TSG-RAN Meeting #10
Bangkok, Thailand, 6 - 8 December 2000

RP-000543

Title: **Agreed CRs to TS 25.222**

Source: **TSG-RAN WG1**

Agenda item: **5.1.3**

No.	R1 T-doc	Spec	CR	Rev	Subject	Cat	V_old	V_new
1	R1-001277	25.222	049	-	Clarification on the Ci formula	F	3.4.0	3.5.0
2	R1-001003	25.222	050	-	Correction on TFCI & TPC Transmission	F	3.4.0	3.5.0
3	R1-001477	25.222	053	1	Editorial corrections in TS 25.222	F	3.4.0	3.5.0

3GPP TSG RAN Meeting #10
Bangkok, Thailand, 6-8, December 2000

Document *R1-00-1277*
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	049	Current Version:	3.4.0

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

For submission to:	RAN#10	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:** 2000-10

Subject: Clarification on the C_i formula

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:

- The value of $\lceil X_i / Z \rceil$ for Z = *unlimited* is not so obvious, all the more that $x \mapsto \lceil x \rceil$ is not a continuous function.
- The code block size computation is not applicable when the number of code block is null
- Y_I was replaced by Y_i somewhere in the section.

Clauses affected: 4.2.2.2

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: This also clarify that the padding up to 40 bits for turbo coding is not applicable for C_i = 0.

4.2.2.2 Code block segmentation

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

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

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

Number of code blocks:

~~$C_i = \lceil X_i / Z \rceil$~~

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

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

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

$K_i = 40$

else

$K_i = \lceil X_i / C_i \rceil$

end if

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

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

$o_{i1k} = 0$

end for

for $k = Y_i + 1$ to K_i

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

end for

$r = 2$ -- Segmentation

while $r \leq C_i$

for $k = 1$ to K_i

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

end for

r = r+1

end while

3GPP TSG RAN Meeting #10
Bangkok, Thailand, 6-8, December 2000

Document R1-00-1003
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	050	Current Version:	3.4.0

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

For submission to:	RAN#10	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:** 2000-10-29

Subject: Correction on TFCI & TPC Transmission

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: Fixes the SF for the TFCI and TPC field to 16 in order to enable high repetition factors on the data field in case of low bit rates and low fixed SF.

Clauses affected: 4.2.7.1

Other specs Affected:

Other 3G core specifications	X	→ List of CRs:	CR221-034
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 clarifies the usage of minimum SF in the RM section.


help.doc

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

4.2.7.1 Determination of rate matching parameters

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

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

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

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

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

Denote the number of data bits in each physical channel by $U_{p,Sp}$, where p refers to the sequence number $1 \le p \le P_{max}$ of this physical channel in the allocation message, and the second index Sp indicates the spreading factor with the possible values {*16, 8, 4, 2, 1*}, respectively. For each physical channel an individual minimum spreading factor Sp_{min} is transmitted by means of the higher layer. There are two options to determine the number N_{data} from the number of possible physical channels and the minimum spreading factor. The option is signalled by means of the higher layers.

If the UE shall increase the spreading factor autonomously, ~~Then,~~ for N_{data} one of the following values in ascending order can be chosen:

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

If the UE shall only use the minimum spreading factor, for N_{data} one of the following values in ascending order can be chosen:

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

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

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

$N_{data,j}$ = min SET1

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

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

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

3GPP TSG RAN Meeting #10
Bangkok, Thailand, 6-8, December 2000

Document R1-00-1477
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	053r1	Current Version:	3.4.0

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

For submission to:	RAN #10	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:** 21-Nov.-2000

Subject: Editorial corrections in TS 25.222

Work item:

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

F	Correction	X	**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			Release 99	X
				Release 00	

Reason for change:
To correct wording in CRC attachment, 1st interleaving and 2nd interleaving sections.
To clarify bits padding and pruning for rectangular matrix of 2nd interleaving.
To align mathematical notations with preferred notations shown in TS 25.201 Annex A.

Clauses affected: 4.2.1, 4.2.1.1, 4.2.1.2, 4.2.5 and 4.2.10 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:

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

4.2.1 CRC attachment~~Error detection~~

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

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~~size of a transport block of TrCH *i*, *m* is the transport block number, and ~~L_i~~L_i is the number of parity bits. L_i can take the values 24, 16, 12, 8, or 0 depending on what is signalled from higher layers.

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

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

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

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

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

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

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

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

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

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

4.2.1.2 Relation between input and output of the CRC attachment block~~Cyclic Redundancy Check~~

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

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

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

4.2.3.3 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 $E_i = C_i Y_i$. The output bits are defined by the following relations:

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

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

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

$\ldots$

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

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

4.2.4 Radio frame size equalisation

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

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

$t_{ik} = c_{ik,}$ for k = 1… E_i and

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

where

$T_i = F_i * N_i$ and

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

4.2.5 1st interleaving

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

1) select the number of columns C1 from table 4~~;~~ depending on the TTI. The columns are numbered 0, 1, …, C1 - 1 from left to right.

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

   ~~–~~R1 = ~~X_I~~ X_i / C1~~,;~~

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

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

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

4) Perform the inter-column permutation for the matrix based on the pattern $\langle P1_{C1}(j)\rangle_{j\in\{0,1,\ldots,C1-1\}}$ shown in table 4, where $P1_{C1}(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by $y_{i,k}$:

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

5) Read the output bit sequence ~~$y_{i1}, y_{i2}, y_{i3}, \ldots, y_{i,(C1\times R1)}$~~ $y_{i,1}, y_{i,2}, y_{i,3}, \ldots, y_{i,(C1\times R1)}$ of the ~~1st~~ block interleaver~~ing~~ column by column from the inter-column permuted R1 × C1 matrix. Bit $y_{i,1}$ corresponds to ~~the first~~ row 0 of ~~the first~~ column 0 and bit $y_{i,(R1\times C1)}$ corresponds to row R1 - 1 of column C1 - 1.

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

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

Table 4 Inter-column permutation patterns for 1st interleaving

TTI	Number of columns C1	Inter-column permutation patterns <$P1_{C1}(0)$, $P1_{C1}(1)$, …, $P1_{C1}(C1-1)$>
10 ms	1	<0>
20 ms	2	<0,1>
40 ms	4	<0,2,1,3>
80 ms	8	<0,4,2,6,1,5,3,7>

4.2.5.1 Relation between input and output of 1st interleaving

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

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

end do

end if

A repeated bit is placed directly after the original one.

4.2.8 TrCH multiplexing

Every 10 ms, one radio frame from each TrCH is delivered to the TrCH multiplexing. These radio frames are serially multiplexed into a coded composite transport channel (CCTrCH).

The bits input to the TrCH multiplexing are denoted by $f_{i,1}, f_{i,2}, f_{i,3}, \ldots, f_{i,V_i}$, where i is the TrCH number and V_i is the number of bits in the radio frame of TrCH i. The number of TrCHs is denoted by I. The bits output from TrCH multiplexing are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits, i.e. $S = \sum_i V_i$. The TrCH multiplexing is defined by the following relations:

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

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

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

…

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

4.2.9 Physical channel segmentation

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

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

Bits on first PhCH after physical channel segmentation:

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

Bits on second PhCH after physical channel segmentation:

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

…

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

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

4.2.10 2nd interleaving

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

4.2.10.1 Frame related 2nd interleaving

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

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

$x_k = u_{1,k}$ k = 1, 2 , …, U_1

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

…

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

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

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

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

~~U~~U ≤ R2 X C2.

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

(3) Write ~~T~~the ~~bits~~ input bit sequence $x_1, x_2, x_3, \ldots, x_U$ ~~to the 2nd interleaving are written~~ into the R2 × C2 ~~rectangular~~ matrix row by row. starting with bit y_1 in column 0 of row 0:

$$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{30} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{60} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{(R2-1)\times 30+1} & x_{(R2-1)\times 30+2} & x_{(R2-1)\times 30+3} & \cdots & x_{R2\times 30} \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_{C2} \\ y_{(C2+1)} & y_{(C2+2)} & y_{(C2+3)} & \cdots & y_{(2\times C2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{((R2-1)\times C2+1)} & y_{((R2-1)\times C2+2)} & y_{((R2-1)\times C2+3)} & \cdots & y_{(R2\times C2)} \end{bmatrix}$$

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

(4) Perform the inter-column permutation for the matrix based on the pattern ~~{P2 (j)} (j = 0, 1, …, C2-1)~~ $\langle P2(j) \rangle_{j\in\{0,1,\ldots,C2-1\}}$ that is shown in table 7, where P_2(~~j~~j) is the original column position of the ~~j~~j-th permuted column. After permutation of the columns, the bits are denoted by ~~y~~y_k y'_k .

$$\begin{bmatrix} y_1 & y_{R2+1} & y_{2\times R2+1} & \cdots & y_{29\times R2+1} \\ y_2 & y_{R2+2} & y_{2\times R2+2} & \cdots & y_{29\times R2+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{R2} & y_{2\times R2} & y_{3\times R2} & \cdots & y_{30\times R2} \end{bmatrix} \begin{bmatrix} y'_1 & y'_{(R2+1)} & y'_{(2\times R2+1)} & \cdots & y'_{((C2-1)\times R2+1)} \\ y'_2 & y'_{(R2+2)} & y'_{(2\times R2+2)} & \cdots & y'_{((C2-1)\times R2+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_{R2} & y'_{(2\times R2)} & y'_{(3\times R2)} & \cdots & y'_{(C2\times R2)} \end{bmatrix}$$

(5) The output of the 2nd block interleaver~~ing~~ is the bit sequence read out column by column from the inter-column permuted R2 × C2 matrix. The output is pruned by deleting dummy bits that were ~~not present~~padded ~~in~~to the

input ~~bit sequence~~of the matrix before the inter-column permutation, i.e. bits ~~y_k~~ y'_k that corresponds to bits ~~x_k~~ y_k with ~~k~~ k > ~~U~~ U are removed from the output. The bits after frame related 2^{nd} interleaving are denoted by $v_1, v_2, \ldots, v_U$, where ~~v_1~~ v_1 corresponds to the bit ~~y_k~~ y'_k with smallest index ~~k~~ k after pruning, ~~v_2~~ v_2 to the bit ~~y_k~~ y'_k with second smallest index ~~k~~ k after pruning, and so on.

4.2.10.2 Timeslot related 2nd interleaving

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

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

$x_{t,k} = u_{t,1,k}$ k = 1, 2 , …, U_{t1}

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

…

$x_{t,(k+U_{t1}+\ldots+U_{t(P_t-1)})} = u_{t,P_t,k}$ k = 1, 2 , …, 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~~Assign C2 = 30 to be the number of columns of the matrix. The columns of the matrix are numbered 0, 1, 2, …, C2 - 1 from left to right.

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

~~U_t~~ U_t ≤ R2 × C2.

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

(3) Writhe ~~T~~the ~~bits~~ input bit sequence $x_{t,1}, x_{t,2}, x_{t,3}, \ldots, x_{t,U_t}$ ~~to the 2nd interleaving are written~~ into the R2 × C2 ~~rectangular~~ matrix row by row~~.~~ starting with bit $y_{t,1}$ in column 0 of row 0:

$$\begin{bmatrix} x_{t,1} & x_{t,2} & x_{t,3} & \cdots & x_{t,30} \\ x_{t,31} & x_{t,32} & x_{t,33} & \cdots & x_{t,60} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{t,((R2-1)\times 30+1)} & x_{t,((R2-1)\times 30+2)} & x_{t,((R2-1)\times 30+3)} & \cdots & x_{t,(R2\times 30)} \end{bmatrix}$$

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

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

(4) Perform the inter-column permutation for the matrix based on the pattern ~~P2(*j*) (*j* = 0, 1, …, C2-1)~~ $\langle P2(j)\rangle_{j\in\{0,1,\ldots,C2-1\}}$ that is shown in table 7, where P2(~~j~~*j*) is the original column position of the *j*-th permuted column. After permutation of the columns, the bits are denoted by ~~$y_{t,k}$~~ $y'_{t,k}$.

$$\begin{bmatrix} y_{t,1} & y_{t,(R2+1)} & y_{t,(2\times R2+1)} & \cdots y_{t,(29\times R2+1)} \\ y_{t,2} & y_{t,(R2+2)} & y_{t,(2\times R2+2)} & \cdots y_{t,(29\times R2+2)} \\ \vdots & \vdots & \vdots & \cdots \quad \vdots \\ y_{t,R2} & y_{t,(2\times R2)} & y_{t,(3\times R2)} & \cdots \; y_{t,(30\times R2)} \end{bmatrix} \begin{bmatrix} y'_{t,1} & y'_{t,(R2+1)} & y'_{t,(2\times R2+1)} & \cdots y'_{t,((C2-1)\times R2+1)} \\ y'_{t,2} & y'_{t,(R2+2)} & y'_{t,(2\times R2+2)} & \cdots y'_{t,((C2-1)\times R2+2)} \\ \vdots & \vdots & \vdots & \cdots \quad \vdots \\ y'_{t,R2} & y'_{t,(2\times R2)} & y'_{t,(3\times R2)} & \cdots \; y'_{t,(C2\times R2)} \end{bmatrix}$$

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

Table 7 Inter-column permutation pattern for 2nd interleaving

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