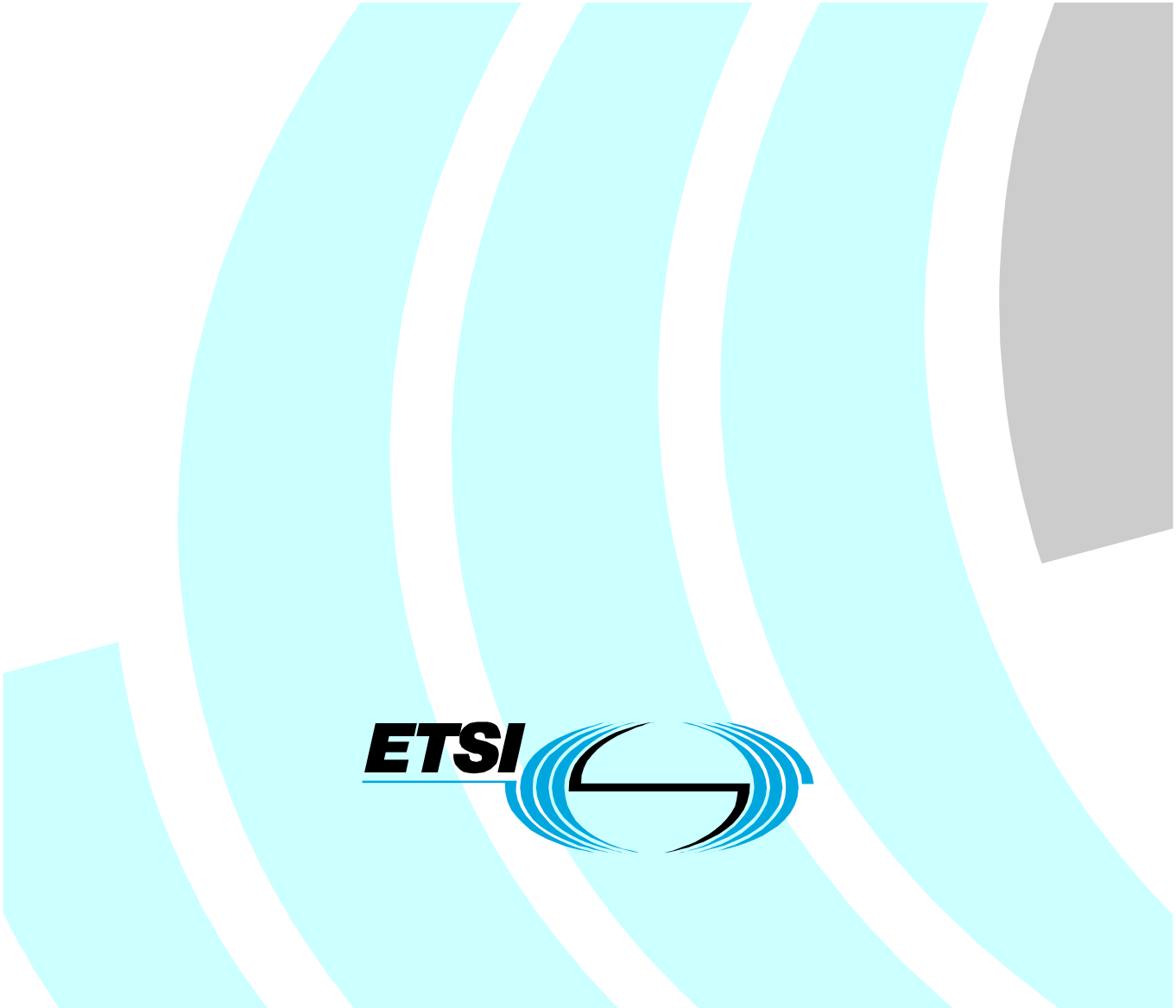
ETSI TS 102 188-3 V1.1.1 (2004-03)

Technical Specification

Satellite Earth Stations and Systems (SES);
Regenerative Satellite Mesh - A (RSM-A) air interface;
Physical layer specification;
Part 3: Channel coding



Reference
DTS/SES-00RSM-A-PHY-P3

Keywords
air interface, broadband, IP, multimedia, satellite

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, send your comment to:
editor@etsi.org

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

The present document is part 3 of a multi-part deliverable covering the BSM Regenerative Satellite Mesh - A (RSM-A) air interface; Physical Layer specifications, as identified below:

Part 1: "General description";

Part 2: "Frame structure";

Part 3: "Channel coding";

Part 4: "Modulation";

Part 5: "Radio transmission and reception";

Part 6: "Radio link control";

Part 7: "Synchronization".

1 Scope

The present document defines the channel coding structure used within the SES BSM Regenerative Satellite Mesh - A (RSM-A) air interface family. It includes code block, scrambling, outer forward error correction encoding, interleaving, and inner forward error correction encoding process definition.

2 References

Void.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Network Operations Control Centre (NOCC): centre that controls the access of the satellite terminal to an IP network and also provides element management functions and control of the address resolution and resource management functionality

satellite payload: part of the satellite that provides air interface functions

NOTE: The satellite payload operates as a packet switch that provides direct unicast and multicast communication between STs at the link layer.

Satellite Terminal (ST): terminal installed in the user premises

terrestrial host: entity on which application level programs are running

NOTE: It may be connected directly to the Satellite Terminal or through one or more networks.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

FEC	Forward Error Correction
IP	Internet Protocol
LSB	Least Significant Bit
MSB	Most Significant Bit
NOCC	Network Operations Control Centre
PHY	PHYsical
PTP	Point-to-Point
RS	Reed-Solomon
RSM	Regenerative Satellite Mesh
SLC	Satellite Link Control
ST	Satellite Terminal
TDMA	Time Division Multiple Access

4 General

The functions of the physical layer are different for the uplink and downlink. The major functions are illustrated in figure 4.

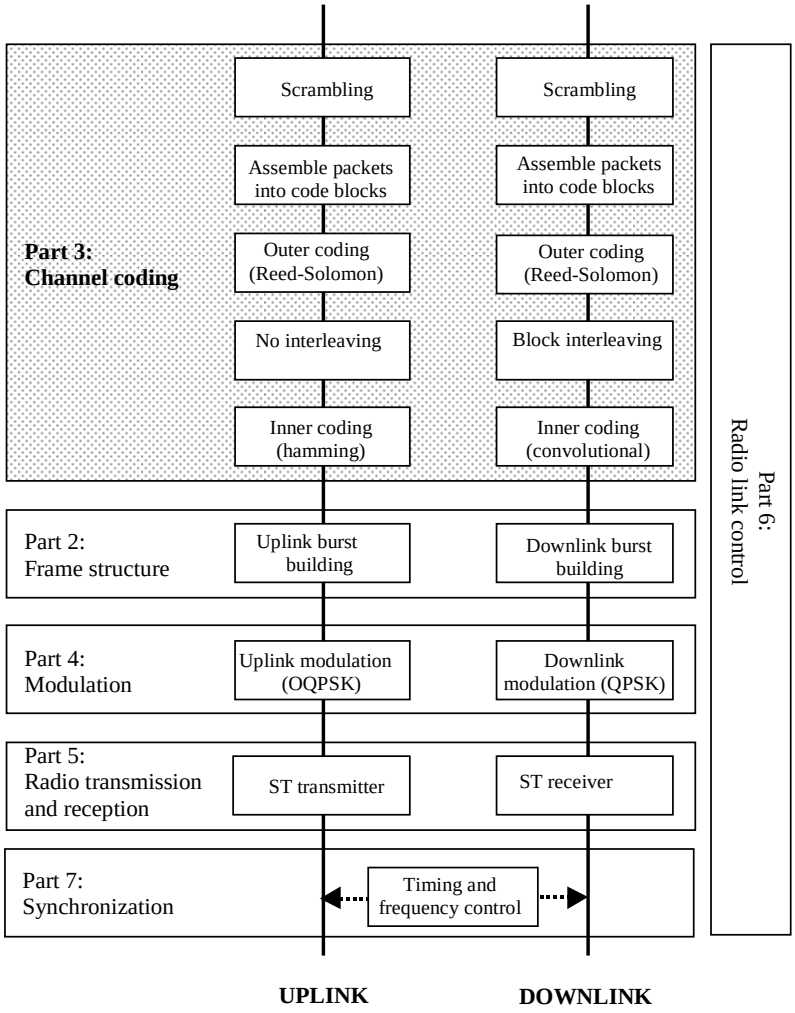


Figure 4: Physical layer functions

The present document describes the channel coding functions - this group of functions is highlighted in figure 4.

The uplink channel coding is described in clause 5 and the downlink channel coding is described in clause 6.

5 Uplink

5.1 Uplink code block structure

Uplink code blocks are the basic unit in the formation of an uplink TDMA burst. The number of code blocks constituting an uplink burst depends on the carrier mode. Uplink code blocks are formed with a set of user data packets and an access control field that have been processed with FEC to achieve acceptable packet error rates.

Uplink code blocks are generated as shown in figure 5.1. This is described in two stages:

- Assembly of an uncoded block containing two user data packets plus an access control field.
- Forward Error Correction coding.

The FEC on the uplink uses a set of two concatenated error correction codes, with no interleaving in between the codes. The outer code consists of a t=12 symbol error correcting (244,220) Reed-Solomon code followed by an inner shortened Hamming (12,8) block code.

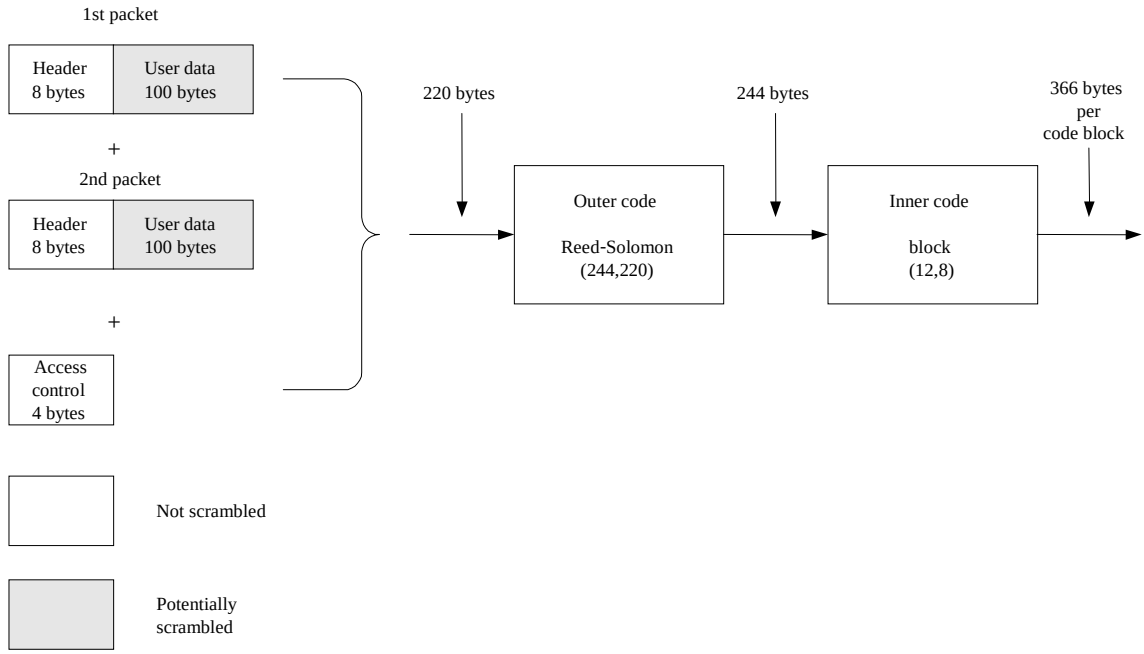


Figure 5.1: Uplink code block generation

5.2 Uplink data scrambling

The ST shall scramble the information payload field of all packets (i.e. byte 8 through byte 107 of a 108-byte packet) except those destined only to the satellite, as defined in table 5.2.

The destination type is specified in the destination type sub-field of the header satellite routing field as described in TS 102 189-2. The scrambling is performed on a packet-by-packet basis. Scrambling starts and stops at the beginning and ending of the information payload field, respectively.

Table 5.2: Scrambling according to packet type

Destination Type	Scrambling
Null packets	Scrambled
PTP or shaped-broadcast packets	Scrambled
Packet replication packets	Scrambled
Satellite terminated packets (except null packets)	Not scrambled

The scrambling sequence is generated by a LFSR with connection polynomial:

$$h(X) = 1 + X + X^{15}$$

as illustrated in figure 5.2, where the adders perform modulo-2 arithmetic. The scrambler is initialized at the beginning of every packet. The initial sequence is given by 110100101011001 (X_0 …. X_{14}).

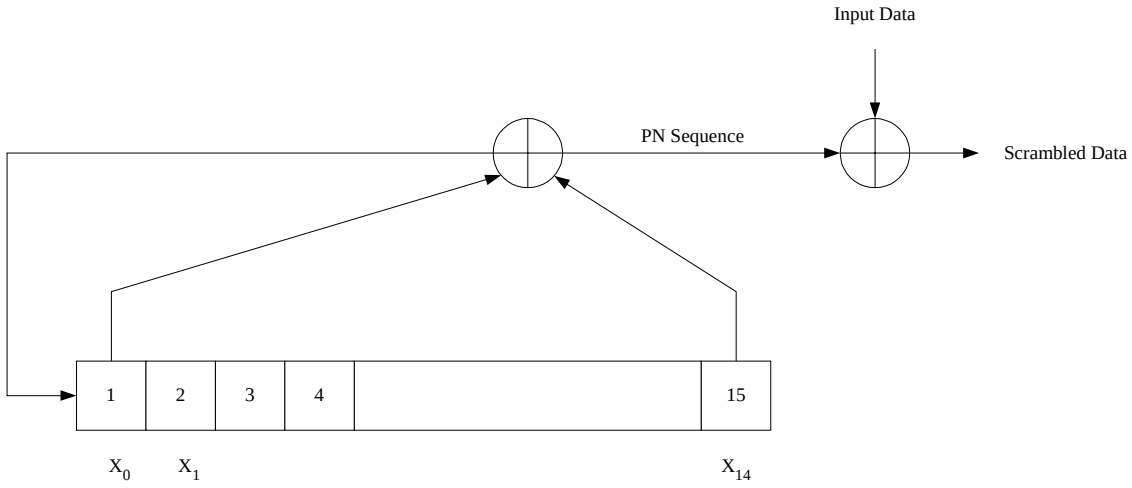


Figure 5.2: Uplink data scrambler

5.3 Uplink Forward Error Correction processing

In order to achieve acceptable packet error rates, a concatenated outer and inner coding scheme is used on each uplink code block. The error correcting codes are both block codes. The outer code is a 12-symbol error correcting Reed-Solomon (RS) code, and the inner code is a one-bit error correcting binary code. The system does not use interleaving between the uplink outer code and the inner code. The FEC order of processing is encoding with the outer code followed by the inner code.

5.3.1 Uplink outer code

The ST encodes two uplink packets and the access control byte data using a Reed-Solomon systematic block code with 24-byte Reed-Solomon parity check field, as shown in figure 5.3.1.1.

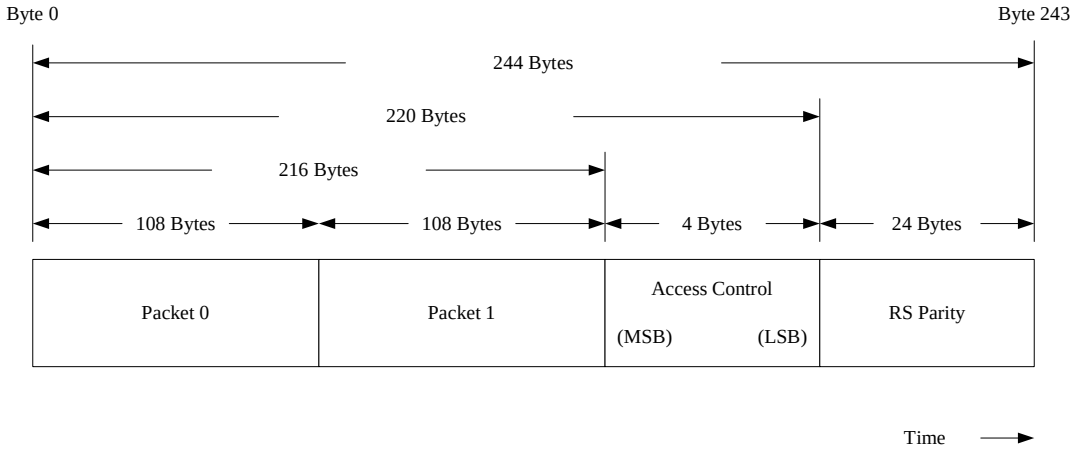


Figure 5.3.1.1: Uplink outer code word

The arrangement of each packet within a Reed-Solomon code word is by increasing byte number (0, 1, 2, ..., 219), and within each byte, the order of the bits is MSB first as shown in figure 5.3.1.2.

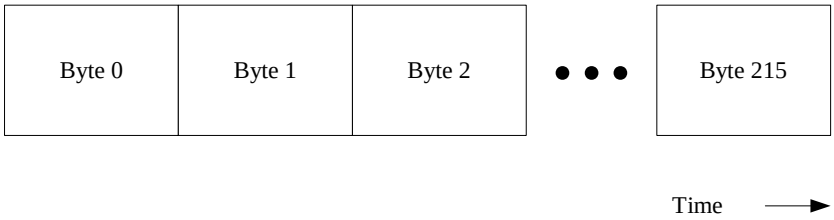


Figure 5.3.1.2: Packets order of presentation to outer code encoder

The uplink Reed-Solomon code is a systematic block code where each code word has 220 information symbols followed by 24-byte parity symbols. The resulting RS code is a (244,220) code. Each symbol is an element of a GF(2^8) field. Thus, each symbol is made up of one byte or eight bits. The symbols for each code word are derived as described in the following operations:

Let:

M(x) = a polynomial of degree less than 220, where the coefficients are the symbols represented by each byte of the two user data packets and the access control field. The highest degree coefficient is taken from byte 0 of user data packet 0. The next coefficient is taken from byte 1, and so on, until the 0-degree coefficient is taken from byte 219. The value of the coefficients of the polynomial M(X) are represented by the respective value of each of the 220 bytes, interpreted as elements of a GF(2^8) field.

G(X) = generator polynomial for the code. The generator polynomial G(X) is defined to be a monic polynomial of degree 24 with coefficients in a GF(2^8) field as defined in table 5.3.1.

Table 5.3.1: Generator function coefficients

Index, decimal	Coefficient in GF(2^8) (8-tuple)								Exponent of coefficient term, decimal
	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	
0	1	0	0	0	0	0	1	1	45
1	0	0	1	1	0	1	1	0	250
2	1	1	1	0	0	0	1	1	118
3	0	0	0	0	1	0	1	1	108
4	1	0	1	1	0	1	0	1	252
5	1	1	1	1	0	0	1	0	136
6	1	0	1	1	0	1	0	0	18
7	1	0	1	0	0	0	0	1	128
8	1	1	0	1	1	1	1	1	234
9	1	0	1	1	1	1	1	0	243
10	0	0	1	1	0	1	0	0	240
11	1	1	1	0	0	1	0	1	205
12	0	1	1	0	0	0	1	1	164
13	0	1	1	0	1	0	0	1	180
14	0	1	1	1	0	1	0	1	190
15	0	0	1	1	1	1	1	1	168
16	0	1	0	1	1	0	1	1	134
17	0	0	0	1	0	0	0	0	3
18	1	0	1	0	0	0	1	1	123
19	1	1	0	0	0	0	1	1	216
20	0	0	1	0	1	0	0	0	52
21	1	0	0	0	0	1	0	0	138
22	1	0	1	0	0	0	1	1	123
23	0	0	1	0	1	1	1	1	230
24	1	0	0	0	0	0	0	0	0

NOTE: *G(X)* contains as roots α^n where α is the primitive field element and n is an integer in the range from 0 to 24.

P(X) = a polynomial of degree less than or equal to 23, where the coefficients are the parity symbols. The order of transmission for the parity symbols is as follows: the coefficient for the term of degree 23 of P(X) is transmitted first, followed by the coefficient of the degree 22 term and so on, ending with the coefficient associated with the 0 degree term.

The parity polynomial P(X) is formed by computing the remainder of the shifted information polynomial M(X) with respect to a generator polynomial G(X) of degree 24. All operations are performed using the arithmetic of GF(2^8). The version of GF(2^8) used has as a primitive element a root α of the (binary) polynomial $f(X) = X^8 + X^4 + X^3 + X^2 + 1$, or in octal 435, where the high-order coefficient is to the left.

C(X) = a polynomial of degree less than 244, where the coefficients are the transmitted symbols for the code word. The order of transmission for the code word symbols (polynomial coefficients) is by decreasing exponent value.

$$C(X) = M(X) \bullet X^{24} + P(X)$$

where

$$P(X) = \left[M(X) \bullet X^{24}\right] \text{Modulo } G(X)$$

$$G(X) = \prod_{i=1}^{24} \left(X - \alpha^i\right)$$

$$\alpha \quad = \text{a root of } X^8 + X^4 + X^3 + X^2 + 1 = 0 \text{ in } GF\left(2^8\right)$$

That is, the outer code word is structured as a polynomial C(X) made up of a shifted (by 24 positions) information polynomial M(X) and a parity polynomial P(X), with all coefficients being treated as elements of GF(2^8).

5.3.2 Uplink block interleaving

There is no block interleaving requirement for the uplink.

5.3.3 Uplink inner code

Following the outer code encoding, the inner code encoder takes each symbol of the Reed-Solomon outer code code-word as the information source i(X). The ST uses a shortened Hamming inner code consisting of a systematic (12,8) binary block code that expands each 8-bit RS symbol to a 12-bit inner code word. Each inner code word includes a 4-bit parity field appended to each symbol of the outer code words as depicted in figure 5.3.3.

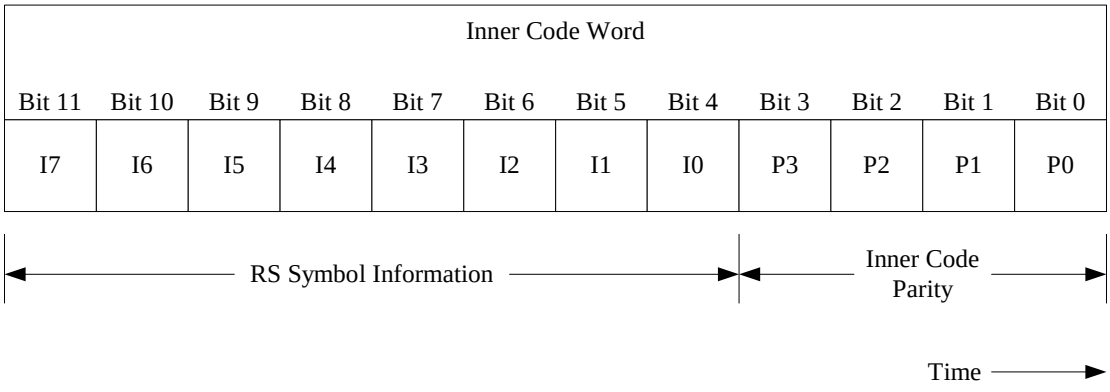


Figure 5.3.3: Uplink inner code word format

The four parity bits of the inner code are formulated in accordance with the following equations:

$$p_3 = i_1 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_7$$
$$p_2 = i_0 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_7$$
$$p_1 = i_0 \oplus i_2 \oplus i_3 \oplus i_6 \oplus i_7$$
$$p_0 = i_0 \oplus i_1 \oplus i_3 \oplus i_5 \oplus i_7$$

In the equations above the sign $\oplus$ is to be interpreted as addition modulo 2.

As it appears in a TDMA burst, an uplink code block with user packets consists of a set of 244 inner code words, each with 12 coded bits. Each 12-bit inner code word is divided into two strings of six binary symbols each and applied independently to the I and the Q arms of the uplink modulator.

A symbol consists of a pair of bits (I, Q). The ST order of presentation for the binary symbols (coded bits) of the inner code words to the modulated symbols is as defined in table 5.3.3. That is, each inner code word is associated with six modulated symbols, with the odd and even indexed coded bits going to the I and Q arms respectively, and with coded bit indices decreasing as time advances.

Table 5.3.3: Inner code word coded bits to modulated symbol transmission order

Arm	Coded bit	Coded bit	Coded bit	Coded bit	Coded bit	Coded bit
I	11	9	7	5	3	1
Q	10	8	6	4	2	0
	Sent first					**Sent last**

The STs transmit a code block in sequential order starting with byte 0 to 243.

6 Downlink

6.1 Downlink code block structure

Downlink code blocks are the basic unit in the formation of a downlink Shaped-Broadcast or PTP burst. There are six interleaved downlink code blocks per downlink burst. Downlink code blocks are formed with a set of twelve scrambled packets that have been processed with FEC to achieve acceptable packet error rates.

Downlink code blocks are generated as shown in figure 6.1. This is described in two stages:

- Assembly of an uncoded block containing user data packets.
- Forward Error Correction coding.

The FEC on the downlink uses a set of two concatenated error correction codes, with interleaving in between the codes. The outer code consists of a t=10 symbol error correcting (236,216) Reed-Solomon code. Following the outer code, the RS code words are 6-way block interleaved. The output of the interleaver is then processed with a set of convolutional encoders.

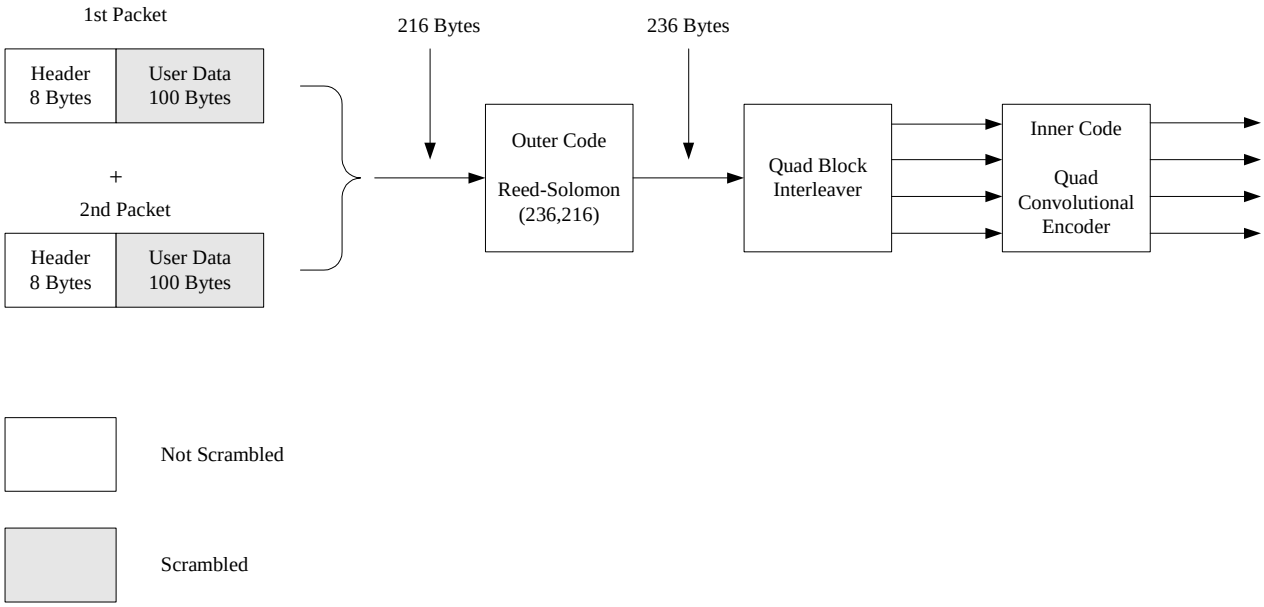


Figure 6.1: Downlink code block generation

6.2 Downlink data scrambling

The information payload portion of all packets (i.e. byte 8 through byte 107 of a 108-byte packet), is scrambled. The packet header portion (byte 0 through byte 7) is not scrambled.

The packet type is specified in the destination type and destination sub-address subfields as described TS 102 189-2. The scrambling is performed on a packet-by-packet basis. Scrambling starts and stops at the beginning and ending of the information payload, respectively.

The downlink uses the same scrambling algorithm as the uplink. The scrambling sequence is generated by a LFSR with connection polynomial:

$$h(X) = 1 + X + X^{15}$$

as illustrated in figure 6.2, where the adders perform modulo-2 arithmetic. The scrambler is initialized at the beginning of every packet. The initial sequence is given by 110100101011001 (X_0 …. X_{14}).

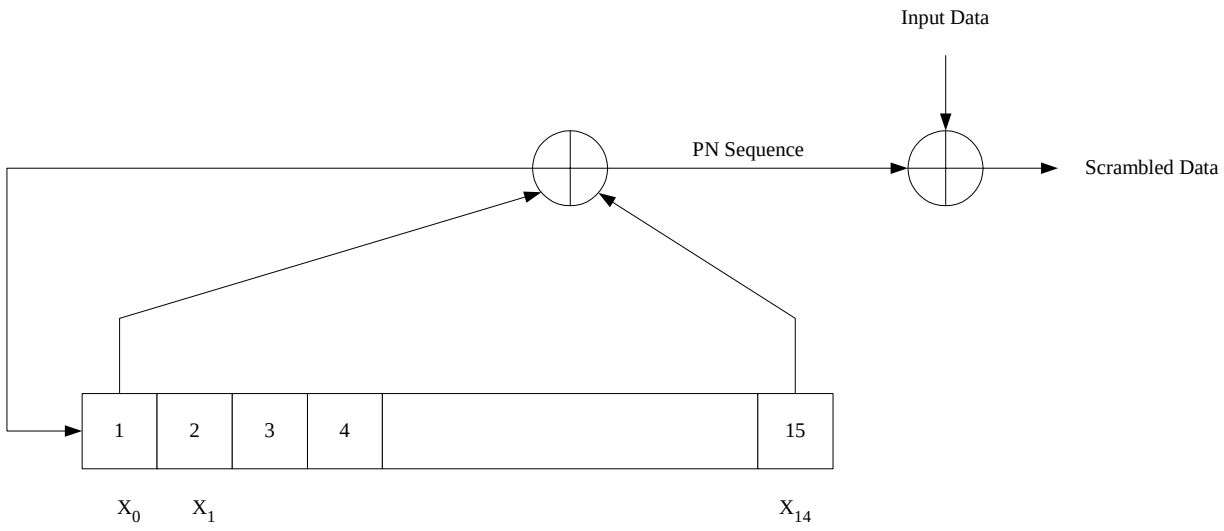


Figure 6.2: Downlink data scrambler

6.3 Downlink forward error correction processing

In order to achieve acceptable packet error rates, a concatenated outer and inner coding scheme is used on each downlink code block. The outer code is a 10-symbol error correcting Reed-Solomon (RS) code, and the inner code is a convolutional R2/3 code. The system uses 6-way interleaving between the uplink outer code and the inner code. The FEC order of processing is encoding with the outer code followed by the inner code.

6.3.1 Downlink outer code

Two downlink packets are encoded using a Reed-Solomon systematic block code with 20-byte Reed-Solomon parity check field, as shown in figure 6.3.1.1.

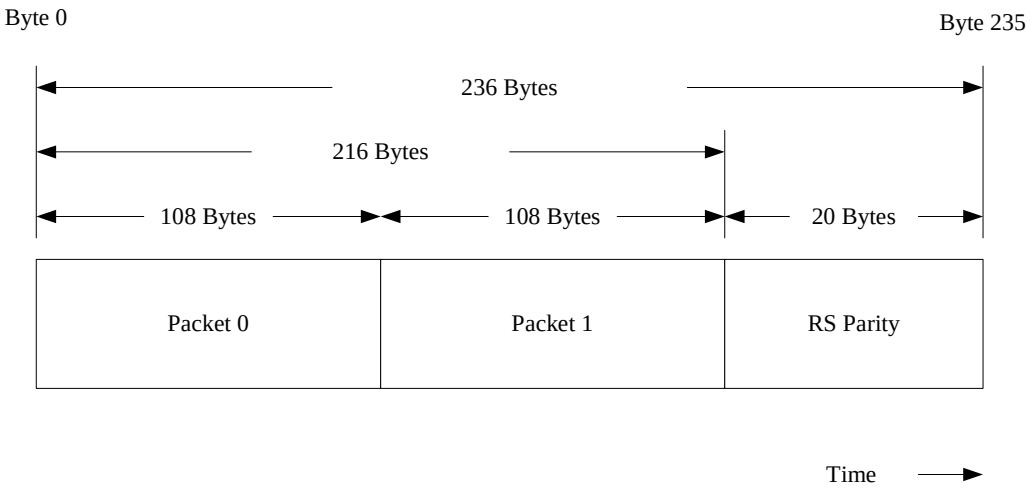


Figure 6.3.1.1: Downlink Read-Solomon code word format

The arrangement of each packet within a Reed-Solomon code word is by increasing byte number (0, 1, 2, ..., 215), and within each byte, the order of the bits is MSB first as shown in figure 6.3.1.2.

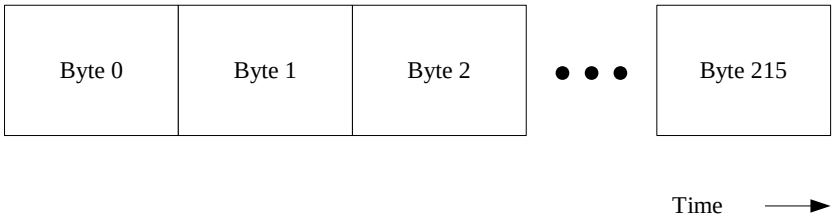


Figure 6.3.1.2: Packets order of presentation to outer code encoder

The downlink Reed-Solomon code is a systematic block code where each code word has 216 information symbols followed by 20-byte parity symbols. The resulting RS code is a (236,216) code. Each symbol is an element of a GF(2^8) field. Thus, each symbol is made up of one byte or eight bits. The symbols for each code word are derived as described in the following operations:

Let:

M(x) = a polynomial of degree less than 216, where the coefficients are the symbols represented by each byte of the two user data packets. The highest degree coefficient is taken from byte 0 of user data packet 0. The next coefficient is taken from byte 1, and so on, until the 0-degree coefficient is taken from byte 215. The value of the coefficients of the polynomial M(X) are represented by the respective value of each of the 216 bytes, interpreted as elements of a GF(2^8) field.

G(X) = generator polynomial for the code. The generator polynomial G(X) is defined to be a monic polynomial of degree 20 with coefficients in a GF(2^8) field as defined in table 6.3.1.

Table 6.3.1: Generator function coefficients

Index, decimal	Coefficient in GF(2^8) (8-tuple) α^0 α^1 α^2 α^3 α^4 α^5 α^6 α^7	Exponent of coefficient term, decimal
0	1 0 0 1 1 0 1 0	146
1	0 1 1 0 0 1 0 1	72
2	0 0 1 0 1 1 1 1	69
3	0 1 0 1 1 1 1 0	70
4	0 1 1 0 1 0 0 1	58
5	1 1 0 0 1 1 0 1	12
6	1 1 1 0 0 0 1 0	95
7	1 0 0 1 1 0 1 1	217
8	1 1 0 1 0 0 0 1	161
9	1 1 1 0 1 1 1 1	215
10	0 1 1 1 0 1 0 1	21
11	0 1 0 0 1 1 0 1	145
12	0 0 1 0 0 1 1 0	15
13	1 1 1 0 0 1 0 0	156
14	1 0 1 0 0 1 1 1	205
15	1 0 0 0 0 1 1 0	99
16	0 0 0 0 1 0 1 0	51
17	1 1 0 0 1 0 1 1	236
18	0 1 1 1 1 0 1 1	172
19	1 0 1 1 0 1 0 0	20
20	1 0 0 0 0 0 0 0	0
NOTE: G*(X)* contains as roots α^n where α is the primitive field element and n is an integer in the range from 0 to 20.		

P(X) = a polynomial of degree less than or equal to 19, where the coefficients are the parity symbols. The order of transmission for the parity symbols is as follows: The coefficient for the term of degree 19 of P(X) is transmitted first, followed by the coefficient of the degree 18 term and so on, ending with the coefficient associated with the 0 degree term.

The parity polynomial P(X) is formed by computing the remainder of the shifted information polynomial M(X) with respect to a generator polynomial G(X) of degree 20. All operations are performed using the arithmetic of GF(2^8). The version of GF(2^8) used has as a primitive element a root α of the (binary) polynomial $f(X) = X^8 + X^4 + X^3 + X^2 + 1$, or in octal 435, where the high-order coefficient is to the left.

C(X) = a polynomial of degree less than 236, where the coefficients are the transmitted symbols for the code word. The order of transmission for the code word symbols (polynomial coefficients) is by decreasing exponent value.

$$C(X) = M(X) \bullet X^{20} + P(X)$$

where

$$P(X) = \left[M(X) \bullet X^{20}\right] \text{Modulo } G(X)$$

$$G(X) = \prod_{i=1}^{20} \left(X - \alpha^i\right)$$

$$\alpha \quad = \text{a root of } X^8 + X^4 + X^3 + X^2 + 1 = 0 \text{ in } GF\left(2^8\right)$$

That is, the outer code word is structured as a polynomial C(X) made up of a shifted (by 20 positions) information polynomial M(X) and a parity polynomial P(X), with all coefficients being treated as elements of GF(2^8).

6.3.2 Downlink block interleaving

The downlink implements a block-interleaving algorithm. Downlink block interleaving for each downlink code block is best described in terms of writing and reading the RS code words into a two dimensional (6 × 236)-byte element array, as shown in table 6.3.2.1.

When writing data into the array, the contents of each row are filled with the 236 symbols of an outer code (RS) code word. Each symbol of the RS code word occupies one-byte element of each row of the array.

The rows of the array are referenced as A through F. The order in which RS code words are written into the rows is sequential, starting with row A and ending with row F.

Table 6.3.2.1: Data input order into block interleaver

Input row	Byte column 0	Byte column 1	Byte column 2		Byte column 233	Byte column 234	Byte column 235
A	A_0	A_1	A_2	• • •	A_{233}	A_{234}	A_{235}
B	B_0	B_1	B_2	• • •	B_{233}	B_{234}	B_{235}
C	C_0	C_1	C_2	• • •	C_{233}	C_{234}	C_{235}
D	D_0	D_1	D_2	• • •	D_{233}	D_{234}	D_{235}
E	E_0	E_1	E_2	• • •	E_{233}	E_{234}	E_{235}
F	F_0	F_1	F_2	• • •	F_{233}	F_{234}	F_{235}

The outputs of the array are divided into four independent streams. These output streams are referred as streams 0 through 3 as shown in table 6.3.2.2. Each output stream consists of a total of 354-byte elements and six 0 bits (flush bits).

Table 6.3.2.2: Data output order of block interleaver

Output stream number	Output order of array byte elements
0 (I_0)	$A_0\ B_0\ C_0\ D_0\ E_0\ F_0\ A_4\ B_4\ C_4\ D_4\ E_4\ F_4\ A_8\ B_8\ C_8\ D_8\ E_8\ F_8 \ldots A_{232}\ B_{232}\ C_{232}\ D_{232}\ E_{232}\ F_{232}$ 0 (see note)
1 (I_1)	$A_1\ B_1\ C_1\ D_1\ E_1\ F_1\ A_5\ B_5\ C_5\ D_5\ E_5\ F_5\ A_9\ B_9\ C_9\ D_9\ E_9\ F_9 \ldots A_{233}\ B_{233}\ C_{233}\ D_{233}\ E_{233}\ F_{233}$ 0 (see note)
2 (Q_0)	$A_2\ B_2\ C_2\ D_2\ E_2\ F_2\ A_6\ B_6\ C_6\ D_6\ E_6\ F_6\ A_{10}\ B_{10}\ C_{10}\ D_{10}\ E_{10}\ F_{10} \ldots A_{234}\ B_{234}\ C_{234}\ D_{234}\ E_{234}\ F_{234}$ 0 (see note)
3 (Q_1)	$A_3\ B_3\ C_3\ D_3\ E_3\ F_3\ A_7\ B_7\ C_7\ D_7\ E_7\ F_7\ A_{11}\ B_{11}\ C_{11}\ D_{11}\ E_{11}\ F_{11} \ldots A_{235}\ B_{235}\ C_{235}\ D_{235}\ E_{235}\ F_{235}$ 0 (see note)
NOTE: Represents 6 bits of 0 (flush bits).	

6.3.3 Downlink inner code

The bit order of presentation of each byte element output of the block interleaver into the input serial stream of the convolutional encoder is the MSB first and the LSB last as shown in figure 6.3.3.1.

The input of each of the four convolutional encoders consists of one of the block interleaver output stream (with six zeros appended at the end of the stream to flush the convolutional encoder). After the encoder flush has been appended, each of the four interleaver output streams are composed of 59 set columns (6 bytes per column) followed by six bits for encoder, for a total of 2 838 bits (59 × 6 × 8 + 6 = 2 838).

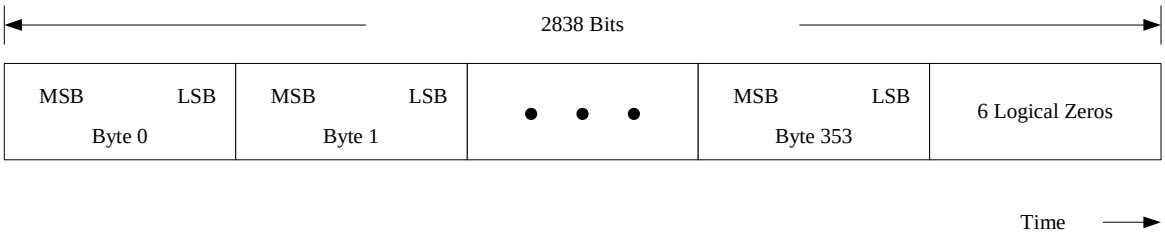


Figure 6.3.3.1: Order of presentation for inner code encoder serial input

The downlink uses a convolutional inner code with rate 2/3 code generated from a punctured convolutional code of rate 1/2 and constraint length 7. The encoder taps for R = 1/2 code are shown figure 6.3.3.2, in octal representation:

$$G_0 = (171)_8$$

$$G_1 = (133)_8$$

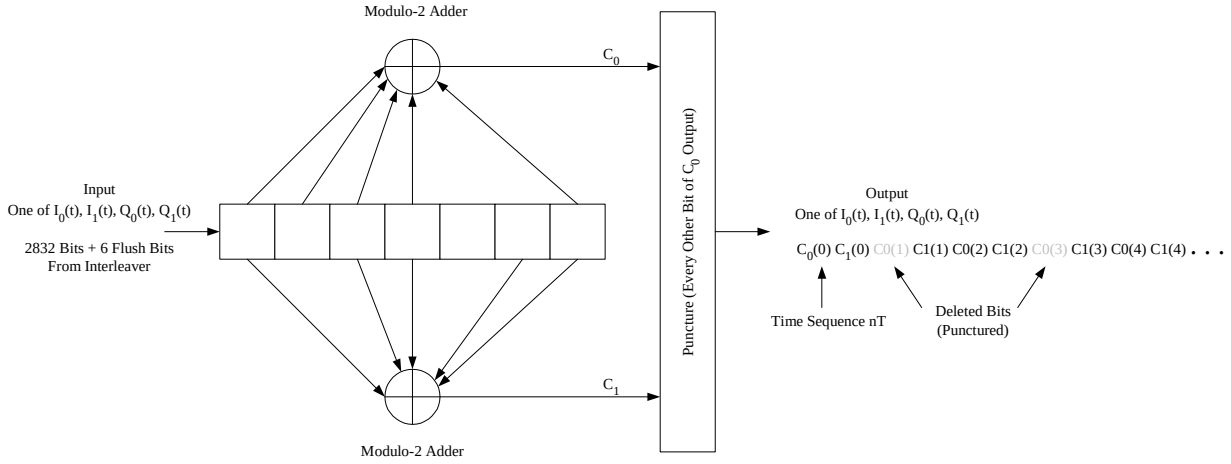


Figure 6.3.3.2: Convolutional encoder structure

Systematic puncturing or deletion of some of the output bits of the rate 1/2 convolutional code results in the generation of a rate 2/3 convolutional code with K = 7. Puncturing is to be accomplished by deleting every other bit of the C0 encoder output, starting with the second C0 output bit. Thus, for every two input bits, there are only three output bits to be transmitted, and thus the effective code rate is 2/3. The punctured output sequence of the encoder is the following:

$$I_0(t), I_1(t), Q_0(t), Q_1(t) = \; C_0(0), C_1(0), C_1(1), C_0(2), C_1(2), C_1(3), C_0(4), C_1(4), \cdots,$$
$$C_0(2830), C_1(2830), C_1(2831), 9 \text{ coded flush bits}$$

Prior to any input bits, the seven registers are initialized to the binary zero state.

After encoding each block interleaver output stream, the 2 838 bits (354 bytes plus 6 flush bits) at the input of the encoder becomes 4 257 bits at the output of the encoder.

The output of the two I inner encoders (i.e. I0(t), I1(t)) are combined into one I(t) stream by alternating between I0(t) and I1(t) where we start with I0(t) as shown in figure 6.3.3.3.

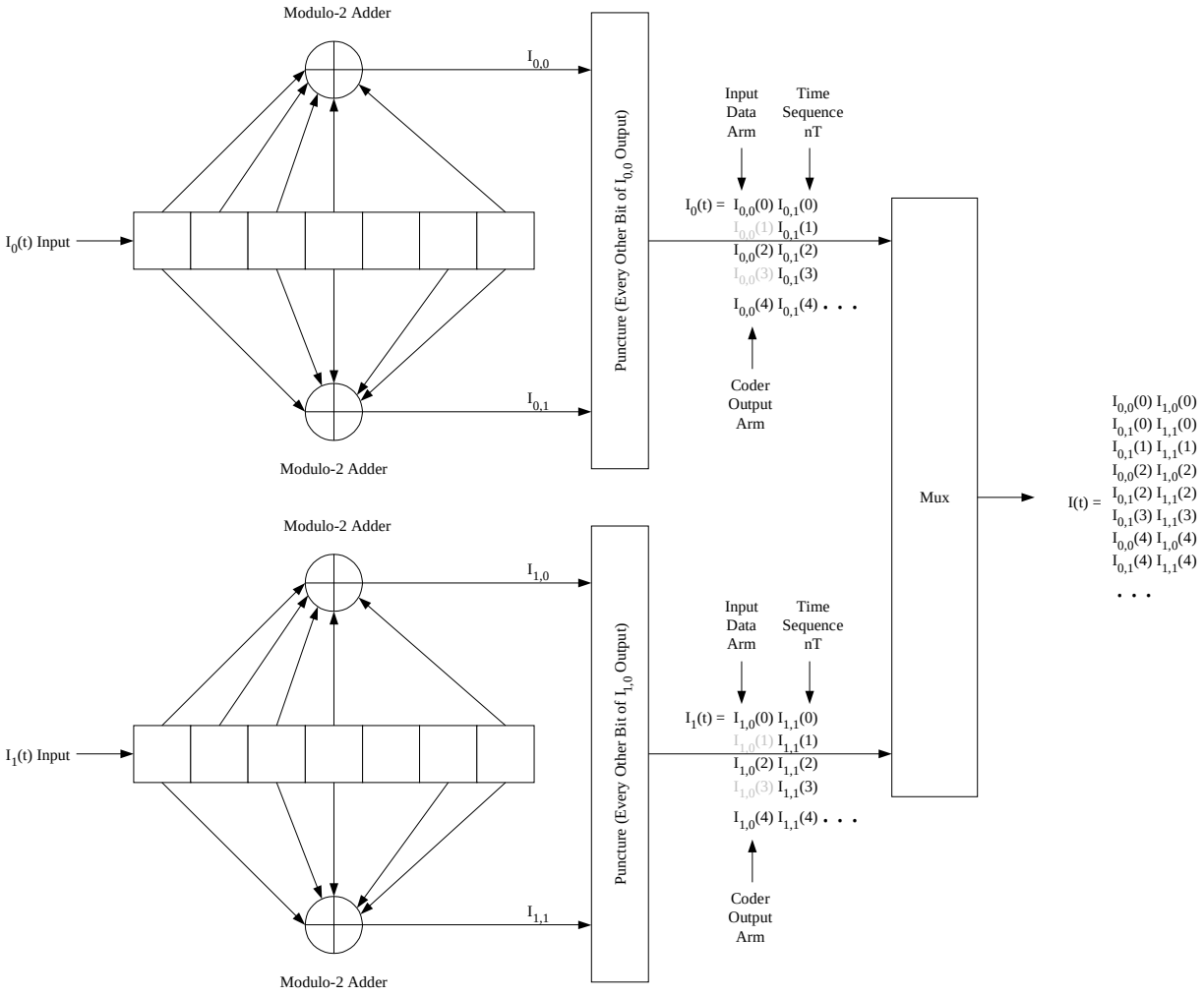


Figure 6.3.3.3: Combining convolutional codes into one data stream of the in-phase(I) arm

The same applies for the Q arm, where 'I' is replaced by 'Q' in figure 6.3.3.3. The following is the output of each parallel convolutional encoder being combined to a single stream:

$$I(t) = I_{o,o}(0), I_{1,0}(0), I_{0,1}(0), I_{1,1}(0), I_{0,1}(1), I_{1,1}(1), I_{o,o}(2), I_{1,0}(2), I_{0,1}(2), I_{1,1}(2), I_{0,1}(3), I_{1,1}(3), \cdots$$

$$Q(t) = Q_{o,o}(0), Q_{1,0}(0), Q_{0,1}(0), Q_{1,1}(0), Q_{0,1}(1), Q_{1,1}(1), Q_{o,o}(2), Q_{1,0}(2), Q_{0,1}(2), Q_{1,1}(2), I_{0,1}(3), Q_{1,1}(3), \cdots$$

where the first index refers either to arm or I in the arm case and or in the arm case and the second index refers to the order of the coded bit in that particular arm. For example:

$$\begin{aligned} I_{o,o}(0) &= \text{arm } I_0, \text{coded bit } C_0(0) \\ I_{1,o}(0) &= \text{arm } I_1, \text{coded bit } C_0(0) \\ I_{o,1}(0) &= \text{arm } I_0, \text{coded bit } C_1(0) \\ I_{1,1}(0) &= \text{arm } I_1, \text{coded bit } C_1(0) \\ I_{o,1}(1) &= \text{arm } I_0, \text{coded bit } C_1(1)\ \ldots \end{aligned}$$

and

$$\begin{aligned} Q_{o,o}(0) &= \text{arm } Q_0, \text{coded bit } C_0(0) \\ Q_{1,o}(0) &= \text{arm } Q_1, \text{coded bit } C_0(0) \\ Q_{o,1}(0) &= \text{arm } Q_0, \text{coded bit } C_1(0) \\ Q_{1,1}(0) &= \text{arm } Q_1, \text{coded bit } C_1(0) \\ Q_{o,1}(1) &= \text{arm } Q_0, \text{coded bit } C_1(1)\ \ldots \end{aligned}$$

Annex A (informative): Bibliography

ETSI TR 101 984: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; Services and Architectures".

ETSI TS 102 188-1: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface; Physical Layer specification; Part 1: General description".

ETSI TS 102 188-2: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface; Physical Layer specification; Part 2: Frame structure".

ETSI TS 102 188-4: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface; Physical Layer specification; Part 4: Modulation".

ETSI TS 102 188-5: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface; Physical Layer specification; Part 5: Radio transmission and reception".

ETSI TS 102 188-6: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface; Physical Layer specification; Part 6: Radio link control".

ETSI TS 102 188-7: "Satellite Earth Stations and Systems (SES); RSM-A Air Interface Physical Layer specification; Part 7: Synchronization".

ETSI TS 102 189-2: "Satellite Earth Stations and Systems (SES); Regenerative Satellite Mesh - A (RSM-A) air interface; MAC/SLC layer specification; Part 2: MAC layer".

History

Document history		
V1.1.1	March 2004	Publication