Synthesis Algorithm for Reversible Logic

J. Hu
Major Laboratories of integrated circuits
College of Electronic Engineering
Heilongjiang University
Harbin, China

***Abstract*—In this paper, we propose a synthesis method based on Gröbner basis. We have tested the proposed algorithm on a set of the reversible benchmark circuits. Compared with existing synthesis method, this heuristic reduces area 9% on average.**

Keywords-logic synthesis; reversible logic; Gröbner basis

I. INTRODUCTION

Reversible circuits have low power property naturally. They map each input pattern to a unique output pattern. Landauer's principle states that logic computations that are not reversible, necessarily generate heat for information that is lost [1]. Therefore, using reversible logic circuits enables restoring consumed energy. Reversible circuits are of high interest in low-power CMOS design, optical computing, nanotechnology and quantum computing. Moreover, reversible computing is applied to other areas, including cryptography, digital signal processing, communication, and computer graphics, requiring that all the information encoded in the inputs be preserved in the outputs. Z. Zilic and A. D. Vos illustrates reversible logic implementation into classical MOS electronics in [2]. They have built several reversible circuits, powered only by their input signals.

In reversible logic, feedbacks and fan-outs are not permitted. It makes reversible circuits synthesis differ substantially from conventional logic synthesis. Synthesis approaches are developed for reversible circuits with small numbers of inputs and outputs. In [3], researchers considered the use of vast templates to simplify a reversible circuit initially found by other means. These heuristics are simply because of no using complicated computation. However, they do not scale well and require extensive use of template matching. They may suffer from exponential explosion with increasing number of inputs. P. Kerntopf, in [4], proposed a heuristic algorithm to reversible logic synthesis using a new complexity measure based on shared binary decision diagrams with complemented edges. But, it is not avoidable for extensive searching and feasible for optimizing reversible circuits with a large number of inputs. D. Maslov, in [5], showed that $|Q| = n \cdot 3^{n-1}$, if Q is the set of all possible gates with n inputs. Given the model for function implementation, the problem of synthesis is to construct a function in terms of a sequence of gates from the set Q. For each step the algorithm investigates $n \cdot 3^{n-1}$ possible gates. This is only feasible for small values of n. To alleviate these problems, we use an XOR sum of products expression of the output function to synthesize the circuit. Use of such a Reed-Muller expansion of the function was also suggested in [6]. The method, however, fails to follow three points. To begin with, applications of Reed-Muller circuits have so far not become popular due to one obstacle: slow speed [7]. The special structure of reversible circuit is cascade. The propagation delay of the cascade tends to be large. However, all above-mentioned approaches have only one constraint, area. They have neglected the impact of path delay. Moreover, as it is applied to functions with a large number of inputs, synthesis of Reed-Muller functions becomes difficult and synthesis is hardly a process. Finally, it claims that it does not require output permutation or extra garbage lines (such lines are required to equalize the number of inputs and outputs). There are too much limitations.

This paper presents several key contributions to synthesize reversible circuits. The organization of the paper is as follows. Mathematical model is built in section 2. The detailed descriptions of the proposed algorithm to synthesize reversible circuits are given in section 3. Experiment results when run on a set of benchmark circuits are shown in Section 4. The paper concludes with section 5.

II. MATHEMATICAL MODEL

In reference [8], we built the Boolean polynomial modeling for reversible circuits. Using an XOR sum of products as each output expression of the reversible circuit, we derived the positive-polarity Reed-Muller (PPRM) expansion from the function's sum of products expansion,

$$F(x_1, x_2, \cdots, x_n) = a_0 \oplus a_1 x_n \oplus a_2 x_{n-1} \oplus a_3 x_{n-1} x_n \oplus \cdots \oplus a_{2^n-1} x_1 x_2 \cdots x_n ,$$

where the $a_i \in \{0,1\}$, $1 \le i \le 2^n\text{-}1$. The $x_1, x_2, \cdots, x_n$ are all uncomplemented (positive polarity) and the $\oplus$ denotes the Boolean XOR(exclusive-or) operation. The PPRM of a function is unique and regular.

The XOR and AND Boolean operations of the PPRM form are mapped into the additive and multiplicative ones of Boolean polynomials that are similar to ones of real-valued polynomials, only a constraint is required, i.e. Boolean variable $x \bullet x = x$. Boolean operations have algebra properties. Let F be a set of PPRM expansions and "$\oplus$" and "$\bullet$" be the XOR and AND Boolean operations. They satisfy all the following properties:

(i) Commutativity. For every $f, g \in F$; $f \oplus g = g \oplus f$; $f \bullet g = g \bullet f$.
(ii) Associativity. For every $f, g, h \in F$;
$f \bullet g \bullet h = f \bullet (g \bullet h) = (f \bullet g) \bullet h$;
$f \oplus g \oplus h = f \oplus (g \oplus h) = (f \oplus g) \oplus h$.
(iii) Distributivity. For every $f, g, h \in F$;
$f \bullet (g \oplus h) = f \bullet g \oplus f \bullet h$; $(g \oplus h) \bullet f = g \bullet f \oplus h \bullet f$.
(iv) Idempotence. For every $f \in F$; $f \bullet f = f$.
(v) Zero element. There is a zero element 0 such that $f \bullet 0 = 0$.
(vi) Identity element. There is an identity element 0 such that $f \oplus 0 = f$.
(vii) Inverse. For every $f \in F$; $f \oplus f = 0$ $f \oplus 1 = \overline{f}$.

III. ALGORITHM

On the base of Dongmei Li's GVW algorithm [9, 10, 11] we add the Boolean algebra properties to it in order to computer Gröbner basis for Boolean polynomials ideal. According to the generators in reduced Gröbner basis, we build decomposition tree and find simpler structure for reversible circuits.

Variables:

U a list of terms $T_i = lm(u_i)$, representing signatures for $(u_i, v_i) \in M$;

V a list of polynomials for $v \in I$;

H a list for $lm(u)$ were $u \in R^m$ is a syzygy found so far,

JP a list of pairs $ax^{\alpha}(T_i, v_i)$, where ax^{α} is a term so that $ax^{\alpha}(u_i, v_i)$ is the J-pair of (u_i, v_i) and (u_j, v_j) for some $i \neq j$.

JCP a list of pairs $dx^{\alpha}(T_i, v_i)$, the JC-pair of (u_i, v_i) and (u_j, v_j); $i \neq j$

P is the polynomial expression of reversible circuit synthesized

Input: $g_1, g_2, \cdots, g_m \in K[x_1, x_2, \cdots, x_n]$, a term order for R, and a term order on R^m

Output: solution-tree ST

Step 0. $U = [E_1, E_2, \cdots, E_m]$, and $V = [g_1, g_2, \cdots, g_m]$, $B = \varnothing$.

Compute all the principle syzygies $g_j E_i - g_i E_j$ for $1 \le i < j \le m$,

and store the leading terms of these syzygies in H;

Compute all J-pairs of $(E_1, g_1), (E_2, g_2), \ldots, (E_m, g_m)$ and store them in JP, storing only one J-pair for each distinct signature, the one with v-part minimal.

Compute all JC-pairs of $(E_1, g_1), (E_2, g_2), \ldots, (E_m, g_m)$ and store them in JCP.

Step 1 while $JP = \varnothing$ or $JCP \neq \varnothing$;

begin

Step 1a while $JCP \neq \varnothing$;

begin

Take any pair (T, v) from JCP (say with minimal signature)

Delete it from JCP.

The pair (T, v) is top-reducible by some pairs in $[U, V]$

Check the pair (T, v) by some pairs in $[U, V]$

if the pair (T, v) be top-reducible then

discard (T, v) and go to step 1a.

else if $v \neq 0$ then

begin

Append T to U and v to V,

$B \leftarrow B \cup \{(T, v)\}$

end

Step 1b. while $JP \neq \varnothing$

begin

Step 1ba. Take any pair (T, v_1) from JP (say with minimal signature)

Delete it from JP.

if (T, v_1) satisfies the minimality criterion with respect to

G = $[U, V]$ then

Discard (T, v_1) and go to step 1ba.

Reduce the pair (T, v_1) repeatedly and as much as possible by

pairs in $[U, V]$ using only regular top-reductions , say to get (T, v).

if $v = 0$ then

Append T to H, and delete every J-pair (T_2, v_2) in JP whose signature T2 is divisible by T.

if $v \neq 0$ then

begin

Append T to U and v to V,

$B \leftarrow B \cup \{(T, v)\}$

end

Step 1c. Add the leading terms of principle syzygies,

$v_i T_j - v_j T_i$, $1 \le i < |B|$ $1 \le i \le |U|$ to H,

where $(T_i, v_i) \in B$ and $(T_j, v_j) \in [U, V]$.

form new JC-pairs of (T_i, v_i) in B and (T_j, v_j),

$JCP \leftarrow$ all such JC-pairs whose signature are not reducible by H.

Form new *J*-pairs of (T_i, v_i) in *B* and (T_j, v_j),

$JP \leftarrow$ all such *J*-pairs whose signature are not reducible by *H*,

Storing only one J-pair for each distinct signature, the one with *v*-part minimal.

$B \leftarrow \varnothing$

end

Return *V* and *H*.

Step 2. Using function GuidedDecomposition(*V*,*P*) return the solution-tree, and

if the length of a solution-tree W > the length of a solution-tree Y during building all solution-tree then

Delete W

Step 3. Storing only the shortest solution-tree *ST*

IV. EXPERIMENTAL RESULTS

The experimental results presented here were run on a PC (Intel Core i7 3770 3.9G Hz, 4GB RAM). We implemented the described algorithm in C++ programming language and compared the experimental results of the presented algorithm and other approach.

TABLE I. SIMPLIFICATION RESULTS OF THE REVERSIBLE BENCHMARK CIRCUITS

	Our	GT
circuits	**#gates**	**#gates**
4mod5	5	8
6sym	13	13
rd53#1	30	30
rd53#2	12	13
rd53#3	12	13
rd53#4	20	23
rd53#5	16	17
ham7	24	23
ham15	109	132
hwb8	739	634
C3540	928	1007
C5315	1256	1401
C6288	2129	2216

Table 1 gives results for a number of circuits taken from D. Maslov's benchmark web site or provided directly by references with various algorithms. The "Our" columns indicates our optimization cost. The GT columns are the synthesis results of the generalized Toffoli gates family [5]. The #gates denotes the number of reversible gates in reversible circuit.

The first two reversible circuits in Table 1 are single output functions, 4mod5 (Divisibility checkers) and 6sym. The next five ones are circuit rd53 that is the 5-input 3-output symmetric RD-input weight function. Circuits rd73 and rd84 are 7-inputs 3-outputs and 8-inputs 4-outputs. Circuits ham7 and ham15 are the size 7 and 15 Hamming optimal coding function. Circuit hwb8 is the size 8 hidden weight bit function. Its output equals its input shifted on the number of positions equal to the number of ones in the input pattern. Hidden weight bit function is known to have an exponential size BDD for any variable ordering. Circuit C3540 is ALU and control. It has 50 primary inputs, 23 primary outputs. Circuit C5315 is ALU and selector. It has primary 178 inputs, 123 primary outputs. Circuit C6288 is 16-bit multiplier, with 16-bit input word sizes and with each full adder realized. Experimental results on a set of benchmarks show that our algorithm is indeed effective in solving synthesis problem for reversible circuits. Using our heuristic, the average number of reversible gates and garbage lines are decreased. We can achieve area improvement of 9%.

V. CONCLUSIONS

Symbolic computer algebra has been effectively applied to achieve optimized designs for combinational arithmetic designs, classical scheduling and resource sharing. A synthesis method of reversible logic is introduced. Basing on Gröbner basis, we presented a symbolic algebra method to reduce Boolean polynomial so as to get better reversible circuits structure. The presented method has been tested on many examples and proved very promising. The method is more effective to complement synthesis of reversible functions with a large number of inputs than existing methods.

ACKNOWLEDGMENTS

This research was supported in part by Key Fund Project of Heilongjiang Provincial Department of Education (No. 12541604).

REFERENCES

[1] R. Landauer. Irreversibility and heat generation in the computing process. IBM J. Res., 5, pp.183- 191, 1961.

[2] Zilic Z, Radecka K, Kazamiphur A. Reversible Circuit Technology Mapping from Non-Reversible Specifications. Proc. of the conference on Design, Automation and Test in Europe, pp.558-563,2007.

[3] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In Proc. Design Automation Conf., Anaheim, CA, pp.318-323, June 2003.

[4] P. Kerntopf. A New Heuristic Algorithm for Reversible Logic Synthesis. In Proc. Design Automation Conf., pp. 7–11, 2004.

[5] G. W. Dueck and D. Maslov. Reversible function synthesis with minimum garbage outputs. In 6th International Symposium on Representations and Methodology of Future Computing Technologies, pp. 154-161, 2003.

[6] A. Agrawal and N. K. Jha. Synthesis of Reversible Logic. Proc. of the Design, Automation and Test in Europe Conference and Exhibition (DATE'04), pp.710-722, 2004.

[7] Xia Y.,Ali B. and Almaina AEA. Area and power optimization of FPRM function based circuits. IEEE Proc. of IEEE ISCAS'2003, Bangkok, Thailand, pp.329-332, 2003.

[8] Hu Jing, Ma Guang-sheng Li Dong-hai, and Feng Gang. Considering Crosstalk Symbolic Synthesis Method for Reversible Circuits. Chinese Journal of Electronics. 36(5),pp.1029-1034,2008.

[9] Adams W, Loustaunau P. An Introduction to Gröbner Basis. Graduate studies in Mathematics 3. Amer Math Soc, Providence, 1994.

[10] Dongmei Li, Jingwang Liu, and Weijun Liu. W-Gröbner Basis and Monomial Ideals under Polynomial Composition. Appl. Math. J. Chinese Univ. Ser. B 26(3), pp.287-294,2011.

[11] Dongmei Li, Jinwang Liu, and Weijun Liu. GVW Algorithm over Principal Ideal Domains. Journal of Systems Science and Complexity, 26(4), pp.619-633,2013.