

ETSI TS 125 213 V3.3.0 (2000-06)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
Spreading and modulation (FDD)
(3G TS 25.213 version 3.3.0 Release 1999)**



Reference

RTS/TSGR-0125213UR2

Keywords

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

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2000.

All rights reserved.

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 the ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

Contents

Foreword.....	5
1 Scope.....	6
2 References.....	6
3 Symbols and abbreviations	6
3.1 Symbols	6
3.2 Abbreviations.....	6
4 Uplink spreading and modulation.....	7
4.1 Overview.....	7
4.2 Spreading	7
4.2.1 DPCCH/DPDCH.....	7
4.2.2 PRACH	9
4.2.2.1 PRACH preamble part.....	9
4.2.2.2 PRACH message part	9
4.2.3 PCPCH.....	9
4.2.3.1 PCPCH preamble part	9
4.2.3.2 PCPCH message part.....	9
4.3 Code generation and allocation.....	10
4.3.1 Channelization codes	10
4.3.1.1 Code definition	10
4.3.1.2 Code allocation for DPCCH/DPDCH	11
4.3.1.3 Code allocation for PRACH message part	11
4.3.1.4 Code allocation for PCPCH message part	11
4.3.1.5 Channelisation code for PCPCH power control preamble.....	12
4.3.2 Scrambling codes	12
4.3.2.1 General	12
4.3.2.2 Long scrambling sequence	12
4.3.2.3 Short scrambling sequence	13
4.3.2.4 DPCCH/DPDCH scrambling code	14
4.3.2.5 PRACH message part scrambling code.....	15
4.3.2.6 PCPCH message part scrambling code.....	15
4.3.2.7 PCPCH power control preamble scrambling code	15
4.3.3 PRACH preamble codes.....	15
4.3.3.1 Preamble code construction.....	15
4.3.3.2 Preamble scrambling code.....	16
4.3.3.3 Preamble signature	16
4.3.4 PCPCH preamble codes	16
4.3.4.1 Access preamble.....	16
4.3.4.1.1 Access preamble code construction	16
4.3.4.1.2 Access preamble scrambling code	17
4.3.4.1.3 Access preamble signature.....	17
4.3.4.2 CD preamble.....	17
4.3.4.2.1 CD preamble code construction	17
4.3.4.2.2 CD preamble scrambling code.....	17
4.3.4.2.3 CD preamble signature	18
4.4 Modulation.....	18
4.4.1 Modulating chip rate	18
4.4.2 Modulation	18
5 Downlink spreading and modulation.....	18
5.1 Spreading	18
5.2 Code generation and allocation.....	19
5.2.1 Channelization codes	19
5.2.2 Scrambling code.....	20
5.2.3 Synchronisation codes.....	21
5.2.3.1 Code generation.....	21

5.2.3.2 Code allocation of SSC..... 22

5.3 Modulation..... 24

5.3.1 Modulating chip rate 24

5.3.2 Modulation..... 24

Annex A (informative): Generalised Hierarchical Golay Sequences 25

A.1 Alternative generation..... 25

Annex B (informative): Change history 26

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

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] 3G TS 25.201: "Physical layer - general description".
- [2] 3G TS 25.211: "Physical channels and mapping of transport channels onto physical channels (FDD)".
- [3] 3G TS 25.101: "UE Radio transmission and Reception (FDD)".
- [4] 3G TS 25.104: "UTRA (BS) FDD; Radio transmission and Reception".
-

3 Symbols and abbreviations

3.1 Symbols

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

$C_{ch,SF,n}$:	n :th channelisation code with spreading factor SF
$C_{pre,n,s}$:	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
$S_{dpch,n}$:	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_{c-acc} :	n :th PCPCH access preamble scrambling code
S_{c-cd} :	n :th PCPCH CD preamble scrambling code
$S_{c-msg,n}$:	n :th PCPCH 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
AP	Access Preamble
BCH	Broadcast Control Channel
CCPCH	Common Control Physical Channel

CD	Collision Detection
CPCH	Common Packet 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 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 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, 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_c , while the n :th DPDCH called $DPDCH_n$ is spread to the chip rate by the channelization code $c_{d,n}$. One DPCCH and up to six parallel DPDCHs can be transmitted simultaneously, i.e. $1 \leq n \leq 6$.

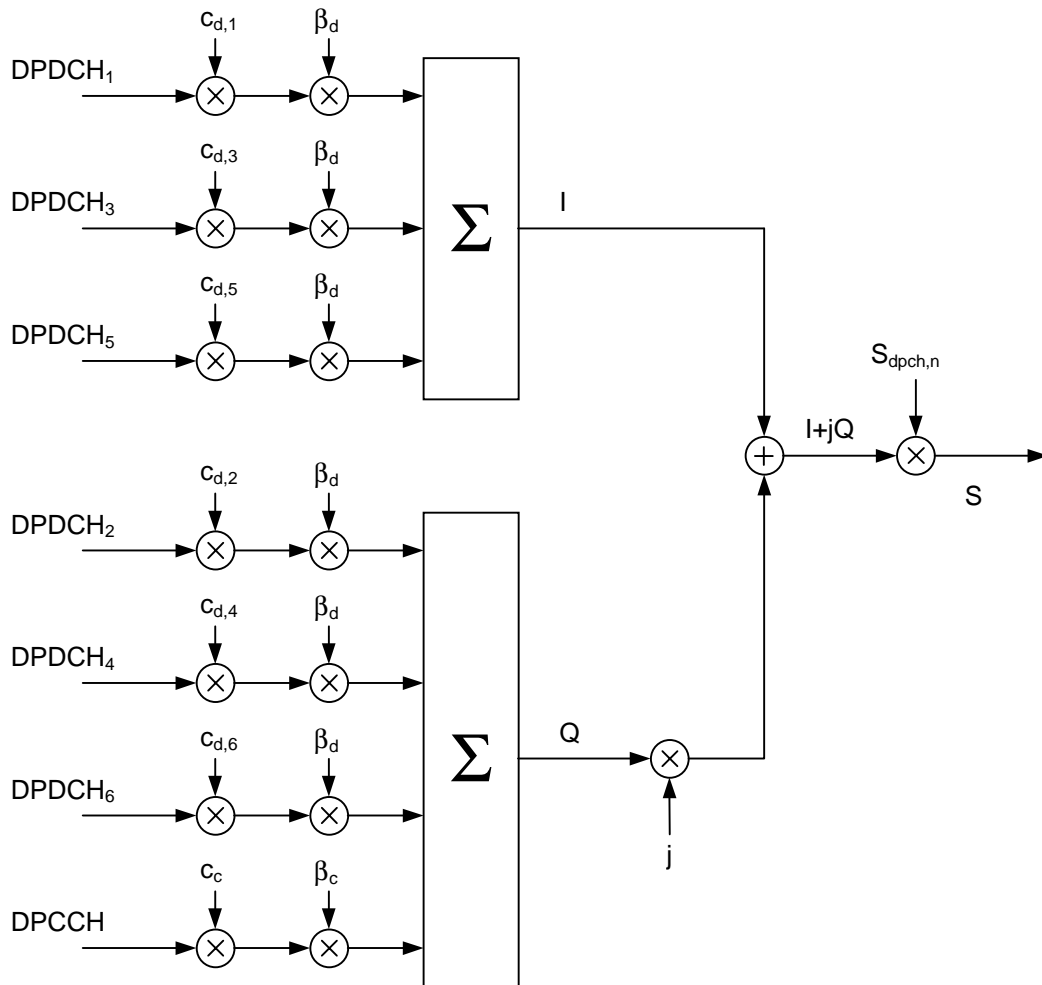


Figure 1: Spreading for uplink DPCCH and DPDCCHs

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

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 $S_{\text{dpch},n}$. 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 section 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 .

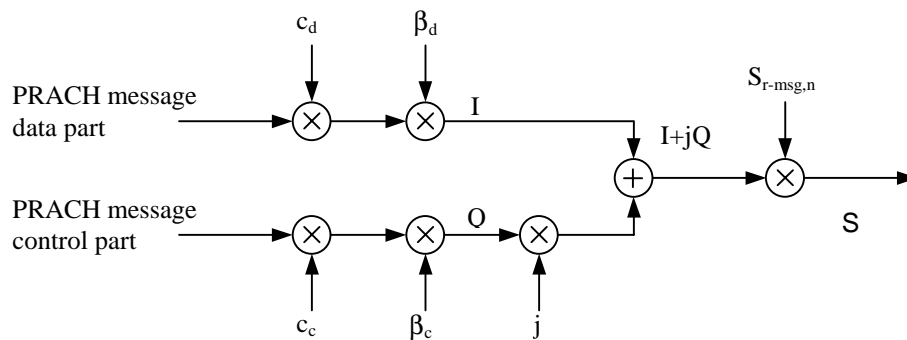


Figure 2: Spreading of PRACH message part

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 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 $S_{\text{r-msg},n}$. 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, described in section 4.3.4.

4.2.3.2 PCPCH message part

Figure 3 illustrates the principle of the spreading 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 .

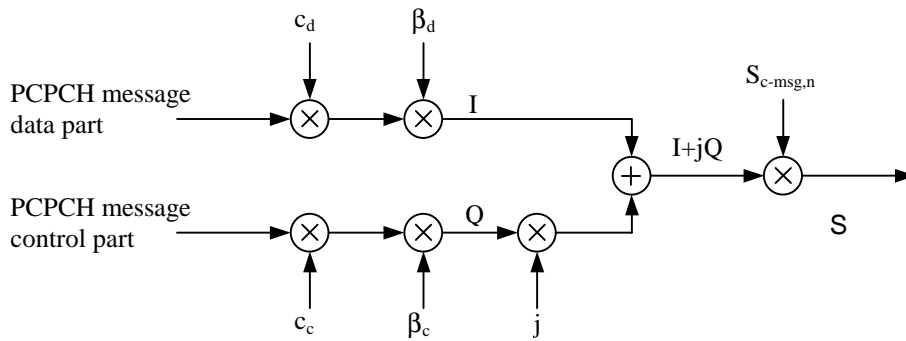


Figure 3: Spreading of PCPCH message part

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 $S_{c\text{-msg},n}$. 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.

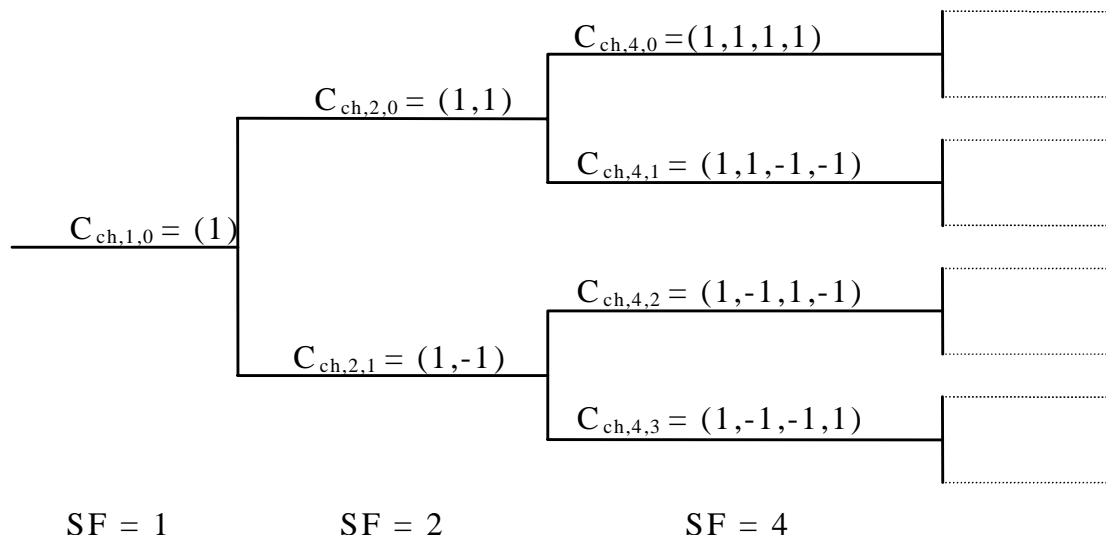


Figure 4: Code-tree for generation of Orthogonal Variable Spreading Factor (OVSF) codes

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 $c_c = C_{ch,256,0}$.
- 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 initialise a DCH, the channelisation 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 , $0 \leq s \leq 15$, 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 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 $m = 16 \times s + 15$. 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

For the control part and data part the following applies:

- The control part is always spread by code $c_c = C_{ch,256,0}$.
- The data part is spread by code $c_d = C_{ch,SF,k}$ where SF is the spreading factor of the data part and $k = SF / 4$.

The data part may use the code from spreading factor 4 to 256. A UE is allowed to increase SF during the message transmission on a frame by frame basis.

4.3.1.5 Channelisation code for PCPCH power control preamble

The channelisation code for the PCPCH power control preamble is the same as that used for the control part of the message part, as described in section 4.3.1.4 above.

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.

4.3.2.2 Long scrambling sequence

The long scrambling sequences $c_{\text{long},1,n}$ and $c_{\text{long},2,n}$ are constructed from 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 sequence $c_{\text{long},2,n}$ is a 16777232 chip shifted version of the sequence $c_{\text{long},1,n}$.

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:

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

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

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

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

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

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

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

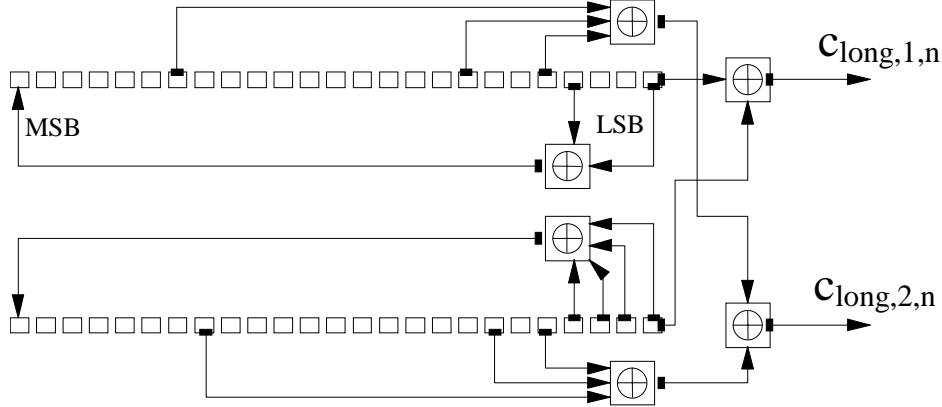


Figure 5: Configuration of uplink scrambling sequence generator

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 $n_{23}n_{22}\dots n_0$ 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 $d(i)$ is generated recursively by the polynomial $g_2(x) = x^8 + x^7 + x^5 + x^4 + 1$ as:

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

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

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.

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

$z_n(i)$	$c_{short,1,n}(i)$	$c_{short,2,n}(i)$
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:

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

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.

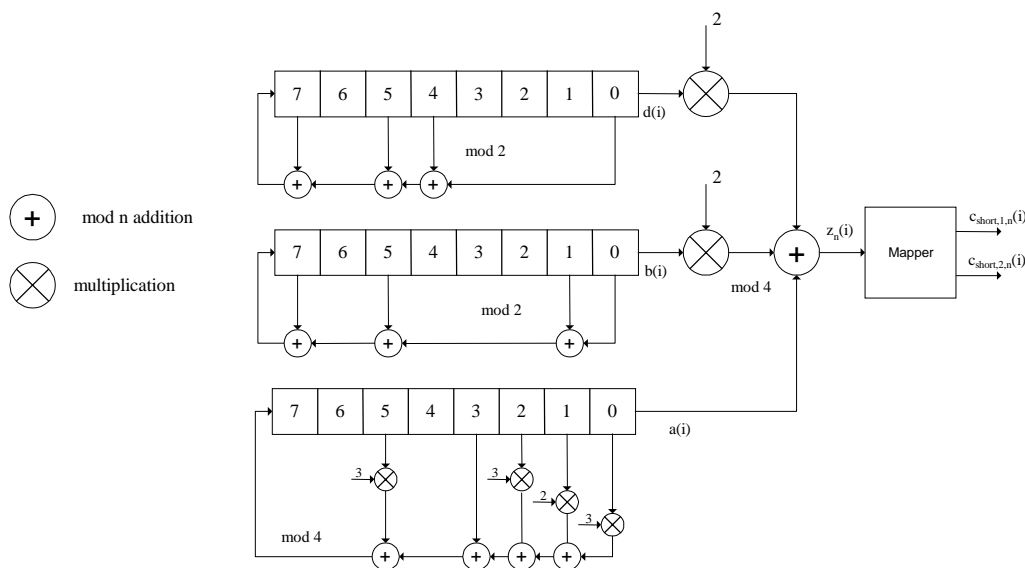


Figure 6: Uplink short scrambling sequence generator for 255 chip sequence

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:

$$S_{dpch,n}(i) = C_{long,n}(i), \quad i = 0, 1, \dots, 38399, \text{ when using long scrambling codes;}$$

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

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

$$S_{dpch,n}(i) = C_{short,n}(i), \quad i = 0, 1, \dots, 38399, \text{ when using short scrambling codes;}$$

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

If a power control preamble is used to initialise 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 8192 different PRACH scrambling codes defined.

The n :th PRACH message part scrambling code, denoted $S_{r\text{-msg},n}$, where $n = 0, 1, \dots, 8191$, is based on the long scrambling sequence and is defined as:

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

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

The set of scrambling codes used for the PCPCH message part are 10 ms long, cell-specific, and each scrambling code has a one-to-one correspondence to the signature sequence and the access sub-channel used by the access preamble part. Both long or short scrambling codes can be used to scramble the PCPCH message part. There are 64 uplink scrambling codes defined per cell and 32768 different PCPCH scrambling codes defined in the system.

The n :th PCPCH message part scrambling code, denoted $S_{c\text{-msg},n}$, where $n = 8192, 8193, \dots, 40959$ is based on the scrambling sequence and is defined as:

In the case when the long scrambling codes are used:

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

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

In the case the short scrambling codes are used:

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

The 32768 PCPCH scrambling codes are divided into 512 groups with 64 codes in each group. There is a one-to-one correspondence between the group of PCPCH preamble scrambling codes in a cell and the primary scrambling code used in the downlink of the cell. The k :th PCPCH scrambling code within the cell with downlink primary scrambling code m , $k = 16, 17, \dots, 79$ and $m = 0, 1, 2, \dots, 511$, is $S_{c\text{-msg},n}$ as defined above with $n = 64 \times m + k + 8176$.

4.3.2.7 PCPCH power control preamble scrambling code

The scrambling code for the PCPCH power control preamble is the same as for the PCPCH message part, as described in section 4.2.3.6 above. 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.

4.3.3 PRACH preamble codes

4.3.3.1 Preamble code construction

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

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

where $k=0$ corresponds to the chip transmitted first in time and $S_{r\text{-pre},n}$ and $C_{\text{sig},s}$ are defined in 4.3.3.2 and 4.3.3.3 below 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 8192 PRACH preamble scrambling codes in total.

The n :th preamble scrambling code, $n = 0, 1, \dots, 8191$, is defined as:

$$S_{r\text{-pre},n}(i) = c_{\text{long},1,n}(i), i = 0, 1, \dots, 4095;$$

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

The 8192 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 , $k = 0, 1, 2, \dots, 15$ and $m = 0, 1, 2, \dots, 511$, 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 $P_s(n)$, $n=0\dots 15$. This is defined as follows:

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

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

4.3.4 PCPCH preamble codes

4.3.4.1 Access preamble

4.3.4.1.1 Access preamble code construction

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

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

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

4.3.4.1.2 Access preamble scrambling code

The scrambling code for the PCPCH preamble part is constructed from the long scrambling sequences. There are 40960 PCPCH access preamble scrambling codes in total.

The n :th PCPCH access preamble scrambling code, where $n = 0, \dots, 40959$ is defined as:

$$- S_{c\text{-acc},n}(i) = c_{\text{long},1,n}(i), i = 0, 1, \dots, 4095;$$

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

The 40960 PCPCH access preamble scrambling codes are divided into 512 groups with 80 codes in each group. There is a one-to-one correspondence between the group of PCPCH access preamble scrambling codes in a cell and the primary scrambling code used in the downlink of the cell. The k :th PCPCH scrambling code within the cell with downlink primary scrambling code m , for $k = 0, \dots, 79$ and $m = 0, 1, 2, \dots, 511$, is $S_{c\text{-acc},n}$ as defined above with $n = 16 \times m + k$ for $k = 0, \dots, 15$ and $n = 64 \times m + (k - 16) + 8192$ for $k = 16, \dots, 79$.

The index $k = 0, \dots, 15$ may only be used as a PCPCH access preamble part scrambling code if the same code is also used for a PRACH.

The index $k = 16, \dots, 79$ correspond to PCPCH access preamble scrambling codes which are not shared together with a PRACH. This leads to 32768 PCPCH specific preamble scrambling codes divided into 512 groups with 64 elements.

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 preamble

4.3.4.2.1 CD preamble code construction

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

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

where $S_{c\text{-cd},n}$ and $C_{\text{sig},s}$ are 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

There are 40960 PCPCH-CD preamble scrambling codes in total.

The n :th PCPCH CD access preamble scrambling code, where $n = 0, \dots, 40959$, is defined as:

$$- S_{c\text{-cd},n}(i) = c_{\text{long},1,n}(i), i = 0, 1, \dots, 4095;$$

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

The 40960 PCPCH scrambling codes are divided into 512 groups with 80 codes in each group. There is a one-to-one correspondence between the group of PCPCH CD preamble scrambling codes in a cell and the primary scrambling code used in the downlink of the cell. The k :th PCPCH scrambling code within the cell with downlink primary scrambling code m , $k = 0, 1, \dots, 79$ and $m = 0, 1, 2, \dots, 511$, is $S_{c\text{-cd},n}$ as defined above with $n = 16 \times m + k$ for $k = 0, \dots, 15$ and $n = 64 \times m + (k - 16) + 8192$ for $k = 16, \dots, 79$.

The index $k = 0, \dots, 15$ may only be used as a PCPCH CD preamble part scrambling code if the same code is also used for a PRACH.

The index $k = 16, \dots, 79$ correspond to PCPCH CD preamble scrambling codes which are not shared together with a PRACH. This leads to 32768 PCPCH specific preamble scrambling codes divided into 512 groups with 64 elements.

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.

4.4 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 7 below:

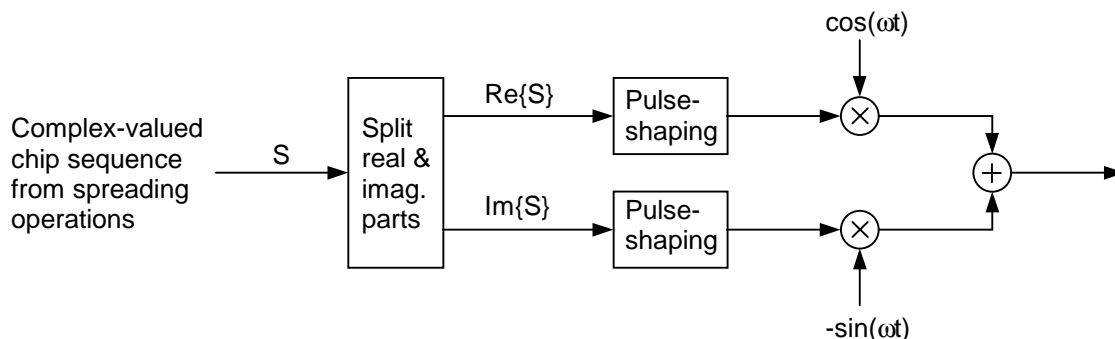


Figure 7: Uplink modulation

The pulse-shaping characteristics are described in [3].

5 Downlink spreading and modulation

5.1 Spreading

Figure 8 illustrates the spreading operation for all downlink physical channels except SCH, i.e. for P-CCPCH, S-CCPCH, CPICH, AICH, 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. For AICH, the symbol values depend on the exact combination of acquisition indicators to be transmitted, compare [2] Section 5.3.3.6.

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. For all channels except AICH, symbol number zero is defined as the first symbol in each frame. For AICH, symbol number zero is defined as the first symbol in each access slot. 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.

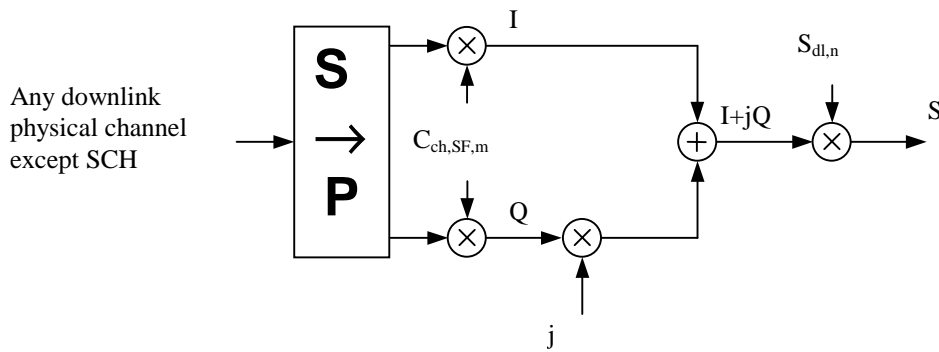


Figure 8: Spreading for all downlink physical channels except SCH

Figure 9 illustrates how different downlink channels are combined. Each complex-valued spread channel, corresponding to point S in Figure 8, is separately weighted by a weight factor G_i . The complex-valued P-SCH and S-SCH, as described in [1], section 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.

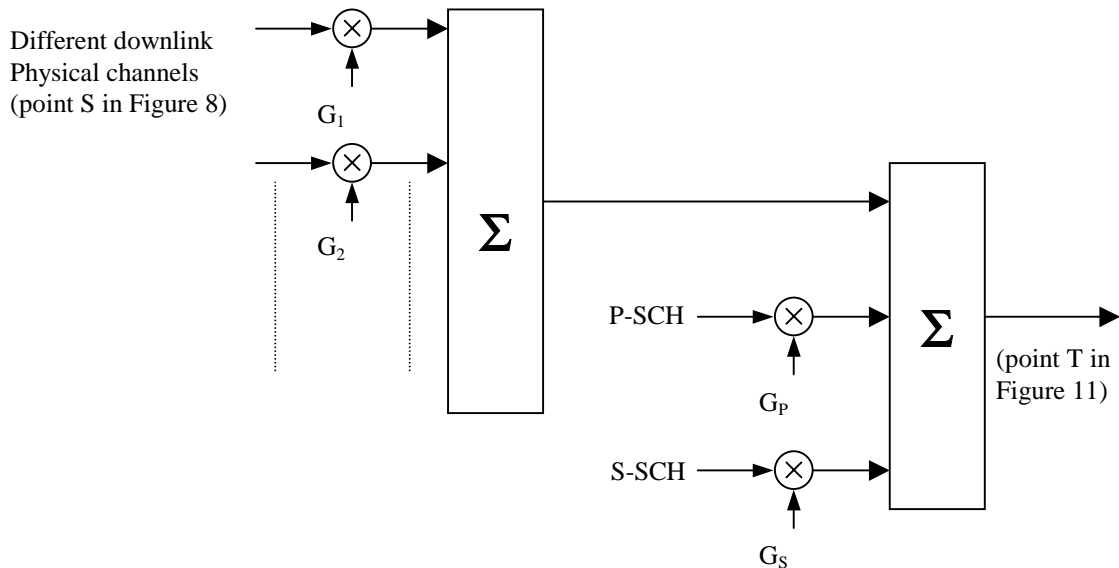


Figure 9: Spreading and modulation for SCH and P-CCPCH

5.2 Code generation and allocation

5.2.1 Channelization codes

The channelization codes of figure 8 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 4 in section 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 $n=0,2,4,\dots,510$, 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 $n=1,3,5,\dots,511$ 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 synchronised 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 OVFSF 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 section 5.2.2);

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

In case the OVFSF code on the PDSCH varies from frame to frame, the OVFSF codes shall be allocated such a way that the OVFSF 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 OVFSF 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 secondary scrambling codes.

Hence, according to the above, scrambling codes $k = 0, 1, \dots, 8191$ 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 $k + 8192$, while the right alternative scrambling code corresponding to scrambling code k is scrambling code number $k + 16384$. 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 $n \geq SF/2$, 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 $16*8*j+16*k$, 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. However, in the case of the CCTrCH of type DSCH then all the PDSCH channelisation 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 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 $1+X^3+X^7+X^{10}+X^{18}$.

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

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

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

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

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

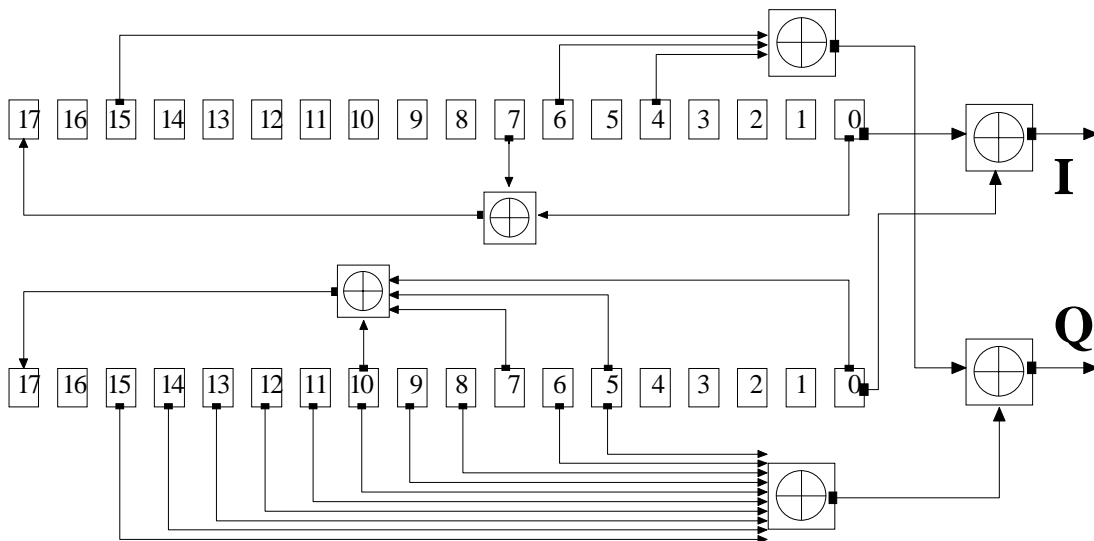


Figure 10: Configuration of downlink scrambling code generator

5.2.3 Synchronisation codes

5.2.3.1 Code generation

The primary synchronisation code (PSC), C_{psc} is constructed as a so-called generalised hierarchical Golay sequence. The PSC is furthermore chosen to have good aperiodic auto correlation properties.

Define:

- $a = \langle x_1, x_2, x_3, \dots, x_{16} \rangle = \langle 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, -1, 1 \rangle$

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:

$$- C_{\text{psc}} = (1 + j) \times \langle a, a, a, -a, -a, a, -a, -a, a, a, a, -a, a, -a, a, a \rangle;$$

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

The 16 secondary synchronization codes (SSCs), $\{C_{\text{ssc},1}, \dots, C_{\text{ssc},16}\}$, 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:

- $z = \langle b, b, b, -b, b, b, -b, -b, b, -b, b, -b, -b, -b, -b \rangle$, where
- $b = \langle x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, -x_9, -x_{10}, -x_{11}, -x_{12}, -x_{13}, -x_{14}, -x_{15}, -x_{16} \rangle$ and $x_1, x_2, \dots, x_{15}, x_{16}$, 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, $n = 0, 1, 2, \dots, 255$, 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 $i = 0, 1, 2, \dots, 255$ and $i = 0$ corresponds to the leftmost symbol.

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

$$- C_{\text{ssc},k} = (1 + j) \times \langle h_m(0) \times z(0), h_m(1) \times z(1), h_m(2) \times z(2), \dots, h_m(255) \times z(255) \rangle;$$

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 $C_{\text{ssc},7}$ 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
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

Scrambling Code Group	slot number														
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
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

In the downlink, the complex-valued chip sequence generated by the spreading process is QPSK modulated as shown in Figure 11 below.

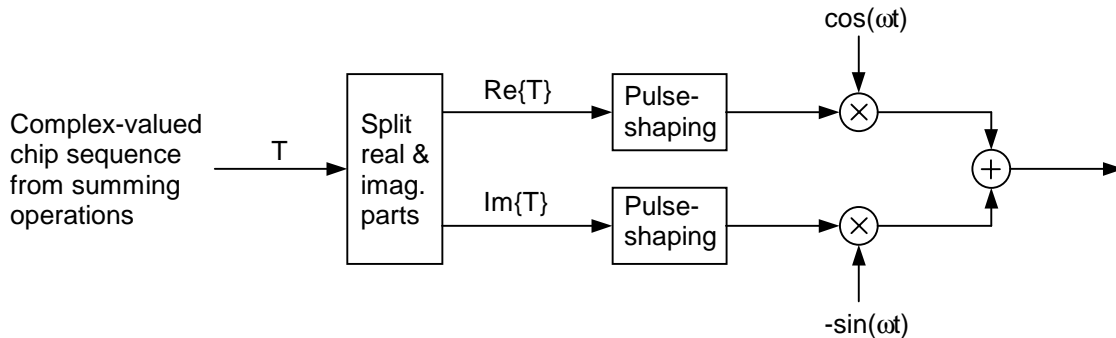


Figure 11: Downlink modulation

The pulse-shaping characteristics are described in [4].

Annex A (informative): Generalised Hierarchical Golay Sequences

A.1 Alternative generation

The generalised hierarchical Golay sequences for the PSC described in 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:

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

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

$$\begin{aligned} a_0(k) &= \delta(k) \text{ and } b_0(k) = \delta(k); \\ a_n(k) &= a_{n-1}(k) + W_n^{(j)} \cdot b_{n-1}(k - D_n^{(j)}); \\ b_n(k) &= a_{n-1}(k) - W_n^{(j)} \cdot b_{n-1}(k - D_n^{(j)}); \\ k &= 0, 1, 2, \dots, 2 * N^{(j)} - 1; \\ n &= 1, 2, \dots, N^{(j)}. \end{aligned}$$

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		
V3.1.1	January 2000	Publication
V3.2.0	March 2000	Publication
V3.3.0	June 2000	Publication