3GPP TSG-RAN WG1 Meeting #18 **R1-01-0182**
Boston, USA, Jan.15-18, 2001

CR-Form-v3

CHANGE REQUEST

✍ **25.222** **CR** **CR-Num** ✍ rev **-** ✍ Current version: **3.4.0** ✍

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ✍ symbols.*

Proposed change affects: ✍ (U)SIM [ ] ME/UE [X] Radio Access Network [X] Core Network [ ]

Title: ✍	CR for TS25.222 regarding the 1.28 Mcps TDD
Source: ✍	CWTS/CATT
Work item code: ✍	1.28 Mcps TDD Physical Layer
Date: ✍	
Category: ✍	**B**
Release: ✍	REL-4

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change: ✍	This CR collects the principally agreed wording of the changes necessary for introducing the feature 'Low Chip Rate TDD option' in the TS25.222. In its last revision it should be editorially changed to reflect the correct changes needed for the latest approved version of this specification.
Summary of change: ✍	
Consequences if not approved: ✍	

Clauses affected: ✍	
Other specs affected: ✍	[ ] Other core specifications ✍ [ ] Test specifications [ ] O&M Specifications
Other comments: ✍	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ✍ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

2 References

<For clarity, this chapter will currently collect only the references that are needed in addition to the already existing abbreviations. In its last version this chapter has to be modified, so that it includes the revisions with respect to the latest versions of TS25.222.>

3 Definitions, symbols and abbreviations

<For clarity, this chapter will currently collect only the definitions, symbols and abbreviations that are needed in addition to the already existing ones. In its last version this chapter has to be modified, so that it includes the revisions with respect to the latest versions of TS25.222.>

3.1 Definitions

3.2 Symbols

3.3 Abbreviations

4 Multiplexing, channel coding and interleaving

4.1 General

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

In the UTRA-TDD mode, the total number of basic physical channels (a certain time slot one spreading code on a certain carrier frequency) per frame is given by the maximum number of time slots which is 15 and the maximum number of CDMA codes per time slot.

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);
- physical channel segmentation (see subclause 4.2.9);
- sub-frame segmentation(see subclause 4.2.11 only for 1.28Mcps TDD)
- mapping to physical channels (see subclause 4.2.11).

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

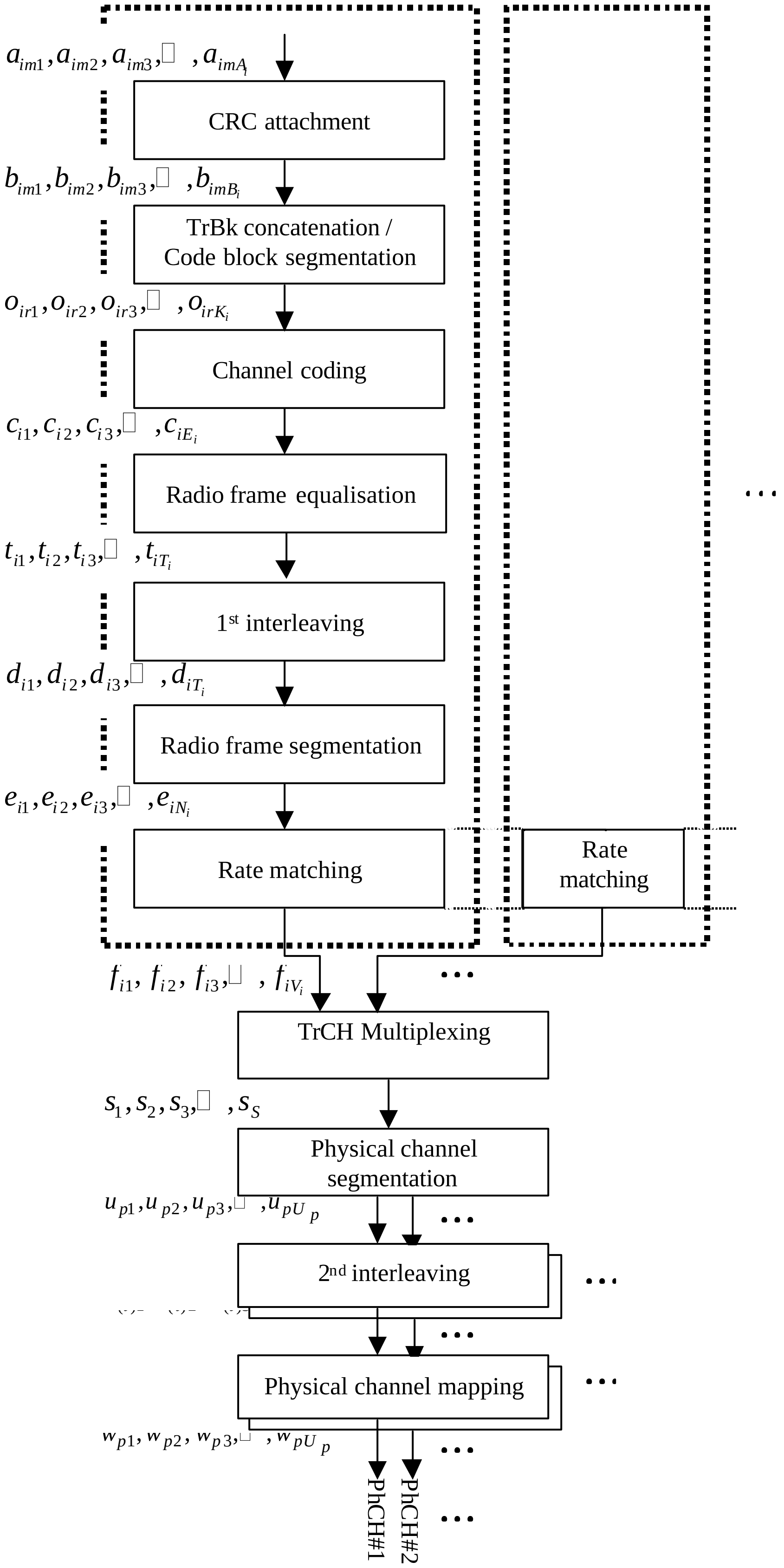

Figure 1: Transport channel multiplexing structure for uplink and downlink for 3.84Mcps TDD

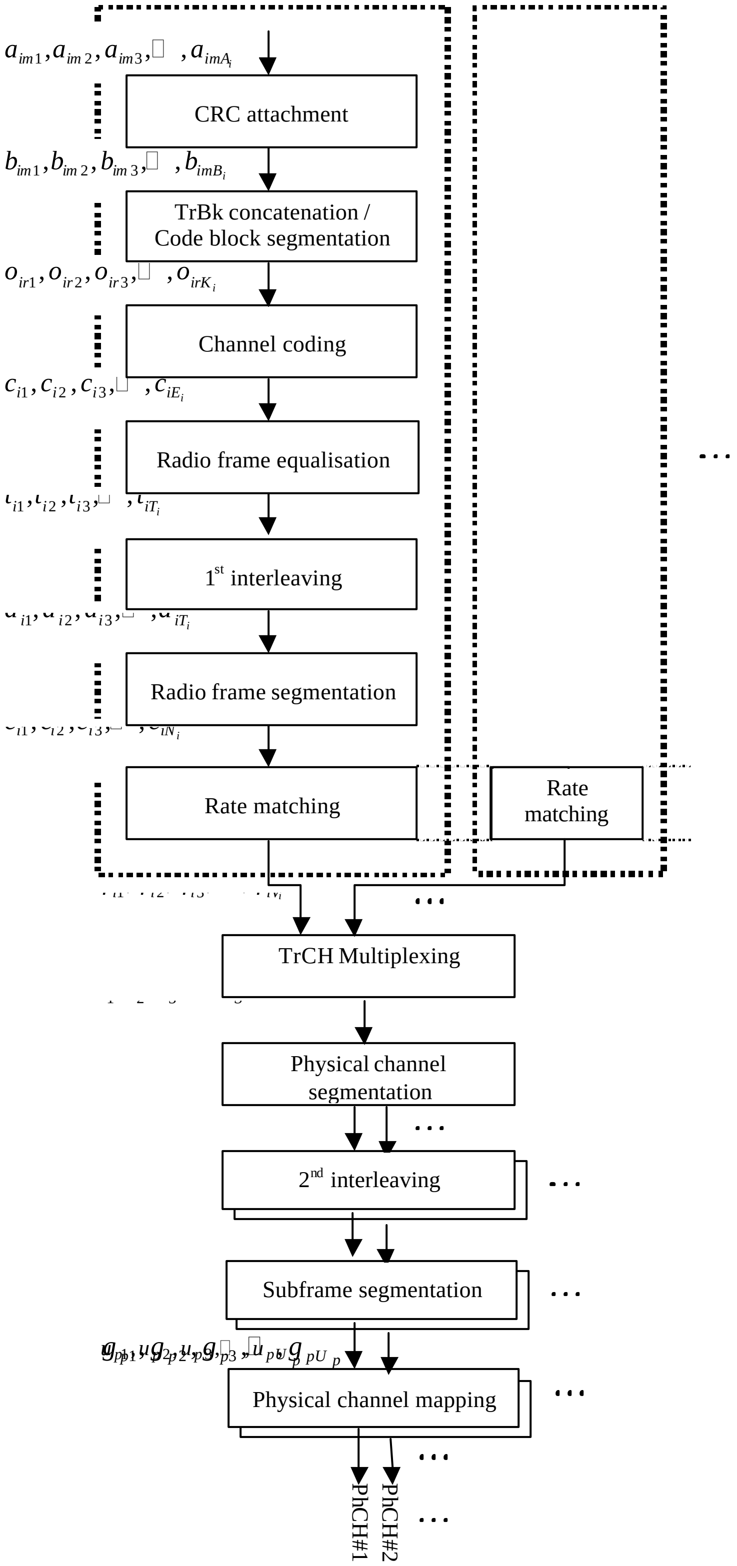

Figure X1: Transport channel multiplexing structure for uplink and downlink of 1.28Mcps TDD

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

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH *i* is denoted by C_i. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$, where Y_i is the number of encoded bits. The relation between o_{irk} and y_{irk} and between K_i and Y_i is dependent on the channel coding scheme.

The following channel coding schemes can be applied to transport channels:

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

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

- convolutional coding with rate 1/2: $Y_i = 2*K_i + 16$; rate 1/3: $Y_i = 3*K_i + 24$;
- turbo coding with rate 1/3: $Y_i = 3*K_i + 12$;
- no coding: $Y_i = K_i$.

Table 11: Usage of channel coding scheme and coding rate for 3.84Mcps TDD

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	½
PCH		
RACH		
DCH, DSCH, FACH, USCH		1/3, ½
	Turbo coding	1/3
	No coding	

Table 22: Usage of channel coding scheme and coding rate for 1.28Mcps TDD

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

4.2.11 Sub-frame segmentation for the 1.28 Mcps option

In the 1.28Mcps TDD, it is needed to add a sub-frame segmentation unit between 2nd interleaving unit and physical channel mapping unit. The operation of rate-matching guarantees that the bit streams is a even number and can be subdivided into 2 sub-frames. The transport channel multiplexing structure for uplink and downlink is shown in figure X1.

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where i is the TrCH number and Xi is the number bits.

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

$y_{i,n_i k} = x_{i,(n_i-1)Y_i+k}$, ni = 1 or 2, k = 1…Yi

where

Yi = (Xi / 2) is the number of bits per sub-frame,

x_{ik} is the kth bit of the input bit sequence and

$y_{i,n_i k}$ is the kth bit of the output bit sequence corresponding to the nth sub-frame

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

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

4.2.12 Physical channel mapping

4.2.12.1 Physical channel mapping for the 3.84 Mcps option

<No changes will be made in this chapter in this CR, only the title and numbering have to be changed. [former section 4.2.11]>

4.2.12.2 Physical channel mapping for the 1.28 Mcps option

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

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

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

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

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

if

SF1 >= SF2 then bs1 = 1 ; bs2 = SF1/SF2 ;

else

SF2 > SF1 then bs1 = SF2/SF1; bs2 = 1 ;

end if

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

4.2.12.2.1 Mapping scheme

Notation used in this subclause:

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

Utp: capacity in bits for the physical channel p in timeslot t

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

bsp: number of consecutive bits to assign per code

for downlink all bsp = 1

for uplink if SF1 >= SF2 then bs1 = 1 ; bs2 = SF1/SF2 ;

if SF2 > SF1 then bs1 = SF2/SF1; bs2 = 1 ;

fbp: 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
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 mod P t) + 1 ;
end do
if (p mod 2) == 0
pos = Utp - fbp                 -- reverse order
else
pos = fbp + 1                   -- forward order
end if
wtp,pos = gt,k                  -- assignment
fbp = fbp + 1                   -- Increment number of already written bits
If (fbp mod bsp) == 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

<No changes will be made in this chapter in this CR, only the numbering has to be changed. >

4.2.~~13~~14 Transport format detection

<No changes will be made in this chapter in this CR, only the numbering has to be changed. >

4.3 Coding for layer 1 control for the 3.84 Mcps option

<No changes will be made in this chapter in this CR, only the numbering has to be changed. >

4.4 Coding for layer 1 control for the 1.28 Mcps option

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

~~4.4.1.1 Coding of long TFCI lengths~~

~~4.4.1.2 Coding of short TFCI lengths~~

~~4.4.1.2.1 Coding very short TFCIs by repetition~~

~~4.4.1.2.2 Coding short TFCIs using bi-orthogonal codes~~

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

4.4.1.~~3~~1 Mapping of TFCI word

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

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

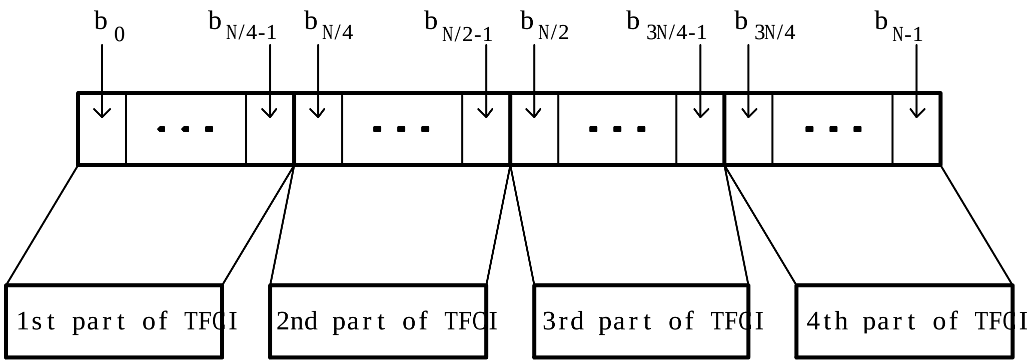

Figure [X1]: Mapping of TFCI word bits to TFCI position in 1.28 Mcps TDD option, where $N = N_{TFCI}$.

When the number of bits in the TFCI is 4 , then the TFCI word is equally divided into two parts for the consecutive two subframe and mapped onto the end of the first data field in each of the consecutive subframes.The mapping for $N_{TFCI}=4$ is show in figure [X2]:

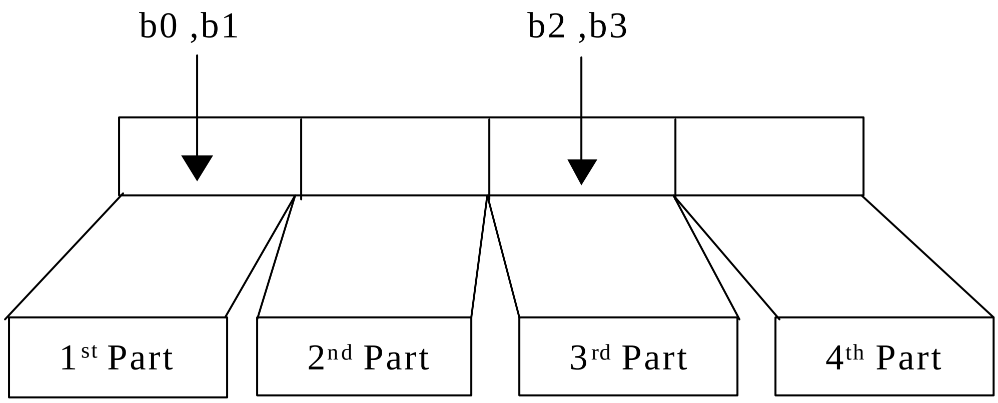

Figure [X2]: Mapping of TFCI word bits to TFCI position in 1.28 Mcps TDD option, when N_{TFCI}=4

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

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

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

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

4.4.2.1 Coding of long TFCI lengths

When the number of TFCI bits are 6 – 10, the TFCI bits are encoded by using a (64,10) sub-code of the second order Reed-Muller code, then 16 bits out of 64 bits are punctured (Puncturing positions are 0, 4, 8, 13, 16, 20, 27, 31, 34, 38, 41, 44, 50, 54, 57, 61st bits). The coding procedure is shown in Figure [F1].

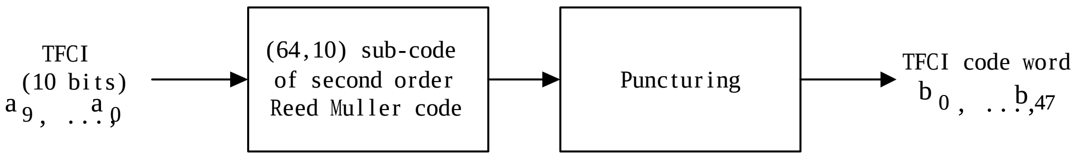

Figure [F1]: Channel coding of long TFCI bits for 8PSK

The code words of the punctured (48,10) sub-code of the second order Reed-Muller codes are linear combination of 10 basis sequences. The basis sequences are shown in Table [T1].

Table [T1]: Basis sequences for (48,10) TFCI code

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

Let's define the TFCI ~~information~~ bits as a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9, where a_0 is the LSB and a_9 is the MSB. The TFCI ~~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…47. N_{TFCI}=48.

4.4.2.2 Coding of short TFCI lengths

4.4.2.2.1 Coding very short TFCIs by repetition

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

4.4.2.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits ~~are~~is in the range of 3 to 5, the TFCI bits are encoded using a (32,5) first order Reed-Muller code, then 8 bits out of 32 bits are punctured (Puncturing positions are 0, 1, 2, 3, 4, 5, 6, 7th bits). The coding procedure is shown in Figure [F2].

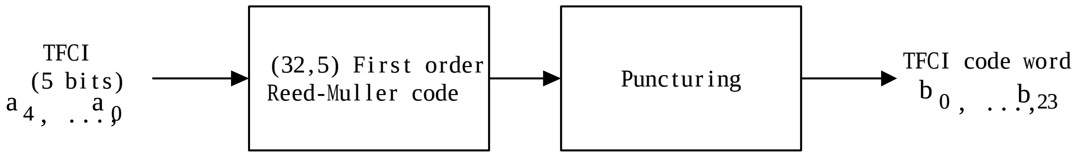

Figure [F2]: Channel coding of short TFCI bits for 8PSK

The code words of the punctured (32,5) first order Reed-Muller codes are linear combination of 5 basis sequences shown in Table [T2].

Table [T2]: Basis sequences for (24,5) TFCI code

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

Let's define the TFCI ~~information~~ bits as a_0 , a_1 , a_2 , a_3 , a_4, where a_0 is the LSB and a_4 is the MSB. The TFCI ~~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:

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

where i=0…23. N_{TFCI}=24.

4.4.2.3 Mapping of TFCI code word

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

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

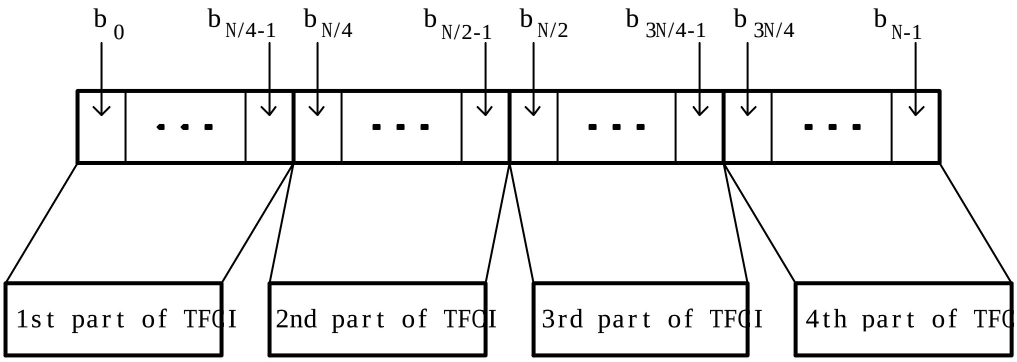

Figure [F3]: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option, where N = N_{TFCI}.

When the number of bits in the TFCI code word is 6, the TFCI code word is equally divided into two parts for the consequtive two sub-frames and mapped onto the first data field in each of the consecutive sub-frames. The mapping of the TFCI code word to the TFCI bit positions in a time slot shall be as shown in figure [F4].

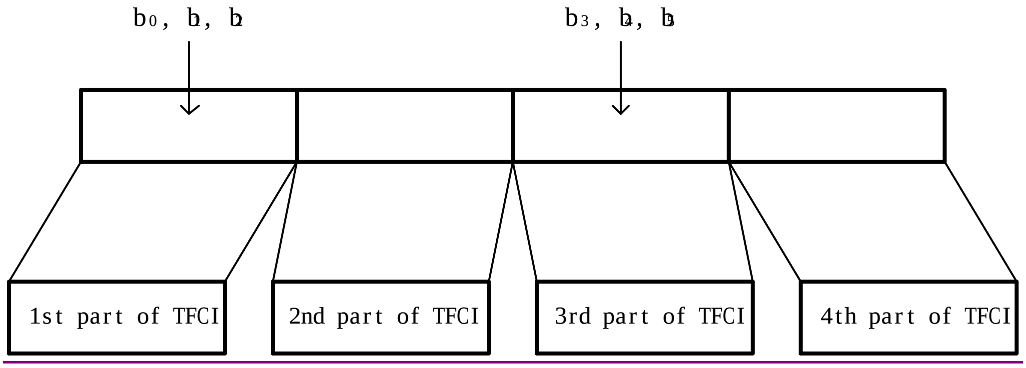

Figure [F4]: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option when N_{TFCI} = 6

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

4.4.3 Coding of Paging Indicator (PI)

The coding of Paging Indicator for 1.28Mcps TDD is same as that for 3.84Mcps TDD, cf.[4.3.2 'Coding of Paging Indicator'].

~~4.4.4 Coding of Transmit Power Control (TPC)~~

~~4.4.4.1 Coding of TPC for QPSK~~

~~4.4.4.2 Coding of TPC for 8PSK~~

~~4.4.5 Coding of Synchronisation Shift Control (SS)~~

~~4.4.5.1 Coding of SS for QPSK~~

~~4.4.5.2 Coding of SS for 8PSK~~

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

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

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

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

3. To adjust the size of the data block c(k) to the size of the FPACH burst, 8 bits are punctured as described in sub-clause 4.2.7.
The 88 bits after rate matching are then delivered to the intra-frame interleaving.

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

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

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

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