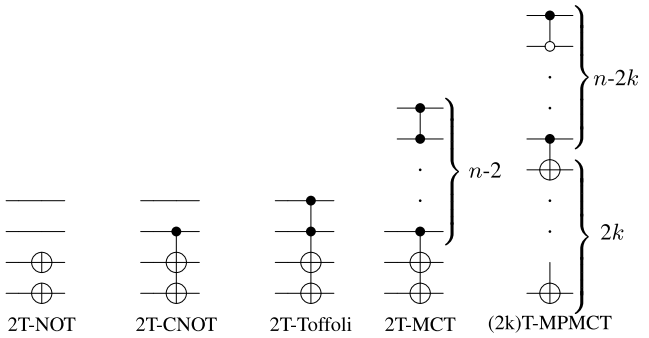

FIGURE 2. Circuit representations of all the gates in the ET-MPMCT gate library.

and it means input values can be deduced from the output, since each input permutation has exactly one counterpart in the output set. On the other hand, an irreversible gate might have different number of inputs and outputs where the mapping is injective or surjective.

A circuit is **reversible** iff it only consists of reversible gates. The generalized and conventional reversible gate library which we use in this study, is called **M**ixed **P**olarity **M**ultiple **C**ontrol **T**offoli (**MPMCT**) gate library. Definition of gates are as follow, with their circuit representations given in Figure 1. The symbols •, ∘ and ⊕ denote positive control, negative control, and Toffoli target line, respectively.

- **N**OT: a 1-bit gate performing NOT operation.
- **C**NOT: a 2-bit gate performing 1 bit NOT operation on its target bit iff its control bit is 1.
- **T**offoli: a 3-bit gate performing 1 bit NOT operation on its target bit iff its control bits are both 1.
- **M**ultiple **C**ontrol **T**offoli: an n-bit gate, $n = 1, 2, 3, 4, \ldots$, performing 1 bit NOT operation on its target bit iff all of its control bits are 1.
- **M**ixed **P**olarity **M**ultiple **C**ontrol **T**offoli: an n-bit gate, $n = 1, 2, 3, 4, \ldots$, 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.

B. REVERSIBLE FAULT TOLERANCE

The following lemmas from [17] demonstrate the reason behind reversible circuits being latent-fault-free.

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

Proof 1: The proof is by contradiction. Suppose that a transition in a node does not cause any change at the output. Since subcircuit of a reversible circuit is also reversible, the node can be considered as an input node of a reversible circuit. Also we know that a reversible circuit has one-to-one matching between its inputs and outputs, so a change in an input should change the output. As a result, there is a contradiction.

Since reversible logic is latent-fault-free, we would like to utilize parity preservative reversible gates to achieve perfect CED circuits. By definition, a reversible gate is **parity preservative** iff a reversible gate with inputs $I_1, I_2, \ldots, I_n$ and outputs $O_1, O_2, \ldots, O_n$, satisfies $I_1 \oplus I_2 \oplus \ldots \oplus I_n = O_1 \oplus O_2 \oplus \ldots \oplus O_n$, where $\oplus$ represents a bitwise XOR logic operation.

Lemma 2: Consider a reversible circuit consisting of only parity preservative gates. For this circuit, 100% *fault detection is possible if a switching fault occurs in an intermediate node of the circuit.*

Proof 2: A change at any intermediate node results in upsetting evenness or oddness of input value and since all the gates are parity preservative then the faulty signals will be carried out to the output with a parity different than the one applied at the input side. Hence by XORing the output bits, one can always detect occurrence of a fault. In other words, parity preservative reversible circuits always result in 100% fault detection upon occurrence of a switching fault at any node of the circuit.

We have proven that reversible circuits are latent-fault-free and by synthesizing a reversible circuit solely from parity preservative gates, perfect CED is achievable. Now we can go through our proposed approaches to achieve 100% CED CMOS circuits using parity preservative reversible gates.

III. PROPOSED PRESERVATIVE GATES AND SYNTHESIS TECHNIQUES

By considering area efficiency both in reversible and CMOS domains, we have proposed an EOP gate library, called ET-MPMCT which is constructed based on the MPMCT gate library. In our proposed library, with a set of control signals, we control even number of targets in a single stage, as shown in Figure 2. This extra even number of target bits guarantee evenness or oddness of 1's at the output, is the same as input. Using the proposed gate library, we achieve 100% or perfect CED.

The general design flow of our fault tolerance scheme, consisting of two steps, is presented in Figure 3. In step 1, to synthesize a given function, reversible or irreversible, we have two approaches: MPMCT and ESOP based synthesis techniques. At the end of step 1, we produce two 100% fault preservative reversible circuits consisting of ET-MPMCT gates corresponding to our two approaches. Next, the circuit having smaller number of reversible gates is preferred to be

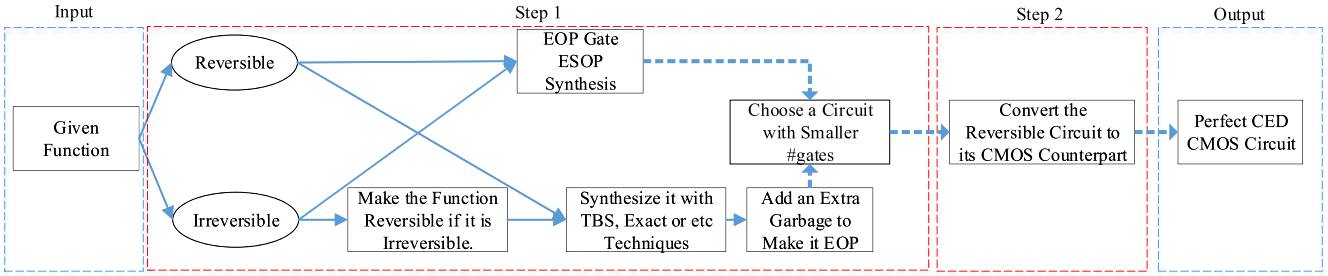

FIGURE 3. General design flow of our proposed fault detection scheme.

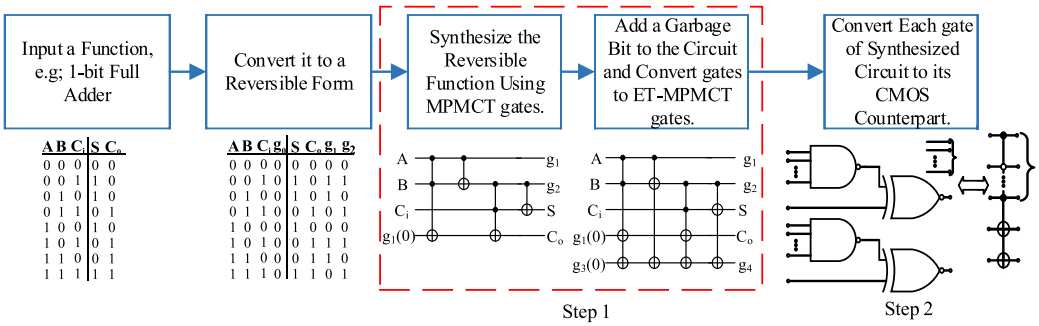

FIGURE 4. Approach 1 design flow.

used in the next step of our design. In Step 2, we convert each gate in our fault preservative synthesized reversible circuit to its CMOS counterpart. At the end, we have a CMOS circuit capable of 100% fault detection.

1) STEP 1 – APPROACH 1 MPMCT BASED SYNTHESIS

For any given function, firstly if it is irreversible, we need to convert it to a reversible function form. Garbage bits are added to the function to satisfy one-to-one matching constraint of reversible logic. By trying all the possible permutations of one-to-one matching of the input and the output, and then synhtesizing all the possible reversible functions, we choose the one with the best results in terms of number of used reversible gates. Note that required number of garbage bits to be added to an irreversible function to make it reversible is $2^{log\lceil M \rceil}$, where M is the number of repeated output pattern [18]. Hence, for an irreversible function to be converted to a reversible form, we have to check $2^{log\lceil M \rceil}$ different permutations to find a form of reversible function that after synthesis, results in a solution with least number of used reversible gates. This can be time consuming for functions where they require large number of garbage bits. Next, we synthesize the reversible function with TBS technique [19]. At this point we have synthesized a function using solely MPMCT gates which is not a fault preservative gate library. To make it fault preservative we add an extra garbage bit to our synthesized circuit. Then for each gate in our circuit, we substitute it with its counterpart gate in our ET-MPMCT gate library. In this approach we just need to use gates with 2 targets from our gate library where their first target bit is kept intact but the second target bit is mapped to this newly added garbage line. Eventually, by applying this procedure to all gates in our MPMCT synthesized circuit, EOP condition is achieved.

With this approach, we might face problems for relatively large functions regarding that the TBS technique does result in large circuits for large functions where the results are much larger than optimal or near optimal solutions. As a result, number of used reversible gates in TBS technique can grow exponentially as function size increases due to greedy nature of TBS algorithm [19]. It must be noted, described conversion technique can be applied to any MPMCT synthesized circuit regardless of their synthesis technique.

To elucidate Approach 1, we utilize a 1-bit full adder as an example.

Example 1: According to the design flow shown in Figure 4, first we define a function that we want to synthesize with perfect fault detection. In this example we use 1-bit full adder. Since 1-bit full adder's function is not reversible, a garbage bit is added to the input and two garbage bits are added to the output of its truth table to make it reversible. Next, we synthesize our reversible function using MPMCT gates shown in Figure 4. Next, we need to convert the MPMCT synthesized reversible circuit to an EOP circuit. By adding an extra garbage bit to the circuit with a value of 0 or 1 arbitrarily – 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. 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

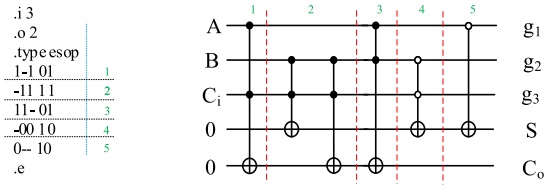

FIGURE 5. ESOP synthesis technique on 1-bit full adder.

target bit and control bits are left untouched and the second target bit of 2T-Toffoli is placed on the newly added garbage bit. And in the second stage, there is a CNOT gate which we can replace it with 2T-CNOT from our proposed gate library and keeping everything intact except mapping the second target on the new garbage bit. For the rest of the gates in the circuit we can follow the same procedure. The converted EOP circuit is shown in Step 1 of our design flow in Figure 4.

2) STEP 1 – APPROACH 2 ESOP BASED SYNTHESIS TECHNIQUE

Our proposed ESOP synthesis technique is a sort of hybrid of ESOP synthesis technique using MPMCT gate library [20] and our proposed conversion technique discussed in Section III-1. Unlike our proposed conversion technique where it only uses 2T-MPMCT gates, ESOP based synthesis technique incorporates as many as possible even number of targets in its synthesis procedure.

ESOP synthesis technique [20] procedure using MPMCT gate library is as follow. For a given n input and m output function, we need to convert it to its ESOP cube list form. And then we create a $n + m$ network and initialize m output bits to zero to synthesize a given ESOP cube list using MPMCT gates. Next, we go through all the rows in the cube list. For each input row, if the bit value is not "-", we use it as control signal for gates in MPMCT gate library. For control signals, if an input bit has a value of zero, we use negatively activated control signal otherwise we use positively activated control signal. Next, for each output bit value, we map the Toffoli target on a bit where it has the value of "1" in that row of ESOP cube list form. The synthesis procedure is shown for a 1-bit full adder in Figure 5.

Modified ESOP based synthesis for our ET-MPMCT gate library can be described as follows. For a given function, reversible or irreversible, we have n input bits and m output bits. We convert the given function into minimized ESOP cube-list form. Next, we create $n + m$ network to synthesize our function. However $n + m$ network is not enough for satisfying EOP condition. Hence we need to add one more output bit to have a $n + (m + 1)$ network. We initialize $m + 1$ output bits to zero. As a next step, we go through each row of our ESOP list and we select control bits as ESOP based synthesis technique described previously for MPMCT gate library. But target bit is chosen differently since, we are using ET-MPMCT gate library where they have even number of targets. To assign targets, we count the number of needed bits at each output row of our ESOP cube-list form. Two scenarios can happen while going through each output row of our ESOP cube list. First scenario is, for a given row, we need to assign even number of targets which means we have even number of ones in that output row. For this scenario, we just choose a proper gate from our proposed gate library with required target bits (number of target bits is equal to the number of ones in each output row). The second scenario is that for a given row, there is a need to assign odd number of targets. In this case, this stage of circuit would not be EOP because all our gates have even number of targets. So we need to map an extra target on the $(m + 1)^{th}$ output bit. We continue this procedure on all rows of ESOP list. Following example demonstrates our synthesis technique on a randomly generated function.

Example 2: As the design flow in Figure 6 shows, we need to convert the given function into its ESOP cube form. The function given in Figure 6 is an irreversible function and has $n = 3$ input bits and $m = 4$ output bits. Next, we need to create a network of size $n + m + 1$ and initialize the $m + 1$ output bits to zero. After forming the network, we go through each row of ESOP cube-list and try to synthesize it. For the first row of ESOP cube-list, last two bits of the input are 1, hence we use these two bits as control signals for our ET-MPMCT gate. And in the output of first row we have two 1's. Hence we need 2T-Toffoli gate. Target bits are mapped on the bit lines where on that row of output cube list we have 1's. For the second row, in the input we have two bits that are not "-", hence we use them as control signals. But for the second input row, the bit with a value of 0, requires a negatively activated control signal and the other one requires positively activated control signal due to having value of 1. And in the output row of the second row we have four 1's. So we need a 4T-MPMCT gate and target bits are mapped on the first m bits of output lines. Though for the last row of ESOP list, selection of control signal is done as described for previous stages, target bit selection is different. Since we have three 1's in binary representation of the output row. And gates in our EOP gate library have even number of targets, we need to select 4T-MPMCT gate, and map the first 3 targets on the first 3 output bits where they have value of 1 and map the 4^{th} target on the $m + 1^{th}$ bit line to satisfy the EOP condition for the third stage of the circuit.

It is worth to mention that, in some cases, negatively activated control signals are not preferred. As an example, for the second stage in our example, we can substitute the negatively activated control signal with a NOT gate and positive control signal as shown in Figure 6 Step 1. It should be noted that by adding this extra NOT gate, EOP condition of the second stage is ruined. To make up for disruption of EOP condition in the second stage, we add an extra NOT gate on $m + 1^{th}$ bit. Hence in the second stage to satisfy EOP condition we have a 2T-NOT gate plus a 4T-Toffoli gate. The same procedure can be done for the $3^{r}d$ stage of our circuit.

3) STEP 2 REVERSIBLE TO CMOS CONVERSION

We convert the synthesized reversible circuits using ET-MPMCT gate library to its CMOS realization. The goal is

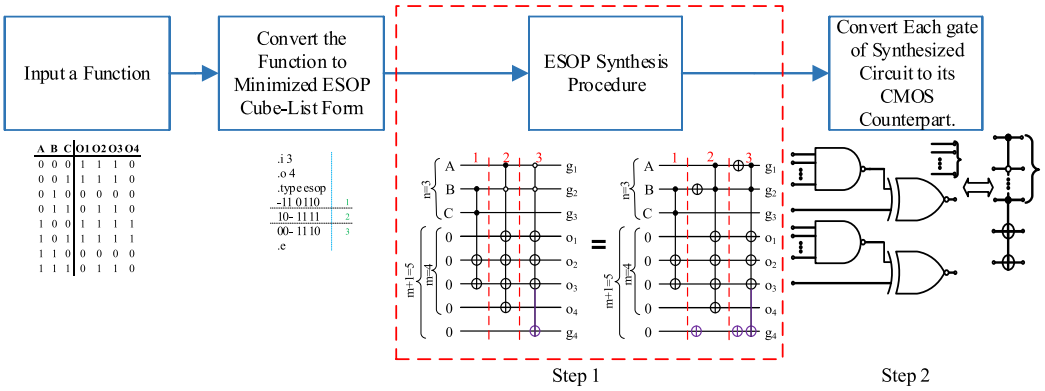

FIGURE 6. Approach 2 design flow.

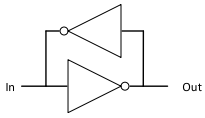

FIGURE 7. Cascaded Inverter ("repeater") design.

to achieve 100% concurrent error detectability using CMOS gates.

In conventional CMOS gates such as two-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, AND and OR gates as well. On the other hand, XOR, XNOR and Inverter gates distinguish the occurrence of fault at their inputs. 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 for negatively controlled ET-MPMCT gates, we utilize a cascaded Inverter gate (also known as "repeater") which is shown in Figure 7 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 and fault masking at the output.

CMOS implementation of our ET-MPMCT gate library is shown in Figure 8. It should be noted that utilizing one NAND/AND gate block to control even number of XNOR/XOR gates is hazardous because in case of occurrence of a fault at the output of the NAND gate, results in masking of the fault at the output side of our gate. Consequently, we need to utilize as many NAND gates as number of targets of

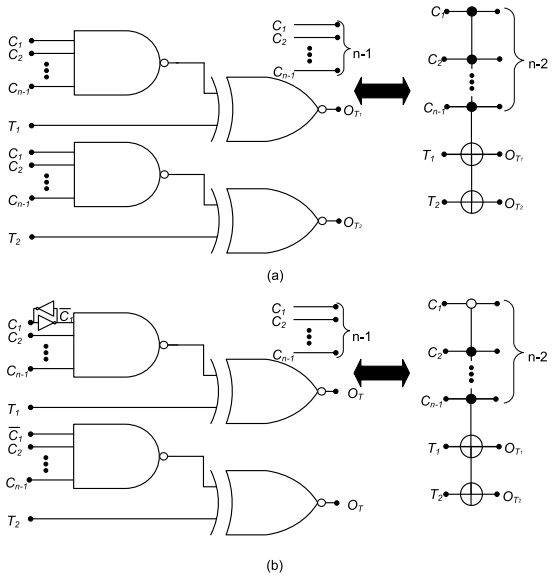

FIGURE 8. CMOS implementation of our proposed ET-MPMCT gate library (a) ET-MPMCT gate CMOS implementation with all positive control signal (b) ET-MPMCT gate CMOS implementation with both negative and positive control signal.

our ET-MPMCT gate to achieve internally fault preservative CMOS circuit. Besides, implementation of target bits of each ET-MPMCT gate library must be with XOR/XNOR gate. Since XNOR/XOR gates are aware of fault at their input. And if a fault occurs at intermediate node of CMOS realization (fault occurrence at the output of AND/NAND gate block) of our parity preservative reversible gates, then it does not get masked at the output.

IV. EXPERIMENTAL RESULTS

We have demonstrated our results both on irreversible benchmarks and reversbile benchmarks. Irreversible benchmarks are from [21] and reversible benchmarks are chosen from [22] and [23]. We compared the performance of our approach with other CED approaches in the literature in terms of area, power consumption, delay and fault-detection rate. Cadence Genus

TABLE 1. Comparison of CMOS area, power consumption, delay and Detection Rate (D.R.) among various CED schemes and our proposed scheme.

Benchmark	Irreversible DMR				Perfect Irreversible DMR				Proposed Approach			
	A	P	D	D.R.	A	P	D	D.R.	A	P	D	D.R.
clip[2,3]	412.25	5.45	1.03	4.07%	856.95	29.88	1.29	100%	736.65	13.33	4.04	100%
con1[2,3]	38.46	0.36	0.73	17.20%	89.00	0.47	0.90	100%	70.56	1.28	1.37	100%
inc[2,3]	267.42	2.84	0.99	3.87%	578.95	3.70	1.16	100%	323.81	4.86	2.32	100%
misex1[2,3]	157.17	1.90	0.95	4.68%	331.28	2.35	1.06	100%	177.11	8.31	1.52	100%
rd53[2,3]	95.96	1.40	1.38	9.53%	200.39	1.65	1.00	100%	79.00	1.42	1.32	100%
rd84[2,3]	539.43	7.47	1.10	12.78%	1095.80	44.55	1.88	100%	511.21	11.48	5.15	100%
sao2[2,3]	380.67	5.08	1.59	1.51%	847.43	6.54	1.45	100%	479.46	15.20	2.90	100%
squar5[2,3]	136.89	1.55	0.88	3.20%	264.95	1.85	1.04	100%	748.29	14.70	3.43	100%
sym9[2,3]	561.13	6.85	0.87	6.76%	1244.68	9.31	1.37	100%	620.00	12.75	6.10	100%
t481[2,3]	176.22	0.85	0.97	21.02%	363.03	5.50	1.32	100%	70.00	1.16	1.56	100%
xor5[2,3]	16.76	0.43	0.71	100.00%	16.76	0.43	0.71	100%	16.23	0.42	0.86	100%
5xp1[2,3]	274.13	3.03	0.96	3.38%	596.59	3.91	1.22	100%	264.00	3.30	1.64	100%
3_17[1,4]	34.22	0.40	0.80	69.51%	51.16	0.44	1.31	100%	21.25	0.39	1.05	100%
4_49[1,4]	87.14	1.07	0.86	45.10%	156.47	1.12	1.00	100%	79.03	1.52	2.31	100%
aj_e11[1,4]	70.56	0.77	0.80	49.20%	125.77	0.84	0.93	100%	52.57	0.98	1.84	100%
wim[1,4]	152.06	0.07	0.79	0.00%	252.78	0.26	0.93	100%	67.39	0.76	1.25	100%
sym6[1,4]	196.33	1.88	1.02	20.73%	361.44	8.41	1.18	100%	208.51	3.66	3.54	100%
alu-v2_30[1,4]	137.77	1.57	0.91	12.39%	280.12	2.11	1.07	100%	61.74	1.12	1.63	100%
hwb7[1,4]	1265.67	17.62	1.15	3.36%	2388.10	21.05	1.56	100%	1319.83	38.25	28.87	100%
4mod7[1,4]	146.59	1.93	0.92	7.27%	284.36	4.39	1.10	100%	44.81	1.00	1.43	100%
FA[1,3]	21.52	0.28	0.76	43.10%	35.63	0.32	0.76	100%	16.93	0.26	0.84	100%
Average	**246.11**	**2.99**	**0.96**	**20.89%**	**496.27**	**7.10**	**1.15**	**100%**	**284.21**	**6.48**	**3.57**	**100%**

[1] denotes, benchmark is synthesized using Approach 1 in Step 1.
[2] denotes, benchmark is synthesized using Approach 2 in Step 1.
[3] denotes, benchmark is irreversible.
[4] denotes, benchmark is reversible.

tool with TSMC 40nm technology is used for simulations. Analysis of detection rate were done using Monte-Carlo simulation, where we injected a switching fault randomly in an intermediate node of a circuit and the output was checked. Detection Rate (D.R.) is reported as faults that are observed at the output and also, it is detectable by the used scheme at the output over total number of trials, shown as follow:

$$D.R. = \frac{\text{Observable and Detectable Faults at Output}}{\text{Total Number of Trials}}.$$

We compared our results with DMR and perfect CED DMR. By using DMR, faults can be detected at the output if a fault occurs at any internal node and the effect of fault is observed at the output. However, perfect CED DMR is a bit different. All the intermediate nodes of each modules are XORed except input terminal of XNOR/XOR and Inverter gates. That is because XNOR/XOR and Inverter can distinguish occurrence of a fault at their output in all cases. Next output of XORed nodes will be ORed and in case of fault occurrence, output of OR gate will be set to logic 1. Perfect CED DMR is shown in Figure 9.

Conventional DMR and perfect DMR schemes are synthesized using ABC tool [24]. It should be noted that perfect DMR scheme prefers XOR and Inverter gates in comparison to other logic gates because these gates do not require XORing their input terminals. Hence, while synthesizing our benchmarks for DMR and perfect DMR, we forced the ABC tool to prefer XOR and Inverter gates while mapping. To trick ABC tool to prefer XOR and Inverter gates, we set the XOR and Inverter gates cost to zero and other gates cost were left untouched.

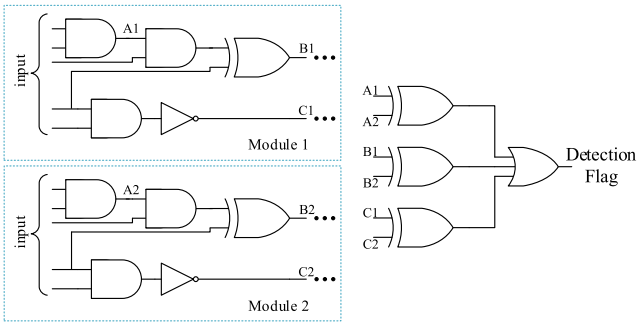

FIGURE 9. Perfect CED DMR scheme.

Besides, for our first approach used in Step 1 of our design flow, discussed in Section III-1, we synthesized our benchmarks using Revkit [25] and then applied our conversion technique to the synthesized circuits. For our second approach used in Step 1 of our design flow, discussed in Section III-2, to find minimized ESOP cube-list form of a given function we used Revkit [25] to perform exorcism and generate the ESOP form of the functions. Next we input ESOP cube-list to our python code to synthesize it using our proposed ET-MPMCT gate library.

Detection circuitry for our EOP reversible circuits is shown in Figure 10. Furthermore, the results for power consumption (**P**), area cost (**A**), delay (**D**) and detection rate (**D.R.**) are shown in Table 1. In Table 1 power consumption is reported in μW, area is reported in μm^2 and delay is reported in nS.

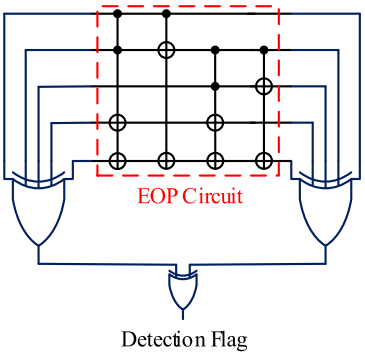

FIGURE 10. Detection unit for our EOP circuits.

From Table 1, on average our approach yields better results in terms of die area occupation and power consumption. Our proposed approach while always yields 100% D.R., on average requires 15% more area in comparison to conventional DMR which is comparable and has 42% less area in comparison to perfect DMR. Also, power consumption of our proposed approach is more by 116% on average in comparison to the conventional DMR technique. And it is lower by 9.5% on average in comparison with perfect irreversible DMR technique. However, delay of our proposed approach is on average 3.1 times more than the delay of perfect irreversible DMR technique. This is due to the cascade architecture of reversible logic where it is a must for a circuit to have neither feedback nor feedforward. On average our reversible based CED approaches require more time for computation than perfect irreversible DMR technique but it has lower power consumption and area occupation.

It should be noted, in some cases in Table 1, our approach yields better area results but worse power consumption results are achieved in comparison to the perfect irreversible DMR technique. And due to low area occupation, we would expect low power consumption but this is not the case for all of our benchmarks in Table 1. This can be justified as follow. Since, our approach is based on reversible logic and all the gates in the reversible logic must be cascaded to form a circuit. Besides, the CMOS realization of our reversible gates have XOR, XNOR and Inverters. While switching the input values, signals propagates through the circuit and they are toggled through the XOR, XNOR and Inverter gates. And upon the time signals reaches to the output, inputted signals are switched many times. And this causes to increase the switching activity and this explains why power consumption is more in some cases in our reversible CED technique.

V. CONCLUSION

In this study, we proposed a low cost universal library of EOP gates and a straightforward technique to convert any reversible MPMCT synthesized circuit to an EOP one and a synthesis technique using our proposed EOP gate library. Our synthesis technique resulted in descent performance for huge irreversible benchmarks and our conversion technique yielded descent results on optimally synthesized circuits using MPMCT gates. Next, we converted our synthesized circuits to their CMOS realizations, and then we compared the results with other CED approaches in the literature such as DMR and perfect DMR. Clearly our approaches showed a good potential in terms of fault-detection rate, area cost and power consumption. However, the delay of our proposed techniques are not good in complex functions, but for smaller benchmarks it yielded a comparable results with other CED techniques. This is due to the requirement of reversible circuits to have serial configuration. As a future work, we would like to investigate CMOS realization of reversible gates that yields better delay results and more optimized synthesis techniques using our proposed gate library for large and complex functions. Moreover, we would like to investigate our proposed gate library in **Q**uantum **C**ellular **A**utomata realm. Finally, our approach can be applied to applications where it requires perfect CED, low area and power efficient design.

REFERENCES

[1] M. J. Iacoponi, "Optimal control of latent fault accumulation," in *19th Int. Symp. Fault-Tolerant Comput. Dig. Papers (FTCS)*, Jun. 1989, pp. 382–388.
[2] E. Dubrova, *Fault-Tolerant Design*. New York, NY, USA: Springer-Verlag, 2013.
[3] M. Gabel, A. Schuster, R.-G. Bachrach, and N. Bjorner, "Latent fault detection in large scale services," in *Proc. 42nd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2012, pp. 1–12.
[4] T. Nakura, K. Nose, and M. Mizuno, "Fine-grain redundant logic using defect-prediction flip-flops," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, Feb. 2007, pp. 402–611.
[5] W. G. Brown, J. Tierney, and R. Wasserman, "Improvement of electronic-computer reliability through the use of redundancy," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 407–416, Sep. 1961.
[6] D. Das and N. A. Touba, "Weight-based codes and their application to concurrent error detection of multilevel circuits," in *Proc. 17th IEEE VLSI Test Symp.*, Apr. 1999, pp. 370–376.
[7] J.-C. Lo, S. Thanawastien, and T. R. N. Rao, "Concurrent error detection in arithmetic and logical operations using Berger codes," in *Proc. 9th Symp. Comput. Arithmetic*, Sep. 1989, pp. 233–240.
[8] D. Das and N. A. Touba, "Synthesis of circuits with low-cost concurrent error detection based on Bose-Lin codes," in *Proc. 16th IEEE VLSI Test Symp.*, Apr. 1998, pp. 309–315.
[9] N. A. Touba and E. J. McCluskey, "Logic synthesis of multilevel circuits with concurrent error detection," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 16, no. 7, pp. 783–789, Jul. 1997.
[10] N. Alves, A. Buben, K. Nepal, J. Dworak, and R. I. Bahar, "A cost effective approach for online error detection using invariant relationships," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 5, pp. 788–801, May 2010.
[11] R. Vemu, A. Jas, J. A. Abraham, S. Patil, and R. Galivanche, "A low-cost concurrent error detection technique for processor control logic," in *Proc. Conf. Design, Autom. Test Europe*, Mar. 2008, pp. 897–902.
[12] S. Parvin and M. Altun, "Implementation of CMOS logic circuits with perfect fault detection using preservative reversible gates," in *Proc. 25th IEEE Int. Symp. On-Line Test. Robust Syst. Design*, Jul. 2019, pp. 1–4.
[13] R. Khandelwal and S. Saini, "Parity preserving adder/subtractor using a novel reversible gate," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 885–888.
[14] 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 Circuits Syst. Int. Conf. Test. Diagnosis*, Apr. 2009, pp. 1–4.
[15] L. Jamal, M. M. Rahman, and H. M. H. Babu, "An optimal design of a fault tolerant reversible multiplier," in *Proc. IEEE 26th Int. SOC Conf. (SOCC)*, Apr. 2013, pp. 37–42.

[16] B. Parhami, "Fault-tolerant reversible circuits," in *Proc. 14th Asilomar Conf. Signals, Syst. Comput.*, Oct./Nov. 2006, pp. 1726–1729.
[17] 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. 74475–74484, 2018.
[18] D. Maslov and G. W. Dueck, "Reversible cascades with minimal garbage," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 23, no. 11, pp. 1497–1509, Nov. 2004.
[19] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proc. ACM/EDAC/IEEE Design Autom. Conf.*, Jun. 2003, pp. 318–323.
[20] K. Fazel, M. A. Thornton, and J. E. Rice, "ESOP-based Toffoli gate cascade generation," in *Proc. IEEE Pacific Rim Conf. Commun. Comput. Signal Process.*, Aug. 2007, pp. 206–209.
[21] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," MCNC, Tech. Rep. 1991-IWLS-UG-Saeyang, 1991.
[22] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Proc. Int. Symp. Multiple Valued Logic*, May 2008, pp. 220–225. [Online]. Available: http://www.revlib.org
[23] D. Maslov, G. W. Dueck, and N. Scott. (2005). *Reversible Logic Synthesis Benchmarks Page*. [Online]. Available: http://webhome.cs.uvic.ca/~dmaslov
[24] Berkeley Logic Synthesis and Verification Group. *ABC: A System for Sequential Synthesis and Verification, Release 110213*. [Online]. Available: http://www.eecs.berkeley.edu/~alanmi/abc/
[25] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: An open source toolkit for the design of reversible circuits," in *Reversible Computation*, A. De Vos and R. Wille, Eds. Berlin, Germany: Springer, 2012, pp. 64–76.

SAJJAD PARVIN received the B.Sc. degree in electronic engineering from Hormozgan University, Iran, in 2016. He is currently pursuing the M.Sc. degree in electronic engineering with Istanbul Technical University. He is also a Researcher with TUBITAK. His current research interests include electronic design automation (EDA), artificial intelligence, and efficient hardware implementation of machine learning algorithms.

MUSTAFA ALTUN received the B.Sc. and M.Sc. degrees in electronics engineering from Istanbul Technical University, in 2004 and 2007, respectively, and the Ph.D. degree in electrical engineering with a Ph.D. minor in mathematics from the University of Minnesota, in 2012. Since 2013, he has been a Professor with Istanbul Technical University and runs the Emerging Circuits and Computation (ECC) Group. He is the author of more than 50 peer-reviewed articles and a book chapter. Dr. Altun has been served as a Principal Investigator/Researcher for various research projects, including EU H2020 RISE, National Science Foundation of USA (NSF) and TUBITAK projects. He was a recipient of the TUBITAK Success, TUBITAK Career, and Werner von Siemens Excellence awards.