

# ETSI TS 125 222 V7.4.0 (2007-10)

---

*Technical Specification*

**Universal Mobile Telecommunications System (UMTS);  
Multiplexing and channel coding (TDD)  
(3GPP TS 25.222 version 7.4.0 Release 7)**

---



---

**Reference**

RTS/TSGR-0125222v740

---

**Keywords**

UMTS

**ETSI**

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

---

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

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

---

**Important notice**

---

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

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

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

---

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

© European Telecommunications Standards Institute 2007.  
All rights reserved.

**DECT**<sup>TM</sup>, **PLUGTESTS**<sup>TM</sup> and **UMTS**<sup>TM</sup> are Trade Marks of ETSI registered for the benefit of its Members.  
**TIPHON**<sup>TM</sup> and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.  
**3GPP**<sup>TM</sup> is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

---

## Intellectual Property Rights

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

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

---

## Foreword

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

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

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

# Contents

Intellectual Property Rights .....	2
Foreword.....	2
Foreword.....	7
1 Scope .....	8
2 References .....	8
3 Definitions, symbols and abbreviations .....	8
3.1 Definitions .....	8
3.2 Symbols.....	9
3.3 Abbreviations .....	9
4 Multiplexing, channel coding and interleaving .....	11
4.1 General .....	11
4.2 General coding/multiplexing of TrCHs.....	11
4.2.1 CRC attachment.....	14
4.2.1.1 CRC calculation .....	14
4.2.1.2 Relation between input and output of the CRC attachment block.....	14
4.2.2 Transport block concatenation and code block segmentation.....	15
4.2.2.1 Concatenation of transport blocks.....	15
4.2.2.2 Code block segmentation .....	15
4.2.3 Channel coding .....	16
4.2.3.1 Convolutional coding .....	17
4.2.3.2 Turbo coding.....	18
4.2.3.2.1 Turbo coder .....	18
4.2.3.2.2 Trellis termination for Turbo coder .....	19
4.2.3.2.3 Turbo code internal interleaver.....	19
4.2.3.2.3.1 Bits-input to rectangular matrix with padding .....	20
4.2.3.3 Concatenation of encoded blocks.....	23
4.2.4 Radio frame size equalisation .....	23
4.2.5 1st interleaving.....	23
4.2.5.1 Relation between input and output of 1 <sup>st</sup> interleaving.....	24
4.2.6 Radio frame segmentation .....	24
4.2.7 Rate matching .....	25
4.2.7.1 Determination of rate matching parameters .....	26
4.2.7.1.1 Uncoded and convolutionally encoded TrCHs .....	27
4.2.7.1.2 Turbo encoded TrCHs .....	27
4.2.7.2 Bit separation and collection for rate matching.....	28
4.2.7.2.1 Bit separation.....	30
4.2.7.2.2 Bit collection .....	31
4.2.7.3 Rate matching pattern determination .....	31
4.2.8 TrCH multiplexing.....	32
4.2.9 Bit Scrambling .....	33
4.2.10 Physical channel segmentation .....	33
4.2.11 2nd interleaving .....	33
4.2.11.1 Frame related 2nd interleaving.....	34
4.2.11.2 Timeslot related 2 <sup>nd</sup> interleaving .....	35
4.2.11A Sub-frame segmentation for the 1.28 Mcps option.....	37
4.2.12 Physical channel mapping .....	37
4.2.12.1 Physical channel mapping for the 3.84 Mcps and 7.68Mcps options .....	37
4.2.12.1.1 Mapping scheme.....	38
4.2.12.2 Physical channel mapping for the 1.28 Mcps option .....	39
4.2.12.2.1 Mapping scheme.....	39
4.2.13 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels .....	40
4.2.13.1 Allowed CCTrCH combinations for one UE .....	41

4.2.13.1.1	Allowed CCTrCH combinations on the uplink .....	41
4.2.13.1.2	Allowed CCTrCH combinations on the downlink .....	41
4.2.14	Transport format detection.....	41
4.2.14.1	Blind transport format detection .....	41
4.2.14.2	Explicit transport format detection based on TFCI .....	41
4.2.14.2.1	Transport Format Combination Indicator (TFCI).....	41
4.3	Coding for layer 1 control for the 3.84 Mcps and 7.68Mcps TDD options.....	42
4.3.1	Coding of transport format combination indicator (TFCI) .....	42
4.3.1.1	Coding of long TFCI lengths .....	42
4.3.1.2	Coding of short TFCI lengths.....	43
4.3.1.2.1	Coding very short TFCIs by repetition.....	43
4.3.1.2.2	Coding short TFCIs using bi-orthogonal codes.....	43
4.3.1.3	Mapping of TFCI code word.....	44
4.3.2	Coding and Bit Scrambling of the Paging Indicator .....	44
4.3.3	Coding and Bit Scrambling of the MBMS Notification Indicator .....	45
4.4	Coding for layer 1 control for the 1.28 Mcps option.....	45
4.4.1	Coding of transport format combination indicator (TFCI) for QPSK and 16QAM.....	45
4.4.1.1	Mapping of TFCI code word.....	45
4.4.2	Coding of transport format combination indicator (TFCI) for 8PSK .....	47
4.4.2.1	Coding of long TFCI lengths .....	47
4.4.2.2	Coding of short TFCI lengths.....	49
4.4.2.2.1	Coding very short TFCIs by repetition.....	49
4.4.2.2.2	Coding short TFCIs using bi-orthogonal codes.....	49
4.4.2.3	Mapping of TFCI code word.....	50
4.4.3	Coding and Bit Scrambling of the Paging Indicator .....	51
4.4.4	Coding of the Fast Physical Access Channel (FPACH) information bits .....	51
4.4.5	Coding and Bit Scrambling of the MBMS Notification Indicator .....	52
4.4.6	Coding of PLCCCH .....	52
4.5	Coding for HS-DSCH .....	53
4.5.1	CRC attachment for HS-DSCH .....	55
4.5.2	Code block segmentation for HS-DSCH .....	55
4.5.3	Channel coding for HS-DSCH.....	56
4.5.4	Hybrid ARQ for HS-DSCH.....	56
4.5.4.1	HARQ bit separation.....	56
4.5.4.2	HARQ First Rate Matching Stage.....	56
4.5.4.3	HARQ Second Rate Matching Stage .....	57
4.5.4.4	HARQ bit collection .....	58
4.5.5	Bit scrambling.....	58
4.5.6	Interleaving for HS-DSCH .....	58
4.5.7	Constellation re-arrangement for 16 QAM.....	60
4.5.8	Physical channel mapping for HS-DSCH.....	60
4.6	Coding/Multiplexing for HS-SCCH.....	61
4.6.1	HS-SCCH information field mapping .....	63
4.6.1.1	Channelisation code set information mapping .....	63
4.6.1.1.1	1.28Mcps TDD and 3.84Mcps TDD.....	63
4.6.1.1.2	7.68Mcps TDD.....	63
4.6.1.2	Timeslot information mapping.....	64
4.6.1.2.1	1.28 Mcps TDD.....	64
4.6.1.2.2	3.84 Mcps TDD and 7.68Mcps TDD .....	64
4.6.1.3	Modulation scheme information mapping .....	65
4.6.1.4	Redundancy and constellation version information mapping .....	65
4.6.1.5	HS-SCCH cyclic sequence number.....	65
4.6.1.6	UE identity .....	65
4.6.1.7	HARQ process identifier mapping.....	66
4.6.1.8	Transport block size index mapping .....	66
4.6.2	Multiplexing of HS-SCCH information.....	66
4.6.3	CRC attachment for HS-SCCH .....	66
4.6.4	Channel coding for HS-SCCH.....	66
4.6.5	Rate matching for HS-SCCH.....	67
4.6.6	Interleaving for HS-SCCH.....	67
4.6.7	Physical Channel Segmentation for HS-SCCH .....	67
4.6.8	Physical channel mapping for HS-SCCH .....	67

4.7	Coding for HS-SICH .....	67
4.7.1	HS-SICH information field mapping .....	68
4.7.1.1	RMF information mapping .....	68
4.7.1.2	RTBS information mapping .....	68
4.7.1.3	ACK/NACK information mapping .....	68
4.7.2	Coding for HS-SICH .....	69
4.7.2.1	Field Coding of ACK/NACK .....	69
4.7.2.2	Field Coding of CQI .....	69
4.7.2.2.1	Field Coding of CQI for 1.28 Mcps TDD .....	69
4.7.2.2.2	Field Coding of CQI for 3.84 Mcps TDD and 7.68Mcps TDD .....	70
4.7.3	Multiplexing of HS-SICH information fields .....	70
4.7.4	Interleaver for HS-SICH .....	70
4.7.5	Physical channel mapping for HS-SICH .....	70
4.8	Coding for E-DCH .....	70
4.8.1	CRC attachment for E-DCH .....	72
4.8.2	Code block segmentation for E-DCH .....	72
4.8.3	Channel coding for E-DCH .....	73
4.8.4	Physical layer HARQ functionality and rate matching for E-DCH .....	73
4.8.4.1	Determination of SF, modulation and number of physical channels .....	73
4.8.4.2	HARQ bit separation .....	73
4.8.4.3	HARQ Rate Matching Stage .....	73
4.8.4.4	HARQ bit collection .....	74
4.8.5	Bit scrambling .....	74
4.8.6	Interleaving for E-DCH .....	74
4.8.7	Constellation re-arrangement for 16 QAM .....	74
4.8.8	Physical channel mapping for E-DCH .....	74
4.9	Coding for E-UCCH .....	75
4.9.1	Coding for E-UCCH for the 3.84Mcps and 7.68Mcps TDD options .....	75
4.9.1.1	Overview .....	75
4.9.1.2	E-UCCH part 1 .....	76
4.9.1.2.1	Information field mapping of E-TFCI .....	76
4.9.1.2.2	Channel coding for E-UCCH part 1 .....	76
4.9.1.2.3	Physical channel mapping for E-UCCH part 1 .....	76
4.9.1.3	E-UCCH part 2 .....	76
4.9.1.3.1	Information field mapping of retransmission sequence number .....	76
4.9.1.3.2	Information field mapping of HARQ process ID .....	76
4.9.1.3.3	Multiplexing of E-UCCH part 2 information .....	77
4.9.1.3.4	Channel coding for E-UCCH part 2 .....	77
4.9.1.3.5	Physical channel mapping for E-UCCH part 2 .....	77
4.9.2	Coding for E-UCCH for the 1.28Mcps TDD option .....	77
4.9.2.1	E-UCCH information field mapping .....	78
4.9.2.1.1	Information field mapping of E-TFCI .....	78
4.9.2.1.2	RSN information mapping .....	78
4.9.2.1.3	HARQ information mapping .....	79
4.9.2.2	Multiplexing for E-UCCH .....	79
4.9.2.3	Coding for E-UCCH .....	79
4.9.2.4	Physical channel mapping for E-UCCH .....	79
4.10	Coding for E-AGCH .....	80
4.10.1	Information Field Mapping .....	81
4.10.1.1	Mapping of the Absolute Grant (Power) Value .....	81
4.10.1.2	Mapping of the Code Resource Related Information .....	82
4.10.1.3	Mapping of the Timeslot Resource Related Information .....	83
4.10.1.4	Mapping of the E-AGCH Cyclic Sequence Number (ECSN) .....	83
4.10.1.5	Mapping of the Resource Duration Indicator .....	83
4.10.1.6	Mapping of the E-HICH Indicator (1.28Mcps option only) .....	83
4.10.1.7	Mapping of the E-UCCH Number Indicator (1.28Mcps option only) .....	83
4.10.2	Field Multiplexing .....	83
4.10.3	CRC attachment .....	84
4.10.4	Channel Coding .....	84
4.10.5	Rate Matching .....	85
4.10.6	Interleaving .....	85
4.10.7	Physical Channel Segmentation .....	85

4.10.8	Physical Channel Mapping .....	85
4.11	Coding for E-HICH ACK/NACK .....	85
4.11.1	Coding for E-HICH ACK/NACK for the 3.84Mcps and 7.68Mcps options .....	85
4.11.1.1	Overview .....	85
4.11.1.2	Coding of the HARQ acknowledgement indicator .....	85
4.11.1.3	Bit scrambling of the E-HICH .....	87
4.11.1.4	Physical channel mapping of the E-HICH .....	87
4.11.2	Coding for E-HICH for the 1.28Mcps option only .....	87
4.11.2.1	Overview .....	87
4.11.2.2	Coding of the HARQ acknowledgement indicator and TPC/SS .....	88
4.11.2.3	Bit scrambling and Physical channel mapping of the E-HICH .....	90
4.12	Coding for E-RUCCH .....	90
4.12.1	CRC attachment for E-RUCCH .....	90
4.12.2	Channel coding for E-RUCCH .....	91
4.12.3	Rate matching for E-RUCCH .....	91
4.12.4	Bit scrambling for E-RUCCH .....	91
4.12.5	Interleaving for E-RUCCH .....	91
4.12.6	Physical channel mapping for E-RUCCH .....	91
<b>Annex A (informative):</b>	<b>Change history .....</b>	<b>92</b>
History .....		94

---

# Foreword

This Technical Specification (TS) has been produced by the 3<sup>rd</sup> 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 multiplexing, channel coding and interleaving 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 25.202: "UE capabilities".
- [2] 3GPP TS 25.211: "Transport channels and 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: "Transport channels and physical channels (TDD)".
- [9] 3GPP TS 25.223: "Spreading and modulation (TDD)".
- [10] 3GPP TS 25.224: "Physical layer procedures (TDD)".
- [11] 3GPP TS 25.225: "Measurements".
- [12] 3GPP TS 25.331: "RRC Protocol Specification".
- [13] 3GPP TS 25.308: "High Speed Downlink Packet Access (HSDPA): Overall description (stage 2)".
- [14] ITU-T Recommendation X.691 (12/97) "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)".
- [15] 3GPP TS 25.321: 'Medium Access Control (MAC) protocol specification'

---

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

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

**TrCH number:** The transport channel number identifies a TrCH in the context of L1. The L3 transport channel identity (TrCH ID) maps onto the L1 transport channel number. The mapping between the transport channel number and the TrCH ID is as follows: TrCH 1 corresponds to the TrCH with the lowest TrCH ID, TrCH 2 corresponds to the TrCH with the next lowest TrCH ID and so on.

## 3.2 Symbols

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

$\lceil x \rceil$	round towards $\infty$ , i.e. integer such that $x \leq \lceil x \rceil < x+1$
$\lfloor x \rfloor$	round towards $-\infty$ , i.e. integer such that $x-1 < \lfloor x \rfloor \leq x$
$ x $	absolute value of $x$

Unless otherwise is explicitly stated when the symbol is used, the meaning of the following symbols are:

$i$	TrCH number
$j$	TFC number
$k$	Bit number
$l$	TF number
$m$	Transport block number
$n$	Radio frame number
$p$	PhCH number
$r$	Code block number
$I$	Number of TrCHs in a CCTrCH.
$C_i$	Number of code blocks in one TTI of TrCH $i$ .
$F_i$	Number of radio frames in one TTI of TrCH $i$ .
$M_i$	Number of transport blocks in one TTI of TrCH $i$ .
$N_{TFCI \text{ code word}}$	Number of TFCI code word bits after TFCI encoding
$P$	Number of PhCHs used for one CCTrCH.
$PL$	Puncturing Limit. Signalled from higher layers
$RM_i$	Rate Matching attribute for TrCH $i$ . Signalled from higher layers.

Temporary variables, i.e. variables used in several (sub)clauses with different meaning.

$x, X$   
 $y, Y$   
 $z, Z$

## 3.3 Abbreviations

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

<ACRONYM>	<Explanation>
ARQ	Automatic Repeat on Request
BCH	Broadcast Channel
BER	Bit Error Rate
BS	Base Station
BSS	Base Station Subsystem
CBR	Constant Bit Rate
CCCH	Common Control Channel
CCTrCH	Coded Composite Transport Channel
CDMA	Code Division Multiple Access
CFN	Connection Frame Number
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
DCA	Dynamic Channel Allocation
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
E-AGCH	E-DCH Absolute Grant Channel
ECSN	E-AGCH Cyclic Sequence Number
E-DCH	Enhanced Dedicated Channel
E-HICH	E-DCH Hybrid ARQ Indicator Channel

E-PUCH	E-DCH Physical Uplink Channel
E-RUCCH	E-DCH Random Access Uplink Control Channel
E-TFCI	E-DCH Transport Format Combination Indicator
E-UCCH	E-DCH Uplink Control Channel
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Control
FER	Frame Error Rate
GF	Galois Field
HARQ	Hybrid Automatic Repeat reQuest
HS-DSCH	High Speed Downlink Shared Channel
HS-PDSCH	High Speed Physical Downlink Shared Channel
HS-SCCH	Shared Control Channel for HS-DSCH
HS-SICH	Shared Information Channel for HS-DSCH
JD	Joint Detection
L1	Layer 1
L2	Layer 2
LLC	Logical Link Control
MA	Multiple Access
MAC	Medium Access Control
MBSFN	MBMS over a Single Frequency Network
MICH	MBMS Indicator Channel
MS	Mobile Station
MT	Mobile Terminated
NRT	Non-Real Time
OVSF	Orthogonal Variable Spreading Factor
PC	Power Control
PCCC	Parallel Concatenated Convolutional Code
PICH	Paging Indicator Channel
PCH	Paging Channel
PhCH	Physical Channel
PI	Paging Indicator (value calculated by higher layers)
PLCCH	Physical Layer Common Control Channel
P <sub>q</sub>	Paging Indicator (indicator set by physical layer)
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RACH	Random Access Channel
RF	Radio Frequency
RLC	Radio Link Control
RMF	Recommended Modulation Format
RRC	Radio Resource Control
RRM	Radio Resource Management
RSC	Recursive Systematic Convolutional Coder
RSN	Retransmission Sequence Number
RT	Real Time
RTBS	Recommended Transport Block Size
RU	Resource Unit
RV	Redundancy Version
SCCC	Serial Concatenated Convolutional Code
SCH	Synchronization Channel
SNR	Signal to Noise Ratio
TCH	Traffic channel
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TFRI	Transport Format Resource Indicator
TPC	Transmit Power Control
TrBk	Transport Block
TrCH	Transport Channel
TTI	Transmission Time Interval

UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
USCH	Uplink Shared Channel
UTRA	UMTS Terrestrial Radio Access
VBR	Variable Bit Rate

---

## 4 Multiplexing, channel coding and interleaving

### 4.1 General

Data stream from/to MAC and higher layers (Transport block / Transport block set) is encoded/decoded to offer transport services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting (including rate matching), and interleaving and transport channels mapping onto/splitting from physical channels.

In the UTRA-TDD mode, the total number of basic physical channels (a certain time slot one spreading code on a certain carrier frequency) per frame is given by the maximum number of time slots and the maximum number of CDMA codes per time slot.

### 4.2 General coding/multiplexing of TrCHs

This section only applies to the transport channels: DCH, RACH, DSCH, USCH, BCH, FACH and PCH. Other transport channels which do not use the general method are described separately below.

Figure 1 illustrates the overall concept of transport-channel coding and multiplexing. Data arrives to the coding/multiplexing unit in form of transport block sets, once every transmission time interval. The transmission time interval is transport-channel specific from the set {5 ms<sup>(\*)</sup>, 10 ms, 20 ms, 40 ms, 80 ms}.

Note: <sup>(\*)</sup> may be applied for PRACH for 1.28 Mcps TDD

The following coding/multiplexing steps can be identified:

- add CRC to each transport block (see subclause 4.2.1);
- TrBk concatenation / Code block segmentation (see subclause 4.2.2);
- channel coding (see subclause 4.2.3);
- radio frame size equalization (see subclause 4.2.4);
- interleaving (two steps, see subclauses 4.2.5 and 4.2.11);
- radio frame segmentation (see subclause 4.2.6);
- rate matching (see subclause 4.2.7);
- multiplexing of transport channels (see subclause 4.2.8);
- bit scrambling (see subclause 4.2.9);
- physical channel segmentation (see subclause 4.2.10);
- sub-frame segmentation(see subclause 4.2.11A only for 1.28Mcps TDD)
- mapping to physical channels (see subclause 4.2.12).

The coding/multiplexing steps for uplink and downlink are shown in figures 1 and 1A.

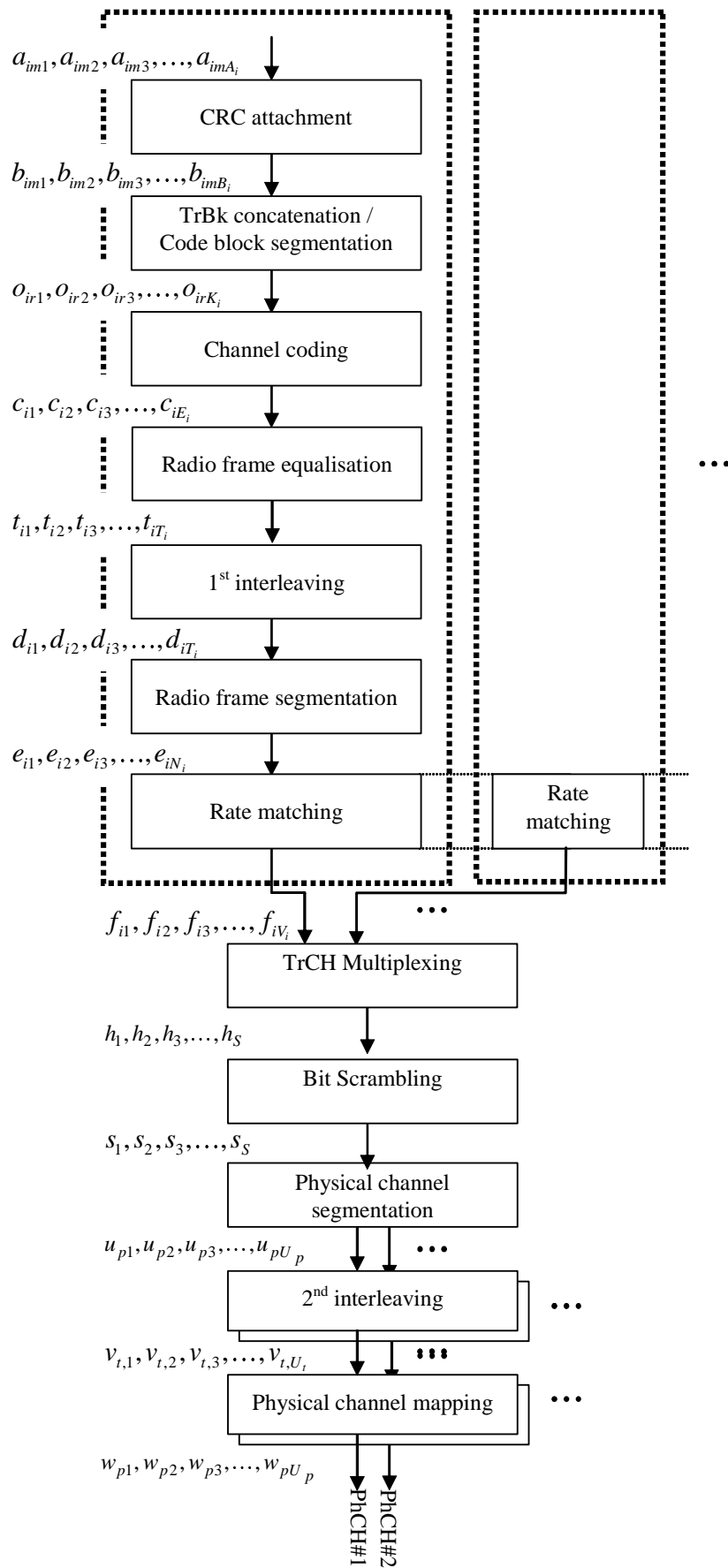


Figure 1: Transport channel multiplexing structure for uplink and downlink for 3.84Mcps and 7.68 Mcps TDD

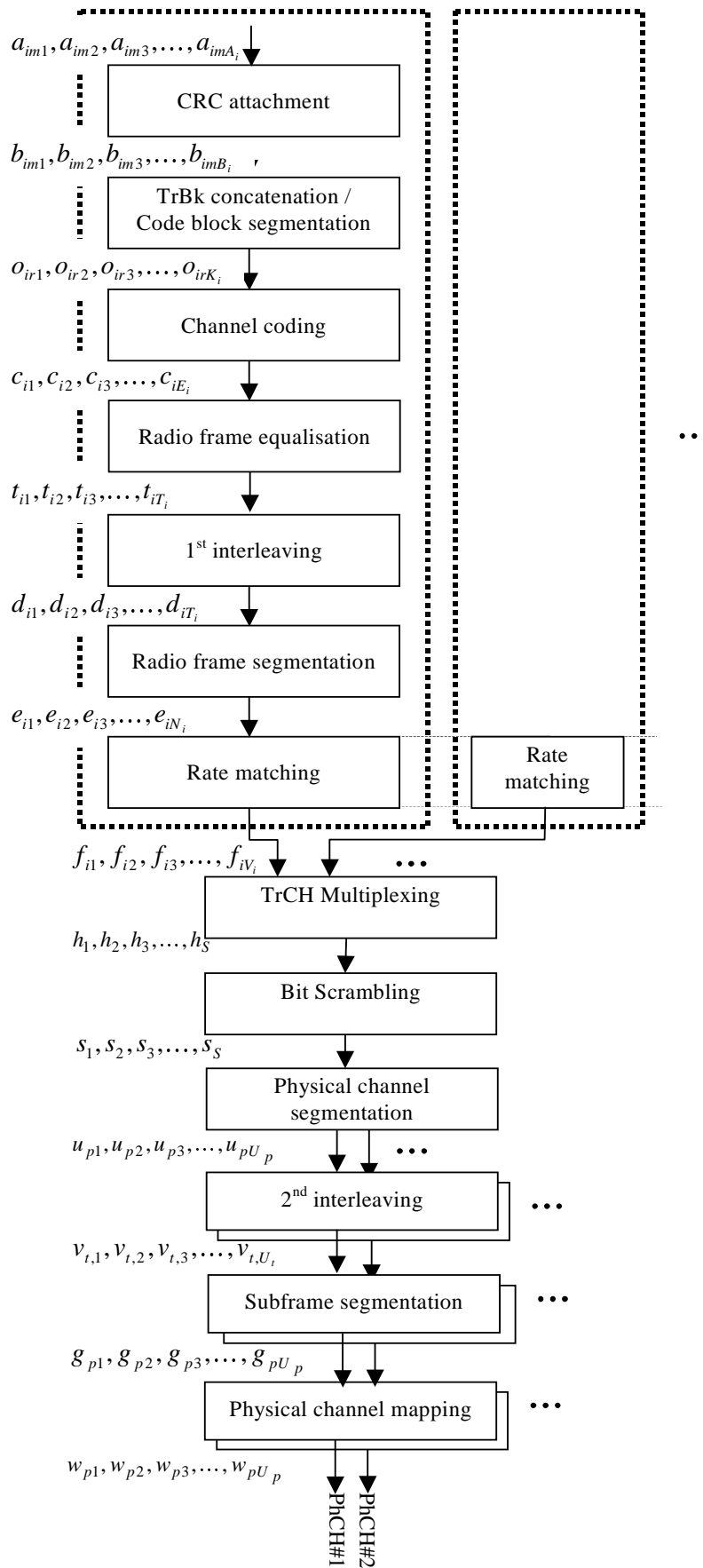


Figure 1A: Transport channel multiplexing structure for uplink and downlink of 1.28Mcps TDD

Primarily, transport channels are multiplexed as described above, i.e. into one data stream mapped on one or several physical channels. However, an alternative way of multiplexing services is to use multiple CCTrCHs (Coded Composite Transport Channels), which corresponds to having several parallel multiplexing chains as in figures 1 and 1A, resulting in several data streams, each mapped to one or several physical channels.

## 4.2.1 CRC attachment

Error detection is provided on transport blocks through a Cyclic Redundancy Check (CRC). The size of the CRC is 24, 16, 12, 8 or 0 bits and it is signalled from higher layers what CRC size that should be used for each transport channel.

### 4.2.1.1 CRC calculation

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

$$g_{\text{CRC24}}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$$

$$g_{\text{CRC16}}(D) = D^{16} + D^{12} + D^5 + 1$$

$$g_{\text{CRC12}}(D) = D^{12} + D^{11} + D^3 + D^2 + D + 1$$

$$g_{\text{CRC8}}(D) = D^8 + D^7 + D^4 + D^3 + D + 1$$

Denote the bits in a transport block delivered to layer 1 by  $a_{im1}, a_{im2}, a_{im3}, \dots, a_{imA_i}$ , and the parity bits by  $p_{im1}, p_{im2}, p_{im3}, \dots, p_{imL_i}$ .  $A_i$  is the size of a transport block of TrCH  $i$ ,  $m$  is the transport block number, and  $L_i$  is the number of parity bits.  $L_i$  can take the values 24, 16, 12, 8, or 0 depending on what is signalled from higher layers.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_{im1}D^{A_i+23} + a_{im2}D^{A_i+22} + \dots + a_{imA_i}D^{24} + p_{im1}D^{23} + p_{im2}D^{22} + \dots + p_{im23}D^1 + p_{im24}$$

yields a remainder equal to 0 when divided by  $g_{\text{CRC24}}(D)$ , polynomial:

$$a_{im1}D^{A_i+15} + a_{im2}D^{A_i+14} + \dots + a_{imA_i}D^{16} + p_{im1}D^{15} + p_{im2}D^{14} + \dots + p_{im15}D^1 + p_{im16}$$

yields a remainder equal to 0 when divided by  $g_{\text{CRC16}}(D)$ , polynomial:

$$a_{im1}D^{A_i+11} + a_{im2}D^{A_i+10} + \dots + a_{imA_i}D^{12} + p_{im1}D^{11} + p_{im2}D^{10} + \dots + p_{im11}D^1 + p_{im12}$$

yields a remainder equal to 0 when divided by  $g_{\text{CRC12}}(D)$  and the polynomial:

$$a_{im1}D^{A_i+7} + a_{im2}D^{A_i+6} + \dots + a_{imA_i}D^8 + p_{im1}D^7 + p_{im2}D^6 + \dots + p_{im7}D^1 + p_{im8}$$

yields a remainder equal to 0 when divided by  $g_{\text{CRC8}}(D)$ .

If no transport blocks are input to the CRC calculation ( $M_i = 0$ ), no CRC attachment shall be done. If transport blocks are input to the CRC calculation ( $M_i \neq 0$ ) and the size of a transport block is zero ( $A_i = 0$ ), CRC shall be attached, i.e. all parity bits equal to zero.

### 4.2.1.2 Relation between input and output of the CRC attachment block

The bits after CRC attachment are denoted by  $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$ , where  $B_i = A_i + L_i$ . The relation between  $a_{imk}$  and  $b_{imk}$  is:

$$b_{imk} = a_{imk} \quad k = 1, 2, 3, \dots, A_i$$

$$b_{imk} = p_{im(L_i+1-(k-A_i))} \quad k = A_i + 1, A_i + 2, A_i + 3, \dots, A_i + L_i$$

## 4.2.2 Transport block concatenation and code block segmentation

All transport blocks in a TTI are serially concatenated. If the number of bits in a TTI is larger than the maximum size of a code block, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional, turbo coding or no coding is used for the TrCH.

### 4.2.2.1 Concatenation of transport blocks

The bits input to the transport block concatenation are denoted by  $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$  where  $i$  is the TrCH number,  $m$  is the transport block number, and  $B_i$  is the number of bits in each block (including CRC). The number of transport blocks on TrCH  $i$  is denoted by  $M_i$ . The bits after concatenation are denoted by  $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$ , where  $i$  is the TrCH number and  $X_i = M_i B_i$ . They are defined by the following relations:

$$x_{ik} = b_{i1k} \quad k = 1, 2, \dots, B_i$$

$$x_{ik} = b_{i,2,(k-B_i)} \quad k = B_i + 1, B_i + 2, \dots, 2B_i$$

$$x_{ik} = b_{i,3,(k-2B_i)} \quad k = 2B_i + 1, 2B_i + 2, \dots, 3B_i$$

...

$$x_{ik} = b_{i,M_i,(k-(M_i-1)B_i)} \quad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \dots, M_i B_i$$

### 4.2.2.2 Code block segmentation

Segmentation of the bit sequence from transport block concatenation is performed if  $X_i > Z$ . The code blocks after segmentation are of the same size. The number of code blocks on TrCH  $i$  is denoted by  $C_i$ . If the number of bits input to the segmentation,  $X_i$ , is not a multiple of  $C_i$ , filler bits are added to the beginning of the first block. If turbo coding is selected and  $X_i < 40$ , filler bits are added to the beginning of the code block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

- convolutional coding:  $Z = 504$ ;
- turbo coding:  $Z = 5114$ ;
- no channel coding:  $Z = \textit{unlimited}$ .

The bits output from code block segmentation, for  $C_i \neq 0$ , are denoted by  $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$ , where  $i$  is the TrCH number,  $r$  is the code block number, and  $K_i$  is the number of bits per code block.

Number of code blocks:

$$C_i = \begin{cases} \lceil X_i / Z \rceil & \text{when } Z \neq \textit{unlimited} \\ 0 & \text{when } Z = \textit{unlimited} \text{ and } X_i = 0 \\ 1 & \text{when } Z = \textit{unlimited} \text{ and } X_i \neq 0 \end{cases}$$

Number of bits in each code block (applicable for  $C_i \neq 0$  only):

if  $X_i < 40$  and Turbo coding is used, then

$$K_i = 40$$

else

$$K_i = \lceil X_i / C_i \rceil$$

end if



Number of filler bits:  $Y_i = C_i K_i - X_i$

for  $k = 1$  to  $Y_i$  -- Insertion of filler bits

$$o_{ilk} = 0$$

end for

for  $k = Y_i + 1$  to  $K_i$

$$o_{ilk} = x_{i,(k-Y_i)}$$

end for

$r = 2$  -- Segmentation

while  $r \leq C_i$

for  $k = 1$  to  $K_i$

$$o_{irk} = x_{i,(k+(r-1) \cdot K_i - Y_i)}$$

end for

$r = r + 1$

end while

### 4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by  $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$ , where  $i$  is the TrCH number,  $r$  is the code block number, and  $K_i$  is the number of bits in each code block. The number of code blocks on TrCH  $i$  is denoted by  $C_i$ . After encoding the bits are denoted by  $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$ , where  $Y_i$  is the number of encoded bits. The relation between  $o_{irk}$  and  $y_{irk}$  and between  $K_i$  and  $Y_i$  is dependent on the channel coding scheme.

The following channel coding schemes can be applied to transport channels:

- convolutional coding;
- turbo coding;
- no coding.

Usage of coding scheme and coding rate for the different types of TrCH is shown in tables 1 and 1A. The values of  $Y_i$  in connection with each coding scheme:

- convolutional coding with rate 1/2:  $Y_i = 2 \cdot K_i + 16$ ; rate 1/3:  $Y_i = 3 \cdot K_i + 24$ ;
- turbo coding with rate 1/3:  $Y_i = 3 \cdot K_i + 12$ ;
- no coding:  $Y_i = K_i$ .

**Table 1: Usage of channel coding scheme and coding rate for 3.84Mcps TDD**

Type of TrCH	Coding scheme	Coding rate
BCH	Turbo coding	1/3
PCH	Convolutional coding	1/2
RACH		
DCH, DSCH, FACH, USCH	Turbo coding	1/3, 1/2
	Turbo coding	1/3
	No coding	

Note: Rate 1/3 Turbo coding is only applied to BCH when the beacon timeslot uses burst type 4; rate 1/2 convolutional coding is only applied to BCH when the beacon timeslot uses burst type 1. For the MBSFN FACH, only rate 1/3 Turbo coding shall be applied.

**Table 1A: Usage of channel coding scheme and coding rate for 1.28Mcps TDD**

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/3
PCH		1/3, 1/2
RACH		1/2
DCH, DSCH, FACH, USCH		1/3, 1/2
	Turbo coding	1/3
	No coding	

Note: For the MBSFN FACH, only rate 1/3 Turbo coding shall be applied.

#### 4.2.3.1 Convolutional coding

Convolutional codes with constraint length 9 and coding rates 1/3 and 1/2 are defined.

The configuration of the convolutional coder is presented in figure 2.

Output from the rate 1/3 convolutional coder shall be done in the order output 0, output 1, output 2, output 0, output 1, output 2, output 0, ..., output 2. Output from the rate 1/2 convolutional coder shall be done in the order output 0, output 1, output 0, output 1, output 0, ..., output 1.

8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.

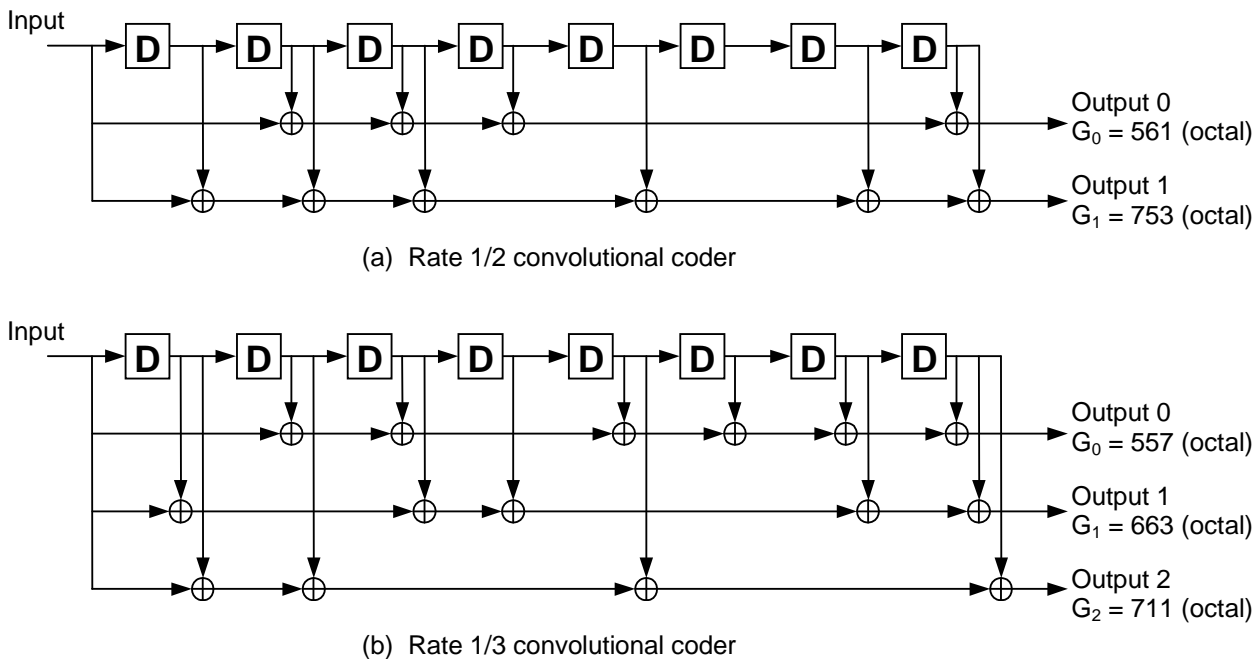


Figure 2: Rate 1/2 and rate 1/3 convolutional coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The scheme of Turbo coder is a Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 3.

The transfer function of the 8-state constituent code for PCCC is:

$$G(D) = \begin{bmatrix} 1, & g_1(D) \\ & g_0(D) \end{bmatrix},$$

where

$$g_0(D) = 1 + D^2 + D^3,$$

$$g_1(D) = 1 + D + D^3.$$

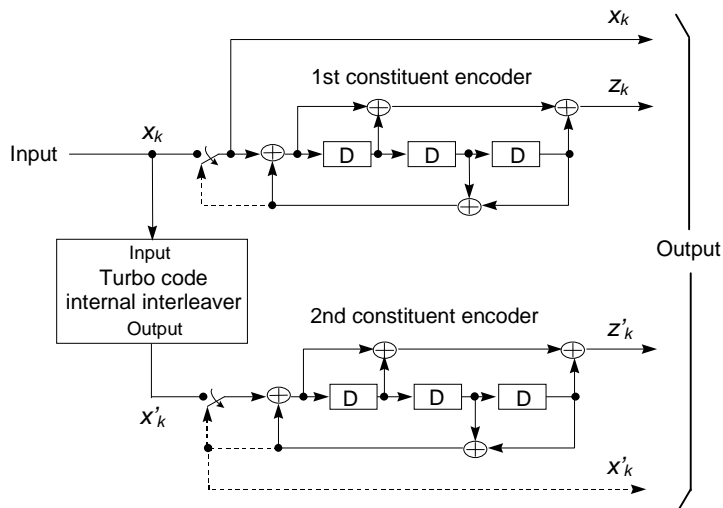
The initial value of the shift registers of the 8-state constituent encoders shall be all zeros when starting to encode the input bits.

Output from the Turbo coder is , Y'(0), X(1), Y(1), Y'(1), etc:

$$x_1, z_1, z'_1, x_2, z_2, z'_2, \dots, x_K, z_K, z'_K,$$

where  $x_1, x_2, \dots, x_K$  are the bits input to the Turbo coder i.e. both first 8-state constituent encoder and Turbo code internal interleaver, and  $K$  is the number of bits, and  $z_1, z_2, \dots, z_K$  and  $z'_1, z'_2, \dots, z'_K$  are the bits output from first and second 8-state constituent encoders, respectively.

The bits output from Turbo code internal interleaver are denoted by  $x'_1, x'_2, \dots, x'_K$ , and these bits are to be input to the second 8-state constituent encoder.



**Figure 3: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)**

4.2.3.2.2 Trellis termination for Turbo coder

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 3 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 3 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be:

$$x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}.$$

4.2.3.2.3 Turbo code internal interleaver

The Turbo code internal interleaver consists of bits-input to a rectangular matrix with padding, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with pruning. The bits input to the Turbo code internal interleaver are denoted by  $x_1, x_2, x_3, \dots, x_K$ , where  $K$  is the integer number of the bits and takes one value of  $40 \leq K \leq 5114$ . The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by  $x_k = o_{irk}$  and  $K = K_i$ .

**The following subclause specific symbols are used in subclauses 4.2.3.2.3.1 to 4.2.3.2.3.3 :**

- $K$  Number of bits input to Turbo code internal interleaver
- $R$  Number of rows of rectangular matrix
- $C$  Number of columns of rectangular matrix
- $p$  Prime number
- $v$  Primitive root
- $\langle s(j) \rangle_{j \in \{0,1,\dots,p-2\}}$  Base sequence for intra-row permutation
- $q_i$  Minimum prime integers
- $r_i$  Permuted prime integers

$\langle T(i) \rangle_{i \in \{0,1,\dots,R-1\}}$	Inter-row permutation pattern
$\langle U_i(j) \rangle_{j \in \{0,1,\dots,C-1\}}$	Intra-row permutation pattern of $i$ -th row
$i$	Index of row number of rectangular matrix
$j$	Index of column number of rectangular matrix
$k$	Index of bit sequence

#### 4.2.3.2.3.1 Bits-input to rectangular matrix with padding

The bit sequence  $x_1, x_2, x_3, \dots, x_K$  input to the Turbo code internal interleaver is written into the rectangular matrix as follows.

- (1) Determine the number of rows of the rectangular matrix,  $R$ , such that:

$$R = \begin{cases} 5, & \text{if } (40 \leq K \leq 159) \\ 10, & \text{if } ((160 \leq K \leq 200) \text{ or } (481 \leq K \leq 530)) \\ 20, & \text{if } (K = \text{any other value}) \end{cases} .$$

The rows of rectangular matrix are numbered 0, 1, ...,  $R - 1$  from top to bottom.

- (2) Determine the prime number to be used in the intra-permutation,  $p$ , and the number of columns of rectangular matrix,  $C$ , such that:

if  $(481 \leq K \leq 530)$  then

$$p = 53 \text{ and } C = p.$$

else

Find minimum prime number  $p$  from table 2 such that

$$K \leq R \times (p + 1),$$

and determine  $C$  such that

$$C = \begin{cases} p - 1 & \text{if } K \leq R \times (p - 1) \\ p & \text{if } R \times (p - 1) < K \leq R \times p \\ p + 1 & \text{if } R \times p < K \end{cases}$$

end if

The columns of rectangular matrix are numbered 0, 1, ...,  $C - 1$  from left to right.

**Table 2: List of prime number  $p$  and associated primitive root  $v$**

$p$	$v$	$p$	$v$	$p$	$v$	$p$	$v$	$p$	$v$
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

(3) Write the input bit sequence  $x_1, x_2, x_3, \dots, x_K$  into the  $R \times C$  rectangular matrix row by row starting with bit  $y_1$  in column 0 of row 0:

$$\begin{bmatrix} y_1 & y_2 & y_3 & \dots & y_C \\ y_{(C+1)} & y_{(C+2)} & y_{(C+3)} & \dots & y_{2C} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{((R-1)C+1)} & y_{((R-1)C+2)} & y_{((R-1)C+3)} & \dots & y_{R \times C} \end{bmatrix}$$

where  $y_k = x_k$  for  $k = 1, 2, \dots, K$  and if  $R \times C > K$ , the dummy bits are padded such that  $y_k = 0$  or  $1$  for  $k = K + 1, K + 2, \dots, R \times C$ . These dummy bits are pruned away from the output of the rectangular matrix after intra-row and inter-row permutations.

**4.2.3.2.3.2 Intra-row and inter-row permutations**

After the bits-input to the  $R \times C$  rectangular matrix, the intra-row and inter-row permutations for the  $R \times C$  rectangular matrix are performed stepwise by using the following algorithm with steps (1) – (6).

(1) Select a primitive root  $v$  from table 2 in section 4.2.3.2.3.1, which is indicated on the right side of the prime number  $p$ .

(2) Construct the base sequence  $\langle s(j) \rangle_{j \in \{0,1,\dots,p-2\}}$  for intra-row permutation as:

$$s(j) = (v \times s(j-1)) \bmod p, \quad j = 1, 2, \dots, (p-2), \text{ and } s(0) = 1.$$

(3) Assign  $q_0 = 1$  to be the first prime integer in the sequence  $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$ , and determine the prime integer  $q_i$  in the sequence  $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$  to be a least prime integer such that  $\text{g.c.d}(q_i, p-1) = 1, q_i > 6$ , and  $q_i > q_{(i-1)}$  for each  $i = 1, 2, \dots, R-1$ . Here g.c.d. is greatest common divisor.

(4) Permute the sequence  $\langle q_i \rangle_{i \in \{0,1,\dots,R-1\}}$  to make the sequence  $\langle r_i \rangle_{i \in \{0,1,\dots,R-1\}}$  such that

$$r_{T(i)} = q_i, \quad i = 0, 1, \dots, R-1,$$

where  $\langle T(i) \rangle_{i \in \{0,1,\dots,R-1\}}$  is the inter-row permutation pattern defined as the one of the four kind of patterns, which are shown in table 3, depending on the number of input bits  $K$ .

**Table 3: Inter-row permutation patterns for Turbo code internal interleaver**

Number of input bits $K$	Number of rows $R$	Inter-row permutation patterns $\langle T(0), T(1), \dots, T(R-1) \rangle$
$(40 \leq K \leq 159)$	5	$\langle 4, 3, 2, 1, 0 \rangle$
$(160 \leq K \leq 200)$ or $(481 \leq K \leq 530)$	10	$\langle 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 \rangle$

$(2281 \leq K \leq 2480)$ or $(3161 \leq K \leq 3210)$	20	<19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10>
$K = \text{any other value}$	20	<19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11>

(5) Perform the  $i$ -th ( $i = 0, 1, \dots, R - 1$ ) intra-row permutation as:

if ( $C = p$ ) then

$$U_i(j) = s((j \times r_i) \bmod (p-1)), \quad j = 0, 1, \dots, (p-2), \text{ and } U_i(p-1) = 0,$$

where  $U_i(j)$  is the original bit position of  $j$ -th permuted bit of  $i$ -th row.

end if

if ( $C = p + 1$ ) then

$$U_i(j) = s((j \times r_i) \bmod (p-1)), \quad j = 0, 1, \dots, (p-2). \quad U_i(p-1) = 0, \text{ and } U_i(p) = p,$$

where  $U_i(j)$  is the original bit position of  $j$ -th permuted bit of  $i$ -th row, and

if ( $K = R \times C$ ) then

Exchange  $U_{R-1}(p)$  with  $U_{R-1}(0)$ .

end if

end if

if ( $C = p - 1$ ) then

$$U_i(j) = s((j \times r_i) \bmod (p-1)) - 1, \quad j = 0, 1, \dots, (p-2),$$

where  $U_i(j)$  is the original bit position of  $j$ -th permuted bit of  $i$ -th row.

end if

(6) Perform the inter-row permutation for the rectangular matrix based on the pattern  $\langle T(i) \rangle_{i \in \{0, 1, \dots, R-1\}}$ ,

where  $T(i)$  is the original row position of the  $i$ -th permuted row.

#### 4.2.3.2.3.3 Bits-output from rectangular matrix with pruning

After intra-row and inter-row permutations, the bits of the permuted rectangular matrix are denoted by  $y'_k$ :

$$\begin{bmatrix} y'_1 & y'_{(R+1)} & y'_{(2R+1)} & \cdots & y'_{((C-1)R+1)} \\ y'_2 & y'_{(R+2)} & y'_{(2R+2)} & \cdots & y'_{((C-1)R+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_R & y'_{2R} & y'_{3R} & \cdots & y'_{C \times R} \end{bmatrix}$$

The output of the Turbo code internal interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted  $R \times C$  rectangular matrix starting with bit  $y'_1$  in row 0 of column 0 and ending with bit  $y'_{CR}$  in row  $R - 1$  of column  $C - 1$ . The output is pruned by deleting dummy bits that were padded to the input of the rectangular matrix before intra-row and inter row permutations, i.e. bits  $y'_k$  that corresponds to bits  $y_k$  with  $k > K$  are removed from the output. The bits output from Turbo code internal interleaver are denoted by  $x'_1, x'_2, \dots, x'_K$ , where  $x'_1$  corresponds to the bit  $y'_k$  with smallest index  $k$  after pruning,  $x'_2$  to the bit  $y'_k$  with second smallest index  $k$  after pruning, and so on. The number of bits output from Turbo code internal interleaver is  $K$  and the total number of pruned bits is:

$$R \times C - K.$$

### 4.2.3.3 Concatenation of encoded blocks

After the channel coding for each code block, if  $C_i$  is greater than 1, the encoded blocks are serially concatenated so that the block with lowest index  $r$  is output first from the channel coding block, otherwise the encoded block is output from channel coding block as it is. The bits output are denoted by  $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$ , where  $i$  is the TrCH number and  $E_i = C_i Y_i$ . The output bits are defined by the following relations:

$$c_{ik} = y_{i1k} \quad k = 1, 2, \dots, Y_i$$

$$c_{ik} = y_{i,2,(k-Y_i)} \quad k = Y_i + 1, Y_i + 2, \dots, 2Y_i$$

$$c_{ik} = y_{i,3,(k-2Y_i)} \quad k = 2Y_i + 1, 2Y_i + 2, \dots, 3Y_i$$

...

$$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)} \quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \dots, C_i Y_i$$

If no code blocks are input to the channel coding ( $C_i = 0$ ), no bits shall be output from the channel coding, i.e.  $E_i = 0$ .

### 4.2.4 Radio frame size equalisation

Radio frame size equalisation is padding the input bit sequence in order to ensure that the output can be segmented in  $F_i$  data segments of same size as described in the subclause 4.2.6.

The input bit sequence to the radio frame size equalisation is denoted by  $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$ , where  $i$  is TrCH number and  $E_i$  the number of bits. The output bit sequence is denoted by  $t_{i1}, t_{i2}, t_{i3}, \dots, t_{iT_i}$ , where  $T_i$  is the number of bits. The output bit sequence is derived as follows:

$$t_{ik} = c_{ik}, \text{ for } k = 1 \dots E_i \text{ and}$$

$$t_{ik} = \{0, 1\} \text{ for } k = E_i + 1 \dots T_i, \text{ if } E_i < T_i$$

where

$$T_i = F_i * N_i \text{ and}$$

$$N_i = \lceil E_i / F_i \rceil \text{ is the number of bits per segment after size equalisation.}$$

### 4.2.5 1st interleaving

The 1<sup>st</sup> interleaving is a block interleaver with inter-column permutations. The input bit sequence to the block interleaver is denoted by  $x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,X_i}$ , where  $i$  is TrCH number and  $X_i$  the number of bits. Here  $X_i$  is guaranteed to be an integer multiple of the number of radio frames in the TTI. The output bit sequence from the block interleaver is derived as follows:

- 1) select the number of columns C1 from table 4 depending on the TTI. The columns are numbered 0, 1, ..., C1 - 1 from left to right.

- 2) determine the number of rows of the matrix, R1 defined as

$$R1 = X_i / C1.$$

The rows of the matrix are numbered 0, 1, ..., R1 - 1 from top to bottom.

- 3) write the input bit sequence into the  $R1 \times C1$  matrix row by row starting with bit  $x_{i,1}$  in column 0 of row 0 and ending with bit  $x_{i,(R1 \times C1)}$  in column C1 - 1 of row R1 - 1:



$$\begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & \dots & x_{i,C1} \\ x_{i,(C1+1)} & x_{i,(C1+2)} & x_{i,(C1+3)} & \dots & x_{i,(2 \times C1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{i,((R1-1) \times C1+1)} & x_{i,((R1-1) \times C1+2)} & x_{i,((R1-1) \times C1+3)} & \dots & x_{i,(R1 \times C1)} \end{bmatrix}$$

- 4) Perform the inter-column permutation for the matrix based on the pattern  $\langle P1_{C1}(j) \rangle_{j \in \{0,1,\dots,C1-1\}}$  shown in table 4, where  $P1_{C1}(j)$  is the original column position of the  $j$ -th permuted column. After permutation of the columns, the bits are denoted by  $y_{i,k}$ :

$$\begin{bmatrix} y_{i,1} & y_{i,(R1+1)} & y_{i,(2 \times R1+1)} & \dots & y_{i,((C1-1) \times R1+1)} \\ y_{i,2} & y_{i,(R1+2)} & y_{i,(2 \times R1+2)} & \dots & y_{i,((C1-1) \times R1+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{i,R1} & y_{i,(2 \times R1)} & y_{i,(3 \times R1)} & \dots & y_{i,(C1 \times R1)} \end{bmatrix}$$

- 5) Read the output bit sequence  $y_{i,1}, y_{i,2}, y_{i,3}, \dots, y_{i,(C1 \times R1)}$  of the block interleaver column by column from the inter-column permuted  $R1 \times C1$  matrix. Bit  $y_{i,1}$  corresponds to row 0 of column 0 and bit  $y_{i,(R1 \times C1)}$  corresponds to row  $R1 - 1$  of column  $C1 - 1$ .

**Table 4 Inter-column permutation patterns for 1st interleaving**

TTI	Number of columns C1	Inter-column permutation patterns $\langle P1_{C1}(0), P1_{C1}(1), \dots, P1_{C1}(C1-1) \rangle$
5ms <sup>(*)</sup> , 10 ms	1	$\langle 0 \rangle$
20 ms	2	$\langle 0, 1 \rangle$
40 ms	4	$\langle 0, 2, 1, 3 \rangle$
80 ms	8	$\langle 0, 4, 2, 6, 1, 5, 3, 7 \rangle$

<sup>(\*)</sup> can be used for PRACH for 1.28 Mcps TDD

#### 4.2.5.1 Relation between input and output of 1<sup>st</sup> interleaving

The bits input to the 1<sup>st</sup> interleaving are denoted by  $t_{i,1}, t_{i,2}, t_{i,3}, \dots, t_{i,T_i}$ , where  $i$  is the TrCH number and  $T_i$  the number of bits. Hence,  $x_{i,k} = t_{i,k}$  and  $X_i = T_i$ .

The bits output from the 1<sup>st</sup> interleaving are denoted by  $d_{i,1}, d_{i,2}, d_{i,3}, \dots, d_{i,T_i}$ , and  $d_{i,k} = y_{i,k}$ .

#### 4.2.6 Radio frame segmentation

When the transmission time interval is longer than 10 ms, the input bit sequence is segmented and mapped onto consecutive  $F_i$  radio frames. Following radio frame size equalisation the input bit sequence length is guaranteed to be an integer multiple of  $F_i$ .

The input bit sequence is denoted by  $x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,X_i}$  where  $i$  is the TrCH number and  $X_i$  is the number bits. The  $F_i$  output bit sequences per TTI are denoted by  $y_{i,n_1}, y_{i,n_2}, y_{i,n_3}, \dots, y_{i,n_{Y_i}}$  where  $n_i$  is the radio frame number in current TTI and  $Y_i$  is the number of bits per radio frame for TrCH  $i$ . The output sequences are defined as follows:

$$y_{i,n_k} = x_{i,((n_i-1)Y_i)+k}, n_i = 1 \dots F_i, k = 1 \dots Y_i$$

where

$$Y_i = (X_i / F_i) \text{ is the number of bits per segment.}$$

The  $n_i$ -th segment is mapped to the  $n_i$ -th radio frame of the transmission time interval.

The input bit sequence to the radio frame segmentation is denoted by  $d_{i1}, d_{i2}, d_{i3}, \dots, d_{iT_i}$ , where  $i$  is the TrCH number and  $T_i$  the number of bits. Hence,  $x_{ik} = d_{ik}$  and  $X_i = T_i$ .

The output bit sequence corresponding to radio frame  $n_i$  is denoted by  $e_{i1}, e_{i2}, e_{i3}, \dots, e_{iN_i}$ , where  $i$  is the TrCH number and  $N_i$  is the number of bits. Hence,  $e_{i,k} = y_{i,n_i,k}$  and  $N_i = Y_i$ .

## 4.2.7 Rate matching

Rate matching means that bits on a TrCH are repeated or punctured. Higher layers assign a rate-matching attribute for each TrCH. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

The number of bits on a TrCH can vary between different transmission time intervals. When the number of bits between different transmission time intervals is changed, bits are repeated to ensure that the total bit rate after TrCH multiplexing is identical to the total channel bit rate of the allocated physical channels.

If no bits are input to the rate matching for all TrCHs within a CCTrCH, the rate matching shall output no bits for all TrCHs within the CCTrCH.

### Notation used in subclause 4.2.7 and subclauses:

- $N_{ij}$  : Number of bits in a radio frame before rate matching on TrCH  $i$  with transport format combination  $j$ .
- $\Delta N_{i,j}$  : If positive – number of bits to be repeated in each radio frame on TrCH  $i$  with transport format combination  $j$ .  
If negative – number of bits to be punctured in each radio frame on TrCH  $i$  with transport format combination  $j$ .
- $RM_i$  : Semi-static rate matching attribute for TrCH  $i$ . Signalled from higher layers.
- $PL$  : Puncturing limit. This value limits the amount of puncturing that can be applied in order to minimise the number of physical channels. Signalled from higher layers. The allowed puncturing in % is actually equal to  $(1-PL)*100$ .
- $N_{data,j}$  : Total number of bits that are available for a CCTrCH in a radio frame with transport format combination  $j$ .
- $P$  : number of physical channels used in the current frame.
- $P_{max}$  : maximum number of physical channels allocated for a CCTrCH.
- $U_p$  : Number of data bits in the physical channel  $p$  with  $p = 1 \dots P$  during a radio frame.
- $I$  : Number of TrCHs in a CCTrCH.
- $Z_{ij}$  : Intermediate calculation variable.
- $F_i$  : Number of radio frames in the transmission time interval of TrCH  $i$ .
- $n_i$  : Radio frame number in the transmission time interval of TrCH  $i$  ( $0 \leq n_i < F_i$ ).
- $q$  : Average puncturing or repetition distance (normalised to only show the remaining rate matching on top of an integer number of repetitions).
- $PI_F(n_i)$  : The column permutation function of the 1<sup>st</sup> interleaver,  $PI_F(x)$  is the original position of column with number  $x$  after permutation.  $PI$  is defined on table 4 of section 4.2.5 (note that  $PI_F$  self-inverse).
- $S[n]$  : The shift of the puncturing or repetition pattern for radio frame  $n_i$  when  $n = PI_{F_i}(n_i)$ .
- $TF_i(j)$  : Transport format of TrCH  $i$  for the transport format combination  $j$ .
- $TFS(i)$  : The set of transport format indexes  $l$  for TrCH  $i$ .

- $e_{ini}$  : Initial value of variable  $e$  in the rate matching pattern determination algorithm of subclause 4.2.7.3.
- $e_{plus}$  : Increment of variable  $e$  in the rate matching pattern determination algorithm of subclause 4.2.7.3.
- $e_{minus}$  : Decrement of variable  $e$  in the rate matching pattern determination algorithm of subclause 4.2.7.3.
- $b$  : Indicates systematic and parity bits.
- $b=1$ : Systematic bit.  $X(t)$  in subclause 4.2.3.2.1.
- $b=2$ : 1<sup>st</sup> parity bit (from the upper Turbo constituent encoder).  $Y(t)$  in subclause 4.2.3.2.1.
- $b=3$ : 2<sup>nd</sup> parity bit (from the lower Turbo constituent encoder).  $Y'(t)$  in subclause 4.2.3.2.1.

Note: when the TTI is 5msec for 1.28Mcps, the above notation refers to a sub-frame rather than a radio frame. In this case,  $F_i = 1$  and  $n_i = 0$ .

#### 4.2.7.1 Determination of rate matching parameters

The following relations, defined for all TFC  $j$ , are used when calculating the rate matching pattern:

$$Z_{0,j} = 0$$

$$Z_{i,j} = \left\lfloor \frac{\left( \left( \sum_{m=1}^i RM_m \times N_{m,j} \right) \times N_{data,j} \right)}{\sum_{m=1}^I RM_m \times N_{m,j}} \right\rfloor \quad \text{for all } i = 1 \dots I(1)$$

$$\Delta N_{i,j} = Z_{i,j} - Z_{i-1,j} - N_{i,j} \quad \text{for all } i = 1 \dots I$$

Puncturing can be used to minimise the required transmission capacity. The maximum amount of puncturing that can be applied is 1-PL, PL is signalled from higher layers. The possible values for  $N_{data}$  depend on the number of physical channels  $P_{max}$ , allocated to the respective CCTrCH, and on their characteristics (spreading factor, length of midamble and TFCI code word, usage of TPC and multiframe structure), which is given in [7].

For each physical channel an individual minimum spreading factor  $Sp_{min}$  is transmitted by means of the higher layers. Denote the number of data bits in each physical channel by  $U_{p,Sp}$ , where  $p$  indicates the sequence number  $1 \leq p \leq P_{max}$  and  $Sp$  indicates the spreading factor of this physical channel:  $Sp$  takes the possible values  $\{16, 8, 4, 2, 1\}$  for 1.28Mcps TDD and 3.84Mcps TDD,  $Sp$  takes the possible values  $\{32, 16, 8, 4, 2, 1\}$  for 7.68Mcps TDD. The index  $p$  is described in section 4.2.12 with the following modifications: spreading factor ( $Q$ ) is replaced by the minimum spreading factor  $Sp_{min}$  and  $k$  is replaced by the channelization code index at  $Q = Sp_{min}$ . Then, for  $N_{data}$  one of the following values in ascending order can be chosen:

$$\left\{ U_{1,Sp_{min}}, U_{1,Sp_{min}} + U_{2,Sp_{min}}, U_{1,Sp_{min}} + U_{2,Sp_{min}} + \dots + U_{P_{max},(Sp_{max})_{min}} \right\}$$

Optionally, if indicated by higher layers for the UL the UE shall vary the spreading factor autonomously, so that  $N_{data}$  is one of the following values in ascending order:

$$\left\{ U_{1,16}, \dots, U_{1,Sp_{min}}, U_{1,Sp_{min}} + U_{2,16}, \dots, U_{1,Sp_{min}} + U_{2,Sp_{min}}, \dots, U_{1,Sp_{min}} + U_{2,Sp_{min}} + \dots + U_{P_{max},16}, \dots, U_{1,Sp_{min}} + U_{2,Sp_{min}} + \dots + U_{P_{max},(Sp_{max})_{min}} \right\}$$

$N_{data,j}$  for the transport format combination  $j$  is determined by executing the following algorithm:

$$\text{SET1} = \left\{ N_{data} \text{ such that } \left( \min_{1 \leq y \leq I} \{ RM_y \} \right) \times N_{data} - PL \times \sum_{x=1}^I RM_x \times N_{x,j} \text{ is non negative} \right\}$$

$$N_{data,j} = \min \text{SET1}$$

The number of bits to be repeated or punctured,  $\Delta N_{i,j}$ , within one radio frame (one sub-frame when the TTI is 5msec) for each TrCH  $i$  is calculated with the relations given at the beginning of this subclause for all possible transport format combinations  $j$  and selected every radio frame (sub-frame). The number of physical channels corresponding to  $N_{\text{data},j}$ , shall be denoted by  $P$ .

If  $\Delta N_{i,j} = 0$  then the output data of the rate matching is the same as the input data and the rate matching algorithm of subclause 4.2.7.3 does not need to be executed.

Otherwise, the rate matching pattern is calculated with the algorithm described in subclause 4.2.7.3. For this algorithm the parameters  $e_{\text{ini}}$ ,  $e_{\text{plus}}$ ,  $e_{\text{minus}}$ , and  $X_i$  are needed, which are calculated according to the equations in subclauses 4.2.7.1.1 and 4.2.7.1.2.

#### 4.2.7.1.1 Uncoded and convolutionally encoded TrCHs

$$a = 2$$

$$\Delta N_i = \Delta N_{i,j}$$

$$X_i = N_{i,j}$$

$$R = \Delta N_{i,j} \bmod N_{i,j} \text{ -- note: in this context } \Delta N_{i,j} \bmod N_{i,j} \text{ is in the range of } 0 \text{ to } N_{i,j}-1 \text{ i.e. } -1 \bmod 10 = 9.$$

if  $R \neq 0$  and  $2 \times R \leq N_{i,j}$

$$\text{then } q = \lceil N_{i,j} / R \rceil$$

else

$$q = \lceil N_{i,j} / (R - N_{i,j}) \rceil$$

endif

NOTE 1:  $q$  is a signed quantity.

If  $q$  is even

$$\text{then } q' = q + \text{gcd}(|q|, F_i) / F_i \text{ -- where } \text{gcd}(|q|, F_i) \text{ means greatest common divisor of } |q| \text{ and } F_i$$

NOTE 2:  $q'$  is not an integer, but a multiple of 1/8.

else

$$q' = q$$

endif

for  $x = 0$  to  $F_i-1$

$$S[\lfloor x \times q' \rfloor \bmod F_i] = (\lfloor x * q' \rfloor \text{ div } F_i)$$

end for

$$e_{\text{ini}} = (a \times S[\text{P1}_{F_i}(n_i)] \times |\Delta N_i| + 1) \bmod (a \times N_{i,j})$$

$$e_{\text{plus}} = a \times X_i$$

$$e_{\text{minus}} = a \times |\Delta N_i|$$

puncturing for  $\Delta N_i < 0$ , repetition otherwise.

#### 4.2.7.1.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e.  $\Delta N_{i,j} > 0$ , the parameters in subclause 4.2.7.1.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index  $b$  is used to indicate systematic ( $b=1$ ), 1<sup>st</sup> parity ( $b=2$ ), and 2<sup>nd</sup> parity bit ( $b=3$ ).

$a = 2$  when  $b=2$

$a = 1$  when  $b=3$

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{i,j} / 2 \rfloor, & b = 2 \\ \lfloor \Delta N_{i,j} / 2 \rfloor, & b = 3 \end{cases}$$

If  $\Delta N_i$  is calculated as 0 for  $b=2$  or  $b=3$ , then the following procedure and the rate matching algorithm of subclause 4.2.7.3 don't need to be performed for the corresponding parity bit stream.

$$X_i = \lfloor N_{i,j} / 3 \rfloor,$$

$$q = \lfloor X_i / |\Delta N_i| \rfloor$$

if( $q \leq 2$ )

for  $r=0$  to  $F_i-1$

$$S[(3 \times r + b - 1) \bmod F_i] = r \bmod 2;$$

end for

else

if  $q$  is even

then  $q' = q - \gcd(q, F_i) / F_i$  -- where  $\gcd(q, F_i)$  means greatest common divisor of  $q$  and  $F_i$

NOTE:  $q'$  is not an integer, but a multiple of  $1/8$ .

else  $q' = q$

endif

for  $x=0$  to  $F_i - 1$

$$r = \lceil x \times q' \rceil \bmod F_i;$$

$$S[(3 \times r + b - 1) \bmod F_i] = \lceil x \times q' \rceil \div F_i;$$

endfor

endif

For each radio frame, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.3, where:

$X_i$  is as above,

$$e_{imi} = (a \times S[P1 \ F_i(n_i)] \times |\Delta N_i| + X_i) \bmod (a \times X_i), \text{ if } e_{imi} = 0 \text{ then } e_{imi} = a \times X_i$$

$$e_{plus} = a \times X_i$$

$$e_{minus} = a \times |\Delta N_i|$$

#### 4.2.7.2 Bit separation and collection for rate matching

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits may be punctured. The systematic bits, first parity bits, and second parity bits in the bit sequence input to the rate matching block are therefore separated into three sequences.

The first sequence contains:

- All of the systematic bits that are from turbo encoded TrCHs.

- From 0 to 2 first and/or second parity bits that are from turbo encoded TrCHs. These bits come into the first sequence when the total number of bits in a block after radio frame segmentation is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second and third sequences shall be of equal length, whereas the first sequence can contain from 0 to 2 more bits. Puncturing is applied only to the second and third sequences.

The bit separation function is transparent for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition. The bit separation and bit collection are illustrated in figures 4 and 5.

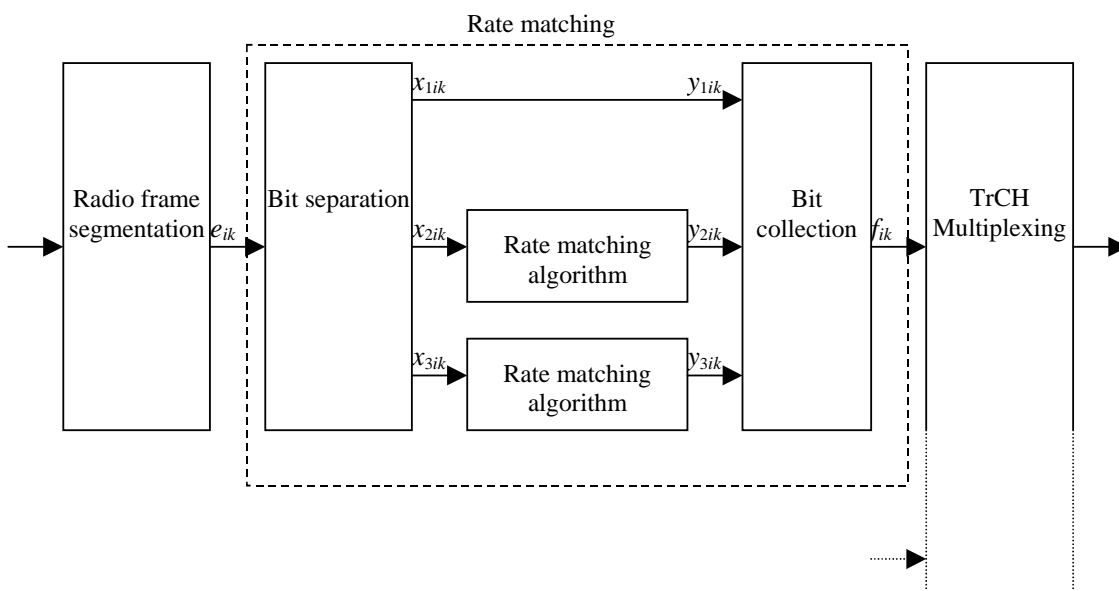
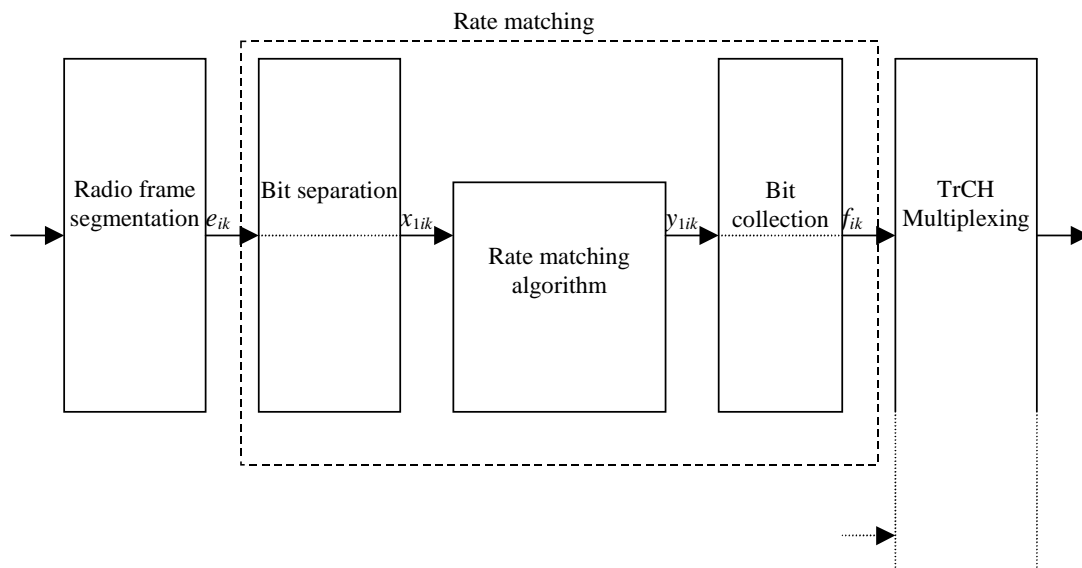


Figure 4: Puncturing of turbo encoded TrCHs



**Figure 5: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition**

The bit separation is dependent on the 1<sup>st</sup> interleaving and offsets are used to define the separation for different TTIs.  $b$  indicates the three sequences defined in this section, with  $b=1$  indicating the first sequence,  $b = 2$  the second one, and  $b = 3$  the third one.

The offsets  $\alpha_b$  for these sequences are listed in table 5.

**Table 5: TTI dependent offset needed for bit separation**

TTI (ms)	$\alpha_1$	$\alpha_2$	$\alpha_3$
5, 10, 40	0	1	2
20, 80	0	2	1

The bit separation is different for different radio frames in the TTI. A second offset is therefore needed. The radio frame number for TrCH  $i$  is denoted by  $n_i$ . and the offset by  $\beta_{n_i}$ .

**Table 6: Radio frame dependent offset needed for bit separation**

TTI (ms)	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$	$\beta_7$
5, 10	0	NA	NA	NA	NA	NA	NA	NA
20	0	1	NA	NA	NA	NA	NA	NA
40	0	1	2	0	NA	NA	NA	NA
80	0	1	2	0	1	2	0	1

4.2.7.2.1 Bit separation

The bits input to the rate matching are denoted by  $e_{i,1}, e_{i,2}, e_{i,3}, \dots, e_{i,N_i}$ , where  $i$  is the TrCH number and  $N_i$  is the number of bits input to the rate matching block. Note that the transport format combination number  $j$  for simplicity has been left out in the bit numbering, i.e.  $N_i=N_{ij}$ . The bits after separation are denoted by  $x_{b,i,1}, x_{b,i,2}, x_{b,i,3}, \dots, x_{b,i,X_i}$ . For turbo encoded TrCHs with puncturing,  $b$  indicates the three sequences defined in section 4.2.7.2, with  $b=1$  indicating the first sequence, and so forth. For all other cases  $b$  is defined to be 1.  $X_i$  is the number of bits in each separated bit sequence. The relation between  $e_{i,k}$  and  $x_{b,i,k}$  is given below.

For turbo encoded TrCHs with puncturing:

$$x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i / 3 \rfloor$$

$$x_{1,i,\lfloor N_i/3 \rfloor+k} = e_{i,3\lfloor N_i/3 \rfloor+k} \quad k = 1, \dots, N_i \bmod 3 \quad \text{Note: When } (N_i \bmod 3) = 0 \text{ this row is not needed.}$$

$$x_{2,i,k} = e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i/3 \rfloor$$

$$x_{3,i,k} = e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i/3 \rfloor$$

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$x_{1,i,k} = e_{i,k} \quad k = 1, 2, 3, \dots, X_i \quad X_i = N_i$$

#### 4.2.7.2.2 Bit collection

The bits  $x_{b,i,k}$  are input to the rate matching algorithm described in subclause 4.2.7.3. The bits output from the rate matching algorithm are denoted  $y_{b,i,1}, y_{b,i,2}, y_{b,i,3}, \dots, y_{b,i,Y_i}$ .

Bit collection is the inverse function of the separation. The bits after collection are denoted by

$z_{b,i,1}, z_{b,i,2}, z_{b,i,3}, \dots, z_{b,i,Y_i}$ . After bit collection, the bits indicated as punctured are removed and the bits are then denoted by  $f_{i,1}, f_{i,2}, f_{i,3}, \dots, f_{i,V_i}$ , where  $i$  is the TrCH number and  $V_i = N_{i,j} + \Delta N_{i,j}$ . The relations between  $y_{b,i,k}$ ,  $z_{b,i,k}$ , and  $f_{i,k}$  are given below.

For turbo encoded TrCHs with puncturing ( $Y_i = X_i$ ):

$$z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3\lfloor N_i/3 \rfloor+k} = y_{1,i,\lfloor N_i/3 \rfloor+k} \quad k = 1, \dots, N_i \bmod 3 \quad \text{Note: When } (N_i \bmod 3) = 0 \text{ this row is not needed.}$$

$$z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3} = y_{2,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3} = y_{3,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

After the bit collection, bits  $z_{i,k}$  with value  $\delta$ , where  $\delta \notin \{0, 1\}$ , are removed from the bit sequence. Bit  $f_{i,1}$  corresponds to the bit  $z_{i,k}$  with smallest index  $k$  after puncturing, bit  $f_{i,2}$  corresponds to the bit  $z_{i,k}$  with second smallest index  $k$  after puncturing, and so on.

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$z_{i,k} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

When repetition is used,  $f_{i,k} = z_{i,k}$  and  $Y_i = V_i$ .

When puncturing is used,  $Y_i = X_i$  and bits  $z_{i,k}$  with value  $\delta$ , where  $\delta \notin \{0, 1\}$ , are removed from the bit sequence. Bit  $f_{i,1}$  corresponds to the bit  $z_{i,k}$  with smallest index  $k$  after puncturing, bit  $f_{i,2}$  corresponds to the bit  $z_{i,k}$  with second smallest index  $k$  after puncturing, and so on.

#### 4.2.7.3 Rate matching pattern determination

The bits input to the rate matching are denoted by  $x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,X_i}$ , where  $i$  is the TrCH and  $X_i$  is the parameter given in subclauses 4.2.7.1.1 and 4.2.7.1.2.

NOTE: The transport format combination number  $j$  for simplicity has been left out in the bit numbering.

The rate matching rule is as follows:

if puncturing is to be performed



```

e = eini    -- initial error between current and desired puncturing ratio
m = 1        -- index of current bit
do while m <= Xi
    e = e - eminus    -- update error
    if e <= 0 then    -- check if bit number m should be punctured
        set bit xi,m to δ where δ ∈ {0, 1}
        e = e + eplus    -- update error
    end if
    m = m + 1        -- next bit
end do
else
e = eini    -- initial error between current and desired puncturing ratio
m = 1        -- index of current bit
do while m <= Xi
    e = e - eminus    -- update error
    do while e <= 0    -- check if bit number m should be repeated
        repeat bit xi,m
        e = e + eplus    -- update error
    end do
    m = m + 1        -- next bit
end do
end if

```

A repeated bit is placed directly after the original one.

## 4.2.8 TrCH multiplexing

Every 10 ms, one radio frame from each TrCH is delivered to the TrCH multiplexing. These radio frames are serially multiplexed into a coded composite transport channel (CCTrCH). If the TTI is smaller than 10ms, then no TrCH multiplexing is performed.

The bits input to the TrCH multiplexing are denoted by  $f_{i,1}, f_{i,2}, f_{i,3}, \dots, f_{i,V_i}$ , where  $i$  is the TrCH id number and  $V_i$  is the number of bits in the radio frame of TrCH  $i$ . The number of TrCHs is denoted by  $I$ . The bits output from TrCH multiplexing are denoted by  $h_1, h_2, h_3, \dots, h_S$ , where  $S$  is the number of bits, i.e.  $S = \sum_i V_i$ . The TrCH multiplexing is defined by the following relations:

$$h_k = f_{1,k} \quad k = 1, 2, \dots, V_1$$

$$h_k = f_{2,(k-V_1)} \quad k = V_1+1, V_1+2, \dots, V_1+V_2$$

$$h_k = f_{3,(k-(V_1+V_2))} \quad k = (V_1+V_2)+1, (V_1+V_2)+2, \dots, (V_1+V_2)+V_3$$

...

$$h_k = f_{I,(k-(V_1+V_2+\dots+V_{I-1}))} \quad k = (V_1+V_2+\dots+V_{I-1})+1, (V_1+V_2+\dots+V_{I-1})+2, \dots, (V_1+V_2+\dots+V_{I-1})+V_I$$

## 4.2.9 Bit Scrambling

The bits output from the TrCH multiplexer are scrambled in the bit scrambler. The bits input to the bit scrambler are denoted by  $h_1, h_2, h_3, \dots, h_S$ , where  $S$  is the number of bits input to the bit scrambling block equal to the total number of bits on the CCTrCH. The bits after bit scrambling are denoted  $s_1, s_2, s_3, \dots, s_S$ .

Bit scrambling is defined by the following relation:

$$s_k = h_k \oplus p_k \quad k = 1, 2, \dots, S$$

and  $p_k$  results from the following operation:

$$p_k = \left( \sum_{i=1}^{16} g_i \cdot p_{k-i} \right) \bmod 2; \quad p_k = 0; k < 1; \quad p_1 = 1; \quad g = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1\}$$

## 4.2.10 Physical channel segmentation

When more than one PhCH is used, physical channel segmentation divides the bits among the different PhCHs. The bits input to the physical channel segmentation are denoted by  $s_1, s_2, s_3, \dots, s_S$ , where  $S$  is the number of bits input to the physical channel segmentation block. The number of PhCHs after rate matching is denoted by  $P$ , as defined in subclause 4.2.7.1.

The bits after physical channel segmentation are denoted  $u_{p,1}, u_{p,2}, u_{p,3}, \dots, u_{p,U_p}$ , where  $p$  is PhCH number and  $U_p$  is the in general variable number of bits in the respective radio frame for each PhCH. The relation between  $s_k$  and  $u_{p,k}$  is given below.

Bits on first PhCH after physical channel segmentation:

$$u_{1,k} = s_k \quad k = 1, 2, \dots, U_1$$

Bits on second PhCH after physical channel segmentation:

$$u_{2,k} = s_{(k+U_1)} \quad k = 1, 2, \dots, U_2$$

...

Bits on the  $P^{\text{th}}$  PhCH after physical channel segmentation:

$$u_{P,k} = s_{(k+U_1+\dots+U_{P-1})} \quad k = 1, 2, \dots, U_P$$

## 4.2.11 2nd interleaving

The 2<sup>nd</sup> interleaving is a block interleaver and consists of bits input to a matrix with padding, the inter-column permutation for the matrix and bits output from the matrix with pruning. The 2nd interleaving can be applied jointly to all data bits transmitted during one frame, or separately within each timeslot, on which the CCTrCH is mapped. The selection of the 2nd interleaving scheme is controlled by higher layer.

#### 4.2.11.1 Frame related 2nd interleaving

In case of frame related 2<sup>nd</sup> interleaving, the bits input to the block interleaver are denoted by  $x_1, x_2, x_3, \dots, x_U$ , where  $U$  is the total number of bits after TrCH multiplexing transmitted during the respective radio frame with

$$S=U = \sum_p U_p .$$

The relation between  $x_k$  and the bits  $u_{p,k}$  in the respective physical channels is given below:

$$x_k = u_{1,k} \quad k = 1, 2, \dots, U_1$$

$$x_{(k+U_1)} = u_{2,k} \quad k = 1, 2, \dots, U_2$$

...

$$x_{(k+U_1+\dots+U_{p-1})} = u_{p,k} \quad k = 1, 2, \dots, U_p$$

The following steps have to be performed once for each CCTrCH:

- (1) Assign  $C2 = 30$  to be the number of columns of the matrix. The columns of the matrix are numbered  $0, 1, 2, \dots, C2 - 1$  from left to right.
- (2) Determine the number of rows of the matrix,  $R2$ , by finding minimum integer  $R2$  such that:

$$U \leq R2 \times C2.$$

The rows of rectangular matrix are numbered  $0, 1, 2, \dots, R2 - 1$  from top to bottom.

- (3) Write the input bit sequence  $x_1, x_2, x_3, \dots, x_U$  into the  $R2 \times C2$  matrix row by row starting with bit  $y_1$  in column 0 of row 0:

$$\begin{bmatrix} y_1 & y_2 & y_3 & \dots & y_{C2} \\ y_{(C2+1)} & y_{(C2+2)} & y_{(C2+3)} & \dots & y_{(2 \times C2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{((R2-1) \times C2+1)} & y_{((R2-1) \times C2+2)} & y_{((R2-1) \times C2+3)} & \dots & y_{(R2 \times C2)} \end{bmatrix}$$

where  $y_k = x_k$  for  $k = 1, 2, \dots, U$  and if  $R2 \times C2 > U$ , the dummy bits are padded such that  $y_k = 0$  or  $1$  for  $k = U + 1, U + 2, \dots, R2 \times C2$ . These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

- (4) Perform the inter-column permutation for the matrix based on the pattern  $\langle P_2(j) \rangle_{j \in \{0,1,\dots,C2-1\}}$  that is shown in table 7, where  $P_2(j)$  is the original column position of the  $j$ -th permuted column. After permutation of the columns, the bits are denoted by  $y'_k$ .

$$\begin{bmatrix} y'_1 & y'_{(R2+1)} & y'_{(2 \times R2+1)} & \dots & y'_{((C2-1) \times R2+1)} \\ y'_2 & y'_{(R2+2)} & y'_{(2 \times R2+2)} & \dots & y'_{((C2-1) \times R2+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y'_{R2} & y'_{(2 \times R2)} & y'_{(3 \times R2)} & \dots & y'_{(C2 \times R2)} \end{bmatrix}$$

- (5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted  $R2 \times C2$  matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits  $y'_k$  that corresponds to bits  $y_k$  with  $k > U$  are removed from the output. The bits at the output of the block interleaver are denoted by  $z_1, z_2, \dots, z_U$ , where  $z_1$

corresponds to the bit  $y'_k$  with smallest index  $k$  after pruning,  $z_2$  to the bit  $y'_k$  with second smallest index  $k$  after pruning, and so on.

The bits  $z_1, z_2, \dots, z_U$  shall be segmented as follows:

$$u_{1,k} = z_k \quad k = 1, 2, \dots, U_1$$

$$u_{2,k} = z_{(k+U_1)} \quad k = 1, 2, \dots, U_2$$

...

$$u_{p,k} = z_{(k+U_1+\dots+U_{p-1})} \quad k = 1, 2, \dots, U_p$$

The bits after frame related  $2^{\text{nd}}$  interleaving are denoted by  $v_{t,1}, v_{t,2}, \dots, v_{t,U_t}$ , where  $t$  refers to the timeslot sequence number and  $U_t$  is the number of bits transmitted in this timeslot during the respective radio frame.

Let  $T$  be the number of time slots in a CCTrCH during the respective radio frame (where for 1.28Mcps TDD, the respective radio frame includes subframes 1 and 2), and  $t = 1, \dots, T$ . The physical layer shall assign the time slot sequence number  $t$  in ascending order of the allocated time slots in the CCTrCH in the respective radio frame. In time slot  $t$ ,  $R_t$  refers to the number of physical channels within the respective time slot and  $r = 1, \dots, R_t$ . The relation between  $r$  and  $t$  and the physical channel sequence number  $p$  as detailed in 4.2.12.1 is given by:

$$p = r \quad t = 1$$

$$p = R_1 + R_2, \dots, R_{t-1} + r \quad 1 < t \leq T$$

Defining the relation  $u_{t,r,k} = u_{p,k}$  and denoting  $U_{tr}$  as the number of bits for physical channel  $r$  in time slot  $t$ , the relation between  $v_{t,k}$  and  $u_{t,r,k}$  is given below:

$$v_{t,k} = u_{t,1,k} \quad k = 1, 2, \dots, U_{t1}$$

$$v_{t,(k+U_{t1})} = u_{t,2,k} \quad k = 1, 2, \dots, U_{t2}$$

...

$$v_{t,(k+U_{t1}+\dots+U_{t(R_t-1)})} = u_{t,R_t,k} \quad k = 1, 2, \dots, U_{tR_t}$$

#### 4.2.11.2 Timeslot related $2^{\text{nd}}$ interleaving

In case of timeslot related  $2^{\text{nd}}$  interleaving, the bits input to the block interleaver are denoted by  $x_{t,1}, x_{t,2}, x_{t,3}, \dots, x_{t,U_t}$ , where  $t$  is the timeslot sequence number, and  $U_t$  is the number of bits transmitted in this timeslot during the respective radio frame.

Let  $T$  be the number of time slots in a CCTrCH during the respective radio frame (where for 1.28Mcps TDD, the respective radio frame includes subframes 1 and 2), and  $t = 1, \dots, T$ . The physical layer shall assign the time slot sequence number  $t$  in ascending order of the allocated time slots in the CCTrCH in the respective radio frame. In timeslot  $t$ ,  $R_t$  refers to the number of physical channels within the respective timeslot and  $r = 1, \dots, R_t$ . The relation between  $r$  and  $t$  and the physical channel sequence number  $p$  as detailed in 4.2.12.1 is given by:

$$p = r \quad t = 1$$

$$p = R_1 + R_2, \dots, R_{t-1} + r \quad 1 < t \leq T$$

Defining the relation  $u_{t,r,k} = u_{p,k}$  and denoting  $U_{tr}$  as the number of bits for physical channel  $r$  in time slot  $t$ , the relation between  $x_{t,k}$  and  $u_{t,r,k}$  is given below:

$$\begin{aligned} x_{t,k} &= u_{t,1,k} & k &= 1, 2, \dots, U_{t1} \\ x_{t,(k+U_{t1})} &= u_{t,2,k} & k &= 1, 2, \dots, U_{t2} \\ &\dots & & \\ x_{t,(k+U_{t1}+\dots+U_{t(R_t-1)})} &= u_{t,R_t,k} & k &= 1, 2, \dots, U_{tR_t} \end{aligned}$$

The following steps have to be performed for each timeslot  $t$ , on which the respective CCTrCH is mapped:

- (1) Assign  $C2 = 30$  to be the number of columns of the matrix. The columns of the matrix are numbered  $0, 1, 2, \dots, C2 - 1$  from left to right.
- (2) Determine the number of rows of the matrix,  $R2$ , by finding minimum integer  $R2$  such that:
 
$$U_t \leq R2 \times C2.$$
 The rows of rectangular matrix are numbered  $0, 1, 2, \dots, R2 - 1$  from top to bottom.
- (3) Write the input bit sequence  $x_{t,1}, x_{t,2}, x_{t,3}, \dots, x_{t,U_t}$  into the  $R2 \times C2$  matrix row by row starting with bit  $y_{t,1}$  in column 0 of row 0:

$$\begin{bmatrix} y_{t,1} & y_{t,2} & y_{t,3} & \dots & y_{t,C2} \\ y_{t,(C2+1)} & y_{t,(C2+2)} & y_{t,(C2+3)} & \dots & y_{t,(2 \times C2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{t,((R2-1) \times C2+1)} & y_{t,((R2-1) \times C2+2)} & y_{t,((R2-1) \times C2+3)} & \dots & y_{t,(R2 \times C2)} \end{bmatrix}$$

where  $y_{t,k} = x_{t,k}$  for  $k = 1, 2, \dots, U_t$  and if  $R2 \times C2 > U_t$ , the dummy bits are padded such that  $y_{t,k} = 0$  or  $1$  for  $k = U_t + 1, U_t + 2, \dots, R2 \times C2$ . These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

- (4) Perform the inter-column permutation for the matrix based on the pattern  $\langle P2(j) \rangle_{j \in \{0,1,\dots,C2-1\}}$  that is shown in table 7, where  $P2(j)$  is the original column position of the  $j$ -th permuted column. After permutation of the columns, the bits are denoted by  $y'_{t,k}$ .

$$\begin{bmatrix} y'_{t,1} & y'_{t,(R2+1)} & y'_{t,(2 \times R2+1)} & \dots & y'_{t,((C2-1) \times R2+1)} \\ y'_{t,2} & y'_{t,(R2+2)} & y'_{t,(2 \times R2+2)} & \dots & y'_{t,((C2-1) \times R2+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y'_{t,R2} & y'_{t,(2 \times R2)} & y'_{t,(3 \times R2)} & \dots & y'_{t,(C2 \times R2)} \end{bmatrix}$$

- (5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted  $R2 \times C2$  matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits  $y'_{t,k}$  that corresponds to bits  $y_{t,k}$  with  $k > U_t$  are removed from the output. The bits after time slot  $2^{\text{nd}}$  interleaving are denoted by  $v_{t,1}, v_{t,2}, \dots, v_{t,U_t}$ , where  $v_{t,1}$  corresponds to the bit  $y'_{t,k}$  with smallest index  $k$  after pruning,  $v_{t,2}$  to the bit  $y'_{t,k}$  with second smallest index  $k$  after pruning, and so on.

**Table 7 Inter-column permutation pattern for 2nd interleaving**

Number of Columns C2	Inter-column permutation pattern < P2(0), P2(1), ..., P2(C2-1) >
30	<0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17>

#### 4.2.11A Sub-frame segmentation for the 1.28 Mcps option

In the 1.28Mcps TDD, it is needed to add a sub-frame segmentation unit between 2nd interleaving unit and physical channel mapping unit when the TTI of the CCTrCh is greater than 5msec. In this case, the operation of rate-matching guarantees that the size of bit streams is an even number and can be subdivided into 2 sub-frames. The transport channel multiplexing structure for uplink and downlink is shown in figure 1A.

The input to the sub-frame segmentation unit is segmented into timeslot chunks, where each timeslot chunk contains all of the bits that are to be transmitted in a given timeslot position in both of the sub-frames.

The input bit sequence is denoted by  $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$  where  $i$  is the timeslot number and  $X_i$  is the number of bits transmitted in timeslot  $i$  in a radio frame. The two output bit sequences per radio frame are denoted by  $y_{i,n,1}, y_{i,n,2}, y_{i,n,3}, \dots, y_{i,n,Y_i}$  where  $n$  is the sub-frame number in current radio frame and  $Y_i$  is the number of bits per sub-frame for timeslot  $i$ . The output sequences are defined as follows:

$$y_{i,n,k} = x_{i,((n-1)Y_i)+k}, n = 1 \text{ or } 2, k = 1 \dots Y_i$$

where

$Y_i = (X_i / 2)$  is the number of bits in timeslot  $i$  per sub-frame,

$x_{ik}$  is the  $k^{\text{th}}$  bit of the input bit sequence and

$y_{i,n,k}$  is the  $k^{\text{th}}$  bit of the output bit sequence corresponding to the  $n^{\text{th}}$  sub-frame

#### 4.2.12 Physical channel mapping

##### 4.2.12.1 Physical channel mapping for the 3.84 Mcps and 7.68Mcps options

The PhCH for both uplink and downlink is defined in [7]. The bits after physical channel mapping are denoted by  $w_{p,1}, w_{p,2}, \dots, w_{p,U_p}$ , where  $p$  is the PhCH number corresponding to the sequence number  $1 \leq p \leq P$  of this physical channel as detailed below,  $U_p$  is the number of bits in one radio frame for the respective PhCH, and  $P \leq P_{max}$ . The bits  $w_{p,k}$  are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to  $k$ .

The physical layer shall assign the physical channel sequence number  $p$  to the physical channels of the CCTrCH in the respective radio frame, treating each allocated timeslot in ascending order. If within a timeslot there are multiple physical channels they shall first be ordered in ascending order of the spreading factor ( $Q$ ) and subsequently by channelisation code index ( $k$ ), as shown in [9].

The mapping of the bits  $v_{t,1}, v_{t,2}, \dots, v_{t,U_t}$  is performed like block interleaving, writing the bits into columns, but a PhCH with an odd number is filled in forward order, whereas a PhCH with an even number is filled in reverse order.

The mapping scheme, as described in the following subclause, shall be applied individually for each timeslot  $t$  used in the current frame. Therefore, the bits  $v_{t,1}, v_{t,2}, \dots, v_{t,U_t}$  are assigned to the bits of the physical channels

$w_{t,1,1 \dots U_{t1}}, w_{t,2,1 \dots U_{t2}}, \dots, w_{t,P,1 \dots U_{tP}}$  in each timeslot.

In uplink there are at most two codes allocated ( $P \leq 2$ ). If there is only one code, the same mapping as for downlink is applied. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. For the number of consecutive bits to assign per code  $bs_k$  the following rule is applied:

if

SF1  $\geq$  SF2 then  $bs_1 = 1$  ;  $bs_2 = SF1/SF2$  ;

else

SF2  $>$  SF1 then  $bs_1 = SF2/SF1$ ;  $bs_2 = 1$  ;

end if

In the downlink case  $bs_p$  is 1 for all physical channels.

#### 4.2.12.1.1 Mapping scheme

Notation used in this subclause:

$P_t$ : number of physical channels for timeslot  $t$ ,  $P_t = 1..2$  for uplink ;  $P_t = 1..16$  for downlink for 1.28Mcps TDD and 3.84Mcps TDD,  $P_t = 1..32$  for 7.68Mcps TDD

$U_{t,p}$ : capacity in bits for the physical channel  $p$  in timeslot  $t$

$U_t$ : total number of bits to be assigned for timeslot  $t$

$bs_p$ : number of consecutive bits to assign per code

for downlink all  $bs_p = 1$

for uplink if SF1  $\geq$  SF2 then  $bs_1 = 1$  ;  $bs_2 = SF1/SF2$  ;

if SF2  $>$  SF1 then  $bs_1 = SF2/SF1$ ;  $bs_2 = 1$  ;

$fb_p$ : number of already written bits for each code

pos: intermediate calculation variable

for  $p=1$  to  $P_t$  -- reset number of already written bits for every physical channel

$fb_p = 0$

end for

$p = 1$  -- start with PhCH #1

for  $k=1$  to  $U_t$

do while ( $fb_p == U_{t,p}$ ) -- physical channel filled up already ?

$p = (p \bmod P_t) + 1$ ;

end do

if ( $p \bmod 2 == 0$ )

$pos = U_{t,p} - fb_p$  -- reverse order

else

$pos = fb_p + 1$  -- forward order

endif

$w_{t,p,pos} = v_{t,k}$  -- assignment

$fb_p = fb_p + 1$  -- Increment number of already written bits

```

if (fbp mod bsp) == 0          -- Conditional change to the next physical channel
    p = (p mod Pt) + 1;
end if

```

```

end for

```

#### 4.2.12.2 Physical channel mapping for the 1.28 Mcps option

The bit streams from the sub-frame segmentation unit are mapped onto code channels of time slots in sub-frames.

The bits after physical channel mapping are denoted by  $W_{p1}, W_{p2}, \dots, W_{pU_p}$ , where  $p$  is the PhCH number and  $U_p$  is the number of bits in one sub-frame for the respective PhCH. The bits  $w_{pk}$  are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to  $k$ .

The mapping of the bits  $y_{t,n,1}, y_{t,n,2}, y_{t,n,3}, \dots, y_{t,n,U_t}$  is performed like block interleaving, writing the bits into columns, but a PhCH with an odd number is filled in forward order, were as a PhCH with an even number is filled in reverse order.

The mapping scheme, as described in the following subclause, shall be applied individually for each timeslot  $t$  used in the current subframe. Therefore, the bits  $y_{t,n,1}, y_{t,n,2}, y_{t,n,3}, \dots, y_{t,n,U_t}$  are assigned to the bits of the physical channels

$W_{t1,1\dots U_{t1}}, W_{t2,1\dots U_{t2}}, \dots, W_{tP_t,1\dots U_{tP_t}}$  in each timeslot.

In uplink there are at most two codes allocated ( $P \leq 2$ ). If there is only one code, the same mapping as for downlink is applied. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. For the number of consecutive bits to assign per code  $bs_k$  the following rule is applied:

```

if
SF1 >= SF2 then bs1 = 1 ; bs2 = SF1/SF2 ;
else
SF2 > SF1 then bs1 = SF2/SF1; bs2 = 1 ;
end if

```

In the downlink case  $bs_p$  is 1 for all physical channels.

##### 4.2.12.2.1 Mapping scheme

Notation used in this subclause:

$P_t$ : number of physical channels for timeslot  $t$ ,  $P_t = 1..2$  for uplink ;  $P_t = 1..16$  for downlink

$U_{tp}$ : capacity in bits for the physical channel  $p$  in timeslot  $t$  in the current sub-frame

$U_t$ : total number of bits to be assigned for timeslot  $t$  in the current sub-frame

$n$  = index of the current sub-frame (1 or 2)

$bs_p$ : number of consecutive bits to assign per code

for downlink all  $bs_p = 1$

for uplink if SF1 >= SF2 then  $bs_1 = 1$  ;  $bs_2 = SF1/SF2$  ;

if SF2 > SF1 then  $bs_1 = SF2/SF1$ ;  $bs_2 = 1$  ;

$fb_p$ : number of already written bits for each code

$pos$ : intermediate calculation variable

for  $p=1$  to  $P_t$  -- reset number of already written bits for every physical channel



```

fbp = 0
end for
p = 1          -- start with PhCH #1
for k=1 to Ut.
do while (fbp == Ut,p)  -- physical channel filled up already ?
p = (p mod Pt) + 1 ;
end do
if (p mod 2) == 0
pos = Ut,p - fbp      -- reverse order
else
pos = fbp + 1        -- forward order
end if
wtp,pos = yt,n,k      -- assignment
fbp = fbp + 1      -- Increment number of already written bits
If (fbp mod bsp) == 0  -- Conditional change to the next physical channel
p = (p mod Pt) + 1 ;
end if
end for

```

#### 4.2.13 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels

Different transport channels can be encoded and multiplexed together into one Coded Composite Transport Channel (CCTrCH). The following rules shall apply to the different transport channels which are part of the same CCTrCH:

- 1) Transport channels multiplexed into one CCTrCh shall have co-ordinated timings. When the TFCS of a CCTrCH is changed because one or more transport channels are added to the CCTrCH or reconfigured within the CCTrCH, or removed from the CCTrCH, the change may only be made at the start of a radio frame with CFN fulfilling the relation

$$\text{CFN} \bmod F_{\max} = 0,$$

where  $F_{\max}$  denotes the maximum number of radio frames within the transmission time intervals of all transport channels which are multiplexed into the same CCTrCH, including any transport channels  $i$  which are added reconfigured or have been removed, and CFN denotes the connection frame number of the first radio frame of the changed CCTrCH.

After addition or reconfiguration of a transport channel  $i$  within a CCTrCH, the TTI of transport channel  $i$  may only start in radio frames with CFN fulfilling the relation

$$\text{CFN}_i \bmod F_i = 0.$$

- 2) Different CCTrCHs cannot be mapped onto the same physical channel.
- 3) One CCTrCH shall be mapped onto one or several physical channels.
- 4) Dedicated Transport channels and common transport channels cannot be multiplexed into the same CCTrCH.
- 5) For the common transport channels, only the FACH and PCH may belong to the same CCTrCH.

6) Each CCTrCH carrying a BCH shall carry only one BCH and shall not carry any other Transport Channel.

7) Each CCTrCH carrying a RACH shall carry only one RACH and shall not carry any other Transport Channel.

Hence, there are two types of CCTrCH.

CCTrCH of dedicated type, corresponding to the result of coding and multiplexing of one or several DCH.

CCTrCH of common type, corresponding to the result of the coding and multiplexing of a common channel, i.e. RACH and USCH in the uplink and DSCH, BCH, FACH or PCH in the downlink, respectively.

Transmission of TFCI is possible for CCTrCH containing Transport Channels of:

- dedicated type;
- USCH type;
- DSCH type;
- FACH and/or PCH type.

#### 4.2.13.1 Allowed CCTrCH combinations for one UE

##### 4.2.13.1.1 Allowed CCTrCH combinations on the uplink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

- 1) several CCTrCH of dedicated type;
- 2) several CCTrCH of common type.

##### 4.2.13.1.2 Allowed CCTrCH combinations on the downlink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

- 3) several CCTrCH of dedicated type;
- 4) several CCTrCH of common type.

#### 4.2.14 Transport format detection

Transport format detection can be performed both with and without Transport Format Combination Indicator (TFCI). If a TFCI is transmitted, the receiver detects the transport format combination from the TFCI. When no TFCI is transmitted, so called blind transport format detection may be used, i.e. the receiver side uses the possible transport format combinations as a priori information.

##### 4.2.14.1 Blind transport format detection

Blind Transport Format Detection is optional both in the UE and the UTRAN. Therefore, for all CCTrCH a TFCI shall be transmitted, including the possibility of a TFCI code word length zero, if only one TFC is defined.

##### 4.2.14.2 Explicit transport format detection based on TFCI

###### 4.2.14.2.1 Transport Format Combination Indicator (TFCI)

The Transport Format Combination Indicator (TFCI) informs the receiver of the transport format combination of the CCTrCHs. As soon as the TFCI is detected, the transport format combination, and hence the individual transport channels' transport formats are known, and decoding of the transport channels can be performed.

## 4.3 Coding for layer 1 control for the 3.84 Mcps and 7.68Mcps TDD options

### 4.3.1 Coding of transport format combination indicator (TFCI)

Encoding of the TFCI depends on its length. If there are 6-10 bits of TFCI the channel encoding is done as described in subclause 4.3.1.1. Also specific coding of less than 6 bits is possible as explained in subclause 4.3.1.2.

#### 4.3.1.1 Coding of long TFCI lengths

The TFCI is encoded using a (32, 10) sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 6.

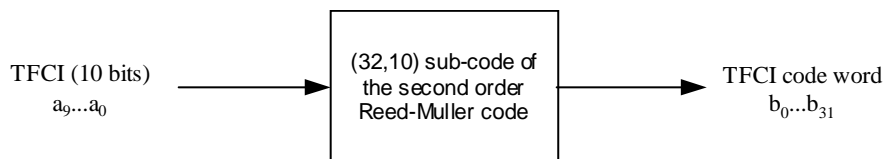


Figure 6: Channel coding of the TFCI bits

If the TFCI consists of less than 10 bits, it is padded with zeros to 10 bits, by setting the most significant bits to zero. TFCI is encoded by the (32,10) sub-code of second order Reed-Muller code. The code words of the (32,10) sub-code of second order Reed-Muller code are linear combination of some among 10 basis sequences. The basis sequences are as follows in table 8.

Table 8: Basis sequences for (32,10) TFCI code

$l$	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$
0	1	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0
2	1	1	0	0	0	1	0	0	0	1
3	0	0	1	0	0	1	1	0	1	1
4	1	0	1	0	0	1	0	0	0	1
5	0	1	1	0	0	1	0	0	1	0
6	1	1	1	0	0	1	0	1	0	0
7	0	0	0	1	0	1	0	1	1	0
8	1	0	0	1	0	1	1	1	1	0
9	0	1	0	1	0	1	1	0	1	1
10	1	1	0	1	0	1	0	0	1	1
11	0	0	1	1	0	1	0	1	1	0
12	1	0	1	1	0	1	0	1	0	1
13	0	1	1	1	0	1	1	0	0	1
14	1	1	1	1	0	1	1	1	1	1
15	1	0	0	0	1	1	1	1	0	0
16	0	1	0	0	1	1	1	1	0	1
17	1	1	0	0	1	1	1	0	1	0
18	0	0	1	0	1	1	0	1	1	1
19	1	0	1	0	1	1	0	1	0	1
20	0	1	1	0	1	1	0	0	1	1
21	1	1	1	0	1	1	0	1	1	1
22	0	0	0	1	1	1	0	1	0	0
23	1	0	0	1	1	1	1	1	0	1
24	0	1	0	1	1	1	1	0	1	0
25	1	1	0	1	1	1	1	0	0	1
26	0	0	1	1	1	1	0	0	1	0
27	1	0	1	1	1	1	1	1	0	0
28	0	1	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	0	0	0	0	0	1	0	0	0	0
31	0	0	0	0	1	1	1	0	0	0

The TFCI bits  $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$  (where  $a_0$  is LSB and  $a_9$  is MSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

The output TFCI code word bits  $b_i$  are given by:

$$b_i = \sum_{n=0}^9 (a_n \times M_{i,n}) \text{ mod } 2$$

where  $i = 0, \dots, 31$ .  $N_{\text{TFCI code word}} = 32$ .

### 4.3.1.2 Coding of short TFCI lengths

#### 4.3.1.2.1 Coding very short TFCIs by repetition

If the number of TFCI bits is 1 or 2, then repetition will be used for coding. In this case each bit is repeated to a total of 4 times giving 4-bit transmission ( $N_{\text{TFCI code word}}=4$ ) for a single TFCI bit and 8-bit transmission ( $N_{\text{TFCI code word}}=8$ ) for 2 TFCI bits. The TFCI bit(s)  $a_0$  (or  $a_0$  and  $a_1$  where  $a_0$  is the LSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

In the case of  $N_{\text{TFCI code word}}=4$ , the TFCI codeword  $\{b_0, b_1, b_2, b_3\}$  is equal to the sequence  $\{a_0, a_0, a_0, a_0\}$ .

In the case of  $N_{\text{TFCI code word}}=8$ , the TFCI codeword  $\{b_0, b_1, \dots, b_7\}$  is equal to the sequence  $\{a_0, a_1, a_0, a_1, a_0, a_1, a_0, a_1\}$ .

#### 4.3.1.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits is in the range 3 to 5 the TFCI is encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 7.

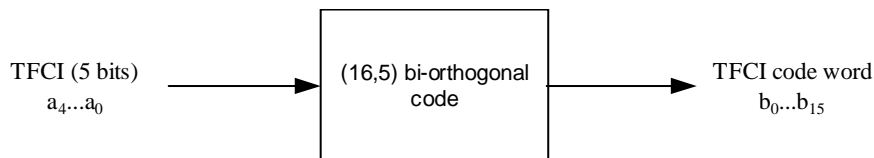


Figure 7: Channel coding of short length TFCI bits

If the TFCI consists of less than 5 bits, it is padded with zeros to 5 bits, by setting the most significant bits to zero. The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 9.

Table 9: Basis sequences for (16,5) TFCI code

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$
0	1	0	0	0	1
1	0	1	0	0	1
2	1	1	0	0	1
3	0	0	1	0	1
4	1	0	1	0	1
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	0	1	1
8	1	0	0	1	1
9	0	1	0	1	1
10	1	1	0	1	1
11	0	0	1	1	1
12	1	0	1	1	1
13	0	1	1	1	1
14	1	1	1	1	1
15	0	0	0	0	1

The TFCI bits  $a_0, a_1, a_2, a_3, a_4$  (where  $a_0$  is LSB and  $a_4$  is MSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

The output code word bits  $b_j$  are given by:

$$b_i = \sum_{n=0}^4 (a_n \times M_{i,n}) \text{ mod } 2$$

where  $i = 0, \dots, 15$ .  $N_{\text{TFCI code word}} = 16$ .

### 4.3.1.3 Mapping of TFCI code word

The mapping of the TFCI code word to the TFCI bit positions in a timeslot shall be as follows.

Denote the number of bits in the TFCI code word by  $N_{\text{TFCI code word}}$ , denote the TFCI code word bits by  $b_k$  where  $k=0 \dots N_{\text{TFCI code word}}-1$ .

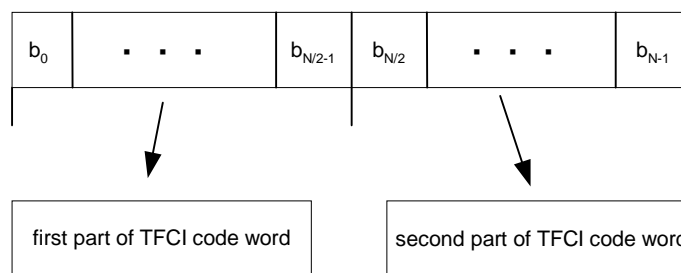


Figure 8: Mapping of TFCI code word bits to timeslot

The locations of the first and second parts of the TFCI code word in the timeslot is defined in [7].

If the shortest transmission time interval of any constituent TrCH is at least 20 ms the successive TFCI code words in the frames in the TTI shall be identical. If TFCI is transmitted on multiple timeslots in a frame each timeslot shall have the same TFCI code word.

### 4.3.2 Coding and Bit Scrambling of the Paging Indicator

The paging indicator  $P_q, q = 0, \dots, N_{PI}-1, P_q \in \{0, 1\}$  is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI, calculated by higher layers, and the associated paging indicator  $P_q$ . The length  $L_{PI}$  of the paging indicator is  $L_{PI}=2, L_{PI}=4$  or  $L_{PI}=8$  symbols.  $N_{PIB} = 2 * N_{PI} * L_{PI}$  bits are used for the paging indicator transmission in one radio frame. The mapping of the paging indicators to the bits  $e_i, i = 1, \dots, N_{PIB}$  is shown in table 10.

Table 10: Mapping of the paging indicator

$P_q$	Bits $\{e_{2L_{PI} \cdot q+1}, e_{2L_{PI} \cdot q+2}, \dots, e_{2L_{PI} \cdot (q+1)}\}$	Meaning
0	$\{0, 0, \dots, 0\}$	There is no necessity to receive the PCH
1	$\{1, 1, \dots, 1\}$	There is the necessity to receive the PCH

If the number  $S$  of bits in one radio frame available for the PICH is bigger than the number  $N_{PIB}$  of bits used for the transmission of paging indicators, the sequence  $e = \{e_1, e_2, \dots, e_{N_{PIB}}\}$  is extended by  $S-N_{PIB}$  bits that are set to zero, resulting in a sequence  $h = \{h_1, h_2, \dots, h_S\}$ :

$$h_k = e_k, \quad k = 1, \dots, N_{PIB}$$

$$h_k = 0, \quad k = N_{PIB} + 1, \dots, S$$

The bits  $h_k, k = 1, \dots, S$  on the PICH then undergo bit scrambling as defined in section 4.2.9.

The bits  $s_k, k = 1, \dots, S$  output from the bit scrambler are then transmitted over the air as shown in [7].

### 4.3.3 Coding and Bit Scrambling of the MBMS Notification Indicator

The MBMS notification indicator  $N_q, q = 0, \dots, N_n-1, N_q \in \{0, 1\}$ , is an identifier to instruct UEs whether there is an MBMS notification indication for the groups of MBMS services that are associated to the NI, calculated by higher layers, and the associated MBMS notification indicator  $N_q$ . The length  $L_{NI}$  of the MBMS notification indicator is  $L_{NI}=2, L_{NI}=4$  or  $L_{NI}=8$  symbols.  $N_{NIB} = 2 * N_n * L_{NI}$  bits are used for the MBMS notification indicator transmission in one MICH. The mapping of the MBMS notification indicators to the bits  $e_i, i = 1, \dots, N_{NIB}$  is shown in table 10A.

**Table 10A: Mapping of the MBMS notification indicator**

$N_q$	Bits $\{e_{2L_{NI} * q + 1}, e_{2L_{NI} * q + 2}, \dots, e_{2L_{NI} * (q+1)}\}$
0	$\{0, 0, \dots, 0\}$
1	$\{1, 1, \dots, 1\}$

If the number  $S$  of bits available for the MICH is bigger than the number  $N_{NIB}$  of bits used for the transmission of MBMS notification indicators, the sequence  $e = \{e_1, e_2, \dots, e_{N_{NIB}}\}$  is extended by  $S - N_{NIB}$  bits that are set to zero, resulting in a sequence  $h = \{h_1, h_2, \dots, h_S\}$ :

$$h_k = e_k, \quad k = 1, \dots, N_{NIB}$$

$$h_k = 0, \quad k = N_{NIB} + 1, \dots, S$$

The bits  $h_k, k = 1, \dots, S$  on the MICH then undergo bit scrambling as defined in section 4.2.9.

The bits  $s_k, k = 1, \dots, S$  output from the bit scrambler are then transmitted over the air as shown in [7].

## 4.4 Coding for layer 1 control for the 1.28 Mcps option

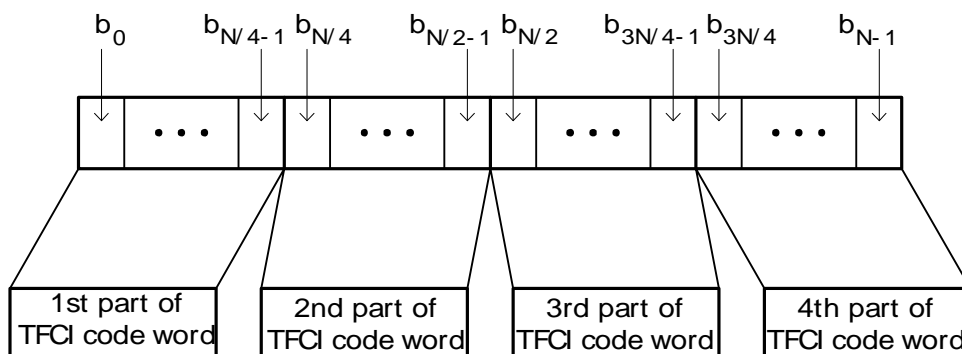
### 4.4.1 Coding of transport format combination indicator (TFCI) for QPSK and 16QAM

The coding of TFCI for 1.28Mcps TDD is same as that of 3.84Mcps TDD.cf.[4.3.1 'Coding of transport format combination indicator'].

#### 4.4.1.1 Mapping of TFCI code word

Denote the number of bits in the TFCI code word by  $N_{TFCI \text{ code word}}$ , and denote the TFCI code word bits by  $b_k$ , where  $k = 0, \dots, N_{TFCI \text{ code word}} - 1$

When the number of bits in the TFCI code word is 8, 16, 32, the mapping of the TFCI code word to the TFCI bit positions shall be as follows:



**Figure 9: Mapping of TFCI code word bits to TFCI position in 1.28 Mcps TDD option, where  $N = N_{TFCI \text{ code word}}$**

For MBSFN transmissions with 16QAM, the coded bits  $b_k$ , are mapped to the transmitted TFCI bits according to the following two group formulas:

Formula a, mapping onto the outer-corners of the 16QAM constellation

$$d_{4k} = b_{2k},$$

$$d_{4k+1} = b_{2k+1},$$

$$d_{4k+2} = 1,$$

$$d_{4k+3} = 1,$$

Formula b, mapping onto the inner-corners of the 16QAM constellation

$$d_{4k} = b_{2k},$$

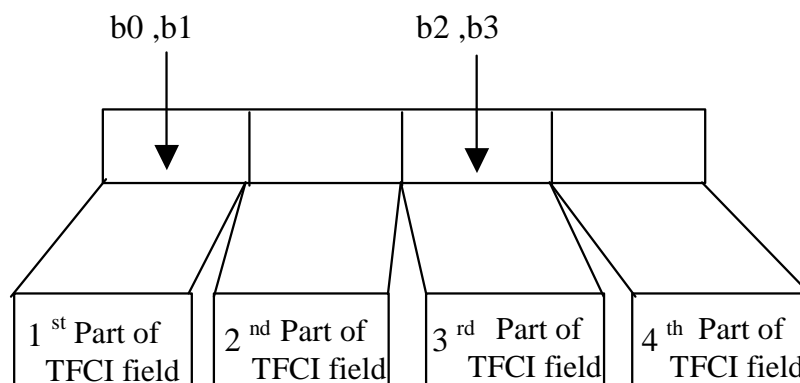
$$d_{4k+1} = b_{2k+1},$$

$$d_{4k+2} = 0,$$

$$d_{4k+3} = 0,$$

The 1st and the 3rd part of TFCI code word will use the Mapping Formula a, and the 2nd and 4th part of TFCI code word will use the Mapping Formula b. As the TTI of S-CCPCH is 20ms or 40ms or 80ms, the TFCI bits will be repeated with a period of 10 ms frame. In the second 10ms frame, the 1st and the 3rd part of TFCI code word will use the Mapping Formula b, and the 2nd and 4th part of TFCI code word will use the Mapping Formula a, and so on for the consecutive frames.

When the number of bits of the TFCI code word is 4, then the TFCI code word is equally divided into two parts for the consecutive two subframe and mapped onto the end of the first data field in each of the consecutive subframes. The mapping for  $N_{\text{TFCI code word}}=4$  is shown in figure 10:



**Figure 10: Mapping of TFCI code word bits to TFCI position in 1.28 Mcps TDD option, when  $N_{\text{TFCI code word}}=4$**

The location of the 1st to 4th parts of the TFCI code word in the timeslot is defined in [7].

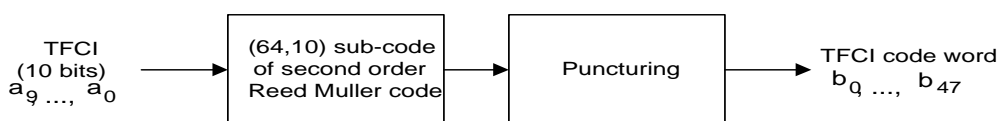
If the shortest transmission time interval of any constituent TrCH is at least 20 ms, then successive TFCI code words in the frames within the TTI shall be identical. If a TFCI is transmitted on multiple timeslots in a frame each timeslot shall have the same TFCI code word.

## 4.4.2 Coding of transport format combination indicator (TFCI) for 8PSK

Encoding of TFCI bits depends on the number of them and the modulation in use. When 2 Mcps service is transmitted, 8PSK modulation is applied in 1.28 Mcps TDD option. The encoding scheme for TFCI when the number of bits are 6 – 10, and less than 6 bits is described in section 4.4.2.1 and 4.4.2.2, respectively.

### 4.4.2.1 Coding of long TFCI lengths

When the number of TFCI bits is 6 – 10, the TFCI bits are encoded by using a (64,10) sub-code of the second order Reed-Muller code, then 16 bits out of 64 bits are punctured (Puncturing positions are 0, 4, 8, 13, 16, 20, 27, 31, 34, 38, 41, 44, 50, 54, 57, 61<sup>st</sup> bits). The coding procedure is shown in Figure 11.



**Figure 11: Channel coding of long TFCI bits for 8PSK**

If the TFCI consists of less than 10 bits, it is padded with zeros to 10 bits, by setting the most significant bits to zero. The code words of the punctured (48,10) sub-code of the second order Reed-Muller codes are linear combination of 10 basis sequences. The basis sequences are shown in Table 11.



Table 11: Basis sequences for (48,10) TFCI code

<b>I</b>	<b>M<sub>i,0</sub></b>	<b>M<sub>i,1</sub></b>	<b>M<sub>i,2</sub></b>	<b>M<sub>i,3</sub></b>	<b>M<sub>i,4</sub></b>	<b>M<sub>i,5</sub></b>	<b>M<sub>i,6</sub></b>	<b>M<sub>i,7</sub></b>	<b>M<sub>i,8</sub></b>	<b>M<sub>i,9</sub></b>
0	1	0	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	1	0	0
2	1	1	0	0	0	0	1	1	0	1
3	1	0	1	0	0	0	1	1	1	0
4	0	1	1	0	0	0	1	0	1	0
5	1	1	1	0	0	0	1	1	1	0
6	1	0	0	1	0	0	1	1	1	1
7	0	1	0	1	0	0	1	1	0	1
8	1	1	0	1	0	0	1	0	1	0
9	0	0	1	1	0	0	1	1	0	0
10	0	1	1	1	0	0	1	1	0	1
11	1	1	1	1	0	0	1	1	1	1
12	1	0	0	0	1	0	1	0	1	1
13	0	1	0	0	1	0	1	1	1	0
14	1	1	0	0	1	0	1	0	0	1
15	1	0	1	0	1	0	1	0	1	1
16	0	1	1	0	1	0	1	1	0	0
17	1	1	1	0	1	0	1	1	1	0
18	0	0	0	1	1	0	1	0	0	1
19	1	0	0	1	1	0	1	0	1	1
20	0	1	0	1	1	0	1	0	1	0
21	0	0	1	1	1	0	1	0	1	0
22	1	0	1	1	1	0	1	1	0	1
23	0	1	1	1	1	0	1	1	1	0
24	0	0	0	0	0	1	1	1	0	1
25	1	0	0	0	0	1	1	1	1	0
26	1	1	0	0	0	1	1	1	1	1
27	0	0	1	0	0	1	1	0	1	1
28	1	0	1	0	0	1	1	1	0	1
29	1	1	1	0	0	1	1	0	1	1
30	0	0	0	1	0	1	1	0	0	1
31	0	1	0	1	0	1	1	0	0	1
32	1	1	0	1	0	1	1	1	1	1
33	1	0	1	1	0	1	1	0	0	1
34	0	1	1	1	0	1	1	1	1	0
35	1	1	1	1	0	1	1	1	0	1
36	0	0	0	0	1	1	1	1	1	0
37	1	0	0	0	1	1	1	0	1	1
38	1	1	0	0	1	1	1	1	1	1
39	0	0	1	0	1	1	1	1	0	0
40	1	0	1	0	1	1	1	1	0	0
41	1	1	1	0	1	1	1	1	1	1
42	0	0	0	1	1	1	1	1	1	1
43	0	1	0	1	1	1	1	0	1	0
44	1	1	0	1	1	1	1	0	1	0
45	0	0	1	1	1	1	1	0	1	1
46	0	1	1	1	1	1	1	0	0	1
47	1	1	1	1	1	1	1	1	0	0

Let's define the TFCI bits as  $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$ , where  $a_0$  is the LSB and  $a_9$  is the MSB. The TFCI bits shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

The output TFCI code word bits  $b_i$  are given by:

$$b_i = \sum_{n=0}^9 (a_n \times M_{i,n}) \text{ mod } 2$$

where  $i=0 \dots 47$ .  $N_{\text{TFCI code word}}=48$ .

#### 4.4.2.2 Coding of short TFCI lengths

##### 4.4.2.2.1 Coding very short TFCIs by repetition

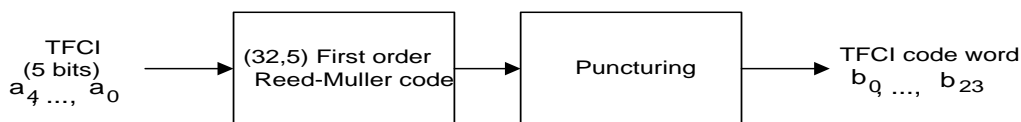
When the number of TFCI bits is 1 or 2, then repetition will be used for the coding. In this case, each bit is repeated to a total of 6 times giving 6-bit transmission ( $N_{\text{TFCI code word}} = 6$ ) for a single TFCI bit and 12-bit transmission ( $N_{\text{TFCI code word}} = 12$ ) for 2 TFCI bits. The TFCI bit(s)  $a_0$  (or  $a_0$  and  $a_1$  where  $a_0$  is the LSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

In the case of  $N_{\text{TFCI code word}}=6$ , the TFCI codeword  $\{b_0, b_1, b_2, b_3, b_4, b_5\}$  is equal to the sequence  $\{a_0, a_0, a_0, a_0, a_0, a_0\}$ .

In the case of  $N_{\text{TFCI code word}}=12$ , the TFCI codeword  $\{b_0, b_1, \dots, b_{11}\}$  is equal to the sequence  $\{a_0, a_1, a_0, a_1, a_0, a_1, a_0, a_1, a_0, a_1, a_0, a_1\}$ .

##### 4.4.2.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits is in the range of 3 to 5, the TFCI bits are encoded using a (32,5) first order Reed-Muller code, then 8 bits out of 32 bits are punctured (Puncturing positions are 0, 1, 2, 3, 4, 5, 6, 7<sup>th</sup> bits). The coding procedure is shown in Figure 12.



**Figure 12: Channel coding of short TFCI bits for 8PSK**

If the TFCI consists of less than 5 bits, it is padded with zeros to 5 bits, by setting the most significant bits to zero. The code words of the punctured (32,5) first order Reed-Muller codes are linear combination of 5 basis sequences shown in Table 12.

**Table 12: Basis sequences for (24,5) TFCI code**

<b>i</b>	<b>M<sub>i,0</sub></b>	<b>M<sub>i,1</sub></b>	<b>M<sub>i,2</sub></b>	<b>M<sub>i,3</sub></b>	<b>M<sub>i,4</sub></b>
0	0	0	0	1	0
1	1	0	0	1	0
2	0	1	0	1	0
3	1	1	0	1	0
4	0	0	1	1	0
5	1	0	1	1	0
6	0	1	1	1	0
7	1	1	1	1	0
8	0	0	0	0	1
9	1	0	0	0	1
10	0	1	0	0	1
11	1	1	0	0	1
12	0	0	1	0	1
13	1	0	1	0	1
14	0	1	1	0	1
15	1	1	1	0	1
16	0	0	0	1	1
17	1	0	0	1	1
18	0	1	0	1	1
19	1	1	0	1	1
20	0	0	1	1	1
21	1	0	1	1	1
22	0	1	1	1	1
23	1	1	1	1	1

Let's define the TFCI bits as  $a_0, a_1, a_2, a_3, a_4$ , where  $a_0$  is the LSB and  $a_4$  is the MSB. The TFCI bits shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated radio frame.

The output code word bits  $b_i$  are given by:

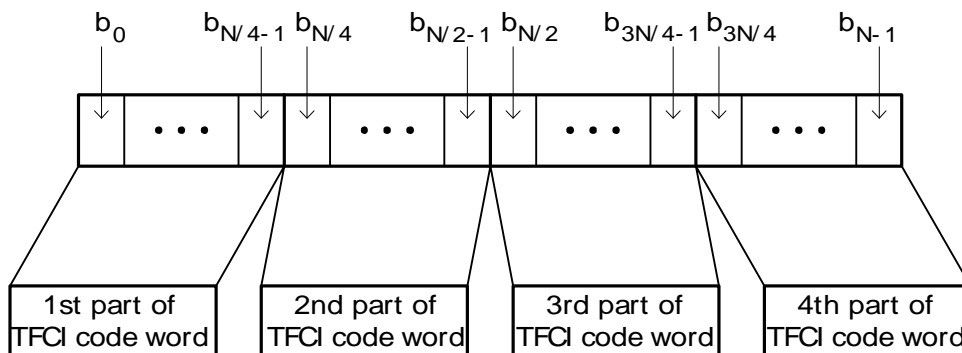
$$b_i = \sum_{n=0}^4 (a_n \times M_{i,n}) \text{ mod } 2$$

where  $i=0 \dots 23$ .  $N_{\text{TFCI code word}} = 24$ .

### 4.4.2.3 Mapping of TFCI code word

Denote the number of bits in the TFCI code word by  $N_{\text{TFCI code word}}$ , and denote the TFCI code word bits by  $b_k$ , where  $k = 0, \dots, N_{\text{TFCI code word}} - 1$ .

When the number of bits in the TFCI code word is 12, 24 or 48, the mapping of the TFCI code word to the TFCI bit positions in a time slot shall be as follows.



**Figure 13: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option, where  $N = N_{\text{TFCI code word}}$**

When the number of bits in the TFCI code word is 6, the TFCI code word is equally divided into two parts for the consecutive two sub-frames and mapped onto the first data field in each of the consecutive sub-frames. The mapping of the TFCI code word to the TFCI bit positions in a time slot shall be as shown in figure 14.

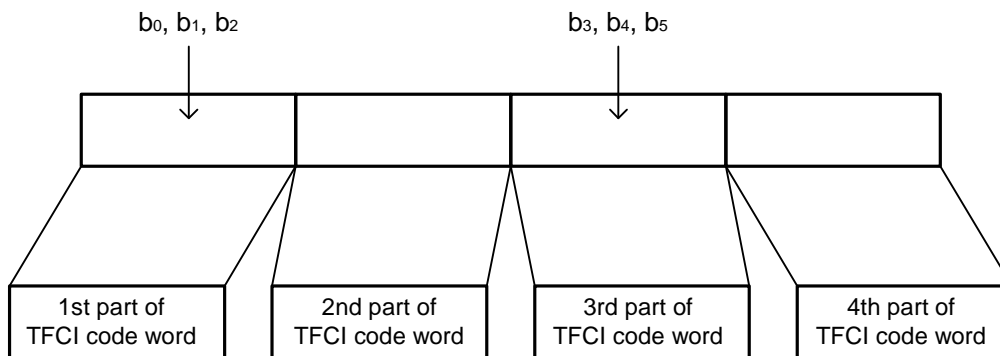


Figure 14: Mapping of TFCI code word bits to timeslot in 1.28 Mcps TDD option when  $N_{\text{TFCI code word}} = 6$

The location of the 1st to 4th parts of the TFCI code word in the timeslot is defined in [7].

#### 4.4.3 Coding and Bit Scrambling of the Paging Indicator

The paging indicator  $P_q$ ,  $q = 0, \dots, N_{\text{PI}}-1$ ,  $P_q \in \{0, 1\}$  is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI, calculated by higher layers, and the associated paging indicator  $P_q$ . The length  $L_{\text{PI}}$  of the paging indicator is  $L_{\text{PI}}=2$ ,  $L_{\text{PI}}=4$  or  $L_{\text{PI}}=8$  symbols.  $N_{\text{PIB}} = 2 * N_{\text{PI}} * L_{\text{PI}}$  bits are used for the paging indicator transmission in one radio frame. The mapping of the paging indicators to the bits  $e_i$ ,  $i = 1, \dots, N_{\text{PIB}}$  is shown in table 13.

Table 13: Mapping of the paging indicator

$P_q$	Bits $\{e_{2L_{\text{PI}}*q+1}, e_{2L_{\text{PI}}*q+2}, \dots, e_{2L_{\text{PI}}*(q+1)}\}$	Meaning
0	$\{0, 0, \dots, 0\}$	There is no necessity to receive the PCH
1	$\{1, 1, \dots, 1\}$	There is the necessity to receive the PCH

If the number  $S$  of bits in one radio frame available for the PICH is bigger than the number  $N_{\text{PIB}}$  of bits used for the transmission of paging indicators, the sequence  $e = \{e_1, e_2, \dots, e_{N_{\text{PIB}}}\}$  is extended by  $S - N_{\text{PIB}}$  bits that are set to zero, resulting in a sequence  $h = \{h_1, h_2, \dots, h_S\}$ :

$$h_k = e_k, \quad k = 1, \dots, N_{\text{PIB}}$$

$$h_k = 0, \quad k = N_{\text{PIB}} + 1, \dots, S$$

The bits  $h_k$ ,  $k = 1, \dots, S$  on the PICH then undergo bit scrambling as defined in section 4.2.9.

The bits  $s_k$ ,  $k = 1, \dots, S$  output from the bit scrambler are then transmitted over the air as shown in [7].

#### 4.4.4 Coding of the Fast Physical Access Channel (FPACH) information bits

The FPACH burst is composed by 32 information bits which are block coded and convolutional coded, and then delivered in one sub-frame as follows:

1. The 32 information bits are protected by 8 parity bits for error detection as described in sub-clause 4.2.1.1.
2. Convolutional code with constraint length 9 and coding rate  $\frac{1}{2}$  is applied as described in sub-clause 4.2.3.1. The size of data block  $c(k)$  after convolutional encoder is 96 bits.
3. To adjust the size of the data block  $c(k)$  to the size of the FPACH burst, 8 bits are punctured as described in sub-clause 4.2.7 with the following clarifications:

- $N_{i,j}=96$  is the number of bits in a radio sub-frame before rate matching
- $\Delta N_{i,j} = -8$  is the number of bits to punctured in a radio sub-frame
- $e_{ini} = a \times N_{ij}$

The 88 bits after rate matching are then delivered to the intra-frame interleaving.

4. The bits in input to the interleaving unit are denoted as  $\{x(0), \dots, x(87)\}$ . The coded bits are block rectangular interleaved according to the following rule: the input is written row by row, the output is read column by column.

$$\begin{bmatrix} x(0) & x(1) & x(2) & \dots & x(7) \\ x(8) & x(9) & x(10) & \dots & x(15) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x(80) & x(81) & x(82) & \dots & x(87) \end{bmatrix}$$

Hence, the interleaved sequence is denoted by  $y(i)$  and are given by:

$$y(0), y(1), \dots, y(87) = x(0), x(8), \dots, x(80), x(1), \dots, x(87).$$

#### 4.4.5 Coding and Bit Scrambling of the MBMS Notification Indicator

The MBMS notification indicator  $N_q$ ,  $q = 0, \dots, N_n - 1$ ,  $N_q \in \{0, 1\}$  is an identifier to instruct the UE whether there is an MBMS notification indication for the groups of MBMS services that are associated to the NI, calculated by higher layers, and the associated MBMS notification indicator  $N_q$ . The length  $L_{NI}$  of the MBMS notification indicator is  $L_{NI}=2$ ,  $L_{NI}=4$  or  $L_{NI}=8$  symbols.  $N_{NIB} = 2 * N_n * L_{NI}$  bits are used for the MBMS notification indicator transmission in one MICH. The mapping of the MBMS notification indicators to the bits  $e_i$ ,  $i = 1, \dots, N_{NIB}$  is shown in table 13A.

**Table 13A: Mapping of the MBMS notification indicator**

$N_q$	Bits $\{e_{2L_{NI} \cdot q + 1}, e_{2L_{NI} \cdot q + 2}, \dots, e_{2L_{NI} \cdot (q+1)}\}$
0	$\{0, 0, \dots, 0\}$
1	$\{1, 1, \dots, 1\}$

If the number  $S$  of bits available for the MICH is bigger than the number  $N_{NIB}$  of bits used for the transmission of MBMS notification indicators, the sequence  $e = \{e_1, e_2, \dots, e_{N_{NIB}}\}$  is extended by  $S - N_{NIB}$  bits that are set to zero, resulting in a sequence  $h = \{h_1, h_2, \dots, h_S\}$ :

$$\begin{aligned} h_k &= e_k, & k &= 1, \dots, N_{NIB} \\ h_k &= 0, & k &= N_{NIB} + 1, \dots, S \end{aligned}$$

The bits  $h_k$ ,  $k = 1, \dots, S$  on the MICH then undergo bit scrambling as defined in section 4.2.9.

The bits  $s_k$ ,  $k = 1, \dots, S$  output from the bit scrambler are then transmitted over the air as shown in [7].

#### 4.4.6 Coding of PLCCH

The PLCCH is a Node-B terminated channel used to carry dedicated (UE-specific) TPC and SS information to multiple UEs. Each TPC/SS command pair for a given UE is mapped to 3 bits as shown in table 13B.

Table 13B: Mapping of the TPC/SS pair

3-bit TPC/SS command (MSB on left)	TPC command	SS command
000	"DOWN"	"DOWN"
100	"UP"	"DOWN"
011	"DOWN"	"UP"
111	"UP"	"UP"
001	"DOWN"	"Do Nothing"
101	"UP"	"Do Nothing"

Let  $I=14$  be the number of TPC/SS command pairs that can be carried by a single PLCCCH. The 3 bits corresponding to the  $i^{\text{th}}$  TPC/SS command pair ( $i=1\dots I$ ), are denoted  $\{B_0^i, B_1^i, B_2^i\}$  where  $B_0^i$  is the MSB.

The bit sequence corresponding to the  $I=14$  TPC/SS command pairs is denoted  $\alpha_k$  ( $k=0,1,2,\dots,41$ ).  $B_n^i$  ( $n=0,1,2$ ) is mapped to  $\alpha_k$  such that:

$$\alpha_{3(i-1)+n} = B_n^i$$

The PLCCCH burst is composed of 44 information bits  $\{b_{plcch}(0), b_{plcch}(1), \dots, b_{plcch}(43)\}$  which are repetition coded, and then delivered in one sub-frame as follows:

1.  $b_{plcch}(m) = \alpha_m$  (for  $m=0,1,\dots,41$ ) and  $b_{plcch}(m) = 0$  (for  $m=42,43$ )
2. Repetition coding with code rate  $\frac{1}{2}$  is applied to the sequence  $\{b_{plcch}(0), \dots, b_{plcch}(43)\}$  in order to form the sequence  $\{x(0), x(1), x(2), x(3), \dots, x(86), x(87)\}$ . The size of the data block after the repetition encoder is 88 bits. The encoded codeword  $\{x(0), x(1), x(2), x(3), \dots, x(86), x(87)\}$  is equal to  $\{b_{plcch}(0), b_{plcch}(0), b_{plcch}(1), b_{plcch}(1), \dots, b_{plcch}(43), b_{plcch}(43)\}$ .
3. The bits output from the repetition encoder  $\{x(0), \dots, x(87)\}$  are input to an interleaving unit. The coded bits are block rectangular interleaved according to the following rule: the input is written row by row, the output is read column by column.

$$\begin{bmatrix} x(0) & x(1) & x(2) & \dots & x(7) \\ x(8) & x(9) & x(10) & \dots & x(15) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x(80) & x(81) & x(82) & \dots & x(87) \end{bmatrix}$$

Hence, the interleaved sequence is denoted by  $y(i)$  and is given by:

$$y(0), y(1), \dots, y(87) = x(0), x(8), \dots, x(80), x(1), \dots, x(87).$$

The bit sequence  $y(0), y(1), \dots, y(87)$  is mapped to the PLCCCH burst in order of bit index, with the lowest bit index being the first bit (in time) to be transmitted.

## 4.5 Coding for HS-DSCH

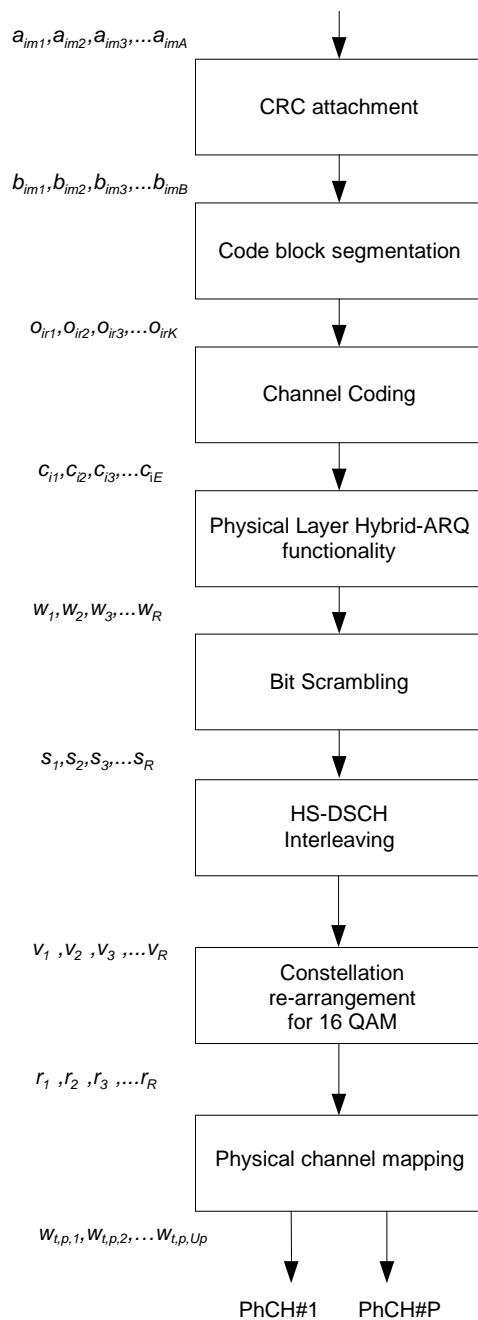
Figure 15 illustrates the overall concept of transport-channel coding for HS-DSCH. Data arrives at the coding unit in the form of one transport block once every TTI. The TTI is 5 ms for 1.28 Mcps TDD and 10 ms for 3.84 Mcps TDD / 7.68 Mcps TDD.

For 1.28 Mcps TDD, in the case of multiple-frequency transmission in one TTI, a number of transport blocks arrive at the coding unit, in which the number of transport blocks equals the number of frequencies used. Each transport block for each frequency shall be coded separately as the following coding step.

The following coding steps for HS-DSCH can be identified:

- add CRC to each transport block (see subclause 4.5.1);
- code block segmentation (see subclause 4.5.2);
- channel coding (see subclause 4.5.3);
- hybrid ARQ (see subclause 4.5.4);
- bit scrambling (see subclause 4.5.5);
- interleaving for HS-DSCH (see subclause 4.5.6);
- constellation re-arrangement for 16QAM (see subclause 4.5.7);
- mapping to physical channels (see subclause 4.5.8).

The coding steps for HS-DSCH are shown in figure 15.



**Figure 15: Coding chain for HS-DSCH**

In the following the number of transport blocks is always one. When referencing non HS-DSCH formulae which are used in correspondence with HS-DSCH formulae the convention is used that transport block subscripts may be omitted (e.g.  $X_i$  when  $i$  is always 1 may be written  $X$ ).

### 4.5.1 CRC attachment for HS-DSCH

A CRC of size 24 bits is calculated and added per HS-DSCH TTI. The CRC polynomial is defined in 4.2.1.1 with the following specific parameters:  $i = 1$ ,  $L_i = 24$  bits.

### 4.5.2 Code block segmentation for HS-DSCH

Code block segmentation for the HS-DSCH transport channel shall be done with the general method described in 4.2.2.2 above with the following specific parameters.



There will only be one transport block,  $i = 1$ . The bits  $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB}$  input to the block are mapped to the bits  $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iXL}$  directly. It follows that  $X_I = B$ . Note that the bits  $x$  referenced here refer only to the internals of the code block segmentation function. The output bits from the code block segmentation function are  $O_{ir1}, O_{ir2}, O_{ir3}, \dots, O_{irK}$ .

The value of  $Z = 5114$  for turbo coding shall be used.

### 4.5.3 Channel coding for HS-DSCH

Channel coding for the HS-DSCH transport channel shall be done with the general method described in 4.2.3 above with the following specific parameters.

There will be a maximum of one transport block,  $i = 1$ . The rate 1/3 turbo coding shall be used.

### 4.5.4 Hybrid ARQ for HS-DSCH

The hybrid ARQ functionality matches the number of bits at the output of the channel coder to the total number of bits of the HS-PDSCH set to which the HS-DSCH is mapped. The hybrid ARQ functionality is controlled by the redundancy version (RV) parameters. The exact set of bits at the output of the hybrid ARQ functionality depends on the number of input bits, the number of output bits, and the RV parameters.

The hybrid ARQ functionality consists of two rate-matching stages and a virtual buffer as shown in the figure below.

The first rate matching stage matches the number of input bits to the virtual IR buffer, information about which is provided by higher layers. Note that, if the number of input bits does not exceed the virtual IR buffering capability, the first rate-matching stage is transparent.

The second rate matching stage matches the number of bits after first rate matching stage to the number of physical channel bits available in the HS-PDSCH set in the TTI.

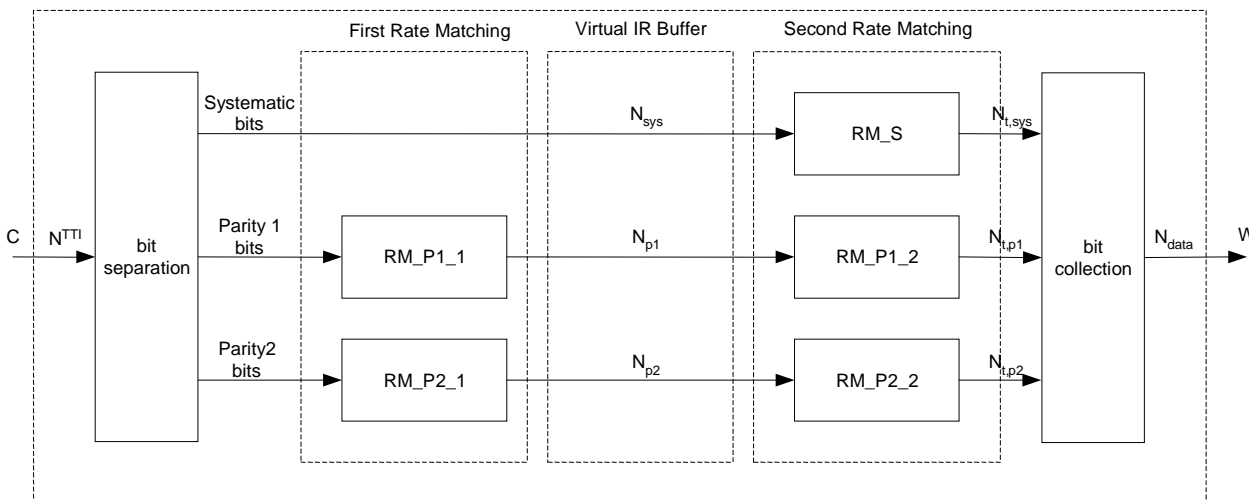


Figure 16: HS-DSCH hybrid ARQ functionality

#### 4.5.4.1 HARQ bit separation

The HARQ bit separation function shall be performed in the same way as bit separation for turbo encoded TrCHs in 4.2.7.2 above.

#### 4.5.4.2 HARQ First Rate Matching Stage

HARQ first stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.1.2 above with the following specific parameters.

The maximum number of soft channel bits available in the virtual IR buffer is  $N_{IR}$  which is signalled from higher layers for each HARQ process. The number of coded bits in a TTI before rate matching is  $N^{TTI}$  this is deduced from

information signalled from higher layers and parameters signalled on the HS-SCCH for each TTI. Note that HARQ processing and physical layer storage occurs independently for each HARQ process currently active.

If  $N_{IR}$  is greater than or equal to  $N^{TTI}$  (i.e. all coded bits of the corresponding TTI can be stored) the first rate matching stage shall be transparent. This can, for example, be achieved by setting  $e_{minus} = 0$ . Note that no repetition is performed.

If  $N_{IR}$  is smaller than  $N^{TTI}$  the parity bit streams are punctured as in 4.2.7.1.2 above by setting the rate matching parameter  $\Delta N_{il}^{TTI} = N_{IR} - N^{TTI}$  where the subscripts  $i$  and  $l$  refer to transport channel and transport format in the referenced sub-clause. Note the negative value is expected when the rate matching implements puncturing. Bits selected for puncturing which appear as  $\delta$  in the algorithm in 4.2.7 above shall be discarded and not counted in the totals for the streams through the virtual IR buffer.

#### 4.5.4.3 HARQ Second Rate Matching Stage

HARQ second stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.3 above with the following specific parameters. Bits selected for puncturing which appear as  $\delta$  in the algorithm in 4.2.7.3 above shall be discarded and are not counted in the streams towards the bit collection.

The parameters of the second rate matching stage depend on the value of the RV parameters  $s$  and  $r$ . The parameter  $s$  can take the value 0 or 1 to distinguish between transmissions that prioritise systematic bits ( $s = 1$ ) and non systematic bits ( $s = 0$ ). The parameter  $r$  (range 0 to  $r_{max}-1$ ) changes the initial error variable  $e_{ini}$  in the case of puncturing. In case of repetition both parameters  $r$  and  $s$  change the initial error variable  $e_{ini}$ . The parameters  $X_i$ ,  $e_{plus}$  and  $e_{minus}$  are calculated as per table 14 below.

Denote the number of bits before second rate matching as  $N_{sys}$  for the systematic bits,  $N_{p1}$  for the parity 1 bits, and  $N_{p2}$  for the parity 2 bits, respectively. For the HS-DSCH, denote the number of timeslots used as  $T$ , the number of codes per timeslot as  $C$  and the number of bits available in timeslot  $t$  as  $U_t$ , where  $U_t = C \times N_{Data/Slot}$  and  $N_{Data/Slot}$  is as defined in

[7] for timeslot  $t$ .  $N_{data}$  is the number of bits available to the HS-DSCH in one TTI and is defined as  $N_{data} = \sum_{t=1}^T U_t$ .

The rate matching parameters are determined as follows.

For  $N_{data} \leq N_{sys} + N_{p1} + N_{p2}$ , puncturing is performed in the second rate matching stage. The number of transmitted systematic bits in a transmission is  $N_{t,sys} = \min\{N_{sys}, N_{data}\}$  for a transmission that prioritises systematic bits and  $N_{t,sys} = \max\{N_{data} - (N_{p1} + N_{p2}), 0\}$  for a transmission that prioritises non systematic bits.

For  $N_{data} > N_{sys} + N_{p1} + N_{p2}$  repetition is performed in the second rate matching stage. A similar repetition rate in

all bit streams is achieved by setting the number of transmitted systematic bits to  $N_{t,sys} = \left\lfloor N_{sys} \cdot \frac{N_{data}}{N_{sys} + 2N_{p1}} \right\rfloor$ .

The number of parity bits in a transmission is:  $N_{t,p1} = \left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor$  and  $N_{t,p2} = \left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor$  for the parity 1 and parity 2 bits, respectively.

Table 14 below summarizes the resulting parameter choice for the second rate matching stage.

**Table 14: Parameters for HARQ second rate matching**

	$X_i$	$e_{plus}$	$e_{minus}$
Systematic RM S	$N_{sys}$	$N_{sys}$	$ N_{sys} - N_{t,sys} $
Parity 1 RM P1_2	$N_{p1}$	$2 \cdot N_{p1}$	$2 \cdot  N_{p1} - N_{t,p1} $
Parity 2 RM P2_2	$N_{p2}$	$N_{p2}$	$ N_{p2} - N_{t,p2} $

The rate matching parameter  $e_{ini}$  is calculated for each bit stream according to the RV parameters  $r$  and  $s$  using

$e_{ini}(r) = \left\{ \left( X_i - \left\lfloor r \cdot e_{plus} / r_{max} \right\rfloor - 1 \right) \bmod e_{plus} \right\} + 1$  in the case of puncturing, i.e.,  $N_{data} \leq N_{sys} + N_{p1} + N_{p2}$ ,  
and

$e_{ini}(r) = \left\{ \left( X_i - \left\lfloor (s + 2 \cdot r) \cdot e_{plus} / (2 \cdot r_{max}) \right\rfloor - 1 \right) \bmod e_{plus} \right\} + 1$  for repetition, i.e.,  $N_{data} > N_{sys} + N_{p1} + N_{p2}$ .

Where  $r \in \{0, 1, \dots, r_{max} - 1\}$  and  $r_{max}$  is the total number of redundancy versions allowed by varying  $r$  as defined in 4.6.1.4. Note that  $r_{max}$  varies depending on the modulation mode, i.e. for 16QAM  $r_{max} = 2$  and for QPSK  $r_{max} = 4$ .

Note: For the modulo operation the following clarification is used: the value of  $(x \bmod y)$  is strictly in the range of 0 to  $y-1$  (i.e.  $-1 \bmod 10 = 9$ ).

#### 4.5.4.4 HARQ bit collection

The HARQ bit collection is achieved using a rectangular interleaver of size  $N_{row} \times N_{col}$ .

The number of rows and columns are determined from:

$$N_{row} = 4 \text{ for 16QAM and } N_{row} = 2 \text{ for QPSK}$$

$$N_{col} = N_{data} / N_{row}$$

where  $N_{data}$  is used as defined in 4.5.4.3 above.

Data is written into the interleaver column by column, and read out of the interleaver column by column, starting from the first column.

$N_{t,sys}$  is the number of transmitted systematic bits. Intermediate values  $N_r$  and  $N_c$  are calculated using:

$$N_r = \left\lfloor \frac{N_{t,sys}}{N_{col}} \right\rfloor \text{ and } N_c = N_{t,sys} - N_r \cdot N_{col}.$$

If  $N_c = 0$  and  $N_r > 0$ , the systematic bits are written into rows  $1 \dots N_r$ .

Otherwise systematic bits are written into rows  $1 \dots N_r + 1$  in the first  $N_c$  columns and, if  $N_r > 0$ , also into rows  $1 \dots N_r$  in the remaining  $N_{col} - N_c$  columns.

The remaining space is filled with parity bits. The parity bits are written column wise into the remaining rows of the respective columns. Parity 1 and 2 bits are written in alternating order, starting with a parity 2 bit in the first available column with the lowest index number.

In the case of 16QAM for each column the bits are read out of the interleaver in the order row 1, row 2, row 3, row 4. In the case of QPSK for each column the bits are read out of the interleaver in the order row1, row2.

#### 4.5.5 Bit scrambling

The bit scrambling for HS-DSCH shall be done with the general method described in subclause 4.2.9.

#### 4.5.6 Interleaving for HS-DSCH

The interleaving for TDD is done over all bits in the TTI, as shown in figure 17 when QPSK modulation is being used for the HS-DSCH, and figure 18 when 16-QAM modulation is being used. The bits input to the block interleaver are denoted by  $s_1, s_2, s_3, \dots, s_R$ , where  $R$  is the number of bits in one TTI.

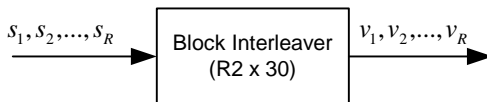


Figure 17: Interleaver structure for HS-DSCH with QPSK modulation

For QPSK, the interleaver is a block interleaver and consists of bits input to a matrix with padding, the inter-column permutation for the matrix and bits output from the matrix with pruning. The output bit sequence from the block interleaver is derived as follows:

- (1) The number of columns of the matrix is 30. The columns of the matrix are numbered 0, 1, 2, ..., 29 from left to right.
- (2) Determine the number of rows of the matrix,  $R2$ , by finding minimum integer  $R2$  such that  $R \leq 30 \times R2$ . The rows of rectangular matrix are numbered 0, 1, 2, ...,  $R2 - 1$  from top to bottom.
- (3) Write the input bit sequence  $s_1, s_2, s_3, \dots, s_R$  into the  $R2 \times 30$  matrix row by row starting with bit  $y_1$  in column 0 of row 0:

$$\begin{bmatrix} y_1 & y_2 & y_3 & \dots & y_{30} \\ y_{31} & y_{32} & y_{33} & \dots & y_{60} \\ \vdots & \vdots & \vdots & & \vdots \\ y_{30 \cdot (R2-1)+1} & y_{30 \cdot (R2-1)+2} & y_{30 \cdot (R2-1)+3} & \dots & y_{30 \cdot R2} \end{bmatrix}$$

where  $y_k = s_k$  for  $k = 1, 2, \dots, R$  and, if  $R < 30 \times R2$ , dummy bits are inserted for  $k = R+1, R+2, \dots, 30 \times R2$ . These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

- (4) Perform the inter-column permutation for the matrix based on the pattern  $\langle P2(j) \rangle_{j \in \{0,1,\dots,29\}}$  that is shown in Table 7, where  $P2(j)$  is the original column position of the  $j$ -th permuted column. After permutation of the columns, the bits are denoted by  $y''_k$ .

$$\begin{bmatrix} y'_1 & y'_{R2+1} & y'_{2 \cdot R2+1} & \dots & y'_{29 \cdot R2+1} \\ y'_2 & y'_{R2+2} & y'_{2 \cdot R2+2} & \dots & y'_{29 \cdot R2+2} \\ \vdots & \vdots & \vdots & & \vdots \\ y'_{R2} & y'_{2 \cdot R2} & y'_{3 \cdot R2} & \dots & y'_{30 \cdot R2} \end{bmatrix}$$

- (5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted  $R2 \times 30$  matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits  $y''_k$  that corresponds to bits  $y_k$  with  $k > R$  are removed from the output. The bits after interleaving are denoted by  $v_1, v_2, v_3, \dots, v_R$ , where  $v_1$  corresponds to the bit  $y''_k$  with smallest index  $k$  after pruning,  $v_2$  to the bit  $y''_k$  with second smallest index  $k$  after pruning, and so on.

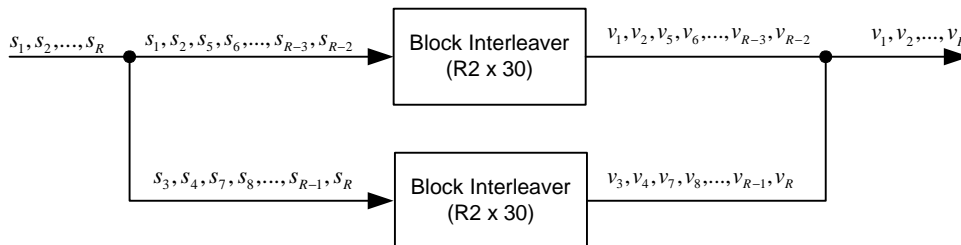


Figure 18: Interleaver structure for HS-DSCH with 16-QAM modulation

For 16QAM, a second identical interleaver operates in parallel to the first. For both interleavers,  $R2$  is chosen to be the minimum integer that satisfies  $R \leq 60 \times R2$ . The output bits from the bit scrambling operation are divided pairwise between the interleavers: bits  $s_k$  and  $s_{k+1}$  go to the first interleaver and bits  $s_{k+2}$  and  $s_{k+3}$  go to the second interleaver,

where  $k \bmod 4 = 1$ . Bits are collected pairwise from the interleavers: bits  $v_k$  and  $v_{k+1}$  are obtained from the first interleaver and bits  $v_{k+2}$  and  $v_{k+3}$  are obtained from the second interleaver, where again  $k \bmod 4 = 1$ .

#### 4.5.7 Constellation re-arrangement for 16 QAM

This function only applies to 16 QAM modulated bits. In case of QPSK it is transparent.

The following table 15 describes the operations that produce the different rearrangements.

The bits of the input sequence are mapped in groups of 4 so that  $v_k, v_{k+1}, v_{k+2}, v_{k+3}$  are used, where  $k \bmod 4 = 1$ .

**Table 15: Constellation re-arrangement for 16 QAM**

Constellation version parameter $b$	Output bit sequence	Operation
0	$v_k v_{k+1} v_{k+2} v_{k+3}$	None
1	$v_{k+2} v_{k+3} v_k v_{k+1}$	Swapping MSBs with LSBs
2	$v_k v_{k+1} v_{k+2} v_{k+3}$	Inversion of the logical values of LSBs
3	$v_{k+2} v_{k+3} v_k v_{k+1}$	Swapping MSBs with LSBs, and inversion of the logical values of LSBs

The output bit sequences from the table above map to the output bits in groups of 4, i.e.  $r_k, r_{k+1}, r_{k+2}, r_{k+3}$ , where  $k \bmod 4 = 1$ .

#### 4.5.8 Physical channel mapping for HS-DSCH

The HS-PDSCH is defined in [7]. The bits input to the physical channel mapping are denoted by  $r_1, r_2, \dots, r_R$ , where  $R$  is the number of physical channel bits in the allocation for the current TTI. These bits are mapped to the physical channel bits,  $\{w_{t,p,j} : t = 1, 2, \dots, T; p = 1, 2, \dots, C; j = 1, 2, \dots, U_t\}$ , where  $t$  is the timeslot index,  $T$  is the number of timeslots in the allocation message,  $p$  is the physical channel index,  $C$  is the number of codes per timeslot in the allocation message,  $j$  is the physical channel bit index and  $U_t$  is the number of bits per physical channel in timeslot  $t$ . The timeslot index,  $t$ , increases with increasing timeslot number; the physical channel index,  $p$ , increases with increasing channelisation code index, and the physical channel bit index,  $j$ , increases with increasing physical channel bit position in time.

The bits  $r_k$  shall be mapped to the PhCHs according to the following rule :

Define  $\{y_{t,k} : k = 1, 2, \dots, C \cdot U_t\}$  to be the set of bits to be transmitted in timeslot  $t$  as follows :

$$y_{1,k} = r_k \quad \text{for } k = 1, 2, \dots, C \cdot U_1$$

$$y_{2,k} = r_{k+C \cdot U_1} \quad \text{for } k = 1, 2, \dots, C \cdot U_2$$

...

$$y_{T,k} = r_{k+C \sum_{t=1}^{T-1} U_t} \quad \text{for } k = 1, 2, \dots, C \cdot U_T$$

When the modulation level applied to the physical channels is 16- QAM :

The physical channel  $p$  used to transmit the  $k^{\text{th}}$  bit in the sequence  $y_{t,k}$  is :

$$p = \left\lfloor \frac{k-1}{4} \right\rfloor \bmod C + 1$$

If  $p$  is odd then :

$$w_{t,p,j} = y_{t,k} \text{ where } j = 4 \cdot \left\lfloor \frac{k-1}{4 \cdot C} \right\rfloor + (k-1) \bmod 4 + 1$$

If  $p$  is even then :

$$w_{t,p,j} = y_{t,k} \text{ where } j = U_t - 4 \cdot \left\lfloor \frac{k-1}{4 \cdot C} \right\rfloor - 3 + (k-1) \bmod 4$$

Otherwise, when the modulation level applied to the physical channels is QPSK :

The physical channel  $p$  used to transmit the  $k^{\text{th}}$  bit in the sequence  $y_{t,k}$  is :

$$p = (k-1) \bmod C + 1$$

If  $p$  is odd then :

$$w_{t,p,j} = y_{t,k} \text{ where } j = \left\lfloor \frac{k-1}{C} \right\rfloor + 1$$

If  $p$  is even then :

$$w_{t,p,j} = y_{t,k} \text{ where } j = U_t - \left\lfloor \frac{k-1}{C} \right\rfloor$$

## 4.6 Coding/Multiplexing for HS-SCCH

The following information, provided by higher layers, is transmitted by means of the HS-SCCH physical channel.

For 1.28 Mcps TDD, in the case of multi-frequency HS-DSCH transmission in one TTI, HS-PDSCH on each frequency shall be configured with associated HS-SCCH(s) which is coded and multiplexed as following.

- Channelisation-code-set information ( $q$  bits where  $q = 8$  for 1.28Mcps TDD / 3.84Mcps TDD and  $q = 10$  for 7.68Mcps TDD):  $x_{ccs,1}, x_{ccs,2}, \dots, x_{ccs,q}$
- Time slot information ( $n$  bits where  $n = 5$  for 1.28 Mcps TDD and  $n = 13$  for 3.84 Mcps TDD / 7.68Mcps TDD):  $x_{ts,1}, x_{ts,2}, \dots, x_{ts,n}$
- Modulation scheme information (1 bit):  $x_{ms,1}$
- Transport-block size information (m bits where  $m = 6$  for 1.28 Mcps TDD and  $m = 9$  for 3.84 Mcps TDD / 7.68Mcps TDD):  $x_{tbs,1}, x_{tbs,2}, \dots, x_{tbs,m}$
- Hybrid-ARQ process information (3 bits):  $x_{hap,1}, x_{hap,2}, x_{hap,3}$
- Redundancy version information (3 bits):  $x_{rv,1}, x_{rv,2}, x_{rv,3}$
- New data indicator (1 bit):  $x_{nd,1}$
- HS-SCCH cyclic sequence number (3 bits):  $x_{hcsn,1}, x_{hcsn,2}, x_{hcsn,3}$
- UE identity (16 bits):  $x_{ue,1}, x_{ue,2}, \dots, x_{ue,16}$

The following coding/multiplexing steps can be identified:

- multiplexing of HS-SCCH information (see subclause 4.6.2)
- CRC attachment (see subclause 4.6.3);
- channel coding (see subclause 4.6.4);

- rate matching (see subclause 4.6.5);
- interleaving for HS-SCCH (see subclause 4.6.6);
- mapping to physical channels (see subclauses 4.6.7 and 4.6.8).

The general coding/multiplexing flow is shown in Figure 19.

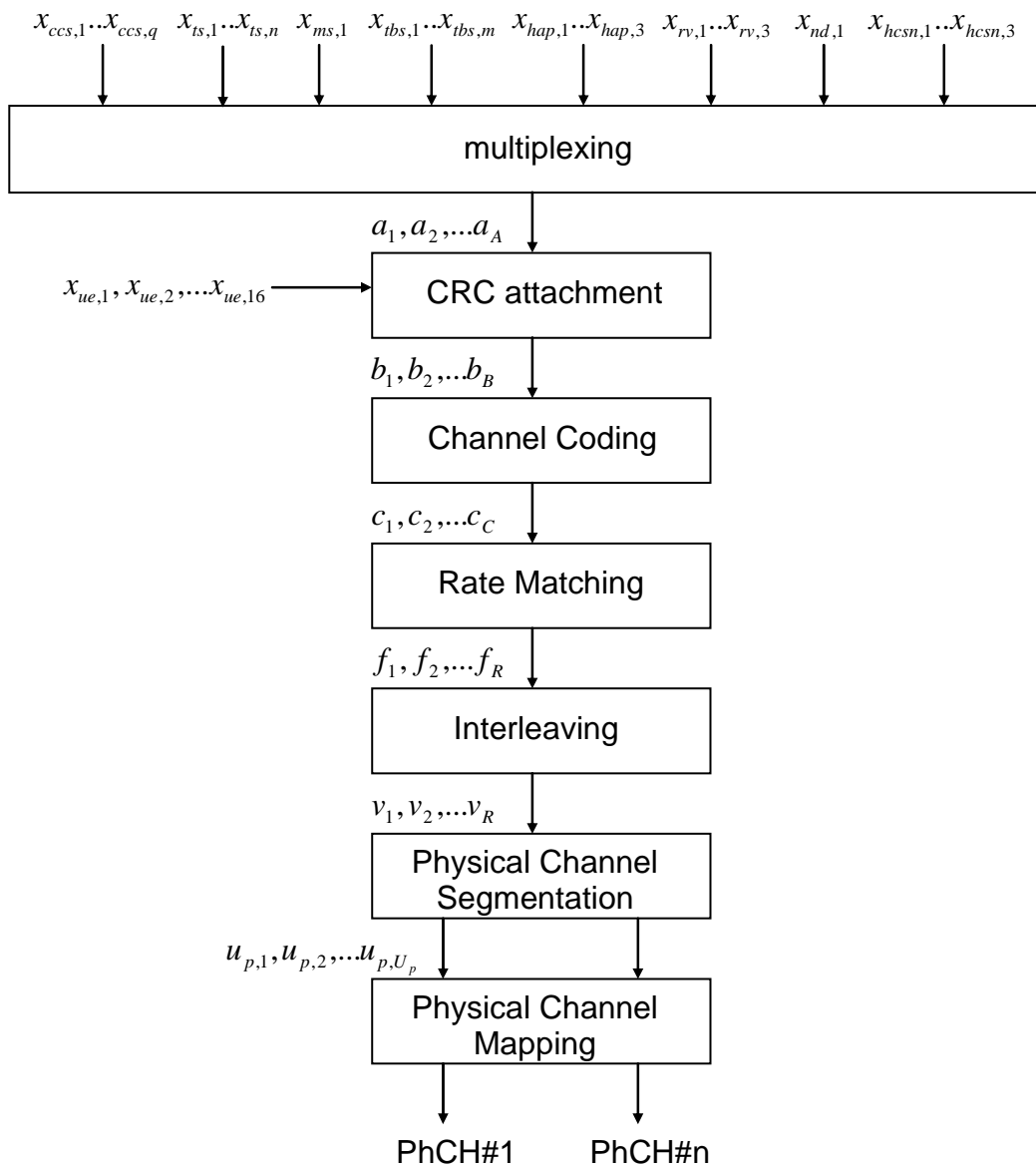


Figure 19 Coding and Multiplexing for HS-SCCH

## 4.6.1 HS-SCCH information field mapping

### 4.6.1.1 Channelisation code set information mapping

#### 4.6.1.1.1 1.28Mcps TDD and 3.84Mcps TDD

HS-PDSCH channelisation codes are allocated contiguously from a signalled start code to a signalled stop code, and the allocation includes both the start and stop code. The start code  $k_{start}$  is signalled by the bits  $x_{ccs,1}, x_{ccs,2}, x_{ccs,3}, x_{ccs,4}$  and the stop code  $k_{stop}$  by the bits  $x_{ccs,5}, x_{ccs,6}, x_{ccs,7}, x_{ccs,8}$ . The mapping in Table 16 below applies.

**Table 16: Channelisation code set information mapping for 1.28Mcps and 3.84Mcps TDD**

$k_{start}$	$x_{ccs,1}$	$x_{ccs,2}$	$x_{ccs,3}$	$x_{ccs,4}$	$k_{stop}$	$x_{ccs,5}$	$x_{ccs,6}$	$x_{ccs,7}$	$x_{ccs,8}$
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	2	0	0	0	1
3	0	0	1	0	3	0	0	1	0
4	0	0	1	1	4	0	0	1	1
5	0	1	0	0	5	0	1	0	0
6	0	1	0	1	6	0	1	0	1
7	0	1	1	0	7	0	1	1	0
8	0	1	1	1	8	0	1	1	1
9	1	0	0	0	9	1	0	0	0
10	1	0	0	1	10	1	0	0	1
11	1	0	1	0	11	1	0	1	0
12	1	0	1	1	12	1	0	1	1
13	1	1	0	0	13	1	1	0	0
14	1	1	0	1	14	1	1	0	1
15	1	1	1	0	15	1	1	1	0
16	1	1	1	1	16	1	1	1	1

If a value of  $k_{start} = 16$  and  $k_{stop} = 1$  is signalled, a spreading factor of SF=1 shall be used for the HS-PDSCH resources. Other than this case,  $k_{start} > k_{stop}$  shall be treated as an error by the UE.

#### 4.6.1.1.2 7.68Mcps TDD

HS-PDSCH channelisation codes are allocated contiguously from a signalled start code to a signalled stop code, and the allocation includes both the start and stop code. The start code  $k_{start}$  is signalled by the bits  $x_{ccs,1}, x_{ccs,2}, x_{ccs,3}, x_{ccs,4}, x_{ccs,5}$  and the stop code  $k_{stop}$  by the bits  $x_{ccs,6}, x_{ccs,7}, x_{ccs,8}, x_{ccs,9}, x_{ccs,10}$ . The mapping in Table 16A below applies.



Table 16A: Channelisation code set information mapping for 7.68Mcps TDD

$k_{start}$	$x_{ccs,1}$	$x_{ccs,2}$	$x_{ccs,3}$	$x_{ccs,4}$	$x_{ccs,5}$	$k_{stop}$	$x_{ccs,6}$	$x_{ccs,7}$	$x_{ccs,8}$	$x_{ccs,9}$	$x_{ccs,10}$
1	0	0	0	0	0	1	0	0	0	0	0
2	0	0	0	0	1	2	0	0	0	0	1
3	0	0	0	1	0	3	0	0	0	1	0
4	0	0	0	1	1	4	0	0	0	1	1
5	0	0	1	0	0	5	0	0	1	0	0
6	0	0	1	0	1	6	0	0	1	0	1
7	0	0	1	1	0	7	0	0	1	1	0
8	0	0	1	1	1	8	0	0	1	1	1
9	0	1	0	0	0	9	0	1	0	0	0
10	0	1	0	0	1	10	0	1	0	0	1
11	0	1	0	1	0	11	0	1	0	1	0
12	0	1	0	1	1	12	0	1	0	1	1
13	0	1	1	0	0	13	0	1	1	0	0
14	0	1	1	0	1	14	0	1	1	0	1
15	0	1	1	1	0	15	0	1	1	1	0
16	0	1	1	1	1	16	0	1	1	1	1
17	1	0	0	0	0	17	1	0	0	0	0
18	1	0	0	0	1	18	1	0	0	0	1
19	1	0	0	1	0	19	1	0	0	1	0
20	1	0	0	1	1	20	1	0	0	1	1
21	1	0	1	0	0	21	1	0	1	0	0
22	1	0	1	0	1	22	1	0	1	0	1
23	1	0	1	1	0	23	1	0	1	1	0
24	1	0	1	1	1	24	1	0	1	1	1
25	1	1	0	0	0	25	1	1	0	0	0
26	1	1	0	0	1	26	1	1	0	0	1
27	1	1	0	1	0	27	1	1	0	1	0
28	1	1	0	1	1	28	1	1	0	1	1
29	1	1	1	0	0	29	1	1	1	0	0
30	1	1	1	0	1	30	1	1	1	0	1
31	1	1	1	1	0	31	1	1	1	1	0
32	1	1	1	1	1	32	1	1	1	1	1

If a value of  $k_{start} = 32$  and  $k_{stop} = 1$  is signalled, a spreading factor of SF=1 shall be used for the HS-PDSCH resources. Other than this case,  $k_{start} > k_{stop}$  shall be treated as an error by the UE.

#### 4.6.1.2 Timeslot information mapping

##### 4.6.1.2.1 1.28 Mcps TDD

For 1.28 Mcps, the timeslots to be used for HS-PDSCH resources are signalled by the bits  $x_{ts,1}, x_{ts,2}, \dots, x_{ts,5}$ , where bit  $x_{ts,n}$  carries the information for timeslot n+1. Timeslots 0 and 1 cannot be used for HS-PDSCH resources. If the signalling bit is set (i.e. equal to 1), then the corresponding timeslot shall be used for HS-PDSCH resources. Otherwise, the timeslot shall not be used. All used timeslots shall use the same channelisation code set, as signalled by the channelisation code set information bits.

##### 4.6.1.2.2 3.84 Mcps TDD and 7.68Mcps TDD

For 3.84 Mcps, the timeslots to be used for HS-PDSCH resources are signalled by the bits  $x_{ts,1}, x_{ts,2}, \dots, x_{ts,13}$ , where bit  $x_{ts,n}$  carries the information for the n<sup>th</sup> available timeslot for HS-PDSCH resources, where the order of the timeslots available for HS-PDSCH resources shall be the same as the order of the 15 time slots within each frame with the following two slots removed:

- The slot containing the P-CCPCH
- The first slot in a frame containing the PRACH

If the P-CCPCH and/or PRACH are assigned to some, but not all frames, then the corresponding time slots shall remain unavailable for these frames as well..

If the bit is set (i.e. equal to 1), then the corresponding timeslot shall be used for HS-PDSCH resources. Otherwise, the timeslot shall not be used. All used timeslots shall use the same channelisation code set, as signalled by the channelisation code set information bits.

#### 4.6.1.3 Modulation scheme information mapping

The modulation scheme to be used by the HS-PDSCH resources shall be signalled by bit  $x_{ms,1}$ . The mapping scheme in Table 17 shall apply.

**Table 17: Modulation scheme information mapping**

$x_{ms,1}$	Modulation Scheme
0	QPSK
1	16-QAM

#### 4.6.1.4 Redundancy and constellation version information mapping

The redundancy version (RV) parameters  $r$ ,  $s$  and constellation version parameter  $b$  are mapped jointly to produce the value  $X_{rv}$ .  $X_{rv}$  is alternatively represented as the sequence  $x_{rv,1}$ ,  $x_{rv,2}$ ,  $x_{rv,3}$  where  $x_{rv,1}$  is the MSB. This is done according to the following tables according to the modulation mode used:

**Table 18: RV mapping for 16 QAM**

$X_{rv}$ (value)	$s$	$r$	$b$
0	1	0	0
1	0	0	0
2	1	1	1
3	0	1	1
4	1	0	1
5	1	0	2
6	1	0	3
7	1	1	0

**Table 19: RV mapping for QPSK**

$X_{rv}$ (value)	$s$	$r$
0	1	0
1	0	0
2	1	1
3	0	1
4	1	2
5	0	2
6	1	3
7	0	3

#### 4.6.1.5 HS-SCCH cyclic sequence number

The HS-SCCH cyclic sequence number is mapped such that  $x_{hcsn,1}$  corresponds to the MSB and  $x_{hcsn,3}$  to the LSB.

#### 4.6.1.6 UE identity

The UE identity is the HS-DSCH Radio Network Identifier (H-RNTI) defined in [12]. This is mapped such that  $x_{ue,1}$  corresponds to the MSB and  $x_{ue,16}$  to the LSB, cf. [14].

#### 4.6.1.7 HARQ process identifier mapping

The hybrid-ARQ process information  $x_{hap,1}, x_{hap,2}, x_{hap,3}$  is unsigned binary representation of the HARQ process identifier where  $x_{hap,1}$  is MSB.

#### 4.6.1.8 Transport block size index mapping

The transport-block size information  $x_{tbs,1}, x_{tbs,2}, \dots, x_{tbs,m}$  is unsigned binary representation of the transport block size index where  $x_{tbs,1}$  is MSB.

### 4.6.2 Multiplexing of HS-SCCH information

The information carried on the HS-SCCH is multiplexed onto the bits  $a_1, a_2, \dots, a_A$  according to the following rule :

$$a_1, a_2 \dots a_q = x_{ccs,1}, x_{ccs,2} \dots x_{ccs,q}$$

$$a_{q+1}, a_{q+2} \dots a_{q+n} = x_{ts,1}, x_{ts,2} \dots x_{ts,n}$$

$$a_{q+n+1} = x_{ms,1}$$

$$a_{q+n+2}, a_{q+n+3} \dots a_{q+n+m+1} = x_{tbs,1}, x_{tbs,2} \dots x_{tbs,m}$$

$$a_{q+n+m+2}, a_{q+n+m+3}, a_{q+n+m+4} = x_{hap,1}, x_{hap,2}, x_{hap,3}$$

$$a_{q+n+m+5}, a_{q+n+m+6}, a_{q+n+m+7} = x_{rv,1}, x_{rv,2}, x_{rv,3}$$

$$a_{q+n+m+8} = x_{nd,1}$$

$$a_{q+n+m+9}, a_{q+n+m+10}, a_{q+n+m+11} = x_{hcsn,1}, x_{hcsn,2}, x_{hcsn,3}$$

### 4.6.3 CRC attachment for HS-SCCH

From the sequence of bits  $a_1, a_2, \dots, a_A$  a 16 bit CRC is calculated according to Section 4.2.1.1. This gives a sequence of bits  $y_1, y_2, \dots, y_{16}$  where

$$y_k = p_{im(17-k)} \quad k = 1, 2, \dots, 16$$

This latter sequence of bits is then masked with the UE identity and appended to the sequence of bits  $a_1, a_2, \dots, a_A$ . The bits at the output of the CRC attachment block is the sequence of bits  $b_1, b_2, \dots, b_B$ , where

$$b_i = a_i \quad i=1, 2, \dots, A$$

$$b_i = (y_{i-A} + x_{ue,i-A}) \bmod 2 \quad i=A+1 \dots B$$

### 4.6.4 Channel coding for HS-SCCH

Channel coding for the HS-SCCH shall be done with the general method described in 4.2.3 with the following specific parameters:

The rate 1/3 convolutional coding shall be used for HS-SCCH.

#### 4.6.5 Rate matching for HS-SCCH

Rate matching for HS-SCCH shall be done with the general method described in 4.2.7.

#### 4.6.6 Interleaving for HS-SCCH

Interleaving for HS-SCCH shall be done with the general method described in 4.2.11.1.

#### 4.6.7 Physical Channel Segmentation for HS-SCCH

Physical channel segmentation for HS-SCCH shall be done with the general method described in 4.2.10. For 1.28 Mcps TDD, the HS-SCCH consists of two physical channels HS-SCCH1 and HS-SCCH2; for 3.84 Mcps TDD and 7.68 Mcps TDD the HS-SCCH only uses one physical channel, see [7].

#### 4.6.8 Physical channel mapping for HS-SCCH

Physical channel mapping for the HS-SCCH shall be done with the general method described in subclause 4.2.12.

### 4.7 Coding for HS-SICH

The following information, provided by higher layers, is transmitted by means of the HS-SICH physical channel.

For 1.28 Mcps TDD, in the case of multi-frequency HS-DSCH transmission in one TTI, HS-PDSCH on each frequency shall be configured with associated HS-SICH(s) which is coded as following.

- Recommended Modulation Format (RMF) (1 bit):  $x_{rmf,1}$
- Recommended Transport-block size (RTBS) (n bits where n = 6 for 1.28 Mcps TDD and n = 9 for 3.84 Mcps TDD and 7.68 Mcps TDD):  $x_{tbs,1}, x_{tbs,2}, \dots, x_{tbs,n}$
- Hybrid-ARQ information ACK/NACK (1 bit):  $x_{an,1}$

The following coding/multiplexing steps can be identified:

- separate coding of RMF, RTBS and ACK/NACK (see subclause 4.7.2);
- multiplexing of HS-SICH information (4.7.3);
- interleaving for HS-SICH (see subclause 4.7.4);
- mapping to physical channels (see subclause 4.7.5).

The general coding/multiplexing flow is shown in the figure 20.

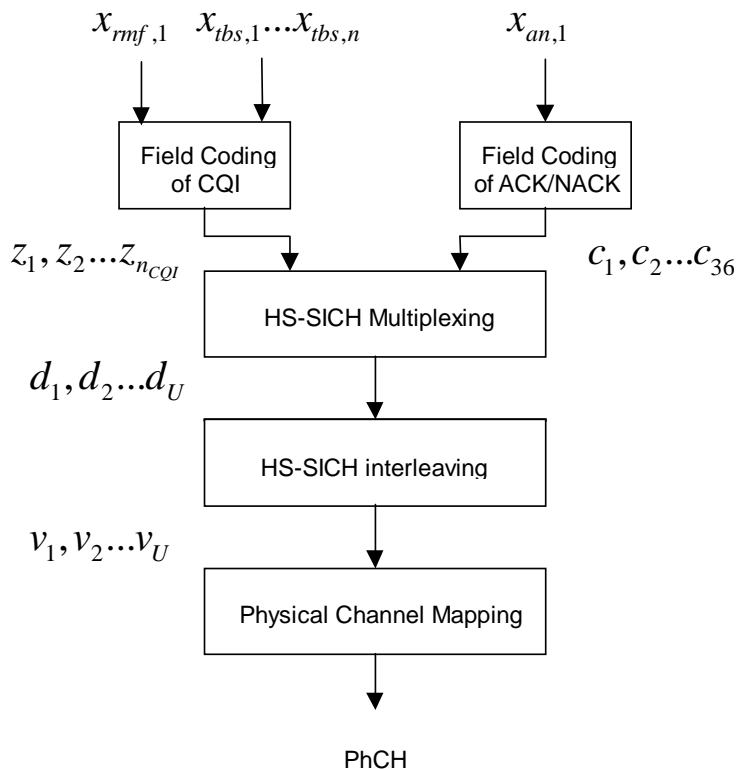


Figure 20 Coding and multiplexing for HS-SICH

### 4.7.1 HS-SICH information field mapping

#### 4.7.1.1 RMF information mapping

The RMF information bit,  $x_{rmf,1}$ , shall be mapped according to the mapping specified in subclause 4.6.1.3.

#### 4.7.1.2 RTBS information mapping

The RTBS information bits,  $x_{tbs,1}, x_{tbs,2}, \dots, x_{tbs,n}$ , shall be mapped according to the same mapping as is used for the transport block size information bits in subclause 4.6. This mapping is defined by higher layers [12].

#### 4.7.1.3 ACK/NACK information mapping

The ACK/NACK information bit  $x_{an,1}$  shall be mapped according to the mapping given in Table 20 below.

Table 20: ACK/NACK information mapping

ACK/NACK	$x_{an,1}$
ACK	1
NACK	0

## 4.7.2 Coding for HS-SICH

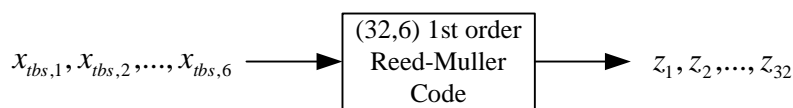
### 4.7.2.1 Field Coding of ACK/NACK

The ACK/NACK bit  $x_{an,l}$  shall be repetition coded to 36 bits. The coded bits are defined as  $c_1 \dots c_{36}$

### 4.7.2.2 Field Coding of CQI

#### 4.7.2.2.1 Field Coding of CQI for 1.28 Mcps TDD

The quality information consists of Recommended Transport Block Size (RTBS) and Recommended Modulation Format (RMF) fields. The 6 bits of the RTBS field are coded to 32 bits using a (32, 6) 1<sup>st</sup> order Reed-Muller code. The coding procedure is as shown in figure 21.



**Figure 21: Field coding of RTBS information bits**

The coding uses a subset basis sequences as the TFCI coder as described in subclause 4.3.1.1. The basis sequences that are used for RTBS coding are as follows in table 21.

**Table 21: Basis sequences for (32,6) RTBS code**

i	M <sub>i,0</sub>	M <sub>i,1</sub>	M <sub>i,2</sub>	M <sub>i,3</sub>	M <sub>i,4</sub>	M <sub>i,5</sub>
0	1	0	0	0	0	1
1	0	1	0	0	0	1
2	1	1	0	0	0	1
3	0	0	1	0	0	1
4	1	0	1	0	0	1
5	0	1	1	0	0	1
6	1	1	1	0	0	1
7	0	0	0	1	0	1
8	1	0	0	1	0	1
9	0	1	0	1	0	1
10	1	1	0	1	0	1
11	0	0	1	1	0	1
12	1	0	1	1	0	1
13	0	1	1	1	0	1
14	1	1	1	1	0	1
15	1	0	0	0	1	1
16	0	1	0	0	1	1
17	1	1	0	0	1	1
18	0	0	1	0	1	1
19	1	0	1	0	1	1
20	0	1	1	0	1	1
21	1	1	1	0	1	1
22	0	0	0	1	1	1
23	1	0	0	1	1	1
24	0	1	0	1	1	1
25	1	1	0	1	1	1
26	0	0	1	1	1	1
27	1	0	1	1	1	1
28	0	1	1	1	1	1
29	1	1	1	1	1	1
30	0	0	0	0	0	1
31	0	0	0	0	1	1

The output RTBS code word bits  $\{z_i : i = 1, \dots, 32\}$  are given by:

$$z_i = \left( \sum_{n=1}^6 x_{tbs,n} \cdot M_{i-1,n-1} \right) \bmod 2$$

The RMF bit  $x_{rmf,1}$  is repetition coded to 16 bits to produce the bits  $z_{33}, z_{34}, \dots, z_{n_{CQI}}$  where  $n_{CQI} = 48$ .

#### 4.7.2.2.2 Field Coding of CQI for 3.84 Mcps TDD and 7.68Mcps TDD

RTBS and RMF bits are multiplexed onto the bits  $y_1, y_2 \dots y_{10}$  according to the following rule :

$$y_1 = x_{rmf,1}$$

$$y_2, y_3 \dots y_{10} = x_{tbs,1}, x_{tbs,2} \dots x_{tbs,9}$$

The bits  $y_1, y_2 \dots y_{10}$  are coded to produce the bits  $w_1, w_2, \dots, w_{32}$  using a (32,10) sub-code of the second order Reed-Muller code as defined in subclause 4.3.1.1..

The bits  $w_1, w_2, \dots, w_{32}$  are used to produce the CQI bits  $z_1, z_2 \dots z_{n_{CQI}}$  using a (4,1) repetition code, where  $n_{CQI}=128$ , such that:

$$z_n, z_{n+32}, z_{n+64}, z_{n+96} = w_n \quad n=1 \dots 32$$

### 4.7.3 Multiplexing of HS-SICH information fields

The CQI bits  $z_1, z_2 \dots z_{n_{CQI}}$  are multiplexed with the repetition coded ACK/NACK bits  $c_1 \dots c_{36}$  to produce the bits  $d_1, d_2 \dots d_U$  where  $U$  is the number of physical channel bits carried by HS-SICH, according to the following rule.:

$$d_1, d_2 \dots d_{n_{CQI}} = z_1, z_2 \dots z_{n_{CQI}}$$

$$d_{n_{CQI}+1}, d_{n_{CQI}+2} \dots d_{n_{CQI}+36} = c_1, c_2 \dots c_{36}$$

$$d_{n_{CQI}+37}, d_{n_{CQI}+38} \dots d_U = 0, 0 \dots 0$$

#### 4.7.4 Interleaver for HS-SICH

Interleaver for HS-SICH shall be done with the general method described in 4.2.11.1.

#### 4.7.5 Physical channel mapping for HS-SICH

Physical channel mapping for HS-SICH shall be done with the general method described in 4.2.12.

### 4.8 Coding for E-DCH

Figure 22 shows the processing structure for the E-DCH transport channel mapped onto a separate CCTrCH. Data arrives to the coding unit in form of a maximum of one transport block once every transmission time interval (TTI). A

10ms TTI is used for E-DCH for 3.84Mcps and 7.68Mcps TDD whilst for 1.28Mcps TDD, a TTI of 5ms will be used. The following coding steps for E-DCH can be identified:

- add CRC to each transport block (see subclause 4.8.1);
- code block segmentation (see subclause 4.8.2);
- channel coding (see subclause 4.8.3);
- hybrid ARQ (see subclause 4.8.4);
- bit scrambling (see subclause 4.8.5);
- interleaving for E-DCH (see subclause 4.8.6);
- constellation re-arrangement for 16QAM (see subclause 4.8.7);
- mapping to physical channels (see subclause 4.8.8).



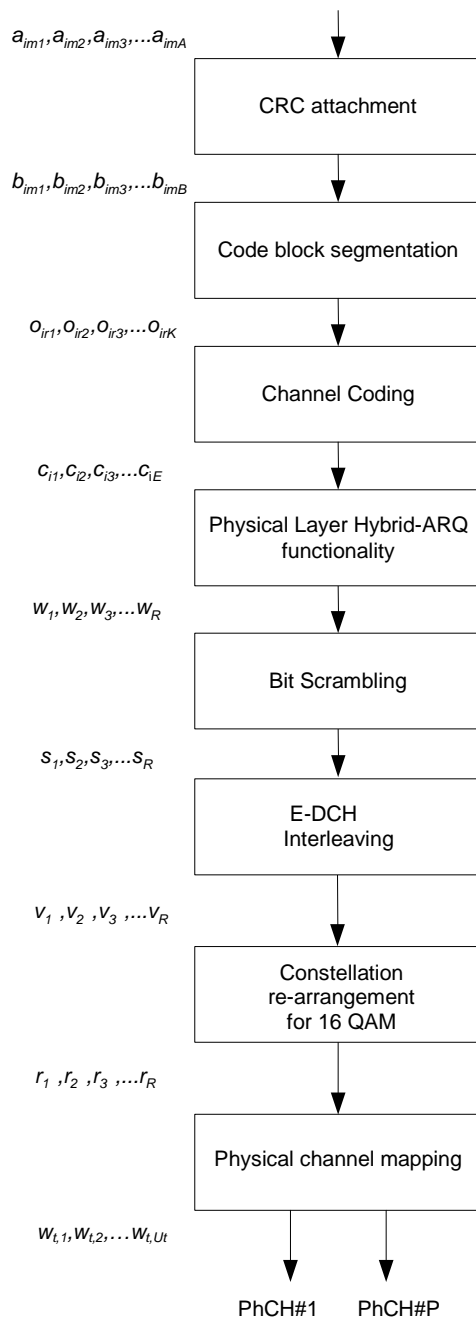


Figure 22. Coding chain for E-DCH

In the following the number of transport blocks is always one. When referencing non E-DCH formulae which are used in correspondence with E-DCH formulae the convention is used that transport block subscripts may be omitted (e.g.  $X_i$  when  $i$  is always 1 may be written  $X$ ).

### 4.8.1 CRC attachment for E-DCH

CRC attachment for the E-DCH transport channel shall be performed according to the general method described in 4.2.1 above with the following specific parameters.

The CRC length shall always be  $L_c=24$  bits.

### 4.8.2 Code block segmentation for E-DCH

Code block segmentation for the E-DCH transport channel shall be performed according to the general method described in 4.2.2.2 with the following specific parameters.

There is a maximum of one transport block. The bits  $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$  input to the block are mapped to the bits  $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$  directly. It follows that  $X_i = B_i$ . Note that the bits  $x$  referenced here refer only to the internals of the code block segmentation function. The output bits from the code block segmentation function are  $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK}$ .

The value of  $Z = 5114$  for turbo coding shall be used.

### 4.8.3 Channel coding for E-DCH

Channel coding for the E-DCH transport channel shall be performed according to the general method described in section 4.2.3 above with the following specific parameters.

There is a maximum of one transport block,  $i=1$ . The rate 1/3 turbo coding shall be used.

### 4.8.4 Physical layer HARQ functionality and rate matching for E-DCH

The hybrid ARQ functionality matches the number of bits at the output of the channel coder to the total number of bits of the E-PUCH set to which the E-DCH transport channel is mapped. The hybrid ARQ functionality is controlled by the redundancy version (RV) parameters.

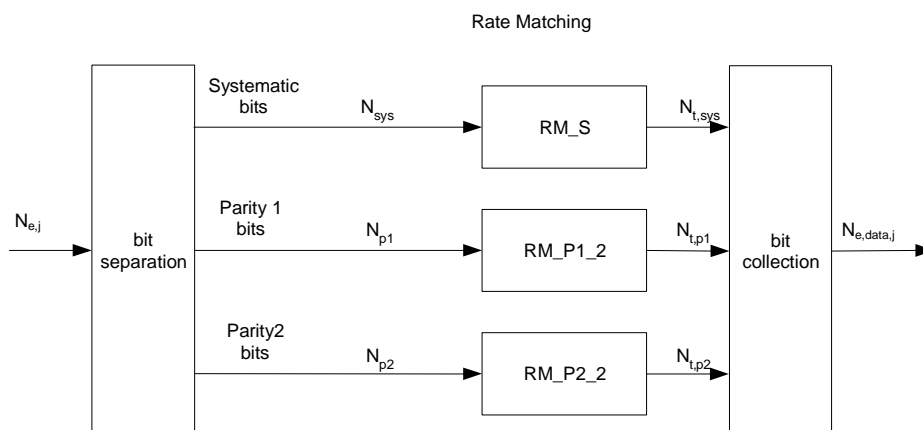


Figure 23: E-DCH hybrid ARQ functionality

#### 4.8.4.1 Determination of SF, modulation and number of physical channels

The SF, modulation type and number of E-PUCHs in the E-PUCH set is determined by higher layers (see [15]). These correspond to a value of  $N_{e,data,j}$ .

#### 4.8.4.2 HARQ bit separation

The HARQ bit separation function shall be performed in the same way as bit separation for turbo encoded TrCHs with puncturing in 4.2.7.2.1 above.

#### 4.8.4.3 HARQ Rate Matching Stage

The hybrid ARQ rate matching for the E-DCH transport channel shall be done with the general method described in 4.2.7.3 with the following specific parameters.

The parameters of the rate matching stage depend on the value of the RV parameters  $s$  and  $r$ . The  $s$  and  $r$  combinations corresponding to each RV allowed for the E-DCH are listed in table 22 below.

Table 22: RV for E-DCH

E-DCH RV Index	<i>s</i>	<i>r</i>
0	1	0
1	0	0
2	1	1
3	0	1

The parameter  $e_{plus}$ ,  $e_{minus}$  and  $e_{ini}$  are calculated with the general method described in 4.5.4.3 above. The following parameters are used as input:

$$N_{sys} = N_{p1} = N_{p2} = N_{e,j}/3$$

$$N_{data} = N_{e,data,j}$$

$$r_{max} = 2 \text{ (for both QPSK and 16-QAM)}$$

#### 4.8.4.4 HARQ bit collection

HARQ bit collection for E-DCH shall be performed according to the general method described for HS-DSCH in subclause 4.5.4.4.

#### 4.8.5 Bit scrambling

The bit scrambling for E-DCH shall be performed in accordance with the general method described in subclause 4.2.9.

#### 4.8.6 Interleaving for E-DCH

Interleaving for E-DCH shall be performed in accordance with the general method described for HS-DSCH in subclause 4.5.6.

#### 4.8.7 Constellation re-arrangement for 16 QAM

Constellation rearrangement shall be performed in the case of 16-QAM in accordance with the general method described for HS-DSCH in subclause 4.5.7. For QPSK this function is transparent.

For 3.84Mcps and 7.68Mcps, the constellation version parameter  $b$  is associated with the E-DCH RV index as shown in table 23 below.

**Table 23: Mapping of RV to constellation rearrangement parameter  $b$  for E-DCH (3.84Mcps and 7.68Mcps options)**

E-DCH RV Index	$b$
0	0
1	1
2	2
3	3

For 1.28Mcps option, the constellation version parameter  $b$  is associated with the retransmission sequence number (RSN). The mapping between RSN and  $b$  parameters for constellation re-arrangement is listed in table 25A in subclause 4.9.2.1.2.

#### 4.8.8 Physical channel mapping for E-DCH

The E-PUCH is defined in [7]. The bits input to the physical channel mapping are denoted by  $r_1, r_2, \dots, r_R$ , where  $R = N_{e,data,j}$  and is the number of physical channel data bits to be transmitted in the current TTI. These bits are mapped to the physical channel bits,  $\{w_{t,k} : t = 1, 2, \dots, T; \text{ and } k = 1, 2, \dots, U_t\}$ , where  $t$  is the timeslot index,  $T$  is the number of timeslots in the allocation message,  $k$  is the physical channel bit index and  $U_t$  is the number of bits in the E-PUCH physical channel in timeslot  $t$ . The timeslot index,  $t$ , increases with increasing timeslot number and the physical channel bit index,  $k$ , increases with increasing physical channel bit position in time.

The bits  $r_1, r_2, \dots, r_R$  shall be mapped to the physical channel bits  $w_{t,k}$  according to the following rule :

$$w_{1,k} = r_k \quad \text{for } k = 1, 2, \dots, U_1$$

$$w_{2,k} = r_{k+U_1} \quad \text{for } k = 1, 2, \dots, U_2$$

...

$$w_{T,k} = r_{k + \sum_{i=1}^{T-1} U_i} \quad \text{for } k = 1, 2, \dots, U_T$$

## 4.9 Coding for E-UCCH

### 4.9.1 Coding for E-UCCH for the 3.84Mcps and 7.68Mcps TDD options

E-UCCH is coded in two parts, E-UCCH part 1 and E-UCCH part 2. Both parts of the E-UCCH are transmitted on the E-PUCH (see [7]).

The following information is transmitted by means of E-UCCH part 1:

- E-TFCI information:  $x_{tfc1,1}, x_{tfc1,2}, \dots, x_{tfc1,7}$

The following information is transmitted by means of E-UCCH part 2:

- Retransmission sequence number (RSN):  $x_{rsn,1}, x_{rsn,2}$
- HARQ process ID (HARQ\_ID):  $x_{harq,1}, x_{harq,2}$

#### 4.9.1.1 Overview

Figure 24 below illustrates the overall coding chain for E-UCCH.

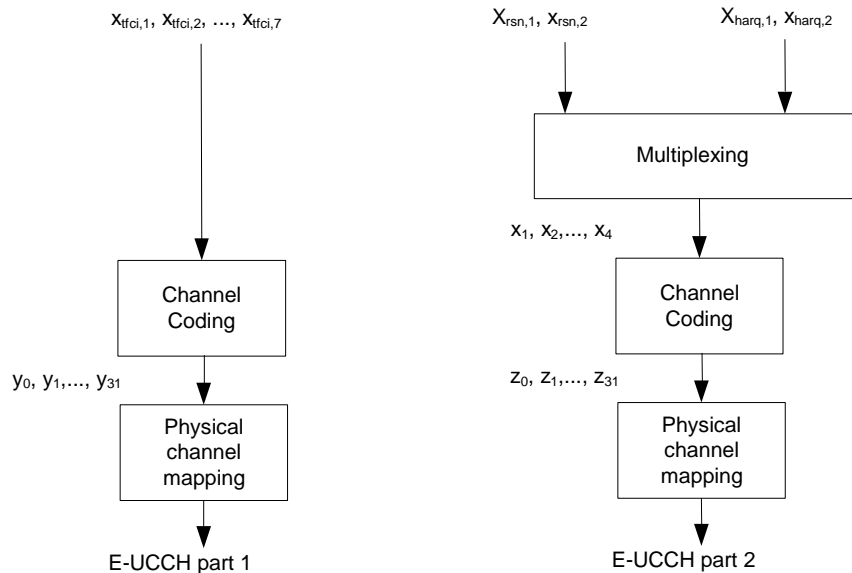


Figure 24: Coding chain for E-UCCH parts 1 and 2

#### 4.9.1.2 E-UCCH part 1

##### 4.9.1.2.1 Information field mapping of E-TFCI

The E-TFCI is mapped such that  $x_{tfc,i,1}$  corresponds to the MSB.

##### 4.9.1.2.2 Channel coding for E-UCCH part 1

The E-TFCI bits  $x_{tfc,i,1}, x_{tfc,i,2}, \dots, x_{tfc,i,7}$  are zero-padded and coded to produce the bits  $y_0, y_1, \dots, y_{31}$  using a (32,10) sub-code of the second order Reed-Muller code as defined in subclause 4.3.1.1. Bits  $x_{tfc,i,1}, x_{tfc,i,2}, \dots, x_{tfc,i,7}$  correspond to bits  $a_6, a_5, \dots, a_0$  of subclause 4.3.1.1, and bits  $a_9, a_8, a_7$  of subclause 4.3.1.1 are set to zero.

##### 4.9.1.2.3 Physical channel mapping for E-UCCH part 1

E-UCCH part 1 is described in [7]. The sequence of bits  $y_0, y_1, \dots, y_{31}$  output from the E-UCCH part 1 channel coding is mapped to the E-UCCH part 1 indicator field of each E-PUCH of the E-DCH TTI configured to carry E-UCCH. The bits  $y_i \{i=0,1,\dots,31\}$  are mapped in an identical manner to that described for TFCI in subclause 4.3.1.3 such that they are transmitted over the air in ascending order with respect to  $i$ .

#### 4.9.1.3 E-UCCH part 2

##### 4.9.1.3.1 Information field mapping of retransmission sequence number

To indicate the redundancy version (RV) of each HARQ transmission and to assist the Node B soft buffer management a two bit retransmission sequence number (RSN) is signalled from the UE to the Node B. The Node B can avoid soft buffer corruption by flushing the soft buffer associated to one HARQ process in case the last received RSN for that HARQ process is incompatible with the current one.

The RSN is set by higher layers as described in [15]. For a given HARQ process, once the maximum RSN value of 3 is reached, the RSN alternates between the values of 2 and 3 for any further retransmissions. The RSN thus follows the pattern 0,1,2,3,2,3,2,3,2,3,... The RSN is transmitted by the physical layer in E-UCCH part 2.

The bits ( $x_{rsn,1}, x_{rsn,2}$ ) of the RSN field of E-UCCH part 2 are mapped such that  $x_{rsn,1}$  corresponds to the MSB of the RSN and  $x_{rsn,2}$  corresponds to the LSB of the RSN.

The applied E-DCH RV index specifying the used RV ( $s$  and  $r$  parameter) and in the case of 16-QAM, also the used constellation rearrangement parameter ( $b$ ) both depend only on the values of RSN, and on  $N_{sys} / N_{e,data,j}$  as shown in table 25 below.

**Table 25: Relation between RSN value and E-DCH RV Index**

RSN Value	$N_{sys} / N_{e,data,j} < 1/2$	$1/2 \leq N_{sys} / N_{e,data,j}$
	E-DCH RV Index	E-DCH RV Index
0	0	0
1	2	3
2	0	2
3	2	1

The UE shall use either:

- an RV index as indicated in Table 25 and according to the value of RSN
- or, if signalled by higher layers only E-DCH RV index 0 independently of the value of RSN.

##### 4.9.1.3.2 Information field mapping of HARQ process ID

The HARQ process ID bits ( $x_{harq,1}, x_{harq,2}$ ) transmitted on E-UCCH part 2 correspond to the two LSBs of the 3-bit HARQ ID indicated by higher layers (the MSB of the higher layer HARQ ID is not transmitted). Thus, with the higher layer HARQ ID represented as  $h_1, h_2, h_3$  (with  $h_1$  as the MSB), bits  $x_{harq,1}, x_{harq,2}$  correspond to bits  $h_2, h_3$ .

#### 4.9.1.3.3 Multiplexing of E-UCCH part 2 information

The retransmission sequence number information ( $x_{rsn,1}, x_{rsn,2}$ ) and the HARQ process ID information ( $x_{harq,1}, x_{harq,2}$ ) are multiplexed together to give a sequence of bits  $x_1, x_2, \dots, x_4$  where:

$$x_k = x_{rsn,k} \quad k=1,2$$

$$x_k = x_{harq,k-2} \quad k=3,4$$

#### 4.9.1.3.4 Channel coding for E-UCCH part 2

The bits  $x_1, x_2, \dots, x_4$  are zero-padded and coded to produce the bits  $z_0, z_1, \dots, z_{31}$  using a (32,10) sub-code of the second order Reed-Muller code as defined in subclause 4.3.1.1. Bits  $x_1, x_2, \dots, x_4$  correspond to bits  $a_3, a_2, \dots, a_0$  of subclause 4.3.1.1, and bits  $a_9, a_8, \dots, a_4$  of subclause 4.3.1.1 are set to zero.

#### 4.9.1.3.5 Physical channel mapping for E-UCCH part 2

E-UCCH part 2 is described in [7]. The sequence of bits  $z_0, z_1, \dots, z_{31}$  output from the E-UCCH part 2 channel coding is mapped to the E-UCCH part 2 indicator field of each E-PUCH of the E-DCH TTI configured to carry E-UCCH. The bits  $z_i \{i=0,1,\dots,31\}$  are mapped such that they are transmitted over the air in ascending order with respect to  $i$ .

### 4.9.2 Coding for E-UCCH for the 1.28Mcps TDD option

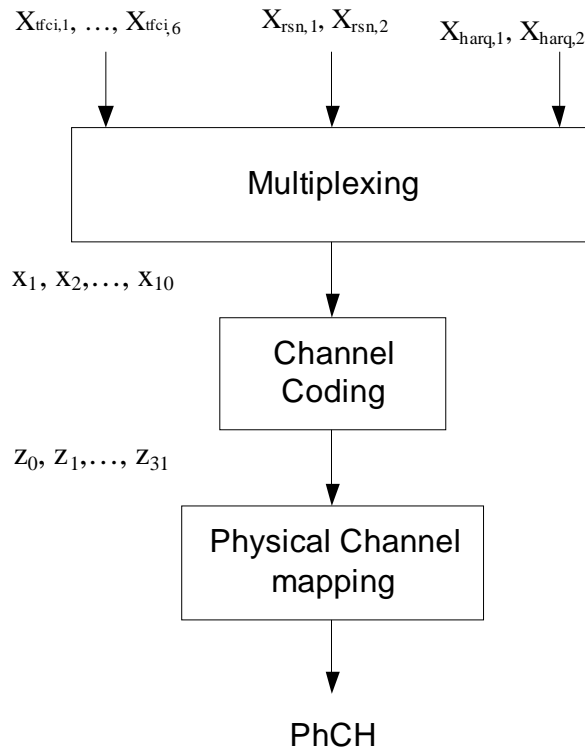
The E-UCCH is used to convey the following information:

- The modulation type of the selected E-TFC – 0 bits (see note 1)
- The transport block size of the selected E-TFC – 6 bits
- The retransmission sequence number (RSN) – 2 bits
- The HARQ process ID – 2 bits

*Note 1: The occupied modulation type is not explicitly signaled, which is inferred from the transport block size.*

The E-UCCH is transmitted on the E-PUCH and is coded using a (32, 10) sub code of the second order Reed Muller code as defined in subclause 4.3.1.1

Figure 24A below illustrates the overall coding chain for E-UCCH.



**Figure 24A: Coding chain for E-UCCH**

#### 4.9.2.1 E-UCCH information field mapping

##### 4.9.2.1.1 Information field mapping of E-TFCI

The E-TFCI is mapped such that  $x_{tfc1,1}$  corresponds to the MSB.

##### 4.9.2.1.2 RSN information mapping

To indicate the redundancy version (RV) of each HARQ transmission and to assist the Node B soft buffer management a two bit retransmission sequence number (RSN) is signalled from the UE to the Node B. The Node B can avoid soft buffer corruption by flushing the soft buffer associated to one HARQ process in case the last received RSN for that HARQ process is incompatible with the current one.

The RSN is set by higher layers as described in [15]. For a given HARQ process, once the maximum RSN value of 3 is reached, the RSN alternates between the values of 2 and 3 for any further retransmissions. The RSN thus follows the pattern 0,1,2,3,2,3,2,3,2,3,...

The bits ( $x_{rsn,1}$ ,  $x_{rsn,2}$ ) of the RSN field are mapped such that  $x_{rsn,1}$  corresponds to the MSB of the RSN and  $x_{rsn,2}$  corresponds to the LSB of the RSN.

The used RV is implicitly linked to the transmitted RSN, as such the Node-B is always able to determine the correct RV if the RSN information is correctly obtained.

The constellation rearrangement parameter linkage with RSN is shown in Table 25A below

**Table 25A: mapping between RSN and b parameters for CoRe**

RSN	$N_{sys} / N_{e,data,j} < 1/2$	$1/2 \leq N_{sys} / N_{e,data,j}$
	b	b
0	0	0
1	2	3
2	3	0
3	1	1

In addition to being associated with the value of RSN, the redundancy version (RV) of the E-DCH transmission is also associated with the coding rate of the E-DCH transmission according to Table 25B and Table 25C below.

**Table 25B: Relation between RSN and E-DCH RV index for QPSK**

RSN	$N_{sys} / N_{e,data,j} < 1/2$	$1/2 \leq N_{sys} / N_{e,data,j}$
	E-DCH RV Index	E-DCH RV Index
0	0	0
1	2	3
2	0	2
3	2	1

**Table 25C: Relation between RSN and E-DCH RV index for 16QAM**

RSN	$N_{sys} / N_{e,data,j} < 1/2$	$1/2 \leq N_{sys} / N_{e,data,j}$
	E-DCH RV Index	E-DCH RV Index
0	0	0
1	0	3
2	2	1
3	2	2

#### 4.9.2.1.3 HARQ information mapping

The bits  $(x_{harq,1}, x_{harq,2})$  of the HARQ field are mapped such that  $x_{harq,1}$  corresponds to the MSB of the HARQ process ID and  $x_{harq,2}$  corresponds to the LSB of the HARQ process ID.

#### 4.9.2.2 Multiplexing for E-UCCH

The transport block size information  $(x_{tbs,1}, \dots, x_{tbs,6})$ , retransmission sequence number information  $(x_{rsn,1}, x_{rsn,2})$  and the HARQ process ID information  $(x_{harq,1}, x_{harq,2})$  are multiplexed together to give a sequence of bits  $x_1, x_2, \dots, x_{10}$  where:

$$x_k = x_{tbs,k} \quad k=1,2,3,4,5,6$$

$$x_k = x_{rsn,k-6} \quad k=7,8$$

$$x_k = x_{harq,k-8} \quad k=9,10$$

#### 4.9.2.3 Coding for E-UCCH

The bits  $x_1, x_2, \dots, x_{10}$  are coded to produce the bits  $z_0, z_1, \dots, z_{31}$  using a (32,10) sub-code of the second order Reed-Muller code as defined in subclause 4.3.1.1.

#### 4.9.2.4 Physical channel mapping for E-UCCH

The E-UCCH is described in [7]. The sequence of bits  $z_0, z_1, \dots, z_{31}$  output from the E-UCCH channel coding is mapped with E-DCH to E-PUCH of the E-DCH TTI configured to carry E-UCCH.



## 4.10 Coding for E-AGCH

The E-AGCH carries the following fields multiplexed into  $w$  bits  $x_{ag,1}, x_{ag,2}, \dots, x_{ag,w}$ .  $w$  is within the range 14 to 28 bits (for the 3.84Mcps option) and 15 to 29 bits (for the 7.68Mcps option) and 23 to 26 bits (for the 1.28Mcps option)

- Absolute grant (power) value ( $x_{pg,1}, x_{pg,2}, \dots, x_{pg,5}$ ) {5 bits},
- Code resource related information ( $x_{c,1}, x_{c,2}, \dots, x_{c,N_c}$ ), { $N_c=5$  bits for the 1.28Mcps and 3.84Mcps options,  $N_c=6$  bits for the 7.68Mcps option}
- Timeslot resource related information ( $x_{t,1}, x_{t,2}, \dots, x_{t,n_{TRRI}}$ ), { $n_{TRRI}$  bits} [ $n_{TRRI}=5$  for 1.28Mcps, and is configured by higher layers for 3.84Mcps and 7.68Mcps options]
- E-AGCH Cyclic Sequence Number (ECSN) ( $x_{e,1}, x_{e,2}, x_{e,3}$ ) (3 bits)
- Resource duration indicator (3 bits if present) ( $x_{r,1}, x_{r,2}, x_{r,3}$ ) – [the presence of this field is configured by higher layers]
- E-HICH Indicator ( $x_{EI,1}, x_{EI,2}$ ) (2 bits) (for 1.28Mcps TDD only)
- E-UCCH Number Indicator ( $x_{ENI,1}, x_{ENI,2}, x_{ENI,3}$ ) (3 bits) (for 1.28Mcps TDD only)

Figure 25 illustrates the overall coding chain for the E-AGCH.

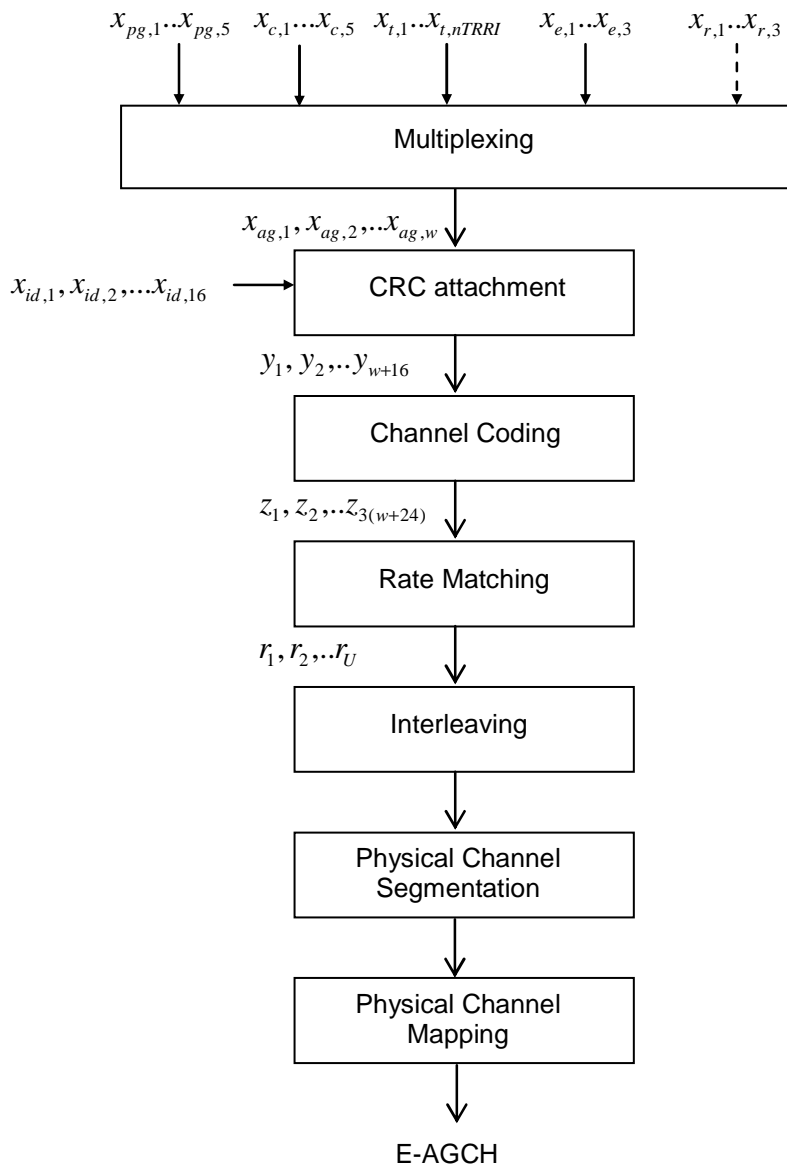


Figure 25 – TrCH processing of E-AGCH

### 4.10.1 Information Field Mapping

#### 4.10.1.1 Mapping of the Absolute Grant (Power) Value

The absolute grant (power) value ( $x_{pg,1}, x_{pg,2}, \dots, x_{pg,5}$ ) is represented by 5 bits and corresponds to a dB value as specified in table 26 below. The values are mapped such that  $x_{pg,1}$  corresponds to the MSB of the index.



		$C_9^{(16)}$ [23]	$C_{17}^{(32)}$ [47]
	$C_5^{(8)}$ [11]		$C_{18}^{(32)}$ [48]
		$C_{10}^{(16)}$ [24]	$C_{19}^{(32)}$ [49]
$C_3^{(4)}$ [5]		$C_{11}^{(16)}$ [25]	$C_{20}^{(32)}$ [50]
	$C_6^{(8)}$ [12]	$C_{12}^{(16)}$ [26]	$C_{21}^{(32)}$ [51]
$C_2^{(2)}$ [2]		$C_{13}^{(16)}$ [27]	$C_{22}^{(32)}$ [52]
	$C_7^{(8)}$ [13]	$C_{14}^{(16)}$ [28]	$C_{23}^{(32)}$ [53]
		$C_{15}^{(16)}$ [29]	$C_{24}^{(32)}$ [54]
$C_4^{(4)}$ [6]	$C_8^{(8)}$ [14]	$C_{16}^{(16)}$ [30]	$C_{25}^{(32)}$ [55]
			$C_{26}^{(32)}$ [56]
			$C_{27}^{(32)}$ [57]
			$C_{28}^{(32)}$ [58]
			$C_{29}^{(32)}$ [59]
			$C_{30}^{(32)}$ [60]
			$C_{31}^{(32)}$ [61]
			$C_{32}^{(32)}$ [62]

#### 4.10.1.3 Mapping of the Timeslot Resource Related Information

For 3.84Mcps and 7.68Mcps options, the timeslot resource related information ( $x_{t,1}, x_{t,2}, \dots, x_{t,n_{TRRI}}$ ) is a bitmap of length  $n_{TRRI}$  indicating which of the timeslots configured for E-DCH use by higher layers have been allocated. The length of the field ( $n_{TRRI}$ ) is configured by higher layers up to a maximum of 12 bits. For 1.28Mcps option, the timeslot resource related information ( $x_{t,1}, x_{t,2}, \dots, x_{t,n_{TRRI}}$ ) is also a bitmap of length  $n_{TRRI}$ , which equals 5 indicating the allocation for E-DCH resources from TS1 to TS5. The bitmap is arranged such that the corresponding timeslots are in ascending numerical order, with  $x_{t,1}$  corresponding to LSB and lowest-numbered timeslot configured for E-DCH use.

#### 4.10.1.4 Mapping of the E-AGCH Cyclic Sequence Number (ECSN)

The E-AGCH cyclic sequence number is mapped such that  $x_{e,1}$  corresponds to the MSB and  $x_{e,3}$  to the LSB.

#### 4.10.1.5 Mapping of the Resource Duration Indicator

The resource duration indicator ( $x_{r,1}, x_{r,2}, x_{r,3}$ ) is mapped such that  $x_{r,1}$  corresponds to the MSB of the resource duration index described in [15].

#### 4.10.1.6 Mapping of the E-HICH Indicator (1.28Mcps option only)

The E-HICH indicator consists of 2 bits used to indicate the UE which E-HICH will be used to convey the acknowledgement indicator in the following schedule period for 1.28Mcps TDD only. The bits ( $x_{EI,1}, x_{EI,2}$ ) are mapped such that  $x_{EI,1}$  corresponds to the MSB and  $x_{EI,2}$  to the LSB.

#### 4.10.1.7 Mapping of the E-UCCH Number Indicator (1.28Mcps option only)

The E-UCCH number indicator is composed of 3bits which is used to indicate the detailed number of E-UCCH for 1.28Mcps TDD only. The bits ( $x_{ENI,1}, x_{ENI,2}, x_{ENI,3}$ ) are mapped such that  $x_{ENI,1}$  corresponds to the MSB and  $x_{ENI,3}$  to the LSB.

### 4.10.2 Field Multiplexing

The absolute grant (power) value, code resource related information, timeslot resource related information, resource duration indicator (if present), ECSN, E-HICH indicator (for 1.28Mcps TDD only) and E-UCCH number indicator (for 1.28Mcps TDD only) are multiplexed together to give a sequence of bits  $x_{ag,1}, x_{ag,2}, \dots, x_{ag,w}$  where:

For the 1.28Mcps option:

$$\begin{aligned} x_{ag,k} &= x_{pg,k} & k=1, 2, \dots, 5 \\ x_{ag,k} &= x_{c,k-5} & k=6, 7, \dots, 10 \\ x_{ag,k} &= x_{t,k-10} & k=11, \dots, 15 \end{aligned}$$

$$X_{ag,k} = X_{e,k-15} \quad k=16,17,18$$

if resource duration indicator (RDI) present:

$$X_{ag,k} = X_{r,k-18} \quad k=19,20,21$$

$$X_{ag,k} = X_{EI,k-21} \quad k=22,23$$

$$X_{ag,k} = X_{ENI,k-23} \quad k=24,25,26$$

or if resource duration indicator (RDI) does not present:

$$X_{ag,k} = X_{EI,k-18} \quad k=19,20$$

$$X_{ag,k} = X_{ENI,k-20} \quad k=21,22,23$$

For the 3.84Mcps option:

$$X_{ag,k} = X_{pg,k} \quad k=1, 2, \dots, 5$$

$$X_{ag,k} = X_{c,k-5} \quad k=6, 7, \dots, 10$$

$$X_{ag,k} = X_{t,k-10} \quad k=11, \dots, 10+n_{TRRI}$$

$$X_{ag,k} = X_{e,k-10-n_{TRRI}} \quad k=11+n_{TRRI}, \dots, 13+n_{TRRI}$$

$$X_{ag,k} = X_{r,k-13-n_{TRRI}} \quad k=14+n_{TRRI}, \dots, 16+n_{TRRI} \text{ \{if resource duration indicator present\}}$$

and for the 7.68Mcps option:

$$X_{ag,k} = X_{pg,k} \quad k=1, 2, \dots, 5$$

$$X_{ag,k} = X_{c,k-5} \quad k=6, 7, \dots, 11$$

$$X_{ag,k} = X_{t,k-11} \quad k=12, \dots, 11+n_{TRRI}$$

$$X_{ag,k} = X_{e,k-11-n_{TRRI}} \quad k=12+n_{TRRI}, \dots, 14+n_{TRRI}$$

$$X_{ag,k} = X_{r,k-14-n_{TRRI}} \quad k=15+n_{TRRI}, \dots, 17+n_{TRRI} \text{ \{if resource duration indicator present\}}$$

### 4.10.3 CRC attachment

The E-RNTI ( $x_{id,1}, x_{id,2}, \dots, x_{id,16}$ ) is the E-DCH Radio Network Identifier defined in [12]. It is mapped such that  $x_{id,1}$  corresponds to the MSB.

From the sequence of bits  $x_{ag,1}, x_{ag,2}, \dots, x_{ag,w}$  a 16 bit CRC is calculated according to section 4.2.1.1. This gives the sequence of bits  $c_1, c_2, \dots, c_{16}$  where:

$$c_k = P_{im(17-k)} \quad k=1,2,\dots,16$$

This sequence of bits is then masked with  $x_{id,1}, x_{id,2}, \dots, x_{id,16}$  and appended to the sequence of bits  $x_{ag,1}, x_{ag,2}, \dots, x_{ag,w}$  to form the sequence of bits  $y_1, y_2, \dots, y_{w+16}$  where

$$y_i = x_{ag,i} \quad i=1,2, \dots, w$$

$$y_i = (c_{i-w} + x_{id,i-w}) \bmod 2 \quad i=w+1, \dots, w+16$$

### 4.10.4 Channel Coding

1/3 rate convolutional channel coding is applied to the sequence  $y_1, y_2, \dots, y_{w+16}$  in accordance with section 4.2.3.1, resulting in the sequence of bits  $z_1, z_2, \dots, z_{3(w+24)}$ .

## 4.10.5 Rate Matching

Rate matching is applied to the input sequence  $z_1, z_2, \dots, z_{3(w+24)}$  to obtain the output sequence  $r_1, r_2, \dots, r_U$ , where  $U = 242$  for burst type 1 and  $U = 274$  for burst type 2.

## 4.10.6 Interleaving

Interleaving is performed as per section 4.2.11.1 (frame-related 2<sup>nd</sup> interleaving).

## 4.10.7 Physical Channel Segmentation

Physical channel segmentation is performed as per section 4.2.10. For 1.28Mcps TDD, the E-AGCH consists of two physical channels E-AGCH1 and E-AGCH2, whereas for 3.84Mcps and 7.68Mcps TDD the E-AGCH only uses one physical channel, see [7]. Note that physical channel segmentation is transparent when only one physical channel exists

## 4.10.8 Physical Channel Mapping

Physical channel mapping is performed as per section 4.2.12.

## 4.11 Coding for E-HICH ACK/NACK

### 4.11.1 Coding for E-HICH ACK/NACK for the 3.84Mcps and 7.68Mcps options

#### 4.11.1.1 Overview

The ACK/NACK is transmitted on the E-HICH as described in [7].

The value of a binary HARQ acknowledgement indicator for user  $h$  is denoted ' $a_h$ ' and may assume the value 0 or 1. The value of the indicator is mapped as shown in table 28.

**Table 28 – Mapping of HARQ acknowledgement indicator**

Command	HARQ acknowledgement indicator value ( $a_h$ )
NACK	0
ACK	1

A HARQ acknowledgement indicator is mapped to one of 240 signature sequences of length 240 bits and represented by the bit sequence  $b_{h,0}, b_{h,1}, \dots, b_{h,239}$  for the  $h^{\text{th}}$  acknowledgement indicator. The signature sequence number ' $r$ ' is selected as described in [7].

The signature sequence  $b_{h,0}, b_{h,1}, \dots, b_{h,239}$  is constructed via coding of  $a_h$  followed by bit scrambling. Spare bits are inserted during the physical channel mapping stage to produce the output sequence  $d_{h,0}, d_{h,1}, \dots, d_{h,U}$ .

#### 4.11.1.2 Coding of the HARQ acknowledgement indicator

Bit  $a_h$  is used to form the sequence  $s_{2,v}$  ( $v=0,1,\dots,239$ ) via a two-stage serialised binary spreading process as shown in figure 26.

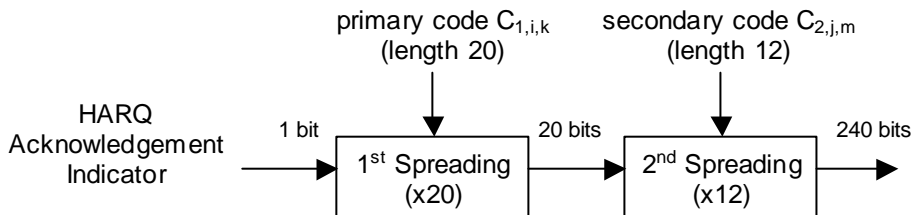


Figure 26

The output of the first spreading stage is  $s_{1,k} = a_h \oplus C_{1,i,k}$ , where  $k=0, 1, 2, \dots, 19$ .

The output of the second spreading stage is  $s_{2,v} = s_{1,w} \oplus C_{2,j,m}$ , where  $v=0, 1, \dots, 239$  and where  $w = \left\lfloor \frac{v}{12} \right\rfloor$  and,  $m = v \bmod 12$ .

The binary sequences selected for the first ( $C_{1,i,k}$ ) and second ( $C_{2,j,m}$ ) spreading operations are derived as a function of the HARQ acknowledgement sequence number  $r$  (see [7]) such that:

$$i = \left\lfloor \frac{r}{12} \right\rfloor$$

$$j = r \bmod 12$$

The first orthogonal sequence set ( $C_{1,i,k}$ ) is given by table 29 and the second orthogonal sequence set ( $C_{2,j,m}$ ) is given by table 30.

Table 29 – Primary code sequences for HARQ acknowledgement indicator

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$C_{1,0,k}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$C_{1,1,k}$	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1
$C_{1,2,k}$	1	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0
$C_{1,3,k}$	1	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0
$C_{1,4,k}$	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1
$C_{1,5,k}$	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1
$C_{1,6,k}$	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0
$C_{1,7,k}$	1	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0
$C_{1,8,k}$	1	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0
$C_{1,9,k}$	1	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0
$C_{1,10,k}$	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1
$C_{1,11,k}$	1	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0
$C_{1,12,k}$	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1
$C_{1,13,k}$	1	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0
$C_{1,14,k}$	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1
$C_{1,15,k}$	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1
$C_{1,16,k}$	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1
$C_{1,17,k}$	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1
$C_{1,18,k}$	1	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0
$C_{1,19,k}$	1	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0

**Table 30 – Secondary code sequences for HARQ acknowledgement indicator**

m	0	1	2	3	4	5	6	7	8	9	10	11
$C_{2,0,m}$	1	1	1	1	1	1	1	1	1	1	1	1
$C_{2,1,m}$	1	0	1	0	1	1	1	0	0	0	1	0
$C_{2,2,m}$	0	1	1	0	1	0	0	0	1	1	1	0
$C_{2,3,m}$	1	1	0	0	1	0	1	1	1	0	0	0
$C_{2,4,m}$	1	0	1	0	0	1	0	1	1	1	0	0
$C_{2,5,m}$	0	1	1	0	1	1	0	1	0	0	0	1
$C_{2,6,m}$	0	1	1	1	0	1	1	0	1	0	0	0
$C_{2,7,m}$	0	0	1	1	1	0	1	1	0	1	0	0
$C_{2,8,m}$	1	1	1	0	0	0	1	0	0	1	0	1
$C_{2,9,m}$	0	0	0	0	1	1	1	0	1	1	0	1
$C_{2,10,m}$	0	1	0	0	0	1	1	1	0	1	1	0
$C_{2,11,m}$	1	1	0	1	1	1	0	0	0	1	0	0

#### 4.11.1.3 Bit scrambling of the E-HICH

The bit sequence  $b_{h,0}, b_{h,1}, \dots, b_{h,239}$  is formed by applying bit scrambling (as defined in subclause 4.2.9) to the sequence  $S_{2,v}$ .

#### 4.11.1.4 Physical channel mapping of the E-HICH

The bit sequence  $b_{h,0}, b_{h,1}, \dots, b_{h,239}$  is segmented into two halves,  $b_{h,0}, \dots, b_{h,119}$ , and  $b_{h,120}, \dots, b_{h,239}$ . A sequence of  $U$  spare bits  $z_u$  ( $u=0 \dots U-1$ ) are inserted between the first and second half of the sequence to form:

$$d_h = \{b_{h,0}, b_{h,1}, \dots, b_{h,119}, z_0, z_1, \dots, z_{U-1}, b_{h,120}, b_{h,121}, \dots, b_{h,239}\}$$

$U$  is equal to 4 or 36 dependant on the burst type (see [7]). The spare bit sequence  $z_u$  is not defined.

### 4.11.2 Coding for E-HICH for the 1.28Mcps option only

#### 4.11.2.1 Overview

The scheduled and non-Scheduled transmissions on different E-HICHs are described in [7].

The value of a binary HARQ acknowledgement indicator for user  $h$  is denoted ' $a_h$ ' and may assume the value 0 or 1. The value of the indicator is mapped as same as that of 3.84Mcps shown in subclause 4.11.1.1.

Construction of the bit sequence for the  $h^{\text{th}}$  acknowledgement indicator is achieved via a spreading process using an orthogonal sequence which is the row of an orthogonal matrix of order 80. This orthogonal matrix ( $C_{80}$ ) is Kronecker tensor product of one Hadamard matrix of order 20 ( $C_{20}$ ) and another Hadamard matrix of order 4 ( $C_4$ ),

$$C_{80} = C_{20} \otimes C_4$$

$\otimes$  is Kronecker tensor product. (note: Kronecker product is not commutative, i.e.  $A \otimes B \neq B \otimes A$ )

These two Hadamard matrices are given by table 31 and table 32.

**Table 31: Hadamard matrix of order 4**

m	0	1	2	3
$C_{4,0,m}$	1	1	1	1
$C_{4,1,m}$	1	0	1	0



$$C_{4,2,m} \quad 1 \quad 1 \quad 0 \quad 0$$

$$C_{4,3,m} \quad 0 \quad 1 \quad 1 \quad 0$$

**Table 32: Hadamard matrix of order 20**

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
C <sub>20,0,k</sub>	1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	1	1	0
C <sub>20,1,k</sub>	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	1	0	1	1
C <sub>20,2,k</sub>	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	1
C <sub>20,3,k</sub>	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	1	1	0	1	0
C <sub>20,4,k</sub>	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1
C <sub>20,5,k</sub>	0	1	1	1	1	1	0	0	0	0	0	1	0	0	1	1	1	0	0	1
C <sub>20,6,k</sub>	1	0	1	1	1	0	1	0	0	0	1	0	1	0	0	1	1	1	0	0
C <sub>20,7,k</sub>	1	1	0	1	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0
C <sub>20,8,k</sub>	1	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1
C <sub>20,9,k</sub>	1	1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1
C <sub>20,10,k</sub>	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1
C <sub>20,11,k</sub>	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1
C <sub>20,12,k</sub>	1	0	0	0	1	1	0	1	0	1	0	0	1	0	0	1	1	0	1	1
C <sub>20,13,k</sub>	1	1	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	1
C <sub>20,14,k</sub>	0	1	1	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0
C <sub>20,15,k</sub>	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0
C <sub>20,16,k</sub>	1	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0
C <sub>20,17,k</sub>	0	1	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0
C <sub>20,18,k</sub>	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0
C <sub>20,19,k</sub>	1	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1

The binary orthogonal sequence (C<sub>80,r,n</sub>) used for spreading operation is selected from the r<sup>th</sup> row of the orthogonal matrix of order 80 (C<sub>80</sub>). A HARQ acknowledgement indicator is synchronously linked with the E-DCH TTI transmission to which it relates. There is thus a one-to-one association between an E-DCH TTI transmission and its respective HARQ acknowledgement indicator.

#### 4.11.2.2 Coding of the HARQ acknowledgement indicator and TPC/SS

For scheduled transmissions, E-HICHs carry HARQ acknowledgement indicators only.

A logical allocation resource tag ID 'r' (r=0,1,2,...,79) is calculated first for the E-DCH resource allocation associated with the HARQ acknowledgement indicator.

$$r = 16(t_0 - 1) + (q_0 - 1) \frac{16}{Q_0}$$

where:

$t_0$  is the first (lowest-numbered) allocated timeslot (1,2,...,5)

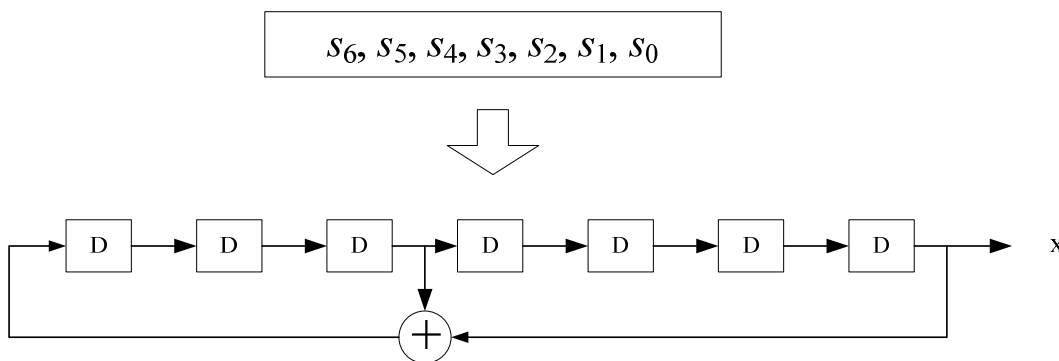
$q_0$  is the lowest-numbered channelisation code index allocated in timeslot  $t_0$  (1,2,...,  $Q_0$ )

$Q_0$  is the spreading factor of the lowest-numbered channelisation code index allocated in timeslot  $t_0$

The logical resource tag ID  $r$  is then mapped to a physical allocation resource tag ID  $r'$ ,

$$r' = P(r, SFN', MidambleCode),$$

where  $P$  is a permutation function depends on the logical signature index  $r$ , system sub-frame number  $SFN'$  and the cell specific midamble sequence (MidambleCode). A 7-tap linear feedback shift register (LFSR) is used to generate pseudo-random numbers which are then used to generate the pseudo-random permutation  $P$ . The generator polynomial of the 7-tap LFSR is  $g(x) = 1 + x^3 + x^7$ , as shown in Figure 27.



**Figure 27: Structure of PN register**

The pseudo-random permutation is generated according to the following procedures:

1. Initialization

- (a) Initialize  $M = 80$ , initialize  $P$  as  $P(m) = m, m = 0, 1, \dots, M-1$ ;
- (b) Initialize  $N = 7$ ;
- (c) Initialize PN register with seed  $s$ , where  $s = \text{LSB}(SFN', N) \square \text{LSB}(\text{MidambleCode}, N)$ ;
- (d) Initialize  $i = 0$ .

2. Repeat the following steps until  $i = M - 3$ .

- (a) Find the smallest  $p$  such that  $M - i - 1 < 2^p$ ;
- (b) Clock the PN register  $N$  times to obtain an  $N$ -bit pseudorandom number  $x$ . Set  $k = \text{LSB}(x, p)$ ;
- (c) If  $k > M - i - 1$ , set  $k = k - (M - i)$ ;
- (d) Swap the  $i$ -th and the  $(k+i)$ -th element of  $P$ , i.e.,  $\text{tmp} = P(i), P(i) = P(k+i), P(k+i) = \text{tmp}$ ;
- (e) Increment  $i$  by 1.

where 'LSB( $x, n$ )' means the  $n$ -LSBs of  $x$ , ' $\square$ ' means modulo 2 addition. The resulting  $P$  is the output permutation and the physical signature sequence index is given by  $r' = P(r)$ .

The output of the spreading stage is equal to  $b_{h,n} = a_h \Theta C_{80,r',n}$ , where  $n=0,1,\dots,79$ .

For Non-Scheduled transmissions, E-HICHs carry HARQ acknowledgement indicators and TPC/SS commands. The 80 orthogonal sequences are divided into 20 groups while each group includes 4 sequences of contiguous logical resource tag ID. The mapping between the logical resource tag ID and the physical tag ID is same as scheduled transmissions. Each non-scheduled user is assigned one group by higher layer to indicate the HARQ acknowledgement indicator and TPC/SS command. The first one of the four sequences is used for the acknowledgement indicator's spreading operation and one of the other three is used to indicate TPC/SS command implicitly. The mapping relations between them are described in [7].

The HARQ acknowledgement indicator is spread by the assigned orthogonal sequence ( $C_{80,s',n}$ ), where  $s'$  is the physical resource tag ID. The output of the spreading stage is equal to  $c_{h,n} = a_h \Theta C_{80,s',n}$ , where  $n=0,1,\dots,79$ . The sequence chosen to indicate TPC/SS command is denoted as ' $e_{h,n}$ ',  $e_{h,n} = p_h \Theta C_{80,t',n}$ .

#### 4.11.2.3 Bit scrambling and Physical channel mapping of the E-HICH

For scheduled transmission, the bit sequence  $b_{h,0}, b_{h,1}, \dots, b_{h,79}$  is segmented into two halves,  $b_{h,0}, \dots, b_{h,39}$ , and  $b_{h,40}, \dots, b_{h,79}$ . 8 spare bits  $z_u$  ( $u=0\dots7$ ) are inserted between the first and second half of the sequence to form:

$$d_h = \{b_{h,0}, b_{h,1}, \dots, b_{h,39}, z_0, z_1, \dots, z_7, b_{h,40}, b_{h,41}, \dots, b_{h,79}\}$$

The spare bit sequence  $z_u$  is not defined. For Non-Scheduled transmission, the corresponding output bit sequences are:

$$d_{h1} = \{c_{h,0}, c_{h,1}, \dots, c_{h,39}, z_0, z_1, \dots, z_7, c_{h,40}, c_{h,41}, \dots, c_{h,79}\}$$

$$d_{h2} = \{e_{h,0}, e_{h,1}, \dots, e_{h,39}, z_0, z_1, \dots, z_7, e_{h,40}, e_{h,41}, \dots, e_{h,79}\}$$

Then the corresponding bit sequence  $d_h$  or  $d_{h1}/d_{h2}$  is formed by applying bit scrambling (as defined in subclause 4.2.9) to the sequence  $s_{h,n}$  or  $s_{h1,n}/s_{h2,n}$ ,  $n=0,1,\dots,87$ .

## 4.12 Coding for E-RUCCH

For 1.28Mcps TDD, E-RUCCH supports both a 5ms and a 10ms TTI. For 3.84Mcps and 7.68Mcps TDD, E-RUCCH supports a TTI of 10ms only. The sequence of 39 information bits to be transmitted on one E-RUCCH as supplied by higher layers is defined as bits  $x_{e-rucch,1}, a_{e-rucch,2}, \dots, a_{e-rucch,39}$ .

The following processing steps (in sequential order) are defined for E-RUCCH:

- CRC attachment
- Channel coding
- Rate matching
- Bit scrambling
- Frame related  $2^{\text{nd}}$  interleaving
- Mapping to physical channels

### 4.12.1 CRC attachment for E-RUCCH

From the sequence of  $A_i=39$  bits ( $x_{e-rucch,1}, a_{e-rucch,2}, \dots, a_{e-rucch,39}$ ) a CRC of length  $L_i=16$  is appended according to Section 4.2.1.1. The bits output from CRC attachment are denoted  $b_{im,k}$ , where  $k=1,2,3, \dots, A_i+L_i$  and  $A_i=39$

For 3.84Mcps and 7.68Mcps TDD, each parity bit  $b_{im,k}$ , (where  $k=A_i+1, A_i+2, \dots, A_i+L_i$ ) is then XOR'd with 1 such that its polarity is reversed.

## 4.12.2 Channel coding for E-RUCCH

The bits output from CRC attachment are subject to 1/3 rate convolutional coding in accordance with the general method of subclause 4.2.3.

## 4.12.3 Rate matching for E-RUCCH

Rate matching for E-RUCCH shall be performed in accordance with the general method described in subclause 4.2.7.

## 4.12.4 Bit scrambling for E-RUCCH

Bit scrambling for E-RUCCH shall be performed in accordance with the general method described in subclause 4.2.9.

## 4.12.5 Interleaving for E-RUCCH

Frame-related 2<sup>nd</sup> interleaving for E-RUCCH shall be performed in accordance with the general method described in subclause 4.2.11.1.

## 4.12.6 Physical channel mapping for E-RUCCH

Physical channel mapping for the E-RUCCH shall be performed in accordance with the general method described in subclause 4.2.12.1.

## Annex A (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
14/01/00	RAN_05	RAN_05	-		Approved at TSG RAN #5 and placed under Change Control	-	3.0.0
14/01/00	RAN_06	RP-99694	001	3	Correction of rate matching parameters for repetition after 1st Interleaving in 25.222	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	002	1	Clarification of bit separation and collection	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	003	-	Changing the initial offset value for convolutional code rate matching	3.0.0	3.1.0
14/01/00	RAN_06	RP-99693	004	1	Editorial corrections to TS 25.222	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	007	-	Update of rate matching rule for TDD	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	009	1	Modified physical channel mapping scheme	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	013	-	Introduction of TFCI for S-CCPCH in TDD mode	3.0.0	3.1.0
14/01/00	RAN_06	RP-99694	015	-	TFCI coding and mapping in TDD	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-000068	017	-	Corrections to TS 25.222	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	018	-	Refinements of Physical Channel Mapping	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	019	1	TFCI coding specification in TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	021	-	Modification of Turbo code internal interleaver	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	023	-	Update of TS 25.222 - clarification of BTFD for TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	025	-	Change of TFCI basis for TDD	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	026	-	Padding Function for Turbo coding of small blocks	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	027	-	Editorial modification of shifting parameter calculation for turbo code puncturing	3.1.1	3.2.0
31/03/00	RAN_07	RP-000068	029	1	Editorial changes of channel coding section	3.1.1	3.2.0
26/06/00	RAN_08	RP-000272	030	-	Parity bit attachment to 0 size transport block	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	031	-	Correction of the mapping formula	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	034	-	Alignment of Multiplexing for TDD	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	036	2	Bit separation of the Turbo encoded data	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	038	2	Revision of code block segmentation description	3.2.0	3.3.0
26/06/00	RAN_08	RP-000272	039	-	Editorial corrections in channel coding section	3.2.0	3.3.0
23/09/00	RAN_09	RP-000345	040	1	Update of TS 25.222	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	041	1	Editorial corrections in Turbo code internal interleaver section	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	042	-	Paging Indicator Terminology	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	043	1	Bit separation and collection for rate matching	3.3.0	3.4.0
23/09/00	RAN_09	RP-000345	048	-	Puncturing Limit definition in WG1 specification	3.3.0	3.4.0
15/12/00	RAN_10	RP-000543	049	-	Clarification on the Ci formula	3.4.0	3.5.0
15/12/00	RAN_10	RP-000543	050	-	Correction on TFCI & TPC Transmission	3.4.0	3.5.0
15/12/00	RAN_10	RP-000543	053	1	Editorial corrections in TS 25.222	3.4.0	3.5.0
16/03/01	RAN_11	-	-	-	Approved as Release 4 specification (v4.0.0) at TSG RAN #11	3.5.0	4.0.0
16/03/01	RAN_11	RP-010063	051	1	Bit Scrambling for TDD	3.5.0	4.0.0
16/03/01	RAN_11	RP-010063	054	1	Corrections & Clarifications for TS25.222	3.5.0	4.0.0
16/03/01	RAN_11	RP-010071	055	1	Inclusion of 1.28Mcps TDD in TS 25.222	3.5.0	4.0.0
21/09/01	RAN_13	RP-010523	057	-	TFCI Terminology	4.0.0	4.1.0
21/09/01	RAN_13	RP-010529	058	-	5ms TTI for PRACH for 1.28 Mcps TDD	4.0.0	4.1.0
21/09/01	RAN_13	RP-010529	060	-	A correction on the meaning of FPACH in TS 25.222	4.0.0	4.1.0
14/12/01	RAN_14	RP-010747	059	-	Bit Scrambling for TDD	4.1.0	4.2.0
14/12/01	RAN_14	RP-010747	061	-	Corrections in clause 4.1 and 4.2 of TS 25.222	4.1.0	4.2.0
08/03/02	RAN_15	RP-020050	063	1	Correction to addition of padding zeros to PICH in TDD	4.2.0	4.3.0
08/03/02	RAN_15	RP-020050	065	3	Clarification of the requirement for the determination of the rate matching parameters and editorial corrections to 25.222	4.2.0	4.3.0
08/03/02	RAN_15	RP-020058	066	2	Inclusion of HSDPA in 25.222	4.2.0	5.0.0
07/06/02	RAN_16	RP-020311	077	-	Second Stage Interleaving and Physical Channel Mapping	5.0.0	5.1.0
07/06/02	RAN_16	RP-020311	076	1	Zero padding for TFCI (3.84Mcps TDD)	5.0.0	5.1.0
07/06/02	RAN_16	RP-020314	073	-	Correction to addition of padding zeros to PICH in 1.28 Mcps TDD	5.0.0	5.1.0
07/06/02	RAN_16	RP-020314	086	-	Zero padding for TFCI (1.28Mcps TDD)	5.0.0	5.1.0
07/06/02	RAN_16	RP-020317	078	2	Removal of inconsistencies and ambiguities in the HARQ description	5.0.0	5.1.0
07/06/02	RAN_16	RP-020317	079	4	Corrections to HS-DSCH coding	5.0.0	5.1.0
07/06/02	RAN_16	RP-020317	082	1	Corrections to HSDPA Multiplexing and Coding	5.0.0	5.1.0
07/06/02	RAN_16	RP-020317	084	-	Introduction of HS-SCCH Cyclic Sequence Counter for TDD	5.0.0	5.1.0
19/09/02	RAN_17	RP-020570	097	1	Clarification of the definition of layer 1 transport channel numbers	5.1.0	5.2.0
19/09/02	RAN_17	RP-020585	093	-	HS-DSCH Interleaving for TDD	5.1.0	5.2.0
19/09/02	RAN_17	RP-020585	091	1	HS-SCCH corrections for TDD	5.1.0	5.2.0
19/09/02	RAN_17	RP-020585	089	1	Clarification of TFRI bits for 3.84Mcps HSDPA TDD	5.1.0	5.2.0
15/10/02	-	-	-	-	Moving of misplaced sections 4.6.1.5 and 4.6.1.6	5.2.0	5.2.1

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
22/12/02	RAN_18	RP-020843	100	-	Corrections to TFCI encoding of very short TFCI lengths	5.2.1	5.3.0
22/12/02	RAN_18	RP-020843	102	-	Corrections to TFCI encoding of very short TFCI lengths	5.2.1	5.3.0
22/12/02	RAN_18	RP-020849	103	-	Correction of editorial Error	5.2.1	5.3.0
22/12/02	RAN_18	RP-020849	104	-	Miscellaneous Minor HSDPA Corrections	5.2.1	5.3.0
22/12/02	RAN_18	RP-020852	107	-	Editorial modification to the section numberings	5.2.1	5.3.0
25/03/03	RAN_19	RP-030139	108	1	HSDPA corrections	5.3.0	5.4.0
25/03/03	RAN_19	RP-030139	109	3	Miscellaneous Corrections	5.3.0	5.4.0
24/06/03	RAN_20	RP-030276	111	-	Corrections to field coding of CQI for HS-SICH (3.84Mcps TDD)	5.4.0	5.5.0
24/06/03	RAN_20	RP-030276	112	-	Correction to definition of number of bits available to HS-DSCH in one TTI.	5.4.0	5.5.0
06/01/04	RAN_22	RP-030646	118	1	Correction of subframe segmentation, physical channel mapping & rate matching for 1.28Mcps TDD	5.5.0	5.6.0
06/01/04	RAN_22	RP-030650	119	-	Clarification of the CRC attachment procedure for HS-SCCH (TDD)	5.5.0	5.6.0
06/01/04	RAN_22	RP-030660	120	-	HARQ process identifier mapping	5.5.0	5.6.0
06/01/04	RAN_22	RP-030712	121	-	Alignment of "Soft channel bits" terminology with 25.306	5.5.0	5.6.0
13/01/04	RAN_22	-	-	-	Created for M.1457 update	5.6.0	6.0.0
07/09/04	RAN_25	RP-040315	124	1	Correction of symbol Xi defined in sub-frame segmentation step	6.0.0	6.1.0
13/12/04	RAN_26	RP-040451	125	-	Introduction of MICH	6.1.0	6.2.0
20/03/06	RAN_31	RP-060078	0126	-	Introduction of the Physical Layer Common Control Channel (PLCCH)	6.2.0	7.0.0
20/03/06	RAN_31	RP-060079	0127	-	Introduction of 7.68Mcps TDD option	6.2.0	7.0.0
29/09/06	RAN_33	RP-060492	0129	-	Introduction of E-DCH for 3.84Mcps and 7.68Mcps TDD	7.0.0	7.1.0
29/09/06	RAN_33	RP-060491	0130	-	Correction to coding of PLCCH for 1.28Mcps TDD	7.0.0	7.1.0
13/03/07	RAN_35	RP-070118	0131	2	Introduction of E-DCH for 1.28Mcps TDD	7.1.0	7.2.0
30/05/07	RAN_36	RP-070385	0132	1	Support for MBSFN operation	7.2.0	7.3.0
30/05/07	RAN_36	RP-070383	0133	1	Modification on HARQ process ID signalling for 1.28Mcps TDD	7.2.0	7.3.0
30/05/07	RAN_36	RP-070385	0135	-	Support for 1.28Mcps TDD MBSFN operation	7.2.0	7.3.0
11/09/07	RAN_37	RP-070650	0136	-	Introduction of multi-frequency operation for 1.28Mcps TDD	7.3.0	7.4.0
11/09/07	RAN_37	RP-070647	0137	-	TFCI mapping for S-CCPCH and 16QAM for 1.28Mcps TDD MBSFN	7.3.0	7.4.0
11/09/07	RAN_37	RP-070640	0138	1	Improved E-HICH signature sequence allocation for LCR TDD	7.3.0	7.4.0

---

## History

<b>Document history</b>		
V7.0.0	March 2006	Publication
V7.1.0	September 2006	Publication
V7.2.0	March 2007	Publication
V7.3.0	May 2007	Publication
V7.4.0	October 2007	Publication