3GPP TSG RAN WG1 (Radio) Meeting #9
Dresden, Germany, 30 Nov - 3 Dec 1999

***Document* R1-99k04**
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 004r1 Current Version: V3.0.0

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

For submission to: RAN #6	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: Nokia, Siemens   **Date:** 30 Nov 1999

Subject: Editorial corrections to TS 25.222

Work item:

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

F	Correction		**Release:**	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	

Reason for change:
- Old document numbers corrected in the list of references
- One term added to list of abbreviations
- 'No coding' –option addressed as a separate coding scheme in table 4.2.3.1
- Description of figure 4.4 has more detail added
- The order of reading out data from turbo code internal interleaver is clarified
- a couple of small typographical errors are corrected

Clauses affected: 2, 3.1, 3.3, 4.2.2, 4.2.3, 4.2.3.2.1, 4.2.3.2.3, 4.2.6

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 new document 25.222 CR004r1 is a merge of Nokia CR004 and Siemens CR005

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

1 Scope

This 3GPP Report describes multiplexing, channel coding and interleaving for UTRA Physical Layer TDD mode.

Text without revision marks has been approved in the previous TSG-RAN WG1 meetings, while text with revision marks is subject to approval.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] TS 25.202 ~~(V1.0.0)~~: “UE capabilities”

[2] TS 25.211 ~~(V1.0.0)~~: “Transport channels and physical channels (FDD)”

[3] TS 25.212 ~~(V1.0.0)~~: “Multiplexing and channel coding (FDD)”

[4] TS 25.213 ~~(V1.0.0)~~: “Spreading and modulation (FDD)”

[5] TS 25.214 ~~(V1.0.0)~~: “Physical layer procedures (FDD)”

[6] TS 25.215: “Physical layer – Measurements (FDD)”

[7] TS 25.221 ~~(V1.0.0)~~: “Transport channels and physical channels (TDD)”

~~[9]TS 25.222 (V1.0.0): “Multiplexing and channel coding (TDD)”~~

~~[10]~~[8] TS 25.223 ~~(V1.0.0)~~: “Spreading and modulation (TDD)”

~~[11]~~[9] TS 25.224 ~~(V1.0.0)~~: “Physical layer procedures (TDD)”

~~[12]~~[10] TS 25.225~~31 (V1.0.0)~~: “Physical layer Measurements (TDD)”

~~[13]~~[11] TS S2.01 ~~(V1.0.0)~~: “Radio Interface Protocol Architecture”

3 Definitions, ~~S~~symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

<defined term>: <definition>.

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:

éxù	round towards ¥, i.e. integer such that *x* £ *éxù* < *x*+1
ëxû	round towards -¥, i.e. integer such that *x*-1 < *ëxû* £ *x*
çxç	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

3.3 Abbreviations

ARQ	Automatic Repeat on Request
BCH	Broadcast Channel
BER	Bit Error Rate
~~BPSK~~	~~Binary Phase Shift Keying~~
BS	Base Station
BSS	Base Station Subsystem
~~CA~~	~~Capacity Allocation~~
~~CAA~~	~~Capacity Allocation Acknowledgement~~
CBR	Constant Bit Rate
CCCH	Common Control Channel
CCTrCH	Coded Composite Transport Channel
~~CD~~	~~Capacity Deallocation~~
~~CDA~~	~~Capacity Deallocation Acknowledgement~~
CDMA	Code Division Multiple Access
~~CTDMA~~	~~Code Time Division Multiple Access~~
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
~~HCS~~	~~Hierarchical Cell Structure~~
JD	Joint Detection
L1	Layer 1
L2	Layer 2
LLC	Logical Link Control
MA	Multiple Access
MAC	Medium Access Control
~~MAHO~~	~~Mobile Assisted Handover~~
~~MO~~	~~Mobile Originated~~
~~MOHO~~	~~Mobile Originated Handover~~
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
~~SDCCH~~	~~Stand-alone Dedicated Control Channel~~
~~SFN~~	~~System Frame Number~~
SNR	Signal to Noise Ratio
~~SP~~	~~Switching Point~~
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

4.2.2 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,~~ the maximum size of a code block , then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on ~~if~~whether convolutional ~~or~~ ,turbo coding or no coding is used for the TrCH.

4.2.2.1 Concatenation of transport blocks

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

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

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

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

…

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

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 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: $K_i = \lceil X_i / C_i \rceil$

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

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

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

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

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

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

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 $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$. 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, …, Y_i

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

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

…

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

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~~		
~~FACH~~		
~~RACH~~		
~~DCH~~		~~1/3, 1/2, or no coding~~
~~DCH~~	~~Turbo code~~	~~1/3, or 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	

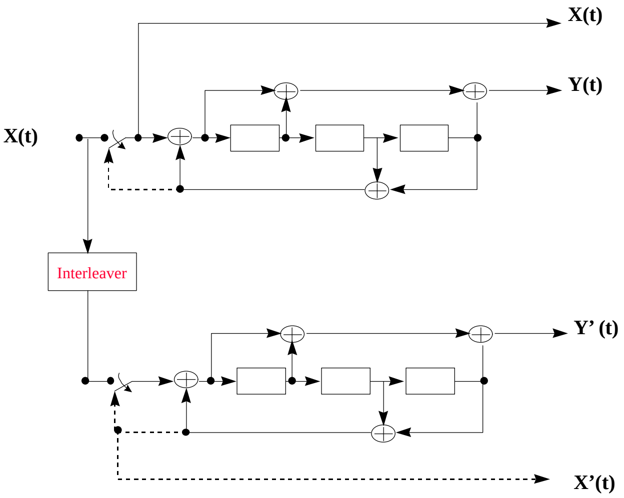

Figure 4-3. Structure of the 8-state PCCC encoder (dotted lines effective for trellis termination only)

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 ~~1/3~~. 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),

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 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.

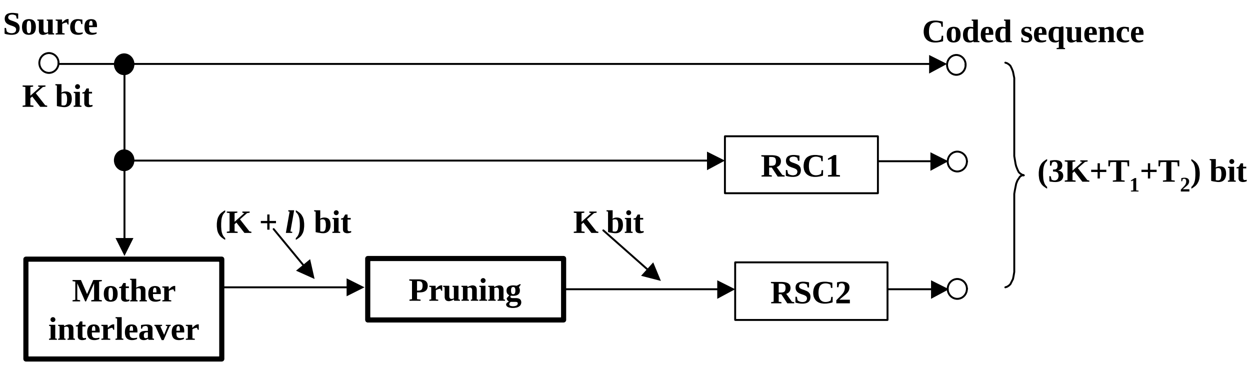

Figure 4-4. Overall 8 State PCCC Turbo Coding

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 to 5114 bits).

First Stage:

(1) Determine the number of rows~~a row number~~ R such that

R=10 (K = 481 to 530 bits; Case-1)

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

(2) Determine the number of columns~~a column number~~ C such that

Case-1; C = p = 53

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 RxC 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,…(p-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:

$c_j(i) = c([i \times p_j] \bmod (p-1))$, i =0,1,2,…, (p-2)., and $c_j(p-1) = 0$,

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

<u>If C = *p*+1</u>

(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:

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

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 x R) then exhange $c_{R-1}(p)$ with $c_{R-1}(0)$.

<u>If C = p-1</u>

(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:

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

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

<u>Third Stage:</u>

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 <u>starting from column 0</u>.

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

p	g_o	~~*P*~~*P*	g_o	*p*	g_o	~~*P*~~*P*	g_o	*p*	g_o
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.

4.2.4 Radio frame size equalisation

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

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

$t_{ik} = c_{ik},$ for k = 1… E_i and

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

where

$T_i = F_i * N_i$ and

$N_i = \lfloor (E_i - 1)/F_i \rfloor + 1$ is the number of bits per segment after size equalisation.

4.2.5 1st interleaving

The 1st interleaving is a block interleaver with inter-column permutations. The input bit sequence to the 1st interleaver is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where *i* is TrCH number and X_i the number of bits (at this stage X_i is assumed and guaranteed to be an integer multiple of TTI). The output bit sequence is derived as follows:

1) Select the number of columns C_I from table 4.2.5-1.

2) Determine the number of rows R_I defined as $R_I = X_i/C_I$

3) Write the input bit sequence into the $R_I \times C_I$ rectangular matrix row by row starting with bit $x_{i,1}$ in the first column of the first row and ending with bit $x_{i,(R_I C_I)}$ in column C_I of row R_I:

$$\begin{bmatrix} x_{i1} & x_{i2} & x_{i3} & \ldots & x_{iC_I} \\ x_{i,(C_I+1)} & x_{i,(C_I+2)} & x_{i,(C_I+3)} & \ldots & x_{i,(2C_I)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{i,((R_I-1)C_I+1)} & x_{i,((R_I-1)C_I+2)} & x_{i,((R_I-1)C_I+3)} & \ldots & x_{i,(R_IC_I)} \end{bmatrix}$$

4) Perform the inter-column permutation based on the pattern $\{P_1(j)\}$ (j=0,1, …, C-1) shown in table 4.2.5-1, where $P_1(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{ik}:

$$\begin{bmatrix} y_{i1} & y_{i,(R_I+1)} & y_{i,(2R_I+1)} & \ldots & y_{i,((C_I-1)R_I+1)} \\ y_{i2} & y_{i,(R_I+2)} & y_{i,(2R_I+2)} & \ldots & y_{i,((C_I-1)R_I+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{iR_I} & y_{i,(2R_I)} & y_{i,(3R_I)} & \ldots & y_{i,(C_IR_I)} \end{bmatrix}$$

5) Read the output bit sequence $y_{i1}, y_{i2}, y_{i3}, \ldots, y_{i,(C_IR_I)}$ of the 1st interleaving column by column from the inter-column permuted $R_I \times C_I$ matrix. Bit $y_{i,1}$ corresponds to the first row of the first column and bit $y_{i,(R_IC_I)}$ corresponds to row R_I of column C_I.

The bits input to the 1st interleaving are denoted by $t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$, where i is the TrCH number and ~~E~~$\underline{T}_i$ the number of bits. Hence, $x_{ik} = t_{ik}$ and $X_i = T_i$.

The bits output from the 1st interleaving are denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$, and $d_{ik} = y_{ik}$.

Table 4.2.5-1

TTI	Number of columns C_1	Inter-column permutation patterns
10 ms	1	{0}
20 ms	2	{0,1}
40 ms	4	{0,2,1,3}
80 ms	8	{0,4,2,6,1,5,3,7}

4.2.6 Radio frame segmentation

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

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

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

where

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

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

y_{i,n_ik} is the k[th] bit of the output bit sequence corresponding to the n[th] radio frame