

**TSG-RAN Meeting #6
Nice, France, 13 – 15 December 1999**

TSGRP#6(99)682

Title: Agreed CRs of category "D" (Editorial) to TS 25.213

Source: TSG-RAN WG1

Agenda item: 5.1.3

Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Version-New	Doc
25.213	005	1	R99	Harmonization of notations for downlink scrambling codes	D	3.0.0	3.1.0	R1-99k05
25.213	007	1	R99	Update of TS 25.213 uplink parts	D	3.0.0	3.1.0	R1-99k12
25.213	014	1	R99	Editorial Change	D	3.0.0	3.1.0	R1-99l14

NOTE: The source of this document is TSG-RAN WG1. The source shown on each CR cover sheet is the originating organisation.

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

Document R1-99k05

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

<h2 style="margin: 0;">CHANGE REQUEST</h2>		<i>Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.</i>
25.213	CR 005r1	Current Version: v3.0.0
<small>GSM (AA.BB) or 3G (AA.BBB) specification number ↑</small>	<small>↑ CR number as allocated by MCC support team</small>	
For submission to: RAN #6 <small>list expected approval meeting # here ↑</small>	for approval <input checked="" type="checkbox"/> for information <input type="checkbox"/>	strategic <input type="checkbox"/> non-strategic <input type="checkbox"/> <small>(for SMG use only)</small>

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 UTRAN / Radio Core Network
(at least one should be marked with an X)

Source: Nokia **Date:** 1 Nov 1999

Subject: Harmonization of notations for downlink scrambling codes

Work item:

Category:	F Correction <input type="checkbox"/> A Corresponds to a correction in an earlier release <input type="checkbox"/> B Addition of feature <input type="checkbox"/> C Functional modification of feature <input type="checkbox"/> D Editorial modification <input checked="" type="checkbox"/>	Release:	Phase 2 <input type="checkbox"/> Release 96 <input type="checkbox"/> Release 97 <input type="checkbox"/> Release 98 <input type="checkbox"/> Release 99 <input checked="" type="checkbox"/> Release 00 <input type="checkbox"/>
------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(only one category shall be marked with an X)

Reason for change: The current text in 5.2.2 is misleading, notation is clarified.

Clauses affected: 5.2.2 of TS25.213

Other specs affected:	Other 3G core specifications <input type="checkbox"/> Other GSM core specifications <input type="checkbox"/> MS test specifications <input type="checkbox"/> BSS test specifications <input type="checkbox"/> O&M specifications <input type="checkbox"/>	→ List of CRs: → List of CRs: → List of CRs: → List of CRs: → List of CRs:
------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------

Other comments:

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

In case the OVSF code on the PDSCH varies from frame to frame, the OVSF codes shall be allocated such a way that the OVSF code(s) below the smallest spreading factor will be from the branch of the code tree pointed by the smallest spreading factor used for the connection. This means that all the codes for UE for the PDSCH connection can be generated according to the OVSF code generation principle from smallest spreading factor code used by the UE on PDSCH.

In case of mapping the DSCH to multiple parallel PDSCHs, the same rule applies, but all of the branches identified by the multiple codes, corresponding to the smallest spreading factor, may be used for higher spreading factor allocation.

5.2.2 Scrambling code

A total of $2^{18}-1 = 262,143$ scrambling codes, numbered $0 \dots 262,142$ can be generated. However not all the scrambling codes are used. The scrambling codes are divided into 512 sets each of a primary scrambling code and 15 secondary scrambling codes.

The primary scrambling codes consist of scrambling codes $n=16*i$ where $i=0 \dots 511$. The i :th set of secondary scrambling codes consists of scrambling codes $16*i+k$, where $k=1 \dots 15$.

There is a one-to-one mapping between each primary scrambling code and 15 secondary scrambling codes in a set such that i :th primary scrambling code corresponds to i :th set of scrambling codes.

Hence, according to the above, scrambling codes $k = 0, 1, \dots, 8191$ are used. Each of these codes are associated with an even alternative scrambling code and an odd alternative scrambling code, that may be used for compressed frames. The even alternative scrambling code corresponding to scrambling code k is scrambling code number $k + 8192$, while the odd alternative scrambling code corresponding to scrambling code k is scrambling code number $k + 16384$.

The set of primary scrambling codes is further divided into 64 scrambling code groups, each consisting of 8 primary scrambling codes. The j :th scrambling code group consists of primary scrambling codes $16*8*j+16*k$, where $j=0 \dots 63$ and $k=0 \dots 7$.

Each cell is allocated one and only one primary scrambling code. The primary CCPCH ~~and primary CPICH are~~ always transmitted using the primary scrambling code. The other downlink physical channels can be transmitted with either the primary scrambling code or a secondary scrambling code from the set associated with the primary scrambling code of the cell.

The mixture of primary scrambling code and secondary scrambling code for one CCTrCH is allowable.

The scrambling code sequences are constructed by combining two real sequences into a complex sequence. Each of the two real sequences are constructed as the position wise modulo 2 sum of 38400 chip segments of two binary m -sequences generated by means of two generator polynomials of degree 18. The resulting sequences thus constitute segments of a set of Gold sequences. The scrambling codes are repeated for every 10 ms radio frame. Let x and y be the two sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $1+X^7+X^{18}$. The y sequence is constructed using the polynomial $1+X^5+X^7+X^{10}+X^{18}$.

The sequence depending on the chosen scrambling code number n is denoted z_n , in the sequel. Furthermore, let $x(i)$, $y(i)$ and $z_n(i)$ denote the i :th symbol of the sequence x , y , and z_n , respectively

The m -sequences x and y are constructed as:

Initial conditions:

x is constructed with $x(0)=1, x(1)=x(2)=\dots=x(16)=x(17)=0$

$y(0)=y(1)=\dots=y(16)=y(17)=1$

Recursive definition of subsequent symbols:

$x(i+18) = x(i+7) + x(i) \text{ modulo } 2, i=0, \dots, 2^{18}-20,$

$y(i+18) = y(i+10)+y(i+7)+y(i+5)+y(i) \text{ modulo } 2, i=0, \dots, 2^{18}-20.$

The n :th Gold code sequence $z_n, n=0,1,2,\dots,2^{18}-2$, is then defined as

$$z_n(i) = x((i+n) \text{ modulo } (2^{18} - 1)) + y(i) \text{ modulo } 2, i=0, \dots, 2^{18}-2.$$

These binary ~~sequences code words~~ are converted to real valued sequences Z_n by the following transformation: ~~'0' → '+1', '1' → '-1'~~.

$$Z_n(i) = \begin{cases} +1 & \text{if } z_n(i) = 0 \\ -1 & \text{if } z_n(i) = 1 \end{cases} \text{ for } i = 0, 1, \dots, 2^{18} - 2.$$

Finally, the n:th complex scrambling code sequence $S_{dl,n}$ is defined as ~~(the lowest index corresponding to the chip scrambled first in each radio frame)(where N is the period in chips and M is 131,072)~~:

$$S_{dl,n}(i) = Z_n(i) + j Z_n((i + 131072M) \text{ modulo } (2^{18} - 1)), i=0, 1, \dots, 38399N-1.$$

Note that the pattern from phase 0 up to the phase of 38399 is repeated.

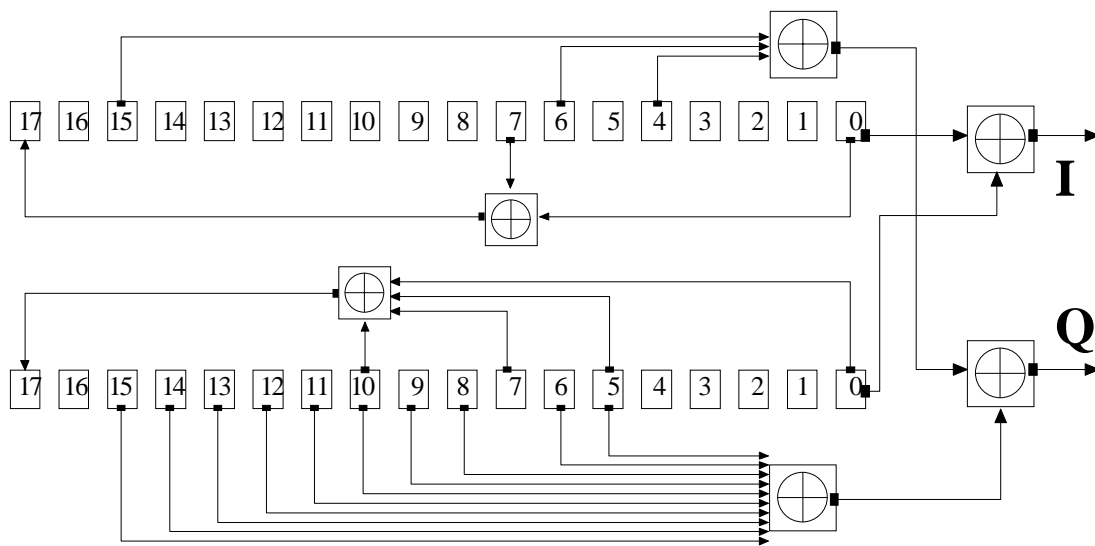


Figure 11: Configuration of downlink scrambling code generator

Agenda item:

Source: Ericsson

Title: CR 25.213-007r1: Update of uplink spreading sections

Document for: Decision

The current structure of the uplink parts of TS 25.213 V3.0.0 is a bit "un-structured":

- There is no clear distinction between "uplink scrambling codes" and those codes use for DPCCH/DPDCH and PRACH and PCPCH.
- The scrambling codes are described in several different places.
- The channelization code allocation is described in several different places.
- The sub-clause structure for PCPCH preamble code is not aligned with the PRACH preamble code structure.
- Although the spreading for preamble and message parts are very different, this is not reflected in the structure.

Moreover, there is room for improvement of the description in the following areas:

- Explicit information about how the scrambling code is applied to a radio frame is missing.
- The use of both $C_{ch,n}$ and $C_{ch,SF,n}$ to denote channelization codes is a bit confusing.
- The description of the HPSK transformation is not very clear.
- The distinction between 0/1 and 1/-1 sequences is not very clear.
- There is an error in the terminology used for the allocation of channelization code used for PRACH message data part. The names of the codes are old and should be updated.
- The list with symbol definitions is not consistent and not complete
- The short scrambling code section is a bit unclear:
 - The connection between code number and initialisation of registers is not clear.
 - How the sequence is used together with HPSK is not clearly defined.
 - The letter "c" is used for both the final codes and one of the component codes.
 - "Shift suspend" is not very clear in the figure of the generator.

To fix all this, a CR has been generated that updates the structure and makes the other clarifying changes. It should be noted that no technical change is done, i.e. the chips that come out from the spreading/scrambling is identical to what was assumed before. Also note that this CR only deals with the uplink parts.

It is acknowledged that it might have been better to address smaller issues in separate CRs, however, it was not clear how this would be accomplished together with an structural change in another CR. Hence, only one big CR has been produced.

Revision information

In revision 1 of the CR, the following has been updated:

- How the scrambling code is applied to the PRACH/PCPCH message part radio frames has been clarified, by talking now about "message part radio frames" and not "radio frames".
- The mathematical formula describing the HPSK transformation has been reformulated.

CHANGE REQUEST

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

25.213 CR 007

Current Version: **3.0.0**

GSM (AA.BB) or 3G (AA.BBB) specification number ↑

↑ CR number as allocated by MCC support team

For submission to: **TSG-RAN #6**
list expected approval meeting # here ↑

for approval
for information

strategic (for SMG use only)
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 UTRAN / Radio Core Network
(at least one should be marked with an X)

Source: Ericsson **Date:** 1999-11-18

Subject: Update of TS 25.213 uplink parts

Work item:

Category: F Correction **Release:** Phase 2
(only one category shall be marked with an X) 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
Release 00

Reason for change: Improves the readability of TS 25.213:

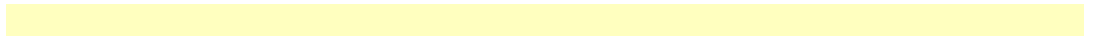
- Makes a clear distinction between "uplink scrambling codes" and those codes use for DPCC/DPDCH and PRACH and PCPCH.
- The scrambling codes to use and allocation of channelization codes are now described in a more consistent way (not spread over different sections).
- The structure of PCPCH preamble code definition is now aligned with the PRACH preamble code definition.
- Spreading for preamble and message parts are very different. This is now reflected in the structure.
- Explicit information how scrambling code is applied to radio frame has been added
- Channelization codes now have consistent names in Figure 1.
- The description of the HPSK transformation is now clear.
- The distinction between 0/1 and 1/-1 sequences is now clear.
- The terminology used for the allocation of channelization code used for PRACH message data part has been updated to the new code names.
- The list with symbol definitions is now consistent and more complete
- The connection between code number and initialisation of registers for short scrambling codes is now clear.
- The use of HPSK for short scrambling codes is now clearly defined.
- Figure of short sequence generator has been simplified.
- Codes have been renamed to not interfere with each other.

Clauses affected: 3.2, all sub-clauses to clause 4

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:



1 Scope

The present document describes spreading and modulation for UTRA Physical Layer FDD mode.

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 25.201: "Physical layer - general description".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$C_{ch,SF,n}$:	n :th channelisation code with spreading factor SF
C_{scramb}:	scrambling code for uplink
$C_{pre,n,s}$:	PRACH preamble code for n:th preamble scrambling code and signature s
$C_{c-acc,n,s}$:	PCPCH access preamble code for n:th preamble scrambling code and signature s
$C_{c-cd,n,s}$:	PCPCH CD preamble code for n:th preamble scrambling code and signature s
$C_{sig,s}$:	PRACH/PCPCH signature code for signature s
$S_{long,n}$:	n:th DPCCH/DPDCH long uplink UL scrambling code for dedicated channels
$S_{short,n}$:	n:th DPCCH/DPDCH short uplink scrambling code
$S_{r-pre,n}$:	n :th PRACH preamble scrambling code
$S_{r-msg,n}$:	n :th PRACH message scrambling code
$S_{c-acc,n}$:	n :th PCPCH access preamble scrambling code
$S_{c-cd,n}$:	n :th PCPCH CD preamble scrambling code
$S_{c-msg,n}$:	n :th PCPCH message scrambling code
$S_{dl,n}$:	DL scrambling code
$C_{sch,n}$:	n :th SCH code (primary or secondary)
C_{psc} :	PSC code
$C_{ssc,n}$:	n :th SSC code

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AICH	Acquisition Indicator Channel
AP	Access Preamble
BCH	Broadcast Control Channel

CCPCH	Common Control Physical Channel
CD	Collision Detection
CPCH	Common Packet Channel
DCH	Dedicated Channel
DPCH	Dedicated Physical Channel
DPCCH	Dedicated Physical Control Channel
DPDCH	Dedicated Physical Data Channel
FDD	Frequency Division Duplex
Mcps	Mega Chip Per Second
OVSF	Orthogonal Variable Spreading Factor (codes)
PDSCH	Physical Dedicated Shared Channel
PICH	Page Indication Channel
PRACH	Physical Random Access Channel
PSC	Primary Synchronisation Code
RACH	Random Access Channel
SCH	Synchronisation Channel
SSC	Secondary Synchronisation Code
SF	Spreading Factor
UE	User Equipment

4 Uplink spreading and modulation

4.1 Overview

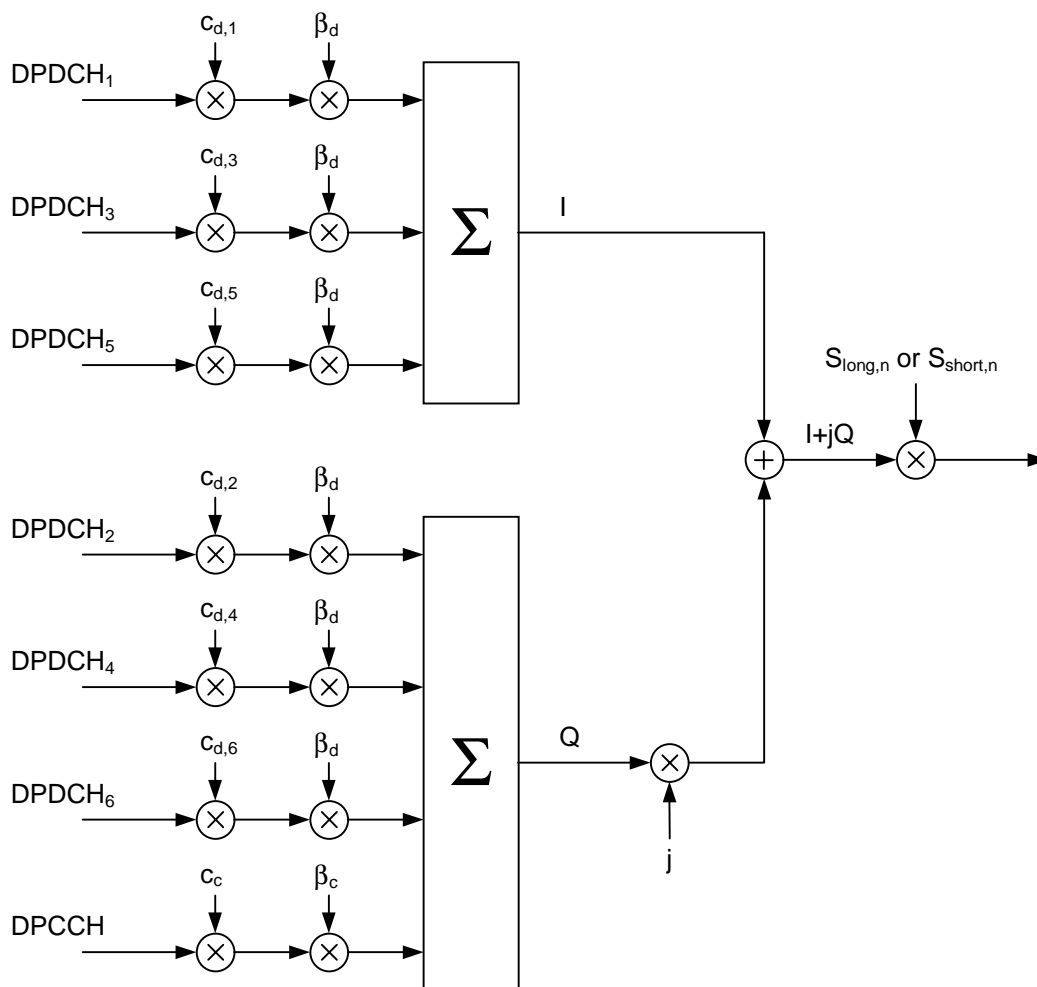
Spreading is applied to the physical channels. It consists of two operations. The first is the channelization operation, which transforms every data symbol into a number of chips, thus increasing the bandwidth of the signal. The number of chips per data symbol is called the Spreading Factor (SF). The second operation is the scrambling operation, where a scrambling code is applied to the spread signal.

With the channelization, data symbols on so-called I- and Q-branches are independently multiplied with an OVSF code. With the scrambling operation, the resultant signals on the I- and Q-branches are further multiplied by complex-valued scrambling code, where I and Q denote real and imaginary parts, respectively.

4.2 Spreading

4.2.1 Uplink Dedicated Physical Channels (uplink DPDCH/DPDCH)

Figure 1 illustrates the principle of the uplink spreading of DPCCCH and DPDCHs. The binary DPCCCH and DPDCHs to be spread are represented by real-valued sequences, i.e. the binary value "0" is mapped to the real value +1, while the binary value "1" is mapped to the real value -1. The DPCCCH is spread to the chip rate by the channelization code c_c , while the n :th DPDCH called DPDCH _{n} is spread to the chip rate by the channelization code $c_{d,n}$. One DPCCCH and up to six parallel DPDCHs can be transmitted simultaneously, i.e. $0 \leq n \leq 6$.



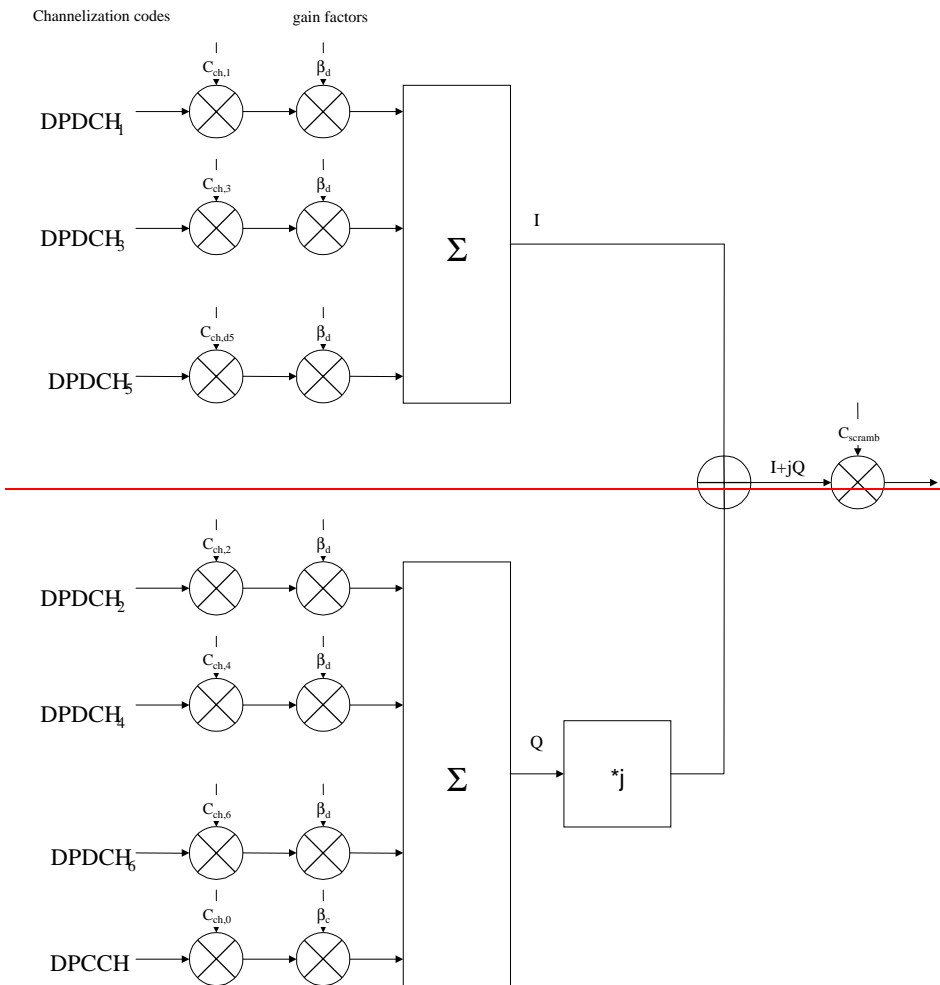


Figure 1: Spreading/modulation for uplink DPCCH and DPDCHs

After channelization, the real-valued spread signals are weighted by gain factors, β_c for DPCCH and β_d for all DPDCHs.

At every instant in time, at least one of the values β_c and β_d has the amplitude 1.0. The β -values are quantized into 4 bit words. The quantization steps are given in table 1.

Table 1: The quantization of the gain parameters

Signalling values for β_c and β_d	Quantized amplitude ratios β_c and β_d
15	1.0
14	0.9333
13	0.8666
12	0.8000
11	0.7333
10	0.6667
9	0.6000
8	0.5333
7	0.4667
6	0.4000
5	0.3333
4	0.2667
3	0.2000
2	0.1333
1	0.0667
0	Switch off

After the weighting, the stream of real-valued chips on the I- and Q-branches are then summed and treated as a complex-valued stream of chips. This complex-valued signal is then scrambled by the complex-valued scrambling code $S_{\text{long},n}$ or $S_{\text{short},n}$, depending on if long or short scrambling codes are used C_{scramb} . After pulse shaping (described in [1]), QPSK modulation is performed. The scrambling code is applied aligned with the radio frames, i.e. the first scrambling chip corresponds to the beginning of a radio frame.

4.2.2 PRACH

4.2.2.1 PRACH preamble part

The PRACH preamble part consist of a complex-valued code, that after pulse shaping is transmitted using QPSK. The preamble code is described in section 4.3.3.1.

4.2.2.2 PRACH message part

The spreading and modulation of the message part of the PRACH message part is basically the same as for the uplink dedicated physical channels.

Figure 2 illustrates the principle of the spreading and scrambling of the PRACH message part, consisting of data and control parts. The binary control and data parts to be spread are represented by real-valued sequences, i.e. the binary value "0" is mapped to the real value +1, while the binary value "1" is mapped to the real value -1. The control part is spread to the chip rate by the channelization code c_c , while the data part is spread to the chip rate by the channelization code c_d .

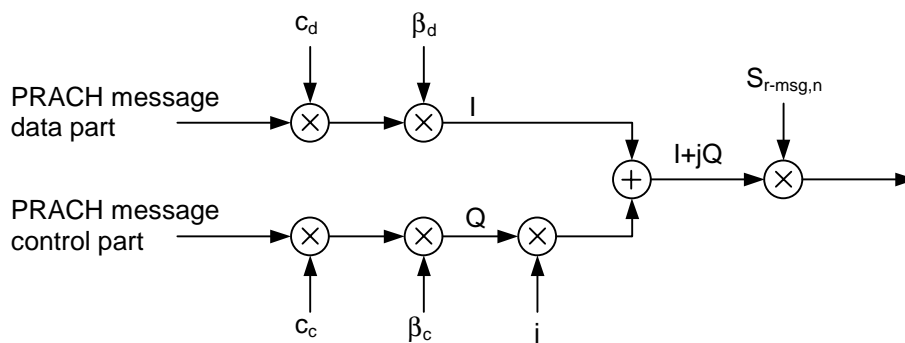


Figure 2: Spreading and scrambling of PRACH message part

After channelization, the real-valued spread signals are weighted by gain factors, β_c for the control part and β_d for the data part. At every instant in time, at least one of the values β_c and β_d has the amplitude 1.0. The β -values are quantized into 4 bit words. The quantization steps are given in section 4.2.1.

After the weighting, the stream of real-valued chips on the I- and Q-branches are treated as a complex-valued stream of chips. This complex-valued signal is then scrambled by the complex-valued scrambling code $S_{\text{r-msg},n}$. After pulse shaping (described in [1]), QPSK modulation is performed. The 10 ms scrambling code is applied aligned with the 10 ms message part radio frames, i.e. the first scrambling chip corresponds to the beginning of a message part radio frame.

4.2.3 PCPCH

4.2.3.1 PCPCH preamble part

The PCPCH preamble part consists of a complex-valued code, that after pulse shaping is transmitted using QPSK. The preamble code is described in section 4.3.4.3.

4.2.3.2 PCPCH message part

Figure 3 illustrates the principle of the spreading and scrambling of the PCPCH message part, consisting of data and control parts. The binary control and data parts to be spread are represented by real-valued sequences, i.e. the binary

value "0" is mapped to the real value +1, while the binary value "1" is mapped to the real value -1. The control part is spread to the chip rate by the channelization code c_c , while the data part is spread to the chip rate by the channelization code c_d .

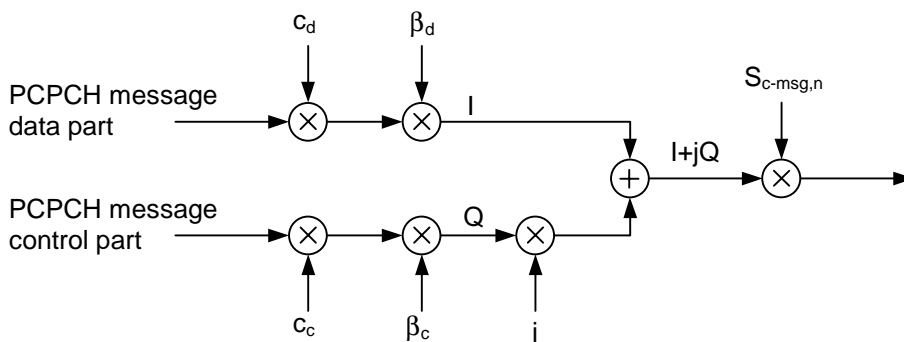


Figure 3: Spreading and scrambling of PCPCH message part

After channelization, the real-valued spread signals are weighted by gain factors, β_c for the control part and β_d for the data part. At every instant in time, at least one of the values β_c and β_d has the amplitude 1.0. The β -values are quantized into 4 bit words. The quantization steps are given in section 4.2.1.

After the weighting, the stream of real-valued chips on the I- and Q-branches are treated as a complex-valued stream of chips. This complex-valued signal is then scrambled by the complex-valued scrambling code $S_{c\text{-msg},n}$. After pulse-shaping (described in [1]), QPSK modulation is performed. The 10 ms scrambling code is applied aligned with the 10 ms message part radio frames, i.e. the first scrambling chip corresponds to the beginning of a message part radio frame.

4.3 Code generation and allocation

4.3.1 Channelization codes

4.3.1.1 Code definition

The channelization codes of figure 1 are Orthogonal Variable Spreading Factor (OVSF) codes that preserve the orthogonality between a user's different physical channels. The OVSF codes can be defined using the code tree of figure 4.

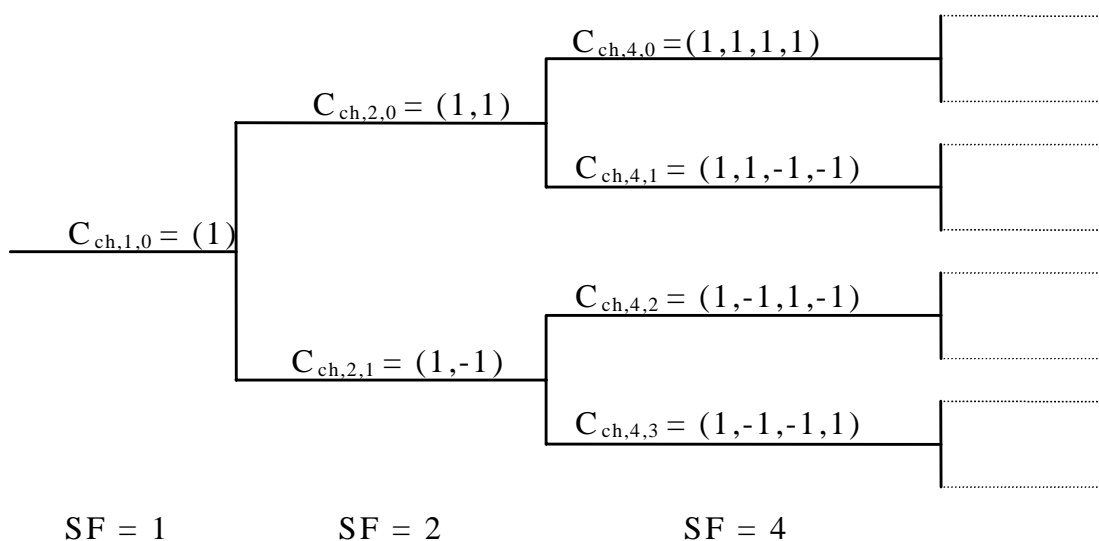


Figure 4: Code-tree for generation of Orthogonal Variable Spreading Factor (OVSF) codes

In figure 4, the channelization codes are uniquely described as $C_{ch,SF,k}$, where SF is the spreading factor of the code and k is the code number, $0 \leq k \leq SF-1$.

Each level in the code tree defines channelization codes of length SF, corresponding to a spreading factor of SF in figure 4.

The generation method for the channelization code is defined as:

$$C_{ch,1,0} = 1,$$

$$\begin{bmatrix} C_{ch,2,0} \\ C_{ch,2,1} \end{bmatrix} = \begin{bmatrix} C_{ch,1,0} & C_{ch,1,0} \\ C_{ch,1,0} & -C_{ch,1,0} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} C_{ch,2^{(n+1)},0} \\ C_{ch,2^{(n+1)},1} \\ C_{ch,2^{(n+1)},2} \\ C_{ch,2^{(n+1)},3} \\ \vdots \\ C_{ch,2^{(n+1)},2^{(n+1)-2}} \\ C_{ch,2^{(n+1)},2^{(n+1)-1}} \end{bmatrix} = \begin{bmatrix} C_{ch,2^n,0} & C_{ch,2^n,0} \\ C_{ch,2^n,0} & -C_{ch,2^n,0} \\ C_{ch,2^n,1} & C_{ch,2^n,1} \\ C_{ch,2^n,1} & -C_{ch,2^n,1} \\ \vdots & \vdots \\ C_{ch,2^n,2^{n-1}} & C_{ch,2^n,2^{n-1}} \\ C_{ch,2^n,2^{n-1}} & -C_{ch,2^n,2^{n-1}} \end{bmatrix}$$

The leftmost value in each channelization code word corresponds to the chip transmitted first in time.

4.3.1.2 Code allocation for DPCCH/DPDCH

For the DPCCH and DPDCHs the following applies:

- The DPCCH is always spread by code $c_c C_{ch,0} = C_{ch,256,0}$.
- When only one DPDCH is to be transmitted, DPDCH₁ is spread by code $c_{d,1} C_{ch,SF,k}$ where SF is the spreading factor of DPDCH₁ and $k = SF_{d,t} / 4$
- When more than one DPDCH is to be transmitted, all DPDCHs have spreading factors equal to 4. DPDCH_n is spread by the the code $c_{d,n} C_{ch,n} = C_{ch,4,k}$, where $k = 1$ if $n \in \{1, 2\}$, $k = 3$ if $n \in \{3, 4\}$, and $k = 2$ if $n \in \{5, 6\}$.

4.3.1.3 Code allocation for PRACH message part

The preamble signature s , $1 \leq s \leq 16$, points to one of the 16 nodes in the code-tree that corresponds to channelization codes of length 16. The sub-tree below the specified node is used for spreading of the message part. The control part is spread with the channelization code c_c (as shown in section 4.2.2.2) of spreading factor 256 in the lowest branch of the sub-tree, i.e. $c_c = C_{ch,256,m}$ where $m = 16(s - 1) + 15$. The data part uses any of the channelization codes from spreading factor 32 to 256 in the upper-most branch of the sub-tree. To be exact, the data part is spread by channelization code $c_d = C_{ch,SF,m}$ and SF is the spreading factor used for the data part and $m = SF \times (s - 1) / 16$.

4.3.1.4 Code allocation for PCPCH message part

The signature in the preamble specifies one of the 16 nodes in the code-tree that corresponds to channelization codes of length 16. The sub-tree below the specified node is used for spreading of the message part. The control part is always spread with a channelization code of spreading factor 256. The code is chosen from the lowest branch of the sub-tree. The data part may use channelization codes from spreading factor 4 to 64. A UE is allowed to increase its spreading factor during the message transmission by choosing any channelization code from the uppermost branch of the sub-tree code. For channelization codes with spreading factors less than 16, the node is located on the same sub-tree as the channelization code of the access preamble.

4.3.2 Scrambling codes

4.3.2.1 General

All uplink physical channels are subjected to scrambling with a complex-valued scrambling code. The DPCCH/DPDCH may be scrambled by either long or short scrambling codes, defined in section 4.3.2.4. The PRACH message part is scrambled with a long scrambling code, defined in section 4.3.2.5. Also the PCPCH message part is scrambled with a long scrambling code, defined in section 4.3.2.6.

There are 2^{24} long and 2^{24} short uplink scrambling codes. Uplink scrambling codes are assigned by higher layers.

The long scrambling code is built from constituent long sequences defined in section 4.3.2.2, while the constituent short sequences used to build the short scrambling code are defined in section 4.3.2.3.

There are 2^{24} uplink scrambling codes. All uplink channels shall use either short or long scrambling codes, except for the PRACH, for which only the long scrambling code is used. Both short and long scrambling codes are represented with complex value.

The uplink scrambling generator (either short or long) shall be initialised by a 24 bit value.

Both short and long uplink scrambling codes are formed as follows:

$$S_{ul,n} = C_{seramb,n}$$

where

$$C_{seramb,n} = e_1(w_0 + j e_2' w_1)$$

where w_0 and w_1 are chip rate sequences defined as repetitions of:

$$w_0 = \{1 \dots 1\}$$

$$w_1 = \{1 \dots -1\}$$

Also, e_1 is a real chip rate code, and e_2' is a decimated version of the real chip rate code e_2 .

With a decimation factor 2, e_2' is given as:

$$e_2'(2k) = e_2(2k+1) = e_2(2k), \quad k=0,1,2,\dots$$

The constituent codes e_1 and e_2 are formed differently for the short and long scrambling codes as described in sections 4.3.2.2 and 4.3.2.3.

4.3.2.2 Long scrambling sequencecode

The long scrambling sequences $c_{long,1,n}$ and $c_{long,2,n}$ codes are formed as described in section 4.3.2, where e_1 and e_2 are constructed fromas the position wise modulo 2 sum of 38400 chip segments of two binary m -sequences generated by means of two generator polynomials of degree 25. Let x , and y be the two m -sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $X^{25} + X^3 + 1$. The y sequence is constructed using the polynomial $X^{25} + X^3 + X^2 + X + 1$. The resulting sequences thus constitute segments of a set of Gold sequences.

The code,sequence $c_{long,2,n}$, used in generating the quadrature component of the complex spreading code is a 16,777,232 chip shifted version of the sequencecode, $c_{long,1,n}$, used in generating the in phase component.

The uplink scrambling code word has a period of one radio frame.

Let $n_{23} \dots n_0$ be the 24 bit binary representation of the scrambling sequencecode number n (decimal) with n_0 being the least significant bit. The x sequence depends on the chosen scrambling sequencecode number n and is denoted x_n , in the sequel. Furthermore, let $x_n(i)$ and $y(i)$ denote the i :th symbol of the sequence x_n and y , respectively

The m -sequences x_n and y are constructed as:

Initial conditions:

$$x_n(0)=n_0, x_n(1)= n_1, \dots, x_n(22)= n_{22}, x_n(23)= n_{23}, x_n(24)=1$$

$$y(0)=y(1)= \dots =y(23)= y(24)=1$$

Recursive definition of subsequent symbols:

$$x_n(i+25) = x_n(i+3) + x_n(i) \text{ modulo } 2, i=0, \dots, 2^{25}-27,$$

$$y(i+25) = y(i+3)+y(i+2) +y(i+1) +y(i) \text{ modulo } 2, i=0, \dots, 2^{25}-27.$$

The definition of the n -th scrambling code word for the in phase and quadrature components follows as (the left most index correspond to the chip scrambled first in each radio frame):

Define the binary Gold sequence z_n by

$$z_n(i) = x_n(i) + y(i) \text{ modulo } 2, i = 0, 1, 2, \dots, 2^{25}-2,$$

$$e_{+n} = \langle x_n(0)+y(0), x_n(1)+y(1), \dots, x_n(N-1)+y(N-1) \rangle,$$

$$e_{-n} = \langle x_n(M)+y(M), x_n(M+1)+y(M+1), \dots, x_n(M+N-1) + y(M+N-1) \rangle,$$

again all sums being modulo 2 additions.

Where N is the period in chips and $M = 16,777,232$.

These binary code words are converted to real valued sequences by the transformation '0' \rightarrow '+1', '1' \rightarrow '-1'.

The real valued Gold sequence Z_n is defined by

$$Z_n(i) = \begin{cases} +1 & \text{if } z_n(i) = 0 \\ -1 & \text{if } z_n(i) = 1 \end{cases} \text{ for } i = 0, 1, \dots, 2^{25} - 2.$$

Now, the real-valued long scrambling sequences $c_{long,1,n}$ and $c_{long,2,n}$ are defined as follows:

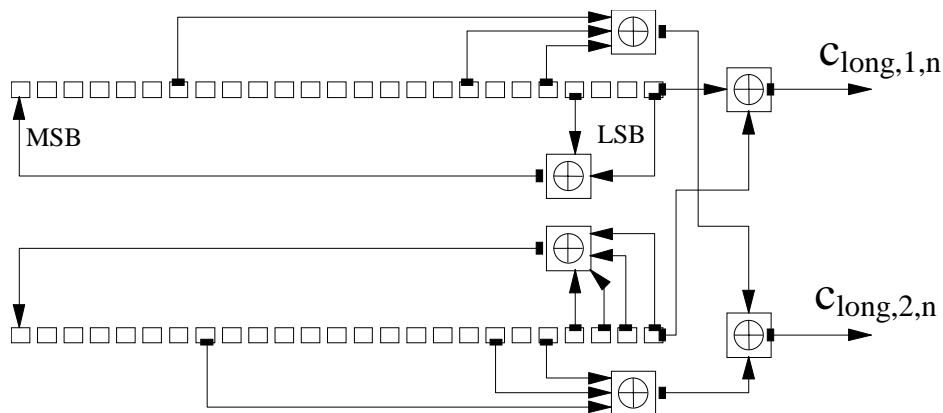
$$c_{long,1,n}(i) = Z_n(i), i = 0, 1, 2, \dots, 2^{25} - 2 \text{ and}$$

$$c_{long,2,n}(i) = Z_n((i + 16777232) \text{ modulo } (2^{25} - 1)), i = 0, 1, 2, \dots, 2^{25} - 2.$$

Finally, the complex-valued long scrambling sequence $C_{long,n}$ is defined as

$$C_{long,n}(i) = c_{long,1,n}(i) (1 + j(-1)^i c_{long,2,n}(2 \lfloor i/2 \rfloor))$$

where $i = 0, 1, \dots, 2^{25} - 2$ and $\lfloor \cdot \rfloor$ denotes rounding to nearest lower integer.



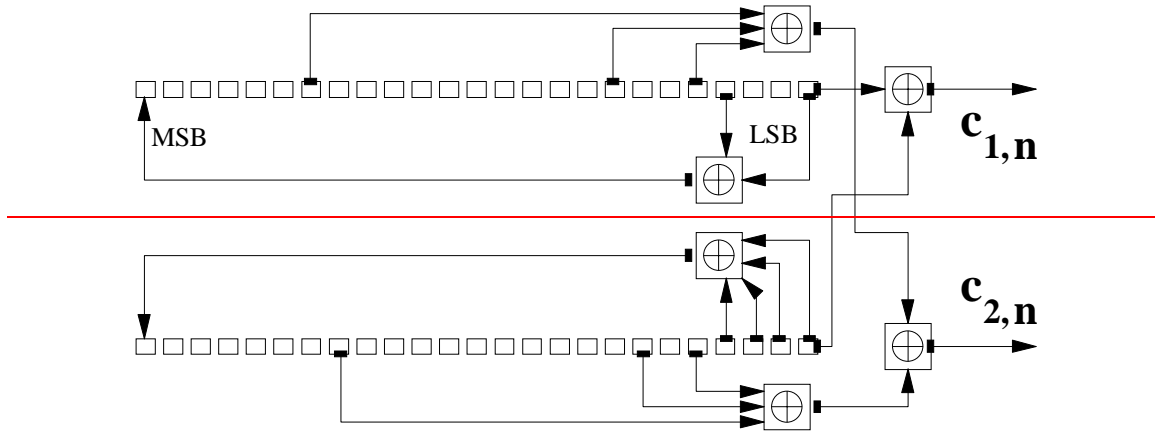


Figure 5: Configuration of uplink scrambling sequence generator

4.3.2.3 Short scrambling sequence

The short scrambling sequences $c_{short,1,n}(i)$ and $c_{short,2,n}(i)$ are defined from a sequence from the family of periodically extended S(2) codes.

Let $n_{23}n_{22}...n_0$ be the 24 bit binary representation of the code number n .

The n :th quaternary S(2) sequence $z_n(i)$, $0 \leq i \leq 16777215$, is obtained by modulo 4 addition of three sequences, a quaternary sequence $a(i)$ and two binary sequences $b(i)$ and $d(i)$, where the initial loading of the three sequences is determined from the code number n . The sequence $z_n(i)$ of length 255 is generated according to the following relation:

$$z_n(i) = a(i) + 2b(i) + 2d(i) \text{ modulo } 4, i = 0, 1, \dots, 254.$$

where the quaternary sequence $a(i)$ is generated recursively by the polynomial $g_0(x) = x^8 + x^5 + 3x^3 + x^2 + 2x + 1$ as

$$a(0) = 2n_0 + 1 \text{ modulo } 4,$$

$$a(i) = 2n_i \text{ modulo } 4, i = 1, 2, \dots, 7,$$

$$a(i) = 3a(i-3) + a(i-5) + 3a(i-6) + 2a(i-7) + 3a(i-8) \text{ modulo } 4, i = 8, 9, \dots, 254,$$

and the binary sequence $b(i)$ is generated recursively by the polynomial $g_1(x) = x^8 + x^7 + x^5 + x + 1$ as

$$b(i) = n_{8+i} \text{ modulo } 2, i = 0, 1, \dots, 7,$$

$$b(i) = b(i-1) + b(i-3) + b(i-7) + b(i-8) \text{ modulo } 2, i = 8, 9, \dots, 254,$$

and the binary sequence $c(i)$ is generated recursively by the polynomial $g_2(x) = x^8 + x^7 + x^5 + x^4 + 1$ as

$$d(i) = n_{16+i} \text{ modulo } 2, i = 0, 1, \dots, 7,$$

$$d(i) = d(i-1) + d(i-3) + d(i-4) + d(i-8) \text{ modulo } 2, i = 8, 9, \dots, 254.$$

The sequence $z_n(i)$ is extended to length 256 chips by setting $z_n(255) = z_n(0)$.

The mapping from $z_n(i)$ to the real-valued binary sequences $c_{short,1,n}(i)$ and $c_{short,2,n}(i)$, $i = 0, 1, \dots, 255$ is defined in Table 2.

$z_n(i)$	$c_{short,1,n}(i)$	$c_{short,2,n}(i)$
0	+1	+1
1	-1	+1
2	-1	-1
3	+1	-1

Table 2. Mapping from $z_n(i)$ to $c_{short,1,n}(i)$ and $c_{short,2,n}(i)$, $i = 0, 1, \dots, 255$.

Finally, the complex-valued short scrambling sequence $C_{short,n}$ is defined as

$$C_{short,n}(i) = c_{short,1,n}(i \bmod 256)(1 + j(-1)^i c_{short,2,n}(2\lfloor(i \bmod 256)/2\rfloor))$$

where $i = 0, 1, 2, \dots$ and $\lfloor \cdot \rfloor$ denotes rounding to nearest lower integer.

An implementation of the short scrambling sequence generator for the 255 chip sequence to be extended by one chip is shown in Figure 6.

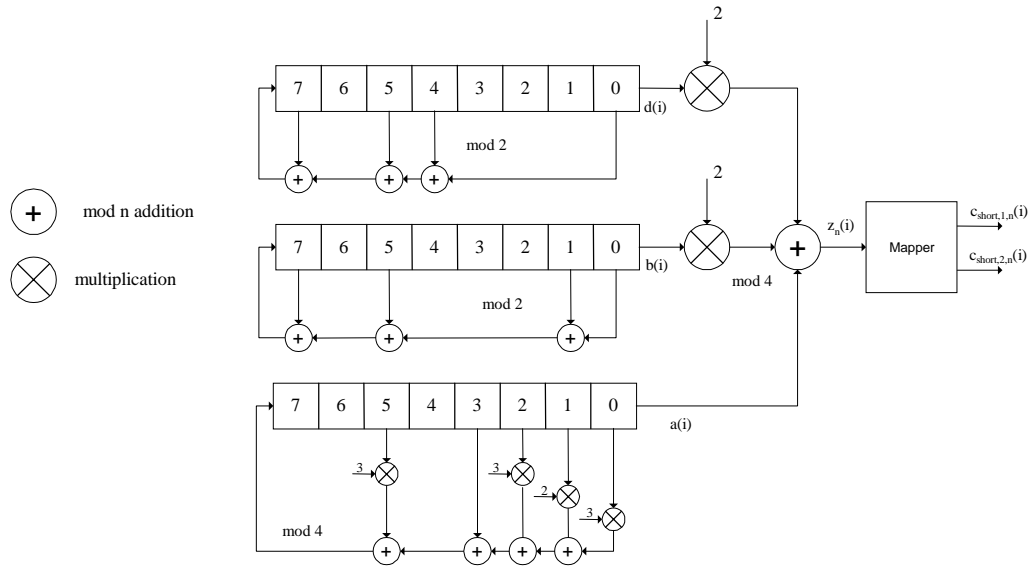


Figure 6. Uplink short scrambling sequence generator for 255 chip sequence.

The short scrambling codes are formed as described in section 4.3.2.1, where c_1 and c_2 are the real and imaginary components of a complex sequence from the family of periodically extended $S(2)$ -codes.

The uplink short codes $S_v(n)$, $n=0,1,\dots,255$, of length 256 chips are obtained by one chip periodic extension of $S(2)$ sequences of length 255. It means that the first chip ($S_v(0)$) and the last chip ($S_v(255)$) of any uplink short scrambling code are the same.

The quaternary $S(2)$ -sequence $z_v(n)$, $0 \leq v \leq 16,777,215$, of length 255 is obtained by modulo 4 addition of three sequences, a quaternary sequence $a_r(n)$ and two binary sequences $b_s(n)$ and $c_t(n)$, according to the following relation:

$$z_v(n) = a_r(n) + 2b_s(n) + 2c_t(n) \pmod{4}, \quad n = 0, 1, \dots, 254.$$

The user index v determines the indexes r , s , and t of the constituent sequences in the following way:

$$v = t \cdot 2^{16} + s \cdot 2^8 + r,$$

$$r = 0, 1, 2, \dots, 255,$$

$$s = 0, 1, 2, \dots, 255,$$

$$t = 0, 1, 2, \dots, 255.$$

The quaternary sequence $a_r(n)$ is generated by the recursive generator G_0 defined by the polynomial

$$g_0(x) = x^8 + x^5 + 3x^3 + x^2 + 2x + 1 \text{ as}$$

$$a_r(n) = 3 \cdot a_r(n-3) + 1 \cdot a_r(n-5) + 3 \cdot a_r(n-6) + 2 \cdot a_r(n-7) + 3 \cdot a_r(n-8) \pmod{4}.$$

$$n = 8 \dots 254.$$

The binary sequence $b_s(n)$ is generated by the recursive generator G_1 defined by the polynomial

$$g_1(x) = x^8 + x^7 + x^5 + x + 1 \text{ as}$$

$$b_s(n) = b_s(n-1) + b_s(n-3) + b_s(n-7) + b_s(n-8) \pmod{2}$$

The binary sequence $c_i(n)$ is generated by the recursive generator G_2 defined by the polynomial

$$g_2(x) = x^8 + x^7 + x^5 + x^4 + 1$$

$$c_i(n) = c_i(n-1) + c_i(n-3) + c_i(n-4) + c_i(n-8) \pmod{2}$$

An implementation of the short scrambling code generator is shown in figure 6. The initial states for the binary generators G_1 and G_2 are the two 8 bit words representing the indexes s and t in the 24 bit binary representation of the user index v , as it is shown in figure 7.

The initial state for the quaternary generator G_0 is according to figure 7 obtained after the transformation of 8 bit word representing the index r . This transformation is given by

$$a_r(0) = 2v(0) + 1 \pmod{4}, \quad a_r(n) = 2v(n) \pmod{4}, \quad n = 1, \dots, 7$$

The complex quadriphase sequence $S_v(n)$ is obtained from quaternary sequence $z_v(n)$ by the mapping function given in table 2.

The $\text{Re}\{S_v(n)\}$ and $\text{Im}\{S_v(n)\}$ of the $S(2)$ code are the pair of two binary sequences corresponding to input binary sequences c_1 and c_2 respectively described in 4.3.2.

Table 2: Mapping between $S_v(n)$ and $z_v(n)$

$z_v(n)$	$S_v(n)$
0	$+1 + j1$
1	$-1 + j1$
2	$-1 - j1$
3	$+1 - j1$

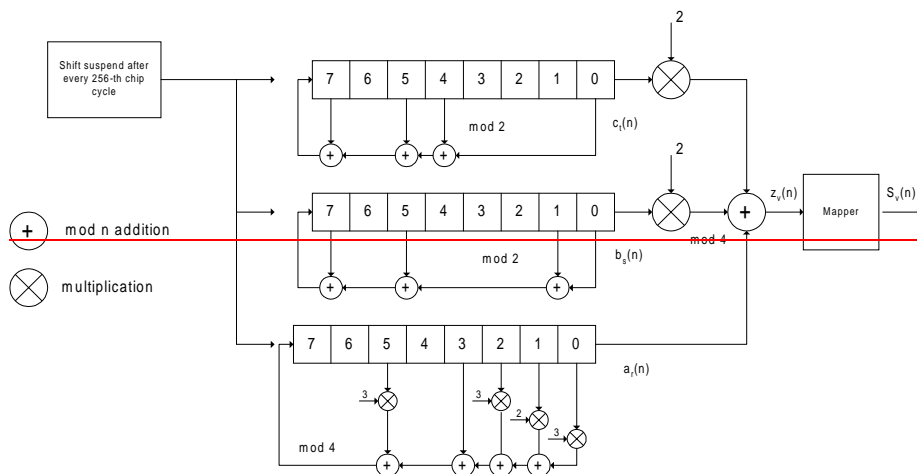


Figure 6: Uplink short scrambling code generator

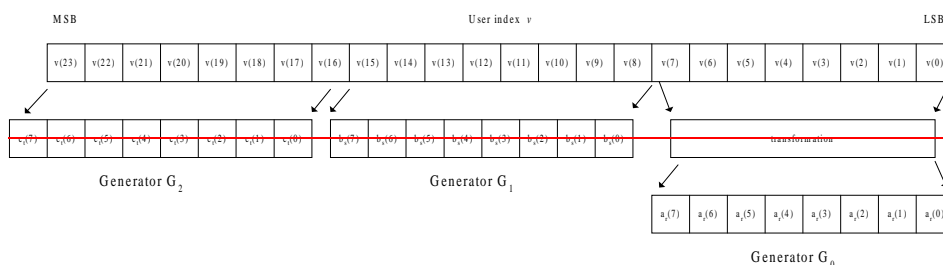


Figure 7: Uplink short scrambling code generator state initialisation

4.3.2.4 DPCCH/DPDCH scrambling code

The code used for scrambling of the uplink DPCCH/DPDCH may be of either long or short type. When the scrambling code is formed, different constituent codes are used for the long and short type as defined below.

The *n*:th long uplink scrambling code for DPCCH/DPDCH, denoted $S_{long,n}$, is defined as

$$S_{long,n}(i) = C_{long,n}(i), \quad i = 0, 1, \dots, 38399,$$

where the lowest index corresponds to the chip transmitted first in time and $C_{long,n}$ is defined in section 4.3.2.2.

The *n*:th short uplink scrambling code for DPCCH/DPDCH, denoted $S_{short,n}$, is defined as

$$S_{short,n}(i) = C_{short,n}(i), \quad i = 0, 1, \dots, 38399,$$

where the lowest index corresponds to the chip transmitted first in time and $C_{short,n}$ is defined in section 4.3.2.3.

4.3.2.5 PRACH message part scrambling code

The scrambling code used for the PRACH message part is 10 ms long, cell-specific and has a one-to-one correspondence to the scrambling code used for the preamble part.

The *n*:th PRACH message part scrambling code, denoted $S_{r-msg,n}$, is based on the long scrambling sequence and is defined as

$$S_{r-msg,n}(i) = C_{long,n}(i + 4096), \quad i = 0, 1, \dots, 38399$$

where the lowest index corresponds to the chip transmitted first in time and $C_{long,n}$ is defined in section 4.3.2.2.

4.3.2.6 PCPCH message part scrambling code

The scrambling code used for the PCPCH message part is 10 ms long, cell-specific and has a one-to-one correspondence to the scrambling code used for the preamble part.

The *n*:th PCPCH message part scrambling code, denoted $S_{c-msg,n}$, is based on the long scrambling sequence and is defined as

$$S_{r-msg,n}(i) = C_{long,n}(i + 8192), \quad i = 0, 1, \dots, 38399$$

where the lowest index corresponds to the chip transmitted first in time and $C_{long,n}$ is defined in section 4.3.2.2.

In the case when the access resources are shared between the RACH and CPCH, then $S_{c-msg,n}$ is defined as

$$S_{r-msg,n}(i) = C_{long,n}(i + 4096), \quad i = 0, 1, \dots, 38399$$

where the lowest index corresponds to the chip transmitted first in time and $C_{long,n}$ is defined in section 4.3.2.2.

NOTE: Use of short scrambling code for CPCH message part is ffs.

4.3.3 ~~Random access~~PRACH preamble codes

~~4.3.3.1~~ Preamble Codes

~~4.3.3.1.1~~ Preamble code construction

The random access preamble code $C_{pre,n,s}$, is a complex valued sequence. It is built from a preamble scrambling code $S_{pre,n}$ and a preamble signature $C_{sig,s}$ as follows:

$$C_{pre,n,s}(k) = S_{pre,n}(k) \times C_{sig,s}(k) \times e^{j\left(\frac{\pi}{4} + \frac{\pi}{2}k\right)}, k = 0, 1, 2, 3, \dots, 4095,$$

where $k=0$ corresponds to the chip transmitted first in time and $S_{pre,n}$ and $C_{sig,s}$ are defined in ~~4.3.3.1.2~~ and ~~4.3.3.32~~ below respectively.

~~4.3.3.1.2~~ Preamble scrambling code

The scrambling code for the PRACH preamble part is ~~constructed from the long scrambling sequences as follows.~~

~~The code generating method is the same as for the real part of the uplink long scrambling codes on dedicated channels, see 4.3.2.1 and 4.3.2.2. Only the first 4096 chips of the code are used for preamble scrambling.~~

~~The definition of the n :th preamble scrambling code sequence is defined as follows (the left most index correspond to the chip transmitted first in each slot):~~

$$S_{pre,n}(i) = c_{long,1,n}(i), i = 0, 1, \dots, 4095, \text{Re}\{C_{scramb,n}\}, \text{for chip indexes } 0 \dots 4095 \text{ of } C_{scramb,n}$$

~~where the sequence $c_{long,1,n}$ is defined in section 4.3.2.2.~~

~~4.3.3.32~~ Preamble signature

The preamble signature corresponding to a signature s consists of 256 repetitions of a length 16 signature $P_s(n)$, $n=0 \dots 15$. This is defined as follows:

$$C_{sig,s}(i) = P_s(i \text{ modulo } 16), i = 0, 1, \dots, 4095.$$

The signature $P_s(n)$ is from the set of 16 Hadamard codes of length 16. These are listed in table 3.

Table 3: Preamble signatures

Preamble signature	Value of n															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P_0(n)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$P_1(n)$	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
$P_2(n)$	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
$P_3(n)$	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
$P_4(n)$	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
$P_5(n)$	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
$P_6(n)$	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
$P_7(n)$	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
$P_8(n)$	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
$P_9(n)$	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
$P_{10}(n)$	1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
$P_{11}(n)$	1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
$P_{12}(n)$	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1
$P_{13}(n)$	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
$P_{14}(n)$	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
$P_{15}(n)$	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1

4.3.3.3 Channelization codes for the message part

The preamble signature s , $1 \leq s \leq 16$, points to one of the 16 nodes in the code-tree that corresponds to channelization codes of length 16. The sub-tree below the specified node is used for spreading of the message part. The control part is spread with the channelization code c_e (as shown in section 4.2.2) of spreading factor 256 in the lowest branch of the sub-tree, i.e. $c_e = C_{ch,256,m}$ where $m = 16(s-1) + 15$. The data part uses any of the channelization codes from spreading factor 32 to 256 in the upper most branch of the sub-tree. To be exact, the data part is spread by channelization code $C_{ch,d}$ where $C_{ch,d} = c_{SF,m}$ and SF is the spreading factor used for the data part and $m = SF \times (s-1) / 16$.

4.3.3.4 Scrambling code for the message part

In addition to spreading, the message part is also subject to scrambling with a 10 ms complex code. The scrambling code is cell-specific and has a one-to-one correspondence to the scrambling code used for the preamble part:

$$S_{r,msg,n} = C_{seramb,n} \text{ for chip indexes } 4096 \dots 42495 \text{ of } C_{seramb,n}$$

The generation of these codes is explained in 4.3.2.2. The mapping of these codes to provide a complex scrambling code is also the same as for the dedicated uplink channels and is described in 4.3.2.1.

4.3.4 Common packet channel PCPCH preamble codes

4.3.4.1 Access preamble

4.3.4.1.1 Access Pp preamble code construction

Similar to PRACH access preamble codes, the PCPCH access preamble codes $C_{c-acc,n,s}$ are complex valued sequences. The PCPCH access preamble codes are built from the preamble scrambling codes $S_{c-acc,n}$ and a preamble signature $C_{sig,s}$ as follows:

$$C_{c-acc,n,s}(k) = S_{c-acc,n}(k) \times C_{sig,s}(k) \times e^{j\left(\frac{\pi}{4} + \frac{\pi}{2}k\right)}, k = 0, 1, 2, 3, \dots, 4095,$$

—where $S_{c-acc,n}$ and $C_{sig,s}$ are defined in sections 4.3.4.1.2 and 4.3.4.1.3, below respectively.

4.3.4.1.2 Access preamble scrambling code

The access preamble scrambling code generation is done in a way similar to that of PRACH with a difference of the initialisation of the x_m sequence in section 4.3.2.2. The long code $C_{seramb,n}$ (as described in sections 4.3.2.1 and 4.3.2.2) for the in-phase component is used directly on both in-phase and quadrature branches without offset between branches. Only the first 4096 chips of the code are used for preamble scrambling. In the case when the access resources are shared between the RACH and CPCH, the scrambling codes used in the RACH preamble will be used for the CPCH preamble as well.

The n :th definition of the PCPCH access preamble scrambling code is defined as sequence follows (the left most index correspond to the chip transmitted first in each slot):

$$S_{c-acc,n}(i) = c_{long,1,n}(i), i = 0, 1, \dots, 4095, \text{Re}\{C_{seramb,n}\}, \text{ for chip indexes } 0 \dots 4095 \text{ of } C_{seramb,n}$$

where the sequence $c_{long,1,n}$ is defined in section 4.3.2.2.

In the case when the access resources are shared between the PRACH and PCPCH, the scrambling codes used in the PRACH preamble are used for the PCPCH preamble as well.

4.3.4.1.3 Access preamble signature

The access preamble part of the CPCH-access burst carries one of the sixteen different orthogonal complex signatures identical to the ones used by the preamble part of the random-access burst.

4.3.4.2 CD ~~P~~preamble

4.3.4.2.1 CD ~~p~~Preamble code construction

Similar to ~~P~~RACH access preamble codes, the ~~PC~~PCCH CD preamble codes $C_{c-cd,n,s}$ are complex valued sequences. The ~~PC~~PCCH CD preamble codes are built from the preamble scrambling codes $S_{c-cd,n}$ and a preamble signature $C_{sig,s}$ as follows:

$$C_{c-cd,n,s}(k) = S_{c-cd,n}(k) \times C_{sig,s}(k) \times e^{j\left(\frac{\pi}{4} + \frac{\pi}{2}k\right)}, k = 0, 1, 2, 3, \dots, 4095,$$

—where $S_{c-cd,n}$ and $C_{sig,s}$ is defined in sections 4.3.4.2.2 and 4.3.4.2.3 below respectively.

4.3.4.2.2 CD preamble scrambling code

The ~~PC~~PCCH CD preamble scrambling code is derived from the same scrambling code used in the ~~PC~~PCCH access preamble. ~~The long code $C_{scramb,n}$ (as described in sections 4.3.2.1 and 4.3.2.2) for the in-phase component is used directly on both in-phase and quadrature branches without offset between branches. The 4096 chips of the code from 4096 to 8191 are used for CPCH CD preamble scrambling.~~

The ~~n th definition of the CPCH CD access preamble scrambling code is defined as sequence follows (the left most index correspond to the chip transmitted first in each slot):~~

$$S_{c-cd,n}(i) = \underline{c_{long,l,n}(i + 4096)}, i = 0, 1, \dots, 4095, \text{Re}\{C_{scramb,n}\}, \text{for chip indexes } 4096 \dots 8191 \text{ of } C_{scramb,n}$$

~~where the sequence $c_{long,l,n}$ is defined in section 4.3.2.2.~~

In the case when the access resources are shared between the RACH and CPCH, the scrambling codes used in the RACH preamble will be used for the CPCH CD preamble as well.

4.3.4.2.3 CD preamble signature

The CD-preamble part of the CPCH-access burst carries one of sixteen different orthogonal complex signatures identical to the ones used by the preamble part of the random-access burst.

~~4.3.4.3 CPCH preamble signatures~~

~~4.3.4.3.1 Access preamble signature~~

~~The access preamble part of the CPCH access burst carries one of the sixteen different orthogonal complex signatures identical to the ones used by the preamble part of the random access burst.~~

~~4.3.4.2.2 CD preamble signature~~

~~The CD preamble part of the CPCH access burst carries one of sixteen different orthogonal complex signatures identical to the ones used by the preamble part of the random access burst.~~

~~4.3.4.3 Channelization codes for the CPCH message part~~

~~The signature in the preamble specifies one of the 16 nodes in the code tree that corresponds to channelization codes of length 16. The sub tree below the specified node is used for spreading of the message part. The control part is always spread with a channelization code of spreading factor 256. The code is chosen from the lowest branch of the sub tree. The data part may use channelization codes from spreading factor 4 to 64. A UE is allowed to increase its spreading factor during the message transmission by choosing any channelization code from the uppermost branch of the sub tree code. For channelization codes with spreading factors less than 16, the node is located on the same sub tree as the channelization code of the access preamble.~~

~~4.3.4.4 Scrambling code for the CPCH message part~~

~~In addition to spreading, the message part is also subject to scrambling with a 10 ms complex code. The scrambling code is cell specific and has a one to one correspondence to the scrambling code used for the preamble part.~~

~~$$S_{e\text{-msg},n} = C_{\text{scramb},n}, \text{ for chip indexes } 8192 \dots 46591 \text{ of } C_{\text{scramb},n}.$$~~

~~In the case when the access resources are shared between the RACH and CPCH,~~

~~$$S_{e\text{-msg},n} = C_{\text{scramb},n}, \text{ for chip indexes } 4096 \dots 42495 \text{ of } C_{\text{scramb},n}.$$~~

~~The generation of these codes is explained in 4.3.2.2. The mapping of these codes to provide a complex scrambling code is also the same as for the dedicated uplink channels and is described in 4.3.2.1.~~

~~NOTE:—Use of short scrambling code for CPCH message part is ffs.~~

4.4 Modulation

4.4.1 Modulating chip rate

The modulating chip rate is 3.84 Mcps.

4.4.2 Modulation

In the uplink, the modulation of both DPCCH and DPDCH is BPSK.

**TSG RAN Working Group 1#9
Dresden, Germany
Nov 30 to Dec 3, 1999**

TSGR1#9(99)L14

Agenda: AH14
Source GBT
Title: CR 014 for TS25.213 “Editorial Change”
Document for Approval

There seems to be an inconsistency in 25.213 and 25.211 on the issue Spreading Factor for CPCH. Use of all Spreading Factors for CPCH is an agreement in WG1 and therefore we propose to resolve the issue by adding the appropriate Spreading Factors to 25.213 text.

Revision information

This CR and contribution is the revised version of J32 to comply with the acceptable CR format.

CHANGE REQUEST

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

25.213

CR 014 r1.0

Current Version: 3.0.0

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

For submission to: RAN

list expected approval meeting # here
↑

for approval X

for information

strategic

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 UTRAN / Radio Core Network
(at least one should be marked with an X)

Source: GBT **Date:** 1 Dec 1999

Subject: Editorial Change

Work item: TS25.213

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

(only one category shall be marked with an X)

Reason for change: Inconsistency on the value of Spreading Factor for CPCH

Clauses affected: 4.3.4.3

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

Other comments:

4.3.4.3 Channelization codes for the CPCH message part

The signature in the preamble specifies one of the 16 nodes in the code-tree that corresponds to channelization codes of length 16. The sub-tree below the specified node is used for spreading of the message part. The control part is always spread with a channelization code of spreading factor 256. The code is chosen from the lowest branch of the sub-tree. The data part may use channelization codes from spreading factor 4 to ~~256~~ 64. A UE is allowed to increase its spreading factor during the message transmission by choosing any channelization code from the uppermost branch of the sub-tree code. For channelization codes with spreading factors less than 16, the node is located on the same sub-tree as the channelization code of the access preamble.