

3GPP2/3GPP Coding

Turbo code and Turbo Interleaver Summary

Jungsub Byun

CDMA-2000 3GPP2 C.S0002-C Version 2.0 Physical Layer Standard for cdma2000 Spread Spectrum Systems Revision C - Page from 197 to 207

2.1.3.1.4.1 Convolutional Encoding

2.1.3.1.4.2 Turbo Encoding

2.1.3.1.4.2 Turbo Encoding

Figure 2.1.3.1.4.2.1.1. Turbo Encoder

2.1.3.1.4.2.2 Turbo Code Termination

2.1.3.1.4.2.3 Turbo Interleavers

Figure 2.1.3.1.4.2.3-1. Turbo Interleaver output Address Calculation Procedure

Table 2.1.3.4.2.3-1. Turbo Interleaver Parameter

W-CDMA 3GPP TS 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD) - Page from 15 to 21

4.2.3 Channel coding

4.2.3.1 Convolutional coding

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

Figure 4: Structure of rate 1/3 Turbo coder

4.2.3.2.3 Turbo code internal interleaver

Table 2: List of prime number p and associated primitive root v

Table 3: Inter-row permutation patterns for Turbo code internal interleaver



CDMA-2000 3GPP2 C.S0002-C Version 2.0 Date: July 23, 2004

Physical Layer Standard for cdma2000 Spread Spectrum Systems
Revision C

2.1.3.1.4.1 Convolutional Encoding

All convolutional codes shall have a constraint length of 9.

Convolutional encoding involves the modulo-2 addition of selected taps of a serially time-delayed data sequence. The length of the data sequence delay is equal to $K-1$, where K is the constraint length of the code.

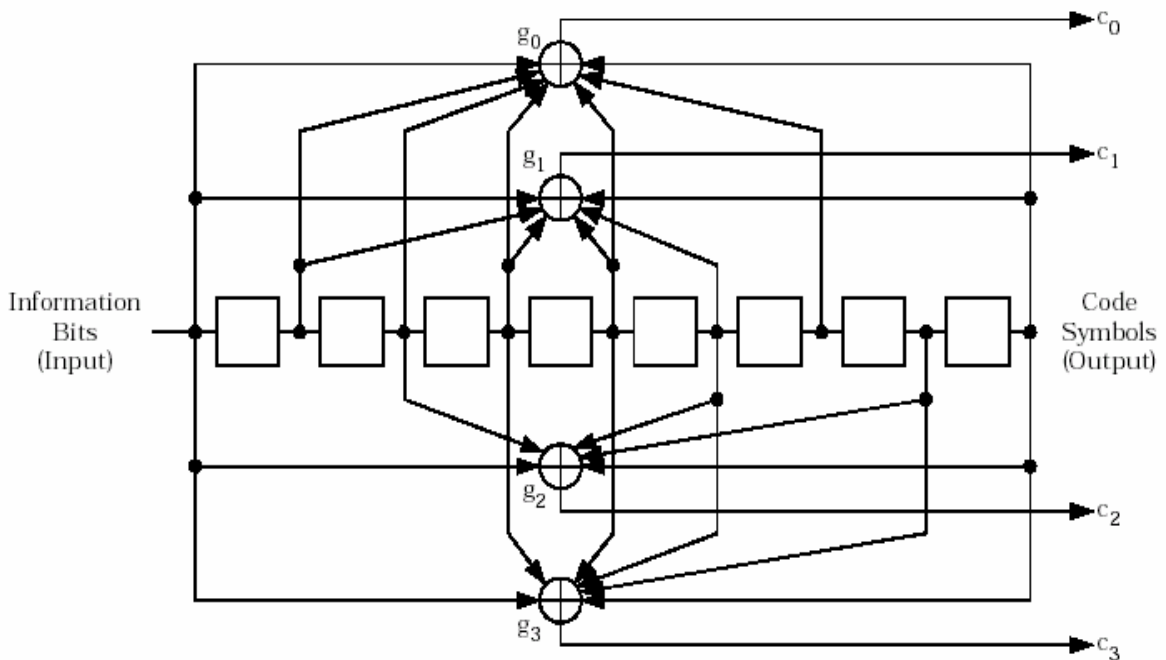


Figure 2.1.3.1.4.1.1-1. $K = 9$, Rate 1/4 Convolutional Encoder

2.1.3.1.4.1.2 Rate 1/3 Convolutional Code

The generator functions for this code shall be g_0 equals 557 (octal), g_1 equals 663 (octal), and g_2 equals 711 (octal). This code generates three code symbols for each data bit input to the encoder. These code symbols shall be output so that the code symbol (c_0) encoded with generator function g_0 shall be output first, the code symbol (c_1) encoded with generator function g_1 shall be output second, and the code symbol (c_2) encoded with generator function g_2 shall be output last. The state of the convolutional encoder, upon initialization, shall be the all-zero state. The first code symbol output after initialization shall be a code symbol encoded with generator function g_0 . The encoder for this code is illustrated in Figure 2.1.3.1.4.1.2-1.

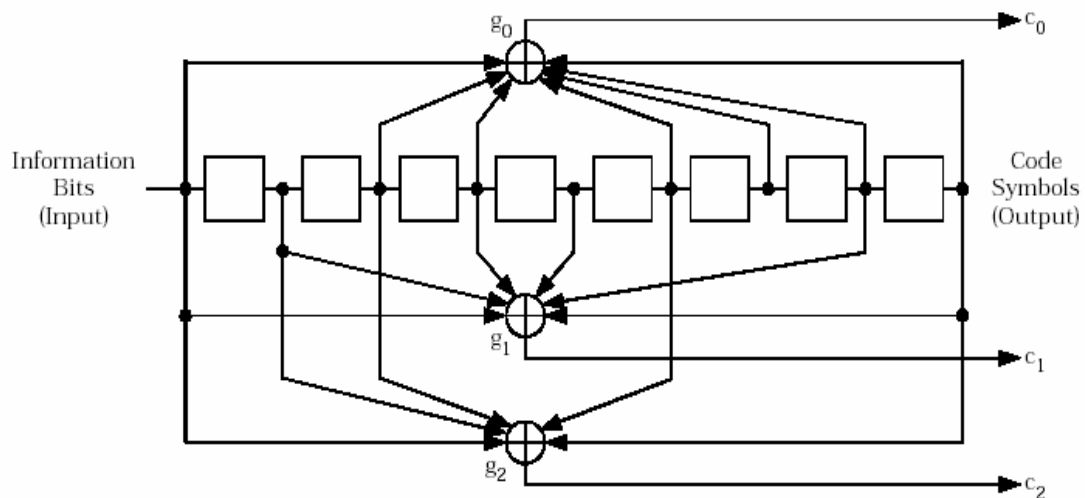


Figure 2.1.3.1.4.1.2-1. K = 9, Rate 1/3 Convolutional Encoder

2.1.3.1.4.1.3 Rate 1/2 Convolutional Code

The generator functions for this code shall be g_0 equals 753 (octal) and g_1 equals 561 (octal). This code generates two code symbols for each data bit input to the encoder. These code symbols shall be output so that the code symbol (c_0) encoded with generator function g_0 shall be output first and the code symbol (c_1) encoded with generator function g_1 shall be output last. The state of the convolutional encoder, upon initialization,

shall be the all-zero state. The first code symbol output after initialization shall be a code symbol encoded with generator function g_0 . The encoder for this code is illustrated in Figure 2.1.3.1.4.1.3-1.

2.1.3.1.4.1.1 Rate 1/4 Convolutional Code

The generator functions for the rate 1/4 code shall be g_0 equals 765 (octal), g_1 equals 671 (octal), g_2 equals 513 (octal), and g_3 equals 473 (octal). This code generates four code symbols for each data bit input to the encoder. These code symbols shall be output so that the code symbol (c_0) encoded with generator function g_0 is output first, the code symbol (c_1) encoded with generator function g_1 is output second, the code symbol (c_2) encoded with generator function g_2 is output third, and the code symbol (c_3) encoded with generator function g_3 is output last. The state of the convolutional encoder, upon initialization, shall be the all-zero state. The first code symbol that is output after initialization shall be a code symbol encoded with generator function g_0 . The encoder for this code is illustrated in Figure 2.1.3.1.4.1.1-1.

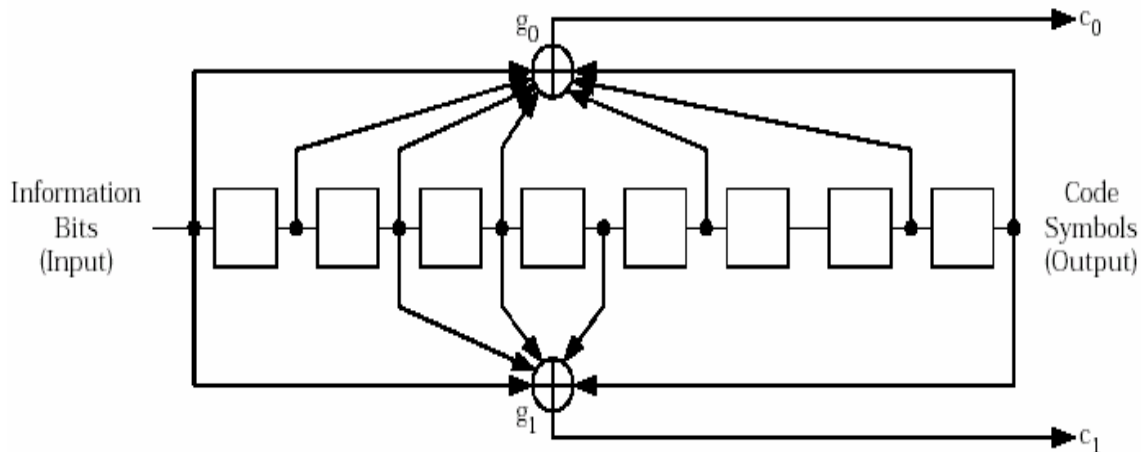


Figure 2.1.3.1.4.1.3-1. K = 9, Rate 1/2 Convolutional Encoder

2.1.3.1.4.2 Turbo Encoding

The turbo encoder encodes the data, frame quality indicator (CRC), and two reserved bits. During encoding, an encoder output tail sequence is added. If the total number of data, frame quality, and reserved input bits is N turbo, the turbo encoder generates N turbo/ R encoded data output symbols followed by $6/R$ tail output symbols, where R is the code rate of $1/2$, $1/3$, $1/4$, or $1/5$. The turbo encoder employs two systematic, recursive, convolutional encoders connected in parallel, with an interleaver, the turbo interleaver, preceding the second recursive convolutional encoder. The two recursive convolutional codes are called the constituent codes of the turbo code. The outputs of the constituent encoders are punctured and repeated to achieve the $(N \text{ turbo} + 6)/R$ output symbols.

2.1.3.1.4.2.1 Rate $1/2$, $1/3$, $1/4$, and $1/5$ Turbo Encoders

A common constituent code shall be used for the turbo codes of rate $1/2$, $1/3$, $1/4$, and $1/5$. The transfer function for the constituent code shall be

$$G(D) = \left[1 \quad \frac{n_0(D)}{d(D)} \quad \frac{n_1(D)}{d(D)} \right]$$

where $d(D) = 1 + D^2 + D^3$, $n_0(D) = 1 + D + D^3$, and $n_1(D) = 1 + D + D^2 + D^3$.

The turbo encoder shall generate an output symbol sequence that is identical to the one generated by the encoder shown in Figure 2.1.3.1.4.2.1-1. Initially, the states of the constituent encoder registers in this figure are set to zero. Then, the constituent encoders are clocked with the switches in the positions noted. The encoded data output symbols are generated by clocking the constituent encoders N turbo times with the switches in the up positions and puncturing the outputs as specified in Table 2.1.3.1.4.2.1-1. Within a puncturing pattern, a '0' means that the symbol shall be deleted and a '1' means that a symbol shall be passed. The constituent encoder outputs for each bit period shall be output in the sequence X, Y_0, Y_1, X, Y_0, Y_1 with the X output first. Symbol repetition is not used in generating the encoded data output symbols.

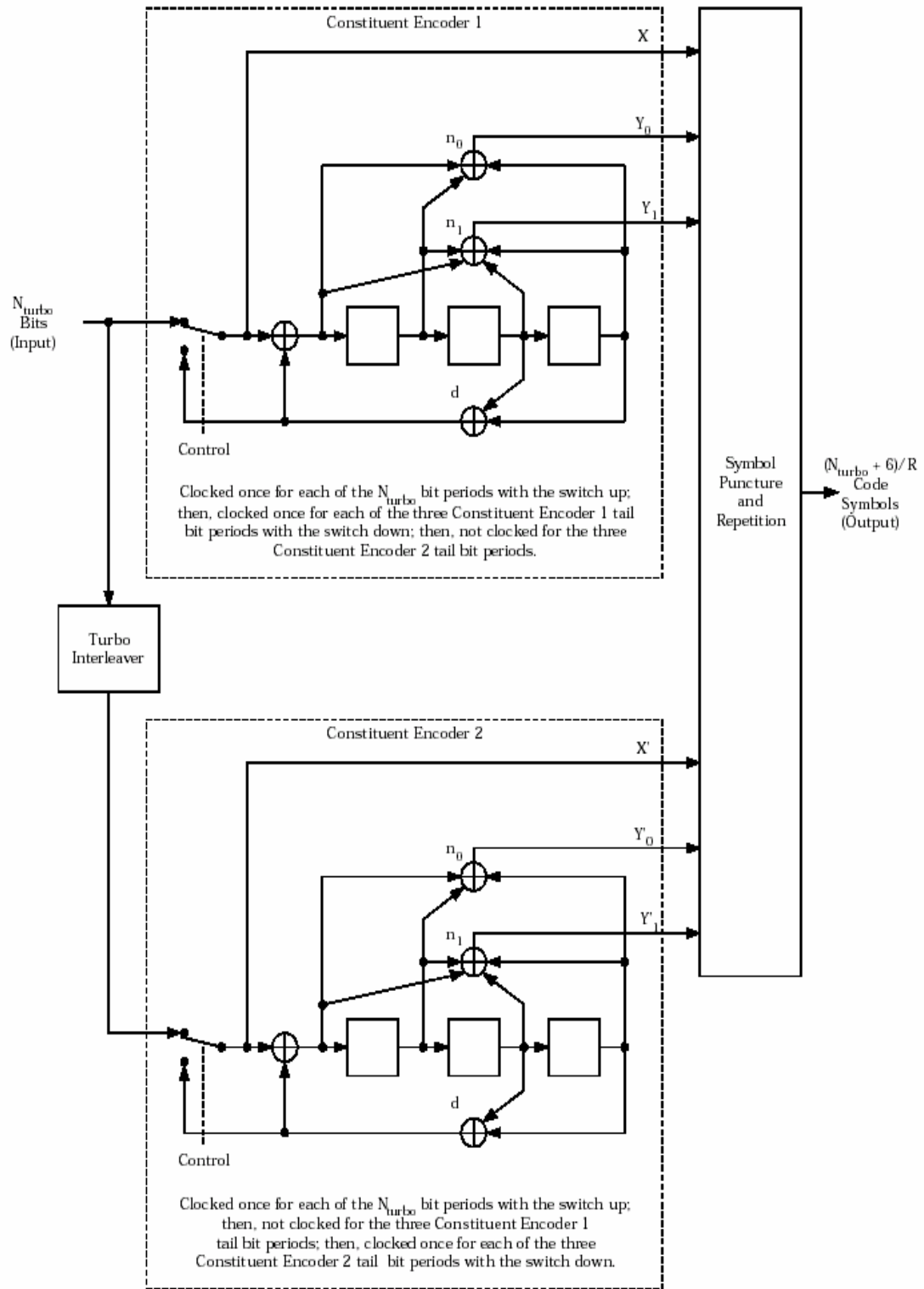


Figure 2.1.3.1.4.2.1-1. Turbo Encoder

Table 2.1.3.1.4.2.1-1. Puncturing Patterns for the Data Bit Periods

Output	Code Rate			
	1/2	1/3	1/4	1/5
X	11	11	11	11
Y ₀	10	11	11	11
Y ₁	00	00	10	11
X'	00	00	00	00
Y' ₀	01	11	01	11
Y' ₁	00	00	11	11

Note: For each rate, the puncturing table shall be read first from top to bottom and then from left to right.

2.1.3.1.4.2.2 Turbo Code Termination

The turbo encoder shall generate 6/R tail output symbols following the encoded data output symbols. This tail output symbol sequence shall be identical to the one generated by the encoder shown in Figure 2.1.3.1.4.2.1-1. The tail output symbols are generated after the constituent encoders have been clocked N_{turbo} times with the switches in the up position. The first 3/R tail output symbols are generated by clocking Constituent Encoder 1 three times with its switch in the down position while Constituent Encoder 2 is not clocked and puncturing and repeating the resulting constituent encoder output symbols. The last 3/R tail output symbols are generated by clocking Constituent Encoder 2 three times with its switch in the down position while Constituent Encoder 1 is not clocked and puncturing and repeating the resulting constituent encoder output symbols. The constituent encoder outputs for each bit period shall be output in the sequence X, Y₀, Y₁, X', Y'₀, Y'₁ with the X output first. The tail output symbol puncturing and symbol repetition shall be as specified in Table 2.1.3.1.4.2.2-1. Within a puncturing pattern, a '0' means that the symbol shall be deleted and a '1'

means that a symbol shall be passed. A 2 or a 3 means that two or three copies of the symbol shall be passed. For rate 1/2 turbo codes, the tail output symbols for each of the first three tail bit periods shall be XY0, and the tail output symbols for each of the last three tail bit periods shall be X Y 0. For rate 1/3 turbo codes, the tail output symbols for each of the first three tail bit periods shall be XXY0, and the tail output symbols for each of the last three tail bit periods shall be X X Y 0. For rate 1/4 turbo codes, the tail output symbols for each of the first three tail bit periods shall be XXY0Y1, and the tail output symbols for each of the last three tail bit periods shall be X X Y 0Y 1. For rate 1/5 turbo codes, the tail output symbols for each of the first three tail bit periods shall be XXXY0Y1, and the tail output symbols for each of the last three tail bit periods shall be X X X Y 0Y 1.

Table 2.1.3.1.4.2.2-1. Puncturing and Symbol Repetition Patterns for the Tail Bit Periods

Output	Code Rate			
	1/2	1/3	1/4	1/5
X	111 000	222 000	222 000	333 000
Y ₀	111 000	111 000	111 000	111 000
Y ₁	000 000	000 000	111 000	111 000
X'	000 111	000 222	000 222	000 333
Y' ₀	000 111	000 111	000 111	000 111
Y' ₁	000 000	000 000	000 111	000 111

Note: The puncturing table shall be read first from top to bottom and then from left to right with the number of repetitions for each symbol indicated by the number in the table.

2.1.3.1.4.2.3 Turbo Interleavers

The turbo interleaver, which is part of the turbo encoder, shall block interleave the data, frame quality indicator (CRC), and reserved bits input to the turbo encoder.

The turbo interleaver shall be functionally equivalent to an approach where the entire sequence of turbo interleaver input bits are written sequentially into an array at a sequence of addresses, and then the entire sequence is read out from a sequence of addresses that are defined by the procedure described below.

Let the sequence of input addresses be from 0 to $N_{\text{turbo}} - 1$, where N_{turbo} is the total number of information bits, frame quality indicator bits, and reserved bits in the turbo interleaver. Then, the sequence of interleaver output addresses shall be equivalent to those generated by the procedure illustrated in Figure 2.1.3.1.4.2.3-1 and described below.

- 1. Determine the turbo interleaver parameter, n , where n is the smallest integer such that $N_{\text{turbo}} \geq 2n + 5$. Table 2.1.3.1.4.2.3-1 gives this parameter for the numbers of bits per frame that are available without flexible data rates.**
- 2. Initialize an $(n + 5)$ -bit counter to 0.**
- 3. Extract the n most significant bits (MSBs) from the counter and add one to form a new value. Then, discard all except the n least significant bits (LSBs) of this value.**
- 4. Obtain the n -bit output of the table lookup defined in Table 2.1.3.1.4.2.3-2 with a read address equal to the five LSBs of the counter. Note that this table depends on the value of n .**
- 5. Multiply the values obtained in Steps 3 and 4, and discard all except the n LSBs.**
- 6. Bit-reverse the five LSBs of the counter.**
- 7. Form a tentative output address that has its MSBs equal to the value obtained in Step 6 and its LSBs equal to the value obtained in Step 5.**
- 8. Accept the tentative output address as an output address if it is less than N_{turbo} ; otherwise, discard it.**
- 9. Increment the counter and repeat Steps 3 through 8 until all N_{turbo} interleaver output addresses are obtained.**

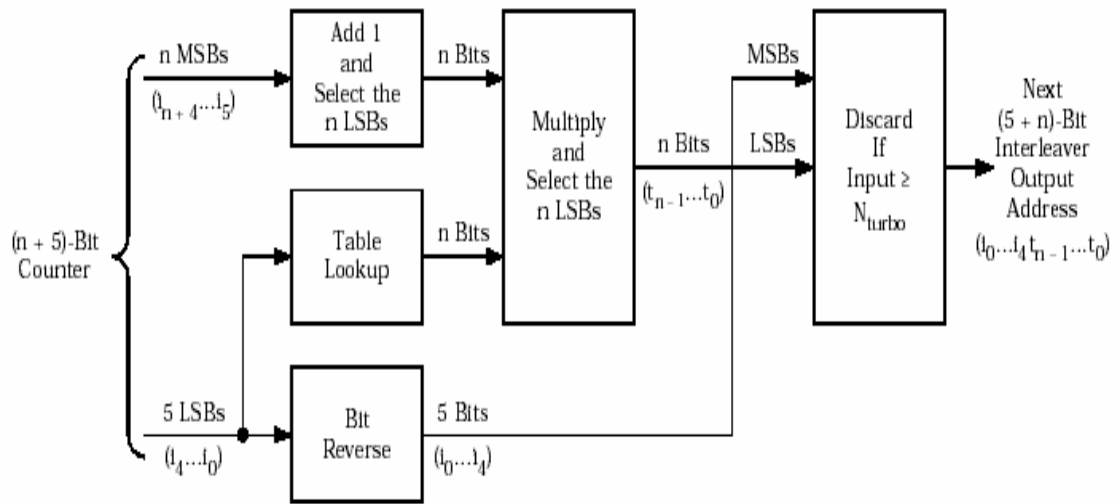


Figure 2.1.3.1.4.2.3-1. Turbo Interleaver Output Address Calculation Procedure

Table 2.1.3.1.4.2.3-1. Turbo Interleaver Parameter

Turbo Interleaver Block Size N_{turbo}	Turbo Interleaver Parameter n
378	4
570	5
762	5
1,146	6
1,530	6
2,298	7
3,066	7
4,602	8
6,138	8
9,210	9
12,282	9
20,730	10

W-CDMA 3GPP TS 25.212 V6.3.0 (2004-12)

3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD) (Release 6)

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH i is denoted by C_i . After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$, where Y_i is the number of encoded bits. The relation between o_{irK} and y_{irK} and between K_i and Y_i is dependent on the channel coding scheme.

The following channel coding schemes can be applied to TrCHs:

- convolutional coding;
- turbo coding.

Usage of coding scheme and coding rate for the different types of TrCH is shown in table 1.

The values of Y_i in connection with each coding scheme:

- convolutional coding with rate 1/2: $Y_i = 2 * K_i + 16$; rate 1/3: $Y_i = 3 * K_i + 24$;
- turbo coding with rate 1/3: $Y_i = 3 * K_i + 12$.

Table 1: Usage of channel coding scheme and coding rate

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		

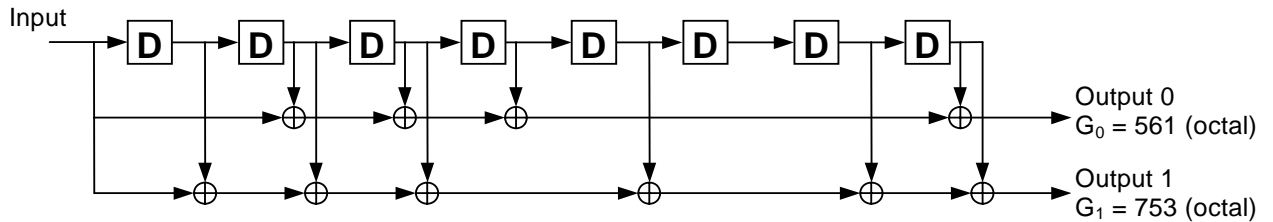
RACH		
CPCH, DCH, DSCH, FACH		1/3, 1/2
	Turbo coding	1/3

4.2.3.1 Convolutional coding

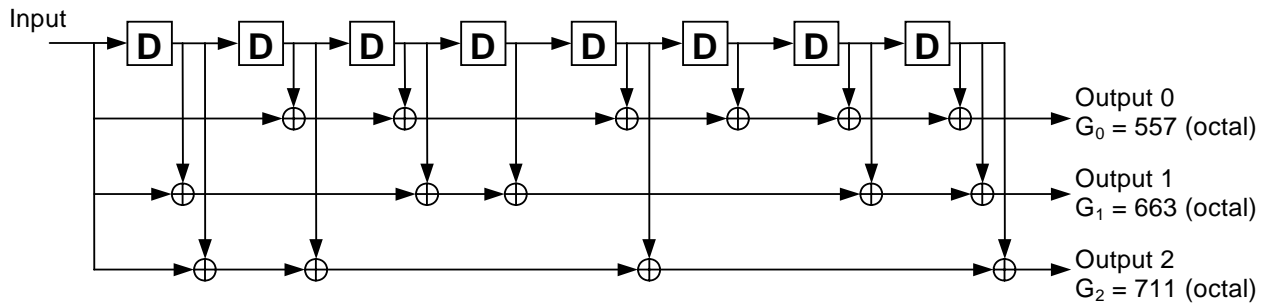
Convolutional codes with constraint length 9 and coding rates 1/3 and 1/2 are defined.

The configuration of the convolutional coder is presented in figure 3.

Output from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output 2, output , ...,output2. Output from the rate 1/2 convolutional coder shall be done in the order output 0, output 1, output 0, output 1, output 0, ..., output 1.8 tail bits with binary value 0 shall be added to the end of the code block before encoding. The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.



(a) Rate 1/2 convolutional coder



(b) Rate 1/3 convolutional coder

Figure 3: Rate 1/2 and rate 1/3 convolutional coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The scheme of Turbo coder is a Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 4.

The transfer function of the 8-state constituent code for PCCC is:

$$G(D) = \begin{bmatrix} 1, \frac{g_1(D)}{g_0(D)} \\ g_0(D) \end{bmatrix},$$

where

$$g_0(D) = 1 + D^2 + D^3,$$

$$g_1(D) = 1 + D + D^3.$$

The initial value of the shift registers of the 8-state constituent encoders shall be all zeros when starting to encode the input bits.

Output from the Turbo coder is

$$x_1, z_1, z'_1, x_2, z_2, z'_2, \dots, x_K, z_K, z'_K,$$

where x_1, x_2, \dots, x_K are the bits input to the Turbo coder i.e. both first 8-state constituent encoder and Turbo code internal interleaver, and K is the number of bits, and z_1, z_2, \dots, z_K and z'_1, z'_2, \dots, z'_K are the bits output from first and second 8-state constituent encoders, respectively.

The bits output from Turbo code internal interleaver are denoted by x'_1, x'_2, \dots, x'_K , and these bits are to be input to the second 8-state constituent encoder.

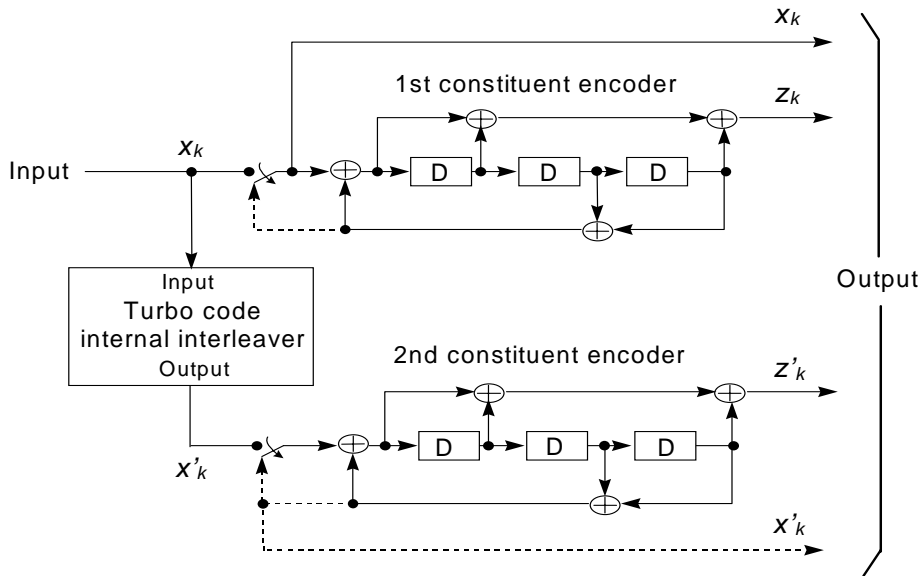


Figure 4: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

4.2.3.2.2 Trellis termination for Turbo coder

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be:

$$x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}.$$

4.2.3.2.3 Turbo code internal interleaver

The Turbo code internal interleaver consists of bits-input to a rectangular matrix with padding, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with

pruning. The bits input to the Turbo code internal interleaver are denoted by $x_1, x_2, x_3, \dots, x_K$, where K is the integer number of the bits and takes one value of $40 \leq K \leq 5114$. The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by $x_k = o_{irk}$ and $K = Ki$.

The following subclause specific symbols are used in subclauses 4.2.3.2.3.1 to 4.2.3.2.3.3:

K Number of bits input to Turbo code internal interleaver

R Number of rows of rectangular matrix

C Number of columns of rectangular matrix

p Prime number

v Primitive root

$\langle s(j) \rangle_{j \in \{0,1,\dots,p-2\}}$ Base sequence for intra-row permutation

q_i Minimum prime integers

r_i Permuted prime integers

$\langle T(i) \rangle_{i \in \{0,1,\dots,R-1\}}$ Inter-row permutation pattern

$\langle U_i(j) \rangle_{j \in \{0,1,\dots,C-1\}}$ Intra-row permutation pattern of i -th row

i Index of row number of rectangular matrix

j Index of column number of rectangular matrix

k Index of bit sequence

4.2.3.2.3.1 Bits-input to rectangular matrix with padding

The bit sequence $x_1, x_2, x_3, \dots, x_K$ input to the Turbo code internal interleaver is written into the rectangular matrix as follows.

(1) Determine the number of rows of the rectangular matrix, R , such that:

$$R = \begin{cases} 5, & \text{if } (40 \leq K \leq 159) \\ 10, & \text{if } ((160 \leq K \leq 200) \text{ or } (481 \leq K \leq 530)) \\ 20, & \text{if } (K = \text{any other value}) \end{cases}$$

The rows of rectangular matrix are numbered 0, 1, ..., R - 1 from top to bottom.

(2) Determine the prime number to be used in the intra-permutation, p, and the number of columns of rectangular matrix, C, such that:

if $(481 \leq K \leq 530)$ then

$p = 53$ and $C = p$.

else

Find minimum prime number p from table 2 such that

$$K \leq R \times (p + 1),$$

and determine C such that

$$C = \begin{cases} p - 1 & \text{if } K \leq R \times (p - 1) \\ p & \text{if } R \times (p - 1) < K \leq R \times p \\ p + 1 & \text{if } R \times p < K \end{cases}$$

end if

The columns of rectangular matrix are numbered 0, 1, ..., C - 1 from left to right.

Table 2: List of prime number p and associated primitive root v

p	v	p	v	p	v	p	v	p	v
7	3	47	5	$\begin{matrix} 10 \\ 1 \end{matrix}$	2	$\begin{matrix} 15 \\ 7 \end{matrix}$	5	$\begin{matrix} 22 \\ 3 \end{matrix}$	3
11	2	53	2	$\begin{matrix} 10 \\ 3 \end{matrix}$	5	$\begin{matrix} 16 \\ 3 \end{matrix}$	2	$\begin{matrix} 22 \\ 7 \end{matrix}$	2
13	2	59	2	$\begin{matrix} 10 \\ 7 \end{matrix}$	2	$\begin{matrix} 16 \\ 7 \end{matrix}$	5	$\begin{matrix} 22 \\ 9 \end{matrix}$	6

p	v	p	v	p	v	p	v	p	v
17	3	61	2	10 ₉	6	17 ₃	2	23 ₃	3
19	2	67	2	11 ₃	3	17 ₉	2	23 ₉	7
23	5	71	7	12 ₇	3	18 ₁	2	24 ₁	7
29	2	73	5	13 ₁	2	19 ₁	1	25 ₁	6
31	3	79	3	13 ₇	3	19 ₃	5	25 ₇	3
37	2	83	2	13 ₉	2	19 ₇	2		
41	6	89	3	14 ₉	2	19 ₉	3		
43	3	97	5	15 ₁	6	21 ₁	2		

(3) Write the input bit sequence $x_1, x_2, x_3, \dots, x_K$ into the $R \times C$ rectangular matrix row by row starting with bit y_1 in column 0 of row 0:

$$\begin{bmatrix} y_1 & y_2 & y_3 & \dots & y_C \\ y_{(C+1)} & y_{(C+2)} & y_{(C+3)} & \dots & y_{2C} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{((R-1)C+1)} & y_{((R-1)C+2)} & y_{((R-1)C+3)} & \dots & y_{R \times C} \end{bmatrix}$$

where $y_k = x_k$ for $k = 1, 2, \dots, K$ and if $R \times C > K$, the dummy bits are padded such that $y_k = 0 \text{ or } 1$ for $k = K + 1, K + 2, \dots, R \times C$. These dummy bits are pruned away from the output of the rectangular matrix after intra-row and inter-row permutations.

4.2.3.2.3.2 Intra-row and inter-row permutations

After the bits-input to the $R \times C$ rectangular matrix, the intra-row and inter-row permutations for the $R \times C$ rectangular matrix are performed stepwise by using the following algorithm with steps (1) – (6):

- (1) Select a primitive root v from table 2 in section 4.2.3.2.3.1, which is indicated on the right side of the prime number p .
- (2) Construct the base sequence $\langle s(j) \rangle_{j \in \{0,1,\dots,p-2\}}$ for intra-row permutation as:
 $s(j) = (v \times s(j-1)) \bmod p$, $j = 1, 2, \dots, (p - 2)$, and $s(0) = 1$.
- (3) Assign $q_0 = 1$ to be the first prime integer in the sequence $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$, and determine the prime integer q_i in the sequence $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$ to be a least prime integer such that $\text{g.c.d}(q_i, p - 1) = 1$, $q_i > 6$, and $q_i > q_{(i - 1)}$ for each $i = 1, 2, \dots, R - 1$. Here g.c.d . is greatest common divisor.
- (4) Permute the sequence $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$ to make the sequence $\langle r_i \rangle_{i \in \{0,1,\dots,R-1\}}$ such that

$$rT(i) = q_i, \quad i = 0, 1, \dots, R - 1,$$

where $\langle T(i) \rangle_{i \in \{0,1,\dots,R-1\}}$ is the inter-row permutation pattern defined as the one of the four kind of patterns, which are shown in table 3, depending on the number of input bits K .

Table 3: Inter-row permutation patterns for Turbo code internal interleaver

Number of input bits K	Nu m b e r o f w	Inter-row permutation patterns $\langle T(0), T(1), \dots, T(R - 1) \rangle$
-----------------------------	---------------------------------------	---

	S R	
$(40 \leq K \leq 159)$	5	$\langle 4, 3, 2, 1, 0 \rangle$
$(160 \leq K \leq 200)$ or $(481 \leq K \leq 530)$	10	$\langle 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 \rangle$
$(2281 \leq K \leq 2480)$ or $(3161 \leq K \leq 3210)$	20	$\langle 19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10 \rangle$
$K = \text{any other value}$	20	$\langle 19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11 \rangle$

(5) Perform the i -th ($i = 0, 1, \dots, R - 1$) intra-row permutation as:

if ($C = p$) then

$$U_i(j) = s((j \times r_i) \bmod (p-1)), \quad j = 0, 1, \dots, (p-2), \text{ and } U_i(p-1) = 0,$$

where $U_i(j)$ is the original bit position of j -th permuted bit of i -th row.

end if

if ($C = p + 1$) then

$$U_i(j) = s((j \times r_i) \bmod (p-1)), \quad j = 0, 1, \dots, (p-2). \quad U_i(p-1) = 0, \text{ and } U_i(p) = p,$$

where $U_i(j)$ is the original bit position of j -th permuted bit of i -th row, and

if ($K = R \times C$) then

Exchange $UR-1(p)$ with $UR-1(0)$.

end if

end if

if ($C = p - 1$) then

$$U_i(j) = s((j \times r_i) \bmod (p-1))^{-1}, \quad j = 0, 1, \dots, (p-2),$$

where $U_i(j)$ is the original bit position of j -th permuted bit of i -th row.

end if

(6) Perform the inter-row permutation for the rectangular matrix based on the pattern $\langle T(i) \rangle_{i \in \{0,1,\dots,R-1\}}$,

where $T(i)$ is the original row position of the i -th permuted row.

4.2.3.2.3.3 Bits-output from rectangular matrix with pruning

After intra-row and inter-row permutations, the bits of the permuted rectangular matrix are denoted by y'_k :

$$\begin{bmatrix} y'_1 & y'_{(R+1)} & y'_{(2R+1)} & \cdots & y'_{((C-1)R+1)} \\ y'_2 & y'_{(R+2)} & y'_{(2R+2)} & \cdots & y'_{((C-1)R+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_R & y'_{2R} & y'_{3R} & \cdots & y'_{C \times R} \end{bmatrix}$$

The output of the Turbo code internal interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted $R \times C$ rectangular matrix starting with bit y'_1 in row 0 of column 0 and ending with bit y'_{CR} in row $R - 1$ of column $C - 1$. The output is pruned by deleting dummy bits that were padded to the input of the rectangular matrix before intra-row and inter row permutations, i.e. bits y'_k that corresponds to bits y_k with $k > K$ are removed from the output. The bits output from Turbo code internal interleaver are denoted by x'_1, x'_2, \dots, x'_K , where x'_1 corresponds to the bit y'_k with smallest index k after pruning, x'_2 to the bit y'_k with second smallest index k after pruning, and so on. The number of bits output from Turbo code internal interleaver is K and the total number of pruned bits is:

$$R \times C - K.$$

4.2.3.3 Concatenation of encoded blocks

After the channel coding for each code block, if C_i is greater than 1, the encoded blocks are serially concatenated so that the block with lowest index r is output first from the channel coding block, otherwise the encoded block is output from channel coding block as it is. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where i is the TrCH number and $E_i = C_i Y_i$. The output bits are defined by the following relations:

$$c_{ik} = y_{ik} \quad k = 1, 2, \dots, Y_i$$

$$c_{ik} = y_{i,2,(k-Y_i)} \quad k = Y_i + 1, Y_i + 2, \dots, 2Y_i$$

$$c_{ik} = y_{i,3,(k-2Y_i)} \quad k = 2Y_i + 1, 2Y_i + 2, \dots, 3Y_i$$

...

$$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)} \quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \dots, C_i Y_i$$

If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.