2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems

Logic Synthesis in Reversible PLA

Nazma Tara, Hafiz Md. Hasan Babu and Nawshi Matin
Dept. of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh
nazmatara2000@gmail.com, hafizbabu@hotmaill.com and nawshi.matin@yahoo.com

***Abstract*— Reversible logic have been motivated by consideration of zero-energy computation. Re-configurability and structural regularity of Programmable Logic Devices caused wide use of it by the logic designers. In this paper, we propose a design algorithm for a PLA (Programmable Logic Array) with a newly designed low cost 3 × 3 reversible NMG (New Mux Gate) circuit for implementing multi-output ESOP (Exclusive-OR Sum of Product) functions. In addition, we propose a heuristic to sort and to realize the product terms of ESOP functions in order to share the internal sub-products to reduce the number of gates in the proposed circuit. The proposed algorithms make the design efficient with improvement 9.05% in number of gates, 25.5% in garbage count and 14.5% quantum cost metric than existing techniques averagely. Performance is also analyzed by using MCNC benchmark circuits.**

***Keywords*— Reversible logic; RPLA; ESOP; Quantum cost**

I. INTRODUCTION

Irreversible hardware computation regardless of its realization technique results in energy dissipation due to information loss [1], proved by Landaurer in the early 1960. The amount of this energy dissipation is k×T×ln2 joules for each bit of information, where k is Boltzman's constant and T is absolute temperature. To avoid this energy consumption was a big challenge for the researchers. As the consequence, Benett in [2] showed that this huge energy dissipation can be resolved by adopting the digital circuit with reversible logic instead of irreversible logic. A reversible gate is a unit component of reversible circuit which gate has the equal number of inputs and outputs and one-to-one mapping between input vectors and output vectors. Reversible circuits are of particular interest in low power CMOS design [3], optical computing [4], quantum computing [5] and nanotechnology [6].

On the other hand, programmable logic devices (PLDs) are standard, off-the-shelf parts that offer customers a wide range of logic capacity, features, speed, and voltage characteristics - and these devices can be changed at any time to perform any number of functions. With programmable logic devices, designers use inexpensive software tools to quickly develop, simulate, and test their designs. Then, a design can be quickly programmed into a device, and immediately tested in a live circuit. Another advantage of PLD is, there are no NRE costs and the final design is completed much faster than that of a custom, fixed logic device. This Array Logic was introduced by Fleisher and Maissel [7] based on AND, OR and NOT synthesis to implement SOP or POS whereas Reversible Logic prefers Ex-OR operations as well as Exclusive Sun-Of-Product (ESOP) synthesis. ESOP synthesis gives out better result than SOP realization where many useful methods are proposed for minimizing multi-output Boolean functions into ESOP form [8], [9]. A regular structure of reversible wave cascade of ESOP synthesis is proposed in [10]. The generalized structure of Reversible PLA was first proposed in [11] based on ESOP realization of multi-output functions. Finally, this paper has proposed a new approach of designing Reversible Programmable Logic Arrays as well as compared with the proposed design with existing [11,12,13] others.

Three main contributions are addressed in this paper-

1. A 3×3 reversible gate with quantum cost 4 is invented.
2. Efficient algorithm is introduced, which reduced the complexity parameters of reversible circuit, such as, number of gates, garbage outputs and quantum cost of the proposed architecture of Reversible PLA.
3. The proposed circuit is compared with other designs [11, 12, 13] with the help of benchmark functions.

The organization of this paper is as follows- In the next Section, basic definitions and properties of Reversible Logic and Programmable Logic Array are given. In Section III, we describe the earlier approaches of Reversible PLA design and their limitations. In Section IV, we propose algorithm of new architecture. Finally, the construction of reversible PLA is illustrated. In Section V, we show the result of performance analysis of the proposed circuits. Last of all, the paper is concluded in Section VI.

978-1-4673-8700-2/16 $31.00 © 2016 IEEE
DOI 10.1109/VLSID.2016.40

IEEE computer society

II. Basic Definitions And Literature Review

In this section, basic definitions and ideas related to Reversible Logic and Programmable Logic Array are presented with illustrative figures and examples.

a) A **Reversible Gate** is an $k \times k$ data stripe block which uniquely maps between input vector $I_v = (I_0, I_1, \ldots, I_k)$ and output vector $O_v = (O_0, O_1, \ldots, O_k)$ denoted as $I_v \times O_v$. A $k \times k$ reversible gate is shown in Fig. 1.

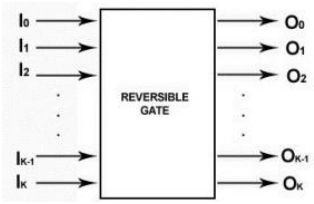
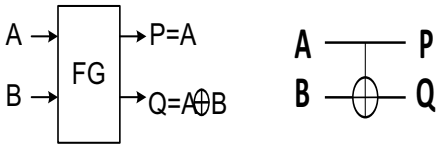
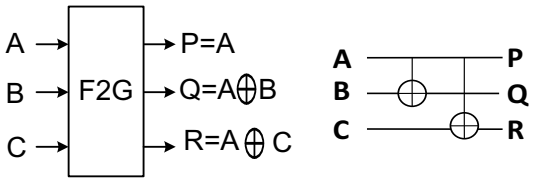


Fig. 1. A k × k Reversible Gate



(a) (b)

Fig. 2. (a). Feyman Gate and (b). Quantum realization of Feyman Gate

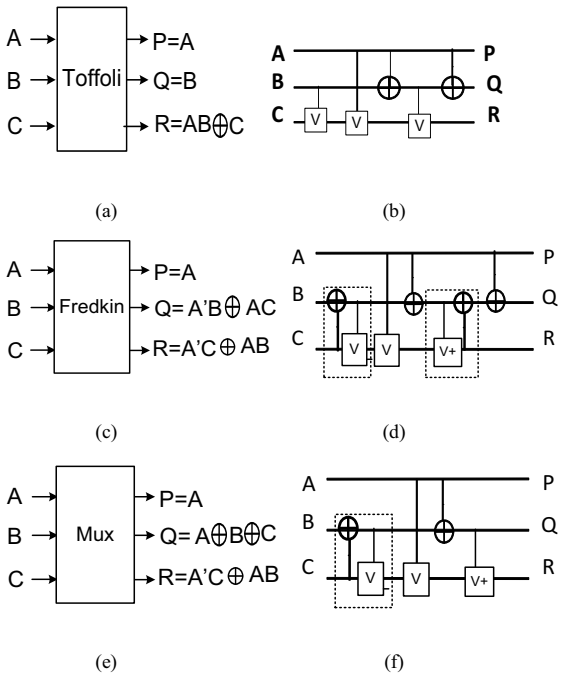


(a) (b)

Fig. 3. (a). Feyman Double Gate and (b). Quantum realization of Feyman Double Gate.

b) Every gate's output, that is not used as input to other gates or as a primary output is known as ***Garbage.*** For example, for an Ex-OR operation, P output of Feyman gate is a garbage output.

c) Every quantum circuit is built from 1 × 1 and 2 × 2 quantum primitives and its cost is calculated as a total sum of 2 × 2 gates used since 1 × 1 gate costs nothing i.e. zero. Basically the quantum primitives are matrix operation which is applied on qubits state. All the gates of the form 2 × 2 has equal ***Quantum cost*** and the cost is unity i.e. 1 Since every reversible gate is a combination of 1 × 1 or 2 × 2 quantum gate, therefore the Quantum Cost of a reversible circuit calculates the total number of 2 × 2 gates used.

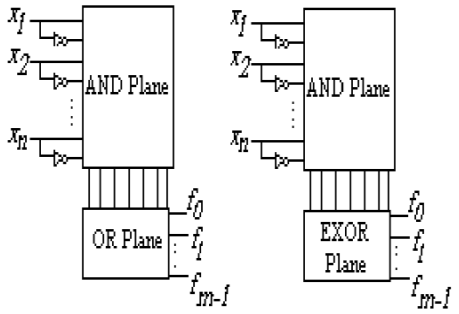


(a) (b) (c) (d) (e) (f)

Fig..4. Three 3 × 3 Reversible Gates with their quantum circuits ; (a) & (b) Toffoli gate, (c) &(d) Fredkin Gate and (e) & (f) MUX gate.

The quantum cost of Feynman gate in Fig. 2(a) is 1 and the quantum cost of Feynman Double gate in Fig. 3(a) is 2. Quantum circuit of Feynman and Feynman Double gate are shown in Fig. 2(b) and Fig. 3(b), respectively. Some other gates with their quantum costs are also shown in Fig. 4.

d) ***Programmable Array Logic Array (PLA)*** consists of two planes, the first one is programmable AND plane and the second one is programmable OR plane which is all together known as AND-OR PLA. When the second plane works as Ex-OR, then it is called AND-Ex-OR PLA.



(a) (b)

Fig. 5. Architecture of Programmable Logic Array (a). AND-OR array , (b). Ex-OR-OR Array

III. Design Analysis of Existing Techniques

Here we discuss about different types of existing Reversible PLAs [11,12,13] .

The design of Reversible PLA was first proposed in [11], which has used Feynman and Toffoli gates to realize Reversible PLA for multi-output ESOP operation where Toffoli gate is used for AND operation and Feynman gate is used for Ex-OR operation which used Conventional Architecture (Complement and non-complement lines for copying input variables). In [12] Fredkin gate is used in AND plane of reversible PLA for AND operation which costs 5 for each gate resulting higher quantum cost. Whereas, in [13], MUX gate has been used for AND operation. But all the designs suffer from extra overhead in terms of gate counting, garbage and quantum cost metrics.

IV. Logic Analysis In The Proposed Reversible PLA

A. Proposed Reversible gate:

A 3×3 reversible gate namely "New Mux Gate" gate or "NMG" is proposed which is drawn in Fig.6. The NMG gate can be defined as $I_v = (A, B, C)$ and $O_v = (P = A \oplus B)$, $Q = B'C \oplus AC'$, $R = BC \oplus AC'$), Where I_v and O_v are the input and output vectors respectively. The corresponding truth table of the NMG gate is shown in TABLE I. It can be verified from the truth table that the input pattern corresponding to a particular output pattern can be uniquely determined.

B. Design of Reversible Programmable Logic Array (RPLA)

In this section, we will describe our proposed design of Reversible Programmable Logic Array. In this structure, we will synthesize the multiple-output ESOP functions using different configurations of NMG gates.

TABLE I. TRUTH TABLE OF THE PROPOSED REVERSIBLE NMG GATE

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	0	1	0

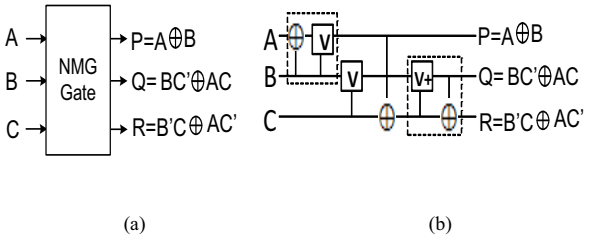


Fig. 6. (a) Block diagram of NMG gate(b) Quantum realization of NMG gate.

In multi-output ESOP, some products may be common among different output functions. We will also take the advantage of those terms like as irreversible PLA. The main constraint is that the fan-out of all signals in a reversible gate is one. Therefore, a single wire for a single product, having several cross points like those that irreversible PLA is not allowed in the proposed design. So, the appropriate copy in each product should be ensured to handle fan-out problem. In the synthesis method for multi-output ESOP using the RPLA, it is assumed that the multi-output circuit has been already minimized and is available in an ESOP format. For convenience we will denote different modes of operation of NMG, Feyman and Feyman Double gate as shown in Fig. 7.

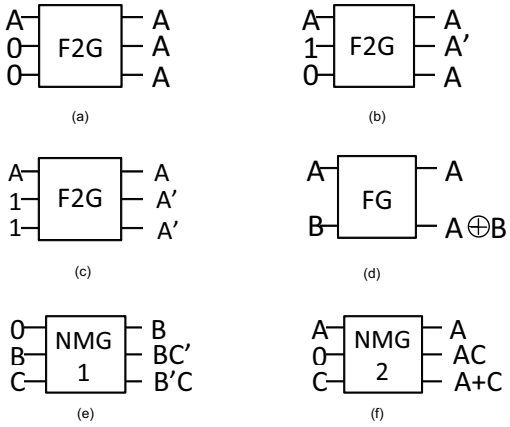


Fig. 7. Different modes of operation of (a), (b) & (c) Feyman double gate, (d). Feyman gate and (e) & (f). NMG gate

Now we will present the algorithms to realize the RPLA circuit of multi-output ESOP functions. When we describe the algorithms, we used two special terms. One is the Frequency of a product term, which is the number of output functions that share the product term. And the term DOT, that denotes the cross point in RPLA, in which no gate is used.

Algorithm I: Algorithm for the proposed AND plane of reversible PLA

Input : Several Benchmark Functions

Output : Number of gate used and number of garbage outputs produced to realize AND plane of RPLA circuit

1. **Begin**

2. Sort the products in ascending form based on their frequency of a benchmark function, then their index.

3. If Product X and Y has the same frequency and $index_y = index_x + 1$, realize product Y prior to the product X if all the conditions hold:

i. Both X and Y are the part of same output function O_p.

ii. X is the first product of O_p and Y is the second product of O_p and not shared by any other output function O_q, where $p<q<=o$, o denotes number of output functions.

4. If In_j is not the first input variable and used in Product X , then produce product X in later.

5. At each step of procedure of step 7, if any sub-product (first two variable only) of current product is already produced, then Update DOT. Otherwise repeat procedure of step 7.

6. Set DOT:=0, $\text{Gate}_{\text{count}} := 0, \text{Garbage}_{\text{count}} := 0$, $\text{Gate}_{\text{count}}$ = no. of total gate and $\text{Garbage}_{\text{count}}$ = no. of total garbase;

7. Put all literals into the stack

while for each ordered product (P_i) do

if In_i is the first variable of Product(P_i) then

if In_i or In_i' is used only once in total AND plane then

DOT + + ;

Update Stack;

else

Apply **F2G;**

$\text{Gate}_{\text{count}} + +$;

end if

else if In_i or In_i' is the second variable of Product (P_i) then

Apply **NMG** Gate;

$\text{Gate}_{\text{count}} + +$;

Update Stack;

else

Apply **NMG** Gate;

$\text{Gate}_{\text{count}} + +$;

$\text{Garbage}_{\text{count}}$ ++;

Update Stack;

end if

end while

6. end

Algorithm II: Algorithm for the proposed Ex-OR plane of reversible PLA

Input : Several Benchmark Functions

Output : No. of gate used and no. of garbage output produced to

realize Ex-OR plane of RPLA circuit.

1. **Begin**

2. For each output Functions F_i

if F_i has only one product, then generate F_i at last.

Loop

For each product P_j of F_i

Loop

if P_j is the first product in F_iI

then

if Freq(P_j) = 1 then

Place **DOT;**

else

Place **Feyman Gate**;

$\text{Gate}_{\text{count}} + +$;

end if;

else if F_i is the last function that shares P_j then

Place **Feyman Gate**;

$\text{Gate}_{\text{count}} + +$

$\text{Garbage}_{\text{count}} + +$;

else

Place **Feyman Gate;**

$\text{Gate}_{\text{count}} + +$;

end if;

end loop.

end loop.

3. end.

Example 1 : Consider the following ESOP functions:

$$\left.\begin{aligned} F_1 &= AB'C \oplus A'B \\ F_2 &= A'B'C \oplus AC \\ F_3 &= AB'C \oplus BC' \oplus A'B \\ F_4 &= AC \\ F_5 &= BC' \oplus A'B \oplus AC \end{aligned}\right\} \quad \ldots (1)$$

As a demonstration of the above Algorithm I and Algorithm- II, we constructed Reversible PLA circuit from Equation (1).

For the multi-output function given in Equation (1), the frequency of $A'B$ is 3 as it is shared among the output functions F_1, F_3, F_5; whereas the frequency of $A'B'C$ is one, as it is shared only by F_2.Table II shows the frequency of each product terms for Equation (1).

TABLE II. FREQUENCY TABLE FOR THE ESOP GIVEN IN EQUATION (1)

Products of the given ESOP function	A′B′C	A′B	AB′C	AC	BC′
Frequency of the corresponding product	1	3	2	3	2

According to the Algorithm I and Algorithm II the RPLA for the multi-output ESOP functions given in Equation (1) is presented in Fig. 8, which requires 14 gates and produces 6 garbage outputs.

The Feynman Double gates in AND plane are used to copy the input variables and their complement which removes the fan-out problems. Whereas, the NMG gates are used for AND operations. The generation of

complementary forms of input literals are unnecessary for the proposed AND plane because NMG and F2G are used together to generate all the products of two variables without having any dedicated lines of complemented forms of input variables. The realization of AND plane generates the order of Products illustrated in Algorithm I and EX-OR plane will be constructed according to this order by using Algorithm II. By using Algorithms I and II, the realization of the proposed reversible PLA is shown in Fig. 8.

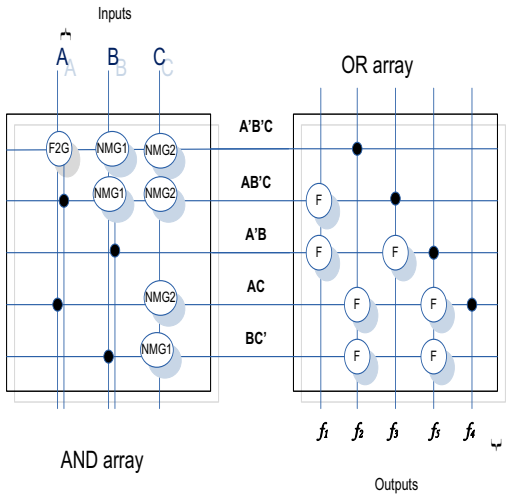


Fig. 8. Architecture of the proposed Reversible PLA for the ESOP functions given in Equation (1).

V. Simulation Result And Performance Analysis

Our algorithms for realization and minimization of RPLA have been written using language C and have tested extensively on windows microcomputer. Several experimental results are given below using an Intel core i5 Desktop CPU 2.3 GHz under Microsoft Windows 7 edition with 2 GB RAM. During the execution it was ensured that no other application is running. Fig.9. shows the simulation result of NMG by using Microwind DSCH2 [14]. Table III shows the experimental results comparing the proposed methods with the methods presented in [11, 12, 13] in terms of the number of garbage outputs and the number of gates for ESOPs given in Equation (1). Comparisons are also given in Table IV for corresponding benchmark functions.

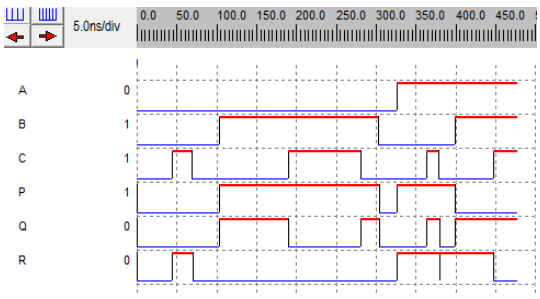


Fig. 9. Simulation result of NMG gate.

TABLE III. COMPARISON AMONG DIFFERENT DESIGNS FOR EQUATION (1).

	Gate	Garbage	Quantum cost
Existing[11]	19	11	47
Existing[12]	18	7	39
Existing[13]	17	19	55
Proposed design	14	6	31

VI. Conclution

In this paper, compact structure of Reversible Programmable Logic Array (RPLA) is presented. A reversible gate with low cost is proposed to generate the AND terms of RPLA which can produce two different AND terms at the same time. Using this property we can ensure the reduction of gate count. Two algorithms are proposed to minimize the previous architecture of RPLA that can realize any multi-output ESOP (Exclusive-OR Sum of Product) function. In addition, simulation of the proposed gate has shown that it works correctly. Finally, benchmark analysis proves the optimization of all the parameters. This design methodology improved 14.9%, 2.29% and 9.96% than [11], [12] and [13], respectively in terms of number of gates. It produced 39% and 12% less garbage outputs than [12] and [13], respectively. Whereas, quantum cost is reduced 17% with respect to [11] and 12% to [12]. The different number of input-output plays a major role in computing these comparisons. Working with more benchmark function is open for future research. In embedded circuits and other technologies for making low power consumption, reversible PLAs are useful [3, 7].

Acknowledgement

We would like to thank the ministry of ICT of Government of the People's Republic of Bangladesh for the innovation fund of researches.

References

[1] R. Landauer,"Irreversibility and heat generation in the computing pro-cess." IBM J. Res. Dev., vol.5,no. 3, pp. 183-191, 1961.

[2] C.H. Bennett,"Logical reversibility of computation". IBM J. Res. Dev.,vol.17, no. 6, pp. 525-532, 1973.

[3] G. Schrom and S. Selberherr. Ultra-low-power cmos technology. In Semiconductor Conf., Romania, 2002. E. Knill, R. Laamme, and G. J. Milburn. A scheme fore-cient quantum computation with linear optics. Nature, 409:46–52, January 2001.

[4] M. A. Nielsen and I. L. Chuang. Quantum computation and quantum information, 2001.

[5] R. C. Merkle. Two types of mechanical reversible logic. Nanotechnology, 4(2):114–131, January 1993.

[6] H. Fleisher and L. I. Maissel. An introduction to array logic. IBM J. of Research and Development, 19, 1975.

[7] T. Sasao. Exmin2: A simplification algorithm for exclusive-or-sum-of-products expressions for multiple-

valued input two-valued output functions. IEEE Transactions on Computer-Aided Desig n of Integrated Circuits and Systems, 12(5):621–632, 1993.

[8] A. Mishchenko and M. Perkowski. Logic synthesis of reversible wave cascades. In International Workshop on Logic Synthesis, pages 197–202, June 2002.

[9] M. Perkowski, A. B. P. Kerntof and et al. Regularity and symmetry as a base of efficient realization of reversible logic circuits. In International Workshop on Logic Synthesis, pages 245–252, June 2001.

[10] D. Maslov and G. Dueck. Reversible cascades with minimal garbage. IEEE Transactions on CAD, 23(11):1497–1509, November 2004.

[11] A. R. Chowdhury, R. Nazmul, and H. M. H. Babu. A new approach to synthesize multiple-output functions using reversible programmable logic arrays. In IEEE 19rd International Conf. on VLSI Design, pages 311–316, Hyderabad, India, 2006.

[12] R. Rahman, L. Jamal and H.M.H. Babu, ”Design of Reversible Fault Tolerant Programmable Logic Arrays with Vector Orientation”, Interna-tional Journal of Information and Communication Technology Research, pp. 337-342, 2011

[13] S.K. Mitra, L. Jamal, Mineo Kaneko and H.M.H. Babu, ”Design and Minimization of Reversible Programmable Logic Arrays”, Proc. of Symposium on VLSI,GLSVLSI’12, NY, USA, pp. 215-220, 2012.

[14] DSCH:, Microwind and dsch information page, Available at, http://www.microwind.org.

TABLE IV. COMPARATIVE RESULTS OF OUR POPOSED METHOS AND EXISTING METHODS [12,13,14] WITH MCNC BENCHMARK FUNCTION

Benchmark Function in ESOP form			**Existing Design[11]**			**Existing Design[12]**			**Existing Design[13]**			**Proposed Design**		
Function name	In /Out	No. of product	No. of gate	No. of garbage	Quantum cost	No. of gate	No. of garbage	Quantum cost	No. of gate	No. of garbage	Quantum cost	No. of gate	No. of garbage	Quantum cost
5xp1	7/10	31	170	80	508	140	157	468	166	112	418	136	102	402
9sym	9/1	52	439	330	1737	402	411	1456	427	385	1405	396	354	1377
adr3	6/4	16	69	50	189	58	61	176	67	48	157	58	40	155
b12	15/9	28	170	90	562	147	151	490	159	132	453	144	115	546
bw	5/28	22	350	46	499	296	293	686	305	64	446	296	55	448
rd53	5/3	14	55	23	162	48	50	148	56	42	134	46	33	126