ETSI TS 125 222 V5.0.0 (2002-03)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
Multiplexing and channel coding (TDD)
(3GPP TS 25.222 version 5.0.0 Release 5)

Reference

RTS/TSGR-0125222Uv5

Keywords

UMTS

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

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 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes multiplexing, channel coding and interleaving for UTRA Physical Layer TDD mode.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 25.202: "UE capabilities".

[2] 3GPP TS 25.211: "Transport channels and physical channels (FDD)".

[3] 3GPP TS 25.212: "Multiplexing and channel coding (FDD)".

[4] 3GPP TS 25.213: "Spreading and modulation (FDD)".

[5] 3GPP TS 25.214: "Physical layer procedures (FDD)".

[6] 3GPP TS 25.215: "Physical layer – Measurements (FDD)".

[7] 3GPP TS 25.221: "Transport channels and physical channels (TDD)".

[9] 3GPP TS 25.223: "Spreading and modulation (TDD)".

[10] 3GPP TS 25.224: "Physical layer procedures (TDD)".

[11] 3GPP TS 25.225: "Measurements".

[12] 3GPP TS 25.331: "RRC Protocol Specification".

[13] 3GPP TS 25.308: "High Speed Downlink Packet Access (HSDPA): Overall description (stage 2)".

3 Definitions, symbols and abbreviations

3.1 Definitions

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

TrCH number: transport channel number represents a TrCH ID assigned to L1 by L2. Transport channels are multiplexed to the CCTrCH in the ascending order of these IDs.

3.2 Symbols

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

$\lceil x \rceil$ round towards ∞, i.e. integer such that $x \le \lceil x \rceil < x+1$
$\lfloor x \rfloor$ round towards $-\infty$, i.e. integer such that $x-1 < \lfloor x \rfloor \le x$

$/x/$	absolute value of x

Unless otherwise is explicitly stated when the symbol is used, the meaning of the following symbols are:

i	TrCH number
j	TFC number
k	Bit number
l	TF number
m	Transport block number
n	Radio frame number
p	PhCH number
r	Code block number
I	Number of TrCHs in a CCTrCH.
C_i	Number of code blocks in one TTI of TrCH i.
F_i	Number of radio frames in one TTI of TrCH i.
M_i	Number of transport blocks in one TTI of TrCH i.
$N_{TCFI\ code\ word}$	Number of TFCI code word bits after TFCI encoding
P	Number of PhCHs used for one CCTrCH.
PL	Puncturing Limit. Signalled from higher layers
RM_i	Rate Matching attribute for TrCH i. Signalled from higher layers.

Temporary variables, i.e. variables used in several (sub)clauses with different meaning.

x, X
y, Y
z, Z

3.3 Abbreviations

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

<ACRONYM>	<Explanation>
ARQ	Automatic Repeat on Request
BCH	Broadcast Channel
BER	Bit Error Rate
BS	Base Station
BSS	Base Station Subsystem
CBR	Constant Bit Rate
CCCH	Common Control Channel
CCTrCH	Coded Composite Transport Channel
CDMA	Code Division Multiple Access
CFN	Connection Frame Number
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
DCA	Dynamic Channel Allocation
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Control
FER	Frame Error Rate
GF	Galois Field
HARQ	Hybrid Automatic Repeat reQuest
HS-DSCH	High Speed Downlink Shared Channel
HS-PDSCH	High Speed Physical Downlink Shared Channel
HS-SCCH	Shared Control Channel for HS-DSCH
HS-SICH	Shared Information Channel for HS-DSCH

JD	Joint Detection
L1	Layer 1
L2	Layer 2
LLC	Logical Link Control
MA	Multiple Access
MAC	Medium Access Control
MS	Mobile Station
MT	Mobile Terminated
NRT	Non-Real Time
OVSF	Orthogonal Variable Spreading Factor
PC	Power Control
PCCC	Parallel Concatenated Convolutional Code
PCH	Paging Channel
PhCH	Physical Channel
PI	Paging Indicator (value calculated by higher layers)
P_q	Paging Indicator (indicator set by physical layer)
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RACH	Random Access Channel
RF	Radio Frequency
RLC	Radio Link Control
RMF	Recommended Modulation Format
RRC	Radio Resource Control
RRM	Radio Resource Management
RSC	Recursive Systematic Convolutional Coder
RT	Real Time
RTBS	Recommended Transport Block Size
RU	Resource Unit
RV	Redundancy Version
SCCC	Serial Concatenated Convolutional Code
SCH	Synchronization Channel
SNR	Signal to Noise Ratio
TCH	Traffic channel
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TFRC	Transport Format Resouce Combination
TFRI	Transport Format Resouce Indicator
TPC	Transmit Power Control
TrBk	Transport Block
TrCH	Transport Channel
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
USCH	Uplink Shared Channel
UTRA	UMTS Terrestrial Radio Access
VBR	Variable Bit Rate

4 Multiplexing, channel coding and interleaving

4.1 General

Data stream from/to MAC and higher layers (Transport block / Transport block set) is encoded/decoded to offer transport services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting (including rate matching), and interleaving and transport channels mapping onto/splitting from physical channels.

In the UTRA-TDD mode, the total number of basic physical channels (a certain time slot one spreading code on a certain carrier frequency) per frame is given by the maximum number of time slots and the maximum number of CDMA codes per time slot.

4.2 General coding/multiplexing of TrCHs

This section only applies to the transport channels: DCH, RACH, DSCH, USCH, BCH, FACH and PCH. Other transport channels which do not use the general method are described separately below.

Figure 1 illustrates the overall concept of transport-channel coding and multiplexing. Data arrives to the coding/multiplexing unit in form of transport block sets, once every transmission time interval. The transmission time interval is transport-channel specific from the set {5 ms(*1), 10 ms, 20 ms, 40 ms, 80 ms}.

Note: (*1) may be applied for PRACH for 1.28 Mcps TDD

The following coding/multiplexing steps can be identified:

- add CRC to each transport block (see subclause 4.2.1);
- TrBk concatenation / Code block segmentation (see subclause 4.2.2);
- channel coding (see subclause 4.2.3) ;
- radio frame size equalization (see subclause 4.2.4);
- interleaving (two steps, see subclauses 4.2.5 and 4.2.10);
- radio frame segmentation (see subclause 4.2.6);
- rate matching (see subclause 4.2.7);
- multiplexing of transport channels (see subclause 4.2.8);
- bit scrambling (see subclause 4.2.9);
- physical channel segmentation (see subclause 4.2.10);
- sub-frame segmentation(see subclause 4.2.12 only for 1.28Mcps TDD)
- mapping to physical channels (see subclause 4.2.13).

The coding/multiplexing steps for uplink and downlink are shown in figures 1 and 2.

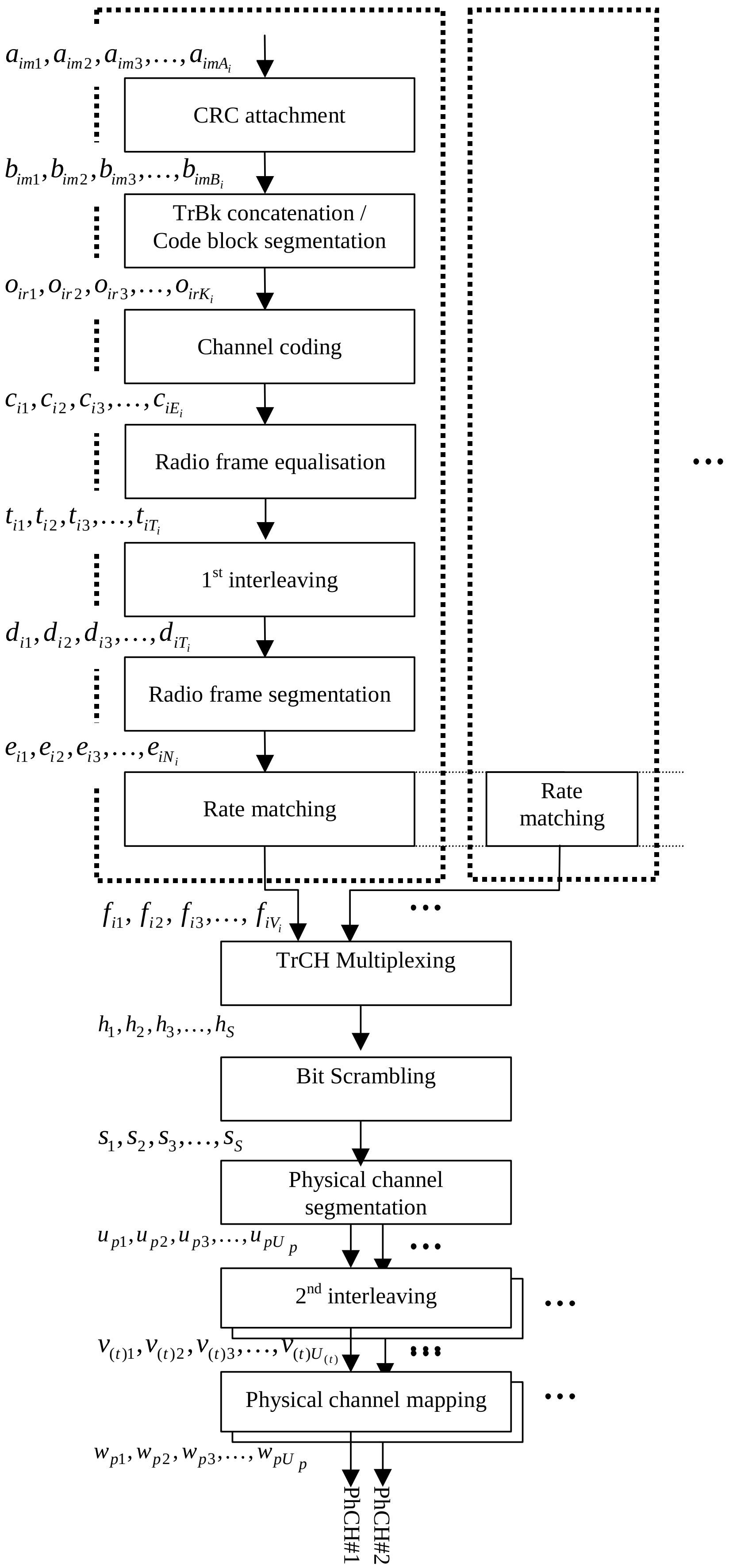


Figure 1: Transport channel multiplexing structure for uplink and downlink for 3.84Mcps TDD

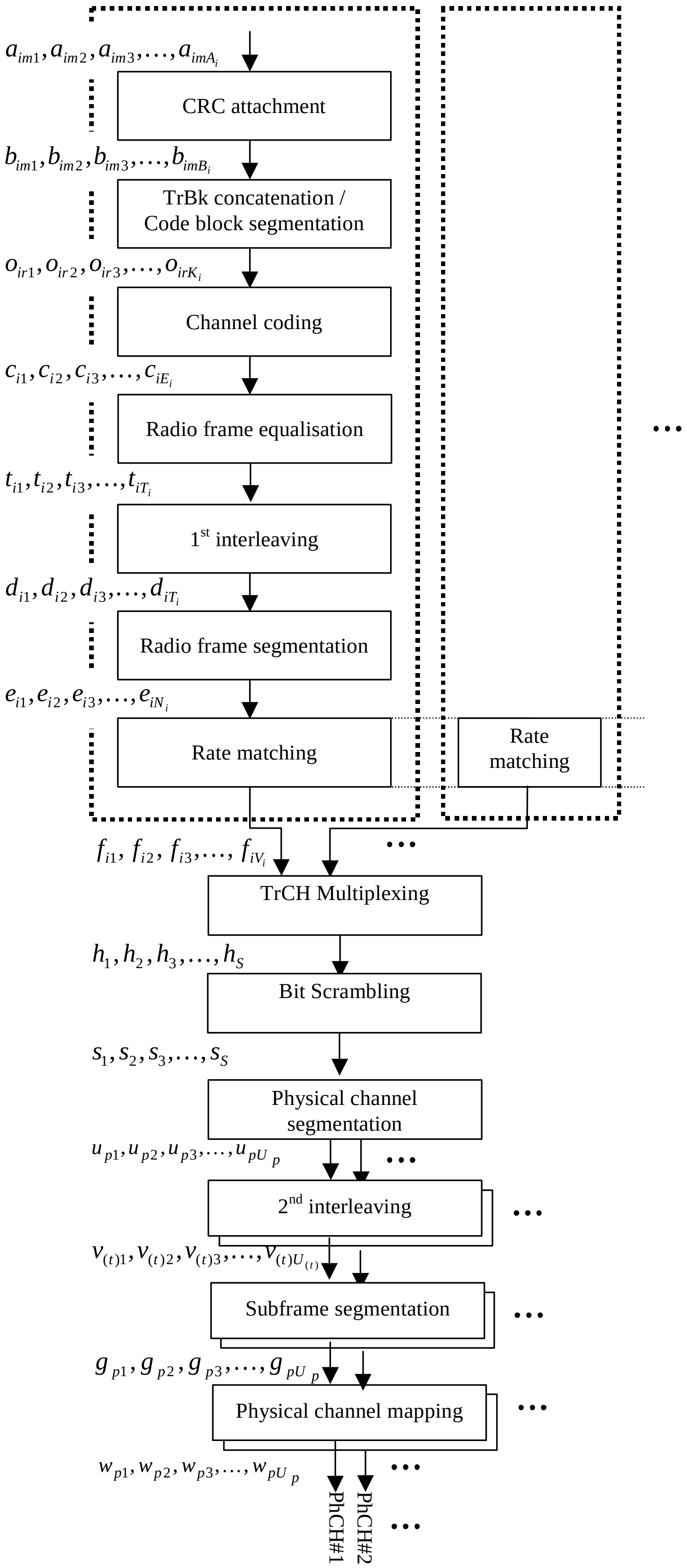


Figure 2: Transport channel multiplexing structure for uplink and downlink of 1.28Mcps TDD

Primarily, transport channels are multiplexed as described above, i.e. into one data stream mapped on one or several physical channels. However, an alternative way of multiplexing services is to use multiple CCTrCHs (Coded Composite Transport Channels), which corresponds to having several parallel multiplexing chains as in figures 1 and 2, resulting in several data streams, each mapped to one or several physical channels.

4.2.1 CRC attachment

Error detection is provided on transport blocks through a Cyclic Redundancy Check (CRC). The size of the CRC is 24, 16, 12, 8 or 0 bits and it is signalled from higher layers what CRC size that should be used for each transport channel.

4.2.1.1 CRC calculation

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

$g_{CRC24}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$

$g_{CRC16}(D) = D^{16} + D^{12} + D^5 + 1$

$g_{CRC12}(D) = D^{12} + D^{11} + D^3 + D^2 + D + 1$

$g_{CRC8}(D) = D^8 + D^7 + D^4 + D^3 + D + 1$

Denote the bits in a transport block delivered to layer 1 by $a_{im1}, a_{im2}, a_{im3}, \ldots, a_{imA_i}$, and the parity bits by $p_{im1}, p_{im2}, p_{im3}, \ldots, p_{imL_i}$. A_i is the size of a transport block of TrCH i, m is the transport block number, and L_i is the number of parity bits. L_i can take the values 24, 16, 12, 8, or 0 depending on what is signalled from higher layers.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_{im1}D^{A_i+23} + a_{im2}D^{A_i+22} + \ldots + a_{imA_i}D^{24} + p_{im1}D^{23} + p_{im2}D^{22} + \ldots + p_{im23}D^1 + p_{im24}$$

yields a remainder equal to 0 when divided by $g_{CRC24}(D)$, polynomial:

$$a_{im1}D^{A_i+15} + a_{im2}D^{A_i+14} + \ldots + a_{imA_i}D^{16} + p_{im1}D^{15} + p_{im2}D^{14} + \ldots + p_{im15}D^1 + p_{im16}$$

yields a remainder equal to 0 when divided by $g_{CRC16}(D)$, polynomial:

$$a_{im1}D^{A_i+11} + a_{im2}D^{A_i+10} + \ldots + a_{imA_i}D^{12} + p_{im1}D^{11} + p_{im2}D^{10} + \ldots + p_{im11}D^1 + p_{im12}$$

yields a remainder equal to 0 when divided by $g_{CRC12}(D)$ and the polynomial:

$$a_{im1}D^{A_i+7} + a_{im2}D^{A_i+6} + \ldots + a_{imA_i}D^{8} + p_{im1}D^{7} + p_{im2}D^{6} + \ldots + p_{im7}D^1 + p_{im8}$$

yields a remainder equal to 0 when divided by $g_{CRC8}(D)$.

If no transport blocks are input to the CRC calculation ($M_i = 0$), no CRC attachment shall be done. If transport blocks are input to the CRC calculation ($M_i \neq 0$) and the size of a transport block is zero ($A_i = 0$), CRC shall be attached, i.e. all parity bits equal to zero.

4.2.1.2 Relation between input and output of the CRC attachment block

The bits after CRC attachment are denoted by $b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$, where $B_i = A_i + L_i$. The relation between a_{imk} and b_{imk} is:

$$b_{imk} = a_{imk} \qquad k = 1, 2, 3, \ldots, A_i$$

$$b_{imk} = p_{im(L_i+1-(k-A_i))} \quad k = A_i + 1, A_i + 2, A_i + 3, \ldots, A_i + L_i$$

4.2.2 Transport block concatenation and code block segmentation

All transport blocks in a TTI are serially concatenated. If the number of bits in a TTI is larger than the maximum size of a code block, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional, turbo coding or no coding is used for the TrCH.

4.2.2.1 Concatenation of transport blocks

The bits input to the transport block concatenation are denoted by $b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$ where i is the TrCH number, m is the transport block number, and B_i is the number of bits in each block (including CRC). The number of transport blocks on TrCH i is denoted by M_i. The bits after concatenation are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where i is the TrCH number and $X_i=M_iB_i$. They are defined by the following relations:

$x_{ik} = b_{i1k}$ $k = 1, 2, \ldots, B_i$

$x_{ik} = b_{i,2,(k-B_i)}$ $k = B_i + 1, B_i + 2, \ldots, 2B_i$

$x_{ik} = b_{i,3,(k-2B_i)}$ $k = 2B_i + 1, 2B_i + 2, \ldots, 3B_i$

…

$x_{ik} = b_{i,M_i,(k-(M_i-1)B_i)}$ $k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \ldots, M_iB_i$

4.2.2.2 Code block segmentation

Segmentation of the bit sequence from transport block concatenation is performed if $X_i>Z$. The code blocks after segmentation are of the same size. The number of code blocks on TrCH i is denoted by C_i. If the number of bits input to the segmentation, X_i, is not a multiple of C_i, filler bits are added to the beginning of the first block. If turbo coding is selected and $X_i < 40$, filler bits are added to the beginning of the code block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

- convolutional coding: $Z = 504$;
- turbo coding: $Z = 5114$;
- no channel coding: $Z =$ *unlimited*.

The bits output from code block segmentation, for $C_i \neq 0$, are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits per code block.

Number of code blocks:

$$C_i = \begin{cases} \lceil X_i/Z \rceil & \text{when } Z \neq \textit{unlimited} \\ 0 & \text{when } Z = \textit{unlimited} \text{ and } X_i = 0 \\ 1 & \text{when } Z = \textit{unlimited} \text{ and } X_i \neq 0 \end{cases}$$

Number of bits in each code block (applicable for $C_i \neq 0$ only):

if $X_i < 40$ and Turbo coding is used, then

$K_i = 40$

else

$K_i = \lceil X_i / C_i \rceil$

end if

Number of filler bits: $Y_i = C_i K_i - X_i$

for k = 1 to Y_i -- Insertion of filler bits

$$o_{i1k} = 0$$

end for

for $k = Y_i$+1 to K_i

$$o_{i1k} = x_{i,(k-Y_i)}$$

end for

r = 2 -- Segmentation

while $r \le C_i$

for k = 1 to K_i

$$o_{irk} = x_{i,(k+(r-1)\cdot K_i - Y_i)}$$

end for

r = r+1

end while

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, 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 $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$, 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 transport channels:

- convolutional coding;
- turbo coding;
- no coding.

Usage of coding scheme and coding rate for the different types of TrCH is shown in tables 1 and 2. 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: $Y_i = 3*K_i + 12$;
- no coding: $Y_i = K_i$.

Table 1: Usage of channel coding scheme and coding rate for 3.84Mcps TDD

<table>
<tr><th>Type of TrCH</th><th>Coding scheme</th><th>Coding rate</th></tr>
<tr><td>BCH</td><td rowspan="4">Convolutional coding</td><td rowspan="3">1/2</td></tr>
<tr><td>PCH</td></tr>
<tr><td>RACH</td></tr>
<tr><td rowspan="3">DCH, DSCH, FACH, USCH</td><td>1/3, 1/2</td></tr>
<tr><td>Turbo coding</td><td>1/3</td></tr>
<tr><td colspan="2">No coding</td></tr>
</table>

Table 2: Usage of channel coding scheme and coding rate for 1.28Mcps TDD

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/3
PCH		1/3, 1/2
RACH		1/2
DCH, DSCH, FACH, USCH		1/3, 1/2
	Turbo coding	1/3
	No coding	

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 output 0, output 1, output 2, output 0, output 1, output 2, output 0,…,output 2. 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.

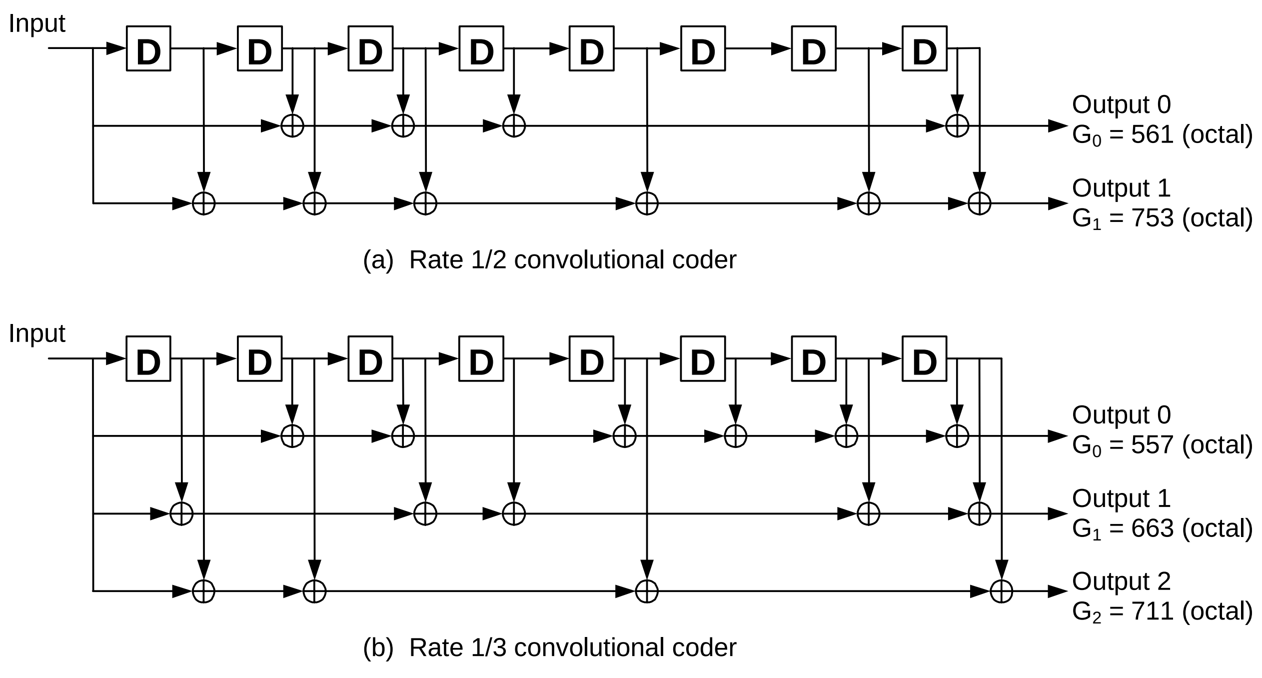


Figure 3: Rate 1/2 and rate 1/3 convolutional coders

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:

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right],$$

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 , Y'(0), X(1), Y(1), Y'(1), etc:

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

where $x_1, x_2, \ldots, x_K$ 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 $z_1, z_2, \ldots, z_K$ and $z'_1, z'_2, \ldots, z'_K$ are the bits output from first and second 8-state constituent encoders, respectively.

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

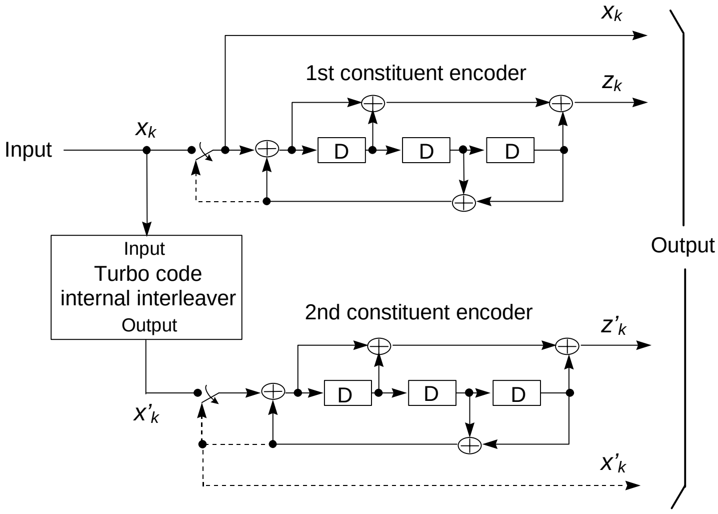


Figure 4: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

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 $x_1, x_2, x_3, \ldots, x_K$, 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 $K = K_i$.

The following subclause specific symbols are used in subclauses 4.2.3.2.3.1 to 4.2.3.4.3.3:

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
$\langle s(j) \rangle_{j \in \{0,1,\cdots,p-2\}}$	Base sequence for intra-row permutation
q_i	Minimum prime integers
r_i	Permuted prime integers
$\langle T(i) \rangle_{i \in \{0,1,\cdots,R-1\}}$	Inter-row permutation pattern
$\langle U_i(j) \rangle_{j \in \{0,1,\cdots,C-1\}}$	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 $x_1, x_2, x_3, \ldots, x_K$ 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 3 such that

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

and determine C such that

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

end if

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

Table 3: List of prime number *p* and associated primitive root *v*

p	*v*	*p*	*v*	*p*	*v*	*p*	*v*	*p*	*v*
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

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

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

where $y_k = x_k$ for k = 1, 2, …, K and if $R \times C > K$, the dummy bits are padded such that $y_k = 0 or 1$ for $k = K + 1$, K + 2, …, $R \times C$. 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 3 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 $\langle s(j) \rangle_{j \in \{0,1,\cdots,p-2\}}$ for intra-row permutation as:

$s(j) = (v \times s(j-1)) \bmod p$, j = 1, 2,… (p - 2), and $s(0)$ = 1.

(3) Assign q_0 = 1 to be the first prime integer in the sequence $\langle q_i \rangle_{i \in \{0,1,\cdots,R-1\}}$, and determine the prime integer q_i in the sequence $\langle q_i \rangle_{i \in \{0,1,\cdots,R-1\}}$ to be a least prime integer such that g.c.d(q_i, p - 1) = 1, $q_i > 6$, and $q_i > q_{(i-1)}$ for each i = 1, 2, …, R – 1. Here g.c.d. is greatest common divisor.

(4) Permute the sequence $\langle q_i \rangle_{i \in \{0,1,\cdots,R-1\}}$ to make the sequence $\langle r_i \rangle_{i \in \{0,1,\cdots,R-1\}}$ such that

$r_{T(i)} = q_i$, i = 0, 1, …. , R - 1, where $\langle T(i) \rangle_{i \in \{0,1,\cdots,R-1\}}$ is the inter-row permutation pattern defined as the one of the four kind of patterns, which are shown in table 4, depending on the number of input bits *K*.

Table 4: Inter-row permutation patterns for Turbo code internal interleaver

Number of input bits *K*	Number of rows *R*	Inter-row permutation patterns <*T*(0), *T*(1), …, *T*(*R* - 1)>
(40 ≤ *K* ≤ 159)	5	<4, 3, 2, 1, 0>
(160 ≤ *K* ≤ 200) or (481 ≤ *K* ≤ 530)	10	<9, 8, 7, 6, 5, 4, 3, 2, 1, 0>
(2281 ≤ *K* ≤ 2480) or (3161 ≤ *K* ≤ 3210)	20	<19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10>
K = any other value	20	<19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11>

(5) Perform the *i*-th (*i* = 0,1, …, *R* - 1) intra-row permutation as:

if (*C* = *p*) then

$U_i(j) = s((j \times r_i) \bmod (p-1))$, j = 0, 1, …, (p - 2), 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

$U_i(j) = s((j \times r_i) \bmod (p-1))$, j = 0, 1, …, (p - 2). $U_i(p-1) = 0$, 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

Exhange $U_{R-1}(p)$ with $U_{R-1}(0)$.

end if

end if

if (*C* = *p* - 1) then

$U_i(j) = s((j \times r_i) \bmod (p-1)) - 1$, j =0, 1, …, (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 $\langle T(i) \rangle_{i \in \{0,1,\cdots,R-1\}}$,

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 $x'_1, x'_2, \ldots, x'_K$, 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 $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and $E_i = C_iY_i$. The output bits are defined by the following relations:

$c_{ik} = y_{i1k}$ $k = 1, 2, \ldots, Y_i$

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

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

…

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

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

4.2.4 Radio frame size equalisation

Radio frame size equalisation is padding the input bit sequence in order to ensure that the output can be segmented in F_i data segments of same size as described in the subclause 4.2.6.

The input bit sequence to the radio frame size equalisation is denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is TrCH number and E_i the number of bits. The output bit sequence is denoted by $t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$, where T_i is the number of bits. The output bit sequence is derived as follows:

$t_{ik} = c_{ik}$, for k = 1… E_i and

$t_{ik} = \{0, 1\}$ for k= E_i +1… T_i, if $E_i < T_i$

where

$T_i = F_i * N_i$ and

$N_i = \lceil E_i / F_i \rceil$ is the number of bits per segment after size equalisation.

4.2.5 1st interleaving

The 1st interleaving is a block interleaver with inter-column permutations. The input bit sequence to the block interleaver is denoted by $x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,X_i}$, where i is TrCH number and X_i the number of bits. Here X_i is guaranteed to be an integer multiple of the number of radio frames in the TTI. The output bit sequence from the block interleaver is derived as follows:

1) select the number of columns C1 from table 5 depending on the TTI. The columns are numbered 0, 1, …, C1 - 1 from left to right.

2) determine the number of rows of the matrix, R1 defined as

   R1 = X_i / C1.

The rows of the matrix are numbered 0, 1, …, R1 - 1 from top to bottom.

3) write the input bit sequence into the R1 × C1 matrix row by row starting with bit $x_{i,1}$ in column 0 of row 0 and ending with bit $x_{i,(\mathrm{R1}\times\mathrm{C1})}$ in column C1 - 1 of row R1 – 1:

$$\begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & \ldots & x_{i,\mathrm{C1}} \\ x_{i,(\mathrm{C1}+1)} & x_{i,(\mathrm{C1}+2)} & x_{i,(\mathrm{C1}+3)} & \ldots & x_{i,(2\times\mathrm{C1})} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+1)} & x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+2)} & x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+3)} & \ldots & x_{i,(\mathrm{R1}\times\mathrm{C1})} \end{bmatrix}$$

4) Perform the inter-column permutation for the matrix based on the pattern $\left\langle \mathrm{P1}_{\mathrm{C1}}(j) \right\rangle_{j\in\{0,1,\ldots,\mathrm{C1}-1\}}$ shown in table 5, where $\mathrm{P1}_{\mathrm{C1}}(j)$ is the original column position of the *j*-th permuted column. After permutation of the columns, the bits are denoted by $y_{i,k}$:

$$\begin{bmatrix} y_{i,1} & y_{i,(\mathrm{R1}+1)} & y_{i,(2\times\mathrm{R1}+1)} & \cdots & y_{i,((\mathrm{C1}-1)\times\mathrm{R1}+1)} \\ y_{i,2} & y_{i,(\mathrm{R1}+2)} & y_{i,(2\times\mathrm{R1}+2)} & \cdots & y_{i,((\mathrm{C1}-1)\times\mathrm{R1}+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{i,\mathrm{R1}} & y_{i,(2\times\mathrm{R1})} & y_{i,(3\times\mathrm{R1})} & \cdots & y_{i,(\mathrm{C1}\times\mathrm{R1})} \end{bmatrix}$$

5) Read the output bit sequence $y_{i,1}, y_{i,2}, y_{i,3}, \ldots, y_{i,(\mathrm{C1}\times\mathrm{R1})}$ of the block interleaver column by column from the inter-column permuted R1 × C1 matrix. Bit $y_{i,1}$ corresponds to row 0 of column 0 and bit $y_{i,(\mathrm{R1}\times\mathrm{C1})}$ corresponds to row R1 - 1 of column C1 - 1.

Table 5 Inter-column permutation patterns for 1st interleaving

TTI	Number of columns C1	Inter-column permutation patterns <P1C1(0), P1C1(1), …, P1C1(C1-1)>
5ms(*1), 10 ms	1	<0>
20 ms	2	<0,1>
40 ms	4	<0,2,1,3>
80 ms	8	<0,4,2,6,1,5,3,7>

(*1) can be used for PRACH for 1.28 Mcps TDD

4.2.5.1 Relation between input and output of 1st interleaving

The bits input to the 1st interleaving are denoted by $t_{i,1}, t_{i,2}, t_{i,3}, \ldots, t_{i,T_i}$, where *i* is the TrCH number and T_i the number of bits. Hence, $x_{i,k} = t_{i,k}$ and $X_i = T_i$.

The bits output from the 1st interleaving are denoted by $d_{i,1}, d_{i,2}, d_{i,3}, \ldots, d_{i,T_i}$, and $d_{i,k} = y_{i,k}$.

4.2.6 Radio frame segmentation

When the transmission time interval is longer than 10 ms, the input bit sequence is segmented and mapped onto consecutive F_i radio frames. Following radio frame size equalisation the input bit sequence length is guaranteed to be an integer multiple of F_i.

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where *i* is the TrCH number and X_i is the number bits. The F_i output bit sequences per TTI are denoted by $y_{i,n_i1}, y_{i,n_i2}, y_{i,n_i3}, \ldots, y_{i,n_iY_i}$ where n_i is the radio frame number in current TTI and Y_i is the number of bits per radio frame for TrCH *i*. The output sequences are defined as follows:

$y_{i,n_ik} = x_{i,((n_i-1)Y_i)+k}$, $n_i = 1\ldots F_i$, $k = 1\ldots Y_i$

where

$Y_i = (X_i / F_i)$ is the number of bits per segment.

The n_i –th segment is mapped to the n_i –th radio frame of the transmission time interval.

The input bit sequence to the radio frame segmentation is denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$, where i is the TrCH number and T_i the number of bits. Hence, $x_{ik} = d_{ik}$ and $X_i = T_i$.

The output bit sequence corresponding to radio frame n_i is denoted by $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$, where i is the TrCH number and N_i is the number of bits. Hence, $e_{i,k} = y_{i,n_i k}$ and $N_i = Y_i$.

4.2.7 Rate matching

Rate matching means that bits on a TrCH are repeated or punctured. Higher layers assign a rate-matching attribute for each TrCH. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

The number of bits on a TrCH can vary between different transmission time intervals. When the number of bits between different transmission time intervals is changed, bits are repeated to ensure that the total bit rate after TrCH multiplexing is identical to the total channel bit rate of the allocated physical channels.

If no bits are input to the rate matching for all TrCHs within a CCTrCH, the rate matching shall output no bits for all TrCHs within the CCTrCH.

Notation used in subclause 4.2.7 and subclauses:

N_{ij} : Number of bits in a radio frame before rate matching on TrCH *i* with transport format combination *j*.

$\Delta N_{i,j}$: If positive – number of bits to be repeated in each radio frame on TrCH *i* with transport format

If negative – number of bits to be punctured in each radio frame on TrCH *i* with transport format combination *j*.

RM_i : Semi-static rate matching attribute for TrCH *i*. Signalled from higher layers.

PL : Puncturing limit. This value limits the amount of puncturing that can be applied in order to minimise the number of physical channels. Signalled from higher layers. The allowed puncturing in % is actually equal to (1-PL)*100.

$N_{data,j}$: Total number of bits that are available for a CCTrCH in a radio frame with transport format combination j.

P : number of physical channels used in the current frame.

P_{max} : maximum number of physical channels allocated for a CCTrCH.

U_p : Number of data bits in the physical channel p with p = 1...P.

I : Number of TrCHs in a CCTrCH.

Z_{ij} : Intermediate calculation variable.

F_i : Number of radio frames in the transmission time interval of TrCH *i*.

n_i : Radio frame number in the transmission time interval of TrCH *i* ($0 \leq n_i < F_i$).

q : Average puncturing or repetition distance(normalised to only show the remaining rate matching on top of an integer number of repetitions).

$P1_F(n_i)$: The column permutation function of the 1st interleaver, $P1_F(x)$ is the original position of column with number x after permutation. P1 is defined on table 5 of section 4.2.5 (note that $P1_F$ self-inverse).

S[n] : The shift of the puncturing or repetition pattern for radio frame n_i when $n = P1_{F_i}(n_i)$.

$TF_i(j)$: Transport format of TrCH i for the transport format combination j.

$TFS(i)$: The set of transport format indexes *l* for TrCH i.

e_{ini} : Initial value of variable e in the rate matching pattern determination algorithm of subclause 4.2.7.3.

e_{plus} : Increment of variable *e* in the rate matching pattern determination algorithm of subclause 4.2.7.3.

e_{minus} : Decrement of variable *e* in the rate matching pattern determination algorithm of subclause 4.2.7.3.

b : Indicates systematic and parity bits.

b=1: Systematic bit. *X(t)* in subclause 4.2.3.2.1.

b=2: 1st parity bit (from the upper Turbo constituent encoder). *Y(t)* in subclause 4.2.3.2.1.

b=3: 2nd parity bit (from the lower Turbo constituent encoder). *Y'(t)* in subclause 4.2.3.2.1.

4.2.7.1 Determination of rate matching parameters

The following relations, defined for all TFC *j*, are used when calculating the rate matching pattern:

$$Z_{0,j} = 0$$

$$Z_{i,j} = \left\lfloor \frac{\left(\left(\sum_{m=1}^{i} RM_m \times N_{m,j}\right) \times N_{data,j}\right)}{\sum_{m=1}^{I} RM_m \times N_{m,j}} \right\rfloor \quad \text{for all i = 1 ... } I \text{ (1)}$$

$$\Delta N_{i,j} = Z_{i,j} - Z_{i-1,j} - N_{i,j} \quad \text{for all i = 1 ... } I$$

Puncturing can be used to minimise the required transmission capacity. The maximum amount of puncturing that can be applied is 1-PL, PL is signalled from higher layers. The possible values for N_{data} depend on the number of physical channels P_{max}, allocated to the respective CCTrCH, and on their characteristics (spreading factor, length of midamble and TFCI code word, usage of TPC and multiframe structure), which is given in [7].

For each physical channel an individual minimum spreading factor Sp_{min} is transmitted by means of the higher layers. Denote the number of data bits in each physical channel by $U_{p,Sp}$, where *p* indicates the sequence number $1 \le p \le P_{max}$ and *Sp* indicates the spreading factor with the possible values {*16, 8, 4, 2, 1*} of this physical channel. The index *p* is described in section 4.2.13 with the following modifications: spreading factor (*Q*) is replaced by the minimum spreading factor Sp_{min} and *k* is replaced by the channelization code index at $Q = Sp_{min}$. Then, for N_{data} one of the following values in ascending order can be chosen:

$$\left\{U_{1,S1_{min}}, U_{1,S1_{min}} + U_{2,S2_{min}}, U_{1,S1_{min}} + U_{2,S2_{min}} + ... + U_{P_{max},(SP_{max})_{min}}\right\}$$

Optionally, if indicated by higher layers for the UL the UE shall vary the spreading factor autonomously, so that N_{data} is one of the following values in ascending order:

$$\left\{U_{1,16},...,U_{1,S1_{min}}, U_{1,S1_{min}} + U_{2,16},...,U_{1,S1_{min}} + U_{2,S2_{min}},...,U_{1,S1_{min}} + U_{2,S2_{min}} + ... + U_{P_{max},16},...,U_{1,S1_{min}} + U_{2,S2_{min}} + ... + U_{P_{max},(SP_{max})_{min}}\right\}$$

$N_{data,j}$ for the transport format combination j is determined by executing the following algorithm:

$$\text{SET1} = \{\ N_{data} \text{ such that } \left(\min_{1\le y\le I}\left\{RM_y\right\}\right) \times N_{data} - PL \times \sum_{x=1}^{I} RM_x \times N_{x,j} \text{ is non negative }\}$$

$$N_{data,j} = \min \text{SET1}$$

The number of bits to be repeated or punctured, $\Delta N_{i,j}$, within one radio frame for each TrCH i is calculated with the relations given at the beginning of this subclause for all possible transport format combinations j and selected every radio frame.

If $\Delta N_{i,j} = 0$ then the output data of the rate matching is the same as the input data and the rate matching algorithm of subclause 4.2.7.3 does not need to be executed.

Otherwise, the rate matching pattern is calculated with the algorithm described in subclause 4.2.7.3. For this algorithm the parameters e_{ini}, e_{plus}, e_{minus}, and X_i are needed, which are calculated according to the equations in subclauses 4.2.7.1.1 and 4.2.7.1.2.

4.2.7.1.1 Uncoded and convolutionally encoded TrCHs

$a = 2$

$\Delta N_i = \Delta N_{i,j}$

$X_i = N_{i,j}$

$R = \Delta N_{i,j} \bmod N_{i,j}$ -- note: in this context $\Delta N_{i,j} \bmod N_{i,j}$ is in the range of 0 to $N_{i,j}$-1 i.e. -1 mod 10 = 9.

if $R \neq 0$ and $2\times R \leq N_{i,j}$

then $q = \lceil N_{i,j} / R \rceil$

else

$q = \lceil N_{i,j} / (R - N_{i,j}) \rceil$

endif

NOTE 1: q is a signed quantity.

If q is even

then $q' = q + \gcd(|q|, F_i) / F_i$ -- where gcd ($|q|$, F_i) means greatest common divisor of $|q|$ and F_i

NOTE 2: q' is not an integer, but a multiple of 1/8.

else

$q' = q$

endif

for x = 0 to F_i-1

$S[|\lfloor x\times q' \rfloor| \bmod F_i] = (|\lfloor x * q' \rfloor| \operatorname{div} F_i)$

end for

$e_{ini} = (a \times S[P1_{Fi}(n_i)] \times |\Delta N_i| + 1) \bmod (a \times N_{i,j})$

$e_{plus} = a \times X_i$

$e_{minus} = a \times |\Delta N_i|$

puncturing for $\Delta N_i < 0$, repetition otherwise.

4.2.7.1.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{i,j} > 0$, the parameters in subclause 4.2.7.1.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index b is used to indicate systematic (b=1), 1st parity (b=2), and 2nd parity bit (b=3).

a = 2 when b=2

a = 1 when b=3

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{i,j} / 2 \rfloor, & b = 2 \\ \lceil \Delta N_{i,j} / 2 \rceil, & b = 3 \end{cases}$$

If ΔN_i is calculated as 0 for b=2 or b=3, then the following procedure and the rate matching algorithm of subclause 4.2.7.3 don't need to be performed for the corresponding parity bit stream.

$X_i = \lfloor N_{i,j} / 3 \rfloor$,

$q = \lfloor X_i / |\Delta N_i| \rfloor$

if(q ≤ 2)

for r=0 to F_i-1

S[(3×r+b-1) mod F_i] = r mod 2;

end for

else

if q is even

then q′ = q – gcd(q, F_i)/ F_i -- where gcd (q, F_i) means greatest common divisor of q and F_i

NOTE: q′ is not an integer, but a multiple of 1/8.

else q′ = q

endif

for x=0 to F_i –1

r = $\lceil x \times q' \rceil$ mod F_i;

S[(3×r+b-1) mod F_i] = $\lceil x \times q' \rceil$ div F_i;

endfor

endif

For each radio frame, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.3, where:

X_i is as above,

e_{ini} = (a×S[P1 F_i (n_i)]×·|ΔN_i| + X_i) mod (a×X_i), if e_{ini} = 0 then e_{ini} = a×X_i

e_{plus} = a×X_i

e_{minus} = a×|ΔN_i|

4.2.7.2 Bit separation and collection for rate matching

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits may be punctured. The systematic bits, first parity bits, and second parity bits in the bit sequence input to the rate matching block are therefore separated into three sequences.

The first sequence contains:

- All of the systematic bits that are from turbo encoded TrCHs.

- From 0 to 2 first and/or second parity bits that are from turbo encoded TrCHs. These bits come into the first sequence when the total number of bits in a block after radio frame segmentation is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second and third sequences shall be of equal length, whereas the first sequence can contain from 0 to 2 more bits. Puncturing is applied only to the second and third sequences.

The bit separation function is transparent for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition. The bit separation and bit collection are illustrated in figures 5 and 6.

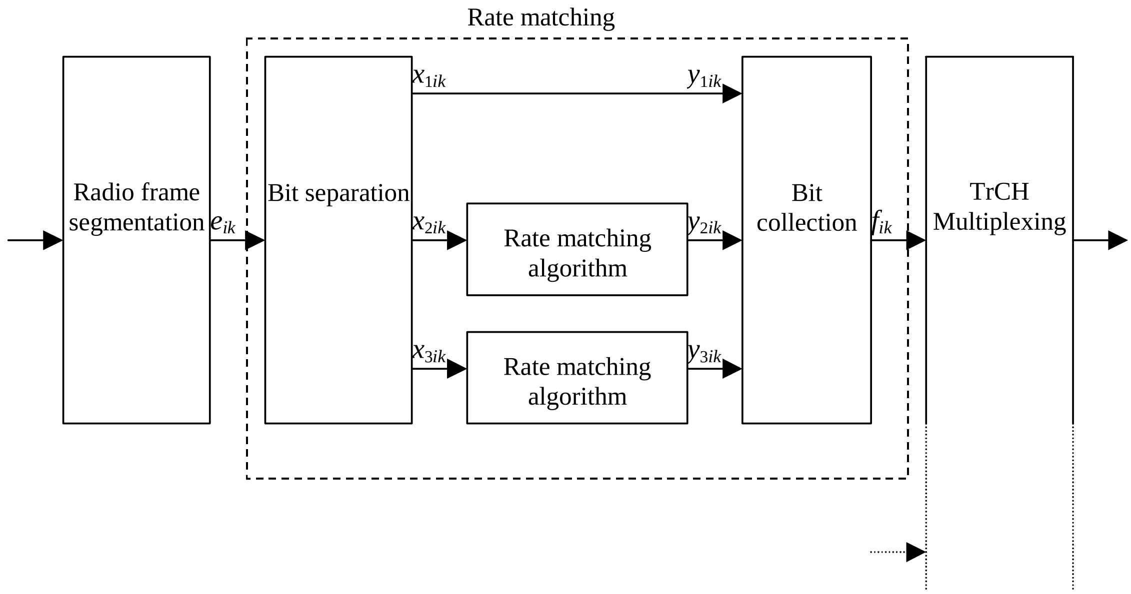


Figure 5: Puncturing of turbo encoded TrCHs

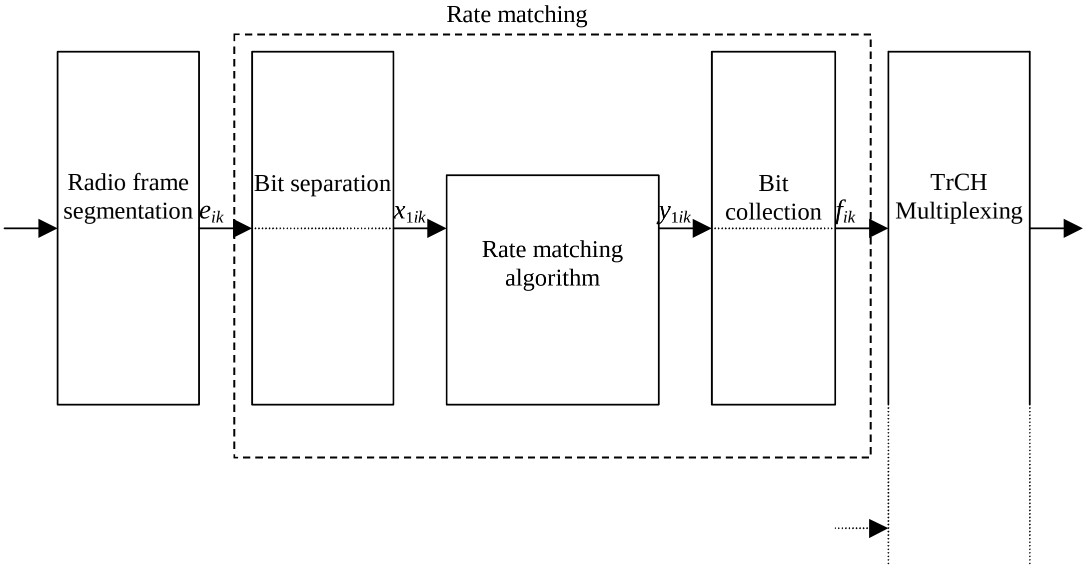


Figure 6: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. *b* indicates the three sequences defined in this section, with *b*=1 indicating the first sequence, *b* = 2 the second one, and *b* = 3 the third one.

The offsets α_b for these sequences are listed in table 6.

Table 6: TTI dependent offset needed for bit separation

TTI (ms)	α_1	α_2	α_3
10, 40	0	1	2
20, 80	0	2	1

The bit separation is different for different radio frames in the TTI. A second offset is therefore needed. The radio frame number for TrCH *i* is denoted by n_i. and the offset by β_{n_i} .

Table 7: Radio frame dependent offset needed for bit separation

TTI (ms)	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
10	0	NA	NA	NA	NA	NA	NA	NA
20	0	1	NA	NA	NA	NA	NA	NA
40	0	1	2	0	NA	NA	NA	NA
80	0	1	2	0	1	2	0	1

4.2.7.2.1 Bit separation

The bits input to the rate matching are denoted by $e_{i,1}, e_{i,2}, e_{i,3}, \ldots, e_{i,N_i}$, where *i* is the TrCH number and N_i is the number of bits input to the rate matching block. Note that the transport format combination number *j* for simplicity has been left out in the bit numbering, i.e. N_i=N_{ij}. The bits after separation are denoted by $x_{b,i,1}, x_{b,i,2}, x_{b,i,3}, \ldots, x_{b,i,X_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.2, with *b*=1 indicating the first sequence, and so forth. For all other cases *b* is defined to be 1. X_i is the number of bits in each separated bit sequence. The relation between $e_{i,k}$ and $x_{b,i,k}$ is given below.

For turbo encoded TrCHs with puncturing:

$$x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3} \qquad k = 1, 2, 3, \ldots, X_i \qquad X_i = \lfloor N_i / 3 \rfloor$$

$x_{1,i,\lfloor N_i/3 \rfloor + k} = e_{i,3\lfloor N_i/3 \rfloor + k}$ $k = 1, \ldots, N_i \bmod 3$ Note: When (N_i mod 3) = 0 this row is not needed.

$x_{2,i,k} = e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3}$ $k = 1, 2, 3, \ldots, X_i$ $X_i = \lfloor N_i/3 \rfloor$

$x_{3,i,k} = e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3}$ $k = 1, 2, 3, \ldots, X_i$ $X_i = \lfloor N_i/3 \rfloor$

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$x_{1,i,k} = e_{i,k}$ $k = 1, 2, 3, \ldots, X_i$ $X_i = N_i$

4.2.7.2.2 Bit collection

The bits $x_{b,i,k}$ are input to the rate matching algorithm described in subclause 4.2.7.3. The bits output from the rate matching algorithm are denoted $y_{b,i,1}, y_{b,i,2}, y_{b,i,3}, \ldots, y_{b,i,Y_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{b,i,1}, z_{b,i,2}, z_{b,i,3}, \ldots, z_{b,i,Y_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i,1}, f_{i,2}, f_{i,3}, \ldots, f_{i,V_i}$, where i is the TrCH number and $V_i = N_{i,j} + \Delta N_{i,j}$. The relations between $y_{b,i,k}$, $z_{b,i,k}$, and $f_{i,k}$ are given below.

For turbo encoded TrCHs with puncturing (Y_i=X_i):

$z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3} = y_{1,i,k}$ $k = 1, 2, 3, \ldots, Y_I$

$z_{i,3\lfloor N_i/3 \rfloor + k} = y_{1,i,\lfloor N_i/3 \rfloor + k}$ $k = 1, \ldots, N_i \bmod 3$ Note: When (N_i mod 3) = 0 this row is not needed.

$z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3} = y_{2,i,k}$ $k = 1, 2, 3, \ldots, Y_i$

$z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3} = y_{3,i,k}$ $k = 1, 2, 3, \ldots, Y_i$

After the bit collection, bits $z_{i,k}$ with value δ, where $\delta \notin \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$z_{i,k} = y_{1,i,k}$ $k = 1, 2, 3, \ldots, Y_i$

When repetition is used, $f_{i,k}$=$z_{i,k}$ and Y_i=V_i.

When puncturing is used, Y_i=X_i and bits $z_{i,k}$ with value δ, where $\delta \notin \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

4.2.7.3 Rate matching pattern determination

The bits input to the rate matching are denoted by $x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,X_i}$, where i is the TrCH and X_i is the parameter given in subclauses 4.2.7.1.1 and 4.2.7.1.2.

NOTE: The transport format combination number j for simplicity has been left out in the bit numbering.

The rate matching rule is as follows:

if puncturing is to be performed

```
    e = e_ini       -- initial error between current and desired puncturing ratio
    m = 1           -- index of current bit
    do while m <= X_i
        e = e – e_minus         -- update error
        if e <= 0 then          -- check if bit number m should be punctured
            set bit x_i,m to δ where δ∉{0, 1}
            e = e + e_plus      -- update error
        end if
        m = m + 1               -- next bit
    end do
else
    e = e_ini       -- initial error between current and desired puncturing ratio
    m = 1                 -- index of current bit
    do while m <= X_i
        e = e – e_minus         -- update error
        do while e <= 0         -- check if bit number m should be repeated
            repeat bit x_i,m
            e = e + e_plus  -- update error
        end do
        m = m + 1               -- next bit
    end do
end if
```

A repeated bit is placed directly after the original one.

4.2.8 TrCH multiplexing

Every 10 ms, one radio frame from each TrCH is delivered to the TrCH multiplexing. These radio frames are serially multiplexed into a coded composite transport channel (CCTrCH). If the TTI is smaller than 10ms, then no TrCH multiplexing is performed.

The bits input to the TrCH multiplexing are denoted by $f_{i,1}, f_{i,2}, f_{i,3}, \ldots, f_{i,V_i}$, where i is the TrCH id number and V_i is the number of bits in the radio frame of TrCH i. The number of TrCHs is denoted by I. The bits output from TrCH multiplexing are denoted by $h_1, h_2, h_3, \ldots, h_S$, where S is the number of bits, i.e. $S = \sum_i V_i$. The TrCH multiplexing is defined by the following relations:

$h_k = f_{1,k}$ $\quad k = 1, 2, \ldots, V_1$

$h_k = f_{2,(k-V_1)}$ $\quad k = V_1+1, V_1+2, \ldots, V_1+V_2$

$h_k = f_{3,(k-(V_1+V_2))}$ $\quad k = (V_1+V_2)+1, (V_1+V_2)+2, \ldots, (V_1+V_2)+V_3$

…

$$h_k = f_{I,(k-(V_1+V_2+\ldots+V_{I-1}))} \quad k = (V_1+V_2+\ldots+V_{I-1})+1,\ (V_1+V_2+\ldots+V_{I-1})+2,\ \ldots,\ (V_1+V_2+\ldots+V_{I-1})+V_I$$

4.2.9 Bit Scrambling

The bits output from the TrCH multiplexer are scrambled in the bit scrambler. The bits input to the bit scrambler are denoted by $h_1, h_2, h_3, \ldots, h_S$, where S is the number of bits input to the bit scrambling block equal to the total number of bits on the CCTrCH. The bits after bit scrambling are denoted $s_1, s_2, s_3, \ldots, s_S$.

Bit scrambling is defined by the following relation:

$$s_k = h_k \oplus p_k \quad k = 1,2\ldots,S$$

and p_k results from the following operation:

$$p_k = \left(\sum_{i=1}^{16} g_i \cdot p_{k-i}\right) \bmod 2\,;\ p_k = 0; k < 1\,;\ p_1 = 1\,;\ g = \{0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,1\}$$

4.2.10 Physical channel segmentation

When more than one PhCH is used, physical channel segmentation divides the bits among the different PhCHs. The bits input to the physical channel segmentation are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits input to the physical channel segmentation block. The number of PhCHs is denoted by P.

The bits after physical channel segmentation are denoted $u_{p,1}, u_{p,2}, u_{p,3}, \ldots, u_{p,U_p}$, where p is PhCH number and U_p is the in general variable number of bits in the respective radio frame for each PhCH. The relation between s_k and $u_{p,k}$ is given below.

Bits on first PhCH after physical channel segmentation:

$$u_{1,k} = s_k \quad k = 1, 2, \ldots, U_1$$

Bits on second PhCH after physical channel segmentation:

$$u_{2,k} = s_{(k+U_1)} \quad k = 1, 2, \ldots, U_2$$

…

Bits on the P^{th} PhCH after physical channel segmentation:

$$u_{P,k} = s_{(k+U_1+\ldots+U_{P-1})} \quad k = 1, 2, \ldots, U_P$$

4.2.11 2nd interleaving

The 2nd interleaving is a block interleaver and consists of bits input to a matrix with padding, the inter-column permutation for the matrix and bits output from the matrix with pruning. The 2nd interleaving can be applied jointly to all data bits transmitted during one frame, or separately within each timeslot, on which the CCTrCH is mapped. The selection of the 2nd interleaving scheme is controlled by higher layer.

4.2.11.1 Frame related 2nd interleaving

In case of frame related 2nd interleaving, the bits input to the block interleaver are denoted by $x_1, x_2, x_3, \ldots, x_U$, where U is the total number of bits after TrCH multiplexing transmitted during the respective radio frame with $S = U = \sum_p U_p$.

The relation between x_k and the bits $u_{p,k}$ in the respective physical channels is given below:

$x_k = u_{1,k}$ $k = 1, 2, \ldots, U_1$

$x_{(k+U_1)} = u_{2,k}$ $k = 1, 2, \ldots, U_2$

…

$x_{(k+U_1+\ldots+U_{P-1})} = u_{P,k}$ $k = 1, 2, \ldots, U_P$

The following steps have to be performed once for each CCTrCH:

(1) Assign C2 = 30 to be the number of columns of the matrix. The columns of the matrix are numbered 0, 1, 2, …, C2 - 1 from left to right.

(2) Determine the number of rows of the matrix, R2, by finding minimum integer R2 such that:

$U \leq \text{R2} \times \text{C2}$.

The rows of rectangular matrix are numbered 0, 1, 2, …, R2 - 1 from top to bottom.

(3) Write the input bit sequence $x_1, x_2, x_3, \ldots, x_U$ into the R2 $\times$ C2 matrix row by row starting with bit y_1 in column 0 of row 0:

$$\begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_{\text{C2}} \\ y_{(\text{C2}+1)} & y_{(\text{C2}+2)} & y_{(\text{C2}+3)} & \cdots & y_{(2\times\text{C2})} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{((\text{R2}-1)\times\text{C2}+1)} & y_{((\text{R2}-1)\times\text{C2}+2)} & y_{((\text{R2}-1)\times\text{C2}+3)} & \cdots & y_{(\text{R2}\times\text{C2})} \end{bmatrix}$$

where $y_k = x_k$ for $k = 1, 2, \ldots, U$ and if R2 $\times$ C2 > U, the dummy bits are padded such that y_k = 0 or 1 for $k = U + 1, U + 2, \ldots$, R2 $\times$ C2. These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

(4) Perform the inter-column permutation for the matrix based on the pattern $\langle \text{P}2(j) \rangle_{j\in\{0,1,\ldots,\text{C2}-1\}}$ that is shown in table 8, where $\text{P}_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y'_k.

$$\begin{bmatrix} y'_1 & y'_{(\text{R2}+1)} & y'_{(2\times\text{R2}+1)} & \cdots & y'_{((\text{C2}-1)\times\text{R2}+1)} \\ y'_2 & y'_{(\text{R2}+2)} & y'_{(2\times\text{R2}+2)} & \cdots & y'_{((\text{C2}-1)\times\text{R2}+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_{\text{R2}} & y'_{(2\times\text{R2})} & y'_{(3\times\text{R2})} & \cdots & y'_{(\text{C2}\times\text{R2})} \end{bmatrix}$$

(5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted R2 $\times$ C2 matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits y'_k that corresponds to bits y_k with $k > U$ are removed from the output. The bits after frame related 2nd interleaving are denoted by $v_1, v_2, \ldots, v_U$, where v_1 corresponds to the bit y'_k with smallest index k after pruning, v_2 to the bit y'_k with second smallest index k after pruning, and so on.

4.2.11.2 Timeslot related 2nd interleaving

In case of timeslot related 2nd interleaving, the bits input to the block interleaver are denoted by $x_{t,1}, x_{t,2}, x_{t,3}, \ldots, x_{t,U_t}$, where t refers to a certain timeslot, and U_t is the number of bits transmitted in this timeslot during the respective radio frame.

In each timeslot t the relation between $x_{t,k}$ and $u_{t,p,k}$ is given below with P_t refering to the number of physical channels within the respective timeslot:

$x_{t,k} = u_{t,1,k}$ $\quad k = 1, 2, \ldots, U_{t1}$

$x_{t,(k+U_{t1})} = u_{t,2,k}$ $\quad k = 1, 2, \ldots, U_{t2}$

…

$x_{t,(k+U_{t1}+\ldots+U_{t(P_t-1)})} = u_{t,P_t,k}$ $\quad k = 1, 2, \ldots, U_{tP_t}$

The following steps have to be performed for each timeslot t, on which the respective CCTrCH is mapped:

(1) Assign C2 = 30 to be the number of columns of the matrix. The columns of the matrix are numbered 0, 1, 2, …, C2 - 1 from left to right.

(2) Determine the number of rows of the matrix, R2, by finding minimum integer R2 such that:

$U_t \le \mathrm{R2} \times \mathrm{C2}$.

The rows of rectangular matrix are numbered 0, 1, 2, …, R2 - 1 from top to bottom.

(3) Write the input bit sequence $x_{t,1}, x_{t,2}, x_{t,3}, \ldots, x_{t,U_t}$ into the R2 × C2 matrix row by row starting with bit $y_{t,1}$ in column 0 of row 0:

$$\begin{bmatrix} y_{t,1} & y_{t,2} & y_{t,3} & \cdots & y_{t,\mathrm{C2}} \\ y_{t,(\mathrm{C2}+1)} & y_{t,(\mathrm{C2}+2)} & y_{t,(\mathrm{C2}+3)} & \cdots & y_{t,(2\times\mathrm{C2})} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{t,((\mathrm{R2}-1)\times\mathrm{C2}+1)} & y_{t,((\mathrm{R2}-1)\times\mathrm{C2}+2)} & y_{t,((\mathrm{R2}-1)\times\mathrm{C2}+3)} & \cdots & y_{t,(\mathrm{R2}\times\mathrm{C2})} \end{bmatrix}$$

where $y_{t,k} = x_{t,k}$ for k = 1, 2, …, U_t and if R2 × C2 > U_t, the dummy bits are padded such that $y_{t,k}$ = 0 or 1 for $k = U_t + 1, U_t + 2, \ldots$, R2 × C2. These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

(4) Perform the inter-column permutation for the matrix based on the pattern $\langle \mathrm{P2}(j) \rangle_{j\in\{0,1,\ldots,\mathrm{C2}-1\}}$ that is shown in table 8, where P2(j) is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by $y'_{t,k}$.

$$\begin{bmatrix} y'_{t,1} & y'_{t,(\mathrm{R2}+1)} & y'_{t,(2\times\mathrm{R2}+1)} & \cdots & y'_{t,((\mathrm{C2}-1)\times\mathrm{R2}+1)} \\ y'_{t,2} & y'_{t,(\mathrm{R2}+2)} & y'_{t,(2\times\mathrm{R2}+2)} & \cdots & y'_{t,((\mathrm{C2}-1)\times\mathrm{R2}+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y'_{t,\mathrm{R2}} & y'_{t,(2\times\mathrm{R2})} & y'_{t,(3\times\mathrm{R2})} & \cdots & y'_{t,(\mathrm{C2}\times\mathrm{R2})} \end{bmatrix}$$

(5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted R2 × C2 matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits $y'_{t,k}$ that corresponds to bits $y_{t,k}$ with $k > U_t$ are removed from the output. The bits after time slot 2nd interleaving are denoted by $v_{t,1}, v_{t,2}, \ldots, v_{t,U_t}$, where $v_{t,1}$

corresponds to the bit $y'_{t,k}$ with smallest index *k* after pruning, $v_{t,2}$ to the bit $y'_{t,k}$ with second smallest index *k* after pruning, and so on.

Table 8 Inter-column permutation pattern for 2nd interleaving

Number of Columns C2	Inter-column permutation pattern < P2(0), P2(1), …, P2(C2-1) >
30	<0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17>

4.2.12 Sub-frame segmentation for the 1.28 Mcps option

In the 1.28Mcps TDD, it is needed to add a sub-frame segmentation unit between 2nd interleaving unit and physical channel mapping unit. The operation of rate-matching guarantees that the bit streams is a even number and can be subdivided into 2 sub-frames. The transport channel multiplexing structure for uplink and downlink is shown in figure 2.

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where i is the TrCH number and X_i is the number bits. The two output bit sequences per radio frame are denoted by $y_{i,n_i1}, y_{i,n_i2}, y_{i,n_i3}, \ldots, y_{i,n_iY_i}$ where ni is the sub-frame number in current radio frame and Y_i is the number of bits per radio frame for TrCH i. The output sequences are defined as follows:

$y_{i,n_ik} = x_{i,((n_i-1)\cdot Y_i)+k}$, n_i = 1 or 2, k = 1…Y_i

where

$Y_i = (X_i / 2)$ is the number of bits per sub-frame,

x_{ik} is the k^th^ bit of the input bit sequence and

y_{i,n_ik} is the k^th^ bit of the output bit sequence corresponding to the n^th^ sub-frame

The input bit sequence to the sub-frame segmentation is denoted by $v_{(t)1}, v_{(t)2}, \ldots, v_{(t)U_{(t)}}$, $x_{ik} = v_{(t)k}$ and $X_i = U_{(t)}$.

The output bit sequence corresponding to subframe n_i is denoted by $g_{p1}, g_{p2}, \ldots, g_{pU_p}$, where p is the PhCH number and U_p is the number of bits in one subframe for the respective PhCH. Hence, $g_{pk} = y_{i,n_ik}$ and $U_p = Y_i$.

4.2.13 Physical channel mapping

4.2.13.1 Physical channel mapping for the 3.84 Mcps option

The PhCH for both uplink and downlink is defined in [6]. The bits after physical channel mapping are denoted by $w_{p,1}, w_{p,2}, \ldots, w_{p,U_p}$, where *p* is the PhCH number corresponding to the sequence number $1 \le p \le P_{max}$ of this physical channel as detailed below, and U_p is the number of bits in one radio frame for the respective PhCH. The bits $w_{p,k}$ are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to *k*.

The physical channel sequence number *p* are to be allocated by the physical layer in ascending order of the timeslots in which they appear. If more than one physical channel appears in a timeslot, they shall be allocated the sequence number in order of the timeslot first and then of their channelisation codes. The channelisation codes shall be ordered in ascending order of the spreading factor (*Q*) and then channelisation code index (*k*), as shown in [9].

The mapping of the bits $v_{(t),1}, v_{(t),2}, \ldots, v_{(t),U_{(t)}}$ is performed like block interleaving, writing the bits into columns, but a PhCH with an odd number is filled in forward order, were as a PhCH with an even number is filled in reverse order.

The mapping scheme, as described in the following subclause, shall be applied individually for each timeslot *t* used in the current frame. Therefore, the bits $v_{t,1}, v_{t,2}, \ldots, v_{t,U_t}$ are assigned to the bits of the physical channels $w_{t,1,1\ldots U_{t1}}, w_{t,2,1\ldots U_{t2}}, \ldots, w_{t,P_t,1\ldots U_{tP_t}}$ in each timeslot.

In uplink there are at most two codes allocated (P≤2). If there is only one code, the same mapping as for downlink is applied. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. For the number of consecutive bits to assign per code bs_k the following rule is applied:

if

SF1 >= SF2 then $bs_1 = 1$; $bs_2 =$ SF1/SF2 ;

else

SF2 > SF1 then $bs_1 =$ SF2/SF1; $bs_2 = 1$;

end if

In the downlink case bs_p is 1 for all physical channels.

4.2.13.1.1 Mapping scheme

Notation used in this subclause:

P_t: number of physical channels for timeslot t , P_t = 1..2 for uplink ; P_t = 1...16 for downlink

$U_{t,p}$: capacity in bits for the physical channel p in timeslot t

$U_{t.}$: total number of bits to be assigned for timeslot t

bs_p: number of consecutive bits to assign per code

for downlink all $bs_p = 1$

for uplink if SF1 >= SF2 then $bs_1 = 1$; $bs_2 =$ SF1/SF2 ;

if SF2 > SF1 then $bs_1 =$ SF2/SF1; $bs_2 = 1$;

fb_p: number of already written bits for each code

pos: intermediate calculation variable

```
for p=1 to P_t                      -- reset number of already written bits for every physical channel
    fb_p = 0
end for
p = 1                               -- start with PhCH #1
for k=1 to U_t.
    do while (fb_p == U_t,p)          -- physical channel filled up already ?
        p = (p mod P_t) + 1;
    end do
    if (p mod 2) == 0
        pos = U_t,p - fb_p            -- reverse order
    else
        pos = fb_p + 1                -- forward order
```

endif

$w_{t,p,pos} = v_{t,k}$ -- assignment

$fb_p = fb_p + 1$ -- Increment number of already written bits

if (fb_p mod bs_p) == 0 -- Conditional change to the next physical channel

p = (p mod P_t) + 1;

end if

end for

4.2.13.2 Physical channel mapping for the 1.28 Mcps option

The bit streams from the sub-frame segmentation unit are mapped onto code channels of time slots in sub-frames.

The bits after physical channel mapping are denoted by $w_{p1}, w_{p2}, \ldots, w_{pU_p}$, where p is the PhCH number and Up is the number of bits in one sub-frame for the respective PhCH. The bits wpk are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to k.

The mapping of the bits $g_{p1}, g_{p2}, \ldots, g_{pU_p}$ is performed like block interleaving, writing the bits into columns, but a PhCH with an odd number is filled in forward order, were as a PhCH with an even number is filled in reverse order.

The mapping scheme, as described in the following subclause, shall be applied individually for each timeslot t used in the current subframe. Therefore, the bits $g_{p1}, g_{p2}, \ldots, g_{pU_p}$ are assigned to the bits of the physical channels $w_{t1,1\ldots U_{t1}}, w_{t2,1\ldots U_{t2}}, \ldots, w_{tP_t,1\ldots U_{tP_t}}$ in each timeslot.

In uplink there are at most two codes allocated (P≤2). If there is only one code, the same mapping as for downlink is applied. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. For the number of consecutive bits to assign per code bsk the following rule is applied:

if

SF1 >= SF2 then bs_1 = 1 ; bs_2 = SF1/SF2 ;

else

SF2 > SF1 then bs_1 = SF2/SF1; bs_2 = 1 ;

end if

In the downlink case bs_p is 1 for all physical channels.

4.2.13.2.1 Mapping scheme

Notation used in this subclause:

P_t: number of physical channels for timeslot t , P_t = 1..2 for uplink ; P_t = 1...16 for downlink

U_{tp}: capacity in bits for the physical channel p in timeslot t

U_t.: total number of bits to be assigned for timeslot t

bs_p: number of consecutive bits to assign per code

for downlink all bs_p = 1

for uplink if SF1 >= SF2 then bs_1 = 1 ; bs_2 = SF1/SF2 ;

if SF2 > SF1 then bs_1 = SF2/SF1; bs_2 = 1 ;

fb_p: number of already written bits for each code

pos: intermediate calculation variable

```
for p=1 to P_t                -- reset number of already written bits for every physical channel
fb_p = 0
end for
p = 1                         -- start with PhCH #1
for k=1 to U_t.
do while (fb_p == U_t,p)      -- physical channel filled up already ?
p = (p mod P_t) + 1 ;
end do
if (p mod 2) == 0
pos = U_t,p - fb_p            -- reverse order
else
pos = fb_p + 1                -- forward order
end if
w_tp,pos = g_t,k              -- assignment
fb_p = fb_p + 1               -- Increment number of already written bits
If (fb_p mod bs_p) == 0       -- Conditional change to the next physical channel
p = (p mod P t) + 1 ;
end if
end for
```

4.2.14 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels

Different transport channels can be encoded and multiplexed together into one Coded Composite Transport Channel (CCTrCH). The following rules shall apply to the different transport channels which are part of the same CCTrCH:

1) Transport channels multiplexed into one CCTrCh shall have co-ordinated timings. When the TFCS of a CCTrCH is changed because one or more transport channels are added to the CCTrCH or reconfigured within the CCTrCH, or removed from the CCTrCH, the change may only be made at the start of a radio frame with CFN fulfilling the relation

   $CFN \bmod F_{max} = 0$,

   where F_{max} denotes the maximum number of radio frames within the transmission time intervals of all transport channels which are multiplexed into the same CCTrCH, including any transport channels *i* which are added reconfigured or have been removed, and CFN denotes the connection frame number of the first radio frame of the changed CCTrCH.

   After addition or reconfiguration of a transport channel *i* within a CCTrCH, the TTI of transport channel *i* may only start in radio frames with CFN fulfilling the relation

   $CFN_i \bmod F_i = 0$.

2) Different CCTrCHs cannot be mapped onto the same physical channel.

3) One CCTrCH shall be mapped onto one or several physical channels.

4) Dedicated Transport channels and common transport channels cannot be multiplexed into the same CCTrCH.

5) For the common transport channels, only the FACH and PCH may belong to the same CCTrCH.

6) Each CCTrCH carrying a BCH shall carry only one BCH and shall not carry any other Transport Channel.

7) Each CCTrCH carrying a RACH shall carry only one RACH and shall not carry any other Transport Channel.

Hence, there are two types of CCTrCH.

CCTrCH of dedicated type, corresponding to the result of coding and multiplexing of one or several DCH.

CCTrCH of common type, corresponding to the result of the coding and multiplexing of a common channel, i.e. RACH and USCH in the uplink and DSCH, BCH, FACH or PCH in the downlink, respectively.

Transmission of TFCI is possible for CCTrCH containing Transport Channels of:

- dedicated type;
- USCH type;
- DSCH type;
- FACH and/or PCH type.

4.2.14.1 Allowed CCTrCH combinations for one UE

4.2.14.1.1 Allowed CCTrCH combinations on the uplink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

1) several CCTrCH of dedicated type;

2) several CCTrCH of common type.

4.2.14.1.2 Allowed CCTrCH combinations on the downlink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

3) several CCTrCH of dedicated type;

4) several CCTrCH of common type.

4.2.15 Transport format detection

Transport format detection can be performed both with and without Transport Format Combination Indicator (TFCI). If a TFCI is transmitted, the receiver detects the transport format combination from the TFCI. When no TFCI is transmitted, so called blind transport format detection may be used, i.e. the receiver side uses the possible transport format combinations as a priori information.

4.2.15.1 Blind transport format detection

Blind Transport Format Detection is optional both in the UE and the UTRAN. Therefore, for all CCTrCH a TFCI shall be transmitted, including the possibility of a TFCI code word length zero, if only one TFC is defined.

4.2.15.2 Explicit transport format detection based on TFCI

4.2.15.2.1 Transport Format Combination Indicator (TFCI)

The Transport Format Combination Indicator (TFCI) informs the receiver of the transport format combination of the CCTrCHs. As soon as the TFCI is detected, the transport format combination, and hence the individual transport channels' transport formats are known, and decoding of the transport channels can be performed.

4.3 Coding for layer 1 control for the 3.84 Mcps option

4.3.1 Coding of transport format combination indicator (TFCI)

Encoding of the TFCI depends on its length. If there are 6-10 bits of TFCI the channel encoding is done as described in subclause 4.3.1.1. Also specific coding of less than 6 bits is possible as explained in subclause 4.3.1.2.

4.3.1.1 Coding of long TFCI lengths

The TFCI is encoded using a (32, 10) sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 7.

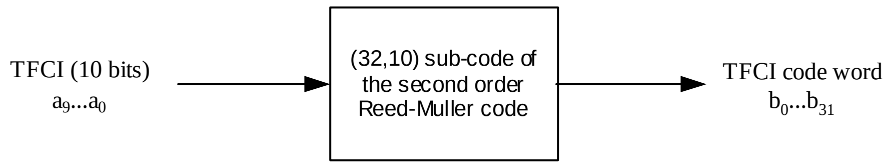


Figure 7: Channel coding of the TFCI bits

TFCI is encoded by the (32,10) sub-code of second order Reed-Muller code. The code words of the (32,10) sub-code of second order Reed-Muller code are linear combination of some among 10 basis sequences. The basis sequences are as follows in table 9.

Table 9: Basis sequences for (32,10) TFCI code

I	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{I,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$
0	1	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0
2	1	1	0	0	0	1	0	0	0	1
3	0	0	1	0	0	1	1	0	1	1
4	1	0	1	0	0	1	0	0	0	1
5	0	1	1	0	0	1	0	0	1	0
6	1	1	1	0	0	1	0	1	0	0
7	0	0	0	1	0	1	0	1	1	0
8	1	0	0	1	0	1	1	1	1	0
9	0	1	0	1	0	1	1	0	1	1
10	1	1	0	1	0	1	0	0	1	1
11	0	0	1	1	0	1	0	1	1	0
12	1	0	1	1	0	1	0	1	0	1
13	0	1	1	1	0	1	1	0	0	1
14	1	1	1	1	0	1	1	1	1	1
15	1	0	0	0	1	1	1	1	0	0
16	0	1	0	0	1	1	1	1	0	1
17	1	1	0	0	1	1	1	0	1	0
18	0	0	1	0	1	1	0	1	1	1
19	1	0	1	0	1	1	0	1	0	1
20	0	1	1	0	1	1	0	0	1	1
21	1	1	1	0	1	1	0	1	1	1
22	0	0	0	1	1	1	0	1	0	0
23	1	0	0	1	1	1	1	1	0	1
24	0	1	0	1	1	1	1	0	1	0
25	1	1	0	1	1	1	1	0	0	1
26	0	0	1	1	1	1	0	0	1	0
27	1	0	1	1	1	1	1	1	0	0
28	0	1	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	0	0	0	0	0	1	0	0	0	0
31	0	0	0	0	1	1	1	0	0	0

The TFCI bits a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9 (where a_0 is LSB and a_9 is MSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

The output TFCI code word bits b_i are given by:

$$b_i = \sum_{n=0}^{9} (a_n \times M_{i,n}) \bmod 2$$

where $i = 0,\ldots,31$. $N_{\text{TFCI code word}} = 32$.

4.3.1.2 Coding of short TFCI lengths

4.3.1.2.1 Coding very short TFCIs by repetition

If the number of TFCI bits is 1 or 2, then repetition will be used for coding. In this case each bit is repeated to a total of 4 times giving 4-bit transmission ($N_{\text{TFCI code word}}$ =4) for a single TFCI bit and 8-bit transmission ($N_{\text{TFCI code word}}$ =8) for 2 TFCI bits. The TFCI bit(s) b_0 (or b_0 and b_1 where b_0 is the LSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame. In the case of two TFCI bits denoted b_0 and b_1 the TFCI code word shall be { $b_0, b_1, b_0, b_1, b_0, b_1, b_0, b_1$ }.

4.3.1.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits is in the range 3 to 5 the TFCI is encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 8.

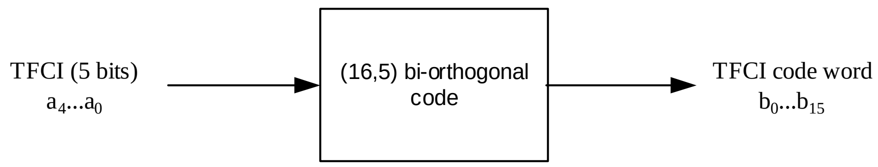


Figure 8: Channel coding of short length TFCI bits

The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 10.

Table 10: Basis sequences for (16,5) TFCI code

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$
0	1	0	0	0	1
1	0	1	0	0	1
2	1	1	0	0	1
3	0	0	1	0	1
4	1	0	1	0	1
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	0	1	1
8	1	0	0	1	1
9	0	1	0	1	1
10	1	1	0	1	1
11	0	0	1	1	1
12	1	0	1	1	1
13	0	1	1	1	1
14	1	1	1	1	1
15	0	0	0	0	1

The TFCI bits a_0 , a_1 , a_2 , a_3 , a_4 (where a_0 is LSB and a_4 is MSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

The output code word bits b_j are given by:

$$b_i = \sum_{n=0}^{4} (a_n \times M_{i,n}) \bmod 2$$

where $i = 0,\ldots,15$. $N_{\text{TFCI code word}} = 16$.

4.3.1.3 Mapping of TFCI code word

The mapping of the TFCI code word to the TFCI bit positions in a timeslot shall be as follows.

Denote the number of bits in the TFCI code word by $N_{\text{TFCI code word}}$, denote the TFCI code word bits by b_k where k=0… $N_{\text{TFCI code word}}$ -1.

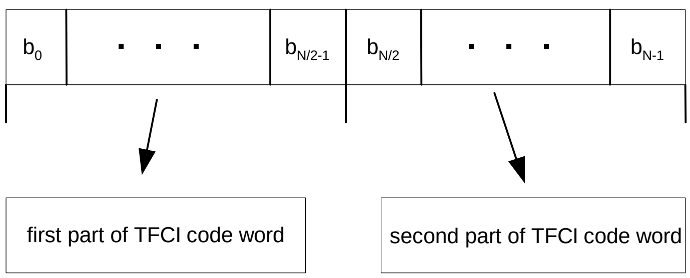


Figure 9: Mapping of TFCI code word bits to timeslot

The locations of the first and second parts of the TFCI code word in the timeslot is defined in [7].

If the shortest transmission time interval of any constituent TrCH is at least 20 ms the successive TFCI code words in the frames in the TTI shall be identical. If TFCI is transmitted on multiple timeslots in a frame each timeslot shall have the same TFCI code word.

4.3.2 Coding and Bit Scrambling of the Paging Indicator

The paging indicator P_q, $q = 0, ..., N_{PI}-1$, $P_q \in \{0, 1\}$ is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI, calculated by higher layers, and the associated paging indicator P_q. The length L_{PI} of the paging indicator is $L_{PI}=2$, $L_{PI}=4$ or $L_{PI}=8$ symbols. $N_{PIB} = 2*N_{PI}*L_{PI}$ bits are used for the paging indicator transmission in one radio frame. The mapping of the paging indicators to the bits e_i, $i = 1, ..., N_{PIB}$ is shown in table 11.

Table 11: Mapping of the paging indicator

P_q	Bits $\{e_{2Lpi*q+1}, e_{2Lpi*q+2}, \dots, e_{2Lpi*(q+1)}\}$	Meaning
0	{0, 0, ..., 0}	There is no necessity to receive the PCH
1	{1, 1, ..., 1}	There is the necessity to receive the PCH

If the number S of bits in one radio frame available for the PICH is bigger than the number N_{PIB} of bits used for the transmission of paging indicators, the sequence $e = \{e_1, e_2, ..., e_{NPIB}\}$ is extended by $S-N_{PIB}$ bits that are set to zero, resulting in a sequence $h = \{h_1, h_2, ..., h_S\}$:

$$h_k = e_k, \quad k = 1, ..., N_{PIB}$$
$$h_k = 0, \quad k = N_{PIB} + 1, ..., S$$

The bits h_k , $k = 1, ..., S$ on the PICH then undergo bit scrambling as defined in section 4.2.9.

The bits s_k, $k = 1, ..., S$ output from the bit scrambler are then transmitted over the air as shown in [7].

4.4 Coding for layer 1 control for the 1.28 Mcps option

4.4.1 Coding of transport format combination indicator (TFCI) for QPSK

The coding of TFCI for 1.28Mcps TDD is same as that of 3.84Mcps TDD.cf.[4.3.1 'Coding of transport format combination indicator'].

4.4.1.1 Mapping of TFCI code word

Denote the number of bits in the TFCI code word by $N_{TFCI\ code\ word}$, and denote the TFCI code word bits by b_k, where $k = 0, \dots, N_{TFCI\ code\ word} - 1$

When the number of bits in the TFCI code word is 8, 16, 32, the mapping of the TFCI code word to the TFCI bit positions shall be as follows:

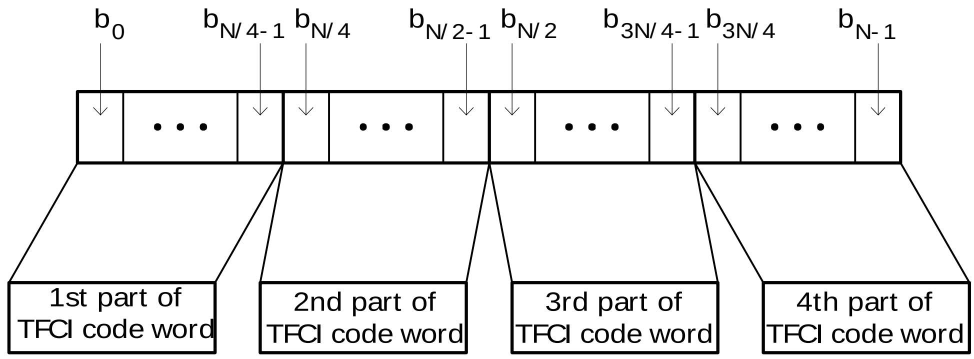


Figure 10: Mapping of TFCI code word bits to TFCI position in 1.28 Mcps TDD option, where N = $N_{TFCI\ code\ word}$.

When the number of bits of the TFCI code word is 4 , then the TFCI code word is equally divided into two parts for the consecutive two subframe and mapped onto the end of the first data field in each of the consecutive subframes.The mapping for $N_{TFCI\ code\ word}$ =4 is shown in figure 11:

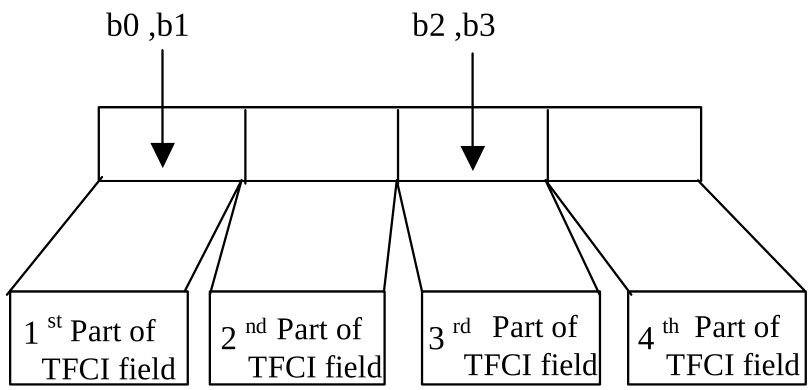


Figure 11: Mapping of TFCI code word bits to TFCI position in 1.28 Mcps TDD option, when $N_{TFCI\ code\ word.}$=4

The location of the 1st to 4th parts of the TFCI code word in the timeslot is defined in [7].

If the shortest transmission time interval of any constituent TrCH is at least 20 ms, then successive TFCI code words in the frames within the TTI shall be identical. If a TFCI is transmitted on multiple timeslots in a frame each timeslot shall have the same TFCI code word.

4.4.2 Coding of transport format combination indicator (TFCI) for 8PSK

Encoding of TFCI bits depends on the number of them and the modulation in use. When 2 Mcps service is transmitted, 8PSK modulation is applied in 1.28 Mcps TDD option. The encoding scheme for TFCI when the number of bits are 6 – 10, and less than 6 bits is described in section 4.4.2.1 and 4.4.2.2, respectively.

4.4.2.1 Coding of long TFCI lengths

When the number of TFCI bits is 6 – 10, the TFCI bits are encoded by using a (64,10) sub-code of the second order Reed-Muller code, then 16 bits out of 64 bits are punctured (Puncturing positions are 0, 4, 8, 13, 16, 20, 27, 31, 34, 38, 41, 44, 50, 54, 57, 61st bits). The coding procedure is shown in Figure 12.

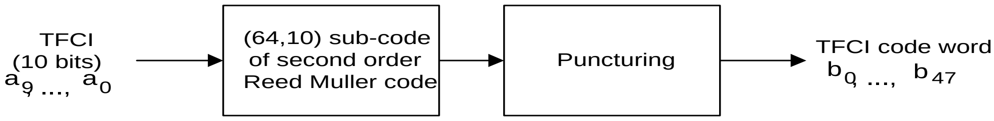


Figure 12: Channel coding of long TFCI bits for 8PSK

The code words of the punctured (48,10) sub-code of the second order Reed-Muller codes are linear combination of 10 basis sequences. The basis sequences are shown in Table 12.

Table 12: Basis sequences for (48,10) TFCI code

I	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{I,4}$	$M_{i,5}$	$M_{i,6}$	$M_{I,7}$	$M_{I,8}$	$M_{i,9}$
0	1	0	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	1	0	0
2	1	1	0	0	0	0	1	1	0	1
3	1	0	1	0	0	0	1	1	1	0
4	0	1	1	0	0	0	1	0	1	0
5	1	1	1	0	0	0	1	1	1	0
6	1	0	0	1	0	0	1	1	1	1
7	0	1	0	1	0	0	1	1	0	1
8	1	1	0	1	0	0	1	0	1	0
9	0	0	1	1	0	0	1	1	0	0
10	0	1	1	1	0	0	1	1	0	1
11	1	1	1	1	0	0	1	1	1	1
12	1	0	0	0	1	0	1	0	1	1
13	0	1	0	0	1	0	1	1	1	0
14	1	1	0	0	1	0	1	0	0	1
15	1	0	1	0	1	0	1	0	1	1
16	0	1	1	0	1	0	1	1	0	0
17	1	1	1	0	1	0	1	1	1	0
18	0	0	0	1	1	0	1	0	0	1
19	1	0	0	1	1	0	1	0	1	1
20	0	1	0	1	1	0	1	0	1	0
21	0	0	1	1	1	0	1	0	1	0
22	1	0	1	1	1	0	1	1	0	1
23	0	1	1	1	1	0	1	1	1	0
24	0	0	0	0	0	1	1	1	0	1
25	1	0	0	0	0	1	1	1	1	0
26	1	1	0	0	0	1	1	1	1	1
27	0	0	1	0	0	1	1	0	1	1
28	1	0	1	0	0	1	1	1	0	1
29	1	1	1	0	0	1	1	0	1	1
30	0	0	0	1	0	1	1	0	0	1
31	0	1	0	1	0	1	1	0	0	1
32	1	1	0	1	0	1	1	1	1	1
33	1	0	1	1	0	1	1	0	0	1
34	0	1	1	1	0	1	1	1	1	0
35	1	1	1	1	0	1	1	1	0	1
36	0	0	0	0	1	1	1	1	1	0
37	1	0	0	0	1	1	1	0	1	1
38	1	1	0	0	1	1	1	1	1	1
39	0	0	1	0	1	1	1	1	0	0
40	1	0	1	0	1	1	1	1	0	0
41	1	1	1	0	1	1	1	1	1	1
42	0	0	0	1	1	1	1	1	1	1
43	0	1	0	1	1	1	1	0	1	0
44	1	1	0	1	1	1	1	0	1	0
45	0	0	1	1	1	1	1	0	1	1
46	0	1	1	1	1	1	1	0	0	1
47	1	1	1	1	1	1	1	1	0	0

Let's define the TFCI bits as a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9, where a_0 is the LSB and a_9 is the MSB. The TFCI bits shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

The output TFCI code word bits b_i are given by:

$$b_i = \sum_{n=0}^{9} (a_n \times M_{i,n}) \bmod 2$$

where i=0…47. $N_{TFCI\ code\ word}$ =48.

4.4.2.2 Coding of short TFCI lengths

4.4.2.2.1 Coding very short TFCIs by repetition

When the number of TFCI bits is 1 or 2, then repetition will be used for the coding. In this case, each bit is repeated to a total of 6 times giving 6-bit transmission ($N_{TFCI\ code\ word}$ = 6) for a single TFCI bit and 12-bit transmission ($N_{TFCI\ code\ word}$ = 12) for 2 TFCI bits. For a single TFCI bit b_0, the TFCI code word shall be {b_0, b_0, b_0, b_0, b_0, b_0}. For two TFCI bits b_0 and b_1, the TFCI code word shall be {b_0, b_1, b_0, b_1, b_0, b_1, b_0, b_1, b_0, b_1, b_0, b_1}.

4.4.2.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits is in the range of 3 to 5, the TFCI bits are encoded using a (32,5) first order Reed-Muller code, then 8 bits out of 32 bits are punctured (Puncturing positions are 0, 1, 2, 3, 4, 5, 6, 7th bits). The coding procedure is shown in Figure 13.

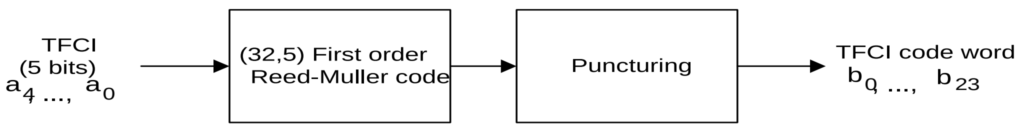


Figure 13: Channel coding of short TFCI bits for 8PSK

The code words of the punctured (32,5) first order Reed-Muller codes are linear combination of 5 basis sequences shown in Table 13.

Table 13: Basis sequences for (24,5) TFCI code

I	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$
0	0	0	0	1	0
1	1	0	0	1	0
2	0	1	0	1	0
3	1	1	0	1	0
4	0	0	1	1	0
5	1	0	1	1	0
6	0	1	1	1	0
7	1	1	1	1	0
8	0	0	0	0	1
9	1	0	0	0	1
10	0	1	0	0	1
11	1	1	0	0	1
12	0	0	1	0	1
13	1	0	1	0	1
14	0	1	1	0	1
15	1	1	1	0	1
16	0	0	0	1	1
17	1	0	0	1	1
18	0	1	0	1	1
19	1	1	0	1	1
20	0	0	1	1	1
21	1	0	1	1	1
22	0	1	1	1	1
23	1	1	1	1	1

Let's define the TFCI bits as a_0 , a_1 , a_2 , a_3 , a_4, where a_0 is the LSB and a_4 is the MSB. The TFCI bits shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

The output code word bits b_i are given by:

$$b_i = \sum_{n=0}^{4} (a_n \times M_{i,n}) \bmod 2$$

where i=0…23. $N_{\text{TFCI code word}}$ =24.

4.4.2.3 Mapping of TFCI code word

Denote the number of bits in the TFCI code word by $N_{\text{TFCI code word}}$, and denote the TFCI code word bits by b_k, where k = 0, …, $N_{\text{TFCI code word}}$ -1.

When the number of bits in the TFCI code word is 12, 24 or 48, the mapping of the TFCI code word to the TFCI bit positions in a time slot shall be as follows.

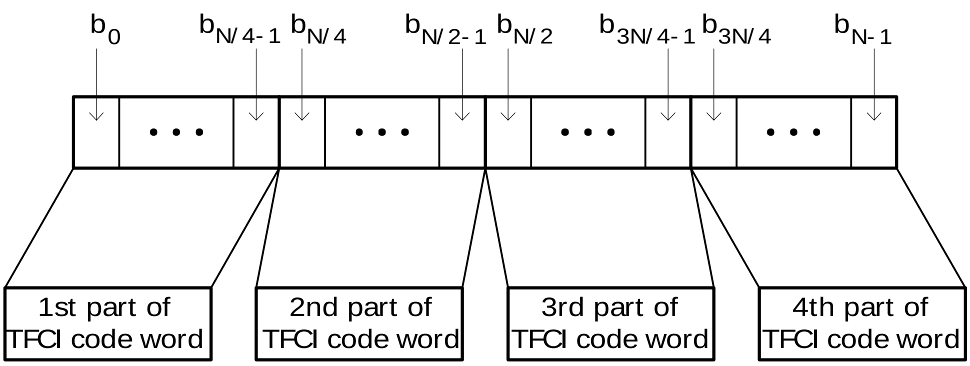


Figure 14: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option, where N = $N_{\text{TFCI code word}}$.

When the number of bits in the TFCI code word is 6, the TFCI code word is equally divided into two parts for the consecutive two sub-frames and mapped onto the first data field in each of the consecutive sub-frames. The mapping of the TFCI code word to the TFCI bit positions in a time slot shall be as shown in figure 15.

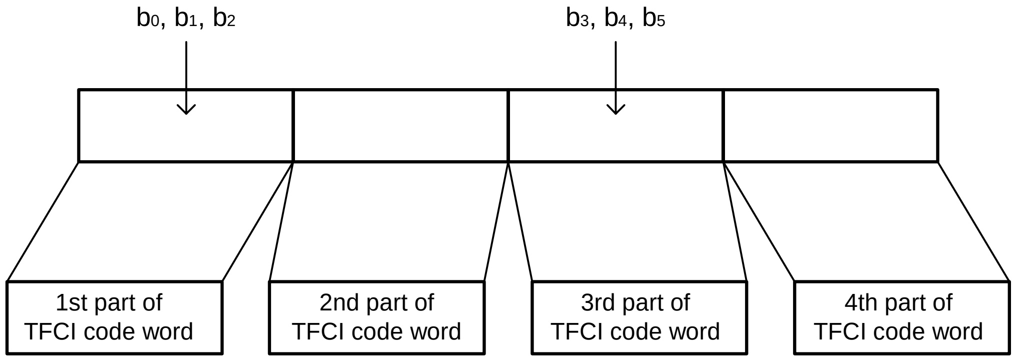


Figure 15: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option when $N_{TFCI\ code\ word}$ = 6

The location of the 1st to 4th parts of the TFCI code word in the timeslot is defined in [7].

4.4.3 Coding and Bit Scrambling of the Paging Indicator

The paging indicator P_q, q = 0, ..., N_{PI}-1, $P_q \in \{0, 1\}$ is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI, calculated by higher layers, and the associated paging indicator P_q. The length L_{PI} of the paging indicator is L_{PI}=2, L_{PI}=4 or L_{PI}=8 symbols. $N_{PIB} = 2*N_{PI}*L_{PI}$ bits are used for the paging indicator transmission in one radio frame. The mapping of the paging indicators to the bits e_i, i = 1, ..., N_{PIB} is shown in table 14.

Table 14: Mapping of the paging indicator

P_q	Bits $\{e_{2L_{PI}*q+1}, e_{2L_{PI}*q+2}, \ldots, e_{2L_{PI}*(q+1)}\}$	Meaning
0	{0, 0, ..., 0}	There is no necessity to receive the PCH
1	{1, 1, ..., 1}	There is the necessity to receive the PCH

If the number S of bits in one radio frame available for the PICH is bigger than the number N_{PIB} of bits used for the transmission of paging indicators, the sequence $e = \{e_1, e_2, ..., e_{NPIB}\}$ is extended by S-N_{PIB} bits that are set to zero, resulting in a sequence $h = \{h_1, h_2, ..., h_S\}$:

$$h_k = e_k, \quad k = 1, ..., N_{PIB}$$
$$h_k = 0, \quad k = N_{PIB}, ..., S$$

The bits h_k , k = 1, ..., S on the PICH then undergo bit scrambling as defined in section 4.2.9.

The bits s_k, k = 1, ..., S output from the bit scrambler are then transmitted over the air as shown in [7].

4.4.4 Coding of the Fast Physical Access Channel (FPACH) information bits

The FPACH burst is composed by 32 information bits which are block coded and convolutional coded, and then delivered in one sub-frame as follows:

1. The 32 information bits are protected by 8 parity bits for error detection as described in sub-clause 4.2.1.1.

2. Convolutional code with constraint length 9 and coding rate ½ is applied as described in sub-clause 4.2.3.1. The size of data block c(k) after convolutional encoder is 96 bits.

3. To adjust the size of the data block c(k) to the size of the FPACH burst, 8 bits are punctured as described in sub-clause 4.2.7 with the following clarifications:

- $N_{i;j}$ =96 is the number of bits in a radio sub-frame before rate matching
- $\Delta N_{i,,j}$ = -8 is the number of bits to punctured in a radio sub-frame
- e_{ini} = a x N_{ij}

The 88 bits after rate matching are then delivered to the intra-frame interleaving.

4. The bits in input to the interleaving unit are denoted as {x(0), …, x(87)}. The coded bits are block rectangular interleaved according to the following rule: the input is written row by row, the output is read column by column.

$$\begin{bmatrix} x(0) & x(1) & x(2) & \dots & x(7) \\ x(8) & x(9) & x(10) & \dots & x(15) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x(80) & x(81) & x(82) & \dots & x(87) \end{bmatrix}$$

Hence, the interleaved sequence is denoted by y (i) and are given by:

y(0), y(1), …, y(87)=x(0), x(8), …,x(80),x(1), …, x(87).

4.5 Coding for HS-DSCH

Figure 16 illustrates the overall concept of transport-channel coding and multiplexing for HS-DSCH. Data arrives to the coding/multiplexing unit in form of one transport block once every transmission time interval. The transmission time interval is 5 ms for 1.28Mcps TDD and 10ms for 3.84 Mcps TDD.

The following coding/multiplexing steps for HS-DSCH can be identified:

- add CRC to each transmission time interval (see subclause 4.5.1);
- code block segmentation (see subclause 4.5.2);
- channel coding (see subclause 4.5.3);
- hybrid ARQ (see subclause 4.5.4);
- bit scrambling (see subclause 4.5.5);
- physical channel segmentation (see subclause 4.5.6);
- interleaving for HS-DSCH (see subclauses 4.5.7);
- mapping to physical channels (see subclause 4.5.8);
- constellation re-arrangement for 16QAM (see subclause 4.5.9).

The coding steps for HS-DSCH are shown in figure 16.

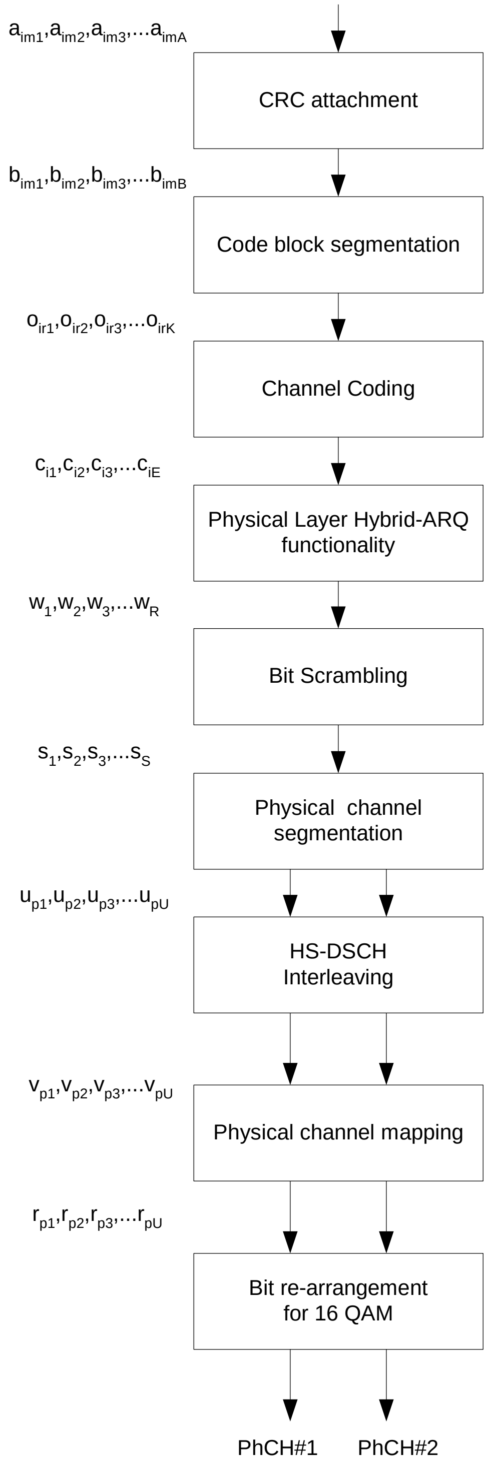


Figure 16. Transport channel multiplexing structure for HS-DSCH

In the following the number of transport blocks is always one. When referencing non HS-DSCH formulae which are used in correspondence with HS-DSCH formulae the convention is used that transport block subscripts may be omitted (e.g. X_i when i is always 1 may be written X).

4.5.1 CRC attachment for HS-DSCH

A CRC of size 24 bits is calculated and added per HS-DSCH TTI. The CRC polynomial is defined in 4.2.1.1 with the following specific parameters: i=1, L_1 = 24 bits.

4.5.2 Code block segmentation for HS-DSCH

Code block segmentation for the HS-DSCH transport channel shall be done with the general method described in 4.2.2.2 above with the following specific parameters.

There will only be one transport block, i=1. The bits b_{im1}, b_{im2}, b_{im3},…b_{imB} input to the block are mapped to the bits x_{i1}, x_{i2}, x_{i3},…x_{iX_i} directly. It follows that X_1 = B. Note that the bits x referenced here refer only to the internals of the code block segmentation function.

The value of Z = 5114 for turbo coding shall be used.

4.5.3 Channel coding for HS-DSCH

Channel coding for the HS-DSCH transport channel shall be done with the general method described in 4.2.3.2 above with the following specific parameters.

There will be a maximum of one transport block, i=1. The rate 1/3 turbo coding shall be used.

4.5.4 Hybrid ARQ for HS-DSCH

The hybrid ARQ functionality matches the number of bits at the output of the channel coder to the total number of bits of the HS-PDSCH set to which the HS-DSCH is mapped. The hybrid ARQ functionality is controlled by the redundancy version (RV) parameters. The exact set of bits at the output of the hybrid ARQ functionality depends on the number of input bits, the number of output bits, and the RV parameters.

The hybrid ARQ functionality consists of two rate-matching stages and a buffer as shown in the figure below.

The first rate matching stage matches the number of input bits to the virtual IR buffer, information about which is provided by higher layers. Note that, if the number of input bits does not exceed the virtual IR buffering capability, the first rate-matching stage is transparent.

The second rate matching stage matches the number of bits after first rate matching stage to the number of physical channel bits available in the HS-PDSCH set in the TTI.

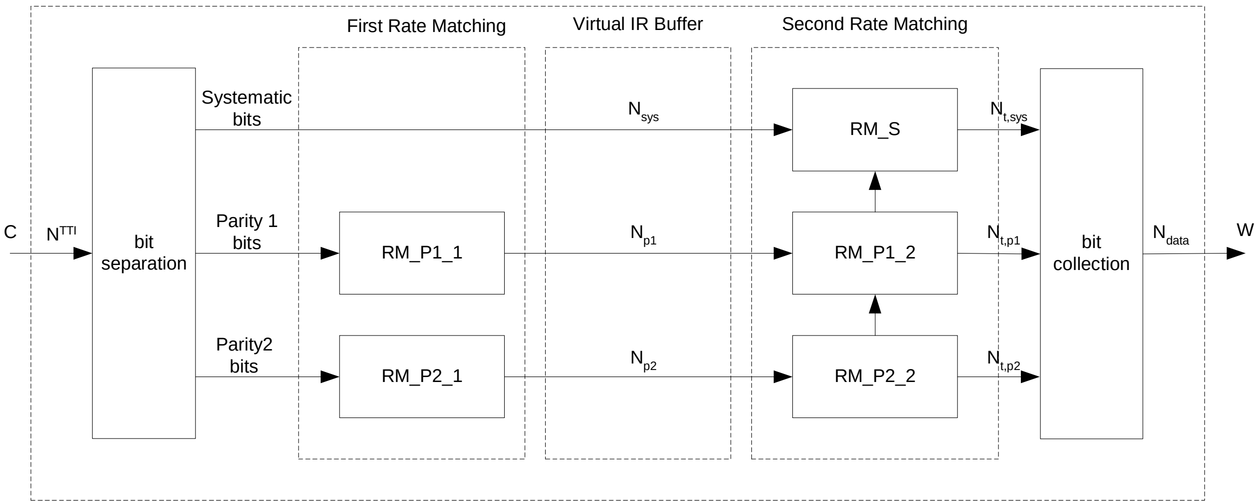


Figure 17: HS-DSCH hybrid ARQ functionality

4.5.4.1 HARQ bit separation

The HARQ bit separation function shall be performed in the same way as bit separation for turbo encoded TrCHs in 4.2.7.2 above.

4.5.4.2 HARQ First Rate Matching Stage

HARQ first stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.1.2 above with the following specific parameters.

The maximum number of soft bits available in the virtual IR buffer is N_{IR} which is signalled from higher layers for each HARQ process. The number of coded bits in a TTI before rate matching is N^{TTI} this is deduced from information signalled from higher layers and parameters signalled on the HS-SCCH for each TTI. Note that HARQ processing and physical layer storage occurs independently for each HARQ process currently active.

If N_{IR} is greater than or equal to N^{TTI} (i.e. all coded bits of the corresponding TTI can be stored) the first rate matching stage shall be transparent. This can, for example, be achieved by setting $e_{minus} = 0$. Note that no repetition is performed.

If N_{IR} is smaller than N^{TTI} the parity bit streams are punctured as in 4.2.7.1.2 above by setting the rate matching parameter $\Delta N_{il}^{TTI} = N_{IR} - N^{TTI}$ where the subscripts i and l refer to transport channel and transport format in the referenced sub-clause. Note the negative value is expected when the rate matching implements puncturing. Bits selected for puncturing which appear as δ in the algorithm in 4.2.7 above shall be discarded and not counted in the totals for the streams through the virtual IR buffer.

4.5.4.3 HARQ Second Rate Matching Stage

HARQ second stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.3 above with the following specific parameters.

The parameters of the second rate matching stage depend on the value of the RV parameters s and r. The parameter s can take the value 0 or 1 to distinguish self-decodable (s = 1) and non self-decodable (s = 0) transmissions. The parameter r (range 0 to r_{max}) changes the initial error variable e_{ini} in the case of puncturing. In case of repetition both parameters *r* and *s* change the initial error variable e_{ini}. The parameters X, e_{plus} and e_{minus} are calculated as per table 15 below.

Denote the number of bits before second rate matching as N_{sys} for the systematic bits, N_{p1} for the parity 1 bits, and N_{p2} for the parity 2 bits, respectively. Denote the number of physical channels used for the CCTrCH by *P*. N_{data} is the number of bits available to the CCTrCH in one radio frame and defined as $N_{data}=P\times 3\times N_{data1}$, where N_{data1} is defined in [2]. The rate matching parameters are determined as follows.

For $N_{data} \le N_{sys} + N_{p1} + N_{p2}$, puncturing is performed in the second rate matching stage. The number of transmitted systematic bits in a retransmission is $N_{t,sys} = \min\{N_{sys}, N_{data}\}$ for a transmission of self-decodable type and $N_{t,sys} = \max\{N_{data} - (N_{p1} + N_{p2}), 0\}$ in the non self-decodable case.

For $N_{data} > N_{sys} + N_{p1} + N_{p2}$ repetition is performed in the second rate matching stage. A similar repetition rate in all bit streams is achieved by setting the number of transmitted systematic bits to $N_{t,sys} = \left\lfloor N_{sys} \cdot \frac{N_{data}}{N_{sys} + 2N_{p2}} \right\rfloor$.

The number of parity bits in a transmission is: $N_{t,p1} = \left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor$ and $N_{t,p2} = \left\lceil \frac{N_{data} - N_{t,sys}}{2} \right\rceil$ for the parity 1 and parity 2 bits, respectively.

Table 15 below summarizes the resulting parameter choice for the second rate matching stage. The parameter *a* in the table is chosen using *a* = 2 for parity 1 and *a* = 1 for parity 2.

Table 15: Parameters for HARQ second rate matching

	X_i	e_{plus}	e_{minus}
Systematic RM S	N_{sys}	N_{sys}	$\left\| N_{sys} - N_{t,sys} \right\|$
Parity 1 RM P1_2	N_{p1}	$a \cdot N_{p1}$	$a \cdot \left\| N_{p1} - N_{t,p1} \right\|$
Parity 2 RM P2_2	N_{p2}	$a \cdot N_{p2}$	$a \cdot \left\| N_{p2} - N_{t,p2} \right\|$

The rate matching parameter e_{ini} is calculated for each bit stream according to the RV parameters r and s using

$e_{ini}(r) = \left\{ \left[X_i - \left(r \cdot e_{plus} / r_{\max} \right) - 1 \right] \bmod e_{plus} \right\} + 1$ in the case of puncturing , i.e., $N_{data} \le N_{sys} + N_{p1} + N_{p2}$, and

$e_{ini}(r) = \left\{ \left[X_i - \left(\left(s + 2 \cdot r \right) \cdot e_{plus} / \left(2 \cdot r_{\max} \right) \right) - 1 \right] \bmod e_{plus} \right\} + 1$ for repetition, i.e., $N_{data} > N_{sys} + N_{p1} + N_{p2}$.

Where $r \in \{0, 1, \cdots, r_{\max} - 1\}$ and $r_{\max}$ is the total number of redundancy versions allowed by varying r . Note that r_{max} varies depending on the modulation mode.

Note: For the modulo operation the following clarification is used: the value of (x mod y) is strictly in the range of 0 to *y*-1 (i.e. -1 mod 10 = 9).

4.5.4.4 HARQ bit collection

The HARQ bit collection is achieved using a rectangular interleaver of size $N_{row} \times N_{col}$.

The number of rows and columns are determined from:

$$N_{row} = \log_2(M)$$
$$N_{col} = F / N_{row}$$

where M is the modulation size and F is the number of coded and rate-matched bits to be transmitted.

Data is written into the interleaver column by column, and read out of the interleaver column by column.

$N_{t,sys}$ is the number of transmitted systematic bits. Intermediate values N_r and N_c are calculated using:

$N_r = \left\lfloor \frac{N_{t,sys}}{N_{col}} \right\rfloor$ and $N_c = \left(\frac{N_{t,sys}}{N_{col}} - N_r \right) \cdot N_{col}$.

If N_c=0, the systematic bits are written into rows $1 \ldots N_r$.

Otherwise systematic bits are written into rows $1 \ldots N_r+1$ in the first N_c columns and rows $1 \ldots N_r$ in the remaining N_c columns. The remaining space is filled with parity bits. The parity bits are written column wise into the remaining rows of the respective columns. Parity 1 and 2 bits are written in alternating order.

In the case of 16QAM for each column the bits are read out of the interleaver in the order row 1, row 3, row 2, row 4. In the case of QPSK for each column the bits are read out of the interleaver in the order row1, row2.

4.5.5 Bit scrambling

The bit scrambling for HS-DSCH shall be done with the general method described in subclause 4.2.9.

4.5.6 Physical channel segmentation for HS-DSCH

When more than one HS-PDSCH is used, physical channel segmentation divides the bits among the different physical channels. The bits input to the physical channel segmentation are denoted by $w_1, w_2, w_3, \ldots w_R$, where R is the number of bits input to the physical channel segmentation block. The number of PhCHs is denoted by *P*.

The bits after physical channel segmentation are denoted $u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU}$, where *p* is PhCH number and *U* is the number of bits in one TTI for each HS-PDSCH, i.e. $U = \frac{R}{P}$. The relation between w_k and u_{pk} is given below.

For all modes, some bits of the input flow are mapped to each code until the number of bits on the code is *U*.

Bits on first PhCH after physical channel segmentation:

$u_{1,k} = w_k \quad k = 1, 2, \ldots, U$

Bits on second PhCH after physical channel segmentation:

$u_{2,k} = w_{k+U} \quad k = 1, 2, \ldots, U$

…

Bits on the P^{th} PhCH after physical channel segmentation:

$u_{P,k} = w_{k+(P-1)\times U} \quad k = 1, 2, \ldots, U$

4.5.7 Interleaving for HS-DSCH

The interleaving for TDD is done as shown in figure 18 below, separately for each physical channel. The bits input to the block interleaver are denoted by $u_1, u_2, u_3, \ldots, u_U$, where *U* is the number of bits in one TTI. For QPSK the interleaver is the same as Rel99 2nd interleaver described in Section 4.2.11.1. The interleaver is of fixed size: R2=32 rows and C2=30 columns.

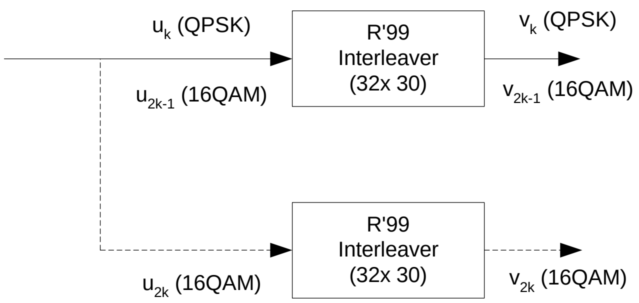


Figure 18: Interleaver structure for HSDPA

For 16QAM, there are two identical interleavers of the same fixed size R2×C2 = 32×30. The output bits from the physical channel segmentation are divided between the interleavers: all odd numbered bits to interleaver one and all even numbered bits to interleaver two.

Note: the outputs of the interleavers will result in mapping to 16QAM symbols such that the output of first interleaver is mapped to the more reliable positions (i_1 and q_1) whereas the output of the second interleaver is mapped to the less reliable positions (i_2 and q_2).

4.5.8 Physical channel mapping for HS-DSCH

The HS-PDSCH is defined in [7]. The bits input to the physical channel mapping are denoted by $v_{p1}, v_{p2}, \ldots, v_{pU}$, where *p* is the physical channel number and *U* is the number of bits in one TTI for one HS-PDSCH. The bits v_{pk} are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to *k*.

4.5.9 Constellation re-arrangement for 16 QAM

This function only applies to 16 QAM modulated bits. In case of QPSK it is transparent.

The following table 16 describes the operations that produce the different rearrangements.

The bits of the input sequence are mapped in groups of 4 so that v_{pk}, v_{pk+1}, v_{pk+2}, v_{pk+3} map to $i_a i_b q_a q_b$, where k mod 4 = 0.

Table 16: Constellation re-arrangement for 16 QAM

Constellation version parameter ***b***	Output bit sequence	Operation
0	$i_a q_a i_b q_b$	None
1	$i_b q_b i_a q_a$	Swapping i_a with i_b and q_a with q_b
2	$i_a q_a \overline{i_b} \overline{q_b}$	XOR with 0011 (equivalent to inversion of the logical values of i_b and q_b)
3	$i_b q_b \overline{i_a} \overline{q_a}$	Swapping i_a with i_b and q_a with q_b and XOR with 0011

The output bit sequences from the table above map to the output bits in groups of 4, i.e. r_{pk}, r_{pk+1}, r_{pk+2}, r_{pk+3}, where k mod 4 = 0.

4.6 Coding/Multiplexing for HS-SCCH

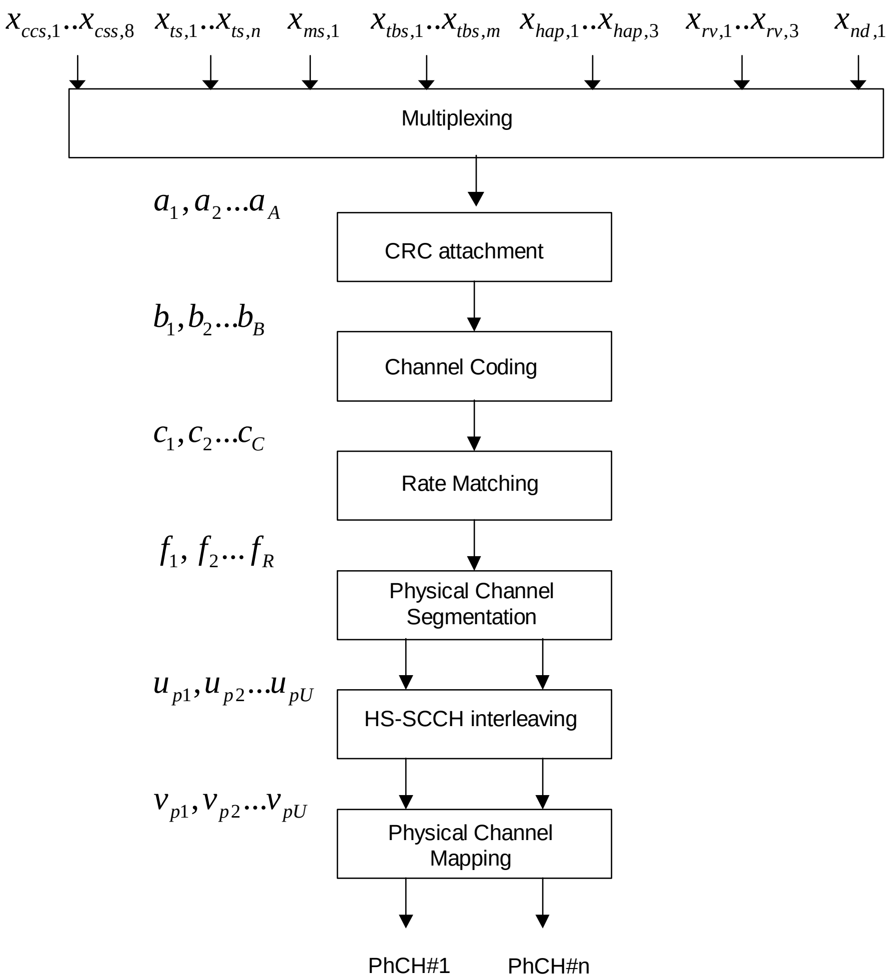
The following information, provided by higher layers, is transmitted by means of the HS-SCCH physical channel.

- Channelisation-code-set information (8 bits): $x_{ccs,1}, x_{ccs,2}, \ldots, x_{ccs,8}$
- Time slot information (*n* bits where *n* = 5 for 1.28 Mcps TDD and *n* = 13 for 3.84 Mcps TDD): $x_{ts,1}, x_{ts,2}, \ldots, x_{ts,n}$
- Modulation scheme information (1 bit): $x_{ms,1}$
- Transport-block size information (m bits where m = 6 for 1.28 Mcps TDD and m = 9 for 3.84 Mcps TDD): $x_{tbs,1}, x_{tbs,2}, \ldots, x_{tbs,m}$
- Hybrid-ARQ process information (3 bits): $x_{hap,1}, x_{hap,2}, x_{hap,3}$
- Redundancy version information (3 bits): $x_{rv,1}, x_{rv,2}, x_{rv,3}$
- New data indicator (1 bit): $x_{nd,1}$
- UE identity (10 bits): $x_{ue,1}, x_{ue,2}, \ldots, x_{ue,10}$

The following coding/multiplexing steps can be identified:

- multiplexing of HS-SCCH information (see subclause 4.6.1)
- CRC attachment (see subclause 4.6.2);
- channel coding (see subclause 4.6.3);
- rate matching (see subclause 4.6.4);
- interleaving for HS-SCCH (see subclause 4.6.5);
- mapping to physical channels (see subclause 4.6.6).

The general coding/multiplexing flow is shown in the figure below.



4.6.1 Multiplexing of HS-SCCH information

The information carried on the HS-SCCH is multiplexed onto the bits $a_1, a_2, ... a_A$ according to the following rule :

$$a_1, a_2 ... a_8 = x_{ccs,1}, x_{css,2} ... x_{css,8}$$

$$a_9, a_{10} ... a_{9+n-1} = x_{ts,1}, x_{ts,2} ... x_{ts,n}$$

$$a_{9+n} = x_{ms,1}$$

$$a_{9+n+1}, a_{9+n+2} ... a_{9+n+m} = x_{tbs,1}, x_{tbs,2} ... x_{tbs,m}$$

$$a_{10+n+m}, a_{11+n+m}, a_{12+n+m} = x_{hap,1}, x_{hap,2}, x_{hap,3}$$

$$a_{13+n+m}, a_{14+n+m}, a_{15+n+m} = x_{rv,1}, x_{rv,2}, x_{rv,3}$$

$$a_{16+n+m} = x_{nd,1}$$

4.6.2 CRC attachment for HS-SCCH

The bits b_1, ..., b_B are generated by adding the computed CRC of length 16 as described in the general section 4.2.1.1, and then scrambling the computed CRC by the modulo 2 addition of an extended UE identifier. The MSBs of the UE identifier shall be extended to 16 bits by zero padding.

4.6.3 Channel coding for HS-SCCH

Channel coding for the HS-SCCH shall be done with the general method described in 4.2.3 with the following specific parameters:

The rate 1/3 convolutional coding shall be used for HS-SCCH.

4.6.4 Rate matching for HS-SCCH

Rate matching for HS-SCCH shall be done with the general method described in 4.2.7.

4.6.5 Physical Channel Segmentation for HS-SCCH

Physical Channel Segmentation for HS-SCCH shall be done with the general method described in 4.2.10. For 1.28 Mcps TDD, the HS-SCCH consists of two physical channels HS-SCCH1 and HS-SCCH2; for 3.84 Mcps TDD the HS-SCCH only uses one physical channel, see [7].

4.6.6 Interleaving for HS-SCCH

Interleaving for HS-SCCH shall be done with the general method described in 4.2.11.1.

4.6.7 Physical channel mapping for HS-SCCH

Physical channel mapping for the HS-SCCH shall be done with the general method described in subclause 4.2.13.

4.7 Coding for HS-SICH

The following information, provided by higher layers, is transmitted by means of the HS-SICH physical channel.

- Recommended Modulation Format (RMF) (1 bit): $x_{rmf,1}$
- Recommended Transport-block size (RTBS) (n bits where n = 6 for 1.28 Mcps TDD and n = 9 for 3.84 Mcps TDD): $x_{tbs,1}$, $x_{tbs,2}$, …, $x_{tbs,n}$
- Hybrid-ARQ information ACK/NACK (1 bit): $x_{an,1}$

The following coding/multiplexing steps can be identified:

- separate coding of RMF, RTBS and ACK/NACK (see subclause 4.7.1);
- multiplexing of HS-SICH information (4.7.2);
- interleaving for HS-SICH (see subclause 4.7.3);
- mapping to physical channels (see subclause 4.7.4).

The general coding/multiplexing flow is shown in the figure 19.

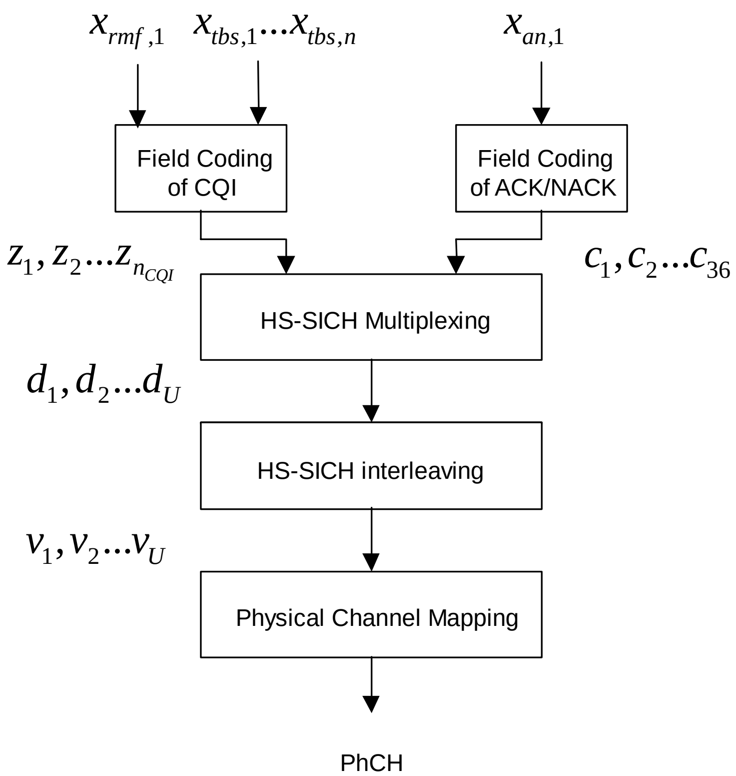


Figure 19 Coding and multiplexing for HS-SICH

4.7.1 Coding for HS-SICH

4.7.1.1 Field Coding of ACK/NACK

The ACK/NACK field of the HS-SICH is repetition coded to 36 bits. The coded bits are defined as $c_1 \dots c_{36}$

4.7.1.2 Field Coding of CQI

4.7.1.2.1 Field Coding of CQI for 1.28 Mcps TDD

The quality information consists of Recommended Transport Block Size (RTBS) and Recommended Modulation Format (RMF) fields. The 6 bits of the RTBS field are coded to 32 bits using a (32, 6) 1st order Reed-Muller code. The coding procedure is as shown in figure 20.

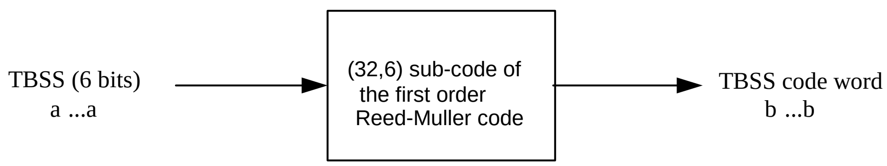


Figure 20 Field coding of RTBS information bits

The coding uses a subset basis sequences as the TFCI coder as described in subclause 4.3.1.1. The basis sequences that are used for RTBS coding are as follows in table 17.

Table 17: Basis sequences for (32,6) RTBS code

I	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{I,4}$	$M_{i,5}$
0	1	0	0	0	0	1
1	0	1	0	0	0	1
2	1	1	0	0	0	1
3	0	0	1	0	0	1
4	1	0	1	0	0	1
5	0	1	1	0	0	1
6	1	1	1	0	0	1
7	0	0	0	1	0	1
8	1	0	0	1	0	1
9	0	1	0	1	0	1
10	1	1	0	1	0	1
11	0	0	1	1	0	1
12	1	0	1	1	0	1
13	0	1	1	1	0	1
14	1	1	1	1	0	1
15	1	0	0	0	1	1
16	0	1	0	0	1	1
17	1	1	0	0	1	1
18	0	0	1	0	1	1
19	1	0	1	0	1	1
20	0	1	1	0	1	1
21	1	1	1	0	1	1
22	0	0	0	1	1	1
23	1	0	0	1	1	1
24	0	1	0	1	1	1
25	1	1	0	1	1	1
26	0	0	1	1	1	1
27	1	0	1	1	1	1
28	0	1	1	1	1	1
29	1	1	1	1	1	1
30	0	0	0	0	0	1
31	0	0	0	0	1	1

The output RTBS code word bits b_i are given by:

$$b = \sum_{=} (a \times M \ ) \bmod 2$$

where $i = 0,\ldots,31$. $N_{\text{RTBS code word}} = 32$.

The 1 bit of the RMF is repetition coded to 16 bits to produce the bits . $b_{32}, b_{33} \ldots b_{47}$.

The CQI is composed of the bits $z_1, z_2 \ldots z_{n_{CQI}}$ where :

$$z_1, z_2 \ldots z_{n_{CQI}} = b_0, b_1 \ldots b_{47}$$

4.7.1.2.2 Field Coding of CQI for 3.84 Mcps TDD

RTBS and RMF bits are multiplexed onto the bits $y_1, y_2 \ldots y_{10}$ according to the following rule :

$$y_1 = x_{rmf,1}$$

$$y_2, y_3 \ldots y_{10} = x_{tbs,1}, x_{tbs,2} \ldots x_{tbs,9}$$

The bits $y_1, y_2 ... y_{10}$ are coded to produce the CQI bits $z_1, z_2 ... z_{n_{CQI}}$ using a (32,10) sub-code of the second order Reed-Muller code as defined in subclause 4.3.1.1, where $n_{CQI} = 32$.

4.7.2 Multiplexing of HS-SICH information fields

The CQI bits $z_1, z_2 ... z_{n_{CQI}}$ are multiplexed with the repetition coded ACK/NACK bits $c_1 ... c_{36}$ to produce the bits $d_1, d_2 ... d_U$ where U is the number of physical channel bits carried by HS-SICH, according to the following rule.:

$$d_1, d_2 ... d_{n_{CQI}} = z_{1,} z_2 ... z_{n_{CQI}}$$

$$d_{n_{CQI}+1}, d_{n_{CQI}+2} ... d_{n_{CQI}+36} = c_{1,} c_2 ... c_{36}$$

$$d_{n_{CQI}+37}, d_{n_{CQI}+38} ... d_U = 0,0....0$$

4.7.3 Interleaver for HS-SICH

Interleaver for HS-SICH shall be done with the general method described in 4.2.11.1.

4.7.4 Physical channel mapping for HS-SICH

Physical channel mapping for HS-SICH shall be done with the general method described in 4.2.13.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
14/01/00	RAN_05	RAN_05	-		Approved at TSG RAN #5 and placed under Change Control	-	3.0.0
14/01/00	RAN_06	RP-99694	001	3	Correction of rate matching parameters for repetition after 1st Interleaving in 25.222	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	002	1	Clarification of bit separation and collection	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	003	-	Changing the initial offset value for convolutional code rate matching	3.0.0	3.1.0
14/01/00	RAN_06	RP-99693	004	1	Editorial corrections to TS 25.222	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	007	-	Update of rate matching rule for TDD	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	009	1	Modified physical channel mapping scheme	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	013	-	Introduction of TFCI for S-CCPCH in TDD mode	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	015	-	TFCI coding and mapping in TDD	3.0.0	3.1.0
14/01/00	-	-	-		Change history was added by the editor	3.1.0	3.1.1
31/03/00	RAN_07	RP-000068	017	-	Corrections to TS 25.222	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	018	-	Refinements of Physical Channel Mapping	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	019	1	TFCI coding specification in TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	021	-	Modification of Turbo code internal interleaver	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	023	-	Update of TS 25.222 - clarification of BTFD for TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	025	-	Change of TFCI basis for TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	026	-	Padding Function for Turbo coding of small blocks	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	027	-	Editorial modification of shifting parameter calculation for turbo code puncturing	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	029	1	Editorial changes of channel coding section	3.1.1	3.2.0
26/06/00	RAN_08	RP-000272	030	-	Parity bit attachment to 0 size transport block	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	031	-	Correction of the mapping formula	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	034	-	Alignment of Multiplexing for TDD	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	036	2	Bit separation of the Turbo encoded data	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	038	2	Revision of code block segmentation description	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	039	-	Editorial corrections in channel coding section	3.2.0	3.3.0
23/09/00	RAN_09	RP-000345	040	1	Update of TS 25.222	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	041	1	Editorial corrections in Turbo code internal interleaver section	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	042	-	Paging Indicator Terminology	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	043	1	Bit separation and collection for rate matching	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	048	-	Puncturing Limit definition in WG1 specification	3.3.0	3.4.0
15/12/00	RAN_10	RP-000543	049	-	Clarification on the Ci formula	3.4.0	3.5.0
15/12/00	RAN_10	RP-000543	050	-	Correction on TFCI & TPC Transmission	3.4.0	3.5.0
15/12/00	RAN_10	RP-000543	053	1	Editorial corrections in TS 25.222	3.4.0	3.5.0
16/03/01	RAN_11	-	-	-	Approved as Release 4 specification (v4.0.0) at TSG RAN #11	3.5.0	4.0.0
16/03/01	RAN_11	RP-010063	051	1	Bit Scrambling for TDD	3.5.0	4.0.0
16/03/01	RAN_11	RP-010063	054	1	Corrections & Clarifications for TS25.222	3.5.0	4.0.0
16/03/01	RAN_11	RP-010071	055	1	Inclusion of 1.28Mcps TDD in TS 25.222	3.5.0	4.0.0
21/09/01	RAN_13	RP-010523	057	-	TFCI Terminology	4.0.0	4.1.0
21/09/01	RAN_13	RP-010529	058	-	5ms TTI for PRACH for 1.28 Mcps TDD	4.0.0	4.1.0
21/09/01	RAN_13	RP-010529	060	-	A correction on the meaning of FPACH in TS 25.222	4.0.0	4.1.0
14/12/01	RAN_14	RP-010747	059	-	Bit Scrambling for TDD	4.1.0	4.2.0
14/12/01	RAN_14	RP-010747	061	-	Corrections in clause 4.1 and 4.2 of TS 25.222	4.1.0	4.2.0
08/03/02	RAN_15	RP-020050	063	1	Correction to addition of padding zeros to PICH in TDD	4.2.0	4.3.0
08/03/02	RAN_15	RP-020050	065	3	Clarification of the requirement for the determination of the rate matching parameters and editorial corrections to 25.222	4.2.0	4.3.0
08/03/02	RAN_15	RP-020058	066	2	Inclusion of HSDPA in 25.222	4.2.0	5.0.0

History

Document history		
V5.0.0	March 2002	Publication