Quine-McCluskey Repair Technique for Evolutionary Design of Combinational Logic Circuits

Xinjie Huang, Ning Wu, and Xiaoqiang Zhang

***Abstract*—Evolvable hardware has been implemented in the design of combinational logic circuits. But it always takes too much time if circuits are generated with evolutionary algorithm directly. So the repair techniques have been more and more important in evolvable hardware. In this paper, Quine-McCluskey repair technique (QMRT), which generates the repair circuits with Quine-McCluskey (Q-M) algorithm, is proposed for evolutionary design of combinational logic circuits. Based on the QMRT, a novel evolutionary design algorithm, named EA-QMRT, is presented. The experimental results demonstrate that the EA-QMRT could realize combinational logic circuit with acceptable time cost and gates needed.**

***Keywords*—evolvable hardware, evolutionary algorithm, combinational logic circuits, repair technique**

I. Introduction

THE evolvable hardware (EHW) is a new technique which designs the structure of hardware adopting evolutionary algorithms (EAs) [1], and it has been applied in the design of combinational logic circuits [2]. Compared with traditional methods, evolutionary design method of combinational logic circuits needs not too much specialized experience, but could design creative circuit structures. Simultaneously, the EAs with the constraint of gates cost could optimize the resource consumption of the circuits [4].

However, the scalability problem [5], one of the main problems of the EHW, needs to be solved. As the circuit scale expands, the length of the chromosome in EA is increasing. It means that the search space of the EA is larger [6], causing the increase of the corresponding time cost significantly. When designing combinational logic circuits with EA, the Stalling effect [6] is common. As the Stalling effect appears, the best fitness of population is improved very slowly, and its tiny increase generally needs many evolutionary generations. So, currently, it's very difficult for EA to design a relatively large scale combinational logic circuit.

In order to overcome the scalability problem and improve the efficiency of EA, some methods have been proposed. And these methods can be classified into two categories: the decomposition techniques and the repair techniques.

Torresen introduced the divide-and-conquer in [7]. The divide-and-conquer approach means dividing the circuit into several smaller parts, each of which would be realized with EA, and then combining all parts into the expected circuit. Bidirectional Incremental Evolution (BIE) and Generalized Disjunction Decomposition (GDD) were proposed by Stomeo and Kalganova in [6], and Liang et al put forward a three-step decomposition method [8]. These hybrids of evolutionary algorithms and the decomposition techniques are effective for relatively large scale circuits, but still difficult to design large scale circuits efficiently.

The repair idea is proposed by Liang Houjun in [9] firstly, so is EA-Repair method. In EA-Repair method, an almost but not fully functional circuit is generated with EA firstly, and repairing the uncompleted circuit with Liang's repair technique is the next step. Compared to EA, the main advantage of the EA-Repair is that the time taken to design circuit is considerably shorter, whereas it is at the cost of a large number of gates consumption.

Thus, under the guidance of repair idea, Xin Zhang and his colleagues proposed the evolutionary repair technique in [10] trying to find a balance between the gates and the time consumed, by increasing the time cost and reducing the consumption of gates resource. However, its repair time takes large percent in the total time.

In this paper, an EA combined with Quine-McCluskey repair technique (QMRT), i.e. EA-QMRT, is proposed to improve the performance of EA for designing combinational logic circuits. And the kernel of the EA-QMRT is the QMRT, which realizes the repair circuit with the Quine-McCluskey (Q-M) algorithm. The QMRT can simplify the function of the repair circuit to save numerous redundant gates. Therefore the QMRT takes fewer gates compared to Liang's repair method and the evolutionary repair technique. Besides, the Q-M algorithm has a strong regularity. Even if facing multivariate functions, it can always find the simplest result. Meanwhile, since QMRT is targeted at the circuit directly, its repair time is shorter than that of the evolutionary repair technique.

Manuscript received December 30, 2014. This work was supported by the National Natural Science Foundation of China (61376025), Industry-academic Joint Technological Innovations Fund Project of Jiangsu (BY2013003-11).

Xinjie Huang is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 210016 China (e-mail: huangxj@nuaa.edu.cn).

Ning Wu is with the College of Electronic and Information Engineering, NUAA, Nanjing 210016 China (e-mail: wunee@nuaa.edu.cn).

Xiaoqiang Zhang is with the College of Electronic and Information Engineering, NUAA, Nanjing 210016 China (e-mail: zxq198111@qq.com).

ISBN: 978-988-19253-9-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The rest of this paper is organized as follows. Section II introduces the repair technique, including the repair principle and the QMRT. Section III combines EA with the QMRT. Some experimental results are given at section IV. Section V shows some discussions about this paper, and section VI summarizes the whole paper briefly.

II. The Repair Technique

A. *The Repair Principle*

In order to repair the uncompleted circuit into the correct one, the repair circuit has two necessary properties [11]:

--Before and after repair, the correct output is right still.

--After adding the repair circuit, the final output is opposite to the previous error output, e.g. an error output of the uncompleted circuit is 0, so the final output should be 1.

At the same time, the XOR has the following properties:

$$A \oplus 0 = A,\ A \oplus 1 = \overline{A}. \tag{1}$$

A could represent not only a logical variable, but also a logic function. So the XOR of a logic function and “0” is equal to the logic function; And, if a logic function is XORed with “1”, the final output is opposite to the logic function.

B. *The QMRT*

The Q-M Algorithm

The Q-M algorithm is an essential method to simplify the logic function of circuits. Compared with other methods, such as the formula and Karnaugh map, it could take advantage of the computer aided calculation with stronger regularity, which is particularly significant to simplify multivariate logic function.

The core idea of Q-M algorithm is to find out all prime implicants and essential prime implicants to get the simplest expression.

The QMRT

The target circuit is composed of the uncompleted circuit and the repair circuit as shown in Fig. 1. The uncompleted circuit is designed by EA, and the repair circuit by QMRT.

The uncompleted function could always be expressed by the sum of minterms. So the beginning of repair is to get the logical expression of repair circuit. The true table of a repair circuit could always be concluded, based on the repair principle as well as the input-output combinations of the uncompleted circuit and the expected circuit. An example is as followed.

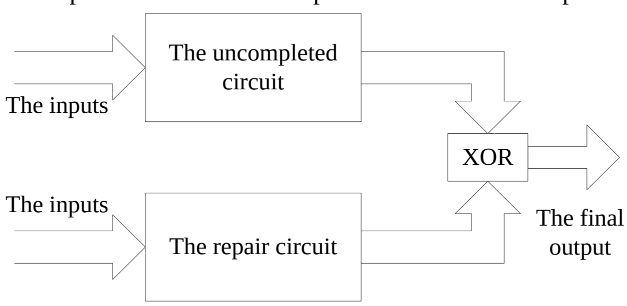

Fig.1. The target circuit.

TABLE I
THE TRUE TABLE OF THE EXPECTED CIRCUIT

A	*B*	*C*	*Out*
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

A, B, C are the circuit inputs; *Out* is the circuit output.

TABLE II
THE TRUE TABLE OF THE UNCOMPLETED CIRCUIT

A	*B*	*C*	*Oute*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Oute is the uncompleted circuit output.

TABLE III
THE TRUE TABLE OF THE REPAIR CIRCUIT

A	*B*	*C*	*Outr*
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Outr is the repair circuit output.

TABLE I gives the expected output of the target circuit. TABLE II lists the input-output combinations of the uncompleted circuit. Comparing the two tables, three input-output combinations of the uncompleted circuit are incorrect. When the input is “000”, “001”and “101”, the output is “0”, “1” and “0” respectively. But the expected output is “1”, “0” and “1”. So these have to be repaired.

As for the repair circuit, the output values corresponding to the input vector of “000”, “001” and “101” are “1”, and other outputs should be “0”. Based on the above analysis, TABLE III is filled with the expected outputs of the repair circuit. So the repair function could be expressed with the expression

$$Outr = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}C. \tag{2}$$

The key of QMRT is to obtain the simplest expression of the repair function with Q-M algorithm. $\overline{A}\overline{B}\overline{C}$ and $\overline{A}\overline{B}C$ are adjacent minterms, so are $\overline{A}\overline{B}C$ and $A\overline{B}C$. The prime implicant of the former group is $\overline{A}\overline{B}$ and the latter is $\overline{B}C$. So

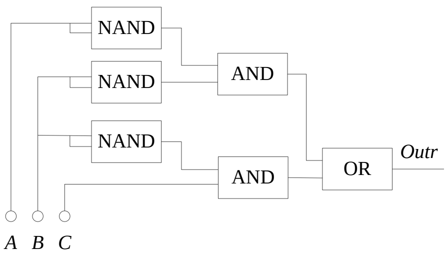

Fig.2. The repair circuit designed by QMRT.

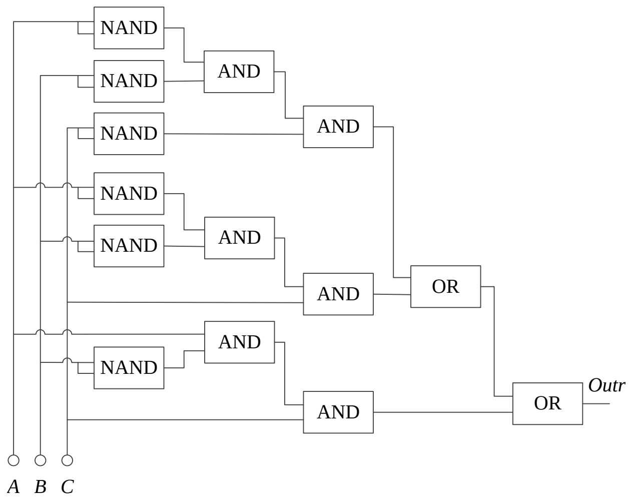

Fig.3. The repair circuit with Liang's method.

the simplest expression of the repair circuit is

$$Outr = \bar{A}\bar{B} + \bar{B}C. \quad (3)$$

So the final repair circuit could be shown in Fig. 2.

If Liang's repair method [9] is adopted to repair the same uncompleted function, the repair circuit should be described as in Fig. 3.

Comparing the two repair circuits, it's apparent that the QMRT consumes fewer gates.

III. The Evolutionary Algorithm Based on the QMRT

The circuit designed by EA based on the QMRT includes two parts: the evolved uncompleted circuit and the repair circuit, as shown in Fig 1.

Different from the EA-Repair [9] and the erEDA [10], the selected EA is the genetic algorithm (GA). The repair technique adopts QMRT. And the layout of evolved circuit is Cartesian Genetic Programming (CGP) model [12]. The steps in the design process of target circuit are shown in Fig. 4. The details of EA-QMRT are presented as the following.

1) Randomly initializing the population is the first step. And the parameters, including *P* (which stands for the population quantity), P_c (crossover probability), P_m (mutation probability), and *Bests* (the replication parameter) need to be set in this step. S_G is the count of generations in which the best fitness doesn't increase.
2) Evaluating the population according to the truth table of target circuit and evolving these individuals with EA need to be completed in this step. The EA including replication, crossover and mutation is carried out with the parameters of *Bests*, *Pc*, *Pm* and F_B^n. The replication is to replicate *Bests* excellent individuals from the current generation to the next generation. F_B^n is the best fitness in the current generation.
3) Comparing F_B^n with F_S (the fitness used to identify the Stalling effect), and if F_B^n is smaller, the algorithm will go to Step 4; if not, the algorithm will turn into Step 6.
4) F_B^{n-1} denotes the best fitness of the last generation. If F_B^n identically equals to F_B^{n-1}, S_G adds 1, otherwise S_G is 0.
5) S_G denotes the generation number used to identify the Stalling effect. If $S_G >= S\_G$, the algorithm will turn to Step 6. Otherwise, the algorithm will turn back to Step 2.
6) The QMRT starts to work in this step.
7) The final circuit including the evolved uncompleted circuit and the repair circuit comes out and then the algorithm is finished.

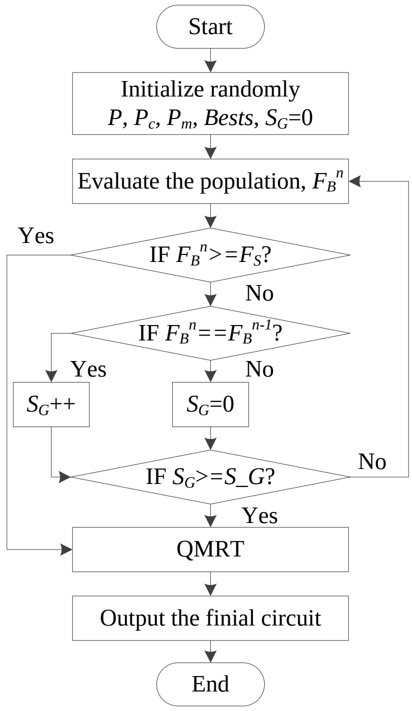

Fig.4. The flowchart of the EA-QMRT

It's necessary to emphasize the starting point of the QMRT. There are two things marking the beginning of the QMRT. One is that S_G reaches S_G; the other is that large enough best fitness appears. Either of them is regarded as the Stalling effect, which triggers the repair process.

IV. Experiments

In order to test the performance of the EA-QMRT, the designs of adders and multipliers have been completed.

Here is an example of 3-input and 2-output circuit to show the method of chromosome encoding in Fig. 5. The layout of the circuit is 1(row) ×10(column). And the gates used in the circuit are listed in TABLE IV. So the chromosome of the circuit is "3,2,1 4,2,3 0,1,2 2,5,0 3,2,0 3,5,4 1,6,6 4,0,7 2,6,3 2,1,10".

There are several cells in the chromosome. Each cell includes three parts. The first one is the code of the gate,

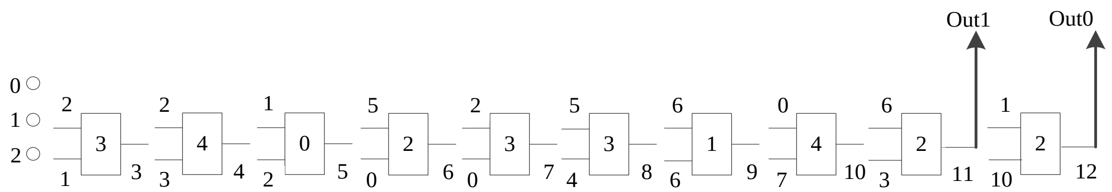

Fig.5. The CGP model of 3-input and 2-output circuit.

TABLE IV
THE INFORMATION OF THE GATE USED IN CIRCUITS

The code of gate	Type of gate	Logical function
0	NAND	!(a&b)
1	XOR	a ⊕ b
2	OR	a|b
3	AND	a&b
4	NOR	!(a|b)

judging the gate type. The second and third ones refer to the serial number of the input signals, which are the outputs of the front connected gates.

There is another aspect to stress. The output position of a gate with the largest fitness is the final output position of a chromosome.

The experiments are run on a notebook computer with a Dual Core (Intel core i3 2350) at 2.30 GHz and 2GB of RAM. The crossover rate is 0.8, the mutation rate 0.01, and the population quantity 200. The gates used in these experiments are NAND, XOR, OR, AND and NOR with 2 inputs.

In this set of experiments, the candidate adder circuit layout is 1×100, and the candidate multiplier circuit layout 1×200. Each experiment runs independently 20 times. If the best fitness doesn't increase in 300 generations, the Stalling effect would be thought to occur. And the best fitness can be considered large enough, when it reaches 95% of the maximal fitness value.

The results of these experiments are listed in TABLE V, where "Gates" means the average gates consumption, and "Time" stands for the product of the average time.

From TABLE V, it can be observed that the gate resource of the EA-QMRT is much less than that of the EA-Repair and the erEDA. Typically, as for "4×4"Adder, the gates consumption is 1648 when the Adder is evolved by the EA-Repair and 219.85 by the erEDA. However, the EA-QMRT only spends 139.7 gates on average. Overall, the EA-QMRT has a better performance on the gates consumption than the EA-Repair and erEDA.

The EA-QMRT consumes a small amount of gates, but its time cost doesn't increase significantly, even less than the other two improved methods about the "3×3" and "3×4" Adder. As the example with the "3×3" Adder, the time consumption of the erEDA is 11.938, whereas the EA-QMRT needs 6.455 only. Compared with EA, the improvement in time of the EA-QMRT is obvious.

Without the pertinent information about the repair time in [10], some experiments have been completed with Xin Zhang's repair strategy to find the relations linking repair time and total time. These experiments adopt GA. The candidate adder circuit layout is 1×100, and the candidate multiplier circuit layout is 1×200. *Stalling_Generation* [10] in all these experiments is 200. TABLE VI lists the values of *Max_Repair_Number* [10] for each circuit. Each experiment runs independently 10 times.

TABLE VII gives the circuit Repair Time and the Rate. "Repair Time" refers to the average time spent in the repair process. And "Rate" is the Repair Time to the total time ratio on average. Form TABLE VII, it's apparent that the repair time of EA-QMRT is much shorter than that of the erEDA.

TABLE VI
PARAMETERS SETTING OF ADDER AND MULTIPLIER

Ciruit Name	*Max_Repair_Number*
3×3 Adder	1
3×4 Adder	2
4×4 Adder	4
4×5 Adder	5
3×3 Mult	1
4×4 Mult	7

TABLE V
THE INFORMATION OF THE GATE USED IN CIRCUITS

Repair technique		EA[9]	EA-Repair[9]	erEDA[10]	EA-QMRT
3×3 Adder	Gates	61	239	84.85	22.95
	Time(s)	83	2.8	11.938	6.455
3×4 Adder	Gates	56	703	——	61.55
	Time(s)	210	26	——	12.141
4×4 Adder	Gates	68	1648	219.85	139.7
	Time(s)	753	39	58.014	66.112
4×5 Adder	Gates	74	3626	——	203.4
	Time(s)	3896	262	——	238.828
3×3 Mult	Gates	45.40	187	104.45	87.75
	Time(s)	425	17	38.208	44.152
4×4 Mult	Gates	——	1685	784.35	748.62
	Time(s)	——	237	428.915	493.86

ISBN: 978-988-19253-9-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE VII
THE REPAIR INFORMATION OF ADDERS AND MULTIPLIERS

Circuit Name		3×3 Adder	3×4 Adder	4×4 Adder	4×5 Adder	3×3 Mult	4×4 Mult
Repair Time(ms)	EA-QMRT	5.15	6.3	10.5	31.2	13.1	607.45
	erEDA	3571	18561	156080	834673	48794	1073462
Rate (%)	EA-QMRT	0.080	0.052	0.016	0.013	0.030	0.123
	erEDA	36.457	62.137	69.618	78.481	53.074	68.416

Meanwhile, the EA-QMRT has a better performance in the Rate than erEDA.

V. DISCUSSION

The scalability problem is an open question in the field of evolutionary design of combinational logic circuits.

The EA-Repair [9] repairs the evolved uncompleted circuit directly by the repair technique. It's effective to terminate the evolutionary process when the Stalling effect occurs for making up its impact. But it is required too many gates in the final solution, and the problem is unavoidable even if the decomposition methods are adopted.

Compared with EA-Repair, the erEDA [10] is more efficient with fewer gates. The evolutionary repair technique is the kernel of the erEDA. With its repair strategy, there would be several repair processes, which result in several Stalling effects. So the repair time takes a large percentage of the total time.

In this paper, EA-QMRT is proposed to improve the performance of EA in the design of combinational logic circuits. As the kernel of EA-QMRT, the QMRT can take advantage of Q-M algorithm to simplify and realize the function of repair circuit, which can save many redundant gates and much repair time.

Compared with EA, the EA-QMRT needs less time, whereas the gate resource consumption is only a little larger. Compared with the EA-Repair, the EA-QMRT can save many redundant gates, and the time spent doesn't increase significantly. Meanwhile, the EA-QMRT has a better performance in terms of the repair time to the erEDA.

VI. CONCLUSION

An efficient evolutionary design algorithm, i.e. the EA-QMRT, for combinational logic circuits is proposed in this paper， which is based on the QMRT. The primary idea of the EA-QMRT is to utilize the QMRT to generate the repair circuit. The experiment results demonstrate that the EA-QMRT proposed can accelerate the design process obviously and keep a good balance between the time cost and the gates needed. Overall, the EA-QMRT is more effective.

In addition, the EA-QMRT could be combined with the evolutionary repair technique [10] into an EA with a hybrid repair technique, and it's worthy studying on the combination of the EA-QMRT and the decomposition techniques [6]. In the future, more works should be done to improve the EA-QMRT for designing more effective repair techniques.

REFERENCES

[1] X. Yao and T. Higuchi, "Promises and Challenges of Evolvable Hardware," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and R*eviews, vol. 29, no.1, pp. 87-97, 1999.
[2] T. Higuchi, M. Iwata, D. Keymeulen, et al, "Real-world applications of analog and digital evolvable hardware," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 3, pp.220-235, 1999.
[3] T. Kalganova, "Bidirectional incremental evolution in extrinsic evolvable hardware," *Proc. the Second NASA/DoD Workshop on Evolvable Hardware (EH2000)*. Palo Alto, California, USA, 2000, pp. 65-74.
[4] G. Zbysek and S. Lukas, "An Efficient Selection Strategy for Digital Circuit Evolution," *Proc. Evolvable Systems: From Biology to Hardware*. York, UK, 2010, pp. 13-24.
[5] A. Stoica, R. Zebulum, D. Keymeulen, M.I. Ferguson, Xin Guo, "Scalability Issues in Evolutionary Synthesis of Electronic Circuits: Lessons Learned and Chal-lenges Ahead," *AAAI Spring Symposium Series*, Stanford University, CA, March 24-27, 2003.
[6] E. Stomeo, T. Kalganova, and C. Lambert, "Generalized Disjunction Decomposition for Evolvable Hardware," *IEEE Transactions on Systems, Man and Cybernetics*, PART B: Cybernetics, vol. 36, no. 5, pp. 1024-1043, 2006.
[7] J. Torresen, "A Divide-and-Conquer Approach to Evolvable Hardware," *Proc. The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98)*. Lausanne, Switzerland, 1998, pp. 57-65.
[8] H. Liang, W. Luo, Z. Li, and X. Wang, "A Three-Step Decomposition Method for the Evolutionary Design of Sequential Logic Circuits," *Genetic Programming and Evolvable Machines*, vol. 10, no. 3, pp. 231-262, 2009.
[9] H. Liang, W. Luo, Z. Li, and X. Wang, "Designing Combinational circuits with an Evolutionary Algorithm Based on the Repair Technique," *Proc. The 9th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2010)*. York, UK, 2010, pp. 193-201.
[10] X. Zhang and W. Luo, "Evolutionary Repair for Evolutionary Design of Combinational Logic Circuit," *Pro, the 2012 IEEE Congress on Evolutionary Computation*. Brisbane, Australia, 2012.
[11] H. Liang, "Research on Evolutionary Design Algorithms for Circuits," Ph.D. dissertation, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, 2009 (In Chinese).
[12] L. Sekanina, "Evolutionary Design of Gate-level Polymorphic Digital Circuits," *Proc. Applications of Evolutionary Computing: EvoWorkshops 2005*. Lausanne, Switzerland, 2005, pp. 185-194.