On structural properties of Reed-Muller codes and their use in channels with synchronization and substitution errors

Lara Dolecek

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-43
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-43.html

April 27, 2006

On Structural Properties of Reed-Muller RM(1,m) Codes and Their Use in Channels with Synchronization Errors

Lara Dolecek

dolecek@eecs.berkeley.edu

April 25, 2006

1 Introduction

In this report we address the problem of using a Reed-Muller(1,m) code in channels which, due to imperfect synchronization, permit one repetition and one deletion of a bit per transmitted codeword and we provide new results on the run-length structure of this code. The motivation, while briefly presented here, is fully provided in [1], along with the summary of relevant existing work. This report and its contribution should be viewed in the context of the problem addressed in [1].

Typically, in a communication system, a binary input message $\mathbf{x}$ is encoded at the transmitter using a substitution-error correcting code C into a coded sequence $\mathbf{c}$ = $C(\mathbf{x})$, which we assume is also binary. The received waveform after matched filtering may be written as

$$r(t) = \sum_i c_i h(t - iT) + n(t), \tag{1}$$

where c_i is the i^{th}bit of $\mathbf{c}$, $h(t)$ is convolution of the modulating pulse and the matched filter, and $n(t)$ represents the additive noise introduced by the channel. The receiver first samples the waveform $r(t)$ at specific time instances, followed by decoding of the transmitted message based on these discrete values. For the decoding to work properly, it is crucial that the sampling is done at correct places. However, as the operating conditions under which sampling must be performed become more stringent (such as in modern magnetic recording and wireless applications), it becomes ever more difficult to accurately sample the incoming waveform. As a result, when the adequate synchronization is missing, some symbol may be skipped or sampled twice. Let us suppose that we are operating in the infinite SNR regime, and that transmitted codewords can be analyzed individually, so that, under perfect synchronization, the received sampled sequence would be precisely the transmitted codeword. If the synchronization is not perfect, some bit can then be repeated or deleted. As a result, each codeword gives rise to a whole set of possible binary sequences obtained from it by different repetition-deletion patterns, as dictated by the accuracy of the synchronization scheme. An important consequence is that different codewords can then result in the same string, thus making the correct input message retrieval impossible. We call such codewords *identification problem* causing codewords.

In this report we focus on the Reed-Muller (1, m) code. The goal is to address the scenario when due to imperfect synchronization, both a repetition and a deletion can occur within the transmitted codeword, and to propose a way to eliminate the identification problem causing codewords. The subcase of a single synchronization error is studied in detail in [1]. In addition, we provide run-length properties of this code that may be of independent interest.

The report is outlined as follows. In the next section we briefly review the RM(1, m) code. In Section 3 we present several useful structural properties of this code, and in Section 4 we provide a detailed analysis of the RM(1, m) code where both a repetition and a deletion are possible. The report is concluded with Section 5.

2 Review of the RM(1, m) code

First order Reed-Muller codes (RM(1,m)) are an instance of linear substitution error-correcting codes [2]. This code is described by a $k \times n$ generator matrix $\mathbf{G_m}$ where $k = m + 1$ and $n = 2^m$. Let $\mathbf{g_0}(\mathbf{m})$ denote an all ones vector of length 2^m, and let $\mathbf{g_1}(\mathbf{m})$ be an $m \times 2^m$ matrix whose columns are binary m-tuples in the decreasing order.

The generator matrix of the RM(1,m) code is then

$$\mathbf{G_m} = \begin{bmatrix} \mathbf{g_0(m)} \\ \mathbf{g_1(m)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \dots & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & \dots & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{2}$$

Observe that the first row of $\mathbf{G_m}$ consists of all ones, and the i^{th} row of $\mathbf{G_m}$, for $1 < i \leq m+1$ consists of 2^{i-1} alternating runs of ones and zeros, where each run is of size 2^{m-i+1}, and the leftmost run in each row is a run of ones. Let $C(m)$ denote the RM(1,m) code. Note that every codeword in $C(m+1)$ code is either the concatenation of a codeword in $C(m)$ with itself or with its bitwise complement.

3 Runlength properties of the Reed-Muller(1,m) Code

In addition to the properties of the code presented in [1], we now prove additional interesting structural results.

3.1 Relationship between the input message and the run-lengths of its codeword

It is sometimes useful to determine the number of runs of a particular codeword based on its input message and vice versa. In this section we provide an explicit relationship between these two quantities. Let $\mathbf{a_m} = (a_0, a_m, a_{m-1}, ..., a_2, a_1)$ be a binary string of length $m+1$ and let $\mathbf{c}$ be a codeword in $C(m)$ such that $\mathbf{c} = \mathbf{a_m}\mathbf{G_m}$. The bit a_0 multiplies the all-ones row of $\mathbf{G_m}$ and therefore does not affect the number of runs of the resulting codeword, i.e. $\mathbf{a_m} = (a_0, a_m, a_{m-1}, ..., a_2, a_1)$ and $\mathbf{a_m} = (\overline{a_0}, a_m, a_{m-1}, ..., a_2, a_1)$ result in complement codewords (with the same number of runs).

In the following we replace a_0 by x to indicate that the value of a_0 does not matter.

We denote by $R_m(a_0, a_1, ..., a_{m-1}, a_m)$ the total number of runs in $\mathbf{c}$. The following result provides a closed-form expression for $R_m(a_0, a_1, ..., a_{m-1}, a_m)$ in terms of $\mathbf{a_m}$.

Lemma 1 *The number of runs in the codeword* $\mathbf{c}$ *given by* $\mathbf{c} = \mathbf{a_m G_m}$ *where* $\mathbf{a_m} = (a_0, a_m, a_{m-1}, ..., a_2, a_1)$ *is* $R_m(a_0, a_1, ..., a_{m-1}, a_m) = 2^{m-1}a_1 + 2^{m-2} + 1/2 - \sum_{k=2}^{m} 2^{m-k-1}(-1)^{\sum_{i=1}^{k} a_i}$.

Proof: By construction the bottom $m-1$ rows in $\mathbf{G_m}$ when viewed as a $m-1$ by 2^m matrix, are the same as the matrix obtained by concatenating the matrix consisting of the bottom $m-1$ rows in $\mathbf{G_{m-1}}$ with itself. If the runs at the point of concatenation are the same, the concatenation results in the merging of two runs, otherwise no runs are altered.

Therefore, the linear combination of the bottom $m-1$ rows in $\mathbf{G_m}$ produces a codeword in $C(m)$ which has either $2R$ or $2R-1$ runs, where R denotes the number of runs of the codeword produced by the same linear combination of rows in $\mathbf{G_{m-1}}$. In particular, the number of runs is $2R$ if the auxiliary codeword in $C(m-1)$ (the one constructed from the same linear combination) had different outermost bits, and the number of runs is $2R-1$ if the outermost bits are the same. The former (latter) case occurs when the linear combination consists of an odd (even) number of participating rows.

Then, when $a_m = 0$ we have the following:

$$R_m(x, a_1, a_2, ..., a_{m-1}, 0) = \begin{cases} 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}), & \text{if } \sum_{i=1}^{m-1} a_i \bmod 2 \equiv 1, \\ 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}) - 1, & \text{if } \sum_{i=1}^{m-1} a_i \bmod 2 \equiv 0 \end{cases}$$

Now, $a_m = 1$ has the effect of complementing the left half of the codeword obtained from a linear combination of rows of $\mathbf{G_m}$ that does not involve second row of $\mathbf{G_m}$, and leaving the right half intact.

Therefore,

$$R_m(x, a_1, a_2, ..., a_{m-1}, 1) = \begin{cases} 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}), & \text{if } \sum_{i=1}^{m-1} a_i \bmod 2 \equiv 0, \\ 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}) - 1, & \text{if } \sum_{i=1}^{m-1} a_i \bmod 2 \equiv 1 \end{cases}$$

We can jointly write these two expressions as

$$R_m(x, a_1, a_2, ..., a_{m-1}, a_m) = 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}) - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2.$$

To obtain the formula for $R_m(x, a_1, ..., a_{m-1}, a_m)$, iterate recursively as follows,

$$\begin{aligned}
&R_m(x, a_1, ..., a_m) \\
&\quad = 2R_{m-1}(x, a_1, a_2, ..., a_{m-1}) - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2 \\
&\quad = 2\left[2R_{m-2}(x, a_1, a_2, ..., a_m) - 1/2(-1)^{\sum_{i=1}^{m-1} a_i} - 1/2\right] - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2 \\
&\quad = 4R_{m-2}(x, a_1, a_2, ..., a_{m-2}) - (-1)^{\sum_{i=1}^m a_i} - 1 - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2 \\
&\quad = 4\left[2R_{m-3}(x, a_1, a_2, ..., a_{m-3}) - 1/2(-1)^{\sum_{i=1}^{m-2} a_i} - 1/2\right] - (-1)^{\sum_{i=1}^m a_i} - 1 - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2 \\
&\quad = 8R_{m-3}(x, a_1, a_2, ..., a_{m-3}) - 2(-1)^{\sum_{i=1}^{m-2} a_i} - 2 - (-1)^{\sum_{i=1}^m a_i} - 1 - 1/2(-1)^{\sum_{i=1}^m a_i} - 1/2 \\
&\quad \vdots \\
&\quad = 2^{m-2}R_2(x, a_1, a_2) - 2^{m-3-1}(-1)^{\sum_{i=1}^3 a_i} - 2^{m-3-1} - 2^{m-4-1}(-1)^{\sum_{i=1}^4 a_i} - 2^{m-4-1} - ... \\
&\qquad -2^{m-(m-1)-1}(-1)^{\sum_{i=1}^{m-1} a_i} - 2^{m-(m-1)-1} - 2^{m-m-1}(-1)^{\sum_{i=1}^m a_i} - 2^{m-m-1} \\
&\quad = 2^{m-1}R_1(x, a_1) - 2^{m-2}1/2(-1)^{\sum_{i=1}^2 a_i} - 2^{m-2}1/2 - 2^{m-4}(-1)^{\sum_{i=1}^3 a_i} - 2^{m-4} - ... \\
&\qquad -2^0(-1)^{\sum_{i=1}^{m-1} a_i} - 2^0 - 2^{-1}(-1)^{\sum_{i=1}^m a_i} - 2^{-1} \\
&\quad = 2^{m-1}(1 + a_1) - \\
&\quad \left[2^{m-3}(-1)^{\sum_{i=1}^2 a_i} + 2^{m-4}(-1)^{\sum_{i=1}^3 a_i} + ... + 2^{-1}(-1)^{\sum i=1^m a_i} + 2^{m-3} + 2^{m-4} + ... + 1 + 1/2\right] \\
&\quad = 2^{m-1}a_1 - \sum_{k=2}^m 2^{m-k-1}(-1)^{\sum_{i=1}^k a_i} + 2^{m-1} - \left[2^{m-2} - 1 + 1/2\right] \\
&\quad = 2^{m-1}a_1 + 2^{m-2} + 1/2 - \sum_{k=2}^m 2^{m-k-1}(-1)^{\sum_{i=1}^k a_i},
\end{aligned}$$

which completes the proof. ■

It is also useful to know how to quickly determine the input message based on the number of runs in the codeword it generates. Let N_m be the integer denoting the number of runs, and let $S_m(N_m) = (a_0, a_1, a_2, ..., a_{m-1}, a_m)$ be the binary string consisting of the entries in the input message, so that the mapping S_m is from $\mathbb{N}^+$ to $\{0, 1\}^{m+1}$.

First observe that $|\sum_{k=2}^m 2^{m-k-1}(-1)^{\sum_{i=2}^k a_i}| \leq 2^{m-2} - 1/2$. Thus, for $a_1 = 1$, $R_m(x, 1, ..., a_{m-1}, a_m)$ is at least $2^{m-1}+1$, and for $a_1 = 0$, $R_m(x, 0, ..., a_{m-1}, a_m)$ is at most 2^{m-1}. Moreover, $R_m(x, 1, ..., a_{m-1}, a_m)$

+ $R_m(x, 0, ..., a_{m-1}, a_m)$ evaluates to $2^m + 1$.

Thus, for the given m, if $N_m \geq 2^{m-1} + 1$, a_1 must be 1, otherwise it must be zero.

We can now subtract the contribution of a_1 to N_m, which is zero for $a_1 = 0$ and is $2^m + 1 - N_m$ for $a_1 = 1$, where by contribution we mean the difference in the number of runs of the codewords whose input messages are $(x, a_1, a_2, ..., a_m)$ and $(x, 0, a_2, ..., a_m)$, respectively. Denote the result by N'_m.

Having subtracted the contribution of a_1 from N_m, to determine a_2, observe that $R_m(x, 0, a_2, ..., a_{m-1}, a_m)$ = $R_{m-1}(x, a_2, ..., a_{m-1}, a_m)$, since the i^{th} row of $\mathbf{G_m}$ for $1 \leq i \leq m$ is constructed from the $(i+1)^{st}$ row of $\mathbf{G_{m-1}}$ by duplicating each entry twice. Thus, a codeword constructed from the linear combination of a subset of these particular rows of $\mathbf{G_m}$ has the same number of runs as the codeword in $C(m-1)$ constructed from the counterpart rows of $\mathbf{G_{m-1}}$.

We now view a_2 as the value that multiplies the last row of $\mathbf{G_{m-1}}$, just like a_1 did for $\mathbf{G_m}$. By using the same line of arguments as for a_1, we conclude that if $N'_m \geq 2^{(m-1)-1} + 1$, a_2 is 1, otherwise it is 0. To determine a_3 we need to subtract the contribution of a_2 from N'_m, and compare the result to $2^{(m-1)-1} + 1$, and so on.

The steps for determining a_1 through a_m can be outlined as follows:

Algorithm 1

1. Initialization: $N_c = N_m$, $l = 0$, $a_1 = a_2 = ... = a_m = 0$
2. Find the largest integer p, $p \geq 0$ such that $N_c > 2^p$. If no such p exists, stop and return the current values of a_1 through a_m. If such p exists, proceed to Step 3.
3. Let $i = m - p$, and update the value a_i to be 1 (a_j's for $l < j < i$ remain 0). Proceed to Step 4.
4. Let $N_c := 2^{p+1} + 1 - N_c$ and let $l = i$. Return to Step 2.

Example: $m = 4$, $N_m = 10$.

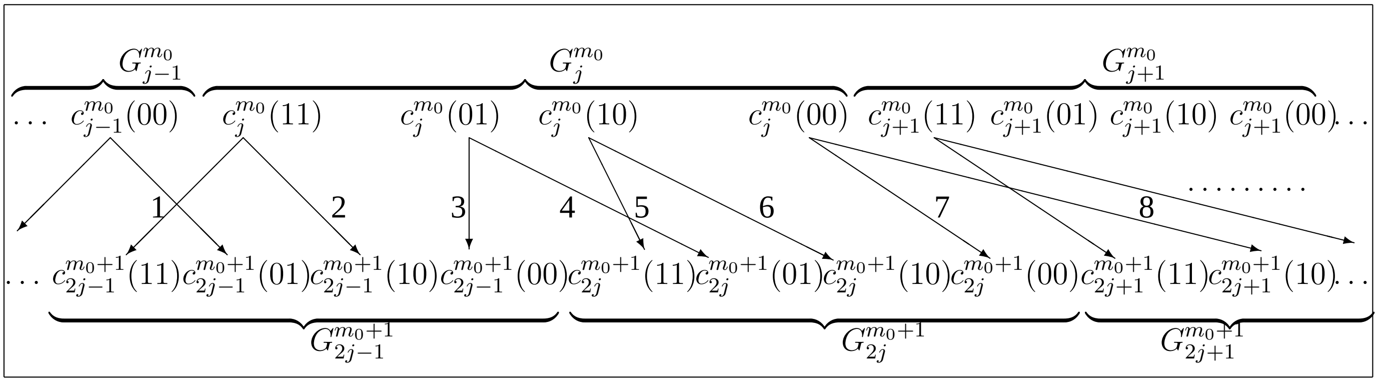

Figure 1: Construction of codewords in $C(m_0+1)$ from codewords in $C(m_0)$.

- Step 1: Initialize $(a_1, a_2, a_3, a_4) = (0, 0, 0, 0)$, $N_c = 10$, $l = 0$
- Step 2.a: Since $8 < N_c < 16 \Rightarrow p = 3$
- Step 3.a: Set $i = 1$, $a_1 = 1$
- Step 4.a: Set $N_c = 7$, $l = 1$
- Step 2.b: Since $4 < N_c < 8 \Rightarrow p = 2$
- Step 3.b: Set $i = 2$, $a_2 = 1$
- Step 4.b: Set $N_c = 2$, $l = 2$
- Step 2.c: Since $1 < N_c \leq 2 \Rightarrow p = 0$
- Step 3.c: Set $i = 4$, $a_4 = 1$
- Step 4.c: Set $N_c = 1$, $l = 4$
- Step 2.d. No p exists, return $(a_1, a_2, a_3, a_4) = (1, 1, 0, 1)$

It can be easily checked that the messages $[0, 1, 0, 1, 1]$ and $[1, 1, 0, 1, 1]$ both result in codewords with 10 runs each.

3.2 Run-length distribution

Lemma 2 *The codewords in* $C(m)$ *can be partitioned into* $2^{m-1}+1$ *distinct non-empty groups* G_j^m*, for* $0 \leq j \leq 2^{m-1}$*. Here* G_j^m *is comprised of those codewords in* $C(m)$ *that have* j *runs of ones.* G_0^m *is comprised of exactly one codeword, namely the all-zero codeword. This codeword will be denoted* $c_0^m(00)$*. There are 4 distinct codewords in each group* G_j^m*, for* $1 \leq j < 2^{m-1}$*. These codewords may be uniquely identified by their first and last bit. They may thus be unambiguously denoted as* $c_j^m(11)$*,* $c_j^m(10)$*,* $c_j^m(01)$*, and* $c_j^m(00)$ *respectively. There are 3 distinct codewords in the group* $G_{2^{m-1}}^m$*. These codewords may also be uniquely identified by their first and last bit and may be unambiguously denoted as* $c_{2^{m-1}}^m(11)$*,* $c_{2^{m-1}}^m(10)$*, and* $c_{2^{m-1}}^m(01)$ *respectively.*

Proof: See Lemma 1 in [1]. ■

Lemma 3 *Consider a codeword* c *in* $C(m)$*. Either* c *has all its runs of the same length, which is a power of* 2*, or the runs in* c *are of at most two different lengths, and these two lengths are consecutive powers of 2. In addition, if there are runs of two different lengths in* c*, the outer runs (i.e. the leftmost run and the rightmost run) in* c *are of the smaller length.*

Proof: See Lemma 3 in [1]. ■

Using the notation introduced in the preceding Lemmas, we can prove the following result on the run-length distribution of $C(m)$.

Lemma 4 *With the exception of the all-ones codeword, all codewords belonging to the group* G_j^m *for* $2^{p-1} < j \leq 2^p$ *for some* p*,* $0 \leq p \leq m-1$ *have all runs of ones either of length* 2^{m-p-1} *or of length* 2^{m-p}*. Moreover,* $(j-2^{p-1}) \times 2$ *runs out of these* j *runs have length* 2^{m-p-1}*, and the remaining* $2^p - j$ *runs have length* 2^{m-p}*.*

Proof: To prove the statement we use induction on m. For small values of m, the proposed statement can be verified directly. Suppose now that the assertion holds for some $m = m_0$.

By Lemma 2, the group $G_{j'}^{m_0}$ for $2^{p-1} < j' \leq 2^p$ for some p, $0 \leq p \leq m_0 - 1$ contains codewords $c_{j'}^{m_0}(10)$, $c_{j'}^{m_0}(01)$, and $c_{j'}^{m_0}(11)$. If $j' \neq 2^{m_0-1}$ it also contains $c_{j'}^{m_0}(00)$. There is a single codeword in $G_0^{m_0}$ (the all-zeros codeword). Let us now analyze all the possible concatenations of the codewords belonging to the group $G_j^{m_0}, 0 \leq j \leq 2^{m_0-1}$ i.e. of each codeword with itself and with its complement. By Lemma 2 there are at most 4 codewords in $G_j^{m_0}$ so we have to consider at most 8 different concatenations. In doing so, the similar cases will be presented together.

- The concatenation of $c_j^{m_0}(11)$, if it exists, with itself produces a codeword in $G_{2j-1}^{m_0+1}$ (see Arrow 1 in Figure 1).

  If $j = 1$, $c_j^{m_0}(11)$ is the all-ones codeword in $C(m_0)$, and the concatenation with itself produces the all-ones codeword in $C(m_0+1)$. If $j > 1$, the outer runs in $c_j^{m_0}(11)$ must be of size 2^{m_0-p-1}. (To see this not that if j is a complete power of 2, i.e. $j = 2^p$ then all runs of ones, including the outer runs, are of size 2^{m_0-p-1} by assumption, and if j is not a complete power of 2, i.e. $2^{p-1} < j < 2^p$ then the outer runs must have size 2^{m_0-p-1} by Lemma 3). In the process of concatenation, two outer, smaller runs merge into one larger run and all other runs of ones are unaltered. Therefore, in the resulting codeword in $G_{2j-1}^{m_0+1}$, where $j > 1$, and $2^p < 2j-1 < 2^{p+1}$, there are $2 \times (j - 2^{p-1}) \times 2 - 2 = ((2j-1) - 2^p) \times 2$ runs of ones of size $2^{m_0-p-1} = 2^{(m_0+1)-(p+1)-1}$, and $2 \times (2^p - j) + 1 = 2^{p+1} - (2j-1)$ runs of ones of size $2^{m_0-p} = 2^{(m_0+1)-(p+1)}$.

- The concatenation of $c_j^{m_0}(11)$, if it exists, with its complement produces a codeword in $G_{2j-1}^{m_0+1}$ (see Arrow 2 in Figure 1).

  The complement of $c_j^{m_0}(11)$ is $c_{j-1}^{m_0}(00)$. By assumption, $c_j^{m_0}(11)$ has $(j - 2^{p-1}) \times 2$ runs of ones of size 2^{m_0-p-1}, and $2^p - j$ runs of ones of size 2^{m_0-p}, for $j > 1$. If $j = 1$, then $p = 0$, and the complement is the all-zero codeword, so the result of the concatenation has a single run of ones, of size $2^{m_0} = 2^{(m_0+1)-1}$.

  Suppose now that $j > 1$. Then there is a corresponding p such that $2^{p-1} < j \leq 2^p$ and $0 < p \leq m_0 - 1$. Note that $2^{p-1} \leq j - 1 < 2^p$.

  Case 1: $j - 1 = 2^{p-1}$

  Under this condition, the codeword $c_{j-1}^{m_0}(00)$ has all $j-1$ runs of ones of size $2^{m_0-(p-1)-1}$ each. The concatenation of $c_j^{m_0}(11)$ and $c_{j-1}^{m_0}(00)$ then has $(j - 2^{p-1}) \times 2 = 2$ runs of ones of size 2^{m_0-p-1}, and $2^p - j + j - 1 = 2^p - 1$ runs of ones of size 2^{m_0-p}. Using the fact that $2 = ((2j-1) - 2^p) \times 2$, $2^p - 1 = 2^{p+1} - (2j-1)$ and that $2^p < 2j-1 < 2^{p+1}$, we conclude that the resulting codeword satisfies the proposed assertion.

Case 2: $j-1>2^{p-1}$

The codeword $c_{j-1}^{m_0}(00)$ has $((j-1)-2^{p-1})\times 2$ runs of ones of size 2^{m_0-p-1} and $2^p-(j-1)$ runs of ones of size 2^{m_0-p}. The result of the concatenation has $(j-2^{p-1})\times 2+((j-1)-2^{p-1})\times 2=((2j-1)-2^p)\times 2$ runs of ones of size 2^{m_0-p-1}, and $2^p-j+2^p-(j-1)=2^{p+1}-(2j-1)$ runs of ones of size 2^{m_0-p}. Since $2^p<2j-1<2^{p+1}$, the proposed assertion holds for this choice of $j-1$ as well.

- The concatenation of $c_j^{m_0}(01)$, if it exists, with its complement produces a codeword in $G_{2j-1}^{m_0+1}$ (see Arrow 3 in Figure 1).

  First note that the complement of $c_j^{m_0}(01)$ is $c_j^{m_0}(10)$, and since they both belong to the same group $G_j^{m_0}$, by assumption they both have $(j-2^{p-1})\times 2$ runs of ones of size 2^{m_0-p-1}, and 2^p-j runs of ones of size 2^{m_0-p}.

  As established in Lemma 3, the outer runs are of the smaller size (here 2^{m_0-p-1}), so in the process of concatenating $c_j^{m_0}(01)$ and $c_j^{m_0}(10)$, the rightmost run of ones in $c_j^{m_0}(01)$ merges with the leftmost run of ones in $c_j^{m_0}(10)$, resulting in a run of ones of size 2^{m_0-p}. All other runs of ones are unaltered. We will treat the cases $j=1$ and $j>1$ separately.

  If $j=1$, both $c_j^{m_0}(01)$ and $c_j^{m_0}(10)$ have one run of ones of size 2^{m_0-1}, so their concatenation results in a codeword in $G_1^{m_0+1}$ whose sole run of ones is of size 2^{m_0}, which is consistent with the proposed assertion.

  For $j>1$, the concatenation of $c_j^{m_0}(01)$ with its complement has $(2j-2^p)\times 2-2=((2j-1)-2^p)\times 2$ runs of ones of size $2^{(m_0+1)-(p+1)-1}$, and $2\times(2^p-j)+1=2^{p+1}-(2j-1)$ runs of ones of size $2^{(m_0+1)-(p+1)}$. Since $j>1$, $2^p<2j-1<2^{p+1}$ holds, and we can conclude that the codeword in $G_{2j-1}^{m_0+1}$ obtained by concatenating $c_j^{m_0}(01)$ with its complement satisfies the proposed assertion.

- The concatenation of $c_j^{m_0}(10)$, if it exists, with its complement produces a codeword in $G_{2j}^{m_0+1}$ (see Arrow 5 in Figure 1).

  Note that both $c_j^{m_0}(01)$ and its complement $c_j^{m_0}(10)$ have $(j-2^{p-1})\times 2$ runs of ones of size 2^{m_0-p-1}, and 2^p-j runs of ones of size 2^{m_0-p}. Consequently, the result of the concatenation has $(j-2^{p-1})\times 2\times 2=(2j-2^p)\times 2$ runs of ones of size $2^{(m_0+1)-(p+1)-1}$, and $(2^p-j)\times 2=2^{p+1}-2j$ runs of ones of size 2^{m_0-p}. Since $2^p<2j\leq 2^{p+1}$, we can conclude that the proposed assertion holds for a codeword in $G_{2j}^{m_0+1}$ obtained by concatenating $c_j^{m_0}(10)$ with its complement.

- The concatenation of $c_j^{m_0}(10)$, if it exists, with itself produces a codeword in $G_{2j}^{m_0+1}$ (see Arrow 6 in Figure 1).

  Now, for $2^p < 2j \leq 2^{p+1}$) the resulting codeword has $2 \times (j - 2^{p-1}) \times 2 = (2j - 2^p) \times 2$ runs of ones of size $2^{m_0-p-1} = 2^{(m_0+1)-(p+1)-1}$, and $2 \times (2^p - j) = 2^{p+1} - 2j$ runs of ones of size $2^{m_0-p} = 2^{(m_0+1)-(p+1)}$. No runs of ones are altered, they are merely duplicated. This same argument applies to the concatenation of $c_j^{m_0}(01)$ with itself (Arrow 4 in Figure 1), and to the concatenation of $c_j^{m_0}(00)$ with itself (Arrow 7 in Figure 1).

- The concatenation of $c_j^{m_0}(00)$, if it exists, with its complement produces a codeword in $G_{2j+1}^{m_0+1}$ (see Arrow 8 in Figure 1).

  If $j = 0$, $c_j^{m_0}(00)$ is the all-zeros codeword. The concatenation with its complement (the all-ones codeword) produces a codeword in $G_1^{m_0+1}$ that has a single run of ones of size $2^{(m_0+1)-1}$.

  By assumption, for $j > 0$, the codeword $c_j^{m_0}(00)$ has $(j - 2^{p-1}) \times 2$ runs of ones of size 2^{m_0-p-1}, and $2^p - j$ runs of ones of size 2^{m_0-p}. The complement of $c_j^{m_0}(00)$ is $c_{j+1}^{m_0}(11)$. We will analyze the cases when $2^{p-1} < j < 2^p$ and $j = 2^p$ separately.

  Case 1: $2^{p-1} < j < 2^p$.

  Here we have that $2^{p-1} < j+1 \leq 2^p$, and $2^p < 2j+1 < 2^{p+1}$. By assumption, $c_{j+1}^{m_0}(11)$ has $((j+1) - 2^{p-1}) \times 2$ runs of ones of size 2^{m_0-p-1} and $2^p - (j+1)$ runs of ones of size 2^{m_0-p}. Consequently, the concatenation has $(j - 2^{p-1}) \times 2 + ((j+1) - 2^{p-1}) \times 2$=$((2j+1) - 2^p) \times 2$ runs of ones of size $2^{m_0-p-1} = 2^{(m_0+1)-(p+1)-1}$, and $2^p - j + 2^p - (j+1) = 2^{p+1} - (2j+1)$ runs of ones of size $2^{m_0-p} = 2^{(m_0+1)-(p+1)}$. The assertion therefore holds for the codeword in $G_{2j+1}^{m_0+1}$, obtained by concatenating $c_j^{m_0}(00)$ with its complement, when $2^{p-1} < j < 2^p$.

  Case 2: $j = 2^p$.

  Now we have that $2^p < j+1 \leq 2^{p+1}$ and $2^{p+1} < 2j+1 < 2^{p+2}$. In this case, $c_j^{m_0}(00)$ has all $j = 2^p$ runs of ones of size 2^{m_0-p-1}. Its complement $c_{j+1}^{m_0}(11)$ has $((j+1) - 2^p) \times 2$ runs of ones of size $2^{m_0-(p+1)-1} = 2^{m_0-p-2}$, and $2^{p+1} - (j+1)$ runs of ones of size $2^{m_0-(p+1)} = 2^{m_0-p-1}$. The result of the concatenation has $2^p + 2^{p+1} - (j+1) = 2^{p+1} - 1$ runs of ones of size 2^{m_0-p-1}, and $((j+1) - 2^p) \times 2$ runs of ones of size 2^{m_0-p-2}. Since $j = 2^p$, we can replace $2^{p+1}-1$ with $2^{p+2}-(2j+1)$ and $((j+1)-2^p)\times 2$ with $((2j+1)-2^{p+1})\times 2$. Thus, for $j=2^p$, the result of the concatenation of $c_j^{m_0}(00)$ with its complement is a codeword in $G_{2j+1}^{m_0+1}$ that has $2^{p+2}-(2j+1)$ runs of ones of size $2^{(m_0+1)-(p+2)}$, and $((2j+1)-2^{p+1})\times 2$ runs of ones of size $2^{(m_0+1)-(p+2)-1}$, where $2^{p+1}<2j+1<2^{p+2}$.

Combining the results stated so far in the proof, we conclude that Lemma 4 holds for $C(m_0+1)$. ■

4 One Deletion, One Repetition Case: Identification-problem Causing Codewords

Recall the discussion from Section 1. Let us assume that the error correction code is Reed-Muller(1,m) and that we are operating in the noise-free regime. We further assume that the received strings (codewords with synchronization errors) can be observed in isolation so that from the length of the received string, the total difference between the number of repetitions and deletions is known. Suppose we allow at most one deletion and at most one repetition. If n denotes the code length, and if $l=n-1$ bits are received, than one deletion case is declared. Similarly, if $l=n+1$ bits are received, one repetition case is declared. If $l=n$ bits are received it is either the case that no synchronization errors occurred or one of each kind occurred.

Our goal in this section is to analyze the case of at most one deletion and at most one repetition and to determine all pairs of codewords of $C(m)$ that can result in the same string under these assumptions. This enumeration will yield to the pruned code with improved synchronization error correction capabilities.

For small values of m, we have the following result.

Remark 4.1 *For $m=0,1,2$ we can show by inspection the following.*

$m=0$ *The only codewords are '0' and '1' and they can both result in an empty string.*

$m = 1$ *The codewords are '00', '11', '01', and '10'. Any two pair of codewords, except for '00' and '11', can result in the same string.*

$m = 2$ *The codewords are '0000', '1100', '0011','0110', '1111', '1010','0101', and '1001'. The codeword '0011' and any one of '0110', '0101', and '1001' can result in the same string. Similarly, the codeword '1100' and any one of '1001', '1010', and '0110' can result in the same string. The same is true for '0110', and any one of '1010' and '0101' as well as for '1001' and any one of '0101' and '1010'. Also, '1010' and '0101' can result in the same string.*

$m = 3$ *The confusable codewords are as follows. The codeword '11001100' can be confused with either one of '01100110', '10011001', and '10010110'. The codeword '01100110' can be confused with either one of '10011001', '01101001', '10010110', and '01011010'. The codeword '01101001' can be confused with either one of '10100101', '01011010', and '01010101'. The codeword '10100101' can be confused with either '10101010' or '01010101', or '01011010' and finally '10101010' and '01010101' can be confused. To complete the list of confusable codewords, take the complement of those listed explicitly.*

■

Before proceeding with the main theorem, we first establish a couple of useful results:

Remark 4.2 *Complementarity: Consider two distinct codewords* $\mathbf{c_a}$ *and* $\mathbf{c_b}$*, and their complements* $\overline{\mathbf{c_a}}$ *and* $\overline{\mathbf{c_b}}$*, all in* $C(m)$*. Then, if and only if* $\mathbf{c_a}$ *and* $\mathbf{c_b}$ *give rise to the same string after experiencing at most one deletion and one repetition each, then so do* $\overline{\mathbf{c_a}}$ *and* $\overline{\mathbf{c_b}}$*. In particular, iff such deletion in* $\mathbf{c_a}$ *occurs in a run of ones, then the corresponding deletion in* $\overline{\mathbf{c_a}}$ *occurs in a run of zeros.*

Lemma 5 *If* $\mathbf{c_a}$ *is a codeword belonging to* $C(m)$*, then* $\mathbf{c_a^*} = B(\mathbf{c_a})$ *is also a codeword in* $C(m)$ *where* $B(\mathbf{c_a})$ *is the string obtained by reading* $\mathbf{c_a}$ *backwards.*

Proof : To see this recall that $C(m)$ is described by a $k \times n$ generator matrix $\mathbf{G_m}$ where $k = m+1$ and $n = 2^m$. The first row of $\mathbf{G_m}$ consists of all ones, and the i^{th} row of $\mathbf{G_m}$, for $1 < i \leq m+1$

consists of 2^{i-1} alternating runs of ones and zeros, where each run is of size 2^{m-i+1}, and the leftmost run in each row is a run of ones. Now, we can describe the same code with the alternative generator matrix $\mathbf{G'_m}$, where each row i for $1 < i \leq m+1$ consists of 2^{i-1} alternating runs of zeros and ones, by simply replacing the i^{th} row of $\mathbf{G_m}$ with the sum of the first and the i^{th} row. Note that all rows in $\mathbf{G'_m}$ are equal to the corresponding rows of $\mathbf{G_m}$ read backwards, so that every linear combination of the rows of $\mathbf{G'_m}$ is equal to the same linear combination of the rows of $\mathbf{G_m}$ when read backwards.

Remark 4.3 *Reversibility: Consider two distinct codewords $\mathbf{c_a}$ and $\mathbf{c_b}$, and their reverses $\mathbf{c_a^*}$ and $\mathbf{c_b^*}$, all in $C(m)$, where by $\mathbf{c_a^*}$ we denote the codeword $\mathbf{c_a}$ read backwards. Then if and only if $\mathbf{c_a}$ and $\mathbf{c_b}$ give rise to the same string after experiencing at most one deletion and one repetition each, the same must be true for $\mathbf{c_a^*}$ and $\mathbf{c_b^*}$.*

The previous two remarks will be used throughout the proof of the main Theorem.

We now also introduce a useful auxiliary set of strings and state several properties of these strings. The proofs for the given statements are contained in [1]. These results will also be used in the proof of the main Theorem.

Definition 1 *For a codeword $\mathbf{c} \in C(m)$ let $\mathbf{d} = d(\mathbf{c})$ be the string whose entries are the lengths of consecutive runs in $\mathbf{c}$, read from left to right. Let $\mathcal{D}_m = \{\mathbf{d} | \mathbf{d} = d(\mathbf{c}), \mathbf{c} \in C(m)\}$, so that $\mathcal{D}_m$ represents the collection of all possible sequences of run lengths associated with the codewords of $C(m)$.* ■

Lemma 6 *[mirror-symmetry] $\forall \mathbf{c} \in C(m)$, the string $\mathbf{d} = d(\mathbf{c})$ possesses the mirror-symmetry property, i.e. the entry in position p in $\mathbf{d}$, denoted by $\mathbf{d}(p)$, is the same as the entry in position $l-p+1$, denoted by $\mathbf{d}(l-p+1)$, where l represents the length of string $\mathbf{d}$.*

Lemma 7 *If all entries in $\mathbf{d} = d(\mathbf{c})$ are either 1 or 2, with at least one entry being 1 and one being 2, then the leftmost entry equal to 2 must be in position 2^p, for some $p \geq 1$.*

Lemma 8 *If all entries in $\mathbf{d} = d(\mathbf{c})$ are either 1 or 2, with at least one entry being 1 and one being 2, then each run of 2's in $\mathbf{d}$ is of length $2^p - 1$, for some $p \geq 1$.*

Lemma 9 *If all entries in $\mathbf{d} = d(\mathbf{c})$ are either 1 or 2, with at least one entry being 1 and one being 2, then each inner run of 1's (where the inner run denotes a run with neighboring runs on each side) in $\mathbf{d}$ is of length $2^p - 2$, for some $p \geq 1$.*

Lemma 10 *If $\mathbf{d_a} = d(\mathbf{c_a})$ and $\mathbf{d_b} = d(\mathbf{c_b})$, for $\mathbf{c_a}$, $\mathbf{c_b} \in C(m)$ ($\mathbf{d_a}$, $\mathbf{d_b} \in \mathcal{D}_m$) and $m > 2$, are such that they have $2k+1$ and $2k$ entries respectively, and all their entries are 1 or 2, then in the first leftmost position in which they differ, call it p, the entry is 1 in $\mathbf{d_a}$ and is 2 in $\mathbf{d_b}$, and $p < k$.*

Proof: For the proofs please see Lemmas 4–8 in [1]. ■

We now state the main result.

Theorem 1 *For $m \geq 3$, the following codewords can result in the same string after each experiences at most one deletion and at most one repetition. (For the ease of proving the result, they are categorized into different groups).*

1. $c_j^m(10)$ *and* $c_j^m(01)$ } *Group 1*

2. $c_j^m(10)$ *and* $c_j^m(11)$
3. $c_j^m(10)$ *and* $c_{j-1}^m(00)$
4. $c_j^m(01)$ *and* $c_j^m(11)$
5. $c_j^m(01)$ *and* $c_{j-1}^m(00)$

} *Group 2*

6. $c_j^m(01)$ *and* $c_{j-1}^m(01)$
7. $c_j^m(10)$ *and* $c_{j-1}^m(10)$

} *Group 3*

8. $c_k^m(01)$ *and* $c_k^m(00)$
9. $c_k^m(01)$ *and* $c_{k+1}^m(11)$
10. $c_k^m(10)$ *and* $c_k^m(00)$
11. $c_k^m(10)$ *and* $c_{k+1}^m(11)$

} *Group 4*

12. $c^m_{j-1}(00)$ *and* $c^m_j(11)$ } *Group 5*

13. $c^m_k(00)$ *and* $c^m_{k+1}(11)$ } *Group 6*

14. $c^m_{j-1}(11)$ *and* $c^m_{j-1}(10)$
15. $c^m_{j-1}(11)$ *and* $c^m_{j-1}(01)$
16. $c^m_{j-2}(00)$ *and* $c^m_{j-1}(10)$
17. $c^m_{j-2}(00)$ *and* $c^m_{j-1}(01)$

} *Group 7*

18. $c^m_{k+1}(11)$ *and* $c^m_{k+1}(10)$
19. $c^m_{k+1}(11)$ *and* $c^m_{k+1}(01)$
20. $c^m_k(00)$ *and* $c^m_{k+1}(10)$
21. $c^m_k(00)$ *and* $c^m_{k+1}(01)$

} *Group 8*

22. $c^m_l(00)$ *and* $c^m_l(10)$
23. $c^m_l(00)$ *and* $c^m_l(01)$
24. $c^m_{l+1}(11)$ *and* $c^m_l(10)$
25. $c^m_{l+1}(11)$ *and* $c^m_l(01)$

} *Group 9*

26. $c^m_k(00)$ *and* $c^m_{k+1}(00)$
27. $c^m_{k+1}(11)$ *and* $c^m_{k+2}(11)$

} *Group 10*

28. $c^m_k(01)$ *and* $c^m_{k+1}(01)$
29. $c^m_k(10)$ *and* $c^m_{k+1}(10)$

} *Group 11*

30. $c^m_{j-1}(11)$ *and* $c^m_j(11)$
31. $c^m_{j-2}(00)$ *and* $c^m_{j-1}(00)$

} *Group 12*

32. $c^m_j(11)$ *and* $c^m_{j-1}(10)$
33. $c^m_j(11)$ *and* $c^m_{j-1}(01)$
34. $c^m_{j-1}(00)$ *and* $c^m_{j-1}(10)$
35. $c^m_{j-1}(00)$ *and* $c^m_{j-1}(01)$

} *Group 13*

where $j = 2^{m-1}$, $k = 2^{m-2}$, *and* $l = 3 * 2^{m-3}$.

Note that we have already shown this result for $m = 3$, by Remark 4.1. In the remainder we will assume that $m > 3$. Since the cases where exactly one synchronization error per codeword occurs are contained in the case under current consideration, we first enumerate the ones that cause the identification problem under a single deletion. These pairs are listed in 1). through 11). in Theorem 1, and are obtained from Theorem 2 in [1]. It also shown in [1] that no two codewords can result in the same string when each experiences a repetition.

Thus, the remaining possibilities are when two distinct codewords give rise to the same string, one of the codewords experiences one repetition and one deletion in different runs and the other codeword experiences no synchronization errors, and another possibility is that they each experience one deletion and one repetition. In the former case, it would be necessary that there exist two codewords, call them $\mathbf{c_a}$ and $\mathbf{c_b}$ such that when $\mathbf{c_a}$ experiences a repetition in position say p_1 and a deletion in position say p_2, the resulting string would again be a valid codeword. It is now sufficient to consider the codewords $\mathbf{c_a}$ and $\mathbf{c_b}$ which satisfy the following: $\mathbf{c_a}$ experiences a deletion in position p_1 and $\mathbf{c_b}$ experiences a deletion in position p_2 where the bit in $\mathbf{c_b}$ in position p_2 belongs to a run of size at least 2. The set of such pairs ($\mathbf{c_a}$, $\mathbf{c_b}$) is contained in the collection of pairs listed in Theorem 2 in [1]. Specifically, they are already listed in 2). through 11). in Theorem 1.

We now focus on the latter case, namely when $\mathbf{c_a}$ and $\mathbf{c_b}$ both experience a deletion and a repetition. We will insist that both $\mathbf{c_a}$ and $\mathbf{c_b}$ experience both types of errors in different runs (otherwise the analysis can be reduced to the earlier cases of having one synchronization error per codeword). As discussed before, it is sufficient to consider the cases when the total number of runs in $\mathbf{c_a}$ and $\mathbf{c_b}$ differs by 0, 1, and 2. Without loss of generality assume that the total number of runs in $\mathbf{c_a}$ is at least equal to the total number of runs in $\mathbf{c_b}$. Let $d_a = d(\mathbf{c_a})$ and $d_b = d(\mathbf{c_b})$. We treat the cases $d_a = d_b$, $d_a = d_b + 1$, and $d_a = d_b + 2$ separately.

1. length($\mathrm{d_a}$)=length($\mathrm{d_b}$)

As in the case of single deletion, it is necessary that $\mathbf{c_a}$ and $\mathbf{c_b}$ are complements of each other. Since $\mathbf{c_a}$ and $\mathbf{c_b}$ disagree in the leftmost bit, it is necessary that one of them experiences a deletion in the leftmost bit.

Without loss of generality we can assume that $\mathbf{c_a}$ experiences a deletion in the leftmost bit.

Then the leftmost bit in $\mathbf{c_b}$ is the same as the second leftmost bit in $\mathbf{c_a}$ (the deletion in $\mathbf{c_b}$ cannot occur in its leftmost bit as then it would be impossible to construct the same string by applying a repetition to the strings consisting of the remaining $n-1$ bits in $\mathbf{c_a}$ and $\mathbf{c_b}$), so we conclude that both $\mathbf{c_a}$ and $\mathbf{c_b}$ start with runs of length 1. By Lemma 3, the rightmost runs in $\mathbf{c_a}$ and $\mathbf{c_b}$ are also of length 1, and all other runs in between are of length 1 or 2.

It is further necessary that the deletion in $\mathbf{c_b}$ occurs in the rightmost bit in a run of size 1, since otherwise the resulting strings obtained by applying a deletion and a repetition to $\mathbf{c_a}$ and $\mathbf{c_b}$, would end in different types of runs. Now we are left with the task of determining possible locations of repetitions in inner runs.

By starting with the leftmost bit in $\mathbf{c_b}$, and by matching up the appropriate bits in $\mathbf{c_a}$ and $\mathbf{c_b}$ (i.e. the bit in position $i+1$ in $\mathbf{c_a}$ is the same as the bit in position i in $\mathbf{c_b}$, and is the complement of the bit in position $i+1$ in $\mathbf{c_b}$) up until the very next synchronization error in either codeword, we conclude that $\mathbf{c_a}$ (and $\mathbf{c_b}$) starts with a substring consisting of alternating bits. Now suppose that the very next error, occurring say at position p_2 is again in $\mathbf{c_a}$ (i.e. it must be that the bit in position p_2 in $\mathbf{c_a}$ is being repeated). This would imply that the bit in $\mathbf{c_b}$ in position p_2 is the same as bit in the same position in $\mathbf{c_a}$, which is impossible for complement codewords. Therefore the very next error must be a repetition in $\mathbf{c_b}$. Since all runs in $\mathbf{c_a}$ and $\mathbf{c_b}$ are of size 1 or 2 only, it is necessary that the repeated bit in $\mathbf{c_b}$ in position p_2 belongs to a run of size 1, and that a 2-bit run of the same type contains bits in positions p_2+1 and p_2+2 in $\mathbf{c_a}$. Now suppose that the bit in position p_3 ($p_3 > p_2$) in $\mathbf{c_a}$ is repeated. By matching up the bits in positions between the repetitions we conclude that the substrings starting at position p_2+1 and ending at position p_3-1 consist of alternating runs of size 2 (we can think of $\mathbf{c_b}$ as trailing $\mathbf{c_a}$ by two bits). The repetition in $\mathbf{c_a}$ must occur in the run of size 1, and in the remainder from position p_3+1 onwards, $\mathbf{c_b}$ is trailing $\mathbf{c_a}$ by one bit, so that substring in both codewords consists of alternating bits. Therefore, $\mathbf{c_a}$ has the following format: it consists of alternating runs of size 1, followed by alternating runs of size 2, followed again by alternating runs of size 1. Since $\mathbf{c_a}$ can be viewed as a concatenation of a codeword from $C(m-1)$ either with itself or with its complement, it is further necessary that $\mathbf{c_a}$ consists of a single run of size 1, followed by alternating runs of size 2, followed again by a single run of size 1, or that $\mathbf{c_a}$ consists of alternating runs of size 1, followed by a single run of size 2, followed again by alternating runs of size 1. Since the weight of $\mathbf{c_a}$ is even, by definition of $C(m)$, its outermost bits must be of the same type, and we can conclude that the only choices for the pair

$(\mathbf{c_a}, \mathbf{c_b})$ is $\mathbf{c_a} = c^m_{k+1}(11)$ and $\mathbf{c_b} = c^m_k(00)$ for $k = 2^{m-2}$, or vice versa, as well as $\mathbf{c_a} = c^m_j(11)$ and $\mathbf{c_b} = c^m_{j-1}(00)$ for $j = 2^{m-1}$. Note that by Remarks 4.2, and 4.3, these are the only such pairs. These are listed under Group 5 and Group 6.

2. length($\mathrm{d_a}$)=length($\mathrm{d_b}$)+1

It can be either the case that $\mathbf{c_a}$ experiences a deletion of the outermost 1-bit run and $\mathbf{c_b}$ experiences a deletion in a run of size at least 2 or that $\mathbf{c_a}$ experiences a deletion in an inner run of size 1 and $\mathbf{c_b}$ experiences a deletion in an outermost run of size 1.

a) Suppose first that $\mathbf{d_a}$ has even length.

Then $\mathbf{d_a}$ and $\mathbf{d_b}$ differ in the middle locations (since $\mathbf{c_a}$ and $\mathbf{c_b}$ can be viewed as the result of concatenation applied to the same codeword in $C(m-1)$ whereby no runs are altered in creating $\mathbf{c_a}$ and the outermost runs are merged in creating $\mathbf{c_b}$) so that $\mathbf{d_a}$ can be expressed as $\mathbf{d_a}$=[A11B] and $\mathbf{d_b}$ as $\mathbf{d_b}$=[A2B] where A and B are substrings of $\mathbf{d_a}$ and $\mathbf{d_b}$ and are mirror images of each other.

a.I.-If $\mathbf{c_a}$ experiences a deletion in the outermost 1-bit run, by reversibility property we can assume that the leftmost bit in $\mathbf{c_a}$ is deleted. Starting with substrings B, $\mathbf{d_a}$ and $\mathbf{d_b}$ look the same, so the last error is either a repetition in $\mathbf{c_a}$ or a deletion in $\mathbf{c_b}$, in the place that would correspond to the entry in $\mathbf{d_a}$ or $\mathbf{d_b}$ immediately preceding B.

We are then left with placing a deletion in $\mathbf{c_b}$ (or a repetition in $\mathbf{c_a}$), and a repetition in $\mathbf{c_b}$. This further necessitates the case of the left halves in $\mathbf{c_a}$ and $\mathbf{c_b}$ having the following property: there are exactly two consecutive entries that are different from each other, so that the left half of $\mathbf{c_a}$ consists of a run of 1's followed by a run of 2's followed by a run of 1's. Then the left half of $\mathbf{d_a}$ has the format: 1.12.21.1 (here and in the remainder by '1.1' we assume a substring consisting of 1's only), so by structural properties of the string $\mathbf{d_a}$, it must be either 1.121.121 or 12.2112.21. For the former case, $\mathbf{c_a}$ and $\mathbf{c_b}$ have $2^m - 2$ and $2^m - 3$ runs respectively, and it can be verified that $\mathbf{c_a}$ is $c^m_j(01)$ or $c^m_j(10)$ and $\mathbf{c_b}$ is $c^m_j(11)$ or $c^m_{j-1}(00)$ for $j = 2^{m-1} - 1$. These are given in the Group 7. For the latter case, $\mathbf{c_a}$ and $\mathbf{c_b}$ have $2^{m-1} + 2$ and $2^{m-1} + 1$ runs each. Then $\mathbf{c_a}$ is either $c^m_j(01)$ or $c^m_j(10)$ and $\mathbf{c_b}$ is either $c^m_j(11)$ or $c^m_{j-1}(00)$ for $j = 2^{m-2} + 1$. These pairs are listed in Group 8. Again by Remarks 4.2, and 4.3, these are the only such pairs.

a.II.-Now suppose that $\mathbf{c_b}$ experiences a deletion in its outermost run, which we can assume to be rightmost run, and that $\mathbf{c_a}$ experiences a deletion in an inner 1-bit run.

We are left with the task of placing repetitions in $\mathbf{c_a}$ and $\mathbf{c_b}$ as well as a segment of three consecutive runs in $\mathbf{c_a}$, the middle of which is a 1-bit run that gets deleted, and this segment either corresponds to a 2-bit or a 1-bit run in $\mathbf{c_b}$. As a consequence of the current assumptions on the deletions, we have the following changes in $\mathbf{d_a}$ and $\mathbf{d_b}$ as a result of synchronization errors, whereby a ”joint” error means that the deletion in $\mathbf{c_a}$ and the repetition in $\mathbf{c_b}$ occur in the same segment.

case	in $\mathbf{d_a}$	in $\mathbf{d_b}$	comment
1	211$\rightarrow$ 3(del), 1$\rightarrow$ 2 (rep)	2$\rightarrow$ 3(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
2	112$\rightarrow$ 3(del), 1$\rightarrow$ 2 (rep)	2$\rightarrow$ 3(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
3	111$\rightarrow$ 2(del), 1$\rightarrow$ 2 (rep)	1$\rightarrow$ 2(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
4	111$\rightarrow$ 2(del), 1$\rightarrow$ 2 (rep)	1$\rightarrow$ 2(rep), 1$\rightarrow$ $\{\}$(del)	

In Cases 1-3 we want to place the repetition in $\mathbf{c_a}$ relative to the ”joint” error. Since $\mathbf{d_a}$ and $\mathbf{d_b}$ agree everywhere except in the middle, we have the following possible situations:

Suppose first that the innermost entries in $\mathbf{d_a}$ and/or $\mathbf{d_b}$ are changed due to the deletion in $\mathbf{c_a}$, and that this is the leftmost error. In particular, if this is the ”joint” error, and the innermost 11 in $\mathbf{d_a}$ is followed by 2 (description given under Case 2), then in the right halves of $\mathbf{d_a}$ and $\mathbf{d_b}$, there must be one change from runs of 2’s to runs of 1’s to accommodate the remaining error (repetition in $\mathbf{c_a}$), so that $\mathbf{d_b}$ is 1.12.21.1. Then $\mathbf{d_b}$ can be 1.121.1 or 12.21, and $\mathbf{d_a}$ can be 1.1 or 12.2112.21. In the former case $\mathbf{d_a}$ has no entries equal to 2, and the latter case corresponds to $\mathbf{c_a}$ and $\mathbf{c_b}$ with $2^{m-1}+2$ and $2^{m-1}+1$ runs each. Then $\mathbf{c_a}$ is either $c_j^m(01)$ or $c_j^m(10)$ and $\mathbf{c_b}$ is either $c_j^m(11)$ or $c_{j-1}^m(00)$ for $j=2^{m-2}+1$. These pairs are listed in Group 8. Since the innermost entries cannot be 21 in $\mathbf{d_a}$, and the innermost entry cannot be 1 in $\mathbf{d_b}$, we can rule out Cases 1 and 3 for when the ”joint” error causes the changes of the innermost entries in $\mathbf{d_a}$ and $\mathbf{d_b}$. However, for Case 4, the innermost 11 in $\mathbf{d_a}$ are altered due to deletion but the innermost 2 in $\mathbf{d_b}$ remains unaltered. Then, to accommodate additional repetitions, we require two changes in the right halves of $\mathbf{d_a}$ and $\mathbf{d_b}$, once in going from runs of 1's to runs of 2's, and once in going from runs of 2's to runs of 1's. Then $\mathbf{d_a}$ is 1.12.21.12.21.1, so that it must be 1.121.121.1 ($\mathbf{d_a}$ could also be 12.2112.21 but then it would not have a run of 1's of size 3 as required). This corresponds to $\mathbf{d_b}$ equal to 1.121.121.121.1, and in turn pairs established in Group 7.

If the "joint" error occurs in the left halves of $\mathbf{c_a}$ and $\mathbf{c_b}$, it must correspond to Case 1. Then suppose that 211 in $\mathbf{d_a}$ which gets altered by the "joint" error starts at position p. By mirror-symmetry p-th rightmost entry in both $\mathbf{d_a}$ and $\mathbf{d_b}$ must be 2, so by matching up the appropriate entries, the $p+1$-th rightmost entry in $\mathbf{d_b}$ is 2 as well. However its mirror image is 1 in $\mathbf{d_b}$ unless 211 in $\mathbf{d_a}$ spans its innermost entries. Moreover, there is one change from runs of 2's to runs of 1's in the right half of $\mathbf{d_b}$ so $\mathbf{d_b}$ itself is 12.21, which then yield codewords listed in Group 8.

If for Case 4, deletion in $\mathbf{c_a}$ occurs in the left half, let us first consider the case when there is at least one 2 in B. Then B must be 2.21 as otherwise there would be more than 2 mismatches between the appropriate entries in the right halves of $\mathbf{d_a}$ and $\mathbf{d_b}$. If B is 1.1, then the resulting codewords are already given in Group 2.

Another choice is if the leftmost error is a repetition in $\mathbf{c_a}$. It then must affect the innermost entries in $\mathbf{d_a}$ and $\mathbf{d_b}$. However, in all cases it is impossible to place the appropriate patterns that correspond to the deletion in $\mathbf{c_a}$ in the substrings B in $\mathbf{d_a}$ and $\mathbf{d_b}$.

b) Now suppose that the length of $\mathbf{d_a}$ is odd.

b.I.-Let us first consider the case when $\mathbf{c_a}$ experiences a deletion in its outermost run, which by reversibility property, can be assumed to be the rightmost run (of size 1).

Then it is necessary that $\mathbf{d_a}$ and $\mathbf{d_b}$ are such that the entry in position i in $\mathbf{d_b}$ is the same as the entry in position i in $\mathbf{d_a}$ except for three pairs of entries. In these exceptions, the entry is 1 in $\mathbf{d_a}$ and its counterpart is 2 in $\mathbf{d_b}$ twice, and the entry is 2 in $\mathbf{d_a}$ and its counterpart in $\mathbf{d_b}$ is 1 once.

Since $\mathbf{d_b}$ has an even number of entries, its two innermost entries must be 1 ($\mathbf{d_b}$ must start with 1, and the left and the right halves of $\mathbf{d_b}$ are the same).

Since $\mathbf{d_a}$ has an odd number of entries, and its outermost entries are both 1, its innermost entry is 2.

Now, this 2 in $\mathbf{d_a}$ is in the same position when counted from the left as a 1 in $\mathbf{d_b}$, so that $\mathbf{d_a}$ and $\mathbf{d_b}$ look like (tentatively, where '...' indicate the current unknown substrings), and the overline indicates the run that disappears as a consequence of having a deletion in the outermost bit in $\mathbf{c_a}$.

$\mathbf{d_a}$=1...2...$\overline{1}$

$\mathbf{d_b}$=1...11...1

We now need to place 2 in $\mathbf{d_b}$ and 1 in $\mathbf{d_a}$ twice in the same positions and have all other entries in the same positions be the same. By the mirror symmetry of $\mathbf{d_b}$, if we place such 2 in $\mathbf{d_b}$ in the l-th place from the left, we must place the remaining 2 in $\mathbf{d_b}$ in the l-th place from the right.

Then,

$\mathbf{d_a}$=1...$\underline{1}$...2...$\underline{1}$...$\overline{1}$

$\mathbf{d_b}$=1...$\underline{2}$...11...$\underline{2}$...1

where the left (right) underlined places are in the l-th position counted from the left (right).

By using mirror symmetry of $\mathbf{d_a}$ and $\mathbf{d_b}$ and the fact that the remaining entries in the same positions in $\mathbf{d_a}$ and $\mathbf{d_b}$ must be the same, we conclude that all remaining entries must be equal to 1, i.e. $\mathbf{d_a}$=1.121.1 and $\mathbf{d_b}$=1.121.121.1 where 1.1 indicates a substring of all 1's.

Then $\mathbf{c_a}$ is either $c^m_{j+1}(11)$ or $c^m_j(00)$ and $\mathbf{c_b}$ is either $c^m_j(10)$ or $c^m_j(01)$ for $j = 2^{m-1} - 1$. It can be verified that in all 4 choices, the same string can result from $\mathbf{c_a}$ and $\mathbf{c_b}$. These pairs are listed in Group 13.

b.II.-Now we consider the case when $\mathbf{c_a}$ experiences a deletion in an inner 1-bit run and $\mathbf{c_b}$ experiences a deletion of the outermost run, which we can take to be its rightmost run.

As in the case analyzed in a.II we write the table indicating various error patterns. We have the following changes in $\mathbf{d_a}$ and $\mathbf{d_b}$ as a result of synchronization errors:

case	in $\mathbf{d_a}$	in $\mathbf{d_b}$	comment
1	211$\rightarrow$ 3(del), 1$\rightarrow$ 2 (rep)	2$\rightarrow$ 3(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
2	112$\rightarrow$ 3(del), 1$\rightarrow$ 2 (rep)	2$\rightarrow$ 3(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
3	111$\rightarrow$ 2(del), 1$\rightarrow$ 2 (rep)	1$\rightarrow$ 2(rep), 1$\rightarrow$ $\{\}$(del)	”joint” error
4	111$\rightarrow$ 2(del), 1$\rightarrow$ 2 (rep)	1$\rightarrow$ 2(rep), 1$\rightarrow$ $\{\}$(del)	

Suppose first that the repetition in $\mathbf{c_a}$ occurs before the ”joint” error, and that it corresponds to the p-th leftmost entries in $\mathbf{d_a}$ and $\mathbf{d_b}$, which are then 1 and 2 respectively. Then the p-th rightmost entries must be 1 and 2 in $\mathbf{d_a}$ and $\mathbf{d_b}$, respectively. In particular, if that 2 in the p-th rightmost position in $\mathbf{d_b}$ is after the ”joint” error, by matching up the appropriate entries in $\mathbf{d_a}$ and $\mathbf{d_b}$, it would follow that last $p-1$ entries in $\mathbf{d_a}$ are all 2, which is impossible. If that 2 is a part of the segment affected by the ”joint” error (can hold for Cases 1 and 2), then the rightmost 2 in $\mathbf{d_a}$ would be in the $p+1$-th position, and the rightmost 2 in $\mathbf{d_b}$ would be in its p-th rightmost position, which by Lemmas 6 and 7 cannot hold simultaneously, or $\mathbf{d_b}$ would end in a run of 2’s which is also impossible. Similarly, if that 2 is in the position before the ”joint” error, for Case 1, the positions of the rightmost 2’s in $\mathbf{d_a}$ and $\mathbf{d_b}$ would violate Lemmas 6 and 7, and for Case 2, $\mathbf{d_b}$ would end in a 2. For the Case 3, this pattern of errors would imply that $\mathbf{d_a}$ and $\mathbf{d_b}$ have all entries equal to 2 inbetween the repetition and its mirror image, and all entries equal to 1 outside of these errors. Then, however, the location of the leftmost 2’s in $\mathbf{d_a}$ and $\mathbf{d_b}$ differs by 1, which is not possible by Lemma 7.

We now consider the error pattern in which the ”joint” error occurs before the repetition. If there is an entry equal to 2 in the segment before the ”joint” error, then by matching up the appropriate entries in $\mathbf{d_a}$ and $\mathbf{d_b}$, the leftmost 2 in that run of 2’s has a mirror image in the right half of $\mathbf{d_b}$ which must be matched with a 1 in $\mathbf{d_a}$, i.e. it corresponds to the repetition in $\mathbf{c_a}$. Moreover, the ”joint” error cannot have the corresponding entry be 1 in $\mathbf{d_b}$ (Case 3) as its mirror image must be matched with 2 in the right half of $\mathbf{d_a}$. If the ”joint” error is as given in Case 1, it would be necessary that in the right halves of $\mathbf{d_a}$ and $\mathbf{d_b}$ exist two runs of 2’s whose sizes are two consecutive numbers, which by Lemma 8 is not possible. Similarly, for Case 2, $\mathbf{d_a}$ and $\mathbf{d_b}$ would contain runs of 2's of two consecutive sizes.

We now consider the case when the "joint" error is the leftmost error, and there are no 2's in $\mathbf{d_a}$ and $\mathbf{d_b}$ prior to the positions that correspond to this "joint" error. In particular, if no entry prior to the "joint" error is equal to 2, then for Case 1, the mirror image of 2 in $\mathbf{d_b}$, say in the p-th rightmost position that corresponds to the "joint" entry, would have to correspond to the repetition location in $\mathbf{c_a}$, and by matching up the appropriate entries in between the locations of the "joint" error and the repetition, it would follow that 2 in the p-th rightmost entry in $\mathbf{d_b}$ is preceded by a 12, whereas 2 in the p-th rightmost entry in $\mathbf{d_a}$ is preceded by 1. Then $\mathbf{d_a}$ and $\mathbf{d_b}$ would have runs of 2's of two consecutive sizes which is impossible by Lemma 8.

For Case 2, again the mirror image of 2 in $\mathbf{d_b}$ that corresponds to the "joint" entry, would have to correspond to the repetition location in $\mathbf{c_a}$, but then the rightmost 2's in $\mathbf{d_a}$ and $\mathbf{d_b}$ would be in positions that differ by 2, which by Lemmas 6 and 7 can only be 4 and 2. However then $\mathbf{d_a}$ starts with 111211 and $\mathbf{d_b}$ starts with 121, and to match the entries in between they must be a repeated sequence 211. Then $\mathbf{d_b}$ is 121121...121121, and $\mathbf{d_a}$ is 1112112...112111, and have lengths $3 \times 2^{m-2}$ and $3 \times 2^{m-2} + 1$ respectively. The resulting codewords $\mathbf{c_a}$ and $\mathbf{c_b}$ are as given in Group 9.

For Case 3, let us suppose that due to the repetition, p-th rightmost entry in $\mathbf{d_b}$ is 2 and $p-1$-th rightmost entry in $\mathbf{d_a}$ is 1. Then if p-th entry in $\mathbf{d_b}$ is strictly located in the right half of $\mathbf{d_b}$, by mirror-symmetry, the p-th leftmost entry in $\mathbf{d_b}$ must be 2 (which is by assumption after the "joint" error, as the "joint" error is not preceded by a 2) so that $p+2$-th entry in $\mathbf{d_a}$ must be 2 as well, which are also the first leftmost entries equal to 2 in both $\mathbf{d_a}$ and $\mathbf{d_b}$. By Lemma 7 these must be in positions 4 and 2 in $\mathbf{d_a}$ and $\mathbf{d_b}$ respectively, and as a result we obtain the same $\mathbf{d_a}$ and $\mathbf{d_b}$ that yield the codewords already listed in 9.

The last case to consider is Case 4 when the errors cause the following modifications in $\mathbf{d_a}$ and $\mathbf{d_b}$: Deletion in $\mathbf{d_a}$ converts 111 to 2, and in $\mathbf{d_b}$ converts 1 to an empty string and the repetitions in both of them convert 1 to 2.

We first want to place the repetition errors relative to the deletion in $\mathbf{c_a}$. We label them (2,1) and

(1,2), for when the repetitions occur in $\mathbf{c_b}$, and in $\mathbf{c_a}$, respectively. If (2,1) precedes the deletion, it itself must be preceded by (1,2) as it is not possible to have 2 in $\mathbf{d_a}$ and 1 in $\mathbf{d_b}$ in the first leftmost position they differ in, by Lemma 8.

It is also not possible to have (1,2) precede the deletion, as then $\mathbf{d_a}$ would end in 2, unless (1,2) and the deletion in $\mathbf{c_a}$ are associated with positions that are mirror images of each other in $\mathbf{d_b}$ (otherwise $\mathbf{d_a}$ would end in 2). In that case, we start matching the entries in $\mathbf{d_a}$ and $\mathbf{d_b}$ that are in between these two positions and we conclude that all those entries must be 1. There is only one remaining mismatched pair, i.e. the one we labelled (2,1), so we conclude that $\mathbf{d_a}$ must have a single 2 in its innermost entry and all other entries equal to 1. This however yields $\mathbf{c_a}$ and $\mathbf{c_b}$ already listed in Group 13.

We now consider the case when both mismatchings (1,2) and (2,1) occur after the deletion in $\mathbf{c_a}$. Suppose the segment 111 in $\mathbf{d_a}$, 2 in $\mathbf{d_b}$ that corresponds to the deletion in $\mathbf{c_a}$ is immediately preceded by a 2 in both $\mathbf{d_a}$ and $\mathbf{d_b}$. Now if, this segment is either followed by 2 and 2 in $\mathbf{d_a}$ and $\mathbf{d_b}$, or by 1 and 1 or by 2 and 1, one of Lemmas 8 and 9 will be violated. If it is followed by 1 in $\mathbf{d_a}$ and 2 in $\mathbf{d_b}$, this would correspond to (1,2) error. Then $\mathbf{d_a}$ would have a segment 21111 and $\mathbf{d_b}$ would have a segment 222 that start in the same position and whose second entry corresponds to the first position in which $\mathbf{d_a}$ and $\mathbf{d_b}$ differ. By mirror-symmetry $\mathbf{d_a}$ would have a segment 1112 and $\mathbf{d_b}$ would have a segment 222 that end in the same position, counted from the right. However, then there would be another mismatch between the corresponding entries that is 1 in $\mathbf{d_a}$ and 2 in $\mathbf{d_b}$. Thus the segment 111 in $\mathbf{d_a}$, 2 in $\mathbf{d_b}$ that corresponds to the deletion in $\mathbf{c_a}$ is immediately preceded by a 1 in both $\mathbf{d_a}$ and $\mathbf{d_b}$. The mirror image of this segment corresponds to the (1,2) error. The (2,1) error cannot occur after the (1,2) error as that would imply that $\mathbf{d_a}$ ends in a 2. Therefore the (2,1) error is in between the deletion in $\mathbf{c_a}$ and the (1,2) error. Moreover by matching up the appropriate entries in $\mathbf{d_a}$ and $\mathbf{d_b}$ the segment preceding the leftmost error as well as the segment following the (1,2) error consist of 1's only, as do the segments in between. Since only (2,1) error remains to be placed somewhere that 2 must correspond to the innermost entry in $\mathbf{d_a}$ which then has all other entries equal to 1. This in turn yields candidate codewords already listed in 13.

3. length($\mathrm{d_a}$)=length($\mathrm{d_b}$)+2

In this case, the deletion in $\mathbf{c_b}$ must occur in a run of size at least 2, and the deletion in $\mathbf{c_a}$

must occur in an inner run of size 1. All runs in $\mathbf{c_a}$ must be of length 1 or 2 each. If $\mathbf{d_b}$ had an entry larger than 2, that entry would have to be 4 but then $\mathbf{d_a}$ would not have any inner 1's as required. Thus $\mathbf{c_b}$ has runs of length 1 and 2 as well. Either the repetition in $\mathbf{c_b}$ and the deletion in $\mathbf{c_b}$ occur "jointly", or the repetition in $\mathbf{c_b}$ is in the run of size 1 and the deletion in $\mathbf{c_a}$ is in a 1-bit run neighbored by two 1-bit runs whereby these errors occur separately.

a) The deletion in $\mathbf{c_a}$ is in an inner 1-bit run that is neighbored by a 1-bit and a 2-bit run, and there is a repetition in $\mathbf{c_b}$ in the corresponding 2-bit run, or the deletion is in an inner 1-bit run in $\mathbf{c_a}$ which is neighbored by two 1-bit runs and there is a repetition in the corresponding 1-bit run in $\mathbf{c_b}$. In either case, we think of these errors (deletion in $\mathbf{c_a}$, repetition in $\mathbf{c_b}$) as "joint" error, and we need to place the remaining errors (deletion in $\mathbf{c_b}$, repetition in $\mathbf{c_a}$). These two remaining errors can be thought of as being of the same kind (repetition in a 1-bit run in $\mathbf{c_a}$ and the deletion in a 2-bit run in $\mathbf{c_b}$) so we call them x and we call the "joint" error y.

Then there are three different orderings of errors, but by reversibility property it is sufficient to only consider orderings yxx and xyx.

Suppose the ordering is yxx. Then $\mathbf{d_a}$ and $\mathbf{d_b}$ are:

1) $\mathbf{d_a}$ = A112B1C1D and $\mathbf{d_b}$ = A2B2C2D or

2) $\mathbf{d_a}$ = A211B1C1D and $\mathbf{d_b}$ = A2B2C2D or

3) $\mathbf{d_a}$ = A111B1C1D and $\mathbf{d_b}$ = A1B2C2D,

for some appropriately chosen substrings A, B, C and D.

In 1) by mirror symmetry |A|=|D|, and we start matching up the appropriate entries in $\mathbf{d_a}$ and $\mathbf{d_b}$. If C is not empty, it must end in 1, and we get that there is a substring '212' in $\mathbf{d_b}$ which by Lemma 9 is not possible. Thus C must be empty, and then we have that B consists of all 2's so that $\mathbf{d_a}$=A112.211D and $\mathbf{d_b}$=A22.22D where D is the reverse of A. Furthermore, A starts with 1, and D ends with 1. Suppose A has at least three runs (of 1's, 2's, and 1's). Then the last run in A cannot be a run of 1's as the inner runs of 1's in $\mathbf{d_a}$ cannot be k+2 and k,k>0 simultaneously by Lemma 9. Thus A ends in a run of 2's or it consists of 1's only. If A is 1.1, by Lemma 7 $\mathbf{d_a}$ would be 1112.2111 which is impossible for $m > 3$. Thus A has a run of 1's followed by a run of 2's. Suppose 11 following A is not in the middle of the left half of $\mathbf{d_a}$. By concatenation principle, A would have another inner 1 somewhere but that 1 in A in $\mathbf{d_b}$ would not have its mirror image in the left half of $\mathbf{d_b}$. Thus 11 in $\mathbf{d_a}$ must be in the middle of its left half which then implies that $\mathbf{d_b}$ has no inner 1's. Thus $\mathbf{d_b}$=1.12.21.1. In particular $\mathbf{d_b}$ is 12.21. Consequently $\mathbf{d_a}$ is 12.2112.2112.21 and the candidate $\mathbf{c_a}$ and $\mathbf{c_b}$ are $c_j^m(11)$ or $c_{j-1}^m(00)$ for $\mathbf{c_a}$ where $j = 2^{m-2} + 2$, and $c_{j'}^m(11)$ or $c_{j'-1}^m(00)$ for $\mathbf{c_b}$ where $j' = 2^{m-2} + 1$. These codewords are listed under Group 10.

For 2) we look at first 2 following A2 in $\mathbf{d_b}$. Suppose it immediately follows that substring. Then by mirror symmetry |A|=|D|-1. If the entry in $\mathbf{d_a}$ immediately following A211 is 1, B would be empty and C would consist of all 1's. However, $\mathbf{d_a}$ and $\mathbf{d_b}$ would have runs of 2's of two consecutive sizes which is not possible by Lemma 8. If A211 in $\mathbf{d_a}$ is followed by 2, then if entry immediately following A22 in $\mathbf{d_b}$ is 1, the runs of 2's starting with the leftmost entry in D in $\mathbf{d_a}$ and starting with 2 immediately preceding D in $\mathbf{d_b}$ would differ in length by 1, which is impossible by Lemma 8. Thus A22 in $\mathbf{d_b}$ is followed by 2. By matching up the appropriate entries, it further follows that C is empty and that B has all 2's. But then by concatenation rule, A has only 1 in its leftmost position and 2's everywhere else and we arrive at $\mathbf{d_a}$ and $\mathbf{d_b}$ as being 12.2112.2112.21 and 12.21, which in turn correspond to codewords already given in Group 10. If A2 in $\mathbf{d_b}$ is followed by 1, then by Lemma 9, B cannot have any 2's. Since B is not empty (by assumption there is a 1 in $\mathbf{d_b}$ following A2 and there is a 2 following B), it has all 1's, and 2 immediately following B in $\mathbf{d_b}$ has 2 immediately following C as its mirror image. By mirror symmetry all entries in C must be 1's as well. Then $\mathbf{d_a}$ and $\mathbf{d_b}$ are A21.1D and A21.121.12D, and their left halves can be viewed as $d_1' = d(c_1')$ and $d_2' = d(c_2')$ for some $c_1', c_2' \in C(m-1)$, whereby two innermost entries in d_1' and d_2' are 11. Then d_1' is A21.1 and d_2' is A21.121.1, but then the mirror symmetry of one of them has to be violated.

For 3), if the substring B has at least one entry equal to 2, then by Lemmas 7 and 9, A must be empty. However, by mirror symmetry $\mathbf{d_a}$ actually starts with 111211, which cannot be matched with $\mathbf{d_b}$ which itself must start with 12112, since then $\mathbf{d_a}$ ends with 112111 and $\mathbf{d_b}$ with 21121, and this rightmost 2 in $\mathbf{d_a}$ cannot be matched with 1 in $\mathbf{d_b}$. If B does not have any 2's then we can think of A being extended by the size of B, so that B is effectively empty and A ends in a run of 1's. We label the new A as A'.Then 2's neighboring C in $\mathbf{d_b}$ must be mirror images of each other, and C itself must be all 1's. Then $\mathbf{d_a}$ and $\mathbf{d_b}$ are A'1.1D and A'121.121D. We now view the left half of $\mathbf{d_b}$ in isolation. If in it 2 following A'1 had a mirror image in A', the mirror of the left half of $\mathbf{d_a}$ would be violated. Thus 2 following A'1 in $\mathbf{d_b}$ is in the middle of its left half and A' has all 1's. This in turn implies that $\mathbf{d_a}$ is all 1's and $\mathbf{d_a}$ and $\mathbf{d_b}$ correspond to $\mathbf{c_a}$ and $\mathbf{c_b}$ listed in Group 3.

Consider now the error pattern xyx.

Then $\mathbf{d_a}$ =A1BEC1D and $\mathbf{d_b}$=A2BFC2D where E in $\mathbf{d_a}$ and F in $\mathbf{d_b}$ contain 112 and 2 or 211 and 2, or 111 and 1. Strings A and D must be mirror images of each other, and by matching up the same entries in $\mathbf{d_a}$ and $\mathbf{d_b}$ in B and C we end up with the unmatched middle which is either of length 4 and 2 in $\mathbf{d_a}$ and $\mathbf{d_b}$ respectively or 3 and 1 in $\mathbf{d_a}$ and $\mathbf{d_b}$ respectively. For the latter case, to preserve the mirror symmetry of $\mathbf{d_a}$, the isolated entries must be 111 in $\mathbf{d_a}$ and 1 in $\mathbf{d_b}$. However the run of 1's to which that 1 in $\mathbf{d_b}$ belongs would be of odd length, which is impossible by Lemma 9. For the former case, the four innermost entries in $\mathbf{d_a}$ are 2112 and are 22 in $\mathbf{d_b}$ or are 1111 in $\mathbf{d_a}$ and are 11 in $\mathbf{d_b}$ (2112 and 11 does not fit the pattern of $\mathbf{d_a}$ and $\mathbf{d_b}$). Then either $\mathbf{d_b}$ has all entries equal to 2, which gives rise to $\mathbf{c_b}$ being $c_j^m(01)$ or $c_j^m(10)$ for $j = 2^{m-2}$ and $\mathbf{c_a}$ being $c_{j+1}^m(01)$ or $c_{j+1}^m(10)$. These pairs are listed in Group 11. For when the innermost entries are 1111 and 11 in $\mathbf{d_a}$ and $\mathbf{d_b}$ respectively, if at least one of B and C contains a 2, we would have that there exist inner runs of 1's in $\mathbf{d_a}$ and $\mathbf{d_b}$ that differ in length by 2, which by Lemma 9 is impossible. Thus B and C must contain only 1's, and 2's bordering A and D in $\mathbf{d_b}$, by mirror symmetry, must be in the middle of the left and the right half of $\mathbf{d_b}$, respectively, and A itself must be all 1's as well. Then $\mathbf{d_a}$ is all 1's and the resulting $\mathbf{c_a}$ and $\mathbf{c_b}$ are already established in Group 3.

b) Now suppose that the deletion in $\mathbf{d_a}$ and the repetition in $\mathbf{d_b}$ are not "joint" errors.

We again think of the deletion in $\mathbf{c_b}$ (must be in a run of size 2) and the repetition in $\mathbf{c_a}$ (must be in a run of size 1) as equivalent errors, and we denote them by x. Let y be the deletion location in $\mathbf{c_a}$ and z be the repetition location in $\mathbf{c_b}$.

We can order x, x, y, z in 12 ways but by reversibility it is sufficient to look at 6 of these. Moreover, z cannot be the rightmost or the leftmost error by an argument similar to that given in Lemma 10. Suppose that $\mathbf{d_a}$ and $\mathbf{d_b}$ have lengths $l+2$ and l respectively. Then there exist another $\mathbf{d_c}$ of length $l+1$ such that $\mathbf{d_c} = d(\mathbf{c_c})$, where $\mathbf{c_c} \in C(m)$. If l is even, we apply Lemma 10 to $\mathbf{d_b}$ and $\mathbf{d_c}$, and conclude that the first leftmost position, say p, in which they differ, $\mathbf{d_b}$ is 2 and $\mathbf{d_c}$

is 1. This p is strictly in the left halves of $\mathbf{d_b}$ and $\mathbf{d_c}$. Since $\mathbf{d_c}$ and $\mathbf{d_a}$ disagree only in the middle, i.e. when counted from the left, first time in position $l_0/2$ where l_0 is the length of $\mathbf{d_a}$, it follows that the leftmost position in which $\mathbf{d_b}$ and $\mathbf{d_a}$ disagree is still p, and in that position $\mathbf{d_b}$ is 2 and $\mathbf{d_a}$ is 1. When l is odd the argument follows similarly.

It is therefore sufficient to analyze the following cases: yzxx, yxzx, and xyzx.

-For xyzx we have that $\mathbf{d_a}$ = A1B111C2D1E and $\mathbf{d_b}$ = A2B2C1D2E

We first observe that $|$A=$|$E$|$. Then irrespective of how are the sizes of B and D related, the mirror symmetry of one of $\mathbf{d_a}$ and $\mathbf{d_b}$ will be violated. In particular for $|\text{B}| = |\text{D}|$, 2 immediately following B would have 1 immediately preceding D as its mirror-image. If $|\text{B}| > |\text{D}|$, 2 immediately preceding D in $\mathbf{d_a}$ and 1 immediately preceding D in $\mathbf{d_b}$, would have mirror images in B in the same positions. If this 2 in $\mathbf{d_b}$ is its own image, $\mathbf{d_a}$ and $\mathbf{d_b}$ would agree up to the innermost entry of $\mathbf{d_b}$, so that $\mathbf{d_a}$ is then A121A' and $\mathbf{d_b}$ is A2A', for A' the reverse of A, which contradicts current assumptions on $\mathbf{d_a}$ and $\mathbf{d_b}$.

-For yzxx we have that $\mathbf{d_a}$ = A111B2C1D1E and $\mathbf{d_b}$ = A2B1C2D2E

If both A and B contain at least one 2 each, there would be an inner run of 1's in $\mathbf{d_a}$ of odd length, which by Lemma 9 is impossible. If A contains no 2's , but B does, then by Lemma 7 A must be of size $2^{l_1} - 1$ for some l_1. Now by Lemma 9 the run of 1's in B immediately preceding its leftmost 2, is of size $2^{l_2} - 2$. Then the leftmost run of 1's in $\mathbf{d_a}$ is of size $2^{l_1} - 1 + 3 + 2^{l_2} - 2$, which is impossible by Lemma 7. Thus B must contain only 1's.

By matching up the appropriate entries in $\mathbf{d_a}$ and $\mathbf{d_b}$ it turns out that D must consist of only 1's as well. Now, the isolated 2 between C and D in $\mathbf{d_b}$ is either the innermost entry in $\mathbf{d_b}$ or it has its mirror image, which must be in C (it cannot be in E as its counterpart in $\mathbf{d_a}$ is 1 and would have its own mirror image in the same position in E). In the former case, $\mathbf{d_b}$ is A21.121.12E where E is the reverse of A. By the concatenation principle, A and E must contain only 1's as well, so that $\mathbf{c_a}$ has $2^m - 1$ runs and $\mathbf{c_b}$ has $2^m - 3$ runs, and $c_1 = c_j^m(zz)$, $c_2 = c_{j-1}^m(zz)$ for $z = 1$ and $j = 2^{m-1}$, or $z = 0$ and $j = 2^{m-1} - 1$. These pairs are listed under Group 12. In the latter case, by mirror symmetry, all entries in C starting with the mirror image of the isolated 2 onwards must be all 2's as well. That substring in C cannot be preceded by a 1, as that would imply that $\mathbf{d_a}$ and $\mathbf{d_b}$ have runs of 2's of two consecutive lengths, which is impossible by Lemma 8. Thus $\mathbf{d_a}$ and $\mathbf{d_b}$ are A1.12.21.1E and A21.12.21.12E respectively, where A is followed by at least three 1's in $\mathbf{d_a}$. If the innermost run of 2's (which is C) in $\mathbf{d_a}$ has length more than 1, by concatenation principle A must start with 12, so that in the remainder of $\mathbf{d_a}$ all runs of 1's are either of length 1 or 2, which contradicts the earlier requirement that there exists a run of size at least 3. Thus, the innermost run of 2's in $\mathbf{d_a}$ has size 1, and $\mathbf{d_a}$ is A1.121.12E, which gives rise to the codeword pairs listed in Group 12.

-Finally, for yxzx, we have: $\mathbf{d_a}$ is A111B2C1D2E and $\mathbf{d_b}$ = A2B1C2D1E.

If $|$A$|$=$|$E$|$, the mirror-symmetry is violated. If $|$A$| < |$E$|$, then 1 and 2 immediately following A in $\mathbf{d_a}$ and $\mathbf{d_b}$ have mirror images in E (if that 2 in c_b is its own image, then $\mathbf{d_a}$ and $\mathbf{d_b}$ would agree up to the innermost entry in $\mathbf{d_b}$, and $\mathbf{d_b}$ would be A2A' and $\mathbf{d_a}$ would be A111A', where A' denotes the reverse of A, but which is impossible) in the same positions, but then the mirror-symmetry property is violated. Similarly, for $|$A$| > |$E$|$, the mirror-symmetry property is violated as well. ■

4.1 Pruning of the code

Recall that the i^{th} row of $\mathbf{G_m}$, for $1 < i \leq m+1$ consists of 2^{i-1} alternating runs of ones and zeros, and that each run is of length 2^{m-i+1}. Observe that the i^{th} row is then precisely $c^m_{2^{i-2}}(10)$. In particular, the last two rows of $\mathbf{G_m}$ are $c^m_{2^{m-2}}(10)$ for i = m and $c^m_{2^{m-1}}(10)$ for i = $m+1$.

We write $\mathbf{c} \in C(m)$ as $\mathbf{xG_m}$, where $\mathbf{x}$ is a $(m+1)$-dimensional message vector so that $c^m_{2^{m-1}}(10)$ = $[0, 0, \ldots, 0, 1]\mathbf{G_m}$ and $c^m_{2^{m-1}}(01)$ = $[1, 0, \ldots, 0, 1]\mathbf{G_m}$. Similarly, $c^m_{2^{m-2}}(10)$ is $[0, 0, \ldots, 0, 1, 0]\mathbf{G_m}$ and $c^m_{2^{m-2}}(01)$ is $[1, 0, \ldots, 0, 1, 0]\mathbf{G_m}$.

Observe that either $c^m_{2^{m-1}}(10)$ or its complement appears in each codeword pair in Groups 1, 2 and 3, and that either $c^m_{2^{m-2}}(10)$ or its complement appears in each codeword pair in Groups 4 and 11. The codeword $c^m_{2^{m-1}}(11)$ corresponds to the input $[1, 1, 0, \ldots 0, 1]$ and its complement, $c^m_{2^{m-1}-1}(00)$, to the input $[0, 1, 0 \ldots, 0, 1]$. At least one of them appears in each codeword pair in Groups 5, 12 and 13. Likewise, the codeword $c^m_{2^{m-2}+1}(11)$ corresponds to the input $[1, 0, \ldots 0, 1, 1]$

and its complement, $c^m_{2^{m-2}}(00)$, to the input $[0, 0 \ldots, 0, 1, 1]$, and at least one of them appears in each codeword pair in Groups 6, 8 and 10. The codeword $c^m_{2^{m-1}-2}(00)$ corresponds to the input $[0, 0, \ldots 0, 1, 0, 1]$ and its complement, $c^m_{2^{m-1}-1}(11)$, to the input $[1, 0, \ldots 0, 1, 0, 1]$, and one of them appears in each pair in Group 7. Lastly, $c^m_{3*2^{m-3}}(10)$ corresponds to the input $[0, 0, \ldots 0, 1, 1, 1]$ and its complement, $c^m_{3*2^{m-3}}(01)$, to the input $[1, 0 \ldots, 0, 1, 1, 1]$, and one of them appears in each pair in Group 9.

Therefore, for each codeword pair, at least one codeword has the message with a nonzero component in either position m or $m + 1$. Consider a matrix consisting of the top $m - 1$ rows of $\mathbf{G_m}$. It has $m - 1$ rows and no linear combinations of its rows give rise to codewords causing the identification problem. Thus, if instead of using $C(m)$ of rate $\frac{m+1}{2^m}$ we use its linear subcode $\hat{C}(m)$ of rate $\frac{m-1}{2^m}$, generated by the top $m - 1$ rows of $\mathbf{G_m}$ we are able to eliminate the identification problem while preserving the linearity of the code and suffering a very small loss in the overall rate.

5 Conclusion

In this report we proved several run-length structural properties of a Reed-Muller(1,m) code and we studied how to modify this code for use in channels in which sampling errors cause one bit repetition and one bit deletion per transmitted codeword. We enumerated all pairs of codewords that can result in the same string under this channel model, and based on this enumeration we provided a simple way of thinning the code to eliminate such codewords. The resulting code only has two fewer information bits than the original code and is also equipped with better synchronization error correction properties.

References

[1] L. Dolecek and V. Anantharam. ”Using Reed-Muller RM(1,m) Codes over Channels with Synchronization and Substitution Errors”. *submitted to IEEE Trans. Inform. Theory*, April 2006.

[2] F. J. MacWilliams and N. J. A. Sloane. *"The Theory Of Error Correcting Codes"*. North Holland Publishing Company, Amsterdam, Holland, 1977.