ETSI TS 101 851-3 V1.1.1 (2000-12)

Technical Specification

Satellite Component of UMTS/IMT 2000;
A-family;
Part 3: Spreading and modulation
(S-UMTS-A 25.213)

Reference
DTS/SES-00-U-1-25-03

Keywords
MES, MSS, satellite, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at http://www.etsi.org/tb/status/

If you find errors in the present document, send your comment to:
editor@etsi.fr

Copyright Notification

ETSI

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

The contents of the present document are subject to continuing work within TC-SES and may change following formal TC-SES approval. Should TC-SES modify the contents of the present document it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version 1.m.n

Where:

- the third digit (n) is incremented when editorial only changes have been incorporated in the specification;

- the second digit (m) is incremented for all other types of changes, i.e. technical enhancements, corrections, updates, etc.

The present document is part 3 of a multi-part deliverable covering the Satellite Component of UMTS/IMT 2000; A-family, as identified below:

Part 1: "Physical channels and mapping of transport channels into physical channels";

Part 2: "Multiplexing and channel coding";

Part 3: "Spreading and modulation";

Part 4: "Physical layer procedures".

Introduction

S-UMTS stands for the Satellite component of the Universal Mobile Telecommunication System. S-UMTS systems will complement the terrestrial UMTS (T-UMTS) and inter-work with other IMT-2000 family members through the UMTS core network. S-UMTS will be used to deliver 3rd generation mobile satellite services (MSS) utilizing either low (LEO) or medium (MEO) earth orbiting, or geostationary (GEO) satellite(s). S-UMTS systems are based on terrestrial 3GPP specifications and will support access to GSM/UMTS core networks.

NOTE 1: The term T-UMTS will be used in the present document to further differentiate the Terrestrial UMTS component.

Due to the differences between terrestrial and satellite channel characteristics, some modifications to the terrestrial UMTS (T-UMTS) standards are necessary. Some specifications are directly applicable, whereas others are applicable with modifications. Similarly, some T-UMTS specifications do not apply, whilst some S-UMTS specifications have no corresponding T-UMTS specification.

Since S-UMTS is derived from T-UMTS, the organization of the S-UMTS specifications closely follows the original 3rd Generation Partnership Project (3GPP) structure. The S-UMTS numbers have been designed to correspond to the 3GPP terrestrial UMTS numbering system. All S-UMTS specifications are allocated a unique S-UMTS number as follows:

S-UMTS-n xx.yyy

Where :

The numbers xx and yyy correspond to the 3GPP numbering scheme.

n (n = A, B, C, …) denotes the family of S-UMTS specifications.

An S-UMTS system is defined by the combination of a family of S-UMTS specifications and 3GPP specifications, as follows:

- If an S-UMTS specification exists it takes precedence over the corresponding 3GPP specification (if any). This precedence rule applies to any references in the corresponding 3GPP specifications.

NOTE 2: Any references to 3GPP specifications within the S-UMTS specifications are not subject to this precedence rule. For example, an S-UMTS specification may contain specific references to the corresponding 3GPP specification.

- If an S-UMTS specification does not exist, the corresponding 3GPP specification may or may not apply. The exact applicability of the complete list of 3GPP specifications shall be defined at a later stage.

1 Scope

The present document defines the Layer 1 transport channels and physical channels used for family A of the satellite component of UMTS (S-UMTS-A).

It is based on the FDD mode of UTRA defined by 3GPP [2], [3], [4], [5] and adapted for operation over satellite transponders.

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 and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies.

[1] ETSI TS 101 851-1: "Satellite Component of UMTS/IMT 2000; A-family; Part 1: Physical channels and mapping of transport channels into physical channels (S-UMTS-A 25.211)".

[2] ETSI TS 125 211: "Universal Mobile Telecommunication System (UMTS); Physical channels and mapping of transport channels onto physical channels (FDD) (3G TS 25.211 version 3.3.0 Release 1999)".

[3] ETSI TS 125 212: "Universal Mobile Telecommunication System (UMTS); Multiplexing and channel coding (FDD) (3G TS 25.212 version 3.3.0 Release 1999)".

[4] ETSI TS 125 213: "Universal Mobile Telecommunication System (UMTS); Spreading and modulation (FDD) (3G TS 25.213 version 3.3.0 Release 1999)".

[5] ETSI TS 125 214: "Universal Mobile Telecommunication System (UMTS); Physical layer procedures (FDD) (3G TS 25.214 version 3.3.0 Release 1999)".

[6] ETSI TS 125 101: "Universal Mobile Telecommunication System (UMTS); UE Radio transmission and Reception (FDD) '3G TS 25.101 version 3.3.0 Release 1999)".

[7] ETSI TS 125 104: "Universal Mobile Telecommunication System (UMTS); UTRA (BS) FDD; Radio transmission and Reception (3G TS 25.104 version 3.3.0 Release 1999)".

3 Symbols and abbreviations

3.1 Symbols

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

$C_{ch,SF,n}$:   n:th channellization code with spreading factor SF
$C_{pre,n,s}$:   PRACH preamble code for n:th preamble scrambling code and signature s
$C_{sig,s}$:   PRACH signature code for signature s
$S_{dpch,n}$: n:th DPCCH/DPDCH uplink scrambling code
$S_{r-pre,n}$: n:th PRACH preamble scrambling code
$S_{r-msg,n}$: n:th PRACH message scrambling code
$S_{dl,n}$:   DL scrambling code

C_{psc}:   PSC code

$C_{ssc,n}$:   n:th SSC code

3.2 Abbreviations

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

AICH	Acquisition Indicator Channel
CCPCH	Common Control Physical Channel
CPICH	Common Pilot 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 Synchronization Code
RACH	Random Access Channel
SCH	Synchronization Channel
SSC	Secondary Synchronization 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 symbol 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/DPCCH)

Figure 1 illustrates the principle of the uplink spreading of DPCCH and DPDCHs. The binary DPCCH 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 DPCCH is spread to the chip rate by the channelization code $C_{ch,0}$, while the n:th DPDCH called $DPDCH_n$ is spread to the chip rate by the channelization code $C_{ch,n}$. One DPCCH and up to six parallel DPDCHs can be transmitted simultaneously, i.e. $0 \leq n \leq 6$.
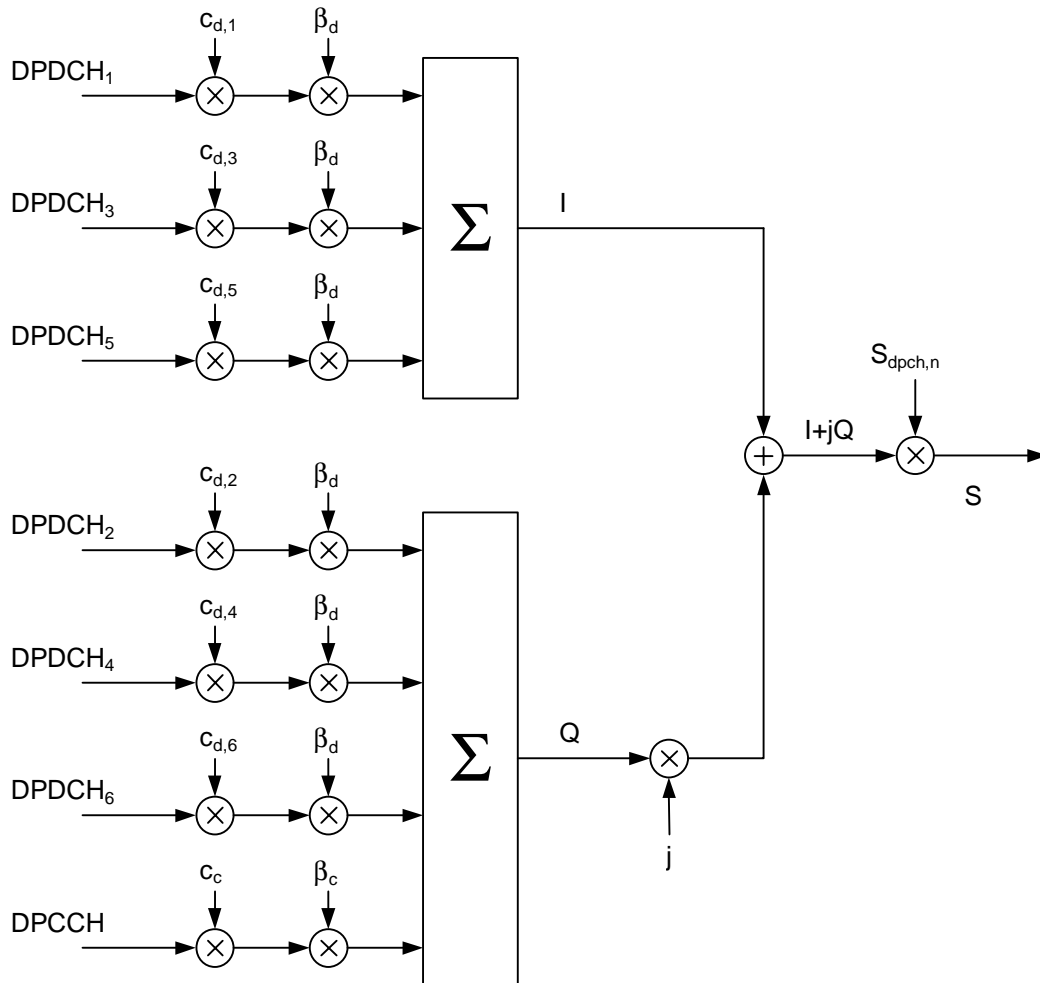
Figure 1: Spreading/modulation for uplink DPCCH and DPDCHs

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	14/15
13	13/15
12	12/15
11	11/15
10	10/15
9	9/15
8	8/15
7	7/15
6	6/15
5	5/15
4	4/15
3	3/15
2	2/15
1	1/15
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 $S_{dpch,n}$. 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 consists of a complex-valued code, described in clause 4.3.3.

4.2.2.2 PRACH message part

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.
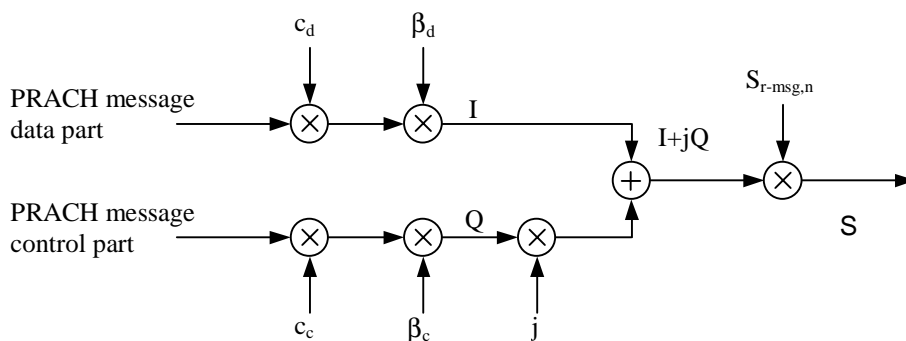
Figure 2: Spreading of PRACH message part

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 clause 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 $S_{r-msg,n}$. 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

This channel is not used in S-UMTS-A.

4.3 Code generation and allocation

4.3.1 Channellization 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 3.

$$C_{ch,4,0} = (1,1,1,1)$$

$$C_{ch,2,0} = (1,1)$$
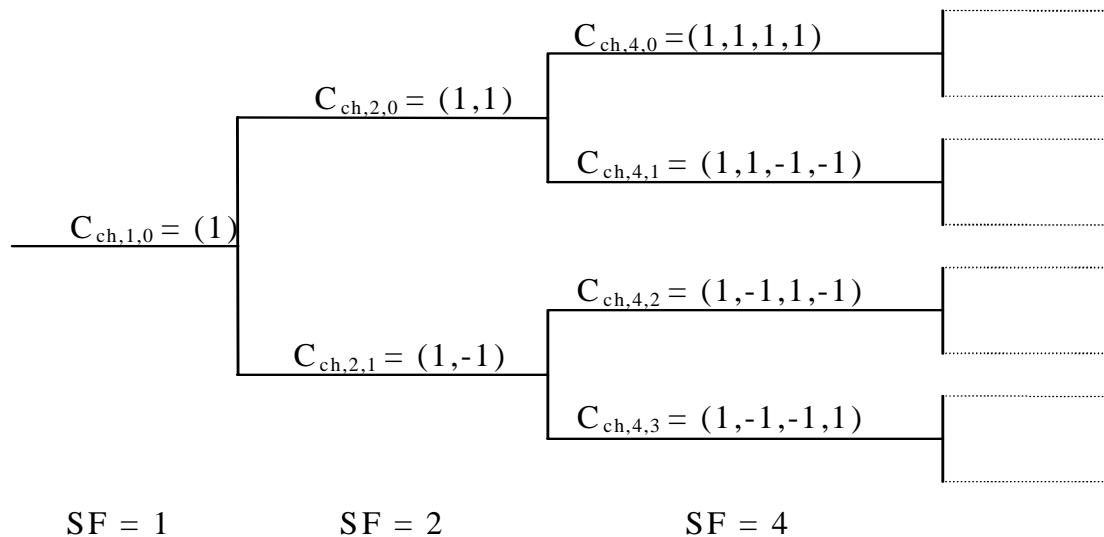
$$C_{ch,4,1} = (1,1,-1,-1)$$

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

$$C_{ch,4,2} = (1,-1,1,-1)$$

$$C_{ch,2,1} = (1,-1)$$

$$C_{ch,4,3} = (1,-1,-1,1)$$

SF = 1 SF = 2 SF = 4

Figure 3: Code-tree for generation of Orthogonal Variable Spreading Factor (OVSF) codes

In figure 3, 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 3.

The generation method for the channelization code is defined as:

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

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

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

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 $c_c = C_{ch,256,0}$.

- When only one DPDCH is to be transmitted, DPDCH$_1$ is spread by code $c_{d,1} = C_{ch,SF,k}$ where SF is the spreading factor of DPDCH$_1$ and $k = SF/4$.

- 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 $c_{d,n} = C_{ch,4,k}$, where $k = 1$ if $n \in \{1, 2\}$, $k = 3$ if $n \in \{3, 4\}$, and $k = 2$ if $n \in \{5, 6\}$.

If a power control preamble is used to initialize a DCH, the channellization code for the DPCCH during the power control preamble shall be the same as that to be used afterwards.

4.3.1.3   Code allocation for PRACH message part

The preamble signature s, $0 \le s \le 15$, 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 clause 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 $m = 16 \times s + 15$. 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 $m = SF \times s/16$.

4.3.1.4   Code allocation for PCPCH message part

This channel (CPCH) is not used in S-UMTS-A.

4.3.1.5   Channellization code for PCPCH power control preamble

This channel (CPCH) is not used in S-UMTS-A.

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 clause 4.3.2.4. The PRACH message part is scrambled with a long scrambling code, defined in clause 4.3.2.5. Also the PCPCH message part is scrambled with a long scrambling code, defined in clause 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 clause 4.3.2.2, while the constituent short sequences used to build the short scrambling code are defined in clause 4.3.2.3.

4.3.2.2   Long scrambling sequence

The long scrambling sequences $c_{long,1,n}$ and $c_{long,2,n}$ are constructed from position wise modulo 2 sum of 38 400 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 sequence $c_{long,2,n}$ is a 16 777 232 chip shifted version of the sequence $c_{long,1,n}$.

Let $n_{23} \dots n_0$ be the 24 bit binary representation of the scrambling sequence number n with n_0 being the least significant bit. The x sequence depends on the chosen scrambling sequence 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:

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

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

Define the binary Gold sequence z_n by:

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

The real valued Gold sequence Z_n is defined by:

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

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

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

$$c_{long,2,n}(i) = Z_n((i + 16\ 777\ 232) \text{ modulo } (2^{25} - 1)), \ i = 0, 1, 2, \ldots, 2^{25} - 2.$$

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

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

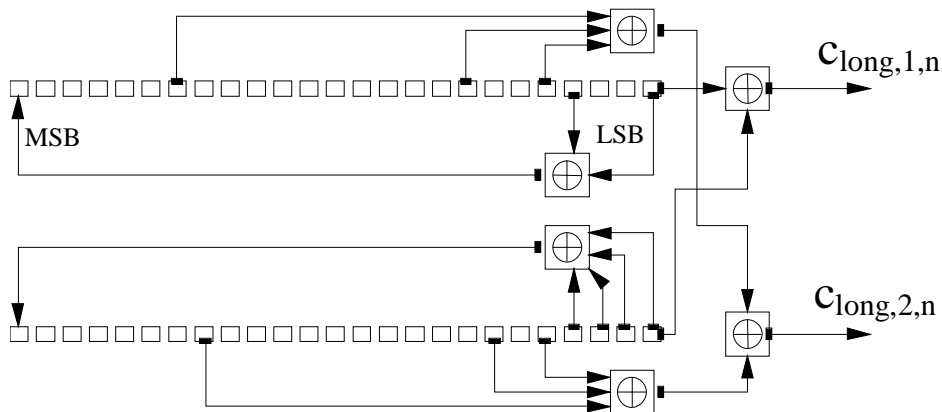Where $i = 0, 1, \ldots, 2^{25} - 2$ and $\lfloor \rfloor$ denotes rounding to nearest lower integer.

Figure 4: 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 $n_{23}n_{22}\ldots n_0$ be the 24 bit binary representation of the code number n.

The n:th quaternary S(2) sequence $z_n(i)$, $0 \le n \le 16\ 777\ 215$, 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:

- $z_n(i) = a(i) + 2b(i) + 2d(i)$ modulo 4, $i = 0, 1, \ldots, 254$;

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:

- $a(0) = 2n_0 + 1$ modulo 4;

- $a(i) = 2n_i$ modulo 4, $i = 1, 2, \ldots, 7$;

- $a(i) = 3a(i-3) + a(i-5) + 3a(i-6) + 2a(i-7) + 3a(i-8)$ modulo 4, $i = 8, 9, \ldots, 254$;

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

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

$b(i) = b(i-1) + b(i-3) + b(i-7) + b(i-8)$ modulo 2, $i = 8, 9, \ldots, 254$;

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

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

$d(i) = d(i-1) + d(i-3) + d(i-4) + d(i-8)$ modulo 2, $i = 8, 9, \ldots, 254$.

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 $c_{short,2,n}(i)$, , $i = 0, 1, \ldots, 255$ is defined in table 2.

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

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

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

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

Where $i = 0, 1, 2, \ldots$ and $\lfloor \rfloor$ 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 5.
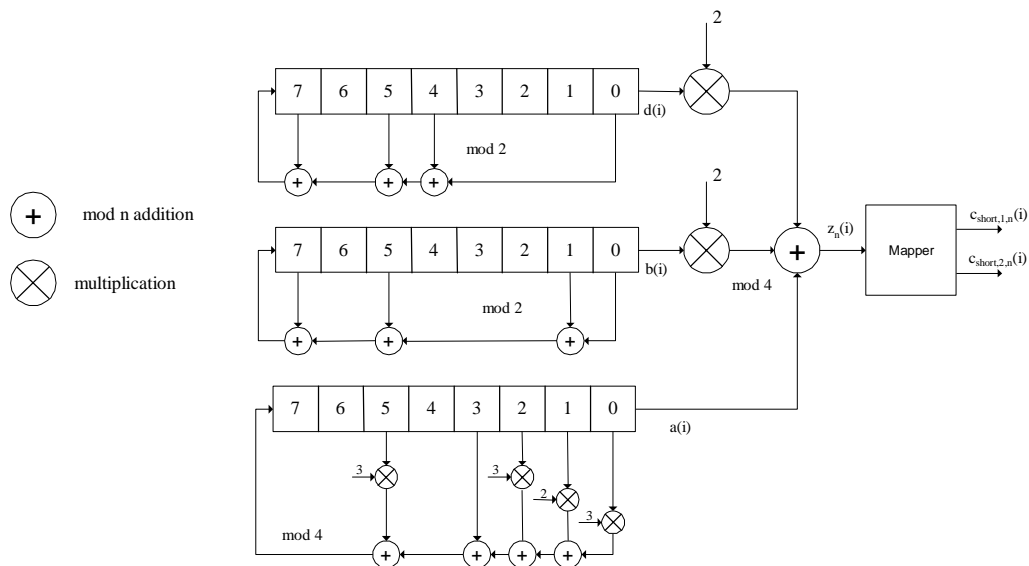

Figure 5: Uplink short scrambling sequence generator for 255 chip sequence

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 uplink scrambling code for DPCCH/DPDCH, denoted $S_{dpch, n}$, is defined as:

$$S_{dpch,n}(i) = C_{long,n}(i), i = 0, 1, \ldots, 38\ 399, \text{ when using long scrambling codes.}$$

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

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

$$S_{dpch,n}(i) = C_{short,n}(i), i = 0, 1, \ldots, 38\ 399, \text{ when using short scrambling codes.}$$

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

If a power control preamble is used to initialize a DCH, the scrambling code for the DPCCH during the power control preamble shall be the same as that to be used afterwards. The phase of the scrambling code shall be such that the end of the code is aligned with the frame boundary at the end of the power control preamble.

4.3.2.5 PRACH message part scrambling code

The scrambling code used for the PRACH message part is 10 ms long, and there are 8 192 different PRACH scrambling codes defined.

The n:th PRACH message part scrambling code, denoted $S_{r\text{-}msg,n}$, where $n = 0, 1, \ldots, 8\ 191$, is based on the long scrambling sequence and is defined as:

$$S_{r\text{-}msg,n}(i) = C_{long,n}(i + 4\ 096), i = 0, 1, \ldots, 38\ 399$$

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

The message part scrambling code has a one-to-one correspondence to the scrambling code used for the preamble part. For one PRACH, the same code number is used for both scrambling codes, i.e. if the PRACH preamble scrambling code used is $S_{r\text{-}pre,m}$ then the PRACH message part scrambling code is $S_{r\text{-}msg,m}$, where the number m is the same for both codes.

4.3.2.6 PCPCH message part scrambling code

This channel (CPCH) is not used for S-UMTS-A.

4.3.2.7 PCPCH power control preamble scrambling code

This channel (CPCH) is not used for S-UMTS-A.

4.3.3 PRACH preamble codes

4.3.3.1 Preamble code construction

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

- $C_{pre,n,s}(k) = S_{r\text{-}pre,n}(k) \times C_{sig,s}(k) \times e^{j(\frac{\pi}{4} + \frac{\pi}{2}k)}$, k = 0, 1, 2, 3, …, 4 095

Where k = 0 corresponds to the chip transmitted first in time and $S_{r\text{-}pre,n}$ and $C_{sig,s}$ are defined in clauses 4.3.3.2 and 4.3.3.3.

This preamble is repeated 8 times, to create a 8 x 4 096-chip-long sequence. This sequence is followed by a short Unique Word marker of length 16 QPSK symbols (SF = 256). This UW is assumed to be differentially encoded. Its value is 0x78DB.

The preamble can thus be seen as 128 symbols with no modulation, followed by 16 modulated symbols (UW).

4.3.3.2 Preamble scrambling code

The scrambling code for the PRACH preamble part is constructed from the long scrambling sequences. There are 8 192 PRACH preamble scrambling codes in total.

The n:th preamble scrambling code, n = 0, 1, …, 8 191, is defined as:

$$S_{r\text{-pre},n}(i) = c_{long,1,n}(i), i = 0, 1, …, 4\ 095$$

Where the sequence $c_{long,1,n}$ is defined in clause 4.3.2.2.

The 8 192 PRACH preamble scrambling codes are divided into 512 groups with 16 codes in each group. There is a one-to-one correspondence between the group of PRACH preamble scrambling codes in a cell and the primary scrambling code used in the downlink of the cell. The k:th PRACH preamble scrambling code within the cell with downlink primary scrambling code m, k = 0, 1, 2, …, 15 and m = 0, 1, 2, …, 511, is $S_{r\text{-pre},n}(i)$ as defined above with $n = 16 \times m + k$.

4.3.3.3 Preamble signature

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

-   $C_{sig,s}(i) = P_s(i \text{ modulo } 16), i = 0, 1, …, 4\ 095$

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

4.4 Modulation

4.4.1 Modulating chip rate

The modulating chip rate is 3,84 Mcps.

4.4.2 Modulation

In the uplink, the complex-valued chip sequence generated by the spreading process is QPSK modulated as shown in figure 6:
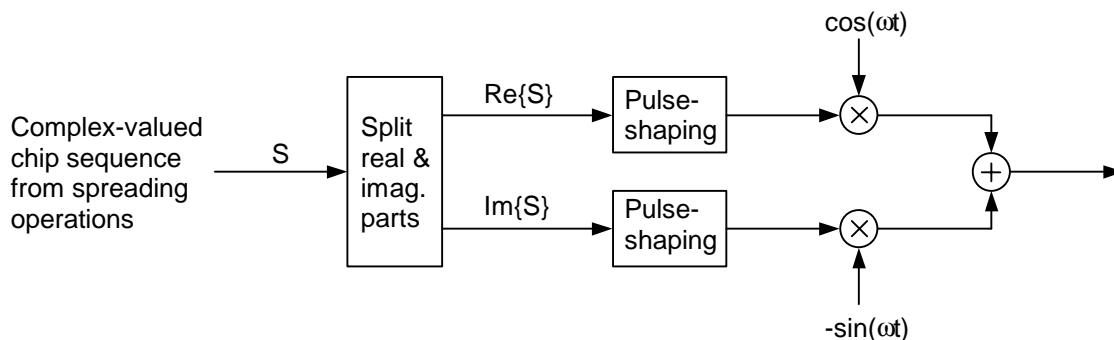
Figure 6: Uplink modulation

The pulse-shaping characteristics are described in [6].

5   Downlink spreading and modulation

5.1   Spreading

Figure 7 illustrates the spreading operation for all downlink physical channels except SCH, i.e. for P-CCPCH, S-CCPCH, CPICH, PICH, PDSCH and downlink DPCH. The non-spread physical channel consists of a sequence of real-valued symbols. For all channels except AICH, the symbols can take the three values + 1, -1, and 0, where 0 indicates DTX.

As already mentioned in [1], HPPICH is neither spread nor scrambled.

Each pair of two consecutive symbols is first serial-to-parallel converted and mapped to an I and Q branch. The mapping is such that even and odd numbered symbols are mapped to the I and Q branch respectively. Symbol number zero is defined as the first symbol in each frame. The I and Q branches are then spread to the chip rate by the same real-valued channelization code $C_{ch,SF,m}$. The sequences of real-valued chips on the I and Q branch are then treated as a single complex-valued sequence of chips. This sequence of chips is scrambled (complex chip-wise multiplication) by a complex-valued scrambling code $S_{dl,n}$. In case of P-CCPCH, the scrambling code is applied aligned with the P-CCPCH frame boundary, i.e. the first complex chip of the spread P-CCPCH frame is multiplied with chip number zero of the scrambling code. In case of other downlink channels, the scrambling code is applied aligned with the scrambling code applied to the P-CCPCH. In this case, the scrambling code is thus not necessarily applied aligned with the frame boundary of the physical channel to be scrambled.
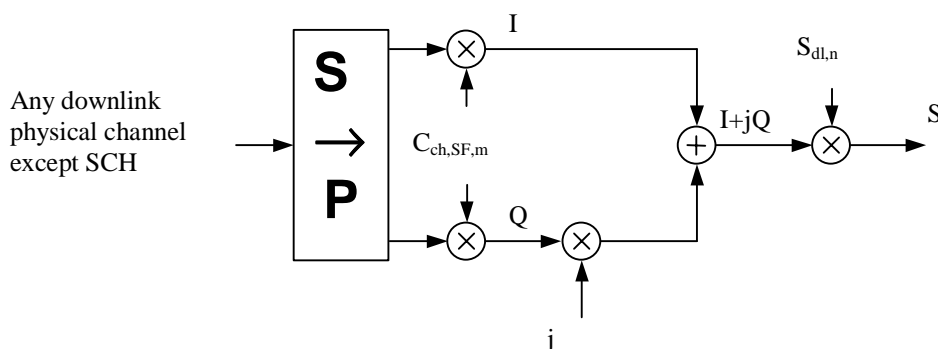
Figure 7: Spreading for all downlink physical channels except SCH

Figure 8 illustrates how different downlink channels are combined. Each complex-valued spread channel, corresponding to point S in figure 7, is separately weighted by a weight factor G_i. The complex-valued SCH and SCH, as described in [1] clause 5.3.3.4, are separately weighted by weight factors G_p and G_s. All downlink physical channels are then combined using complex addition.
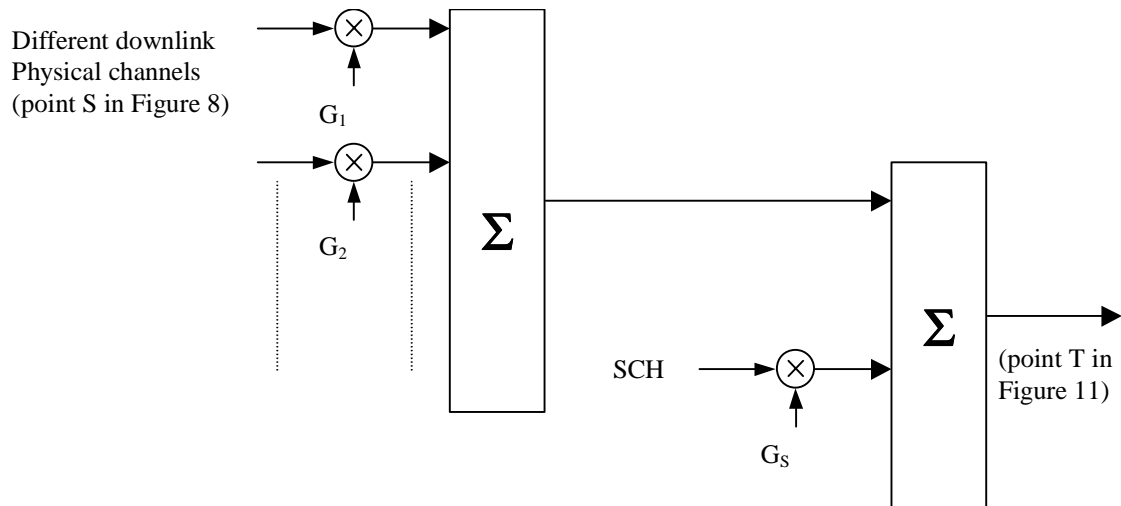
Figure 8: Spreading and modulation for SCH and P-CCPCH

5.2 Code generation and allocation

5.2.1 Channelization codes

The channelization codes of figure 7 are the same codes as used in the uplink, namely Orthogonal Variable Spreading Factor (OVSF) codes that preserve the orthogonality between downlink channels of different rates and spreading factors. The OVSF codes are defined in figure 3 in clause 4.3.1.

The channelization code for the Primary CPICH is fixed to $C_{ch,256,0}$ and the channelization code for the Primary CCPCH is fixed to $C_{ch,256,1}$. The channelization codes for all other physical channels are assigned by UTRAN.

With the spreading factor 512 a specific restriction is applied. When the code word $C_{ch,512,n}$, with n = 0,2,4….510, is used in soft handover, then the code word $C_{ch,512,n+1}$ is not allocated in the Node Bs where timing adjustment is to be used. Respectively if $C_{ch,512,n}$, with n = 1,3,5….511 is used, then the code word $C_{ch,512,n-1}$ is not allocated in the Node B where timing adjustment is to be used. This restriction shall not apply for the softer handover operation or in case UTRAN is synchronized to such a level that timing adjustments in soft handover are not used with spreading factor 512.

When compressed mode is implemented by reducing the spreading factor by 2, the OVSF code used for compressed frames is:

-    $C_{ch,SF/2,\lfloor n/2 \rfloor}$ if ordinary scrambling code is used;

-    $C_{ch,SF/2,n \bmod SF/2}$ if alternative scrambling code is used (see clause 5.2.2);

Where $C_{ch,SF,n}$ is the channelization code used for non-compressed frames.

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.

5.2.3 Scrambling code

5.2.3.1 Long scrambling code

A total of 2^{18}-1 = 262,143 long scrambling codes, numbered 0…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 x i where i = 0…511. The i:th set of secondary scrambling codes consists of scrambling codes 16 x i $_+$ k, where k = 1…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 secondary scrambling codes.

Hence, according to the above, scrambling codes k = 0, 1, …, 8 191 are used. Each of these codes is associated with a left alternative scrambling code and a right alternative scrambling code, that may be used for compressed frames. The left alternative scrambling code corresponding to scrambling code k is scrambling code number k + 8 192, while the right alternative scrambling code corresponding to scrambling code k is scrambling code number k + 16 384. The alternative scrambling codes can be used for compressed frames. In this case, the left alternative scrambling code is used if n<SF/2 and the right alternative scrambling code is used if n≥SF/2, where $c_{ch,SF,n}$ is the channelization code used for non-compressed frames. The usage of alternative scrambling code for compressed frames is signalled by higher layers for each physical channel respectively.

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 x 8 x j + 16 x k, where j = 0..63 and k = 0..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. However, in the case of the CCTrCH of type DSCH then all the PDSCH channellization codes that a single UE may receive shall be under a single scrambling code (either the primary or a secondary scrambling code).

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 38 400 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) = ... = x (16) = x (17) = 0;

- y(0) = y(1) = … = y(16) = y(17) =1.

Recursive definition of subsequent symbols:

- x(i + 18) = x(i + 7) + x(i) modulo 2, i = 0,…,2^{18}-20;

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

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

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

These binary sequences are converted to real valued sequences Z_n by the following transformation:

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

Finally, the n:th complex scrambling code sequence $S_{dl,n}$ is defined as:

- $S_{dl,n}(i) = Z_n(i) + j \ Z_n((i + 131\ 072)\ modulo\ (2^{18}\text{-}1)),\ i = 0,1,\ldots, 38\ 399$

Note that the pattern from phase 0 up to the phase of 38 399 is repeated.

Figure 9: Configuration of downlink scrambling code generator

5.2.3   Short scrambling code

The code used for scrambling of the uplink DPCCH/DPDCH may be of either long or short type. Clause 5.2.2 specifies the long scrambling code, whereas clause 5.2.3 describes the short code option. The decision on the use of whether short or long scrambling codes is taken by higher layers.

A total of $2^8-1 = 255$ Gold-like scrambling codes, numbered 1,…,255, can be generated. Due to the cross-correlation characteristics of Gold codes, only a subset of these can be used simultaneously. Table 4 shows the 85 groups of three Gold codes whose simultaneous use is prohibited. In practice, one code of each group is selected, up to a total of 85 different codes.

Table 4: Groups of Gold codes with bad cross-correlation properties

1, 246, 247	18, 66, 80	37, 133, 160	62, 157, 163	95, 136, 215
2, 237, 239	19, 167, 180	38, 158, 184	63, 84, 107	96, 131, 227
3, 24, 27	20, 130, 150	39, 79, 104	64, 189, 253	99, 155, 248
4, 41, 45	21, 97, 116	40, 218, 242	67, 166, 229	100, 170, 206
5, 219, 222	22, 111, 121	43, 194, 233	68, 144, 212	103, 178, 213
6, 192, 198	23, 142, 153	44, 223, 243	71, 139, 204	105, 185, 208
7, 49, 54	25, 236, 245	47, 196, 235	73, 174, 231	106, 162, 200
8, 164, 172	26, 238, 244	48, 193, 241	74, 182, 252	109, 148, 249
9, 83, 90	28, 46, 50	51, 217, 234	77, 135, 202	110, 143, 225
10, 65, 75	29, 197, 216	52, 220, 232	78, 159, 209	113, 186, 203
11, 183, 188	30, 195, 221	55, 199, 240	81, 181, 228	114, 161, 211
12, 129, 141	31, 42, 53	56, 93, 101	82, 173, 255	117, 151, 226
13, 119, 122	32, 94, 126	57, 146, 171	85, 156, 201	118, 140, 250
14, 98, 108	33, 137, 168	58, 138, 176	86, 132, 210	120, 152, 224
15, 149, 154	34, 145, 179	59, 70, 125	88, 190, 230	123, 128, 251
16, 175, 191	35, 69, 102	60, 76, 112	91, 165, 254	124, 177, 205
17, 72, 89	36, 87, 115	61, 134, 187	92, 147, 207	127, 169, 214

The scrambling code sequences are constructed as the position wise modulo 2 sum of two binary m-sequences generated by means of two generator polynomials of degree 8. The resulting sequences thus constitute segments of a set of Gold sequences. The scrambling codes are repeated for every 0,667 ms radio slot. Let x and y be the two sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $1 + X^2 + X^3 + {}^4 + X^8$. The y-sequence is constructed using the polynomial $1 + X^2 + X^3 + X^4 + X^5 + X^7 + X^8$.

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

Let $n_7 … n_0$ be the 8-bit binary representation of the scrambling code number n (decimal) with n_0 being the least significant bit. The x sequence depends on the chosen scrambling code 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:

   $x_n(0) = n_0$, $x_n(1) = n_1$, … ,$x_n(7) = n_7$;

   $y(0) = 0, y(1) = 1, y(2) = 0, y(3) = 0, y(4) = 1, y(5) = 0, y(6) = 0, y(7) = 1$.

Recursive definition of subsequent symbols:

   $x_n(i + 8) = x_n(i + 4) + x_n(i + 3) + x_n(i + 2) + x_n(i) \bmod 2$;

   $y(i + 8) = y(i + 7) + y(i + 5) + y(i + 4) + y(i + 3) + y(i + 2) + y(i) \bmod 2$.

The n-th Gold code sequence z_n, $n = 1,1,2,…,2^8-1$, is then defined as

   $z_n(i) = x_n(i) + y(i) \bmod 2$, $i = 1,…, 2^8-1$.

To generate an even length scrambling sequence a 0 is added at the end of the sequence, so that the length is 256 chips.

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

Note that the pattern from phase 0 up to the phase of 255 is repeated.

5.2.4 Synchronization codes

5.2.4.1 Code Generation

The code sequence, C_{psc} is constructed as a so-called generalized hierarchical Golay sequence. The SCH is furthermore chosen to have good a-periodic auto-correlation properties.

Letting a = $<x_1, x_2, x_3, \ldots, x_{16}>$ = <0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0> and

$$b = < x_1, x_2, \ldots, x_8, \bar{x}_9, \bar{x}_{10}, \ldots, \bar{x}_{16} > .$$

The PSC code is generated by repeating sequence 'a' modulated by a Golay complementary sequence.

Letting $y = < a, a, a, \bar{a}, \bar{a}, a, \bar{a}, \bar{a}, a, a, a, \bar{a}, a, \bar{a}, a, a > .$

The definition of the PSC code word C_{psc} follows (the left most index corresponds to the chip transmitted first in each time slot):

$$C_{psc} = <y(0), y(1), y(2), \ldots, y(255)>.$$

This PSC binary code words is converted to real valued sequences by the transformation
'0'->' + 1', '1->'-1'.

5.3 Modulation

5.3.1 Modulating rate

The modulating chip rate is 3,84 Mcps. As stated in [1], HPPICH is not spread, and the bit rate is 15 kbps.

5.3.2 Modulation

In the downlink, the complex-valued chip sequence generated by the spreading process is QPSK modulated as shown in figure 10. As stated in [1], the real-valued HPPICH is instead BPSK modulated, thus using the real chain in figure 10.
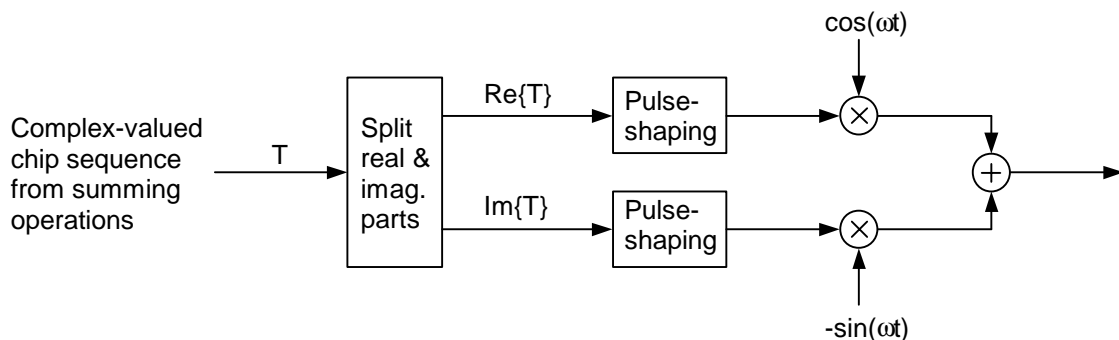

Figure 10: Downlink modulation

The pulse-shaping characteristics are described in [7].

Annex A (informative):
Generalized Hierarchical Golay Sequences

A.1   Alternative generation

The generalized hierarchical Golay sequences for the PSC described in clause 5.2.3.1 may be also viewed as generated (in real valued representation) by the following methods:

Method 1

The sequence y is constructed from two constituent sequences x_1 and x_2 of length n_1 and n_2 respectively using the following formula:

$$y(i) = x_2(i \bmod n_2) \times x_1(i \operatorname{div} n_2), i = 0 \ldots (n_1 \times n_2)\text{-}1.$$

The constituent sequences x_1 and x_2 are chosen to be the following length 16 (i.e. $n_1 = n_2 = 16$) sequences:

- x_1 is defined to be the length 16 ($N^{(1)} = 4$) Golay complementary sequence obtained by the delay matrix $D^{(1)} = [8, 4, 1, 2]$ and weight matrix $W^{(1)} = [1, -1, 1, 1]$.

- x_2 is a generalized hierarchical sequence using the following formula, selecting s = 2 and using the two Golay complementary sequences x_3 and x_4 as constituent sequences. The length of the sequence x_3 and x_4 is called n_3 respectively n_4.

$$x_2(i) = x_4(i \bmod s + s \times (i \operatorname{div} sn_3)) \times x_3((i \operatorname{div} s) \bmod n_3), \; i = 0 \ldots (n_3 \times n_4)\text{-}1;$$

x_3 and x_4 are defined to be identical and the length 4 ($N^{(3)} = N^{(4)} = 2$) Golay complementary sequence obtained by the delay matrix $D^{(3)} = D^{(4)} = [1, 2]$ and weight matrix $W^{(3)} = W^{(4)} = [1, 1]$.

The Golay complementary sequences x_1, x_3 and x_4 are defined using the following recursive relation:

$$a_0(k) = \delta(k) \text{ and } b_0(k) = \delta(k);$$

$$a_n(k) = a_{n-1}(k) + W^{(j)}_n \times b_{n-1}(k\text{-}D^{(j)}_n);$$

$$b_n(k) = a_{n-1}(k)\text{-}W^{(j)}_n \times b_{n-1}(k\text{-}D^{(j)}_n);$$

$$k = 0, 1, 2, \ldots, 2 \times N^{(j)}\text{-}1;$$

$$n = 1, 2, \ldots, N^{(j)}.$$

The wanted Golay complementary sequence x_j is defined by a_n assuming $n = N^{(j)}$. The Kronecker delta function is described by δ, k,j and n are integers.

Method 2

The sequence y can be viewed as a pruned Golay complementary sequence and generated using the following parameters which apply to the generator equations for a and b above:

(a) Let $j = 0$, $N^{(0)} = 8$;

(b) $[D_1^0, D_2^0, D_3^0, D_4^0, D_5^0, D_6^0, D_7^0, D_8^0] = [128, 64, 16, 32, 8, 1, 4, 2]$;

(c) $[W_1^0, W_2^0, W_3^0, W_4^0, W_5^0, W_6^0, W_7^0, W_8^0] = [1, -1, 1, 1, 1, 1, 1, 1]$;

(d) For $n = 4, 6$, set $b_4(k) = a_4(k)$, $b_6(k) = a_6(k)$.

Annex B (informative): Bibliography

- ETSI TS 101 851-2: "Satellite Component of UMTS/IMT 2000; A-family; Part 2: Multiplexing and channel coding (S-UMTS-A 25.212)".

- ETSI TS 125 214: "Universal Mobile Telecommunication System (UMTS); Physical layer procedures (FDD) (3G TS 25.214 version 3.3.0 Release 1999)".

History

ETSI		
V1.1.1	December 2000	Publication