Research on Reversible Functions Having Component Functions with Specified Properties: An Overview

Paweł Kerntopf^{1(✉)}, Claudio Moraga², Krzysztof Podlaski³,
and Radomir Stanković⁴

¹ Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland
`pawel.kerntopf@gazeta.pl`

² Faculty of Computer Science, Technical University of Dortmund,
Dortmund, Germany
`claudio.moraga@tu-dortmund.de`

³ Faculty of Physics and Applied Informatics, University of Łódź, Łódź, Poland
`podlaski@uni.lodz.pl`

⁴ Department of Computer Science, Faculty of Electronic Engineering,
University of Niš, Niš, Serbia
`radomir.stankovic@gmail.com`

Abstract. In the traditional logic synthesis, different classifications of non-reversible Boolean functions have found many applications. Recently, some attempts to deal with classifications of reversible functions have been published. In this paper, an overview of our results towards constructing a new classification of reversible functions is presented. These results were obtained due to our discussions during two Short Term Scientific Missions (STSMs) as well as during our further research in the framework of COST Action IC1405 “Reversible Computation - Extending Horizons of Computing” and were published in five papers.

Keywords: Reversible functions · Component functions · Classification

1 Introduction

Recent advances in nanotechnology, low-power design, and quantum computing have renewed interest in reversible logic synthesis since they allow for reducing the power dissipation in related circuits and the potential speed-up in quantum computations. More details can be found in [4, 25] and references therein.

A reversible function is defined as a bijective mapping $f : A^n \rightarrow A^n$, where A is any finite set of elements which can be conveniently identified with non-negative integers $\{0, 1, \dots, p - 1\}$. In particular, for $p = 2$ and $p = 3$, we speak about binary or Boolean and ternary reversible functions, respectively. Therefore, an n -variable reversible function is actually a permutation on A^n , and can

be viewed as a vector of n functions called the *component functions* (CFs), i.e., $F = (f_1, f_2, \dots, f_n)$. In [27], the term *components* is applied in the similar meaning, meanwhile in the literature on cryptography the term *coordinate functions* is used, see e.g., [3, 30]. However, in [30] the term *component function* means a linear combination of *coordinate functions*.

Correspondingly, a reversible circuit is a circuit that realises a reversible function, i.e., performs a bijective mapping of n input signals onto n output signals in a manner specified by the function to be realised.

Recently in [13, 14], we discussed the question if it is possible to extend a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ into a reversible function $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$, *under the condition that all its component functions have a homogeneous property*. The term homogeneous property means that all component functions express the same particular property Boolean functions might have, e.g. all the component functions belong to the same equivalence class in a particular classification of Boolean functions. The motivation was that if such an embedding of a Boolean function into a reversible function is possible, then new classes of reversible functions can be defined. In [15, 17] the same question is explored for ternary functions $F : \{0, 1, 2\}^n \rightarrow \{0, 1, 2\}^n$ and we have shown that there are significant differences in the theory of binary and ternary reversible functions in the case of linear component functions.

As homogeneous properties, we have chosen typical ones considered in classical logic synthesis: symmetry, affinity, linearity, nonlinearity, self-duality, self-complementarity, monotonicity, unateness (see, e.g. [31]). In papers [13–15, 17] the exemplary functions used in proofs of the results were obtained in a constructive manner. In [16] the results on properties of component functions of Boolean reversible functions obtained by the extrapolation approach were demonstrated. An overview of the most relevant of these results in the binary case is presented here. Because of lack of space we have omitted our results on reversible multiple-valued functions. The reader can find them in [15, 17].

The presentation is organised in the following way. For the sake of completeness, necessary definitions and basic results from the theory of standard Boolean as well as from reversible Boolean functions are provided in Sect. 2. In Sect. 3, a brief overview of related and background work is presented. Section 4 demonstrates our theoretical results on properties of component functions of reversible functions. Section 5 describes the results of our research on the existence of Boolean reversible functions with all component functions belonging to different equivalence classes while considering well-known and newly constructed reversible functions defined for any number of variables. Section 6 presents our numerical calculations of all equivalence classes of balanced Boolean functions up to $n = 4$ and all reversible functions up to $n = 3$. Finally, Sect. 7 describes our results on the existence of Boolean reversible functions with specified properties of all component functions obtained by extrapolating some properties of reversible functions. The presented research is summarised in Sect. 8.

2 Preliminaries

In this section, the basic definitions and known results are provided for the convenience of the reader. Let us first briefly survey fundamental notions related to standard Boolean functions and reversible Boolean functions.

First we present notation and terminology for fundamental notions. The symbols $+$, $-$, $\cdot$ denote ordinary addition, subtraction, and multiplication, respectively. For arbitrary elements x and y in the set $\{0, 1\}$ basic operations in this set (one unary and three binary operations) are defined in the usual way:

Negation $x' = 1 - x$, i.e. if the argument x is 0, then the result is 1, otherwise it is 0;

Product $xy = x \cdot y$, i.e. its value is 1 if and only if both arguments are 1;

Sum $x \vee y = x + y - x \cdot y$, i.e. its value is 0 if and only if both arguments are 0;

EXOR $x \oplus y = x + y \pmod{2} = x + y - 2 \cdot x \cdot y$, i.e. its value is 1 if and only if exactly one argument is 1.

In classical logic synthesis, the basic representation of a Boolean function is the Sum-of-Products expression (SOP). In the field of reversible circuit synthesis two other representations are commonly used. Any Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ can be described using an EXOR-sum of products (ESOP) expression. In ESOPs each variable may appear in both uncomplemented and complemented forms. The Positive Polarity Reed-Muller (PPRM) expression is an ESOP expression which uses only uncomplemented variables. It is a canonical expression and for small functions can be easily generated from a truth table or other representations of the Boolean function.

A Boolean function $f(x_1, x_2, \dots, x_n)$ depends essentially on its variable x_i if and only if $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \neq f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$.

Definition 1. A Boolean function depending essentially on all its variables is called non-degenerate, otherwise it is called degenerate.

Example 1. There are 16 functions of two variables x and y : $0, 1, x, x', y, y', xy, x'y, xy', x'y', x \vee y, x' \vee y, x \vee y', x' \vee y', x \oplus y = x' \oplus y', x' \oplus y = x \oplus y'$. The first six of them are degenerate: the first two depend essentially on none of the variables, the next four depend essentially on only one of the variables. **■**

Let us define an order relation in the set $\{0, 1\}$ in the usual way: $0 < 1$ and a partial order relation in the set $\{0, 1\}^n$: for any two vectors $\mathbf{a} = (a_1, a_2, \dots, a_n)$, $\mathbf{b} = (b_1, b_2, \dots, b_n)$ in $\{0, 1\}^n$ $\mathbf{a} \leq \mathbf{b}$ if and only if $a_i \leq b_i$ for $1 \leq i \leq n$.

Definition 2. A Boolean function f is monotone increasing if and only if $\mathbf{a} \leq \mathbf{b}$ implies $f(\mathbf{a}) \leq f(\mathbf{b})$ which will simply be called a monotone function. By changing the inequalities into inverse ones we obtain a definition of monotone decreasing function.

Example 2. Both the constant functions 0 and 1 are monotone increasing and monotone decreasing. There are six monotone increasing functions of two variables x and y : $0, 1, x, y, xy, x \vee y$. Similarly, there are six monotone decreasing functions of two variables x and y : $0, 1, x', y', x'y', x' \vee y'$. **■**

Definition 3. A Boolean function $f(x_1, x_2, \dots, x_n)$ is called *unate* (or *mixed monotone*) if and only if it is a constant or there exists its SOP representation using either uncomplemented or complemented literals for each variable.

Example 3. There are 14 unate functions of two variables x and y : only functions $x \oplus y$ and $x' \oplus y$ are not unate. I

Definition 4. A Boolean function $f(x_1, x_2, \dots, x_n)$ is called *threshold* (or *linearly separable*) if and only if there exist real numbers $a_1, a_2, \dots, a_n$, and b such that $f = 1$ if the sum of all $a_i x_i$, $1 \leq i \leq n$, is greater than or equal to b , and $f = 0$ otherwise.

Example 4. All unate functions of up to three variables are threshold functions. Thus, for two variables there are 14 threshold functions (i.e., all except $x \oplus y$ and $x' \oplus y$), in particular,

$$\begin{aligned} \text{when } a_1 = a_2 = 1 \quad \text{and } b = 1.5 \quad \text{then } f &= xy, \\ \text{when } a_1 = a_2 = 1 \quad \text{and } b = 0.5 \quad \text{then } f &= x \vee y, \\ \text{when } a_1 = a_2 = -1 \quad \text{and } b = -0.5 \quad \text{then } f &= x'y', \\ \text{when } a_1 = a_2 = -1 \quad \text{and } b = -1.5 \quad \text{then } f &= x' \vee y'. \end{aligned}$$

The 4-variable function $f(x_1, x_2, x_3, x_4) = x_1 x_2 \vee x_3 x_4$ is an example of a monotone increasing function which is not a threshold function. I

Definition 5. A Boolean function f on an odd number of arguments is called *majority function* if and only if $f = 1$ when more than half of the arguments are 1.

Example 5. The 3-variable majority function $f(x, y, z) = xy \oplus xz \oplus yz$ is a threshold function, where $a_1 = a_2 = a_3 = 1$ and $b = 2$. I

It is well known that the following result holds.

Lemma 1.

- (1) Every threshold function is a unate function.
- (2) Every majority function is a threshold function.

Definition 6. A Boolean function f is *linear with respect to a variable x_i* if it can be expressed in the form $f = x_i \oplus g$, where $\oplus$ denotes XOR operation and g is a function independent of x_i (then the variable x_i is called *linear in f*). A function has property *LV* if it contains at least one linear variable. A function f is called *affine* if and only if each of variable x_i is either linear in f , or f does not depend on x_i , i.e. $f(x_1, x_2, \dots, x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n$, where $a_0, a_1, a_2, \dots, a_n \in \{0, 1\}$. If $a_0 = 0$ then it is called *linear*. Any affine function which is not linear can be obtained by negating an appropriate linear function. A Boolean function which is not affine is called *nonlinear*.

Example 6. $f_1(x, y, z) = x \oplus y \oplus yz$ is linear with respect to x as then $g = y \oplus yz$ is independent of x , but f_1 is not linear with respect to y as then $g = x \oplus yz$ is dependent of y . Similarly, $f_2(x, y) = x \oplus y \oplus xy$ is neither linear with respect to x , nor to y . I

Definition 7. A Boolean function is (totally) symmetric if any permutation of all its variables does not change the function.

There are 2^{n+1} symmetric Boolean functions.

Definition 8. If any permutation of a proper subset S of the variables of cardinality at least 2 does not change the Boolean function f , then f is called a partially symmetric function with respect to S and S is called a partial symmetry of variables of f . The collection of maximal partial symmetry subsets of variables of f is called a partial symmetry profile and is denoted by S_f . The partial symmetry profile of a totally symmetric Boolean function $f(x_1, x_2, \dots, x_3)$ is equal to $\{\{x_1, x_2, \dots, x_n\}\}$. Let partial symmetry profiles of Boolean functions $f_1, f_2, \dots, f_n$ be denoted by $S_1, S_2, \dots, S_n$, respectively. The intersection of such profiles is the collection of subsets of variables obtained by taking all possibilities of performing intersection operation on an element in S_1 , an element in S_2 , $\dots$, and an element in S_n . If the intersection operation on $S_1, S_2, \dots, S_n$ does not contain an element with at least two variables, then there does not exist a partial symmetry subset of all functions $f_1, f_2, \dots, f_n$.

Example 7. Let $f(u, v, w, x, y, z) = u \oplus vw \oplus xyz$ and $g(u, v, w, x, y, z) = uv \oplus w \oplus xyz$. Then the partial symmetry profile of f is $S_f = \{\{v, w\}, \{x, y, z\}\}$, the partial symmetry profile of g is $S_g = \{\{u, v\}, \{x, y, z\}\}$ and the intersection of S_f and S_g is equal to $\{\{v\}, \phi, \{x, y, z\}\}$, i.e. both functions f and g are partially symmetric with respect to $\{\{x, y, z\}\}$ as well as this subset is the only one partial symmetry subset of both functions f and g . I

Definition 9. A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called balanced if it takes value 1 the same number of times as value 0.

Example 8. There are 70 balanced Boolean functions on 3 arguments, including degenerate ones. Only four of them are totally symmetric, namely parity and majority functions and their negations:

$$\begin{array}{ll} \text{parity} & x \oplus y \oplus z \quad 1 \oplus x \oplus y \oplus z, \\ \text{majority} & xy \oplus xz \oplus yz \quad 1 \oplus xy \oplus xz \oplus yz. \end{array}$$

Eight of the balanced functions, including degenerate ones, are partially symmetric with respect to each 2-element subset of variables, for instance, functions

$$\begin{array}{ll} x \oplus y, & 1 \oplus x \oplus y, \\ xy \oplus z, & 1 \oplus xy \oplus z, \\ x \oplus y \oplus xy \oplus z, & 1 \oplus x \oplus y \oplus xy \oplus z, \\ x \oplus y \oplus xy \oplus xz \oplus yz, & 1 \oplus x \oplus y \oplus xy \oplus xz \oplus yz. \end{array}$$

are partially symmetric with respect to $\{x, y\}$. I

Definition 10. Two Boolean functions are:

- (1) *P-equivalent* if they can be converted to each other by the permutation of variables,

- (2) NP-equivalent if they can be converted to each other by the negation and/or permutation of variables,
- (3) NPN-equivalent if they can be converted to each other by negation of variables, permutation of variables and negation of the function.

Definition 11. A Boolean function f is self-complementary (SC) if f and f' are NP-equivalent.

Definition 12. A Boolean function f is self-dual (SD) if

$$f(x_1, x_2, \dots, x_n) = f'(x'_1, x'_2, \dots, x'_n).$$

The following results are well-known:

- Lemma 2.** (1) All self-complementary functions are balanced,
 (2) All self-dual functions are self-complementary,
 (3) All functions having property LV are self-complementary,
 (4) If a Boolean function f is linear with respect to a variable x_i then

$$f(x_i = 1) = f'(x_i = 0).$$

In the case of Boolean functions, depending on the operations allowed in a particular classification, the P-equivalent, NP-equivalent, and NPN-equivalent functions are distinguished. In some applications, equivalence classes defined with respect to a restricted set of operations are of a particular interest, as for example, in [5,6]. Here, we are particularly interested in P-equivalent functions when studying the properties of component functions.

Definition 13. A mapping $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is called an $n \times n$ reversible function if it is bijective. Functions which are not reversible are called irreversible. An $n \times n$ reversible function F can be considered as a vector of standard Boolean functions called component functions $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, $1 \leq i \leq n$, which are defined at every $x \in \{0, 1\}^n$ by $F(x) = (f_1(x), \dots, f_n(x))$.

In the truth table of a reversible $n \times n$ Boolean function there are n input columns and n output columns. The output rows of such a truth table form a permutation of the input rows. From the bijectivity of reversible functions it follows that all component functions have to be balanced Boolean functions.

By an analogy with the definition of NPN-equivalence classes for standard Boolean functions, the following definition of equivalence classes for Boolean reversible functions can be given.

Definition 14. Two reversible Boolean functions are NPNP-equivalent if they can be transformed to each other by the following operations (including the combinations that do not use all of these operations):

- (1) Negation of variables,
- (2) Permutation of variables,
- (3) Negation of component functions, and
- (4) Permutation of component functions.

Each reversible function can be treated as a permutation. This is why we also recall basic notions connected with permutations. Let A be any finite set. A permutation on a set A is a bijective mapping from A to itself. Every permutation can be considered as a collection of disjoint cycles. Here such a collection will be called a cycle structure. We will write a cycle in the form $\langle a_1, a_2, \dots, a_k \rangle$, meaning that a_1 is mapped onto a_2 , $\dots$, a_k is mapped onto a_1 . It could be written in different ways, e.g. $\langle a_2, a_3, \dots, a_k, a_1 \rangle$. The number of elements in a cycle is called the length of the cycle. A cycle with the length k is called a k -cycle. A 2-cycle is also called a *transposition*.

3 Previous Work

The motivation for our studies of reversible functions toward constructing their classifications is borrowed from the classical logic synthesis by referring to an analogy with related problems. For example, in classical logic synthesis, the equivalence of two functions under permutation of the variables is an important problem due to applications in the synthesis of multiplexer-based field-programmable gate arrays [5,6]. The problem is called Boolean matching, and two functions match if they have the same P-representative. The extension to NP-representatives is done in [7,8] in solving the Boolean matching problem in cell-library binding.

Classification of Boolean functions is a classical problem in logic synthesis due to its various applications, with fast prototyping and unification of testing procedures being just two of them [28]. However, a considerably smaller amount of work has been done in the classification of reversible functions. In [18,19] it is presented an approach to enumerate equivalence classes of reversible functions with the equivalence classes defined as follows. Denote by G and H the groups of permutations acting on the inputs and outputs of Boolean reversible functions, respectively. Two functions $f_1(x)$ and $f_2(x)$ are equivalent if for each n -tuple x , there is a $g \in G$ and an $h \in H$ such that $f_1(x) = h(f_2(g(x)))$. It is also provided a list of all NPNP-equivalence classes of 3-variable reversible functions as well as a classification based on properties of the inverses of the representative functions for the equivalence classes considered. The lists consist of triples of balanced Boolean functions specified by ESOPs. Unfortunately, using “prime” for negation led to a number of typographical errors which has been discovered by us recently [13] (see Sect. 6).

A technical report from 1962 by C. S. Lorens [18] and an article by the same author [19] can be viewed as a starting point of subsequent work on an enumeration of equivalence classes of reversible functions by several authors [10,20–23,29]. With the exception of [22], these publications consider the classification of binary reversible functions. These publications were discussed mainly by researchers in combinatorial mathematics and cryptography but hardly used and correspondingly rarely if at all referred within the reversible functions community, the main reason probably being that the term *invertible* instead of *reversible*

functions has been used. A classification scheme for reversible functions was the subject of a profound study in [24], however, without a concrete solution proposed.

Recently, certain aspects of the classification problem have been addressed. In [26], the list of all NPNP-equivalence classes for three variable reversible functions from [19] is presented in the context of a study of the complexity of reversible circuits with the representative functions for equivalence classes given in the form of permutations (i.e. without considering individual component functions). The minimal number of nonlinear gates needed in the implementation of reversible functions is used as a classification criterion in [9]. The structure of closed classes of reversible functions is described in [1]. Enumeration of equivalence classes under the action of permutation of the inputs and outputs on the domain and the range is presented in [2].

For the first time in the literature, we solved in [13,14] several problems of the existence of binary reversible functions with all component functions having the same known property (e.g., symmetry, affinity, linearity, nonlinearity, self-duality, self-complementarity, monotonicity, unateness). Solutions of such problems for ternary reversible functions are presented by us in [15]. In [17] we presented results on the existence of ternary/multiple-valued reversible functions with all component functions belonging to different P-equivalence classes. In [16] it is shown how we discovered solutions of some problems by extrapolating properties of previously found reversible functions of 3 and 4 variables.

4 Theoretical Results

This section presents basic theoretical results on properties of component functions of reversible functions. We begin with the following general result:

Theorem 1. *If $f(x_1, x_2, \dots, x_n) = (f_1, f_2, \dots, f_n)$ is an $n \times n$ reversible (irreversible) Boolean function, then the function obtained from f by any of the following transformations*

- *negation of variables,*
- *permutation of variables,*
- *negation of a component function,*
- *permutation of component functions,*

is also reversible (irreversible).

Proof. It is sufficient to notice that any of the above transformations corresponds to a permutation of rows in the truth table, i.e. preserves the property of bijectivity. $\square$

The following result follows directly from Theorem 1.

Corollary 1. *$n \times n$ functions belonging to an NPNP-equivalence class either are all reversible or none of them is reversible.*

There are constraints on using totally and partially symmetric functions as component functions of an $n \times n$ reversible function $f(x_1, x_2, \dots, x_n) = (f_1, f_2, \dots, f_n)$. Let partial symmetry profiles of the component functions $f_1, f_2, \dots, f_n$ be denoted by $S_1, S_2, \dots, S_n$, respectively (see Definition 8).

Theorem 2. *A necessary condition for an $n \times n$ function $f(x_1, x_2, \dots, x_n) = (f_1, f_2, \dots, f_n)$ to be reversible is as follows: intersection of all profiles $S_1, S_2, \dots, S_n$, has to be equal to the collection of results each of which has no more than one element.*

Proof. Let us assume that two variables x_i and x_j belong to one subset being an element of the intersection of the profiles $S_1, S_2, \dots, S_n$, i.e. appear in one subset in all these profiles. It is equivalent to the equation:

$$f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n).$$

However, because any reversible function f is a bijective mapping then $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n)$ differs from $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n)$. $\square$

Thus, the following theorem holds.

Theorem 3. *$n \times n$ reversible Boolean functions, $n > 1$, with all totally symmetric CFs being non-degenerate do not exist.*

On the other hand, component functions of a reversible function can be totally or partially symmetric if at least two of them are partially symmetric.

Example 9. *It is easy to show that the following function is reversible*

$$f_1 = x_1 \oplus x_2 \oplus x_3, \quad f_2 = x_1 \oplus x_2, \quad f_3 = x_1 \oplus x_3,$$

where f_1 is a totally symmetric function and both f_2 and f_3 are partially symmetric functions. The simple generalization of the above reversible function to the case of any n can be defined as follows:

$$f_1 = \bigoplus_{i=1}^n x_i, \quad f_k = \bigoplus_{i \neq k} x_i, k \in \{2, \dots, n\},$$

where the symbol $\bigoplus$ denotes summing modulo 2. ■

In some papers, algorithms for synthesis of reversible circuits for (totally) symmetric functions are considered. However, symmetric functions in these papers are first embedded in reversible specifications with additional inputs and/or outputs.

Now let us consider linear and affine CFs. For any n there is only one non-degenerate linear Boolean function:

$$x_1 \oplus x_2 \oplus \dots \oplus x_n.$$

Hence, by Theorem 1 the following result is true:

Theorem 4. *For $n > 1$ $n \times n$ reversible Boolean functions with all linear or affine CFs being non-degenerate do not exist.*

However, reversible Boolean functions having as CFs one of non-degenerate linear (affine) functions and the other functions depending essentially on $k < n$ variables do exist as is shown in Example 9.

Let us consider the following property of monotone Boolean functions.

Lemma 3. *Every monotone Boolean function which is balanced, except projection functions P -equivalent to the identity, cannot be equal to 1 for an assignment with weight 1 (i.e. with only one non-zero entry).*

Proof. Assume that the lemma is not true. Then there exists a balanced monotone Boolean function f , not being a projection function, and an assignment $\mathbf{a} = (a_1, a_2, \dots, a_n)$ with weight 1 for which $f(\mathbf{a})$ is not equal to zero. Without loss of generality let $\mathbf{a} = (1, 0, \dots, 0, 0)$, i.e. $x_1 = 1$, $x_i = 0$ for $2 \leq i \leq n$. Because f is monotone so $f(\mathbf{b}) = 1$ for all assignments $\mathbf{b} = (b_1, b_2, \dots, b_n)$ with $b_1 = 1$. The number of such assignments is equal to 2^{n-1} . As the number of all binary assignments is 2^n hence the number of assignments $\mathbf{c}$ not compatible with assignments $\mathbf{b}$, i.e. having $c_1 = 0$, is equal to $2^n - 2^{n-1} = 2^{n-1}$. A balanced Boolean function takes values 0 and 1 the same number of times so $f = 0$ for all those assignments with $c_1 = 0$. Thus,

$$\begin{aligned} f(\mathbf{a}) &= 0 & \text{for all binary assignments} & \quad \mathbf{a} = (0, a_2, \dots, a_{n-1}, a_n), \\ f(\mathbf{a}) &= 1 & \text{for all binary assignments} & \quad \mathbf{a} = (1, a_2, \dots, a_{n-1}, a_n), \end{aligned}$$

i.e., f is a projection function what is in contradiction with the initial assumption. $\square$

Theorem 5. *An $n \times n$ reversible Boolean function, $n \geq 3$, with all component functions being non-degenerate monotone does not exist.*

Proof. Lemma 3 states that for any monotone balanced Boolean function F and any input assignment $\mathbf{a}$ with weight 1

$$F(\mathbf{a}) = 0.$$

Thus, for any $n \times n$ reversible Boolean function G , $n \geq 3$, with all CFs being monotone and any input assignment $\mathbf{a}$ with weight 1

$$G(0, 0, \dots, 0, 0, 1) = G(0, 0, \dots, 0, 1, 0) = (0, 0, \dots, 0, 0, 0),$$

what contradicts the reversibility constraint as G takes value $(0, 0, \dots, 0, 0, 0)$ more than once. Thus, any $n \times n$ Boolean function, $n \geq 3$, with all component functions being monotone, is not reversible.

By Definition 3, Definition 4, Lemma 1, Theorem 1 and Theorem 5 the following result holds. $\square$

Corollary 2. *An $n \times n$ reversible Boolean function, $n \geq 3$, with all component functions being non-degenerate and threshold, does not exist.*

5 Results Based on Newly Constructed Functions

Let us first introduce simple notions related to Positive Polarity Reed-Muller expressions for Boolean functions. The number of literals in a term will be called its rank. Denote by $T_{i,j}$ the exclusive-or sum of all terms having a rank not smaller than i and not greater than j ($T_{i,i}$ will denote all terms with rank i).

In [13] we introduced the following $n \times n$ reversible function, for arbitrary n , which will be called Negation with Preservation of Constants (in short NPC $n \times n$):

Definition 15. The reversible function $NPCn*n(x_1, x_2, \dots, x_n)$, $n \geq 3$, is defined in such a manner that its component functions $NPCn$ are defined as follows:

$$\begin{array}{llll} f_1 & = x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_{n-2} \oplus x_{n-1} & \oplus T_{2,n-1}, \\ f_2 & = x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_{n-2} \oplus & x_n \oplus T_{2,n-1}, \\ \dots & \dots & \dots \\ f_{n-1} & = x_1 & \oplus x_3 \oplus \cdots \oplus x_{n-2} \oplus x_{n-1} \oplus x_n \oplus T_{2,n-1}, \\ f_n & = & x_2 \oplus x_3 \oplus \cdots \oplus x_{n-2} \oplus x_{n-1} \oplus x_n \oplus T_{2,n-1}, \end{array}$$

i.e., in each of the above equations exactly one variable is missing, namely in the i th equation variable x_{n-i+1} is missing.

The formulas in Definition 15 can be transformed taking into account that

$$1. \quad x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_{n-2} \oplus x_{n-1} \oplus x_n \oplus T_{2,n} = T_{1,n} = \\ = x_1 \vee x_2 \vee x_3 \vee \cdots \vee x_{n-1} \vee x_n,$$

(this transformation can be easily proved by induction starting from the well-known formula for $n = 2$: $x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1 x_2$),

$$\begin{aligned}
2. \quad f_i &= f_i \oplus x_{n-i+1} \oplus x_{n-i+1} \oplus x_1 x_2 \dots x_n \oplus x_1 x_2 \dots x_n = T_{1,n} \oplus x_{n-i+1} \oplus x_1 x_2 \dots x_n \\
&= (x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1} \vee x_n) \oplus x_{n-i+1} \oplus x_1 x_2 \dots x_n,
\end{aligned}$$

where we applied the following three obvious formulas:

$$f_i = f_i \oplus 0, \quad x_{n-i+1} \oplus x_{n-i+1} = 0, \quad x_1 x_2 \dots x_n \oplus x_1 x_2 \dots x_n = 0,$$

and the transformation used in case 1:

$$\begin{aligned} f_i \oplus x_{n-i+1} \oplus x_1 x_2 \dots x_n &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_{n-2} \oplus x_{n-1} \oplus x_n \oplus T_{2,n-1} \oplus x_1, x_2 \dots x_n \\ &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_{n-2} \oplus x_{n-1} \oplus x_n \oplus T_{2,n} = T_{1,n}. \end{aligned}$$

Using the above formulas we will show by example how the values of the function f_i can be calculated. Without loss of generality we will show this for f_1 :

Step 1. Calculate $f_1^{(1)} = x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1} \vee x_n$ (see Table 1).

Step 2. Calculate $f_1^{(2)} = (x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1} \vee x_n) \oplus x_1$ (by negating the lower half of the truth table obtained in Step 1).

Step 3. Calculate $f_1 = (x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1} \vee x_n) \oplus x_1 \oplus x_1 x_2 \dots x_n$ (by negating the output value in the last row of the truth table obtained in Step 2). These steps are performed for $n = 3$ in Table 1.

Thus, f_1 has the well-known property of preserving constants:

$$f_1(0, 0, 0) = 0 \quad \text{and} \quad f_1(1, 1, 1) = 1,$$

as well as is negating the input x_1 for all other vectors of input values. Similarly (see Table 2), the reversible function NPC3*3 is preserving constants:

$$\text{NPC3*3}(0, 0, 0) = (0, 0, 0) \quad \text{and} \quad \text{NPC3*3}(1, 1, 1) = (1, 1, 1),$$

as well as negating all the other input vectors. This is why we gave this reversible function the name Negation with Preservation of Constants.

Table 1. Establishing values of the 3-variable function f_1 in three steps

$x_1 x_2 x_3$	$f_1^{(1)}$	$f_1^{(2)}$	f_1
0 0 0	0	0	0
0 0 1	1	1	1
0 1 0	1	1	1
0 1 1	1	1	1
1 0 0	1	0	0
1 0 1	1	0	0
1 1 0	1	0	0
1 1 1	1	0	1

By analogy with the above example it is easy to show that the following two results hold for any n :

Lemma 4. *Each function NPCn*n is reversible.*

Lemma 5. *Each component function of NPCn*n can be obtained from the Boolean function NPCn as a result of a permutation of its variables.*

Table 2. Truth table for the function NPC3*3

$x_1x_2x_3$	$f_1f_2f_3$
0 0 0	0 0 0
0 0 1	1 1 0
0 1 0	1 0 1
0 1 1	1 0 0
1 0 0	0 1 1
1 0 1	0 1 0
1 1 0	0 0 1
1 1 1	1 1 1

Theorem 6. All component functions of NPCn*n, $n \geq 3$, are (1) nonlinear, (2) self-dual, (3) self-complementary, (4) P-equivalent, and (5) unate.

Proof. (1) The function NPCn is nonlinear because its PPRM contains terms of rank 2 for any $n > 2$.
 (2) Without loss of generality we write $\text{NPCn}(x_1, x_2, \dots, x_n) = (x_1 \vee x_2 \vee \dots \vee x_n) \oplus x_1 \oplus x_1x_2 \dots x_n$. On the other hand, by De Morgan's laws the following two formulas hold:

$$\begin{aligned}(x'_1 \vee x'_2 \vee \dots \vee x'_n) &= (x_1x_2 \dots x_n)' = 1 \oplus x_1x_2 \dots x_n, \\ x'_1x'_2 \dots x'_n &= (x_1 \vee x_2 \vee \dots \vee x_n)' = 1 \oplus (x_1 \vee x_2 \vee \dots \vee x_n).\end{aligned}$$

Thus,

$$\begin{aligned}(\text{NPCn})'(x'_1, x'_2, \dots, x'_n) &= 1 \oplus [(x'_1 \vee x'_2 \vee \dots \vee x'_n) \oplus x'_1 \oplus x'_1x'_2 \dots x'_n] \\ &= 1 \oplus [(x_1x_2 \dots x_n)' \oplus x'_1 \oplus (x_1 \vee x_2 \vee \dots \vee x_n)'] \\ &= 1 \oplus [1 \oplus x_1x_2 \dots x_n \oplus 1 \oplus x_1 \oplus 1 \oplus (x_1 \vee x_2 \vee \dots \vee x_n)] \\ &= (x_1 \vee x_2 \vee \dots \vee x_n) \oplus x_1 \oplus x_1x_2 \dots x_n \\ &= \text{NPCn}(x_1, x_2, \dots, x_n).\end{aligned}$$

and by Definition 12 any NPCn is self-dual.

- (3) From Lemma 2 it follows that it is self-complementary.
 (4) P-equivalence follows from Lemma 5.
 (5) Once again, without loss of generality we can write

$$\text{NPCn}(x_1, x_2, \dots, x_n) = [(x_1 \vee x_2 \vee \dots \vee x_n) \oplus x_1] \oplus x_1x_2 \dots x_n,$$

and transform it using well-known formulas $a \oplus b = ab' + a'b$, $aa' = 0$, $a = a \vee ab$, and De Morgan's laws:

$$\begin{aligned}\text{NPCn}(x_1, x_2, \dots, x_n) &= [(x_1 \vee x_2 \vee \dots \vee x_n)x'_1 \vee (x_1 \vee x_2 \vee \dots \vee x_n)'x_1] \oplus x_1x_2 \dots x_n \\ &= [x'_1x_2 \vee x'_1x_3 \vee \dots \vee x'_1x_n \vee (x'_1x'_2 \dots x'_n)x_1] \oplus x_1x_2 \dots x_n \\ &= (x'_1x_2 \vee x'_1x_3 \vee \dots \vee x'_1x_n)(x_1x_2 \dots x_n)' \vee (x'_1x'_2 \vee x'_1x'_3 \vee \dots \vee x'_1x'_n)'(x_1x_2 \dots x_n) \\ &= x'_1x_2 \vee x'_1x_3 \vee \dots \vee x'_1x_n \vee x_2 \dots x_n.\end{aligned}$$

Thus, in the reduced SOP for NPC_n the variable x_1 appears only as complemented and all the other variables are uncomplemented, i.e. NPC_n is unate. $\square$

Corollary 3. *For any $n \geq 3$ there exist reversible functions having all component functions being:*

- (1) *nonlinear,*
- (2) *self-dual,*
- (3) *self-complementary,*
- (4) *P-equivalent,*
- (5) *unate.*

6 Computational Results

By running simple programs on a laptop we have obtained the computational results described in this section. The configuration of the laptop we used was standard: i7 processor and 4 GB of RAM. Each of the computational tasks took less than one hour. First we calculated in an exhaustive manner all NPN-equivalence classes of balanced Boolean functions of 1, 2, 3 and 4 variables. The results for $n = 1, 2, 3$ are shown in Table 3 and for $n = 4$ in Table 4 together with sizes and functional properties of all these classes. These results were published for the first time in [13]. Each row gives one equivalence class identified by its representative expressed in the form of PPRM expressions. For our purpose considering each component function separately is a more convenient form than permutation which is shorter but in which component functions are not shown explicitly. For each class, the table shows the number of variables (n), the name of the class (Class), the size of the equivalence class (Size), a *Representative* of the class, and the classical *Properties* the class possesses (the meanings of abbreviations L, LV, NL, SC and SD were introduced in Sect. 2). Equivalence classes are sorted first by the size of the number of terms in PPRM expression and in case of a tie by the sizes of the consecutive terms in the expression (the terms of the same size are given in the lexicographic order). To decrease the width of Tables 4 and 5 we used names a, b, c and d to denote variables (instead of x_1, x_2, x_3, x_4 which we use in the rest of the paper).

We have checked that only for the following 18 out of 58 classes of balanced Boolean functions up to 4 variables (B1.1-B4.52) it is *impossible* to find four functions belonging to the same class which would constitute a 4-variable reversible function: B2.1, B3.2, B4.2, B4.3, B4.4, B4.7 (this class includes only 2 functions), B4.13, B4.15, B4.27, B4.28, B.4.31, B4.33, B4.34, B4.35, B4.38, B4.42, B4.48, B4.51.

We used this result for extrapolation of some properties for a larger number of variables. We also expect that several interesting conjectures can be formulated on the basis of the above results.

We have also calculated all NPNP-equivalence classes of 3-variable reversible functions (see Table 5 organised under the same assumptions as Tables 3 and 4).

Table 3. NPN-equivalence classes of n -variable balanced Boolean functions for $n \leq 3$

Class	Size	Representative	Properties				
			L	LV	NL	SC	SD
<i>B1.1</i>	8	a	+	+		+	+
<i>B2.1</i>	12	$a \oplus b$	+	+		+	
<i>B3.1</i>	96	$a \oplus bc$		+	+	+	
<i>B3.2</i>	8	$a \oplus b \oplus c$	+	+		+	+
<i>B3.3</i>	96	$a \oplus ab \oplus bc$			+	+	
<i>B3.4</i>	32	$ab \oplus ac \oplus bc$			+	+	+

For the synthesis of reversible functions, NPNP-equivalence classes are interesting because permutations of component functions do not change values of cost functions of optimal reversible circuits implementing them. It is because a permutation of component functions leads to permutation of lines in the circuit which does not change the cost of the circuit.

As mentioned in Sect. 3 such a table was published in [19] but we were able to find (probably typographic) errors in it. One type of these errors consists in non-reversibility of two classes' representatives. To show precisely where the errors are located let us point that Lorens' Table VI is split into three parts based on properties of the inverses of the classes' representatives:

- (A) 21 functions having their inverses identical to the function (called self-inverse functions),
- (B) 3 classes of functions having their inverses in the same NPNP-equivalence class,
- (C) 28 classes of functions having their inverses in a different NPNP-equivalence class.

It is easy to check that the following two classes' representatives from the Lorens' table are not reversible:

$$f_1 = x'_1 \oplus x_2 x_3, f_2 = x_2 \oplus x_1 x'_3, f_3 = x_3 \oplus x'_1 x_2 \text{ (Part A, row 16),}$$

$$f_1 = x_1 x_2 \oplus x_2 x_3 \oplus x_3 x_1, f_2 = x_1 \oplus x_2 x'_3, f_3 = x_3 \oplus x_2 x'_1 \text{ (Part C, column 1, row 13).}$$

It seems that the correct expressions were supposed to be as follows:

$$f_1 = x'_1 \oplus x_2 x_3, f_2 = x_2 \oplus x_1 x'_3, f_3 = x_3 \oplus x'_1 x'_2 \text{ (adding a "prime" to the last literal),}$$

$$f_1 = x_1 x_2 \oplus x_2 x_3 \oplus x_3 x_1, f_2 = x_1 \oplus x'_2 x_3, f_3 = x_3 \oplus x_2 x'_1 \text{ (swapping the "prime" in the 2nd term in } f_2 \text{).}$$

The last two functions are reversible and belong to our classes R28 and R31, respectively, which are not covered by the other representatives in Table VI in [19].

In Lorens' Table VI we have also found two pairs of representatives that belong to the same class:

Table 4. NPN-equivalence classes of n -variable balanced Boolean functions for $n = 4$

Class	Size	Representative	Properties				
			L	LV	NL	SC	SD
B4.1	64	$a \oplus bcd$		+	+	+	
B4.2	48	$a \oplus b \oplus cd$		+	+	+	
B4.3	192	$a \oplus b \oplus acd$		+	+	+	
B4.4	96	$a \oplus bc \oplus bd$		+	+	+	
B4.5	768	$a \oplus bc \oplus abd$			+		
B4.6	192	$a \oplus abc \oplus bcd$			+	+	
B4.7	2	$a \oplus b \oplus c \oplus d$	+	+		+	
B4.8	192	$a \oplus b \oplus c \oplus abd$		+	+	+	
B4.9	96	$a \oplus b \oplus ac \oplus cd$		+	+	+	
B4.10	384	$a \oplus b \oplus cd \oplus abc$			+		
B4.11	384	$a \oplus b \oplus abc \oplus acd$			+	+	
B4.12	96	$a \oplus ab \oplus bc \oplus bd$			+	+	
B4.13	32	$a \oplus bc \oplus bd \oplus cd$		+	+	+	
B4.14	384	$a \oplus bc \oplus abc \oplus abd$			+	+	
B1.15	64	$a \oplus b \oplus c \oplus d \oplus abc$		+	+	+	
B4.16	192	$a \oplus b \oplus c \oplus abc \oplus abd$			+	+	
B4.17	32	$a \oplus b \oplus ac \oplus ad \oplus cd$		+	+	+	+
B4.18	384	$a \oplus b \oplus ab \oplus ac \oplus bcd$			+	+	
B4.19	384	$a \oplus b \oplus ac \oplus ad \oplus bcd$			+		
B4.20	384	$a \oplus b \oplus ac \oplus acd \oplus bcd$			+		
B4.21	384	$a \oplus b \oplus ac \oplus abd \oplus bcd$			+		
B4.22	384	$a \oplus b \oplus abc \oplus acd \oplus bcd$			+		
B4.23	48	$a \oplus ab \oplus ac \oplus bd \oplus cd$			+	+	+
B4.24	384	$a \oplus ab \oplus bc \oplus bd \oplus acd$			+		
B4.25	384	$a \oplus ab \oplus bc \oplus abd \oplus acd$			+		
B4.26	192	$a \oplus ab \oplus cd \oplus abc \oplus acd$			+	+	
B4.27	384	$a \oplus bc \oplus bd \oplus abd \oplus acd$			+	+	
B4.28	192	$a \oplus bc \oplus bd \oplus acd \oplus bcd$			+		
B4.29	192	$a \oplus ab \oplus abc \oplus abd \oplus bcd$			+	+	
B4.30	384	$a \oplus bc \oplus abc \oplus abd \oplus acd$			+		
B4.31	48	$ab \oplus ac \oplus ad \oplus bc \oplus bd$			+	+	
B4.32	384	$a \oplus b \oplus c \oplus ab \oplus ad \oplus bcd$			+	+	
B4.33	192	$a \oplus b \oplus c \oplus ad \oplus abc \oplus bcd$			+		
B4.34	384	$a \oplus b \oplus c \oplus ad \oplus abd \oplus bcd$			+	+	
B4.35	24	$a \oplus b \oplus ab \oplus ac \oplus bd \oplus cd$			+	+	
B4.36	384	$a \oplus b \oplus ac \oplus cd \oplus abc \oplus abd$			+	+	
B4.37	192	$a \oplus b \oplus ac \oplus cd \oplus abd \oplus acd$			+	+	
B4.38	192	$a \oplus b \oplus ab \oplus abc \oplus abd \oplus acd$			+	+	
B4.39	768	$a \oplus b \oplus ac \oplus abd \oplus acd \oplus bcd$			+		
B4.40	96	$a \oplus b \oplus abc \oplus abd \oplus acd \oplus bcd$			+		
B4.41	96	$a \oplus ab \oplus ac \oplus bc \oplus bd \oplus cd$			+	+	
B4.42	192	$a \oplus ab \oplus bc \oplus bd \oplus acd \oplus bcd$			+	+	
B4.43	384	$a \oplus ab \oplus bc \oplus cd \oplus abd \oplus acd$			+	+	
B4.44	384	$a \oplus ab \oplus cd \oplus abc \oplus abd \oplus acd$			+		
B4.45	384	$a \oplus bc \oplus bd \oplus abc \oplus acd \oplus bcd$			+	+	
B4.46	64	$a \oplus b \oplus c \oplus ad \oplus abd \oplus acd \oplus bcd$			+	+	
B4.47	384	$a \oplus b \oplus ac \oplus cd \oplus abc \oplus abd \oplus acd$			+	+	
B4.48	192	$a \oplus b \oplus ab \oplus abc \oplus abd \oplus acd \oplus bcd$			+		
B4.49	384	$a \oplus b \oplus ac \oplus abc \oplus abd \oplus acd \oplus bcd$			+	+	
B4.50	64	$a \oplus ab \oplus cd \oplus abc \oplus abd \oplus acd \oplus bcd$			+	+	+
B4.51	64	$a \oplus b \oplus c \oplus d \oplus ab \oplus abc \oplus abd \oplus acd$			+	+	
B4.52	64	$a \oplus b \oplus c \oplus ab \oplus cd \oplus abc \oplus abd \oplus acd \oplus bcd$			+	+	+

Table 5. Representatives of NPNP-equivalence classes of reversible Boolean functions for $n = 3$

Class	Size	$f_1(a, b, c)$	$f_2(a, b, c)$	$f_3(a, b, c)$	BF classes
$R1$	48	a	b	c	1.1 1.1 1.1
$R2$	288	a	b	$a \oplus c$	1.1 1.1 2.1
$R3$	576	a	b	$c \oplus ab$	1.1 1.1 3.1
$R4$	144	a	b	$a \oplus b \oplus c$	1.1 1.1 3.2
$R5$	144	a	$a \oplus b$	$a \oplus c$	1.1 2.1 2.1
$R6$	288	a	$a \oplus b$	$b \oplus c$	1.1 2.1 2.1
$R7$	1152	a	$a \oplus b$	$c \oplus ab$	1.1 2.1 3.1
$R8$	288	a	$a \oplus b$	$a \oplus b \oplus c$	1.1 2.1 3.2
$R9$	576	a	$b \oplus c$	$b \oplus ab \oplus ac$	1.1 2.1 3.3
$R10$	1152	a	$b \oplus ac$	$b \oplus c \oplus ab$	1.1 3.1 3.1
$R11$	576	a	$b \oplus ac$	$b \oplus c \oplus ac$	1.1 3.1 3.1
$R12$	2304	a	$b \oplus ac$	$c \oplus ab \oplus ac$	1.1 3.1 3.3
$R13$	576	a	$b \oplus ac$	$a \oplus b \oplus c \oplus ac$	1.1 3.2 3.1
$R14$	576	a	$a \oplus b \oplus c$	$b \oplus ab \oplus ac$	1.1 3.2 3.3
$R15$	288	a	$b \oplus ab \oplus ac$	$c \oplus ab \oplus ac$	1.1 3.3 3.3
$R16$	288	a	$b \oplus ab \oplus ac$	$a \oplus c \oplus ab \oplus ac$	1.1 3.3 3.3
$R17$	144	$a \oplus b$	$a \oplus c$	$a \oplus b \oplus c$	2.1 2.1 3.2
$R18$	576	$a \oplus b$	$a \oplus c$	$ab \oplus ac \oplus bc$	2.1 2.1 3.4
$R19$	576	$a \oplus b$	$c \oplus ab$	$a \oplus c \oplus ab$	2.1 3.1 3.1
$R20$	1152	$a \oplus b$	$c \oplus ab$	$a \oplus ac \oplus bc$	2.1 3.1 3.3
$R21$	1152	$a \oplus b$	$c \oplus ab$	$a \oplus c \oplus ab \oplus ac \oplus bc$	2.1 3.1 3.4
$R22$	576	$a \oplus b$	$a \oplus b \oplus c$	$a \oplus ac \oplus bc$	2.1 3.2 3.3
$R23$	288	$a \oplus b$	$a \oplus ac \oplus bc$	$a \oplus c \oplus ac \oplus bc$	2.1 3.3 3.3
$R24$	288	$a \oplus b$	$a \oplus ac \oplus bc$	$b \oplus c \oplus ac \oplus bc$	2.1 3.3 3.3
$R25$	1152	$a \oplus b$	$a \oplus ac \oplus bc$	$a \oplus c \oplus ab \oplus ac \oplus bc$	2.1 3.3 3.4
$R26$	576	$a \oplus b$	$ab \oplus ac \oplus bc$	$a \oplus c \oplus ab \oplus ac \oplus bc$	2.1 3.4 3.4
$R27$	2304	$a \oplus bc$	$a \oplus b \oplus ac$	$c \oplus ab \oplus ac$	3.1 3.1 3.3
$R28$	384	$a \oplus bc$	$a \oplus b \oplus ac$	$a \oplus b \oplus c \oplus ab$	3.1 3.1 3.1
$R29$	1152	$a \oplus bc$	$a \oplus b \oplus ac$	$a \oplus b \oplus c \oplus ac$	3.1 3.1 3.1
$R30$	1152	$a \oplus bc$	$a \oplus b \oplus ac$	$b \oplus c \oplus ac \oplus bc$	3.1 3.1 3.3
$R31$	1152	$a \oplus bc$	$a \oplus b \oplus ac$	$a \oplus c \oplus ab \oplus ac \oplus bc$	3.1 3.1 3.4
$R32$	576	$a \oplus bc$	$a \oplus b \oplus bc$	$a \oplus c \oplus bc$	3.1 3.1 3.1
$R33$	1152	$a \oplus bc$	$a \oplus b \oplus bc$	$c \oplus ab \oplus bc$	3.1 3.1 3.3
$R34$	576	$a \oplus bc$	$b \oplus ab \oplus ac$	$c \oplus ab \oplus ac$	3.1 3.3 3.3
$R35$	2304	$a \oplus bc$	$b \oplus ab \oplus ac$	$c \oplus ab \oplus bc$	3.1 3.3 3.3
$R36$	576	$a \oplus bc$	$b \oplus ab \oplus ac$	$a \oplus b \oplus c \oplus bc$	3.1 3.3 3.1
$R37$	1152	$a \oplus bc$	$b \oplus ab \oplus ac$	$a \oplus c \oplus ab \oplus ac$	3.1 3.3 3.3
$R38$	1152	$a \oplus bc$	$b \oplus ac \oplus bc$	$b \oplus c \oplus ab \oplus bc$	3.1 3.3 3.3
$R39$	1152	$a \oplus bc$	$b \oplus ac \oplus bc$	$b \oplus c \oplus ac \oplus bc$	3.1 3.3 3.3
$R40$	2304	$a \oplus bc$	$b \oplus ac \oplus bc$	$a \oplus c \oplus ab \oplus ac \oplus bc$	3.1 3.3 3.4
$R41$	576	$a \oplus bc$	$a \oplus b \oplus c \oplus bc$	$a \oplus b \oplus ab \oplus ac \oplus bc$	3.1 3.1 3.4
$R42$	576	$a \oplus bc$	$a \oplus b \oplus ab \oplus ac \oplus bc$	$a \oplus c \oplus ab \oplus ac \oplus bc$	3.1 3.4 3.4
$R43$	1152	$a \oplus b \oplus c$	$a \oplus ab \oplus bc$	$b \oplus ac \oplus bc$	3.2 3.3 3.3
$R44$	288	$a \oplus b \oplus c$	$a \oplus ab \oplus bc$	$c \oplus ab \oplus bc$	3.2 3.3 3.3
$R45$	288	$a \oplus b \oplus c$	$a \oplus ab \oplus bc$	$a \oplus b \oplus ab \oplus bc$	3.2 3.3 3.3
$R46$	384	$a \oplus ab \oplus bc$	$b \oplus ac \oplus bc$	$c \oplus ab \oplus ac$	3.3 3.3 3.3
$R47$	1152	$a \oplus ab \oplus bc$	$b \oplus ac \oplus bc$	$a \oplus c \oplus ac \oplus bc$	3.3 3.3 3.3
$R48$	1152	$a \oplus ab \oplus bc$	$b \oplus ac \oplus bc$	$b \oplus c \oplus ab \oplus ac \oplus bc$	3.3 3.3 3.4
$R49$	576	$a \oplus ab \oplus bc$	$c \oplus ab \oplus bc$	$a \oplus b \oplus ab \oplus bc$	3.3 3.3 3.3
$R50$	576	$a \oplus ab \oplus bc$	$c \oplus ab \oplus bc$	$a \oplus b \oplus ab \oplus ac \oplus bc$	3.3 3.3 3.4
$R51$	576	$a \oplus ab \oplus bc$	$a \oplus b \oplus ab \oplus ac \oplus bc$	$b \oplus c \oplus ab \oplus ac \oplus bc$	3.3 3.4 3.4
$R52$	192	$ab \oplus ac \oplus bc$	$a \oplus b \oplus ab \oplus ac \oplus bc$	$a \oplus c \oplus ab \oplus ac \oplus bc$	3.4 3.4 3.4

- one pair is in Part A, rows 13 and 14 (both representatives belong to our class R40),
- the other pair: Part C, column 1, row 10, and Part C, column 2, row 11 (both representatives belong to our class R45).

Thus, 4 out of 52 NPNP-equivalence classes of 3*3 reversible functions are not represented in Lorens' Table VI.

In Table 4, in comparison with [19], we added sizes of the classes and information showing to which NPN-equivalence class of balanced functions each of the component functions belongs. The latter information was useful in our extrapolation of some properties of reversible functions [16] (for an example see Sect. 7).

7 Extrapolation Based on Cycle Structures

In [11, 12] it has been demonstrated that it is possible to extrapolate some properties of reversible functions by considering their cycle structures. This is why we tried to exploit the same approach to discover infinite sequences of reversible functions with all their component functions being non-degenerate and belonging to different P-classes. We established that there are 26 NPNP-classes of 3-variable functions (R27-R52) that possess all component functions depending essentially on all three variables. Among them, there is only one class that consists of reversible functions all whose component functions belong to different NPN-classes. Below the PPRM expressions for a member of this class are shown:

NPNP-class R40

$$\begin{aligned} A &= a \oplus c \oplus ab \oplus ac \oplus bc, \\ B &= b \oplus ab \oplus ac, \\ C &= c \oplus ab. \end{aligned}$$

The above PPRM expressions show some regular features. However, our experience is so that extrapolation of such features of PPRMs is very difficult because: (1) usually a component function is obtained which is not balanced, (2) even if all PPRMs correspond to balanced functions then their collection does not constitute a reversible function. Therefore we have decided to apply extrapolation based on cycle structures. By considering the appropriate mappings $\{0, 1\}^3 \rightarrow \{0, 1\}^3$ it is easy to establish that the earlier defined member of the NPNP-class R40 has the following cycle structure:

$$< 000 > < 010 > < 011 > < 100 > < 001, 101, 111, 110 > .$$

Let us note that binary n -tuples in the unique cycle having more than one element form a regular pattern:

001,
101,
111,
110.

Namely, it is easy to note that

- the first and the second n -tuples differ only in the 1st bit position,
- the second and the third n -tuples differ only in the 2nd bit position,
- the third and the fourth n -tuples differ only in the 3rd bit position.

Thus, we observe here a certain periodicity which can be easily extrapolated leading to the desired infinite sequence of reversible functions as will be seen later. In this case, extrapolating was quite simple. Let us introduce additional notions.

Definition 16. *A set of variable assignments over $\{0, 1\}$ with specified numbers of p 0s and r 1s is called a block and denoted by $b_{p,r}$.*

Example 10. *The set of all eight variable assignments for 3-variable Boolean functions can be partitioned into the following four blocks:*

$$b_{3,0} = \{000\}, \quad b_{2,1} = \{001, 010, 100\}, \quad b_{1,2} = \{011, 101, 110\}, \quad b_{0,3} = \{111\}.$$

Definition 17. *For any Boolean function f let $B^0(f)$ and $B^1(f)$ denote the sets of blocks including all variable assignments for which f is equal 0 and 1, respectively.*

Example 11. *Let us consider the following Boolean projection functions:*

$$f(x_1, x_2, x_3) = x_1, \quad g(x_1, x_2, x_3) = x_2, \quad h(x_1, x_2, x_3) = x_3.$$

Then,

$$\begin{aligned} B^0(f) &= \{\{000\}, \{001, 010\}, \{011\}\}, & B^1(f) &= \{\{100\}, \{101, 110\}, \{111\}\}, \\ B^0(g) &= \{\{000\}, \{001, 100\}, \{101\}\}, & B^1(g) &= \{\{010\}, \{011, 110\}, \{111\}\}, \\ B^0(h) &= \{\{000\}, \{010, 100\}, \{110\}\}, & B^1(h) &= \{\{001\}, \{011, 101\}, \{111\}\}. \end{aligned}$$

Notice that for each 3-variable Boolean reversible function k the union of $B^0(k)$ and $B^1(k)$ is equal to the set of all 8 Boolean variable assignments. For each of the component functions of an arbitrary reversible function cardinalities of unions of their B^i sets are the same.

Example 12. Let us consider a 3-variable Boolean reversible function $F(x_1, x_2, x_3) = (f_1, f_2, f_3)$ defined in such a manner that the only non-identical mappings of variable assignments in F are as follows

$$\begin{aligned} 001 &\rightarrow 101, \\ 101 &\rightarrow 111, \\ 111 &\rightarrow 110, \\ 110 &\rightarrow 001. \end{aligned}$$

When we consider the reversible function F as a permutation of output assignments it is a single cycle of four elements:

$$< 001, 101, 111, 110 > .$$

Notice that in the above mappings

- in the 1st row the leftmost bit is being negated,
- in the 2nd row the second bit is being negated
- in the 3rd row the third bit is being negated
- in the 4th row all bits are being negated.

This observation will be generalised later to functions of any number of variables.

Now let us note what changes have been done in the sets B_i , $0 \leq i \leq 1$, for functions f_1 , f_2 , and f_3 , in comparison with the sets for the function in Example 11 (the assignments moved to another block are shown **bolded** and underlined):

$$\begin{aligned} B^0(f_1) &= \{\{000\}, \{010\}, \{011, \underline{\mathbf{110}}\}\}, & B^1(f_1) &= \{\{\underline{\mathbf{001}}, 100\}, \{101\}, \{111\}\}, \\ B^0(f_2) &= \{\{000\}, \{001, 100\}, \{\underline{\mathbf{110}}\}\}, & B^1(f_2) &= \{\{010\}, \{011, \underline{\mathbf{101}}\}, \{111\}\}, \\ B^0(f_3) &= \{\{000\}, \{010, 100\}, \{\underline{\mathbf{111}}\}\}, & B^1(f_3) &= \{\{001\}, \{011, 101\}, \{\underline{\mathbf{110}}\}\}. \end{aligned}$$

Let us summarise the above observations.

The values of the function f_1 differ from the values of the projection function x_1 only for the assignments 001 and 110. Namely, we can notice that

$$\begin{aligned} f(0, 0, 1) &= 0 & f_1(0, 0, 1) &= 1, \\ f(1, 1, 0) &= 1 & f_1(1, 1, 0) &= 0. \end{aligned}$$

As a result, the function f_1 can be obtained from the projection function x_1 by swapping its values for variable assignments 001 and 110.

Values of each of the other two component functions, f_2 and f_3 , also differ from the values of the corresponding projection functions only for two assignments.

Swaps for f_2 in comparison with the projection function x_2 are as follows:

$$\begin{array}{ll} g(1, 0, 1) = 0 & f_2(1, 0, 1) = 1, \\ g(1, 1, 0) = 1 & f_2(1, 1, 0) = 0. \end{array}$$

Swaps for f_3 in comparison with the projection function x_3 are as follows:

$$\begin{array}{ll} h(1, 1, 1) = 1 & f_3(1, 1, 1) = 0, \\ h(1, 1, 0) = 0 & f_3(1, 1, 0) = 1. \end{array}$$

Let us show that component functions f_1 and f_2 belong to different P -equivalence classes. Assume that f_1 and f_2 belong to the same P -equivalence class. Then, since any permutation over the variable set $\{x_1, x_2, x_3\}$ does not change the assignment 111 there should be $f_1(1, 1, 1) = f_2(1, 1, 1)$, however, $f_1(1, 1, 1) = 0$ and $f_2(1, 1, 1) = 1$. It is in contradiction with our assumption that f_1 and f_2 belong to the same P -equivalence class. Thus, f_1 and f_2 belong to different P -equivalence classes.

In a similar manner it can be shown that the other two pairs of component functions of F , (f_1, f_3) and (f_2, f_3) , belong to different P -equivalence classes.

Let us show that component functions f_1 and f_3 belong to different P -equivalence classes. Assume that f_1 and f_3 belong to the same P -equivalence class. Then, since any permutation over variable set $\{x_1, x_2, x_3\}$ does not change the assignment 111 there should be $f_1(1, 1, 1) = f_3(1, 1, 1)$, however $f_1(1, 1, 1) = 0$ and $f_3(1, 1, 1) = 1$. It is in contradiction with our assumption that f_1 and f_3 belong to the same P -equivalence class. Thus, f_1 and f_3 belong to different P -equivalence classes.

Let us show that component functions f_2 and f_3 belong to different P -equivalence classes. Assume that f_2 and f_3 belong to the same P -equivalence class. Then, let us consider the permutation of variables consisting in swapping variables x_2 and x_3 . Then there should be $f_2(1, 0, 0) = f_3(1, 0, 0)$, however $f_2(1, 0, 0) = 0$ and $f_3(1, 0, 0) = 1$. It is in contradiction with our assumption that f_2 and f_3 belong to the same P -equivalence class. Thus, f_2 and f_3 belong to different P -equivalence classes. ■

Now the presented above methodology of proving that two component functions of F belong to different P -equivalence classes will be extended to Boolean reversible functions of any number of variables. To prove that Boolean reversible functions with all component functions belonging to different P -equivalence classes exist for any number of variables $n \geq 3$, we will define the following infinite sequence of reversible functions:

Definition 18. The reversible Boolean function $H^n(x_1, x_2, \dots, x_n) = (f_1, f_2, \dots, f_n)$, $n \geq 3$, is defined in such a manner that the only non-identical mappings of variable assignments in H^n are as follows:

- totally symmetric,
 - linear/affine,
 - monotone,
 - majority,
 - threshold.
- (B) For any $n \geq 3$ there exists a Boolean reversible function with all component functions being nondegenerate and
- nonlinear,
 - self-complementary,
 - self-dual,
 - unate.
 - P-equivalent.
- (C) For any $n \geq 3$ there exists a Boolean reversible function with all component functions being non-degenerate and belonging to different P-equivalence classes.

Our work has not been finished. We plan to continue efforts for constructing a classification of reversible Boolean functions which would be useful in the synthesis of reversible circuits.

Acknowledgements. The authors acknowledge partial support of COST Action IC1405 on “Reversible Computation - Extending Horizons of Computing”. They are grateful to Philipp Niemann, one of the reviewers, for many comprehensive remarks.

References

1. Aaronson, S., Grier, D., Schaeffer, L.: The classification of reversible bit operations. Preprint [arXiv:1504.05155](https://arxiv.org/abs/1504.05155) [quant-ph] (2015)
2. Carić, M., Živković, M.: On the number of equivalence classes of invertible Boolean functions under action of permutation of variables on domain and range. *Publications de l’Institut Mathématique* **100**(114), 95–99 (2016)
3. Carlet, C.: Vectorial Boolean functions for cryptography. In: Crama, Y., Hammer, P. (eds.) *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pp. 398–472. Cambridge University Press, Cambridge (2010)
4. De Vos, A.: *Reversible Computing: Fundamentals, Quantum Computing, and Applications*. Wiley-VCH Verlag, Weinheim (2010)
5. Debnath, D., Sasao, T.: Fast Boolean matching under variable permutation using representative. In: *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 359–362 (1999)
6. Debnath, D., Sasao, T.: Efficient computation of canonical form for Boolean matching in large libraries. In: *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 591–596 (2004)
7. Debnath, D., Sasao, T.: Fast Boolean matching under permutation by efficient computation of canonical form. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E87–A**(12), 3134–3140 (2004)
8. Debnath, D., Sasao, T.: Efficient computation of canonical form under variable permutation and negation for Boolean matching in large libraries. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E89–A**(12), 3443–3450 (2006)

9. Draper, T.G.: Nonlinear complexity of Boolean permutations. Ph.D. thesis, Department of Mathematics, University of Maryland, College Park, Maryland, USA (2009)
10. Harrison, M.A.: The number of classes of invertible Boolean functions. *J. ACM* **10**, 25–28 (1963)
11. Jegier, J., Kerntopf, P.: Progress towards constructing sequences of benchmarks for quantum Boolean circuits synthesis. In: Proceedings of the 14th IEEE International Conference on Nanotechnology, pp. 250–255 (2014)
12. Jegier, J., Kerntopf, P., Szyprowski, M.: An approach to constructing reversible multi-qubit benchmarks with provably minimal implementations. In: Proceedings of the 13th IEEE International Conference on Nanotechnology, pp. 99–104 (2013)
13. Kerntopf, P., Moraga, C., Podlaski, K., Stanković, R.: Towards classification of reversible functions. In: Steinbach, B. (ed.) Proceedings of the 12th International Workshop on Boolean Problems, pp. 21–28 (2018)
14. Kerntopf, P., Moraga, C., Podlaski, K., Stanković, R.: Towards classification of reversible functions with homogeneous component functions. In: Steinbach, B. (ed.) Further Improvements in the Boolean Domain, pp. 386–406. Cambridge Scholars Publishing, Newcastle upon Tyne (2018)
15. Kerntopf, P., Podlaski, K., Moraga, C., Stanković, R.: Study of reversible ternary functions with homogeneous component functions. In: Proceedings of the 47th IEEE International Conference on Multiple-Valued Logic, pp. 191–196 (2017)
16. Kerntopf, P., Podlaski, K., Moraga, C., Stanković, R.: New results on reversible Boolean functions having component functions with specified properties. In: Drechsler, R., Soeken, M. (eds.) Advanced Boolean Techniques, pp. 217–236. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20323-8_10
17. Kerntopf, P., Stanković, R., Podlaski, K., Moraga, C.: Ternary/MV reversible functions with component functions from different equivalence classes. In: Proceedings of the 48th IEEE International Conference on Multiple-Valued Logic, pp. 109–114 (2018)
18. Lorens, C.S.: Invertible Boolean functions. Tech. rep. 21, Space-General Corp., El Monte, CA, Research Memorandum (1962)
19. Lorens, C.S.: Invertible Boolean functions. *IEEE Trans. Electron. Comput.* **EC-13**(5), 529–541 (1964)
20. Primenko, E.A.: Invertible Boolean functions and fundamental groups of transformations of algebras of Boolean functions. *Avtomatika & Vychislitel'naya Tekhnika* **3**, 17–21 (1976)
21. Primenko, E.A.: On the number of types of invertible Boolean functions. *Avtomatika & Vychislitel'naya Tekhnika* **6**, 12–14 (1977)
22. Primenko, E.A.: On the number of types of invertible transformations in multivalued logic. *Kibernetika* **5**, 27–29 (1977)
23. Primenko, E.A.: Equivalence classes of invertible Boolean functions. *Kibernetika* **6**, 1–5 (1984)
24. Rice, J.E.: Considerations for determining a classification scheme for reversible Boolean functions. Tech. rep. TR-CSJR2-2007, Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, Canada (2007)
25. Saeedi, M., Markov, I.L.: Synthesis and optimization of reversible circuits: a survey. *ACM Comput. Surv.* **45**(2), 21:1–21:34 (2013)
26. Soeken, M., Abdessaied, N., De Micheli, G.: Enumeration of reversible functions and its application to circuit complexity. In: Devitt, S., Lanese, I. (eds.) RC 2016. LNCS, vol. 9720, pp. 255–270. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40578-0_19

27. Soeken, M., Wille, R., Keszocze, O., Miller, D.M., Drechsler, R.: Embedding of large Boolean functions for reversible logic. *ACM J. Emerg. Technol. Comput. Syst.* **12**(4), 41:1–41:26 (2015)
28. Stanković, R.S., Astola, J.T., Steinbach, B.: Former and recent work in classification of switching functions. In: Steinbach, B. (ed.) *Proceedings of the 8th International Workshop on Boolean Problems*, pp. 115–126 (2008)
29. Strazdins, I.E.: On the number of types of invertible binary networks. *Avtomatika & Vychislitel'naya Tekhnika* **1**, 30–34 (1974)
30. Tokareva, N.: *Bent Functions, Results and Applications to Cryptography*. Academic Press, London (2015)
31. Tsai, C.C., Marek-Sadowska, M.: Boolean functions classification via fixed polarity Reed-Muller forms. *IEEE Trans. Comput.* **46**(2), 173–186 (1997)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.