TSG-RAN Meeting #7 ***RP-000068***
Madrid, Spain, 13 – 15 March 2000

Title: **Agreed CRs to TS 25.222**

Source: **TSG-RAN WG1**

Agenda item: **6.1.3**

No.	Doc #	Spec	CR	Rev	Subject	Cat	Versio	Versio
1	R1-000082	25.222	017	-	Corrections to TS 25.222	F	3.1.1	3.2.0
2	R1-000081	25.222	018	-	Refinements of Physical Channel Mapping	F	3.1.1	3.2.0
3	R1-000193	25.222	019	1	TFCI coding specification in TDD	F	3.1.1	3.2.0
4	R1-000160	25.222	021	-	Modification of Turbo code internal interleaver	B	3.1.1	3.2.0
5	R1-000226	25.222	023	-	Update of TS 25.222 – clarification of BTFD for	F	3.1.1	3.2.0
6	R1-000237	25.222	025	-	Change of TFCI basis for TDD	F	3.1.1	3.2.0
7	R1-000255	25.222	026	-	Padding Function for Turbo coding of small blocks	B	3.1.1	3.2.0
8	R1-000282	25.222	027	-	Editorial modification of shifting parameter	D	3.1.1	3.2.0
9	R1-000437	25.222	029	1	Editorial changes of channel coding section	D	3.1.1	3.2.0

Document R1-00-0082
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 017 Current Version: V3.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG RAN#7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 14.01.2000

Subject: Corrections to TS 25.222

Work item: TS 25.222

Category:				**Release:**	
	F	Correction	X	Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
(only one category	B	Addition of feature		Release 97	
shall be marked	C	Functional modification of feature		Release 98	
with an X)	D	Editorial modification		Release 99	X
				Release 00	

Reason for change: This CR includes the following corrections, mainly due to harmonisation reasons with FDD:

- Inclusion of flexible coding schemes for the FACH
- Modification of the chapter describing convolutional coding
- Clarification of the alignment of TrCHs with different TTI
- Handling of zero length transport blocks
- Clarification of the indices for timeslot related 2nd interleaving

Furthermore, some minor editorial changes have been added.

Clauses affected:

Other specs affected:			
Other 3G core specifications		→ List of CRs:	
Other GSM core specifications		→ List of CRs:	
MS test specifications		→ List of CRs:	
BSS test specifications		→ List of CRs:	
O&M specifications		→ List of CRs:	

Other comments:

help.doc

<---------- double-click here for help and instructions on how to create a CR.

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
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
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
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RACH	Random Access Channel
RF	Radio Frequency
RLC	Radio Link Control
RRC	Radio Resource Control
RRM	Radio Resource Management
RSC	Recursive Systematic Convolutional Coder
RT	Real Time
RU	Resource Unit
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
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

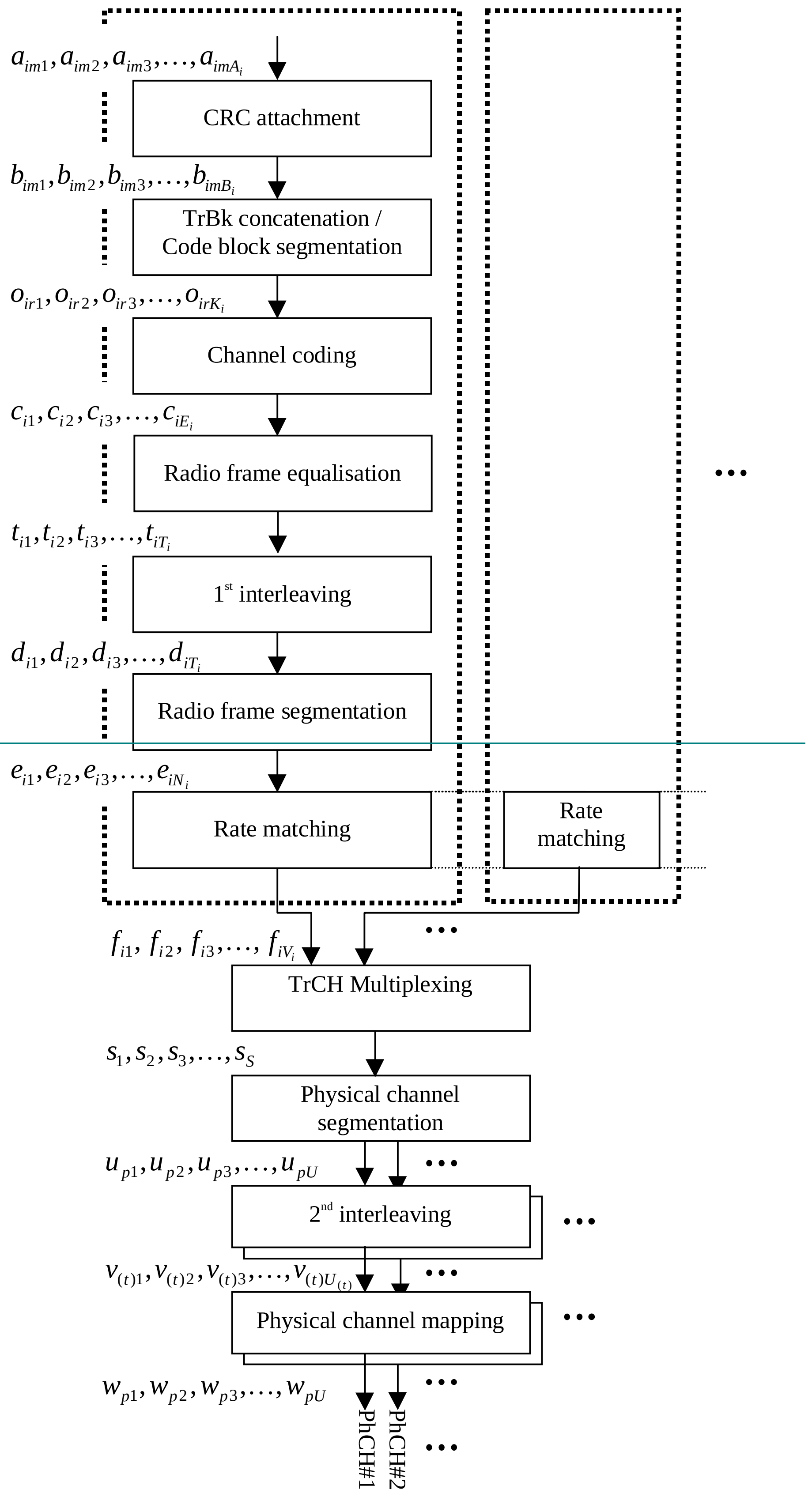
4.2 Transport channel coding/multiplexing

Figure 4-1 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 {10 ms, 20 ms, 40 ms, 80 ms}.

The following coding/multiplexing steps can be identified:

- Add CRC to each transport block (see section 4.2.1)
- TrBk concatenation / Code block segmentation (see section 4.2.2)
- Channel coding (see section 4.2.3)
- Radio frame size equalization (see section 4.2.4)
- Interleaving (two steps, see sections 4.2.5 and 4.2.10)
- Radio frame segmentation (4.2.6)
- Rate matching (see section 4.2.7)
- Multiplexing of transport channels (see section 4.2.8)
- Physical channel segmentation (see section 4.2.9)
- Mapping to physical channels (see section 4.2.11)

The coding/multiplexing steps for uplink and downlink are shown in figure 4-1.



$a_{im1}, a_{im2}, a_{im3}, \ldots, a_{imA_i}$
CRC attachment
$b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$
TrBk concatenation / Code block segmentation
$o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$
Channel coding
$c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$
Radio frame equalisation
$t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$
1st interleaving
$d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$
Radio frame segmentation
$e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$
Rate matching
Rate matching
$f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}$
TrCH Multiplexing
$s_1, s_2, s_3, \ldots, s_S$
Physical channel segmentation
$u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU}$
2nd interleaving
$v_{(t)1}, v_{(t)2}, v_{(t)3}, \ldots, v_{(t)U_{(t)}}$
Physical channel mapping
$w_{p1}, w_{p2}, w_{p3}, \ldots, w_{pU}$
PhCH#1
PhCH#2

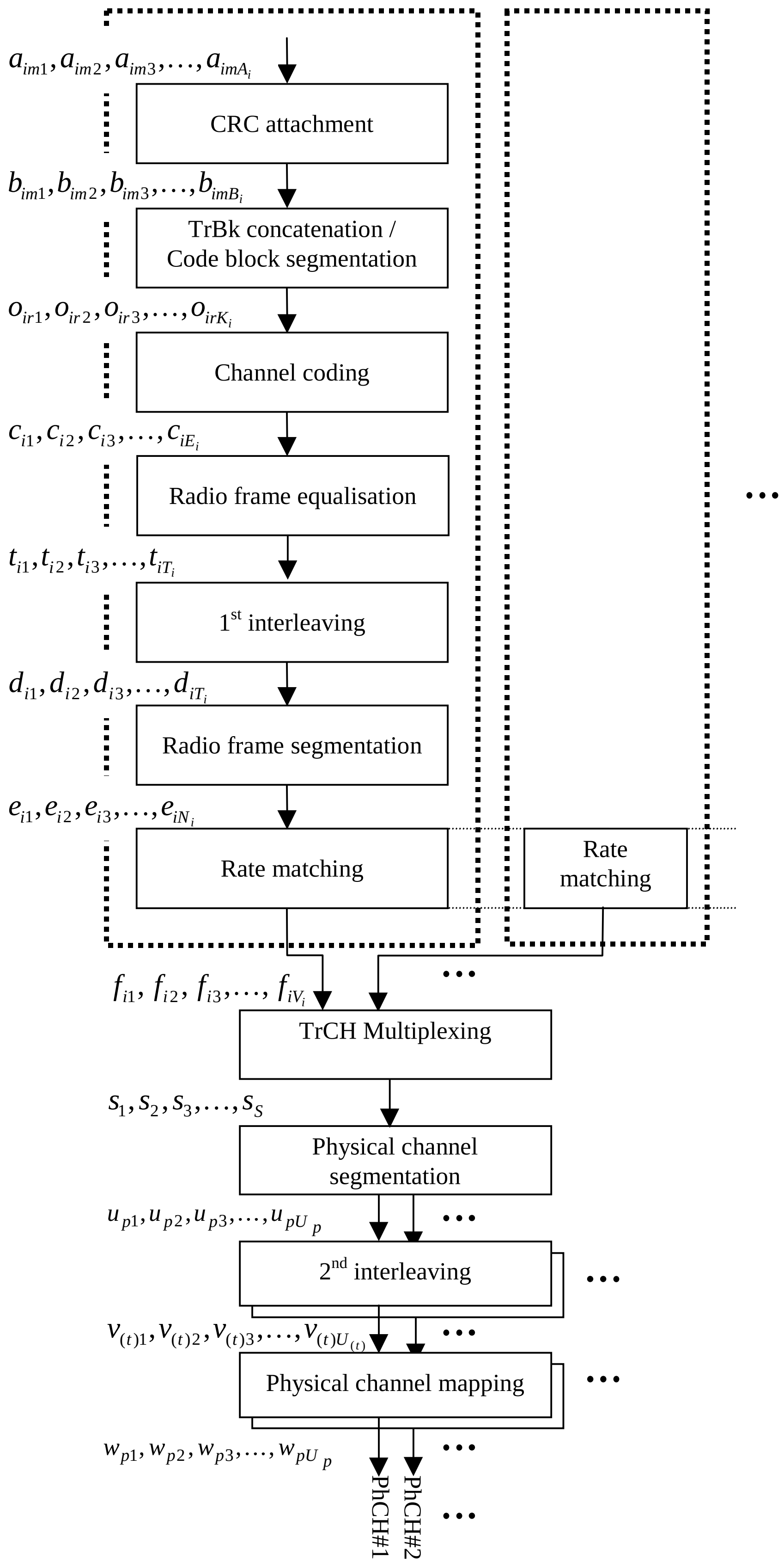


Figure 4–1: Transport channel multiplexing structure for uplink and downlink

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 length of a transport block of TrCH i, m is the transport block number, and L_i is 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

$$\cancel{a_{im1}D^{A_i+11} + a_{im2}D^{A_i+10} + \ldots + a_{imA_i}D^{12} + p_{im1}D^{11} + p_{im2}D^{10} + \ldots + p_{im7}D^1 + p_{im12}}$$

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

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.

4.2.2.2 Code block segmentation

~~NOTE: It is assumed that filler bits are set to 0.~~

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 last 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 are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits.

Number of code blocks: $C_i = \lceil X_i / Z \rceil$

Number of bits in each code block: $K_i = \lceil X_i / C_i \rceil$

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

If $X_i \le Z$, then $o_{i1k} = x_{ik}$, and $K_i = X_i$.

If $X_i \ge Z$, then

$o_{i1k} = x_{ik}$ $\quad k = 1, 2, \dots, K_i$

$o_{i2k} = x_{i,(k+K_i)}$ $\quad k = 1, 2, \dots, K_i$

$o_{i3k} = x_{i,(k+2K_i)}$ $\quad k = 1, 2, \dots, K_i$

$\dots$

$o_{iC_ik} = x_{i(k+(C_i-1)K_i)}$ $\quad k = 1, 2, \dots, K_i - Y_i$

$o_{iC_ik} = 0$ $\quad k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \dots, K_i$

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH i is denoted by C_i. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$. The encoded blocks are serially multiplexed so that the block with lowest index r is output first from the channel coding block. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where i is the TrCH number and $E_i = C_i Y_i$. The output bits are defined by the following relations:

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

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

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

$\dots$

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

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 channel coding

The values of Y_i in connection with each coding scheme:

- Convolutional coding, ½ rate: $Y_i = 2*K_i + 16$; 1/3 rate: $Y_i = 3*K_i + 24$
- Turbo coding, 1/3 rate: $Y_i = 3*K_i + 12$
- No channel coding, $Y_i = K_i$

Table 4.2.3-1: Error Correction Coding Parameters

Transport channel type	Coding scheme	Coding rate
BCH	Convolutional code	1/2
PCH		
RACH		
DCH, DSCH, USCH, FACH		1/3, 1/2
	Turbo code	1/3
	No coding	

~~**Table 4.2.3-1: Error Correction Coding Parameters**~~

~~Transport channel type~~	~~Coding scheme~~	~~Coding rate~~
~~BCH~~	~~Convolutional code~~	~~1/2~~
~~PCH~~		
~~FACH~~		
~~RACH~~		
~~DCH, DSCH, USCH~~		~~1/3, 1/2~~
	~~Turbo code~~	~~1/3~~
	~~No coding~~	

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

4.2.3.1 Convolutional Coding

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

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

Output from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output 2, output 0,…,output2. Output from the rate 1/2 convolutional coder shall be done in the order output 0, output 1, output 0, output 1, output 0, …, output 1.

8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.

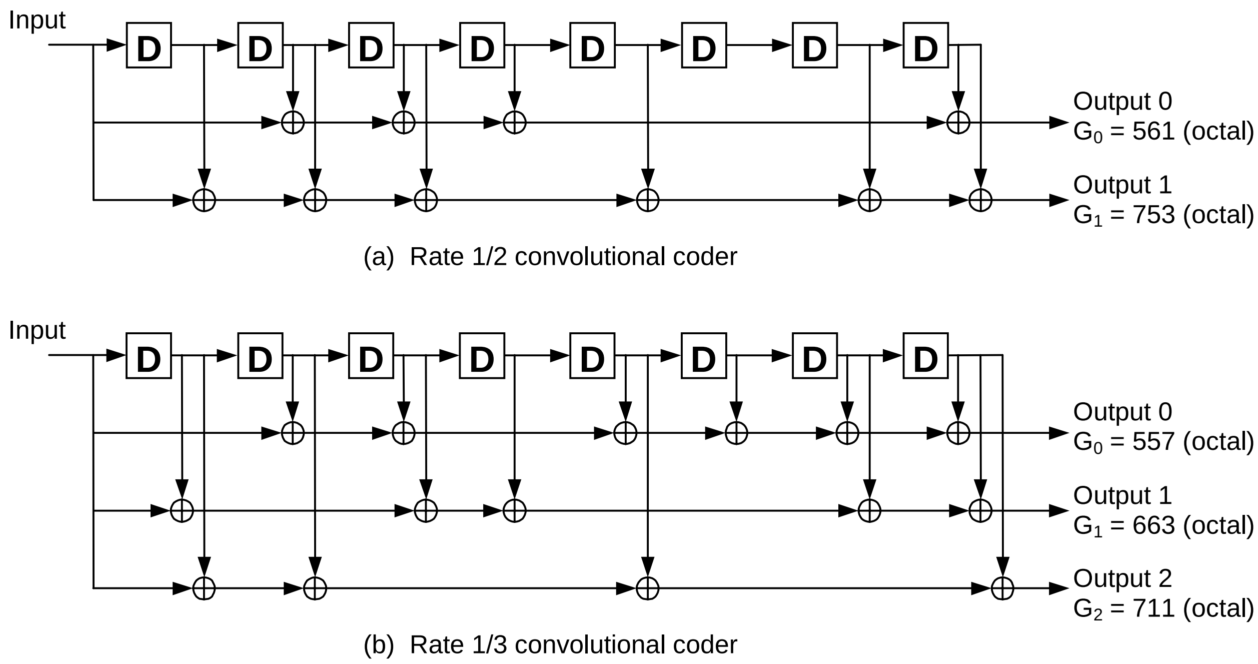
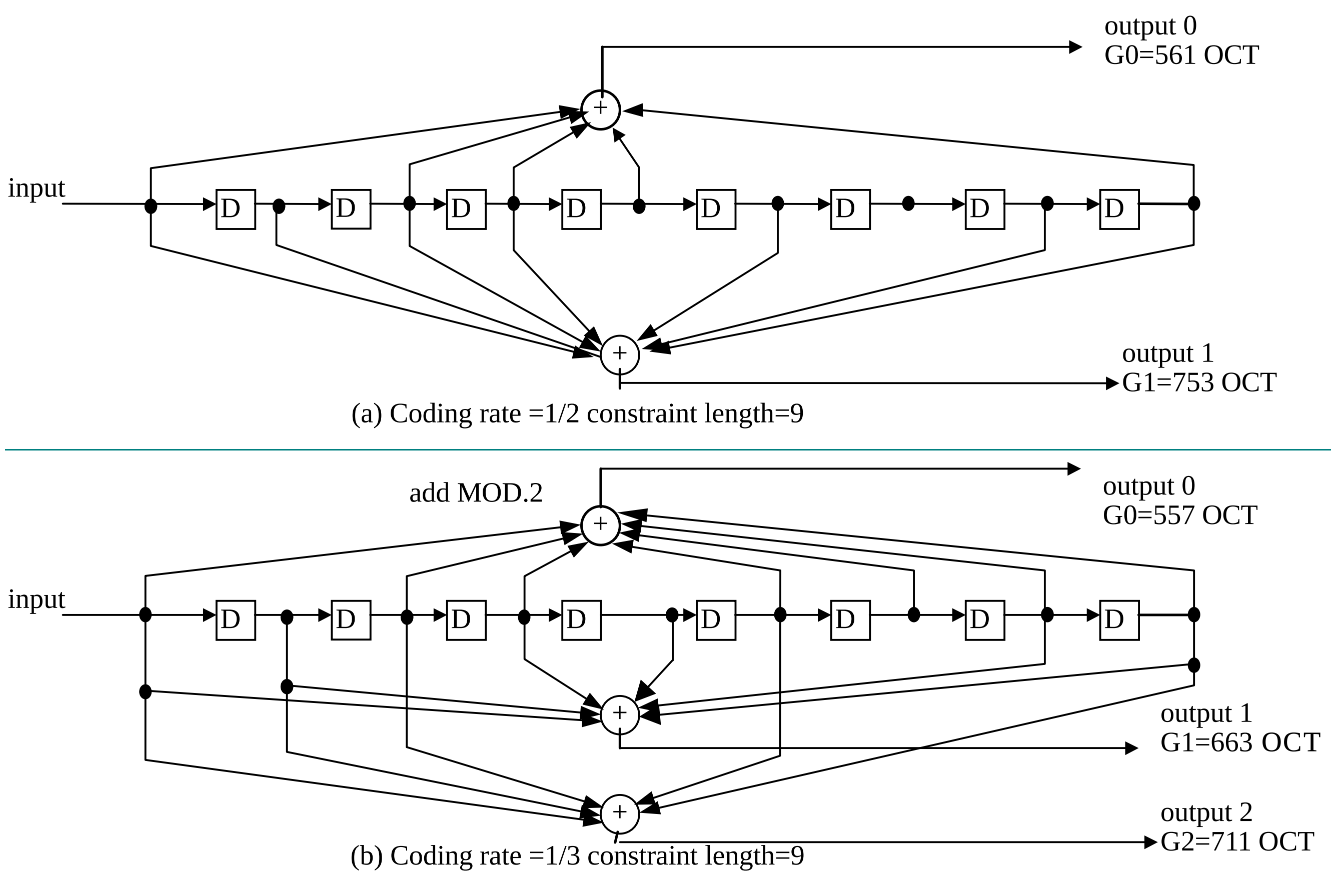


Figure 3: Rate 1/2 and rate 1/3 convolutional coders

- ~~Constraint length K=9. Coding rates 1/2 and 1/3.~~
- ~~The configuration of the convolutional coder is presented in figure 4-2.~~
- ~~The output from the convolutional coder shall be done in the order output0, output1,output2, output0, output1,…, output2. (When coding rate is 1/2, output is done up to output 1).~~
- ~~The initial value of the shift register of the coder shall be "all 0".~~
- ~~K-1 tail bits (value 0) shall be added to the end of the code block before encoding.~~



(a) Coding rate =1/2 constraint length=9

(b) Coding rate =1/3 constraint length=9

~~Figure 4-2: Convolutional Coder~~

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The turbo coding scheme is a parallel concatenated convolutional code (PCCC) with 8-state constituent encoders.

~~For data services requiring quality of service between 10^{-3} and 10^{-6} BER inclusive, parallel concatenated convolutional code (PCCC) with 8-state constituent encoders is used.~~

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

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

where,

$$d(D)=1+D^2+D^3$$

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

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 section 4.2.7 and subsections:

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

ΔN_{ij} : If positive – number of bits to be repeated in each radio frame on TrCH *i* with transport format combination *j*.

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.

$N_{data,j}$: 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.

$Z_{\cancel{m}ij}$: 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).

$I_F(n_i)$: The inverse interleaving function of the 1st interleaver (note that the inverse interleaving function is identical to the interleaving function itself for the 1st interleaver).

$S(n_i)$: The shift of the puncturing or repetition pattern for radio frame 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 section 4.2.7.3.

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

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

b: Indicates systematic and parity bits.

b=~~0~~1: Systematic bit. *X(t)* in 4.2.3.2.1.

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

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

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

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

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

Puncturing can be used to minimise the required transmission capacity. The maximum amount of puncturing that can be applied is signalled from higher layers and denoted by PL. 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, usage of TPC and multiframe structure), which is given in [7].

Denote the number of data bits in each physical channel by ~~N~~$U_{p,Sp}$ (~~$N_{k,Sk}$~~), where ~~k~~ *p* refers to the sequence number $1 \leq$ ~~k~~ $p \leq P_{max}$ of this physical channel in the allocation message, and the second index *S*~~k~~*p* indicates the spreading factor with the possible values {*16, 8, 4, 2, 1*}, respectively. For each physical channel an individual minimum spreading factor *S*~~k~~*p*$_{min}$ is transmitted by means of the higher layer. Then, for N_{data} one of the following values in ascending order can be chosen: ~~{$N_{1,16}$, ..., $N_{1,S1min}$, $N_{1,S1min}+N_{2,16}$, ..., $N_{1,S1min}+N_{2,S2min}$, ..., $N_{1,S1min}+N_{2,S2min}+\ldots+N_{P,16}$, ..., $N_{1,S1min}+N_{2,S2min}+\ldots+N_{P,SPmin}$}.~~

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

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

$$\text{SET1} = \{ N_{data} \text{ such that } N_{data} - PL \cdot \sum_{x=1}^{I} \frac{RM_x}{\min_{1 \leq y \leq I}\{RM_y\}} \cdot N_{x,j} \text{ is non negative } \}$$

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

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

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

Otherwise, the rate matching pattern is calculated with the algorithm described in section 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 section 4.2.7.1.1 and 4.2.7.1.2

4.2.7.3 Rate matching pattern determination

The bits input to the rate matching are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where *i* is the TrCH and X_i is the parameter given in section 4.2.7.1.1 and 4.2.7.1.2.~~The bits output from the rate matching are denoted by $f_{i1}, f_{i2}, f_{13}, \ldots, f_{iV_i}$, where *i* is the TrCH number and V_i=N+ΔN .~~

Note that 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 = eini      -- initial error between current and desired puncturing ratio
    m = 1         -- index of current bit
    do while m <= Xi
        e = e – eminus        -- update error
        if e <= 0 then        -- check if bit number m should be punctured
            set bit xi,m to δ where δ∉{0, 1}
            e = e + eplus     -- update error
        end if
        m = m + 1             -- next bit
    end do
else
    e = eini      -- initial error between current and desired puncturing ratio
    m = 1         -- index of current bit
    do while m <= Xi
        e = e – eminus        -- update error
        do while e <= 0       -- check if bit number m should be repeated
            repeat bit xi,m
            e = e + eplus     -- update error
        end do
        m = m + 1             -- next bit
    end do
end if
```

A repeated bit is placed directly after the original one.

4.2.10 2nd interleaving

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.

4.2.10.1 Frame related 2nd interleaving

In case of frame related interleaving, the bits input to the 2nd interleaver are denoted $x_1, x_2, x_3, \ldots, x_U$, 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_{pk} in the respective physical channels is given below:

$x_k = u_{1k}$ k = 1, 2 , …, U_1

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

…

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

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

(1) Set the number of columns C_2 = 30. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that $U \le R_2C_2$.

(3) The bits input to the 2nd interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_{30} \\ x_{31} & x_{32} & x_{33} & \ldots & x_{60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{(R_2-1)30+1} & x_{(R_2-1)30+2} & x_{(R_2-1)30+3} & \ldots & x_{R_2\cdot 30} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern $\{P_2(j)\}$ (j = 0, 1, …, C_2-1) that is shown in table 4.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_k.

$$\begin{bmatrix} y_1 & y_{R_2+1} & y_{2R_2+1} & \ldots & y_{29R_2+1} \\ y_2 & y_{R_2+2} & y_{2R_2+2} & \ldots & y_{29R_2+2} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{R_2} & y_{2R_2} & y_{3R_2} & \ldots & y_{30R_2} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_k that corresponds to bits x_k with k>U are removed from the output. The bits after 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.

4.2.10.2 Timeslot related 2nd interleaving

In case of timeslot related 2nd interleaving, the bits input to the 2nd interleaver are denoted $x_{t1}, x_{t2}, x_{t3}, \dots, x_{tU_t}$, 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_{tk} and u_{tpk} is given below with P_t refering to the number of physical channels within the respective timeslot:

$\cancel{x_{tk} = u_{1k} \quad k = 1, 2, \dots, U_1}$

$x_{tk} = u_{t1k} \quad k = 1, 2, \dots, U_{t1}$

$\cancel{x_{t(k+U_1)} = u_{2k} \quad k = 1, 2, \dots, U_2}$

$x_{t(k+U_{t1})} = u_{t2k} \quad k = 1, 2, \dots, U_{t2}$

…

$\cancel{x_{t(k+U_1+\dots+U_{P_t-1})} = u_{P_t k} \quad k = 1, 2, \dots,}$

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

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

(1) Set the number of columns $C_2 = 30$. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that $U_t \le R_2 C_2$.

(3) The bits input to the 2nd interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_{t1} & x_{t2} & x_{t3} & \dots & x_{t30} \\ x_{t31} & x_{t32} & x_{t33} & \dots & x_{t60} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{t,((R_2-1)30+1)} & x_{t,((R_2-1)30+2)} & x_{t,((R_2-1)30+3)} & \dots & x_{t,(R_2 30)} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern $\{P_2(j)\}$ $(j = 0, 1, \dots, C_2-1)$ that is shown in table 4.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{tk}.

$$\begin{bmatrix} y_{t1} & y_{t,(R_2+1)} & y_{t,(2R_2+1)} & \dots & y_{t,(29R_2+1)} \\ y_{t2} & y_{t,(R_2+2)} & y_{t,(2R_2+2)} & \dots & y_{t,(29R_2+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{tR_2} & y_{t,(2R_2)} & y_{t,(3R_2)} & \dots & y_{t,(30R_2)} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_{tk} that corresponds to bits x_{tk} with $k>U_t$ are removed from the output. The bits after 2nd interleaving are denoted by

4.2.12 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 sh~~ould~~all have co-ordinated timings~~ in the sense that transport blocks arriving from higher layers on different transport channels of potentially different transmission time intervals shall have aligned transmission time instants as shown in figure 4-6~~. When the TFCS of a CCTrCH is changed because a transport channel *i* is added to the CCTrCH or reconfigured within the CCTrCH, the TTI of transport channel *i* may only start in radio frames with CFN fulfilling the relation

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

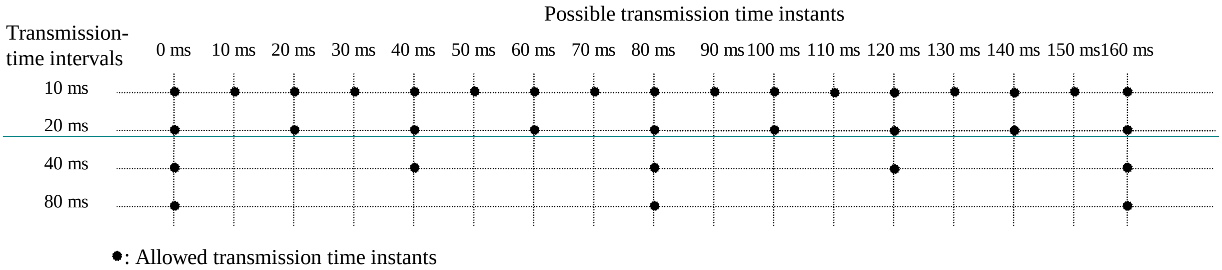
   where F_{max} denotes the maximum number of radio frames within the transmission time intervals of all transport channels which are multiplexed into the same CCTrCH, including transport channel *i* which is added or reconfigured, and CFN_i denotes the connection frame number of the first radio frame within the transmission time interval of transport channel *i*.

   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.



~~Figure 4-6: Possible transmission time instants regarding CCTrCH~~

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

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0081
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222	**CR**	**018**	Current Version:	V3.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN#7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio	X	Core Network	

Source: TSG RAN WG1 **Date:** 14.01.2000

Subject: Refinements of Physical Channel Mapping

Work item: TS 25.222

Category:				**Release:**	
	F	Correction	X	Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
(only one category	B	Addition of feature		Release 97	
shall be marked	C	Functional modification of feature		Release 98	
with an X)	D	Editorial modification		Release 99	X
				Release 00	

Reason for change: This CR does not change the technical content of the mapping scheme, which has been approved during RAN#6. However, the current mathematical notation using indices is replaced by a pseudo c-code, which allows a clearer and uniform representation of the algorithm used for mapping each CCTrCH onto the respective physical channels.

Clauses affected:

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

help.doc

<---------- double-click here for help and instructions on how to create a CR.

4.2.11 Physical channel mapping

The PhCH for both uplink and downlink is defined in [6]. The bits after physical channel mapping are denoted by $w_{p1}, w_{p2}, \ldots, w_{pU_p}$, where p is the PhCH number and U_p is the number of bits in one radio frame for the respective PhCH. The bits w_{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.

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 subsection, shall be applied individually for each timeslot t used in the current frame. Therefore, the bits $v_{t1}, v_{t2}, \ldots, v_{tU_t}$ 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 bs_k the following rule is applied:

```
if
   SF1 >= SF2  then bs1 = 1  ; bs2 =  SF1/SF2 ;
else
   SF2 > SF1  then bs1 = SF2/SF1; bs2 = 1 ;
end if
```

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

4.2.11.1 Mapping scheme

Notation used in this section:

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 Pt                                      --  reset number of already written bits for every physical channel
      fbp = 0
end for

p = 1                                                -- start with PhCH #1

for k=1  to Ut

   do while (fbp == Utp)                             -- physical channel filled up already ?
        p = ((p + 1) mod (Pt +1))  + 1 ;
   end do

   if (p mod 2) == 0
```

pos = U_{tp} - fb_p -- reverse order

else

pos = fb_p + 1 -- forward order

endif

$w_{tp,pos} = v_{t,k}$ -- 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 + 1) mod (P_t +1)) + 1;

end if

end for

~~The mapping scheme depends on the applied 2nd interleaving scheme.~~

~~4.2.11.1 Mapping scheme after frame related 2nd interleaving~~

~~4.2.11.1.1 Mapping scheme after frame related 2nd interleaving in uplink~~

~~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, see section 6.2.11.1.2. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. Then denote the inverse relation of the spreading factors s1: s2 = SF2: SF1, where the smallest possible integers are used for s1 and s2.~~

~~The following mapping rule is applied:~~

~~Bits are mapped on the first PhCH (in forward order) if *(k-1)mod(s1+s2) = 0*, …, s1-1:~~

$$\cancel{w_{1,(k\,div\,(s1+s2))\cdot s1+k \bmod (s1+s2)} = v_k}$$

~~else bits are mapped on the second PhCH (in reverse order):~~

$$\cancel{w_{2,U_2-(k\,div\,(s1+s2))\cdot s2+k \bmod (s1+s2)-s1} = v_k}$$

~~This formula is applied starting with k=1 and increasing k until one of the PhCH is completely filled. From then on, the remaining bits are mapped on the PhCH which has not been filled in the same order (forward or reverse depending on the PhCH) as used previously on that PhCH.~~

~~4.2.11.1.2 Mapping scheme after frame related 2nd interleaving in downlink~~

~~The mapping is equivalent to block interleaving, writing in colomns, 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 following mapping rule is applied:~~

~~Bits are mapped on an odd numbered PhCH (in forward order) according to the following rule, if (k mod P)+1 is odd:~~

$$\cancel{w_{k \bmod P+1,\,k\,div\,P} = v_k}$$

~~Bits are mapped on an even numbered PhCH (in reverse order) according to the following rule, if (k mod P)+1 is even:~~

$$\cancel{w_{k \bmod P+1,\,U_P-1-k\,div\,P} = v_k}$$

~~This formula is applied starting with k=1 and increasing k until all the PhCHs which carry TFCI are completely filled. From then on, the remaining bits are mapped on the remaining PhCHs in the same order (forward or reverse depending on the PhCH) as previously on these PhCHs.~~

~~4.2.11.2 Mapping scheme after timeslot related 2nd interleaving~~

~~For each timeslot only those physical channels with $p = 1, 2, \ldots, P_t$ are considered respectively, which are transmitted in that timeslot, and the following mapping scheme is applied:~~

~~6.2.11.2.1 Mapping scheme after timeslot related 2nd interleaving in uplink~~

~~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, see section 6.2.11.1.2. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. Then denote the inverse relation of the spreading factors s1: s2 = SF2: SF1, where the smallest possible integers are used for s1 and s2.~~

~~The following mapping rule is applied:~~

~~Bits are mapped on the first PhCH (in forward order) if $(k-1) mod(s1+s2) = 0, \ldots, s1-1$:~~

$$\cancel{w_{1,(k\,div(s1+s2))\cdot s1+k \bmod (s1+s2)} = v_{tk}}$$

~~else bits are mapped on the second PhCH (in reverse order):~~

$$\cancel{w_{2,U_2-(k\,div(s1+s2))\cdot s2+k \bmod (s1+s2)-s1} = v_{tk}}$$

~~This formula is applied starting with k=1 and increasing k until one of the PhCH is completely filled. From then on, the remaining bits are mapped on the PhCH which has not been filled in the same order (forward or reverse depending on the PhCH) as used previously on that PhCH.~~

~~6.2.11.2.2 Mapping scheme after timeslot related 2nd interleaving in downlink~~

~~The mapping is equivalent to block interleaving, writing in colomns, 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 following mapping rule is applied:~~

~~Bits are mapped on an odd numbered PhCH (in forward order) according to the following rule, if (k mod P_t)+1 is odd:~~

$$\cancel{w_{k \bmod P_t+1,\,k\,div\,P_t} = v_{tk}}$$

~~Bits are mapped on an even numbered PhCH (in reverse order) according to the following rule, if (k mod P_t)+1 is even:~~

$$\cancel{w_{k \bmod P_t+1,\,U_{P_t}-1-k\,div\,P_t} = v_{tk}}$$

~~This formula is applied starting with k=1 and increasing k until all the PhCHs which carry TFCI are completely filled. From then on, the remaining bits are mapped on the remaining PhCHs in the same order (forward or reverse depending on the PhCH) as previously on these PhCHs.~~

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0193
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 019r1 Current Version: 3.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN#7	for approval	X	strategic	(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic	

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [X] ME [X] UTRAN / Radio [ ] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 2000-1-26

Subject: TFCI coding specification in TDD

Work item:

Category:				**Release:**	
	F	Correction	X	Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
(only one category	B	Addition of feature		Release 97	
shall be marked	C	Functional modification of feature		Release 98	
with an X)	D	Editorial modification		Release 99	X
				Release 00	

Reason for change: To align the TFCI specification with corrections applied to FDD (see R1-00-0123).

Clauses affected: 4.3.1.1, 4.3.1.2.1, 4.1.3.2.2

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

Table 4.3.1-1: 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	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	1	0	0	0
2	1	1	1	0	0	0	0	0	0	1
3	1	0	0	1	0	0	1	0	1	1
4	1	1	0	1	0	0	0	0	0	1
5	1	0	1	1	0	0	0	0	1	0
6	1	1	1	1	0	0	0	1	0	0
7	1	0	0	0	1	0	0	1	1	0
8	1	1	0	0	1	0	1	1	1	0
9	1	0	1	0	1	0	1	0	1	1
10	1	1	1	0	1	0	0	0	1	1
11	1	0	0	1	1	0	0	1	1	0
12	1	1	0	1	1	0	0	1	0	1
13	1	0	1	1	1	0	1	0	0	1
14	1	1	1	1	1	0	1	1	1	1
15	1	1	0	0	0	1	1	1	0	0
16	1	0	1	0	0	1	1	1	0	1
17	1	1	1	0	0	1	1	0	1	0
18	1	0	0	1	0	1	0	1	1	1
19	1	1	0	1	0	1	0	1	0	1
20	1	0	1	1	0	1	0	0	1	1
21	1	1	1	1	0	1	0	1	1	1
22	1	0	0	0	1	1	0	1	0	0
23	1	1	0	0	1	1	1	1	0	1
24	1	0	1	0	1	1	1	0	1	0
25	1	1	1	0	1	1	1	0	0	1
26	1	0	0	1	1	1	0	0	1	0
27	1	1	0	1	1	1	1	1	0	0
28	1	0	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	1	0	0	0	0	0	0	0	0	0
31	1	0	0	0	0	1	1	0	0	0

Let's define the TFCI information bits as a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9 (a_0 is LSB and a_9 is MSB). The TFCI information 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:

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

where i=0…31. N_{TFCI}=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_{TFCI}=4) for a single TFCI bit and 8-bit transmission (N_{TFCI}=8) for 2 TFCI bits. Let's define the TFCI information bit(s) as b_0 (or b_0 and b_1). The TFCI information bit(s) 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 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 bits are encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 4-8.

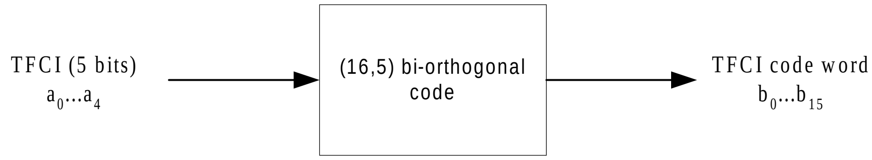


Figure 4-8: Channel coding of short length TFCI bits

The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 4.3.1-2 below.

Table 4.3.1-2: 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	1	0	0	0
1	1	0	1	0	0
2	1	1	1	0	0
3	1	0	0	1	0
4	1	1	0	1	0
5	1	0	1	1	0
6	1	1	1	1	0
7	1	0	0	0	1
8	1	1	0	0	1
9	1	0	1	0	1
10	1	1	1	0	1
11	1	0	0	1	1
12	1	1	0	1	1
13	1	0	1	1	1
14	1	1	1	1	1
15	1	0	0	0	0

Let's define the TFCI information bits as a_0 , a_1 , a_2 , a_3 , a_4 (a_0 is LSB and a_4 is MSB). The TFCI information 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_j are given by:

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

where i=0…15. N_{TFCI}=16.

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0160
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222	**CR**	**021**	Current Version:	3.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ ↑ *CR number as allocated by MCC support team*

For submission to:	RAN #7	for approval	X	strategic		*(for SMG use only)*
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source:	TSG RAN WG1	**Date:**	21-Jan-2000
Subject:	Modification of Turbo code internal interleaver		
Work item:			

Category:	F	Correction		**Release:**	Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature	X		Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification			Release 99	X
					Release 00	

Reason for change: Addition of Turbo code internal interleaver for smaller block size from 40-bit to 319-bit

Clauses affected: 4.2.3.2.3 of TS 25.222

Other specs affected:	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

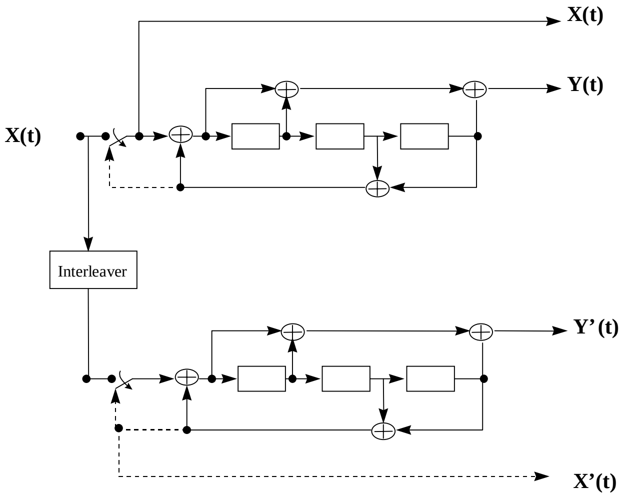


Figure 4-3: Structure of the 8-state PCCC encoder (dotted lines effective for trellis termination only)

The initial value of the shift registers of the PCCC encoder shall be all zeros.

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc.

4.2.3.2.2 Trellis termination in turbo code

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

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4-3 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 4-3 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2).

4.2.3.2.3 Turbo code internal interleaver

Figure 4-4 depicts the overall 8-State PCCC Turbo coding scheme including Turbo code internal interleaver. The Turbo code internal interleaver consists of mother interleaver generation and pruning. For arbitrary given block length K, one mother interleaver is selected from the ~~134~~ 163 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l*-bits are pruned in order to adjust the mother interleaver to the block length K. Tail bits T_1 and T_2 are added for constituent encoders RSC1 and RSC2, respectively. The definition of *l* is shown in section 4.2.3.2.3.2..

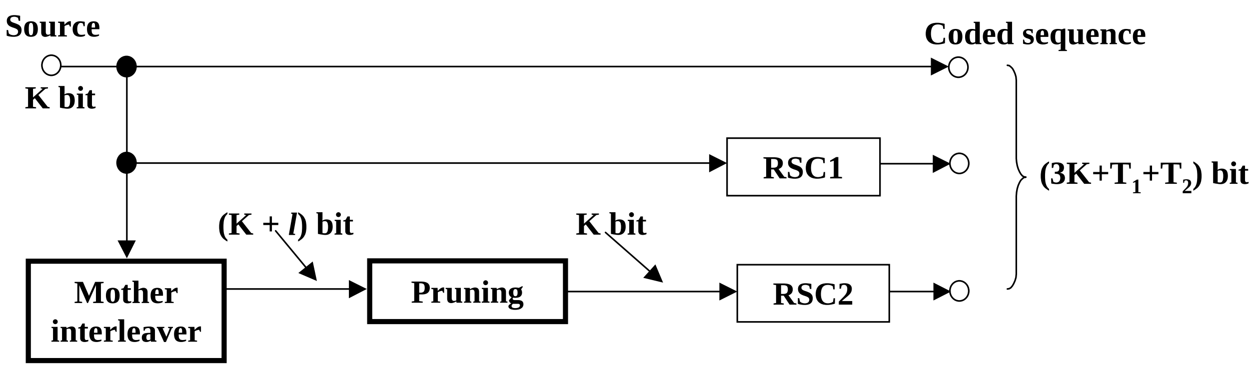


Figure 4-4: Overall 8 State PCCC Turbo Coding

4.2.3.2.3.1 Mother interleaver generation

The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (~~320~~ 40 to 5114 bits).

First Stage:

(1) Determine the number of rows R such that

R = 5 (K = 40 to 159 bits)

R = 10 (K = 160 to 200 bits and 481 to 530 bits~~; Case 1~~)

R = 20 (K = any other block lengths~~ except 481 to 530 bits; Case 2~~)

(2) Determine the number of columns C such that

if K = 481 to 530 then~~Case 1;~~ C = p = 53

else~~Case 2;~~

(i) find minimum prime p such that,

0 =< (p + 1) - K/R

(ii) if (0 =< p - K/R) then go to (iii)

else C = p + 1.

(iii) if (0 =< p - 1 - K/R) then C = p - 1.

else C = p.

(3) The input sequence of the interleaver is written into the R$\times$C rectangular matrix row by row starting from row 0.

Second Stage:

A. If C = p

(A-1) Select a primitive root g_0 from table 4.2.2-2.

(A-2) Construct the base sequence $c(i)$ for intra-row permutation as:

$$c(i) = [g_0 \times c(i-1)] \bmod p,\ i = 1,2,\ldots(p\text{-}2).,\ c(0) = 1.$$

(A-3) Select the minimum prime integer set $\{q_j\}$ (j = 1, 2, …, R-1) such that

g.c.d$\{q_j, p\text{-}1\} = 1$

$q_j > 6$

$q_j > q_{(j-1)}$

where g.c.d. is greatest common divider. And $q_0 = 1$.

(A-4) The set $\{q_j\}$ is permuted to make a new set $\{p_j\}$ such that

$p_{P(j)} = q_j$, j = 0, 1, …. R-1,

where P(j) is the inter-row permutation pattern defined in the third stage.

(A-5) Perform the j-th (j = 0,1, 2, …, C-1) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \bmod (p-1)),\quad i = 0,1,2,\ldots,(p\text{-}2).,\text{ and } c_j(p\text{-}1) = 0,$$

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

If C = p+1

(B-1) Same as case A-1.

(B-2) Same as case A-2.

(B-3) Same as case A-3.

(B-4) Same as case A-4.

(B-5) Perform the j-th (j = 0, 1, 2, …, R-1) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \mathrm{mod}(p-1)), \quad i = 0,1,2,\ldots,(p-2).,\ c_j(p-1) = 0, \text{ and } c_j(p) = p,$$

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

(B-6) If (K = C × R) then exhange $c_{R-1}(p)$ with $c_{R-1}(0)$.

If C = p-1

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \mathrm{mod}(p-1)) -1, \quad i = 0,1,2,\ldots,(p-2).,$$

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

Third Stage:

Perform the inter-row permutation based on the following P(j) (j = 0,1, …, R-1) patterns, where P(j) is the original row position of the j-th permuted row.

P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R = 20

P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R = 20

P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R = 10

P_D:{4, 3, 2, 1, 0} for R = 5

The usage of these patterns is as follows:

Block length K: P(j)

40 to 159-bit: P_D

160 to 200-bit: P_C

201~~320~~ to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

2281 to 2480-bit: P_B

2481 to 3160-bit: P_A

3161 to 3210-bit: P_B

3211 to 5114-bit: P_A

(2) The output of the mother interleaver is the sequence read out column by column from the permuted R × C matrix starting from column 0.

Table 4.2.3-2: Table of prime *p* and associated primitive root g_0

p	g_0	p	g_0	p	g_0	p	g_0	p	g_0
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		

~~p~~	~~g0~~	~~P~~	~~g0~~	~~p~~	~~g0~~	~~P~~	~~G0~~	~~p~~	~~g0~~
~~17~~	~~3~~	~~59~~	~~2~~	~~103~~	~~5~~	~~157~~	~~5~~	~~211~~	~~2~~
~~19~~	~~2~~	~~61~~	~~2~~	~~107~~	~~2~~	~~163~~	~~2~~	~~223~~	~~3~~
~~23~~	~~5~~	~~67~~	~~2~~	~~109~~	~~6~~	~~167~~	~~5~~	~~227~~	~~2~~
~~29~~	~~2~~	~~71~~	~~7~~	~~113~~	~~3~~	~~173~~	~~2~~	~~229~~	~~6~~
~~31~~	~~3~~	~~73~~	~~5~~	~~127~~	~~3~~	~~179~~	~~2~~	~~233~~	~~3~~
~~37~~	~~2~~	~~79~~	~~3~~	~~131~~	~~2~~	~~181~~	~~2~~	~~239~~	~~7~~
~~41~~	~~6~~	~~83~~	~~2~~	~~137~~	~~3~~	~~191~~	~~19~~	~~241~~	~~7~~
~~43~~	~~3~~	~~89~~	~~3~~	~~139~~	~~2~~	~~193~~	~~5~~	~~251~~	~~6~~
~~47~~	~~5~~	~~97~~	~~5~~	~~149~~	~~2~~	~~197~~	~~2~~	~~257~~	~~3~~
~~53~~	~~2~~	~~101~~	~~2~~	~~151~~	~~6~~	~~199~~	~~3~~		

4.2.3.2.3.2 Definition of the number of pruning bits

The output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence. The pruning bits number *l* is defined as:

$$l = R \times C - K,$$

where R is the row number and C is the column number defined in section 4.2.3.2.3.1.

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0226

e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222	**CR**	**023**	Current Version:	V3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑   *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN#7	for approval	X	strategic		*(for SMG use only)*
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG   The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio	X	Core Network	

Source:	TSG RAN WG1	**Date:**	22.02.2000
Subject:	Update of TS 25.222 - clarification of BTFD for TDD		
Work item:	TS 25.222		

Category:	F	Correction	X	**Release:**	Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature			Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification			Release 99	X
					Release 00	

Reason for change:	- Correction concerning the application of the puncturing limit (PL) - Clarification of BTFD for TDD
Clauses affected:	

Other specs affected:	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:	

help.doc

<--------- double-click here for help and instructions on how to create a CR.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] 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 $x-1 < \lfloor x \rfloor \le x$
$\lvert x \rvert$	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.
P	Number of PhCHs used for one CCTrCH.
PL	Puncturing Limit~~ for the uplink~~. 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)sections with different meaning.

x, X
y, Y
z, Z

4.2.13 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.

4.2.13.1 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 possibilty of a TFCI length zero, if only one TFC is defined.

~~Blind transport format detection may be performed in the receiver by trying all possible combinations of the transport format.~~

4.2.13.2 Explicit transport format detection based on TFCI

4.2.13.2.1 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.

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0237
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 025 Current Version: V3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG RAN #7 — for approval [X] / for information [ ] — strategic [ ] / non-strategic [ ] *(for SMG use only)*
list expected approval meeting # here ↑

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source:	TSG RAN WG1
Date:	2000-2-29
Subject:	Change of TFCI basis for TDD
Work item:	

Category: *(only one category Shall be marked With an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: For the most commonality between FDD and TDD TFCI basis, this CR is proposed.

Clauses affected: 4.3.1.1, 4.3.1.2.2

Other specs Affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other GSM core specifications	[ ]	→ List of CRs:
MS test specifications	[ ]	→ List of CRs:
BSS test specifications	[ ]	→ List of CRs:
O&M specifications	[ ]	→ List of CRs:

Other comments:

4.3 Coding for layer 1 control

4.3.1 Coding of transport format combination indicator (TFCI)

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

4.3.1.1 Coding of long TFCI lengths

The TFCI bits are encoded using a (32, 10) sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 4.3.3.1-1.

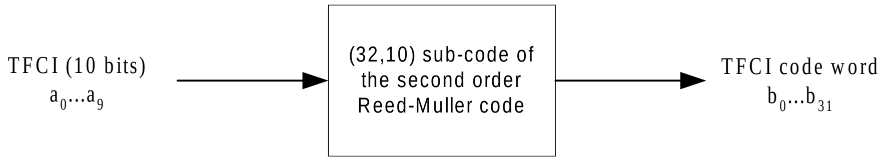


Figure 4.3.3.1-1: Channel coding of TFCI bits

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 4.3.1-1.

Table 4.3.1-1: 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~~1~~	0~~1~~	0~~0~~	0~~0~~	0~~0~~	1~~0~~	0	0	0	0
1	0~~1~~	1~~0~~	0~~1~~	0~~0~~	0~~0~~	1~~0~~	1	0	0	0
2	1~~1~~	1~~1~~	0~~1~~	0~~0~~	0~~0~~	1~~0~~	0	0	0	1
3	0~~1~~	0~~0~~	1~~0~~	0~~1~~	0~~0~~	1~~0~~	1	0	1	1
4	1~~1~~	0~~1~~	1~~0~~	0~~1~~	0~~0~~	1~~0~~	0	0	0	1
5	0~~1~~	1~~0~~	1~~1~~	0~~1~~	0~~0~~	1~~0~~	0	0	1	0
6	1~~1~~	1~~1~~	1~~1~~	0~~1~~	0~~0~~	1~~0~~	0	1	0	0
7	0~~1~~	0~~0~~	0~~0~~	1~~0~~	0~~1~~	1~~0~~	0	1	1	0
8	1~~1~~	0~~1~~	0~~0~~	1~~0~~	0~~1~~	1~~0~~	1	1	1	0
9	0~~1~~	1~~0~~	0~~1~~	1~~0~~	0~~1~~	1~~0~~	1	0	1	1
10	1~~1~~	1~~1~~	0~~1~~	1~~0~~	0~~1~~	1~~0~~	0	0	1	1
11	0~~1~~	0~~0~~	1~~0~~	1~~1~~	0~~1~~	1~~0~~	0	1	1	0
12	1~~1~~	0~~1~~	1~~0~~	1~~1~~	0~~1~~	1~~0~~	0	1	0	1
13	0~~1~~	1~~0~~	1~~1~~	1~~1~~	0~~1~~	1~~0~~	1	0	0	1
14	1~~1~~	1~~1~~	1~~1~~	1~~1~~	0~~1~~	1~~0~~	1	1	1	1
15	1~~1~~	0~~1~~	0~~0~~	0~~0~~	1~~0~~	1~~1~~	1	1	0	0
16	0~~1~~	1~~0~~	0~~1~~	0~~0~~	1~~0~~	1~~1~~	1	1	0	1
17	1~~1~~	1~~1~~	0~~1~~	0~~0~~	1~~0~~	1~~1~~	1	0	1	0
18	0~~1~~	0~~0~~	1~~0~~	0~~1~~	1~~0~~	1~~1~~	0	1	1	1
19	1~~1~~	0~~1~~	1~~0~~	0~~1~~	1~~0~~	1~~1~~	0	1	0	1
20	0~~1~~	1~~0~~	1~~1~~	0~~1~~	1~~0~~	1~~1~~	0	0	1	1
21	1~~1~~	1~~1~~	1~~1~~	0~~1~~	1~~0~~	1~~1~~	0	1	1	1
22	0~~1~~	0~~0~~	0~~0~~	1~~0~~	1~~1~~	1~~1~~	0	1	0	0
23	1~~1~~	0~~1~~	0~~0~~	1~~0~~	1~~1~~	1~~1~~	1	1	0	1
24	0~~1~~	1~~0~~	0~~1~~	1~~0~~	1~~1~~	1~~1~~	1	0	1	0
25	1~~1~~	1~~1~~	0~~1~~	1~~0~~	1~~1~~	1~~1~~	1	0	0	1
26	0~~1~~	0~~0~~	1~~0~~	1~~1~~	1~~1~~	1~~1~~	0	0	1	0
27	1~~1~~	0~~1~~	1~~0~~	1~~1~~	1~~1~~	1~~1~~	1	1	0	0
28	0~~1~~	1~~0~~	1~~1~~	1~~1~~	1~~1~~	1~~1~~	1	1	1	0
29	1~~1~~	1~~1~~	1~~1~~	1~~1~~	1~~1~~	1~~1~~	1	1	1	1
30	0~~1~~	0~~0~~	0~~0~~	0~~0~~	0~~0~~	1~~0~~	0	0	0	0
31	0~~1~~	0~~0~~	0~~0~~	0~~0~~	1~~0~~	1~~1~~	1	0	0	0

For TFCI bits a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9 (a_0 is LSB and a_9 is MSB), the output 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…31. N_{TFCI}=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_{TFCI}=4) for a single TFCI bit and 8-bit transmission (N_{TFCI}=8) for 2 TFCI bits. In the case of two TFCI bits denoted b_0 and b_1 the TFCI 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 bits are encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 4-8.

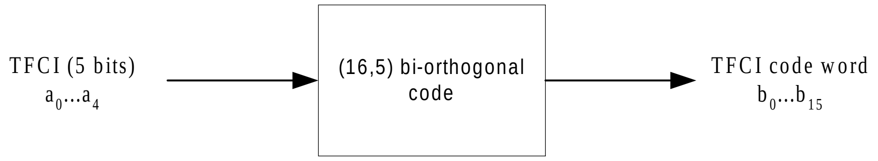


Figure 4-8: Channel coding of short length TFCI bits

The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 4.3.1-2 below.

Table 4.3.1-2: 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~~1~~	0~~1~~	0~~0~~	0~~0~~	1~~0~~
1	0~~1~~	1~~0~~	0~~1~~	0~~0~~	1~~0~~
2	1~~1~~	1~~1~~	0~~1~~	0~~0~~	1~~0~~
3	0~~1~~	0~~0~~	1~~0~~	0~~1~~	1~~0~~
4	1~~1~~	0~~1~~	1~~0~~	0~~1~~	1~~0~~
5	0~~1~~	1~~0~~	1~~1~~	0~~1~~	1~~0~~
6	1~~1~~	1~~1~~	1~~1~~	0~~1~~	1~~0~~
7	0~~1~~	0~~0~~	0~~0~~	1~~0~~	1~~1~~
8	1~~1~~	0~~1~~	0~~0~~	1~~0~~	1~~1~~
9	0~~1~~	1~~0~~	0~~1~~	1~~0~~	1~~1~~
10	1~~1~~	1~~1~~	0~~1~~	1~~0~~	1~~1~~
11	0~~1~~	0~~0~~	1~~0~~	1~~1~~	1~~1~~
12	1~~1~~	0~~1~~	1~~0~~	1~~1~~	1~~1~~
13	0~~1~~	1~~0~~	1~~1~~	1~~1~~	1~~1~~
14	1~~1~~	1~~1~~	1~~1~~	1~~1~~	1~~1~~
15	0~~1~~	0~~0~~	0~~0~~	0~~0~~	1~~0~~

For TFCI information bits a_0, a_1, a_2, a_3, a_4 (a_0 is LSB and a_4 is MSB), the), 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…15. N_{TFCI}=16.

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0255
e.g. for 3GPP use the format TP-99xxx or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 026 Current Version: 3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: WG1 #11	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 29.02.2000

Subject: Padding Function for Turbo coding of small blocks

Work item: TS 25.222

Category: *(only one category shall be marked with an X)*

F	Correction		
A	Corresponds to a correction in an earlier release		
B	Addition of feature	X	
C	Functional modification of feature		
D	Editorial modification		

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: At the last meeting #10 the smallest block size for Turbo coding was set to 40. It was left over to include a functionality to handle blocks of smaller size than 40 bits. This is added in the specification by this CR.

Clauses affected: 4.2.2.2 Code block segmentation

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments: The corresponding CR for FDD mode is 25.212-052.

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 Z, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depend on if convolutional or turbo 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:

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

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

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

…

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

4.2.2.2 Code block segmentation

~~NOTE: It is assumed that filler bits are set to 0.~~

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 ~~last~~ 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 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.

Number of code blocks: $C_i = \lceil X_i / Z \rceil$

Number of bits in each code block:

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: $Y_i = C_iK_i - X_i$

If $X_i \le Z$, then

$$o_{i1k} = 0 \qquad k = 1, 2, \ldots, Y_i$$

$$o_{i1k} = x_{i,(k-Y_i)} \quad \cancel{o_{i1k} = x_{ik}} \qquad k = Y_i+1, Y_i+2, \ldots, K_i$$

end if

~~, and $K_i = X_i$.~~

If X_i > ~~≥~~ Z, then

$o_{i1k} = 0$ $k = 1, 2, \ldots, Y_i$

$o_{i1k} = x_{i,(k-Y_i)}$ $\cancel{o_{i1k} = x_{ik}}$ $k = Y_i+1, Y_i+2, \ldots, K_i$

$o_{i2k} = x_{i,(k+K_i-Y_i)}$ $\cancel{o_{i2k} = x_{i,(k+K_i)}}$ $k = 1, 2, \ldots, K_i$

$o_{i3k} = x_{i,(k+2K_i-Y_i)}$ $\cancel{o_{i3k} = x_{i,(k+2K_i)}}$ $k = 1, 2, \ldots, K_i$

…

$o_{iC_ik} = x_{i,(k+(C_i-1)K_i-Y_i)}$ $\cancel{o_{iC_ik} = x_{i(k+(C_i-1)K_i)}}$ $k = 1, 2, \ldots, K_i$ $\cancel{-Y_i}$

$\cancel{o_{iC_ik} = 0 \quad k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \ldots, K_i}$

end if

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0282
e.g. for 3GPP use the format TP-99xxx or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 027 Current Version: V3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG-RAN #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** Feb 24th, 2000

Subject: Editorial modification of shifting parameter calculation for turbo code puncturing

Work item:

Category:				**Release:**	
	F	Correction		Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
(only one category	B	Addition of feature		Release 97	
shall be marked	C	Functional modification of feature		Release 98	
with an X)	D	Editorial modification	X	Release 99	X
				Release 00	

Reason for change: When ΔN_i is calculated as 0 for parity sequence of turbo code, then q value cannot be defined for the corresponding parity sequence. Actually, in that case nothing is necessary to be done in the rate matching block for that sequence. Therefore some clarification may be necessary.

Clauses affected: 4.2.7.1.2 Turbo encoded TrCHs

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

<--------- double-click here for help and instructions on how to create a CR.

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 section 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 section 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($q \le 2$)

for x=0 to F_i-1

$S[I_F[(3x+b-1) \bmod F_i]] = x \bmod 2$; 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 that q' is not an integer, but a multiple of 1/8

else $q' = q$

endif

for x=0 to F_i –1

$r = \lceil x*q' \rceil \bmod F_i$;

$S[I_F[(3r+b-1) \bmod F_i]] = \lceil x*q' \rceil \operatorname{div} F_i$;

endfor

endif

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

X_i is as above,

$e_{ini} = (a \cdot S(n_i) \cdot |\Delta N_i| + X_i) \bmod (a \cdot X_i)$, if $e_{ini} = 0$ then $e_{ini} = a \cdot X_i$.

$e_{plus} = a \cdot X_i$

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

3GPP TSG RAN Meeting #7
Madrid, Spain, 13-15 March 2000

Document R1-00-0437
e.g. for 3GPP use the format TP-99xxx or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222	**CR**	**029r1**	Current Version:	3.1.1

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	RAN #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 3-Mar-2000

Subject: Editorial changes of channel coding section

Work item:

Category: *(only one category shall be marked with an X)*

F	Correction		Phase 2	
A	Corresponds to a correction in an earlier release		Release 96	
B	Addition of feature		Release 97	
C	Functional modification of feature		Release 98	
D	Editorial modification	X	Release 99	X
			Release 00	

Release: see above

Reason for change: To clarify exact functions of channel coding.

Clauses affected: 4.2.3 of TS25.222

Other specs affected:

Other 3G core specifications		→ List of CRs:	
Other GSM core specifications		→ List of CRs:	
MS test specifications		→ List of CRs:	
BSS test specifications		→ List of CRs:	
O&M specifications		→ List of CRs:	

Other comments: This CR is including content of approved CR25.222-021 and changes for Table 4.2.3.1 as stated in CR25.222-017.

$$o_{iC_ik} = x_{i(k+(C_i-1)K_i)} \quad k = 1, 2, \ldots, K_i - Y_i$$

$$o_{iC_ik} = 0 \quad k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \ldots, K_i$$

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 encoded blocks are serially multiplexed so that the block with lowest index r is output first from the channel coding block. 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:~~

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

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

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

~~…~~

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

The relation between o_{irk} and ~~Y~~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 ~~channel~~ coding

Usage of coding scheme and coding rate for the different types of TrCH is shown in table 4.2.3-1. The values of Y_i in connection with each coding scheme:

- Convolutional coding~~, ½~~ with rate 1/2: $Y_i = 2*K_i + 16$; rate 1/3 ~~rate~~: $Y_i = 3*K_i + 24$
- Turbo coding~~,~~ with rate 1/3 ~~rate~~: $Y_i = 3*K_i + 12$
- No ~~channel~~ coding~~,~~: $Y_i = K_i$

Table 4.2.3-1: Usage of channel coding scheme and coding rate ~~Error Correction Coding Parameters~~

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

~~Transport channel type~~	~~Coding scheme~~	~~Coding rate~~
~~BCH~~	~~Convolutional code~~	~~1/2~~
~~PCH~~		
~~FACH~~		
~~RACH~~		
~~DCH, DSCH, USCH~~		~~1/3, 1/2~~
	~~Turbo code~~	~~1/3~~
	~~No coding~~	

4.2.3.1 Convolutional ~~C~~coding

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

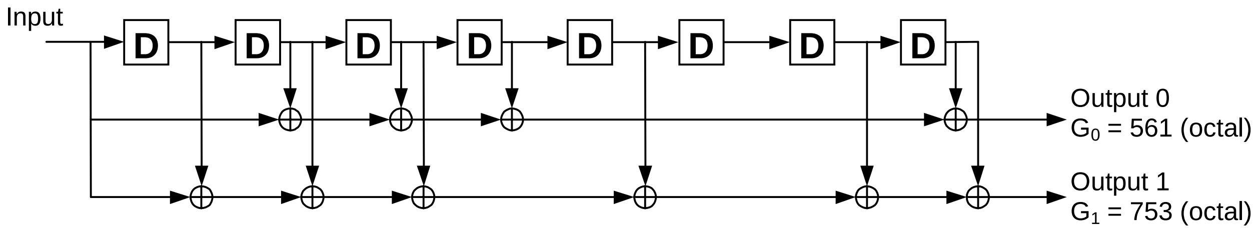
The configuration of the convolutional coder is presented in figure 4-2.

~~The o~~Output from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output2, output0, …, output2. ~~(When coding~~ Output from the rate ~~is~~ 1/2 convolutional coder shall be done in the order~~,~~ output0, output1, output0, output1, output0, …, ~~is done up to~~ output 1~~)~~.

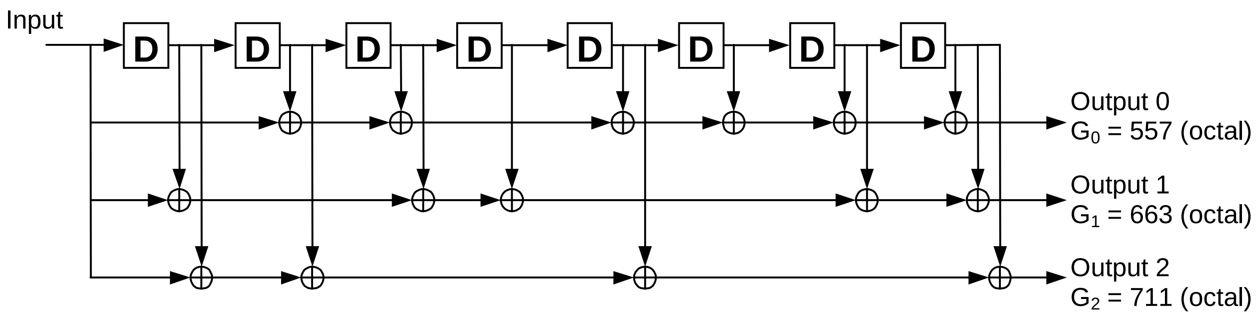
8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.

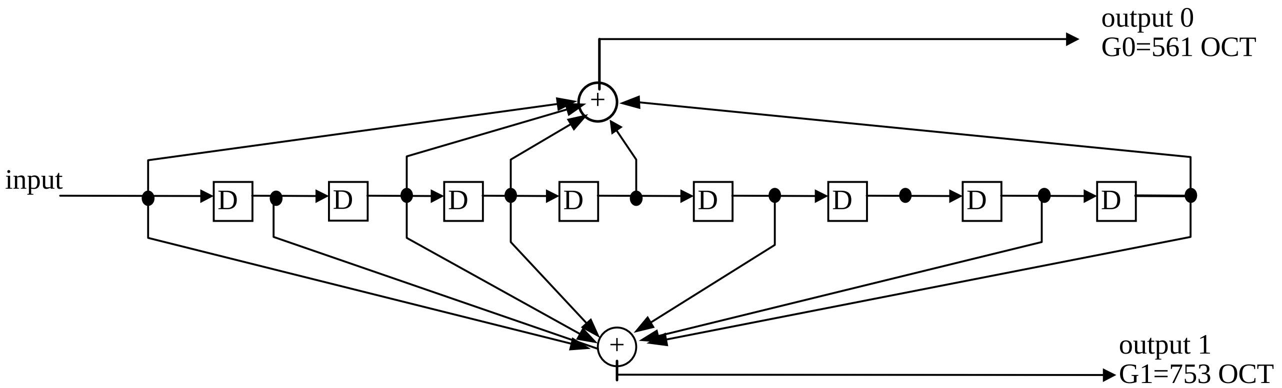
~~K-1 tail bits (value 0) shall be added to the end of the code block before encoding.~~



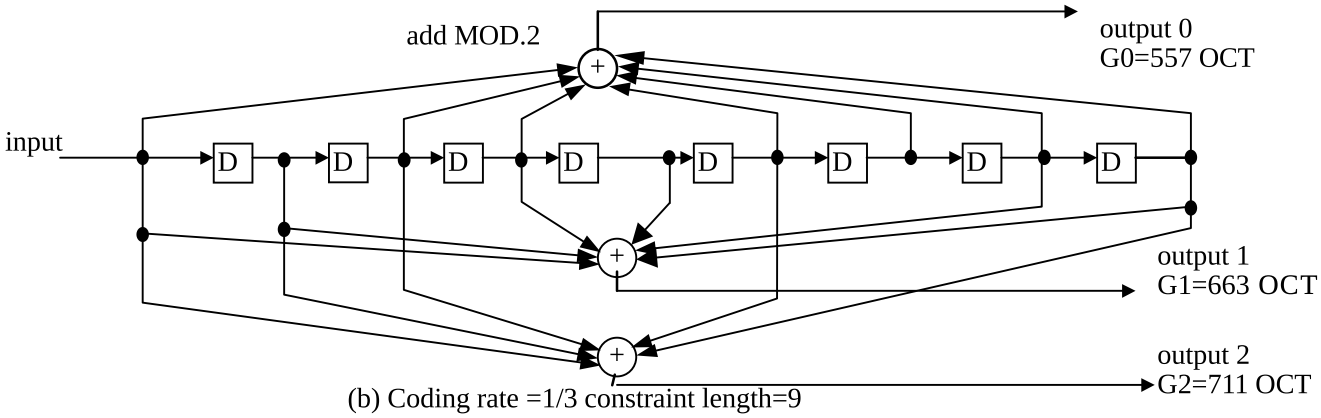
(a) Rate 1/2 convolutional coder



(b) Rate 1/3 convolutional coder



(a) Coding rate =1/2 constraint length=9



(b) Coding rate =1/3 constraint length=9

Figure 4-2: Rate 1/2 and rate 1/3 ~~C~~convolutional ~~C~~coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The scheme of Turbo coder is a ~~For data services requiring quality of service between 10^{-3} and 10^{-6} BER inclusive,~~ ~~p~~Parallel ~~c~~Concatenated ~~c~~Convolutional ~~c~~Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver~~is used~~. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 4-3.

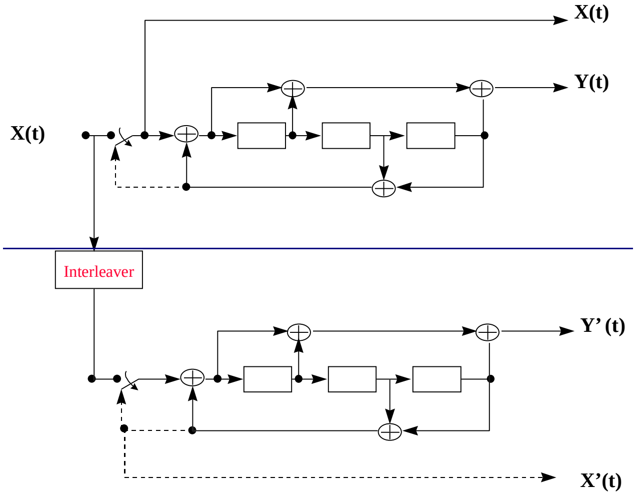
The transfer function of the 8-state constituent code for PCCC is

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

where~~,~~

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

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



~~**Figure 4-3: Structure of the 8-state PCCC encoder (dotted lines effective for trellis termination only)**~~

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

~~The o~~Output ~~of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence~~ from the Turbo coder is ~~X(0), Y(0), 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.

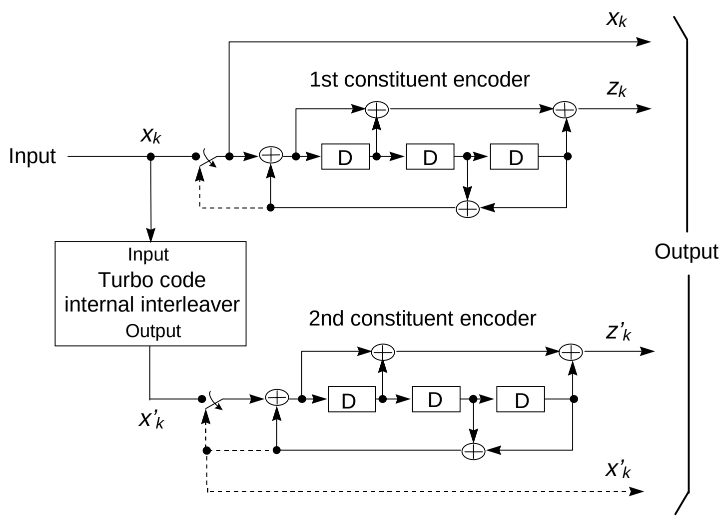


Figure 4-3: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

4.2.3.2.2 Trellis termination ~~in~~for ~~t~~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.

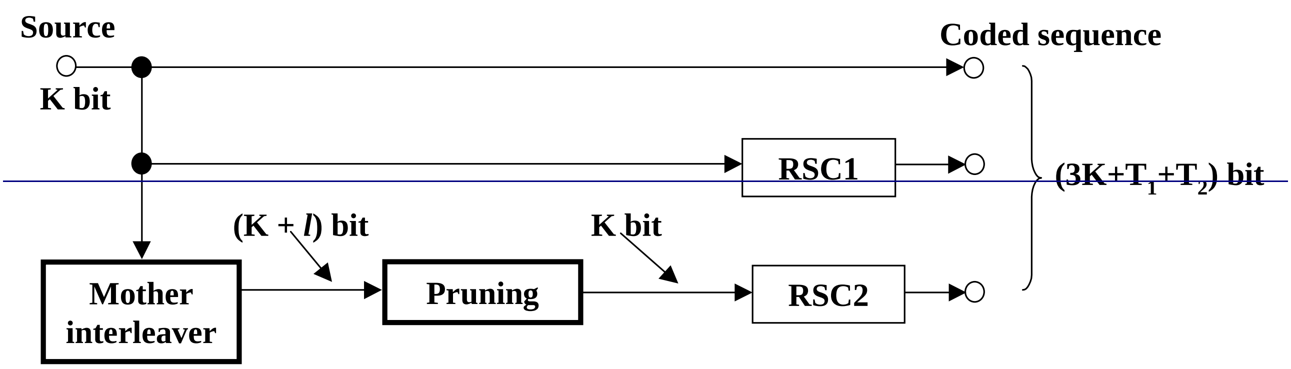
The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4-3 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4-3 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

~~X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2)~~ $x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$.

4.2.3.2.3 Turbo code internal interleaver

~~Figure 4-4 depicts the overall 8-State PCCC Turbo coding scheme including Turbo code internal interleaver.~~ The Turbo code internal interleaver consists of bits-input to a rectangular matrix, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with pruning.~~mother interleaver generation and 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 $K = K_i$.~~For arbitrary given block length K, one mother interleaver is selected from the 134 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l* bits are pruned in order to adjust the mother interleaver to the block length K. Tail bits T_1 and T_2 are added for constituent encoders RSC1 and RSC2, respectively. The definition of *l* is shown in section 4.2.3.2.3.2..~~



~~Figure 4-4: Overall 8 State PCCC Turbo Coding~~

The following section specific symbols are used in sections 4.2.3.2.2.3.1 – 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
$s(i)$	Base sequence for intra-row permutation
q_j	Minimum prime integers
r_j	Permuted prime integers
$T(j)$	Inter-row permutation pattern
$U_j(i)$	Intra-row permutation pattern
i	Index of matrix
j	Index of matrix
k	Index of bit sequence

4.2.3.2.3.1 Bits-input to rectangular matrix~~Mother interleaver generation~~

The bit sequence input to the Turbo code internal interleaver x_k ~~The interleaving consists of three stages. In first stage, the input sequence~~ is written into the rectangular matrix as follows: ~~row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320 to 5114 bits).~~

~~First Stage:~~

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

$$R = \begin{cases} 5, \text{if } (40 \le K \le 159) \\ 10, \text{if } ((160 \le K \le 200) \text{ or } (481 \le K \le 530)) \\ 20, \text{if } (K = \text{any other value}) \end{cases}$$ ~~R=10 (K = 481 to 530 bits; Case-1)~~

~~R=20 (K = any other block length except 481 to 530 bits; Case-2)~~

where the rows of rectangular matrix are numbered 0, 1, 2, …, R - 1 from top to bottom.

(2) Determine the number of columns C of rectangular matrix such that

if (481 $\le K \le$ 530) then
  $p = 53$ and $C = p$.
else
~~Case-1; C = p = 53~~

~~Case-2;~~

  ~~(i) f~~Find minimum prime p such that~~,~~
    ~~0 =<~~ $(p + 1) - K/R \ge 0$,
  and determine C such that
  ~~(ii)~~ if (~~0 =<~~ $p - K/R \ge 0$) then ~~go to (iii)~~
    if ($p - 1 - K/R \ge 0$) then
      $C = p - 1$.
    else
      $C = p$.
    end if
  else
    $C = p + 1$.
  end if
end if

where the columns of rectangular matrix are numbered 0, 1, 2, …, C - 1 from left to right.

~~(iii) if (0 =< p-1-K/R) then C=p-1.~~

~~Else C = p.~~

(3) Write ~~T~~the input bit sequence x_k ~~of the interleaver is written~~ into the $R \times C$ rectangular matrix row by row starting with bit x_1 ~~from~~ in column 0 of row 0~~.~~:

$$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots x_C \\ x_{(C+1)} & x_{(C+2)} & x_{(C+3)} & \cdots x_{2C} \\ \vdots & \vdots & \vdots & \cdots \vdots \\ x_{((R-1)C+1)} & x_{((R-1)C+2)} & x_{((R-1)C+3)} & \cdots x_{RC} \end{bmatrix}.$$

~~**Second Stage:**~~

~~A. If C = p~~

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 are performed by using the following algorithm:

(1) ~~(A-1)~~ Select a primitive root ~~g_0~~ v from table 4.2.~~2~~3-2.

(2) ~~(A-2)~~ Construct the base sequence ~~c~~$s(i)$ for intra-row permutation as~~:~~

~~$c(i) = [g_0 \times c(i-1)] \bmod p$~~ $s(i) = [v \times s(i-1)] \bmod p$, $i = 1, 2, \ldots, (p-2)$., and ~~c~~$s(0) = 1$.

(3) ~~(A-3)~~ Let $q_0 = 1$ be the first prime integer in $\{q_j\}$, and ~~S~~select the consecutive minimum prime integers ~~set~~ $\{q_j\}$ ($j = 1, 2, \ldots, R-1$) such that

g.c.d$\{q_j, p-1\} = 1$,

$q_j > 6$, and

$q_j > q_{(j-1)}$.

where g.c.d. is greatest common ~~divider~~divisor. ~~And $q_0 = 1$.~~

(4) ~~(A-4)~~ Permute ~~The set~~ $\{q_j\}$ ~~is permuted~~ to make ~~a new set~~ $\{$~~p~~$r_j\}$ such that

~~$p_{P(j)}$~~ $r_{T(j)} = q_j$, $j = 0, 1, \ldots, R-1$,

where ~~P~~$T(j)$ indicates the original row position of the j-th permuted row, and $T(j)$ is the inter-row permutation pattern defined as the one of the following four kind of patterns: Pat_1, Pat_2, Pat_3 and Pat_4 depending on the number of input bits K. ~~in the third stage.~~

$$T(j) = \begin{cases} Pat_4 & \text{if } (40 \le K \le 159) \\ Pat_3 & \text{if } (160 \le K \le 200) \\ Pat_1 & \text{if } (201 \le K \le 480) \\ Pat_3 & \text{if } (481 \le K \le 530) \\ Pat_1 & \text{if } (531 \le K \le 2280) \\ Pat_2 & \text{if } (2281 \le K \le 2480) \\ Pat_1 & \text{if } (2481 \le K \le 3160) \\ Pat_2 & \text{if } (3161 \le K \le 3210) \\ Pat_1 & \text{if } (3211 \le K \le 5114) \end{cases},$$

where Pat_1, Pat_2, Pat_3 and Pat_4 have the following patterns respectively.

Pat_1: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}
Pat_2: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10}
Pat_3: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}
Pat_4: {4, 3, 2, 1, 0}

(5) ~~(A-5)~~ Perform the j-th (j = 0, 1, 2, …, ~~C~~ R - 1) intra-row permutation as~~:~~

if ($C = p$) then

$\cancel{c_j(i) = c([i \times p_j] \bmod (p-1))}$ $U_j(i) = s([i \times r_j] \bmod (p - 1))$, i = 0, 1, 2, …, (p - 2), and ~~c~~$U_j(p - 1) = 0$,

where ~~c~~$U_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

~~I~~if (C = p + 1) then

~~(B-1) Same as case A-1.~~

~~(B-2) Same as case A-2.~~

~~(B-3) Same as case A-3.~~

~~(B-4) Same as case A-4.~~

~~(B-5) Perform the *j*-th (*j* = 0,1, 2, …, R-1) intra-row permutation as:~~

$\cancel{c_j(i) = c([i \times p_j] \bmod (p-1))}$ $U_j(i) = s([i \times r_j] \bmod (p - 1))$, i = 0, 1, 2, …, (p - 2), ~~c~~$U_j(p - 1) = 0$, and ~~c~~$U_j(p) = p$,

where ~~c~~$U_j(i)$ is the input bit position of i-th output after the permutation of j-th row~~.~~, and

~~(B-6) I~~if ($K = C \times R$) then

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

end if

end if

~~I~~if ($C = p$ - 1) then

~~(C-1) Same as case A-1.~~

~~(C-2) Same as case A-2.~~

~~(C-3) Same as case A-3.~~

~~(C-4) Same as case A-4.~~

~~(C-5) Perform the *j*-th (*j* = 0,1, 2, …, R-1) intra-row permutation as:~~

$\cancel{c_j(i) = c([i \times p_j] \bmod (p-1))}$ $U_j(i) = s([i \times r_j] \bmod (p - 1))$ - 1, i = 0, 1, 2, …, (p - 2)~~.~~,

where ~~c~~$U_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

end if

~~**Third Stage:**~~

~~Perform the inter-row permutation based on the following P(*j*) (*j*=0,1, …, R-1) patterns, where P(*j*) is the original row position of the *j*-th permuted row.~~

~~P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R=20~~

~~P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R=20~~

~~P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R=10~~

~~The usage of these patterns is as follows:~~

~~Block length K: P(*j*)~~

~~320 to 480-bit: P_A~~

~~481 to 530-bit: P_C~~

~~531 to 2280-bit: P_A~~

~~2281 to 2480-bit: P_B~~

~~2481 to 3160-bit: P_A~~

~~3161 to 3210-bit: P_B~~

~~3211 to 5114-bit: P_A~~

~~(2) The output of the mother interleaver is the sequence read out column by column from the permuted R × C matrix starting from column 0.~~

Table 4.2.3-2: Table of prime *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		

~~p~~	~~g0~~	~~P~~	~~g0~~	~~p~~	~~g0~~	~~P~~	~~g0~~	~~p~~	~~g0~~
~~17~~	~~3~~	~~59~~	~~2~~	~~103~~	~~5~~	~~157~~	~~5~~	~~211~~	~~2~~
~~19~~	~~2~~	~~61~~	~~2~~	~~107~~	~~2~~	~~163~~	~~2~~	~~223~~	~~3~~
~~23~~	~~5~~	~~67~~	~~2~~	~~109~~	~~6~~	~~167~~	~~5~~	~~227~~	~~2~~
~~29~~	~~2~~	~~71~~	~~7~~	~~113~~	~~3~~	~~173~~	~~2~~	~~229~~	~~6~~
~~31~~	~~3~~	~~73~~	~~5~~	~~127~~	~~3~~	~~179~~	~~2~~	~~233~~	~~3~~
~~37~~	~~2~~	~~79~~	~~3~~	~~131~~	~~2~~	~~181~~	~~2~~	~~239~~	~~7~~
~~41~~	~~6~~	~~83~~	~~2~~	~~137~~	~~3~~	~~191~~	~~19~~	~~241~~	~~7~~
~~43~~	~~3~~	~~89~~	~~3~~	~~139~~	~~2~~	~~193~~	~~5~~	~~251~~	~~6~~
~~47~~	~~5~~	~~97~~	~~5~~	~~149~~	~~2~~	~~197~~	~~2~~	~~257~~	~~3~~
~~53~~	~~2~~	~~101~~	~~2~~	~~151~~	~~6~~	~~199~~	~~3~~		

4.2.3.2.3.3~~2~~ Bits-output from rectangular matrix with ~~Definition of the number of~~ pruning ~~bits~~

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

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

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 $R \times C$ 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 bits that were not present in the input bit sequence, i.e. bits y'_k that corresponds to bits x_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 output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence.~~ The number of bits output from Turbo code internal interleaver is K and ~~T~~the total number of pruneding bits ~~number *l*~~ is ~~defined as:~~

~~l =~~ $R \times C - K$.

~~where R is the row number and C is the column number defined in section 4.2.3.2.3.1.~~

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:

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

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

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

$$\ldots$$

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

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