[3] V. S. Pless, "A classification of self-orthogonal codes over GF(2)," *Discr. Math.*, vol. 3, pp. 209–246, 1972.
[4] J. H. Conway and V. S. Pless, "On the enumeration of self-dual codes," *J. Combin. Theory Ser. A*, vol. 28, pp. 26–53, 1980.
[5] J. H. Conway, V. S. Pless, and N. J. A. Sloane, "The binary self-dual codes of length up to 32: A revised enumeration," *J. Combin. Theory Ser. A*, vol. 60, pp. 183–195, 1992.
[6] F. J. MacWilliams, N. J. A. Sloane, and J. G. Thompson, "Good self-dual codes exist," *Discr. Math.*, vol. 3, pp. 153–162, 1972.
[7] C. L. Mallows and N. J. A. Sloane, "An upper bound for self-dual codes," *Inform. Contr.*, vol. 22, pp. 188–200, 1973.
[8] V. S. Pless, "On the uniqueness of the Golay codes," *J. Combin. Theory Ser. A*, vol. 5, pp. 215–228, 1968.
[9] M. Harada, "Existence of new extremal doubly-even codes and extremal singly-even codes," *Des. Codes Cryptogr.*, vol. 8, pp. 273–283, 1996.
[10] V. D. Tonchev, "Block designs of Hadamard type and self-dual codes," *Probl. Pered. Inform.*, vol. 19, pp. 25–30, 1983. English translation in *Probl. Inform. Transm.*, vol. 19, pp. 270–274, 1983.
[11] ——, "Self-orthogonal designs and extremal doubly-even codes," *J. Combin. Theory Ser. A*, vol. 52, pp. 197–205, 1989.
[12] F. C. Bussemaker and V. D. Tonchev, "Extremal doubly-even codes of length 40 derived from Hadamard matrices or order 20," *Discr. Math.*, vol. 80, pp. 317–321, 1990.
[13] V. Y. Yorgov, "Binary self-dual codes with an automorphism of odd order," *Probl. Pered. Inform.*, vol. 19, pp. 11–24, 1983. English translation in *Probl. Inform. Transm.*, vol. 19, pp. 260–270, 1983.
[14] V. Y. Yorgov and N. P. Ziapkov, "Doubly-even self-dual [40, 20, 8] codes with an automorphism of odd order," *Probl. Pered. Inform.*, vol. 32, 1996. English translation in *Probl. Inform. Transm.*, vol. 32, pp. 253–257, 1996.
[15] M. Ozeki, "Hadamard matrices and doubly-even self-dual error-correcting codes," *J. Combin. Theory Ser. A*, vol. 44, pp. 274–287, 1987.
[16] M. Harada, T. A. Gulliver, and H. Kaneta, "Classification of extremal double circulant self-dual codes of length up to 62," *Discr. Math.*, vol. 188, pp. 127–136, 1998.
[17] J. Leech and N. J. A. Sloane, "Sphere packing and error-correcting codes," *Canad. J. Math.*, vol. 23, pp. 718–745, 1971. See also [2, Ch. 5].
[18] B. B. Venkov, "The classification of integral even unimodular 24-dimensional quadratic forms," *Trudy Mat. Instit. Imeni V. A. Steklova*, vol. 148, pp. 65–76, 1978. See also [2, Ch. 16].
[19] O. D. King. A mass formula for unimodular lattices with no roots. [Online]. Available: http://arXiv:math.NT/0012231
[20] Y. Kitaoka, *Arithmetic of Quadratic Forms*. Cambridge, U.K.: Cambridge Univ. Press, 1993, vol. 106, Cambridge Tracts in Mathematics.
[21] H. Katsurada, "An explicit formula for Siegel series," *Amer. J. Math.*, vol. 121, pp. 415–452, 1999.

Reed–Muller Codes: Projections onto GF (4) and Multilevel Construction

Ofer Amrani, *Member, IEEE*, and Yair Be'ery, *Senior Member, IEEE*

***Abstract*—A projection of binary Reed–Muller codes $\mathcal{R}(r, m)$ onto GF $(4)^{m-2}$ is presented. For an $\mathcal{R}(r, m)$ code, this operation yields a linear quaternary code with the same length, dimension, and minimum distance as the Reed–Muller $\mathcal{R}(r-1, m-2)$ code. Based upon this projection, multilevel construction is given for $\mathcal{R}(r, m)$, where the constituent codes applied to the different levels are themselves the Reed–Muller codes $\mathcal{R}(r-2, m-2)$ and $\mathcal{R}(r, m-2)$, as well as the aforementioned quaternary code. This construction of Reed–Muller codes is readily applicable for their efficient decoding.**

***Index Terms*—Mapping, multilevel, multistage, projection, Reed–Muller codes.**

I. Introduction

The continuing search focusing on new constructions of known binary block codes is a most interesting theoretical problem but may also prove beneficial from the practical viewpoint. Different constructions can lead to different decoding algorithms, hence, for practical applications, one is interested in those constructions that assist in the reduction of the decoding complexity. The pioneering work of Hammons *et al.* [9] and Forney *et al.* [8] are two such examples. They state that certain outstanding nonlinear binary codes can be constructed from appropriate linear codes over the integer residue ring Z_4 by using the so-called Gray mapping. Thus, clearly, one can more easily decode the Z_4-linear version rather than the original nonlinear binary code. Another interesting example, closer related to the current work, is the Pless construction of the Golay code [13]. It is based on projecting the binary codewords onto GF (4), a construction which led to some of the most practically attractive soft-decision algorithms for the Golay code and the related Leech lattice [1], [16]. The projection idea has also been successfully employed for the construction and efficient soft decoding of the extended quadratic residue code of length 32 [18] and the Nordstrom–Robinson code [15].

Binary Reed–Muller codes are among the most prominent families of codes in coding theory. They have been extensively studied and employed for practical applications. In this work, a general three-level construction of Reed–Muller codes is presented, based upon the projection of the binary codewords onto codes over GF (4). Interestingly, for a binary Reed–Muller code $\mathcal{R}(r, m)$, the parameters of the projected code over GF (4) are similar to those of the Reed–Muller $\mathcal{R}(r-1, m-2)$ code; furthermore, the codes applied to the other two levels of the construction are the Reed–Muller codes $\mathcal{R}(r-2, m-2)$ and $\mathcal{R}(r, m-2)$. Note that the parameters of the three constituent codes are the same as in Forney's two-level squaring construction of $\mathcal{R}(r, m)$ [7] except that all the constituent codes in [7] are binary.

As in the case of the Golay codes, the proposed construction and, in particular, the projection onto linear codes over GF (4), should enable efficient maximum-likelihood and multistage soft decoding of the Reed–Muller codes. Some examples and complexity considerations are

Manuscript received October 1, 1998; revised September 17, 2000. The material in this correspondence was presented in part at the Israeli–French Workshop on Coding and Information Integrity.

The authors are with the Department of Electrical Engineering–Systems, Tel-Aviv University, Ramat-Aviv 69978, Tel-Aviv, Israel (e-mail: ofera@eng.tau.ac.il; ybeery@eng.tau.ac.il).

Communicated by F. R. Kschischang, Asocciate Editor for Coding Theory.
Publisher Item Identifier S 0018-9448(01)06229-0.

discussed. This should also make $\mathcal{R}(r,\ m)$ codes attractive candidates for Block turbo-code schemes, i.e., product codes with iterative decoding [14]. Moreover, since the family of Barnes–Wall lattices are closely related to Reed–Muller codes [5], [6], the same construction can be employed for generating these lattices and for their efficient soft decoding. Nonlinear relatives of the Reed–Muller codes may also benefit from this construction which would make them easier to decode [15], [2].

Finally, it is noteworthy that 1) Reed–Muller codes are usually not Z_4-linear [9], [10], though they may be regarded as "GF(4)-linear" for any r and m; 2) when they are indeed Z_4-linear, the number of Z_4 codewords is the same as the number of binary codewords in the Reed–Muller code due to the one-to-one mapping, while the number of codewords over GF(4) is considerably smaller due to the projection operation.

In the next section, the projection of binary sequences onto GF(4) is briefly described and some simple Reed–Muller codes are defined in order to demonstrate the main ideas behind the proposed construction. A more formal mathematical definition of this construction, based on the definition of Reed–Muller codes in terms of Boolean functions, is given in Section III for all Reed–Muller codes $\mathcal{R}(r,\ m)$. Several different representations of the obtained construction are also described. Finally, decoding complexity issues and further considerations are discussed in Section IV.

II. Preliminaries and Simple Constructions

Hereafter, the elements of GF$(4) = \{\mathbf{0},\ \mathbf{1},\ \boldsymbol{\alpha},\ \boldsymbol{\beta}\}$ will be referred to as *symbols*. A binary four-tuple $b = (b_1,\ b_2,\ b_3,\ b_4)^t$ with the scalar product

$$(\mathbf{0},\ \mathbf{1},\ \boldsymbol{\alpha},\ \boldsymbol{\beta}) \cdot b = \mu \in \mathrm{GF}(4) \tag{1}$$

will be called a *binary image* of μ. Conversely, μ will be called the *projection* of the binary four-tuple b. Clearly, each GF(4) symbol has two complementary even-weight and two complementary odd-weight images.

Let $\boldsymbol{b} = (b_1,\ b_2,\ \ldots,\ b_n)$ be a binary vector of length n divisible by 4, and arrange its elements in a four rows by $\frac{n}{4}$ columns, $4 \times \frac{n}{4}$, array

$$\boldsymbol{b} = \begin{bmatrix} b_1 & b_5 & & b_{n-3} \\ b_2 & b_6 & & b_{n-2} \\ b_3 & b_7 & \ldots & b_{n-1} \\ b_4 & b_8 & & b_n \end{bmatrix}. \tag{2}$$

The projection of this array onto GF$(4)^{\frac{n}{4}}$ is obtained by taking the $\frac{n}{4}$ projections of the columns. A column will be said to be of type *even*, or simply even, if its Hamming weight is even. Otherwise, the column will be said to be *odd*. The sum of two columns is defined as the component-wise modulo-2 addition of the column's elements.

Finally, recall that for any two integers r and m satisfying $0 \le r \le m$, there is a binary rth-order Reed–Muller code $\mathcal{R}(r,\ m)$ with the following $[n,\ k,\ d]$ parameters: length $n = 2^m$; dimension $k = \sum_{i=0}^{r} \binom{m}{i}$; and minimum distance $d = 2^{m-r}$. Next, we construct some simple binary Reed–Muller codes based on the notation above.

$\mathcal{R}(m,\ m)$ contains all the vectors of length $n = 2^m$, i.e., it is the $[2^m,\ 2^m,\ 1]$ Universe code. Its definition is straightforward.

Proposition 1: The $\mathcal{R}(m,\ m)$ code, $m \ge 2$, is the set of all the $4 \times 2^{m-2}$ binary arrays whose projection is a codeword of the $[2^{m-2},\ 2^{m-2},\ 1]$ Universe code over GF(4).

The zero-order Reed–Muller code $\mathcal{R}(0,\ m)$ is the repetition code with parameters $[2^m,\ 1,\ 2^m]$.

Proposition 2: The $\mathcal{R}(0,\ m)$ code, $m \ge 2$, is the set of all the $4 \times 2^{m-2}$ binary arrays, such that each array satisfies the following conditions.

a) It consists of only even columns.

b) The projection of the array is the all-zero vector over GF$(4)^{2^{m-2}}$.

c) The top row is a codeword of the binary $\mathcal{R}(0,\ m-2)$ code.

It is easy to verify that Proposition 2 generates the all-zero and the all-one vectors of length 2^m by recalling that the even binary image of $\mathbf{0} \in$ GF(4) is either 0000 or 1111, and also that the two possible codewords of $\mathcal{R}(0,\ m-2)$ are the all-zero and the all-one vectors. The above propositions are easily generalized to all codes with length divisible by 4.

$\mathcal{R}(m-1,\ m)$ is a $[2^m,\ 2^m-1,\ 2]$ code consisting of all even-weight vectors. This family of single parity-check codes can be constructed as follows.

Proposition 3: The $\mathcal{R}(m-1,\ m)$ code, $m \ge 2$, is the set of all the $4 \times 2^{m-2}$ binary arrays, such that each array satisfies the following conditions:

a) the number of odd columns is even;

b) the projection of the array is a codeword of the $[2^{m-2},\ 2^{m-2},\ 1]$ Universe code over GF(4).

Proof: Follows immediately from the fact the this definition generates all possible binary words of even weight, due to conditions b) and a) respectively. □

An alternative approach to the proof is discussed in Section III. Note that the following may be regarded as an (additional) condition c) for Proposition 3: the top row is a codeword of the binary $\mathcal{R}(m-2,\ m-2)$ code. Since the $\mathcal{R}(m-2,\ m-2)$ is the Universe code $[2^{m-2},\ 2^{m-2},\ 1]$, this condition is redundant and was hence omitted.

Proposition 3 is easily extended for constructing all single parity-check codes of length $4i$, i being a positive integer. Each of these codes is defined as the set of all the $4 \times i$ binary arrays, such that each array satisfies the following conditions: a) the number of odd columns is even; b) the projection of the array is a codeword of the $[i,\ i,\ 1]$ Universe code over GF(4). The above proof also holds true for these codes.

III. $\mathcal{R}(r,\ m)$: Definition and Multilevel Construction

It is well known that Reed–Muller codes can be defined very simply in terms of Boolean polynomials. Let $v_1,\ v_2,\ \ldots,\ v_m$, be m binary variables and let f denote a Boolean polynomial in these m variables. $\boldsymbol{f}(f)$ is a vector of length 2^m corresponding to f which, in the standard bit ordering [11], is given by

$$\boldsymbol{f}(f) = (f(\mathbf{0}),\ f(\mathbf{1}),\ f(\mathbf{2}),\ \ldots,\ f(\mathbf{2^m - 1})). \tag{3}$$

For $\boldsymbol{n} = 0,\ 1,\ \ldots,\ 2^m - 1$, $(n_1,\ n_2,\ \ldots,\ n_m)$ is the binary expansion of length m of $\boldsymbol{n}$ satisfying

$$\boldsymbol{n} = \sum_{i=0}^{m-1} n_{i+1} 2^i$$

and $f(\boldsymbol{n})$ is the value of the polynomial f evaluated at

$$(v_1,\ v_2,\ \ldots,\ v_m) = (n_1,\ n_2,\ \ldots,\ n_m).$$

Definition [12]: The rth-order binary Reed–Muller code $\mathcal{R}(r,\ m)$ of length $n = 2^m$, for $0 \le r \le m$ is the set of all vectors $\boldsymbol{f}(f)$,

where f is a Boolean polynomial in m variables of degree at most r. In other words, if $\mathcal{R}_f(r, m)$ denotes the set of Boolean polynomial in m variables of degree at most r, then

$$\mathcal{R}(r, m) := \{\boldsymbol{f}(f) : f \in \mathcal{R}_f(r, m)\}.$$

The proposed construction is based on dividing the bit positions into groups of four as in (2), thus, (3) becomes

$$\begin{bmatrix} f(\mathbf{0}) & f(\mathbf{4}) & \cdots & f(4\boldsymbol{k}) & \cdots & f(\mathbf{2}^{\boldsymbol{m}}-\mathbf{4}) \\ f(\mathbf{1}) & f(\mathbf{5}) & \cdots & f(4\boldsymbol{k}+\mathbf{1}) & \cdots & f(\mathbf{2}^{\boldsymbol{m}}-\mathbf{3}) \\ f(\mathbf{2}) & f(\mathbf{6}) & \cdots & f(4\boldsymbol{k}+\mathbf{2}) & \cdots & f(\mathbf{2}^{\boldsymbol{m}}-\mathbf{2}) \\ f(\mathbf{3}) & f(\mathbf{7}) & \cdots & f(4\boldsymbol{k}+\mathbf{3}) & \cdots & f(\mathbf{2}^{\boldsymbol{m}}-\mathbf{1}) \end{bmatrix}. \quad (4)$$

This grouping amounts to fixing a certain value for the variables $v_3, v_4, \ldots, v_m$, and allowing the remaining coordinates v_1 and v_2 to take all the possible values. Thus, we immediately observe that: A) the variables in the first row of (4) are all multiples of 4, hence $v_1 = v_2 = 0$ for all of these variables; B) in the second row $v_1 = 1$, $v_2 = 0$; C) in the third row $v_1 = 0$, $v_2 = 1$; D) in the fourth row $v_1 = v_2 = 1$; E) in each of the four rows, the binary variables $v_3, v_4, \ldots, v_m$ take all the possible values.

As described in Section II, each column in (4) can be projected on GF(4), namely, obtain an element in GF(4) that may be represented by its coefficients with respect to the basis $\{\mathbf{1}, \boldsymbol{\alpha}\}$. Based on the proposed grouping and the representation over GF(4) (1), we shall now describe four different projection mappings.

- $\mathbf{TOP}$ the mapping of a block of four bits to its first element

$$\mathbf{TOP}(b_1, b_2, b_3, b_4) = b_1;$$

- $\boldsymbol{P_1}$ the coefficient of $\mathbf{1}$ in the GF(4)-domain

$$\boldsymbol{P_1}(b_1, b_2, b_3, b_4) = b_2 + b_4;$$

- $\boldsymbol{P_\alpha}$ the coefficient of $\boldsymbol{\alpha}$ in the GF(4)-domain

$$\boldsymbol{P_\alpha}(b_1, b_2, b_3, b_4) = b_3 + b_4;$$

- $\mathbf{PAR}$ the mapping of a block of four bits to its parity

$$\mathbf{PAR}(b_1, b_2, b_3, b_4) = b_1 + b_2 + b_3 + b_4.$$

Clearly, rather than operating on blocks of four bits, the above mappings may be equivalently defined for Boolean functions. For example, consider the mapping $\mathbf{TOP}$. Let A_m be the space of Boolean functions in m binary variables. Then, $\mathbf{TOP}: A_m \mapsto A_{m-2}$ is the mapping that takes $f(v_1, v_2, \ldots, v_m)$ to the Boolean function $\tilde{f}(v_3, v_4, \ldots, v_m)$ for which $\boldsymbol{f}(\tilde{f})$ is the top row of (4).

Let $w = v_{i_1} v_{i_2} \cdots v_{i_k}$ denote a monomial in the m variables $v_1, v_2, \ldots, v_m$, where k is at most r, and $\mathcal{I} \stackrel{\text{def}}{=} \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. Since the above mappings are obviously linear, it suffices to study their effect on the Boolean monomials w of degree at most r. Such monomials form a basis for the code $\mathcal{R}(r, m)$, therefore, from now on, we shall restrict our attention to Boolean vectors of the form

$$\boldsymbol{f}(w) = (w(\mathbf{0}), w(\mathbf{1}), w(\mathbf{2}), \ldots, w(\boldsymbol{n}), \ldots, w(\mathbf{2}^{\boldsymbol{m}} - 1)).$$

A. Derivation of the General Definition

The mapping $\mathbf{TOP}$ takes w in $v_1, v_2, v_3, \ldots, v_m$ to a monomial in $v_3, v_4, \ldots, v_m$ whose corresponding vector is the top row in the array representation (4) of $\boldsymbol{f}(w)$

$$\mathbf{TOP}(w) = w|_{v_1=0, v_2=0} = \begin{cases} w, & \text{if } \{v_1, v_2\} \not\subset \mathcal{I} \\ 0. & \text{otherwise.} \end{cases}$$

This, together with observation E above implies that all (and only) the monomials in $v_3, v_4, \ldots, v_m$ of degree at most r will be mapped to themselves, and the next lemma immediately follows.

Lemma 1: The image of $\mathcal{R}_f(r, m)$ under the mapping $\mathbf{TOP}$ is $\mathcal{R}_f(r, m-2)$.

The mapping $\boldsymbol{P_1}$ takes the monomial w in $v_1, v_2, v_3, \ldots, v_m$ to a monomial in $v_3, v_4, \ldots, v_m$ whose corresponding vector contains the coefficients of $\mathbf{1}$ in the GF(4) projection of the array representation of $\boldsymbol{f}(w)$. The coefficient of $\mathbf{1}$ in

$$b_1 \cdot \mathbf{0} + b_2 \cdot \mathbf{1} + b_3 \cdot \boldsymbol{\alpha} + b_4 \cdot (\mathbf{1} + \boldsymbol{\alpha})$$

is $b_2 + b_4$, corresponding to the second and fourth rows of the array representation. Therefore,

$$\boldsymbol{P_1}(w) = w|_{v_1=1, v_2=0} + w|_{v_1=1, v_2=1} = \begin{cases} 0 + \frac{w}{v_1 v_2} = \frac{w}{v_1 v_2}, & \text{if } \{v_1, v_2\} \subset \mathcal{I} \\ \frac{w}{v_1} + \frac{w}{v_1} = 0, & \text{if } v_1 \subset \mathcal{I},\ v_2 \not\subset \mathcal{I} \\ 0 + \frac{w}{v_2} = \frac{w}{v_2}, & \text{if } v_1 \not\subset \mathcal{I},\ v_2 \subset \mathcal{I} \\ w + w = 0, & \text{if } \{v_1, v_2\} \not\subset \mathcal{I}. \end{cases}$$

Clearly, this mapping yields all the monomials in $v_3, v_4, \ldots, v_m$ of degree at most $r-1$.

Lemma 2: The image of $\mathcal{R}_f(r, m)$ under the mapping $\boldsymbol{P_1}$ is $\mathcal{R}_f(r-1, m-2)$.

Similarly, the mapping $\boldsymbol{P_\alpha}$ takes the monomial w in $v_1, v_2, v_3, \ldots, v_m$ to a monomial in $v_3, v_4, \ldots, v_m$ whose corresponding vector contains the coefficients of $\boldsymbol{\alpha}$ in the GF(4) projection of the array representation of $\boldsymbol{f}(w)$. The coefficient of $\boldsymbol{\alpha}$ in

$$b_1 \cdot \mathbf{0} + b_2 \cdot \mathbf{1} + b_3 \cdot \boldsymbol{\alpha} + b_4 \cdot (\mathbf{1} + \boldsymbol{\alpha})$$

is $b_3 + b_4$. Therefore,

$$\boldsymbol{P}_\alpha(w) = w|_{v_1=0, v_2=1} + w|_{v_1=1, v_2=1} = \begin{cases} \frac{w}{v_1 v_2}, & \text{if } \{v_1, v_2\} \subset \mathcal{I} \\ \frac{w}{v_1}, & \text{if } v_1 \subset \mathcal{I},\ v_2 \not\subset \mathcal{I} \\ 0, & \text{otherwise.} \end{cases}$$

Once again, this mapping yields all the monomials in $v_3, v_4, \ldots, v_m$ of degree at most $r-1$.

Lemma 3: The image of $\mathcal{R}_f(r, m)$ under the mapping $\boldsymbol{P}_\alpha$ is $\mathcal{R}_f(r-1, m-2)$.

The mapping $\mathbf{PAR}$ takes w in $v_1, v_2, v_3, \ldots, v_m$ to a monomial in $v_3, v_4, \ldots, v_m$ whose corresponding vector is the parity image of the array representation (4) of $\boldsymbol{f}(w)$. (The parity image of the array is a vector of length 2^{m-2} whose elements are either 0 or 1 in accordance

TABLE I
DEFINITION OF $\mathcal{R}(r, m)$ CODES

Code	Code Parameters	parity image	Projection	Top row parity
$\mathcal{R}(0,m)$	$[2^m, 1, 2^m]$	$0^{m-2} \in GF(2)$	$0^{m-2} \in GF(4)$	$\mathcal{R}(0,m-2)$
$\mathcal{R}(1,m)$	$[2^m, 1+m, 2^{m-1}]$	$0^{m-2} \in GF(2)$	$\mathcal{R}_4(0,m-2)$	$\mathcal{R}(1,m-2)$
$\mathcal{R}(r,m)$	$[2^m, \sum_{i=0}^{r}\binom{m}{i}, 2^{m-r}]$	$\mathcal{R}(r-2,m-2)$	$\mathcal{R}_4(r-1,m-2)$	$\mathcal{R}(r,m-2)$
$\mathcal{R}(m-1,m)$	$[2^m, 2^m-1, 2]$	$\mathcal{R}(m-3,m-2)$	$\mathcal{R}_4(m-2,m-2)$	$\mathcal{R}(m-2,m-2)$
$\mathcal{R}(m,m)$	$[2^m, 2^m, 1]$	$\mathcal{R}(m-2,m-2)$	$\mathcal{R}_4(m-2,m-2)$	$\mathcal{R}(m-2,m-2)$

with the parity of the corresponding column.) Thus, $\mathbf{PAR}(w) = b_1 + b_2 + b_3 + b_4$, corresponding to the sum of the four elements of a column

$$\begin{aligned}\mathbf{PAR}(w) &= w|_{v_1=0,\,v_2=0} + w|_{v_1=1,\,v_2=0} \\ &\quad + w|_{v_1=0,\,v_2=1} + w|_{v_1=1,\,v_2=1} \\ &= \begin{cases} \frac{w}{v_1 v_2}, & \text{if } \{v_1, v_2\} \subset \mathcal{I} \\ 0, & \text{otherwise.} \end{cases}\end{aligned}$$

This mapping yields all the monomials in $v_3, v_4, \ldots, v_m$ of degree at most $r-2$.

Lemma 4: The image of $\mathcal{R}_f(r, m)$ under the mapping $\mathbf{PAR}$ is $\mathcal{R}_f(r-2, m-2)$.

Since $w = \prod_{i \in I(w)} v_i$ is a monomial in $v_1, v_2, \ldots, v_m$, where $I(w)$ is a nonempty subset of $\{1, 2, \ldots, m\}$, and by defining J as the intersection of $I(w)$ with $\{1, 2\}$, the action of the above mappings on w can be summarized as

	$J=\{1,2\}$	$J=\{1\}$	$J=\{2\}$	$J=\{\}$
$\mathbf{TOP}(w)$	0	0	0	w
$\boldsymbol{P_1}(w)$	$\frac{w}{(v_1 v_2)}$	0	$\frac{w}{v_2}$	0
$\boldsymbol{P_\alpha}(w)$	$\frac{w}{(v_1 v_2)}$	$\frac{w}{v_1}$	0	0
$\mathbf{PAR}(w)$	$\frac{w}{(v_1 v_2)}$	0	0	0

From this, and as a corollary of Lemmas 1 through 4, we see that the mapping

$$\mathbf{ID} = \mathbf{TOP} + v_2\boldsymbol{P_1} + v_1\boldsymbol{P_\alpha} + (v_1 + v_2 + v_1 v_2)\,\mathbf{PAR} \quad (5)$$

is the identity mapping. This means that the images under the four projection mappings uniquely determine the preimage. Let $G_{\mathcal{R}(r,m)}$ denote the generator matrix for the code $\mathcal{R}(r, m)$, and let the combined image of the mappings $\boldsymbol{P_1}$ and $\boldsymbol{P_\alpha}$ be defined as

$$\{\mathbf{1}\cdot\boldsymbol{b}\cdot G_{\mathcal{R}(r-1,m-2)} + \boldsymbol{\alpha}\cdot\boldsymbol{a}\cdot G_{\mathcal{R}(r-1,m-2)} :\ \boldsymbol{a}, \boldsymbol{b} \in \mathrm{GF}(2)^k\} \quad (6)$$

with the operations suitably interpreted over GF(4), and where k is the dimension of $\mathcal{R}(r-1, m-2)$. It follows from Lemmas 2 and 3 and the definition of $\boldsymbol{P_1}$ and $\boldsymbol{P_\alpha}$, that the combined image of the mappings $\boldsymbol{P_1}$ and $\boldsymbol{P_\alpha}$ (6) is identical to the projection of the columns of (4) onto GF(4) as defined in Section II. The next corollary immediately follows.

Corollary 1: The projection of the code $\mathcal{R}(r, m)$, i.e., the set of quaternary vectors obtained by projecting all the codewords of $\mathcal{R}(r, m)$ on GF$(4)^{2^{m-2}}$, is a quarternary linear code whose generator matrix is $G_{\mathcal{R}(r-1,m-2)}$.

This quaternary code will, therefore, be denoted by $\mathcal{R}_4(r-1, m-2)$.

Theorem 1: The $\mathcal{R}(r, m)$ code, $r > 1$ and $m > r+1$, is the set of all the $4 \times 2^{m-2}$ binary arrays, such that each array satisfies the following conditions.

a) The parity-image of the array is a codeword of the $\mathcal{R}(r-2, m-2)$ code.

b) The projection of the array is a codeword of the $\mathcal{R}_4(r-1, m-2)$ code.

c) The top row is a codeword of the $\mathcal{R}(r, m-2)$ code.

Proof: The proof follows from (5), Lemmas 1 and 4, and Corollary 1. □

Theorem 1 does not apply to first-order Reed–Muller codes. However, the following construction is an immediate corollary to the properties of the four projection mappings.

Corollary 2: The $\mathcal{R}(1, m)$ code, $m > 2$, is the set of all the $4 \times 2^{m-2}$ binary arrays, such that each array satisfies the following conditions.

a) It consists of only even columns.

b) The projection of the array is a codeword of a $\mathcal{R}_4(0, m-2)$ code.

c) The top row is a codeword of the $\mathcal{R}(1, m-2)$ code.

Proof: The proof follows from (5), Lemma 1, and Corollary 1, and the fact that first-order Reed–Muller codes include only monomials of maximum degree one, thus, $\mathbf{PAR}(w) = 0$ for any monomial w. □

Similarly, for $\mathcal{R}(m-1, m)$, Proposition 3 follows from the fact that this code contains all the monomials of degree less than r. Thus, the mapping $\mathbf{TOP}$ generates all the monomials in $v_3, v_4, \ldots, v_m$ of degree at most $r-2$, which from the array representation (4) clearly amounts to the Universe code $[2^{m-2}, 2^{m-2}, 1] = \mathcal{R}(m-2, m-2)$. The obtained results are summarized in Table I.

Remark: It is well known that all Reed–Muller codes of length greater than four can be obtained by applying the $(u|u+v)$ construction recursively from the codes $\mathcal{R}(0, 2)$, $\mathcal{R}(1, 2)$, and $\mathcal{R}(2, 2)$. Though we originally derived Theorem 1 by employing this recursive construction, the current approach was finally adopted, as it is more elegant. In the Appendix, we show how Corollary 1 can be derived from the $(u|u+v)$ construction.

B. Multilevel Construction of the Reed–Muller Codes

While the four projection mappings do not result in a multilevel construction, combining the images of $\boldsymbol{P_1}$ and $\boldsymbol{P_\alpha}$ yields the definition of Reed–Muller codes given by Theorem 1, which indeed results in the (true) multilevel construction illustrated in Fig. 1.

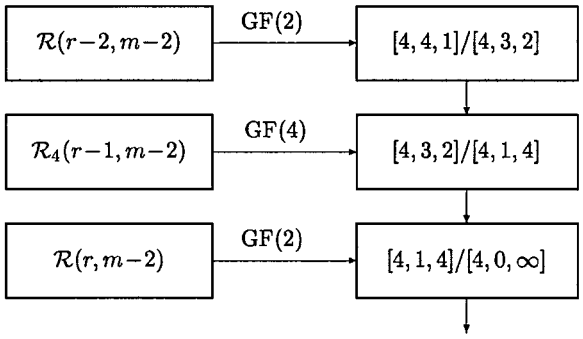

Fig. 1. Multilevel construction of Reed–Muller codes $\mathcal{R}(r,\ m)$, $1 < r < m-1$.

This construction is based on the partition chain of the four-dimensional binary codes $[4,\ 4]/[4,\ 3]/[4,\ 1]/[4,\ 0]$ with Hamming distances $1/2/4/\infty$. It is now shown that this construction is equivalent to that of Theorem 1. The two-way partition $[4,\ 4]/[4,\ 3]$ at the first level distinguishes between the odd and even columns. According to condition 1 part a), the parity image of the array, i.e., the positions of the odd and even columns, is a codeword of $\mathcal{R}(r-2,\ m-2)$. Hence, the $\mathcal{R}(r-2,\ m-2)$ is applied to the partition $[4,\ 4]/[4,\ 3]$ at the first level. The four-way partition $[4,\ 3]/[4,\ 1]$ distinguishes between the binary images of the four symbols $\mathbf{0}$, $\mathbf{1}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ over GF(4). According to condition 1 part b), the projection of the array is a codeword of the quaternary code $\mathcal{R}_4(r-1,\ m-2)$. Hence, the $\mathcal{R}_4(r-1,\ m-2)$ is applied to the partition $[4,\ 3]/[4,\ 1]$ at the second level. At the third level, the two-way partition $[4,\ 1]/[4,\ 0]$ determines whether the top row entry in a column is 0 or 1, or equivalently, it chooses between two complementary binary four-tuples. According to condition 1 part c), the parity of the top row of the array is a codeword of $\mathcal{R}(r,\ m-2)$. Hence, the code $\mathcal{R}(r,\ m-2)$ is applied to the partition $[4,\ 1]/[4,\ 0]$ at the third level.

Another representation for this construction is by means of the code formula notation. Consider the following set of coset representatives for the partition $[4,\ 4]/[4,\ 3]/[4,\ 1]/[4,\ 0]$:

$$G_{[4,\,4]/[4,\,3]} = [\,0 \quad 1 \quad 1 \quad 1\,]$$
$$G_{[4,\,3]/[4,\,1]} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$
$$G_{[4,\,1]/[4,\,0]} = [\,1 \quad 1 \quad 1 \quad 1\,].$$

Based on the foregoing arguments, it is apparent that the $\mathcal{R}(r,\ m)$ code consists of all the vectors of the form

$$\boldsymbol{c_1} \otimes G_{[4,\,4]/[4,\,3]} + \boldsymbol{c_2} \otimes G_{[4,\,3]/[4,\,1]} + \boldsymbol{c_3} \otimes G_{[4,\,1]/[4,\,0]}$$

where $\otimes$ stands for the Kronecker product, $\boldsymbol{c_1} \in \mathcal{R}(r-2,\ m-2)$, $\boldsymbol{c_2} \in \mathcal{R}_4(r-1,\ m-2)$, and $\boldsymbol{c_3} \in \mathcal{R}(r,\ m-2)$ (with the convention that the quaternary elements of the vector $\boldsymbol{c_2}$ are mapped to binary pairs as follows: $\mathbf{0} \mapsto 00$, $\mathbf{1} \mapsto 01$, $\alpha \mapsto 10$ and $\beta \mapsto 11$).

IV. Further Considerations and Conclusion

A multilevel construction for binary Reed–Muller codes $\mathcal{R}(r,\ m)$ has been presented based upon the projection of the codewords of $\mathcal{R}(r,\ m)$ onto a linear quaternary code with the same parameters and generator matrix as the binary Reed–Muller code $\mathcal{R}(r-1,\ m-2)$. The other two codes applied to the first and third levels of the construction are the binary Reed–Muller codes, $\mathcal{R}(r-2,\ m-2)$ and $\mathcal{R}(r,\ m-2)$, respectively. As argued before, this construction is readily applicable for efficient maximum-likelihood or multistage bounded distance decoding. This will be demonstrated by means of some examples.

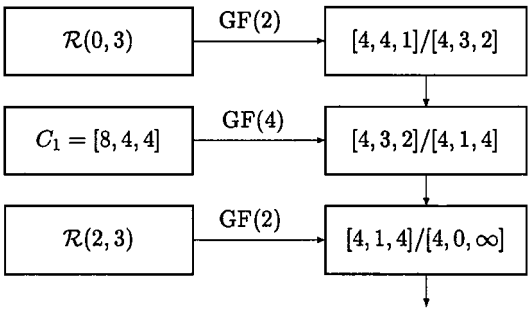

Fig. 2. Multilevel construction of $[32,\ 16,\ 8]\mathcal{R}(2,\ 5)$.

First-order $\mathcal{R}(1,\ m)$ codes may be decoded in a straightforward manner based on the construction of Corollary 2. A brief description of the decoding steps follows. (*Precomputation*) Let us assume an additive white Gaussian noise (AWGN) channel model, and let a sequence of n symbols be observed at the channel output. Arrange the received sequence in a two-dimensional $4 \times \frac{n}{4}$ array S corresponding to (4). For each $\mu \in$ GF(4) and for each of the columns S_i of S compute the metric of the *even representation* of μ [17]. i) For each of the four codewords of $\mathcal{R}_4(0,\ m-2)$, the repetition code over GF(4), find the binary array whose top row is the codeword of $\mathcal{R}(1,\ m-2)$ with the minimum overall metric. ii) Among the four arrays thus obtained, select the one with the minimum metric as the output of the decoder. The decoding complexity $N_{\mathcal{R}(1,\,m)}$ associated with this algorithm can now be easily evaluated recursively

$$N_{\mathcal{R}(1,\,m)} = 10 \cdot 2^{m-2} + 4 \cdot N_{\mathcal{R}(1,\,m-2)} + 3$$

where 10 is the number of real-number operations required for computing the metric of each and every $\mu \in$ GF(4) per coordinate (employing the Gray mapping principle); and $4 \cdot N_{\mathcal{R}(1,\,m-2)}$ is obviously the complexity of decoding step 1. (Note that this may be regarded as an upper bound on the true decoding complexity associated with our construction because we have employed a "brute-force" approach for the complexity evaluation.) For example, decoding the $\mathcal{R}(1,\ 3) = [8,\ 4,\ 4]$ code requires 20 real-number operations for computing the different metrics in the *Precomputation* step, and $4+3$ additional operations for computing the overall metrics of the four arrays (Step i)) and for selecting the best one (Step ii)). Therefore, in a straightforward manner, using the above recursive evaluation, we get $N_{\mathcal{R}(1,\,3)} = 27$, $N_{\mathcal{R}(1,\,5)} = 191$, $N_{\mathcal{R}(1,\,7)} = 1087$. This is comparable to 23, 223, and 1151 operations, respectively, associated with the trellis-based decoder of Forney [7]. A more interesting example is the second-order $\mathcal{R}(2,\ 5) = [32,\ 16,\ 8]$. Along the lines of [17], maximum-likelihood decoding of the $\mathcal{R}(2,\ 5)$ code requires $\sim$ 2000 binary real-number operations as compared to $\sim$3580 operations required for trellis decoding [7]. Along the lines of [1], [3] bounded distance soft decoding of $\mathcal{R}(2,\ 5)$ may be performed with as few as $\sim$320 operations.

While focusing primarily on Reed–Muller codes, we note that similar constructions may be obtained for other interesting codes. For example, Conway and Pless [4] enumerated five nonequivalent binary codes with parameters $[32,\ 16,\ 8]$. At least two of these codes, i.e., the $\mathcal{R}(2,\ 5)$ and the extended quadratic residue (QR) codes, can be defined over GF(4) in very much the same way. $\mathcal{R}(2,\ 5)$ has a true multilevel construction as depicted in Fig. 2, with C_1 being the $[8,\ 4,\ 4]$ quaternary code $\mathcal{R}_4(1,\ 3)$. The QR code is obtained by replacing C_1

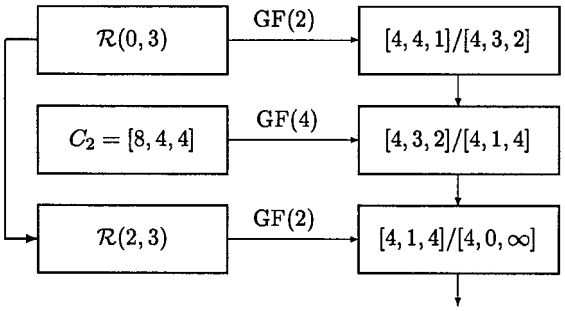

Fig. 3. Multilevel construction of $[32,\ 16,\ 8]$ quadratic residue (QR) code.

with another $[8,\ 4,\ 4]$ quaternary code [18], and by introducing some dependency between the first and last levels. The line connecting the levels in Fig. 3 depicts this fact. The code employed at the last level is either the $[8,\ 7,\ 2]$ even-weight code or a coset of the latter (containing only odd-weight codewords), and is determined by the codeword in the first level. It is plausible that the other three nonequivalent $[32,\ 16,\ 8]$ codes may be constructed in a similar manner.

APPENDIX

Corollary 1 is now derived by invoking the $(u|u+v)$ construction. This derivation requires the following lemmas. Let φ denote the projection mapping taking a binary linear code of length $4n$ to a quaternary code of length n as defined in the correspondence.

Lemma 5: The images of the $(4,\ 4,\ 1)$, $(4,\ 1,\ 4)$, and $(4,\ 3,\ 2)$ binary codes are GF(4)-linear.

Proof: By Proposition 1, Proposition 2 part b) and Proposition 3 part b), respectively. □

Lemma 6: If U and V are binary linear codes whose images are GF(4)-linear, then the code $\varphi(U+V)$ is GF(4)-linear.

Proof: $U+V$ is a vector space, and the mapping φ is a homomorphism with respect to addition, so $\varphi(U+V)$ forms a subgroup of the additive group in GF$(4)^n$. It remains to show that $\varphi(U+V)$ is closed under scalar multiplication. Clearly, $0\cdot x \in \varphi(U+V)$. It suffices to show that for all $x \in \varphi(U+V)$, αx is also in $\varphi(U+V)$, since then, by addition, $x+\alpha x = \beta x \in \varphi(U+V)$. But if $x \in \varphi(U+V)$, then $x=\varphi(u+v)$ for some $u \in U,\ v \in V$. By the homomorphism property, $\varphi(u+v)=\varphi(u)+\varphi(v)$. Since $\varphi(U)$ is GF(4)-linear, there exists $u' \in U$ such that $\varphi(u')=\alpha\varphi(u)$. Likewise, there exists $v' \in V$ such that $\varphi(v')=\alpha\varphi(v)$. It follows that $\varphi(u'+v')=\alpha\varphi(u)+\alpha\varphi(v)=\alpha x$ is an element of $\varphi(U+V)$. □

Lemma 7: If U is a binary linear code whose image is GF(4)-linear, then the code $(U|U)=\{(u,\ u){:}u \in U\}$ is GF(4)-linear. Likewise, the code $(0|V)=\{(0,\ v){:}v \in V\}$ is GF(4)-linear.

Lemma 7 is easy to prove by construction. Combining Lemmas 6 and 7 we see that if U and V are codes whose images are GF(4)-linear, then $(U|U+V)$ is a code whose image is GF(4)-linear. Since all Reed–Muller codes of length larger than four can be obtained by applying the $(U|U+V)$ construction recursively from the codes of Lemma 5, it follows that the images of all Reed–Muller codes (with bit ordering obtained from the recursive construction) are GF(4)-linear.

Now let $G(\varphi(U))$ be a generator matrix for $\varphi(U)$ and $G(\varphi(V))$ be a generator matrix for $\varphi(V)$, then

$$\begin{bmatrix} G(\varphi(U)) & G(\varphi(U)) \\ 0 & G(\varphi(V)) \end{bmatrix}$$

is clearly a generator matrix for $\varphi(U|U+V)$, with $G(\varphi(U))=0$ when $U=(4,\ 0,\ -)$ or $U=(4,\ 1,\ 4)$, and $G(\varphi(U))=1$ when $U=(4,\ 3,\ 2)$ or $U=(4,\ 4,\ 1)$. From the recursive $(u|u+v)$ construction of $\mathcal{R}(r,\ m)$ it follows that the generator matrix for the quaternary image is the same as $\mathcal{R}(r-1,\ m-2)$ and Corollary 1 is obtained.

ACKNOWLEDGMENT

The authors wish to thank S. Litsyn, O. Keren, and Y. Shany for stimulating discussions. The referees are gratefully acknowledged for their helpful comments. In particular, the authors are in debt to J. Lahtonen for his insightful comments that greatly improved the presentation of the results.

REFERENCES

[1] O. Amrani and Y. Be'ery, "Efficient bounded-distance decoding of the hexacode and associated decoders for the Leech lattice and the Golay code," *IEEE Trans. Commun.*, vol. 44, pp. 534–537, Apr. 1996.

[2] ——, "On representing the Nordstrom–Robinson code over GF(4)," Tel-Aviv Univ., Tel-Aviv, Israel, Tech. Rep. EE-S-98-46, Sept. 1998.

[3] ——, "Bounded-distance decoding: Algorithms, decision regions, and pseudo nearest neighbors," *IEEE Trans. Inform. Theory*, vol. 44, pp. 3072–3082, Nov. 1998.

[4] J. H. Conway and V. Pless, "On the enumeration of self-dual codes," *J. Combin. Theory*, ser. A 28, pp. 26–53, 1980.

[5] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1993.

[6] G. D. Forney, Jr., "Coset codes—Part I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123–1151, Sept. 1988.

[7] ——, "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.

[8] G. D. Forney, Jr., N. J. A. Sloane, and M. D. Trott, "The Nordstrom–Robinson code is the binary image of the Octacode," in *Proc. DIMACS/IEEE Worksh. Coding and Quantization*, vol. 14, 1993, pp. 19–26.

[9] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Sole, "The Z_4 linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 301–319, Mar. 1994.

[10] X. Hou, J. T. Lahtonen, and S. Koponen, "The Reed–Muller code $R(r,m)$ is not Z_4-linear for $3 \le r \le m-2$," *IEEE Trans. Inform. Theory*, vol. 44, pp. 798–799, Mar. 1998.

[11] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 242–245, Jan. 1993.

[12] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1997.

[13] V. Pless, "Decoding the Golay codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 561–567, July 1986.

[14] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, Aug. 1998.

[15] A. Vardy, "The Nordstrom–Robinson code: Representation over GF(4) and efficient decoding," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1686–1693, Sept. 1994.

[16] ——, "Even more efficient bounded-distance decoding of the hexacode, the Golay code, and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1495–1499, Sept. 1995.

[17] A. Vardy and Y. Be'ery, "More efficient soft-decision decoding of the Golay codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 667–672, May 1991.

[18] Y. Yuan, C. S. Chen, and S. Ma, "Two-level decoding of (32,16,8) quadratic residue code," *Proc. Inst. Elec. Eng. Pt. 1*, vol. 140, pp. 409–414, 1993.