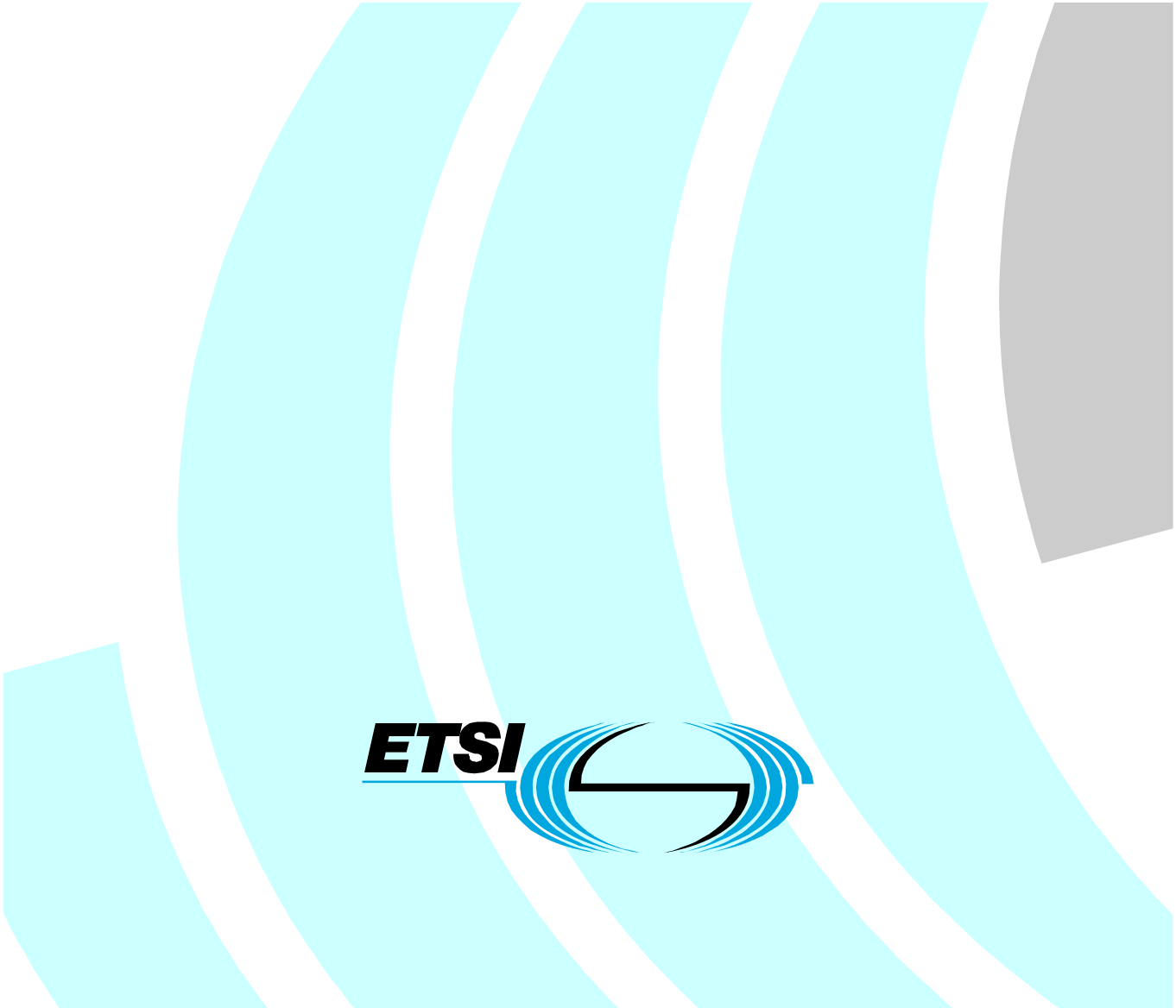
ETSI TS 101 851-3-1 V2.1.1 (2008-01)

Technical Specification

Satellite Earth Stations and Systems (SES);
Satellite Component of UMTS/IMT-2000;
Part 3: Spreading and modulation;
Sub-part 1: G-family (S-UMTS-G 25.213)



Reference
RTS/SES-00298-3-1

Keywords
interface, MES, MSS, radio, 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

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://webapp.etsi.org/IPR/home.asp).

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 present document is specifying the Satellite Radio Interface referenced as SRI Family G at ITU-R, in the frame of the modification of ITU-R Recommendation M.1457 [4]. This modification has been approved at SG8 meeting in November 2005.

The present document is part 3, sub-part 1 of a multi-part deliverable covering Satellite Earth Stations and Systems (SES); Satellite Component of UMTS/IMT-2000; G-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";

Sub-part 1: "G-family (S-UMTS-G 25.213)";

Sub-part 2: "A-family (S-UMTS-A 25.213)";

Part 4: "Physical layer procedures";

Part 5: "UE Radio Transmission and Reception";

Part 6: "Ground stations and space segment radio transmission and reception".

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, etc.) 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.

  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 describes spreading and modulation for the Physical Layer for family G of the satellite component of UMTS (S-UMTS-G).

It is based on the FDD mode of UTRA defined by TS 101 851-1-1 [1], TS 101 851-2-1 [2], TS 101 851-4-1 [3] and adapted for operation over satellite transponders.

2 References

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.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

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

[2] ETSI TS 101 851-2-1: "Satellite Earth Stations and Systems (SES); Satellite Component of UMTS/IMT-2000; Part 2: Multiplexing and channel coding; Sub-part 1: G-family (S-UMTS-G 25.212)".

[3] ETSI TS 101 851-4-1: "Satellite Earth Stations and Systems (SES); Satellite Component of UMTS/IMT-2000; Part 4: Physical layer procedures; Sub-part 1: G-family (S-UMTS-G 25.214)".

[4] ITU-R Recommendation M.1457 (2006): "Detailed specifications of the radio interfaces of International Mobile Telecommunications-2000 (IMT-2000)".

3 Symbols and abbreviations

3.1 Symbols

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

$C_{ch,SF,n}$	n:th channelization 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/PCPCH signature code for signature *s*
$S_{dpch,n}$	*n*:th DPCCH/DPDCH uplink scrambling code
$S_{r\text{-}pre,n}$	*n*:th PRACH preamble scrambling code
$S_{r\text{-}msg,n}$	*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
DPCCH	Dedicated Physical Control CHannel
DPCH	Dedicated Physical CHannel
DPDCH	Dedicated Physical Data CHannel
DTX	Discontinuous Transmission
FDD	Frequency Division Duplex
GEO	Geostationary Earth Orbit
LEO	Low Earth Orbit
Mcps	Mega chip per second
MEO	Medium Earth Orbit
MICH	MBMS Indication CHannel
MSS	Mobile Satellite Services
OVSF	Orthogonal Variable Spreading Factor (codes)
PICH	Page Indication CHannel
PRACH	Physical Random Access CHannel
PSC	Primary Synchronization Code
QPSK	Quaternary Phase Shift Keying
SCH	Synchronization CHannel
SF	Spreading Factor
SSC	Secondary Synchronization Code
USRAN	UMTS Satellite Radio Access Network
UTRA	UMTS Terrestrial Radio Access

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 DPCCH/DPDCH

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, and the binary value "1" is mapped to the real value -1. The DPCCH is spread to the chip rate by the channelization code c_c. The *n*:th DPDCH called $DPDCH_n$ is spread to the chip rate by the channelization code $c_{d,n}$. One DPCCH, up to six parallel DPDCHs, i.e. $1 \leq n \leq 6$.

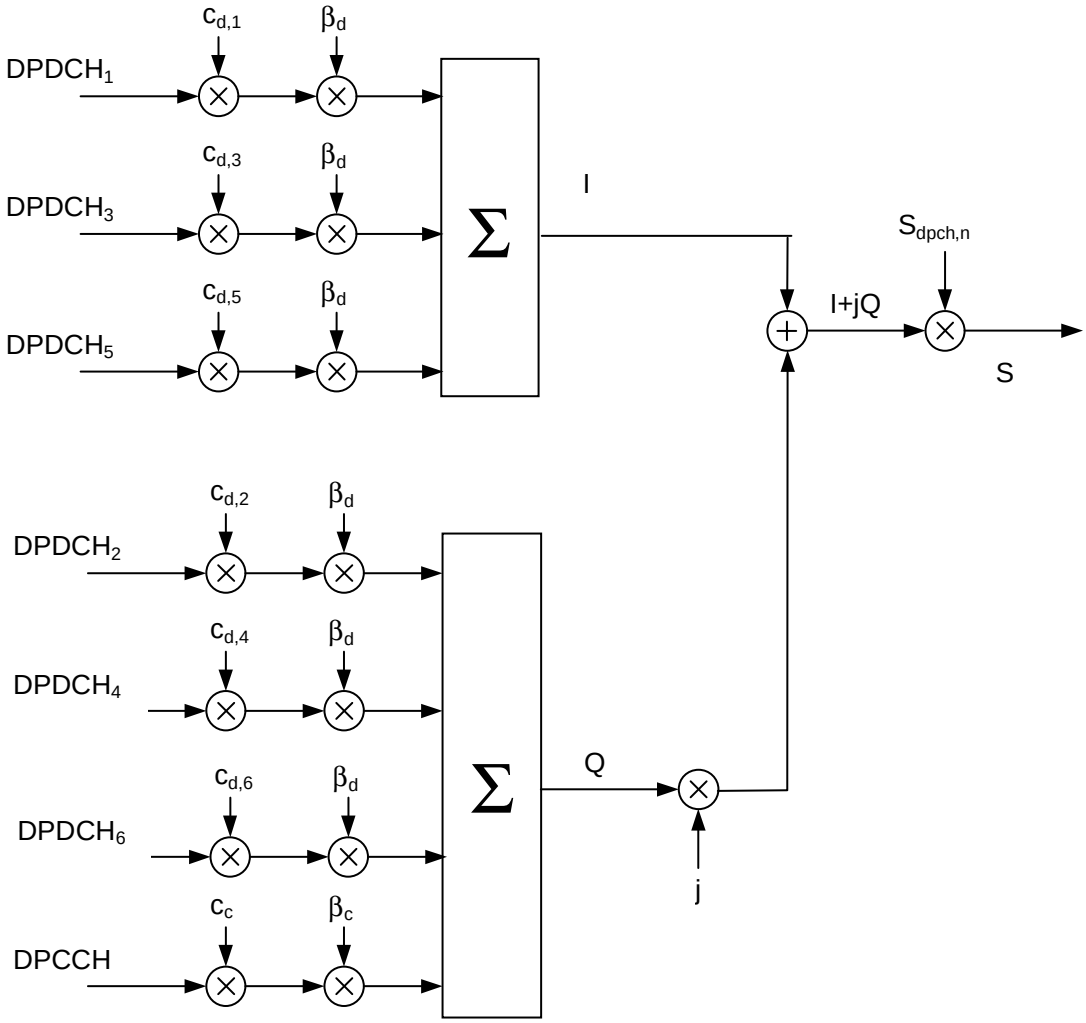


Figure 1: Spreading for uplink DPCCH and DPDCHs

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

The β_c and β_d values are signalled by higher layers or calculated as described in TS 101 851-4-1 [3]. At every instant in time, at least one of the values β_c and β_d has the amplitude 1,0. The β_c and β_d 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.

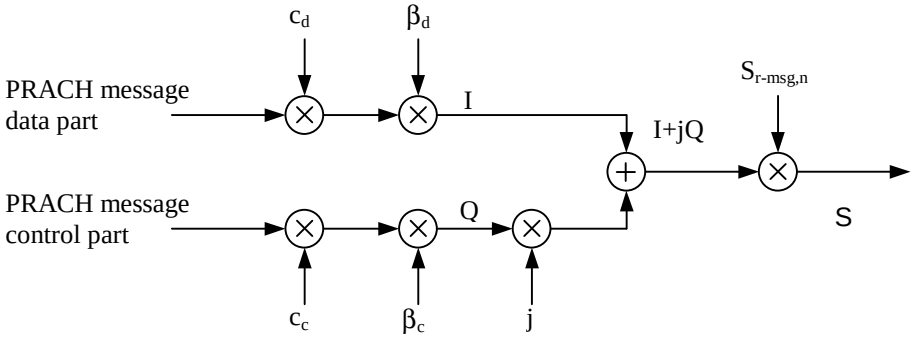


Figure 2: Spreading of PRACH message part

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 is treated as a complex-valued stream of chips. This complex-valued signal is then scrambled by the complex-valued scrambling code $S_{r\text{-}msg,n}$. 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.

4.3 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 3.

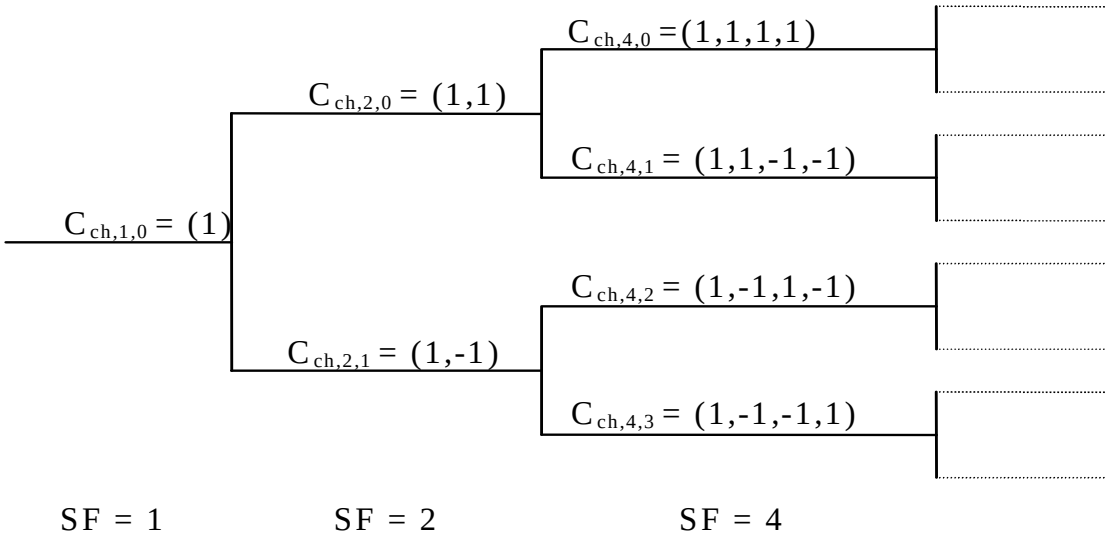


Figure 3: Code-tree for generation of Orthogonal Variable Spreading Factor (OVSF) codes

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:

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

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

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

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 $c_c = C_{ch,256,0}$.
- 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 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 channelization 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 $m = SF \times s / 16$.

4.3.2 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.

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 chips 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 chips shifted version of the sequence $c_{long,1,n}$.

Let $n_{23} \ldots n_0$ 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:

- $x_n(0) = n_0$, $x_n(1) = n_1$, … = $x_n(22) = n_{22}$,$x_n(23) = n_{23}$, $x_n(24) = 1$.
- $y(0) = y(1) = \ldots = y(23) = y(24) = 1$.

Recursive definition of subsequent symbols:

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

Define the binary Gold sequence z_n by:

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

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:

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

$c_{long,2,n}(i) = Z_n((i + 16\ 777\ 232)$ 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.

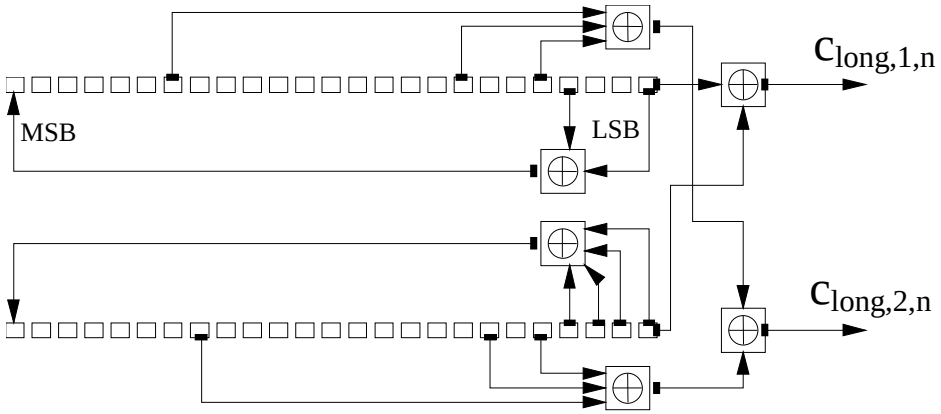


Figure 4: Configuration of uplink scrambling sequence generator

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 $c_{short,2,n}(i)$, $i = 0, 1, \ldots, 255$

$z_n(i)$	$c_{short,1,n}(i)$	$c_{short,2,n}(i)$
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 chips sequence to be extended by one chip is shown in figure 5.

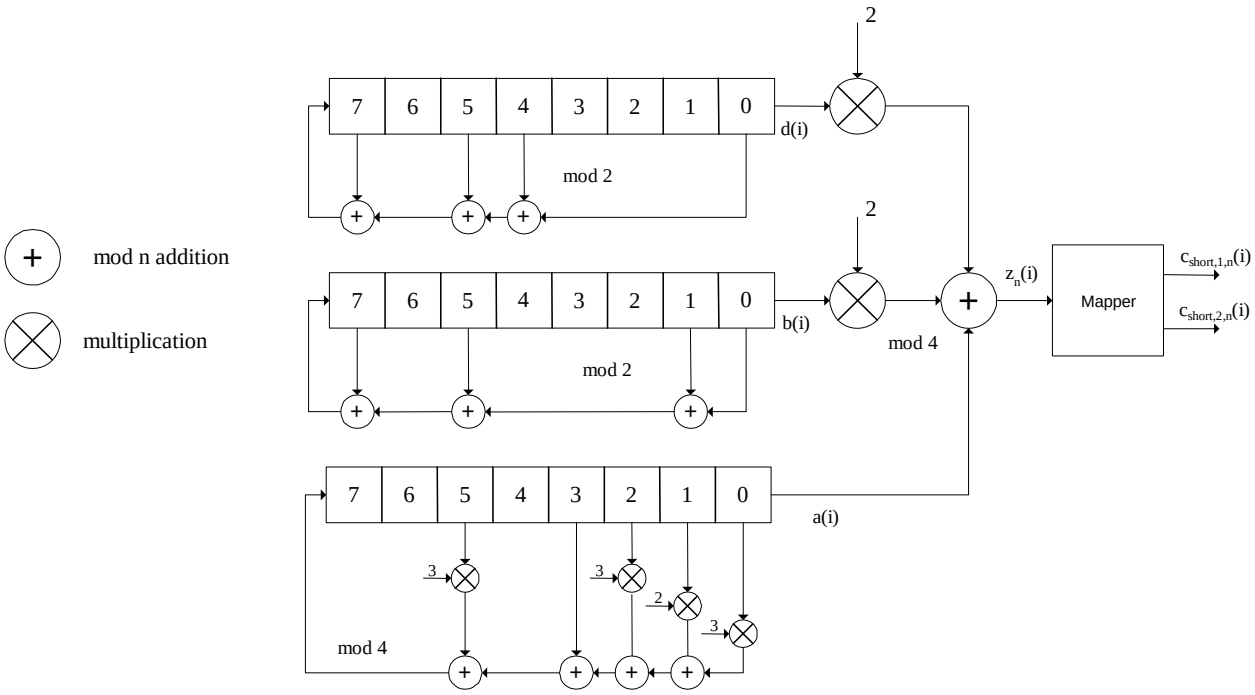


Figure 5: Uplink short scrambling sequence generator for 255 chips sequence

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.

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.

4.3.3 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 respectively.

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

Modulation of the complex-valued chip sequence generated by the spreading process is shown in figure 6.

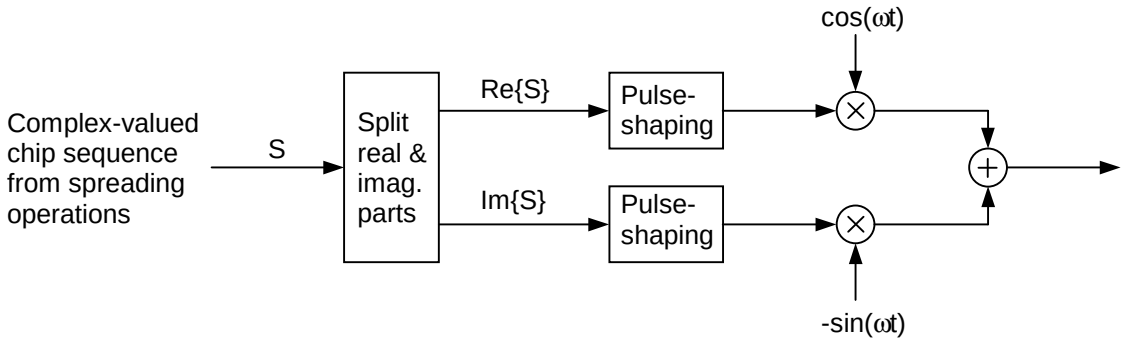


Figure 6: Uplink modulation

The pulse-shaping characteristics are described in TS 101 851-2-1 [2].

5 Downlink spreading and modulation

5.1 Spreading

Figure 7 illustrates the spreading operation for the physical channel except SCH. The downlink physical channels using QPSK are P-CCPCH, S-CCPCH, CPICH, AICH, PICH, MICH and downlink DPCH. The downlink physical channel using either QPSK. The non-spread downlink physical channels, except SCH, AICH, consist of a sequence of 3-valued digits taking the values 0, 1 and "DTX". Note that "DTX" is only applicable to those downlink physical channels that support DTX transmission. In case of QPSK, these digits are mapped to real-valued symbols as follows: the binary value "0" is mapped to the real value +1, the binary value "1" is mapped to the real value -1 and "DTX" is mapped to the real value 0. For the indicator channels using signatures (AICH), the real-valued symbols depend on the exact combination of the indicators to be transmitted.

In case of QPSK, each pair of two consecutive real-valued symbols is first serial-to-parallel converted and mapped to an I and Q branch. The definition of the modulation mapper is such that even and odd numbered symbols are mapped to the I and Q branch respectively. In case of QPSK, for all channels except the indicator channels using signatures, symbol number zero is defined as the first symbol in each frame. For the indicator channels using signatures, symbol number zero is defined as the first symbol in each access slot. The I and Q branches are then both spread to the chip rate by the same real-valued channelization code $C_{ch,SF,m}$. The channelization code sequence shall be aligned in time with the symbol boundary. 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 $S_{dl,n}$.

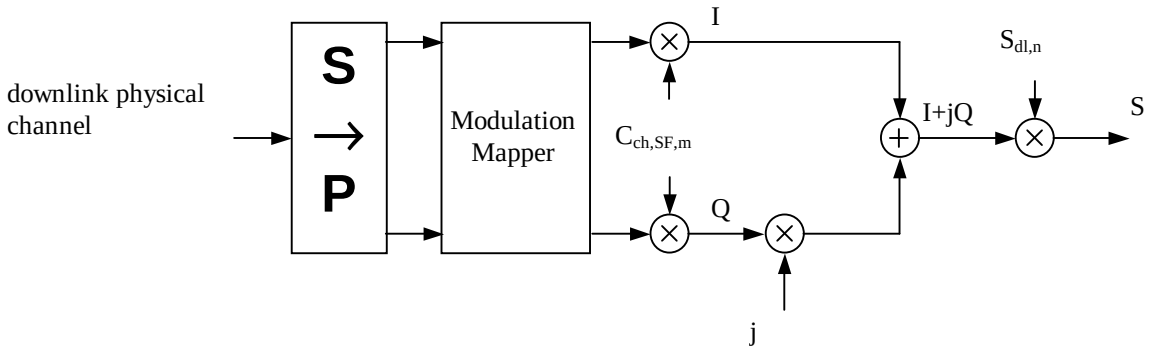


Figure 7: Spreading for all downlink physical channels except SCH

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 P-SCH and S-SCH, are separately weighted by weight factors G_p and G_s. All downlink physical channels are then combined using complex addition.

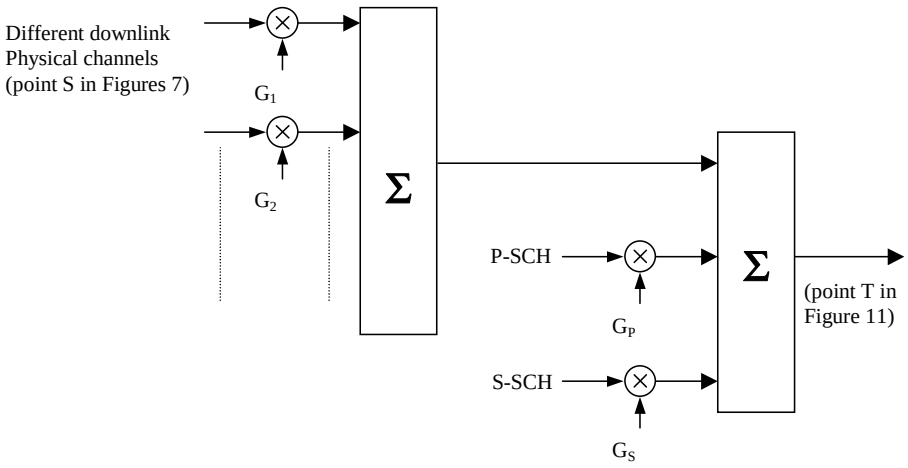


Figure 8: Combining of downlink physical channels

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 $C_{ch,256,0}$.The channelization codes for all other physical channels are assigned by USRAN.

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 cells 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 cells where timing adjustment is to be used. This restriction shall not apply in cases where 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.

5.2.2 Scrambling code

A total of $2^{18} - 1 = 262,143$ 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 are 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, primary CPICH, PICH, MICH, AICH and S-CCPCH carrying PCH 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 no more than one secondary scrambling code for one CCTrCH is allowable. In compressed mode during compressed frames, these can be changed to the associated left or right scrambling codes as described above, i.e. in these frames, the total number of different scrambling codes may exceed two.

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 chips 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, $n = 0,1,2,\ldots,2^{18} - 2$, 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} - 1)), i = 0, 1, … , 38 399.

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

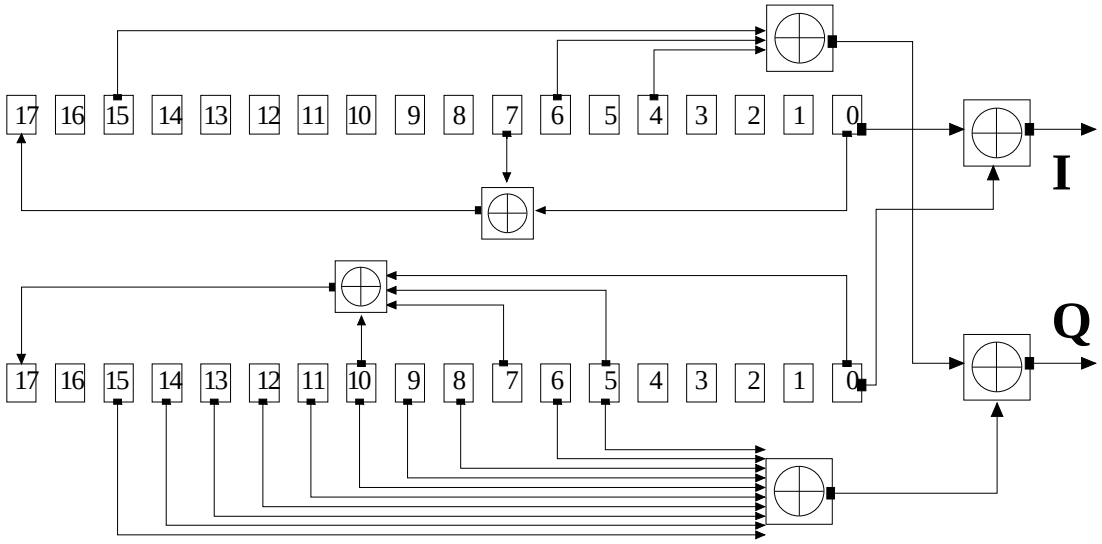


Figure 9: Configuration of downlink scrambling code generator

5.2.3 Synchronization codes

5.2.3.1 Code generation

The Primary Synchronization Code (PSC), C_{psc} is constructed as a so-called generalized hierarchical Golay sequence. The PSC is furthermore chosen to have good aperiodic auto correlation properties.

Define:

- a = <x_1, x_2, x_3, …, x_{16}> = <1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, 1>.

The PSC is generated by repeating the sequence *a* modulated by a Golay complementary sequence, and creating a complex-valued sequence with identical real and imaginary components. The PSC C_{psc} is defined as:

- C_{psc} = (1 + j) × <a, a, a, -a, -a, a, -a, -a, a, a, a, -a, a, -a, a, a>,

where the leftmost chip in the sequence corresponds to the chip transmitted first in time.

The 16 secondary synchronization codes (SSCs), {$C_{ssc,1}$,…,$C_{ssc,16}$}, are complex-valued with identical real and imaginary components, and are constructed from position wise multiplication of a Hadamard sequence and a sequence *z*, defined as:

- z = <b, b, b, -b, b, b, -b, -b, b, -b, b, -b, -b, -b, -b, -b>, where
- *b* = <x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, -x_9, -x_{10}, -x_{11}, -x_{12}, -x_{13}, -x_{14}, -x_{15}, -x_{16}> and x_1, x_2 , …, x_{15}, x_{16}, are same as in the definition of the sequence *a* above.

The Hadamard sequences are obtained as the rows in a matrix H_8 constructed recursively by:

$$H_0 = (1)$$

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}, \quad k \geq 1$$

The rows are numbered from the top starting with row *0* (the all ones sequence).

Denote the *n*:th Hadamard sequence as a row of H_8 numbered from the top, n = 0, 1, 2, …, 255, in the sequel.

Furthermore, let $h_n(i)$ and $z(i)$ denote the i:th symbol of the sequence h_n and z, respectively where i = 0, 1, 2, …, 255 and i = 0 corresponds to the leftmost symbol.

The k:th SSC, $C_{ssc,k}$, k = 1, 2, 3, …, 16 is then defined as:

- $C_{ssc,k} = (1 + j) \times <h_m(0) \times z(0), h_m(1) \times z(1), h_m(2) \times z(2), \ldots, h_m(255) \times z(255)>$,

where $m = 16 \times (k - 1)$ and the leftmost chip in the sequence corresponds to the chip transmitted first in time.

5.2.3.2 Code allocation of SSC

The 64 secondary SCH sequences are constructed such that their cyclic-shifts are unique, i.e., a non-zero cyclic shift less than 15 of any of the 64 sequences is not equivalent to some cyclic shift of any other of the 64 sequences. Also, a non-zero cyclic shift less than 15 of any of the sequences is not equivalent to itself with any other cyclic shift less than 15. Table 4 describes the sequences of SSCs used to encode the 64 different scrambling code groups. The entries in table 4 denote what SSC to use in the different slots for the different scrambling code groups, e.g. the entry "7" means that SSC $C_{ssc,7}$ shall be used for the corresponding scrambling code group and slot.

Table 4: Allocation of SSCs for secondary SCH

Scrambling Code Group	slot number														
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Group 0	1	1	2	8	9	10	15	8	10	16	2	7	15	7	16
Group 1	1	1	5	16	7	3	14	16	3	10	5	12	14	12	10
Group 2	1	2	1	15	5	5	12	16	6	11	2	16	11	15	12
Group 3	1	2	3	1	8	6	5	2	5	8	4	4	6	3	7
Group 4	1	2	16	6	6	11	15	5	12	1	15	12	16	11	2
Group 5	1	3	4	7	4	1	5	5	3	6	2	8	7	6	8
Group 6	1	4	11	3	4	10	9	2	11	2	10	12	12	9	3
Group 7	1	5	6	6	14	9	10	2	13	9	2	5	14	1	13
Group 8	1	6	10	10	4	11	7	13	16	11	13	6	4	1	16
Group 9	1	6	13	2	14	2	6	5	5	13	10	9	1	14	10
Group 10	1	7	8	5	7	2	4	3	8	3	2	6	6	4	5
Group 11	1	7	10	9	16	7	9	15	1	8	16	8	15	2	2
Group 12	1	8	12	9	9	4	13	16	5	1	13	5	12	4	8
Group 13	1	8	14	10	14	1	15	15	8	5	11	4	10	5	4
Group 14	1	9	2	15	15	16	10	7	8	1	10	8	2	16	9
Group 15	1	9	15	6	16	2	13	14	10	11	7	4	5	12	3
Group 16	1	10	9	11	15	7	6	4	16	5	2	12	13	3	14
Group 17	1	11	14	4	13	2	9	10	12	16	8	5	3	15	6
Group 18	1	12	12	13	14	7	2	8	14	2	1	13	11	8	11
Group 19	1	12	15	5	4	14	3	16	7	8	6	2	10	11	13
Group 20	1	15	4	3	7	6	10	13	12	5	14	16	8	2	11
Group 21	1	16	3	12	11	9	13	5	8	2	14	7	4	10	15
Group 22	2	2	5	10	16	11	3	10	11	8	5	13	3	13	8
Group 23	2	2	12	3	15	5	8	3	5	14	12	9	8	9	14
Group 24	2	3	6	16	12	16	3	13	13	6	7	9	2	12	7
Group 25	2	3	8	2	9	15	14	3	14	9	5	5	15	8	12
Group 26	2	4	7	9	5	4	9	11	2	14	5	14	11	16	16
Group 27	2	4	13	12	12	7	15	10	5	2	15	5	13	7	4
Group 28	2	5	9	9	3	12	8	14	15	12	14	5	3	2	15
Group 29	2	5	11	7	2	11	9	4	16	7	16	9	14	14	4
Group 30	2	6	2	13	3	3	12	9	7	16	6	9	16	13	12
Group 31	2	6	9	7	7	16	13	3	12	2	13	12	9	16	6
Group 32	2	7	12	15	2	12	4	10	13	15	13	4	5	5	10
Group 33	2	7	14	16	5	9	2	9	16	11	11	5	7	4	14
Group 34	2	8	5	12	5	2	14	14	8	15	3	9	12	15	9
Group 35	2	9	13	4	2	13	8	11	6	4	6	8	15	15	11
Group 36	2	10	3	2	13	16	8	10	8	13	11	11	16	3	5
Group 37	2	11	15	3	11	6	14	10	15	10	6	7	7	14	3
Group 38	2	16	4	5	16	14	7	11	4	11	14	9	9	7	5
Group 39	3	3	4	6	11	12	13	6	12	14	4	5	13	5	14
Group 40	3	3	6	5	16	9	15	5	9	10	6	4	15	4	10

Scrambling Code Group	slot number														
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Group 41	3	4	5	14	4	6	12	13	5	13	6	11	11	12	14
Group 42	3	4	9	16	10	4	16	15	3	5	10	5	15	6	6
Group 43	3	4	16	10	5	10	4	9	9	16	15	6	3	5	15
Group 44	3	5	12	11	14	5	11	13	3	6	14	6	13	4	4
Group 45	3	6	4	10	6	5	9	15	4	15	5	16	16	9	10
Group 46	3	7	8	8	16	11	12	4	15	11	4	7	16	3	15
Group 47	3	7	16	11	4	15	3	15	11	12	12	4	7	8	16
Group 48	3	8	7	15	4	8	15	12	3	16	4	16	12	11	11
Group 49	3	8	15	4	16	4	8	7	7	15	12	11	3	16	12
Group 50	3	10	10	15	16	5	4	6	16	4	3	15	9	6	9
Group 51	3	13	11	5	4	12	4	11	6	6	5	3	14	13	12
Group 52	3	14	7	9	14	10	13	8	7	8	10	4	4	13	9
Group 53	5	5	8	14	16	13	6	14	13	7	8	15	6	15	7
Group 54	5	6	11	7	10	8	5	8	7	12	12	10	6	9	11
Group 55	5	6	13	8	13	5	7	7	6	16	14	15	8	16	15
Group 56	5	7	9	10	7	11	6	12	9	12	11	8	8	6	10
Group 57	5	9	6	8	10	9	8	12	5	11	10	11	12	7	7
Group 58	5	10	10	12	8	11	9	7	8	9	5	12	6	7	6
Group 59	5	10	12	6	5	12	8	9	7	6	7	8	11	11	9
Group 60	5	13	15	15	14	8	6	7	16	8	7	13	14	5	16
Group 61	9	10	13	10	11	15	15	9	16	12	14	13	16	14	11
Group 62	9	11	12	15	12	9	13	13	11	14	10	16	15	14	16
Group 63	9	12	10	15	13	14	9	14	15	11	11	13	12	16	10

5.3 Modulation

5.3.1 Modulating chip rate

The modulating chip rate is 3,84 Mcps.

5.3.2 Modulation

Modulation of the complex-valued chip sequence generated by the spreading process is shown in figure 10.

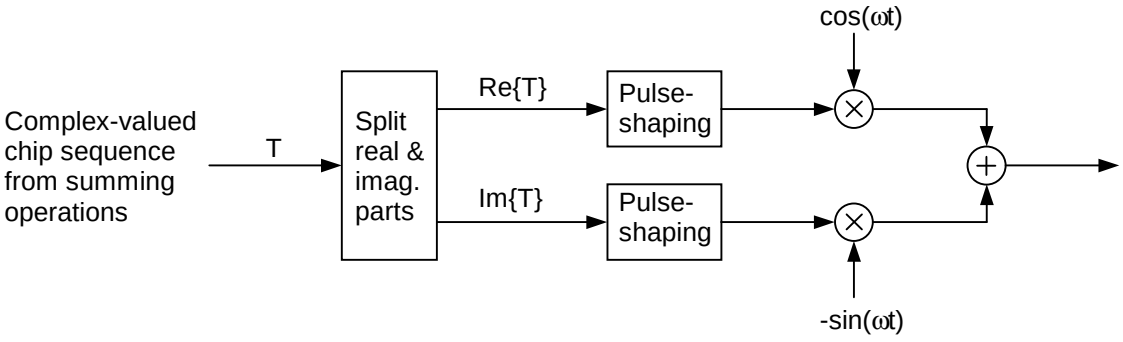


Figure 10: Downlink modulation

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:

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

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

$a_0(k) = \delta(k)$ and $b_0(k) = \delta(k)$;

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

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

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

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

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 $b_4(k) = a_4(k)$, $b_6(k) = a_6(k)$.

History

Document history		
V1.1.1	December 2000	Publication as TS 101 851-3
V1.2.1	January 2006	Publication as TS 101 851-3
V2.1.1	January 2008	Publication