TSG-RAN WORKING GROUP 1 MEETING NO. 17 *TSGR1-00-1340*

STOCKHOLM, SWEDEN, 21-24 NOV. 2000

Agenda Item: AH99

Source: Siemens AG

Title: TRANSMISSION OF LONG SEQUENCES WITH SAME DATA SYMBOL

Document for: Approval

1 ABSTRACT

The spreading/scrambling code sequences have a DC content, when long sequences of the same data symbol are sent the waveform will have a significant DC content. The performance degradation that results when the receiver is AC coupled is examined. A method of scrambling is suggested to avoid the problem.

2 INTRODUCTION

A high DC content in the transmitted waveform can occur under some circumstances, which when the receiver is AC coupled will lead to a performance degradation. This working paper attempts to quantify the performance degradation and describes a method of data symbol scrambling that would solve the problem.

The product of the scrambling/spreading codes can lead to a high DC content. For a spreading factor of 16 individual codes have a mean value with a magnitude of one quarter of the total power (independent of spreading/scrambling code combination). When several spreading codes are summed at the transmitter to produce the transmitted waveform the mean value of this sum of the spreading codes will be highly dependent upon which spreading codes form the waveform and their relative powers. In Figure 1 below the mean value of the sum of the product of scrambling spreading code is given for several of the scrambling codes defined in TS25.221, and for different numbers of spreading codes summed.

Scrambling Code Number	4 Spreading Codes Summed	8 Spreading Codes Summed	16 Spreading Codes Summed
1	-8	0	-16
2	16	16	16
3	8	0	16
4	-8	0	16

5	8	16	16
6	0	0	-16
7	-8	-16	-16
8	0	16	16
9	8	16	16
10	8	16	16

Figure 1 Mean Value of Scrambling/Spreading Code Product

With random data applied to each code the resulting DC value of the transmitted waveform will generally be small, despite the individual scrambling/spreading codes possessing a non-zero mean value. When the data contains long sequences of the same data symbol, for example a long sequence of all zeros, the DC content of the transmitted waveform could be quite high, particularly if this occurs simultaneously on all codes.

3 PERFORMANCE WITH DC CONTENT REMOVED

To examine the impact upon performance when the DC content of the signal is removed (normal in most receiver designs where baseband circuitry is AC coupled to the RF/IF circuitry). Simulations have been performed where the mean value of the received signal is removed. In order to examine the extent of the problem simulations have been conducted where both random data and all-zero data is transmitted.

In the results of Figure 1 the channel is a single static path with only a random phase offset applied. The number of codes transmitted is 8. Both Joint detection and single user detection is considered at the receiver.

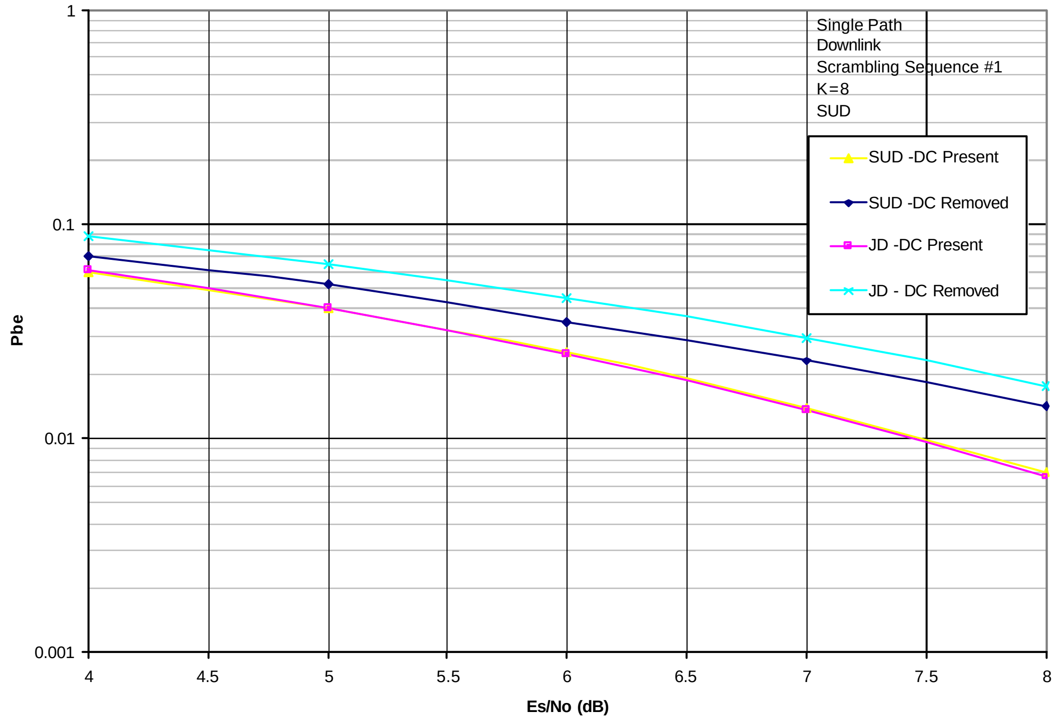

Figure 2 Raw BER for All Zero Data

The results shown above indicate that some degradation does occur when the mean value of the signal is removed. Joint detection is slightly more sensitive to removal of the mean value than single user detection. When the random data replaces all-zero data there is no difference between the performance with and without removal of the mean value of the received signal. Consequently applying data scrambling with a psuedo-noise sequence would eliminate the possible performance degradation that can occur when the receiver is AC coupled.

4 DATA SYMBOL SCRAMBLING

To overcome the problem of long sequences of the same data symbol giving rise to a large DC value in the received signal it possible to apply scrambling to the data symbols or data bits prior to transmission. The use of scrambling sequences for preventing long sequences of the same data being transmitted is well known and typically a pseudo-random sequence is generated and an exor operation performed between the data bits and the pseudo-random sequence. There are many reasons for applying data scrambling including the avoidance of DC offsets and preservation of the transmit power spectrum. The scrambling sequence should ideally possess zero mean (balanced sequence) for all lengths of data it is required to cover. This property is not critical and rather than generating balanced sequences for each length of data used a sequence long enough to cover all lengths of data would be sufficient.

There are a number of places where data scrambling could be applied. One follows transport channel multiplexing, and would be applied to the data for a whole radio frame. The length of data is variable, but upper bounded by the maximum number of bits that can be contained in a radio frame (and a different scrambling sequence could be applied to each physical channel. An alternative point where data scrambling could be applied is after the mapping to physical channels. In this case the scrambling code should be code dependent. The advantage of applying data scrambling to the CCTrCH is that only one sequence need to be specified, whereas 16 sequences would need to be specified if data scrambling were applied at the physical channel level.

Data scrambling applied to the ccTrCH would require a pseudo-noise sequence of length $2^{17} ? 1$ (assuming all 15 time slots contain 4416 bits each). For practical purposes a sequence of length $2^{16} ? 1$ would be sufficient.

If s_k are the bits of the ccTrCH, then the scrambled bits of the ccTrCH are obtained in the following manner:

$$s_k^{?} ? s_k ? p_k \quad k ? [1, S]$$

Where p_k results from the following operation:

$$p_k ? \overset{m}{\underset{i?1}{?}} g_i p_{k?i} \quad g_i ? [0,1] \quad p_k ? 0, k ? 1$$

In the above, the summation is modulo 2.

Data scrambling applied to the physical channels would require 16 pseudo-noise sequences of length $2^{13} ? 1$. Alternatively 16 different initialisations of the same generator are used. It would be generated and applied in exactly the same manner as described for the coded composite transport channel.

5 CONCLUSION

This document has shown the potential performance degradation due to the removal of the DC offset in the TDD bursts that may occur in case of the transmission of long sequences with the same data symbol. A typical example for such a transmission is the PICH, where the PI is coded as long zero and one sequences. Low cost terminals may use cost efficient AC coupled receivers, thus implicitely removing the DC component.

In order to overcome this problem, it is proposed to include a bit scrambling operation after CCTrCH multiplexing. The appropriate CRs for TS25.221 and TS25.222 can be found in this TDoc.

3GPP TSG RAN WG1 Meeting #17
Stockholm, Sweden, 21-24 Nov 2000

Document R1-001340
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.

25.221 CR 037 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
list expected approval meeting # here ↑

for approval	X	strategic	
for information		non-strategic	

(for SMG use only)

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: Siemens AG **Date:** 16-11-00

Subject: Transmission of Long Sequences with same data symbol

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: Removes possible DC offsets in the Tx signals.

Clauses affected: 5.3.7

Other specs affected:

Other 3G core specifications	X	→ List of CRs:	CR222-051
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 reserved PICH bits are changed due to the scrambling.

help.doc

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

5.3.7 The Paging Indicator Channel (PICH)

The Paging Indicator Channel (PICH) is a physical channel used to carry the paging indicators. The PICH is always transmitted at the same reference power level as the P-CCPCH.

Figure 15 depicts the structure of a PICH burst and the numbering of the bits within the burst. The same burst type is used for the PICH in every cell. N_{PIB} bits in a normal burst of type 1 or 2 are used to carry the paging indicators, where N_{PIB} depends on the burst type: N_{PIB}=240 for burst type 1 and N_{PIB}=272 for burst type 2. The bits b_{NPIB},..., b_{NPIB+3} adjacent to the midamble are reserved for possible future use. They shall be set according to the coding and scrambling scheme as defined in [7]~~to 0~~ and transmitted with the same power as the paging indicator carrying bits.

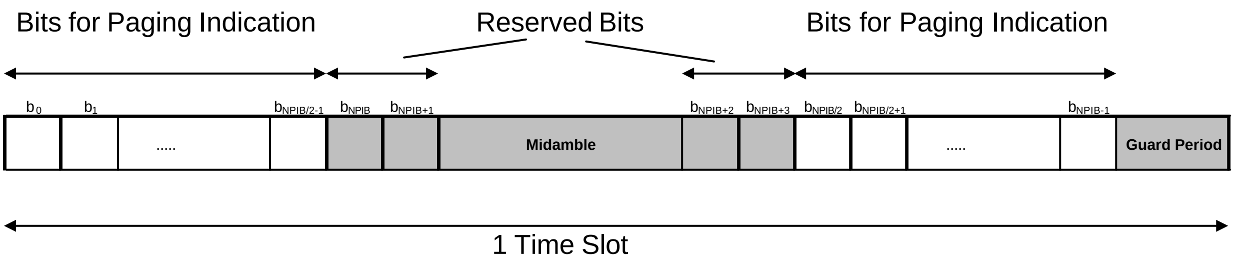

Figure 15: Transmission and numbering of paging indicator carrying bits in a PICH burst

In each time slot, N_{PI} paging indicators are transmitted, using L_{PI}=2, L_{PI}=4 or L_{PI}=8 symbols. L_{PI} is called the paging indicator length. The number of paging indicators N_{PI} per time slot is given by the paging indicator length and the burst type, which are both known by higher layer signalling. In table 8 this number is shown for the different possibilities of burst types and paging indicator lengths.

Table 8: Number N_{PI} of paging indicators per time slot for the different burst types and paging indicator lengths L_{PI}

	L_{PI}=2	L_{PI}=4	L_{PI}=8
Burst Type 1	N_{PI}=60	N_{PI}=30	N_{PI}=15
Burst Type 2	N_{PI}=68	N_{PI}=34	N_{PI}=17

As shown in figure 16, the paging indicators of N_{PICH} consecutive frames form a PICH block, N_{PICH} is configured by higher layers. Thus, $N_P=N_{PICH}*N_{PI}$ paging indicators are transmitted in each PICH block.

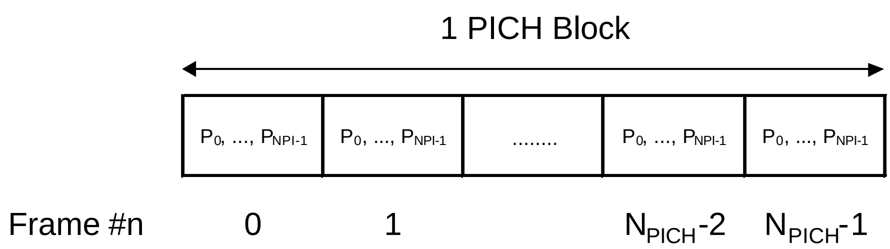

Figure 16: Structure of a PICH block

The value PI (PI = 0, ..., N_P-1) calculated by higher layers for use for a certain UE, see [15], is associated to the paging indicator P_q in the nth frame of one PICH block, where q is given by

$$q = PI \bmod N_{PI}$$

and n is given by

$$n = PI \text{ div } N_{PI}.$$

The PI bitmap in the PCH data frames over Iub contains indication values for all possible higher layer PI values, see [16]. Each bit in the bitmap indicates if the paging indicator P_q associated with that particular PI shall be set to 0 or 1. Hence, the calculation in the formulas above is to be performed in Node B to make the association between PI and P_q.

The paging indicator P_q in one time slot is mapped to the bits $\{b_{Lpi*q},...,b_{Lpi*q+Lpi-1},b_{NPIB/2+Lpi*q},...,b_{NPIB/2+Lpi*q+Lpi-1}\}$ within this time slot, as exemplary shown in figure 17. Thus, half of the L_{PI} symbols used for each paging indicator are transmitted in the first data part, and the other half of the L_{PI} symbols are transmitted in the second data part.

The coding of the paging indicator P_q is given in [7].

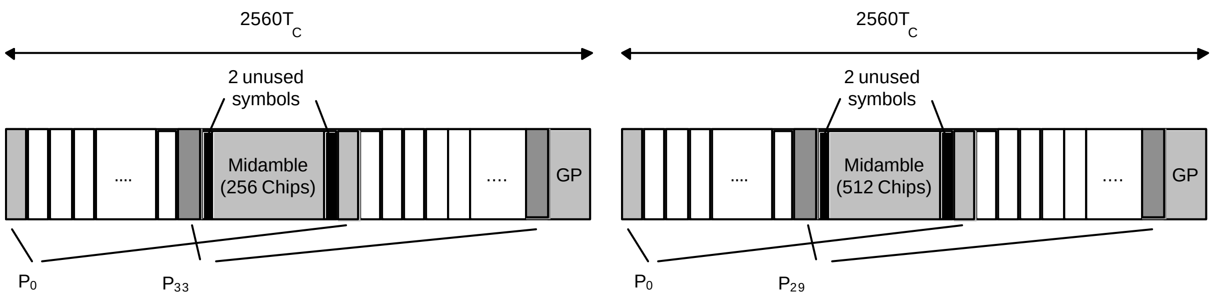

Figure 17: Example of mapping of paging indicators on PICH bits for L_{PI}=4

3GPP TSG RAN WG1 Meeting #17
Stockholm, Sweden, 21-24 Nov 2000

***Document* R1-001340**
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 051 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] — 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-Formv2.doc*

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

Source: Siemens AG **Date:** 16-11-00

Subject: Transmission of Long Sequences with same data symbol

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: Removes possible DC offsets in the Tx signals.

Clauses affected: 4.2, 4.2.8, 4.2.9, 4.2.10, 4.2.10.1, 4.2.10.2, 4.2.11, 4.2.11.1, 4.2.12, 4.2.12.1, 4.2.12.1.1, 4.2.12.1.2, 4.2.13, 4.2.13.1, 4.2.13.2, 4.2.13.2.1, 4.2.3

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: For the sections except 4.2, 4.2.8, 4.2.9, and 4.3.2 only the section number is changed.

help.doc

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

4.2 Transport channel coding/multiplexing

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

The following coding/multiplexing steps can be identified:

- add CRC to each transport block (see subclause 4.2.1);
- TrBk concatenation / Code block segmentation (see subclause 4.2.2);
- channel coding (see subclause 4.2.3) ;
- radio frame size equalization (see subclause 4.2.4);
- interleaving (two steps, see subclauses 4.2.5 and 4.2.10);
- radio frame segmentation (see subclause 4.2.6);
- rate matching (see subclause 4.2.7);
- multiplexing of transport channels (see subclause 4.2.8);
- bit scrambling (see subclause 4.2.9);
- physical channel segmentation (see subclause 4.2.109);
- mapping to physical channels (see subclause 4.2.1211).

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

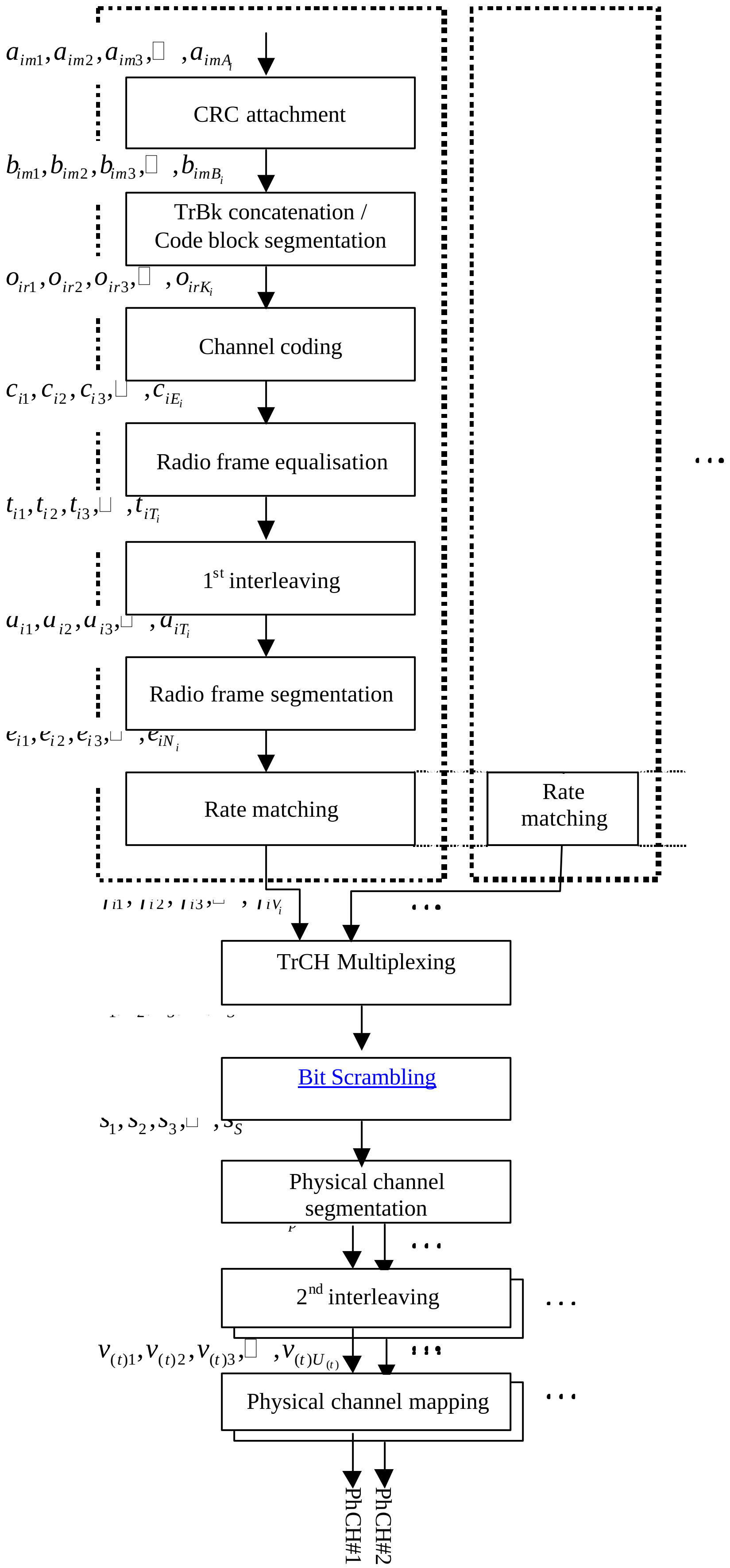

Figure 1: Transport channel multiplexing structure for uplink and downlink

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 ~~$s_1, s_2, s_3, \ldots, s_S$~~ $h_1, h_2, h_3, \ldots, h_S$, where *S* is the number of bits, i.e. $S = \sum_i V_i$. The TrCH multiplexing is defined by the following relations:

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

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

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

…

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

4.2.9 Bit Scrambling

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

Bit scrambling is defined by the following relation:

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

and p_k results from the following operation:

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

4.2.~~9~~10 Physical channel segmentation

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

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

Bits on first PhCH after physical channel segmentation:

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

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~~11 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~~11.1 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 $u_{p,k}$ in the respective physical channels is given below:

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

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

…

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

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

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

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

U ≤ R2 X C2.

(3) The bits input to the 2nd interleaving are written into the R2 × C2 rectangular matrix row by row.

$$\begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_{30} \\ x_{31} & x_{32} & x_{33} & \ldots & x_{60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{(R2-1)\cdot 30+1} & x_{(R2-1)\cdot 30+2} & x_{(R2-1)\cdot 30+3} & \ldots & x_{R2\cdot 30} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern {P2 (j)} (j = 0, 1, …, C2-1) that is shown in table 7, 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.

$$\begin{bmatrix} y_1 & y_{R2+1} & y_{2\cdot R2+1} & \ldots & y_{29\cdot R2+1} \\ y_2 & y_{R2+2} & y_{2\cdot R2+2} & \ldots & y_{29\cdot R2+2} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{R2} & y_{2\cdot R2} & y_{3\cdot R2} & \ldots & y_{30\cdot R2} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted R2 × C2 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~~11.2 Timeslot related 2nd interleaving

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

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

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

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

…

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

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

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

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

$U_t \le R2 \times C2$.

(3) The bits input to the 2nd interleaving are written into the R2 × C2 rectangular matrix row by row.

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

(4) Perform the inter-column permutation based on the pattern P2(*j*) (*j* = 0, 1, …, C2-1) that is shown in table 7, where P2(j) is the original column position of the *j*-th permuted column. After permutation of the columns, the bits are denoted by $y_{t,k}$.

$$\begin{bmatrix} y_{t,1} & y_{t,(R2+1)} & y_{t,(2\times R2+1)} & \ldots & y_{t,(29\times R2+1)} \\ y_{t,2} & y_{t,(R2+2)} & y_{t,(2\times R2+2)} & \ldots & y_{t,(29\times R2+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{t,R2} & y_{t,(2\times R2)} & y_{t,(3\times R2)} & \ldots & y_{t,(30\times R2)} \end{bmatrix}$$

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

Table 7

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

4.2.~~11~~12 Physical channel mapping

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

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

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

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

if

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

else

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

end if

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

4.2.~~11~~12.1 Mapping scheme

Notation used in this subclause:

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

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

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

bs_p: number of consecutive bits to assign per code

for downlink all $bs_p = 1$

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

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

fb_p: number of already written bits for each code

pos: intermediate calculation variable

for p=1 to P_t -- reset number of already written bits for every physical channel

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

4.2.~~12~~13 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels

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

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

   CFN mod F_{max} = 0,

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

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

   CFN_i mod F_i = 0.

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

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

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

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

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

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

Hence, there are two types of CCTrCH.

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

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

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

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

4.2.~~12~~13.1 Allowed CCTrCH combinations for one UE

4.2.~~12~~13.1.1 Allowed CCTrCH combinations on the uplink

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

1) several CCTrCH of dedicated type;

2) several CCTrCH of common type.

4.2.~~12~~13.1.2 Allowed CCTrCH combinations on the downlink

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

3) several CCTrCH of dedicated type;

4) several CCTrCH of common type.

4.2.~~13~~14 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~~14.1 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.

4.2.~~13~~14.2 Explicit transport format detection based on TFCI

4.2.~~13~~14.2.1 Transport Format Combination Indicator (TFCI)

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

4.3.2 Coding, Scrambling, and Interleaving of the Paging Indicator

The paging indicator P_q, $q = 0, ..., N_{PI}-1$, P_q ? {0, 1} is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI, calculated by higher layers, and the associated paging indicator P_q. The length L_{PI} of the paging indicator is L_{PI}=2, L_{PI}=4 or L_{PI}=8 symbols. $N_{PIB} = 2*N_{PI}*L_{PI}$ bits are used for the paging indicator transmission in one radio frame. The mapping coding of the paging indicators to the bits e_i, $i = 1, ..., N_{PIB}$ is shown in table 10.

Table 10: Coding of the paging indicator

Bits	Paging Indicator	Content
All '0'	Not set, P_q='0'	There is no necessity to receive PCH
All '1'	Set, P_q='1'	There is necessity to receive PCH

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

In order to match the number of bits used for the transmission of the paging indicators with the number of bits available in the physical channel, the sequence $e = \{e_1, e_2, ..., e_{NPIB}\}$ is extended by four bits that are set to zero, resulting in a sequence $h = \{h_1, h_2, ..., h_S\}$, where $S = N_{PIB} + 4$ is the number of bits in one radio frame of the PICH:

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

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

The bits s_k, $k = 1, ..., S$ output from the bit scrambler are then interleaved across the two data parts of the bursts by the following relations:

$$w_k ? s_{2k?1}, \quad k ? 1, ..., S/2?2$$
$$w_k ? s_{k?S/2?2}, \quad k ? S/2?1, ..., S/2?2$$
$$w_k ? s_{2k?S?4}, \quad k ? S/2?3, ..., S$$

The bits w_k are mapped to the PICH so that the bits are transmitted over the air in ascending order with respect to k.