Peak-to-Mean Power Control in OFDM, Golay Complementary Sequences and Reed-Muller Codes

James A. Davis*, Jonathan Jedwab
Networked Systems Department
HP Laboratories Bristol

HPL-97-158
December, 1997

E-mail: jij@hplb.hpl.hp.com

power, envelope, OFDM, Golay, complementary, sequence, Reed-Muller, code

We present a range of coding schemes for OFDM transmission at high code rates using binary, quaternary, octary and higher-order modulation. These schemes have tightly bounded peak-to-mean envelope power ratio (PMEPR) and simultaneously have good error correction capability. The key theoretical result is a previously unrecognised connection between Golay complementary sequences and second-order Reed-Muller codes over alphabets $\mathbb{Z}_{2^h}$. We obtain additional flexibility in trading off code rate, PMEPR and error correction capability by partitioning the second-order Reed-Muller code into cosets such that codewords with large values of PMEPR are isolated. For all the proposed schemes, we show that encoding is straightforward and give an efficient decoding algorithm involving multiple fast Hadamard transforms. Since the coding schemes are all based on the same formal generator matrix, we can deal adaptively with varying channel constraints and evolving system requirements.

*Department of Mathematical and Computer Science, University of Richmond, Richmond, Virginia

1 The envelope power problem in OFDM transmission

Orthogonal frequency division multiplexing (OFDM) is a method of transmitting data simultaneously over multiple equally-spaced carrier frequencies, using Fourier transform processing for modulation and demodulation [10]. The method has been proposed or adopted for many types of radio systems such as wireless Local Area Networks [2] and digital audio and digital video broadcasting [1, 39]. OFDM offers many well-documented advantages for multicarrier transmission at high data rates, particularly in mobile applications. Specifically, it has inherent resistance to dispersion in the propagation channel [6]. Furthermore when coding is added it is possible to exploit frequency diversity in frequency selective fading channels to obtain excellent performance under low signal-to-noise conditions [38]. For these reasons OFDM is often preferable to constant envelope modulation with adaptive equalisation (and indeed is arguably less complex to implement [29]).

The principal difficulty with OFDM is that when the sinusoidal signals of the n carriers add mostly constructively the peak envelope power is as much as n times the mean envelope power. If the peak envelope power is subject to a design or regulatory limit then this has the effect of reducing the mean envelope power allowed under OFDM relative to that allowed under constant envelope modulation. If battery power is a constraint, as is typically the case with portable equipment, then the power amplifiers required to behave linearly up to the peak envelope power must be operated inefficiently (with considerable backoff from compression). Digital hard limiting of the transmitted signal has been shown to alleviate the problem [26], but only at the cost of spectral sidelobe growth and consequent performance degradation.

This gives a clear motivation to find other ways of controlling the peak-to-mean envelope power ratio (PMEPR) of the transmitted signal. A promising method which has attracted considerable interest, introduced in [25] and developed in [45], is to use block coding to transmit across the carriers only those polyphase sequences with small PMEPR. As originally described, this entails exhaustive search to identify the best sequences and requires large look-up tables for encoding and decoding. Several authors, for example [15, 46], have proposed simpler implementations of this method using systematic (or at least constrained) methods of coding. Nonetheless [15] declares that "...there are no known rules concerning selection of the allowed signals [having PMEPR below a certain threshold] in a structured way". Moreover these schemes do not address the problem of error correction at all. An alternative method [24] instead takes the transmitted codewords from a coset of a linear error-correcting code, choosing the coset representative or "mask vector" by computationally intensive search in order to reduce the PMEPR. In this way the error correction properties are assured but the appropriate choice of linear code and coset representative for optimal PMEPR remains an open problem.

In this paper we present a highly flexible coding scheme for binary, quaternary, octary and higher-order modulation which incorporates aspects of both of the above methods. It uses theoretical considerations to guarantee low PMEPR and simultaneously to provide good error correction capability. It allows simple changes to properties such as code rate, PMEPR and error correction capability to deal adaptively with varying channel constraints, and provides a clear evolution path for physical systems from binary to quaternary to octary modulation. In all cases we provide straightforward and efficient algorithms for encoding and decoding.

The remainder of the paper is structured as follows. Section 2 motivates the use of Golay sequences (*i.e.* sequences belonging to Golay complementary pairs) as a first solution to the envelope power problem in OFDM. We explicitly determine a large class of Golay sequences over $\mathbb{Z}_{2^h}$ of length 2^m in terms of generalised Boolean functions. Section 3 shows that in the binary case $h = 1$, these Golay sequences occur as cosets of the first-order Reed-Muller code within the second-order Reed-Muller code. This connection between Golay sequences and Reed-Muller codes has not previously been recognised, and is a key result leading to the practical and flexible OFDM coding schemes of this paper. For the non-binary cases $h > 1$ we introduce two new linear codes over the ring $\mathbb{Z}_{2^h}$ as generalisations of the Reed-Muller code and demonstrate a corresponding connection with the non-binary Golay sequences previously determined. We establish the minimum Hamming and Lee distance of these new codes as measures of their error correction capability. Section 4 proposes an OFDM coding scheme, based on the Golay sequences of Section 2, involving cosets of one generalised Reed-Muller code within another. We then show that by varying the set of cosets of the first generalised Reed-Muller code within the second we can obtain a much more general range of solutions to the envelope power problem, not necessarily restricted to Golay sequences. In this way we can make trade-offs between PMEPR, code rate, and error correction capability. The essential observation is that partitioning the second-order Reed-Muller code into cosets in this way appears naturally to isolate those codewords with large values of PMEPR. Section 5 presents highly efficient decoding algorithms for all of the proposed coding schemes. These algorithms apply the fast Hadamard transform repeatedly in a novel manner. For background on classical coding theory, see [27] or [28].

Some of the results of this paper, in particular the connection between Golay sequences and second-order Reed-Muller codes, were announced without proof in [12]. An overview of the coding schemes of this paper, including Tables 4, 5 and 6, was given in [22]. There is limited overlap between the results in Sections 2 and 3 of this paper and recent independent work on OFDM. Translated into the notation of the present paper, van Nee [32] essentially shows how to obtain recursively a subset of the Golay sequences of Corollary 2.4 corresponding to m cosets of $\mathrm{RM}_{2^h}(1,m)$, and Ochiai and Imai [33] do likewise but for a subset corresponding to a single coset rather than to m. In contrast Corollaries 3.1 and 3.4 explicitly identify $m!/2$ such cosets within a specified linear code, and Theorem 2.3 and Corollary 2.5 show how to arrange the identified sequences into Golay complementary pairs. Moreover [32] and [33] do not make the crucial connection between Golay sequences and Reed-Muller codes and consequently do not identify the range of coding options presented here and their attendant advantages. We also note that the claim of [33], that the method announced in [12] "...seems to be valid only for binary sequences", is emphatically incorrect; in fact the principal example of [33], contained in equations (16) and (18) of that paper, consists of the quaternary length 8 sequences $2(x_1x_2+x_2x_3)+\sum_{k=1}^{3}c_kx_k+c$ for $c, c_k \in \mathbb{Z}_4$, which occur as a simple special case of Theorem 2 of [12].

2 Golay sequences

We represent the value assigned to the ith carrier of an OFDM system during a given symbol period as an element a_i of the ring $\mathbb{Z}_H$ for some $H \geq 2$, where $i = 0, 1, \dots, n-1$. In each symbol period, the $\mathbb{Z}_H$-ary sequence $(a_0, a_1, \dots, a_{n-1})$ across the n carriers forms a codeword. Codewords in successive symbol periods belong to a code whose alphabet is $\mathbb{Z}_H$, and in the cases $H = 2$, 4 or 8 the code is called binary, quaternary or octary respectively. In signal processing, it is more common to consider the sequence of complex modulated values $(\xi^{a_0}, \xi^{a_1}, \dots, \xi^{a_{n-1}})$, where $\xi = \exp(2\pi\sqrt{-1}/H)$ is a primitive H-th root of unity. (In some implementations this sequence is multiplied by the constant $\exp(\pi\sqrt{-1}/H)$.) This modulation is called H-phase shift keying, which in the cases $H = 2$ or 4 is also known as binary phase shift keying or quadrature phase shift keying respectively.

The transmitted OFDM signal is the real part of the complex envelope

$$s(t) = \sum_{i=0}^{n-1} \xi^{a_i(t)+Hf_it}, \tag{1}$$

where f_i is the frequency of the ith carrier and $a_i(t)$ is constant over a symbol period. In order to ensure orthogonality the carrier frequencies are related by

$$f_i = f + i\Delta f \tag{2}$$

for some constant f, where Δf is an integer multiple of the OFDM symbol rate. The *instantaneous envelope power* of the signal is the real-valued function $P(t) = |s(t)|^2$, and substitution from (1) and (2) gives

$$P(t) = \sum_{i,j} \xi^{a_i(t)-a_j(t)+H(i-j)\Delta ft}. \tag{3}$$

Let the constant value of $a_i(t)$ over a symbol period such as $0 \leq \Delta ft \leq 1$ be a_i, and call the resulting continuous function $P(t)$ over the symbol period the *envelope power* $P_a(t)$ *of the sequence* $a = (a_0, a_1, \ldots, a_{n-1})$. Then by putting $j = i + u$ in the expression for $P_a(t)$ given by (3) we obtain

$$P_a(t) = n + \sum_{u \neq 0} \sum_{i} \xi^{a_i - a_{i+u} - Hu\Delta ft}, \tag{4}$$

where here and in (5) below the summations are understood to be over only those integer values for which both i and $i + u$ lie within $\{0, 1, \ldots, n-1\}$. Since the *aperiodic autocorrelation of* a *at displacement* u is by definition

$$C_a(u) = \sum_{i} \xi^{a_i - a_{i+u}}, \tag{5}$$

we can rewrite (4) as

$$P_a(t) = n + \sum_{u \neq 0} C_a(u) \xi^{-Hu\Delta ft}. \tag{6}$$

The *peak envelope power* (PEP) of the sequence a is the supremum over a symbol period of $P_a(t)$. From (5) and (6), the mean envelope power of any sequence a over a symbol period is n, and so the *peak-to-mean envelope power ratio* (PMEPR) of a is the ratio PEP$/n$. Alternative names for PMEPR are *peak-to-average power ratio* [30] and *peak factor* [42]; the square root of the PMEPR is called the *crest factor* [7]. A PMEPR of R is often expressed as $10 \log_{10} R$ dB. From (6) we see that $P_a(t) \leq n + \sum_{u \neq 0} |C_a(u)| \cdot 1 \leq n + 2 \sum_{u=1}^{n-1} (n - u) = n^2$, so the PEP of any sequence a is at most n^2 and the PMEPR is at most n. (See [42] for a similar argument giving a general upper bound on the PEP of a in terms of $C_a(u)$, and [16] for the derivation of a lower bound on the PMEPR of a from (6).)

The upper bound of n for PMEPR is attained by the sequence $a = (0, 0, \ldots, 0)$, which can occur in an uncoded OFDM system. But by restricting the set of allowed sequences to Golay sequences we can reduce the PMEPR from its maximum value of n to at most 2, as we now show.

Definition 2.1 *Let* $a = (a_0, a_1, \ldots, a_{n-1})$ *and* $b = (b_0, b_1, \ldots, b_{n-1})$, *where* $a_i, b_i \in \mathbb{Z}_H$. *The sequences* a *and* b *are called a* Golay complementary pair over $\mathbb{Z}_H$ of length n *if* $C_a(u) + C_b(u) = 0$ *for each* $u \neq 0$. *Any sequence which is a member of a Golay complementary pair is called a* Golay sequence.

Theorem 2.2 *The PMEPR of any Golay sequence is at most* 2.

Proof: Let a and b be a Golay complementary pair, so that by definition $C_a(u) + C_b(u) = 0$ for each $u \neq 0$. Then from (6), $P_a(t) + P_b(t) = 2n$ and since $P_b(t) = |s_b(t)|^2 \geq 0$ we deduce $P_a(t) \leq 2n$. The result follows from the definition of PMEPR. □

Theorem 2.2 was obtained by Popović [36] (in terms of the crest factor of the real-valued signal envelope) by generalising earlier work of Boyd [7]. Golay complementary pairs over $\mathbb{Z}_2$ were introduced by Golay [17, 18] in connection with infrared multislit spectrometry and have since found application in fields such as optical time domain reflectometry [31] and acoustic surface-wave encoding [43]. They are known to exist for all lengths $n = 2^\alpha 10^\beta 26^\gamma$, where $\alpha, \beta, \gamma \geq 0$ [44], but do not exist for any length n having a prime factor congruent to 3 modulo 4 [13]. For a survey of results on non-binary Golay complementary pairs, see [14, Chap. 13].

Henceforth we impose the restriction $n = 2^m$ so that the sampled OFDM signal corresponding to the continuous function (1) can be easily generated using the inverse fast Fourier transform. We also assume that $H = 2^h$ for some $h \geq 1$ and then in each symbol period the OFDM signal contains exactly h code bits per carrier. We now give an explicit form for a large class of Golay complementary pairs over $\mathbb{Z}_{2^h}$ of length 2^m, and deduce the form of a set of Golay sequences. We first require some notation.

A *Boolean function* is a function f from $\mathbb{Z}_2^m = \{(x_1, x_2, \ldots, x_m) \mid x_i \in \{0,1\}\}$ to $\mathbb{Z}_2$. We regard each 0-1 variable x_i as itself being a Boolean function $f_i(x_1, x_2, \ldots, x_m) = x_i$ and consider the 2^m monomials

$$1,\ x_1, x_2, \ldots, x_m,\ x_1x_2, x_1x_3, \ldots, x_{m-1}x_m,\ \ldots,\ x_1x_2\cdots x_m. \tag{7}$$

Any Boolean function f can be uniquely expressed as a linear combination over $\mathbb{Z}_2$ of these monomials, where the coefficient of each monomial belongs to $\mathbb{Z}_2$ [28]. The resulting expression for f is called the *algebraic normal form* [37]. We specify a sequence $\mathbf{f}$ of length 2^m corresponding to f by listing the values taken by $f(x_1, x_2, \ldots, x_m)$ as $(x_1, x_2, \ldots, x_m)$ ranges over all its 2^m values in lexicographic order. In other words, if $(i_1, i_2, \ldots, i_m)$ is the binary representation of the integer $i = \sum_{j=1}^{m} i_j 2^{m-j}$ then the ith element of $\mathbf{f}$ (numbering the leftmost element as 0) is $f(i_1, i_2, \ldots, i_m)$. For example, for $m = 3$ we have

$$\mathbf{f} = (f(0,0,0), f(0,0,1), f(0,1,0), f(0,1,1), f(1,0,0), f(1,0,1), f(1,1,0), f(1,1,1))$$

and so $\mathbf{1} = (11111111)$, $\mathbf{x_1} = (00001111)$, $\mathbf{x_2} = (00110011)$, $\mathbf{x_3} = (01010101)$, and $\mathbf{x_1x_2} + \mathbf{x_2x_3} = (00010010)$.

We define a *generalised Boolean function* to be a function f from $\mathbb{Z}_2^m$ to $\mathbb{Z}_{2^h}$, where $h \geq 1$. It is straightforward to modify the proof of the algebraic normal form result stated above to show that any such function can be uniquely expressed as a linear combination over $\mathbb{Z}_{2^h}$ of the monomials (7), where the coefficient of each monomial belongs to $\mathbb{Z}_{2^h}$. As above, we specify a sequence $\mathbf{f}$ of length 2^m corresponding to the generalised Boolean function f. For example, for $h = 2$ and $m = 3$ we have $3\mathbf{x_1} = (00003333)$, $2\mathbf{x_1x_2x_3} = (00000002)$, and $\mathbf{x_1x_2} + 3\mathbf{x_2x_3} + 2 \cdot \mathbf{1} = (22212232)$. (Technically, for such expressions to be valid we must embed the range space $\mathbb{Z}_2^m$ of the monomials (7) in $\mathbb{Z}_{2^h}^m$.) Henceforth we shall drop the distinction between a generalised Boolean function and its corresponding sequence, and use the notation f to refer to both.

With this notation we are now ready to describe the Golay complementary pairs over $\mathbb{Z}_{2^h}$ of length 2^m.

Theorem 2.3 *Let*

$$f(x_1, x_2, \ldots, x_m) = 2^{h-1} \sum_{k=1}^{m-1} x_{\pi(k)} x_{\pi(k+1)} + \sum_{k=1}^{m} c_k x_k, \tag{8}$$

where π is a permutation of the symbols $\{1, 2, \ldots, m\}$ and $c_k \in \mathbb{Z}_{2^h}$. Then the sequences $a(x_1, x_2, \ldots, x_m) = f(x_1, x_2, \ldots, x_m) + c$ and $b(x_1, x_2, \ldots, x_m) = f(x_1, x_2, \ldots, x_m) + 2^{h-1}x_{\pi(1)} + c'$ are a Golay complementary pair over $\mathbb{Z}_{2^h}$ of length 2^m for any $c, c' \in \mathbb{Z}_{2^h}$.

Proof: The case $m = 1$ is easily checked by hand, so assume $m \geq 2$ and fix $u \neq 0$. By the definition of aperiodic autocorrelation (5), $C_a(u) + C_b(u)$ is the sum over i of terms $\xi^{a_i - a_{i+u}} + \xi^{b_i - b_{i+u}}$, where ξ is a primitive 2^h-th root of unity. For a given integer i, set $j = i + u$ and let $(i_1, i_2, \ldots, i_m)$ and $(j_1, j_2, \ldots, j_m)$ be the binary representation of i and j respectively. The sequence element a_i is given by $a(i_1, i_2, \ldots, i_m)$, as discussed above, which implies that

$$b_i - a_i = 2^{h-1} i_{\pi(1)} + c' - c. \tag{9}$$

Case 1*:* $j_{\pi(1)} \neq i_{\pi(1)}$. From (9), over $\mathbb{Z}_{2^h}$ we have $a_i - a_j - b_i + b_j = 2^{h-1}(j_{\pi(1)} - i_{\pi(1)}) = 2^{h-1}$, so $\xi^{a_i - a_j} / \xi^{b_i - b_j} = \xi^{2^{h-1}} = -1$. Therefore $\xi^{a_i - a_j} + \xi^{b_i - b_j} = 0$.

Case 2*:* $j_{\pi(1)} = i_{\pi(1)}$. Since $j \neq i$ we can define v to be the smallest integer for which $i_{\pi(v)} \neq j_{\pi(v)}$. Let i' be the integer whose binary representation $(i_1, i_2, \ldots, 1 - i_{\pi(v-1)}, \ldots, i_m)$ differs from that of i only in position $\pi(v-1)$, and similarly let j' have binary representation $(j_1, j_2, \ldots, 1 - j_{\pi(v-1)}, \ldots, j_m)$. By assumption $i_{\pi(v-1)} = j_{\pi(v-1)}$ and so $j' = i' + u$. We have therefore defined an invertible map from the ordered pair (i, j) to (i', j'), and both pairs contribute to $C_a(u) + C_b(u)$. Now substitution for i and i' in (8) gives $f_{i'} - f_i = 2^{h-1} i_{\pi(v-2)} + 2^{h-1} i_{\pi(v)} + c_{\pi(v-1)} - 2c_{\pi(v-1)} i_{\pi(v-1)}$

(unless $v = 2$, in which case we just delete terms involving $\pi(v-2)$ here and in what follows). Therefore $a_i - a_j - a_{i'} + a_{j'} = 2^{h-1}(j_{\pi(v-2)} - i_{\pi(v-2)}) + 2^{h-1}(j_{\pi(v)} - i_{\pi(v)}) - 2c_{\pi(v-1)}(j_{\pi(v-1)} - i_{\pi(v-1)}) = 2^{h-1}$ by the definition of v. Then (9) implies that $b_i - b_j - b_{i'} + b_{j'} = a_i - a_j - a_{i'} + a_{j'} = 2^{h-1}$. Arguing as in Case 1, we obtain $\xi^{a_i - a_j} + \xi^{a_{i'} - a_{j'}} = 0$ and $\xi^{b_i - b_j} + \xi^{b_{i'} - b_{j'}} = 0$. Therefore $(\xi^{a_i - a_j} + \xi^{b_i - b_j}) + (\xi^{a_{i'} - a_{j'}} + \xi^{b_{i'} - b_{j'}}) = 0$.

Combining these cases we see that $C_a(u) + C_b(u)$ comprises zero contributions (as in Case 1), and contributions which sum to zero in pairs (as in Case 2). Therefore $a(x_1, x_2, \ldots, x_m)$ and $b(x_1, x_2, \ldots, x_m)$ are a Golay complementary pair, by Definition 2.1. □

Corollary 2.4 *For any permutation π of the symbols $\{1, 2, \ldots, m\}$ and for any $c, c_k \in \mathbb{Z}_{2^h}$,*

$$a(x_1, x_2, \ldots, x_m) = 2^{h-1} \sum_{k=1}^{m-1} x_{\pi(k)} x_{\pi(k+1)} + \sum_{k=1}^{m} c_k x_k + c$$

is a Golay sequence over $\mathbb{Z}_{2^h}$ of length 2^m.

Corollary 2.4 explicitly determines $2^{h(m+1)} \cdot m!/2$ Golay sequences over $\mathbb{Z}_{2^h}$ of length 2^m (using the factor $m!/2$ rather than $m!$ because the expression $\sum_{k=1}^{m-1} x_{\pi(k)} x_{\pi(k+1)}$ is invariant under the mapping $\pi \mapsto \pi'$, where $\pi'(k) = \pi(m+1-k)$). Numerical evidence suggests that there are no other Golay sequences over $\mathbb{Z}_{2^h}$ of this length, although we do not have a proof of this. Theorem 2.3 also shows how to form sets of Golay complementary pairs:

Corollary 2.5 *Let $f \equiv f(x_1, x_2, \ldots, x_m) = 2^{h-1} \sum_{k=1}^{m-1} x_{\pi(k)} x_{\pi(k+1)} + \sum_{k=1}^{m} c_k x_k$, where π is a permutation of the symbols $\{1, 2, \ldots, m\}$ and $c_k \in \mathbb{Z}_{2^h}$. Then any sequence in the set*

$$A = \{f + c,\ f + 2^{h-1}(x_{\pi(1)} + x_{\pi(m)}) + c \mid c \in \mathbb{Z}_{2^h}\} \tag{10}$$

forms a Golay complementary pair over $\mathbb{Z}_{2^h}$ of length 2^m with any sequence in the set

$$B = \{f + 2^{h-1} x_{\pi(1)} + c',\ f + 2^{h-1} x_{\pi(m)} + c' \mid c' \in \mathbb{Z}_{2^h}\}. \tag{11}$$

Proof: Consider a single sequence a of the form $f + c$. By Theorem 2.3, this sequence forms a Golay complementary pair with each of the 2^h sequences $\{f + 2^{h-1} x_{\pi(1)} + c' \mid c' \in \mathbb{Z}_{2^h}\}$. Now if π is replaced by the permutation π' defined by $\pi'(k) = \pi(m+1-k)$, $f + c$ is invariant but $f + 2^{h-1} x_{\pi(1)} + c'$ maps to $f + 2^{h-1} x_{\pi(m)} + c'$. Therefore a also forms a Golay complementary pair with each of the 2^h sequences $\{f + 2^{h-1} x_{\pi(m)} + c' \mid c' \in \mathbb{Z}_{2^h}\}$. We have shown that a forms a Golay complementary pair with each sequence $b \in B$, and it follows from Definition 2.1 that for each u, every sequence $b \in B$ has the same value of $C_B(u)$.

Similarly we can show that a single sequence b of the form $f + 2^{h-1} x_{\pi(1)} + c'$ forms a Golay complementary pair with each of the 2^{h+1} sequences $a \in A$ and that, for each u, every sequence $a \in A$ has the same value of $C_A(u)$. Therefore any sequence $a \in A$ forms a Golay complementary pair with any sequence $b \in B$. □

Corollary 2.5 explicitly determines $2^{2(h+1)} \cdot 2^{hm-2} \cdot m!/2$ Golay complementary pairs $\{a, b\}$ over $\mathbb{Z}_{2^h}$ of length 2^m. It also suggests a natural partition of the Golay sequences of Corollary 2.4 into $2^{hm-2} \cdot m!/2$ classes of size 2^{h+2}, each class comprising a set A of 2^{h+1} sequences of the form (10) and a set B of 2^{h+1} sequences of the form (11).

However the true number of Golay complementary pairs $\{a, b\}$ over $\mathbb{Z}_{2^h}$ of length 2^m can be greater than that calculated above because in some cases $C_A(u) = C_{A'}(u)$, for all u, for two distinct sets A, A' of the form (10). For example, for $h = 2$ and $m = 3$, by Corollary 2.5 any of the eight sequences in

$$A = \{2(x_1 x_2 + x_2 x_3) + c,\ 2(x_1 x_2 + x_2 x_3) + 2x_1 + 2x_3 + c \mid c \in \mathbb{Z}_4\}$$

forms a quaternary Golay complementary pair of length 8 with any sequence in

$$B = \{2(x_1x_2 + x_2x_3) + 2x_1 + c',\ 2(x_1x_2 + x_2x_3) + 2x_3 + c' \mid c' \in \mathbb{Z}_4\}.$$

Similarly any one of the eight sequences in

$$A' = \{2(x_2x_1 + x_1x_3) + 3x_2 + x_3 + c,\ 2(x_2x_1 + x_1x_3) + x_2 + 3x_3 + c \mid c \in \mathbb{Z}_4\}$$

forms a Golay complementary pair with any sequence in

$$B' = \{2(x_2x_1 + x_1x_3) + x_2 + x_3 + c',\ 2(x_2x_1 + x_1x_3) + 3x_2 + 3x_3 + c' \mid c' \in \mathbb{Z}_4\}.$$

But in fact direct calculation shows that $(C_A(u) \mid u = 0, 1, \ldots, 7) = (C_{A'}(u) \mid u = 0, 1, \ldots, 7) = (8, -1, 0, 3, 0, 1, 0, 1)$, so these 32 sequences collectively give rise to $16^2 = 256$ Golay complementary pairs rather than the expected $2 \cdot 8^2 = 128$.

In 1961 Golay [19] gave an explicit construction for binary Golay complementary pairs of length 2^m and later noted [20] that the construction implies the existence of at least $2^{m+1} \cdot m!/2$ binary Golay sequences of this length. These results correspond to the binary case $h = 1$ of Theorem 2.3 and Corollary 2.4, and indeed our proof of Theorem 2.3 is modelled on Golay's original construction [19]. However the non-binary cases $h > 1$ of Theorem 2.3 have not been constructed explicitly elsewhere. Moreover we shall prove in Section 3 the new result, announced in [12], that the Golay sequences of Corollary 2.4 form a subcode of the second-order Reed-Muller code (suitably generalised for non-binary cases).

Golay [19] also presented a recursive construction for binary Golay complementary pairs involving concatenation and interleaving of sequences. Budišin [8], building on earlier work of Sivaswamy [41], gave a more general recursive construction for Golay complementary pairs and showed that the set of all binary Golay complementary pairs of length 2^m obtainable from it coincides with those given explicitly by Golay [19] (as described above). Paterson [34] has shown that the set of all Golay complementary pairs over $\mathbb{Z}_{2^h}$ of length 2^m obtainable by Golay's recursive construction ($h = 1$) and by Budišin's ($h \geq 1$) coincides with those given explicitly in Theorem 2.3.

We remark that [19] introduced a definition of equivalence of binary Golay complementary pairs that was taken up by later authors, particularly when counting the number of such pairs of small length by computer search. We believe that the underlying structure of Golay complementary pairs over $\mathbb{Z}_{2^h}$ of length 2^m is more apparent if this definition, and its obvious generalisation for $h > 1$, is not used.

3 Reed-Muller codes

Binary Reed-Muller codes first appeared in print in 1954 and remain "...one of the oldest and best understood families of codes" [28, p. 370]. They have good error correction properties, provided the block length is not too large, and have the important practical advantage of being easy to decode. The rth order binary Reed-Muller code $\mathrm{RM}(r, m)$ of length 2^m is generated by the monomials in the Boolean functions x_i of degree at most r [28]. This allows us to restate the binary case $h = 1$ of Corollary 2.4 as:

Corollary 3.1 *Each of the $m!/2$ cosets of $\mathrm{RM}(1, m)$ in $\mathrm{RM}(2, m)$ having a coset representative of the form $\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)}$ comprises 2^{m+1} binary Golay sequences of length 2^m, where π is a permutation of the symbols $\{1, 2, \ldots, m\}$.*

Note that the PMEPR of a sequence depends on the order in which its elements occur, so here and elsewhere we do not adopt the coding theory convention that regards two codes as equivalent if one can be obtained from the other by a permutation of coordinates.

We wish to make an analogous statement to Corollary 3.1 for the non-binary cases $h > 1$ of Corollary 2.4. To do this, we follow the landmark paper [21] and define a *linear code over $\mathbb{Z}_H$*

of length n to be a subset of $\mathbb{Z}_H^n$ such that the sum of any two codewords is a codeword. [21] demonstrates that defining linear codes in this way, over rings that are not fields, preserves many of the properties of classical codes even though not every element of the code alphabet has a multiplicative inverse. In particular such a code can be specified in terms of a generator matrix such that the code consists of all distinct linear combinations over $\mathbb{Z}_H$ of the rows of the matrix. We now define two new linear codes over $\mathbb{Z}_{2^h}$ of length 2^m in terms of the generalised Boolean functions x_i described in Section 2.

Definition 3.2 *For $h \geq 1$ and $0 \leq r \leq m$, the rth order linear code $\mathrm{RM}_{2^h}(r,m)$ over $\mathbb{Z}_{2^h}$ of length 2^m is generated by the monomials in the x_i of degree at most r.*

Definition 3.3 *For $h > 1$ and $0 \leq r \leq m+1$, the rth order linear code $\mathrm{ZRM}_{2^h}(r,m)$ over $\mathbb{Z}_{2^h}$ of length 2^m is generated by the monomials in the x_i of degree at most $r-1$ together with 2 times the monomials in the x_i of degree r (with the convention that the monomials of degree -1 and $m+1$ are equal to zero).*

The code $\mathrm{RM}_{2^h}(r,m)$ generalises the binary Reed-Muller code $\mathrm{RM}(r,m)$ from the alphabet $\mathbb{Z}_2$ (the case $h = 1$) to the alphabet $\mathbb{Z}_{2^h}$. Likewise the code $\mathrm{ZRM}_{2^h}(r,m)$ generalises the quaternary Reed-Muller code $\mathrm{ZRM}(r,m)$ defined in [21] from the alphabet $\mathbb{Z}_4$ (the case $h = 2$) to the alphabet $\mathbb{Z}_{2^h}$. In both cases the formal generator matrix is unchanged as h varies, but the alphabet over which it is interpreted changes. The number of monomials in the x_i of degree r is $\binom{m}{r}$, so $\mathrm{RM}_{2^h}(r,m)$ contains $2^{h\sum_{i=0}^{r}\binom{m}{i}}$ codewords and $\mathrm{ZRM}_{2^h}(r,m)$ contains $2^{h\sum_{i=0}^{r-1}\binom{m}{i}} \cdot 2^{(h-1)\binom{m}{r}}$ codewords. Note these generalisations of the Reed-Muller code are distinct from the Generalised Reed-Muller code $\mathrm{GRM}(r,m)$ [35], which is defined over a field, and the quaternary Reed-Muller code $\mathrm{QRM}(r,m)$ [21], which generalises the quaternary representation of the Kerdock code.

For example, $\mathrm{RM}_{2^h}(1,4)$ has the generator matrix

$$\left[\begin{array}{cccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1
\end{array}\right]
\begin{array}{l}
1\\ x_1\\ x_2\\ x_3\\ x_4
\end{array} \tag{12}$$

and contains 2^{5h} codewords for $h \geq 1$, and $\mathrm{ZRM}_{2^h}(2,4)$ has the generator matrix

$$\left[\begin{array}{cccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&2&2&2&2\\
0&0&0&0&0&0&0&0&0&0&2&2&0&0&2&2\\
0&0&0&0&0&0&0&0&0&2&0&2&0&2&0&2\\
0&0&0&0&0&0&2&2&0&0&0&0&0&0&2&2\\
0&0&0&0&0&2&0&2&0&0&0&0&0&2&0&2\\
0&0&0&2&0&0&0&2&0&0&0&2&0&0&0&2
\end{array}\right]
\begin{array}{l}
1\\ x_1\\ x_2\\ x_3\\ x_4\\ 2x_1x_2\\ 2x_1x_3\\ 2x_1x_4\\ 2x_2x_3\\ 2x_2x_4\\ 2x_3x_4
\end{array}$$

and contains $2^{5h} \cdot 2^{6(h-1)}$ codewords for $h > 1$.

We are particularly interested in the code $\mathrm{ZRM}_{2^h}(2,m)$, comprising $2^{(h-1)m(m-1)/2}$ cosets of the subcode $\mathrm{RM}_{2^h}(1,m)$, each coset containing $2^{h(m+1)}$ codewords. We can restate the cases $h > 1$ of Corollary 2.4 in terms of these codes as:

Corollary 3.4 *Each of the $m!/2$ cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$ having a coset representative of the form $2^{h-1}\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)}$ comprises $2^{h(m+1)}$ Golay sequences over $\mathbb{Z}_{2^h}$ of length 2^m, where π is a permutation of the symbols $\{1,2,\ldots,m\}$ and $h > 1$.*

We have seen in Theorem 2.2 that the PMEPR of any Golay sequence is at most 2, and Corollaries 3.1 and 3.4 give concise and structured representations for large sets of Golay sequences in the cases $h = 1$ and $h > 1$ respectively. These representations readily lend themselves to implementation in an OFDM coding scheme having tight envelope power control. If we did not wish to consider using sequences other than Golay sequences for OFDM transmission then it would be more natural to replace the multiple 2 in Definition 3.3 by the multiple 2^{h-1} and to extend the definition of $\mathrm{ZRM}_{2^h}(r,m)$ to the case $h = 1$; in that case Corollary 3.4 would hold for all cases $h \geq 1$. However by taking more cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$ we can increase the rate of OFDM transmission at the cost of progressively larger values of PMEPR, as we discuss in Section 4. To allow such design freedom, our objective in defining $\mathrm{ZRM}_{2^h}(r,m)$ was that the linear code $\mathrm{ZRM}_{2^h}(2,m)$ should be the largest superset of the Golay sequences of Corollary 2.4 which does not compromise the minimum Hamming or Lee distance, as we now describe.

Let $a = (a_0, a_1, \ldots, a_{n-1})$ be a sequence over $\mathbb{Z}_H$ of length n. The *Hamming weight* of a is the number of nonzero a_i and the *Lee weight* [35] of a is $\sum_{i=0}^{n-1} \min(a_i, H - a_i)$. The Hamming (or Lee) *distance* between two such sequences a and b is the Hamming (or Lee) weight of $a - b$ (when written as a sequence over $\mathbb{Z}_H$). The Hamming distance measures the number of positions in which a and b differ, whereas the Lee distance takes into account the magnitude of the difference over $\mathbb{Z}_H$ at each position; these coincide in the binary case $H = 2$. For example, the Hamming distance between the sequences $(5,7,0,1)$ and $(3,7,7,6)$ over $\mathbb{Z}_8$ is 3 whereas the Lee distance is $2+0+1+3 = 6$. The minimum Hamming or minimum Lee distance of a code, which is taken over all pairs of distinct codewords, is a measure of its error correction capability: if the (Hamming or Lee) minimum distance is d then we can always correct errors of (Hamming or Lee) weight less than $d/2$. If the transmission channel renders all $H - 1$ possible errors for a given codeword position equally likely then the traditional Hamming distance metric is an appropriate measure. However if errors involving a transition between adjacent values in $\mathbb{Z}_H$ are much more likely than other errors in a given position then the Lee distance metric is more appropriate [35]. We consider both metrics to be useful measures of error correction capability for OFDM transmission and so we now derive the minimum Hamming and Lee distance for the codes $\mathrm{RM}_{2^h}(r,m)$ and $\mathrm{ZRM}_{2^h}(r,m)$. The method uses the fact that the minimum Hamming distance of the binary code $\mathrm{RM}(r,m)$ is 2^{m-r}.

Theorem 3.5 *The following hold for* $0 \leq r \leq m$*:*

	$\mathrm{RM}_{2^h}(r,m)$	$\mathrm{ZRM}_{2^h}(r,m)$
	$(h \geq 1)$	$(h > 1)$
minimum Hamming distance	2^{m-r}	2^{m-r}
minimum Lee distance	2^{m-r}	2^{m-r+1}

Proof: For any linear code the minimum distance equals the minimum weight of the nonzero codewords, in both the Hamming and Lee case. For each of the four values required by the theorem we derive a lower bound on the minimum distance and then exhibit a codeword whose weight equals that lower bound.

We firstly use induction on $h \geq 2$ to establish the minimum Hamming and Lee distance of $\mathrm{ZRM}_{2^h}(r,m)$. The case $r = 0$ is trivial and can be excluded. Let $a = (a_0, a_1, \ldots, a_{2^m-1})$ be any nonzero codeword in $\mathrm{ZRM}_{2^h}(r,m)$ and define $b = (b_0, b_1, \ldots, b_{2^m-1})$ by $b_i \equiv a_i \pmod{2^{h-1}}$ and $b_i \in \mathbb{Z}_{2^{h-1}}$ for each i. Now b is a codeword in $\mathrm{ZRM}_{2^{h-1}}(r,m)$ if $h > 2$ and is a codeword in $\mathrm{RM}(r-1,m)$ if $h = 2$.

Case 1*:* $b = 0$. In this case $a = 2^{h-1}a'$ for a nonzero codeword a' in $\mathrm{RM}(r,m)$, so a' has Hamming weight at least 2^{m-r}. Therefore a has Hamming weight at least 2^{m-r} and Lee weight at least $2^{h-1} \cdot 2^{m-r} \geq 2^{m-r+1}$.

Case 2*:* $b \neq 0$. In this case b has Hamming weight at least 2^{m-r} and Lee weight over $\mathbb{Z}_{2^{h-1}}$ at least 2^{m-r+1}, using the induction hypothesis if $h > 2$. Therefore a has Hamming weight at least

2^{m-r}, and has Lee weight over $\mathbb{Z}_{2^h}$ at least 2^{m-r+1} (since $\min(a_i, 2^h - a_i) \geq \min(b_i, 2^{h-1} - b_i)$ when $a_i = b_i$ or $b_i + 2^{h-1}$).

Furthermore the codeword $2^{h-1}x_1x_2\cdots x_r$ has Hamming weight 2^{m-r}, and the codeword $x_1x_2\cdots x_{r-1}$ (or 1 if $r = 1$) has Lee weight 2^{m-r+1}. This completes the proof for $\mathrm{ZRM}_{2^h}(r,m)$.

By a similar induction on h the minimum Hamming and Lee distance for $\mathrm{RM}_{2^h}(r,m)$ is at least 2^{m-r}, and the codeword $x_1x_2\cdots x_r$ has Hamming and Lee weight 2^{m-r}. □

The proof of Theorem 3.5 demonstrates our earlier claim that the minimum Hamming and Lee distance of $\mathrm{ZRM}_{2^h}(r,m)$ is not compromised by using the multiple 2 in Definition 3.3 instead of the multiple 2^{h-1}.

We conclude this section with a short discussion of bent functions, which will be useful when describing encoding options in Section 4. For m even, a *bent function* is a Boolean function $f(x_1, x_2, \ldots, x_m)$ for which all the Hadamard transform coefficients of the $+1/-1$ sequence $(-1)^{f(x_1,x_2,\ldots,x_m)}$ have magnitude $2^{m/2}$. A bent function is equivalent to a Hadamard difference set in the group $\mathbb{Z}_2^m$. The function $\sum_{k=1}^{m/2} x_{2k-1}x_{2k}$ is bent, and any affine transformation of a bent function is also bent. A *Kerdock code of length* 2^m is the union of 2^{m-1} cosets of $\mathrm{RM}(1,m)$ in $\mathrm{RM}(2,m)$, where $m \geq 4$ is even. One of the coset representatives is 0 (so $\mathrm{RM}(1,m)$ itself is contained in the code), and all the others are bent functions having the property that the sum of any two of them is also a bent function. The minimum Hamming distance of any such code is $2^{m-1} - 2^{(m-2)/2}$. For details of these and other results, see [28]. We now show that for m even, all the binary Golay sequences of Corollary 3.1 are bent functions; since these sequences occur as cosets of $\mathrm{RM}(1,m)$ in $\mathrm{RM}(2,m)$, some may also belong to a Kerdock code.

Theorem 3.6 *For m even, each of the $m!/2$ cosets of* $\mathrm{RM}(1,m)$ *in* $\mathrm{RM}(2,m)$ *having a coset representative of the form $\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)}$ comprises 2^{m+1} bent functions, where π is a permutation of the symbols $\{1, 2, \ldots, m\}$.*

Proof: We show that the function $\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)} + \sum_{k=1}^{m} c_kx_k + c$ can be obtained from $\sum_{k=1}^{m/2} x_{2k-1}x_{2k}$ by a sequence of affine transformations, for any $c, c_k \in \mathbb{Z}_2$. The linear transformation $x_1 \mapsto x_1 + x_3,\quad x_3 \mapsto x_3 + x_5,\quad \ldots,\quad x_{m-3} \mapsto x_{m-3} + x_{m-1}$, and $x_i \mapsto x_i$ for all other x_i, maps $\sum_{k=1}^{m/2} x_{2k-1}x_{2k}$ to $\sum_{k=1}^{m-1} x_kx_{k+1}$. Then the linear transformation $x_i \mapsto x_{\pi(i)} + b_i$, where each $b_i \in \mathbb{Z}_2$ is determined by a single c_k, maps this to $\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)} + \sum_{k=1}^{m} c_kx_k + b$ for $b = c$ or $c + 1$. If necessary, we can apply a translation to add 1 and so obtain the required function. □

4 Encoding

The combination of the new results of Sections 2 and 3 immediately suggests a practical OFDM coding scheme using 2^h-phase shift keying: allow as codewords only those Golay sequences described in Corollaries 3.1 (for $h = 1$) and 3.4 (for $h > 1$). This simultaneously confers tight envelope power control, by Theorem 2.2, and good error correction capability, by Theorem 3.5. The Golay sequences in question occur as $m!/2$ cosets of $\mathrm{RM}_{2^h}(1,m)$ and for convenience of implementation we use 2^w of these cosets, where 2^w is the largest integer power of 2 no greater than $m!/2$. Under this scheme we encode $w + h(m+1)$ information bits per OFDM symbol period. We use w bits to encode the choice of coset representative using a look-up table. The remaining $h(m+1)$ bits are converted to $m+1$ information symbols $u_1, u_2, \ldots u_m, u \in \mathbb{Z}_{2^h}$ by taking each consecutive group of h bits to be the binary representation of an element of $\mathbb{Z}_{2^h}$. The information symbols are then used to form the linear combination $\sum_{i=1}^{m} u_ix_i + u$, in which each symbol multiplies one row of the standard generator matrix for $\mathrm{RM}_{2^h}(1,m)$. This linear combination can be calculated in hardware in 2^m clock cycles using the encoding circuit for $\mathrm{RM}(1,m)$ given in [28, p. 420]. The sum (over $\mathbb{Z}_{2^h}$) of this linear combination with the selected coset representative is the OFDM codeword $(a_0, a_1, \ldots a_{2^m-1})$, which is modulated prior to transmission according to (1). The *code rate*, namely the ratio of the number of information bits to the number of coded bits, is $(w + h(m+1))/(2^m h)$, and we define the *information rate* to be h times the code rate. The information rate describes the increased rate at which information bits are encoded when we change the code from binary to quaternary, from quaternary to octary, and so on.

For example, consider the octary case with 16 carriers ($h = 3$, $m = 4$). The 12 coset representatives given by Corollary 3.4 are

$$\begin{array}{lcl}
(0\ 0\ 0\ 4\ 0\ 0\ 4\ 0\ 0\ 0\ 0\ 4\ 4\ 4\ 0\ 4) & = & 4(x_1x_2 + x_2x_3 + x_3x_4),\\
(0\ 0\ 0\ 4\ 0\ 4\ 0\ 0\ 0\ 0\ 0\ 4\ 4\ 0\ 4\ 4) & = & 4(x_1x_2 + x_2x_4 + x_3x_4),\\
(0\ 0\ 0\ 0\ 0\ 4\ 4\ 0\ 0\ 0\ 4\ 4\ 0\ 4\ 0\ 4) & = & 4(x_1x_3 + x_2x_3 + x_2x_4),\\
(0\ 0\ 0\ 4\ 0\ 4\ 0\ 0\ 0\ 0\ 4\ 0\ 0\ 4\ 4\ 4) & = & 4(x_1x_3 + x_3x_4 + x_2x_4),\\
(0\ 0\ 0\ 0\ 0\ 4\ 4\ 0\ 0\ 4\ 0\ 4\ 0\ 0\ 4\ 4) & = & 4(x_1x_4 + x_2x_4 + x_2x_3),\\
(0\ 0\ 0\ 4\ 0\ 0\ 4\ 0\ 0\ 4\ 0\ 0\ 0\ 4\ 4\ 4) & = & 4(x_1x_4 + x_3x_4 + x_2x_3),\\
(0\ 0\ 0\ 4\ 0\ 0\ 0\ 4\ 0\ 0\ 4\ 0\ 4\ 4\ 0\ 4) & = & 4(x_1x_2 + x_1x_3 + x_3x_4),\\
(0\ 0\ 0\ 4\ 0\ 0\ 0\ 4\ 0\ 4\ 0\ 0\ 4\ 0\ 4\ 4) & = & 4(x_1x_2 + x_1x_4 + x_3x_4),\\
(0\ 0\ 0\ 0\ 0\ 0\ 4\ 4\ 0\ 4\ 4\ 0\ 0\ 4\ 0\ 4) & = & 4(x_2x_3 + x_1x_3 + x_1x_4),\\
(0\ 0\ 0\ 0\ 0\ 4\ 0\ 4\ 0\ 4\ 4\ 0\ 0\ 0\ 4\ 4) & = & 4(x_2x_4 + x_1x_4 + x_1x_3),\\
(0\ 0\ 0\ 0\ 0\ 4\ 0\ 4\ 0\ 0\ 4\ 4\ 4\ 0\ 0\ 4) & = & 4(x_1x_3 + x_1x_2 + x_2x_4),\\
(0\ 0\ 0\ 0\ 0\ 0\ 4\ 4\ 0\ 4\ 0\ 4\ 4\ 0\ 0\ 4) & = & 4(x_2x_3 + x_1x_2 + x_1x_4),
\end{array}$$

of which we choose eight (say the first eight), so $w = 3$. The union of the eight cosets of $\mathrm{RM}_8(1,4)$ having these coset representatives comprises the set of OFDM codewords, all of which have PMEPR of at most 2. The code forms a subcode of $\mathrm{ZRM}_8(2,4)$ and has minimum Hamming and Lee distance 4 and 8 respectively. An error of Hamming weight 1 can always be corrected, as can an error of Lee weight at most 3. The code rate is 3/8 and the information rate is 9/8. Given 18 information bits, three are used to select one of the eight coset representatives and the remaining 15 are regarded as the binary representation of five information symbols u_1, u_2, u_3, u_4, u. The linear combination $u_1x_1 + u_2x_2 + u_3x_3 + u_4x_4 + u$ is calculated with reference to the generator matrix (12) for $\mathrm{RM}_8(1,4)$ and added to the selected coset representative. Suppose the 18 information bits are 011101111011110110. The first three bits 011 select the coset representative (0004040000400444) (labelling the first eight coset representatives $000, 001, \ldots, 111$). The remaining 15 bits select the linear combination $5x_1 + 7x_2 + 3x_3 + 6x_4 + 6 = (6417530631642053)$, so the OFDM codeword is (6413570631242417).

The above coding scheme is restricted to the Golay sequences described in Sections 2 and 3. These sequences occur as $m!/2$ "Golay cosets" of $\mathrm{RM}_{2^h}(1,m)$ within a second-order linear code, where the second-order linear code is $\mathrm{RM}_2(2,m)$ in the binary case $h = 1$ and is $\mathrm{ZRM}_{2^h}(2,m)$ in the non-binary cases $h > 1$. We can increase the code rate, at the cost of progressively larger values of PMEPR, by including additional cosets of $\mathrm{RM}_{2^h}(1,m)$ within the same second-order code. These additional cosets do not necessarily comprise or even contain Golay sequences. Nonetheless we have found that partitioning the second-order code into cosets of $\mathrm{RM}_{2^h}(1,m)$ is an effective means of isolating codewords with large values of PMEPR. Alternatively we can increase the minimum Hamming distance, at the cost of a lower code rate, by choosing fewer than 2^w of the original $m!/2$ Golay cosets. In this way we can trade off code rate, PMEPR and error correction capability to provide a range of solutions to the envelope power problem. For implementation convenience we use $2^{w'}$ cosets of $\mathrm{RM}_{2^h}(1,m)$ for some integer w' to encode $w' + h(m+1)$ information bits, storing the coset representatives in a look-up table. We can determine the possible options for given h and m by arranging all the cosets of $\mathrm{RM}_{2^h}(1,m)$ (within the appropriate second-order code) in increasing order of their maximum PEP over the $2^{h(m+1)}$ codewords in the coset, as we now illustrate.

4.1 The binary case

Consider the binary case with 16 carriers ($h = 1$, $m = 4$). Table 1 lists the $2^{m(m-1)/2} = 64$ cosets of $\mathrm{RM}_2(1,4)$ in $\mathrm{RM}_2(2,4)$ in increasing order of their maximum PEP over the 32 codewords in the coset. The PEP of each codeword is calculated using 2^j *times oversampling*, finding $P(t) = |s(t)|^2$

from (1) at each sample point $t = i/(2^{m+j}\Delta f)$ for $i = 0, 1, \ldots, 2^{m+j} - 1$ and taking the largest sample value of $P(t)$. The value of j is increased until the maximum calculated PEP over the coset is stable. The first 12 cosets of Table 1 are the $m!/2$ Golay cosets of Corollary 3.1, each of which has a maximum PMEPR of at most 2 (since PMEPR = PEP$/n$ and we have fixed $n = 2^m$) in accordance with Theorem 2.2. The final coset in the list is $\mathrm{RM}_2(1,4)$ itself, which has a maximum PMEPR of 2^m since it contains the sequence $(0, 0, \ldots, 0)$. The remaining cosets have intermediate values of maximum PMEPR. Observe that the maximum PMEPR for the cosets in the first half of the list is no greater than 4; we remark that this property holds for the binary case with 8 and 32 carriers too.

Table 4 summarises some possible options for binary coding for 16 and 32 carriers, most of which are derived from the ordered list given in Table 1. The reference option for 16 carriers is Option 3, which uses the first eight (Golay) cosets of this ordered list. Option 4 uses the 32 cosets in the first half of the list and trades an increase in code rate for an increase in maximum PMEPR from 2 to 4. Option 1 uses just the first coset of the list and trades an increase in minimum Hamming distance from 4 to 8 for a reduction in code rate. Option 2 is a compromise between Options 1 and 3, based on the Kerdock code of length 16 whose coset representatives are [27]: 0, $x_1x_2 + x_1x_3 + x_3x_4$, $x_1x_3 + x_2x_3 + x_2x_4$, $x_1x_2 + x_2x_4 + x_3x_4$, $x_1x_4 + x_2x_3 + x_3x_4$, $x_1x_3 + x_1x_4 + x_2x_4$, $x_1x_2 + x_1x_4 + x_2x_3$ and $x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$. Six of these eight coset representatives are of the form $\sum_{k=1}^{m-1} x_{\pi(k)}x_{\pi(k+1)}$ (and so appear in the first 12 places of the list), and by choosing any four of the six we obtain a minimum Hamming distance of 6.

The ordered list for binary coding with 32 carriers (not shown here) contains 1024 cosets of $\mathrm{RM}_2(1,5)$ in $\mathrm{RM}_2(2,5)$ and is headed by the 60 Golay cosets of Corollary 3.1. The reference option for 32 carriers is Option 7, which uses the first 32 of these 60 cosets. Option 6 uses just the first coset of the list. (We could derive a compromise between Options 6 and 7 having minimum Hamming distance 12 based on a Kerdock code of length 32. Although we have given only the classical definition of a Kerdock code, for $m \geq 4$ even, [21] defines a corresponding Kerdock code for $m \geq 3$ odd which can be represented as the union of 2^{m-1} cosets of $\mathrm{RM}(1,m)$ in $\mathrm{RM}(2,m)$ and which has minimum Hamming distance $2^{m-1} - 2^{(m-1)/2}$. The number of information bits of this compromise option will be determined by how many of the 16 Kerdock cosets are also Golay cosets.) Comparing Options 1 and 3 with Options 6 and 7 respectively, we see that doubling the number of carriers from 16 to 32 incurs a penalty in terms of code rate. However it carries the advantage that intersymbol interference in the transmitted signal will be reduced and consequently delay spread in the channel will also be reduced.

Alternatively we can maintain the code rate as the number of carriers doubles, at the cost of increased PMEPR. It is straightforward to show that if a and b are sequences over $\mathbb{Z}_H$ of length n having PMEPR at most R then the sequence formed by interleaving or concatenating the elements of a and b has PMEPR at most $2R$. For example by encoding according to Option 1 twice independently, and either interleaving or concatenating the resulting codeword elements, we obtain the composition coding scheme of Option 8 having the same code rate but a maximum PMEPR of 4. Decoding is likewise carried out by regarding the received codeword as two independent half-length codewords, which is indicated in Table 4 by writing the minimum Hamming distance for Option 8 as $8;8$ (see also Section 5). Examples of this technique of interleaving or concatenating codewords to maintain code rate and to control PMEPR for OFDM transmission have been noted previously [3, 32, 40]. Option 10 is similarly derived from Option 3, with the following modification to improve the code rate slightly. Recall that there are 12 cosets listed in Table 1 having PMEPR at most 2, of which Option 3 uses the first eight. We can therefore form $2^7 < 12 \cdot 12$ ordered pairs of length 16 coset representatives to be added to the respective length 16 linear combinations in $\mathrm{RM}_2(1,4)$ prior to interleaving or concatenating. In this way Option 10 encodes $7 + 2 \cdot 5 = 17$ rather than $2 \cdot 3 + 2 \cdot 5 = 16$ information bits. Likewise Option 2 uses four cosets chosen from six, and since $2^5 < 6 \cdot 6$ we can encode $5 + 2 \cdot 5 = 15$ information bits in the composition coding scheme of Option 9. Finally, Option 5 is a composition coding scheme based on a single Golay coset of $\mathrm{RM}_2(1,3)$.

4.2 The quaternary case

For the non-binary cases $h > 1$ we form similar ordered lists of the $2^{(h-1)m(m-1)/2}$ cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$. Consider the quaternary case with 16 carriers ($h = 2$, $m = 4$). Table 2 lists the 64 cosets of $\mathrm{RM}_4(1,4)$ in $\mathrm{ZRM}_4(2,4)$ in increasing order of their maximum PEP over the 1024 codewords in the coset, headed by the 12 Golay cosets of Corollary 3.4. The maximum PMEPR for the cosets in the first half of the list is no greater than 4 (as in the binary case), and the same is true for 8 and 32 carriers. Table 2 contains a striking feature not present in Table 1: the maximum PMEPR over each coset is an exact power of 2, and the same is true for 4, 8 and 32 carriers.

Table 5 summarises options for quaternary coding for 16 and 32 carriers, mostly derived from the ordered list of Table 2. These options are determined in similar manner to those having the corresponding option number in Table 4. A similar method to the proof of Theorem 3.5 shows that if $m \geq 4$ is even and the set of cosets $\{g_i + \mathrm{RM}_2(1,m)\}$ is a Kerdock code of length 2^m then the minimum Hamming distance of $\{2^{h-1}g_i + \mathrm{RM}_{2^h}(1,m)\}$ over $\mathbb{Z}_{2^h}$ is $2^{m-1} - 2^{(m-2)/2}$ for $h > 1$. Option 2 exploits this result, using coset representatives whose values are twice those of the binary Option 2. Option 5a is a composition coding scheme based on three Golay cosets of $\mathrm{RM}_4(1,3)$. Error correction for this option can be done with respect to Lee distance (though not always with respect to Hamming distance, which is why it does not occur in Table 4). Comparison of Tables 4 and 5 demonstrates that choice of modulation scheme is a further component of design freedom. The quaternary schemes have up to twice the information rate of the corresponding binary schemes for the same minimum Hamming distance, together with enhanced error correction capability based on Lee distance. Their disadvantage is that quaternary modulation leads to a smaller minimum Euclidean distance than binary modulation and so their transmission error rate is larger.

4.3 The octary case

Consider the octary case with 16 carriers ($h = 3$, $m = 4$). Table 3 lists the 4096 cosets of $\mathrm{RM}_8(1,4)$ in $\mathrm{ZRM}_8(1,4)$ in increasing order of their maximum PEP over the 32768 codewords in the coset. The list is headed by the 12 Golay cosets of Corollary 3.4, followed by 48 cosets whose maximum PMEPR is exactly 3. The maximum PMEPR for the cosets in the first quarter of the list is no greater than 4; for 8 carriers this is true for the first half of the list. Table 6 summarises options for octary coding for 16 and 32 carriers, the option numbers corresponding to those in Table 5. Option 2 uses coset representatives whose values are four times those of the binary Option 2. Option 4 has smaller maximum PMEPR than the quaternary Option 4 because it uses 12 Golay cosets together with 20 of the 48 cosets having maximum PMEPR of 3. The parameters of Option 5 coincide with those proposed independently in [3, 32].

4.4 Comments

The coset ordering process illustrated for binary, quaternary and octary modulation can clearly be applied to larger values of h. Since these coding schemes are all based on the same formal generator matrix for $\mathrm{RM}_{2^h}(1,m)$, interpreted over different alphabets $\mathbb{Z}_{2^h}$, it is simple to change adaptively between coding options according to the propagation channel and evolving system requirements. In this way we obtain flexible coding schemes which combine tight control of PMEPR with powerful error correction capability and structured encoding. Efficient methods of decoding will be discussed in Section 5.

The numerical results presented demonstrate, at least for small values of h and m, that partitioning the codewords of $\mathrm{RM}_2(2,m)$ (in the case $h = 1$) or $\mathrm{ZRM}_{2^h}(2,m)$ (in the cases $h > 1$) into cosets of $\mathrm{RM}_{2^h}(1,m)$ is an effective method of isolating those codewords with large values of PMEPR. Indeed, the maximum PMEPR over the entire second-order code space is 2^m, and yet for small values of h and m we typically need reduce this space by a factor of only two or four (losing just one or two encoding bits) to reduce the maximum PMEPR to at most 4.

Based on numerical evidence for the quaternary case we speculate that for all m the maximum PMEPR over any coset of $\mathrm{RM}_4(1,m)$ in $\mathrm{ZRM}_4(2,m)$ is an exact power of 2. Cammarano and Walker [9] have shown that the Golay cosets of Corollary 3.4 always attain the upper bound of 2 on their maximum PMEPR, which establishes this speculation for $m!/2$ of the $2^{m(m-1)/2}$ quaternary cosets. ([9] also shows that the binary Golay cosets of Corollary 3.1 attain the upper bound of 2 on their maximum PMEPR when m is odd and [34] contains further results along these lines.)

We further speculate that a coset of $\mathrm{RM}_4(1,m)$ in $\mathrm{ZRM}_4(2,m)$ having maximum PMEPR of 2^α comprises sequences belonging to a Golay complementary 2^α-tuple (defined analogously to the case $\alpha = 1$ given in Definition 2.1). A straightforward modification of Theorem 2.2 would then give the correct maximum PMEPR. Paterson [34] has made significant progress on this question by showing that each such coset comprises sequences belonging to a Golay complementary 2^β-tuple for some $\beta \geq \alpha$, and that $\beta = \alpha$ in certain cases. These results allow tables such as Table 2 to be predicted at least in part. They also imply good upper bounds, at least for small m, for the values in the corresponding binary tables such as Table 1 without the need for intensive numerical calculation (because the binary coset $g + \mathrm{RM}_2(1,m)$ in $\mathrm{RM}_2(2,m)$ can be embedded in the quaternary coset $2g + \mathrm{RM}_4(1,m)$ in $\mathrm{ZRM}_4(2,m)$). [34] also identifies further coding schemes based on cosets of $\mathrm{RM}_{2^h}(1,m)$.

5 Decoding

An important attraction of the binary Reed-Muller code for applications purposes is that it is easy to decode. In particular, the first-order code $\mathrm{RM}_2(1,m)$ can be decoded very efficiently by means of the fast Hadamard transform (FHT). In this section we give a fast decoding algorithm for $\mathrm{RM}_{2^h}(1,m)$ for any $h \geq 1$, requiring h FHTs and h encoding operations in $\mathrm{RM}_{2^h}(1,m)$. This algorithm acts as a decoder for $\mathrm{RM}_{2^h}(1,m)$ with respect to both Hamming and Lee distance: it always corrects errors of Hamming or Lee weight less than the limit $d/2 = 2^{m-2}$ guaranteed by the minimum Hamming or Lee distance $d = 2^{m-1}$ of the code (see Theorem 3.5). In fact the class of errors which can always be corrected by the algorithm includes many whose Hamming or Lee weight greatly exceeds this limit. The algorithm can be used for soft-decision as well as hard-decision decoding. It is scalable in the sense that the decoder for $\mathrm{RM}_{2^{h+1}}(1,m)$ can be obtained directly from the decoder for $\mathrm{RM}_{2^h}(1,m)$ simply by including one additional iteration. We also extend the decoding algorithm, while maintaining its favourable properties, to deal with an arbitrary union of cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$. This extension efficiently decodes any of the coding schemes of Section 4.

We remark that Ashikhmin and Litsyn [5] give an extension to non-binary cases of the standard FHT method for decoding $\mathrm{RM}_2(1,m)$ but their extension applies to $\mathrm{GRM}(1,m)$ rather than to $\mathrm{RM}_{2^h}(1,m)$ (see Section 3). We also note that van Nee [32] implicitly gives a hard-decision decoder for $\mathrm{RM}_{2^h}(1,3)$ with respect to Hamming (and therefore, by Theorem 3.5, Lee) distance but does not analyse which errors of Hamming weight greater than 1 can be corrected by this decoder and makes no mention of Lee weight.

We begin by summarising the standard FHT method for decoding $\mathrm{RM}_2(1,m)$, as described in [28].

Definition 5.1 *The* Sylvester-Hadamard matrix $H_{2^m} = (H_{ij})$ *of order* 2^m *is given by* $H_{ij} = (-1)^{\sum_{k=1}^{m} i_k j_k}$ *for* $i, j \in \mathbb{Z}_{2^m}$*, where* $(i_1, i_2, \ldots, i_m)$ *and* $(j_1, j_2, \ldots, j_m)$ *are the binary representation of* i *and* j *respectively. The* Hadamard transform *of the row vector* $y = (y_0, y_1, \ldots, y_{2^m-1})$ *is* $\hat{y} = yH_{2^m}$*.*

The Hadamard transform $\hat{y}$ of a sequence y of length 2^m can be calculated rapidly by representing H_{2^m} as the product of m sparse matrices; we then call $\hat{y}$ the *fast Hadamard transform (FHT)* of y. The FHT can be implemented in software with $m2^m$ additions, and in hardware using the *Green machine* with m stages.

If a is a sequence of length n we shall denote by $(a)_i$ the ith element of a for $i = 0, 1, \ldots, n-1$. We shall write $(-1)^a$ for the sequence whose ith element is $(-1)^{(a)_i}$ and write $a \bmod 2^k$ for the sequence whose ith element is $(a)_i \bmod 2^k$ (namely the integer $j \in \mathbb{Z}_{2^k}$ satisfying $(a)_i - j \equiv 0 \pmod{2^k}$).

Now suppose the codeword c of $\mathrm{RM}_2(1,m)$ is received in error as $r = (c+e) \bmod 2$, where e is a sequence over $\mathbb{Z}_2$. The decoding procedure for $\mathrm{RM}_2(1,m)$ calculates the FHT $\hat{y}$ of $(-1)^r$ and determines a value of $j \in \mathbb{Z}_{2^m}$ for which $(\hat{y})_j$ is an element of $\hat{y}$ of largest magnitude. It then sets $w = 0$ or 1 according as $(\hat{y})_j$ is positive or negative, takes $(w_1, w_2, \ldots, w_m)$ to be the binary representation of j, and decodes r as $(\sum_{i=1}^m w_i x_i + w) \bmod 2$. (By truncating intermediate results of the FHT this procedure can actually be implemented in software with fewer than $m2^m$ additions [4].) The decoding procedure relies on the fact that the columns of H_{2^m} together with the columns of $-H_{2^m}$ comprise 2^{m+1} sequences of the form $(-1)^a$, where a ranges over the codewords of $\mathrm{RM}_2(1,m)$. So, in the absence of errors, $(\hat{y})_j$ is $\pm 2^m$ for a unique value $j = J$ and is 0 for each $j \neq J$. The effect of the error e, having Hamming weight $\mathrm{wt}(e)$, is to reduce the magnitude of $(\hat{y})_J$ from 2^m by exactly $2\,\mathrm{wt}(e)$ and to increase the magnitude of $(\hat{y})_j$ for each $j \neq J$ from 0 by at most the same amount $2\,\mathrm{wt}(e)$. Therefore provided $\mathrm{wt}(e) < 2^{m-2}$ the decoding procedure correctly decodes r to c. (See Section 2 for a discussion of the relationship between Boolean functions and binary representations.)

The following definition will be useful in describing the decoding algorithm for $\mathrm{RM}_{2^h}(1,m)$.

Definition 5.2 *Let $a = (a_0, a_1, \ldots, a_{n-1})$ be an integer sequence and let i be an integer. We define* $\mathrm{wt}_{2^k}(i)$ *to be* $\min(i \bmod 2^k, 2^k - (i \bmod 2^k))$ *and* $\mathrm{wt}_{2^k}(a)$ *to be* $\sum_{i=0}^{n-1} \mathrm{wt}_{2^k}(a_i)$.

$\mathrm{wt}_{2^k}(a)$ is equal to the *Lee weight over* $\mathbb{Z}_{2^k}$ of the sequence $a \bmod 2^k$ (see Section 3).

We now introduce the decoding algorithm by outlining the octary case $h = 3$. Suppose the codeword $c \in \mathrm{RM}_8(1,m)$ is received in error as $r = (c+e) \bmod 8$, where e is a sequence over $\mathbb{Z}_8$. Write $c = (\sum_{i=1}^m u_i x_i + u) \bmod 8$, where $u_i, u \in \mathbb{Z}_8$. Let (v_{i2}, v_{i1}, v_{i0}) be the binary representation of u_i and let (v_2, v_1, v_0) be the binary representation of u, so that $u_i = 4v_{i2} + 2v_{i1} + v_{i0}$ and $u = 4v_2 + 2v_1 + v_0$. Then

$$c = (4f_2 + 2f_1 + f_0) \bmod 8, \tag{13}$$

where

$$f_2 = (\sum_{i=1}^{m} v_{i2}x_i + v_2) \bmod 2, \tag{14}$$

$$f_1 = (\sum_{i=1}^{m} v_{i1}x_i + v_1) \bmod 4, \tag{15}$$

$$f_0 = (\sum_{i=1}^{m} v_{i0}x_i + v_0) \bmod 8. \tag{16}$$

Write the error e uniquely as $e = 4e_2 + 2e_1 + e_0$, where each e_k is a sequence over $\mathbb{Z}_2$, so that

$$r = (4(f_2 + e_2) + 2(f_1 + e_1) + (f_0 + e_0)) \bmod 8. \tag{17}$$

Using the FHT, the decoding algorithm recovers the value f_0 by reducing modulo 2, then (assuming f_0 has been determined correctly) the value f_1 by reducing modulo 4, and finally (assuming f_0 and f_1 have been determined correctly) the value f_2; c is then recovered from (13).

Now $r \bmod 2 = (f_0 \bmod 2 + e_0) \bmod 2$, and we know from (16) that $f_0 \bmod 2$ is a codeword in $\mathrm{RM}_2(1,m)$. Therefore provided $\mathrm{wt}_2(e_0) < 2^{m-2}$ we can use the standard binary decoder for $\mathrm{RM}_2(1,m)$ to recover the binary coefficients v_{i0}, v_0 for $f_0 \bmod 2$, and then calculate f_0 from (16).

We next set $r_1 = (r - f_0) \bmod 8$. From (17), $r_1 \bmod 4 = (2(f_1 \bmod 2) + (2e_1 + e_0)) \bmod 4$. From (15), $f_1 \bmod 2$ is a codeword in $\mathrm{RM}_2(1,m)$. We define the sequence y by $(y)_i = 1 - \mathrm{wt}_4((r_1)_i)$ for $i = 0, 1, \ldots, 2^m - 1$ and take $\hat{y}$ to be the FHT of y. Now if $e_0 = 0$ then $y = (-1)^{(f_1 + e_1) \bmod 2}$

and so this stage of the algorithm simply decodes $f_1 \bmod 2$ in the presence of the error e_1 using the standard binary method; $\hat{y}_j$ is $\pm 2^m$ for a unique value $j = J$ and is 0 for each $j \neq J$. However if $e_0 \neq 0$ then $(y)_i = 0$ for all positions i such that $(e_0)_i = 1$. This effectively removes from consideration those elements of y identified as error positions by the FHT from the previous stage. We shall show that the effect of the error e is to reduce the magnitude of $(\hat{y})_J$ from 2^m by exactly $\mathrm{wt}_4(2e_1 + e_0)$, and to increase the magnitude of $(\hat{y})_j$ for each $j \neq J$ from 0 by at most the same amount $\mathrm{wt}_4(2e_1 + e_0)$. Therefore provided $\mathrm{wt}_4(2e_1 + e_0) < 2^{m-1}$ we can recover the binary coefficients v_{i1}, v_1 for $f_1 \bmod 2$ from the position and sign of the transform sequence element of largest magnitude, and then calculate f_1 from (15).

The last stage of the decoding algorithm is to set $r_2 = (r_1 - 2f_1) \bmod 8$. From (17), $r_2 = (4(f_2 \bmod 2) + (4e_2 + 2e_1 + e_0)) \bmod 8$, and from (14), $f_2 \bmod 2$ is a codeword in $\mathrm{RM}_2(1, m)$. We define the sequence y by $(y)_i = 2 - \mathrm{wt}_8((r_2)_i)$ for $i = 0, 1, \ldots, 2^m - 1$ and take $\hat{y}$ to be the FHT of y. If $e_1 = e_0 = 0$ then $y = 2(-1)^{(f_2+e_2) \bmod 2}$ so that this stage reduces to the standard decoding of $f_2 \bmod 2$ in the presence of the error e_2. Otherwise $(y)_i$ takes the value $1, 0$ or -1 for all positions i such that $(2e_1 + e_0)_i \neq 0$; this modifies the result of the FHT according to the error positions identified by both of the previous FHTs. We shall show that provided $\mathrm{wt}_8(4e_2 + 2e_1 + e_0) < 2 \cdot 2^{m-1}$ we can recover $f_2 \bmod 2$ and hence f_2.

Finally we recover c from (13). The conditions for correctly decoding $c + e$ to c are: $\mathrm{wt}_2(e) < 2^{m-2}$, $\mathrm{wt}_4(e) < 2^{m-1}$, and $\mathrm{wt}_8(e) < 2^m$.

We now give a formal description of the decoding algorithm for any value of $h \geq 1$.

Algorithm 5.3 *(Decoding algorithm for* $\mathrm{RM}_{2^h}(1, m)$*)*

1. *Input the received codeword r as a sequence over $\mathbb{Z}_{2^h}$ of length 2^m. Set $k = 0$ and $r_0 = r$.*

2. *Define the sequence y by $(y)_i = 2^{k-1} - \mathrm{wt}_{2^{k+1}}((r_k)_i)$ for $i = 0, 1, \ldots, 2^m - 1$.*

3. *Let $\hat{y}$ be the* FHT *of y and determine a value of $j \in \mathbb{Z}_{2^m}$ for which $(\hat{y})_j$ is an element of $\hat{y}$ of largest magnitude. Let w be 0 or 1 according as $(\hat{y})_j$ is positive or negative, and let $(w_1, w_2, \ldots, w_m)$ be the binary representation of j. Set $f_k = (\sum_{i=1}^m w_i x_i + w) \bmod 2^{h-k}$.*

4. *If $k = h - 1$ then output the decoded codeword $(2^{h-1} f_{h-1} + 2^{h-2} f_{h-2} + \ldots + f_0) \bmod 2^h$. Else set $r_{k+1} = (r_k - 2^k f_k) \bmod 2^h$, then increment k and go to Step 2.*

Theorem 5.4 *Let c be a codeword of* $\mathrm{RM}_{2^h}(1, m)$ *and let e be a sequence over $\mathbb{Z}_{2^h}$. Given the input* $(c + e) \bmod 2^h$, *Algorithm* 5.3 *outputs c provided* $\mathrm{wt}_{2^{k+1}}(e) < 2^{m+k-2}$ *for $k = 0, 1, \ldots, h - 1$.*

<u>**Proof:**</u> Write $c = (\sum_{i=1}^m u_i x_i + u) \bmod 2^h$, where $u_i, u \in \mathbb{Z}_{2^h}$. Let $(v_{i,h-1}, v_{i,h-2}, \ldots, v_{i0})$ be the binary representation of u_i and let $(v_{h-1}, v_{h-2}, \ldots, v_0)$ be the binary representation of u, so that $u_i = 2^{h-1} v_{i,h-1} + 2^{h-2} v_{i,h-2} + \ldots + v_{i0}$ and $u = 2^{h-1} v_{h-1} + 2^{h-2} v_{h-2} + \ldots + v_0$. Then $c = (2^{h-1} f_{h-1} + 2^{h-2} f_{h-2} + \ldots + f_0) \bmod 2^h$, where

$$f_k = (\sum_{i=1}^m v_{ik} x_i + v_k) \bmod 2^{h-k} \tag{18}$$

for $k = 0, 1, \ldots, h - 1$. Write the error e uniquely as

$$e = 2^{h-1} e_{h-1} + 2^{h-2} e_{h-2} + \ldots + e_0, \tag{19}$$

where each e_k is a sequence over $\mathbb{Z}_2$, so that the received codeword $r = (c + e) \bmod 2^h$ is given by

$$r = (2^{h-1}(f_{h-1} + e_{h-1}) + 2^{h-2}(f_{h-2} + e_{h-2}) + \ldots + (f_0 + e_0)) \bmod 2^h. \tag{20}$$

The algorithm has h passes $0, 1, \ldots, h - 1$, and on pass k we determine the value of f_k. Assume that the values $f_0, f_1, \ldots, f_{k-1}$ have been correctly determined. Then Step 4 shows that

$r_k \bmod 2^{k+1} = (r - f_0 - 2f_1 - 2^2 f_2 - \ldots - 2^{k-1} f_{k-1}) \bmod 2^{k+1}$, and by (19) and (20) we obtain $r_k \bmod 2^{k+1} = (2^k(f_k \bmod 2) + e \bmod 2^{k+1}) \bmod 2^{k+1}$. Now it is straightforward to verify the identity

$$2^{k-1} - \mathrm{wt}_{2^{k+1}}(2^k\alpha + \beta) \equiv (-1)^\alpha (2^{k-1} - \mathrm{wt}_{2^{k+1}}(\beta)) \text{ for all } \alpha \in \mathbb{Z}_2,\ \beta \in \mathbb{Z}_{2^{k+1}}$$

for any integer $k \geq 0$. Therefore by Step 2 we have $(y)_i = (-1)^{(f_k \bmod 2)_i}(2^{k-1} - \mathrm{wt}_{2^{k+1}}((e)_i))$. Since $(\hat{y})_j = \sum_{i=0}^{2^m-1}(y)_i H_{ij}$, where $H = (H_{ij})$ is the Sylvester-Hadamard matrix of order 2^m, we then have

$$\begin{aligned}(\hat{y})_j &= 2^{k-1}\sum_{i=0}^{2^m-1}(-1)^{(f_k \bmod 2)_i} H_{ij} - \sum_{i=0}^{2^m-1}(-1)^{(f_k \bmod 2)_i} H_{ij}\,\mathrm{wt}_{2^{k+1}}((e)_i) \\ &= 2^{k-1}((-1)^{f_k \bmod 2} H)_j - \sum_{i=0}^{2^m-1} d_{ij}\,\mathrm{wt}_{2^{k+1}}((e)_i), \end{aligned} \tag{21}$$

where each $d_{ij} = (-1)^{(f_k \bmod 2)_i} H_{ij}$ takes the value 1 or -1. Since $f_k \bmod 2$ is a codeword in $\mathrm{RM}_2(1,m)$, $((-1)^{f_k \bmod 2} H)_j$ is $\pm 2^m$ for a unique value $j = J$ and is 0 for each $j \neq J$. Therefore either $d_{iJ} = 1$ for all i or $d_{iJ} = -1$ for all i. We then see from (21) that the effect of the error e is to reduce the magnitude of $(\hat{y})_j$ from $2^{k-1} \cdot 2^m$ by exactly $\mathrm{wt}_{2^{k+1}}(e)$ for a unique value $j = J$, and to increase the magnitude of $(\hat{y})_j$ for each $j \neq J$ from 0 by at most the same amount. By assumption $\mathrm{wt}_{2^{k+1}}(e) < 2^{m+k-2}$, so we can recover the binary coefficients v_{ik}, v_k for $f_k \bmod 2$ from the position and sign of the transform sequence element of largest magnitude, and then calculate f_k from (18). □

Note that when $k = 0$, Step 2 of Algorithm 5.3 sets $y = (-1)^{r \bmod 2}/2$, so pass 0 of the algorithm is the standard binary decoder for $\mathrm{RM}_2(1,m)$ except that the values $\pm 1/2$ are used instead of ± 1. For implementation convenience we can choose to work with $2y$ instead of y on pass 0. Note also that we can choose in Step 3 to calculate f_k modulo 2^h rather than modulo 2^{h-k} without affecting the result.

Corollary 5.5 *Algorithm* 5.3 *acts as a decoder for* $\mathrm{RM}_{2^h}(1,m)$ *with respect to Hamming distance and with respect to Lee distance.*

Proof: Let c be a codeword of $\mathrm{RM}_{2^h}(1,m)$ and let e be a transmission error having Hamming weight $\mathrm{wt}(e)$. By Theorem 3.5 it is sufficient to show that Algorithm 5.3 correctly decodes $(c + e) \bmod 2^h$ to c provided that $\mathrm{wt}(e) < 2^{m-2}$. This follows from Theorem 5.4 by noting that $\mathrm{wt}_{2^{k+1}}(e) \leq 2^k\,\mathrm{wt}(e)$ for $k = 0, 1, \ldots, h-1$. □

The full power of Algorithm 5.4 is demonstrated not by Corollary 5.5 but by Theorem 5.4. For example, consider the octary case $h = 3$ with $m = 4$. Theorem 3.5 and Corollary 5.5 guarantee only that an error of Hamming (or Lee) weight at most 3 can be corrected and yet by Theorem 5.4 the error $e = (4002101000760400)$, having Hamming weight 7 and Lee weight 15, can be corrected using Algorithm 5.3 because it satisfies $\mathrm{wt}_2(e) = 3$, $\mathrm{wt}_4(e) = 7$ and $\mathrm{wt}_8(c) = 15$. We now illustrate the use of the decoding algorithm for these values of h, m and e, taking the codeword c to be $5x_1 + 7x_2 + 3x_3 + 6x_4 + 6 = (6417530631642053)$. The received codeword is $r_0 = (c + e) \bmod 8 = (2411631631522453)$. On pass $k = 0$ we find $2y = (1,1,-1,-1,1,-1,-1,1,-1,-1,-1,1,1,1,-1,-1)$ and

$$\widehat{2y} = (-2,-2,6,6,-2,-2,-2,-2,2,2,2,2,2,2,10,-6).$$

We therefore set $f_0 = (x_1 + x_2 + x_3) \bmod 8 = (0011112211222233)$ and $r_1 = (r_0 - f_0) \bmod 8 = (2400527420300220)$. On pass $k = 1$ we find $y = (-1,1,1,1,0,-1,0,1,-1,1,0,1,1,-1,-1,1)$ and

$$\hat{y} = (3,-5,-5,3,3,-5,-1,-9,1,1,-3,-3,1,1,1,1).$$

We therefore set $f_1 = (x_2+x_3+x_4+1) \bmod 4 = (1223233012232330)$ and $r_2 = (r_1 - 2f_1) \bmod 8 = (0042141404724440)$. On pass $k = 2$ we find $y = (2,2,-2,0,1,-2,1,-2,2,-2,1,0,-2,-2,-2,2)$ and

$$\hat{y} = (-3,5,1,9,9,1,9,1,3,3,11,-5,-1,-17,3,3).$$

We therefore set $f_2 = (x_1+x_2+x_4+1) \bmod 2 = (1010010101011010)$. The output of the decoding algorithm is $(4f_2 + 2f_1 + f_0) \bmod 8 = (6417530631642053)$, which is the original codeword.

Under the encoding schemes of Section 4 information symbols $u_i, u \in \mathbb{Z}_{2^h}$ are used to form the codeword $(\sum_{i=1}^m u_i x_i + u) \bmod 2^h$ of $\mathrm{RM}_{2^h}(1,m)$. These information symbols can be recovered directly using the above decoding algorithm: in the above example the output is determined as $(4(x_1 + x_2 + x_4 + 1) + 2(x_2 + x_3 + x_4 + 1) + (x_1 + x_2 + x_3)) \bmod 8 = (5x_1 + 7x_2 + 3x_3 + 6x_4 + 6) \bmod 8$. Furthermore the binary representation of the information symbols u_i, u gives the original information bits, so these can also be recovered directly from the algorithm as the coefficients v_{ik}, v_k for $k = 0,1,\ldots,h-1$. Now pass k of the algorithm can incorrectly determine the value f_k if the error e does not satisfy $\mathrm{wt}_{2^{k+1}}(e) < 2^{m+k-2}$. If this happens then subsequent passes can incorrectly determine the values $f_{k+1}, f_{k+2}, \ldots, f_{h-1}$ so that the decoded codeword can have large Lee distance from the original codeword. However provided the values $f_0, f_1, \ldots, f_{k-1}$ are all correctly determined then at least $k(m+1)$ information bits (namely the coefficients v_{ij}, v_j for $i = 1,2,\ldots,m$ and $j = 0,1,\ldots k-1$) out of the original $h(m+1)$ will be correctly determined.

The principal computational requirement for Algorithm 5.3 is h integer-valued FHTs and h summations of the form $(\sum_{i=1}^m w_i x_i + w) \bmod 2^h$. Each summation can be calculated using whatever software or hardware procedure is used to encode the information symbols u_i, u as the element $(\sum_{i=1}^m u_i x_i + u) \bmod 2^h$ of $\mathrm{RM}_{2^h}(1,m)$.

We have presented Algorithm 5.3 as a hard-decision decoder (acting on a sequence whose elements are integers in $\mathbb{Z}_{2^h}$), but it can also be used as a soft-decision decoder (acting on a sequence whose elements are real numbers in the range $[0, 2^h)$). We simply need to extend Definition 5.2 for $\mathrm{wt}_{2^k}(i)$ to deal with real-valued i by taking $i \bmod 2^k$ to be the real number j in the range $[0, 2^k)$ satisfying $i - j \equiv 0 \pmod{2^k}$.

Algorithm 5.3 can be modified as follows. Replace the definition of y in Step 2 by $v = (r_k \bmod 2^{k+1})/2^k$ and $y = (-1)^v$, calculate $e_k = (v + f_k) \bmod 2$ at the end of Step 3, and replace the equation for r_{k+1} in Step 4 by $r_{k+1} = (r_k - 2^k(f_k + e_k)) \bmod 2^h$. Then on pass k, assuming $f_0, f_1, \ldots, f_{k-1}$ have been determined correctly, Step 2 sets $y = (-1)^{(f_k+e_k) \bmod 2}$ and Step 3 uses the standard binary decoder for $\mathrm{RM}_2(1,m)$ to find $f_k \bmod 2$ (and hence f_k) and e_k. The modified conditions for correcting the error e defined by (19) are $\mathrm{wt}_2(e_k) < 2^{m-2}$ for $k = 0,1,\ldots,h-1$. Both the original Algorithm 5.3 and this modification act as decoders for $\mathrm{RM}_{2^h}(1,m)$ with respect to Hamming and Lee distance; beyond the limit guaranteed by the minimum distance of the code both perform well but neither one is uniformly better than the other.

We now extend Algorithm 5.3 to decode efficiently the non-binary ($h > 1$) coding schemes of Section 4, most of which involve the union of several cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$. The *supercode* decoding method for decoding the union of cosets of a code C, as described in [11] for binary codes, involves subtracting each possible coset representative in turn from the received codeword and decoding the result as an element of C; the best decoding result in C determines the coset representative. We shall modify this method by interleaving the subtraction of the coset representatives with the h passes of Algorithm 5.3 to give a substantially faster algorithm than would be obtained by applying Algorithm 5.3 in full to each coset of $\mathrm{RM}_{2^h}(1,m)$.

Algorithm 5.6 *(Decoding algorithm for a union of cosets of* $\mathrm{RM}_{2^h}(1,m)$ *in* $\mathrm{ZRM}_{2^h}(2,m)$*)*

1. *Input the received codeword* r *as a sequence over* $\mathbb{Z}_{2^h}$ *of length* 2^m *and input the predetermined set* $G = \{g\}$ *of coset representatives of* $\mathrm{RM}_{2^h}(1,m)$ *in* $\mathrm{ZRM}_{2^h}(2,m)$*. Set* $k = 0$ *and* $r_0 = r$.

2. *Let* $\{u_1, u_2, \ldots, u_s\}$ *be the distinct values of* $g \bmod 2^{k+1}$ *as* g *takes all values in* G*. Set* $l = 1$ *and* $Y = 0$.

3. *Define the sequence* y *by* $(y)_i = 2^{k-1} - \mathrm{wt}_{2^{k+1}}((r_k - u_l)_i)$ *for* $i = 0,1,\ldots,2^m - 1$.

4. *Let $\hat{y}$ be the* FHT *of y and determine a value of $j \in \mathbb{Z}_{2^m}$ for which $(\hat{y})_j$ is an element of $\hat{y}$ of largest magnitude.*

5. *If $|(\hat{y})_j| > |Y|$ then set $Y = (\hat{y})_j$, $J = j$ and $L = l$.*

6. *If $|Y| > 3 \cdot 2^{m+k-3}$ or $l = s$ then go to Step 7. Else increment l and go to Step 3.*

7. *Let w be 0 or 1 according as Y is positive or negative, and let $(w_1, w_2, \ldots, w_m)$ be the binary representation of J. Set $f_k = (\sum_{i=1}^{m} w_i x_i + w) \bmod 2^{h-k}$. Remove from G each coset representative g for which $g \bmod 2^{k+1} \neq u_L$.*

8. *If $k = h - 1$ then output the decoded codeword $(g + 2^{h-1} f_{h-1} + 2^{h-2} f_{h-2} + \ldots + f_0) \bmod 2^h$ for the single remaining $g \in G$. Else set $r_{k+1} = (r_k - 2^k f_k) \bmod 2^h$, then increment k and go to Step 2.*

Theorem 5.7 *Let $G = \{g\}$ be a set of coset representatives of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$, let c be a codeword of the code $\{g + \mathrm{RM}_{2^h}(1,m) \mid g \in G\}$ and let e be a sequence over $\mathbb{Z}_{2^h}$. Given the input $(c + e) \bmod 2^h$, Algorithm 5.6 outputs c provided that for $k = 0, 1, \ldots, h-1$,*

$$\mathrm{wt}_{2^{k+1}}(e) < \begin{cases} 2^{m+k-3} & \textit{if } G \textit{ contains } g, g' \textit{ which are equal modulo } 2^k \textit{ but distinct modulo } 2^{k+1} \\ 2^{m+k-2} & \textit{otherwise.} \end{cases}$$

Proof: The proof is similar to that of Theorem 5.4. Write $c = (g + \sum_{i=1}^{m} u_i x_i + u) \bmod 2^h$, where $u_i, u \in \mathbb{Z}_{2^h}$ and $g \in G$. Write g uniquely as $g = 2^{h-1} g_{h-1} + 2^{h-2} g_{h-2} + \ldots + g_0$, where each g_k is a sequence over $\mathbb{Z}_2$. Then $c = (g + 2^{h-1} f_{h-1} + 2^{h-2} f_{h-2} + \ldots + f_0) \bmod 2^h$ and the received codeword $r = (c + e) \bmod 2^h$ is given by

$$r = (2^{h-1}(g_{h-1} + f_{h-1} + e_{h-1}) + 2^{h-2}(g_{h-2} + f_{h-2} + e_{h-2}) + \ldots + (g_0 + f_0 + e_0)) \bmod 2^h,$$

where f_k and e_k are as previously.

The algorithm has h passes $0, 1, \ldots, h-1$, and on pass k we determine the value of f_k and g_k and discard any $g' \in G$ for which $g'_k \neq g_k$. On pass k Steps 3 to 6 perform a FHT for each remaining group of coset representatives in G having the same value modulo 2^{k+1}, and select one such group by finding a transform sequence element of largest magnitude amongst all the FHTs. If a sufficiently large magnitude is encountered then Step 6 truncates the selection procedure. Assume that the values $f_0, f_1, \ldots, f_{k-1}$ and $g_0, g_1, \ldots, g_{k-1}$ have been correctly determined. Note that all the remaining coset representatives in G must be equal modulo 2^k. If they are also all equal modulo 2^{k+1} then g_k is determined and f_k can be recovered as in the proof of Theorem 5.4 because by assumption $\mathrm{wt}_{2^{k+1}}(e) < 2^{m+k-2}$. Therefore assume that G contains a coset representative g' for which $g' \bmod 2^{k+1} = 2^k g'_k + 2^{k-1} g_{k-1} + 2^{k-2} g_{k-2} + \ldots + g_0$, where $g'_k \neq g_k$.

Suppose that Step 3 selects the value $u_l = g' \bmod 2^{k+1}$. Then Step 8 shows that $(r_k - u_l) \bmod 2^{k+1} = (2^k((g_k - g'_k + f_k) \bmod 2) + e \bmod 2^{k+1}) \bmod 2^{k+1}$. By a similar argument to that used previously it follows that

$$(\hat{y})_j = 2^{k-1}((-1)^{(g_k - g'_k + f_k) \bmod 2} H)_j - \sum_{i=0}^{2^m-1} d_{ij}\, \mathrm{wt}_{2^{k+1}}((e)_i), \tag{22}$$

where each $d_{ij} = (-1)^{((g_k - g'_k + f_k) \bmod 2)_i} H_{ij}$ takes the value 1 or -1 and $H = (H_{ij})$ is the Sylvester-Hadamard matrix of order 2^m. Now $f_k \bmod 2$ is a codeword in $\mathrm{RM}_2(1,m)$ and we see (by expressing g_k and g'_k in similar manner to (18)) that $(g_k - g'_k) \bmod 2$ is a codeword in $\mathrm{RM}_2(2,m) \backslash \mathrm{RM}_2(1,m)$. Since the minimum Hamming distance of $\mathrm{RM}_2(2,m)$ is 2^{m-2} we conclude that $((-1)^{(g_k - g'_k + f_k) \bmod 2} H)_j$ has magnitude at most $2^m - 2 \cdot 2^{m-2} = 2^{m-1}$ for each j. (22) then implies that $(\hat{y})_j$ has magnitude at most $2^{m+k-2} + \mathrm{wt}_{2^{k+1}}(e)$ for each j.

In contrast if Step 3 selects the value $u_l = g \bmod 2^{k+1}$ we know from the proof of Theorem 5.4 that $(\hat{y})_j$ has magnitude exactly $2^{m+k-1} - \mathrm{wt}_{2^{k+1}}(e)$ for a unique value of j and has magnitude at most $\mathrm{wt}_{2^{k+1}}(e)$ for each other j. By assumption $\mathrm{wt}_{2^{k+1}}(e) < 2^{m+k-3}$ and therefore we can recover f_k and g_k. Furthermore if $(\hat{y})_j$ has magnitude greater than $2^{m+k-1} - 2^{m+k-3} = 3 \cdot 2^{m+k-3}$ then there is no need to consider any further values $u_{l+1}, u_{l+2}, \ldots, u_s$. □

Corollary 5.8 *Algorithm* 5.6 *acts as a decoder for an arbitrary union of cosets of* $\mathrm{RM}_{2^h}(1,m)$ *in* $\mathrm{ZRM}_{2^h}(2,m)$ *with respect to Hamming distance and with respect to Lee distance.*

Proof: The proof for Hamming distance follows from Theorem 3.5 in similar manner to the proof of Corollary 5.5. For Lee distance, note that the condition for $k = 0$ in Theorem 5.7 is $\mathrm{wt}_2(e) < 2^{m-2}$ because all coset representatives of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$ are equal modulo 2. The result follows from Theorems 3.5 and 5.7 since $\mathrm{wt}_{2^{k+1}}(e) \le \mathrm{wt}_{2^h}(e)$ for $k = 0, 1, \ldots, h-1$ and the Lee weight over $\mathbb{Z}_{2^h}$ of e is $\mathrm{wt}_{2^h}(e)$. □

The number of encoding operations in $\mathrm{RM}_{2^h}(1,m)$ required by Algorithm 5.6 is h. The number of FHTs required is at least h and at most $h + |G| - 1$: if $g, g' \in G$ are equal modulo 2^k but distinct modulo 2^{k+1} then the algorithm can choose between them using two FHTs. In fact the expected number of FHTs can be less than $h + (|G|-1)/2$ because the algorithm can choose between groups of coset representatives. For example, consider the code to be the union of the first 32 cosets of $\mathrm{RM}_8(1,4)$ in $\mathrm{ZRM}_8(2,4)$ listed in Table 3 (given as Option 4 in Table 6) and suppose the actual coset representative is not one of the first twelve of the list. Since these twelve cosets are all equal modulo 4 they can be eliminated from consideration with a single FHT on pass 1. Algorithm 5.6 can be further speeded up by calculating in parallel those FHTs which choose between groups of coset representatives.

The decoded coset representative g can be output separately by Algorithm 5.6. The information bits used in any of the encoding schemes of Section 4 to select a coset representative (or an ordered pair of coset representatives, in the case of a composition coding scheme) can be found by inverting the encoding look-up table.

For some codes the decoding procedure of Algorithm 5.6 can be applied with a stronger truncation criterion in Step 6. Suppose that $m \ge 4$ is even and that each coset representative in G is of the form $2^{h-1} g_{h-1}$, where the binary coset $g_{h-1} + \mathrm{RM}_2(1,m)$ belongs to a Kerdock code of length 2^m. Then for distinct $2^{h-1} g_{h-1}, 2^{h-1} g'_{h-1}$ in G we know from Section 3 that $(g_{h-1} - g'_{h-1} + f_{h-1}) \bmod 2$ is a bent function and therefore that $((-1)^{(g_{h-1} - g'_{h-1} + f_{h-1}) \bmod 2} H)_j$ has magnitude $2^{m/2}$ for all j. Then, following the proof of Theorem 5.7, the truncation criterion takes effect only on pass $h-1$ and can be improved to $|Y| > 2^{h-3}(2^m + 2^{m/2})$; the conditions for correcting the error e are

$$\mathrm{wt}_{2^{k+1}}(e) < \begin{cases} 2^{h-3}(2^m - 2^{m/2}) & \text{for } k = h-1 \\ 2^{m+k-2} & \text{for } k = 0, 1, \ldots, h-2. \end{cases}$$

For example, Option 2 of Table 6, described in Section 4, is derived from such a code with $h = 3$ and $m = 4$. The truncation criterion for pass 2 of the decoding algorithm for this coding scheme can be taken as $|Y| > 20$ instead of $|Y| > 24$ and the conditions for correcting e are $\mathrm{wt}_2(e) < 4$ and $\mathrm{wt}_4(e) < 8$ (as before), together with $\mathrm{wt}_8(e) < 12$ instead of $\mathrm{wt}_8(e) < 8$.

Algorithm 5.6 can be used for soft-decision as well as hard-decision decoding. It can also be modified, in similar manner to the modification of Algorithm 5.3 described earlier, to act as an alternative decoder for a union of cosets of $\mathrm{RM}_{2^h}(1,m)$ in $\mathrm{ZRM}_{2^h}(2,m)$ with respect to Hamming and Lee distance. Replace the definition of y in Step 3 by $v = ((r_k - u_l) \bmod 2^{k+1})/2^k$ and $y = (-1)^v$, calculate $e_k = (v + f_k) \bmod 2$ at the end of Step 7, and replace the equation for r_{k+1} in Step 8 by $r_{k+1} = (r_k - 2^k(f_k + e_k)) \bmod 2^h$. The conditions for correcting the error e are then

$$\mathrm{wt}_2(e_k) < \begin{cases} 2^{m-3} & \text{if } G \text{ contains } g, g' \text{ which are equal modulo } 2^k \text{ but distinct modulo } 2^{k+1} \\ 2^{m-2} & \text{otherwise} \end{cases}$$

for $k = 0, 1, \ldots, h-1$.

The decoding procedure of Algorithm 5.6 can be used for an arbitrary union of cosets of $\mathrm{RM}_{2^h}(1, m)$, not necessarily contained within $\mathrm{ZRM}_{2^h}(2, m)$, by modifying the truncation criterion in Step 6 according to the error correction properties of the selected code.

6 Conclusion

The connection between Golay complementary sequences and second-order Reed-Muller codes, together with the coset ordering process, are the keys to obtaining the range of OFDM coding schemes with favourable properties described here. These schemes are highly suitable for practical purposes owing to the efficient decoding algorithms involving multiple fast Hadamard transforms.

We have shown that linear codes over rings, as introduced in [21], arise naturally as solutions to the OFDM power envelope problem. We have also shown that certain Golay sequences possess a high degree of intrinsic structure, whereas many other sequences defined by aperiodic autocorrelation constraints appear not to do so.

Jones and Wilkinson [personal communication, 1997] have demonstrated the potential improvement offered by certain of the OFDM coding schemes of this paper by simulating their end-to-end system performance in a typical indoor radio environment. Jones, Wilkinson and Castle [23] have shown experimentally that a representative one of the coding schemes presented here offers superior adjacent channel interference performance as compared with conventional OFDM coding schemes. Paterson [34] has developed and extended many of the ideas of this paper and has given a theoretical explanation for some of our numerical results.

Acknowledgements

We are grateful to Alan Jones and Tim Wilkinson for initiating this research and for their advice and encouragement throughout. We wish to thank Kenny Paterson for numerous helpful discussions and suggestions which we feel have greatly improved our work. James Davis thanks Hewlett-Packard for generous hospitality and support during his sabbatical year 1995–6 in Bristol and during Summer 1997.

References

[1] M. Alard and R. Lassalle. Principles of modulation and channel coding for digital broadcasting for mobile receivers. *EBU Review*, No. 224:47–69, August 1987.

[2] M. Aldinger. Multicarrier COFDM scheme in high bitrate radio local area networks. In *5th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun., The Hague*, pages 969–973, Sept 1994.

[3] J. Aldis. The WAND Project physical layer — an option for HIPERLAN Type 2. ETSI BRAN Working Group 3 Temporary Document 05, June 1997.

[4] A.E. Ashikhmin and S.N. Litsyn. Fast decoding algorithms for first order Reed-Muller and related codes. *Designs, Codes and Cryptography*, **7**:187–214, 1996.

[5] A.E. Ashikhmin and S.N. Litsyn. Fast decoding of non-binary first order Reed-Muller codes. *Appl. Algebra Eng. Commun. Computing*, **7**:299–308, 1996.

[6] J.A.C. Bingham. Multicarrier modulation for data transmission: an idea whose time has come. *IEEE Commun. Magazine*, **28**:5–14, May 1990.

[7] S. Boyd. Multitone signals with low crest factor. *IEEE Trans. Circuits and Systems*, **CAS-33**:1018–1022, 1986.

[8] S.Z. Budišin. New complementary pairs of sequences. *Electron. Lett.*, **26**:881–883, 1990.

[9] M.W. Cammarano and M.L. Walker. Integer maxima in power envelopes of Golay codewords. Preprint, University of Richmond, 1997.

[10] L.J. Cimini, Jr. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Trans. Commun.*, **COM-33**:665–675, 1985.

[11] J.H. Conway and N.J.A. Sloane. Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice. *IEEE Trans. Inf. Theory*, **IT-32**:41–50, 1986.

[12] J.A. Davis and J. Jedwab. Peak-to-mean power control and error correction for OFDM transmission using Golay sequences and Reed-Muller codes. *Electron. Lett.*, **33**:267–268, 1997.

[13] S. Eliahou, M. Kervaire, and B. Saffari. A new restriction on the lengths of Golay complementary sequences. *J. Combin. Theory (A)*, **55**:49–59, 1990.

[14] P. Fan and M. Darnell. *Sequence Design for Communications Applications*. Communications Systems, Techniques and Applications. Research Studies Press, Taunton, 1996.

[15] M. Friese. Multicarrier modulation with low peak-to-mean average power ratio. *Electron. Lett.*, **32**:713–714, 1996.

[16] M. Friese. Multitone signals with low crest factor. *IEEE Trans. Commun.*, **45**:1338–1344, 1997.

[17] M.J.E. Golay. Multislit spectroscopy. *J. Opt. Soc. Amer.*, **39**:437–444, 1949.

[18] M.J.E. Golay. Static multislit spectrometry and its application to the panoramic display of infrared spectra. *J. Opt. Soc. Amer.*, **41**:468–472, 1951.

[19] M.J.E. Golay. Complementary series. *IRE Trans. Inform. Theory*, **IT-7**:82–87, 1961.

[20] M.J.E. Golay. Sieves for low autocorrelation binary sequences. *IEEE Trans. Inform. Theory*, **IT-23**:43–51, 1977.

[21] A.R. Hammons, Jr., P.V. Kumar, A.R. Calderbank, N.J.A. Sloane, and P. Solé. The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes. *IEEE Trans. Inform. Theory*, **40**:301–319, 1994.

[22] J. Jedwab, A. Jones, K. Paterson, T. Wilkinson, and J. Davis. Options for variable rate coding in OFDM using binary, quaternary and octary modulation. ETSI BRAN Working Group 3 Temporary Document 50, Sept 1997.

[23] A. Jones, T. Wilkinson, and R. Castle. Adjacent channel interference performance of coded OFDM. ETSI BRAN Working Group 3 Temporary Document 79, Oct 1997.

[24] A.E. Jones and T.A. Wilkinson. Combined coding for error control and increased robustness to system nonlinearities in OFDM. In *IEEE 46th Vehicular Technology Conf., Atlanta*, pages 904–908, April–May 1996.

[25] A.E. Jones, T.A. Wilkinson, and S.K. Barton. Block coding scheme for reduction of peak to mean envelope power ratio of multicarrier transmission schemes. *Electron. Lett.*, **30**:2098–2099, 1994.

[26] X. Li and L.J. Cimini, Jr. Effects of clipping and filtering on the performance of OFDM. In *IEEE 47th Vehicular Technology Conf., Phoenix*, pages 1634–1638, May 1997.

[27] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, Berlin, 2nd edition, 1992.

[28] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1986.

[29] J.T.E. McDonnell and T.A. Wilkinson. Comparison of computational complexity of adaptive equalization and OFDM for indoor wireless networks. In *7th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun., Taipei*, pages 1088–1091, Oct 1996.

[30] S. Narahashi and T. Nojima. New phasing scheme of n-multiple carriers for reducing peak-to-average power ratio. *Electron. Lett.*, **30**:1382–1383, 1994.

[31] M. Nazarathy, S.A. Newton, R.P. Giffard, D.S. Moberly, F. Sischka, W.R. Trutna, Jr., and S. Foster. Real-time long range complementary correlation optical time domain reflectometer. *IEEE J. Lightwave Technology*, **7**:24–38, 1989.

[32] R.D.J. van Nee. OFDM codes for peak-to-average power reduction and error correction. In *IEEE Globecom 1996, London*, pages 740–744, Nov 1996.

[33] H. Ochiai and H. Imai. Block coding scheme based on complementary sequences for multicarrier signals. *IEICE Trans. Fundamentals*, Nov 1997. To appear.

[34] K.G. Paterson. Generalised Reed-Muller codes and power control in OFDM modulation. Preprint, Hewlett-Packard Laboratories, Bristol, 1997.

[35] W.W. Peterson and E.J. Weldon, Jr. *Error-Correcting Codes.* MIT Press, Cambridge, 2nd edition, 1972.

[36] B.M. Popović. Synthesis of power efficient multitone signals with flat amplitude spectrum. *IEEE Trans. Commun.*, **39**:1031–1033, 1991.

[37] R.A. Rueppel. Stream ciphers. In G.J. Simmons, editor, *Contemporary Cryptology: the Science of Information Integrity*, pages 65–134. IEEE Press, Piscataway, 1992.

[38] H. Sari, G. Karam, and I. Jeanclaude. Transmission techniques for digital terrestrial TV broadcasting. *IEEE Commun. Magazine*, **33**:100–109, Feb 1995.

[39] P. Shelswell. The COFDM modulation system: the heart of digital audio broadcasting. *Elec. Commun. Eng. J.*, pages 127–136, June 1995.

[40] S.J. Shepherd, P.W.J. van Eetvelt, C.W. Wyatt-Millington, and S.K. Barton. Simple coding scheme to reduce peak factor in QPSK multicarrier modulation. *Electron. Lett.*, **31**:1131–1132, 1995.

[41] R. Sivaswamy. Multiphase complementary codes. *IEEE Trans. Inform. Theory*, **IT-24**:546–552, 1978.

[42] C. Tellambura. Upper bound on peak factor of N-multiple carriers. *Electron. Lett.*, **33**:1608–1609, 1997.

[43] C.-C. Tseng. Signal multiplexing in surface-wave delay lines using orthogonal pairs of Golay's complementary sequences. *IEEE Trans. Sonics Ultrasonics*, **SU-18**:103–107, 1971.

[44] R.J. Turyn. Hadamard matrices, Baumert-Hall units, four-symbol sequences, pulse compression, and surface wave encodings. *J. Combin. Theory (A)*, **16**:313–333, 1974.

[45] T.A. Wilkinson and A.E. Jones. Minimisation of the peak to mean envelope power ratio of multicarrier transmission schemes by block coding. In *IEEE 45th Vehicular Technology Conf., Chicago*, pages 825–829, July 1995.

[46] D. Wulich. Reduction of peak to mean ratio of multicarrier modulation using cyclic coding. *Electron. Lett.*, **32**:432–433, 1996.

u_{12}	u_{13}	u_{14}	u_{23}	u_{24}	u_{34}	Coset representative	Max PEP
1	0	0	1	0	1	(0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 1)	31.59
1	1	0	0	0	1	(0 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1)	31.94
0	1	0	1	1	0	(0 0 0 0 0 1 1 0 0 0 1 1 0 1 0 1)	31.95
1	0	0	0	1	1	(0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 1)	31.98
0	1	0	0	1	1	(0 0 0 1 0 1 0 0 0 0 1 0 0 1 1 1)	31.98
0	0	1	1	0	1	(0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 1)	31.98
0	1	1	0	1	0	(0 0 0 0 0 1 0 1 0 1 1 0 0 0 1 1)	31.98
1	0	1	0	0	1	(0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 1)	31.98
1	1	0	0	1	0	(0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1)	31.99
1	0	1	1	0	0	(0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1)	31.99
0	0	1	1	1	0	(0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 1)	32.00
0	1	1	1	0	0	(0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1)	32.00
1	0	0	0	0	1	(0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 0)	49.82
0	0	1	1	0	0	(0 0 0 0 0 0 1 1 0 1 0 1 0 1 1 0)	49.87
0	1	0	0	1	0	(0 0 0 0 0 1 0 1 0 0 1 1 0 1 1 0)	49.98
0	1	1	0	1	1	(0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0)	50.88
1	0	1	0	1	1	(0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 0)	51.10
1	1	0	1	1	0	(0 0 0 0 0 1 1 0 0 0 1 1 1 0 1 0)	51.12
1	0	0	1	1	1	(0 0 0 1 0 1 1 1 0 0 0 1 1 0 0 0)	51.65
0	0	1	1	1	1	(0 0 0 1 0 1 1 1 0 1 0 0 0 0 1 0)	51.76
1	0	1	1	1	0	(0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 0)	51.81
1	1	1	0	1	0	(0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0)	52.87
1	1	1	1	0	0	(0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 0)	52.90
0	1	1	1	0	1	(0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0)	53.47
1	1	1	0	0	1	(0 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0)	53.56
0	1	0	1	1	1	(0 0 0 1 0 1 1 1 0 0 1 0 0 1 0 0)	53.82
1	1	0	1	0	1	(0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 0)	53.99
0	0	0	0	1	1	(0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0)	64.00
0	0	0	1	0	1	(0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0)	64.00
0	0	0	1	1	0	(0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0)	64.00
0	0	1	0	0	1	(0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0)	64.00
0	0	1	0	1	0	(0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0)	64.00
0	0	1	0	1	1	(0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1)	64.00
0	1	0	0	0	1	(0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0)	64.00
0	1	0	1	0	0	(0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0)	64.00
0	1	0	1	0	1	(0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1)	64.00
0	1	1	0	0	0	(0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0)	64.00
0	1	1	1	1	0	(0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0)	64.00
0	1	1	1	1	1	(0 0 0 1 0 1 1 1 0 1 1 1 0 0 0 1)	64.00
1	0	0	0	1	0	(0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0)	64.00
1	0	0	1	0	0	(0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0)	64.00
1	0	0	1	1	0	(0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1)	64.00
1	0	1	0	0	0	(0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0)	64.00
1	0	1	1	0	1	(0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0)	64.00
1	0	1	1	1	1	(0 0 0 1 0 1 1 1 0 1 0 0 1 1 0 1)	64.00
1	1	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0)	64.00
1	1	0	0	1	1	(0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0)	64.00
1	1	0	1	1	1	(0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 1)	64.00
1	1	1	0	0	0	(0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1)	64.00
1	1	1	0	1	1	(0 0 0 1 0 1 0 0 0 1 1 1 1 1 0 1)	64.00
1	1	1	1	0	1	(0 0 0 1 0 0 1 0 0 1 1 1 1 0 1 1)	64.00
1	1	1	1	1	0	(0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1)	64.00
1	1	1	1	1	1	(0 0 0 1 0 1 1 1 0 1 1 1 1 1 1 0)	98.95
0	1	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1)	99.72
0	1	1	0	0	1	(0 0 0 1 0 0 0 1 0 1 1 1 0 1 1 1)	101.43
1	1	0	1	0	0	(0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 1)	101.56
0	0	0	1	0	0	(0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1)	105.60
1	0	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1)	105.85
0	0	0	1	1	1	(0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1)	106.22
0	0	0	0	1	0	(0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1)	106.41
1	0	1	0	1	0	(0 0 0 0 0 1 0 1 0 1 0 1 1 1 1 1)	106.69
0	0	0	0	0	1	(0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1)	109.48
0	0	1	0	0	0	(0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1)	109.75
0	0	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)	256.00

Table 1: Binary coding with 16 carriers: the 64 cosets of $\mathrm{RM}_2(1,4)$ in $\mathrm{RM}_2(2,4)$, ordered by maximum PEP over the coset. Coset representatives are $\sum_{i<j} u_{ij}x_ix_j$ and the PEP of each sequence is calculated using 256 times oversampling.

u_{12}	u_{13}	u_{14}	u_{23}	u_{24}	u_{34}	Coset representative	Max PEP
0	0	1	1	0	1	(0 0 0 2 0 0 2 0 0 2 0 0 0 2 2 2)	32.00
0	1	0	0	1	1	(0 0 0 2 0 2 0 0 0 0 2 0 0 2 2 2)	32.00
0	1	0	1	1	0	(0 0 0 0 0 2 2 0 0 0 2 2 0 2 0 2)	32.00
0	1	1	1	0	0	(0 0 0 0 0 0 2 2 0 2 2 0 0 2 0 2)	32.00
1	0	0	0	1	1	(0 0 0 2 0 2 0 0 0 0 0 2 2 0 2 2)	32.00
1	0	0	1	0	1	(0 0 0 2 0 0 2 0 0 0 0 2 2 2 0 2)	32.00
1	0	1	0	0	1	(0 0 0 2 0 0 0 2 0 2 0 0 2 0 2 2)	32.00
1	0	1	1	0	0	(0 0 0 0 0 0 2 2 0 2 0 2 2 0 0 2)	32.00
1	1	0	0	0	1	(0 0 0 2 0 0 0 2 0 0 2 0 2 2 0 2)	32.00
1	1	0	0	1	0	(0 0 0 0 0 2 0 2 0 0 2 2 2 0 0 2)	32.00
0	0	1	1	1	0	(0 0 0 0 0 2 2 0 0 2 0 2 0 0 2 2)	32.00
0	1	1	0	1	0	(0 0 0 0 0 2 0 2 0 2 2 0 0 0 2 2)	32.00
0	0	0	1	0	1	(0 0 0 2 0 0 2 0 0 0 0 2 0 0 2 0)	64.00
0	0	0	1	1	0	(0 0 0 0 0 2 2 0 0 0 0 0 0 2 2 0)	64.00
0	0	1	0	1	1	(0 0 0 2 0 2 0 0 0 2 0 0 0 0 0 2)	64.00
0	0	1	1	0	0	(0 0 0 0 0 0 2 2 0 2 0 2 0 2 2 0)	64.00
0	0	1	1	1	1	(0 0 0 2 0 2 2 2 0 2 0 0 0 0 2 0)	64.00
0	1	0	0	0	1	(0 0 0 2 0 0 0 2 0 0 2 0 0 0 2 0)	64.00
0	1	0	0	1	0	(0 0 0 0 0 2 0 2 0 0 2 2 0 2 2 0)	64.00
0	1	0	1	0	1	(0 0 0 2 0 0 2 0 0 0 2 0 0 0 0 2)	64.00
0	1	0	1	1	1	(0 0 0 2 0 2 2 2 0 0 2 0 0 2 0 0)	64.00
0	1	1	0	0	0	(0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 0)	64.00
0	1	1	0	1	1	(0 0 0 2 0 2 0 0 0 2 2 2 0 0 2 0)	64.00
0	1	1	1	0	1	(0 0 0 2 0 0 2 0 0 2 2 2 0 2 0 0)	64.00
0	1	1	1	1	1	(0 0 0 2 0 2 2 2 0 2 2 2 0 0 0 2)	64.00
1	0	0	0	0	1	(0 0 0 2 0 0 0 2 0 0 0 2 2 2 2 0)	64.00
1	0	0	0	1	0	(0 0 0 0 0 2 0 2 0 0 0 0 2 0 2 0)	64.00
1	0	0	1	0	0	(0 0 0 0 0 0 2 2 0 0 0 0 2 2 0 0)	64.00
1	0	0	1	1	0	(0 0 0 0 0 2 2 0 0 0 0 0 2 0 0 2)	64.00
1	0	0	1	1	1	(0 0 0 2 0 2 2 2 0 0 0 2 2 0 0 0)	64.00
1	0	1	0	0	0	(0 0 0 0 0 0 0 0 0 2 0 2 2 0 2 0)	64.00
1	0	1	0	1	1	(0 0 0 2 0 2 0 0 0 2 0 0 2 2 2 0)	64.00
1	0	1	1	1	0	(0 0 0 0 0 2 2 0 0 2 0 2 2 2 0 0)	64.00
1	0	1	1	1	1	(0 0 0 2 0 2 2 2 0 2 0 0 2 2 0 2)	64.00
1	1	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 2 2 2 2 0 0)	64.00
1	1	0	1	0	1	(0 0 0 2 0 0 2 0 0 0 2 0 2 2 2 0)	64.00
1	1	0	1	1	0	(0 0 0 0 0 2 2 0 0 0 2 2 2 0 2 0)	64.00
1	1	1	0	0	0	(0 0 0 0 0 0 0 0 0 2 2 0 2 0 0 2)	64.00
1	1	1	0	0	1	(0 0 0 2 0 0 0 2 0 2 2 2 2 0 0 0)	64.00
1	1	1	0	1	0	(0 0 0 0 0 2 0 2 0 2 2 0 2 2 0 0)	64.00
1	1	1	0	1	1	(0 0 0 2 0 2 0 0 0 2 2 2 2 2 0 2)	64.00
1	1	1	1	0	0	(0 0 0 0 0 0 2 2 0 2 2 0 2 0 2 0)	64.00
0	0	0	0	1	1	(0 0 0 2 0 2 0 0 0 0 0 2 0 2 0 0)	64.00
0	0	1	0	0	1	(0 0 0 2 0 0 0 2 0 2 0 0 0 2 0 0)	64.00
0	0	1	0	1	0	(0 0 0 0 0 2 0 2 0 2 0 2 0 0 0 0)	64.00
0	1	0	1	0	0	(0 0 0 0 0 0 2 2 0 0 2 2 0 0 0 0)	64.00
0	1	1	1	1	0	(0 0 0 0 0 2 2 0 0 2 2 0 0 0 0 0)	64.00
1	0	1	1	0	1	(0 0 0 2 0 0 2 0 0 2 0 0 2 0 0 0)	64.00
1	1	0	0	1	1	(0 0 0 2 0 2 0 0 0 0 2 0 2 0 0 0)	64.00
1	1	0	1	1	1	(0 0 0 2 0 2 2 2 0 0 2 0 2 0 2 2)	64.00
1	1	1	1	0	1	(0 0 0 2 0 0 2 0 0 2 2 2 2 0 2 2)	64.00
1	1	1	1	1	0	(0 0 0 0 0 2 2 0 0 2 2 0 2 2 2 2)	64.00
1	1	0	1	0	0	(0 0 0 0 0 0 2 2 0 0 2 2 2 2 2 2)	128.00
0	0	0	0	0	1	(0 0 0 2 0 0 0 2 0 0 0 2 0 0 0 2)	128.00
0	0	0	0	1	0	(0 0 0 0 0 2 0 2 0 0 0 0 0 2 0 2)	128.00
0	0	0	1	0	0	(0 0 0 0 0 0 2 2 0 0 0 0 0 0 2 2)	128.00
0	0	0	1	1	1	(0 0 0 2 0 2 2 2 0 0 0 2 0 2 2 2)	128.00
0	1	1	0	0	1	(0 0 0 2 0 0 0 2 0 2 2 2 0 2 2 2)	128.00
1	0	1	0	1	0	(0 0 0 0 0 2 0 2 0 2 0 2 2 2 2 2)	128.00
0	0	1	0	0	0	(0 0 0 0 0 0 0 0 0 2 0 2 0 2 0 2)	128.00
0	1	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2)	128.00
1	0	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2)	128.00
1	1	1	1	1	1	(0 0 0 2 0 2 2 2 0 2 2 2 2 2 2 0)	128.00
0	0	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)	256.00

Table 2: Quaternary coding with 16 carriers: the 64 cosets of $\text{RM}_4(1, 4)$ in $\text{ZRM}_4(2, 4)$, ordered by maximum PEP over the coset. Coset representatives are $2\sum_{i<j} u_{ij}x_ix_j$ and the PEP of each sequence is calculated using 1 times oversampling.

u_{12}	u_{13}	u_{14}	u_{23}	u_{24}	u_{34}	Coset representative	Max PEP
0	0	2	2	0	2	(0 0 0 4 0 0 4 0 0 4 0 0 0 4 4 4)	32.00
2	0	0	0	2	2	(0 0 0 4 0 4 0 0 0 0 0 4 4 0 4 4)	32.00
2	0	0	2	0	2	(0 0 0 4 0 0 4 0 0 0 0 4 4 4 0 4)	32.00
2	0	2	0	0	2	(0 0 0 4 0 0 0 4 0 4 0 0 4 0 4 4)	32.00
2	0	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 0 4 4 0 0 4)	32.00
2	2	0	0	0	2	(0 0 0 4 0 0 0 4 0 0 4 0 4 4 0 4)	32.00
2	2	0	0	2	0	(0 0 0 0 0 4 0 4 0 0 4 4 4 0 0 4)	32.00
0	0	2	2	2	0	(0 0 0 0 0 4 4 0 0 4 0 4 0 0 4 4)	32.00
0	2	0	2	2	0	(0 0 0 0 0 4 4 0 0 0 4 4 0 4 0 4)	32.00
0	2	0	0	2	2	(0 0 0 4 0 4 0 0 0 0 4 0 0 4 4 4)	32.00
0	2	2	0	2	0	(0 0 0 0 0 4 0 4 0 4 4 0 0 0 4 4)	32.00
0	2	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 4 0 0 4 0 4)	32.00
1	0	2	2	1	0	(0 0 0 0 0 2 4 6 0 4 0 4 2 0 6 4)	48.00
1	2	1	0	2	0	(0 0 0 0 0 4 0 4 0 2 4 6 2 0 6 4)	48.00
0	0	2	2	1	1	(0 0 0 2 0 2 4 0 0 4 0 6 0 6 4 4)	48.00
0	1	2	2	0	1	(0 0 0 2 0 0 4 6 0 4 2 0 0 4 6 4)	48.00
1	2	0	1	2	0	(0 0 0 0 0 4 2 6 0 0 4 4 2 6 0 4)	48.00
0	2	0	1	2	1	(0 0 0 2 0 4 2 0 0 0 4 6 0 4 6 4)	48.00
0	2	1	0	2	1	(0 0 0 2 0 4 0 6 0 2 4 0 0 6 4 4)	48.00
2	0	0	1	1	2	(0 0 0 4 0 2 2 0 0 0 0 4 4 6 6 4)	48.00
2	0	1	0	1	2	(0 0 0 4 0 2 0 6 0 2 0 6 4 0 4 4)	48.00
2	1	0	1	0	2	(0 0 0 4 0 0 2 6 0 0 2 6 4 4 0 4)	48.00
1	1	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 2 6 2 6 0 4)	48.00
2	1	1	0	0	2	(0 0 0 4 0 0 0 4 0 2 2 0 4 6 6 4)	48.00
0	2	3	0	2	3	(0 0 0 6 0 4 0 2 0 6 4 0 0 2 4 4)	48.00
2	0	0	3	3	2	(0 0 0 4 0 6 6 0 0 0 0 4 4 2 2 4)	48.00
2	0	3	0	3	2	(0 0 0 4 0 6 0 2 0 6 0 2 4 0 4 4)	48.00
2	3	0	3	0	2	(0 0 0 4 0 0 6 2 0 0 6 2 4 4 0 4)	48.00
2	3	3	0	0	2	(0 0 0 4 0 0 0 4 0 6 6 0 4 2 2 4)	48.00
3	2	0	3	2	0	(0 0 0 0 0 4 6 2 0 0 4 4 6 2 0 4)	48.00
3	2	3	0	2	0	(0 0 0 0 0 4 0 4 0 6 4 2 6 0 2 4)	48.00
3	3	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 6 2 6 2 0 4)	48.00
0	0	2	2	1	3	(0 0 0 6 0 2 4 4 0 4 0 2 0 6 4 0)	48.00
0	0	2	2	3	1	(0 0 0 2 0 6 4 4 0 4 0 6 0 2 4 0)	48.00
0	0	2	2	3	3	(0 0 0 6 0 6 4 0 0 4 0 2 0 2 4 4)	48.00
0	2	0	3	2	3	(0 0 0 6 0 4 6 0 0 0 4 2 0 4 2 4)	48.00
0	2	1	0	2	3	(0 0 0 6 0 4 0 2 0 2 4 4 0 6 4 0)	48.00
0	3	2	2	0	3	(0 0 0 6 0 0 4 2 0 4 6 0 0 4 2 4)	48.00
1	3	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 6 2 2 6 4 0)	48.00
3	0	2	2	3	0	(0 0 0 0 0 6 4 2 0 4 0 4 6 0 2 4)	48.00
0	2	0	3	2	1	(0 0 0 2 0 4 6 4 0 0 4 6 0 4 2 0)	48.00
0	1	2	2	0	3	(0 0 0 6 0 0 4 2 0 4 2 4 0 4 6 0)	48.00
2	0	0	1	3	2	(0 0 0 4 0 6 2 4 0 0 0 4 4 2 6 0)	48.00
2	0	1	0	3	2	(0 0 0 4 0 6 0 2 0 2 0 6 4 4 4 0)	48.00
2	1	0	3	0	2	(0 0 0 4 0 0 6 2 0 0 2 6 4 4 4 0)	48.00
1	0	2	2	3	0	(0 0 0 0 0 6 4 2 0 4 0 4 2 4 6 0)	48.00
1	2	3	0	2	0	(0 0 0 0 0 4 0 4 0 6 4 2 2 4 6 0)	48.00
0	2	0	1	2	3	(0 0 0 6 0 4 2 4 0 0 4 2 0 4 6 0)	48.00
0	2	3	0	2	1	(0 0 0 2 0 4 0 6 0 6 4 4 0 2 4 0)	48.00
0	3	2	2	0	1	(0 0 0 2 0 0 4 6 0 4 6 4 0 4 2 0)	48.00
2	0	0	3	1	2	(0 0 0 4 0 2 6 4 0 0 0 4 4 6 2 0)	48.00
2	0	3	0	1	2	(0 0 0 4 0 2 0 6 0 6 0 2 4 4 4 0)	48.00
2	1	3	0	0	2	(0 0 0 4 0 0 0 4 0 6 2 4 4 2 6 0)	48.00
2	3	1	0	0	2	(0 0 0 4 0 0 0 4 0 2 6 4 4 6 2 0)	48.00
1	2	0	3	2	0	(0 0 0 0 0 4 6 2 0 0 4 4 2 6 4 0)	48.00
3	0	2	2	1	0	(0 0 0 0 0 2 4 6 0 4 0 4 6 4 2 0)	48.00
3	1	2	2	0	0	(0 0 0 0 0 0 4 4 0 4 2 6 6 2 4 0)	48.00
2	3	0	1	0	2	(0 0 0 4 0 0 2 6 0 0 6 2 4 4 4 0)	48.00
3	2	0	1	2	0	(0 0 0 0 0 4 2 6 0 0 4 4 6 2 4 0)	48.00
3	2	1	0	2	0	(0 0 0 0 0 4 0 4 0 2 4 6 6 4 2 0)	48.00
0	0	2	2	0	1	(0 0 0 2 0 0 4 6 0 4 0 6 0 4 4 2)	54.63
0	2	0	0	2	1	(0 0 0 2 0 4 0 6 0 0 4 6 0 4 4 2)	54.63

⋮

[4032 lines of table omitted]

⋮

0	0	0	0	0	3	(0 0 0 6 0 0 0 6 0 0 0 6 0 0 0 6)	218.51
0	0	0	0	0	0	(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)	256.00

Table 3: Octary coding with 16 carriers: the 4096 cosets of $\mathrm{RM}_8(1,4)$ in $\mathrm{ZRM}_8(2,4)$, ordered by maximum PEP over the coset. Coset representatives are $2\sum_{i<j} u_{ij}x_ix_j$ and the PEP of each sequence is calculated using 256 times oversampling.

	# carriers	Max possible PMEPR (dB)	Max actual PMEPR (dB)	Min Hamming distance	# info bits per codeword	Code rate	Info rate
1	16	12.0	3.0	8	5	0.31	0.31
2			3.0	6	7	0.44	0.44
3			3.0	4	8	0.50	0.50
4			6.0	4	10	0.62	0.62
5			6.0	4;4	8	0.50	0.50
6	32	15.1	3.0	16	6	0.19	0.19
7			3.0	8	11	0.34	0.34
8			6.0	8;8	10	0.31	0.31
9			6.0	6;6	15	0.47	0.47
10			6.0	4;4	17	0.53	0.53

Table 4: Binary coding options with 16 and 32 carriers. $d;d$ describes minimum distance in a composition coding scheme.

	# carriers	Max possible PMEPR (dB)	Max actual PMEPR (dB)	Min Hamming distance	Min Lee distance	# info bits per codeword	Code rate	Info rate
1	16	12.0	3.0	8	8	10	0.31	0.62
2			3.0	6	8	12	0.38	0.75
3			3.0	4	8	13	0.41	0.81
4			6.0	4	8	15	0.47	0.94
5			6.0	4;4	4;4	16	0.50	1.00
5a			6.0	2;2	4;4	19	0.59	1.19
6	32	15.1	3.0	16	16	12	0.19	0.38
7			3.0	8	16	17	0.27	0.53
8			6.0	8;8	8;8	20	0.31	0.62
9			6.0	6;6	8;8	25	0.39	0.78
10			6.0	4;4	8;8	27	0.42	0.84

Table 5: Quaternary coding options with 16 and 32 carriers.

	# carriers	Max possible PMEPR (dB)	Max actual PMEPR (dB)	Min Hamming distance	Min Lee distance	# info bits per codeword	Code rate	Info rate
1	16	12.0	3.0	8	8	15	0.31	0.94
2			3.0	6	8	17	0.35	1.06
3			3.0	4	8	18	0.38	1.12
4			4.8	4	8	20	0.42	1.25
5			6.0	4;4	4;4	24	0.50	1.50
5a			6.0	2;2	4;4	27	0.56	1.69
6	32	15.1	3.0	16	16	18	0.19	0.56
7			3.0	8	16	23	0.24	0.72
8			6.0	8;8	8;8	30	0.31	0.94
9			6.0	6;6	8;8	35	0.36	1.09
10			6.0	4;4	8;8	37	0.39	1.16

Table 6: Octary coding options with 16 and 32 carriers.