| | |
|---|---|
| Project | **IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>** |
| Title | **Revised Text for Reed-Solomon Encoding sections in TG1 Standard** |
| Date Submitted | **2000-11-07** |
| Source(s) | Karl Stambaugh<br>Motorola<br>8202 E. Roosevelt MS R1106<br>Scottsdale, AZ 85257<br>Voice: 480-441-7842<br>Fax: 480-675-2116<br>mailto:karl.stambaugh@motorola.com |
| Re: | 802.16.1-00/01r4 Air Interface for Fixed Broadband Wireless Access Systems |
| Abstract | This document contains revised text for sections 3.2.2.2.5 and 3.3.3.2.1 of 802161-00_01r4. These sections define the Reed-Solomon encoding procedure used in the Fixed Codeword and Shortened Last Codeword modes for the Burst Downstream (Mode B) and the Upstream cases. |
| Purpose | This text is proposed to replace the corresponding sections in the 802.16.1-00/01r4 document. |
| Notice |  |
| Release |  |
| Patent Policy and Procedures |  |

# Revised Text for Reed-Solomon Encoding sections in TG1 Standard

# Rev 1

*Karl Stambaugh*

*Motorola, Inc.*

This document contains revised text for the document 802.16.1-00/01r4 Air Interface for Fixed Broadband Wireless Access Systems. The affected sections define how the Reed-Solomon encoding shall be accomplished in the Fixed Codeword Mode and Shortened Last Codeword Mode, which was inadvertently omitted from the original draft. The number of Reed-Solomon parity bytes are now a function of the parameter ‘T’, which is defined within the MAC.

## Downstream Reed-Solomon Definition Text

Include the below text in the Downstream PMD section in place of the present 3.2.2.2.5 section (page 283 lines 5-49).

### 3.2.2.2.5 Reed Solomon Encoding (for code types 1-3)

The outer block code for code types 1-3 shall be a shortened, systematic Reed-Solomon code generated from GF(256) with information block lengths (K) variable from 6-255 bytes and error correction capability (T) able to correct from 1 to 16 byte errors. The specified code generator polynomials are given by:

**Code Generator Polynomial:** $g(x) = (x+\mu^0)(x+\mu^1)(x+\mu^2) \ldots (x+\mu^{2T-1})$, where $\mu = 02_{hex}$

**Field Generator Polynomial:** $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

The specified code has a block length of 255 bytes, and shall be configured as a RS(255,255-R) code with information bytes preceded by (255-N) zero symbols, where N is the codeword length and R the number of redundancy bytes R = 2*T = 2 to 32.

The value of K and T are specified for each burst type by the MAC. In the Fixed Codeword Mode, all Reed-Solomon codewords have the same number of information bytes (K), while in the Shortened Last Codeword Mode, the number of information bytes in the final codeword of each byte may be shortened as necessary to efficiently transport the MAC messages.

When using code type 2, the number of information bytes (K) shall always be an even number so that the total codeword size (K+R) is also always an even number.

### 3.2.2.2.5.1 Fixed Codeword Mode

In the Fixed Codeword Mode, the number of information bytes in each Reed-Solomon codeword is always the same (K). If the MAC messages in a burst require fewer bytes than are carried by an integral number of codewords, stuff-bytes ($FF_{hex}$) shall be added between MAC messages or after the last MAC message so that the total message length is an integral multiple of K bytes.

The SS determines the number of codewords in each burst from the Downstream MAP message, which defines the beginning point of each burst, and hence the length. Using this burst length, the SS calculates the number of full-length RS codewords that can be carried by each burst.

The process used by the BS to encode each burst is described below:

When the number of MAC message bytes (M) entering the FEC process is less than K bytes, the following operation is performed:

(A1) Add (K-M) stuff-bytes ($FF_{hex}$) to the M byte block as a suffix

(A2) Randomize the resulting K bytes

(A3) RS Encode the K bytes

(A4) Convert the byte block to a bit block

When the number of bytes (M) entering the FEC process is greater than or equal to K bytes, the following operation is performed:

(B1) Randomize the first K bytes

(B2) RS Encode the first K bytes

(B3) Subtract K from M (Let M=M-K)

(B4) If the new M is greater than or equal to K then repeat with the next set of bytes (go to B1),

If the new M is zero, then stop, otherwise go to step A1 above and process the M<K case.

**3.2.2.2.5.2 Shortened Last Codeword Mode**

In the Shortened Last Codeword Mode, the number of information bytes in the final Reed Solomon block of each burst is reduced from the normal number (K), while the number of parity bytes (R) remains the same. The Base Station tailors the number of information bytes in the last codeword in order to minimize the number of stuff-bytes to add to the end of the MAC message. The length of the burst is then set to the minimum number of Physical Slots required to transport all of the burst's bytes, which include preamble, information and parity bytes. The BS implicitly communicates the number of bytes in the shortened last codeword to the SS via the Downstream MAP message, which defines the starting PS of each burst. The SS uses the Downstream MAP information to calculate the number of full-length RS codewords and the length of the shortened last codeword that can be carried within the specified burst size.

To allow the receiving hardware to decode the previous Reed-Solomon codeword, no Reed-Solomon codeword shall have less than 6 information bytes. The number of information bytes carried by the shortened last codeword shall be between 6 and K bytes inclusive. If the number of information bytes needing to be sent by the BS is less than 6 bytes of data, stuff-bytes ($FF_{hex}$) shall be appended to the end of the data to bring the total number of information bytes up to the minimum of 6.

When using code type 2, the number of information bytes in the shortened last codeword shall always be an even number so that the total codeword size is also always an even number. If an odd number of information bytes needs to be sent, a stuff-byte ($FF_{hex}$) shall be appended to the end of the message to obtain an even number of bytes.

The process used by the BS to encode each burst is described below:

When the number of bytes (M) entering the FEC process is less than K bytes, perform the following operation:

(A1) If M<6 then append (6-M) stuff-bytes ($FF_{hex}$) to the message end. (Let M =6).

(A3) Apply Randomization to the new M bytes

(A4) Add (K-M) zero bytes to the M byte block as a prefix
(A5) Perform RS Encoding
(A6) Discard all of the (K-M) zero RS symbols
(A7) Convert the resulting byte block to bit block
(A8) Perform the Inner coding

When the number of bytes (M) entering the FEC process is greater than or equal to K bytes, perform the following operation:

(B1) Randomize the first K bytes
(B2) RS Encode the first K bytes
(B3) Subtract K from M (Let M=M-K)
(B4) If the new M is greater than or equal to K then repeat with the next set of bytes (go to B1),
If the new M is zero, then stop, otherwise go to step A1 above and process the M<K case.

## Upstream Reed-Solomon Definition Text

Include the below text in the Upstream PMD section in place of the present 3.3.3.2.1 section (page 299 line 41 to page 300 line 19). This textdefines the Upstream Reed-Solomon encoding process, which is similar to the Downstream encoding process defined above.

### 3.3.3.2.1 Reed Solomon Encoding (for code types 1-3)

The outer block code for code types 1-3 shall be a shortened, systematic Reed-Solomon code generated from GF(256) with information block lengths (K) variable from 6-255 bytes and error correction capability (T) able to correct from 1 to 16 byte errors. The specified code generator polynomials are given by:

**Code Generator Polynomial:** $g(x) = (x+\mu^0)(x+\mu^1)(x+\mu^2) \ldots (x+\mu^{2T-1})$, where $\mu = 02_{hex}$

**Field Generator Polynomial:** $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

The specified code has a block length of 255 bytes, and shall be configured as a RS(255,255-R) code with information bytes preceded by (255-N) zero symbols, where N is the codeword length and R the number of redundancy bytes R = 2*T = 2 to 32.

The value of K and T are specified for each burst type by the MAC. In the Fixed Codeword Mode, all Reed-Solomon codewords have the same number of information bytes (K), while in the Shortened Last Codeword Mode, the number of information bytes in the final codeword of each byte may be shortened as necessary to efficiently transport the MAC messages.

When using code type 2, the number of information bytes (K) shall always be an even number so that the total codeword size (K+R) is also always an even number.

#### 3.3.3.2.1.1 Fixed Codeword Mode

In the Fixed Codeword Mode, the number of information bytes in each codeword is always the same (K). If the MAC messages in a burst require fewer bytes than are carried by an integral number of Reed-Solomon codewords, stuffbytes ($FF_{hex}$) shall be added between MAC messages or after the last MAC message so that the total message length is an integral multiple of K bytes.

The SS determines the number of codewords in each burst from the Upstream MAP message, which defines the beginning point of each burst, and hence the length. Using this burst length, the SS calculates the number of full-length RS codewords that can be carried by each burst.

The process used by the SS to encode each burst is described below:

When the number of MAC message bytes (M) entering the FEC process is less than K bytes, the following operation is performed:

(A1) Add (K-M) stuff-bytes ($FF_{hex}$) to the M byte block as a suffix

(A2) Randomize the resulting K bytes

(A3) RS Encode the K bytes

(A4) Convert the byte block to a bit block

When the number of bytes (M) entering the FEC process is greater than or equal to K bytes, the following operation is performed:

(B1) Randomize the first K bytes

(B2) RS Encode the first K bytes

(B3) Subtract K from M (Let M=M-K)

(B4) If the new M is greater than or equal to K then repeat with the next set of bytes (go to B1),

If the new M is zero, then stop, otherwise go to step A1 above and process the M<K case.

**3.3.3.2.1.2 Shortened Last Codeword Mode**

In the Shortened Last Codeword Mode, the number of information bytes in the final Reed Solomon block of each burst is reduced from the normal number (K), while the number of parity bytes (R) remains the same. The SS tailors the number of information bytes in the last codeword in order to transport as many information bytes as possible in each upstream burst. The BS implicitly communicates the number of bytes in the shortened last codeword to the SS via the Upstream MAP message, which defines the starting PS of each burst. The SS uses the Upstream MAP information to calculate the number of full-length RS codewords and the length of the shortened last codeword that can be carried within the specified burst size. This calculation must take into account the number of bytes in the burst used for the preamble and coding bytes as well as the guard time.

To allow the receiving hardware to decode the previous Reed-Solomon codeword, no Reed-Solomon codeword shall have less than 6 information bytes. The number of information bytes carried by the shortened last codeword shall be between 6 and K bytes inclusive. In this mode, the BS shall only allocate bursts that result in shortened last codewords of the proper length.

When using code type 2, the number of information bytes in the shortened last codeword shall always be an even number so that the total codeword size is also always an even number. The SS shall take this into account when calculating the number of information bytes in the last codeword.

The process used by the BS to encode each burst is described below:

First, the full-sized Reed-Solomon codewords that precede the burst’s final codeword are encoded as in the Fixed Codeword Mode above. The number of bytes allocated for the shortened last codeword by the Upstream MAP is k’ bytes, which must be between 6 and K bytes. The remaining M bytes of the message is then encoded into these k’ bytes using the following procedure:

When the number of remaining bytes (M) is less than k’ bytes, perform the following operation:

(A1) If M<k’ then append (k’-M) stuff-bytes ($FF_{hex}$) to the message end.

(A3) Apply Randomization to the new M bytes

(A4) Add (K-k’) zero bytes to the M byte block as a prefix

(A5) Perform RS Encoding

(A6) Discard all of the (K-k’) zero RS symbols

(A7) Convert the resulting byte block to bit block

(A8) Perform the Inner coding