3-2003

Multi-Output ESOP Synthesis with Cascades of New Reversible Gate Family

Mozammel H.A. Khan
East West University, Bangladesh

Marek Perkowski
Korea Advanced Institute of Science and Technology

Multi-Output ESOP Synthesis with Cascades of New Reversible Gate Family

Mozammel H. A. Khan and Marek A. Perkowski*
Computer Science and Engineering Department, East West University, Mohakhali C/A, Dhaka 1212, BANGLADESH. mhakhan@ewubd.edu, mhakhan@accesstel.net
*Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), 373-1 Gusong-dong, Yusong-gu, Daejeon 305-701, KOREA. mperkows@ee.kaist.ac.kr

Abstract: A reversible gate maps each output vector into a unique input vector and vice versa. The importance of reversible logic lies in the technological necessity that most "near-future" and all long-term future technologies will have to use reversible gates in order to reduce power. In this paper, a new generalized *k*k* reversible gate family is proposed. A synthesis method for multi-output (factorized) ESOP using cascades of the new gate family is presented. For utilizing the benefit of product sharing among the ESOPs, two graph-based data structures – connectivity tree and implementation graph are used. Experimental results with some MCNC benchmark functions show that the number of gates in the multi-output ESOP cascades is almost equal to the number of products in the multi-output ESOP. However, this cascaded realization of multi-output ESOP generates a large number of garbage outputs and requires a large number of input constants, which need to be reduced in the future research. This synthesis method is technology-independent and can be used in association with any known or future reversible technology.

1. Introduction

In general ESOP solution yields better result than SOP solution [1] and many useful methods have been developed for minimizing multi-output Boolean functions into ESOP form [2-4]. However, as the Moore's Law is approaching towards its bottom limit, synthesis of ESOPs using the future technologies become inevitable.

Landauer [5] showed that a computational system built using traditional irreversible logic gates such as AND or multiplexer leads inevitably to energy dissipation, regardless of the technology used to realize the gates. Bennett [6] showed that for power not to be dissipated in an arbitrary circuit, it is necessary that the circuit be built from ***reversible gates***. In principle, reversible logic gates dissipate arbitrary little heat and the use of reversible operations is likely to become more attractive in future technologies. It was shown that reversible gates can be built using (i) CMOS technology [7, 8], (ii) optical technologies [9, 10], (iii) quantum logic technologies [11, 12], (iv) DNA technology [10], and (v) mechanical technology (nanotechnology) [13]. Therefore, a worthy approach might be to develop methods for synthesizing multi-output ESOPs using reversible gates.

A reversible gate is a circuit that has the same number of inputs and outputs and has one-to-one mappings between input vectors and output vectors; thus the input vector states can be always uniquely reconstructed from the output vector states. A reversible gate with k inputs and k outputs is called a *k*k* gate. For example, the well known 2*2 ***Feynman gate*** is described by the equations $\{P = A, Q = A \oplus B\}$. It is a reversible gate as for each combination of outputs $\{P, Q\}$ there is exactly one combination of inputs $\{A, B\}$.

Reversible logic synthesis is different and difficult than the classical logic synthesis for the following reasons:

(i) All gates in a reversible circuit must be reversible.
(ii) In reversible circuits, fan-out of every signal, including primary inputs, is one. If two copies of a signal are required, a copying circuit is used.
(iii) All outputs of a reversible gate are not used in the circuit. The unused outputs are called garbage outputs. A good reversible logic synthesis method should minimize the number of garbage outputs.
(iv) In reversible logic synthesis, constants are applied to some of the inputs of a reversible gate. The total number of input constants are kept as minimum as possible.
(v) The graph of the reversible circuit must be a DAG (directed acyclic graph), which means that there must be no loops of gates or internal loops in a gate.

There are some useful binary reversible gates. The only 1*1 reversible gate is the ***inverter*** described by the equation $\{P = A'\}$. The well-known ***Feynman gate*** is a 2*2 gate as mentioned earlier. There are some useful 3*3 reversible gates, they are ***Toffoli gate*** [14] described by the equations $\{P = A, Q = B, R = AB \oplus C\}$, ***Fredkin gate*** [15] described by the equations $\{P = A, Q = \textit{if } A \textit{ then } C \textit{ else } B, R = \textit{if } A \textit{ then } B \textit{ else } C\}$, and ***Kerntopf gate*** [16-18] described by the equations $\{P = 1 \oplus A \oplus B \oplus C \oplus AB, Q = 1 \oplus AB \oplus B \oplus C \oplus BC, R = 1 \oplus A \oplus B \oplus AC\}$. *k*k* generalization of these gates are also reported in [19, 20].

Systematic logic synthesis algorithms for reversible logic are still very immature, and some methods have been reported in the literature [18-25]. In [24], compositional synthesis methods for reversible logic have been presented. The simplest structure for composition is cascades. Reversible logic synthesis using cascades of gates is presented in [19, 20, 25]. The cascades have the same number of intermediate signals at every level. The reversible cascades are recently extensively studied, as many well-known standard logic synthesis methods can be adapted to reversible cascades. ESOP realization with reversible cascades is reported in [19, 20, 25]. In [19], a method for multi-output ESOP cascade is presented, but no systematic design method and experimental results were provided. In [20] and [25], a method for single-output ESOP cascade and a method for multi-output ESOP cascades are presented, respectively, where generalized Maitra terms are used to represent sets of ESOP products. Although the methods from [20,25] can be easily extended to multi-output functions, this extension reduces the search efficiency and only small functions could be synthesized. Therefore it is important to look for efficient methods to synthesize ***multi-output ESOP*** functions in reversible cascades.

In this paper, a new $k*k$ family of reversible gate is proposed and a synthesis method for multi-output (factorized) ESOP is presented using cascades of the proposed gate. In this method, two graph-based data structures – connectivity tree and implementation graph are used. Experimental results with some MCNC benchmark functions show that the number of gates in most of the multi-output ESOP cascades is almost equal to the number of products in the multi-output ESOP. However, this cascaded realization of multi-output ESOP generates a large number of garbage outputs and requires a large number of input constants, which need to be reduced in the future research. This synthesis method is technology-independent and can be used in association with any known or future reversible technology.

2. New Family of $k*k$ Reversible Gates

A new generalized $k*k$ reversible gate family is proposed in Figure 1 described by the equations $\{P_1 = A_1, \cdots, P_{k-2} = A_{k-2}, P_{k-1} = f_{k-2}A_{k-1} \oplus A_k, P_k = f'_{k-2}A'_{k-1} \oplus A'_k\}$, where f_{k-2} is an arbitrary function of $A_1, A_2, \cdots, A_{k-2}$ and f'_{k-2} is the complement of the earlier function. From the truth table it can be shown that the gate is a reversible gate. The gate is a $(k-2)$-through gate, which means that $(k-2)$ inputs are given at the output unaltered. Depending on f_{k-2}, many possible gates can be constructed. Input lines A_{k-1} and A_k are used as control inputs. Depending on the values of these input signals, the gate can be used in many different modes of operations in cascades as shown in Figure 2.

3. Synthesis of Multi-Output (Factorized) ESOP

3.1 Synthesis of Multi-Output ESOP

For ESOP realization using cascades of new $k*k$ gates, we assume that $f_{k-2} = \prod_{\exists i \in \{1,2,\cdots,k-2\}} A_i$, that is, f_{k-2} is a product of some of the input variables $A_1, A_2, \cdots, A_{k-2}$. For realizing an ESOP, we will use the operating modes of Figure 2(a) and 2(e).

In multi-output ESOP, some products may be common among more than one ESOP. Getting advantage of such a common product can not be handled in a method similar to that used in circuits with classical gates. The main constraint is that the fan-out of all signals in a reversible gate is one. Therefore, this problem has to be handled in a different way to achieve maximum benefit.

In the synthesis method for multi-output ESOP using the cascades of new gate family, it is assumed that the multi-output ESOP is already minimized and available as ESOP. The method is illustrated below using the arbitrary multi-output ESOP: $F_1 = AB' \oplus AB'C$, $F_2 = AC \oplus A'B'C$, $F_3 = AB' \oplus AB'C \oplus BC'$, $F_4 = AC$, and $F_5 = AB' \oplus AC \oplus BC'$.

Step 1: Prepare Product Sharing Information. The product sharing information among the different ESOPs is summarized in Table 1. The table shows the occurrence of all product terms in different ESOPs and the total number of occurrences of each product term.

Step 2: Prepare Connectivity Tree. Depending on the information of the product sharing table, the ***connectivity tree*** (or forest of trees) for the product terms is constructed as shown in Figure 3(a). In the connectivity tree each node represents a product term. The product term(s) with the highest number of occurrences is placed at the top level. Then the product terms with the decreasing order occurrence numbers are placed below. The nodes are connected by directed edges so that EXORing the products in a path forms a function. For example, $AB' \oplus AB'C \oplus BC'$ forms the function F_3. A node must not have more than one inward edge and if needed a separate node is created for the product term. For example, two nodes are created for each of the product terms BC' and AC. Though the fan-out of a signal is limited to one, multiple outward edge from a node will be allowed and this will be handled in an efficient way.

Step 3: Prepare Implementation Graph. From the connectivity tree, the ***implementation graph*** is created as shown in Figure 3(b). In this graph a node represents a new $k*k$ gate. The left inward edge represents A_{k-1} input line and the right inward edge represents A_k input line. The left outward edge represents P_{k-1} output line and the right outward edge represents P_k output line. The product inside the node represents the function f_{k-2}. In all nodes of the implementation graph, the A_{k-1} input is set to 0 and the A_k input is used as the signal propagation path. At the top node A_k input is also set to 0 to realize the first product term. In the connectivity tree, some nodes have multiple outward edges requiring multiple fan-out. It is clear that the left outward edge of an implementation node copies the input value at A_k and thus provides the fan-out of that value. This phenomenon is used to manage multiple fan-out requirement of the connectivity tree.

Step 4: Prepare Multi-Output ESOP Cascade. The implementation graph is realized by cascades of new $k*k$ gates as shown in Figure 3(c). The 0s at the P_{k-1} output of the first and the sixth gates are connected to the A_{k-1} inputs of the next gate to reduce the input constants as well as garbage outputs.

In this particular example, only 2 garbage outputs are generated (passing through primary input signals along the cascade is not counted as garbage output) and 7 input constants are needed. In general, the number of input constants is exactly equal to the number of gates in the cascade. However, the number of gates and the number of garbage outputs depend on the product sharing among the ESOPs. For a single-output ESOP realization, the number of gates is equal to the number of product terms in minimized ESOP. The number of garbage outputs is one less than the number of product terms and the number of input constants is exactly equal to the number of gates in the cascade.

3.2 Synthesis of Factorized ESOP

For factorized ESOP realization using cascades of new $k*k$ gates, we assume that f_{k-2} may be either $f_{k-2} = \prod_{3\in\{1,2,\cdots,k-2\}} A_l$ or $f_{k-2} = \sum_{3\in\{1,2,\cdots,k-2\}} \oplus A_l$, that is f_{k-2} may be a product or EXOR-sum of some of the input variables $A_1, A_2, \cdots, A_{k-2}$. For realizing factorized ESOP, we will use the operating modes of Figures 2(a), 2(e), and 2(g).

Synthesis of factorized ESOP is illustrated using the symmetric function $E_2^4 = \sum_{1\le i\le j\le 4} \oplus x_i x_j = x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4$. This function can be factorized as $E_2^4 = (x_1 \oplus x_2)(x_3 \oplus x_4) \oplus x_1x_2 \oplus x_3x_4$. The implementation graph is shown in Figure 4(a). The implementation graph is realized by cascades of new $k*k$ gates as shown in Figure 4(b). In this implementation 4 gates are needed. It generates 3 garbage outputs and needs 4 input constants. In this method, it is assumed that the ESOP is already factorized.

Using the method of Section 3.1, multi-output factorized ESOP can be realized.

4. Experimental Results

A program for synthesizing multi-output ESOP cascades has been developed in C language. In this program, it is assumed that the multi-output ESOP is already minimized and available as ESOP. For this purpose we minimized 38 MCNC benchmark functions into ESOP form using EXORCISM4 program [4]. Multi-output ESOP cascades are then synthesized using the developed program. The experimental results are given in Table 2. In the table, column 5 gives the number of gates in the multi-output ESOP cascade. Column 6 gives the number of garbage outputs generated in the cascade and column 7 gives the input constants (0s) needed for the cascade. In the classical multi-output ESOP realization more than one fan-out of a gate is permissible, but in the reversible circuit fan-out of all signals are restricted to only one. For this reason, in the connectivity tree, the inward edge of a node is restricted to one and if needed additional nodes are created for a product term (see Figure 3(a)). Each of these nodes represents a gate in the resultant cascade. Therefore, it is clear that the number of gates in the cascade will be at least equal to the number of product terms in the multi-output ESOP and in many cases this number will be grater than the number of product terms. In the table, column 8 gives the ratio of number of gates in the cascade to number of products in the multi-output ESOP. This value can be interpreted as the average number of nodes created for a product term. For example, in 5xp1 function on average 1.667 nodes are created for a product term. In the multi-output ESOP cascade, some but not all gates produce garbage outputs. The last column of the table gives the ratio of number of garbage outputs to number of gates in the cascade. This value can be interpreted as the average number of garbage outputs created per gate. For example, in 5xp1 function on average a gate produces 0.818 garbage output. Alternatively, it can be said that 81.8% gates produce garbage output.

For four single-output ESOP functions (9sym, 9symml, t481, and xor5), the number of gates in the cascade is exactly equal to the number of products in the ESOP. For five multi-output ESOP functions (adr2, c8, con1, frg1, and unreg), the number of gates in the cascade is also exactly equal to the number of products in the SOP. For 20 multi-output ESOP functions, the number of gates in the cascade is almost equal to the number of products in the ESOP. ***These experimental results show that the product sharing technique of the developed algorithm is so efficient that the number of reversible gates (with single fan-out capability) in the resultant multi-output ESOP cascade is almost equal to the number of products (classical AND gates with multiple fan-out capability) in the multi-output ESOP.*** For the remaining nine functions, the number of gates in the cascade is more than double or higher times than the number of products in the ESOP. The nature of product sharing among the outputs of these functions is such that in the connectivity tree many nodes are needed to be created for a product term.

For single-output ESOP, the number of garbage outputs (passing-through primary input signals along the cascade are not counted as garbage outputs) generated in the cascade is one less than the number of reversible gates in the cascade and this is reflected in the result of the single-output ESOP functions (9sym, 9symml, t481, and xor5). For multi-output ESOP, the number of garbage outputs generated in the cascades depends on the nature of product sharing among the outputs. The experimental results show that this value is relatively high. However, the number of garbage signals can be minimized by the use of local mirror and spy circuits [15,23]. This would be a topic for further research.

In the new *k*k* gates, constant 0s are applied to the control inputs for selecting the mode of operation of the gate in the cascade (see Figure 2). For both single-output and multi-output ESOPs, the number of input constants is exactly equal to the number of reversible gates in the cascade. This number is a bit large and need to be minimized in the future research.

5. Conclusion

In this paper, ***we proposed a new k*k reversible gate family and associated synthesis method for multi-output (factorized) ESOP cascades of new gates.*** For utilizing the benefit of product sharing among the outputs of the multi-output ESOPs, ***two graph-based data structures*** – connectivity tree and implementation graph are used.

We wrote a C program to synthesize multi-output ESOP cascades and experiment was done with 38 MCNC benchmark functions. ***The experimental results show that the product sharing technique of the developed algorithm is so efficient that the number of reversible gates (with single fan-out capability) in the resultant multi-output ESOP cascade is almost equal to the number of products (classical AND gates with multiple fan-out capability) in the multi-output ESOP.***

All the synthesis methods presented in this paper are technology independent and can be used in association with any known or future reversible technology.

The future research includes: (i) investigating the possibility of reducing the number of garbage outputs in the multi-output ESOP cascades by the use of local mirror and spy circuits or other techniques, (ii) investigating the way of reducing the number of input constants in the multi-output ESOP cascades, (iii) writing C program to synthesize multi-output factorized ESOP cascades, and (iv) comparing various ESOP-based cascades with respect not only to the gate number but also total number of inputs to gates and gate complexity in quantum realization.

References

[1] T. Sasao (ed), *Logic Synthesis and Optimization*, Kluwer Academic Publisher, 1993, pp. 287-312.

[2] T. Sasao, EXMIN2: A simplification algorithm for Exclusive-or-Sum-of-Products expressions for multi-valued input two-valued output function. *IEEE Trans. CAD*, Vol. 12, No. 5, May 1993, pp. 621-623

[3] N. Song and M. Perkowski, Minimization of exclusive sum of products expression for multi-output multi-valued input, incompletely specified functions. *IEEE Trans. CAD*, Vol. 15, No. 4, April 1996, pp. 385-395.

[4] A. Mishchenko, M. Perkowski, Fast heuristic minimization of exclusive sum-of-products. Proceedings of the 5th International Reed-Muller Workshop, August 2001, pp. 242-250.

[5] R. Landauer, Irreversibility and heat generation in the computational process. *IBM Journal of Research and Development*, 5, pp. 183-191, 1961.

[6] C. Bennett, Logical reversibility of computation. *IBM Journal of Research and Development*, 17, pp. 525-532, 1973.

[7] A. De Vos, Towards reversible digital computers. *Proceedings of European Conference on Circuit Theory and Design*, Budapest, pp. 923-931, 1997.

[8] B. Desoete, A. De Vos, M. Sibinski, and T. Widerski, Feynman's reversible logic gates implemented in silicon. *Proceedings of 6th International Conference MIXDES*, pp. 496-502, 1999.

[9] P. Picton, Optoelectronic, multivalued, conservative logic. *International Journal of Optical Computing*, 2, pp. 19-29, 1991.

[10] P. Picton, A universal architecture for multiple-valued reversible logic. *MVL Journal*, 5, pp. 27-37, 2000.
[11] J. A. Smolin, and D. P. DiVincenzo, Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. *Physical Review A*, 53, 1996, pp. 2855-2856.
[12] A. Peres, Reversible logic and quantum computers, *Physical Review A*, 32, pp. 3266-3276, 1985.
[13] R. C. Merkle, Two types of mechanical reversible logic. *Nanotechnology*, 4, pp. 114-131, 1993.
[14] T. Toffoli, Reversible computing. In *Automata, Languages and programming*, Springer Verlag, 1980, pp. 632-644.
[15] E. Fredkin and T. Toffoli, Conservative logic. *International Journal of Theoretical Physics*, 21, pp. 219-253, 1982.
[16] P. Kerntopf, A comparison of logical efficiency of reversible and conventional gates. *Proceedings of 9th IEEE Workshop on Logic Synthesis*, pp. 261-269, 2000.
[17] P. Kerntopf , Synthesis of multipurpose reversible logic gates. *Proceedings of EUROMICRO Symposium on Digital Systems Design*, 2002, pp. 259 – 266.
[18] M. Perkowski, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, X. Song, A. Al-Rabadi, L. Jozwiak, A. Coppola, B. Massey, Regularity and symmetry as a base for efficient realization of reversible logic circuits. *Proceedings of IWLS 2001*, pp. 90-95, 2001.
[19] A. Khlopotine, M. Perkowski, and P. Kerntopf, Reversible logic synthesis by gate composition. *Proceedings of IWLS* 2002. pp. 261 – 266.
[20] A. Mishchenko and M. Perkowski, Reversible Maitra cascades for single-output functions. *Proceedings of IWLS* 2002. pp. 197 – 202.
[21] L. Storme, A. De Vos, and G. Jacobs, Group theoretical aspects of reversible logic gates. *Journal of Universal Computer Science*, 5, pp. 307-321, 1999.
[22] M. Perkowski, A. Al-Rabadi, P. Kerntopf, A. Mishchenko, M. Chrzanowska-Jeske, Three-dimensional realization of multi-valued functions using reversible logic. *Proceedings of ULSI 2001 Workshop*, Warsaw, Poland, May 2001.pp 47 – 53.
[23] M. Perkowski, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, X. Song, A. Al-Rabadi, L. Jozwiak, A. Coppola, B. Massey, Regular realization of symmetric functions using reversible logic. *Proceedings of Euro-Micro 2001*, pp. 245 – 252.
[24] M. Perkowski, L. Jozwiak, P. Kerntopf, A. Mishchenko, A. Al-Rabadi, A. Coppola, A. Buller, X. Song, M. M. H. A. Khan, S. Yanushkevich, V. Shmerko, and M. Chrzanowska-Jeske, A general decomposition for reversible logic. *Proceedings of RM 2001*. pp. 119 – 138.
[25] A. Mishchenko and M. Perkowski, Logic synthesis of reversible wave cascades. *11th IEEE/ACM Workshop on Logic and Synthesis*, 2002, pp. 197-202.

Tables and Figures

Table 1. Product sharing information in multi-output ESOP.

Function / Product	F_1	F_2	F_3	F_4	F_5	No of Occurrence
AB'	X		X		X	3
$AB'C$	X		X			2
AC		X		X	X	3
BC'			X		X	2
$A'B'C$		X				1

Table 2. Experimental results of multi-output ESOP cascades of new reversible gates for some MCNC benchmark functions.

MCNC Benchmark Functions (expressed as ESOP)				Multi-output ESOP Cascades				
Function Name	No. of Inputs	No. of Outputs	No. of Products	No. of Gates	No. of Garbage Outputs*	No. of Input Constants	No. of Gates in Cascade : No. of Products in ESOP	No. of Garbage Output : No. of Gates
5xp1	7	10	33	55	45	55	1.667	0.818
9sym	9	1	52	52	51	52	1.000	0.981
9symml	9	1	52	52	51	52	1.000	0.981
adr2	4	3	7	7	4	7	1.000	0.571
alu2	10	6	72	79	73	79	1.097	0.924
alu4	14	8	440	473	465	473	1.075	0.983
b12	15	9	28	35	26	35	1.250	0.743
bw	5	28	22	202	174	202	9.182	0.861
c8	28	18	50	50	32	50	1.000	0.640
cc	21	20	36	39	19	39	1.083	0.487
clif	9	5	66	101	96	101	1.530	0.950
con1	7	2	9	9	7	9	1.000	0.778
cu	14	11	16	18	7	18	1.125	0.389
duke2	22	29	79	280	251	280	3.544	0.896
ex5	8	63	72	571	508	571	7.931	0.890
f51m	8	8	31	46	38	46	1.484	0.826
frg1	28	3	115	115	112	115	1.000	0.974
i1	25	16	21	24	8	24	1.143	0.333
Inc	7	9	28	70	61	70	2.500	0.871
misex1	8	7	12	35	28	35	2.917	0.800
misex2	25	18	27	36	18	36	1.333	0.500
misex3c	14	14	231	315	301	315	1.364	0.956
pdc	16	40	252	467	427	467	1.853	0.914
rd53	5	3	15	18	15	18	1.200	0.833
rd73	7	3	39	46	43	46	1.179	0.935
rd84	8	4	62	68	64	68	1.097	0.941
sao2	10	4	28	46	42	46	1.643	0.913
seq	41	35	249	1742	1707	1742	6.996	0.980
spla	16	46	270	765	719	765	2.833	0.940
squar5	5	8	20	27	19	27	1.350	0.704
t481	16	1	13	13	12	13	1.000	0.923
table3	14	14	167	768	754	768	4.599	0.982
table5	17	15	156	668	653	668	4.282	0.978
temp	4	3	7	8	5	8	1.143	0.625
unreg	36	16	48	48	32	48	1.000	0.667
vg2	25	8	184	200	192	200	1.087	0.960
xor5	5	1	5	5	4	5	1.000	0.800
Z5xpl	7	10	33	55	45	55	1.667	0.818

* Passing-through primary input signals along the cascade are not counted as garbage output.

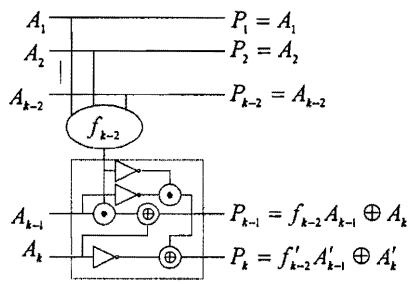


Figure 1. A new generalized $k*k$ reversible gate family.

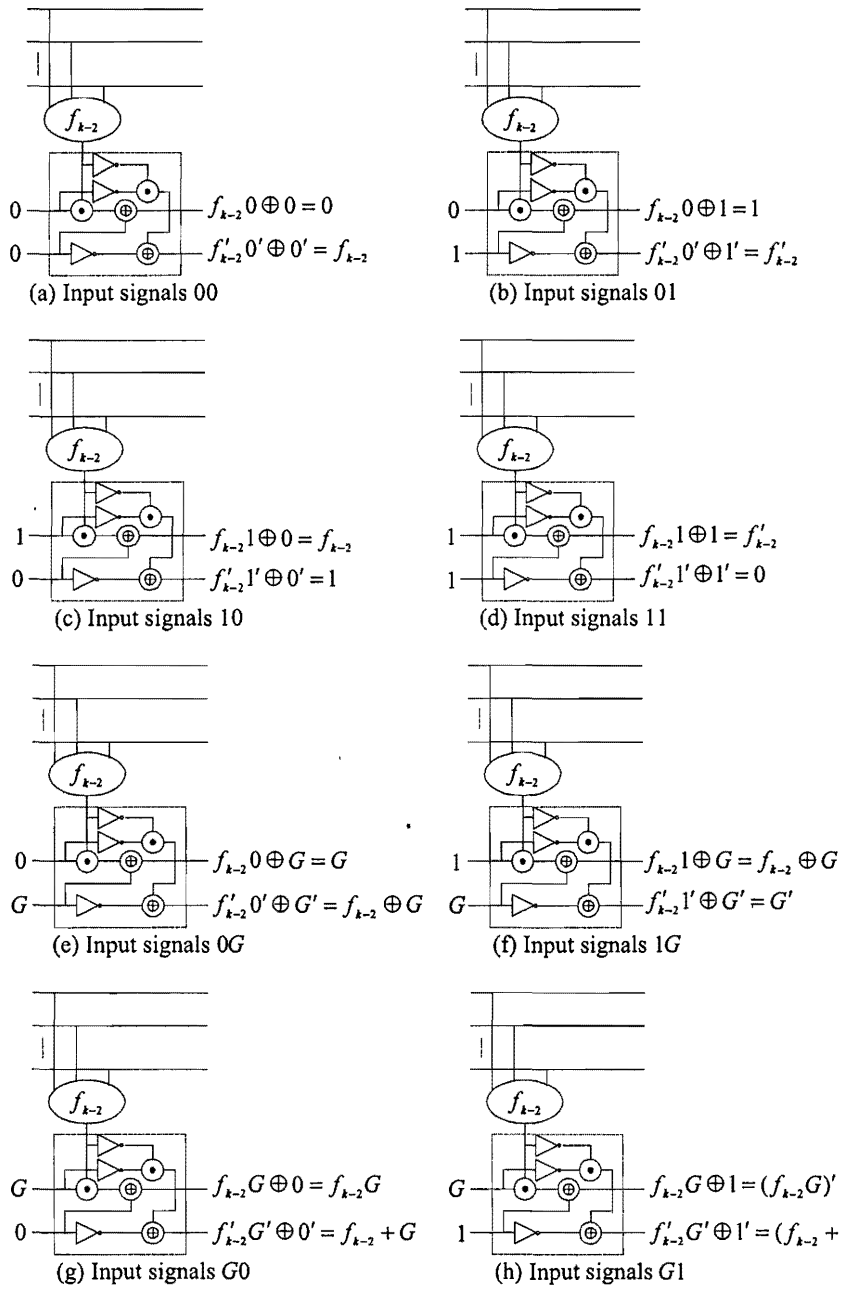
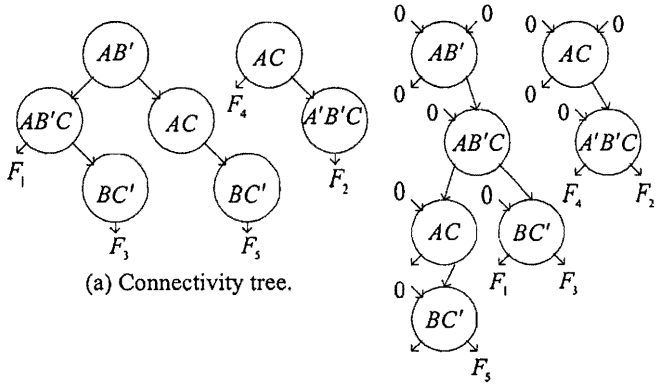
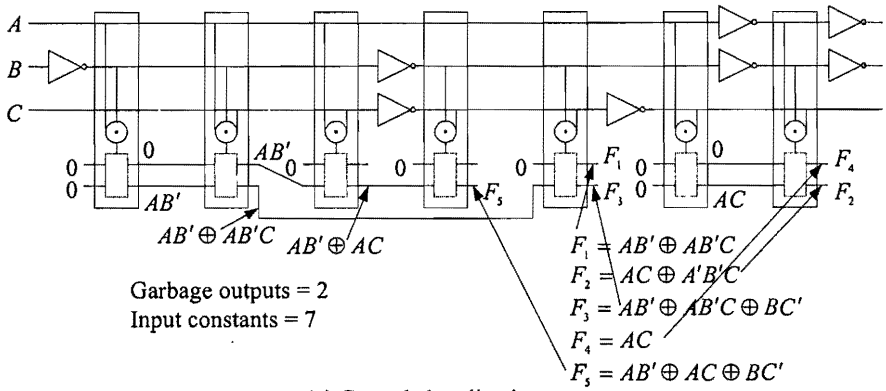


Figure 2. Different modes of operation of the new k^*k generalized reversible gate family in cascades.



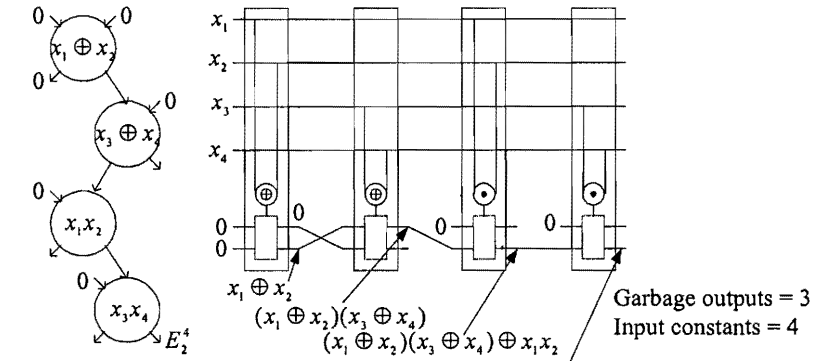
(a) Connectivity tree.

(b) Implementation graph.



(c) Cascaded realization

Figure 3. Synthesis of multi-output ESOP.



(a) Implementation graph

(b) Cascaded realization

Figure 4. Synthesis of single-output factorized ESOP.