

3GPP TS 25.223 V16.0.0 (2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Radio Access Network;

Spreading and modulation (TDD)

(Release 16)

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganisational Partners and shall not be implemented.   
This Specification is provided for future development work within 3GPPonly. The Organisational Partners accept no liability for any use of this Specification.  
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organisational Partners' Publications Offices.

Keywords

UMTS, radio, modulation

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

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

Internet

http://www.3gpp.org

***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.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [5](#__RefHeading___Toc517804246)

1 Scope [6](#__RefHeading___Toc517804247)

2 References [6](#__RefHeading___Toc517804248)

3 Symbols and abbreviations [6](#__RefHeading___Toc517804249)

3.1 Symbols [6](#__RefHeading___Toc517804250)

3.2 Abbreviations [7](#__RefHeading___Toc517804251)

4 General [7](#__RefHeading___Toc517804252)

5 Data modulation for the 3.84 Mcps and 7.68Mcps options [8](#__RefHeading___Toc517804253)

5.1 Symbol rate [8](#__RefHeading___Toc517804254)

5.2 Mapping of bits onto signal point constellation [8](#__RefHeading___Toc517804255)

5.2.1 Mapping for burst type 1 and 2 [8](#__RefHeading___Toc517804256)

5.2.1.1 QPSK modulation [8](#__RefHeading___Toc517804257)

5.2.1.2 16QAM modulation [8](#__RefHeading___Toc517804258)

5.2.2 Mapping for burst type 3 [11](#__RefHeading___Toc517804259)

5.2.3 Mapping for 3.84 Mcps MBSFN IMB [11](#__RefHeading___Toc517804260)

5.2.3.1 Modulation mapping for data [11](#__RefHeading___Toc517804261)

5.2.3.2 Modulation mapping for TFCI [11](#__RefHeading___Toc517804262)

5A Data modulation for the 1.28 Mcps option [11](#__RefHeading___Toc517804263)

5A.1 Symbol rate [11](#__RefHeading___Toc517804264)

5A.2 Mapping of bits onto signal point constellation [11](#__RefHeading___Toc517804265)

5A.2.1 QPSK modulation [11](#__RefHeading___Toc517804266)

5A.2.2 8PSK modulation [12](#__RefHeading___Toc517804267)

5A.2.3 16QAM modulation [12](#__RefHeading___Toc517804268)

5A.2.4 64QAM modulation [12](#__RefHeading___Toc517804269)

6 Spreading modulation [15](#__RefHeading___Toc517804270)

6.1 Basic spreading parameters [15](#__RefHeading___Toc517804271)

6.2 Channelisation codes [16](#__RefHeading___Toc517804272)

6.3 Channelisation Code Specific Multiplier [16](#__RefHeading___Toc517804273)

6.4 Scrambling codes for the 3.84Mcps and 1.28Mcps options [17](#__RefHeading___Toc517804274)

6.4a Scrambling codes for the 7.68Mcps option [18](#__RefHeading___Toc517804275)

6.4a.1 Generation of binary scrambling codes [18](#__RefHeading___Toc517804276)

6.5 Spread signal of data symbols and data blocks [19](#__RefHeading___Toc517804277)

6.6 Modulation for the 3.84Mcps and 7.68Mcps options [19](#__RefHeading___Toc517804278)

6.6.1 Combination of physical channels in uplink [19](#__RefHeading___Toc517804279)

6.6.1a Physical channel transmission for E-PUCH [21](#__RefHeading___Toc517804280)

6.6.2 Combination of physical channels in downlink [21](#__RefHeading___Toc517804281)

6.6.3 Combination of signature sequences for E-HICH [21](#__RefHeading___Toc517804282)

6.7 Modulation for the 1.28 Mcps option [22](#__RefHeading___Toc517804283)

6.7.1 Combination of physical channels in uplink [23](#__RefHeading___Toc517804284)

6.7.1a Physical channel transmission for E-PUCH [23](#__RefHeading___Toc517804285)

6.7.2 Combination of physical channels in downlink [23](#__RefHeading___Toc517804286)

6.7.3 Combination of signature sequences for Scheduled E-HICH [24](#__RefHeading___Toc517804287)

6.7.3a Combination of signature sequences for Non-Scheduled E-HICH [25](#__RefHeading___Toc517804288)

6.8 Spreading modulation for the 3.84 Mcps MBSFN IMB option [26](#__RefHeading___Toc517804289)

6.8.1 Spreading [26](#__RefHeading___Toc517804290)

6.8.2 Code generation and allocation [27](#__RefHeading___Toc517804291)

6.8.2.1 Channelisation codes [27](#__RefHeading___Toc517804292)

6.8.2.2 Scrambling codes [27](#__RefHeading___Toc517804293)

6.8.3 Modulation [27](#__RefHeading___Toc517804294)

7 Synchronisation codes for the 3.84 Mcps option [28](#__RefHeading___Toc517804295)

7.1 Code Generation [28](#__RefHeading___Toc517804296)

7.2 Code Allocation [28](#__RefHeading___Toc517804297)

7.2.1 Code allocation for Case 1 [30](#__RefHeading___Toc517804298)

7.2.2 Code allocation for Case 2 [31](#__RefHeading___Toc517804299)

7.3 Evaluation of synchronisation codes [32](#__RefHeading___Toc517804300)

7.4 Synchronisation codes for 3.84 Mcps MBSFN IMB [33](#__RefHeading___Toc517804301)

7.4.1 Code generation [33](#__RefHeading___Toc517804302)

7.4.2 Code allocation of SSC [33](#__RefHeading___Toc517804303)

7A Synchronisation codes for the 7.68 Mcps option [33](#__RefHeading___Toc517804304)

7A.1 Code Generation [33](#__RefHeading___Toc517804305)

7A.2 Code Allocation [34](#__RefHeading___Toc517804306)

7A.2.1 Code allocation for Case 1 [36](#__RefHeading___Toc517804307)

7A.2.2 Code allocation for Case 2 [37](#__RefHeading___Toc517804308)

7A.3 Evaluation of synchronisation codes [38](#__RefHeading___Toc517804309)

8 Synchronisation codes for the 1.28 Mcps option [39](#__RefHeading___Toc517804310)

8.1 The downlink pilot channel (DwPCH) [39](#__RefHeading___Toc517804311)

8.1.1 Modulation of the SYNC-DL [39](#__RefHeading___Toc517804312)

8.2 The uplink pilot channel (UpPCH) [39](#__RefHeading___Toc517804313)

8.3 Code Allocation [40](#__RefHeading___Toc517804314)

8.3Aa Code Allocation [40](#__RefHeading___Toc517804315)

9 Cell synchronisation codes [41](#__RefHeading___Toc517804316)

Annex A (normative): Scrambling Codes [43](#__RefHeading___Toc517804317)

Annex AA (normative): Synchronisation sequence [46](#__RefHeading___Toc517804318)

AA.1 Basic SYNC-DL sequence [46](#__RefHeading___Toc517804319)

AA.2 Basic SYNC-UL Codes [47](#__RefHeading___Toc517804320)

Annex B (informative): Generalised Hierarchical Golay Sequences [53](#__RefHeading___Toc517804321)

B.1 Alternative generation [53](#__RefHeading___Toc517804322)

Annex C (informative): Change history [54](#__RefHeading___Toc517804323)

# 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 TDD 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. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 25.201: "Physical layer - general description".

[2] 3GPP TS 25.211: "Physical channels and mapping of transport channels onto physical channels (FDD)".

[3] 3GPP TS 25.212: "Multiplexing and channel coding (FDD)".

[4] 3GPP TS 25.213: "Spreading and modulation (FDD)".

[5] 3GPP TS 25.214: "Physical layer procedures (FDD)".

[6] 3GPP TS 25.215: "Physical layer – Measurements (FDD)".

[7] 3GPP TS 25.221: "Physical channels and mapping of transport channels onto physical channels (TDD)".

[8] 3GPP TS 25.222: "Multiplexing and channel coding (TDD)".

[9] 3GPP TS 25.102: "UTRA (UE) TDD; Radio Transmission and Reception".

[10] 3GPP TS 25.105: "UTRA (BS) TDD; Radio Transmission and Reception".

[11] 3GPP TS25.308: "High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2".

[12] 3GPP TS25.224: "Physical Layer Procedures (TDD)"

[13] 3GPP TS25.321: "Medium Access Control (MAC) protocol specification"

# 3 Symbols and abbreviations

## 3.1 Symbols

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

Cp: PSC

Ci: i:th secondary SCH code

C*CSC, m(k)*: CSC derived as *k*:th offset version from *m*:th applicable constituent Golay complementary pair

## 3.2 Abbreviations

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

16QAM 16 Quadrature Amplitude Modulation

CCTrCH Coded Composite Transport Channel

CDMA Code Division Multiple Access

CSC Cell Synchronisation Code

DPCH Dedicated Physical Channel

FDD Frequency Division Duplex

HS-PDSCH High Speed Physical Downlink Shared Channel

IMB Integrated Mobile Broadcast

MBSFN MBSM over a Single Frequency Network

MIB Master Information Block

MU-MIMO Multi-User Multiple Input Multiple Output

OVSF Orthogonal Variable Spreading Factor

P-CCPCH Primary Common Control Physical Channel

PN Pseudo Noise

PRACH Physical Random Access Channel

PSC Primary Synchronisation Code

QPSK Quadrature Phase Shift Keying

RACH Random Access Channel

SCH Synchronisation Channel

SF Spreading Factor

SFN System Frame Number

TDD Time Division Duplex

TFC Transport Format Combination

UE User Equipment

UL Uplink

# 4 General

In the following, a separation between the data modulation and the spreading modulation has been made. The data modulation for 3.84Mcps TDD (including 3.84 Mcps MBSFN IMB) and 7.68Mcps TDD is defined in clause 5 'Data modulation for the 3.84 Mcps and 7.68Mcps options', the data modulation for 1.28Mcps TDD is defined in clause 5A 'Data modulation for the 1.28 Mcps option' and the spreading modulation in clause 6 'Spreading modulation'.

Table 1 shows the basic modulation parameters for the 7.68Mcps, 3.84Mcps (including 3.84 Mcps MBSFN IMB) and 1.28Mcps TDD options.

Table 1: Basic modulation parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Chip rate | 7.68 Mchip/s | same as FDD basic chiprate: 3.84 Mchip/s and 3.84 Mcps MBSFN IMB | Low chiprate:  1.28 Mchip/s |
| Data modulation | QPSK,16QAM (HS-PDSCH, MBSFN S-CCPCH and E-PUCH only) | QPSK,16QAM (HS-PDSCH, MBSFN S-CCPCH and E-PUCH only) | QPSK, 8PSK,16QAM (HS-PDSCH, E-PUCH, MBSFN S-CCPCH only), 64QAM (HS-PDSCH only) |
| Spreading characteristics | Orthogonal  Q chips/symbol, where Q = 2p, 0 <= p <= 5 | Orthogonal  Q chips/symbol, where Q = 2p, 0 <= p <= 4  (For 3.84 Mcps MBSFN IMB Q = 2p, where p = 4 or 8 only) | Orthogonal  Q chips/symbol, where Q = 2p, 0 <= p <= 4 |

# 5 Data modulation for the 3.84 Mcps and 7.68Mcps options

## 5.1 Symbol rate

The symbol duration TS depends on the spreading factor Q and the chip duration TC: Ts = Q  Tc, where Tc = .

## 5.2 Mapping of bits onto signal point constellation

### 5.2.1 Mapping for burst type 1 and 2

#### 5.2.1.1 QPSK modulation

The data modulation is performed to the bits from the output of the physical channel mapping procedure in [8] and combines always 2 consecutive binary bits to a complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

 (1)

KCode is the number of used codes in a time slot: for 3.84Mcps, max KCode =16; for 7.68Mcps, max KCode =32. Nk is the number of symbols per data field for the code k. This number is linked to the spreading factor Qk [7].

Data block  is transmitted before the midamble and data block  after the midamble. Each of the *Nk* data symbols ; i=1, 2; k=1,...,KCode; n=1,...,Nk; of equation 1 has the symbol duration  as already given.

The data modulation is QPSK, thus the data symbols  are generated from two consecutive data bits from the output of the physical channel mapping procedure in [8]:

 (2)

using the following mapping to complex symbols:

|  |  |
| --- | --- |
| consecutive binary bit pattern | complex symbol |
|  |  |
| 00 | +j |
| 01 | +1 |
| 10 | -1 |
| 11 | -j |

The mapping corresponds to a QPSK modulation of the interleaved and encoded data bits **** of equation 2.

#### 5.2.1.2 16QAM modulation

The data modulation is performed to the bits from the output of the physical channel mapping procedure. In case of 16QAM, modulation 4 consecutive binary bits are represented by one complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

 (2b)

Nk is the number of symbols per data field for the user k. This number is linked to the spreading factor Qk.

Data block  is transmitted before the midamble and data block  after the midamble. Each of the Nk data symbols ; i=1, 2; k=1,...,K; n=1,...,Nk; of equation 2b has the symbol duration  as already given.

The data modulation is 16QAM, thus the data symbols  are generated from 4 consecutive data bits from the output of the physical channel mapping procedure in [8]:

(2c)

using the following mapping to complex symbols:

|  |  |
| --- | --- |
| Consecutive binary bit pattern | complex symbol |
|  |  |
| 0000 |  |
| 0001 |  |
| 0010 |  |
| 0011 |  |
| 0100 |  |
| 0101 |  |
| 0110 |  |
| 0111 |  |
| 1000 |  |
| 1001 |  |
| 1010 |  |
| 1011 |  |
| 1100 |  |
| 1101 |  |
| 1110 |  |
| 1111 |  |

The mapping corresponds to a 16QAM modulation of the interleaved and encoded data bits  of the table above and  of equation 2b.

### 5.2.2 Mapping for burst type 3

In case of burst type 3, the definitions in subclause 5.2.1.1 and subclause 5.2.1.2 apply with a modified number of symbols in the second data block. For the burst type 3, the number of symbols in the second data block is decreased by  symbols for 3.84Mcps TDD and is decreased by symbols for 7.68Mcps TDD.

### 5.2.3 Mapping for 3.84 Mcps MBSFN IMB

#### 5.2.3.1 Modulation mapping for data

Mapping of data bits onto a QPSK or 16-QAM signal point constellation shall be accomplished as described in subclause 5.1.1.1 or 5.1.1.2 of [4] respectively.

#### 5.2.3.2 Modulation mapping for TFCI

In the case of S-CCPCH frame type 1 and S-CCPCH frame type 2 using QPSK modulation for data, TFCI bits shall be QPSK modulated according to subclause 5.1.1.1 of [4].

In the case of S-CCPCH frame type 2 using 16-QAM modulation for data, each consecutive pair of binary-valued TFCI bits {b2q, b2q+1}, with q = {0,1,2,…} shall be mapped according to the rotated QPSK constellation given by the following table.

|  |  |  |
| --- | --- | --- |
| **{b2q , b2q+1}** | **I branch** | **Q branch** |
| {0,0} | 0.4472 | 1.3416 |
| {0,1} | 1.3416 | -0.4472 |
| {1,0} | -1.3416 | 0.4472 |
| {1,1} | -0.4472 | -1.3416 |

# 5A Data modulation for the 1.28 Mcps option

## 5A.1 Symbol rate

The symbol duration TS depends on the spreading factor Q and the chip duration TC: Ts = Q  Tc, where Tc = .

## 5A.2 Mapping of bits onto signal point constellation

### 5A.2.1 QPSK modulation

The mapping of bits onto the signal point constellation for QPSK modulation is the same as in the 3.84Mcps TDD cf. [5.2.1.1 QPSK modulation].

### 5A.2.2 8PSK modulation

The data modulation is performed to the bits from the output of the physical channel mapping procedure. In case of 8PSK modulation 3 consecutive binary bits are represented by one complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

 (1a)

Nk is the number of symbols per data field for the code k. This number is linked to the spreading factor Qk.

Data block  is transmitted before the midamble and data block  after the midamble. Each of the Nk data symbols ; i=1, 2; k=1,...,KCode; n=1,...,Nk; of equation 1 has the symbol duration  as already given.

The data modulation is 8PSK, thus the data symbols  are generated from 3 consecutive data bits from the output of the physical channel mapping procedure in [8]:

 (2a)

using the following mapping to complex symbols:

|  |  |
| --- | --- |
| Consecutive binary bit pattern | complex symbol |
|  |  |
| 000 | cos(11pi/8)+ jsin(11pi/8) |
| 001 | cos(9pi/8)+ jsin(9pi/8) |
| 010 | cos(5pi/8)+ jsin(5pi/8) |
| 011 | cos(7pi/8)+ jsin(7pi/8) |
| 100 | cos(13pi/8)+ jsin(13pi/8) |
| 101 | cos(15pi/8)+ jsin(15pi/8) |
| 110 | cos(3pi/8)+ jsin(3pi/8) |
| 111 | cos(pi/8)+ jsin(pi/8) |

The mapping corresponds to a 8PSK modulation of the interleaved and encoded data bits  of the table above and  of equation 1a.

### 5A.2.3 16QAM modulation

The mapping of bits onto the signal point constellation for 16QAM modulation is the same as in the 3.84Mcps TDD cf. [5.2.1.2 16QAM modulation].

### 5A.2.4 64QAM modulation

The data modulation is performed to the bits from the output of the physical channel mapping procedure. In case of 64QAM, modulation 6 consecutive binary bits are represented by one complex valued data symbol. Each user burst has two data carrying parts, termed data blocks:

 (1c)

Nk is the number of symbols per data field for the user k. This number is linked to the spreading factor Qk.

Data block  is transmitted before the midamble and data block  after the midamble. Each of the Nk data symbols ; i=1, 2; k=1,...,K; n=1,...,Nk; of equation 1c has the symbol duration  as already given.

The data modulation is 64QAM, thus the data symbols  are generated from 6 consecutive data bits from the output of the physical channel mapping procedure in [8]:

 (2c)

using the following mapping to complex symbols:

|  |  |  |  |
| --- | --- | --- | --- |
| Consecutive binary bit pattern | complex symbol | Consecutive binary bit pattern | complex symbol |
|  |  |  |  |
| 000000 |  | 100000 |  |
| 000001 |  | 100001 |  |
| 000010 |  | 100010 |  |
| 000011 |  | 100011 |  |
| 000100 |  | 100100 |  |
| 000101 |  | 100101 |  |
| 000110 |  | 100110 |  |
| 000111 |  | 100111 |  |
| 001000 |  | 101000 |  |
| 001001 |  | 101001 |  |
| 001010 |  | 101010 |  |
| 001011 |  | 101011 |  |
| 001100 |  | 101100 |  |
| 001101 |  | 101101 |  |
| 001110 |  | 101110 |  |
| 001111 |  | 101111 |  |
| 010000 |  | 110000 |  |
| 010001 |  | 110001 |  |
| 010010 |  | 110010 |  |
| 010011 |  | 110011 |  |
| 010100 |  | 110100 |  |
| 010101 |  | 110101 |  |
| 010110 |  | 110110 |  |
| 010111 |  | 110111 |  |
| 011000 |  | 111000 |  |
| 011001 |  | 111001 |  |
| 011010 |  | 111010 |  |
| 011011 |  | 111011 |  |
| 011100 |  | 111100 |  |
| 011101 |  | 111101 |  |
| 011110 |  | 111110 |  |
| 011111 |  | 111111 |  |

The mapping corresponds to a 64QAM modulation of the interleaved and encoded data bits  of the table above and  of equation 2c.

# 6 Spreading modulation

Sub-clauses 6.1 to 6.7 do not apply to 3.84 Mcps MBSFN IMB. Spreading modulation for 3.84 Mcps MBSFN IMB is described in clause 6.8.

## 6.1 Basic spreading parameters

Spreading of data consists of two operations: Channelisation and Scrambling. Firstly, each complex valued data symbol  of equation 1 (or of equation 8 in the case of E-HICH) is spread with a real valued channelisation code of length *Qk*: for 3.84Mcps TDD and 1.28Mcps TDD, ; for 7.68Mcps TDD, . The resulting sequence is then scrambled by a complex sequence : the sequence is of length 16 for the 3.84Mcps and 1.28Mcps options; it is of length 32 for the 7.68Mcps option.

## 6.2 Channelisation codes

The elements ; k=1,...,KCode; q=1,...,Qk; of the real valued channelisation codes

; k=1,...,KCode;

shall be taken from the set

**** (3)

The are Orthogonal Variable Spreading Factor (OVSF) codes, allowing to mix in the same timeslot channels with different spreading factors while preserving the orthogonality. The OVSF codes can be defined using the code tree of figure 1.



Figure 1: Code-tree for generation of Orthogonal Variable Spreading Factor (OVSF)  
codes for Channelisation Operation

Each level in the code tree defines a spreading factor indicated by the value of Q in the figure. All codes within the code tree cannot be used simultaneously in a given timeslot. A code can be used in a timeslot if and only if no other code on the path from the specific code to the root of the tree or in the sub-tree below the specific code is used in this timeslot. This means that the number of available codes in a slot is not fixed but depends on the rate and spreading factor of each physical channel.

For the 3.84Mcps and 1.28Mcps TDD options, the spreading factor goes up to QMAX=16; for the 7.68Mcps TDD option, the spreading factor goes up to QMAX=32.

## 6.3 Channelisation Code Specific Multiplier

Associated with each channelisation code is a multiplier  taking values from the set , where is a permutation of the integer set {0, ..., *Qk* -1} and *Qk* denotes the spreading factor. The multiplier is applied to the data sequence modulating each channelisation code. The values of the multiplier for each channelisation code are given in the table below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| k |  |  |  |  |  |  |
| 1 | 1 | 1 | -j | 1 | -1 | -j |
| 2 |  | +j | 1 | +j | -j | -1 |
| 3 |  |  | +j | +j | 1 | -1 |
| 4 |  |  | -1 | -1 | 1 | 1 |
| 5 |  |  |  | -j | +j | -1 |
| 6 |  |  |  | -1 | -1 | -j |
| 7 |  |  |  | -j | -1 | j |
| 8 |  |  |  | 1 | 1 | 1 |
| 9 |  |  |  |  | -j | -1 |
| 10 |  |  |  |  | +j | 1 |
| 11 |  |  |  |  | 1 | 1 |
| 12 |  |  |  |  | +j | -j |
| 13 |  |  |  |  | -j | j |
| 14 |  |  |  |  | -j | -1 |
| 15 |  |  |  |  | +j | j |
| 16 |  |  |  |  | -1 | -j |
| 17 |  |  |  |  |  | -j |
| 18 |  |  |  |  |  | -j |
| 19 |  |  |  |  |  | 1 |
| 20 |  |  |  |  |  | j |
| 21 |  |  |  |  |  | -1 |
| 22 |  |  |  |  |  | -j |
| 23 |  |  |  |  |  | -j |
| 24 |  |  |  |  |  | -j |
| 25 |  |  |  |  |  | -1 |
| 26 |  |  |  |  |  | -1 |
| 27 |  |  |  |  |  | j |
| 28 |  |  |  |  |  | -1 |
| 29 |  |  |  |  |  | -j |
| 30 |  |  |  |  |  | 1 |
| 31 |  |  |  |  |  | -1 |
| 32 |  |  |  |  |  | -1 |

NOTE: the multiplier may only be applied in the 7.68Mcps TDD option.

If the UE autonomously changes the SF, as described in [7], it shall always use the multiplier associated with the channelisation code allocated by higher layers.

## 6.4 Scrambling codes for the 3.84Mcps and 1.28Mcps options

The spreading of data by a real valued channelisation code  of length Qk is followed by a cell specific complex scrambling sequence . The elements  of the complex valued scrambling codes shall be taken from the complex set

 (4)

In equation 4 the letter j denotes the imaginary unit. A complex scrambling code  is generated from the binary scrambling codes  of length 16 shown in Annex A. The relation between the elements and  is given by:

 (5)

Hence, the elements  of the complex scrambling code  are alternating real and imaginary.

The length matching is obtained by concatenating QMAX/Qk spread words before the scrambling. The scheme is illustrated in figure 2 and is described in more detail in subclause 6.5.



Figure 2: Spreading of data symbols

## 6.4a Scrambling codes for the 7.68Mcps option

The spreading of data by a real valued channelisation code  of length Qk is followed by a cell specific complex scrambling sequence . The elements of the complex valued scrambling codes shall be taken from the complex set

 (4a)

In equation 4a the letter j denotes the imaginary unit. A complex scrambling code  is generated from the binary scrambling codes of length 32 that are generated according to the method described in section 6.4a.1. The relation between the elements and  is given by:

(5a)

Hence, the elements  of the complex scrambling code  are alternating real and imaginary.

The length matching is obtained by concatenating QMAX/Qk spread words before the scrambling. The scheme is illustrated in figure 2 and is described in more detail in subclause 6.5.

### 6.4a.1 Generation of binary scrambling codes

The binary scrambling code, , for cell parameter *n* in the 7.68Mcps TDD option is formed from the concatenation of the binary scrambling codes and shown in Annex A:

## 6.5 Spread signal of data symbols and data blocks

The combination of the user specific channelisation and cell specific scrambling codes can be seen as a user and cell specific spreading code  with

, k=1,…,KCode, p=1,…,NkQk.

With the root raised cosine chip impulse filter Cr0(t) the transmitted signal belonging to the data block  of equation 1 transmitted before the midamble is

 (6)

and for the data block of equation 1 transmitted after the midamble

 (7)

where *Lm* is the number of midamble chips*.*

## 6.6 Modulation for the 3.84Mcps and 7.68Mcps options

The complex-valued chip sequence is modulated as shown in figure 3.



Figure 3: Modulation of complex valued chip sequences

The pulse-shaping characteristics are described in [9] and [10].

### 6.6.1 Combination of physical channels in uplink

Figure 4 illustrates the principle of combination of two different physical uplink channels within one timeslot. In the case of E-PUCH, only a single uplink physical channel is transmitted per timeslot and the procedures of subclause 6.6.1a shall instead apply).

The DPCHs to be combined belong to same CCTrCH, did undergo spreading as described in sections before and are thus represented by complex-valued sequences. First, the amplitude of all DPCHs is adjusted according to UL open loop power control as described in [10]. Each DPCH is then separately weighted by a weight factor i and combined using complex addition. After combination of Physical Channels the gain factor j is applied, depending on the actual TFC as described in [10].

In case of different CCTrCH, principle shown in Figure 4 applies to each CCTrCH separately.



Figure 4: Combination of different physical channels in uplink

The values of weight factors i are depending on the spreading factor SF of the corresponding DPCH:

|  |  |
| --- | --- |
| SF of DPCHi | i |
| 32 |  |
| 16 |  |
| 8 |  |
| 4 |  |
| 2 |  |
| 1 |  |

NOTE: in the above table, SF = 32 is only supported in the 7.68Mcps TDD option.

In the case that j (corresponding to the *j*–th TFC) has been explicitly signalled to the UE, the possible values that j can assume are listed in the table below. In the case that j has been calculated by the UE from a reference TFC, j shall not be restricted to the quantised values.

|  |  |
| --- | --- |
| Signalling value for j | Quantized value j |
| 15 | 16/8 |
| 14 | 15/8 |
| 13 | 14/8 |
| 12 | 13/8 |
| 11 | 12/8 |
| 10 | 11/8 |
| 9 | 10/8 |
| 8 | 9/8 |
| 7 | 8/8 |
| 6 | 7/8 |
| 5 | 6/8 |
| 4 | 5/8 |
| 3 | 4/8 |
| 2 | 3/8 |
| 1 | 2/8 |
| 0 | 1/8 |

### 6.6.1a Physical channel transmission for E-PUCH

Figure 4a illustrates the principle of E-PUCH transmission. In a timeslot in which an E-PUCH is transmitted by a UE, no other physical channels may be transmitted by the same UE.

The amplitude of the E-PUCH is adjusted in accordance with the E-PUCH UL power control procedure described in [12]. The power setting procedure of [12] includes appropriate power adjustment factors for the E-PUCH spreading factor and for the E-TFC selected by higher layers [13]. Quantisation of the gain factor used to set the E-PUCH power is not specified.



Figure 4a: Combination of different physical channels in uplink

### 6.6.2 Combination of physical channels in downlink

Figure 5 illustrates how different physical downlink channels are combined within one timeslot. Each complex-valued spread channel is separately weighted by a weight factor Gi. If a timeslot contains the SCH, the complex-valued SCH, as described in [7] is separately weighted by a weight factor GSCH. All downlink physical channels are then combined using complex addition.



Figure 5: Combination of different physical channels in downlink in case of SCH timeslot

### 6.6.3 Combination of signature sequences for E-HICH

Multiple HARQ acknowledgement indicator signature sequences may be mapped onto the same channelisation code. Each signature sequence (described in [8]) is first subjected to QPSK modulation as described in subclause 5.2.1.1 to form the output sequence for the *hth* indicator sequence, where *n=1,2,…,Nk* and *i=1,2*. Code k is the same value for all signature sequences mapped to the same channelisation code.

When multiple signature sequences are to be transmitted on the same channelisation code, the following procedure shall be applied prior to spreading.

Each QPSK-modulated stream is amplitude-weighted by a factor *gh* according to the desired signature sequence power. A summation is then performed across all *H* signature sequences mapped to the same channelisation code as shown in figure 5a. The output of the summation block is the sequence:

(*n = 1,2,…,Nk*) and (*i=1,2*) (8)



Figure 5a: Combination of HARQ acknowledgement indicator sequences prior to spreading

The sequence is mapped to a single channelisation code and subject to spreading at SF=16 (for 3.84Mcps) and at SF=32 (for 7.68Mcps) in accordance with the general method of subclause 6.

## 6.7 Modulation for the 1.28 Mcps option

The complex-valued chip sequence in uplink or downlink on one carrier within one timeslot is modulated as shown in figure 6.



Figure 6: Modulation of complex valued chip sequences

The pulse-shaping characteristics are described in [9] and [10].

### 6.7.1 Combination of physical channels in uplink

The principle of combination of two different physical uplink channels within one timeslot is the same as in the 3.84 Mcps TDD cf. [6.6.1 Combination of physical channels in uplink] In the case of E-PUCH, the procedures of subclause 6.7.1a shall instead apply).

### 6.7.1a Physical channel transmission for E-PUCH

Figure 6a illustrates the principle of E-PUCH transmission when one uplink physical channel is transmitted.

The amplitude of the E-PUCH is adjusted in accordance with the E-PUCH UL power control procedure described in [12]. The power setting procedure of [12] includes appropriate power adjustment factors for the E-PUCH spreading factor and for the E-TFC selected by higher layers [13]. Quantisation of the gain factor used to set the E-PUCH power is not specified.

****

Figure 6a: Combination of different physical channels in uplink

### 6.7.2 Combination of physical channels in downlink

Figure 7 illustrates how different physical downlink channels are combined within one timeslot. Each spread channel is separately weighted by a weight factor Gi.. All downlink physical channels are then combined using complex addition.



Figure 7: Combination of different physical channels in downlink

### 6.7.3 Combination of signature sequences for Scheduled E-HICH

For Scheduled E-HICH, every scheduled user is assigned one signature sequence which is related to the E-DCH resources allocated by Node-B to indicate ACK/NACK. But for the user configured in MU-MIMO mode by higher layers, in case the special default midamble allocation scheme is taken, the signature sequence allocated to the user is related to both the E-DCH resources allocated by Node-B and the variable "offset" which is determined by the special default midamble pattern indicator [7] signalled on E-AGCH. Multiple users' HARQ acknowledgement indicator signature sequences may be mapped onto the same channelisation code. Each signature sequence (described in [8]) is first subjected to QPSK modulation as described in subclause 5.2.1.1 to form the output sequence for the *hth* indicator sequence, where *n=1,2,…,Nk* and *i=1,2*. Code k is the same value for all signature sequences mapped to the same channelisation code.

When multiple signature sequences are to be transmitted on the same channelisation code, the following procedure shall be applied prior to spreading.

Each QPSK-modulated stream is amplitude-weighted by a factor *gh* according to the desired signature sequence power. Each E-HICH physical channel may carry ACK/NACK signature sequence(s) for one UE or multiple UEs decided by Node-B. A summation is then performed across *M* signature sequences mapped to the same channelisation code as shown in figure 8. The output of the summation block is the sequence:

(*n = 1,2,…,Nk*) and (*i=1,2*) (9)



Figure 8: Combination of HARQ acknowledgement indicator sequences prior to spreading for Scheduled E-HICH

The sequence is mapped to a single channelisation code and subject to spreading at SF=16 in accordance with the general method of subclause 6.

### 6.7.3a Combination of signature sequences for Non-Scheduled E-HICH

For Non-Scheduled E-HICH, the 80 signature sequences are divided into 20 groups while each group includes 4 sequences. Every non-scheduled user is assigned one group by higher layer, from that two sequences are selected to indicate ACK/NACK and TPC/SS command. Multiple users' signature sequences may be mapped onto the same channelisation code. Each user's two signature sequences (described in [8]) are first subjected to QPSK modulation as described in subclause 5.2.1.1 to form the two output sequences and for the *hth* user, where *n=1,2,…,Nk* and *i=1,2*. Code k is the same value for all signature sequences mapped to the same channelisation code.

When multiple users' signature sequences are to be transmitted on the same channelisation code, the following procedure shall be applied prior to spreading.

Firstly, each user's QPSK-modulated stream corresponding to TPC/SS signature sequence is amplitude-weighted by a factor f*h* and added to the QPSK-modulated stream corresponding to ACK/NACK signature sequence; Secondly, each user's combined stream is amplitude-weighted by a factor *gh* according to the desired user power. A summation is then performed across *M* users' signature sequences mapped to the same channelisation code as shown in figure 8a. The output of the summation block is the sequence:

(*n = 1,2,…,Nk*) and (*i=1,2*) (9a)



Figure 8a: Combination of ACK/NACK and TPC/SS sequences prior to spreading for Non-Scheduled E-HICH

The sequence is mapped to a single channelisation code and subject to spreading at SF=16 in accordance with the general method of subclause 6.

## 6.8 Spreading modulation for the 3.84 Mcps MBSFN IMB option

### 6.8.1 Spreading

The spreading operation includes a modulation mapper stage successively followed by a channelisation stage, an IQ combining stage and a scrambling stage as illustrated by figure 9.

Modulation mapping is described in subclause 5.2.3.

For all physical channels, except for the Synchronisation Channel (SCH), the I and Q branches shall be spread to the chip rate by the same real-valued channelisation code Cch,SF,m, i.e. the output for each input symbol on the I and the Q branches shall be a sequence of SF chips corresponding to the channelisation code chip sequence multiplied by the real-valued symbol. The channelisation code sequence shall be aligned in time with the symbol boundary. The real-valued chip sequence on the Q-branch shall be complex multiplied with j and summed with the corresponding real-valued chip sequence on the I-branch, resulting in a single complex-valued chip sequence I+jQ.

The sequence of complex-valued chips output from the spreading stage shall be scrambled (complex chip-wise multiplication) by a complex-valued scrambling code Sdl,n.



Figure 9: Spreading for all downlink physical channels except SCH

All complex-valued spread channels are separately weighted and then combined, together with separately weighted Primary SCH and Secondary SCH, into one complex-valued chip sequence by using complex addition, as illustrated by figure 9 in subclause 5.1.5 of [4]. The resulting signal is modulated prior to transmission as described in subclause 6A.3.

### 6.8.2 Code generation and allocation

#### 6.8.2.1 Channelisation codes

The channelisation codes are OVFS codes that preserve the orthogonality between downlink channels of different rates and spreading factors. The channelisation codes are defined in figure 4 of subclause 4.3.1.1 of [3] and are uniquely described as Cch,SF,m, where SF is the spreading factor of the code and *m* is the code number, 0  *m*  SF-1.

The following applies to the MBSFN IMB physical channels:

- The channelisation code for the Primary CPICH is fixed to Cch,256,0 ;

- The channelisation code for the Primary CCPCH is fixed to Cch,256,1 ;

- The channelisation codes for the Secondary CCPCH frame type 1 and MICH are assigned by UTRAN from the codes Cch,256,m ;

- The channelisation codes for the Secondary CCPCH frame type 2 are assigned by UTRAN from the codes Cch,16,m ;

- The channelisation codes for the T-CPICH are Cch,16,1 , Cch,16,2 , …, Cch,16,15.

#### 6.8.2.2 Scrambling codes

The scrambling codes shall be generated as described in subclause 5.2.2 in [4]. For MBSFN IMB operation, only primary scrambling codes shall be used. Out of all possible primary scrambling codes with index n=16\*i where i=0…511 as defined in [4] the following subset shall be supported for the MBSFN option: . No two members of set *n* belong to the same scrambling code group.

Cells that belong to a certain MBSFN IMB cluster shall use the same primary scrambling code.

The primary scrambling code for all physical channels shall be applied aligned with the start of the Primary CCPCH frame. This also applies in the case of a Secondary CCPCH frame type 2 associated with the *k*th sub-frame of a radio frame (*k* = 0,1,…4) [7], such that the start of the scrambling code is always aligned with the start of sub-frame *k* = 0.

### 6.8.3 Modulation

Modulation of the complex-valued chip sequence generated by the spreading process is performed according to sub-clause 6.6. The modulation chip rate is 3.84 Mcps.

# 7 Synchronisation codes for the 3.84 Mcps option

Sub-clauses 7.1, 7.2 and 7.3 do not apply for 3.84 Mcps MBSFN IMB operation. Synchronisation codes for 3.84 Mcps MBSFN IMB are described in sub-clause 7.4.

## 7.1 Code Generation

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

Define a = < x1, x2, x3, …, x16 > = < 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, Cp , is defined as Cp = < y(0),y(1),y(2),...,y(255) >

where

and the left most index corresponds to the chip transmitted first in time.

The 12 secondary synchronization codes, {C0, C1, C3, C4, C5, C6, C8, C10, C12, C13, C14,C15 } are complex valued with identical real and imaginary components, and are constructed from the position wise multiplication of a Hadamard sequence and a sequence z, defined as

z =  , where

b = 

and x1, x2, x3, …, x16 are the 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:



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

Denote the *n*:th Hadamard sequence *hn* as a row of *H8* numbered from the top, n = 0, 1, 2, …, 255,in the sequel.

Furthermore, let *hm(l)* and *z(l)* denote the *l*th symbol of the sequence *hm* and *z,* respectively where *l* = 0, 1, 2, …, 255 and *l* = 0 corresponds to the leftmost symbol.

The i:th secondary SCH code word, Ci, i = 0, 1, 3, 4, 5, 6, 8, 10, 12, 13, 14, 15 is then defined as

Ci = (1 + *j*)  <*hm*(0)  *z*(0), *hm*(1)  *z*(1), *hm*(2)  *z*(2), …, *hm*(255)  *z*(255)>,

where *m* = (16i) and the leftmost chip in the sequence corresponds to the chip transmitted first in time.

## 7.2 Code Allocation

Three secondary SCH codes are QPSK modulated and transmitted in parallel with the primary synchronization code. The QPSK modulation carries the following information:

- the code group that the base station belongs to (32 code groups:5 bits; Cases 1, 2);

- the position of the frame within an interleaving period of 20 msec (2 frames:1 bit, Cases 1, 2);

- the position of the SCH slot(s) within the frame (2 SCH slots:1 bit, Case 2).

The modulated secondary SCH codes are also constructed such that their cyclic-shifts are unique, i.e. a non-zero cyclic shift less than 2 (Case 1) and 4 (Case 2) of any of the sequences is not equivalent to some cyclic shift of any other of the sequences. Also, a non-zero cyclic shift less than 2 (Case 1) and 4 (Case 2) of any of the sequences is not equivalent to itself with any other cyclic shift less than 8. The secondary synchronization codes are partitioned into two code sets for Case 1 and four code sets for Case 2. The set is used to provide the following information:

Case 1:

Table 2: Code Set Allocation for Case 1

|  |  |
| --- | --- |
| Code Set | Code Group |
| 1 | 0-15 |
| 2 | 16-31 |

The code group and frame position information is provided by modulating the secondary codes in the code set.

Case 2:

Table 3: Code Set Allocation for Case 2

|  |  |
| --- | --- |
| Code Set | Code Group |
| 1 | 0-7 |
| 2 | 8-15 |
| 3 | 16-23 |
| 4 | 24-31 |

The slot timing and frame position information is provided by the comma free property of the code word and the Code group is provided by modulating some of the secondary codes in the code set.

The following SCH codes are allocated for each code set:

Case 1

Code set 1: C1, C3, C5.

Code set 2: C10, C13, C14.

Case 2

Code set 1: C1, C3, C5.

Code set 2: C10, C13, C14.

Code set 3: C0, C6, C12.

Code set 4: C4, C8, C15.

The following subclauses 7.2.1 to 7.2.2 refer to the two cases of SCH/P-CCPCH usage as described in [7].

Note that in the tables 4 and 5 corresponding to Cases 1 and 2, respectively, Frame 1 implies the frame with an odd SFN and Frame 2 implies the frame with an even SFN.

### 7.2.1 Code allocation for Case 1

Table 4: Code Allocation for Case 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code Group | Code Set | Frame 1 | | | Frame 2 | | | Associated toffset |
| 0 | 1 | C1 | C3 | C5 | C1 | C3 | -C5 | t0 |
| 1 | 1 | C1 | -C3 | C5 | C1 | -C3 | -C5 | t1 |
| 2 | 1 | -C1 | C3 | C5 | -C1 | C3 | -C5 | t2 |
| 3 | 1 | -C1 | -C3 | C5 | -C1 | -C3 | -C5 | t3 |
| 4 | 1 | jC1 | jC3 | C5 | jC1 | jC3 | -C5 | t4 |
| 5 | 1 | jC1 | -jC3 | C5 | jC1 | -jC3 | -C5 | t5 |
| 6 | 1 | -jC1 | jC3 | C5 | -jC1 | jC3 | -C5 | t6 |
| 7 | 1 | -jC1 | -jC3 | C5 | -jC1 | -jC3 | -C5 | t7 |
| 8 | 1 | jC1 | jC5 | C3 | jC1 | jC5 | -C3 | t8 |
| 9 | 1 | jC1 | -jC5 | C3 | jC1 | -jC5 | -C3 | t9 |
| 10 | 1 | -jC1 | jC5 | C3 | -jC1 | jC5 | -C3 | t10 |
| 11 | 1 | -jC1 | -jC5 | C3 | -jC1 | -jC5 | -C3 | t11 |
| 12 | 1 | jC3 | jC5 | C1 | jC3 | jC5 | -C1 | t12 |
| 13 | 1 | jC3 | -jC5 | C1 | jC3 | -jC5 | -C1 | t13 |
| 14 | 1 | -jC3 | jC5 | C1 | -jC3 | jC5 | -C1 | t14 |
| 15 | 1 | -jC3 | -jC5 | C1 | -jC3 | -jC5 | -C1 | t15 |
| 16 | 2 | C10 | C13 | C14 | C10 | C13 | -C14 | t16 |
| 17 | 2 | C10 | -C13 | C14 | C10 | -C13 | -C14 | t17 |
|  |  |  |  |  |  |  |  |  |
| 20 | 2 | jC10 | jC13 | C14 | jC10 | jC13 | -C14 | t20 |
|  |  |  |  |  |  |  |  |  |
| 24 | 2 | jC10 | jC14 | C13 | jC10 | jC14 | -C13 | t24 |
|  |  |  |  |  |  |  |  |  |
| 31 | 2 | -jC13 | -jC14 | C10 | -jC13 | -jC14 | -C10 | t31 |

NOTE: The code construction for code groups 0 to 15 using only the SCH codes from code set 1 is shown. The construction for code groups 16 to 31 using the SCH codes from code set 2 is done in the same way.

### 7.2.2 Code allocation for Case 2

Table 5: Code Allocation for Case 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code Group | Code Set | Frame 1 | | | | | | Frame 2 | | | | | | Associated toffset |
| Slot k | | | Slot k+8 | | | Slot k | | | Slot k+8 | | |
| 0 | 1 | C1 | C3 | C5 | C1 | C3 | -C5 | -C1 | -C3 | C5 | -C1 | -C3 | -C5 | t0 |
| 1 | 1 | C1 | -C3 | C5 | C1 | -C3 | -C5 | -C1 | C3 | C5 | -C1 | C3 | -C5 | t1 |
| 2 | 1 | jC1 | jC3 | C5 | jC1 | jC3 | -C5 | -jC1 | -jC3 | C5 | -jC1 | -jC3 | -C5 | t2 |
| 3 | 1 | jC1 | -jC3 | C5 | jC1 | -jC3 | -C5 | -jC1 | jC3 | C5 | -jC1 | jC3 | -C5 | t3 |
| 4 | 1 | jC1 | jC5 | C3 | jC1 | jC5 | -C3 | -jC1 | -jC5 | C3 | -jC1 | -jC5 | -C3 | t4 |
| 5 | 1 | jC1 | -jC5 | C3 | jC1 | -jC5 | -C3 | -jC1 | jC5 | C3 | -jC1 | jC5 | -C3 | t5 |
| 6 | 1 | jC3 | jC5 | C1 | jC3 | jC5 | -C1 | -jC3 | -jC5 | C1 | -jC3 | -jC5 | -C1 | t6 |
| 7 | 1 | jC3 | -jC5 | C1 | jC3 | -jC5 | -C1 | -jC3 | jC5 | C1 | -jC3 | jC5 | -C1 | t7 |
| 8 | 2 | C10 | C13 | C14 | C10 | C13 | -C14 | -C10 | -C13 | C14 | -C10 | -C13 | -C14 | t8 |
| 9 | 2 | C10 | -C13 | C14 | C10 | -C13 | -C14 | -C10 | C13 | C14 | -C10 | C13 | -C14 | t9 |
| 10 | 2 | jC10 | jC13 | C14 | jC10 | jC13 | -C14 | -jC10 | -jC13 | C14 | -jC10 | -jC13 | -C14 | t10 |
| 11 | 2 | jC10 | -jC13 | C14 | jC10 | -jC13 | -C14 | -jC10 | jC13 | C14 | -jC10 | jC13 | -C14 | t11 |
| 12 | 2 | jC10 | jC14 | C13 | jC10 | jC14 | -C13 | -jC10 | -jC14 | C13 | -jC10 | -jC14 | -C13 | t12 |
| 13 | 2 | jC10 | -jC14 | C13 | jC10 | -jC14 | -C13 | -jC10 | jC14 | C13 | -jC10 | jC14 | -C13 | t13 |
| 14 | 2 | jC13 | jC14 | C10 | jC13 | jC14 | -C10 | -jC13 | -jC14 | C10 | -jC13 | -jC14 | -C10 | t14 |
| 15 | 2 | jC13 | -jC14 | C10 | jC13 | -jC14 | -C10 | -jC13 | jC14 | C10 | -jC13 | jC14 | -C10 | t15 |
| 16 | 3 | C0 | C6 | C12 | C0 | C6 | -C12 | -C0 | -C6 | C12 | -C0 | -C6 | -C12 | t16 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 | 3 | jC6 | -jC12 | C0 | jC6 | -jC12 | -C0 | -jC6 | jC12 | C0 | -jC6 | jC12 | -C0 | t20 |
| 24 | 4 | C4 | C8 | C15 | C4 | C8 | -C15 | -C4 | -C8 | C15 | -C4 | -C8 | -C15 | t24 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | 4 | jC8 | -jC15 | C4 | jC8 | -jC15 | -C4 | -jC8 | jC15 | C4 | -jC8 | jC15 | -C4 | t31 |

NOTE: The code construction for code groups 0 to 15 using the SCH codes from code sets 1 and 2 is shown. The construction for code groups 16 to 31 using the SCH codes from code sets 3 and 4 is done in the same way.

## 7.3 Evaluation of synchronisation codes

The evaluation of information transmitted in SCH on code group and frame timing is shown in table 6, where the 32 code groups are listed. Each code group is containing 4 specific scrambling codes (cf. subclause 6.4), each scrambling code associated with a specific short and long basic midamble code.

Each code group is additionally linked to a specific tOffset, thus to a specific frame timing. By using this scheme, the UE can derive the position of the frame border due to the position of the SCH sequence and the knowledge of tOffset. The complete mapping of Code Group to Scrambling Code, Midamble Codes and tOffset is depicted in table 6.

Table 6: Mapping scheme for Cell Parameters, Code Groups,  
Scrambling Codes, Midambles and tOffset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CELL PARA-METER | Code Group | Associated Codes | | | Associated tOffset |
| Scrambling Code | Long Basic Midamble Code | Short Basic Midamble Code |
| 0 | Group 0 | Code 0 | mPL0 | mSL0 | t0 |
| 1 | Code 1 | mPL1 | mSL1 |
| 2 | Code 2 | mPL2 | mSL2 |
| 3 | Code 3 | mPL3 | mSL3 |
| 4 | Group 1 | Code 4 | mPL4 | mSL4 | t1 |
| 5 | Code 5 | mPL5 | mSL5 |
| 6 | Code 6 | mPL6 | mSL6 |
| 7 | Code 7 | mPL7 | mSL7 |
| .  .  .  . | | | | | |
| 124 | Group 31 | Code 124 | mPL124 | mSL124 | t31 |
| 125 | Code 125 | mPL125 | mSL125 |
| 126 | Code 126 | mPL126 | mSL126 |
| 127 | Code 127 | mPL127 | mSL127 |

For basic midamble codes mP cf. [7], annex A 'Basic Midamble Codes'.

Each cell shall cycle through two sets of cell parameters in a code group with the cell parameters changing each frame. Table 7 shows how the cell parameters are cycled according to the SFN.

Table 7: Alignment of cell parameter cycling and SFN

|  |  |  |  |
| --- | --- | --- | --- |
| Initial Cell Parameter Assignment | Code Group | Cell Parameter used when SFN mod 2 = 0 | Cell Parameter used when SFN mod 2 = 1 |
| 0 | Group 0 | 0 | 1 |
| 1 | 1 | 0 |
| 2 | 2 | 3 |
| 3 | 3 | 2 |
| 4 | Group 1 | 4 | 5 |
| 5 | 5 | 4 |
| 6 | 6 | 7 |
| 7 | 7 | 6 |
| .  .  .  . | | | |
| 124 | Group 31 | 124 | 125 |
| 125 | 125 | 124 |
| 126 | 126 | 127 |
| 127 | 127 | 126 |

## 7.4 Synchronisation codes for 3.84 Mcps MBSFN IMB

### 7.4.1 Code generation

The primary synchronisation code (PSC), Cpsc, is constructed to be orthogonal to both the primary and secondary SCH codes defined in subclause 7.1. The PSC is furthermore chosen to have good aperiodic auto correlation properties and low aperiodic cross correlations with the primary SCH defined in sub-clause 7.1.

Define:

- a = <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 Cpsc is defined as:

- Cpsc = (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, {Cssc,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 = <1, 1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, 1, 1, -1>.

The Hadamard sequences are obtained as the rows in a matrix *H*8 constructed recursively by:

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 *H8* numbered from the top, n = 0, 1, 2, …, 255,in the sequel.

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

The *k*:th secondary synchronization code, Cssc,k, *k* = 1, 2, 3, …, 16 is then defined as:

- Cssc,k = (1 + *j*)  <*hm*(0)  *z*(0), *hm*(1)  *z*(1), *hm*(2)  *z*(2), …, *hm*(255)  *z*(255)>;

where *m* = 16(*k* – 1) and the leftmost chip in the sequence corresponds to the chip transmitted first in time.

### 7.4.2 Code allocation of SSC

The secondary synchronisation code sequences shall be constructed as described in subclause 5.2.3.2 in [4]. For MBSFN IMB operation, only the first 8 scrambling code groups are utilised.

# 7A Synchronisation codes for the 7.68 Mcps option

## 7A.1 Code Generation

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

Define a = < x1, x2, x3, …, x16 > = < 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, 1 >

The PSC of length 512 chips is generated by repetition coding and repeating the sequence 'a' modulated by a Golay complementary sequence and creating a complex-valued sequence with identical real and imaginary components.

The PSC, Cp , is defined as Cp = < y(0),y(0),y(1),y(1),y(2),y(2)...,y(255),y(255) >

where

and the left most index corresponds to the chip transmitted first in time.

The 12 secondary synchronization codes, {C0, C1, C3, C4, C5, C6, C8, C10, C12, C13, C14,C15 } are complex valued with identical real and imaginary components, and are constructed from repetition coding of the position wise multiplication of a Hadamard sequence and a sequence z, defined as

z =  , where

b = 

and x1, x2, x3, …, x16 are the 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:



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

Denote the *n*:th Hadamard sequence *hn* as a row of *H8* numbered from the top, n = 0, 1, 2, …, 255,in the sequel.

Furthermore, let *hm(l)* and *z(l)* denote the *l*th symbol of the sequence *hm* and *z,* respectively where *l* = 0, 1, 2, …, 255 and *l* = 0 corresponds to the leftmost symbol.

The i:th secondary SCH code word, Ci, i = 0, 1, 3, 4, 5, 6, 8, 10, 12, 13, 14, 15 is of length 512 chips and is then defined as

Ci = (1 + *j*)  <*hm*(0)  *z*(0), *hm*(0)  *z*(0), *hm*(1)  *z*(1), *hm*(1)  *z*(1), …, *hm*(255)  *z*(255) , *hm*(255)  *z*(255)>,

where *m* = (16i) and the leftmost chip in the sequence corresponds to the chip transmitted first in time.

## 7A.2 Code Allocation

Three secondary SCH codes are QPSK modulated and transmitted in parallel with the primary synchronization code. The QPSK modulation carries the following information:

- the code group that the base station belongs to (32 code groups:5 bits; Cases 1, 2);

- the position of the frame within an interleaving period of 20 msec (2 frames:1 bit, Cases 1, 2);

- the position of the SCH slot(s) within the frame (2 SCH slots:1 bit, Case 2).

The QPSK modulation sequences for the 7.68Mcps TDD option are unique to the modulation sequences for the 3.84Mcps TDD option.

The modulated secondary SCH codes are also constructed such that their cyclic-shifts are unique, i.e. a non-zero cyclic shift less than 2 (Case 1) and 4 (Case 2) of any of the sequences is not equivalent to some cyclic shift of any other of the sequences. Also, a non-zero cyclic shift less than 2 (Case 1) and 4 (Case 2) of any of the sequences is not equivalent to itself with any other cyclic shift less than 8. The secondary synchronization codes are partitioned into two code sets for Case 1 and four code sets for Case 2. The set is used to provide the following information:

Case 1:

Table 7A: Code Set Allocation for Case 1

|  |  |
| --- | --- |
| Code Set | Code Group |
| 1 | 0-15 |
| 2 | 16-31 |

The code group and frame position information is provided by modulating the secondary codes in the code set.

Case 2:

Table 7B: Code Set Allocation for Case 2

|  |  |
| --- | --- |
| Code Set | Code Group |
| 1 | 0-7 |
| 2 | 8-15 |
| 3 | 16-23 |
| 4 | 24-31 |

The slot timing and frame position information is provided by the comma free property of the code word and the Code group is provided by modulating some of the secondary codes in the code set.

The following SCH codes are allocated for each code set:

Case 1

Code set 1: C1, C3, C5.

Code set 2: C10, C13, C14.

Case 2

Code set 1: C1, C3, C5.

Code set 2: C10, C13, C14.

Code set 3: C0, C6, C12.

Code set 4: C4, C8, C15.

The following subclauses 7A.2.1 to 7A.2.2 refer to the two cases of SCH/P-CCPCH usage as described in [7].

Note that in the tables 7C and 7D corresponding to Cases 1 and 2, respectively, Frame 1 implies the frame with an odd SFN and Frame 2 implies the frame with an even SFN.

### 7A.2.1 Code allocation for Case 1

Table 7D: Code Allocation for Case 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code Group | Code Set | Frame 1 | | | Frame 2 | | | Associated toffset |
| 0 | 1 | C1 | C3 | jC5 | C1 | C3 | -jC5 | t0 |
| 1 | 1 | C1 | -C3 | jC5 | C1 | -C3 | -jC5 | t1 |
| 2 | 1 | -C1 | C3 | jC5 | -C1 | C3 | -jC5 | t2 |
| 3 | 1 | -C1 | -C3 | jC5 | -C1 | -C3 | -jC5 | t3 |
| 4 | 1 | jC1 | jC3 | jC5 | jC1 | jC3 | -jC5 | t4 |
| 5 | 1 | jC1 | -jC3 | jC5 | jC1 | -jC3 | -jC5 | t5 |
| 6 | 1 | -jC1 | jC3 | jC5 | -jC1 | jC3 | -jC5 | t6 |
| 7 | 1 | -jC1 | -jC3 | jC5 | -jC1 | -jC3 | -jC5 | t7 |
| 8 | 1 | jC1 | C5 | C3 | jC1 | C5 | -C3 | t8 |
| 9 | 1 | jC1 | -C5 | C3 | jC1 | -C5 | -C3 | t9 |
| 10 | 1 | -jC1 | C5 | C3 | -jC1 | C5 | -C3 | t10 |
| 11 | 1 | -jC1 | -C5 | C3 | -jC1 | -C5 | -C3 | t11 |
| 12 | 1 | jC3 | C5 | C1 | jC3 | C5 | -C1 | t12 |
| 13 | 1 | jC3 | -C5 | C1 | jC3 | -C5 | -C1 | t13 |
| 14 | 1 | -jC3 | C5 | C1 | -jC3 | C5 | -C1 | t14 |
| 15 | 1 | -jC3 | -C5 | C1 | -jC3 | -C5 | -C1 | t15 |
| 16 | 2 | C10 | C13 | jC14 | C10 | C13 | -jC14 | t16 |
| 17 | 2 | C10 | -C13 | jC14 | C10 | -C13 | -jC14 | t17 |
|  |  |  |  |  |  |  |  |  |
| 20 | 2 | jC10 | jC13 | jC14 | jC10 | jC13 | -jC14 | t20 |
|  |  |  |  |  |  |  |  |  |
| 24 | 2 | jC10 | C14 | C13 | jC10 | C14 | -C13 | t24 |
|  |  |  |  |  |  |  |  |  |
| 31 | 2 | -jC13 | -C14 | C10 | -jC13 | -C14 | -C10 | t31 |

NOTE: The code construction for code groups 0 to 15 using only the SCH codes from code set 1 is shown. The construction for code groups 16 to 31 using the SCH codes from code set 2 is done in the same way.

### 7A.2.2 Code allocation for Case 2

Table 7C: Code Allocation for Case 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code Group | Code Set | Frame 1 | | | | | | Frame 2 | | | | | | Associated toffset |
| Slot k | | | Slot k+8 | | | Slot k | | | Slot k+8 | | |
| 0 | 1 | C1 | C3 | jC5 | C1 | C3 | -jC5 | -C1 | -C3 | jC5 | -C1 | -C3 | -jC5 | t0 |
| 1 | 1 | C1 | -C3 | jC5 | C1 | -C3 | -jC5 | -C1 | C3 | jC5 | -C1 | C3 | -jC5 | t1 |
| 2 | 1 | jC1 | jC3 | jC5 | jC1 | jC3 | -jC5 | -jC1 | -jC3 | jC5 | -jC1 | -jC3 | -jC5 | t2 |
| 3 | 1 | jC1 | -jC3 | jC5 | jC1 | -jC3 | -jC5 | -jC1 | jC3 | jC5 | -jC1 | jC3 | -jC5 | t3 |
| 4 | 1 | jC1 | C5 | C3 | jC1 | C5 | -C3 | -jC1 | -C5 | C3 | -jC1 | -C5 | -C3 | t4 |
| 5 | 1 | jC1 | -C5 | C3 | jC1 | -C5 | -C3 | -jC1 | C5 | C3 | -jC1 | C5 | -C3 | t5 |
| 6 | 1 | jC3 | C5 | C1 | jC3 | C5 | -C1 | -jC3 | -C5 | C1 | -jC3 | -C5 | -C1 | t6 |
| 7 | 1 | jC3 | -C5 | C1 | jC3 | -C5 | -C1 | -jC3 | C5 | C1 | -jC3 | C5 | -C1 | t7 |
| 8 | 2 | C10 | C13 | jC14 | C10 | C13 | -jC14 | -C10 | -C13 | jC14 | -C10 | -C13 | -jC14 | t8 |
| 9 | 2 | C10 | -C13 | jC14 | C10 | -C13 | -jC14 | -C10 | C13 | jC14 | -C10 | C13 | -jC14 | t9 |
| 10 | 2 | jC10 | jC13 | jC14 | jC10 | jC13 | -jC14 | -jC10 | -jC13 | jC14 | -jC10 | -jC13 | -jC14 | t10 |
| 11 | 2 | jC10 | -jC13 | jC14 | jC10 | -jC13 | -jC14 | -jC10 | jC13 | jC14 | -jC10 | jC13 | -jC14 | t11 |
| 12 | 2 | jC10 | C14 | C13 | jC10 | C14 | -C13 | -jC10 | -C14 | C13 | -jC10 | -C14 | -C13 | t12 |
| 13 | 2 | jC10 | -C14 | C13 | jC10 | -C14 | -C13 | -jC10 | C14 | C13 | -jC10 | C14 | -C13 | t13 |
| 14 | 2 | jC13 | C14 | C10 | jC13 | C14 | -C10 | -jC13 | -C14 | C10 | -jC13 | -C14 | -C10 | t14 |
| 15 | 2 | jC13 | -C14 | C10 | jC13 | -C14 | -C10 | -jC13 | C14 | C10 | -jC13 | C14 | -C10 | t15 |
| 16 | 3 | C0 | C6 | jC12 | C0 | C6 | -jC12 | -C0 | -C6 | jC12 | -C0 | -C6 | -jC12 | t16 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 | 3 | jC6 | -C12 | C0 | jC6 | -C12 | -C0 | -jC6 | C12 | C0 | -jC6 | C12 | -C0 | t20 |
| 24 | 4 | C4 | C8 | jC15 | C4 | C8 | -jC15 | -C4 | -C8 | jC15 | -C4 | -C8 | -jC15 | t24 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | 4 | jC8 | -C15 | C4 | jC8 | -C15 | -C4 | -jC8 | C15 | C4 | -jC8 | C15 | -C4 | t31 |

NOTE: The code construction for code groups 0 to 15 using the SCH codes from code sets 1 and 2 is shown. The construction for code groups 16 to 31 using the SCH codes from code sets 3 and 4 is done in the same way.

## 7A.3 Evaluation of synchronisation codes

The evaluation of information transmitted in SCH on code group and frame timing is shown in table 7E, where the 32 code groups are listed. Each code group contains 4 specific scrambling codes, each scrambling code associated with a specific short and long basic midamble code.

Each code group is additionally linked to a specific tOffset, thus to a specific frame timing. By using this scheme, the UE can derive the position of the frame border due to the position of the SCH sequence and the knowledge of tOffset. The complete mapping of Code Group to Scrambling Code, Midamble Codes and tOffset is depicted in table 7E.

Table 7E: Mapping scheme for Cell Parameters, Code Groups,  
Scrambling Codes, Midambles and tOffset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CELL PARA-METER | Code Group | Associated Codes | | | Associated tOffset |
| Scrambling Code | Long Basic Midamble Code | Short Basic Midamble Code |
| 0 | Group 0 | Code 0 | mPL0 | mSL0 | t0 |
| 1 | Code 1 | mPL1 | mSL1 |
| 2 | Code 2 | mPL2 | mSL2 |
| 3 | Code 3 | mPL3 | mSL3 |
| 4 | Group 1 | Code 4 | mPL4 | mSL4 | t1 |
| 5 | Code 5 | mPL5 | mSL5 |
| 6 | Code 6 | mPL6 | mSL6 |
| 7 | Code 7 | mPL7 | mSL7 |
| .  .  .  . | | | | | |
| 124 | Group 31 | Code 124 | mPL124 | mSL124 | t31 |
| 125 | Code 125 | mPL125 | mSL125 |
| 126 | Code 126 | mPL126 | mSL126 |
| 127 | Code 127 | mPL127 | mSL127 |

Each cell shall cycle through two sets of cell parameters in a code group with the cell parameters changing each frame. Table 7F shows how the cell parameters are cycled according to the SFN.

Table 7F: Alignment of cell parameter cycling and SFN

|  |  |  |  |
| --- | --- | --- | --- |
| Initial Cell Parameter Assignment | Code Group | Cell Parameter used when SFN mod 2 = 0 | Cell Parameter used when SFN mod 2 = 1 |
| 0 | Group 0 | 0 | 1 |
| 1 | 1 | 0 |
| 2 | 2 | 3 |
| 3 | 3 | 2 |
| 4 | Group 1 | 4 | 5 |
| 5 | 5 | 4 |
| 6 | 6 | 7 |
| 7 | 7 | 6 |
| .  .  .  . | | | |
| 124 | Group 31 | 124 | 125 |
| 125 | 125 | 124 |
| 126 | 126 | 127 |
| 127 | 127 | 126 |

# 8 Synchronisation codes for the 1.28 Mcps option

## 8.1 The downlink pilot channel (DwPCH)

The contents of DwPCH is composed of 64 chips of a SYNC-DL sequence, cf.[AA.1 Basic SYNC-DL sequence] and 32 chips of guard period (GP). The SYNC-DL code is not scrambled

There should be 32 different basic SYNC-DL codes for the whole system.

For the generation of the complex valued SYNC-DL codes of length 64, the basic binary SYNC-DL codes  of length 64 shown in Table AA.1 are used. The relation between the elements and  is given by:

 (1)

Hence, the elements  of the complex SYNC-DL code  are alternating real and imaginary.

The SYNC-DL is QPSK modulated and the phase of the SYNC-DL is used to signal the presence of the P-CCPCH in the multi-frame of the resource units of code  and  in time slot #0.

### 8.1.1 Modulation of the SYNC-DL

The SYNC-DL sequences are modulated with respect to the midamble (m(1)) in time slot #0.

Four consecutive phases (phase quadruple) of the SYNC-DL are used to indicate the presence of the P-CCPCH in the following 4 sub-frames. In case the presence of a P-CCPCH is indicated, the next following sub-frame is the first sub-frame of the interleaving period. As QPSK is used for the modulation of the SYNC-DL, the phases 45, 135, 225, and 315° are used.

The total number of different phase quadruples is 2 (S1 and S2). A quadruple always starts with an even system frame number ((SFN mod 2) =0). Table 8 is showing the quadruples and their meaning.

Table 8: Sequences for the phase modulation for the SYNC-DL

|  |  |  |
| --- | --- | --- |
| Name | Phase quadruple | Meaning |
| S1 | 135, 45, 225, 135 | There is a P-CCPCH in the next 4 sub-frames |
| S2 | 315, 225, 315, 45 | There is no P-CCPCH in the next 4 sub-frames |

## 8.2 The uplink pilot channel (UpPCH)

The contents in UpPCH is composed of 128 chips of a SYNC-UL sequence, cf. [AA.2 Basic SYNC-UL sequence] and 32chips of guard period (GP) .The SYNC-UL code is not scrambled.

There should be 256 different basic SYNC-UL codes (see Table AA.2) for the whole system.

For the generation of the complex valued SYNC-UL codes of length 128, the basic binary SYNC-UL codes  of length 128 shown in Table AA.2 are used. The relation between the elements and  is given by:

 (2)

Hence, the elements  of the complex SYNC-UL code  are alternating real and imaginary.

## 8.3 Code Allocation

Relationship between the SYNC-DL and SYNC-UL sequences, the scrambling codes and the midamble codes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code Group | Associated Codes | | | |
| SYNC-DL  ID | SYNC-UL  ID | Scrambling Code  ID | Basic Midamble Code  ID |
| Group 1 | 0 | 0...7 | 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| Group 2 | 1 | 8...15 | 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| .  .  . | | | | |
| Group 32 | 31 | 248...255 | 124 | 124 |
| 125 | 125 |
| 126 | 126 |
| 127 | 127 |

Note: In a multi-frequency cell, primary frequency and secondary frequency use the same scrambling code and basic midamble code.

## 8.3Aa Code Allocation

For the dedicated carrier MBSFN, the basic preamble codes are segemted into two groups, even group and odd group, and the Basic preamble Code is described in [7].

Relationship between the scrambling codes and the preamble codes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code subgroup | Associated Codes | | | |
| PP,even group | | PP,odd group | |
| Scrambling Code ID | Basic preamble Code ID | Scrambling Code ID | Basic preamble Code ID |
| Sub-g1 | 0 | 0 | 4 | 1 |
| Sub-g2 | 8 | 2 | 12 | 3 |
| Sub-g3 | 16 | 4 | 20 | 5 |
| Sub-g4 | 24 | 6 | 28 | 7 |
| .  .  . | | | | |
| Sub-g15 | 112 | 28 | 116 | 29 |
| Sub-g16 | 120 | 30 | 124 | 31 |

# 9 Cell synchronisation codes

The cell synchronisation codes (CSCs) are constructed as so-called CEC sequences, i.e. concatenated and periodically extended complementary sequences. They are complex-valued sequences that are derived as cyclically offset versions from a set of possible constituent Golay complementary pairs.

The CSCs are chosen to have good aperiodic auto correlation properties. The aperiodic auto correlations of the applicable constituent Golay complementary pairs and every pair of their derived cyclically offset versions are complementary. Furthermore, orthogonality is preserved for all CSCs which are derived from the same constituent Golay complementary pair due to this complementary property.

The delay and weight matrices for the set of M = 8 possible constituent Golay complementary pairs are listed in the table below:

|  |  |
| --- | --- |
| Code ID *m* | Delay matrices D*m* and weight matrices W*m* of constituent Golay complementary pairs |
| 0 | D0 = <512, 64, 128, 1, 16, 4, 256, 32, 8, 2>, W0 = <1, 1, 1, 1, -1, -1, 1, 1, 1, 1> |
| 1 | D1 = <2, 16, 32, 256, 1, 8, 128, 4, 512, 64>, W1 = <1, -1, 1, -1, 1, -1, -1, 1, -1, -1> |
| 2 | D2 = <16, 512, 32, 256, 4, 1, 64, 8, 2, 128>, W2 = <-1, 1, 1, -1, -1, 1, -1, 1, -1, -1> |
| 3 | D3 = <512, 16, 8, 4, 2, 256, 128, 64, 32, 1>, W3 = <-1, -1, -1, -1, -1, 1, -1, 1, 1, 1> |
| 4 | D4 = <512, 128, 256, 32, 2, 4, 64, 1, 16, 8>, W4 = <1, -1, 1, -1, -1, -1, -1, -1, -1, 1> |
| 5 | D5 = <1, 2, 4, 64, 512, 16, 32, 256, 128, 8>, W5 = <-1, 1, 1, 1, 1, -1, -1, 1, -1, 1> |
| 6 | D6 = <8, 16, 128, 2, 32, 1, 256, 512, 4, 64>, W6 = <-1, -1, 1, 1, 1, 1, -1, -1, -1, 1> |
| 7 | D7 = <1, 2, 128, 16, 256, 32, 8, 512, 64, 4>, W7 = <1, 1, -1, -1, -1, -1, 1, -1, -1, -1> |

A constituent Golay complementary pair of length N = 1024, defined as:

s*m* = <*sm*(0), *sm*(1), *sm*(2), …, *sm*(1023)> and g*m* = <*gm*(0), *gm*(1), *gm*(2), …, *gm*(1023)>

shall be derived from the selected delay and weight matrices:

D*m* = <*Dm*(0), *Dm*(1), *Dm*(2), …, *Dm*(9)> and W*m* = <*Wm*(0), *Wm*(1), *Wm*(2), …, *Wm*(9)>

as follows.

Define:

a*(0)* = <*a(0)*(0), *a(0)*(1), *a(0)*(2), … , *a(0)*(1023)> = <1, 0, 0, … , 0> and

b*(0)* = <*b(0)*(0), *b(0*)(1), *b(0)*(2), … , *b(0)*(1023)> = <1, 0, 0, … , 0>.

Then, the elements of the set of auxiliary sequences:

a*(n)* = <*a(n)*(0), *a(n)*(1), *a(n)*(2), … , *a(n)*(1023)> and b*(n)* = <*b(n)*(0), *b(n)*(1), *b(n)*(2), … , *b(n)*(1023)>

are given by the recursive relations:

*a(n+1)(i) = a(n)(i) + Wm(n)  b(n)(i – Dm(n))* and

*b(n+1)(i) = a(n)(i) – Wm(n)  b(n)(i – Dm(n))*

with element index *i* = 0, 1, 2, …, 1023 and iteration index *n* = 0, 1, 2, …, 9. Operations on the element index shall be performed modulo 1024.

The elements of the constituent Golay complementary pairs s*m* and g*m* are then obtained from the output of the last iteration step using:

*sm*(*i*) = *a(10)*(*i*) and *gm*(*i*) = *b(10)*(*i*) for *i* = 0, 1, 2, ..., 1023

From each applicable constituent Golay complementary pair s*m* and g*m*, up to K = 8 different cyclically offset pairs s*m(k)* and g*m(k)*, with offset index *k* = 0, 1, 2, …, K-1, of length 1152 chips can be derived. The complementary property of the respective aperiodic auto correlation is preserved for each particular pair of sequences s*m(k)* and g*m(k)*. The generation of the K cyclically offset pairs from s*m* and g*m* is done in a similar way as the generation of the user midambles from a periodic basic midamble sequence as described in [7].

With N = 1024, K = 8, W = 128, the elements of a cyclically offset pair:

s*m(k)* = <*sm(k)*(0), *sm(k)*(1), *sm(k)*(2), …, *sm(k)*(1151)> and g*m(k)* = <g*m(k)*(0), g*m(k)*(1), g*m(k)*(2), …, g*m(k)*(1151)>

for a particular offset *k*, with *k* = 0, 1, 2, …, K-1, shall be derived from the elements of the constituent Golay complementary pairs s*m* and g*m* using:

*sm(k)*(*i*) = (j)*i*  *sm*(*i* + *k* W) and *gm(k)*(*i*) = (j)*i*  *gm*(*i* + *k* W) for *i* = 0, 1, 2, ..., N *– k* W *–* 1,

*sm(k)*(*i*) = (j)*i*  *sm*(*i* *–* N *+ k* W) and *gm(k)*(*i*) = (j)*i*  *gm*(*i* *–* N *+ k* W) for *i* = N *– k* W, N *– k* W + 1, ..., 1151.

Hence, the elements of s*m(k)* and g*m(k)* are alternating real and imaginary.

Note that both s*m(0)* and g*m(0)* simply correspond to s*m* and g*m* respectively, followed by its first W elements as post extension and that both s*m(7)* and g*m(7)* simply correspond to the last W elements of s*m* and g*m* in form of a pre extension, followed by s*m* and g*m* respectively.

Finally, the CSC C*CSC, m(k)* derived from the *m*:th applicable constituent Golay complementary pair s*m* and g*m*, and for the *k*:th offset is then defined as a concatenation of s*m(k)* and g*m(k)* by:

C*CSC, m(k)* = <*sm(k)*(0), *sm(k)*(1), *sm(k)*(2), …, *sm(k)*(1151), *gm(k)*(0), *gm(k)*(1), *gm(k)*(2), …, *gm(k)*(1151)>

where the leftmost element *sm(k)*(0) in the sequence corresponds to the chip to be first transmitted in time. An CSC has therefore length 2304 chips.

Note that due to this construction method, the auto correlations for all CSCs derived from one particular constituent Golay complementary pair s*m* and g*m* can be obtained simultaneously and in sequential order from the sum of partial correlations with s*m* and g*m*, these CSCs remaining orthogonal.

CSCs derived according to above have complex values and shall not be subject to the channelisation or scrambling process, i.e. its elements represent complex chips for usage in the pulse shaping process at modulation.

Annex A (normative):  
Scrambling Codes

The applicable scrambling codes are listed below. Code numbers are referring to table 6 'Mapping scheme for Cell Parameters, Code Groups, Scrambling Codes, Midambles, Preambles and toffset' in subclause 6.3 'Evaluation of synchronisation codes'.

Code{0，4，8，12，…，120，124} are used for downlink MBSFN operation.

| Scrambling Code | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code 0 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 |
| Code 2 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 4 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 5 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 |
| Code 6 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 7 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 |
| Code 8 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 9 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 10 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 11 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 12 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 13 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 14 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 15 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| Code 16 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 |
| Code 17 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 18 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 19 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 20 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 21 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 |
| Code 22 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 23 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 24 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 25 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 26 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 27 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 28 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 29 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 30 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 |
| Code 31 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 |
| Code 32 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 33 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 34 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 35 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 36 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 37 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 38 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 39 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 40 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 41 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 42 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 |
| Code 43 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 |
| Code 44 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 45 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 46 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 |
| Code 47 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 |
| Code 48 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 49 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 |
| Code 50 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 51 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| Code 52 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 53 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 |
| Code 54 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 55 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 56 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 57 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 |
| Code 58 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 59 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 |
| Code 60 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 |
| Code 61 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| Code 62 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 |
| Code 63 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 |
| Code 64 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 |
| Code 65 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 66 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 |
| Code 67 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 |
| Code 68 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 |
| Code 69 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 |
| Code 70 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 71 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 |
| Code 72 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 |
| Code 73 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 |
| Code 74 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 |
| Code 75 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 |
| Code 76 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 |
| Code 77 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 78 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 |
| Code 79 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 |
| Code 80 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 |
| Code 81 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 82 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 |
| Code 83 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 |
| Code 84 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 |
| Code 85 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 |
| Code 86 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 |
| Code 87 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 88 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| Code 89 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| Code 90 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| Code 91 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| Code 92 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 |
| Code 93 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 |
| Code 94 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 |
| Code 95 | 1 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 |
| Code 96 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 |
| Code 97 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 |
| Code 98 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| Code 99 | 1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 |
| Code 100 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 |
| Code 101 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 |
| Code 102 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 103 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| Code 104 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 |
| Code 105 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 |
| Code 106 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 |
| Code 107 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 |
| Code 108 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 |
| Code 109 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 |
| Code 110 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 111 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| Code 112 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 113 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | -1 |
| Code 114 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| Code 115 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 116 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| Code 117 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 |
| Code 118 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 |
| Code 119 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 |
| Code 120 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 |
| Code 121 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| Code 122 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 |
| Code 123 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 |
| Code 124 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 |
| Code 125 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 |
| Code 126 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | -1 |
| Code 127 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 |

Annex AA (normative):  
Synchronisation sequence

## AA.1 Basic SYNC-DL sequence

Table AA.1: Basic SYNC-DL Codes

|  |  |
| --- | --- |
| Code ID | SYNC-DL Codes of length 64 |
| 0 | B3A7CC05A98688E4 |
| 1 | 9D559BD290606791 |
| 2 | 2CE7BA12A017C3A2 |
| 3 | 34511D20672F4712 |
| 4 | 9A772841474603F2 |
| 5 | 9109B1A5CE01F228 |
| 6 | 8FD429B3594501C0 |
| 7 | 25251354AA3F8C19 |
| 8 | C9A3B8E0C043EA56 |
| 9 | BA04B888E5BC1802 |
| 10 | A735354299370207 |
| 11 | 74C3C8DA4415AE51 |
| 12 | F4FD0458A0124663 |
| 13 | A011D4E16C3D6064 |
| 14 | BDA0661B0CAA8C68 |
| 15 | 8E31123F28928698 |
| 16 | F095C1632E2906AB |
| 17 | B60B4A8A664071CF |
| 18 | AA094DCCE91E041A |
| 19 | C0C31CDA8A256807 |
| 20 | D516964FB18C1890 |
| 21 | 30DE01834F4AACCE |
| 22 | 8F700323BA5CAD34 |
| 23 | 1B50F4DEE0C1380C |
| 24 | 443382164F56F2D1 |
| 25 | E1E4005D49B846B4 |
| 26 | 040A97165330BFAA |
| 27 | C48E26881693AD78 |
| 28 | D4354B2FE02361CC |
| 29 | 5383AB6C8A10CE84 |
| 30 | D417A730F2F12244 |
| 31 | ABF0A0D905A939C4 |

## AA.2 Basic SYNC-UL Codes

Table AA.2: Basic SYNC-UL Codes

|  |  |
| --- | --- |
| Code ID | SYNC-UL Codes of length 128 |
| 0 | C11C20F0D1807DB8859175B798EC094A |
| 1 | 91278068081EC8E74543DBC1C9AD4235 |
| 2 | 38F5AEE2E513DB12A663BA04160103E5 |
| 3 | 7AA8A0A210F12A1E4332F2EDD33011FC |
| 4 | C180EA3B9BA1774EB9611BD249C4A508 |
| 5 | B072A2C839489D496B98CE9D0132FBC9 |
| 6 | B2723EAC6EB01667F2B33961C8074234 |
| 7 | C4144AD060F0EC095E227B92CF7C8280 |
| 8 | 653036A10D3054146FCF815986C63A14 |
| 9 | F899CA61435D64DC07FDF04C4A0C053A |
| 10 | B56F2D6893A8051407F4C341D88DC7DC |
| 11 | DC0BE838242142EDE6413A72C88D74AA |
| 12 | 22A2FD86E4086C70A4860B13C76E579F |
| 13 | A3CBC21322C97D2A02728E7875F39588 |
| 14 | D4EC4F694A082CB38E3B1558A0FCC89F |
| 15 | CC891141C4E216D235C15CF5D3F9B002 |
| 16 | A1993114C50B77CB0C0725D1E22FD016 |
| 17 | 24F73A979DE52F82E8800CCB93842A59 |
| 18 | 8F878FA04659842E294D8DEAB20BA2FD |
| 19 | AC90B0442D70662B028CF76A6BECDF09 |
| 20 | D94A284DF64D7B0102F0E084C29C88C8 |
| 21 | 8603200C7596F24E865FD3815693358D |
| 22 | B466B12CF433642BD8B08F1F452E0550 |
| 23 | 86A3A1772C1C99FCA7DBBA0C312E34A0 |
| 24 | 622A1889F72A9A2C042D46F08EFEE1AC |
| 25 | BF220A362BC0D3B0D7CE400954C6CFAE |
| 26 | D28D73C52E89CF57905C502244F63616 |
| 27 | AD4E1C2103697D64D8B9D4C035D90548 |
| 28 | 8F081A9BA12B6C6BD024531AA984D21C |
| 29 | E4092429BE82988E1E3585BF6A6AE550 |
| 30 | 08BD36E0A9C061782CB38B35B335CA56 |
| 31 | 1CDFF3CC2685D1C44F4A1059AB03F40A |
| 32 | 506ED4E88FB1CECE3243F2A27A0221A4 |
| 33 | 846CF58A7AB613C83A24130B5778C0E2 |
| 34 | A2711A99E26A0C75AC026F4CFAECE893 |
| 35 | D846EEEBA2432AC05A01043C62579DCF |
| 36 | 6B16B4E851CAF2121FC4CF88820C89E7 |
| 37 | AA4889A78207674A74E10C6F2BE11D48 |
| 38 | 8534CF8145BC991052814ED5C72709EE |
| 39 | 01AEF15D2290A84A607425746D9963C7 |
| 40 | 999188F758245D5164FE16D852942C71 |
| 41 | CF71C008599287E446E30745BD56E2D2 |
| 42 | 248414BA0DF8CDC4711FE7C8707ED0AD |
| 43 | EB2E263EC016191C81AB714BFE4D2B30 |
| 44 | 862082A7482FAC1C499793A0D8CED670 |
| 45 | DE2C22B2783AB75A7342608DE413840A |
| 46 | E31AA60B727F2CA2A78DAAC10665011D |
| 47 | CEF6CD06509870AC9E0177ACD550921D |
| 48 | E52C84D499FFCDC287581691471540F2 |
| 49 | B33BF6551A4322504BEE0930BCA1EC68 |
| 50 | 555BE6886D0FC43D72315E6C6D384148 |
| 51 | 8444F67451EE23CE1240C90F0B52A492 |
| 52 | 5C290D28E84060E69D09788A261B10FF |
| 53 | 337E0C35E83CD38CCC5D45804241F952 |
| 54 | A7879F0D31A8982A01EE6AC4952984DC |
| 55 | A37F506508928C70A83D69A2373781B9 |
| 56 | 42F55208EE12909803A7CBEB19B5419E |
| 57 | 57E5E268A328FCC9ED04B9E5420AC702 |
| 58 | EB033AD1222F84D8642C4E3FAAD28206 |
| 59 | 98EE1415F026AC0E862C520451697DD0 |
| 60 | 6A0528AEA4B7CD6702660D81F8821E19 |
| 61 | 763D626A87C603BCB09E1A4C800A378F |
| 62 | EEA61897879289340C23F669D6A03762 |
| 63 | A6571B3CC2D0E04F017ACC808B92DCE7 |
| 64 | DDF88B52EA1831D293A803CF23C8C471 |
| 65 | 6CA4D333A2684140475DAB491F61C17A |
| 66 | A7D2AD23043989A13289F7C3E135580A |
| 67 | B1C752FA66B41C81904EDE27EA000E2E |
| 68 | 8694BE3CC1CB36BE2A095F89CC619080 |
| 69 | 9C20334E1BBC596B25E151180BF99940 |
| 70 | 484256214F81070DD9C49A2B05A43DCE |
| 71 | 401A20BCBE29B7438A7AEE44635A9E23 |
| 72 | 8858585C3239CBF628033FA0DF189378 |
| 73 | EFA36404C1BA5118CC5F9052FD28D9C3 |
| 74 | 155609873D8A042D496E6477B747C4F8 |
| 75 | 8446077883A6D7D2549CC9742E3FD023 |
| 76 | E630142B189AA209371A6F0FFDBC30A7 |
| 77 | C46060535AC6DBB2095F1D7826D0CD5C |
| 78 | E00D19E48797148B28DEDA9D429362E2 |
| 79 | 645DE447E938485489416CAFCC1C571F |
| 80 | DA10AFBF2AE61C593A1D88584DE30598 |
| 81 | BB248AEA5FD3FE210CD48FC401E1A686 |
| 82 | A89F146BD9191F445301C081CB6F5625 |
| 83 | 15BBF04F247C59150208949EB6B9CC58 |
| 84 | 08F48BFA7804B5B2CC2E96510232E062 |
| 85 | 9AA2BE74005A3679C626B209580B8D03 |
| 86 | 9D40664A2C808F2F293E255398B37E6A |
| 87 | 6869C98A8AAD81CAE41A23C83FF9EEA0 |
| 88 | 576E8948E61BD0927C4140C3C04C4CF3 |
| 89 | 0F942C67A1137B6EAA058C2A74872C73 |
| 90 | 9D058E27ED546C10632684BBC84E5BC1 |
| 91 | 79D4B840E20148B134F90B51164BCBD0 |
| 92 | 0E35E1D8D1214C05FAC790B69B239150 |
| 93 | FFA1BB0232CD71480BE5CA1C2A269F89 |
| 94 | B2956F5F4E270446F9211584792628DB |
| 95 | F56CCA23421C8EC8F8A41F7DA4A41EA2 |
| 96 | 0B5ECA04F1789A7148C80C39D57D05F6 |
| 97 | A10B538E8A8CFC8F8925C485F2A88660 |
| 98 | 9925C2C715001D9FC78ACCC51DA1AF34 |
| 99 | 0DAC9CFDEA40429A8B12C7D320D60F70 |
| 100 | 377FC9A097017958440914E83118E39D |
| 101 | 8421096FA8B47E4E943B6473671955CC |
| 102 | 574086183477C4F68540CB7E858263B1 |
| 103 | 895B6A8980C6703C779F49F40C5CFC19 |
| 104 | D0D253E157BC19262150CEA668679E71 |
| 105 | B8889C60EBA812BD7F0B6498823296D2 |
| 106 | A13FB9F3A08528E44B13C12CF0D461AA |
| 107 | 8D4DCFBE43D6E2024B1F8470224AA330 |
| 108 | 536D159E119E0893838657B12A074E64 |
| 109 | DCFD49C504AD3A2F049A0CB70238EC8A |
| 110 | D363DB4C46C11757FA8FB18139789102 |
| 111 | 424A1E8A1D4DA256E4CA3BC8C2201BE3 |
| 112 | 417B619ED30FEB0A847CC3A191A20398 |
| 113 | 843FBBC95453C61786D1332612B45B4D |
| 114 | F26CACC0732CF8ED0C5BC1462B1620B4 |
| 115 | 88E0FE440C70E9249A92A7AF94638880 |
| 116 | 99A52B7D8C950308057E0661D7459960 |
| 117 | A5C28218BF5D16E63E42698A0A6B0896 |
| 118 | B2763BEEC784A12E8C50778536921806 |
| 119 | 987B2B6A3A77A059B30A082457AB84E0 |
| 120 | 820DB500F1B206358D7A7F210AB85AA8 |
| 121 | 97760A5CFC5E03EB439C914590045938 |
| 122 | 896A720E8857C8708A59F8C94DE0841E |
| 123 | 2D101F0CF95263843412577340DEBB11 |
| 124 | E8E5214B4DCF5D11A245B0149D49C87C |
| 125 | 51224EAA10099ACDE384834A5ADF03D8 |
| 126 | 64E51253554A230C186FDE4E8781BC09 |
| 127 | A499E391E69ED08890AC1A82A6115BEC |
| 128 | EE54C6E1834210D3EC1B07A456B92AA8 |
| 129 | 949DB5CA82420B54C1E0BCC111E704D9 |
| 130 | 9439EE9A9E4C447D1AA350926495047F |
| 131 | AD095CC0E7438AECE38D60980B3F2D00 |
| 132 | 83089C254C5EE9788072BC3D9282F798 |
| 133 | A27DC1A457BC5A56563D8A9B11203615 |
| 134 | 713053A9C0B1B08B14705FF5A7244DB4 |
| 135 | D36D4B9F4007354E0EC1B0CA8C8C7124 |
| 136 | 82E7C990612114F1CCE1BD9509FD4386 |
| 137 | C8D83FF0B48B14830D2015D53F8C0672 |
| 138 | 08AF223C869A36B169148FDDABB7D120 |
| 139 | B6C284C600AD0A99F86C449F8F4C53A6 |
| 140 | DC741B320C07682AF92AC4DBDE0C28C2 |
| 141 | 89B8D84FA902265850C0FA6FF0EB2C4F |
| 142 | A69445B3A52201DB984BC03D1956D7F3 |
| 143 | 0FE0F7224B7AD72E4D4530D0223F590C |
| 144 | 1B8C06F051434048EB925133AD3BD3F9 |
| 145 | E133D4C3C942726A351300C37E55D0DF |
| 146 | 9E09481D1881A66F562D8B453BC83AB2 |
| 147 | 2397B04B60A3C5700907BDBBA4E818C8 |
| 148 | 8F81F7A08CC6C8DA3D692AD34F50C012 |
| 149 | 9AB325352981BCCFA072F8FDE3009221 |
| 150 | 4FA88B7F1F8A620C31B0D486C52AC2F6 |
| 151 | 097AF0ADD16D7D39851049F0130EE444 |
| 152 | A5027732DACFF11C388D5820A4A9BA49 |
| 153 | 1CD981EA2EDB46218A407C7E20D4BE84 |
| 154 | D0FD94279FA67EC61A3904C0AD8ACA04 |
| 155 | EA73A9415EC2004D49E9D0F645961C75 |
| 156 | 005AF0614A7552041194DEECBF8DD016 |
| 157 | B514481533DA0A731705B93CF634E40D |
| 158 | 983054521841A6E4FF34B2C07B5684FE |
| 159 | C46D927D0FD2B2F509550025677C6871 |
| 160 | 2AD85C08127487C87ECE014D65169102 |
| 161 | 0F617852FA3930AA7EE74B400B2CC831 |
| 162 | AE9D395004C6E27540C378625D36E0D6 |
| 163 | DC4FA55750F10B0636248F12C212FFE4 |
| 164 | D3602B8D6CBF1809C88B827185631ECF |
| 165 | A94825850708E7723EA8F22C44BF78B2 |
| 166 | A62D231C16AEEFE0B0026B306662945A |
| 167 | 9C7BE810A86465A50551F89125D93B12 |
| 168 | 9712D9338B9CC60485C10172F50F121F |
| 169 | A3902CE0E0B9912591FF28C695728257 |
| 170 | 4167057891AB29473A9E0F67F3658921 |
| 171 | B3368B91EC12A284BC414C8F0D7F8D20 |
| 172 | EE21888101ABF06C1175828CB58B598D |
| 173 | E43923A00ECC32CCC2D162A4A44BD7F4 |
| 174 | CC9E30B8538AD51703EEB6F70801AB22 |
| 175 | B908AD2F1501DA1C156811736CD798CD |
| 176 | 2B46302ACCC2F808797FC648A614326D |
| 177 | 8A54494F1BE27235B8764023AA0FBCFA |
| 178 | BC1041E6F636421E89277DC154439103 |
| 179 | 275B39A63029B974E3561AE0A8FC8032 |
| 180 | 9283F6FE819B80492A22B85CE5CE5DC4 |
| 181 | 4CCB52C0CE058A78022C22DF5788CBCC |
| 182 | B0DF9608DE549A6F6C581516919A81E6 |
| 183 | 2CA185163CC36060D1E85BB0A7FBB988 |
| 184 | 66101D2846155CAC986FC790D2124EFC |
| 185 | 8016E3904644D2093579B83BD7AB5071 |
| 186 | 531CAB7085BEC14257439658023647CF |
| 187 | DF2910165AA5051E41F6EB198E4D491C |
| 188 | BA32052042B0FB2188DE7857DA1B6788 |
| 189 | 9E6D075AFF0EA4153615E140BF380666 |
| 190 | 9ACC5A037902534642A3BE391AA40F9B |
| 191 | 4D741A3B4499843010D7E5FA8988DC80 |
| 192 | FA1421C96EDC6092726154560B1C2FC8 |
| 193 | 882946076223CAE0B0BFE3EDA59826D5 |
| 194 | CEBB288C28B7472A0D3917012276C034 |
| 195 | BD35A6E00C9528DB38289CF823C34F30 |
| 196 | E2C93618B6B2800D51171A5F85746A55 |
| 197 | B43EF39A1A64F0E220AF740F9494291B |
| 198 | AC537817C2612744A58132A8AFBC44A3 |
| 199 | 98A321249A821DDBF81C38235A371A14 |
| 200 | AE1D46069090D81BB6B08FED9E687285 |
| 201 | 7EAE2415DC2CD60AE083249A33B56E05 |
| 202 | 3D942AAA9BC9F27289421CE0B301FB98 |
| 203 | 1548BA6D08530727AC6D059C005C6C42 |
| 204 | FF47C21142C65B502DA70647BAE831D1 |
| 205 | C83AA7FEAC5E51A08091E10DB0C233D9 |
| 206 | E86EDD2EC2DAA3104229EDC43471A16A |
| 207 | 22FAFB9C184B78B56EE91B6602C03244 |
| 208 | E45631DC509B1290C08D2C1A1F15DBFE |
| 209 | D203C51207092B56568FDAD9E2D44473 |
| 210 | 2AA87F31A7D1AB1C90024F936006C4A5 |
| 211 | 913136153593DEABC7305BF0C5A62180 |
| 212 | D8DA5FE401F2758642A082C53A6A5CB8 |
| 213 | 23C2295213147F324DE8EC1C103BAE88 |
| 214 | 883AF097FCDE82B366A1844245E0D727 |
| 215 | 79E5E9F8C933159ACADC22A06F900A70 |
| 216 | FE40502B44A9E44B2C336250D47538CC |
| 217 | 670452E19172C843176F1278FE41D584 |
| 218 | B7EAA436078E6886A3024F593AD57580 |
| 219 | 1044D4CDD7230E7B1953AD1232DF07E2 |
| 220 | 4D821ECAC3D845A2E1011695624576FF |
| 221 | 96622ED2FBD44D1B859D70601999F438 |
| 222 | CCC31C3D6D5B41B8D82FF4522A4C0146 |
| 223 | 4A84F7CD62E0C712980E6A0C89BF394F |
| 224 | 10E56751F000927284DBE174E68ECC4C |
| 225 | A3DE70921356F026E084CFE302A210A9 |
| 226 | B12DA0621B343A8C3FE941A32EA5D571 |
| 227 | D653135DE825A74B743E275C19020C71 |
| 228 | 5CAD301BF846B2EE921D33A3D4BB1220 |
| 229 | 1292445ACBB548C668FC3853578474E6 |
| 230 | B94B4B89C0654688C9E007D9061DF5FE |
| 231 | 75A2C91E76061A8680884E8BFD14A64A |
| 232 | 83726F3070B47ECE21504A5065D74A36 |
| 233 | 964A471444A270840919F7FE07382D14 |
| 234 | A582701EBFCA899B8497088C3560F300 |
| 235 | 64FCB63E21CAC63002D1E09FD1543274 |
| 236 | B1E1C83F689ADF422C865F98D288838A |
| 237 | A06A0D822165D3F3416B47419ECCB547 |
| 238 | 1D2068039A32B7EF728914ECE07CB416 |
| 239 | 64C0CF81F78E8823ECC8661A5295422A |
| 240 | 902A7243F593F2180E5A306A8438E6A9 |
| 241 | A4CCED356D56BF1B41C28E1504301FE8 |
| 242 | 82AE90E2F76B3055A2E3A966025CC01A |
| 243 | 8B90D5A62364E18574145C5895CEFF60 |
| 244 | 43F7EA1AB0D19032551AD9DE21307353 |
| 245 | DD5D8424AC60360B1C14E65815C9B15E |
| 246 | C632A67382ECB2681DFB8525140E2878 |
| 247 | 3A6ACF212B6F8B9C53FF224C2E00C16C |
| 248 | 86A90C267B1171093F362FE5CB14E3A0 |
| 249 | EA262EC36E6589C3BB005426AF2590F4 |
| 250 | 200F03126C5B0D7B901128E7757C5F70 |
| 251 | 68FC090C2221AA98BF0D24E85066EFC2 |
| 252 | 9E26CEC67832FC42A87E92FA1015212E |
| 253 | ACD889634F79506F2582EA03240F2A07 |
| 254 | AA65407E1F4A33BF9A62860A3D6A4CC0 |
| 255 | B1B950AC76A608AA32D04B03C7FF24D3 |

Annex B (informative):  
Generalised Hierarchical Golay Sequences

# B.1 Alternative generation

The generalised hierarchical Golay sequences for the PSC described in 7.1 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 *x1* and *x2* of length *n1* and *n2* respectively using the following formula:

- y(i) = x2(i mod n2) \* x1(i div n2), i = 0 ... (n1\* n2) - 1.

The constituent sequences *x1* and *x2* are chosen to be the following length 16 (i.e. *n1* = *n2* =16) sequences:

- x1 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].

- x2 is a generalised hierarchical sequence using the following formula, selecting s=2 and using the two Golay complementary sequences x3 and x4 as constituent sequences. The length of the sequence x3 and x4 is called n3 respectively n4.

*- x2(i) = x4(i* mod *s + s\*(i* div *sn3)) \* x3((i* div *s)* mod *n3),* *i* = 0 ... (*n3\* n4*) - 1.

- x3 and x4 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 x1,x3 and x4 are defined using the following recursive relation:

*a*0(*k*) = (*k*) and *b*0(*k*) = (*k*);

*an*(*k*) = *an*-1(*k*) + *W(j)n*·*bn*-1(*k*-*D(j)n*);

*bn*(*k*) = *an*-1(*k*) - *W(j)n*·*bn*-1(*k*-*D(j)n*);

*k* = 0, 1, 2, …, 2\*\*N(j) -1;

*n* = 1, 2, …, N*(j)*.

The wanted Golay complementary sequence xj is defined by an  assuming n=N(j). The Kronecker delta function is described by , 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) [D10,D20,D30,D40,D50,D60,D70,D80] = [128, 64, 16, 32, 8, 1, 4, 2].

(c) [W10,W20,W30,W40,W50,W60,W70,W80] = [1, -1, 1, 1, 1, 1, 1, 1].

(d) For n = 4, 6, set b4(k) = a4(k), b6(k) = a6(k).

Annex C (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **TSG #** | **TSG Doc.** | **CR** | **Rev** | **Subject/Comment** | **Old** | **New** |
| 14/01/00 | RAN\_05 | RP-99593 | - |  | Approved at TSG RAN #5 and placed under Change Control | - | 3.0.0 |
| 14/01/00 | RAN\_06 | RP-99696 | 001 | 01 | Primary and Secondary CCPCH in TDD | 3.0.0 | 3.1.0 |
| 14/01/00 | RAN\_06 | RP-99695 | 003 | 1 | Alignment of Terminology Regarding Spreading for TDD Mode | 3.0.0 | 3.1.0 |
| 14/01/00 | RAN\_06 | RP-99696 | 004 | - | Code allocation for Case 3 | 3.0.0 | 3.1.0 |
| 14/01/00 | - | - | - |  | Change history was added by the editor | 3.1.0 | 3.1.1 |
| 31/03/00 | RAN\_07 | RP-000069 | 002 | 3 | Cycling of cell parameters | 3.1.1 | 3.2.0 |
| 31/03/00 | RAN\_07 | RP-000069 | 005 | - | Removal of Synchronisation Case 3 in TDD | 3.1.1 | 3.2.0 |
| 31/03/00 | RAN\_07 | RP-000069 | 006 | 1 | Signal Point Constellation | 3.1.1 | 3.2.0 |
| 03/05/00 | - | - | - | - | Revision marks accepted to create clean version | 3.2.0 | 3.2.1 |
| 26/06/00 | RAN\_08 | RP-000273 | 008 | - | Editorial Modifications for 25.223 | 3.2.1 | 3.3.0 |
| 26/06/00 | RAN\_08 | RP-000273 | 009 | - | Editorial modification of 25.223 | 3.2.1 | 3.3.0 |
| 26/06/00 | RAN\_08 | RP-000273 | 010 | - | Editorial modification of 25.223 | 3.2.1 | 3.3.0 |
| 26/06/00 | RAN\_08 | RP-000273 | 011 | 2 | Editorial modification of 25.223 | 3.2.1 | 3.3.0 |
| 26/06/00 | RAN\_08 | RP-000273 | 012 | 2 | Modified code sets on SCH for cell search in UTRA TDD | 3.2.1 | 3.3.0 |
| 26/06/00 | RAN\_08 | RP-000273 | 013 | 1 | Editorial update of TS25.223 | 3.2.1 | 3.3.0 |
| 23/09/00 | RAN\_09 | RP-000346 | 007 | 1 | Gain Factors for TDD Mode | 3.3.0 | 3.4.0 |
| 23/09/00 | RAN\_09 | RP-000346 | 014 | - | Synchronisation codes | 3.3.0 | 3.4.0 |
| 16/03/01 | RAN\_11 | - | - | - | Approved as Release 4 specification (v4.0.0) at TSG RAN #11 | 3.4.0 | 4.0.0 |
| 16/03/01 | RAN\_11 | RP-010064 | 015 | 1 | Code specific phase offsets for TDD | 3.4.0 | 4.0..0 |
| 16/03/01 | RAN\_11 | RP-010073 | 016 | - | Cell synchronisation codes for R'4 Node B sync over air interface in UTRA TDD | 3.4.0 | 4.0.0 |
| 16/03/01 | RAN\_11 | RP-010071 | 017 | 1 | Inclusion of 1.28Mcps TDD in TS 25.223 | 3.4.0 | 4.0.0 |
| 15/06/01 | RAN\_12 | RP-010337 | 019 | - | Addition to the abbreviation list and definition of a constant | 4.0.0 | 4.1.0 |
| 21/09/01 | RAN\_13 | RP-010524 | 021 | 1 | Clarification of notations in TS25.221 and TS25.223 | 4.1.0 | 4.2.0 |
| 21/09/01 | RAN\_13 | RP-010530 | 022 | 1 | Clarification of notations in TS25.221 and TS25.223 | 4.1.0 | 4.2.0 |
| 14/12/01 | RAN\_14 | RP-010748 | 023 | - | A correction of Figure 7 in subclause 7.7.2 of TS 25.223 | 4.2.0 | 4.3.0 |
| 08/03/03 | RAN\_15 | RP-020051 | 025 | 1 | Removal of quantisation of bj gain factor when calculated from a reference TFC | 4.3.0 | 4.4.0 |
| 08/03/03 | RAN\_15 | RP-020051 | 028 | - | Channelisation code-specific multiplier operation under autonomous SF change | 4.3.0 | 4.4.0 |
| 08/03/03 | RAN\_15 | RP-020051 | 030 | - | Alignment of gamma(i) gains of 25.223 with SIR target of WG2 25.331 | 4.3.0 | 4.4.0 |
| 08/03/03 | RAN\_15 | RP-020058 | 026 | 1 | CR to include HSDPA in TS25.223 | 4.4.0 | 5.0.0 |
| 07/06/02 | RAN\_16 | RP-020317 | 031 | - | Correction of SPC for 16QAM in TDD | 5.0.0 | 5.1.0 |
| 22/12/02 | RAN\_18 | RP-020852 | 033 | - | Editorial modification to the section numberings | 5.1.0 | 5.2.0 |
| 25/03/03 | RAN\_19 | RP-030140 | 034 | 3 | Miscellaneous Corrections | 5.2.0 | 5.3.0 |
| 13/01/04 | RAN\_22 | - | - | - | Created for M.1457 update | 5.3.0 | 6.0.0 |
| 12/12/05 | RAN\_30 | RP-050728 | 0037 | - | Correction to 16QAM modulation function | 6.0.0 | 6.1.0 |
| 20/03/06 | RAN\_31 | RP-060079 | 0038 | - | Introduction of 7.68Mcps TDD option | 6.1.0 | 7.0.0 |
| 12/06/06 | RAN\_32 | RP-060295 | 0040 | - | Correction of the values of weight factors | 7.0.0 | 7.1.0 |
| 29/09/06 | RAN\_33 | RP-060492 | 0041 | - | Introduction of E-DCH for 3.84Mcps and 7.68Mcps TDD | 7.1.0 | 7.2.0 |
| 13/03/07 | RAN\_35 | RP-070118 | 0042 | 1 | Introduction of E-DCH for 1.28Mcps TDD | 7.2.0 | 7.3.0 |
| 30/05/07 | RAN\_36 | RP-070385 | 0043 | 1 | Support for MBSFN operation | 7.3.0 | 7.4.0 |
| 30/05/07 | RAN\_36 | RP-070386 | 0045 | - | Support for 1.28Mcps TDD MBSFN operation | 7.3.0 | 7.4.0 |
| 11/09/07 | RAN\_37 | RP-070650 | 0046 | 1 | Introduction of multi-frequency operation for 1.28Mcps TDD | 7.4.0 | 7.5.0 |
| 27/11/07 | RAN\_38 | RP-070943 | 0049 | 1 | More improvement on dedicated carrier for 1.28Mcps TDD MBMS | 7.5.0 | 7.6.0 |
| 04/03/08 | RAN\_39 | - | - | - | Creation of Release 8 further to RAN\_39 decision | 7.6.0 | 8.0.0 |
| 28/05/08 | RAN\_40 | RP-080356 | 0052 | - | Introduction the 64QAM constellation for 1.28 Mcps TDD HSDPA | 8.0.0 | 8.1.0 |
| 28/05/08 | RAN\_40 | RP-080347 | 0054 | - | Correction of uplink multicode capability for 1.28 Mcps TDD EUL | 8.0.0 | 8.1.0 |
| 03/12/08 | RAN\_42 | RP-081118 | 0056 | 1 | Support for 3.84 Mcps MBSFN IMB operation | 8.1.0 | 8.2.0 |
| 03/03/09 | RAN\_43 | RP-090239 | 0057 | - | TFCI for Secondary CCPCH frame type 2 with 16QAM | 8.2.0 | 8.3.0 |
| 03/03/09 | RAN\_43 | RP-090239 | 0058 | - | Specification of scrambling codes and code groups for MBSFN IMB | 8.2.0 | 8.3.0 |
| 26/05/09 | RAN\_44 | RP-090531 | 0059 | - | Minor correction for MBSFN IMB | 8.3.0 | 8.4.0 |
| 01/12/09 | RAN\_46 | - | - | - | Creation of Release 9 (v9.0.0) at RAN#46 | 8.4.0 | 9.0.0 |
| 07/12/10 | RAN\_50 | RP-101317 | 0062 | 2 | Introduction of MC-HSUPA for 1.28Mcps TDD | 9.0.0 | 10.0.0 |
| 07/12/10 | RAN\_50 | RP-101319 | 0063 | 1 | Introduction of MU-MIMO for 1.28Mcps TDD | 9.0.0 | 10.0.0 |
| 15/09/11 | RAN\_53 | RP-111225 | 0066 | 2 | Clarification of 64QAM for LCR TDD | 10.0.0 | 10.1.0 |
| 2012-09 | SP\_57 | - | - | - | Update to Rel-11 version (MCC) | 10.1.0 | 11.0.0 |
| 2014-09 | SP\_65 | - | - | - | Update to Rel-12 version (MCC) | 11.0.0 | 12.0.0 |
| 2015-12 | SP\_70 | - | - | - | Update to Rel-13 version (MCC) | 12.0.0 | 13.0.0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2017-03 | RP-75 | - | - | - | - | Promotion to Release 14 without technical change (MCC) | 14.0.0 |
| 2018-06 | RP-80 | - | - | - | - | Promotion to Release 15 without technical change (MCC) | 15.0.0 |
| 2020-07 | RP-88e | - | - | - | - | Upgrade to Rel-16 version without technical change | 16.0.0 |