UDC 519.7

R. Stanković (Dept. of Computer Science, Serbia)
J. Astola (Dept. of Signal Processing, Finland)
C. Moraga (European Centre for Soft Computing, Spain)

REMARKS ON GENERALIZATIONS OF REED-MULLER EXPRESSIONS FOR BINARY AND MULTIPLE-VALUED FUNCTIONS

This paper discusses extensions and generalizations of Reed-Muller expressions for binary-valued logic functions. Arithmetic expressions are viewed as extensions derived by the change of the range of function values from the Galois field $GF(2)$ which is usually assumed for binary logic functions to the field or rational numbers. Generalizations are concerned with the change of the domain allowing application of these expressions to multiple-valued logic functions and change of the range to define word-level expressions for these functions. The considerations are focused on functional expressions preserving properties of Reed-Muller expressions viewed as counterparts of polynomial (Taylor series) expressions and Fourier series in classical mathematical analysis.

В цій статті отримані деякі розширення та узагальнення представлень Ріда-Малера для двозначних логічних функцій. Арифметичні представлення розглядаються як розширення, що одержуються зміною області значень функції з поля Галуа $GF(2)$, що зазвичай використовується для представлення функцій двозначної логіки, до поля раціональних чисел. Узагальнення дозволяють завдяки розширенню області визначення застосовувати ті самі представлення до функцій багатозначної логіки, а завдяки розширенню області значень, виводити також багато-розрядні представлення для таких функцій. Запропонований підхід сконцентрований на отриманні функціонального представлення із збереженням властивостей представлень Ріда-Малера, що розглядаються як альтернатива поліноміального представлення (ряди Тейлора) і представлення за допомогою рядів Фур'є в класичному математичному аналізі.

Introduction. Discrete functions are usually defined as a mapping

$$f : \times_{i=1}^{n} S_i \to L,$$

where S_i, $i = 1, \ldots, n$ and L are finite non-empty sets of not necessarily equal cardinalities $|S_i|$ and $|L|$, respectively, and $\times$ denotes the direct (Cartesian) product of sets.

Binary-valued (switching) and multiple-valued logic functions are two classes of discrete functions of particular interest in this paper. In these cases, $S_i = L = \{0, 1\}$ and $S_i = L = \{0, 1, \ldots, p-1\}$, for all i, for binary and multiple-valued (p-valued) functions. Cases $p = 3$ and $p = 4$ are most often encountered in practice for practical reasons. Ternary functions ($p = 3$) are most compact in the sense of the number of data which can be encoded with ternary sequences of the given length, while quaternary functions are convenient due to simple encoding of four values by binary sequences and then implementation by two-stable state circuits.

Since discrete functions are mappings between finite sets, the simplest way to specify a discrete function f is to enumerate its values at all the points of the domain of f. This can be done in a tabular form or as a vector of function values, or in some similar way. However, this method cannot be used for functions of a large number of variables, i.e., defined in many points. For that reason, various analytical representations as Sum-of-product or Product-of-sums expressions have been investigated already from the time of the work by De Morgan in 1874 [10].

This subject was of a continuous interest in the past, and nowadays it has a renewed importance due to demands coming from features that are present in contemporary or can be expected in future technologies for realization of digital system. We provide few former and recent references illustrating different attempts to define various functional expressions for discrete functions [2–6, 8, 11, 18, 19, 21, 22, 24–27, 30, 31, 33, 34, 39–41].

Reed-Muller expressions are particular functional expressions for binary valued functions which can be viewed as a discrete analogue of either Taylor series or Fourier series for functions on the real line $\mathcal{R}$. They are defined in terms of a particular set of basis functions defined as elementary products of binary variables. These are products of all possible combinations of n binary-valued variables. For a given function f, no identical products can appear in the Reed-Muller expression for f.

In this paper, we discuss extensions and generalizations of Reed-Muller expressions defined by preserving the same set of basis functions for the binary case or its straightforward generalizations for the multiple-valued case. We attempt to provide explanations of differences and motivation to introduce few different expressions. Various other expressions which can be related to the Reed-Muller expressions in other ways are out of the scope of this paper, however, related references for initial reading on this subject are provided in closing remarks.

1. Reed-Muller expressions. The elementary product of binary variables $x_1, \ldots, x_n$ in Hadamard ordering can be determined as entries of the matrix

$$\mathbf{X}(n) = \bigotimes_{i=1}^{n} \mathbf{X}_i(1), \quad \mathbf{X}_i = \begin{bmatrix} 1 & x_i \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product.

Any binary-valued function $f(x_1, \ldots, x_n)$ can be represented as

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{2^n-1} r_i s(i), \quad \text{calculations in } GF(2),$$

where $s(i)$ are entries of $\mathbf{X}(n)$ and $r_i \in \{0, 1\}$.

In matrix notation, coefficients r_i, written as entries of a vector $\mathbf{R}_f(n) = [r_0, r_1, \ldots, r_{2^n}]^T$ are determined as

$$\mathbf{R}_f(n) = \mathbf{R}(n)\mathbf{F},$$

where

$$\mathbf{R}(n) = \left(\bigotimes_{i=1}^{n} \mathbf{R}_i(1) \right), \quad \mathbf{R}_i(1) \;\; = \;\; \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

and $\mathbf{F} = [f(0), f(1), \ldots, f(2^n - 1)]^T$ is the function vector of f.

Formally, $\mathbf{R}_i(1)$ is the inverse of the numerical version of $\mathbf{X}_i(1)$, which is $\mathbf{X}_i(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$; but in $GF(2)$ this matrix is its own inverse; therefore no explicit inversion is needed.

Table 1 presents basic properties of the Reed-Muller expressions in terms of operations in Boolean algebra and Boolean ring. As noticed in [20], resemblance

Table 1.
Properties of the Reed-Muller transform in terms of Boolean operations.

$h(x) = f(x) \oplus g(x)$	$S_h(w) = S_f(w) \oplus S_g(w)$
$h(x) = f(x) \vee g(x)$	$S_h(w) = \oplus_{u \vee v = w} S_f(u) S_g(v)$
$h(x) = f(x) \wedge g(x)$	$S_h(w) = S_f(w) \oplus S_g(w) \oplus_{u \vee v = w} S_f(u) S_g(v)$,
Convolution theorem	
$S_h(w) = S_f(w) \vee S_g(w)$	$h(x) = \oplus_{y \vee z = x} f(y) g(z)$

to the properties of the Fourier transform on $\mathcal{R}$ is stronger if the Gibbs algebra is assumed as the underlying algebraic structure for study the Reed-Muller expressions.

Definition 1 (Gibbs algebra [20])**.** *The set R of n-tuples $(x_1, \ldots, x_n)$ of elements from $\{0, 1\}$ with pointwise addition modulo* 2

$$(f \oplus g)(x) = f(x) \oplus g(x),$$

where $x = (x_1, \ldots, x_n)$, is a linear space isomorphic to a subspace of the dyadic field $\mathcal{F}$.

The multiplication in $\mathcal{F}$ induces a multiplication in R defined by

$$\begin{aligned} (fg)(0) &= 0, \\ (fg)(x) &= \sum_{s=0}^{\sigma(x)-1} f(\sigma(x) - 1 - s) g(x), \quad x \neq 0, \end{aligned}$$

where $\sigma(x) = \sum_{i=1}^{n} x_i 2^{n-i}$.

Table 2 presents basic properties of the Reed-Muller expressions in terms of operations in the Gibbs algebra [20].

The optimization of the Reed-Muller expressions is usually considered as a reduction of the number of non-zero coefficients, which can be achieved by selecting between the positive x_i and negative $\overline{x}_i = 1 \oplus x_i$ literals for each variable. In this way, 2^n different Fixed-polarity Reed-Muller can be assigned to a given function f. The expansion with the minimum number of non-zero coefficients is usually selected. Since there can be few expressions with the same number of coefficients, the expansion with the smaller number of literals per products is selected.

An alternative (spectral) interpretation of Fixed-polarity Reed-Muller expressions can be given as follows [7].

The negative literal can be viewed as permutation of the values 0 and 1 a binary variable can take into 1 and 0. This results in a particular permutation of the set of basis functions in terms of which the decomposition of a given function f is preformed. This permutation is equivalent to a permutation of elements of the function vector, and for some permutations, the basis functions (entries of function vector) are distributed in a manner which reduces the number of coefficients.

Table 2.
Properties of the Reed-Muller transform in terms of operations in the Gibbs algebra.

Self-inverseness $S_{S_f}(w) = f$
Kronecker property $W(x) \equiv 1, \quad \forall x \in \{0, 1, \ldots, 2^n - 1\}$ $S_{W^q}(w) = \delta(q, w)$, δ - Kronecker delta
Translation formula $S_{W^q f}(w) = \begin{cases} S_f(w-q), & w > q, \\ 0, & w \leq q. \end{cases}$
Parseval relation $\langle f, g \rangle = \sum_{x=0}^{2^n-1} f(x)\overline{g}(x)$ $\overline{g}(x)$ is the dyadic conjugate defined as $\overline{\mathbf{g}} = \mathbf{R}^T \mathbf{g}$, where $\mathbf{g}$ is the function vector of g. $\langle f, g \rangle = \sum_{w=0}^{2^n-1} S_f(w) S_g(w)$
Convolution theorem $S_{f \cdot g}(w) = S_f(w) \cdot S_g(w)$

Arithmetic expressions. When representing multi-output functions, a separate Reed-Muller expression is required for each output. Alternatively, k-outputs of a multiple-output function can be viewed as binary encoding of integers which can be represented by k bits. In this way, a multiple-output function is identified with an integer function which can be represented by the arithmetic expressions defined as integer counterpart of the Reed-Muller expressions [9, 23, 27–29]. This means, we keep the same set of basis functions as determined by the primary products of binary variables, or columns of the Reed-Muller matrix $\mathbf{R}(n)$, however, with function values interpreted as integers 0 and 1 instead the logic values. This matrix we denote by $\mathbf{A}^{-1}(n)$. We take the inverse of it over the field of rational numbers Q, as the arithmetic transform matrix $\mathbf{A}(n)$ which is used to define coefficients in the arithmetic expressions. Since coefficients in arithmetic expressions are integers, which means computer words are required to represent them, these expressions belong to the broad class of various word-level functional expressions for binary-valued functions.

Definition 2. (*Arithmeticexpressions*). *Every function of n binary-valued variables taking values in the set of integers Z can be represented as*

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{2^n-1} a_i s(i), \quad \textit{calculations in } Q,$$

where $s(i)$ are entries of $\mathbf{X}_a(n)$ and $a_i \in Q$, with

$$\mathbf{X}_a(n) = \bigotimes_{i=1}^{n} \mathbf{X}_{a,i}(1), \quad \mathbf{X}_{a,i} = \left[\begin{array}{cc} 1 & x_i \end{array}\right],$$

with x_i taking the values of integers 0 and 1.

The coefficients in arithmetic expressions written as a vector $\mathbf{A}_f(n) = [a_0, a_1, \ldots, a_{2^n}]^T$ are calculated in matrix notation as

$$\mathbf{A}_f(n) = \mathbf{A}(n)\mathbf{F},$$

where

$$\mathbf{A}(n) = \left(\bigotimes_{i=1}^{n} \mathbf{A}_i(1)\right), \quad \mathbf{A}_i(1) \;\; = \;\; \left[\begin{array}{cc} 1 & 0 \\ -1 & 1 \end{array}\right] = (\mathbf{X}_{a,i}(1))^{-1}.$$

Since the arithmetic expressions are defined with respect to the same set of basis functions as the Reed-Muller expressions, the optimization of arithmetic expressions is performed by selecting polarities of variables in the same way as in the Reed-Muller expressions. In this way, Fixed-polarity arithmetic expressions are defined [27, 28].

Generalizations to Multiple-Valued Functions. An approach towards generalizations of the Reed-Muller expressions to multiple-valued functions $f : \{0, 1, \ldots, p-1\}^n \rightarrow \{0, 1, \ldots, p-1\}$ is to use the basis functions which are a direct generalization of basis functions defined by $\mathbf{X}(n)$ and $\mathbf{X}_a(n)$. In the case of the Reed-Muller and the arithmetic expressions, to each binary variable x_i a matrix $\mathbf{X}(1) = \left[\begin{array}{cc} x_i^0 & x_i^1 \end{array}\right]$ is assigned. By an analogy, to each p-valued variable $x_i \in \{0, 1, \ldots, p-1\}$ we assign a matrix

$$\mathbf{X}_{p,i}(1) = \left[\begin{array}{cccc} x_i^0 & x_i^1 & \ldots & x_i^{p-1} \end{array}\right],$$

where the exponentiation is defined as a purposely introduced operation or derived from the multiplication in the underlying algebraic structure assumed for the considered class of p-valued functions. The Reed-Muller-Fourier expressions and Galois field expressions are examples of functional expressions derived in these two different approaches but by using the same principle to define the set of basis functions.

The set of basis functions is defined by elementary products of integer powers of variables, which in matrix notation can be expressed as the Kronecker product of matrices $\mathbf{X}_{p,i}(1)$ in the same way as in the case of Reed-Muller expressions for binary functions

$$\mathbf{X}_p(n) = \bigotimes_{i=1}^{n} \mathbf{X}_{p,i}(1).$$

The coefficients in these expressions are determined by using the matrix inverse to the matrix whose columns are the basis functions determined by $\mathbf{X}_p(n)$.

Galois field expressions. This generalization of Reed-Muller expressions will be illustrated by the example of functions taking their values in $GF(3)$ and $GF(4)$.

Table 3 defines the addition and multiplication in $GF(3)$, which are actually addition and multiplication modulo 3.

Table 3.

Addition and multiplication in $GF(3)$.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Each n-variable three-valued function can be represented as a polynomial of the form

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{3^n-1} a_i g(i), \tag{1}$$

where a_i, $i \in \{0, 1, 2, 3\}$, $g(i)$ are the product terms defined in the Hadamard order as elements of the vector $\mathbf{X}_{3GF}(n)$ defined by

$$\mathbf{X}_{3GF}(n) = \bigotimes_{i=1}^{n} \mathbf{X}_{3GF,i}(1), \quad \mathbf{X}_{3GF,i}(1) = \begin{bmatrix} 1 & x_i^1 & x_i^2 \end{bmatrix}$$

and addition and multiplication are carried out in $GF(3)$, i.e., modulo 3.

Therefore, when written explicitly, the set of basic functions for $n = 1$ is given by columns of the matrix

$$\mathbf{X}_{3GF,i}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}.$$

In matrix notation, for a function f specified by the function vector $\mathbf{F} = [f(0), \ldots, f(3^n - 1)]^T$, the coefficients g_i in the Galois field expression are calculated as

$$\mathbf{G}_f = \mathbf{G}_{3GF}(n)\mathbf{F},$$

where

$$\mathbf{G}_{3GF}(n) = \bigotimes_{i=0}^{n} \mathbf{G}_{3GF,i}(1), \quad \mathbf{G}_{3GF,i}(1) = (\mathbf{X}_{3GF}(1))^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \quad \text{in } GF(3).$$

Example 1. *For $n = 2$, the Galois field transform matrix for $GF(3)$ is defined*

Table 4.

Addition and multiplication in $GF(4)$.

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

as $\mathbf{G}_{3GF}(2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} =$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Table 4 shows addition and multiplication in $GF(4)$.

Each n-variable four-valued function can be represented as a polynomial of the form

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{4^n-1} a_i g(i), \tag{2}$$

where a_i, $i \in \{0, 1, 2, 3\}$, $g(i)$ are the product terms defined in the Hadamard order as elements of the vector $\mathbf{X}_{4GF}(n)$ defined by

$$\mathbf{X}_{4GF}(n) = \bigotimes_{i=1}^{n} \mathbf{X}_{4GF,i}(1), \quad \mathbf{X}_{4GF,i}(1) = \left[\ 1 \quad x_i^1 \quad x_i^2 \quad x_i^3\ \right]$$

and addition and multiplication are carried out in $GF(4)$.

When written explicitly, for $n = 1$,

$$\mathbf{X}_{4GF}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 \end{bmatrix}.$$

The inverse of it over $GF(4)$ is

$$\mathbf{G}_{4GF}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

This matrix is used as the basic matrix to determine by Kronecker product the transform matrix used to calculate coefficients in GF expressions for $p = 4$.

As obvious from Example 1, the GF-transform matrix does not preserve the triangular structure of the Reed-Muller and arithmetic transform matrices for binary-valued functions. A consequence is that in optimization by selecting different polarities for variables we are restricted to p out of $p!$ permutations of values a variable can take. These permutations are defined as $\overset{p-}{x} = x \oplus i$, $i = 1, 2, \ldots, p-1$. All other permutations do not change the number of non-zero coefficients, which follows from the structure of the GF-transform matrix. This disadvantage in reduction of the number possible different functional expansions is overcome in the Reed-Muller-Fourier transform discussed below.

Arithmetic Expressions for Multiple-Valued Functions derived from GF-expressions. Extensions of GF-expressions for multiple-valued functions to the corresponding word-level expressions can be done in two different ways. In the first approach, the form of basis functions is preserved, i.e., basis functions are defined as products of integer powers of variables x_i^k, $k \in \{0, 1, \ldots, p-1\}$, where exponentiation is derived from multiplication in the set of integers.

The following example illustrates the definition of arithmetic expressions for multiple-valued functions by the example of ternary functions.

Example 2. *For ternary functions the basis matrix used in definition of the arithmetic expressions is defined as*

$$\mathbf{X}_3(1) = \begin{bmatrix} x^0 & x^1 & x^2 \end{bmatrix},$$

or in explicit form as

$$\mathbf{X}_3(1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}.$$

The arithmetic transform matrix used to calculate the coefficients in the arithmetic expression is

$$\mathbf{A}_3(1) = \begin{bmatrix} 2 & 0 & 0 \\ -3 & 4 & -1 \\ 1 & -2 & 1 \end{bmatrix}.$$

A problem with this approach towards definition of the arithmetic transform is that values which take basis functions are large, especially for a large value of p, since the exponent of x^{p-1} is taken. This can be overcome if we keep the same set of basis functions as in GF-expressions and interpret their values as integers instead of values in $GF(p)$. Then, we will have the same set of basis functions as in GF-expressions, however, under different interpretation of function values, the coefficients will be integers which when scaled by a normalization factor can be used to represent multi-output functions in multiple-valued variables.

Example 3. *If the basis functions* $\mathbf{X}_{3GF}(1)$*, are interpreted as functions taking the corresponding integer values, the matrix inverse over the filed of rational numbers* Q *defines the basic ternary arithmetic transform. This matrix is given by*

$$\mathbf{A}_3(1) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -2 & 2 \\ -2 & 4 & -2 \end{bmatrix}.$$

The optimization of these both classes of arithmetic expressions can be done by selecting polarities for variables in the same way as in Galois field expressions for binary and multiple-valued case. The restrictions to use p complements of variables remain valid also in this case due to the structure of the transform matrix.

Reed-Muller-Fourier Expressions. A disadvantage of GF-expressions is that except for $GF(2)$, the matrix expressing basis functions, and consequently the matrix used to calculate the coefficients in the expressions, is not triangular. As noticed above, this restricts the possible permutations which can be used in the corresponding fixed-polarity expressions. Thus, that approach to generalizations of Reed-Muller expressions to multiple-valued logic functions has this feature as a limitation in optimization of the expressions by selecting polarities of variables. At the same time, the structure of the transform matrices reflect to the properties of the related expressions in the same ways as discussed in the binary case (see Table 1 and Table 2) and their resemblance to the properties of the classical Fourier transform. That was a motivation for generalizations of the Reed-Muller expressions derived by their interpretation presented in [20]. In this way, the Reed-Muller-Fourier expressions have been defined.

The Gibbs algebra for binary functions can be generalized to p-valued functions in a straightforward manner [33]. The function W playing the role of exponential functions in Fourier analysis is defined as

$$W(x) \equiv p-1, \forall x \in \{0,1,\ldots,p^n-1\},$$

where $x = (x_1,\ldots,x_n)$, $x_i \in \{0,1,\ldots,p-1\}$, and

$$\sigma(x) = \sum_{i=1}^{n} x_i p^{n-1}.$$

The addition is taken as componentwise addition modulo p,

$$(f \oplus g)(x) = f(x) \oplus g(x), \quad \bmod p,$$

and the multiplication is defined in the same way as in the case of binary-valued functions,

$$\begin{aligned} (fg)(0) &= 0, \\ (fg)(x) &= \sum_{s=0}^{\sigma(x)-1} f(\sigma(x)-1-s)g(x), \quad x \neq 0. \end{aligned}$$

Table 5.

Addition and multiplication modulo 3.

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Table 6.

The Gibbs exponentiation $3EXP$.

$*$	0	1	2
0	2	0	0
1	2	1	0
2	2	2	2

The set $S = \{W^1, \ldots, W^{p^n}\}$ is the basis in terms of which the Reed-Muller-Fourier expressions are defined as

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{p^n-1} r_i W^{i+1}, \quad \bmod p.$$

In order to express the Reed-Muller-Fourier expressions in terms of p-valued variables, it is necessary to formally define the exponentiation as an operation derived from multiplication defined above. The definitions will be illustrated by the example of $p = 3$ and $p = 4$.

To define the Reed-Muller-Fourier-expressions for ternary functions, we introduce the following notation and definitions [34, 36, 37].

Table 5 defines addition and multiplication modulo 3, and Table 6 defines a new operation of exponentiation $3EXP$, based on the Gibbs multiplication defined above, and therefore called the Gibbs exponentiation.

In this way, the basis functions are defined as

$$\mathbf{X}_{3RMF}(n) = \bigotimes_{i=1}^{n} \mathbf{X}_{3RMF,i}(1),$$

where

$$\mathbf{X}_{3RMF,i}(1) = \left[\begin{array}{ccc} x_i^{*0} & x_i^{*1} & x_i^{*2} \end{array}\right] = \left[\begin{array}{ccc} 2 & x_i^{*1} & x_i^{*2} \end{array}\right].$$

The Reed-Muller-Fourier expressions for ternary functions are defined in terms of this set of basis functions as

$$f(x_1, \ldots, x_n) = (-1)^n \mathbf{X}_{3RMF}(n)\mathbf{R}_f(n)$$

where

$$\mathbf{R}_f(n) = \mathbf{R}(n)\mathbf{F} = \left(\bigotimes_{i=1}^{n} \mathbf{R}(1)\right)\mathbf{F},$$

Table 7.

Addition and multiplication modulo 4.

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	2
3	3	0	1	2

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	2
2	0	2	0	2
3	0	3	2	1

where

$$\mathbf{R}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and calculations are performed modulo 3.

Example 4. *For $n = 2$, the Reed-Muller-Fourier transform matrix is*

$$\begin{aligned} \mathbf{R}(2) &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

For an illustration, consider the RMF-expression of a function f defined by the function vector $\mathbf{F} = [1, 1, 2, 2, 0, 1, 2, 1, 0]^T$, the RMF-coefficients are given by the vector $\mathbf{R}_f = [1, 0, 1, 2, 1, 1, 2, 0, 1]^T$, and the RMF-expression is

$$f = 1 \oplus 2x_2^{*2} \oplus x_1^{*1} \oplus x_1^{*1}x_2^{*1} \oplus x_1^{*1}x_2^{*2} \oplus x_1^{*2} \oplus x_1^{*2}x_2^{*2}.$$

The definition of RMF-expressions can be uniformly extended to functions for p non-prime. This will be illustrated for the example of $p = 4$.

Tables 7 and 8 define the operations of addition and multiplication modulo 4 and the Gibbs exponentiation $4EXP$.

The Reed-Muller-Fourier expressions for quaternary functions are defined in terms of this set of basis functions as

$$f(x_1, \ldots, x_n) = (-1)^n \mathbf{X}_{4RMF}(n)\mathbf{R}_f(n)$$

where

$$\mathbf{R}_f(n) = \mathbf{R}(n)\mathbf{F} = \left(\bigotimes_{i=1}^{n} \mathbf{R}_i(1) \right) \mathbf{F},$$

Table 8.

The Gibbs exponentiation $4EXP$.

$*$	0	1	2	3
0	3	0	0	0
1	3	1	0	0
2	3	2	3	0
3	3	3	1	1

Table 9.

Number of non-zero coefficients required to represent various two-variable quaternary functions.

Function	zero-polarity		min-polarity	
	GF	RMF	GF	RMF
$x_1 \oplus x_2$	14	7	14	7
$\overline{x}_1\overline{x}_2$ GF	14	3	14	3
$\overline{x}_1\overline{x}_2$ mod 4	12	6	13	6
$x_1 + x_2$	12	7	12	7
$max\{x_1, x_2\}$	13	7	13	7
$min\{x_1, x_2\}$	13	6	13	6
$max\{\overline{x}_1 x_2\}$	13	7	14	10
$min\{\overline{x}_1 x_2\}$	14	8	14	9

$\overline{x}$ defined by the rule $x \oplus \overline{x} = 0$.

where

$$\mathbf{R}_i(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 3 \end{bmatrix}$$

and calculations are performed modulo 4.

Notice that both the matrices defining basis functions and transform matrices used to calculate the coefficients are triangular matrices with upper right part consisting of zero elements. Due to that, it is possible to exploit all $p!$ permutations of a p-valued variable as its complements. In this way, the number of different expressions for a function of n variables is extended from p^n into $(p!)^n$, which increases possibilities to determine expressions with reduced number of non-zero coefficients compared to Galois field expressions. At the same time, all properties corresponding to properties of the Fourier representations, as presented in Table 2 for the binary case, are preserved.

To illustrate the impact of application of an extended set of $(p!)$ logic complements, we present the following examples comparing the number of non-zero coefficients in GF and RMF-expressions [37].

Table 9 compares the number of non-zero coefficients in GF and RMF-expressions for some functions, which are often met in practice and were used as examples also in [38].

Table 10 compares the number of non-zero coefficients in GF and RMF expressions in some benchmark functions for binary-valued functions viewed as quaternary functions after suitable modifications usually used in this area [37].

Table 10.

Number of non-zero coefficients in GF and RMF-expressions for benchmark functions.

	GF	RMF
alu4-3	9696	6301
alu4-4	8609	6393
alu4-5	8515	6070
alu4-6	3117	1553
alu4-7	9308	6668
alu4-8	9266	6596
rd84-1	36	130
rd84-2	8	32
rd84-3	81	1
rd84-4	150	131
sao2-1	350	340
sao2-2	338	646
sao2-3	494	699
sao2-4	510	694
av	3605.57	2589.57

Table 11 shows the average number of nonzero-coefficients in GF and RMF-expressions over 20 sets of 1000 randomly generated quaternary functions in each set. In this table, r is the ratio of the number of coefficients. Considered are functions of two and three variables [37].

Table 11.

Number of non-zero coefficients for randomly generated functions, $p = 4$.

n	GF	RMF	r
2	12.547	7.367	41%
3	44.129	35.032	20%

A detailed analysis and comparison of efficiency of various expressions including GF and RMF expressions is presented in [1].

Arithmetic counterpart of RMF-expressions. To extend applicability of RMF-expressions to integer valued functions, the arithmetic RMF-expressions are defined by using the same basis functions, however, with their values $\{0, 1, \ldots, p-1\}$ interpreted as integers. Then, we calculate the inverse of the matrix $\mathbf{X}_{pRMF}(n)$ over the field of rational numbers Q. This matrix cannot be represented by the Kronecker product, however, presses a recursive structure as will be illustrated below by the example for functions in ternary variables [35, 37].

Definition 3. *The Arithmetic RMF-transform matrix for ternary functions is* *defined as*

$$\mathbf{A}_3(n) = \begin{bmatrix} \mathbf{A}_3(n-1) & \mathbf{O}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{B}_3(n-1) & -\mathbf{B}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{C}_3(n-1) & \mathbf{B}_3(n-1) & \mathbf{A}_3(n-1) \end{bmatrix},$$

where $\mathbf{O}_3(n-1)$ *is the* $(3^{n-1} \times 3^{n-1})$ *zero matrix, and matrices* $\mathbf{B}_3(n-1)$ *and* $\mathbf{C}_3(n-1)$ *also express the same recursive structure,*

$$\mathbf{B}_3(n) = \begin{bmatrix} \mathbf{B}_3(n-1) & \mathbf{O}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{C}_3(n-1) & -\mathbf{C}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{A}_3(n-1) & \mathbf{C}_3(n-1) & \mathbf{B}_3(n-1) \end{bmatrix},$$

and

$$\mathbf{C}_3(n) = \begin{bmatrix} \mathbf{C}_3(n-1) & \mathbf{O}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{A}_3(n-1) & -\mathbf{A}_3(n-1) & \mathbf{O}_3(n-1) \\ \mathbf{B}_3(n-1) & \mathbf{A}_3(n-1) & \mathbf{A}_C(n-1) \end{bmatrix},$$

with $\mathbf{A}_3(0) = 2$, $\mathbf{B}_3(0) = -4$, $\mathbf{C}_3(0) = 2$. *If holds,* $\mathbf{A}_3(n) + \mathbf{B}_3(n) + \mathbf{C}_3(n) = \mathbf{O}_3(n)$. *Thus, for* $n = 1$, *it is*

$$\mathbf{A}_3(1) = \begin{bmatrix} 2 & 0 & 0 \\ -4 & 4 & 0 \\ 2 & -4 & 2 \end{bmatrix}.$$

The matrix $\mathbf{A}_3(n)$ is used to calculate coefficients in the arithmetic RMF-expressions defined as

$$f(x_1, \ldots, x_n) = 2a_0 + \sum_{i=1}^{3^n-1} a_i s(i),$$

where $s(i)$ are columns of the matrix $\mathbf{X}_{3RMF}(n)$, with entries interpreted as integers. In other words, $s(i)$ are $3AND$ product of ternary variables to the integer powers in terms of $3EXP$ in Hadamard ordering.

Closing Remarks. This paper presents a review of definitions of extensions and generalizations of Reed-Muller expressions for binary-valued logic functions. The term extensions refers to the definition of world level expressions used to represent integer valued functions in terms of binary-valued variables. These expressions are known as arithmetic expressions and can be used to efficiently represent multi-output binary logic functions.

The term generalizations refers to definitions of the corresponding expressions for multiple-valued logic functions. In this case, bit-level (multiple-valued bits) and word-level expressions can be also defined.

In this paper, we focus on the Reed-Muller expressions and their extensions and generalizations derived by interpretation of logic AND and EXOR as operations in Galois field $GF(2)$ and the expressions itself as polynomial (Taylor series-like) or spectral (Fourier series-like) expressions. Thus, we consider functional expressions which can be written in terms of variables in functions to be represented and preserving at the same time properties resembling these of Fourier representations.

Word-level expressions for multiple-valued functions which we considered are defined in two ways

1) By assigning to each p-valued variable a polynomial of order p in terms of exponentiation as an operation in the field of rational numbers Q, and using these polynomials to define the set of basis functions for n-variable functions,

2) By preserving the same set of basis functions as in modulo p structures, however, with interpretation of function values in the set of integers, to determine coefficients in the expressions considered.

Various other generalizations in terms of combinations of either modular operations, or *min* and *max* operations, and literal operators, are not discussed. In a way, the expressions based on *min* and *max* operations (as a generalization of logic AND and OR), as well as literal operators, can be rather viewed as generalizations of Sum-of-Product (SOP) expressions and related representations for two-valued (Boolean) logic functions, than as a generalization of polynomial or spectral representations. For these representations we refer to [11, 12, 26].

We also did not discuss representations where the main intention was to reduce the number of non-zero coefficients while preserving fast calculation algorithms, however, at the price of discharging other properties usually expressed by spectral (Fourier series-like) expressions. For such generalizations of Reed-Muller expressions we refer, for instance, to [13–18].

1. *Adams, K.* Optimization of Multiple-Valued Polynomials by Polarities and Affine Transforms. PhD Thesis, University of Ulster, Northern Ireland, UK, March 2007.
2. *Aizenberg I.N., Aizenberg N.N., Vandewalle J.* Multi-Valued and Universal Binary Neurons - Theory, Learning, Applications. Kluwer Academic Publishers, Boston, Dordrecht, L., 2000.
3. *Aizenberg, N.N., Kukharev, G.A., Pak, I.O., Shmerko, V.D.* "Methods of solution of logical equations by using fast orthogonal transforms and techniques of their realizations" in IX Vsesoyuznoe sovetschanie po problemam upravleniya. Tezisy doklada, M., 1983.
4. *Aizenberg, N.N., Rabinovich, Z.L.* Some classes of functional systems of operations and canonical forms of many-valued logical functions // Kibernetika. – 1965. – No. 2. – P. 37–45.
5. *Aizenberg, N.N., Trofimljuk, O.T.* Conjunctive transforms for discrete signals and their applications of tests and the detection of monotone functions. // Kibernetika. – No. 5, K., 1981.
6. *Benjauthrit, B., Reed, I.S.* Galois switching functions and their applications. IEEE Trans. Computers, Vol. C-25 – No.1 – 1976 – P. 79–86.
7. *Besslich, Ph.W.* Efficient computation nethod for XOR logic design. IEE Proc., Part E., Vol. 130 – 1983 – P. 203–206.
8. *Cohn, M.* Switching Function Canonical Form over Integer Fields. PhD thesis, Harvard University, Cambridge, Massachusetts, USA, December 1960.
9. *Copeland, A.H.* Expansion of certain logical functions. Amer. Math. Month., Vol. 44, No. 4 – 1937 – P. 213-218.
10. *De Morgan, A.* Formal logic - or the calculus of inference, necessary and probable. London 1847, XVI+336.
11. *Dubrova, E.V., Muzio, J.C.* Generalized Reed-Muller canonical form for a multiple-valued algebra. Multiple-Valued Logic. – 1996. – Vol. 1. – P. 65–84.
12. *Dubrova, E.V., Muzio, J.C.* Testability of generalized multiple-valued Reed-Muller circuits. Proc. 26th Int. Symp. on Multiple-Valued Logic. May 29-31, 1996, Santiago de Campostela, Spain, 56-61.
13. *Falkowski, B.J., Fu, C.* Properties and experimental results of Fastest linearly independent ternary arithmetic transforms. IEEE Trans. Circuits and Systems, Vol. 53 – No. 4 – 2006 – P. 858-866.
14. *Falkowski, B.J., Lozano, C.C.* Quaternary Fixed-Polarity Reed-Muller expansion computation through operations on disjoint cubes and its comparison with other methods. Computers and Electrical Engineering, Vol. 31 – No. 2 – 2005 – P. 112-131.

15. *Falkowski, B.J., Rahardja, S.* Novel quantized transform for ternary systems. Proc. 25th Int. Symp. on Multiple-Valued Logic, 23.-25.5.1995, Bloomington, Indiana, USA, 117-122.
16. *Falkowski, B.J., Rahardja, S.* Fast construction of polarity coefficient matrices for fixed polarity quaternary Reed-Muller expansions. Proc. 5th Int. Workshop on Spectral Techiques, 15.-17.3.1994, Beijing, China, P. 220-225.
17. *Fu, C., Falkowski, B.J.* Quaternary Arithmetic Helix Transforms based on Kronecker products. Proc. Int. Symp. on Circuits and Systems, (38th ISCAS), Kobe, Japan, May 2005, P. 476-479.
18. *Fu, C., Falkowski, B.J.* Linearly independent ternary Arithmetic Helix Transforms, Their Properties and Relations. IEE Proc., Vision, Image and Signal Processing, UK, Vol. 153 – No. 2 – 2006 – P. 87-94.
19. *Garaev, M.U., Faradzhev, R.G.* On an analog of Fourier expansion over Galois fields and its applications to problem of generalized sequential machines. Izv. Akad. Nauk Azerb. SSR, Ser. Fiz.-Techn. i Math. Nauk – No.6 – 1-68 – 1965.
20. *Gibbs, J.E.* Instant Fourier transform. Electron. Lett. Vol.13 – No.5 – 1977 – P. 122-123.
21. *Green, D.H., Taylor, I.S.* Modular representation of multiple-valued logic systems. Proc. of the IEE, Vol. 121 – 1974 – P. 424-429.
22. *Harking, B., Moraga, C.* Efficient derivation of Reed-Muller expansions in multiple-valued logic systems. Proc. 22nd Int. Symp. on Multiple-Valued Logic, May 27-29, 1992, Sendai, Japan, P. 436-441.
23. *Heidtmann, K.D.* Arithmetic spectrum applied to fault detection for combinatorial networks. IEEE Trans. Computers. Vol.40, No.3, March 1991, P. 320-324.
24. *Hurst, S.L., Miller, D.M., Muzio, J.C.* Spectral Techniques in Digital Logic, Academic Press, Bristol, 1985.
25. *Karpovsky, M.G.* Finite Orthogonal Series in the Design of Digital Devices, Wiley and JUP, N.Y. and Jerusalem, 1976.
26. *Kodandapani, K.L., Setur, R.V.* Multi-valued algebraic generalization of Reed-Muller canonical forms. Proc. Int. Symp. on Multiple-Valued Logic, 1974, P. 505-544.
27. *Malyugin, V.D.* Paralleled Calculations by Means of Arithmetic Polynomials. Physical and Mathematical Publishing Company, Russian Academy of Sciences, M., 1997.
28. *Malyugin, V.D., Sokolov, V.V.* Intensive logic computation. Automatika and Telemekhanika. – 1993 – No.4.
29. *Muzio, J.C.* Stuck fault sensitivity of Reed-Muller and Arithmetic coefficients. C. Moraga, Ed. Theory and Applications of Spectral Techniques, Dortmund, 1989, P. 36-45.
30. *Muzio, J.C., Wesselkamper, T.C.* Multiple-valued Switching Theory. Adam Hilger, Bristol, 1986.
31. *Pradhan, D.K.* A multi-valued algebra based on finite fields. Proc. 1974 International Symp. on Multiple-Valued Logic, 1974, P. 95-112.
32. *Rahardja, S., Falkowski, B.J.* Fast Linearly independent arithmetic expansions. IEEE Trans. Computers, Vol. 48, No. 9 – 1999 – P. 991-999.
33. *Stanković, R.S.* Some remarks on Fourier transforms and differential operators for digital functions. Proc. 22nd Int. Symp. on Multiple-Valued Logic, May 27-29, 1992, Sendai, Japan, P. 365-370.
34. *Stanković, R.S., Moraga, C.* Reed-Muller-Fourier representations of multiple-valued functions over Galois fields of prime cardinality. in: Kebschull, U., Schubert, E., Rosentiel, W., Eds. Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, 16.-17.9.1993, Hamburg, Germany, P. 115-124.
35. *Stanković, R.S., Moraga, C.* An algebraic transform for prime-valued functions. Proc. 5th Int. Workshop on Spectral Techiques, Beijing, China, 15.-17.3.1994, 205-209, also in Stanković, R.S., Moraga, C., An algebraic transform for prime-valued functions", in Moraga, C., (ed.) Advances in Spectral Techniques, Berichte zur Angewandten Informatik, Dortmund University, Dortmund, Germany, ISSN 0946-2341, 1998, P. 34-37.
36. *Stanković, R.S., Moraga, C.* Reed-Muller-Fourier representations of multiple-valued functions. in: Recent Developments in Abstract Harmonic Analysis with Applications in Signal Processing, Nauka, Belgrade – 1996 – P. 205-216.
37. *Stanković, R.S., Janković, D., Moraga, C.* Reed-Muller-Fourier versus Galois field representations of four-valued logic functions. Proc. 28th Int. Symp. on Multiple-Valued Logic – 1998 –

P. 186-191.

38. *Zilić, Ž., Vranešić, Z.* Current-mode CMOS Galois field circuits. Proc. 23rd Int. Symp. on Multiple-Valued Logic, May 24-27, 1993, Sacramento, California, USA, P. 245-250.
39. *Zilić, Ž., Vranešić, Z.G.* A deterministic multivariate polynomial interpolation algorithm for small Finite Fields. IEEE Trans. Computers – Vol. 37 – No. 2 – 2002 – P. 1100-1105.
40. *Yanushkevich, S.N.* Spectral and differential methods to synthesize polynomial forms of MVL function on systolic arrays. Proc. 5th Int. Workshop on Spectral Techiques, 15.-17.3.1994, Beijing, China, P. 78-83.
41. *Yanushkevich, S.N.* Logic Differential Calculus in Multi-Valued Logic Design, Techn. University of Szczecin Academic Publishers, Poland, 1998.

Recived 10.10.2008