Remarks on Codes, Spectral Transforms, and Decision Diagrams

Radomir S. Stanković, Mark G. Karpovsky[1], Claudio Moraga[2]

Dept. of Computer Science, Faculty of Electronics, Niš, Serbia
[1]Dept. of Electrical and Computer Engineering, Boston University,
Boston, Massachusetts, USA
[2]European Centre for Soft Computing, 33600 Mieres, Spain &
Technical University of Dortmund, 44221 Dortmund, Germany

Dedicated to Professor Jaakko Astola on the occasion of his 60th birthday.

Abstract

In this paper, we discuss definitions, features, and relationships of Reed-Muller transforms, Reed-Muller codes and their generalizations to multiple-valued cases, and Reed-Muller decision diagrams. The novelty in this primarily review paper resides in putting together these concepts in the same context and providing a uniform point of view to their definition in terms of a convolutionwise multiplication. In particular, we point out that the Plotkin construction schemes for Reed-Muller codes used in coding theory are a different notation for basic Reed-Muller transform matrices over finite fields or can be alternatively viewed as decomposition rules used to define the Reed-Muller decision diagrams.

1 Introduction

Professor Jaakko Astola started his research career in the area of coding theory where he found a suitable subject for his PhD Thesis. Latter, he intensively studied and used spectral transforms as a classical tool in signal and image processing and system design. In the last ten years Prof. Astola has been efficiently using decision diagrams for representation of discrete signals. For that reason this paper provides some remarks about these three subjects.

In the first part of the paper, we discuss some generalizations of Reed-Muller codes and extensions to non-binary Reed-Muller codes with particular emphasis on methods to construct codes of larger length from codes of the smaller length. We observe that these methods are actually defined by referring to basic transform matrices that are used in spectral techniques to define the Reed-Muller and related transforms.

In the second part, we point out that the generating matrices of non-binary Reed-Muller codes are interpreted as kernels of certain spectral transform and studied from the point of view of functional expressions for representation of discrete functions. Basic functions in terms of which these transforms are defined can be used to define the Reed-Muller related codes.

The third part of the paper presents an approach to exploit spectral interpretation of decision diagrams in an opposite manner. We define a method to construct various spectral transforms (local and global) by assigning various basic transform matrices to nodes of decision diagrams. Some of these basic transform matrices can be selected among generating matrices of certain non-binary codes. In this way, codes will provide a basis to define spectral transforms. As a particular example we presented a Haar-like transform defined in terms of the basic generator matrix for the octacode.

Notice that whenever there is no danger of ambiguities or misinterpretation, the term Reed-Muller codes will be used in a general context to denote either the binary Reed-Muller code or its generalizations and extensions. We will use in the same way the terms Reed-Muller transforms and Reed-Muller decision diagrams. When necessary, more specific terms will be used to denote some particular members of these large families of codes, transforms, and decision diagrams.

2 Background and Related Work in Reed-Muller Codes

In this section, we briefly present some basic facts about Reed-Muller codes and their various generalizations and extensions. We restrict the discussion to the topics most relevant to the presentations in other sections as well as to some recent development in the area, in order to justify actuality of the topic. In this context, the term generalized Reed-Muller codes refers to the Reed-Muller codes where the domain is the same as for the classical binary Reed-Muller codes, i.e., Z_2^n. The term non-binary Reed-Muller codes refers to codes defined over finite fields F_q, $q > 2$ or related algebraic structures.

2.1 Binary Reed-Muller codes

Binary Reed-Muller codes are both theoretically interesting and useful in practice for two main reasons, optimality of their parameters and existence of fast decoding algorithms. For instance, there are such algorithms that are based on Walsh transform in Hadamard ordering [25]. From the point of view accepted for the presentations in this paper, derivation of such algorithms looks natural if we recall tight relationships between the Reed-Muller codes and the Reed-Muller expressions. More precisely, we notice that the Reed-Muller expressions can be derived from the Walsh expressions, if in the latter the Walsh functions are expressed in terms of Boolean variables and then calculations of coefficients are performed modulo 2, see, for instance, [36].

Binary Reed-Muller codes are defined in terms of Boolean functions $f : Z_2^n \to Z_2$, or when we refer to finite fields as underlying algebraic structures, these codes can be viewed as codes over the finite field F_2. The generator matrix is defined by referring to the Reed-Muller matrix or some rows of it selected according to the order of the code. Since permutation of rows of the generating matrix produces identical codes, the order of rows of the Reed-Muller matrix when used for rows of the generating matrix of the Reed-Muller codes can be different from the Hadamard ordering usually preferred in study and applications of the Reed-Muller expressions and transform in switching theory and logic design.

The binary Reed-Muller code $RM(n, k)$ is generated by the componentwise logic AND (the wedge product) of up to k binary variables x_i viewed as trivial switching functions $f(x_1, \ldots, x_n) = x_i$ and represented as binary vectors $\mathbf{x}_i$ of length 2^n. In other words, the binary Reed-Muller are defined in terms of monomials in Boolean variables. The notion will be illustrated by the following example.

Example 1 *(Binary Reed-Muller codes)*
If $n = 3$, we consider the field $F_2^3 = \{(0, 0, \ldots, 0), (0, 0, \ldots, 1), \ldots, (1, 1, \ldots, 1)\}$, the constant vector $\mathbf{1}$ of the length 2^3, and switching variables x_1, x_2, and x_3. Thus,

$$\begin{aligned}
\mathbf{1} &= [11111111], \\
\mathbf{x}_1 &= [00001111], \\
\mathbf{x}_2 &= [00110011], \\
\mathbf{x}_3 &= [01010101].
\end{aligned}$$

The $RM(3,1)$ code is generated by the set $\{\mathbf{1}, x_1, x_2, x_3\}$, or by the rows of the matrix

$$\mathbf{RM}(3,1) = \begin{array}{c} \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} 1 \\ x_3 \\ x_2 \\ x_1. \end{array}$$

Notice again that the code is invariant to the permutation of rows of this matrix.

The $RM(3,2)$ code is generated by the set $\{\mathbf{1}, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3\}$, or by the rows of the matrix

$$\mathbf{RM}(3,2) = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{array}{c} 1 \\ x_3 \\ x_2 \\ x_1 \\ x_2x_3 \\ x_1x_3 \\ x_1x_2 \end{array}$$

The code $RM(n,k)$ is linear, comprises $2^{\sum_{i=0}^{k}\left(\begin{array}{c} n \\ i \end{array}\right)}$ codewords, and has minimum Hamming (and Lee) distance 2^{n-k}.

In general, as noticed above, we can say that the binary Reed-Muller codes are defined in terms of monomials in Boolean variables. In matrix notation these monomials are generated as

$$\mathbf{X} = \bigotimes_{i=1}^{n} \mathbf{X}_i(1), \quad \mathbf{X}_i(1) = \left[\begin{array}{cc} 1 & x_i \end{array}\right], \quad x_i \in \{0,1\}. \tag{1}$$

Recall that in switching theory, the monomials defined above are called the Reed-Muller functions, see, for instance, [4]. In other words, the Reed-Muller codes are generated by selecting subsets of the Reed-Muller functions according to the order of the code. The Reed-Muller functions are a complete set, thus, a basis in the space of all Boolean functions for a given number of variables. This basis is used to express any Boolean function in the so-called *Positive-polarity Reed-Muller expression* (PPRM) [31], also called the Žhegalkin polynomial [45]. Notice that a PPRM is also called *algebraic normal form*, the notation most often used in coding theory [25].

2.2 Binary encoded Reed-Muller codes

Generalized or binary encoded Reed-Muller codes are defined in terms of generalized Boolean functions $f : Z_2^n \rightarrow Z_q$, where $q = 2^h$, $h \in N$, in a manner similar to that used to define the classical binary Reed-Muller codes. Any generalized Boolean function can be uniquely represented by the generalized PPRM, which consists of the same set of monomials as PPRM for Boolean functions, however, with coefficients in Z_{2^h} instead of $Z_2 = \{0,1\}$.

There are several generalizations and the corresponding definitions of the Reed-Muller codes. For example, in [7], generalized Reed-Muller codes are defined as follows.

Definition 1 *(Binary encoded Reed-Muller codes) [7]*
For $h > 1$ and $0 \leq k \leq n$, the kth-order linear code $RM_{2^h}(n,k)$ over Z_{2^h} of length 2^n is generated by the monomials in x_i of degree at most k.

The thus defined $RM_{2^h}(n,k)$ code generalizes the binary Reed-Muller code $RM(n,k)$ from the alphabet Z_2 to the alphabet Z_{2^h}, involving it as the particular case for $h = 1$. The code contains $2^{h\sum_{i=0}^{k}\left(\begin{array}{c} n \\ i \end{array}\right)}$ codewords.

Example 2 *The generator matrix of the code $RM_{2^h}(4,1)$ is*

$$\mathbf{RM}_{2^h}(4,1) = \begin{array}{c} \left[\begin{array}{cccccccccccccccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&1&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{array}\right] \end{array} \begin{array}{c} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}$$

This code contains 2^{5h} codewords for $h \geq 1$.

Definition 2 *(Binary encoded Reed-Muller codes of type ZRM) [7]*
For $h > 1$ and $0 \leq k \leq n+1$, the kth-order linear code $ZRM_{2^h}(n,k)$ over Z_{2^h} of length 2^n is generated by the monomials in the x_i of degree at most $k-1$ together with two times the monomials in the x_i of degree r (with the convention that the monomials of the degree -1 and $n+1$ are equal to zero).

This code is a generalization of the quaternary Reed-Muller code $ZRM(n,k)$ defined in [14] from the alphabet Z_4 to the alphabet Z_{2^h} involving the previous as the particular case for $h = 2$. The code contains $2^{h\sum_{i=0}^{k}\left(\begin{array}{c} n \\ i \end{array}\right)} \cdot 2^{(h-1)\left(\begin{array}{c} n \\ r \end{array}\right)}$ codewords.

Notice that these codes are sometimes mentioned in the literature also as *Generalizer Reed-Muller codes.*

Example 3 *The generator matrix of the code $ZRM_{2^h}(4,2)$ is*

$$\mathbf{RM}_{2^h}(4,2) = \left[\begin{array}{cccccccccccccccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&1&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&2&2&2&2 \\ 0&0&0&0&0&0&0&0&0&0&2&2&0&0&2&2 \\ 0&0&0&0&0&0&0&0&0&2&0&2&0&2&0&2 \\ 0&0&0&0&0&0&2&2&0&0&0&0&0&0&2&2 \\ 0&0&0&0&0&2&0&2&0&0&0&0&0&2&0&2 \\ 0&0&0&2&0&0&0&2&0&0&0&2&0&0&0&2 \end{array}\right] \begin{array}{c} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 2x_1x_2 \\ 2x_1x_3 \\ 2x_1x_4 \\ 2x_2x_3 \\ 2x_2x_4 \\ 2x_3x_4 \end{array}$$

The code contains $2^{5h} \cdot 2^{6(h-1)}$ codewords.

A further generalization of the Reed-Muller codes over the alphabet Z_{2^h} has recently been presented in [32] as follows.

Definition 3 *[32]*
For $h > p$ and $k \geq p$ we define the code $ZRM^p_{2h}(n,k)$ as the set of all vectors of length 2^n that can be associated with a generalized Boolean function $Z_2^n \to Z_{2^h}$ comprising the monomials of order at most $k-p$ and 2^i times the monomials of order $k-p+i$ with $i = 1, 2, \ldots, p$.

In this definition, for $p = 0$ we get the generalized Reed-Muller code $RM_{2^h}(n, k)$ and for $p = 1$ the code $ZRM_{2^h}(n, k)$. For $p \geq 1$ this definition produces new generalized Reed-Muller codes which appear convenient for applications in *Orthogonal Frequency Division Multiplexing* (OFDM) [32] with respect to the peak-to-mean envelope power ratio (PMEPR).

2.3 Non-binary Reed-Muller codes

Many authors refer to [8], [19], and [26] as sources of initial considerations of non-binary Reed-Muller codes. The concept of non-binary Reed-Muller codes will be introduced by the following example.

Example 4 *(Quoternary Reed-Muller codes) [2]*
For instance, the Reed-Muller codes for $q = 4$ and $n = 2$, $RM_4(2, 1)$, are defined by the generator matrix

$$\mathbf{RM}_4(2,1) = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Besides the constant 1 in the last row of the matrix, the first and the second row correspond to the quaternary variables x_1 and x_2.

The Pascal transform defined in terms of entries of the Pascal triangle written as the Pascal matrix has recently received a lot of attention, see for instance [1], [13], [12], [33], [34], [35], [46], and references therein. Referring to this transform is a direct and simple way to observe strong relationships between Reed-Muller codes and spectral transforms for either binary and non-binary cases.

In [26], it was observed that monomials that are used to define classical Binary Reed-Muller codes can be viewed as functions in matrix notation represented by columns of the Pascal matrix whose entries are calculated modulo 2. Therefore, the generator matrix of the classical binary Reed-Muller codes is defined by referring to the Pascal matrix modulo 2. This provides a straightforward way to generalization of the Reed-Muller codes over finite fields F_q of different modules. Such codes are defined by the generating matrices obtained from the Pascal matrix after calculation of its entries modulo q.

Example 5 *First 8 rows of the Pascal matrix are*

$$\mathbf{L}_{(9\times 9)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 & 0 & 0 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 & 0 \\ 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1 \end{bmatrix}.$$

When entries are calculated modulo 2, we get the Reed-Muller matrix that can be defined alternatively as

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

Table 1: Basic generating matrices of non-binary Reed-Muller codes for $q = 3, 4, 5$.

$$\mathbf{G}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \mathbf{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad \mathbf{G}_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 1 & 4 & 1 \end{bmatrix}$$

Table 2: The Plotkin construction schemes for non-binary Reed-Muller codes for $q = 3, 4, 5$.

q	Construction scheme
3	$(u + v + w \mid 2u + v \mid u)$
4	$(u + v + w + x \mid 3u + 2v + w \mid 3u + v \mid u)$
5	$(u + v + w + x + y \mid 4u + 3v + 2w + x \mid u + 3v + w \mid 4u + v \mid u)$

The matrix $\mathbf{X}(1)$ *in (1) can be viewed as the symbolic notation for* $\mathbf{R}(1)$*, and then from there the direct link to the definition of the Reed-Muller codes in terms of monomials in Boolean variables.*

It is not difficult to realize that the Plotkin constructions [27] for the binary Reed-Muller codes follows from $\mathbf{R}(1)$ *since*

$$\begin{bmatrix} v & u \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} u + v & | & u \end{bmatrix}.$$

This example explains the way of defining non-binary Reed-Muller codes as suggested in [26], see also [22]. It also suggests that the following remark can be given.

Remark 1 *(Reed-Muller codes and spectral transforms)*
The Plotkin construction method for Reed-Muller codes is an alternative way to express the Kronecker product structure of the Reed-Muller transform and the same interpretation extends to non-binary cases.

Table 1 shows the generating matrices for non-binary Reed-Muller codes for $q = 3, 4, 5$. Codes of higher dimensions can be defined by using the Plotkin construction schemes that are shown in Table 2.

In [26], it has been remarked as an important feature of the binary Reed-Muller matrix that it is a self-inverse matrix, with a reference to the work of Preparata [28] for applications of this matrix in switching theory. This feature of self-inverseness is lost in generalizations to q-valued case as the generalizations done in [26] for non-binary Reed-Muller codes.

We observe that columns of the Pascal matrix of order $(g \times g)$ can be generated as integer powers of the constant function $W(x) \equiv 1$, for all $x = 0, 1, 2, \ldots, g - 1$, under the exponentiation performed as the convolutionwise (Gibbs) multiplication defined as [1]

$$(fg)(0) \quad = \quad 0, \qquad (2)$$

[1] Notice that $W(x) \equiv 1$, is a vector of the length g whose all entries have the value 1 (the first column in the Pascal matrix) and its integer powers in terms of the convolutionwise (Gibbs) multiplication are also vectors that determine other columns of the Pascal matrix.

Table 3: Basic RMF-matrices for $q = 3, 4, 5$.

$$\mathbf{R}_{3,RMF}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{R}_{4,RMF}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 3 \end{bmatrix} \quad \mathbf{R}_{5,RMF}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Table 4: The Plotkin constructions schemes for the RMF-codes for $q = 3, 4, 5$.

$q = 3$	$(u + v + w \mid u + 2v \mid u)$
$q = 4$	$(u + v + w + x \mid u + 2v + 3w \mid 3u + v \mid 3u)$
$q = 5$	$(u + v + w + x + y \mid u + 2v + 3w + 4x \mid u + 3v + w \mid u + 4v \mid u)$

$$(fg)(x) \quad = \quad \sum_{s=0}^{\sigma(x)-1} f(\sigma(x) - 1 - s)g(s), \quad \forall x = 0, 1, \ldots g - 1, \quad x \neq 0.$$

where σ is a q-adic contraction defined as

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

We further observe that the property of self-inverseness can be regained if instead of the constant function 1, we take the function $W(x) \equiv q - 1$, as it has been done in defining the Reed-Muller-Fourier (RMF) transforms [42]. The basic RMF-transform matrices can be used to define the Reed-Muller-Fourier codes modulo q in the same manner as it was done in [26].

Table 3 and Table 4 show the basic RMF-transform matrices for $q = 3, 4, 5$, and the corresponding Plotkin construction schemes.

Notice that symbolic notation of columns of RMF-matrices in terms of q-valued variables and their integer powers with respect to the convolutionwise (Gibbs) multiplication lead to the q-ary monomials that can be used to define the RMF-codes in the same manner as discussed for the generalized Reed-Muller codes in Subsection 2.2. This approach will be discussed in more details in the example of ternary RMF-codes in Section 4 after presenting some basic facts on RMF-expressions (Section 3) in terms of these ternary monomials.

2.4 Quantum Reed-Muller codes

For the actuality of the topic, in this section we provide few remarks about the *quantum Reed-Muller codes* which recently attract a lot of attention due to the interest in quantum computing and related areas. Moreover, besides applications in OFDM, most of the recent work on non-binary Reed-Muller codes is related to error-correcting problems in quantum computing.

Quantum computing is based on quantum mechanical phenomena, such as superposition and entanglement to perform operation on data, by exploiting quantum properties to represent data and perform operations on them.

Table 5: Truth-table for ternary functions of $n = 2$ variables.

	x_1, x_2	$f(x_1, x_2)$
0.	00	$f(0,0)$
1.	01	$f(0,1)$
2.	02	$f(0,2)$
3.	10	$f(1,0)$
4.	11	$f(1,1)$
5.	12	$f(1,2)$
6.	20	$f(2,0)$
7.	21	$f(2,1)$
8.	22	$f(2,2)$

Quantum information is physical information that is contained in the state of a quantum system, and main task is to protect this information from interaction with environment, which can be expressed in terms of errors as decoherence (the mechanism by which quantum systems interact with their environments to exhibit probabilistically additive behaviour) and quantum noise. In that order *quantum error-correcting codes* are defined. Most widely used are the so-called *binary stabilizer codes* [6], [21], primarily due to their relationships with classical coding theory which permit to use known methods to construct good codes. Non-binary stabilizer codes have been introduced in late 90s, and their theory is quite incomplete.

In this area, codes that have some resemblance to the binary Reed-Muller codes are related to *group character codes* defined in [9]. Such codes have been extended in a quantum analogue of group character codes in [20].

Binary quantum Reed-Muller codes and non-binary quantum Reed-Muller codes were defined in [29] and [30], respectively. Non-binary quantum Reed-Muller codes are derived as a quantum analogue of classical generalized Reed-Muller codes that have been introduced in [19].

3 Ternary Reed-Muller-Fourier Expressions

Ternary Reed-Muller-Fourier codes can be defined very simply in terms of ternary logic functions. We want to define codes of length 3^n and, therefore, we will consider n ternary variables $x_1, \ldots, x_n$, $x_i \in \{0, 1, 2\}$. We assume that $x = (x_1, \ldots, x_n)$ range over the set of all ternary n-tuples V^n. Any function $f(x) = f(x_1, \ldots, x_n)$ which takes the values 0, 1, and 2 is called a *ternary function.* Such a function can be specified by a ternary truth-table that shows the value of f at all its 3^n arguments.

Example 6 *(Ternary functions)*
When $n = 2$, a ternary function is specified by the truth-table as in Table 5, where $f(i) \in \{0, 1, 2\}$.

It is clear that there are 3^{3^n} functions, since $f(i)$ can take any of three values 0, 1, and 2. We assume that rows of the truth-table are in natural (lexicographic) ordering as illustrated in Table 5.

The operations that can be applied to ternary logic functions in order to define the Reed-Muller-Fourier transforms and codes are

1. Addition and multiplication modulo 3, that are for convenience specified by Table 6.

Table 6: Addition, multiplication, and Gibbs exponentiation 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

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

2. Exponentiation defined as $x^{*0} = -1$ modulo 3, and for $j > 0$, x^{*i} is calculated in terms of the convolutionwise (Gibbs) multiplication [11], which in the case of ternary functions can be alternatively specified by the right-most entry in Table 6 [37], [41], [42].

Theorem 1 *(Reed-Muller-Fourier expressions)*
Any ternary function $f(x_1, \ldots, x_n)$ can be expanded in powers of x_i as

$$f(x_1, \ldots, x_n) = (-1)^n \sum_{a \in V^n} q(a) x_1^{a_1} \cdots x_n^{a_n},$$

where $q(a) \in \{0, 1, 2\}$.

In matrix notation, we first define the set of monomials in terms of $3EXP$ as

$$\mathbf{X}(n) = \bigotimes_{i=1}^{n} \mathbf{X}_i(1),$$

where

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

The coefficients $q(a)$ written as entries of a vector $\mathbf{Q} = [q(0), \ldots, q(3^n)]^T$ are calculated by using a matrix that is inverse to $\mathbf{X}(n)$ when its columns written in terms of numeric values variables can take. Thus,

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

where

$$\mathbf{R}_i(1) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{array} \right],$$

$\mathbf{F}$ is the function vector for $f(x_1, \ldots, x_n)$, and calculations are performed modulo 3.

The coefficients in Reed-Muller-Fourier expressions can be calculated by using FFT-like fast algorithm, which can be implemented either over function vectors or Multiple-place decision diagrams [44].

4 Ternary Reed-Muller-Fourier Codes

Ternary monomials defined by (3) can be used to define the ternary Reed-Muller-Fourier codes in the same way as that has been discussed for the generalized Reed-Muller codes in Section 2.2.

Assume that $x = (x_1, \ldots, x_n)$ is a vector that ranges over the set of all ternary n-tuples V^n and $\mathbf{F}$ is a vector of length 3^n specifying a ternary function $f(x_1, \ldots, x_n)$.

Definition 4 *(RMF-codes)*
The k-th order ternary Reed-Muller-Fourier (RMF) code $RM(n,k)$ of length $m = 3^n$ for $0 \leq k \leq n$ is the set of all vectors $\mathbf{F}$ specifying a function $f(x_1, \ldots, x_n)$ that is a ternary function which is a RMF-polynomial of degree at most k.

Example 7 *The first order RMF-code of length 9 consists of 27 codewords*

$$a_0 \cdot \mathbf{1} + a_1 x_1 + a_2 x_2, \quad a_i = 0, 1, 2.$$

These code words are shown in Table 7.

In general, the k-th order RMF-code consists of all linear combinations of the vectors corresponding to the products in $X(n)$, which therefore, form the basis for the code. There are

$$k = 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{r}$$

such basic vectors, which are linearly independent. Therefore, k is the dimension of the code.

Example 8 *Basis vectors for the RMF-code of length 9 are given in Table 8.*

4.0.1 Construction method

The RMF-code of length 3^{n+1} may be obtained from RMF-codes of length 3^n using a ternary generalization of the Plotkin construction for binary Reed-Muller codes [27]. The method is derived by referring to the basic RMF-matrix and by using considerations in [22] and [26].

It is known, see for instance [22], [25], that if U, V, and W are linear (n, k_U), (n, k_V), (n, k_W) codes with the generating matrices $\mathbf{G}_U$, $\mathbf{G}_V$, and $\mathbf{G}_W$, respectively, then the $(3n, k_U k_V k_W)$ code $\varphi = (u + v + w \mid u + 2v \mid u)$, $u \in U$, $v \in V$, $w \in W$, is a linear code with the generator matrix

$$\mathbf{G}_\varphi = \begin{bmatrix} \mathbf{G}_W & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_V & 2\mathbf{G}_V & \mathbf{0} \\ \mathbf{G}_U & \mathbf{G}_U & \mathbf{G}_U \end{bmatrix},$$

where $\mathbf{0}$ is the zero-matrix. The code φ has the distance $d_\varpi = min\{3d_U, 2d_V, d_W\}$, where d_U, d_V, and d_W are distances of U, V, and W, respectively.

The generation the RMF-codes is a particular case of this more general method in the same way as generation of ternary Reed-Muller codes defined in [26] is another particular case of the same method. Due to that observation, it can be shown that the RMF-codes can be generated recursively by using the construction scheme $(u + v + w \mid u + 2v \mid u)$. The method corresponds to that used in [22] to generate ternary Reed-Muller codes defined in [26] and consists in the following.

Table 7: Code words in first order RMF-code of length 9.

Polynomial	Code word
$\mathbf{0}$	000000000
x_1	000111222
$2x_1$	000222111
x_2	012012012
$2x_2$	021021021
$x_1 + x_2$	012120201
$2x_1 + x_2$	012201120
$x_1 + 2x_2$	021102210
$2x_1 + 2x_2$	021210102
$\mathbf{1}$	111111111
$1 + x_1$	111222000
$1 + 2x_1$	111000222
$1 + x_2$	120120120
$1 + 2x_2$	102102102
$1 + x_1 + x_2$	120201012
$1 + 2x_1 + x_2$	120012201
$1 + x_1 + 2x_2$	102210021
$1 + 2x_1 + 2x_2$	102021210
$\mathbf{2}$	222222222
$2 + x_1$	222000111
$2 + 2x_1$	222111000
$2 + x_2$	201201201
$2 + 2x_2$	210210210
$2 + x_1 + x_2$	201012120
$2 + 2x_1 + x_2$	201120012
$2 + x_1 + 2x_2$	210021102
$2 + 2x_1 + 2x_2$	210102021

Table 8: Basis vectors for the RMF-code of length 9.

$\mathbf{2}$	222222222
x_2	012012012
x_2^{*2}	002002002
x_1	000111222
x_1x_2	000021012
$x_1x_2^{*2}$	000001001
x_1^{*2}	000000222
$x_1^{*2}x_2$	000000012
$x_1^{*2}x_2^{*2}$	000000002

Due to its Kronecker product structure, the RMF-matrix $\mathbf{R}_{3,RMF}(n)$ can be defined recursively as

$$\mathbf{R}_{3,RMF}(n) = \begin{bmatrix} \mathbf{R}_{3,RMF}(n-1) & \mathbf{0}(n-1) & \mathbf{0}(n-1) \\ \mathbf{R}_{3,RMF}(n-1) & 2\mathbf{R}_{3,RMF}(n-1) & \mathbf{0}(n-1) \\ \mathbf{R}_{3,RMF}(n-1) & \mathbf{R}_{3,RMF}(n-1) & \mathbf{R}_{3,RMF}(n-1) \end{bmatrix}.$$

By repeating the method used in [26], it can be shown that a code generated by some subset of the rows of $\mathbf{R}_{3,RMF}(n)$ having weights $\{w_1, w_2, \ldots, w_k\}$ has a minimum distance equal to $\min\{w_1, w_2, \ldots, w_k\}$. Therefore, including all rows of $\mathbf{R}_{3,RMF}(n)$ with weight $w \geq d$ yields a code with minimum distance d. By repeating the method in [22] based on induction and exploiting of the recursive definition of $\mathbf{R}_{3,RMF}(n)$, it can be shown that any code generated from the rows of $\mathbf{R}_{3,RMF}(n)$ can be generated recursively using the construction scheme $(u + v + w \mid u + 2v \mid u)$. The proof is omitted since the construction of it is identical to that in [22] with a minor difference originating in the structure of the matrix $\mathbf{R}_{3,RMF}(n)$ compared to the generating matrices of ternary Reed-muller codes in [26].

It should be noticed that $RMF(n, n)$ contains all vectors of length 3^n, $RMF(n, n-1)$ contains all even weight vectors and $RMF(n, 0)$ consists of the vectors 0,1, and 2.

The RMF-expressions are a generalization of the binary Reed-Muller expressions by replacing the function $W(x) \equiv 1$ with $W \equiv q - 1$, $q > 2$. These expressions reduce to the binary Reed-Muller expressions for $q = 2$. The same is for the Reed-Muller-Fourier codes, they become identical to the classical binary Reed-Muller codes for $q = 2$.

5 Reed-Muller-Fourier Decision Diagrams

From their spectral interpretation, decision diagrams can be viewed as graphical representations of some functional (spectral) expressions [39]. In this way, the Reed-Muller-Fourier decision diagrams (RMFDDs) are a subclass of functional decision diagrams for multiple-valued logic functions [38]. The decomposition rule used at the nodes of a decision diagram related to the Reed-Muller transform is derived from the basic Reed-Muller transform matrix. Therefore, the following remark which expresses the relationships between the Reed-Muller codes and Reed-Muller diagrams is possible.

Remark 2 *(Reed-Muller codes and diagrams)*
The Plotkin construction scheme for Reed-Muller codes is a different expression of the expansion rules used in definition of spectral transform decision diagrams related to the Reed-Muller transforms and its generalizations to non-binary cases.

This remark will be illustrated by the following example.

Example 9 *The Plotkin construction C for RMF-codes for $q = 4$ as shown in Table 4 is determined as*

$$\begin{aligned} C &= \begin{bmatrix} x & w & v & u \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 3 \end{bmatrix} \\ &= (u + v + w + x \mid u + 2v + 3w \mid 3u + v \mid 3u). \end{aligned} \qquad (4)$$

In the notation commonly used in decision diagrams theory, the expansion rule corresponding to the RMF-transform for $q = 4$ is defined as

$$f = 3 \begin{bmatrix} 3 & x & x^{*2} & x^{*3} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix},$$

where f_i, $i = 0, 1, 2, 3$ are co-factors of f for $x = i$.

From there,

$$f = f_0 \oplus x(3f_0 \oplus f_1) \oplus x^{*2}(3f_0 \oplus 2f_1 \oplus 3f_2) \oplus x^{*3}(3f_0 \oplus 3f_1 \oplus f_2 \oplus f_3) \tag{5}$$

The equality (5) represents f in terms of its co-factors f_i, $i = 0, 1, 2, 3$, and therefore can be called the RMF-expansion rule for quaternary functions, by the analogy with similar expansion rules in binary and multiple-valued logic [4], [39].

It is easy to realize that this expansion rule up to encoding directly corresponds to the Plotkin construction scheme shown in Table 4. More precisely, assume that in the Plotkin construction scheme for $q = 4$ in Table 4 we perform encoding $u = 3$, $v = x$, $w = x^{*2}$, and $x = x^{*3}$. Then, the Plotkin construction scheme C is derived as in (4), yielding

$$C = (3 + x + x^{*2} + z^{*3} \mid 3x + 2x^{*2} + x^{*3} \mid x^{*2} + 3x^{*3} \mid 3x^{*3}).$$

If the thus determined C is viewed as an (1×4) matrix and multiplied by the function vector $\mathbf{F} = [f_0, f_1, f_2, f_3]^T$, it produces, after a simple recalculation and multiplication by the scalling factor 3, the functional expression (5). Thus, it is possible to write $f = 3C \cdot \mathbf{F}$.

When this expansion rule is performed recursively with respect to all the variables in an n-variable function f, it yields to the RMF-expression for f. The RMFDDs are defined as graphical representations of RMF-expressions. Thus, at each node of the RMFDD, the expansion (5) is performed and the values of constant nodes are the RMF-coefficients of a given function f. Each path from the root node to a constant node corresponds to a column in the RMF-matrix and, therefore, labels at the edges are denoted by 3, x_i, x_i^{*2}, x_i^{*3}.

Fig. 1 shows the Reed-Muller-Fourier decision tree (RMFDT) for a two-variable quaternary function $f(x_1, x_2)$, $x_1, x_2 \in \{0, 1, 2, 3\}$, specified by the vector $\mathbf{F} = [0, 0, 0, 0, 0, 1, 3, 2, 0, 3, 2, 1, 0, 2, 1, 3]^T$. The RMF-spectrum of f is $\mathbf{S}_{f,RMF} = [0, 0, 0, 0, 0, 1, 3, 0, 0, 3, 2, 0, 0, 0, 0, 2]^T$. The spectrum is calculated by using the (16×16) RMF-transform matrix that is the Kronecker product of the basic RMF-transform matrix $R_{4,RMF}(1)$ in Table 3. Fig. 2 shows the Reed-Muller-Fourier decision diagram (RMFDD) for f derived by the reduction of the decision tree by using the correspondingly generalized reduction rules for decision diagrams [4]. Notice that reduction rules assume recalculation of labels at the edges of deleted nodes [2]. For instance, for this reason the label at the leftmost outgoing edge of the root node is $3(3 + x_2 + x_2^{*2} + x^{*3})$, and similar for other edges.

The edge-valued RMF-decision diagrams have been discussed in [40], and Haar-like RMF-diagrams in [43].

In what follows, for an illustration of the general approach to spectral interpretation of decision diagrams and links to the generating matrices of codes we will define the decision diagrams and Haar-like decision diagrams in terms of basic transform matrices used to define the octacode.

[2] Addition of labels at the edges pointing to the same value and multiplication with the labels at the incoming edges [39]

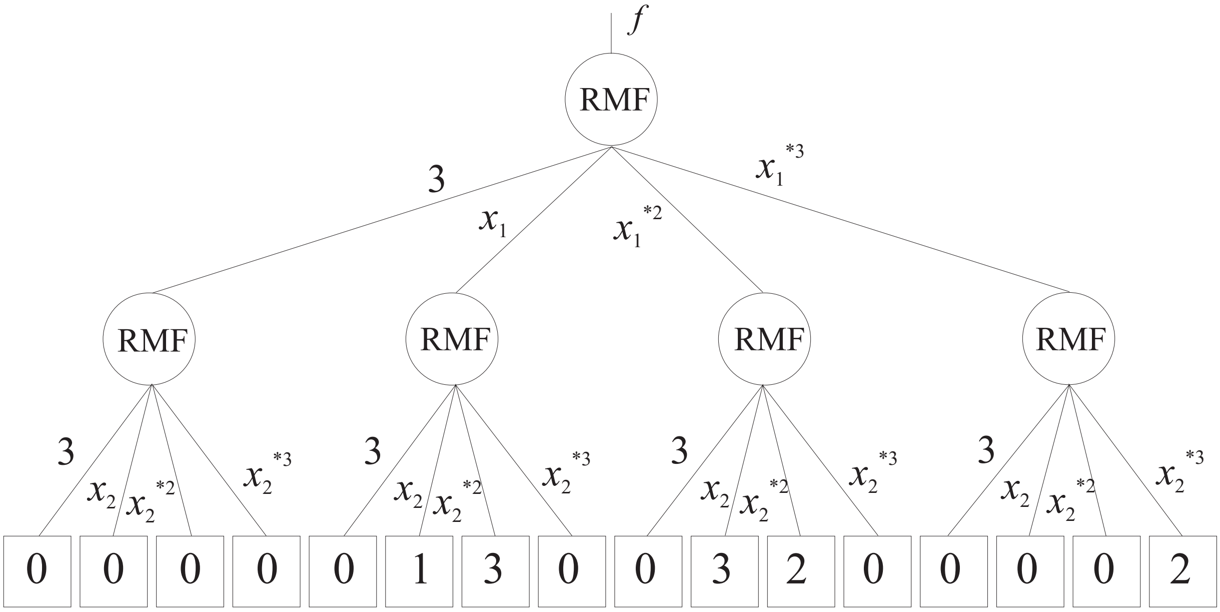

Figure 1: The RMFDT for f in Example 9.

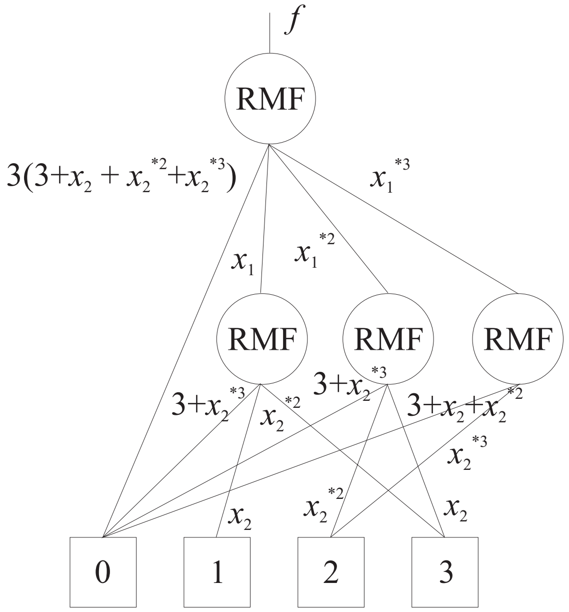

Figure 2: The RMFDD for f in Example 9.

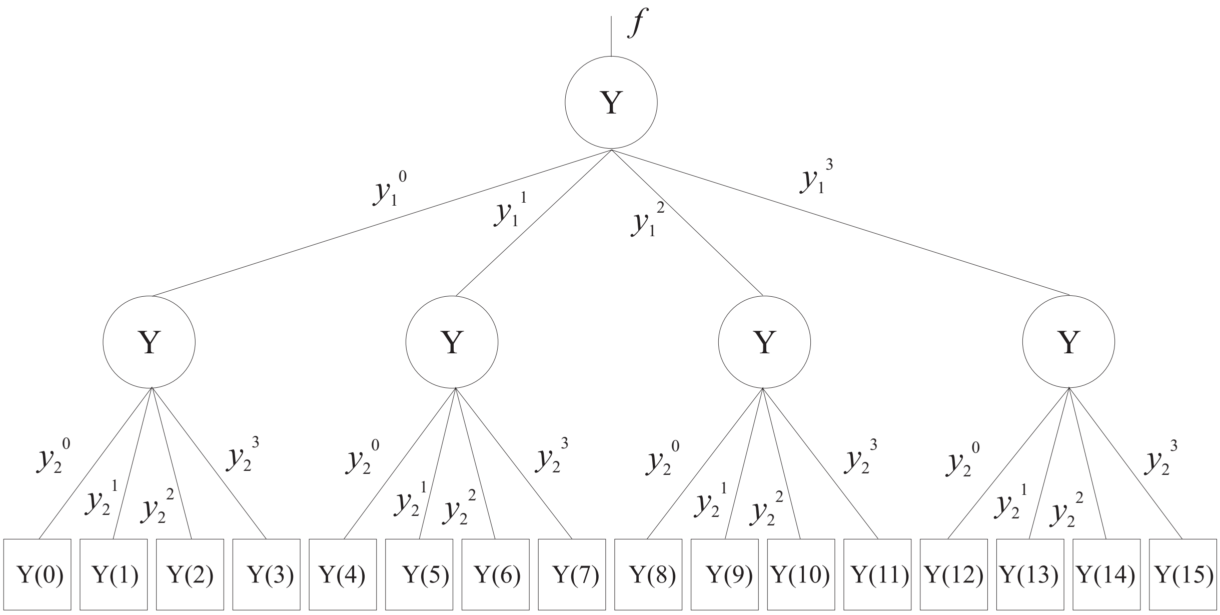

Figure 3: The decision tree defined by using the decomposition rule derived from $\mathbf{Y}(1)$.

This code can be obtained by encoding pairs of binary values in the binary Nordstrom-Robinson code by quaternary values as integers Z_4 modulo 4, see for instance [5], [10] It is a self-dual code, and can be defined by the generating matrix

$$\begin{aligned} \mathbf{G} &= [\mathbf{I}_4|\mathbf{P}] \\ &= \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 2 & 3 & 3 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 1 & 3 & 1 & 2 \end{array}\right]. \end{aligned}$$

We will select the matrix

$$\mathbf{Y}(1) = \begin{bmatrix} 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 1 & 3 & 1 & 2 \end{bmatrix}, \tag{6}$$

as a basic transform matrix to define a spectral transform. This matrix can be also used to derive a decomposition rule that can be assigned to nodes of a quaternary decision tree. Fig. 3 shows the decision tree defined in terms of this decomposition rule that is defined as

$$\begin{aligned} f &= y_0(2f_0 + 3f_1 + 3f_2 + 3f_3) + y_1(f_0 + 2f_1 + 3f_2 + f_3) \\ &+ y_2(f_0 + f_1 + 2f_2 + 3f_3) + y_3(f_0 + 3f_1 + f_2 + 2f_3), \end{aligned} \tag{7}$$

where y_i, $i = 0, 1, 2, 3$ are functions specified by columns of the matrix $\mathbf{Y}(1)$. Values of constant nodes are coefficients in the functional expression obtained by a recursive application of this decomposition rule to the variables in a two-variable quaternary function $f(x_1, x_2)$, $x_1, x_2 \in \{0, 1, 2, 3\}$.

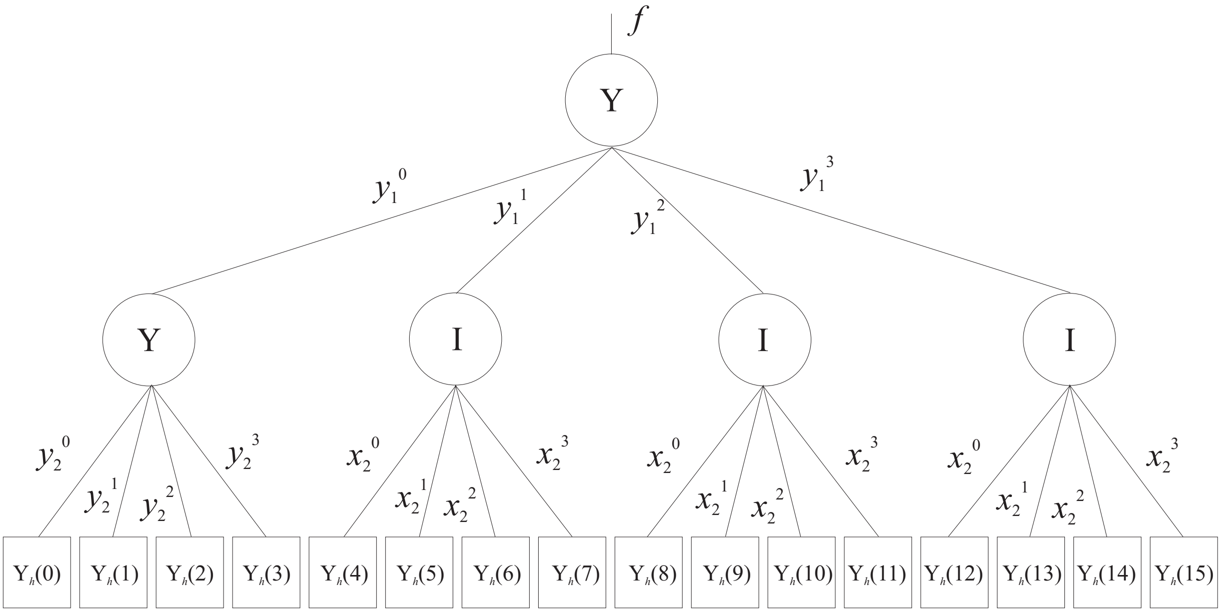

Figure 4: The decision tree defined by using the decomposition $\mathbf{Y}(1)$ and the generalized Shannon decomposition $\mathbf{I}_4$.

In matrix notation, the spectral transform defined by the decision tree in Fig. 3 is specified by the matrix

$$\mathbf{Y}(2) = \mathbf{Y}(1) \otimes \mathbf{Y}(1),$$

where $\otimes$ denotes the Kronecker product.

Fig. 4 shows a decision tree, where the decomposition rule (7) is assigned to the leftmost nodes at a level in the decision tree, while the decomposition rule for all other nodes is the generalized Shannon decomposition rule

$$f = x_i^0 f_0 + x_i^1 f_1 + x_i^2 f(2) + x_i^3 f(3),$$

where x_i is a quaternary variable.

This decomposition rule is alternatively defined by the identity matrix matrix of order 4, $\mathbf{I}_4$, which explains the label at the remaining non-terminal nodes in the decision tree in Fig. 4. This decision tree defines a Haar-like transform, which in matrix notation is specified as

$$\mathbf{Y}_h(n) = \begin{bmatrix} \mathbf{Y}_h(n-1) \otimes \mathbf{y}_0^T \\ \mathbf{I}(n-1) \otimes \mathbf{y}_1^T \\ \mathbf{I}(n-1) \otimes \mathbf{y}_2^T \\ \mathbf{I}(n-1) \otimes \mathbf{y}_3^T \end{bmatrix},$$

where $\mathbf{y}_i$, $i = 0, 1, 2, 3$, are columns of $\mathbf{Y}(1)$.

6 Closing Remarks

Reed-Muller codes are among the oldest and most widely understood concepts in coding theory. There are many generalizations and extensions to non-binary cases. These codes and their non-binary generalizations have been used in cryptography, in particular to design bent functions other perfect nonlinear functions with flat autocorrelations [17], see also [18] For instance, codes with flat dyadic autocorrelation related to the Reed-Muller codes and their generalizations to non-binary case for applications in robust data compressions of test responses have been proposed in [15], [16]. Applications in cryptography and protection of cryptodevices against fault injection attacks have been presented in [23] and [24]. Most of the recent work is related to quantum information transmitting and processing, see for instance [3], [20].

Reed-Muller transforms and their various extensions to multiple-valued functions are a powerful tool in switching theory for binary and multiple-valued logic functions. These transforms are tightly related to the generating matrices of the Reed-Muller codes, and some of their varieties can be used to define new codes with potentially useful features. In this paper we contemplated the Reed-Muller-Fourier transform in this context.

Reed-Muller transforms can be efficiently calculated by fast calculation algorithms performed over function vectors or decision diagrams. At the same time, these transforms can be used to define various spectral transform decision diagrams. Conversely, decision diagrams specified by assigning to their non-terminal nodes the decomposition rules derived from Reed-Muller transforms define various new classes of spectral transforms. In this paper, we show an example of a Haar-like transform derived from the quaternary Reed-Muller-Fourier transform.

There are again links to coding theory, since the Plotkin construction schemes for the Reed-Muller codes can be interpreted as either different notation for the basic transform matrices or a description of the decomposition rules in related decision diagrams.

References

[1] Aburdene, M.F., Goodman, T.J., "The discrete Pascal transform and its applications", *IEEE Signal Processing Letters*, Vol. 12, No. 7, 2005, 493-495.

[2] Ashikhmin, A.E., Litsyn, S.N., " Fast Decoding of Non-Binary First Order Reed-Muller Codes", *Applicable Algebra in Engineering, Communication and Computing, AAECC 7*, 1996, 299-308.

[3] Ashikhmin, A., Knill, E., "Nonbinary quantum stabilizer codes", *IEEE Trans. Inform. Theory*, Vol. 47, 2001, 3065-3072.

[4] Astola, J.T., Stanković, R.S., *Fundamentals of Switching Theory and Logic Design*, Springer, 2006.

[5] Bierbrauer, J., Fridrich, J., "Constructing Good Covering Codes for Applications in Steganography", Research report, 2007, SUNY Binghamton, Department of Electrical and Computer Engineering, T. J. Watson School of Applied Science and Engineering, Binghamton, NY 13902-6000, http://www.ws.binghamton.edu/fridrich/Research/stegocovsurveyOct07.pdf (viewed on September 6, 2008)

[6] Cleve, R., "Quantum stabilizer codes and classical linear codes", *Phys. Rev.*, Vol. A-55, No. 6, 1997, 4054-4059.

[7] Davis, J.A., Jedwab, J., "Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes", *IEEE Trans. Inform. Theory*, Vol. 45, 1999, 2397-2417.

[8] Delsarte, P., Goethals, J.-M., MacWilliams, F.J., "On generalized Reed-Muller codes and their Relatives", *Information and Control*, 16, 1974, 403-442.

[9] Ding, C., Kohel, D., Ling, S., ”Elementary 2-group character codes”, *IEEE Trans. Inf. Theory*, Vol. 46, 2000, 280-284.

[10] Forney, G.D., Sloane, N.J.A., Trott, M.D., ”The Nordstrom Robinson code is the binary image of the octacode”, in R. Calderbank, G.D. Forney Jr., N. Moayeri (eds.), *Coding and Quantization*, Amer. Math. Soc., 1993, 19-26.

[11] Gibbs, J.E., ”Instant Fourier transform”, *Electron. Lett.*, Vol. 13, No. 5, 122-123, 1977.

[12] Goodman, T.J., Aburdene, M.F., ”A hardware implementation of the discrete Pascal transform for image processing”, In Dougherty, E.R., Astola, J.T., Egiazarian, K.O., Nasrabadi, N.M., Rizvi, S.A., *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, Proceedings of the SPIE*, Volume 6064, 2006, 148-155.

[13] Goodman, T.J., Aburdene, M.F., ”On discrete Pascal transform, Poisson sequence and Laguerre polynomials”, *Electronics Letters*, Vol. 43, No. 14, July 5, 2007, 780-781.

[14] Hammons, A.R. Jr., Kumar, P.V., Calderbank, A.R., Sloane, N.J., Solé, P., ”The Z_4-linearity of Kedock, Preparata, Goethals, and realted codes”, *IEEE Trans. Inform. Theory*, Vol. 40, 1994, 301-319.

[15] Karpovsky, M.G., Nagvajara, P., ”Optimal robust compression of test responses”, *IEEE Trans. Computers*, Vol. 39, No. 1, 1990, 138-141.

[16] Karpovsky, M.G., Nagvajara, P., ”Optimal codes for the minimax criterion on error detection”, *IEEE Trans. Inform. Theory*, Vol. Vol. IT-35, No. 6, 1989, 1299-1305.

[17] Karpovsky, M.G., Nagvajara, P., ”Functions with flat autocorrelation and their generalizations”, in C. Moraga (ed.), *Theory and Applications of Spectral Techniques*, (*Proc. 3rd International Workshop on Spectral Techniques*), Forschungsbereicht 286, ISSN 0933-6192, Dortmund University, Dortmund, West Germany, 1988, 56-66.

[18] Karpovsky, M.G., Stanković, R.S., Astola, J.T., *Spectral Logic and Its Applications for the Design of Digital Devices*, Wiley & Sons, 2008.

[19] Kasami, T., Lin, S., Peterson, W.W., ”New generalizations of the Reed-Muller codes Part I : Primitive codes”, *IEEE Trans. Inform. Theory*, Vol. 14, No. 2, 1968, 189-199.

[20] Ketkar, A., Klappenecker, A., Kumar, S., Kiran Sarvepalli, P., ”Nonbinary stabilizer codes over finite fields”, *IEEE Trans. Inform Theory*, Vol. 52, No. 11, 2006, 4892-4914.

[21] Klappenecker, A., Sarvepalli, P.K., ”Clifford code constructions of operator quantum error-correcting codes”, *IEEE Trans. Inform. Theory*, Vol. 54, No. 12, 2008, 5760-5765.

[22] Kschischang, F.R. Pasupathy, S. , ”Some ternary and quaternary codes and associated sphere packings”, *IEEE Trans. Inform Theory, Part 2,* Vol. 38, No. 2, 1992, 227-246.

[23] Kulikowski, K.J., Karpovsky, M.G., Taubin, A., ”Robust codes for fault resistant cryptographic hardware”, *Proc. Int. Workshop on Fault Detection and Tolerance in Cryptography*, August 2005.

[24] Kulikowski, K.J., Karpovsky, M.G., Taubin, A., ”Robust codes and robust, fault tolerant architectures of the advanced encryption standard”, *Journal of System Architecture*, Vol. 53, pp138-149, 2007.

[25] MacWilliams, F.J., Sloane, N.J.A., *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[26] Massey, J.L., Costello, D.J., Jr., Justesen, J., "Polynomial weights and code constructions", *IEEE Trans. Inform. Theory*, Vol. IT-19, No. 1, 1973, 101-110.

[27] Plotkin, M., "Binary codes with specified minimum distances", *IEEE Trans. Inform. Theory*, Vol. 6, No. 4, 1960, 445-450.

[28] Preparata, M., "State-logic relations for autonomous sequential networks", *IEEE Trans. Electron, Comput.*, Vol. EC-13, 1964, 542-548.

[29] Steane, A., "Quantum Reed-Muller codes", *IEEE Trans. Inf. Theory*, Vol. 45, 1999, 1701-1703.

[30] Sarvepalli, P.K., Klappenecker, A., "Nonbinary quantum Reed-Muller codes", *Proc. 2005 IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, 2005, 1023-1027.

[31] Sasao, T., *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.

[32] Schmidt, K.-U., Finger, A., "New codes for OFDM with low PMEPR", *Proc. Int. Symp. on Information Theory, (ISIT 2005)*, September 4-9, 2005, 1136-1140.

[33] Skodras, A.N., "On the computation of the Discrete Pascal transform", *Hellenic Open University: Technical Report HOU-CS-TR-2005-06-EN*, December 2005, 1-9.

[34] Skodras, A.N.,"Fast Discrete Pascal Transform", *Electronics Letters*, Vol. 42, No. 23, 2006, 1367-1368.

[35] Skodras, A.N., "Efficient computation of the Discrete Pascal Transform", *Proc. 14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, September 2006.

[36] Stanković, R.S., "A note on the relation between Reed-Muller and Walsh transform", *IEEE Trans. Electromagnetic Compatibility*, Vol. EMC-24, No. 1, 1982, 68-70.

[37] 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, 365-370.

[38] Stanković, R.S., "Functional decision diagrams for multiple-valued functions", *Proc. 25-th Int. Symp. on Multiple-Valued Logic*, 23-25.5.1995, Bloomington, Indiana, U. S. A., 284-289.

[39] Stanković, R.S., Astola, J.T., *Spectral Interpretation of Decision Diagrams*, Springer, 2003.

[40] Stanković, R.S., Astola, J.T., "Edge-valued decision diagrams based on partial Reed-Muller transforms", in Marchuk, V.I., (ed.), *Practical Aspects of Digital Signal Processing*, Shakhty SRSUES, 2008, 10-20.

[41] Stanković, R.S., Astola, J.T., Moraga, C., "Remarks on generalizations of Reed-Muller expressions for binary and multiple-valued functions", *Visnyk Uzhgorodskogo Natsionalnogo Universytetu. Seria Matematica i Informatica (The News of Uzhgorod National University. Seies Mathematics and Informatics)*, Vol. 17, December 2008, 213-229.

[42] 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, 115-124.

[43] Stanković, R.S., Stanković, M., Moraga, C., "Haar wavelet transforms and Haar spectral transform decision diagrams for switching and multiple-valued functions", *Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 1-2, 2005.

[44] Stanković, R.S., Stanković, M., Moraga, C., Sasao, T., "Calculation of Reed-Muller-Fourier coefficients of multiple-valued functions through multiple-place decision diagrams", *Proc. Twenty-Fourth International Symposium on Multiple-Valued Logic*, May 25-27, 1994, 82-88.

[45] Zhegalkin, I.I., "O tekhnyke vychyslenyi predlozhenyi v symbolytscheskoi logykye", *Math. Sb.*, Vol. 34, 1927, 9-28, in Russian.

[46] Zhong, Q.-C., Nandi, A.K., Aburdene, M.F., "Efficient implementation of discrete Pascal transform using difference operators", *Electronics Letters*, Vol. 43, No. 24, 2007, 1348-1350.