

Draft **ETSI EN 300 909** V8.4.0 (2000-05)

European Standard (Telecommunications series)

**Digital cellular telecommunications system (Phase 2+);
Channel coding
(GSM 05.03 version 8.4.0 Release 1999)**

GSM®

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS



Reference

REN/SMG-020503Q8R1

Keywords

Digital cellular telecommunications system,
Global System for Mobile communications (GSM)

ETSI

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

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

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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).

In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at <http://www.etsi.org/tb/status/>

If you find errors in the present document, send your comment to:

editor@etsi.fr

Copyright Notification

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

© European Telecommunications Standards Institute 2000.
All rights reserved.

Contents

| | |
|--|----|
| Intellectual Property Rights | 10 |
| Foreword | 10 |
| 1 Scope | 11 |
| 1.1 References | 11 |
| 1.2 Abbreviations | 12 |
| 2 General | 12 |
| 2.1 General organization | 12 |
| 2.2 Naming Convention | 16 |
| 3 Traffic Channels (TCH) | 18 |
| 3.1 Speech channel at full rate (TCH/FS and TCH/EFS) | 18 |
| 3.1.1 Preliminary channel coding for EFR only | 18 |
| 3.1.1.1 CRC calculation | 18 |
| 3.1.1.2 Repetition bits | 19 |
| 3.1.1.3 Correspondence between input and output of preliminary channel coding | 19 |
| 3.1.2 Channel coding for FR and EFR | 19 |
| 3.1.2.1 Parity and tailing for a speech frame | 19 |
| 3.1.2.2 Convolutional encoder | 20 |
| 3.1.3 Interleaving | 20 |
| 3.1.4 Mapping on a Burst | 20 |
| 3.2 Speech channel at half rate (TCH/HS) | 21 |
| 3.2.1 Parity and tailing for a speech frame | 21 |
| 3.2.2 Convolutional encoder | 21 |
| 3.2.3 Interleaving | 22 |
| 3.2.4 Mapping on a burst | 22 |
| 3.3 Data channel at full rate, 12.0 kbit/s radio interface rate (9.6 kbit/s services (TCH/F9.6)) | 23 |
| 3.3.1 Interface with user unit | 23 |
| 3.3.2 Block code | 23 |
| 3.3.3 Convolutional encoder | 23 |
| 3.3.4 Interleaving | 23 |
| 3.3.5 Mapping on a Burst | 24 |
| 3.4 Data channel at full rate, 6.0 kbit/s radio interface rate (4.8 kbit/s services (TCH/F4.8)) | 24 |
| 3.4.1 Interface with user unit | 24 |
| 3.4.2 Block code | 24 |
| 3.4.3 Convolutional encoder | 24 |
| 3.4.4 Interleaving | 24 |
| 3.4.5 Mapping on a Burst | 25 |
| 3.5 Data channel at half rate, 6.0 kbit/s radio interface rate (4.8 kbit/s services (TCH/H4.8)) | 25 |
| 3.5.1 Interface with user unit | 25 |
| 3.5.2 Block code | 25 |
| 3.5.3 Convolutional encoder | 25 |
| 3.5.4 Interleaving | 25 |
| 3.5.5 Mapping on a Burst | 25 |
| 3.6 Data channel at full rate, 3.6 kbit/s radio interface rate (2.4 kbit/s and less services (TCH/F2.4)) | 25 |
| 3.6.1 Interface with user unit | 25 |
| 3.6.2 Block code | 25 |
| 3.6.3 Convolutional encoder | 26 |
| 3.6.4 Interleaving | 26 |
| 3.6.5 Mapping on a Burst | 26 |
| 3.7 Data channel at half rate, 3.6 kbit/s radio interface rate (2.4 kbit/s and less services (TCH/H2.4)) | 26 |
| 3.7.1 Interface with user unit | 26 |
| 3.7.2 Block code | 26 |
| 3.7.3 Convolutional encoder | 27 |
| 3.7.4 Interleaving | 27 |

| | | |
|----------|---|----|
| 3.7.5 | Mapping on a Burst | 27 |
| 3.8 | Data channel at full rate, 14.5 kbit/s radio interface rate (14.4 kbit/s services (TCH/F14.4))..... | 27 |
| 3.8.1 | Interface with user unit | 27 |
| 3.8.2 | Block code | 27 |
| 3.8.3 | Convolutional encoder..... | 27 |
| 3.8.4 | Interleaving | 27 |
| 3.8.5 | Mapping on a Burst | 28 |
| 3.9 | Adaptive multi rate speech channel at full rate (TCH/AFS)..... | 28 |
| 3.9.1 | SID_UPDATE | 28 |
| 3.9.1.1 | Coding of in-band data..... | 28 |
| 3.9.1.2 | Parity and convolutional encoding for the comfort noise parameters..... | 29 |
| 3.9.1.3 | Identification marker | 30 |
| 3.9.1.4 | Interleaving | 30 |
| 3.9.1.5 | Mapping on a Burst..... | 30 |
| 3.9.2 | SID_FIRST | 30 |
| 3.9.2.1 | Coding of in-band data..... | 30 |
| 3.9.2.2 | Identification marker | 30 |
| 3.9.2.3 | Interleaving | 30 |
| 3.9.2.4 | Mapping on a Burst..... | 30 |
| 3.9.3 | ONSET | 31 |
| 3.9.3.1 | Coding of in-band data..... | 31 |
| 3.9.3.2 | Interleaving | 31 |
| 3.9.3.3 | Mapping on a Burst..... | 31 |
| 3.9.4 | SPEECH | 31 |
| 3.9.4.1 | Coding of the in-band data | 32 |
| 3.9.4.2 | Ordering according to subjective importance..... | 32 |
| 3.9.4.3 | Parity for speech frames..... | 32 |
| 3.9.4.4 | Convolutional encoder | 34 |
| 3.9.4.5 | Interleaving | 41 |
| 3.9.4.6 | Mapping on a Burst..... | 41 |
| 3.9.5 | RATSCCH..... | 41 |
| 3.9.5.1 | Coding of in-band data..... | 41 |
| 3.9.5.2 | Parity and convolutional encoding for the RATSCCH message | 41 |
| 3.9.5.3 | Identification marker | 42 |
| 3.9.5.4 | Interleaving | 42 |
| 3.9.5.5 | Mapping on a Burst..... | 42 |
| 3.10 | Adaptive multi rate speech channel at half rate (TCH/AHS)..... | 42 |
| 3.10.1 | SID_UPDATE | 43 |
| 3.10.1.1 | Coding of in-band data..... | 43 |
| 3.10.1.2 | Parity and convolutional encoding for the comfort noise parameters..... | 43 |
| 3.10.1.3 | Identification marker | 44 |
| 3.10.1.4 | Interleaving | 44 |
| 3.10.1.5 | Mapping on a Burst..... | 44 |
| 3.10.2 | SID_UPDATE_INH..... | 45 |
| 3.10.2.1 | Coding of in-band data..... | 45 |
| 3.10.2.2 | Identification marker | 45 |
| 3.10.2.3 | Interleaving | 45 |
| 3.10.2.4 | Mapping on a Burst..... | 45 |
| 3.10.3 | SID_FIRST_P1..... | 46 |
| 3.10.3.1 | Coding of in-band data..... | 46 |
| 3.10.3.2 | Identification marker | 46 |
| 3.10.3.3 | Interleaving | 46 |
| 3.10.3.4 | Mapping on a Burst..... | 46 |
| 3.10.4 | SID_FIRST_P2..... | 46 |
| 3.10.4.1 | Coding of in-band data..... | 46 |
| 3.10.4.2 | Interleaving | 46 |
| 3.10.4.3 | Mapping on a Burst..... | 47 |
| 3.10.5 | SID_FIRST_INH..... | 47 |
| 3.10.5.1 | Coding of in-band data..... | 47 |
| 3.10.5.2 | Identification marker | 47 |

| | | |
|----------|--|----|
| 3.10.5.3 | Interleaving | 47 |
| 3.10.5.4 | Mapping on a Burst | 47 |
| 3.10.6 | ONSET | 47 |
| 3.10.6.1 | Coding of in-band data | 47 |
| 3.10.6.2 | Interleaving | 48 |
| 3.10.6.3 | Mapping on a Burst | 48 |
| 3.10.7 | SPEECH | 48 |
| 3.10.7.1 | Coding of the in-band data | 48 |
| 3.10.7.2 | Ordering according to subjective importance | 48 |
| 3.10.7.3 | Parity for speech frames | 49 |
| 3.10.7.4 | Convolutional encoder | 50 |
| 3.10.7.5 | Interleaving | 54 |
| 3.10.7.6 | Mapping on a Burst | 54 |
| 3.10.8 | RATSCCH_MARKER | 54 |
| 3.10.8.1 | Coding of in-band data | 55 |
| 3.10.8.2 | Identification marker | 55 |
| 3.10.8.3 | Interleaving | 55 |
| 3.10.8.4 | Mapping on a Burst | 55 |
| 3.10.9 | RATSCCH_DATA | 55 |
| 3.10.9.1 | Coding of in-band data | 55 |
| 3.10.9.2 | Parity and convolutional encoding for the RATSCCH message | 55 |
| 3.10.9.3 | Interleaving | 56 |
| 3.10.9.4 | Mapping on a Burst | 56 |
| 3.11 | Data channel for ECSD at full rate, 29.0 kbit/s radio interface rate (28.8 kbit/s services (E-TCH/F28.8))..... | 56 |
| 3.11.1 | Interface with user unit | 56 |
| 3.11.2 | Block code | 56 |
| 3.11.2.1 | Repetition bits | 56 |
| 3.11.2.2 | Reed Solomon encoder | 56 |
| 3.11.3 | Convolutional encoder | 58 |
| 3.11.3.1 | Tailing bits for a data frame | 58 |
| 3.11.3.2 | Convolutional encoding for a data frame | 58 |
| 3.11.4 | Interleaving | 59 |
| 3.11.5 | Mapping on a Burst | 59 |
| 3.12 | Data channel for ECSD at full rate, 32.0 kbit/s radio interface rate (32.0 kbit/s services (E-TCH/F32.0))..... | 60 |
| 3.12.1 | Interface with user unit | 60 |
| 3.12.2 | Block code | 60 |
| 3.12.3 | Convolutional encoder | 61 |
| 3.12.3.1 | Tailing bits for a data frame | 61 |
| 3.12.3.2 | Convolutional encoding for a data frame | 61 |
| 3.12.4 | Interleaving | 61 |
| 3.12.5 | Mapping on a Burst | 61 |
| 3.13 | Data channel for ECSD at full rate, 43.5 kbit/s radio interface rate (43.2 kbit/s services (E-TCH/F43.2))..... | 62 |
| 3.13.1 | Interface with user unit | 62 |
| 3.13.2 | Convolutional encoder | 62 |
| 3.13.2.1 | Tailing bits for a data frame | 62 |
| 3.13.2.2 | Convolutional encoding for a data frame | 62 |
| 3.13.3 | Interleaving | 62 |
| 3.13.4 | Mapping on a Burst | 62 |
| 4 | Control Channels..... | 62 |
| 4.1 | Slow associated control channel (SACCH) | 62 |
| 4.1.1 | Block constitution | 62 |
| 4.1.2 | Block code | 63 |
| 4.1.3 | Convolutional encoder | 63 |
| 4.1.4 | Interleaving | 63 |
| 4.1.5 | Mapping on a Burst | 64 |
| 4.2 | Fast associated control channel at full rate (FACCH/F) | 64 |
| 4.2.1 | Block constitution | 64 |
| 4.2.2 | Block code | 64 |
| 4.2.3 | Convolutional encoder | 64 |
| 4.2.4 | Interleaving | 64 |

| | | |
|-----------|--|----|
| 4.2.5 | Mapping on a Burst | 64 |
| 4.3 | Fast associated control channel at half rate (FACCH/H)..... | 65 |
| 4.3.1 | Block constitution | 65 |
| 4.3.2 | Block code | 65 |
| 4.3.3 | Convolutional encoder | 65 |
| 4.3.4 | Interleaving | 65 |
| 4.3.5 | Mapping on a Burst | 65 |
| 4.4 | Broadcast control, Paging, Access grant, Notification and Cell broadcast channels (BCCH, PCH, AGCH, NCH, CBCH), CTS Paging and Access grant channels (CTSPCH, CTSAGCH) | 66 |
| 4.5 | Stand-alone dedicated control channel (SDCCH) | 66 |
| 4.6 | Random access channel (RACH) | 66 |
| 4.7 | Synchronization channel (SCH), Compact synchronization channel (CSCH), CTS Beacon and Access request channels (CTSBCH-SB, CTSARCH) | 67 |
| 4.8 | Access Burst on circuit switched channels other than RACH | 67 |
| 4.9 | Access Bursts for uplink access on a channel used for VGCS | 68 |
| 4.10 | Fast associated control channel at ECSD E-TCH/F (E-FACCH/F)..... | 68 |
| 4.10.1 | Block constitution | 68 |
| 4.10.2 | Block code | 68 |
| 4.10.3 | Convolutional encoder | 68 |
| 4.10.4 | Interleaving | 68 |
| 4.10.5 | Mapping on a Burst | 68 |
| 5 | Packet Switched Channels | 69 |
| 5.1 | Packet data traffic channel (PDTCH)..... | 69 |
| 5.1.1 | Packet data block type 1 (CS-1)..... | 69 |
| 5.1.2 | Packet data block type 2 (CS-2)..... | 69 |
| 5.1.2.1 | Block constitution | 69 |
| 5.1.2.2 | Block code | 69 |
| 5.1.2.3 | Convolutional encoder | 70 |
| 5.1.2.4 | Interleaving | 70 |
| 5.1.2.5 | Mapping on a burst..... | 70 |
| 5.1.3 | Packet data block type 3 (CS-3)..... | 70 |
| 5.1.3.1 | Block constitution | 70 |
| 5.1.3.2 | Block code | 70 |
| 5.1.3.3 | Convolutional encoder | 71 |
| 5.1.3.4 | Interleaving | 71 |
| 5.1.3.5 | Mapping on a burst..... | 71 |
| 5.1.4 | Packet data block type 4 (CS-4)..... | 71 |
| 5.1.4.1 | Block constitution | 71 |
| 5.1.4.2 | Block code | 72 |
| 5.1.4.3 | Convolutional encoder | 72 |
| 5.1.4.4 | Interleaving | 72 |
| 5.1.4.5 | Mapping on a burst..... | 72 |
| 5.1.5 | Packet data block type 5 (MCS-1) | 73 |
| 5.1.5.1 | Downlink (MCS-1 DL) | 73 |
| 5.1.5.1.1 | Block constitution..... | 73 |
| 5.1.5.1.2 | USF precoding..... | 73 |
| 5.1.5.1.3 | Header coding..... | 73 |
| 5.1.5.1.4 | Data coding..... | 74 |
| 5.1.5.1.5 | Interleaving..... | 74 |
| 5.1.5.1.6 | Mapping on a burst..... | 75 |
| 5.1.5.2 | Uplink (MCS-1 UL)..... | 75 |
| 5.1.5.2.1 | Block constitution..... | 75 |
| 5.1.5.2.2 | Header coding..... | 75 |
| 5.1.5.2.3 | Data coding..... | 76 |
| 5.1.5.2.4 | Interleaving..... | 76 |
| 5.1.5.2.5 | Mapping on a burst..... | 76 |
| 5.1.6 | Packet data block type 6 (MCS-2) | 77 |
| 5.1.6.1 | Downlink (MCS-2 DL) | 77 |
| 5.1.6.1.1 | Block constitution..... | 77 |
| 5.1.6.1.2 | USF precoding..... | 77 |

- 5.1.6.1.3 Header coding.....77
- 5.1.6.1.4 Data coding.....77
- 5.1.6.1.5 Interleaving.....78
- 5.1.6.1.6 Mapping on a burst.....78
- 5.1.6.2 Uplink (MCS-2 UL).....78
 - 5.1.6.2.1 Block constitution.....78
 - 5.1.6.2.2 Header coding.....78
 - 5.1.6.2.3 Data coding.....78
 - 5.1.6.2.4 Interleaving.....78
 - 5.1.6.2.5 Mapping on a burst.....78
- 5.1.7 Packet data block type 7 (MCS-3).....78
 - 5.1.7.1 Downlink (MCS-3 DL).....78
 - 5.1.7.1.1 Block constitution.....78
 - 5.1.7.1.2 USF precoding.....78
 - 5.1.7.1.3 Header coding.....79
 - 5.1.7.1.4 Data coding.....79
 - 5.1.7.1.5 Interleaving.....79
 - 5.1.7.1.6 Mapping on a burst.....80
 - 5.1.7.2 Uplink (MCS-3 UL).....80
 - 5.1.7.2.1 Block constitution.....80
 - 5.1.7.2.2 Header coding.....80
 - 5.1.7.2.3 Data coding.....80
 - 5.1.7.2.4 Interleaving.....80
 - 5.1.7.2.5 Mapping on a burst.....80
- 5.1.8 Packet data block type 8 (MCS-4).....80
 - 5.1.8.1 Downlink (MCS-4 DL).....80
 - 5.1.8.1.1 Block constitution.....80
 - 5.1.8.1.2 USF precoding.....80
 - 5.1.8.1.3 Header coding.....80
 - 5.1.8.1.4 Data coding.....80
 - 5.1.8.1.5 Interleaving.....81
 - 5.1.8.1.6 Mapping on a burst.....81
 - 5.1.8.2 Uplink (MCS-4 UL).....81
 - 5.1.8.2.1 Block constitution.....81
 - 5.1.8.2.2 Header coding.....81
 - 5.1.8.2.3 Data coding.....81
 - 5.1.8.2.4 Interleaving.....82
 - 5.1.8.2.5 Mapping on a burst.....82
- 5.1.9 Packet data block type 9 (MCS-5).....82
 - 5.1.9.1 Downlink (MCS-5 DL).....82
 - 5.1.9.1.1 Block constitution.....82
 - 5.1.9.1.2 USF precoding.....82
 - 5.1.9.1.3 Header coding.....82
 - 5.1.9.1.4 Data coding.....83
 - 5.1.9.1.5 Interleaving.....84
 - 5.1.9.1.6 Mapping on a burst.....84
 - 5.1.9.2 Uplink (MCS-5 UL).....85
 - 5.1.9.2.1 Block constitution.....85
 - 5.1.9.2.2 Header coding.....85
 - 5.1.9.2.3 Data coding.....86
 - 5.1.9.2.4 Interleaving.....86
 - 5.1.9.2.5 Mapping on a burst.....86
- 5.1.10 Packet data block type 10 (MCS-6).....87
 - 5.1.10.1 Downlink (MCS-6 DL).....87
 - 5.1.10.1.1 Block constitution.....87
 - 5.1.10.1.2 USF precoding.....87
 - 5.1.10.1.3 Header coding.....87
 - 5.1.10.1.4 Data coding.....87
 - 5.1.10.1.5 Interleaving.....88
 - 5.1.10.1.6 Mapping on a burst.....88

| | | |
|------------|---|----|
| 5.1.10.2 | Uplink (MCS-6 UL) | 88 |
| 5.1.10.2.1 | Block constitution | 88 |
| 5.1.10.2.2 | Header coding | 88 |
| 5.1.10.2.3 | Data coding | 88 |
| 5.1.10.2.4 | Interleaving | 88 |
| 5.1.10.2.5 | Mapping on a burst | 88 |
| 5.1.11 | Packet data block type 11 (MCS-7) | 89 |
| 5.1.11.1 | Downlink (MCS-7 DL) | 89 |
| 5.1.11.1.1 | Block constitution | 89 |
| 5.1.11.1.2 | USF precoding | 89 |
| 5.1.11.1.3 | Header coding | 89 |
| 5.1.11.1.4 | Data coding | 90 |
| 5.1.11.1.5 | Interleaving | 91 |
| 5.1.11.1.6 | Mapping on a burst | 91 |
| 5.1.11.2 | Uplink (MCS-7 UL) | 91 |
| 5.1.11.2.1 | Block constitution | 91 |
| 5.1.11.2.2 | Header coding | 92 |
| 5.1.11.2.3 | Data coding | 92 |
| 5.1.11.2.4 | Interleaving | 92 |
| 5.1.11.2.5 | Mapping on a burst | 93 |
| 5.1.12 | Packet data block type 12 (MCS-8) | 93 |
| 5.1.12.1 | Downlink (MCS-8 DL) | 93 |
| 5.1.12.1.1 | Block constitution | 93 |
| 5.1.12.1.2 | USF precoding | 93 |
| 5.1.12.1.3 | Header coding | 93 |
| 5.1.12.1.4 | Data coding | 93 |
| 5.1.12.1.5 | Interleaving | 94 |
| 5.1.12.1.6 | Mapping on a burst | 95 |
| 5.1.12.2 | Uplink (MCS-8 UL) | 95 |
| 5.1.12.2.1 | Block constitution | 95 |
| 5.1.12.2.2 | Header coding | 95 |
| 5.1.12.2.3 | Data coding | 95 |
| 5.1.12.2.4 | Interleaving | 95 |
| 5.1.12.2.5 | Mapping on a burst | 95 |
| 5.1.13 | Packet data block type 13 (MCS-9) | 95 |
| 5.1.13.1 | Downlink (MCS-9 DL) | 95 |
| 5.1.13.1.1 | Block constitution | 95 |
| 5.1.13.1.2 | USF precoding | 95 |
| 5.1.13.1.3 | Header coding | 95 |
| 5.1.13.1.4 | Data coding | 95 |
| 5.1.13.1.5 | Interleaving | 96 |
| 5.1.13.1.6 | Mapping on a burst | 96 |
| 5.1.13.2 | Uplink (MCS-9 UL) | 97 |
| 5.1.13.2.1 | Block constitution | 97 |
| 5.1.13.2.2 | Header coding | 97 |
| 5.1.13.2.3 | Data coding | 97 |
| 5.1.13.2.4 | Interleaving | 97 |
| 5.1.13.2.5 | Mapping on a burst | 97 |
| 5.2 | Packet control channels (PACCH, PBCCH, PAGCH, PPCH, PNCH, PTCCH, CPBCCCH, CPAGCH, CPPCH, and CPNCH) | 97 |
| 5.3 | Packet random access channel (PRACH and CPRACH) | 97 |
| 5.3.1 | Packet Access Burst | 97 |
| 5.3.2 | Extended Packet Access Burst | 97 |
| 5.4 | Access Burst on packet switched channels other than PRACH and CPRACH | 98 |

Annex A (informative): Summary of Channel Types114
Annex B (informative): Summary of Polynomials Used for Convolutional Codes.....116
Annex C (informative): Change control history117
History118

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 SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in 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 European Standard (Telecommunications series) has been produced by ETSI Technical Committee Special Mobile Group (SMG), and is now submitted for the ETSI standards One-step Approval Procedure.

The present document specifies the data blocks given to the encryption unit. It includes the specification of encoding, reordering, interleaving and the stealing flag within the digital cellular telecommunications system (Phase 2+).

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

Version 8.x.y

where:

- 8 GSM Phase 2+ Release 1999.
- x the second digit is incremented for changes of substance, i.e. technical enhancements, corrections, updates, etc.;
- y the third digit is incremented when editorial only changes have been incorporated in the specification.

| Proposed national transposition dates | |
|--|---------------------------------|
| Date of latest announcement of this EN (doa): | 3 months after ETSI publication |
| Date of latest publication of new National Standard or endorsement of this EN (dop/e): | 6 months after doa |
| Date of withdrawal of any conflicting National Standard (dow): | 6 months after doa |

1 Scope

A reference configuration of the transmission chain is shown in GSM 05.01 [4]. According to this reference configuration, the present document specifies the data blocks given to the encryption unit.

It includes the specification of encoding, reordering, interleaving and the stealing flag. It does not specify the channel decoding method.

The definition is given for each kind of logical channel, starting from the data provided to the channel encoder by the speech coder, the data terminal equipment, or the controller of the Mobile Station (MS) or Base Transceiver Station (BTS). The definitions of the logical channel types used in this technical specification are given in GSM 05.02 [5], a summary is in annex A.

1.1 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.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.
- For this Release 1999 document, references to GSM documents are for Release 1999 versions (version 8.x.y).

- [1] GSM 01.04: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".
- [2] GSM 04.08: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification".
- [3] GSM 04.21: "Digital cellular telecommunications system (Phase 2+); Rate adaption on the Mobile Station - Base Station System (MS - BSS) interface".
- [4] GSM 05.01: "Digital cellular telecommunications system (Phase 2+); Physical layer on the radio path General description".
- [5] GSM 05.02: "Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path".
- [6] GSM 05.05: "Digital cellular telecommunications system (Phase 2+); Radio Transmission and Reception".
- [7] GSM 05.09: "Digital cellular telecommunications system (Phase 2+); Link adaptation".
- [8] GSM 06.10: "Digital cellular telecommunications system; Full rate speech transcoding".
- [9] GSM 06.20: "Digital cellular telecommunications system; Half rate speech transcoding".
- [10] GSM 06.60: "Digital cellular telecommunications system; Enhanced Full Rate (EFR) speech transcoding".
- [11] GSM 06.90: "Digital cellular telecommunications system; Adaptive Multi-Rate speech transcoding".
- [12] GSM 06.93: "Digital cellular telecommunications system; Discontinuous transmission (DTX) for Adaptive Multi-Rate speech traffic channels".

- [13] GSM 03.64: "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Overall description of the GPRS Radio Interface; Stage 2".
- [14] GSM 03.52: "Digital cellular telecommunications system (Phase 2+); GSM Cordless Telephony System (CTS), Phase 1; Lower layers of the CTS Radio Interface; Stage 2".

1.2 Abbreviations

Abbreviations used in the present document are listed in GSM 01.04.

2 General

2.1 General organization

Each channel has its own coding and interleaving scheme. However, the channel coding and interleaving is organized in such a way as to allow, as much as possible, a unified decoder structure.

Each channel uses the following sequence and order of operations:

- the information bits are coded with a systematic block code, building words of information + parity bits;
- these information + parity bits are encoded with a convolutional code, building the coded bits;
- reordering and interleaving the coded bits, and adding a stealing flag, gives the interleaved bits.

All these operations are made block by block, the size of which depends on the channel. However, most of the channels use a block of 456 coded bits which is interleaved and mapped onto bursts in a very similar way for all of them. Figures 1a and 1b give a diagram showing the general structure of the channel coding.

This block of 456 coded bits is the basic structure of the channel coding scheme. In the case of full rate speech TCH, this block carries the information of one speech frame. In case of control channels, it carries one message.

In the case of half rate speech TCH, the information of one speech frame is carried in a block of 228 coded bits.

In the case of the Enhanced full rate speech the information bits coming out of the source codec first go through a preliminary channel coding, then the channel coding as described above takes place.

In the case of a packet switched channel the block of 456 or 1384 coded bits carries one radio block.

In the case of an enhanced circuit switched channel the block of 1368 coded bits (456 coded symbols) carries one radio block.

In the case of FACCH, a coded message block of 456 bits is divided into eight sub-blocks. The first four sub-blocks are sent by stealing the even numbered bits of four timeslots in consecutive frames used for the TCH. The other four sub-blocks are sent by stealing the odd numbered bits of the relevant timeslot in four consecutive used frames delayed 2 or 4 frames relative to the first frame. Along with each block of 456 coded bits there is, in addition, a stealing flag (8 bits), indicating whether the block belongs to the TCH or to the FACCH. In the case of SACCH, BCCH, CCCH or CTSCCH, this stealing flag is dummy. In the case of a packet switched channel, these bits are used to indicate the coding scheme used.

In the case of E-FACCH/F, a coded message block of 456 bits is divided into four sub-blocks. The four sub-blocks are sent by stealing all symbols of four timeslots in consecutive frames used for the E-TCH and using GMSK modulation. The indication of the E-FACCH/F is based on the identification of the modulation. Along with each block of 456 coded bits there is, in addition, a stealing flag (8 bits), indicating whether the block belongs to the E-FACCH, FACCH or TCH.

Some cases do not fit in the general organization, and use short blocks of coded bits which are sent completely in one timeslot. They are the random access messages of:

- the RACH;
- or PRACH and CPRACH;

on uplink and the synchronization information broadcast on the SCH or CSCH on the downlink. In CTS, they are the access request message of the CTSARCH on uplink and the information broadcast on the CTSBCH-SB on downlink.

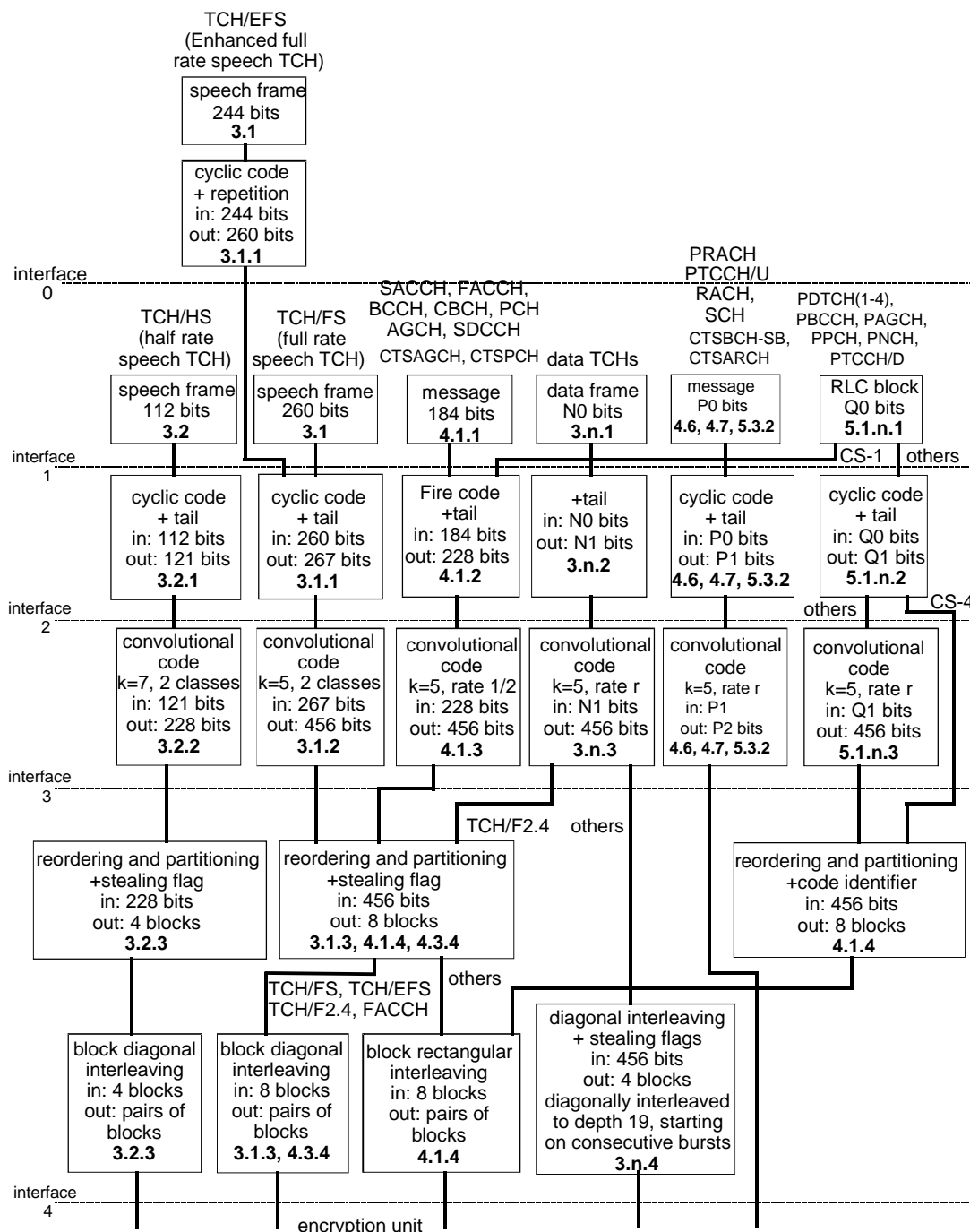


Figure 1a: Channel Coding and Interleaving Organization

In each box, the last line indicates the chapter defining the function. In the case of RACH, P0 = 8 and P1 = 18; in the case of SCH, CSCH, CTSBCH-SB and CTSARCH, P0 = 25 and P1 = 39. In the case of data TCHs, N0, N1 and n depend on the type of data TCH.

Interfaces:

- 1) information bits (d);
- 2) information + parity + tail bits (u);
- 3) coded bits (c);
- 4) interleaved bits (e).

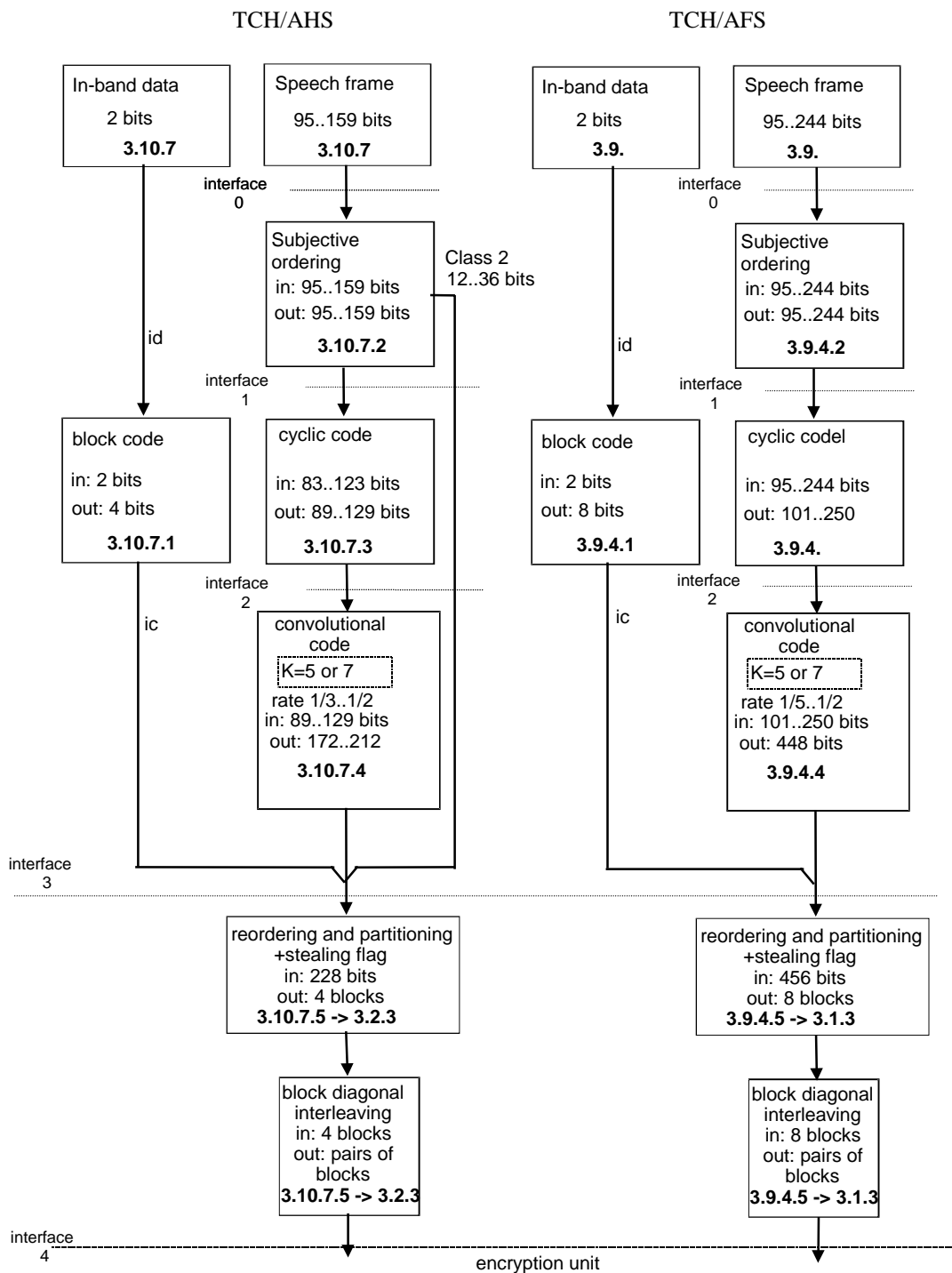


Figure 1b: Channel Coding and Interleaving Organization, adaptive multi-rate speech

In each box, the last line indicates the chapter defining the function.

Interfaces:

- 0) speech bits from the speech encoder (s);
- 1) reordered speech bits (d);
- 2) speech + parity + tail bits (u);
- 3) coded bits (c);
- 4) interleaved bits (e).

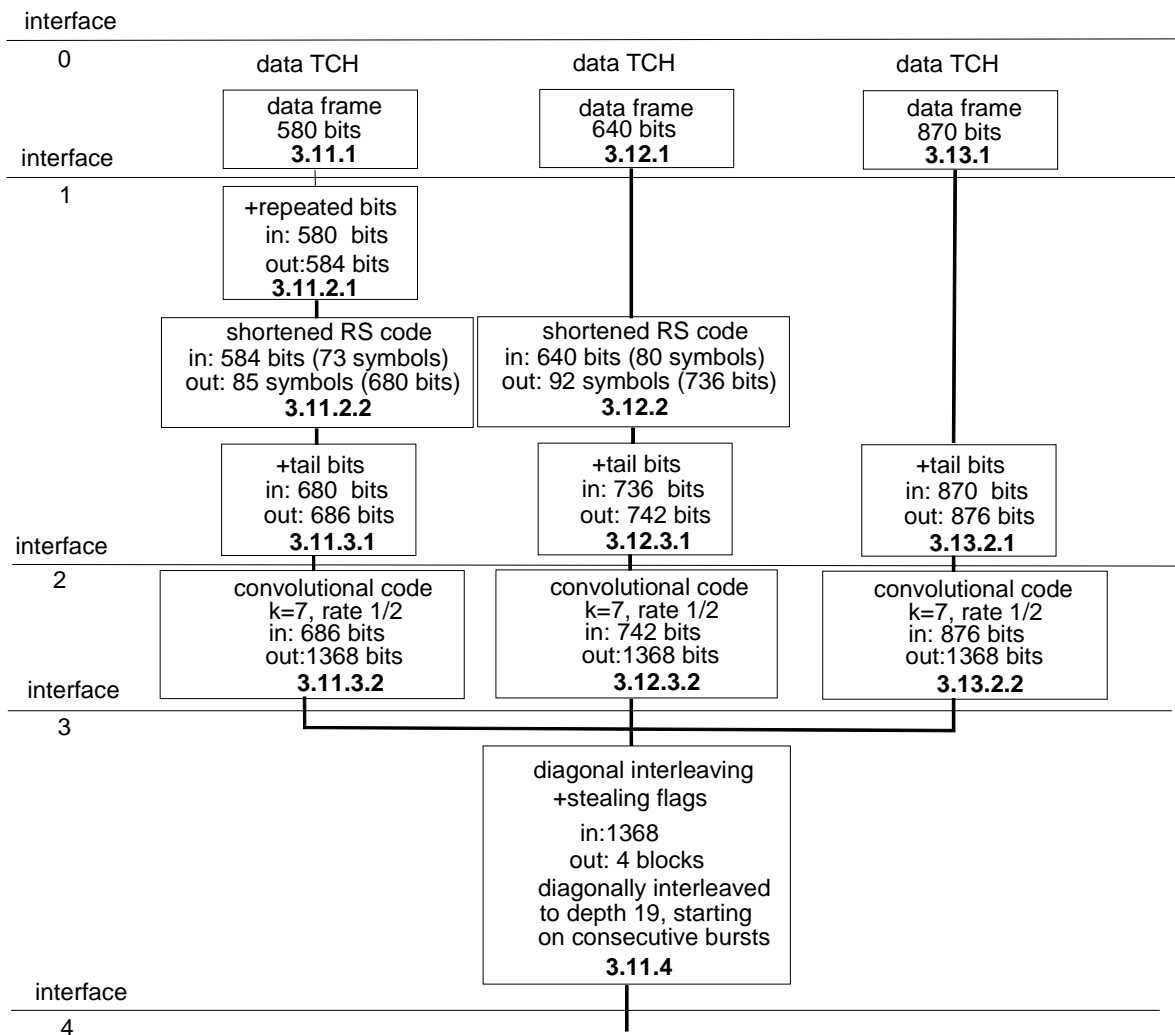


Figure 2a: Channel Coding and Interleaving Organization for ECSD 8-PSK modulated signals

In each box, the last line indicates the chapter defining the function.

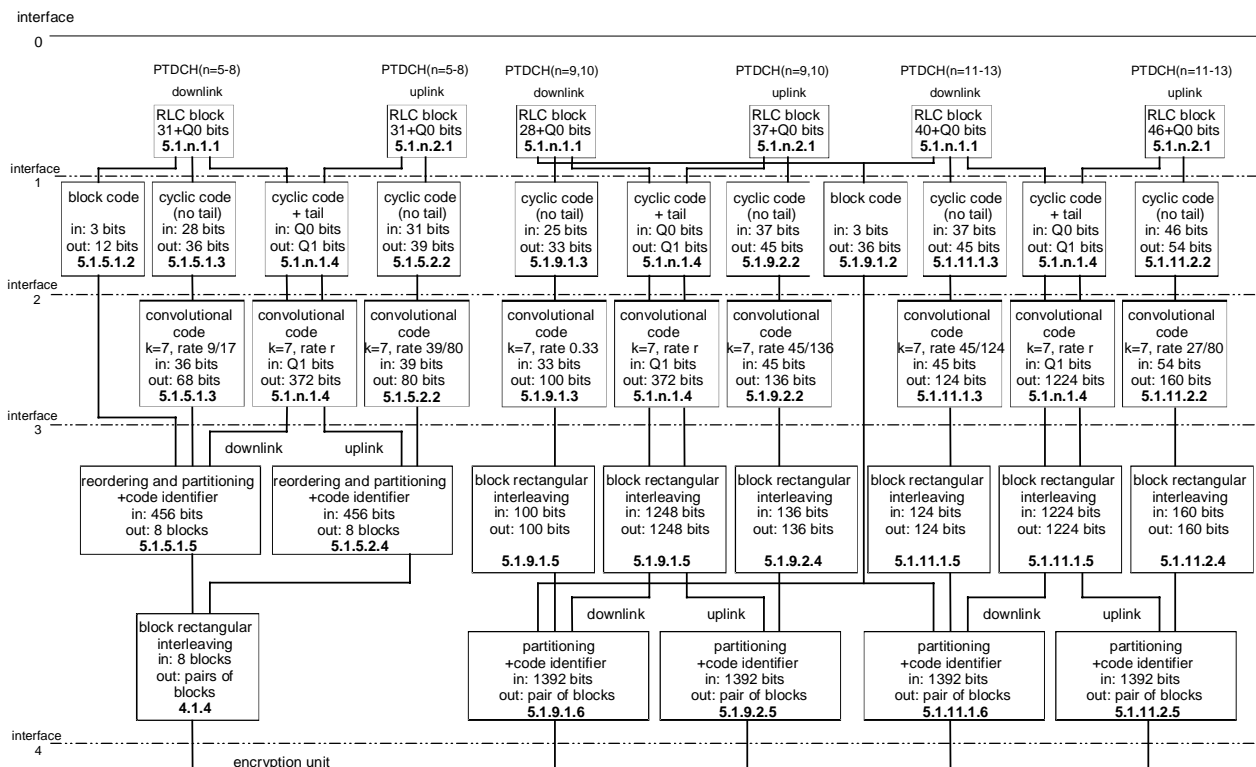


Figure 2b: Channel Coding and Interleaving Organization for EGPRS Packet Data Channels

In each box, the last line indicates the chapter defining the function.

2.2 Naming Convention

For ease of understanding a naming convention for bits is given for use throughout the technical specification:

- General naming:
 - "k" and "j" for numbering of bits in data blocks and bursts;
 - " K_x " gives the amount of bits in one block, where "x" refers to the data type;
 - "n" is used for numbering of delivered data blocks where;
 - "N" marks a certain data block;
 - "B" is used for numbering of bursts or blocks where;
 - " B_0 " marks the first burst or block carrying bits from the data block with $n = 0$ (first data block in the transmission).
- Data delivered to the preliminary channel encoding unit (for EFR only):
 - $s(k)$ for $k = 1, \dots, K_S$
- Data delivered by the preliminary channel encoding unit (for EFR only) before bits rearrangement
 - $w(k)$ for $k = 1, \dots, K_W$
- Data bits delivered to the encoding unit (interface 1 in figure 1):
 - $d(k)$ for $k = 0, 1, \dots, K_d - 1$
- Data symbols delivered to the encoding unit:

$D(k)$ for $k = 0, 1, \dots, K_D - 1$

- Input in-band data bits (for TCH/AMR only):

$id(k)$ for $k = 0, 1$

- Encoded in-band data bits (for TCH/AMR only):

$ic(k)$ for $k = 0, 1, \dots, 3$ TCH/AHS speech frames or
 $k = 0, 1, \dots, 7$ TCH/AFS speech frames or
 $k = 0, 1, \dots, 15$ TCH/AMR, SID frames

- Code identifying the used coding scheme (for packet switched channels only):

$q(k)$ for $k = 0, 1, \dots, 7$

- Data bits after the first encoding step (block code, cyclic code; interface 2 in figure 1):

$u(k)$ for $k = 0, 1, \dots, K_U - 1$

- Data symbols after the first encoding step (block code):

$U(k)$ for $k = 0, 1, \dots, K_U - 1$

- Data put into the shift register of the convolutional code and calculated from the data bits $u(k)$ and the feedback bits in recursive systematic convolutional codes

$r(k)$ for $k = 0, 1, \dots, K_r - 1$

- Data after the second encoding step (convolutional code ; interface 3 in figure 1):

$c(n, k)$ or $c(k)$ for $k = 0, 1, \dots, K_C - 1$
 $n = 0, 1, \dots, N, N+1, \dots$

- Interleaved data bits:

$i(B, k)$ for $k = 0, 1, \dots, K_i - 1$
 $B = B_0, B_0 + 1, \dots$

- Interleaved data symbols:

$I(B, k)$ for $k = 0, 1, \dots, K_I - 1$
 $B = B_0, B_0 + 1, \dots$

- Bits in one burst (interface 4 in figure 1):

$e(B, k)$ for $k = 0, 1, \dots, 114, 115$
 $B = B_0, B_0 + 1, \dots$

- Symbols in one burst (interface 4 in figure 2):

$E(B, k)$ for $k = 0, 1, \dots, 114, 115$
 $B = B_0, B_0 + 1, \dots$

- E-IACCH messages delivered to the block coding of inband signalling (for ECSD only):

$im(k)$ or $im(n, k)$
for $k = 0, 1, 2$
 $n = 0, 1, \dots, N, N+1, \dots$

- E-IACCH bits delivered to the mapping on one burst (for ECSD only):

$$ib(B,k) \quad \text{for} \quad k = 0,1,\dots,5$$

$$B = B_0, B_0+1,\dots$$

- E-IACCH symbols in one burst (for ECSD only):

$$HL(B) \text{ and } HU(B)$$

$$\text{for} \quad B = B_0, B_0+1,\dots$$

3 Traffic Channels (TCH)

Two kinds of traffic channel are considered: speech and data. Both of them use the same general structure (see figure 1), and in both cases, a piece of information can be stolen by the FACCH.

3.1 Speech channel at full rate (TCH/FS and TCH/EFS)

The speech coder (whether Full rate or Enhanced full rate) delivers to the channel encoder a sequence of blocks of data. In case of a full rate and enhanced full rate speech TCH, one block of data corresponds to one speech frame.

For the full rate coder each block contains 260 information bits, including 182 bits of class 1 (protected bits), and 78 bits of class 2 (no protection), (see table 2).

The bits delivered by the speech coder are received in the order indicated in GSM 06.10 and have to be rearranged according to table 2 before channel coding as defined in subclauses 3.1.1 to 3.1.4. The rearranged bits are labelled $\{d(0),d(1),\dots,d(259)\}$, defined in the order of decreasing importance.

For the EFR coder each block contains 244 information bits. The block of 244 information bits, labelled $s(1).., s(244)$, passes through a preliminary stage, applied only to EFR (see figure 1) which produces 260 bits corresponding to the 244 input bits and 16 redundancy bits. Those 16 redundancy bits correspond to 8 CRC bits and 8 repetition bits, as described in subclause 3.1.1. The 260 bits, labelled $w(1)..w(260)$, have to be rearranged according to table 7 before they are delivered to the channel encoding unit which is identical to that of the TCH/FS. The 260 bits block includes 182 bits of class 1(protected bits) and 78 bits of class 2 (no protection). The class 1 bits are further divided into the class 1a and class 1b, class 1a bits being protected by a cyclic code and the convolutional code whereas the class 1b are protected by the convolutional code only.

3.1.1 Preliminary channel coding for EFR only

3.1.1.1 CRC calculation

An 8-bit CRC is used for error-detection. These 8 parity bits (bits $w253-w260$) are generated by the cyclic generator polynomial: $g(D) = D^8 + D^4 + D^3 + D^2 + 1$ from the 65 most important bits (50 bits of class 1a and 15 bits of class 1b). These 65 bits ($b(1)-b(65)$) are taken from the table 5 in the following order (read row by row, left to right):

| | | | | | | | | | |
|------|------|------|------|------|-----|------|------|------|------|
| s39 | s40 | s41 | s42 | s43 | s44 | s48 | s87 | s45 | s2 |
| s3 | s8 | s10 | s18 | s19 | s24 | s46 | s47 | s142 | s143 |
| s144 | s145 | s146 | s147 | s92 | s93 | s195 | s196 | s98 | s137 |
| s148 | s94 | s197 | s149 | s150 | s95 | s198 | s4 | s5 | s11 |
| s12 | s16 | s9 | s6 | s7 | s13 | s17 | s20 | s96 | s199 |
| s1 | s14 | s15 | s21 | s25 | s26 | s28 | s151 | s201 | s190 |
| s240 | s88 | s138 | s191 | s241 | | | | | |

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

- $b(1)D^{72} + b(2)D^{71} + \dots + b(65)D^8 + p(1)D^7 + p(2)D^6 + \dots + p(7)D^1 + p(8)$;
- $p(1) - p(8)$: the parity bits ($w253-w260$);

- $b(1) - b(65)$ = the data bits from the table above;

when divided by $g(D)$, yields a remainder equal to 0.

3.1.1.2 Repetition bits

The repeated bits are s_{70} , s_{120} , s_{173} and s_{223} . They correspond to one of the bits in each of the PULSE_5, the most significant one not protected by the channel coding stage.

3.1.1.3 Correspondence between input and output of preliminary channel coding

The preliminary coded bits $w(k)$ for $k = 1$ to 260 are hence defined by:

$$w(k) = s(k) \quad \text{for } k = 1 \text{ to } 71$$

$$w(k) = s(k-2) \text{ for } k = 74 \text{ to } 123$$

$$w(k) = s(k-4) \text{ for } k = 126 \text{ to } 178$$

$$w(k) = s(k-6) \text{ for } k = 181 \text{ to } s_{230}$$

$$w(k) = s(k-8) \text{ for } k = 233 \text{ to } s_{252}$$

Repetition bits:

$$w(k) = s(70) \quad \text{for } k = 72 \text{ and } 73$$

$$w(k) = s(120) \text{ for } k = 124 \text{ and } 125$$

$$w(k) = s(173) \text{ for } k = 179 \text{ and } 180$$

$$w(k) = s(223) \text{ for } k = 231 \text{ and } 232$$

Parity bits:

$$w(k) = p(k-252) \text{ for } k = 253 \text{ to } 260$$

3.1.2 Channel coding for FR and EFR

3.1.2.1 Parity and tailing for a speech frame

a) Parity bits:

The first 50 bits of class 1 (**known as class 1a for the EFR**) are protected by three parity bits used for error detection. These parity bits are added to the 50 bits, according to a degenerate (shortened) cyclic code (53,50,2), using the generator polynomial:

$$g(D) = D^3 + D + 1$$

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

$$d(0)D^{52} + d(1)D^{51} + \dots + d(49)D^3 + p(0)D^2 + p(1)D + p(2)$$

where $p(0)$, $p(1)$, $p(2)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to:

$$1 + D + D^2$$

b) Tailing bits and reordering:

The information and parity bits of class 1 are reordered, defining 189 information + parity + tail bits of class 1, $\{u(0), u(1), \dots, u(188)\}$ defined by:

$$u(k) = d(2k) \quad \text{and} \quad u(184-k) = d(2k+1) \quad \text{for } k = 0, 1, \dots, 90$$

$$\begin{aligned} u(91+k) &= p(k) && \text{for } k = 0,1,2 \\ u(k) &= 0 && \text{for } k = 185,186,187,188 \text{ (tail bits)} \end{aligned}$$

3.1.2.2 Convolutional encoder

The class 1 bits are encoded with the 1/2 rate convolutional code defined by the polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

The coded bits $\{c(0), c(1), \dots, c(455)\}$ are then defined by:

$$\begin{aligned} \text{- class 1: } c(2k) &= u(k) + u(k-3) + u(k-4) \\ c(2k+1) &= u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0,1,\dots,188 \\ u(k) &= 0 \text{ for } k < 0 \\ \text{- class 2: } c(378+k) &= d(182+k) \quad \text{for } k = 0,1,\dots,77 \end{aligned}$$

3.1.3 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$\begin{aligned} i(B,j) &= c(n,k), \quad \text{for } k = 0,1,\dots,455 \\ n &= 0,1,\dots,N,N+1,\dots \\ B &= B_0 + 4n + (k \bmod 8) \\ j &= 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4) \end{aligned}$$

See table 1. The result of the interleaving is a distribution of the reordered 456 bits of a given data block, $n = N$, over 8 blocks using the even numbered bits of the first 4 blocks ($B = B_0 + 4N + 0, 1, 2, 3$) and odd numbered bits of the last 4 blocks ($B = B_0 + 4N + 4, 5, 6, 7$). The reordered bits of the following data block, $n = N+1$, use the even numbered bits of the blocks $B = B_0 + 4N + 4, 5, 6, 7$ ($B = B_0 + 4(N+1) + 0, 1, 2, 3$) and the odd numbered bits of the blocks $B = B_0 + 4(N+1) + 4, 5, 6, 7$. Continuing with the next data blocks shows that one block always carries 57 bits of data from one data block ($n = N$) and 57 bits of data from the next block ($n = N+1$), where the bits from the data block with the higher number always are the even numbered data bits, and those of the data block with the lower number are the odd numbered bits.

The block of coded data is interleaved "block diagonal", where a new data block starts every 4th block and is distributed over 8 blocks.

3.1.4 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \quad \text{and} \quad e(B,59+j) = i(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$e(B,57) = hl(B) \quad \text{and} \quad e(B,58) = hu(B)$$

The two bits, labelled $hl(B)$ and $hu(B)$ on burst number B are flags used for indication of control channel signalling. For each TCH/FS block not stolen for signalling purposes:

$$hu(B) = 0 \text{ for the first 4 bursts (indicating status of even numbered bits)}$$

$$hl(B) = 0 \text{ for the last 4 bursts (indicating status of odd numbered bits)}$$

For the use of $hl(B)$ and $hu(B)$ when a speech frame is stolen for signalling purposes see subclause 4.2.5.

3.2 Speech channel at half rate (TCH/HS)

The speech coder delivers to the channel encoder a sequence of blocks of data. In case of a half rate speech TCH, one block of data corresponds to one speech frame. Each block contains 112 bits, including 95 bits of class 1 (protected bits), and 17 bits of class 2 (no protection), see tables 3a and 3b.

The bits delivered by the speech coder are received in the order indicated in GSM 06.20 and have to be arranged according to either table 3a or table 3b before channel encoding as defined in subclauses 3.2.1 to 3.2.4. The rearranged bits are labelled $\{d(0),d(1),\dots,d(111)\}$. Table 3a has to be taken if parameter Mode = 0 (which means that the speech encoder is in unvoiced mode), while table 3b has to be taken if parameter Mode = 1, 2 or 3 (which means that the speech encoder is in voiced mode).

3.2.1 Parity and tailing for a speech frame

a) Parity bits:

The most significant 22 class 1 bits $d(73),d(74),\dots,d(94)$ are protected by three parity bits used for error detection. These bits are added to the 22 bits, according to a cyclic code using the generator polynomial:

$$g(D) = D^3 + D + 1$$

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

$$d(73)D^{24} + d(74)D^{23} + \dots + d(94)D^3 + p(0)D^2 + p(1)D + p(2)$$

where $p(0), p(1), p(2)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to:

$$1 + D + D^2.$$

b) Tail bits and reordering:

The information and parity bits of class 1 are reordered, defining 104 information + parity + tail bits of class 1, $\{u(0),u(1),\dots,u(103)\}$ defined by:

$$u(k) = d(k) \quad \text{for } k = 0,1,\dots,94$$

$$u(k) = p(k-95) \quad \text{for } k = 95,96,97$$

$$u(k) = 0 \quad \text{for } k = 98,99,\dots,103 \text{ (tail bits)}$$

3.2.2 Convolutional encoder

The class 1 bits are encoded with the punctured convolutional code defined by the mother polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_5 = 1 + D + D^4 + D^6$$

$$G_6 = 1 + D + D^2 + D^3 + D^4 + D^6$$

and the puncturing matrices:

$$(1,0,1) \quad \text{for } \{u(0),u(1),\dots,u(94)\} \text{ (class 1 information bits);}$$

$$\text{and } \{u(98),u(99),\dots,u(103)\} \text{ (tail bits).}$$

$$(1,1,1) \quad \text{for } \{u(95),u(96),u(97)\} \text{ (parity bits)}$$

In the puncturing matrices, a 1 indicates no puncture and a 0 indicates a puncture.

The coded bits $\{c(0),c(1),\dots,c(227)\}$ are then defined by:

class 1 information bits:

$$c(2k) = u(k)+u(k-2)+u(k-3)+u(k-5)+u(k-6)$$

$$c(2k+1) = u(k)+u(k-1)+u(k-2)+u(k-3)+u(k-4)+u(k-6) \quad \text{for } k = 0,1,\dots,94; u(k) = 0 \text{ for } k < 0$$

parity bits:

$$c(3k-95) = u(k)+u(k-2)+u(k-3)+u(k-5)+u(k-6)$$

$$c(3k-94) = u(k)+u(k-1)+u(k-4)+u(k-6)$$

$$c(3k-93) = u(k)+u(k-1)+u(k-2)+u(k-3)+u(k-4)+u(k-6) \quad \text{for } k = 95,96,97$$

tail bits:

$$c(2k+3) = u(k)+u(k-2)+u(k-3)+u(k-5)+u(k-6)$$

$$c(2k+4) = u(k)+u(k-1)+u(k-2)+u(k-3)+u(k-4)+u(k-6) \quad \text{for } k = 98,99,\dots,103$$

class 2 information bits:

$$c(k+211) = d(k+95) \text{ for } k = 0,1,\dots,16$$

3.2.3 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k) \quad \text{for } \begin{aligned} k &= 0,1,\dots,227 \\ n &= 0,1,\dots,N,N+1,\dots \\ B &= B_0 + 2n + b \end{aligned}$$

The values of b and j in dependence of k are given by table 4.

The result of the interleaving is a distribution of the reordered 228 bits of a given data block, $n = N$, over 4 blocks using the even numbered bits of the first 2 blocks ($B = B_0+2N+0,1$) and the odd numbered bits of the last 2 blocks ($B = B_0+2N+2,3$). The reordered bits of the following data block, $n = N + 1$, use the even numbered bits of the blocks $B = B_0 + 2N + 2,3$ ($B = B_0+2(N+1)+0,1$) and the odd numbered bits of the blocks $B = B_0 + 2(N+1) + 2,3$. Continuing with the next data blocks shows that one block always carries 57 bits of data from one data block ($n = N$) and 57 bits from the next block ($n = N+1$), where the bits from the data block with the higher number always are the even numbered data bits, and those of the data block with the lower number are the odd numbered bits. The block of coded data is interleaved "block diagonal", where a new data block starts every 2nd block and is distributed over 4 blocks.

3.2.4 Mapping on a burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \text{ and } e(B,59+j) = i(B,57+j) \text{ for } j = 0,1,\dots,56$$

and

$$e(B,57) = hl(B) \text{ and } e(B,58) = hu(B)$$

The two bits, labelled $hl(B)$ and $hu(B)$ on burst number B are flags used for indication of control channel signalling. For each TCH/HS block not stolen for signalling purposes:

$$hu(B) = 0 \text{ for the first 2 bursts (indicating status of the even numbered bits)}$$

$$hl(B) = 0 \text{ for the last 2 bursts (indicating status of the odd numbered bits)}$$

For the use of $hl(B)$ and $hu(B)$ when a speech frame is stolen for signalling purposes, see subclause 4.3.5.

3.3 Data channel at full rate, 12.0 kbit/s radio interface rate (9.6 kbit/s services (TCH/F9.6))

The definition of a 12.0 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.3.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 60 information bits (data frames) every 5 ms. Four such blocks are dealt with together in the coding process $\{d(0), \dots, d(239)\}$. For non-transparent services those four blocks shall align with one 240-bit RLP frame.

3.3.2 Block code

The block of $4 * 60$ information bits is not encoded, but only increased with 4 tail bits equal to 0 at the end of the block.

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 239 \\ u(k) &= 0 && \text{for } k = 240, 241, 242, 243 \text{ (tail bits)} \end{aligned}$$

3.3.3 Convolutional encoder

This block of 244 bits $\{u(0), \dots, u(243)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$\begin{aligned} G_0 &= 1 + D^3 + D^4 \\ G_1 &= 1 + D + D^3 + D^4 \end{aligned}$$

resulting in 488 coded bits $\{C(0), C(1), \dots, C(487)\}$ with

$$\begin{aligned} C(2k) &= u(k) + u(k-3) + u(k-4) \\ C(2k+1) &= u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 243 ; u(k) = 0 \text{ for } k < 0 \end{aligned}$$

The code is punctured in such a way that the following 32 coded bits:

$$\{C(11+15j) \text{ for } j = 0, 1, \dots, 31\} \text{ are not transmitted.}$$

The result is a block of 456 coded bits, $\{c(0), c(1), \dots, c(455)\}$

3.3.4 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$\begin{aligned} i(B, j) &= c(n, k) \text{ for } k = 0, 1, \dots, 455 \\ n &= 0, 1, \dots, N, N + 1, \dots \\ B &= B_0 + 4n + (k \bmod 19) + (k \text{ div } 114) \\ j &= (k \bmod 19) + 19 (k \text{ div } 6) \end{aligned}$$

The result of the interleaving is a distribution of the reordered 114 bit of a given data block, $n = N$, over 19 blocks, 6 bits equally distributed in each block, in a diagonal way over consecutive blocks.

Or in other words the interleaving is a distribution of the encoded, reordered 456 bits from four given input data blocks, which taken together give $n = N$, over 22 bursts, 6 bits equally distributed in the first and 22nd bursts, 12 bits distributed in the second and 21st bursts, 18 bits distributed in the third and 20th bursts and 24 bits distributed in the other 16 bursts.

The block of coded data is interleaved "diagonal", where a new block of coded data starts with every fourth burst and is distributed over 22 bursts.

3.3.5 Mapping on a Burst

The mapping is done as specified for TCH/FS in subclause 3.1.4. On bitstealing by a FACCH, see subclause 4.2.5.

3.4 Data channel at full rate, 6.0 kbit/s radio interface rate (4.8 kbit/s services (TCH/F4.8))

The definition of a 6.0 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.4.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 60 information bits (data frames) every 10 ms, $\{d(0), d(1), \dots, d(59)\}$.

In the case where the user unit delivers to the encoder a bit stream organized in blocks of 240 information bits every 40 ms (e.g. RLP frames), the bits $\{d(0), d(1), \dots, d(59), d(60), \dots, d(60+59), d(2*60), \dots, d(2*60+59), d(3*60), \dots, d(3*60+59)\}$ shall be treated as four blocks of 60 bits each as described in the remainder of this clause. To ensure end-to-end synchronization of the 240 bit blocks, the resulting block after coding of the first 120 bits $\{d(0), d(1), \dots, d(60+59)\}$ shall be transmitted in one of the transmission blocks B0, B2, B4 of the channel mapping defined in GSM 05.02.

3.4.2 Block code

Sixteen bits equal to 0 are added to the 60 information bits, the result being a block of 76 bits, $\{u(0), u(1), \dots, u(75)\}$, with:

$$u(19k+p) = d(15k+p) \quad \text{for } k = 0, 1, 2, 3 \text{ and } p = 0, 1, \dots, 14;$$

$$u(19k+p) = 0 \quad \text{for } k = 0, 1, 2, 3 \text{ and } p = 15, 16, 17, 18.$$

Two such blocks forming a block of 152 bits $\{u'(0), u'(1), \dots, u'(151)\}$ are dealt with together in the rest of the coding process:

$$u'(k) = u_1(k), \quad k = 0, 1, \dots, 75 \text{ (} u_1 = \text{1st block)}$$

$$u'(k+76) = u_2(k), \quad k = 0, 1, \dots, 75 \text{ (} u_2 = \text{2nd block)}$$

3.4.3 Convolutional encoder

This block of 152 bits is encoded with the convolutional code of rate 1/3 defined by the following polynomials:

$$G_1 = 1 + D + D^3 + D^4$$

$$G_2 = 1 + D^2 + D^4$$

$$G_3 = 1 + D + D^2 + D^3 + D^4$$

The result is a block of $3 * 152 = 456$ coded bits, $\{c(0), c(1), \dots, c(455)\}$:

$$c(3k) = u'(k) + u'(k-1) + u'(k-3) + u'(k-4)$$

$$c(3k+1) = u'(k) + u'(k-2) + u'(k-4)$$

$$c(3k+2) = u'(k) + u'(k-1) + u'(k-2) + u'(k-3) + u'(k-4) \quad \text{for } k = 0, 1, \dots, 151;$$

$$u'(k) = 0 \text{ for } k < 0$$

3.4.4 Interleaving

The interleaving is done as specified for the TCH/F9.6 in subclause 3.3.4.

3.4.5 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4. On bitstealing for signalling purposes by a FACCH, see subclause 4.2.5.

3.5 Data channel at half rate, 6.0 kbit/s radio interface rate (4.8 kbit/s services (TCH/H4.8))

The definition of a 6.0 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.5.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 60 information bits (data frames) every 10 ms. Four such blocks are dealt with together in the coding process, {d(0),d(1),...,d(239)}.

For non-transparent services those four blocks shall align with one complete 240-bit RLP frame.

3.5.2 Block code

The block encoding is done as specified for the TCH/F9.6 in subclause 3.3.2.

3.5.3 Convolutional encoder

The convolutional encoding is done as specified for the TCH/F9.6 in subclause 3.3.3.

3.5.4 Interleaving

The interleaving is done as specified for the TCH/F9.6 in subclause 3.3.4.

3.5.5 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4. On bitstealing for signalling purposes by a FACCH, see subclause 4.3.5.

3.6 Data channel at full rate, 3.6 kbit/s radio interface rate (2.4 kbit/s and less services (TCH/F2.4))

The definition of a 3.6 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.6.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 36 information bits (data frames) every 10 ms. Two such blocks are dealt with together in the coding process, {d(0),d(1),...,d(71)}.

3.6.2 Block code

This block of 72 information bits is not encoded, but only increased with four tail bits equal to 0 at the end of the block.

$$u(k) = d(k), \quad k = 0, 1, \dots, 71$$

$$u(k) = 0, \quad k = 72, 73, 74, 75 \text{ (tail bits);}$$

3.6.3 Convolutional encoder

This block of 76 bits $\{u(0),u(1),\dots,u(75)\}$ is encoded with the convolutional code of rate 1/6 defined by the following polynomials:

$$G1 = 1 + D + D^3 + D^4$$

$$G2 = 1 + D^2 + D^4$$

$$G3 = 1 + D + D^2 + D^3 + D^4$$

$$G1 = 1 + D + D^3 + D^4$$

$$G2 = 1 + D^2 + D^4$$

$$G3 = 1 + D + D^2 + D^3 + D^4$$

The result is a block of 456 coded bits:

$\{c(0), c(1), \dots, c(455)\}$, defined by

$$c(6k) = c(6k+3) = u(k) + u(k-1) + u(k-3) + u(k-4)$$

$$c(6k+1) = c(6k+4) = u(k) + u(k-2) + u(k-4)$$

$$c(6k+2) = c(6k+5) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-4), \text{ for } k = 0, 1, \dots, 75;$$

$$u(k) = 0 \text{ for } k < 0$$

3.6.4 Interleaving

The interleaving is done as specified for the TCH/FS in subclause 3.1.3.

3.6.5 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4.

3.7 Data channel at half rate, 3.6 kbit/s radio interface rate (2.4 kbit/s and less services (TCH/H2.4))

The definition of a 3.6 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.7.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 36 information bits (data frames) every 10 ms. Two such blocks are dealt with together in the coding process, $\{d(0), d(1), \dots, d(71)\}$.

3.7.2 Block code

The block of 72 information bits is not encoded, but only increased with 4 tail bits equal to 0, at the end of the block.

Two such blocks forming a block of 152 bits $\{u(0), u(1), \dots, u(151)\}$ are dealt with together in the rest of the coding process.

$$u(k) = d1(k), \quad k = 0, 1, \dots, 75 \text{ (d1 = 1st information block)}$$

$$u(k+76) = d2(k), \quad k = 0, 1, \dots, 75 \text{ (d2 = 2nd information block)}$$

$$u(k) = 0, \quad k = 72, 73, 74, 75, 148, 149, 150, 151 \text{ (tail bits)}$$

3.7.3 Convolutional encoder

The convolutional encoding is done as specified for the TCH/F4.8 in subclause 3.4.3.

3.7.4 Interleaving

The interleaving is done as specified for the TCH/F9.6 in subclause 3.3.4.

3.7.5 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4. On bit stealing for signalling purposes by a FACCH, see subclause 4.3.5.

3.8 Data channel at full rate, 14.5 kbit/s radio interface rate (14.4 kbit/s services (TCH/F14.4))

The definition of a 14.5 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.8.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 290 information bits (data frames) every 20 ms.

3.8.2 Block code

The block of 290 information bits is not encoded, but only increased with 4 tail bits equal to 0 at the end of the block.

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 289$$

$$u(k) = 0 \quad \text{for } k = 290, 291, 292, 293 \text{ (tail bits)}$$

3.8.3 Convolutional encoder

This block of 294 bits $\{u(0), \dots, u(293)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

resulting in 588 coded bits $\{C(0), C(1), \dots, C(587)\}$ with

$$C(2k) = u(k) + u(k-3) + u(k-4)$$

$$C(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \text{ for } k = 0, 1, \dots, 293 ; u(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following 132 coded bits:

$$\{C(18*j+1), C(18*j+6), C(18*j+11), C(18*j+15) \text{ for } j = 0, 1, \dots, 31\} \text{ and the bits } C(577), C(582), C(584) \text{ and } C(587) \text{ are not transmitted.}$$

The result is a block of 456 coded bits, $\{c(0), c(1), \dots, c(455)\}$

3.8.4 Interleaving

The interleaving is done as specified for the TCH/F9.6 in section 3.3.4.

3.8.5 Mapping on a Burst

The mapping is done as specified for TCH/FS in section 3.1.4. On bitstealing by a FACCH, see section 4.2.5.

3.9 Adaptive multi rate speech channel at full rate (TCH/AFS)

This section describes the coding for the different frame formats used for TCH/AFS. The formats used are (in the order they are described):

| | |
|------------|--|
| SID_UPDATE | Used to convey comfort noise parameters during DTX |
| SID_FIRST | Marker to define end of speech, start of DTX |
| ONSET | Used to signal the Codec mode for the first speech frame after DTX |
| SPEECH | Speech frames |
| RATSCCH | Frames used to convey RATSCCH messages |

In this chapter, sub chapters 3.9.1 to 3.9.5 describe the channel coding for the different formats listed above.

Common to all the formats is that in-band information is conveyed, the coding for the in-band channel is described in the table below.

| Identifier (defined in GSM 05.09 [7]) | Received in-band data id(1), id(0) | Encoded in-band data for SID and RATSCCH frames ic(15),..., ic(0) | Encoded in-band data for speech frames ic(7),..., ic(0) |
|---|--|---|---|
| CODEC_MODE_1 | 00 | 0101001100001111 | 00000000 |
| CODEC_MODE_2 | 01 | 0011111010111000 | 10111010 |
| CODEC_MODE_3 | 10 | 1000100001100011 | 01011101 |
| CODEC_MODE_4 | 11 | 1110010111010100 | 11100111 |

3.9.1 SID_UPDATE

The speech encoder delivers 35 bits of comfort noise parameters. Also delivered is two in-band channels, id0(0,1) and id1(0,1), id0 corresponding to Mode Commands or Mode Requests and id1 to Mode Indication. The general coding is as: the two in-band data channels are coded to 16 bits each, a 14-bit CRC is added to the 35 CN bits which are then coded by a rate 1/4 RSC coder to 212 bits. Finally a 212 bit identification field is added thereby giving a total size of 456 bits. These 456 bits are then block interleaved in the same way as SACCH frames.

3.9.1.1 Coding of in-band data

The two in-band data fields, id0(0,1) and id1(0,1), are encoded, giving ic0(0..15) and ic1(0..15).

The ic0 and ic1 data is moved to the coded data c as:

| | |
|--------------------|--------------------------|
| $c(k) = ic0(k)$ | for $k = 0, 1, 2, 3$ |
| $c(k) = ic1(k-4)$ | for $k = 4, 5, 6, 7$ |
| $c(k) = ic0(k-4)$ | for $k = 8, 9, 10, 11$ |
| $c(k) = ic1(k-8)$ | for $k = 12, 13, 14, 15$ |
| $c(k) = ic0(k-8)$ | for $k = 16, 17, 18, 19$ |
| $c(k) = ic1(k-12)$ | for $k = 20, 21, 22, 23$ |
| $c(k) = ic0(k-12)$ | for $k = 24, 25, 26, 27$ |
| $c(k) = ic1(k-16)$ | for $k = 28, 29, 30, 31$ |

3.9.1.2 Parity and convolutional encoding for the comfort noise parameters

a) Parity bits:

A 14-bit CRC is used for error-detection. These 14 parity bits are generated by the cyclic generator polynomial: $g(D) = D^{14} + D^{13} + D^5 + D^3 + D^2 + 1$ from the 35 comfort noise parameter bits. The encoding of the cyclic code is performed in a systematic form, which means that, in GF(2), the polynomial:

$$d(0)D(48) + d(1)D(47) + \dots + d(34)D(14) + p(0)D^{(13)} + \dots + p(12)D + p(13)$$

where $p(0), p(1) \dots p(13)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to $1 + D + D^2 + D^3 + D^4 + D^5 + D^6 + D^7 + D^8 + D^9 + D^{10} + D^{11} + D^{12} + D^{13}$

The information and parity bits are merged:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 34$$

$$u(k) = p(k-35) \quad \text{for } k = 35, 36, \dots, 48$$

b) Convolutional encoder

The comfort noise parameters with parity bits ($u(0..48)$) are encoded with the 1/4 rate convolutional code defined by the polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 212 coded bits, $\{C(0) \dots C(211)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k) + r(k-2) + r(k-4)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k) \quad \text{for } k = 0, 1, \dots, 48; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k) + r(k-2) + r(k-4)$$

$$C(4k+2) = r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k+3) = r(k-1) + r(k-2) + r(k-3) + r(k-4) \quad \text{for } k = 49, 50, \dots, 52$$

This block of data is moved to the coded data (c) as:

$$c(8*k+32) = C(4*k)$$

$$c(8*k+33) = C(4*k+1)$$

$$c(8*k+34) = C(4*k+2)$$

$$c(8*k+35) = C(4*k+3) \quad \text{for } k = 0, 1, \dots, 52$$

3.9.1.3 Identification marker

The identification marker, IM(0..211), is constructed by repeating the following 9-bit sequence: { 0, 1, 0, 0, 1, 1, 1, 1, 0 } 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$\begin{aligned} c(8*k+36) &= \text{IM}(4*k) \\ c(8*k+37) &= \text{IM}(4*k+1) \\ c(8*k+38) &= \text{IM}(4*k+2) \\ c(8*k+39) &= \text{IM}(4*k+3) \quad \text{for } k = 0, 1, \dots, 52 \end{aligned}$$

3.9.1.4 Interleaving

The interleaving is done as specified for the SACCH in subclause 4.1.4.

3.9.1.5 Mapping on a Burst

The interleaving is done as specified for the SACCH in subclause 4.1.5 with the exception that hl(B) and hu(B) is set to "0".

3.9.2 SID_FIRST

This frame type contains no source data from the speech coder, what is transmitted is the in-band channel (signalling Mode Indication or Mode Command/Mode Request depending on the current frame number) and an identification marker.

3.9.2.1 Coding of in-band data

The in-band data, id(0,1), is encoded to ic(0..15) which is moved to the coded data c as:

$$\begin{aligned} c(k) &= ic(k) \quad \text{for } k = 0, 1, 2, 3 \\ c(k) &= ic(k-4) \quad \text{for } k = 8, 9, 10, 11 \\ c(k) &= ic(k-8) \quad \text{for } k = 16, 17, 18, 19 \\ c(k) &= ic(k-12) \quad \text{for } k = 24, 25, 26, 27 \end{aligned}$$

3.9.2.2 Identification marker

The identification marker, IM(0..211), is constructed by repeating the following 9-bit sequence: { 0, 1, 0, 0, 1, 1, 1, 1, 0 } 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$\begin{aligned} c(8*k+32) &= \text{IM}(4*k) \\ c(8*k+33) &= \text{IM}(4*k+1) \\ c(8*k+34) &= \text{IM}(4*k+2) \\ c(8*k+35) &= \text{IM}(4*k+3) \quad \text{for } k = 0, 1, \dots, 52 \end{aligned}$$

3.9.2.3 Interleaving

The interleaving is done as specified for the TCH/FS in subclause 3.1.3.

3.9.2.4 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4. The last 4 bursts shall not be transmitted unless the SID_FIRST frame is immediately followed by a speech frame.

3.9.3 ONSET

Onset frames are used to preset the interleaver buffer after a period of no speech activity in DTX mode. This frame type contains no source data from the speech coder, what is transmitted is the in-band channel signalling the Mode Indication for the speech frame following the onset marker.

3.9.3.1 Coding of in-band data

The in-band data, Mode Indication $id1(0,1)$, is encoded to $ic1(0..15)$. This sequence is then repeated 14 times more, and the last 12 bits are discarded ($15*16-12=228$) giving the sequence $ic1(0..227)$.

This sequence is then moved to c as:

$$\begin{aligned} c(8*k+4) &= ic1(4*k) \\ c(8*k+5) &= ic1(4*k+1) \\ c(8*k+6) &= ic1(4*k+2) \\ c(8*k+7) &= ic1(4*k+3) \quad \text{for } k = 0, 1, \dots, 56 \end{aligned}$$

3.9.3.2 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$\begin{aligned} i(B,j) &= c(n,k), \quad \text{for } k = 4,5,6,7, 12,13,14,15,20,21,22,23 \dots,455 \\ n &= 0,1,\dots,N,N+1,\dots \\ B &= B_0 + 4n + (k \bmod 8) - 4 \\ j &= 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4) \end{aligned}$$

See table 1. The result of the interleaving is a distribution of the defined 228 bits of a given data block of size 456 bits, $n = N$, over 4 blocks using the odd numbered bits. The even numbered bits of these 4 blocks will be filled by the speech frame for which this frame is the ONSET.

3.9.3.3 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \quad \text{and} \quad e(B,59+j) = i(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$e(B,57) = hl(B)$$

The bit labelled $hl(B)$ on burst number B is a flag used for indication of control channel signalling. For each ONSET block not stolen for signalling purposes:

$$hl(B) = 0 \quad \text{for the 4 bursts} \quad (\text{indicating status of odd numbered bits})$$

For the use of $hl(B)$ when an ONSET is stolen for signalling purposes see subclause 4.2.5.

3.9.4 SPEECH

The speech coder delivers to the channel encoder a sequence of blocks of data. One block of data corresponds to one speech frame and the block length is different in each of the eight channel codec modes. Adjoining each block of data is information of the channel codec mode to use when encoding the block. Also delivered is the in-band data $id(0,1)$ representing Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.9.4.1 Coding of the in-band data

The two input in-band bits ($id(0,1)$) are coded to eight coded in-band bits ($ic(0..7)$).

The encoded in-band bits are moved to the coded bits, c , as

$$c(k) = ic(k) \quad \text{for } k = 0, 1, \dots, 7.$$

3.9.4.2 Ordering according to subjective importance

The bits delivered by the speech encoder, $\{s(1),s(2),\dots,s(K_s)\}$, are rearranged according to subjective importance before channel coding. Tables 7 to 16 define the correct rearrangement for the speech codec modes 12.2 kbit/s, 10.2 kbit/s, 7.95 kbit/s, 7.40 kbit/s, 6.70 kbit/s, 5.90 kbit/s, 5.15 kbit/s and 4.75 kbit/s, respectively. In the tables speech codec parameters are numbered in the order they are delivered by the corresponding speech encoder according to GSM 06.90 [11] and the rearranged bits are labelled $\{d(0),d(1),\dots,d(K_d-1)\}$, defined in the order of decreasing importance. Index K_d refers to the number of bits delivered by the speech encoder, see below:

| Codec mode | Number of speech bits delivered per block (K_d) |
|-------------|---|
| TCH/AFS12.2 | 244 |
| TCH/AFS10.2 | 204 |
| TCH/AFS7.95 | 159 |
| TCH/AFS7.4 | 148 |
| TCH/AFS6.7 | 134 |
| TCH/AFS5.9 | 118 |
| TCH/AFS5.15 | 103 |
| TCH/AFS4.75 | 95 |

The ordering algorithm is in pseudo code as:

$$\text{for } j = 0 \text{ to } K_d-1 \quad d(j) := s(\text{table}(j)+1); \quad \text{where table}(j) \text{ is read line by line left to right}$$

The rearranged bits are further divided into two different classes to perform unequal error protection for different bits according to subjective importance.

The protection classes are:

- 1a - Data protected with the CRC and the convolution code.
 - 1b - Data protected with the convolution code.
- No unprotected bits are used.

The number of class 1 (sum of class 1a and 1b), class 1a and class 1b bits for each codec mode is shown below:

| Codec Mode | Number of speech bits delivered per block | Number of class 1 bits per block | Number of class 1a bits per block | Number of class 1b bits per block |
|-------------|---|----------------------------------|-----------------------------------|-----------------------------------|
| TCH/AFS12.2 | 244 | 244 | 81 | 163 |
| TCH/AFS10.2 | 204 | 204 | 65 | 139 |
| TCH/AFS7.95 | 159 | 159 | 75 | 84 |
| TCH/AFS7.4 | 148 | 148 | 61 | 87 |
| TCH/AFS6.7 | 134 | 134 | 55 | 79 |
| TCH/AFS5.9 | 118 | 118 | 55 | 63 |
| TCH/AFS5.15 | 103 | 103 | 49 | 54 |
| TCH/AFS4.75 | 95 | 95 | 39 | 56 |

3.9.4.3 Parity for speech frames

The basic parameters for each codec mode for the first encoding step are shown below:

| Codec mode | Speech encoded bits (K_d) | CRC protected bits (K_{d1a}) | Number of bits after first encoding step ($K_u = K_d + 6$) |
|-------------|-------------------------------|----------------------------------|--|
| TCH/AFS12.2 | 244 | 81 | 250 |
| TCH/AFS10.2 | 204 | 65 | 210 |
| TCH/AFS7.95 | 159 | 75 | 165 |
| TCH/AFS7.4 | 148 | 61 | 154 |
| TCH/AFS6.7 | 134 | 55 | 140 |
| TCH/AFS5.9 | 118 | 55 | 124 |
| TCH/AFS5.15 | 103 | 49 | 109 |
| TCH/AFS4.75 | 95 | 39 | 101 |

A 6-bit CRC is used for error-detection. These 6 parity bits are generated by the cyclic generator polynomial: $g(D) = D^6 + D^5 + D^3 + D^2 + D^1 + 1$ from the first K_{d1a} bits of class 1, where K_{d1a} refers to number of bits in protection class 1a as shown above for each codec mode. The encoding of the cyclic code is performed in a systematic form, which means that, in GF(2), the polynomial:

$$d(0)D^{(K_{d1a}+5)} + d(1)D^{(K_{d1a}+4)} + \dots + d(K_{d1a}-1)D^{(6)} + p(0)D^{(5)} + \dots + p(4)D + p(5)$$

where $p(0), p(1) \dots p(5)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to:

$$1 + D + D^2 + D^3 + D^4 + D^5.$$

The information and parity bits are merged:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, K_{d1a}-1 \\ u(k) &= p(k-K_{d1a}) && \text{for } k = K_{d1a}, K_{d1a}+1, \dots, K_{d1a}+5 \\ u(k) &= d(k-6) && \text{for } k = K_{d1a}+6, K_{d1a}+7, \dots, K_u-1 \end{aligned}$$

Thus, after the first encoding step $u(k)$ will be defined by the following contents for each codec mode:

TCH/AFS12.2:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 80 \\ u(k) &= p(k-81) && \text{for } k = 81, 82, \dots, 86 \\ u(k) &= d(k-6) && \text{for } k = 87, 88, \dots, 249 \end{aligned}$$

TCH/AFS10.2:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 64 \\ u(k) &= p(k-65) && \text{for } k = 65, 66, \dots, 70 \\ u(k) &= d(k-6) && \text{for } k = 71, 72, \dots, 209 \end{aligned}$$

TCH/AFS7.95:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 74 \\ u(k) &= p(k-75) && \text{for } k = 75, 76, \dots, 80 \\ u(k) &= d(k-6) && \text{for } k = 81, 82, \dots, 164 \end{aligned}$$

TCH/AFS7.4:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 60 \\ u(k) &= p(k-61) && \text{for } k = 61, 62, \dots, 66 \\ u(k) &= d(k-6) && \text{for } k = 67, 68, \dots, 153 \end{aligned}$$

TCH/AFS6.7:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 54 \\ u(k) &= p(k-55) && \text{for } k = 55, 56, \dots, 60 \\ u(k) &= d(k-6) && \text{for } k = 61, 62, \dots, 139 \end{aligned}$$

TCH/AFS5.9:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 54 \\ u(k) &= p(k-55) && \text{for } k = 55, 56, \dots, 60 \\ u(k) &= d(k-6) && \text{for } k = 61, 62, \dots, 123 \end{aligned}$$

TCH/AFS5.15:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 48 \\ u(k) &= p(k-49) && \text{for } k = 49, 50, \dots, 54 \\ u(k) &= d(k-6) && \text{for } k = 55, 56, \dots, 108 \end{aligned}$$

TCH/AFS4.75:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 38 \\ u(k) &= p(k-39) && \text{for } k = 39, 40, \dots, 44 \\ u(k) &= d(k-6) && \text{for } k = 45, 46, \dots, 100 \end{aligned}$$

3.9.4.4 Convolutional encoder

The bits from the first encoding step ($u(k)$) are encoded with the recursive systematic convolutional codes as summarised below. The number of output bits after puncturing is 448 for all codec modes.

| Codec mode | Rate | Number of input bits to conv. coder | Number of output bits from conv. coder | Number of punctured bits |
|-------------|------|-------------------------------------|--|--------------------------|
| TCH/AFS12.2 | 1/2 | 250 | 508 | 60 |
| TCH/AFS10.2 | 1/3 | 210 | 642 | 194 |
| TCH/AFS7.95 | 1/3 | 165 | 513 | 65 |
| TCH/AFS7.4 | 1/3 | 154 | 474 | 26 |
| TCH/AFS6.7 | 1/4 | 140 | 576 | 128 |
| TCH/AFS5.9 | 1/4 | 124 | 520 | 72 |
| TCH/AFS5.15 | 1/5 | 109 | 565 | 117 |
| TCH/AFS4.75 | 1/5 | 101 | 535 | 87 |

Below the coding for each codec mode is specified in detail.

TCH/AFS12.2:

The block of 250 bits $\{u(0)\dots u(249)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0/G_0 = 1$$

$$G_1/G_0 = 1 + D + D^3 + D^4 / 1 + D^3 + D^4$$

resulting in 508 coded bits, $\{C(0)\dots C(507)\}$ defined by:

$$r(k) = u(k) + r(k-3) + r(k-4)$$

$$C(2k) = u(k)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 0, 1, \dots, 249; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(2k) = r(k-3) + r(k-4)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 250, 251, \dots, 253$$

The code is punctured in such a way that the following 60 coded bits:

C(321), C(325), C(329), C(333), C(337), C(341), C(345), C(349), C(353), C(357), C(361), C(363), C(365), C(369), C(373), C(377), C(379), C(381), C(385), C(389), C(393), C(395), C(397), C(401), C(405), C(409), C(411), C(413), C(417), C(421), C(425), C(427), C(429), C(433), C(437), C(441), C(443), C(445), C(449), C(453), C(457), C(459), C(461), C(465), C(469), C(473), C(475), C(477), C(481), C(485), C(489), C(491), C(493), C(495), C(497), C(499), C(501), C(503), C(505) and C(507)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the in-band bits in c as

$$c(k+8) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS10.2:

The block of 210 bits {u(0)... u(209)} is encoded with the 1/3 rate convolutional code defined by the following polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

resulting in 642 coded bits, {C(0)... C(641)} defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(3k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = u(k) \quad \text{for } k = 0, 1, \dots, 209$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(3k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 210, 211, \dots, 213$$

The code is punctured in such a way that the following 194 bits:

C(1), C(4), C(7), C(10), C(16), C(19), C(22), C(28), C(31), C(34), C(40), C(43), C(46), C(52), C(55), C(58), C(64), C(67), C(70), C(76), C(79), C(82), C(88), C(91), C(94), C(100), C(103), C(106), C(112), C(115), C(118), C(124), C(127), C(130), C(136), C(139), C(142), C(148), C(151), C(154), C(160), C(163), C(166), C(172), C(175), C(178), C(184), C(187), C(190), C(196), C(199), C(202), C(208), C(211), C(214), C(220), C(223), C(226), C(232), C(235), C(238), C(244), C(247), C(250), C(256), C(259), C(262), C(268), C(271), C(274), C(280), C(283), C(286), C(292), C(295), C(298), C(304), C(307), C(310), C(316), C(319), C(322), C(325), C(328), C(331), C(334), C(337), C(340), C(343), C(346), C(349), C(352), C(355), C(358), C(361), C(364), C(367), C(370), C(373), C(376), C(379), C(382), C(385), C(388), C(391), C(394), C(397), C(400),

C(403), C(406), C(409), C(412), C(415), C(418), C(421), C(424), C(427), C(430), C(433), C(436), C(439), C(442), C(445), C(448), C(451), C(454), C(457), C(460), C(463), C(466), C(469), C(472), C(475), C(478), C(481), C(484), C(487), C(490), C(493), C(496), C(499), C(502), C(505), C(508), C(511), C(514), C(517), C(520), C(523), C(526), C(529), C(532), C(535), C(538), C(541), C(544), C(547), C(550), C(553), C(556), C(559), C(562), C(565), C(568), C(571), C(574), C(577), C(580), C(583), C(586), C(589), C(592), C(595), C(598), C(601), C(604), C(607), C(609), C(610), C(613), C(616), C(619), C(621), C(622), C(625), C(627), C(628), C(631), C(633), C(634), C(636), C(637), C(639) and C(640)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the in-band bits in c as:

$$c(k+8) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS7.95:

The block of 165 bits {u(0)... u(164)} is encoded with the 1/3 rate convolutional code defined by the following polynomials:

$$G4/G4 = 1$$

$$G5/G4 = 1 + D + D^4 + D^6 / 1 + D^2 + D^3 + D^5 + D^6$$

$$G6/G4 = 1 + D + D^2 + D^3 + D^4 + D^6 / 1 + D^2 + D^3 + D^5 + D^6$$

resulting in 513 coded bits, {C(0)... C(512)} defined by:

$$r(k) = u(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(3k) = u(k)$$

$$C(3k+1) = r(k)+r(k-1)+r(k-4)+r(k-6)$$

$$C(3k+2) = r(k)+r(k-1) + r(k-2)+r(k-3)+r(k-4)+r(k-6) \quad \text{for } k = 0, 1, \dots, 164; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(3k) = r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(3k+1) = r(k)+r(k-1)+r(k-4)+r(k-6)$$

$$C(3k+2) = r(k)+r(k-1) + r(k-2)+r(k-3)+r(k-4)+r(k-6) \quad \text{for } k = 165, 166, \dots, 170$$

The code is punctured in such a way that the following 65 coded bits:

C(1), C(2), C(4), C(5), C(8), C(22), C(70), C(118), C(166), C(214), C(262), C(310), C(317), C(319), C(325), C(332), C(334), C(341), C(343), C(349), C(356), C(358), C(365), C(367), C(373), C(380), C(382), C(385), C(389), C(391), C(397), C(404), C(406), C(409), C(413), C(415), C(421), C(428), C(430), C(433), C(437), C(439), C(445), C(452), C(454), C(457), C(461), C(463), C(469), C(476), C(478), C(481), C(485), C(487), C(490), C(493), C(500), C(502), C(503), C(505), C(506), C(508), C(509), C(511) and C(512)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the in-band bits in c as

$$c(k+8) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS7.4:

The block of 154 bits {u(0)... u(153)} is encoded with the 1/3 rate convolutional code defined by the following polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

resulting in 474 coded bits, $\{C(0)... C(473)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(3k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = u(k) \quad \text{for } k = 0, 1, \dots, 153$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(3k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 154, 155, \dots, 157$$

The code is punctured in such a way that the following 26 bits:

$C(0), C(355), C(361), C(367), C(373), C(379), C(385), C(391), C(397), C(403), C(409), C(415), C(421), C(427), C(433), C(439), C(445), C(451), C(457), C(460), C(463), C(466), C(468), C(469), C(471)$ and $C(472)$

are not transmitted. The result is a block of 448 coded and punctured bits, $P(0)...P(447)$ which are appended to the in-band bits in c as:

$$c(k+8) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS6.7:

The block of 140 bits $\{u(0)... u(139)\}$ is encoded with the 1/4 rate convolutional code defined by the following polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 576 coded bits, $\{C(0)... C(575)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k) \quad \text{for } k = 0, 1, \dots, 139; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4)$$

$$C(4k+3) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 140, 141, \dots, 143$$

The code is punctured in such a way that the following 128 coded bits:

C(1), C(3), C(7), C(11), C(15), C(27), C(39), C(55), C(67), C(79), C(95), C(107), C(119), C(135), C(147), C(159), C(175), C(187), C(199), C(215), C(227), C(239), C(255), C(267), C(279), C(287), C(291), C(295), C(299), C(303), C(307), C(311), C(315), C(319), C(323), C(327), C(331), C(335), C(339), C(343), C(347), C(351), C(355), C(359), C(363), C(367), C(369), C(371), C(375), C(377), C(379), C(383), C(385), C(387), C(391), C(393), C(395), C(399), C(401), C(403), C(407), C(409), C(411), C(415), C(417), C(419), C(423), C(425), C(427), C(431), C(433), C(435), C(439), C(441), C(443), C(447), C(449), C(451), C(455), C(457), C(459), C(463), C(465), C(467), C(471), C(473), C(475), C(479), C(481), C(483), C(487), C(489), C(491), C(495), C(497), C(499), C(503), C(505), C(507), C(511), C(513), C(515), C(519), C(521), C(523), C(527), C(529), C(531), C(535), C(537), C(539), C(543), C(545), C(547), C(549), C(551), C(553), C(555), C(557), C(559), C(561), C(563), C(565), C(567), C(569), C(571), C(573) and C(575)

are not transmitted. The result is a block of 448 coded bits, P(0)...P(447) which are appended to the in-band bits in c as

$$c(k+8) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS5.9:

The block of 124 bits {u(0)... u(123)} is encoded with the 1/4 rate convolutional code defined by the following polynomials:

$$G4/G6 = 1 + D^2 + D^3 + D^5 + D^6 / 1 + D + D^2 + D^3 + D^4 + D^6$$

$$G5/G6 = 1 + D + D^4 + D^6 / 1 + D + D^2 + D^3 + D^4 + D^6$$

$$G6/G6 = 1$$

$$G6/G6 = 1$$

resulting in 520 coded bits, {C(0)... C(519)} defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4) + r(k-6)$$

$$C(4k) = r(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(4k+1) = r(k) + r(k-1) + r(k-4) + r(k-6)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k)$$

$$\text{for } k = 0, 1, \dots, 123; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k)+r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(4k+1) = r(k)+r(k-1)+r(k-4)+r(k-6)$$

$$C(4k+2) = r(k-1)+r(k-2)+ r(k-3)+r(k-4)+r(k-6)$$

$$C(4k+3) = r(k-1)+r(k-2)+ r(k-3)+r(k-4)+r(k-6)$$

$$\text{for } k = 124, 125, \dots, 129$$

The code is punctured in such a way that the following 72 coded bits:

C(0), C(1), C(3), C(5), C(7), C(11), C(15), C(31), C(47), C(63), C(79), C(95), C(111), C(127), C(143), C(159), C(175), C(191), C(207), C(223), C(239), C(255), C(271), C(287), C(303), C(319), C(327), C(331), C(335), C(343), C(347), C(351), C(359), C(363), C(367), C(375), C(379), C(383), C(391), C(395), C(399),

C(407), C(411), C(415), C(423), C(427), C(431), C(439), C(443), C(447), C(455), C(459), C(463), C(467), C(471), C(475), C(479), C(483), C(487), C(491), C(495), C(499), C(503), C(507), C(509), C(511), C(512), C(513), C(515), C(516), C(517) and C(519)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the in-band bits in c as

$$c(8+k) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS5.15:

The block of 109 bits {u(0)... u(108)} is encoded with the 1/5 rate convolutional code defined by the following polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 565 coded bits, {C(0)... C(564)} defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(5k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(5k+1) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(5k+2) = r(k)+r(k-2)+r(k-4)$$

$$C(5k+3) = u(k)$$

$$C(5k+4) = u(k)$$

$$\text{for } k = 0, 1, \dots, 108; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(5k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(5k+1) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(5k+2) = r(k)+r(k-2)+r(k-4)$$

$$C(5k+3) = r(k-1)+r(k-2)+r(k-3)+r(k-4)$$

$$C(5k+4) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 109, 110, \dots, 112$$

The code is punctured in such a way that the following 117 coded bits:

C(0), C(4), C(5), C(9), C(10), C(14), C(15), C(20), C(25), C(30), C(35), C(40), C(50), C(60), C(70), C(80), C(90), C(100), C(110), C(120), C(130), C(140), C(150), C(160), C(170), C(180), C(190), C(200), C(210), C(220), C(230), C(240), C(250), C(260), C(270), C(280), C(290), C(300), C(310), C(315), C(320), C(325), C(330), C(334), C(335), C(340), C(344), C(345), C(350), C(354), C(355), C(360), C(364), C(365), C(370), C(374), C(375), C(380), C(384), C(385), C(390), C(394), C(395), C(400), C(404), C(405), C(410), C(414), C(415), C(420), C(424), C(425), C(430), C(434), C(435), C(440), C(444), C(445), C(450), C(454), C(455), C(460), C(464), C(465), C(470), C(474), C(475), C(480), C(484), C(485), C(490), C(494), C(495), C(500), C(504), C(505), C(510), C(514), C(515), C(520), C(524), C(525), C(529), C(530), C(534), C(535), C(539), C(540), C(544), C(545), C(549), C(550), C(554), C(555), C(559), C(560) and C(564)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the in-band bits in c as

$$c(8+k) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

TCH/AFS4.75:

The block of 101 bits {u(0)... u(100)} is encoded with the 1/5 rate convolutional code defined by the following polynomials:

$$G4/G6 = 1 + D^2 + D^3 + D^5 + D^6 / 1 + D + D^2 + D^3 + D^4 + D^6$$

$$G4/G6 = 1 + D^2 + D^3 + D^5 + D^6 / 1 + D + D^2 + D^3 + D^4 + D^6$$

$$G5/G6 = 1 + D + D^4 + D^6 / 1 + D + D^2 + D^3 + D^4 + D^6$$

$$G6/G6 = 1$$

$$G6/G6 = 1$$

resulting in 535 coded bits, {C(0)... C(534)} defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4) + r(k-6)$$

$$C(5k) = r(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(5k+1) = r(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(5k+2) = r(k) + r(k-1) + r(k-4) + r(k-6)$$

$$C(5k+3) = u(k)$$

$$C(5k+4) = u(k)$$

$$\text{for } k = 0, 1, \dots, 100; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(5k) = r(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(5k+1) = r(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(5k+2) = r(k) + r(k-1) + r(k-4) + r(k-6)$$

$$C(5k+3) = r(k-1) + r(k-2) + r(k-3) + r(k-4) + r(k-6)$$

$$C(5k+4) = r(k-1) + r(k-2) + r(k-3) + r(k-4) + r(k-6)$$

$$\text{for } k = 101, 102, \dots, 106$$

The code is punctured in such a way that the following 87 coded bits:

C(0), C(1), C(2), C(4), C(5), C(7), C(9), C(15), C(25), C(35), C(45), C(55), C(65), C(75), C(85), C(95),
C(105), C(115), C(125), C(135), C(145), C(155), C(165), C(175), C(185), C(195), C(205), C(215), C(225),
C(235), C(245), C(255), C(265), C(275), C(285), C(295), C(305), C(315), C(325), C(335), C(345), C(355),
C(365), C(375), C(385), C(395), C(400), C(405), C(410), C(415), C(420), C(425), C(430), C(435), C(440),
C(445), C(450), C(455), C(459), C(460), C(465), C(470), C(475), C(479), C(480), C(485), C(490), C(495),
C(499), C(500), C(505), C(509), C(510), C(515), C(517), C(519), C(520), C(522), C(524), C(525), C(526),
C(527), C(529), C(530), C(531), C(532) and C(534)

are not transmitted. The result is a block of 448 coded and punctured bits, P(0)...P(447) which are appended to the inband bits in c as

$$c(8+k) = P(k) \quad \text{for } k = 0, 1, \dots, 447.$$

3.9.4.5 Interleaving

The interleaving is done as specified for the TCH/FS in subclause 3.1.3.

3.9.4.6 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4.

3.9.5 RATSCCH

The RATSCCH message consists of 35 bits. Also delivered are two in-band channels, id0(0,1) and id1(0,1), id0 corresponding to Mode Commands or Mode Requests and id1 to Mode Indication. The general coding is as: the two in-band data channels are coded to 16 bits each, a 14-bit CRC is added to the 35 RATSCCH bits which are then coded by a rate 1/4 RSC coder to 212 bits. Finally a 212 bit identification field is added thereby giving a total size of 456 bits. These 456 bits are then block interleaved in the same way as a normal speech frame.

3.9.5.1 Coding of in-band data

The two n-band data fields, id0(0,1) and id1(0,1), are encoded, giving ic0(0..15) and ic1(0..15). These bits are moved to the coded bits c as:

$$c(k) = ic1(k) \text{ for } k = 0, 1, \dots, 15$$

$$c(k+228) = ic0(k) \text{ for } k = 0, 1, \dots, 15$$

3.9.5.2 Parity and convolutional encoding for the RATSCCH message

a) Parity bits:

A 14-bit CRC is used for error-detection. These 14 parity bits are generated by the cyclic generator polynomial: $g(D) = D^{14} + D^{13} + D^5 + D^3 + D^2 + 1$ from the 35 comfort noise parameter bits. The encoding of the cyclic code is performed in a systematic form, which means that, in GF(2), the polynomial:

$$d(0)D^{48} + d(1)D^{47} + \dots + d(34)D^{14} + p(0)D^{13} + \dots + p(12)D + p(13)$$

$$\text{where } p(0), p(1) \dots p(13) \text{ are the parity bits, when divided by } g(D), \text{ yields a remainder equal to } 1 + D + D^2 + D^3 + D^4 + D^5 + D^6 + D^7 + D^8 + D^9 + D^{10} + D^{11} + D^{12} + D^{13}$$

The information and parity bits are merged:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 34$$

$$u(k) = p(k-35) \quad \text{for } k = 35, 36, \dots, 48$$

b) Convolutional encoder

The comfort noise parameters with parity and tail bits (u(0..48)) are encoded with the 1/4 rate convolutional code defined by the polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 212 coded bits, {C(0)... C(211)} defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k) \quad \text{for } k = 0, 1, \dots, 48; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4)$$

$$C(4k+3) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 49, 50, \dots, 52$$

This block of data is moved to the coded data (c) as:

$$c(k+244) = C(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.9.5.3 Identification marker

The identification marker, IM(0..211), is constructed by repeating the following 11-bit sequence: { 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1 } 20 times and then discarding the last 8 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = IM(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.9.5.4 Interleaving

The interleaving is done as specified for the TCH/FS in subclause 3.1.3.

3.9.5.5 Mapping on a Burst

The mapping is done as specified for the TCH/FS in subclause 3.1.4.

3.10 Adaptive multi rate speech channel at half rate (TCH/AHS)

This section describes the coding for the different frame formats used for TCH/AHS. The formats used are (in the order they are described):

| | |
|----------------|--|
| SID_UPDATE | Used to convey comfort noise parameters during DTX |
| SID_UPDATE_INH | Used to inhibit the second part of a SID_UPDATE frame if there is a speech onset |
| SID_FIRST_P1 | First part of marker to define end of speech, start of DTX |
| SID_FIRST_P2 | Second part of marker to define end of speech, start of DTX |
| SID_FIRST_INH | Used to inhibit the second part of a SID_FIRST_P1 frame if there is a speech onset |
| ONSET | Used to signal the Codec mode for the first speech frame after DTX |
| SPEECH | Speech frames |
| RATSCCH_MARKER | Marker to identify RATSCCH frames |
| RATSCCH_DATA | Frame that conveys the actual RATSCCH message |

In this chapter, sub chapters 3.10.1 to 3.10.9 describe the channel coding for the different formats listed above.

Common to all the formats is that in-band information is conveyed, the coding for the in-band channel is described in the table below:

| Identifier (defined in GSM 05.09 [7]) | Received in-band data id(1), id(0) | Encoded in-band data for SID and RATSCCH frames ic(15),..., ic(0) | Encoded in-band data for speech framesic(3),..., ic(0) |
|---|---------------------------------------|---|--|
| CODEC_MODE_1 | 00 | 0101001100001111 | 0000 |
| CODEC_MODE_2 | 01 | 0011111010111000 | 1001 |
| CODEC_MODE_3 | 10 | 1000100001100011 | 0111 |
| CODEC_MODE_4 | 11 | 1110010111010100 | 1110 |

3.10.1 SID_UPDATE

The speech encoder delivers 35 bits of comfort noise parameters. Also delivered is two in-band channels, id0(0,1) and id1(0,1), id0 corresponding to Mode Commands/Mode Requests and id1 to Mode Indication. The general coding is as: the two in-band data channels are coded to 16 bits each, a 14-bit CRC is added to the 35 CN bits which are then coded by a rate 1/4 RSC coder to 212 bits. Finally a 212 bit identification field is added thereby giving a total size of 456 bits. These 456 bits are block interleaved over 4 bursts.

3.10.1.1 Coding of in-band data

The two in-band data fields, id0(0,1) and id1(0,1), are encoded, giving ic0(0..15) and ic1(0..15).

The ic0 and ic1 data is moved to the coded data c as:

$$c(k) = ic1(k) \quad \text{for } k = 0, 1, \dots, 15$$

$$c(k) = ic0(k-228) \quad \text{for } k = 228, 229, \dots, 243$$

3.10.1.2 Parity and convolutional encoding for the comfort noise parameters

a) Parity bits:

A 14-bit CRC is used for error-detection. These 14 parity bits are generated by the cyclic generator polynomial: $g(D) = D^{14} + D^{13} + D^5 + D^3 + D^2 + 1$ from the 35 comfort noise parameter bits. The encoding of the cyclic code is performed in a systematic form, which means that, in GF(2), the polynomial:

$$d(0)D(48) + d(1)D(47) + \dots + d(34)D(14) + p(0)D^{13} + \dots + p(12)D + p(13)$$

where $p(0), p(1) \dots p(13)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to $1 + D + D^2 + D^3 + D^4 + D^5 + D^6 + D^7 + D^8 + D^9 + D^{10} + D^{11} + D^{12} + D^{13}$

The information and parity bits are merged:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 34$$

$$u(k) = p(k-35) \quad \text{for } k = 35, 36, \dots, 48$$

b) Convolutional encoder

The comfort noise parameters with parity bits ($u(0..48)$) are encoded with the 1/4 rate convolutional code defined by the polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 212 coded bits, $\{C(0) \dots C(211)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k) \quad \text{for } k = 0, 1, \dots, 48; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4)$$

$$C(4k+3) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 49, 50, \dots, 52$$

This block of data is moved to the coded data (c) as:

$$c(k+244) = C(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.1.3 Identification marker

The identification marker, IM(0..211), is constructed by repeating the following 9-bit sequence:

{ 1, 0, 1, 1, 0, 0, 0, 0, 1 } 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = \text{IM}(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.1.4 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k) \text{ for } k = 0,1,\dots,227$$

$$n = 0,1,\dots,N,N+1,\dots$$

$$B = B_0 + 2n + b$$

$$i(B,j) = c(n,k+228) \text{ for } k = 0,1,\dots,227$$

$$n = 0,1,\dots,N,N+1,\dots$$

$$B = B_0 + 2n + ((b + 2) \bmod 4)$$

The values of b and j in dependence of k are given by table 4.

The result of the interleaving is a distribution of the 456 bits of a given data block, $n = N$, over 4 blocks using all bits for each block. The block of coded data is interleaved "block rectangular" where a new data block starts every 4th block and is distributed over 4 blocks.

3.10.1.5 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \text{ and } e(B,59+j) = i(B,57+j) \text{ for } j = 0,1,\dots,56$$

and

$$e(B,57) = \text{hl}(B) \text{ and } e(B,58) = \text{hu}(B)$$

The two bits, labelled $hl(B)$ and $hu(B)$ on burst number B are flags used for indication of control channel signalling. For each block not stolen for FACCH signalling purposes:

$$hu(B) = 0 \quad \text{for all 4 bursts}$$

$$hl(B) = 0 \quad \text{for all 4 bursts}$$

For the use of $hl(B)$ and $hu(B)$ when frame is stolen for signalling purposes, see subclause 4.3.5.

3.10.2 SID_UPDATE_INH

This special frame is used when the first 2 burst of a SID_UPDATE frame have been transmitted but the second two bursts cannot be transmitted due to a speech frame. The general coding is as: the in-band data (Note that this must be the same Mode Indication bits as $id1(0,1)$ for the SID_UPDATE frame that is being inhibited) is encoded, a marker that is the opposite of the SID_UPDATE marker is appended and the data is interleaved in such a way that the odd bits of two bursts are filled.

3.10.2.1 Coding of in-band data

The in-band data, Mode Indication $id1(0,1)$, is encoded to $ic1(0..15)$ which is moved to the coded data c as:

$$c(k) = ic1(k) \quad \text{for } k = 0, 1, \dots, 15$$

3.10.2.2 Identification marker

The identification marker, $IM(0..211)$, is constructed by repeating the following 9-bit sequence: $\{ 0, 1, 0, 0, 1, 1, 1, 1, 0 \}$ 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = IM(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.2.3 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k) \quad \text{for } \begin{aligned} k &= 1, 3, 5, 7, \dots, 227 \\ n &= 0, 1, \dots, N, N+1, \dots \\ B &= B_0 + 2n + b - 2 \end{aligned}$$

The values of b and j in dependence of k are given by table 4.

The result of the interleaving is a distribution of 114 of the reordered 228 bits of a given data block, $n = N$, over 2 blocks using the odd numbered bits. The even numbered bits of these 2 blocks will be filled by the speech frame that following immediately after this frame.

3.10.2.4 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \text{ and } e(B,59+j) = i(B,57+j) \text{ for } j = 0, 1, \dots, 56$$

and

$$e(B,57) = hl(B)$$

The bit labelled $hl(B)$ on burst number B is a flag used for indication of control channel signalling. For each SID_FIRST_INH block not stolen for signalling purposes:

$$hl(B) = 0 \quad \text{for the 2 bursts (indicating status of the odd numbered bits)}$$

For the use of $hl(B)$ when a SID_UPDATE_INH is stolen for signalling purposes, see subclause 4.3.5.

3.10.3 SID_FIRST_P1

This frame type contains no source data from the speech coder. What is generated is the in-band channel and an identification marker. The in-band data $id(0,1)$ represents Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.10.3.1 Coding of in-band data

The in-band data, $id(0,1)$, is encoded to $ic(0..15)$ which is moved to the coded data c as:

$$c(k) = ic(k) \quad \text{for } k = 0, 1, \dots, 15$$

3.10.3.2 Identification marker

The identification marker, $IM(0..211)$, is constructed by repeating the following 9-bit sequence:

{ 0, 1, 0, 0, 1, 1, 1, 1, 0 } 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = IM(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.3.3 Interleaving

The interleaving is done as specified for the TCH/HS in subclause 3.2.3.

3.10.3.4 Mapping on a Burst

The mapping is done as specified for the TCH/HS in subclause 3.2.4.

3.10.4 SID_FIRST_P2

This frame type contains no source data from the speech coder. What is generated is the in-band channel and, derived from that, an identification marker. The in-band data $id(0,1)$ represents Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.10.4.1 Coding of in-band data

The in-band data, $id(0,1)$, is encoded to $ic(0..15)$. This sequence is then repeated 7 times more, and the last 14 bits are discarded ($8*16-14=114$) giving the sequence $ic(0..113)$.

This sequence is then moved to c as:

$$c(2*k) = ic(k) \quad \text{for } k = 0, 1, \dots, 113$$

3.10.4.2 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k) \quad \text{for} \quad \begin{aligned} k &= 0, 2, 4, 6, \dots, 226 \\ n &= 0, 1, \dots, N, N+1, \dots \\ B &= B_0 + 2n + b \end{aligned}$$

The values of b and j in dependence of k are given by table 4.

The result of the interleaving is a distribution of 114 of the reordered 228 bits of a given data block, $n = N$, over 2 blocks using the even numbered bits. The odd numbered bits of these 2 blocks have already been filled by the SID_FIRST_P1 frame.

3.10.4.3 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \text{ and } e(B,59+j) = i(B,57+j) \text{ for } j = 0,1,\dots,56$$

and

$$e(B,58) = hu(B)$$

The bit labelled $hu(B)$ on burst number B is a flag used for indication of control channel signalling. For each SID_FIRST_P2 block not stolen for signalling purposes:

$$hu(B) = 0 \text{ for the 2 bursts (indicating status of the even numbered bits)}$$

For the use of $hu(B)$ when a SID_FIRST_P2 is stolen for signalling purposes, see subclause 4.3.5.

3.10.5 SID_FIRST_INH

This special frame is used when the first 2 burst of a SID_FIRST_P1 frame have been transmitted but the second two bursts cannot be transmitted due to a SPEECH frame. The general coding is as: the in-band data (Note that this must be the same data as for the SID_FIRST_P1 frame that is being inhibited) is encoded, a marker that is the opposite of the SID_FIRST_P1 marker is appended and the data is interleaved in such a way that the odd bits of two bursts are filled.

3.10.5.1 Coding of in-band data

The coding of the in-band data is done as specified for the SID_FIRST_P1 frame in subclause 3.10.3.1.

3.10.5.2 Identification marker

The identification marker, $IM(0..211)$, is constructed by repeating the following 9-bit sequence:

{ 1, 0, 1, 1, 0, 0, 0, 0, 1 } 24 times and then discarding the last 4 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = IM(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.5.3 Interleaving

The interleaving is done as specified for the SID_UPDATE_INH in subclause 3.10.2.3.

3.10.5.4 Mapping on a Burst

The mapping is done as specified for the SID_UPDATE_INH in subclause 3.10.2.4.

3.10.6 ONSET

Onset frames are used to preset the interleaver buffer after a period of no speech activity in DTX mode. This frame type contains no source data from the speech coder. What is transmitted is the in-band channel signalling the Mode Indication for the speech frame following the onset marker.

3.10.6.1 Coding of in-band data

The in-band data, Mode Indication $id1(0,1)$, will be encoded to $ic1(0..15)$. This sequence is then repeated 7 times more, and the last 14 bits are discarded ($8*16-14=114$) giving the sequence $ic1(0..113)$.

This sequence is then moved to c as:

$$c(2*k+1) = ic1(k) \quad \text{for } k = 0, 1, \dots, 113$$

3.10.6.2 Interleaving

The interleaving is done as specified for the SID_UPDATE_INH in subclause 3.10.2.3.

3.10.6.3 Mapping on a Burst

The mapping is done as specified for the SID_UPDATE_INH in subclause 3.10.2.4.

3.10.7 SPEECH

The speech coder delivers to the channel encoder a sequence of blocks of data. One block of data corresponds to one speech frame and the block length is different in each of the six channel codec modes. Adjoining each block of data is information of the channel codec mode to use when encoding the block Also delivered is the in-band data id(0,1) representing Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.10.7.1 Coding of the in-band data

The two bits to be in-band encoded, id(0,1), are encoded into ic(0..3).

The encoded in-band data (4 bits) are then moved to c(k) as:

$$c(k) = ic(k) \quad \text{for } k = 0, 1, \dots, 3$$

3.10.7.2 Ordering according to subjective importance

The bits delivered by the speech encoder, {s(1),s(2),...,s(K_s)}, are rearranged according to subjective importance before channel coding, Tables 9, 10, 11, 12, 13, 14 define the correct rearrangement for the speech codec modes 7.95 kbit/s, 7.40 kbit/s, 6.70 kbit/s, 5.90 kbit/s, 5.15 kbit/s and 4.75 kbit/s, respectively. In the tables speech codec parameters are numbered in the order they are delivered by the corresponding speech encoder according to GSM 06.90 [11] and the rearranged bits are labelled {d(0),d(1),...,d(K_d-1)}, defined in the order of decreasing importance. Index K_d refers to the number of bits delivered by the speech encoder, see below:

| Codec mode | Number of speech bits delivered per block (K _d) |
|-------------|---|
| TCH/AHS7.95 | 159 |
| TCH/AHS7.4 | 148 |
| TCH/AHS6.7 | 134 |
| TCH/AHS5.9 | 118 |
| TCH/AHS5.15 | 103 |
| TCH/AHS4.75 | 95 |

The ordering algorithm is in pseudo code as:

$$\text{for } j = 0 \text{ to } K_d-1 \quad d(j) := s(\text{table}(j)+1); \quad \text{where table}(j) \text{ is read line by line left to right}$$

The rearranged bits are further divided into three different classes to perform unequal error protection for different bits according to subjective importance.

The protection classes are:

- 1a - Data protected with the CRC and the convolution code.
- 1b - Data protected with the convolution code.
- 2 - Data sent without protection.

The number of class 1 (sum of class 1a and 1b), class 1a, class 1b and class 2 bits for each codec mode is shown below:

| Codec mode | Number of speech bits delivered per block | Number of class 1 bits per block | Number of class 1a bits per block | Number of class 1b bits per block | Number of class 2 bits per block |
|-------------|---|----------------------------------|-----------------------------------|-----------------------------------|----------------------------------|
| TCH/AHS7.95 | 159 | 123 | 67 | 56 | 36 |
| TCH/AHS7.4 | 148 | 120 | 61 | 59 | 28 |
| TCH/AHS6.7 | 134 | 110 | 55 | 55 | 24 |
| TCH/AHS5.9 | 118 | 102 | 55 | 47 | 16 |
| TCH/AHS5.15 | 103 | 91 | 49 | 42 | 12 |
| TCH/AHS4.75 | 95 | 83 | 39 | 44 | 12 |

3.10.7.3 Parity for speech frames

The basic parameters for each codec mode for the first encoding step are shown below:

| Mode number | Number of class 1 bits (K_{d1}) | CRC protected bits (K_{d1a}) | Number of output bits from first encoding step ($K_u = K_{d1} + 6$) |
|-------------|-------------------------------------|----------------------------------|---|
| TCH/AHS7.95 | 123 | 67 | 129 |
| TCH/AHS7.4 | 120 | 61 | 126 |
| TCH/AHS6.7 | 110 | 55 | 116 |
| TCH/AHS5.9 | 102 | 55 | 108 |
| TCH/AHS5.15 | 91 | 49 | 97 |
| TCH/AHS4.75 | 83 | 39 | 89 |

A 6-bit CRC is used for error-detection. These 6 parity bits are generated by the cyclic generator polynomial: $g(D) = D^6 + D^5 + D^3 + D^2 + D^1 + 1$ from the first K_{d1a} bits of class 1, where K_{d1a} refers to number of bits in protection class 1a. The value of K_{d1a} for each codec mode is shown above.

The encoding of the cyclic code is performed in a systematic form, which means that, in GF(2), the polynomial: $d(0)D^{(K_{d1a}+5)} + d(1)D^{(K_{d1a}+4)} + \dots + d(K_{d1a}-1)D^6 + p(0)D^5 + \dots + p(4)D + p(5)$

where $p(0), p(1) \dots p(5)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to:

$$1 + D + D^2 + D^3 + D^4 + D^5.$$

The information and parity bits are merged:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, K_{d1a}-1 \\ u(k) &= p(k-K_{d1a}) && \text{for } k = K_{d1a}, K_{d1a}+1, \dots, K_{d1a}+5 \\ u(k) &= d(k-6) && \text{for } k = K_{d1a}+6, K_{d1a}+7, \dots, K_u-1 \end{aligned}$$

Thus, after the first encoding step $u(k)$ will be defined by the following contents for each codec mode:

TCH/AHS7.95:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 66 \\ u(k) &= p(k-67) && \text{for } k = 67, 68, \dots, 72 \\ u(k) &= d(k-6) && \text{for } k = 73, 74, \dots, 128 \end{aligned}$$

TCH/AHS7.4:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 60 \\ u(k) &= p(k-61) && \text{for } k = 61, 62, \dots, 66 \\ u(k) &= d(k-6) && \text{for } k = 67, 68, \dots, 125 \end{aligned}$$

TCH/AHS6.7:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 54 \\ u(k) &= p(k-55) && \text{for } k = 55, 56, \dots, 60 \\ u(k) &= d(k-6) && \text{for } k = 61, 62, \dots, 115 \end{aligned}$$

TCH/AHS5.9:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 54 \\ u(k) &= p(k-55) && \text{for } k = 55, 56, \dots, 60 \\ u(k) &= d(k-6) && \text{for } k = 61, 62, \dots, 107 \end{aligned}$$

TCH/AHS5.15:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 48 \\ u(k) &= p(k-49) && \text{for } k = 49, 50, \dots, 54 \\ u(k) &= d(k-6) && \text{for } k = 55, 56, \dots, 96 \end{aligned}$$

TCH/AHS4.75:

$$\begin{aligned} u(k) &= d(k) && \text{for } k = 0, 1, \dots, 38 \\ u(k) &= p(k-39) && \text{for } k = 39, 40, \dots, 44 \\ u(k) &= d(k-6) && \text{for } k = 45, 46, \dots, 88 \end{aligned}$$

3.10.7.4 Convolutional encoder

The bits from the first encoding step ($u(k)$) are encoded with the recursive systematic convolution code as summarised below:

| Codec mode | Number of input bits to conv. code | Rate | Number of output bits from conv. code | Number of punctured bits |
|-------------|------------------------------------|------|---------------------------------------|--------------------------|
| TCH/AHS7.95 | 129 | 1/2 | 266 | 78 |
| TCH/AHS7.4 | 126 | 1/2 | 260 | 64 |
| TCH/AHS6.7 | 116 | 1/2 | 240 | 40 |
| TCH/AHS5.9 | 108 | 1/2 | 224 | 16 |
| TCH/AHS5.15 | 97 | 1/3 | 303 | 91 |
| TCH/AHS4.75 | 89 | 1/3 | 285 | 73 |

Below the coding for each codec mode is specified in detail.

TCH/AHS7.95:

The block of 129 bits $\{u(0) \dots u(128)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0/G_0 = 1$$

$$G_1/G_0 = 1 + D + D^3 + D^4 / 1 + D^3 + D^4$$

resulting in 266 coded bits, $\{C(0) \dots C(265)\}$ defined by:

$$r(k) = u(k) + r(k-3) + r(k-4)$$

$$C(2k) = u(k)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 0, 1, \dots, 128; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(2k) = r(k-3) + r(k-4)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 129, 130 \dots, 132$$

The code is punctured in such a way that the following 78 coded bits:

C(1), C(3), C(5), C(7), C(11), C(15), C(19), C(23), C(27), C(31), C(35), C(43), C(47), C(51), C(55), C(59), C(63), C(67), C(71), C(79), C(83), C(87), C(91), C(95), C(99), C(103), C(107), C(115), C(119), C(123), C(127), C(131), C(135), C(139), C(143), C(151), C(155), C(159), C(163), C(167), C(171), C(175), C(177), C(179), C(183), C(185), C(187), C(191), C(193), C(195), C(197), C(199), C(203), C(205), C(207), C(211), C(213), C(215), C(219), C(221), C(223), C(227), C(229), C(231), C(233), C(235), C(239), C(241), C(243), C(247), C(249), C(251), C(255), C(257), C(259), C(261), C(263) and C(265)

are not transmitted. The result is a block of 188 coded and punctured bits, P(0)...P(187) which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 187.$$

Finally the 36 class 2 bits are appended to c

$$c(192+k) = d(123+k) \quad \text{for } k = 0, 1, \dots, 35.$$

TCH/AHS7.4:

The block of 126 bits {u(0)... u(125)} is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0/G_0 = 1$$

$$G_1/G_0 = 1 + D + D^3 + D^4 / 1 + D^3 + D^4$$

resulting in 260 coded bits, {C(0)... C(259)} defined by:

$$r(k) = u(k) + r(k-3) + r(k-4)$$

$$C(2k) = u(k)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 0, 1, \dots, 125; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(2k) = r(k-3) + r(k-4)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 126, 127 \dots, 129$$

The code is punctured in such a way that the following 64 coded bits:

C(1), C(3), C(7), C(11), C(19), C(23), C(27), C(35), C(39), C(43), C(51), C(55), C(59), C(67), C(71), C(75), C(83), C(87), C(91), C(99), C(103), C(107), C(115), C(119), C(123), C(131), C(135), C(139), C(143), C(147), C(151), C(155), C(159), C(163), C(167), C(171), C(175), C(179), C(183), C(187), C(191), C(195), C(199), C(203), C(207), C(211), C(215), C(219), C(221), C(223), C(227), C(229), C(231), C(235), C(237), C(239), C(243), C(245), C(247), C(251), C(253), C(255), C(257) and C(259)

are not transmitted. The result is a block of 196 coded and punctured bits, P(0)...P(195) which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 195.$$

Finally the 28 class 2 bits are appended to c

$$c(200+k) = d(120+k) \quad \text{for } k = 0, 1, \dots, 27.$$

TCH/AHS6.7:

The block of 116 bits $\{u(0)\dots u(115)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0/G_0 = 1$$

$$G_1/G_0 = 1 + D + D^3 + D^4 / 1 + D^3 + D^4$$

resulting in 240 coded bits, $\{C(0)\dots C(239)\}$ defined by:

$$r(k) = u(k) + r(k-3) + r(k-4)$$

$$C(2k) = u(k)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 0, 1, \dots, 115; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(2k) = r(k-3) + r(k-4)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 116, 117 \dots, 119$$

The code is punctured in such a way that the following 40 coded bits:

$C(1), C(3), C(9), C(19), C(29), C(39), C(49), C(59), C(69), C(79), C(89), C(99), C(109), C(119), C(129), C(139), C(149), C(159), C(167), C(169), C(177), C(179), C(187), C(189), C(197), C(199), C(203), C(207), C(209), C(213), C(217), C(219), C(223), C(227), C(229), C(231), C(233), C(235), C(237)$ and $C(239)$

are not transmitted. The result is a block of 200 coded and punctured bits, $P(0)\dots P(199)$ which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 199.$$

Finally the 24 class 2 bits are appended to c

$$c(204+k) = d(110+k) \quad \text{for } k = 0, 1, \dots, 23.$$

TCH/AHS5.9:

The block of 108 bits $\{u(0)\dots u(107)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_0/G_0 = 1$$

$$G_1/G_0 = 1 + D + D^3 + D^4 / 1 + D^3 + D^4$$

resulting in 224 coded bits, $\{C(0)\dots C(223)\}$ defined by:

$$r(k) = u(k) + r(k-3) + r(k-4)$$

$$C(2k) = u(k)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 0, 1, \dots, 107; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(2k) = r(k-3) + r(k-4)$$

$$C(2k+1) = r(k)+r(k-1)+r(k-3)+r(k-4) \quad \text{for } k = 108, 109 \dots, 111$$

The code is punctured in such a way that the following 16 coded bits:

$C(1), C(15), C(71), C(127), C(139), C(151), C(163), C(175), C(187), C(195), C(203), C(211), C(215),$
 $C(219), C(221)$ and $C(223)$

are not transmitted. The result is a block of 208 coded and punctured bits, $P(0)...P(207)$ which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 207.$$

Finally the 16 class 2 bits are appended to c

$$c(212+k) = d(102+k) \quad \text{for } k = 0, 1, \dots, 15.$$

TCH/AHS5.15:

The block of 97 bits $\{u(0)...u(96)\}$ is encoded with the 1/3 rate convolutional code defined by the following polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

resulting in 303 coded bits, $\{C(0)...C(302)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(3k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = u(k) \quad \text{for } k = 0, 1, \dots, 96$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(3k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(3k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(3k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 97, 98, \dots, 100$$

The code is punctured in such a way that the following 91 coded bits:

$C(0), C(1), C(3), C(4), C(6), C(9), C(12), C(15), C(18), C(21), C(27), C(33), C(39), C(45), C(51), C(54),$
 $C(57), C(63), C(69), C(75), C(81), C(87), C(90), C(93), C(99), C(105), C(111), C(117), C(123), C(126),$
 $C(129), C(135), C(141), C(147), C(153), C(159), C(162), C(165), C(168), C(171), C(174), C(177), C(180),$
 $C(183), C(186), C(189), C(192), C(195), C(198), C(201), C(204), C(207), C(210), C(213), C(216), C(219),$
 $C(222), C(225), C(228), C(231), C(234), C(237), C(240), C(243), C(244), C(246), C(249), C(252), C(255),$
 $C(256), C(258), C(261), C(264), C(267), C(268), C(270), C(273), C(276), C(279), C(280), C(282), C(285),$
 $C(288), C(289), C(291), C(294), C(295), C(297), C(298), C(300)$ and $C(301)$

are not transmitted. The result is a block of 212 coded and punctured bits, $P(0)...P(211)$ which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 211.$$

Finally the 12 class 2 bits are appended to c

$$c(216+k) = d(91+k) \quad \text{for } k = 0, 1, \dots, 11.$$

TCH/AHS4.75:

The block of 89 bits $\{u(0)\dots u(88)\}$ is encoded with the 1/3 rate convolutional code defined by the following polynomials:

$$G4/G4 = 1$$

$$G5/G4 = 1 + D + D^4 + D^6 / 1 + D^2 + D^3 + D^5 + D^6$$

$$G6/G4 = 1 + D + D^2 + D^3 + D^4 + D^6 / 1 + D^2 + D^3 + D^5 + D^6$$

resulting in 285 coded bits, $\{C(0)\dots C(284)\}$ defined by:

$$r(k) = u(k) + r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(3k) = u(k)$$

$$C(3k+1) = r(k)+r(k-1)+r(k-4)+r(k-6)$$

$$C(3k+2) = r(k)+r(k-1) + r(k-2)+r(k-3)+r(k-4)+r(k-6) \quad \text{for } k = 0, 1, \dots, 88; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(3k) = r(k-2) + r(k-3) + r(k-5) + r(k-6)$$

$$C(3k+1) = r(k)+r(k-1)+r(k-4)+r(k-6)$$

$$C(3k+2) = r(k)+r(k-1) + r(k-2)+r(k-3)+r(k-4)+r(k-6) \quad \text{for } k = 89, 90, \dots, 94$$

The code is punctured in such a way that the following 73 coded bits:

$C(1), C(2), C(4), C(5), C(7), C(8), C(10), C(13), C(16), C(22), C(28), C(34), C(40), C(46), C(52), C(58), C(64), C(70), C(76), C(82), C(88), C(94), C(100), C(106), C(112), C(118), C(124), C(130), C(136), C(142), C(148), C(151), C(154), C(160), C(163), C(166), C(172), C(175), C(178), C(184), C(187), C(190), C(196), C(199), C(202), C(208), C(211), C(214), C(220), C(223), C(226), C(232), C(235), C(238), C(241), C(244), C(247), C(250), C(253), C(256), C(259), C(262), C(265), C(268), C(271), C(274), C(275), C(277), C(278), C(280), C(281), C(283)$ and $C(284)$

are not transmitted. The result is a block of 212 coded and punctured bits, $P(0)\dots P(211)$ which are appended to the in-band bits in c as

$$c(k+4) = P(k) \quad \text{for } k = 0, 1, \dots, 211.$$

Finally the 12 class 2 bits are appended to c

$$c(216+k) = d(83+k) \text{ for } k = 0, 1, \dots, 11.$$

3.10.7.5 Interleaving

The interleaving is done as specified for the TCH/HS in subclause 3.2.3.

3.10.7.6 Mapping on a Burst

The mapping is done as specified for the TCH/HS in subclause 3.2.4.

3.10.8 RATSCCH_MARKER

This frame type contains the in-band channel and an identification marker. The in-band data $id(0,1)$ represents Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.10.8.1 Coding of in-band data

The in-band data, $ic(0,1)$, is encoded to $ic(0..15)$ which is moved to the coded data c as:

$$c(k) = ic(k) \quad \text{for } k = 0, 1, \dots, 15$$

3.10.8.2 Identification marker

The identification marker, $IM(0..211)$, is constructed by repeating the following 11-bit sequence:

{ 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1 } 20 times and then discarding the last 8 bits. This block of data is moved to the coded data (c) as:

$$c(k+16) = IM(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.8.3 Interleaving

The interleaving is done as specified for the TCH/HS in subclause 3.2.3.

3.10.8.4 Mapping on a Burst

The mapping is done as specified for the TCH/HS in subclause 3.2.4.

3.10.9 RATSCCH_DATA

This frame contains the RATSCCH data and an inband channel. The RATSCCH data consists of 35 bits. The in-band data $id(0,1)$ represents Mode Indication or Mode Command/Mode Request depending on the current frame number.

3.10.9.1 Coding of in-band data

The in-band data, $ic(0,1)$, is encoded to $ic(0..15)$ which is moved to the coded data c as:

$$c(k) = ic(k) \quad \text{for } k = 0, 1, \dots, 15$$

3.10.9.2 Parity and convolutional encoding for the RATSCCH message

a) Parity bits:

A 14-bit CRC is used for error-detection. These 14 parity bits are generated by the cyclic generator polynomial: $g(D) = D^{14} + D^{13} + D^5 + D^3 + D^2 + 1$ from the 35 comfort noise parameter bits. The encoding of the cyclic code is performed in a systematic form, which means that, in $GF(2)$, the polynomial:

$$d(0)D^{48} + d(1)D^{47} + \dots + d(34)D^{14} + p(0)D^{13} + \dots + p(12)D + p(13)$$

where $p(0), p(1) \dots p(13)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to $1 + D + D^2 + D^3 + D^4 + D^5 + D^6 + D^7 + D^8 + D^9 + D^{10} + D^{11} + D^{12} + D^{13}$

The information and parity bits are merged:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 34$$

$$u(k) = p(k-35) \quad \text{for } k = 35, 36, \dots, 48$$

b) Convolutional encoder

The comfort noise parameters with parity and tail bits ($u(0..48)$) are encoded with the 1/4 rate

convolutional code defined by the polynomials:

$$G1/G3 = 1 + D + D^3 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G2/G3 = 1 + D^2 + D^4 / 1 + D + D^2 + D^3 + D^4$$

$$G3/G3 = 1$$

$$G3/G3 = 1$$

resulting in 212 coded bits, $\{C(0)... C(211)\}$ defined by:

$$r(k) = u(k) + r(k-1) + r(k-2) + r(k-3) + r(k-4)$$

$$C(4k) = r(k) + r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = u(k)$$

$$C(4k+3) = u(k) \quad \text{for } k = 0, 1, \dots, 48; r(k) = 0 \text{ for } k < 0$$

and (for termination of the coder):

$$r(k) = 0$$

$$C(4k) = r(k)+r(k-1) + r(k-3) + r(k-4)$$

$$C(4k+1) = r(k)+r(k-2)+r(k-4)$$

$$C(4k+2) = r(k-1)+r(k-2)+r(k-3)+r(k-4)$$

$$C(4k+3) = r(k-1)+r(k-2)+r(k-3)+r(k-4) \quad \text{for } k = 49, 50, \dots, 52$$

This block of data is moved to the coded data (c) as:

$$c(k+16) = C(k) \quad \text{for } k = 0, 1, \dots, 211$$

3.10.9.3 Interleaving

The interleaving is done as specified for the TCH/HS in subclause 3.2.3.

3.10.9.4 Mapping on a Burst

The mapping is done as specified for the TCH/HS in subclause 3.2.4.

3.11 Data channel for ECSD at full rate, 29.0 kbit/s radio interface rate (28.8 kbit/s services (E-TCH/F28.8))

The definition of a 28.8 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.11.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 580 information bits (data frames) every 20 ms.

3.11.2 Block code

3.11.2.1 Repetition bits

To match to RS alphabet 4 extra data bits are added to the end of each block of 580 bits: $d(k)=0, k=580,\dots,583$.

3.11.2.2 Reed Solomon encoder

The block of 584 information bits is encoded by shortened systematic Reed Solomon (RS) code over Galois field $GF(2^8)$. The Galois field $GF(2^8)$ is built as an extension of $GF(2)$. The characteristic of $GF(2^8)$ is equal to 2.

The code used is systematic RS₈ (85,73), which is shortened systematic RS₈(255,243) code over GF(2⁸) with the primitive polynomial $p(x)=x^8+x^4+x^3+x^2+1$. The primitive element a is the root of the primitive polynomial, i.e.

$$a^8 = a^4 + a^3 + a^2 + 1.$$

Generator polynomial for RS₈(255,243) code is:

$g(x)=\prod_{i=0}^{11}(x-a^{i+122})$; that results in symmetrical form for the generator polynomial with coefficients given in decimal notation

$$g(x)=x^{12} + 18x^{11} + 157x^{10} + 162x^9 + 134x^8 + 157x^7 + 253x^6 + 157x^5 + 134x^4 + 162x^3 + 157x^2 + 18x + 1$$

where binary presentation of polynomial coefficients in GF(256) is $\{a^7, a^6, a^5, a^4, a^3, a^2, a, 1\}$.

Specifically, decimal, power and polynomial presentations for the generator polynomial coefficients are the following:

$$x^{12}: 1$$

$$x^{11}: 18 = a^{224} = a^4 + a$$

$$x^{10}: 157 = a^{32} = a^7 + a^4 + a^3 + a^2 + 1$$

$$x^9: 162 = a^{209} = a^7 + a^5 + a$$

$$x^8: 134 = a^{99} = a^7 + a^2 + a$$

$$x^7: 157 = a^{32} = a^7 + a^4 + a^3 + a^2 + 1$$

$$x^6: 253 = a^{80} = a^7 + a^6 + a^5 + a^4 + a^3 + a^2 + 1$$

$$x^5: 157 = a^{32} = a^7 + a^4 + a^3 + a^2 + 1$$

$$x^4: 134 = a^{99} = a^7 + a^2 + a$$

$$x^3: 162 = a^{209} = a^7 + a^5 + a$$

$$x^2: 157 = a^{32} = a^7 + a^4 + a^3 + a^2 + 1$$

$$x^1: 18 = a^{224} = a^4 + a$$

$$x^0: 1 = a^{255} = 1$$

The RS encoding is performed in the following three steps:

a) Bit to symbol conversion

The information bits $\{d(0),d(1),\dots,d(583)\}$ are converted into 73 information 8-bit symbols $\{D(0),\dots,D(72)\}$ as the following:

$$D(k) = 128d(8k+7) + 64d(8k+6) + 32d(8k+5) + 16d(8k+4) + 8d(8k+3) + 4d(8k+2) + 2d(8k+1) + d(8k) \\ \text{for } k = 0,1,\dots,72$$

Resulting 8-bit symbols are presented as

$$D(k) = \{d(8k+7), d(8k+6), d(8k+5), d(8k+4), d(8k+3), d(8k+2), d(8k+1), d(8k)\} \quad \text{for } k = 0,1,\dots,72$$

where $d(8k+7),\dots,d(8k)$ are ordered from the most significant bit (MSB) to the less significant bit (LSB).

The polynomial representation of a single information symbol over GF(2⁸) in terms of a is given by

$$D_a(k) = a^7d(8k+7) + a^6d(8k+6) + a^5d(8k+5) + a^4d(8k+4) + a^3d(8k+3) + a^2d(8k+2) + ad(8k+1) + d(8k)$$

b) Encoding

The information symbols $D(0)\dots D(72)$ are encoded by shortened systematic $RS_8(85,73)$ code with output symbols $U(0)\dots U(84)$ ordered as

$$U(k)=D(k) \text{ for } k=0,1,\dots,72; U(k)=R(k) \text{ for } k=73,74,\dots,84;$$

where $R(k)$ are parity check symbols added by $RS_8(85,73)$ encoder.

Information symbols are ordered in the descending polynomial order such that $D_a(72)$ corresponds to the lowest

degree term of $D(x) = D_a(72) + D_a(71)x + \dots + D_a(1)x^{71} + D_a(0)x^{72}$, where $D(x)$ is the polynomial representation of

information symbols $\{D(0),D(1),\dots,D(72)\}$ over Galois field .

Parity check symbols in polynomial representation over Galois field are ordered in the descending polynomial order such that $R_a(84)$ corresponds to the lowest degree of $R(x)=R_a(84) + R_a(83)x + \dots + R_a(74)x^{10} + R_a(73)x^{11}$. The parity check symbols are calculated as $R(x) = \text{remainder}[x^{12}D(x)/g(x)]$, and $U(x) = R(x) + x^{12}D(x)$, i.e.,

$$U_a(k) = D_a(k) \text{ for } k=0,1,\dots,72; U_a(k) = R_a(k) \text{ for } k=73,74,\dots,84.$$

The encoding operation with the shortened $RS_8(85,73)$ code may be presented as the following:

- Expanding 73 information symbols to the block of 243 symbols by adding 170 dump (zero) symbols
- Encoding 243 symbols by systematic $RS_8(255,243)$ encoder with outer block of 255 symbols
- Removing 170 dump symbols, resulting in the output block of 85 symbols.

c) Symbol to bit conversion

The output symbols $\{U_a(0),\dots,U_a(84)\}$ are converted back into symbols $\{U(0),\dots,U(84)\}$ and then back into binary form with LSB coming out first, resulting in the block of 680 bits $\{u(0),\dots,u(679)\}$.

3.11.3 Convolutional encoder**3.11.3.1 Tailing bits for a data frame**

Before convolutional encoding 6 tail bits $\{u(k)=0, k=680,\dots,685\}$ are added to the end of each data block .

3.11.3.2 Convolutional encoding for a data frame

This block of 686 bits $\{u(0),\dots,u(685)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_7 = 1 + D + D^2 + D^3 + D^6$$

resulting in 1372 coded bits $\{c(0), c(1),\dots, c(1371)\}$ with

$$c(2k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6);$$

$$c(2k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6) \text{ for } k = 0,1,\dots,685; u(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following 4 coded bits:

$$c(363), c(723), c(1083) \text{ and } c(1299) \text{ are not transmitted.}$$

The result is a block of 1368 coded bits, $\{c(0),c(1),\dots, c(1367)\}$.

3.11.4 Interleaving

The interleaving scheme is presented below.

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k), \quad \text{for } k = 0,1,\dots,1367$$

$$n = 0,1,\dots,N,N+1,\dots$$

$$B = B_0 + 4n + (k \bmod 19) + (k \text{ div } 342)$$

$$j = (k \bmod 19) + 19(k \text{ div } 18)$$

The result of the interleaving is a distribution of the reordered 342 bit of a given data block, $n = N$, over 19 blocks, 18 bits equally distributed in each block, in a diagonal way over consecutive blocks.

Or in other words the interleaving is a distribution of the encoded, reordered 1368 bits from four given input data blocks, which taken together give $n = N$, over 22 bursts, 18 bits equally distributed in the first and 22nd bursts, 36 bits distributed in the second and 21st bursts, 54 bits distributed in the third and 20th bursts and 72 bits distributed in the other 16 bursts.

The block of coded data is interleaved "diagonal", where a new block of coded data starts with every fourth burst and is distributed over 22 bursts.

3.11.5 Mapping on a Burst

Before mapping on a burst the interleaved bits $\{i(0)\dots i(1367)\}$ are converted into 3-bit symbols $\{I(0),I(1), \dots,I(455)\}$ according to Table 1 in GSM 05.04, the symbol $I(k)$ depends on $i(3k+2)$, $i(3k+1)$ and $i(3k)$ for $k=0,1,\dots,455$.

The E-IACCH message delivered to the encoder on every 20ms has a fixed size of 3 information bits $\{im(0), im(1), im(2)\}$. The contents of the bits are defined in GSM 05.08 for both uplink and downlink.

The E-IACCH information bits $\{im(n,0),im(n,1),im(n,2)\}$ are coded into 24 bits $ib(B,k)$, $B_0 + 4n \leq B < B_0 + 4n + 4$, $k = 0,1,\dots,5$ according to the following table:

| <u>$im(n,0),im(n,1),im(n,2)$</u> | <u>$ib(B_0+4n,0),\dots,ib(B_0+4n,5),\dots,$ $ib(B_0+4n+3,0),\dots,ib(B_0+4n+3,5)$</u> |
|---|---|
| <u>000</u> | <u>000000 000000 000000 000000</u> |
| <u>001</u> | <u>001111 110100 100101 110100</u> |
| <u>010</u> | <u>011100 010111 111001 100011</u> |
| <u>011</u> | <u>010011 100011 011100 010111</u> |
| <u>100</u> | <u>100110 011001 110110 001101</u> |
| <u>101</u> | <u>101001 101101 010011 111001</u> |
| <u>110</u> | <u>111010 001110 001111 101110</u> |
| <u>111</u> | <u>110101 111010 101010 011010</u> |

Before mapping on a burst the E-IACCH bits $\{ib(B,0)\dots ib(B,5)\}$ are converted into 3-bit symbols $\{HL(B),HU(B)\}$ according to Table 1 in GSM 05.04. The symbol $HL(B)$ depends on $ib(B,2)$, $ib(B,1)$ and $ib(B,0)$ and , the symbol $HU(B)$ on $ib(B,5)$, $ib(B,4)$ and $ib(B,3)$.

The mapping is given by the rule:

$$E(B,j) = I(B,j) \quad \text{and} \quad E(B,59+j) = I(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$E(B,57) = HL(B) \quad \text{and} \quad E(B,58) = HU(B).$$

The two symbols, labelled $HL(B)$ and $HU(B)$ on burst number B are flags used for E-IACCH.

3.12 Data channel for ECSD at full rate, 32.0 kbit/s radio interface rate (32.0 kbit/s services (E-TCH/F32.0))

The definition of a 32.0 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.12.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 640 information bits (data frames) every 20 ms.

3.12.2 Block code

The block of 640 information bits is encoded by shortened systematic Reed Solomon (RS) code over Galois field GF(2⁸). The Galois field GF(2⁸) is built as an extension of GF(2). The characteristic *a* of GF(2⁸) is equal to 2.

The code used is systematic RS₈(92,80), which is shortened systematic RS₈(255,243) code over GF(2⁸) with the primitive polynomial $p(x)=x^8+x^4+x^3+x^2+1$. The primitive element *a* is the root of the primitive polynomial, i.e.

$$a^8 = a^4 + a^3 + a^2 + 1.$$

Generator polynomial for RS₈(255,243) code is:

$$g(x) = \prod_{i=0}^{11} (x - a^{i+122});$$

that results in symmetrical form for the generator polynomial with coefficients given in decimal notation

$$g(x) = x^{12} + 18x^{11} + 157x^{10} + 162x^9 + 134x^8 + 157x^7 + 253x^6 + 157x^5 + 134x^4 + 162x^3 + 157x^2 + 18x + 1$$

where binary presentation of polynomial coefficients in GF(256) is {*a*⁷, *a*⁶, *a*⁵, *a*⁴, *a*³, *a*², *a*, 1}

Specifically, decimal, power and polynomial presentations for the generator polynomial coefficients are presented in 3.11.2.2.

The RS encoding is performed in the following three steps:

a) Bit to symbol conversion

The information bits {*d*(0),*d*(1),...,*d*(639)} are converted into 80 information 8-bit symbols {*D*(0),...,*D*(79)} as the following:

$$D(k) = 128d(8k+7) + 64d(8k+6) + 32d(8k+5) + 16d(8k+4) + 8d(8k+3) + 4d(8k+2) + 2d(8k+1) + d(8k)$$

for *k* = 0,1,...,79

Resulting 8-bit symbols are presented as

$$D(k) = \{d(8k+7), d(8k+6), d(8k+5), d(8k+4), d(8k+3), d(8k+2), d(8k+1), d(8k)\} \quad \text{for } k = 0,1,\dots,79$$

where *d*(8*k*+7),...,*d*(8*k*) are ordered from the most significant bit (MSB) to the less significant bit (LSB).

The polynomial representation of a single information symbol over GF(2⁸) in terms of *a* is given by

$$D_a(k) = a^7d(8k+7) + a^6d(8k+6) + a^5d(8k+5) + a^4d(8k+4) + a^3d(8k+3) + a^2d(8k+2) + ad(8k+1) + d(8k)$$

b) Encoding

The information symbols *D*(0)...*D*(79) are encoded by shortened systematic RS₈(92,80) code with output symbols *U*(0)...*U*(91) ordered as

$$U(k)=D(k) \text{ for } k=0,1,\dots,79; U(k)=R(k) \text{ for } k=80,81,\dots,91;$$

where *R*(*k*) are parity check symbols added by RS₈(92,80) encoder.

Information symbols are ordered in the descending polynomial order such that $D_a(79)$ corresponds to the lowest degree term of $D(x) = D_a(79) + D_a(78)x + \dots + D_a(1)x^{78} + D_a(0)x^{79}$, where $D(x)$ is the polynomial representation of information symbols $\{D(0), D(1), \dots, D(79)\}$ over Galois field.

Parity check symbols in polynomial representation over Galois field are ordered in the descending polynomial order such that $R_a(91)$ corresponds to the lowest degree of $R(x) = R_a(91) + R_a(92)x + \dots + R_a(81)x^{10} + R_a(80)x^{11}$. The parity check symbols are calculated as $R(x) = \text{remainder}[x^{12}D(x)/g(x)]$ and $U(x) = R(x) + x^{12}D(x)$, i.e.,

$$U_a(k) = D_a(k) \text{ for } k=0,1,\dots,79; U_a(k) = R_a(k) \text{ for } k=80,\dots,91.$$

The encoding operation with the shortened $RS_8(92,80)$ code may be presented as the following:

- Expanding 80 information symbols to the block of 243 symbols by adding 163 dump (zero) symbols
- Encoding 243 symbols by systematic $RS_8(255,243)$ encoder with outer block of 255 symbols
- Removing 163 dump symbols, resulting in the output block of 92 symbols.

c) Symbol to bit conversion

The output symbols $\{U_a(0), \dots, U_a(91)\}$ are converted back into symbols $\{U(0), \dots, U(91)\}$ and then back into binary form with LSB coming out first, resulting in the block of 736 bits $\{u(0), \dots, u(735)\}$.

3.12.3 Convolutional encoder

3.12.3.1 Tailing bits for a data frame

Before convolutional encoding 6 tail bits $\{u(k)=0, k=736, \dots, 741\}$ are added to the end of each data block .

3.12.3.2 Convolutional encoding for a data frame

This block of 742 bits $\{u(0), \dots, u(741)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_7 = 1 + D + D^2 + D^3 + D^6$$

resulting in 1484 coded bits $\{c(0), c(1), \dots, c(1483)\}$ with

$$c(2k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6);$$

$$c(2k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6) \text{ for } k = 0, 1, \dots, 741; u(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following 116 coded bits:

$$c(3+12(k-1)) \text{ for } k=1, \dots, 15, 17, \dots, 31, 33, \dots, 47, 49, \dots, 63, 65, \dots, 79, 81, \dots, 95, 97, \dots, 111, 113, \dots, 123.$$

are not transmitted.

The result is a block of 1368 coded bits, $\{c(0), c(1), \dots, c(1367)\}$.

3.12.4 Interleaving

The interleaving is done as specified for E-TCH/F28.8 in subclause 3.11.4.

3.12.5 Mapping on a Burst

The mapping is done as specified for E-TCH/F28.8 in subclause 3.11.5.

3.13 Data channel for ECSD at full rate, 43.5 kbit/s radio interface rate (43.2 kbit/s services (E-TCH/F43.2))

The definition of a 43.5 kbit/s radio interface rate data flow for data services is given in GSM 04.21.

3.13.1 Interface with user unit

The user unit delivers to the encoder a bit stream organized in blocks of 870 information bits (data frames) every 20 ms.

3.13.2 Convolutional encoder

3.13.2.1 Tailing bits for a data frame

Before convolutional encoding 6 tail bits $\{d(k)=0, k=870, \dots, 875\}$ are added to the end of each data block .

3.13.2.2 Convolutional encoding for a data frame

This block of 876 bits $\{d(0), \dots, d(875)\}$ is encoded with the 1/2 rate convolutional code defined by the following polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

resulting in 1752 coded bits $\{c(0), c(1), \dots, c(1751)\}$ with

$$c(2k) = d(k) + d(k-2) + d(k-3) + d(k-5) + d(k-6);$$

$$c(2k+1) = d(k) + d(k-1) + d(k-2) + d(k-3) + d(k-6) \text{ for } k = 0, 1, \dots, 875; \text{ u}(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following 384 coded bits:

$$c(2+8(k-1)) \text{ for } k=1:219; c(4+16(k-1)) \text{ for } k=1:110; c(6+32(k-1)) \text{ for } k=1:55$$

are not transmitted.

The result is a block of 1368 coded bits, $\{c(0), c(1), \dots, c(1367)\}$.

3.13.3 Interleaving

The interleaving is done as specified for E-TCH/F28.8 in subclause 3.11.4.

3.13.4 Mapping on a Burst

The mapping is done as specified for E-TCH/F28.8 in subclause 3.11.5.

4 Control Channels

4.1 Slow associated control channel (SACCH)

4.1.1 Block constitution

The message delivered to the encoder has a fixed size of 184 information bits $\{d(0), d(1), \dots, d(183)\}$. It is delivered on a burst mode.

4.1.2 Block code

a) Parity bits:

The block of 184 information bits is protected by 40 extra bits used for error correction and detection. These bits are added to the 184 bits according to a shortened binary cyclic code (FIRE code) using the generator polynomial:

$$g(D) = (D^{23} + 1) * (D^{17} + D^3 + 1)$$

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

$$d(0)D^{223} + d(1)D^{222} + \dots + d(183)D^{40} + p(1)D^{38} + \dots + p(38)D + p(39)$$

where $\{p(0), p(1), \dots, p(39)\}$ are the parity bits, when divided by $g(D)$ yields a remainder equal to:

$$1 + D + D^2 + \dots + D^{39}.$$

b) Tail bits

Four tail bits equal to 0 are added to the information and parity bits, the result being a block of 228 bits.

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 183$$

$$u(k) = p(k-184) \quad \text{for } k = 184, 185, \dots, 223$$

$$u(k) = 0 \quad \text{for } k = 224, 225, 226, 227 \text{ (tail bits)}$$

4.1.3 Convolutional encoder

This block of 228 bits is encoded with the 1/2 rate convolutional code (identical to the one used for TCH/FS) defined by the polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

This results in a block of 456 coded bits: $\{c(0), c(1), \dots, c(455)\}$ defined by:

$$c(2k) = u(k) + u(k-3) + u(k-4)$$

$$c(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 227; u(k) = 0 \text{ for } k < 0$$

4.1.4 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B, j) = c(n, k) \text{ for } k = 0, 1, \dots, 455$$

$$n = 0, 1, \dots, N, N+1, \dots$$

$$B = B_0 + 4n + (k \bmod 4)$$

$$j = 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4)$$

See table 1. The result of the reordering of bits is the same as given for a TCH/FS (subclause 3.1.3) as can be seen from the evaluation of the bit number-index j , distributing the 456 bits over 4 blocks on even numbered bits and 4 blocks on odd numbered bits. The resulting 4 blocks are built by putting blocks with even numbered bits and blocks with odd numbered bits together into one block.

The block of coded data is interleaved "block rectangular" where a new data block starts every 4th block and is distributed over 4 blocks.

4.1.5 Mapping on a Burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \quad \text{and} \quad e(B,59+j) = i(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$e(B,57) = hl(B) \quad \text{and} \quad e(B,58) = hu(B)$$

The two bits labelled $hl(B)$ and $hu(B)$ on burst number B are flags used for indication of control channel signalling. They are set to "1" for a SACCH.

4.2 Fast associated control channel at full rate (FACCH/F)

4.2.1 Block constitution

The message delivered to the encoder has a fixed size of 184 information bits. It is delivered on a burst mode.

4.2.2 Block code

The block encoding is done as specified for the SACCH in subclause 4.1.2.

4.2.3 Convolutional encoder

The convolutional encoding is done as specified for the SACCH in subclause 4.1.3.

4.2.4 Interleaving

The interleaving is done as specified for the TCH/FS in subclause 3.1.3.

4.2.5 Mapping on a Burst

A FACCH/F frame of 456 coded bits is mapped on 8 consecutive bursts as specified for the TCH/FS in subclause 3.1.4. As a FACCH is transmitted on bits which are stolen in a burst from the traffic channel, the even numbered bits in the first 4 bursts and the odd numbered bits of the last 4 bursts are stolen.

To indicate this to the receiving device the flags $hl(B)$ and $hu(B)$ have to be set according to the following rule:

$hu(B) = 1$ for the first 4 bursts (even numbered bits are stolen);

$hl(B) = 1$ for the last 4 bursts (odd numbered bits are stolen).

The consequences of this bitstealing by a FACCH/F is for a:

- speech channel (TCH/FS) and data channel (TCH/F2.4):

One full frame of data is stolen by the FACCH.

- Data channel (TCH/F14.4):

The bitstealing by a FACCH/F disturbs a maximum of 96 of the 456 coded bits generated from an input data block of 290 bits.

- Data channel (TCH/F9.6):

The bitstealing by a FACCH/F disturbs a maximum of 96 coded bits generated from an input frame of four data blocks. A maximum of 24 of the 114 coded bits resulting from one input data block of 60 bits may be disturbed.

- Data channel (TCH/F4.8):

The bit stealing by FACCH/F disturbs a maximum of 96 coded bits generated from an input frame of two data blocks. A maximum of 48 of the 228 coded bits resulting from one input data block of 60 bits may be disturbed.

NOTE: In the case of consecutive stolen frames, a number of bursts will have both the even and the odd bits stolen and both flags $hu(B)$ and $hl(B)$ must be set to 1.

4.3 Fast associated control channel at half rate (FACCH/H)

4.3.1 Block constitution

The message delivered to the encoder has a fixed size of 184 information bits. It is delivered on a burst mode.

4.3.2 Block code

The block encoding is done as specified for the SACCH in subclause 4.1.2.

4.3.3 Convolutional encoder

The convolutional encoding is done as specified for the SACCH in subclause 4.1.3.

4.3.4 Interleaving

The coded bits are reordered and interleaved according to the following rule:

$$i(B,j) = c(n,k) \text{ for } k = 0,1,\dots,455$$

$$n = 0,1,\dots,N,N+1,\dots$$

$$B = B_0 + 4n + (k \bmod 8) - 4((k \bmod 8) \text{ div } 6)$$

$$j = 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4)$$

See table 1. The result of the reordering of bits is the same as given for a TCH/FS (subclause 3.1.3) as can be seen from the evaluation of the bit number-index j , distributing the 456 bits over 4 blocks on even numbered bits and 4 blocks on odd numbered bits. The 2 last blocks with even numbered bits and the 2 last blocks with odd numbered bits are put together into 2 full middle blocks.

The block of coded data is interleaved "block diagonal" where a new data block starts every 4th block and is distributed over 6 blocks.

4.3.5 Mapping on a Burst

A FACCH/H frame of 456 coded bits is mapped on 6 consecutive bursts by the rule:

$$e(B,j) = i(B,j) \quad \text{and} \quad e(B,59+j) = i(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$e(B,57) = hl(B) \quad \text{and} \quad e(B,58) = hu(B)$$

As a FACCH/H is transmitted on bits which are stolen from the traffic channel, the even numbered bits of the first 2 bursts, all bits of the middle 2 bursts and the odd numbered bits of the last 2 bursts are stolen.

To indicate this to the receiving device the flags $hl(B)$ and $hu(B)$ have to be set according to the following rule:

$$hu(B) = 1 \quad \text{for the first 2 bursts (even numbered bits are stolen)}$$

$$hu(B) = 1 \text{ and } hl(B) = 1 \text{ for the middle 2 bursts (all bits are stolen)}$$

$hl(B) = 1$ for the last 2 bursts (odd numbered bits are stolen)

The consequences of this bitstealing by a FACCH/H is for a:

- speech channel (TCH/HS):

two full consecutive speech frames are stolen by a FACCH/H.

- data channel (TCH/H4.8):

The bitstealing by FACCH/H disturbs a maximum of 96 coded bits generated from an input frame of four data blocks. A maximum of 24 out of the 114 coded bits resulting from one input data block of 60 bits may be disturbed.

- data channel (TCH/H2.4):

The bitstealing by FACCH/H disturbs a maximum of 96 coded bits generated from an input frame of four data blocks. A maximum of 24 out of the 114 coded bits resulting from one input data block of 36 bits may be disturbed.

NOTE: In the case of consecutive stolen frames, two overlapping bursts will have both the even and the odd numbered bits stolen and both flags $hu(B)$ and $hl(B)$ must be set to 1.

4.4 Broadcast control, Paging, Access grant, Notification and Cell broadcast channels (BCCH, PCH, AGCH, NCH, CBCH), CTS Paging and Access grant channels (CTSPCH, CTSAGCH)

The coding scheme used for the broadcast control, paging, access grant, notification and cell broadcast messages is the same as for the SACCH messages, specified in subclause 4.1. In CTS, the coding scheme used for the paging and access grant messages is also the same as for the SACCH messages, specified in subclause 4.1.

4.5 Stand-alone dedicated control channel (SDCCH)

The coding scheme used for the dedicated control channel messages is the same as for SACCH messages, specified in subclause 4.1.

4.6 Random access channel (RACH)

The burst carrying the random access uplink message has a different structure. It contains 8 information bits $d(0), d(1), \dots, d(7)$.

Six parity bits $p(0), p(1), \dots, p(5)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(0)D^{13} + \dots + d(7)D^6 + p(0)D^5 + \dots + p(5), \text{ when divided by } D^6 + D^5 + D^3 + D^2 + D + 1 \text{ yields a remainder equal to } D^5 + D^4 + D^3 + D^2 + D + 1.$$

The six bits of the BSIC, $\{B(0), B(1), \dots, B(5)\}$, of the BS to which the Random Access is intended, are added bitwise modulo 2 to the six parity bits, $\{p(0), p(1), \dots, p(5)\}$. This results in six colour bits, $C(0)$ to $C(5)$ defined as $C(k) = b(k) + p(k)$ ($k = 0$ to 5) where:

$b(0)$ = MSB of PLMN colour code

$b(5)$ = LSB of BS colour code.

This defines $\{u(0), u(1), \dots, u(17)\}$ by:

$u(k) = d(k)$ for $k = 0, 1, \dots, 7$

$u(k) = C(k-8)$ for $k = 8, 9, \dots, 13$

$$u(k) = 0 \quad \text{for } k = 14, 15, 16, 17 \text{ (tail bits)}$$

The bits $\{e(0), e(1), \dots, e(35)\}$ are obtained by the same convolutional code of rate 1/2 as for TCH/FS, defined by the polynomials:

$$G0 = 1 + D^3 + D^4$$

$$G1 = 1 + D + D^3 + D^4$$

and with:

$$e(2k) = u(k) + u(k-3) + u(k-4)$$

$$e(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 17 ; u(k) = 0 \text{ for } k < 0$$

4.7 Synchronization channel (SCH), Compact synchronization channel (CSCH), CTS Beacon and Access request channels (CTSBCH-SB, CTSARCH)

The burst carrying the synchronization information on the downlink BCCH, the downlink CPBCCCH for Compact, and in CTS the information of the CTSBCH-SB and the access request message of the CTSARCH, has a different structure. It contains 25 information bits $\{d(0), d(1), \dots, d(24)\}$, 10 parity bits $\{p(0), p(1), \dots, p(9)\}$ and 4 tail bits. The precise ordering of the information bits is given in GSM 04.08.

The ten parity bits $\{p(0), p(1), \dots, p(9)\}$ are defined in such a way that in GF(2) the binary polynomial:

$$d(0)D^{34} + \dots + d(24)D^{10} + p(0)D^9 + \dots + p(9), \text{ when divided by:}$$

$$D^{10} + D^8 + D^6 + D^5 + D^4 + D^2 + 1, \text{ yields a remainder equal to:}$$

$$D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

Thus the encoded bits $\{u(0), u(1), \dots, u(38)\}$ are:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 24$$

$$u(k) = p(k-25) \quad \text{for } k = 25, 26, \dots, 34$$

$$u(k) = 0 \quad \text{for } k = 35, 36, 37, 38 \text{ (tail bits)}$$

The bits $\{e(0), e(1), \dots, e(77)\}$ are obtained by the same convolutional code of rate 1/2 as for TCH/FS, defined by the polynomials:

$$G0 = 1 + D^3 + D^4$$

$$G1 = 1 + D + D^3 + D^4$$

and with:

$$e(2k) = u(k) + u(k-3) + u(k-4)$$

$$e(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 77 ; u(k) = 0 \text{ for } k < 0$$

4.8 Access Burst on circuit switched channels other than RACH

The encoding of this burst is as defined in subclause 4.6 for the random access channel (RACH). The BSIC used shall be the BSIC of the BTS to which the burst is intended.

4.9 Access Bursts for uplink access on a channel used for VGCS

The encoding of this burst is as defined in subclause 4.5 for the RACH. The BSIC used by the Mobile Station shall be the BSIC indicated by network signalling, or if not thus provided, the last received BSIC on the SCH of the current cell.

4.10 Fast associated control channel at ECSD E-TCH/F (E-FACCH/F)

4.10.1 Block constitution

The message delivered to the encoder has a fixed size of 184 information bits. It is delivered on a burst mode.

4.10.2 Block code

The block encoding is done as specified for the SACCH in subclause 4.1.2.

4.10.3 Convolutional encoder

The convolutional encoding is done as specified for the SACCH in subclause 4.1.3.

4.10.4 Interleaving

The interleaving is done as specified for the SACCH in subclause 4.1.4.

4.10.5 Mapping on a Burst

A E-FACCH/F frame of 456 coded bits is mapped on 4 full consecutive bursts. As a E-FACCH/F is transmitted on bits, which are stolen in a burst from the ECSD traffic channel, the four full bursts are stolen.

The mapping on is given by the rule:

$$e(B,j)=i(B,j) \text{ and } e(B,59+j)=i(B,57+j) \text{ for } j=0,1,\dots,56$$

and

$$e(B,57)=hl(B) \text{ and } e(B,58)=hu(B).$$

To indicate to the receiving device the flags $hl(B)$ and $hu(B)$ have to be set according to the following rule:

$$hu(B)=1 \text{ and } hl(B)=1 \text{ for the all 4 bursts (4 full bursts are stolen).}$$

The consequences of this bitstealing by a E-FACCH/F is for a:

- Data channel (E-TCH/F43.2)

The bitstealing by a E-FACCH/F disturbs a maximum of 288 of the 1368 coded bits generated from an input data block of 870 bits.

- Data channel (E-TCH/F32)

The bitstealing by a E-FACCH/F disturbs a maximum of 288 of the 1368 coded bits generated from an input data block of 640 bits.

- Data channel (E-TCH/F28.8)

The bitstealing by a E-FACCH/F disturbs a maximum of 288 of the 1368 coded bits generated from an input data block of 580 bits.

5 Packet Switched Channels

5.1 Packet data traffic channel (PDTCH)

Thirteen coding schemes are specified for the packet data traffic channels. For the three coding schemes CS-2 to CS-4, the first three bits (USF-bits) of the data block are encoded such that the first twelve coded bits are representing the same bit pattern, irrespective of the coding scheme, depending only on the USF-bits. For these coding schemes, the USF-bits can therefore always be decoded from these twelve bits in the same way. It should be noted that the USF precoding is done in the uplink direction for coding schemes CS-2 – CS-4, despite the fact that uplink RLC data block structure (GSM 04.60) does not define USF-field.

For the nine coding schemes MCS-1 to MCS-9, the block structure differs between uplink and downlink since header sizes before coding are not the same.

5.1.1 Packet data block type 1 (CS-1)

The coding scheme used for packet data block type 1 is the same as for SACCH as specified in section 4.1.

The flags hl(B) and hu(B) set to “1” identify the coding scheme CS-1.

5.1.2 Packet data block type 2 (CS-2)

5.1.2.1 Block constitution

The message delivered to the encoder has a fixed size of 271 information bits {d(0),d(1),...,d(270)}. It is delivered on a burst mode.

5.1.2.2 Block code

a) USF precoding:

The first three bits d(0),d(1),d(2) are precoded into six bits u'(0),u'(1),...,u'(5) according to the following table:

| d(0),d(1),d(2) | u'(0),u'(1),...,u'(5) |
|----------------|-----------------------|
| 000 | 000 000 |
| 001 | 001 011 |
| 010 | 010 110 |
| 011 | 011 101 |
| 100 | 100 101 |
| 101 | 101 110 |
| 110 | 110 011 |
| 111 | 111 000 |

b) Parity bits:

Sixteen parity bits p(0),p(1),...,p(15) are defined in such a way that in GF(2) the binary polynomial:

$$d(0)D^{286} + \dots + d(270)D^{16} + p(0)D^{15} + \dots + p(15), \text{ when divided by:}$$

$$D^{16} + D^{12} + D^5 + 1, \text{ yields a remainder equal to:}$$

$$D^{15} + D^{14} + D^{13} + D^{12} + D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

c) Tail bits:

Four tail bits equal to 0 are added to the information and parity bits, the result being a block of 294 bits {u(0),u(1),...,u(293)}:

$$u(k) = u'(k) \quad \text{for } k = 0, 1, \dots, 5$$

$$u(k) = d(k-3) \quad \text{for } k = 6, 7, \dots, 273$$

$$u(k) = p(k-274) \quad \text{for } k = 274, 275, \dots, 289$$

$$u(k) = 0 \quad \text{for } k = 290, 291, 292, 293 \text{ (tail bits)}$$

5.1.2.3 Convolutional encoder

This block of 294 bits $\{u(0), u(1), \dots, u(293)\}$ is encoded with the 1/2 rate convolutional code (identical to the one used for TCH/FS) defined by the polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

This results in a block of 588 coded bits: $\{C(0), C(1), \dots, C(587)\}$ defined by:

$$C(2k) = u(k) + u(k-3) + u(k-4)$$

$$C(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 293 ; u(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following coded bits:

$$\{C(3+4j) \text{ for } j = 3, 4, \dots, 146 \text{ except for } j = 9, 21, 33, 45, 57, 69, 81, 93, 105, 117, 129, 141\}$$
 are not transmitted

The result is a block of 456 coded bits, $\{c(0), c(1), \dots, c(455)\}$.

5.1.2.4 Interleaving

The interleaving is done as specified for SACCH in section 4.1.4.

5.1.2.5 Mapping on a burst

The mapping is given by the rule:

$$e(B, j) = i(B, j) \text{ and } e(B, 59+j) = i(B, 57+j) \quad \text{for } j = 0, 1, \dots, 56$$

and

$$e(B+m, 57) = q(2m) \text{ and } e(B+m, 58) = q(2m+1) \text{ for } m = 0, 1, 2, 3$$

where

$$q(0), q(1), \dots, q(7) = 1, 1, 0, 0, 1, 0, 0, 0 \text{ identifies the coding scheme CS-2.}$$

5.1.3 Packet data block type 3 (CS-3)

5.1.3.1 Block constitution

The messages delivered to the encoder has a fixed size of 315 information bits $\{d(0), d(1), \dots, d(314)\}$. It is delivered on a burst mode.

5.1.3.2 Block code

a) USF precoding:

The first three bits $d(0), d(1), d(2)$ are precoded into six bits $u'(0), u'(1), \dots, u'(5)$ as specified for CS-2 in section 5.1.2.2.a).

b) Parity bits:

Sixteen parity bits $p(0), p(1), \dots, p(15)$ are defined in such a way that in GF(2) the binary polynomial:

$$d(0)D^{330} + \dots + d(314)D^{16} + p(0)D^{15} + \dots + p(15), \text{ when divided by:}$$

$D^{16} + D^{12} + D^5 + 1$, yields a remainder equal to:

$$D^{15} + D^{14} + D^{13} + D^{12} + D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

c) Tail bits:

Four tail bits equal to 0 are added to the information and parity bits, the result being a block of 338 bits $\{u(0), u(1), \dots, u(337)\}$:

$$\begin{aligned} u(k) &= u'(k) && \text{for } k = 0, 1, \dots, 5 \\ u(k) &= d(k-3) && \text{for } k = 6, 7, \dots, 317 \\ u(k) &= p(k-318) && \text{for } k = 318, 319, \dots, 333 \\ u(k) &= 0 && \text{for } k = 334, 335, 336, 337 \text{ (tail bits)} \end{aligned}$$

5.1.3.3 Convolutional encoder

This block of 338 bits $\{u(0), u(1), \dots, u(337)\}$ is encoded with the 1/2 rate convolutional code (identical to the one used for TCH/FS) defined by the polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

This results in a block of 676 coded bits: $\{C(0), C(1), \dots, C(675)\}$ defined by:

$$\begin{aligned} C(2k) &= u(k) + u(k-3) + u(k-4) \\ C(2k+1) &= u(k) + u(k-1) + u(k-3) + u(k-4) \text{ for } k = 0, 1, \dots, 337; u(k) = 0 \text{ for } k < 0 \end{aligned}$$

The code is punctured in such a way that the following coded bits:

$$\{C(3+6j) \text{ and } C(5+6j) \text{ for } j = 2, 3, \dots, 111\}$$
 are not transmitted

The result is a block of 456 coded bits, $\{c(0), c(1), \dots, c(455)\}$.

5.1.3.4 Interleaving

The interleaving is done as specified for SACCH in subclause 4.1.4.

5.1.3.5 Mapping on a burst

The mapping is given by the rule:

$$e(B, j) = i(B, j) \text{ and } e(B, 59+j) = i(B, 57+j) \text{ for } j = 0, 1, \dots, 56$$

and

$$e(B+m, 57) = q(2m) \text{ and } e(B+m, 58) = q(2m+1) \text{ for } m = 0, 1, 2, 3$$

where

$$q(0), q(1), \dots, q(7) = 0, 0, 1, 0, 0, 0, 0, 1 \text{ identifies the coding scheme CS-3.}$$

5.1.4 Packet data block type 4 (CS-4)

5.1.4.1 Block constitution

The message delivered to the encoder has a fixed size of 431 information bits $\{d(0), d(1), \dots, d(430)\}$. It is delivered on a burst mode.

5.1.4.2 Block code

a) USF precoding:

The first three bits $d(0), d(1), d(2)$ are block coded into twelve bits $u'(0), u'(1), \dots, u'(11)$ according to the following table:

| $d(0), d(1), d(2)$ | $u'(0), u'(1), \dots, u'(11)$ |
|--------------------|-------------------------------|
| 000 | 000 000 000 000 |
| 001 | 000 011 011 101 |
| 010 | 001 101 110 110 |
| 011 | 001 110 101 011 |
| 100 | 110 100 001 011 |
| 101 | 110 111 010 110 |
| 110 | 111 001 111 101 |
| 111 | 111 010 100 000 |

b) Parity bits:

Sixteen parity bits $p(0), p(1), \dots, p(15)$ are defined in such a way that in GF(2) the binary polynomial:

$d(0)D^{446} + \dots + d(430)D^{16} + p(0)D^{15} + \dots + p(15)$, when divided by:

$D^{16} + D^{12} + D^5 + 1$, yields a remainder equal to:

$D^{15} + D^{14} + D^{13} + D^{12} + D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1$.

The result is a block of 456 coded bits, $\{c(0), c(1), \dots, c(455)\}$:

$c(k) = u'(k) \quad \text{for } k = 0, 1, \dots, 11$

$c(k) = d(k-9) \quad \text{for } k = 12, 13, \dots, 439$

$c(k) = p(k-440) \quad \text{for } k = 440, 441, \dots, 455$

5.1.4.3 Convolutional encoder

No convolutional coding is done.

5.1.4.4 Interleaving

The interleaving is done as specified for SACCH in section 4.1.4.

5.1.4.5 Mapping on a burst

The mapping is given by the rule:

$e(B, j) = i(B, j) \quad \text{and} \quad e(B, 59+j) = i(B, 57+j) \quad \text{for } j = 0, 1, \dots, 56$

and

$e(B+m, 57) = q(2m) \quad \text{and} \quad e(B+m, 58) = q(2m+1) \quad \text{for } m = 0, 1, 2, 3$

where

$q(0), q(1), \dots, q(7) = 0, 0, 0, 1, 0, 1, 1, 0$ identifies the coding scheme CS-4.

5.1.5 Packet data block type 5 (MCS-1)

5.1.5.1 Downlink (MCS-1 DL)

5.1.5.1.1 Block constitution

The message delivered to the encoder has a fixed size of 209 information bits $\{d(0),d(1),\dots,d(208)\}$. It is delivered on a burst mode.

5.1.5.1.2 USF precoding

The first three bits $d(0),d(1),d(2)$ are block coded into twelve bits $u'(0),u'(1),\dots,u'(11)$ as for Packet data block type 4 (CS-4) in subclause 5.1.4.2.

5.1.5.1.3 Header coding

a) Parity bits:

Eight header parity bits $p(0),p(1),\dots,p(7)$ are defined in such a way that in GF(2) the binary polynomial:

$d(3)D^{35} + \dots + d(30)D^8 + p(0)D^7 + \dots + p(7)$, when divided by:

$D^8 + D^6 + D^3 + 1$, yields a remainder equal to:

$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1$.

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 42 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(35)\}$ with six negative indexes:

$u''(k-6) = p(k+2)$ for $k = 0,1,\dots,5$

$u''(k) = d(k+3)$ for $k = 0,1,\dots,27$

$u''(k) = p(k-28)$ for $k = 28,29,\dots,35$

c) Convolutional encoder

This block of 42 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(35)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$G_4 = 1 + D^2 + D^3 + D^5 + D^6$

$G_7 = 1 + D + D^2 + D^3 + D^6$

$G_5 = 1 + D + D^4 + D^6$

This results in a block of 108 coded bits: $\{C(0),C(1),\dots,C(107)\}$ defined by:

$C(3k) = u''(k) + u''(k-2) + u''(k-3) + u''(k-5) + u''(k-6)$

$C(3k+1) = u''(k) + u''(k-1) + u''(k-2) + u''(k-3) + u''(k-6)$

$C(3k+2) = u''(k) + u''(k-1) + u''(k-4) + u''(k-6)$ for $k = 0,1,\dots,35$

The code is punctured in such a way that the following coded bits:

$\{C(2+3j)$ for $j = 0,1,\dots,35\}$ as well as $\{C(k)$ for $k = 34,58,82,106\}$ are not transmitted

The result is a block of 68 coded bits, $\{hc(0),hc(1),\dots,hc(67)\}$.

5.1.5.1.4 Data coding

a) Parity bits:

Twelve data parity bits $p(0), p(1), \dots, p(11)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$d(31)D^{189} + \dots + d(208)D^{12} + p(0)D^{11} + \dots + p(11)$, when divided by:

$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1$, yields a remainder equal to:

$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1$.

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 196 bits $\{u(0), u(1), \dots, u(195)\}$:

$$u(k) = d(k+31) \quad \text{for } k = 0, 1, \dots, 177$$

$$u(k) = p(k-178) \quad \text{for } k = 178, 179, \dots, 189$$

$$u(k) = 0 \quad \text{for } k = 190, 191, \dots, 195 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 196 bits $\{u(0), u(1), \dots, u(195)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 588 coded bits: $\{C(0), C(1), \dots, C(587)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0, 1, \dots, 195; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Two puncturing schemes named P1 or P2 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | $\{C(2+21j), C(5+21j), C(8+21j), C(10+21j), C(11+21j), C(14+21j), C(17+21j), C(20+21j)\}$ for $j = 0, 1, \dots, 27$ are not transmitted except $\{C(k) \text{ for } k = 73, 136, 199, 262, 325, 388, 451, 514\}$ which are transmitted |
| P2 | $\{C(1+21j), C(4+21j), C(7+21j), C(9+21j), C(13+21j), C(15+21j), C(16+21j), C(19+21j)\}$ for $j = 0, 1, \dots, 27$ are not transmitted except $\{C(k) \text{ for } k = 78, 141, 204, 267, 330, 393, 456, 519\}$ which are transmitted |

The result is a block of 372 coded bits, $\{dc(0), dc(1), \dots, dc(371)\}$.

5.1.5.1.5 Interleaving

The USF, header and data are put together as one entity as described by the following rule:

$$c(k) = u'(k) \quad \text{for } k = 0, 1, \dots, 11$$

$$c(k) = hc(k-12) \quad \text{for } k = 12, 13, \dots, 79$$

$$c(k) = dc(k-80) \quad \text{for } k = 80, 81, \dots, 451$$

$$c'(n, k) = c(n, k) \quad \text{for } k = 0, 1, \dots, 24$$

$$c'(n, k) = c(n, k-1) \quad \text{for } k = 26, 27, \dots, 81$$

$$c'(n,k) = c(n,k-2) \quad \text{for } k = 83,84,\dots,138$$

$$c'(n,k) = c(n,k-3) \quad \text{for } k = 140,141,\dots,423$$

$$c'(n,k) = c(n,k-4) \quad \text{for } k = 425,426,\dots,455$$

$$c'(n,25) = q(8) \quad c'(n,82) = q(9) \quad c'(n,139) = q(10) \quad c'(n,424) = q(11)$$

$c(n,k)$ are the coded bits and $q(8),q(9),\dots,q(11) = 0,0,0,0$ are four extra stealing flags

The resulting block is interleaved according to the following rule:

$$i(B,j) = c'(n,k) \quad \text{for } k = 0,1,\dots,455$$

$$n = 0,1,\dots,N,N+1,\dots$$

$$B = B_0 + 4n + (k \bmod 4)$$

$$j = 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4)$$

5.1.5.1.6 Mapping on a burst

The mapping is given by the rule:

$$e(B,j) = i(B,j) \quad \text{and} \quad e(B,59+j) = i(B,57+j) \quad \text{for } j = 0,1,\dots,56$$

and

$$e(B+m,57) = q(2m) \quad \text{and} \quad e(B+m,58) = q(2m+1) \quad \text{for } m = 0,1,2,3$$

where

$q(0),q(1),\dots,q(7) = 0,0,0,1,0,1,1,0$ identifies the coding scheme as CS-4.

5.1.5.2 Uplink (MCS-1 UL)

5.1.5.2.1 Block constitution

The message delivered to the encoder has a fixed size of 209 information bits $\{d(0),d(1),\dots,d(208)\}$. It is delivered on a burst mode.

5.1.5.2.2 Header coding

a) Parity bits:

Eight header parity bits $p(0),p(1),\dots,p(7)$ are defined in such a way that in GF(2) the binary polynomial:

$$d(0)D^{38} + \dots + d(30)D^8 + p(0)D^7 + \dots + p(7), \text{ when divided by:}$$

$$D^8 + D^6 + D^3 + 1, \text{ yields a remainder equal to:}$$

$$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 45 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(38)\}$ with six negative indexes:

$$u''(k-6) = p(k+2) \quad \text{for } k = 0,1,\dots,5$$

$$u''(k) = d(k) \quad \text{for } k = 0,1,\dots,30$$

$$u''(k) = p(k-31) \quad \text{for } k = 31,32,\dots,38$$

c) Convolutional encoder

This block of 45 bits $\{u^{(-6)}, \dots, u^{(0)}, u^{(1)}, \dots, u^{(38)}\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_7 = 1 + D + D^2 + D^3 + D^6$$

$$G_5 = 1 + D + D^4 + D^6$$

This results in a block of 117 coded bits: $\{C(0), C(1), \dots, C(116)\}$ defined by:

$$C(3k) = u^{(k)} + u^{(k-2)} + u^{(k-3)} + u^{(k-5)} + u^{(k-6)}$$

$$C(3k+1) = u^{(k)} + u^{(k-1)} + u^{(k-2)} + u^{(k-3)} + u^{(k-6)}$$

$$C(3k+2) = u^{(k)} + u^{(k-1)} + u^{(k-4)} + u^{(k-6)} \quad \text{for } k = 0, 1, \dots, 38$$

The code is punctured in such a way that the following coded bits:

$$\{C(5+12j), C(8+12j), C(11+12j), \text{ for } j = 0, 1, \dots, 8\} \text{ as well as } \{C(k) \text{ for } k = 26, 38, 50, 62, 74, 86, 98, 110, 113, 116\}$$
 are not transmitted

The result is a block of 80 coded bits, $\{hc(0), hc(1), \dots, hc(79)\}$.

5.1.5.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.5.1.4.

5.1.5.2.4 Interleaving

The header and data are put together as one entity as described by the following rule:

$$c(k) = hc(k) \quad \text{for } k = 0, 1, \dots, 79$$

$$c(k) = dc(k-80) \quad \text{for } k = 80, 81, \dots, 451$$

$$c'(n, k) = c(n, k) \quad \text{for } k = 0, 1, \dots, 24$$

$$c'(n, k) = c(n, k-1) \quad \text{for } k = 26, 27, \dots, 81$$

$$c'(n, k) = c(n, k-2) \quad \text{for } k = 83, 84, \dots, 138$$

$$c'(n, k) = c(n, k-3) \quad \text{for } k = 140, 141, \dots, 423$$

$$c'(n, k) = c(n, k-4) \quad \text{for } k = 425, 426, \dots, 455$$

$$c'(n, 25) = q(8) \quad c'(n, 82) = q(9) \quad c'(n, 10) = q(2) \quad c'(n, 424) = q(11)$$

$c(n, k)$ are the coded bits and $q(8), q(9), \dots, q(11) = 0, 0, 0, 0$ are four extra stealing flags

The resulting block is interleaved according to the following rule:

$$i(B, j) = c'(n, k) \quad \text{for } k = 0, 1, \dots, 455$$

$$n = 0, 1, \dots, N, N+1, \dots$$

$$B = B_0 + 4n + (k \bmod 4)$$

$$j = 2((49k) \bmod 57) + ((k \bmod 8) \text{ div } 4)$$

5.1.5.2.5 Mapping on a burst

The mapping is the same as for MCS-1 DL as specified in subclause 5.1.5.1.6.

5.1.6 Packet data block type 6 (MCS-2)

5.1.6.1 Downlink (MCS-2 DL)

5.1.6.1.1 Block constitution

The message delivered to the encoder has a fixed size of 257 information bits $\{d(0),d(1),\dots,d(256)\}$. It is delivered on a burst mode.

5.1.6.1.2 USF precoding

The first three bits $d(0),d(1),d(2)$ are block coded into twelve bits $u'(0),u'(1),\dots,u'(11)$ as for Packet data block type 4 (CS-4) in subclause 5.1.4.2.

5.1.6.1.3 Header coding

A block of 68 coded bits $\{hc(0),hc(1),\dots,hc(67)\}$ is derived from $\{d(3),d(4),\dots,d(30)\}$ as described for MCS-1 DL in subclause 5.1.5.1.3.

5.1.6.1.4 Data coding

a) Parity bits:

Twelve data parity bits $p(0),p(1),\dots,p(11)$ are defined in such a way that in GF(2) the binary polynomial:

$d(31)D^{237} + \dots + d(256)D^{12} + p(0)D^{11} + \dots + p(11)$, when divided by:

$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1$, yields a remainder equal to:

$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1$.

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 244 bits $\{u(0),u(1),\dots,u(243)\}$:

$u(k) = d(k+31)$ for $k = 0,1,\dots,225$

$u(k) = p(k-226)$ for $k = 226,227,\dots,237$

$u(k) = 0$ for $k = 238,239,\dots,243$ (tail bits)

c) Convolutional encoder

This block of 244 bits $\{u(0),u(1),\dots,u(243)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$G_4 = 1 + D^2 + D^3 + D^5 + D^6$

$G_7 = 1 + D + D^2 + D^3 + D^6$

$G_5 = 1 + D + D^4 + D^6$

This results in a block of 732 coded bits: $\{C(0),C(1),\dots,C(731)\}$ defined by:

$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$

$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$

$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6)$ for $k = 0,1,\dots,243$; $u(k) = 0$ for $k < 0$

The code is punctured depending on the value of the CPS field as defined in 04.60. Two puncturing schemes named P1 or P2 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | $\{C(6j), C(1+6j), C(5+6j) \text{ for } j = 0, 1, \dots, 121\}$ and $\{C(k) \text{ for } k = 57, 171, 285, 399, 513, 627\}$ are transmitted |
| P2 | $\{C(2+6j), C(3+6j), C(4+6j) \text{ for } j = 0, 1, \dots, 121\}$ and $\{C(k) \text{ for } k = 108, 222, 336, 450, 564, 678\}$ are transmitted |

The result is a block of 372 coded bits, $\{dc(0), dc(1), \dots, dc(371)\}$.

5.1.6.1.5 Interleaving

The interleaving is done as specified for MCS-1 DL in subclause 5.1.5.1.5.

5.1.6.1.6 Mapping on a burst

The mapping is done as specified for MCS-1 DL in subclause 5.1.5.1.6.

5.1.6.2 Uplink (MCS-2 UL)

5.1.6.2.1 Block constitution

The message delivered to the encoder has a fixed size of 257 information bits $\{d(0), d(1), \dots, d(256)\}$. It is delivered on a burst mode.

5.1.6.2.2 Header coding

A block of 80 coded bits $\{hc(0), hc(1), \dots, hc(79)\}$ is derived from $\{d(0), d(1), \dots, d(30)\}$ as described for MCS-1 UL in subclause 5.1.5.2.2.

5.1.6.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.6.1.4..

5.1.6.2.4 Interleaving

The interleaving is the same as for MCS-1 UL as specified in subclause 5.1.5.2.4..

5.1.6.2.5 Mapping on a burst

The mapping is the same as for MCS-1 DL as specified in subclause 5.1.5.1.6.

5.1.7 Packet data block type 7 (MCS-3)

5.1.7.1 Downlink (MCS-3 DL)

5.1.7.1.1 Block constitution

The message delivered to the encoder has a fixed size of 329 information bits $\{d(0), d(1), \dots, d(328)\}$. It is delivered on a burst mode. This packet data block can be reused to retransmit MCS-8 blocks, the size of the corresponding block is 305 bits. This block shall be padded with zeroes to acquire the proper 329-bit size.

5.1.7.1.2 USF precoding

The first three bits $d(0), d(1), d(2)$ are block coded into twelve bits $u'(0), u'(1), \dots, u'(11)$ as for Packet data block type 4 (CS-4) in subclause 5.1.4.2.

5.1.7.1.3 Header coding

A block of 68 coded bits {hc(0),hc(1),...,hc(67)} is derived from {d(3),d(4),...,d(30)} as described for MCS-1 DL in subclause 5.1.5.1.3.

5.1.7.1.4 Data coding

a) Parity bits:

Twelve data parity bits p(0),p(1),...,p(11) are defined in such a way that in GF(2) the binary polynomial:

$$d(31)D^{309} + \dots + d(328)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 316 bits {u(0),u(1),...,u(315)}:

$$u(k) = d(k+31) \quad \text{for } k = 0,1,\dots,297$$

$$u(k) = p(k-298) \quad \text{for } k = 298,299,\dots,309$$

$$u(k) = 0 \quad \text{for } k = 310,311,\dots,315 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 316 bits {u(0),u(1),...,u(315)} is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 948 coded bits: {C(0),C(1),...,C(947)} defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0,1,\dots,315; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Three puncturing schemes named P1, P2 or P3 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | {C(18j), C(1+18j), C(3+18j), C(6+18j), C(10+18j), C(14+18j), C(17+18j) for j = 0,1,...,51} and {C(k) for k = 241,475,709, 936,937,939,942,946 } are transmitted |
| P2 | {C(2+18j), C(5+18j), C(6+18j), C(7+18j), C(9+18j), C(12+18j), C(16+18j) for j = 0,1,...,51} and {C(k) for k = 121,355,589, 938,941,942,943,945 } are transmitted |
| P3 | {C(18j), C(4+18j), C(8+18j), C(11+18j), C(12+18j), C(13+18j), C(15+18j) for j = 0,1,...,51} and {C(k) for k = 181,289,523,811, 936,940,944,947 } are transmitted |

The result is a block of 372 coded bits, {dc(0),dc(1),...,dc(371)}.

5.1.7.1.5 Interleaving

The interleaving is done as specified for MCS-1 DL in subclause 5.1.5.1.5.

5.1.7.1.6 Mapping on a burst

The mapping is done as specified for MCS-1 DL in subclause 5.1.5.1.6.

5.1.7.2 Uplink (MCS-3 UL)

5.1.7.2.1 Block constitution

The message delivered to the encoder has a fixed size of 329 information bits $\{d(0),d(1),\dots,d(328)\}$. It is delivered on a burst mode. This packet data block can be reused to retransmit MCS-8 blocks, the size of the corresponding block is 305 bits. This block shall be padded with zeroes to acquire the proper 329-bit size.

5.1.7.2.2 Header coding

A block of 80 coded bits $\{hc(0),hc(1),\dots,hc(79)\}$ is derived from $\{d(0),d(1),\dots,d(30)\}$ as described for MCS-1 UL in subclause 5.1.5.2.2.

5.1.7.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.7.1.4..

5.1.7.2.4 Interleaving

The interleaving is the same as for MCS-1 UL as specified in subclause 5.1.5.2.4.

5.1.7.2.5 Mapping on a burst

The mapping is the same as for MCS-1 DL as specified in subclause 5.1.5.1.6.

5.1.8 Packet data block type 8 (MCS-4)

5.1.8.1 Downlink (MCS-4 DL)

5.1.8.1.1 Block constitution

The message delivered to the encoder has a fixed size of 385 information bits $\{d(0),d(1),\dots,d(384)\}$. It is delivered on a burst mode.

5.1.8.1.2 USF precoding

The first three bits $d(0),d(1),d(2)$ are block coded into twelve bits $u'(0),u'(1),\dots,u'(11)$ as for Packet data block type 4 (CS-4) in subclause 5.1.4.2.

5.1.8.1.3 Header coding

A block of 68 coded bits $\{hc(0),hc(1),\dots,hc(67)\}$ is derived from $\{d(3),d(4),\dots,d(30)\}$ as described for MCS-1 DL in subclause 5.1.5.1.3.

5.1.8.1.4 Data coding

a) Parity bits:

Twelve data parity bits $p(0),p(1),\dots,p(11)$ are defined in such a way that in GF(2) the binary polynomial:

$$d(31)D^{365} + \dots + d(384)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 372 bits $\{u(0),u(1),\dots,u(371)\}$:

$$u(k) = d(k+31) \quad \text{for } k = 0,1,\dots,353$$

$$u(k) = p(k-354) \quad \text{for } k = 354,355,\dots,365$$

$$u(k) = 0 \quad \text{for } k = 366,367,\dots,371 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 372 bits $\{u(0),u(1),\dots,u(371)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 1116 coded bits: $\{C(0),C(1),\dots,C(1115)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0,1,\dots, 371; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Three puncturing schemes named P1, P2 or P3 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | $\{C(3j) \text{ for } j = 0,1,\dots,371\}$ are transmitted |
| P2 | $\{C(1+3j) \text{ for } j = 0,1,\dots,371\}$ are transmitted |
| P3 | $\{C(2+3j) \text{ for } j = 0,1,\dots,371\}$ are transmitted |

The result is a block of 372 coded bits, $\{dc(0),dc(1),\dots,dc(371)\}$.

5.1.8.1.5 Interleaving

The interleaving is done as specified for MCS-1 DL in subclause 5.1.5.1.5.

5.1.8.1.6 Mapping on a burst

The mapping is done as specified for MCS-1 DL in subclause 5.1.5.1.6.

5.1.8.2 Uplink (MCS-4 UL)

5.1.8.2.1 Block constitution

The message delivered to the encoder has a fixed size of 385 information bits $\{d(0),d(1),\dots,d(384)\}$. It is delivered on a burst mode.

5.1.8.2.2 Header coding

A block of 80 coded bits $\{hc(0),hc(1),\dots,hc(79)\}$ is derived from $\{d(0),d(1),\dots,d(30)\}$ as described for MCS-1 UL in subclause 5.1.5.2.2.

5.1.8.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.8.1.4.

5.1.8.2.4 Interleaving

The interleaving is the same as for MCS-1 UL as specified in subclause 5.1.5.2.4.

5.1.8.2.5 Mapping on a burst

The mapping is the same as for MCS-1 DL as specified in subclause 5.1.5.1.6.

5.1.9 Packet data block type 9 (MCS-5)

5.1.9.1 Downlink (MCS-5 DL)

5.1.9.1.1 Block constitution

The message delivered to the encoder has a fixed size of 478 information bits {d(0),d(1),...,d(477)}. It is delivered on a burst mode.

5.1.9.1.2 USF precoding

The first three bits d(0),d(1),d(2) are block coded into 36 bits u'(0),u'(1),...,u'(35) according to the following table:

| d(0),d(1),d(2) | u'(0),u'(1),...,u'(35) | | | |
|----------------|------------------------|---------------------|---------------------|---------------------|
| | burst 0 | burst 1 | burst 2 | burst 3 |
| 000 | 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 |
| 001 | 1 1 1 1 1 0 0 0 0 0 | 1 1 1 1 1 0 0 0 0 0 | 1 1 1 1 1 1 0 0 0 0 | 1 1 1 1 1 0 0 0 0 1 |
| 010 | 1 1 1 0 0 1 1 1 1 0 | 1 1 1 0 1 1 1 1 0 0 | 1 1 0 0 0 0 1 1 0 | 1 1 0 0 0 1 1 0 0 |
| 011 | 1 0 0 1 1 1 1 1 0 0 | 1 1 0 0 0 0 0 1 1 | 1 0 1 1 1 0 1 1 1 | 0 0 1 0 0 1 1 1 1 |
| 100 | 0 0 0 1 1 0 0 1 1 | 0 0 1 0 1 1 0 1 0 | 1 0 0 0 0 1 1 0 1 | 1 1 1 1 1 1 1 1 0 |
| 101 | 1 1 0 1 0 1 0 1 1 | 0 0 0 1 1 0 1 0 1 | 0 1 1 1 0 1 0 1 1 | 1 0 0 1 0 1 0 1 1 |
| 110 | 0 0 1 0 0 1 1 0 1 | 1 0 1 1 1 1 1 1 1 | 0 1 1 0 1 0 0 0 1 | 0 0 1 1 1 0 1 0 0 |
| 111 | 0 1 1 0 1 0 1 1 1 | 0 1 0 1 0 1 1 1 1 | 0 0 0 1 1 1 1 1 0 | 0 1 0 0 1 0 0 1 1 |

5.1.9.1.3 Header coding

a) Parity bits:

Eight header parity bits p(0),p(1),...,p(7) are defined in such a way that in GF(2) the binary polynomial:

$$d(3)D^{32} + \dots + d(27)D^8 + p(0)D^7 + \dots + p(7), \text{ when divided by:}$$

$$D^8 + D^6 + D^3 + 1, \text{ yields a remainder equal to:}$$

$$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 39 bits {u''(-6),...,u''(0),u''(1),...,u''(32)} with six negative indexes:

$$u''(k-6) = p(k+2) \quad \text{for } k = 0, 1, \dots, 5$$

$$u''(k) = d(k+3) \quad \text{for } k = 0, 1, \dots, 24$$

$$u''(k) = p(k-25) \quad \text{for } k = 25, 26, \dots, 32$$

c) Convolutional encoder

This block of 39 bits {u''(-6),...,u''(0),u''(1),...,u''(32)} is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 99 coded bits: $\{C(0), C(1), \dots, C(98)\}$ defined by:

$$C(3k) = u''(k) + u''(k-2) + u''(k-3) + u''(k-5) + u''(k-6)$$

$$C(3k+1) = u''(k) + u''(k-1) + u''(k-2) + u''(k-3) + u''(k-6)$$

$$C(3k+2) = u''(k) + u''(k-1) + u''(k-2) + u''(k-6) \quad \text{for } k = 0, 1, \dots, 32$$

A spare bit is added at the end of this block:

$$hc(k) = C(k) \quad \text{for } k = 0, 1, \dots, 98$$

$$hc(99) = C(98)$$

The result is a block of 100 coded bits, $\{hc(0), hc(1), \dots, hc(99)\}$.

5.1.9.1.4 Data coding

a) Parity bits:

Twelve data parity bits $p(0), p(1), \dots, p(11)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(28)D^{461} + \dots + d(477)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 468 bits $\{u(0), u(1), \dots, u(467)\}$:

$$u(k) = d(k+28) \quad \text{for } k = 0, 1, \dots, 449$$

$$u(k) = p(k-450) \quad \text{for } k = 450, 451, \dots, 461$$

$$u(k) = 0 \quad \text{for } k = 462, 463, \dots, 467 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 468 bits $\{u(0), u(1), \dots, u(467)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 1404 coded bits: $\{C(0), C(1), \dots, C(1403)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0, 1, \dots, 467; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Two puncturing schemes named P1 or P2 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | {C(2+9j) for j = 0,1,...,153} as well as {C(1388+3j) for j = 0,1,...,5} are not transmitted except {C(k) for k = 47,371,695,1019} which are transmitted |
| P2 | {C(1+9j) for j = 0,1,...,153} as well as {C(1387+3j) for j = 0,1,...,5} are not transmitted except {C(k) for k = 136,460,784,1108} which are transmitted |

The result is a block of 1248 coded bits, {dc(0),dc(1),...,dc(1247)}.

5.1.9.1.5 Interleaving

a) Header

The 100 coded bits of the header, {hc(0),hc(1),...,hc(99)}, are interleaved according to the following rule:

$$h_i(j) = hc(k) \quad \text{for } k = 0,1,\dots,99$$

$$j = 25(k \bmod 4) + ((17k) \bmod 25)$$

b) Data

There is no closed expression describing the interleaver, but it has been derived taking the following approach:

1. A block interleaver with a 1392 bit block size is defined:

The kth input data bit is mapped to the jth bit of the Bth burst, where

$$k = 0, \dots, 1391$$

$$B = \text{mod}(k, 4)$$

$$d = \text{mod}(k, 464)$$

$$j = 3 * (2 \bmod(25d, 58) + \text{div}(\text{mod}(d, 8), 4) + 2(-1)^B \text{div}(d, 232)) + \text{mod}(k, 3)$$

2. The data bit positions being mapped onto header positions in the interleaved block are removed (the header positions are $j = 156, 157, \dots, 191$ when the header is placed next to the training sequence. This leaves 1248 bits in the mapping.
3. The bits are renumbered to fill out the gaps both in j and k , without changing the relative order

The resulting interleaver transform the block of 1248 coded bits, {dc(0),dc(1),...,dc(1247)} into a block of 1248 interleaved bits, {di(0),di(1),...,di(1247)}.

$$d_i(j') = dc(k') \quad \text{for } k' = 0,1,\dots,1247$$

(An explicit relation between j' and k' is given in table 15)

5.1.9.1.6 Mapping on a burst

a) Straightforward Mapping

The mapping is given by the rule:

For $B=0,1,2,3$, let

$$e(B,j) = d_i(312B+j) \quad \text{for } j = 0,1,\dots,155$$

$$e(B,j) = h_i(25B+j-156) \quad \text{for } j = 156,157,\dots,167$$

$$e(B,j) = u'(9B+j-168) \quad \text{for } j = 168,169,\dots,173$$

$$e(B,j) = q(2B+j-174) \quad \text{for } j = 174,175$$

$$e(B,j) = u'(9B+j-170) \quad \text{for } j = 176,177,178$$

$$e(B,j) = h_i(25B+j-167) \quad \text{for } j = 179,180,\dots,191$$

$$e(B,j) = d_i(312B+j-36) \quad \text{for } j = 192,193,\dots,347$$

where

$$q(0),q(1),\dots,q(7) = 0,0,0,0,0,0,0,0 \text{ identifies the coding scheme MCS-5 or MCS-6.}$$

b) Bit swapping

After this mapping the following bits are swapped:

For $B = 0,1,2,3$,

Swap $e(B,142)$ with $e(B,155)$

Swap $e(B,144)$ with $e(B,158)$

Swap $e(B,145)$ with $e(B,161)$

Swap $e(B,147)$ with $e(B,164)$

Swap $e(B,148)$ with $e(B,167)$

Swap $e(B,150)$ with $e(B,170)$

Swap $e(B,151)$ with $e(B,173)$

Swap $e(B,176)$ with $e(B,195)$

Swap $e(B,179)$ with $e(B,196)$

Swap $e(B,182)$ with $e(B,198)$

Swap $e(B,185)$ with $e(B,199)$

Swap $e(B,188)$ with $e(B,201)$

Swap $e(B,191)$ with $e(B,202)$

Swap $e(B,194)$ with $e(B,204)$.

5.1.9.2 Uplink (MCS-5 UL)

5.1.9.2.1 Block constitution

The message delivered to the encoder has a fixed size of 487 information bits $\{d(0),d(1),\dots,d(486)\}$. It is delivered on a burst mode.

5.1.9.2.2 Header coding

a) Parity bits:

Eight header parity bits $p(0),p(1),\dots,p(7)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(0)D^{44} + \dots + d(36)D^8 + p(0)D^7 + \dots + p(7), \text{ when divided by:}$$

$$D^8 + D^6 + D^3 + 1, \text{ yields a remainder equal to:}$$

$$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 51 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(44)\}$ with six negative indexes:

$$u''(k-6) = p(k+2) \quad \text{for } k = 0,1,\dots,5$$

$$u''(k) = d(k) \quad \text{for } k = 0, 1, \dots, 36$$

$$u''(k) = p(k-37) \quad \text{for } k = 37, 38, \dots, 44$$

c) Convolutional encoder

This block of 51 bits $\{u''(-6), \dots, u''(0), u''(1), \dots, u''(44)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 135 coded bits: $\{C(0), C(1), \dots, C(134)\}$ defined by:

$$C(3k) = u''(k) + u''(k-2) + u''(k-3) + u''(k-5) + u''(k-6)$$

$$C(3k+1) = u''(k) + u''(k-1) + u''(k-2) + u''(k-3) + u''(k-6)$$

$$C(3k+2) = u''(k) + u''(k-1) + u''(k-4) + u''(k-6) \quad \text{for } k = 0, 1, \dots, 44$$

The code is punctured in such a way that the following coded bits:

$$hc(k) = C(k) \quad \text{for } k = 0, 1, \dots, 134$$

$$hc(135) = C(134)$$

The result is a block of 136 coded bits, $\{hc(0), hc(1), \dots, hc(135)\}$.

5.1.9.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.9.1.4 where bits $\{d(28), d(29), \dots, d(477)\}$ are replaced by bits $\{d(37), d(38), \dots, d(486)\}$.

5.1.9.2.4 Interleaving

a) Header

The 136 coded bits of the header, $\{hc(0), hc(1), \dots, hc(135)\}$, are interleaved according to the following rule:

$$hi(j) = hc(k) \quad \text{for } k = 0, 1, \dots, 135$$

$$j = 34(k \bmod 4) + 2((11k) \bmod 17) + [(k \bmod 8)/4]$$

b) Data

The data interleaving is the same as for MCS-5 DL as specified in subclause 5.1.9.1.5.

5.1.9.2.5 Mapping on a burst

a) Straightforward Mapping

The mapping is given by the rule:

For $B=0, 1, 2, 3$, let

$$e(B, j) = di(312B+j) \quad \text{for } j = 0, 1, \dots, 155$$

$$e(B, j) = hi(34B+j-156) \quad \text{for } j = 156, 157, \dots, 173$$

$$e(B, j) = q(2B+j-174) \quad \text{for } j = 174, 175$$

$$e(B, j) = hi(34B+j-158) \quad \text{for } j = 176, 177, \dots, 191$$

$$e(B,j) = d_i(312B+j-36) \quad \text{for } j = 192,193,\dots,347$$

where

$$q(0),q(1),\dots,q(7) = 0,0,0,0,0,0,0,0 \text{ identifies the coding scheme MCS-5 or MCS-6.}$$

b) Bit swapping

The bit swapping is the same as for MCS-5 DL as specified in subclause 5.1.9.1.6.

5.1.10 Packet data block type 10 (MCS-6)

5.1.10.1 Downlink (MCS-6 DL)

5.1.10.1.1 Block constitution

The message delivered to the encoder has a fixed size of 622 information bits $\{d(0),d(1),\dots,d(621)\}$. It is delivered on a burst mode. This packet data block can be reused to retransmit MCS-8 blocks, the size of the corresponding block is 575 bits. This block shall be padded with zeroes to acquire the proper 622-bit size.

5.1.10.1.2 USF precoding

A block of 36 bits $\{u'(0),u'(1),\dots,u'(35)\}$ is derived from $\{d(0),d(1),d(2)\}$ as described for MCS-5 DL in subclause 5.1.9.1.2.

5.1.10.1.3 Header coding

A block of 100 coded bits $\{hc(0),hc(1),\dots,hc(99)\}$ is derived from $\{d(3),d(4),\dots,d(27)\}$ as described for MCS-5 DL in subclause 5.1.9.1.3.

5.1.10.1.4 Data coding

a) Parity bits:

Twelve data parity bits $p(0),p(1),\dots,p(11)$ are defined in such a way that in GF(2) the binary polynomial:

$$d(28)D^{605} + \dots + d(621)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 612 bits $\{u(0),u(1),\dots,u(611)\}$:

$$u(k) = d(k+28) \quad \text{for } k = 0,1,\dots,593$$

$$u(k) = p(k-594) \quad \text{for } k = 594,595,\dots,605$$

$$u(k) = 0 \quad \text{for } k = 606,607,\dots,611 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 612 bits $\{u(0),u(1),\dots,u(611)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_7 = 1 + D + D^2 + D^3 + D^6$$

$$G_5 = 1 + D + D^4 + D^6$$

This results in a block of 1836 coded bits: {C(0),C(1),...,C(1835)} defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0,1,\dots,611; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Two puncturing schemes named P1 or P2 are applied in such a way that the following coded bits:

| | |
|----|--|
| P1 | {C(2+3j) for j = 0,1,...,611} are not transmitted except {C(k) for k = 32,98,164,230,296,428,494,560,626,692,824,890,956,1022,1088,1220,1286,1352,1418,1484,1616,1682,1748,1814} which are transmitted |
| P2 | {C(1+3j) for j = 0,1,...,611} are not transmitted except {C(k) for k = 16,82,148,214,280,412,478,544,610,676,808,874,940,1006,1072,1204,1270,1336,1402,1468,1600,1666,1732,1798} which are transmitted |

The result is a block of 1248 coded bits, {dc(0),dc(1),...,dc(1247)}.

5.1.10.1.5 Interleaving

The interleaving is done as specified for MCS-5 DL in subclause 5.1.9.1.5.

5.1.10.1.6 Mapping on a burst

The mapping is done as specified for MCS-5 DL in subclause 5.1.9.1.6.

5.1.10.2 Uplink (MCS-6 UL)

5.1.10.2.1 Block constitution

The message delivered to the encoder has a fixed size of 631 information bits {d(0),d(1),...,d(630)}. It is delivered on a burst mode. This packet data block can be reused to retransmit MCS-8 blocks, the size of the corresponding block is 584 bits. This block shall be padded with zeroes to acquire the proper 631-bit size.

5.1.10.2.2 Header coding

A block of 136 coded bits {hc(0),hc(1),...,hc(135)} is derived from {d(0),d(1),...,d(36)} as described for MCS-5 UL in subclause 5.1.9.2.2.

5.1.10.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.10.1.4 where bits {d(28),d(29),...,d(621)} are replaced by bits {d(37),d(38),...,d(630)}.

5.1.10.2.4 Interleaving

The interleaving is the same as for MCS-5 UL as specified in subclause 5.1.9.2.4.

5.1.10.2.5 Mapping on a burst

The mapping is the same as for MCS-5 UL as specified in subclause 5.1.9.2.5.

5.1.11 Packet data block type 11 (MCS-7)

5.1.11.1 Downlink (MCS-7 DL)

5.1.11.1.1 Block constitution

The message delivered to the encoder has a fixed size of 940 information bits $\{d(0),d(1),\dots,d(939)\}$. It is delivered on a burst mode.

5.1.11.1.2 USF precoding

A block of 36 bits $\{u'(0),u'(1),\dots,u'(35)\}$ is derived from $\{d(0),d(1),d(2)\}$ as described for MCS-5 DL in subclause 5.1.9.1.2.

5.1.11.1.3 Header coding

a) Parity bits:

Eight header parity bits $p(0),p(1),\dots,p(7)$ are defined in such a way that in GF(2) the binary polynomial:

$d(3)D^{44} + \dots + d(39)D^8 + p(0)D^7 + \dots + p(7)$, when divided by:

$D^8 + D^6 + D^3 + 1$, yields a remainder equal to:

$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1$.

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 51 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(44)\}$ with six negative indexes:

$u''(k-6) = p(k+2)$ for $k = 0,1,\dots,5$

$u''(k) = d(k+3)$ for $k = 0,1,\dots,36$

$u''(k) = p(k-37)$ for $k = 37,38,\dots,44$

c) Convolutional encoder

This block of 51 bits $\{u''(-6),\dots,u''(0),u''(1),\dots,u''(44)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$G_4 = 1 + D^2 + D^3 + D^5 + D^6$

$G_7 = 1 + D + D^2 + D^3 + D^6$

$G_5 = 1 + D + D^4 + D^6$

This results in a block of 135 coded bits: $\{C(0),C(1),\dots,C(134)\}$ defined by:

$C(3k) = u''(k) + u''(k-2) + u''(k-3) + u''(k-5) + u''(k-6)$

$C(3k+1) = u''(k) + u''(k-1) + u''(k-2) + u''(k-3) + u''(k-6)$

$C(3k+2) = u''(k) + u''(k-1) + u''(k-4) + u''(k-6)$ for $k = 0,1,\dots,44$

The code is punctured in such a way that the following coded bits:

$\{C(k)$ for $k = 14,23,33,50,59,69,86,95,105,122,131\}$ are not transmitted

The result is a block of 124 coded bits, $\{hc(0),hc(1),\dots,hc(123)\}$.

5.1.11.1.4 Data coding

I) First half:

a) Parity bits:

Twelve data parity bits $p(0), p(1), \dots, p(11)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(40)D^{461} + \dots + d(489)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 468 bits $\{u(0), u(1), \dots, u(467)\}$:

$$u(k) = d(k+40) \quad \text{for } k = 0, 1, \dots, 449$$

$$u(k) = p(k-450) \quad \text{for } k = 450, 451, \dots, 461$$

$$u(k) = 0 \quad \text{for } k = 462, 463, \dots, 467 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 468 bits $\{u(0), u(1), \dots, u(467)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 1404 coded bits: $\{C(0), C(1), \dots, C(1403)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0, 1, \dots, 467; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Three puncturing schemes named P1, P2 or P3 are applied in such a way that the following coded bits:

| | |
|----|---|
| P1 | $\{C(18j), C(1+18j), C(4+18j), C(8+18j), C(11+18j), C(12+18j), C(13+18j), C(15+18j)$ for $j = 0, 1, \dots, 77\}$ are transmitted except $\{C(k) \text{ for } k = 1, 19, 37, 235, 415, 595, 775, 955, 1135, 1351, 1369, 1387\}$ which are not transmitted |
| P2 | $\{C(2+18j), C(3+18j), C(5+18j), C(6+18j), C(10+18j), C(14+18j), C(16+18j), C(17+18j)$ for $j = 0, 1, \dots, 77\}$ are transmitted except $\{C(k) \text{ for } k = 16, 34, 52, 196, 376, 556, 736, 916, 1096, 1366, 1384, 1402\}$ which are not transmitted |
| P3 | $\{C(2+18j), C(5+18j), C(6+18j), C(7+18j), C(9+18j), C(12+18j), C(13+18j), C(16+18j)$ for $j = 0, 1, \dots, 77\}$ are transmitted except $\{C(k) \text{ for } k = 13, 31, 49, 301, 481, 661, 841, 1021, 1201, 1363, 1381, 1399\}$ which are not transmitted |

The result is a block of 612 coded bits, $\{c1(0), c1(1), \dots, c1(611)\}$.

II) Second half:

The same data coding as for first half is proceeded with bits $\{d(40), d(41), \dots, d(489)\}$ replaced by bits $\{d(490), d(491), \dots, d(939)\}$. The result is a block of 612 coded bits, $\{c2(0), c2(1), \dots, c2(611)\}$.

5.1.11.1.5 Interleaving

a) Header

The 124 coded bits of the header, $\{hc(0),hc(1),\dots,hc(123)\}$, are interleaved according to the following rule:

$$hi(j) = hc(k) \quad \text{for } k = 0,1,\dots,123$$

$$j = 31(k \bmod 4) + ((17k) \bmod 31)$$

b) Data

Data are put together as one entity as described by the following rule:

$$dc(k) = c1(k) \quad \text{for } k = 0,1,\dots,611$$

$$dc(k) = c2(k-612) \quad \text{for } k = 612,613,\dots,1223$$

The resulting block is interleaved according to the following rule:

$$di(j) = dc(k) \quad \text{for } k = 0,1,\dots,1223$$

$$j = 306(k \bmod 4) + 3((44k) \bmod 102 + (k \operatorname{div} 4) \bmod 2) + (k + 2 - (k \operatorname{div} 408)) \bmod 3$$

5.1.11.1.6 Mapping on a burst

a) Straightforward Mapping

The mapping is given by the rule:

For $B=0,1,2,3$, let

$$e(B,j) = di(306B+j) \quad \text{for } j = 0,1,\dots,152$$

$$e(B,j) = hi(31B+j-153) \quad \text{for } j = 153,154,\dots,167$$

$$e(B,j) = u'(9B+j-168) \quad \text{for } j = 168,169,\dots,173$$

$$e(B,j) = q(2B+j-174) \quad \text{for } j = 174,175$$

$$e(B,j) = u'(9B+j-170) \quad \text{for } j = 176,177,178$$

$$e(B,j) = hi(31B+j-164) \quad \text{for } j = 179,180,\dots,194$$

$$e(B,j) = di(306B+j-42) \quad \text{for } j = 195,196,\dots,347$$

where

$$q(0),q(1),\dots,q(7) = 1,1,1,0,0,1,1,1 \text{ identifies the coding scheme MCS-7, MCS-8 or MCS-9.}$$

b) Bit swapping

The bit swapping is the same as for MCS-5 DL as specified in subclause 5.1.9.1.6.

5.1.11.2 Uplink (MCS-7 UL)

5.1.11.2.1 Block constitution

The message delivered to the encoder has a fixed size of 946 information bits $\{d(0),d(1),\dots,d(945)\}$. It is delivered on a burst mode.

5.1.11.2.2 Header coding

a) Parity bits:

Eight header parity bits $p(0), p(1), \dots, p(7)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(0)D^{53} + \dots + d(45)D^8 + p(0)D^7 + \dots + p(7), \text{ when divided by:}$$

$$D^8 + D^6 + D^3 + 1, \text{ yields a remainder equal to:}$$

$$D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail biting:

The six last header parity bits are added before information and parity bits, the result being a block of 60 bits $\{u''(-6), \dots, u''(0), u''(1), \dots, u''(53)\}$ with six negative indexes:

$$u''(k-6) = p(k+2) \quad \text{for } k = 0, 1, \dots, 5$$

$$u''(k) = d(k) \quad \text{for } k = 0, 1, \dots, 45$$

$$u''(k) = p(k-46) \quad \text{for } k = 46, 47, \dots, 53$$

c) Convolutional encoder

This block of 60 bits $\{u''(-6), \dots, u''(0), u''(1), \dots, u''(53)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G_4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G_7 = 1 + D + D^2 + D^3 + D^6$$

$$G_5 = 1 + D + D^4 + D^6$$

This results in a block of 162 coded bits: $\{C(0), C(1), \dots, C(161)\}$ defined by:

$$C(3k) = u''(k) + u''(k-2) + u''(k-3) + u''(k-5) + u''(k-6)$$

$$C(3k+1) = u''(k) + u''(k-1) + u''(k-2) + u''(k-3) + u''(k-6)$$

$$C(3k+2) = u''(k) + u''(k-1) + u''(k-4) + u''(k-6) \quad \text{for } k = 0, 1, \dots, 53$$

The code is punctured in such a way that the following coded bits:

$$\{C(k) \text{ for } k = 35, 131\} \text{ are not transmitted}$$

The result is a block of 160 coded bits, $\{hc(0), hc(1), \dots, hc(159)\}$.

5.1.11.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.11.1.4 where bits $\{d(40), d(41), \dots, d(939)\}$ are replaced by bits $\{d(46), d(47), \dots, d(945)\}$.

5.1.11.2.4 Interleaving

a) Header

The 160 coded bits of the header, $\{hc(0), hc(1), \dots, hc(159)\}$, are interleaved according to the following rule:

$$hi(j) = hc(k) \quad \text{for } k = 0, 1, \dots, 159$$

$$j = 40(k \bmod 4) + 2((13(k \operatorname{div} 8)) \bmod 20) + ((k \bmod 8) \operatorname{div} 4)$$

b) Data

The data interleaving is the same as for MCS-7 DL as specified in subclause 5.1.11.1.5.

5.1.11.2.5 Mapping on a burst

a) Straightforward Mapping

The mapping is given by the rule:

For $B=0,1,2,3$, let

$$e(B,j) = d_i(306B+j) \quad \text{for } j = 0,1,\dots,152$$

$$e(B,j) = h_i(40B+j-153) \quad \text{for } j = 153,154,\dots,173$$

$$e(B,j) = q(2B+j-174) \quad \text{for } j = 174,175$$

$$e(B,j) = h_i(40B+j-155) \quad \text{for } j = 176,177,\dots,194$$

$$e(B,j) = d_i(306B+j-42) \quad \text{for } j = 195,196,\dots,347$$

where

$q(0),q(1),\dots,q(7) = 1,1,1,0,0,1,1,1$ identifies the coding scheme MCS-7, MCS-8 or MCS-9.

b) Bit swapping

The bit swapping is the same as for MCS-5 DL as specified in subclause 5.1.9.1.6.

5.1.12 Packet data block type 12 (MCS-8)

5.1.12.1 Downlink (MCS-8 DL)

5.1.12.1.1 Block constitution

The message delivered to the encoder has a fixed size of 1132 information bits $\{d(0),d(1),\dots,d(1131)\}$. It is delivered on a burst mode.

5.1.12.1.2 USF precoding

A block of 36 bits $\{u'(0),u'(1),\dots,u'(35)\}$ is derived from $\{d(0),d(1),d(2)\}$ as described for MCS-5 DL in subclause 5.1.9.1.2.

5.1.12.1.3 Header coding

A block of 124 coded bits $\{hc(0),hc(1),\dots,hc(123)\}$ is derived from $\{d(3),d(4),\dots,d(39)\}$ as described for MCS-7 DL in subclause 5.1.11.1.3.

5.1.12.1.4 Data coding

I) First half:

a) Parity bits:

Twelve data parity bits $p(0),p(1),\dots,p(11)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(40)D^{557} + \dots + d(585)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 564 bits $\{u(0),u(1),\dots,u(563)\}$:

$$u(k) = d(k+40) \quad \text{for } k = 0,1,\dots,545$$

$$u(k) = p(k-546) \quad \text{for } k = 546,547,\dots,557$$

$$u(k) = 0 \quad \text{for } k = 558,559,\dots,563 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 564 bits $\{u(0),u(1),\dots,u(563)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 1692 coded bits: $\{C(0),C(1),\dots,C(1691)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0,1,\dots,563; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Three puncturing schemes named P1, P2 or P3 are applied in such a way that the following coded bits:

| | |
|----|---|
| P1 | $\{C(36j), C(2+36j), C(5+36j), C(6+36j), C(10+36j), C(13+36j), C(16+36j), C(20+36j), C(23+36j), C(24+36j), C(27+36j), C(31+36j), C(35+36j), \text{ for } j = 0,1,\dots,46\}$ as well as $\{C(845)\}$ are transmitted |
| P2 | $\{C(1+36j), C(4+36j), C(8+36j), C(11+36j), C(12+36j), C(15+36j), C(17+36j), C(19+36j), C(22+36j), C(25+36j), C(28+36j), C(30+36j), C(33+36j), \text{ for } j = 0,1,\dots,46\}$ as well as $\{C(582)\}$ are transmitted |
| P3 | $\{C(2+36j), C(3+36j), C(7+36j), C(9+36j), C(14+36j), C(17+36j), C(18+36j), C(21+36j), C(26+36j), C(27+36j), C(29+36j), C(32+36j), C(34+36j), \text{ for } j = 0,1,\dots,46\}$ as well as $\{C(1156)\}$ are transmitted |

The result is a block of 612 coded bits, $\{c1(0),c1(1),\dots,c1(611)\}$.

II) Second half:

The same data coding as for first half is proceeded with bits $\{d(40),d(41),\dots,d(585)\}$ replaced by bits $\{d(586),d(587),\dots,d(1131)\}$. The result is a block of 612 coded bits, $\{c2(0),c2(1),\dots,c2(611)\}$.

5.1.12.1.5 Interleaving

a) Header

The header interleaving is the same as for MCS-7 DL as specified in subclause 5.1.11.1.5.

b) Data

Data are put together as one entity as described by the following rule:

$$dc(k) = c1(k) \quad \text{for } k = 0,1,\dots,611$$

$$dc(k) = c2(k-612) \quad \text{for } k = 612,613,\dots,1223$$

The resulting block is interleaved according to the following rule:

$$di(j) = dc(k) \quad \text{for } k = 0,1,\dots,1223$$

$$j = 306(2(k \text{ div } 612) + (k \text{ mod } 2) + 3((74k) \text{ mod } 102 + (k \text{ div } 2) \text{ mod } 2) + (k + 2 - (k \text{ div } 204)) \text{ mod } 3$$

5.1.12.1.6 Mapping on a burst

The mapping is the same as for MCS-7 DL as specified in subclause 5.1.11.1.6.

5.1.12.2 Uplink (MCS-8 UL)

5.1.12.2.1 Block constitution

The message delivered to the encoder has a fixed size of 1138 information bits $\{d(0),d(1),\dots,d(1137)\}$. It is delivered on a burst mode.

5.1.12.2.2 Header coding

A block of 160 coded bits $\{hc(0),hc(1),\dots,hc(159)\}$ is derived from $\{d(0),d(1),\dots,d(45)\}$ as described for MCS-7 UL in subclause 5.1.11.2.2.

5.1.12.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.12.1.4 where bits $\{d(40),d(41),\dots,d(1131)\}$ are replaced by bits $\{d(46),d(47),\dots,d(1137)\}$.

5.1.12.2.4 Interleaving

a) Header

The header interleaving is the same as for MCS-7 UL as specified in subclause 5.1.11.2.4.

b) Data

The data interleaving is the same as for MCS-8 DL as specified in subclause 5.1.12.1.5.

5.1.12.2.5 Mapping on a burst

The mapping is the same as for MCS-7 UL as specified in subclause 5.1.11.2.5.

5.1.13 Packet data block type 13 (MCS-9)

5.1.13.1 Downlink (MCS-9 DL)

5.1.13.1.1 Block constitution

The message delivered to the encoder has a fixed size of 1228 information bits $\{d(0),d(1),\dots,d(1227)\}$. It is delivered on a burst mode.

5.1.13.1.2 USF precoding

A block of 36 bits $\{u'(0),u'(1),\dots,u'(35)\}$ is derived from $\{d(0),d(1),d(2)\}$ as described for MCS-5 DL in subclause 5.1.9.1.2.

5.1.13.1.3 Header coding

A block of 124 coded bits $\{hc(0),hc(1),\dots,hc(123)\}$ is derived from $\{d(3),d(4),\dots,d(39)\}$ as described for MCS-7 DL in subclause 5.1.11.1.3.

5.1.13.1.4 Data coding

I) First half:

a) Parity bits:

Twelve data parity bits $p(0), p(1), \dots, p(11)$ are defined in such a way that in $GF(2)$ the binary polynomial:

$$d(40)D^{605} + \dots + d(633)D^{12} + p(0)D^{11} + \dots + p(11), \text{ when divided by:}$$

$$D^{12} + D^{11} + D^{10} + D^8 + D^5 + D^4 + 1, \text{ yields a remainder equal to:}$$

$$D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D^2 + D + 1.$$

b) Tail bits:

Six tail bits equal to 0 are added to the information and parity bits, the result being a block of 612 bits $\{u(0), u(1), \dots, u(611)\}$:

$$u(k) = d(k+40) \quad \text{for } k = 0, 1, \dots, 593$$

$$u(k) = p(k-594) \quad \text{for } k = 594, 595, \dots, 605$$

$$u(k) = 0 \quad \text{for } k = 606, 607, \dots, 611 \text{ (tail bits)}$$

c) Convolutional encoder

This block of 612 bits $\{u(0), u(1), \dots, u(611)\}$ is encoded with the 1/3 rate convolutional mother code defined by the polynomials:

$$G4 = 1 + D^2 + D^3 + D^5 + D^6$$

$$G7 = 1 + D + D^2 + D^3 + D^6$$

$$G5 = 1 + D + D^4 + D^6$$

This results in a block of 1836 coded bits: $\{C(0), C(1), \dots, C(1835)\}$ defined by:

$$C(3k) = u(k) + u(k-2) + u(k-3) + u(k-5) + u(k-6)$$

$$C(3k+1) = u(k) + u(k-1) + u(k-2) + u(k-3) + u(k-6)$$

$$C(3k+2) = u(k) + u(k-1) + u(k-4) + u(k-6) \quad \text{for } k = 0, 1, \dots, 611; u(k) = 0 \text{ for } k < 0$$

The code is punctured depending on the value of the CPS field as defined in 04.60. Three puncturing schemes named P1, P2 or P3 are applied in such a way that the following coded bits:

| | |
|----|---|
| P1 | $\{C(3j) \text{ for } j = 0, 1, \dots, 611\}$ are transmitted |
| P2 | $\{C(1+3j) \text{ for } j = 0, 1, \dots, 611\}$ are transmitted |
| P3 | $\{C(2+3j) \text{ for } j = 0, 1, \dots, 611\}$ are transmitted |

The result is a block of 612 coded bits, $\{c1(0), c1(1), \dots, c1(611)\}$.

II) Second half:

The same data coding as for first half is proceeded with bits $\{d(40), d(41), \dots, d(633)\}$ replaced by bits $\{d(634), d(635), \dots, d(1227)\}$. The result is a block of 612 coded bits, $\{c2(0), c2(1), \dots, c2(611)\}$.

5.1.13.1.5 Interleaving

The interleaving is the same as for MCS-8 DL as specified in subclause 5.1.12.1.5.

5.1.13.1.6 Mapping on a burst

The mapping is the same as for MCS-7 DL as specified in subclause 5.1.11.1.6.

5.1.13.2 Uplink (MCS-9 UL)

5.1.13.2.1 Block constitution

The message delivered to the encoder has a fixed size of 1234 information bits $\{d(0),d(1),\dots,d(1233)\}$. It is delivered on a burst mode.

5.1.13.2.2 Header coding

A block of 160 coded bits $\{hc(0),hc(1),\dots,hc(159)\}$ is derived from $\{d(0),d(1),\dots,d(45)\}$ as described for MCS-7 UL in subclause 5.1.11.2.2.

5.1.13.2.3 Data coding

The data coding is the same as for downlink as specified in subclause 5.1.13.1.4 where bits $\{d(40),d(41),\dots,d(1227)\}$ are replaced by bits $\{d(46),d(47),\dots,d(1233)\}$.

5.1.13.2.4 Interleaving

The interleaving is the same as for MCS-8 UL as specified in subclause 5.1.12.2.4.

5.1.13.2.5 Mapping on a burst

The mapping is the same as for MCS-7 UL as specified in subclause 5.1.11.2.5.

5.2 Packet control channels (PACCH, PBCCH, PAGCH, PPCH, PNCH, PTCCH, CPBCCH, CPAGCH, CPPCH, and CPNCH)

The coding scheme used for PACCH, PBCCH, PAGCH, PPCH, PNCH, downlink PTCCH, CPBCCH, CPAGCH, CPPCH, and CPNCH is the same as for SACCH as specified in section 4.1.

The coding scheme used for uplink PTCCH is the same as for PRACH as specified in section 5.3.

5.3 Packet random access channel (PRACH and CPRACH)

Two coding schemes are specified for access bursts on the packet switched channels. The packet access burst containing 8 information bits and the extended packet access burst containing 11 information bits. Only the 11 information bits access burst may be transmitted on the CPRACH.

5.3.1 Packet Access Burst

The encoding of this burst is as defined in section 4.6 for the random access channel (RACH). The BSIC used shall be the BSIC of the BTS to which the burst is intended.

5.3.2 Extended Packet Access Burst

The burst carrying the extended packet random access uplink message contains 11 information bits $d(0),d(1),\dots,d(10)$.

Six parity bits $p(0),p(1),\dots,p(5)$ are defined in such a way that in GF(2) the binary polynomial:

$$d(0)D^{16} + \dots + d(10)D^6 + p(0)D^5 + \dots + p(5), \text{ when divided by } D^6 + D^5 + D^3 + D^2 + D + 1 \text{ yields a remainder equal to } D^5 + D^4 + D^3 + D^2 + D + 1.$$

The six bits of the BSIC, $\{B(0),B(1),\dots,B(5)\}$, of the BTS to which the Random Access is intended, are added bitwise modulo 2 to the six parity bits, $\{p(0),p(1),\dots,p(5)\}$. This results in six colour bits, $C(0)$ to $C(5)$ defined as $C(k) = b(k) + p(k)$ ($k = 0$ to 5) where:

$b(0)$ = MSB of PLMN colour code

$b(5)$ = LSB of BS colour code.

This defines $\{u(0), u(1), \dots, u(20)\}$ by:

$$u(k) = d(k) \quad \text{for } k = 0, 1, \dots, 10$$

$$u(k) = C(k-11) \quad \text{for } k = 11, 12, \dots, 16$$

$$u(k) = 0 \quad \text{for } k = 17, 18, 19, 20 \text{ (tail bits)}$$

The coded bits $\{c(0), c(1), \dots, c(41)\}$ are obtained by the same convolutional code of rate 1/2 as for TCH/FS, defined by the polynomials:

$$G_0 = 1 + D^3 + D^4$$

$$G_1 = 1 + D + D^3 + D^4$$

and with:

$$c(2k) = u(k) + u(k-3) + u(k-4)$$

$$c(2k+1) = u(k) + u(k-1) + u(k-3) + u(k-4) \quad \text{for } k = 0, 1, \dots, 20 ; u(k) = 0 \text{ for } k < 0$$

The code is punctured in such a way that the following coded bits:

$c(0)$, $c(2)$, $c(5)$, $c(37)$, $c(39)$, $c(41)$ are not transmitted.

This results in a block of 36 coded bits, $\{e(0), e(1), \dots, e(35)\}$.

5.4 Access Burst on packet switched channels other than PRACH and CPRACH

The encoding of this burst is as defined in section 5.3 for the packet random access channel (PRACH). The BSIC used shall be the BSIC of the BTS to which the burst is intended.

Table 1: Reordering and partitioning of a coded block of 456 bits into 8 sub-blocks

| k mod 8= | 0 | 1 | 2 | 3 | k mod 8= | 4 | 5 | 6 | 7 |
|----------|-----|-----|-----|-----|----------|-----|-----|-----|-----|
| j=0 | k=0 | 57 | 114 | 171 | j=1 | 228 | 285 | 342 | 399 |
| 2 | 64 | 121 | 178 | 235 | 3 | 292 | 349 | 406 | 7 |
| 4 | 128 | 185 | 242 | 299 | 5 | 356 | 413 | 14 | 71 |
| 6 | 192 | 249 | 306 | 363 | 7 | 420 | 21 | 78 | 135 |
| 8 | 256 | 313 | 370 | 427 | 9 | 28 | 85 | 142 | 199 |
| 10 | 320 | 377 | 434 | 35 | 11 | 92 | 149 | 206 | 263 |
| | 384 | 441 | 42 | 99 | | 156 | 213 | 270 | 327 |
| | 448 | 49 | 106 | 163 | | 220 | 277 | 334 | 391 |
| | 56 | 113 | 170 | 227 | | 284 | 341 | 398 | 455 |
| | 120 | 177 | 234 | 291 | | 348 | 405 | 6 | 63 |
| 20 | 184 | 241 | 298 | 355 | 21 | 412 | 13 | 70 | 127 |
| | 248 | 305 | 362 | 419 | | 20 | 77 | 134 | 191 |
| | 312 | 369 | 426 | 27 | | 84 | 141 | 198 | 255 |
| | 376 | 433 | 34 | 91 | | 148 | 205 | 262 | 319 |
| | 440 | 41 | 98 | 155 | | 212 | 269 | 326 | 383 |
| 30 | 48 | 105 | 162 | 219 | 31 | 276 | 333 | 390 | 447 |
| | 112 | 169 | 226 | 283 | | 340 | 397 | 454 | 55 |
| | 176 | 233 | 290 | 347 | | 404 | 5 | 62 | 119 |
| | 240 | 297 | 354 | 411 | | 12 | 69 | 126 | 183 |
| | 304 | 361 | 418 | 19 | | 76 | 133 | 190 | 247 |
| 40 | 368 | 425 | 26 | 83 | 41 | 140 | 197 | 254 | 311 |
| | 432 | 33 | 90 | 147 | | 204 | 261 | 318 | 375 |
| | 40 | 97 | 154 | 211 | | 268 | 325 | 382 | 439 |
| | 104 | 161 | 218 | 275 | | 332 | 389 | 446 | 47 |
| | 168 | 225 | 282 | 339 | | 396 | 453 | 54 | 111 |
| 50 | 232 | 289 | 346 | 403 | 51 | 4 | 61 | 118 | 175 |
| | 296 | 353 | 410 | 11 | | 68 | 125 | 182 | 239 |
| | 360 | 417 | 18 | 75 | | 132 | 189 | 246 | 303 |
| | 424 | 25 | 82 | 139 | | 196 | 253 | 310 | 367 |
| | 32 | 89 | 146 | 203 | | 260 | 317 | 374 | 431 |
| 60 | 96 | 153 | 210 | 267 | 61 | 324 | 381 | 438 | 39 |
| | 160 | 217 | 274 | 331 | | 388 | 445 | 46 | 103 |
| | 224 | 281 | 338 | 395 | | 452 | 53 | 110 | 167 |
| | 288 | 345 | 402 | 3 | | 60 | 117 | 174 | 231 |
| | 352 | 409 | 10 | 67 | | 124 | 181 | 238 | 295 |
| 70 | 416 | 17 | 74 | 131 | 71 | 188 | 245 | 302 | 359 |
| | 24 | 81 | 138 | 195 | | 252 | 309 | 366 | 423 |
| | 88 | 145 | 202 | 259 | | 316 | 373 | 430 | 31 |
| | 152 | 209 | 266 | 323 | | 380 | 437 | 38 | 95 |
| | 216 | 273 | 330 | 387 | | 444 | 45 | 102 | 159 |
| 80 | 280 | 337 | 394 | 451 | 81 | 52 | 109 | 166 | 223 |
| | 344 | 401 | 2 | 59 | | 116 | 173 | 230 | 287 |
| | 408 | 9 | 66 | 123 | | 180 | 237 | 294 | 351 |
| | 16 | 73 | 130 | 187 | | 244 | 301 | 358 | 415 |
| | 80 | 137 | 194 | 251 | | 308 | 365 | 422 | 23 |
| 90 | 144 | 201 | 258 | 315 | 91 | 372 | 429 | 30 | 87 |
| | 208 | 265 | 322 | 379 | | 436 | 37 | 94 | 151 |
| | 272 | 329 | 386 | 443 | | 44 | 101 | 158 | 215 |
| | 336 | 393 | 450 | 51 | | 108 | 165 | 222 | 279 |
| | 400 | 1 | 58 | 115 | | 172 | 229 | 286 | 343 |
| 100 | 8 | 65 | 122 | 179 | 101 | 236 | 293 | 350 | 407 |
| | 72 | 129 | 186 | 243 | | 300 | 357 | 414 | 15 |
| | 136 | 193 | 250 | 307 | | 364 | 421 | 22 | 79 |
| | 200 | 257 | 314 | 371 | | 428 | 29 | 86 | 143 |
| | 264 | 321 | 378 | 435 | | 36 | 93 | 150 | 207 |
| 110 | 328 | 385 | 442 | 43 | 111 | 100 | 157 | 214 | 271 |
| 112 | 392 | 449 | 50 | 107 | 113 | 164 | 221 | 278 | 335 |

**Table 2: Subjective importance of encoded bits for the full rate speech TCH
(Parameter names and bit indices refer to GSM 06.10)**

| Importance class | Parameter name | Parameter number | Bit index | Label | Class |
|------------------|------------------------|------------------|-----------|----------------------|------------------------------|
| 1 | Log area ratio 1 | 1 | 5 | d0 d1, d2, d3, d4 | 1 with parity check |
| | block amplitude | 12,29,46,63 | 5 | | |
| 2 | Log area ratio 1 | 1 | 4 | ...d48, d49 d50 | |
| | Log area ratio 2 | 2 | 5 | | |
| | Log area ratio 3 | 3 | 4 | | |
| 3 | Log area ratio 1 | 1 | 3 | | |
| | Log area ratio 2 | 2 | 4 | | |
| | Log area ratio 3 | 3 | 3 | | |
| | Log area ratio 4 | 4 | 4 | | |
| | LPT lag | 9,26,43,60 | 6 | | |
| | block amplitude | 12,29,43,63 | 4 | | |
| | Log area ratio 2,5,6 | 2,5,6 | 3 | | |
| | LPT lag | 9,26,43,60 | 5 | | |
| | LPT lag | 9,26,43,60 | 4 | | |
| | LPT lag | 9,26,43,60 | 3 | | |
| LPT lag | 9,26,43,60 | 2 | | | |
| 4 | block amplitude | 12,29,43,63 | 3 | | |
| | Log area ratio 1 | 1 | 2 | | |
| | Log area ratio 4 | 4 | 3 | | |
| | Log area ratio 7 | 7 | 2 | | |
| | LPT lag | 9,26,43,60 | 1 | | |
| | Log area ratio 5,6 | 5,6 | 2 | | |
| | LPT gain | 10,27,44,61 | 1 | | |
| | LPT lag | 9,26,43,60 | 0 | | |
| | Grid position | 11,28,45,62 | 1 | | |
| 5 | Log area ratio 1 | 1 | 1 | ...d181 d182 | 1 with parity check |
| | Log area ratio 2,3,8,4 | 2,3,8,4 | 2 | | |
| | Log area ratio 5,7 | 5,7 | 1 | | |
| | LPT gain | 10,27,44,61 | 0 | | |
| | block amplitude | 12,29,43,63 | 2 | | |
| | RPE pulses | 13..25 | 2 | | |
| | RPE pulses | 30..42 | 2 | | |
| | RPE pulses | 47..59 | 2 | | |
| | RPE pulses | 64..76 | 2 | | |
| | Grid position | 11,28,45,62 | 0 | | |
| | block amplitude | 12,29,43,63 | 1 | | |
| | RPE pulses | 13..25 | 1 | | |
| | RPE pulses | 30..42 | 1 | | |
| | RPE pulses | 47..59 | 1 | | |
| | RPE pulses | 64..67 | 1 | | |
| RPE pulses | 68..76 | 1 | | | |
| 6 | Log area ratio 1 | 1 | 0 | ...d259 | 2 |
| | Log area ratio 2,3,6 | 2,3,6 | 1 | | |
| | Log area ratio 7 | 7 | 0 | | |
| | Log area ratio 8 | 8 | 1 | | |
| | Log area ratio 8,3 | 8,3 | 0 | | |
| | Log area ratio 4 | 4 | 1 | | |
| | Log area ratio 4,5 | 4,5 | 0 | | |
| | block amplitude | 12,29,43,63 | 0 | | |
| | RPE pulses | 13..25 | 0 | | |
| | RPE pulses | 30..42 | 0 | | |
| | RPE pulses | 47..59 | 0 | | |
| | RPE pulses | 64..67 | 0 | | |
| | Log area ratio 2,6 | 2,6 | 0 | | |

Table 3a: Subjective importance of encoded bits for the half rate speech TCH for unvoiced speech frames (Parameter names and bit indices refer to GSM 06.20)

| Parameter name | Bit index | Label | Class |
|----------------|-----------|----------|----------------------------|
| R0 | 1 | d0 | |
| LPC 3 | 7 | d1 | |
| GSP 0-1 | 2 | d2 | |
| GSP 0-2 | 2 | d3 | |
| GSP 0-3 | 2 | d4 | |
| GSP 0-4 | 2 | d5 | |
| LPC 1 | 0 | d6 | |
| LPC 2 | 5...1 | d7...d11 | |
| LPC 3 | 6...1 | d12... | |
| Code 1-2 | 0 | | |
| Code 2-2 | 6...0 | | |
| Code 1-3 | 6...0 | | 1 |
| Code 2-3 | 6...3 | | |
| LPC3 | 0 | | without parity check |
| R0 | 0 | | |
| INT-LPC | 0 | | |
| Code 1-2 | 1...6 | | |
| Code 2-1 | 0...6 | | |
| Code 1-1 | 0...6 | | |
| GSP 0-4 | 0 | | |
| GSP 0-3 | 0 | | |
| GSP 0-2 | 0 | | |
| GSP 0-1 | 0 | | |
| LPC 2 | 0 | | |
| GSP 0-4 | 1 | | |
| GSP 0-3 | 1 | | |
| GSP 0-2 | 1 | | |
| GSP 0-1 | 1 | | |
| LPC 1 | 1...4 | ...d72 | |
| LPC 1 | 5 | d73... | |
| GSP 0-4 | 3 | | |
| GSP 0-3 | 3 | | |
| GSP 0-2 | 3 | | |
| GSP 0-1 | 3 | | |
| LPC2 | 6...8 | | 1 |
| GSP 0-4 | 4 | | |
| GSP 0-3 | 4 | | with parity check |
| GSP 0-2 | 4 | | |
| GSP 0-1 | 4 | | |
| LPC 1 | 6...9 | | |
| R0 | 2 | | |
| LPC 1 | 10 | | |
| R0 | 3,4 | | |
| Mode | 0,1 | ...d94 | |
| Code 2-4 | 0...6 | d95... | |
| Code 1-4 | 0...6 | | 2 |
| Code 2-3 | 0...2 | ...d111 | |

Table 3b: Subjective importance of encoded bits for the half rate speech TCH for voiced speech frames (Parameter names and bit indices refer to GSM 06.20)

| Parameter name | Bit index | Label | Class |
|----------------|-----------|--------|----------------------|
| LPC 1 | 2,1 | d0, d1 | |
| LPC 2 | 6...4 | d2... | |
| GSP 0-1 | 4 | | |
| GSP 0-2 | 4 | | |
| GSP 0-3 | 4 | | |
| GSP 0-4 | 4 | | |
| GSP 0-1 | 3 | | |
| GSP 0-2 | 3 | | |
| GSP 0-3 | 3 | | |
| GSP 0-4 | 3 | | |
| GSP 0-1 | 2 | | |
| GSP 0-2 | 2 | | |
| GSP 0-3 | 2 | | |
| GSP 0-4 | 2 | | |
| Code 1 | 8...0 | | |
| Code 2 | 8...5 | | |
| Code 2 | 2...0 | | |
| Code 3 | 8 | | |
| Code 2 | 4,3 | | |
| GSP 0-1 | 1 | | |
| GSP 0-2 | 1 | | |
| GSP 0-3 | 1 | | |
| GSP 0-4 | 1 | | 1 |
| GSP 0-1 | 0 | | |
| GSP 0-2 | 0 | | without parity check |
| GSP 0-3 | 0 | | |
| GSP 0-4 | 0 | | |
| INT-LPC | 0 | | |
| LPC 2 | 0 | | |
| LPC 3 | 0 | | |
| LAG 4 | 0 | | |
| LPC 3 | 1 | | |
| LPC 2 | 1 | | |
| LAG 4 | 1 | | |
| LAG 3 | 0 | | |
| LAG 2 | 0 | | |
| LAG 1 | 0 | | |
| LAG 4 | 2 | | |
| LAG 3 | 1 | | |
| LAG 2 | 1 | | |
| LAG 1 | 1 | | |
| LPC 3 | 2...4 | | |
| LPC 2 | 2 | | |
| LPC 3 | 5,6 | | |
| LPC 2 | 3 | | |
| R0 | 0 | | |
| LPC 3 | 7 | | |
| LPC 1 | 0 | | |
| LAG 4 | 3 | | |
| LAG 3 | 2 | | |
| LAG 2 | 2 | | |
| LAG 1 | 2 | | |
| R0 | 1 | ...d72 | |

| Parameter name | Bit index | Label | Class |
|----------------|-----------|---------|-------------------|
| LAG 3 | 3 | d73... | |
| LAG 2 | 3 | | |
| LAG 1 | 3,4 | | 1 |
| LPC 2 | 7,8 | | |
| LPC 1 | 3...6 | | with parity check |
| R0 | 2 | | |
| LAG 1 | 5...7 | | |
| LPC 1 | 7...10 | | |
| R0 | 3,4 | | |
| Mode | 0,1 | ...d94 | |
| Code 4 | 0...8 | d95... | 2 |
| Code 3 | 0...7 | ...d111 | |

Table 4: Reordering and partitioning of a coded block of 228 bits into 4 sub-blocks for TCH/HS

| b= | 0 | 1 | b= | 2 | 3 |
|-----------|----------|----------|-----------|----------|----------|
| i=0 | k=0 | 150 | i=1 | k=1 | 151 |
| 2 | 38 | 188 | 3 | 39 | 189 |
| 4 | 76 | 226 | 5 | 77 | 227 |
| 6 | 114 | 14 | 7 | 115 | 15 |
| 8 | 152 | 52 | 9 | 153 | 53 |
| 10 | 190 | 90 | 11 | 191 | 91 |
| | 18 | 128 | | 19 | 129 |
| | 56 | 166 | | 57 | 167 |
| | 94 | 204 | | 95 | 205 |
| | 132 | 32 | | 133 | 33 |
| 20 | 170 | 70 | 21 | 171 | 71 |
| | 208 | 108 | | 209 | 109 |
| | 8 | 146 | | 9 | 147 |
| | 46 | 184 | | 47 | 185 |
| | 84 | 222 | | 85 | 223 |
| 30 | 122 | 10 | 31 | 123 | 11 |
| | 160 | 48 | | 161 | 49 |
| | 198 | 86 | | 199 | 87 |
| | 28 | 124 | | 29 | 125 |
| | 66 | 162 | | 67 | 163 |
| 40 | 104 | 200 | 41 | 105 | 201 |
| | 142 | 30 | | 143 | 31 |
| | 180 | 68 | | 181 | 69 |
| | 218 | 106 | | 219 | 107 |
| | 4 | 144 | | 5 | 145 |
| 50 | 42 | 182 | 51 | 43 | 183 |
| | 80 | 220 | | 81 | 221 |
| | 118 | 6 | | 119 | 7 |
| | 156 | 44 | | 157 | 45 |
| | 194 | 82 | | 195 | 83 |
| 60 | 22 | 120 | 61 | 23 | 121 |
| | 60 | 158 | | 61 | 159 |
| | 98 | 196 | | 99 | 197 |
| | 136 | 24 | | 137 | 25 |
| | 174 | 62 | | 175 | 63 |
| 70 | 212 | 100 | 71 | 213 | 101 |
| | 12 | 138 | | 13 | 139 |
| | 50 | 176 | | 51 | 177 |
| | 88 | 214 | | 89 | 215 |
| | 126 | 2 | | 127 | 3 |
| 80 | 164 | 40 | 81 | 165 | 41 |
| | 202 | 78 | | 203 | 79 |
| | 34 | 116 | | 35 | 117 |
| | 72 | 154 | | 73 | 155 |
| | 110 | 192 | | 111 | 193 |
| 90 | 148 | 26 | 91 | 149 | 27 |
| | 186 | 64 | | 187 | 65 |
| | 224 | 102 | | 225 | 103 |
| | 16 | 140 | | 17 | 141 |
| | 54 | 178 | | 55 | 179 |
| 100 | 92 | 216 | 101 | 93 | 217 |
| | 130 | 20 | | 131 | 21 |
| | 168 | 58 | | 169 | 59 |
| | 206 | 96 | | 207 | 97 |
| | 36 | 134 | | 37 | 135 |
| 110 | 74 | 172 | 111 | 75 | 173 |
| 112 | 112 | 210 | 113 | 113 | 211 |

Table 5: Enhanced Full rate Source Encoder output parameters in order of occurrence and bit allocation within the speech frame of 244 bits/20 ms(Parameter names and bit indices refer to GSM 06.60)

| Bits (MSB-LSB) | Description |
|-----------------------|--|
| s1 - s7 | index of 1st LSF submatrix |
| s8 - s15 | index of 2nd LSF submatrix |
| s16 - s23 | index of 3rd LSF submatrix |
| s24 | sign of 3rd LSF submatrix |
| s25 - s32 | index of 4th LSF submatrix |
| s33 - s38 | index of 5th LSF submatrix |
| subframe 1 | |
| s39 - s47 | adaptive codebook index |
| s48 - s51 | adaptive codebook gain |
| s52 | sign information for 1st and 6th pulses |
| s53 - s55 | position of 1st pulse |
| s56 | sign information for 2nd and 7th pulses |
| s57 - s59 | position of 2nd pulse |
| s60 | sign information for 3rd and 8th pulses |
| s61 - s63 | position of 3rd pulse |
| s64 | sign information for 4th and 9th pulses |
| s65 - s67 | position of 4th pulse |
| s68 | sign information for 5th and 10th pulses |
| s69 - s71 | position of 5th pulse |
| s72 - s74 | position of 6th pulse |
| s75 - s77 | position of 7th pulse |
| s78 - s80 | position of 8th pulse |
| s81 - s83 | position of 9th pulse |
| s84 - s86 | position of 10th pulse |
| s87 - s91 | fixed codebook gain |
| subframe 2 | |
| s92 - s97 | adaptive codebook index (relative) |
| s98 - s141 | same description as s48 - s91 |
| subframe 3 | |
| s142 - s194 | same description as s39 - s91 |
| subframe 4 | |
| s195 - s244 | same description as s92 - s141 |

**Table 6: Ordering of enhanced full rate speech parameters for the channel encoder
(subjective importance of encoded bits) (after preliminary channel coding)
(Parameter names refers to GSM 06.60)**

| Description | Bits (Table 5) | Bit index within parameter |
|--|-------------------|--------------------------------|
| CLASS 1a: 50 bits (protected by 3 bit TCH-FS CRC) | | |
| LTP-LAG 1 | w39 - w44 | b8, b7, b6, b5, b4, b3 |
| LTP-LAG 3 | w146 - w151 | b8, b7, b6, b5, b4, b3 |
| LTP-LAG 2 | w94 - w95 | b5, b4 |
| LTP-LAG 4 | w201 - w202 | b5, b4 |
| LTP-GAIN 1 | n48 | b3 |
| FCB-GAIN 1 | w89 | b4 |
| LTP-GAIN 2 | w100 | b3 |
| FCB-GAIN 2 | w141 | b4 |
| LTP-LAG 1 | w45 | b2 |
| LTP-LAG 3 | w152 | b2 |
| LTP-LAG 2 | w96 | b3 |
| LTP-LAG 4 | w203 | b3 |
| LPC 1 | w2 - w3 | b5, b4 |
| LPC 2 | w8 | b7 |
| LPC 2 | w10 | b5 |
| LPC 3 | w18 - w19 | b6, b5 |
| LPC 3 | w24 | b0 |
| LTP-LAG 1 | w46 - w47 | b1, b0 |
| LTP-LAG 3 | w153 - w154 | b1, b0 |
| LTP-LAG 2 | w97 | b2 |
| LTP-LAG 4 | w204 | b2 |
| LPC 1 | w4 - w5 | b3, b2 |
| LPC 2 | w11 - w12 | b4, b3 |
| LPC 3 | w16 | b8 |
| LPC 2 | w9 | b6 |
| LPC 1 | w6 - w7 | b1, b0 |
| LPC 2 | w13 | b2 |
| LPC 3 | w17 | b7 |
| LPC 3 | w20 | b4 |
| LTP-LAG 2 | w98 | b1 |
| LTP-LAG 4 | w205 | b1 |
| CLASS 1b: 132 bits (protected) | | |
| LPC 1 | w1 | b6 |
| LPC 2 | w14 - w15 | b1, b0 |
| LPC 3 | w21 | b3 |
| LPC 4 | w25 - w26 | b7, b6 |
| LPC 4 | w28 | b4 |
| LTP-GAIN 3 | w155 | b3 |
| LTP-GAIN 4 | w207 | b3 |
| FCB-GAIN 3 | w196 | b4 |
| FCB-GAIN 4 | w248 | b4 |
| FCB-GAIN 1 | w90 | b3 |
| FCB-GAIN 2 | w142 | b3 |
| FCB-GAIN 3 | w197 | b3 |
| FCB-GAIN 4 | w249 | b3 |
| CRC-POLY | w253 - w260 | b7, b6, b5, b4, b3, b2, b1, b0 |
| LTP-GAIN 1 | w49 | b2 |
| (continued) | | |

**Table 6 (continued): Ordering of enhanced full rate speech parameters for the channel encoder
(subjective importance of encoded bits) (after preliminary channel coding)
(Parameter names refers to GSM 06.60)**

| Description | Bits (Table 5) | Bit index within parameter |
|-------------|-------------------|----------------------------|
| LTP-GAIN 2 | w101 | b2 |
| LTP-GAIN 3 | w156 | b2 |
| LTP-GAIN 4 | w208 | b2 |
| LPC 3 | w22 - w23 | b2, b1 |
| LPC 4 | w27 | b5 |
| LPC 4 | w29 | b3 |
| PULSE 1_1 | w52 | b3 |
| PULSE 1_2 | w56 | b3 |
| PULSE 1_3 | w60 | b3 |
| PULSE 1_4 | w64 | b3 |
| PULSE 1_5 | w68 | b3 |
| PULSE 2_1 | w104 | b3 |
| PULSE 2_2 | w108 | b3 |
| PULSE 2_3 | w112 | b3 |
| PULSE 2_4 | w116 | b3 |
| PULSE 2_5 | w120 | b3 |
| PULSE 3_1 | w159 | b3 |
| PULSE 3_2 | w163 | b3 |
| PULSE 3_3 | w167 | b3 |
| PULSE 3_4 | w171 | b3 |
| PULSE 3_5 | w175 | b3 |
| PULSE 4_1 | w211 | b3 |
| PULSE 4_2 | w215 | b3 |
| PULSE 4_3 | w219 | b3 |
| PULSE 4_4 | w223 | b3 |
| PULSE 4_5 | w227 | b3 |
| FCB-GAIN 1 | w91 | b2 |
| FCB-GAIN 2 | w143 | b2 |
| FCB-GAIN 3 | w198 | b2 |
| FCB-GAIN 4 | w250 | b2 |
| LTP-GAIN 1 | w50 | b1 |
| LTP-GAIN 2 | w102 | b1 |
| LTP-GAIN 3 | w157 | b1 |
| LTP-GAIN 4 | w209 | b1 |
| LPC 4 | w30 - w32 | b2, b1, b0 |
| LPC 5 | w33 - w36 | b5, b4, b3, b2 |
| LTP-LAG 2 | w99 | b0 |
| LTP-LAG 4 | w206 | b0 |
| PULSE 1_1 | w53 | b2 |
| PULSE 1_2 | w57 | b2 |

(continued)

**Table 6 (continued): Ordering of enhanced full rate speech parameters for the channel encoder
(subjective importance of encoded bits) (after preliminary channel coding)
(Parameter names refers to GSM 06.60)**

| Description | Bits (Table 5) | Bit index within parameter |
|-------------|-------------------|----------------------------|
| PULSE 1_3 | w61 | b2 |
| PULSE 1_4 | w65 | b2 |
| PULSE 1_5 | w69 | b2 |
| PULSE 2_1 | w105 | b2 |
| PULSE 2_2 | w109 | b2 |
| PULSE 2_3 | w113 | b2 |
| PULSE 2_4 | w117 | b2 |
| PULSE 2_5 | w121 | b2 |
| PULSE 3_1 | w160 | b2 |
| PULSE 3_2 | w164 | b2 |
| PULSE 3_3 | w168 | b2 |
| PULSE 3_4 | w172 | b2 |
| PULSE 3_5 | w176 | b2 |
| PULSE 4_1 | w212 | b2 |
| PULSE 4_2 | w216 | b2 |
| PULSE 4_3 | w220 | b2 |
| PULSE 4_4 | w224 | b2 |
| PULSE 4_5 | w228 | b2 |
| PULSE 1_1 | w54 | b1 |
| PULSE 1_2 | w58 | b1 |
| PULSE 1_3 | w62 | b1 |
| PULSE 1_4 | w66 | b1 |
| PULSE 2_1 | w106 | b1 |
| PULSE 2_2 | w110 | b1 |
| PULSE 2_3 | w114 | b1 |
| PULSE 2_4 | w118 | b1 |
| PULSE 3_1 | w161 | b1 |
| PULSE 3_2 | w165 | b1 |
| PULSE 3_3 | w169 | b1 |
| PULSE 3_4 | w173 | b1 |
| PULSE 4_1 | w213 | b1 |
| PULSE 4_3 | w221 | b1 |
| PULSE 4_4 | w225 | b1 |
| FCB-GAIN 1 | w92 | b1 |
| FCB-GAIN 2 | w144 | b1 |
| FCB-GAIN 3 | s199 | b1 |
| FCB-GAIN 4 | w251 | b1 |
| LTP-GAIN 1 | w51 | b0 |
| LTP-GAIN 2 | w103 | b0 |
| LTP-GAIN 3 | w158 | b0 |
| LTP-GAIN 4 | w210 | b0 |
| FCB-GAIN 1 | w93 | b0 |
| FCB-GAIN 2 | w145 | b0 |
| FCB-GAIN 3 | w200 | b0 |
| FCB-GAIN 4 | w252 | b0 |
| PULSE 1_1 | w55 | b0 |
| PULSE 1_2 | w59 | b0 |
| PULSE 1_3 | w63 | b0 |
| PULSE 1_4 | w67 | b0 |
| PULSE 2_1 | w107 | b0 |
| PULSE 2_2 | w111 | b0 |
| PULSE 2_3 | w115 | b0 |
| PULSE 2_4 | w119 | b0 |
| PULSE 3_1 | w162 | b0 |
| PULSE 3_2 | w166 | b0 |
| PULSE 3_3 | w170 | b0 |

(continued)

**Table 6 (continued): Ordering of enhanced full rate speech parameters for the channel encoder
(subjective importance of encoded bits) (after preliminary channel coding)
(Parameter names refers to GSM 06.60)**

| Description | Bits (Table 5) | Bit index within parameter |
|---------------------------------------|-------------------|----------------------------|
| PULSE 3_4 | w174 | b0 |
| PULSE 4_1 | w214 | b0 |
| PULSE 4_3 | w222 | b0 |
| PULSE 4_4 | w226 | b0 |
| LPC 5 | w37 - w38 | b1, b0 |
| CLASS 2: 78 bits (unprotected) | | |
| PULSE 1_5 | w70 | b1 |
| PULSE 1_5 | w72 - w73 | b1, b1 |
| PULSE 2_5 | w122 | b1 |
| PULSE 2_5 | w124 - s125 | b1, b1 |
| PULSE 3_5 | w177 | b1 |
| PULSE 3_5 | w179 - w180 | b1, b1 |
| PULSE 4_5 | w229 | b1 |
| PULSE 4_5 | w231 - w232 | b1, b1 |
| PULSE 4_2 | w217 - w218 | b1, b0 |
| PULSE 1_5 | w71 | b0 |
| PULSE 2_5 | w123 | b0 |
| PULSE 3_5 | w178 | b0 |
| PULSE 4_5 | w230 | b0 |
| PULSE 1_6 | w74 | b2 |
| PULSE 1_7 | w77 | b2 |
| PULSE 1_8 | w80 | b2 |
| PULSE 1_9 | w83 | b2 |
| PULSE 1_10 | w86 | b2 |
| PULSE 2_6 | w126 | b2 |
| PULSE 2_7 | w129 | b2 |
| PULSE 2_8 | w132 | b2 |
| PULSE 2_9 | w135 | b2 |
| PULSE 2_10 | w138 | b2 |
| PULSE 3_6 | w181 | b2 |
| PULSE 3_7 | w184 | b2 |
| PULSE 3_8 | w187 | b2 |
| PULSE 3_9 | w190 | b2 |
| PULSE 3_10 | w193 | b2 |
| PULSE 4_6 | w233 | b2 |
| PULSE 4_7 | w236 | b2 |
| PULSE 4_8 | w239 | b2 |
| PULSE 4_9 | w242 | b2 |
| PULSE 4_10 | w245 | b2 |
| PULSE 1_6 | w75 | b1 |
| PULSE 1_7 | w78 | b1 |
| PULSE 1_8 | w81 | b1 |
| PULSE 1_9 | w84 | b1 |
| PULSE 1_10 | w87 | b1 |
| PULSE 2_6 | w127 | b1 |
| PULSE 2_7 | w130 | b1 |
| PULSE 2_8 | w133 | b1 |
| PULSE 2_9 | w136 | b1 |
| PULSE 2_10 | w139 | b1 |
| PULSE 3_6 | w182 | b1 |
| PULSE 3_7 | w185 | b1 |
| PULSE 3_8 | w188 | b1 |
| PULSE 3_9 | w191 | b1 |
| PULSE 3_10 | w194 | b1 |
| PULSE 4_6 | w234 | b1 |
| PULSE 4_7 | w237 | b1 |

(continued)

**Table 6 (concluded): Ordering of enhanced full rate speech parameters for the channel encoder
(subjective importance of encoded bits) (after preliminary channel coding)
(Parameter names refers to GSM 06.60)**

| Description | Bits (Table 5) | Bit index within parameter |
|-------------|-------------------|----------------------------|
| PULSE 4_8 | w240 | b1 |
| PULSE 4_9 | w243 | b1 |
| PULSE 4_10 | w246 | b1 |
| PULSE 1_6 | w76 | b0 |
| PULSE 1_7 | w79 | b0 |
| PULSE 1_8 | w82 | b0 |
| PULSE 1_9 | w85 | b0 |
| PULSE 1_10 | w88 | b0 |
| PULSE 2_6 | w128 | b0 |
| PULSE 2_7 | w131 | b0 |
| PULSE 2_8 | w134 | b0 |
| PULSE 2_9 | w137 | b0 |
| PULSE 2_10 | w140 | b0 |
| PULSE 3_6 | w183 | b0 |
| PULSE 3_7 | w186 | b0 |
| PULSE 3_8 | w189 | b0 |
| PULSE 3_9 | w192 | b0 |
| PULSE 3_10 | w195 | b0 |
| PULSE 4_6 | w235 | b0 |
| PULSE 4_7 | w238 | b0 |
| PULSE 4_8 | w241 | b0 |
| PULSE 4_9 | w244 | b0 |
| PULSE 4_10 | w247 | b0 |

Table 7: Sorting of the speech encoded bits for TCH/AFS12.2

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 23 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 24 | 25 | 26 | 27 | 28 | 38 |
| 141 | 39 | 142 | 40 | 143 | 41 | 144 | 42 | 145 | 43 |
| 146 | 44 | 147 | 45 | 148 | 46 | 149 | 47 | 97 | 150 |
| 200 | 48 | 98 | 151 | 201 | 49 | 99 | 152 | 202 | 86 |
| 136 | 189 | 239 | 87 | 137 | 190 | 240 | 88 | 138 | 191 |
| 241 | 91 | 194 | 92 | 195 | 93 | 196 | 94 | 197 | 95 |
| 198 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 50 | 100 |
| 153 | 203 | 89 | 139 | 192 | 242 | 51 | 101 | 154 | 204 |
| 55 | 105 | 158 | 208 | 90 | 140 | 193 | 243 | 59 | 109 |
| 162 | 212 | 63 | 113 | 166 | 216 | 67 | 117 | 170 | 220 |
| 36 | 37 | 54 | 53 | 52 | 58 | 57 | 56 | 62 | 61 |
| 60 | 66 | 65 | 64 | 70 | 69 | 68 | 104 | 103 | 102 |
| 108 | 107 | 106 | 112 | 111 | 110 | 116 | 115 | 114 | 120 |
| 119 | 118 | 157 | 156 | 155 | 161 | 160 | 159 | 165 | 164 |
| 163 | 169 | 168 | 167 | 173 | 172 | 171 | 207 | 206 | 205 |
| 211 | 210 | 209 | 215 | 214 | 213 | 219 | 218 | 217 | 223 |
| 222 | 221 | 73 | 72 | 71 | 76 | 75 | 74 | 79 | 78 |
| 77 | 82 | 81 | 80 | 85 | 84 | 83 | 123 | 122 | 121 |
| 126 | 125 | 124 | 129 | 128 | 127 | 132 | 131 | 130 | 135 |
| 134 | 133 | 176 | 175 | 174 | 179 | 178 | 177 | 182 | 181 |
| 180 | 185 | 184 | 183 | 188 | 187 | 186 | 226 | 225 | 224 |
| 229 | 228 | 227 | 232 | 231 | 230 | 235 | 234 | 233 | 238 |
| 237 | 236 | 96 | 199 | | | | | | |

Table 8: Sorting of the speech encoded bits for TCH/AFS10.2

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 16 | 15 |
| 14 | 13 | 12 | 11 | 10 | 9 | 8 | 26 | 27 | 28 |
| 29 | 30 | 31 | 115 | 116 | 117 | 118 | 119 | 120 | 72 |
| 73 | 161 | 162 | 65 | 68 | 69 | 108 | 111 | 112 | 154 |
| 157 | 158 | 197 | 200 | 201 | 32 | 33 | 121 | 122 | 74 |
| 75 | 163 | 164 | 66 | 109 | 155 | 198 | 19 | 23 | 21 |
| 22 | 18 | 17 | 20 | 24 | 25 | 37 | 36 | 35 | 34 |
| 80 | 79 | 78 | 77 | 126 | 125 | 124 | 123 | 169 | 168 |
| 167 | 166 | 70 | 67 | 71 | 113 | 110 | 114 | 159 | 156 |
| 160 | 202 | 199 | 203 | 76 | 165 | 81 | 82 | 92 | 91 |
| 93 | 83 | 95 | 85 | 84 | 94 | 101 | 102 | 96 | 104 |
| 86 | 103 | 87 | 97 | 127 | 128 | 138 | 137 | 139 | 129 |
| 141 | 131 | 130 | 140 | 147 | 148 | 142 | 150 | 132 | 149 |
| 133 | 143 | 170 | 171 | 181 | 180 | 182 | 172 | 184 | 174 |
| 173 | 183 | 190 | 191 | 185 | 193 | 175 | 192 | 176 | 186 |
| 38 | 39 | 49 | 48 | 50 | 40 | 52 | 42 | 41 | 51 |
| 58 | 59 | 53 | 61 | 43 | 60 | 44 | 54 | 194 | 179 |
| 189 | 196 | 177 | 195 | 178 | 187 | 188 | 151 | 136 | 146 |
| 153 | 134 | 152 | 135 | 144 | 145 | 105 | 90 | 100 | 107 |
| 88 | 106 | 89 | 98 | 99 | 62 | 47 | 57 | 64 | 45 |
| 63 | 46 | 55 | 56 | | | | | | |

Table 9: Sorting of the speech encoded bits for TCH/AFS7.95 and TCH/AHS7.95

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 14 | 16 | 9 |
| 10 | 12 | 13 | 15 | 11 | 17 | 20 | 22 | 24 | 23 |
| 19 | 18 | 21 | 56 | 88 | 122 | 154 | 57 | 89 | 123 |
| 155 | 58 | 90 | 124 | 156 | 52 | 84 | 118 | 150 | 53 |
| 85 | 119 | 151 | 27 | 93 | 28 | 94 | 29 | 95 | 30 |
| 96 | 31 | 97 | 61 | 127 | 62 | 128 | 63 | 129 | 59 |
| 91 | 125 | 157 | 32 | 98 | 64 | 130 | 1 | 0 | 25 |
| 26 | 33 | 99 | 34 | 100 | 65 | 131 | 66 | 132 | 54 |
| 86 | 120 | 152 | 60 | 92 | 126 | 158 | 55 | 87 | 121 |
| 153 | 117 | 116 | 115 | 46 | 78 | 112 | 144 | 43 | 75 |
| 109 | 141 | 40 | 72 | 106 | 138 | 36 | 68 | 102 | 134 |
| 114 | 149 | 148 | 147 | 146 | 83 | 82 | 81 | 80 | 51 |
| 50 | 49 | 48 | 47 | 45 | 44 | 42 | 39 | 35 | 79 |
| 77 | 76 | 74 | 71 | 67 | 113 | 111 | 110 | 108 | 105 |
| 101 | 145 | 143 | 142 | 140 | 137 | 133 | 41 | 73 | 107 |
| 139 | 37 | 69 | 103 | 135 | 38 | 70 | 104 | 136 | |

Table 10: Sorting of the speech encoded bits for TCH/AFS7.4 and TCH/AHS7.4

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 26 | 87 | 27 |
| 88 | 28 | 89 | 29 | 90 | 30 | 91 | 51 | 80 | 112 |
| 141 | 52 | 81 | 113 | 142 | 54 | 83 | 115 | 144 | 55 |
| 84 | 116 | 145 | 58 | 119 | 59 | 120 | 21 | 22 | 23 |
| 17 | 18 | 19 | 31 | 60 | 92 | 121 | 56 | 85 | 117 |
| 146 | 20 | 24 | 25 | 50 | 79 | 111 | 140 | 57 | 86 |
| 118 | 147 | 49 | 78 | 110 | 139 | 48 | 77 | 53 | 82 |
| 114 | 143 | 109 | 138 | 47 | 76 | 108 | 137 | 32 | 33 |
| 61 | 62 | 93 | 94 | 122 | 123 | 41 | 42 | 43 | 44 |
| 45 | 46 | 70 | 71 | 72 | 73 | 74 | 75 | 102 | 103 |
| 104 | 105 | 106 | 107 | 131 | 132 | 133 | 134 | 135 | 136 |
| 34 | 63 | 95 | 124 | 35 | 64 | 96 | 125 | 36 | 65 |
| 97 | 126 | 37 | 66 | 98 | 127 | 38 | 67 | 99 | 128 |
| 39 | 68 | 100 | 129 | 40 | 69 | 101 | 130 | | |

Table 11: Sorting of the speech encoded bits for TCH/AFS6.7 and TCH/AHS6.7

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 4 | 3 | 5 | 6 | 13 | 7 | 2 | 8 |
| 9 | 11 | 15 | 12 | 14 | 10 | 28 | 82 | 29 | 83 |
| 27 | 81 | 26 | 80 | 30 | 84 | 16 | 55 | 109 | 56 |
| 110 | 31 | 85 | 57 | 111 | 48 | 73 | 102 | 127 | 32 |
| 86 | 51 | 76 | 105 | 130 | 52 | 77 | 106 | 131 | 58 |
| 112 | 33 | 87 | 19 | 23 | 53 | 78 | 107 | 132 | 21 |
| 22 | 18 | 17 | 20 | 24 | 25 | 50 | 75 | 104 | 129 |
| 47 | 72 | 101 | 126 | 54 | 79 | 108 | 133 | 46 | 71 |
| 100 | 125 | 128 | 103 | 74 | 49 | 45 | 70 | 99 | 124 |
| 42 | 67 | 96 | 121 | 39 | 64 | 93 | 118 | 38 | 63 |
| 92 | 117 | 35 | 60 | 89 | 114 | 34 | 59 | 88 | 113 |
| 44 | 69 | 98 | 123 | 43 | 68 | 97 | 122 | 41 | 66 |
| 95 | 120 | 40 | 65 | 94 | 119 | 37 | 62 | 91 | 116 |
| 36 | 61 | 90 | 115 | | | | | | |

Table 12: Sorting of the speech encoded bits for TCH/AFS5.9 and TCH/AHS5.9

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

Table 13: Sorting of the speech encoded bits for TCH/AFS5.15 and TCH/AHS5.15

| | | | | | | | | | |
|----|----|----|-----|-----|-----|----|----|----|----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 |
| 13 | 12 | 11 | 10 | 9 | 8 | 23 | 24 | 25 | 26 |
| 27 | 46 | 65 | 84 | 45 | 44 | 43 | 64 | 63 | 62 |
| 83 | 82 | 81 | 102 | 101 | 100 | 42 | 61 | 80 | 99 |
| 28 | 47 | 66 | 85 | 18 | 41 | 60 | 79 | 98 | 29 |
| 48 | 67 | 17 | 20 | 22 | 40 | 59 | 78 | 97 | 21 |
| 30 | 49 | 68 | 86 | 19 | 16 | 87 | 39 | 38 | 58 |
| 57 | 77 | 35 | 54 | 73 | 92 | 76 | 96 | 95 | 36 |
| 55 | 74 | 93 | 32 | 51 | 33 | 52 | 70 | 71 | 89 |
| 90 | 31 | 50 | 69 | 88 | 37 | 56 | 75 | 94 | 34 |
| 53 | 72 | 91 | | | | | | | |

Table 14: Sorting of the speech encoded bits for TCH/AFS4.75 and TCH/AHS4.75

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 23 | 24 | 25 | 26 |
| 27 | 28 | 48 | 49 | 61 | 62 | 82 | 83 | 47 | 46 |
| 45 | 44 | 81 | 80 | 79 | 78 | 17 | 18 | 20 | 22 |
| 77 | 76 | 75 | 74 | 29 | 30 | 43 | 42 | 41 | 40 |
| 38 | 39 | 16 | 19 | 21 | 50 | 51 | 59 | 60 | 63 |
| 64 | 72 | 73 | 84 | 85 | 93 | 94 | 32 | 33 | 35 |
| 36 | 53 | 54 | 56 | 57 | 66 | 67 | 69 | 70 | 87 |
| 88 | 90 | 91 | 34 | 55 | 68 | 89 | 37 | 58 | 71 |
| 92 | 31 | 52 | 65 | 86 | | | | | |

Table 15: Interleaving table for MCS5 and MCS6:

| m\n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|------|------|------|------|------|------|-----|------|------|------|------|------|------|------|-----|------|
| 0 | 0 | 463 | 890 | 1038 | 220 | 371 | 795 | 946 | 582 | 733 | 1160 | 63 | 490 | 641 | 277 | 428 |
| 1 | 852 | 1003 | 185 | 333 | 1223 | 120 | 547 | 698 | 1122 | 28 | 915 | 1066 | 242 | 390 | 817 | 968 |
| 2 | 610 | 761 | 1185 | 85 | 512 | 660 | 305 | 453 | 880 | 1031 | 204 | 355 | 782 | 1242 | 148 | 575 |
| 3 | 723 | 1150 | 50 | 474 | 625 | 1088 | 267 | 418 | 845 | 993 | 169 | 320 | 1207 | 113 | 537 | 688 |
| 4 | 1115 | 12 | 902 | 1050 | 232 | 383 | 807 | 958 | 594 | 745 | 1172 | 75 | 502 | 653 | 289 | 440 |
| 5 | 864 | 1015 | 197 | 345 | 1235 | 132 | 559 | 710 | 1134 | 40 | 927 | 1078 | 254 | 402 | 829 | 980 |
| 6 | 159 | 622 | 773 | 1197 | 97 | 524 | 672 | 1099 | 5 | 465 | 892 | 1043 | 216 | 367 | 794 | 942 |
| 7 | 587 | 735 | 1162 | 62 | 486 | 637 | 279 | 430 | 857 | 1005 | 181 | 332 | 1219 | 125 | 549 | 700 |
| 8 | 1127 | 24 | 914 | 1062 | 244 | 395 | 819 | 970 | 606 | 757 | 1184 | 87 | 514 | 665 | 301 | 452 |
| 9 | 876 | 1027 | 209 | 357 | 784 | 1247 | 144 | 571 | 722 | 1146 | 52 | 479 | 627 | 1090 | 266 | 414 |
| 10 | 841 | 992 | 171 | 322 | 1209 | 109 | 536 | 684 | 1111 | 17 | 904 | 1055 | 228 | 379 | 806 | 954 |
| 11 | 599 | 747 | 1174 | 74 | 498 | 649 | 291 | 442 | 869 | 1017 | 193 | 344 | 1231 | 137 | 561 | 712 |
| 12 | 1139 | 36 | 926 | 1074 | 256 | 407 | 831 | 982 | 158 | 618 | 769 | 1196 | 99 | 526 | 677 | 1101 |
| 13 | 7 | 458 | 894 | 1033 | 227 | 363 | 802 | 941 | 577 | 740 | 1152 | 70 | 485 | 645 | 284 | 420 |
| 14 | 859 | 998 | 189 | 328 | 1215 | 127 | 542 | 702 | 1117 | 35 | 922 | 1061 | 246 | 385 | 824 | 960 |
| 15 | 605 | 765 | 1180 | 92 | 504 | 667 | 309 | 448 | 887 | 1023 | 211 | 350 | 786 | 1237 | 155 | 567 |
| 16 | 730 | 1145 | 54 | 469 | 632 | 1080 | 274 | 413 | 849 | 988 | 176 | 312 | 1202 | 117 | 532 | 695 |
| 17 | 1107 | 19 | 906 | 1045 | 239 | 375 | 814 | 953 | 589 | 752 | 1164 | 82 | 497 | 657 | 296 | 432 |
| 18 | 871 | 1010 | 201 | 340 | 1227 | 139 | 554 | 714 | 1129 | 47 | 934 | 1073 | 258 | 397 | 836 | 972 |
| 19 | 166 | 617 | 777 | 1192 | 104 | 516 | 679 | 1094 | 9 | 460 | 899 | 1035 | 223 | 362 | 798 | 937 |
| 20 | 579 | 742 | 1157 | 66 | 481 | 644 | 286 | 425 | 861 | 1000 | 188 | 324 | 1214 | 129 | 544 | 707 |
| 21 | 1119 | 31 | 918 | 1057 | 251 | 387 | 826 | 965 | 601 | 764 | 1176 | 94 | 509 | 669 | 308 | 444 |
| 22 | 883 | 1022 | 213 | 352 | 791 | 1239 | 151 | 566 | 726 | 1141 | 59 | 471 | 634 | 1085 | 270 | 409 |
| 23 | 848 | 984 | 178 | 317 | 1204 | 116 | 528 | 691 | 1106 | 21 | 911 | 1047 | 235 | 374 | 810 | 949 |
| 24 | 591 | 754 | 1169 | 78 | 493 | 656 | 298 | 437 | 873 | 1012 | 200 | 336 | 1226 | 141 | 556 | 719 |
| 25 | 1131 | 43 | 930 | 1069 | 263 | 399 | 838 | 977 | 162 | 613 | 776 | 1188 | 106 | 521 | 681 | 1096 |
| 26 | 2 | 462 | 889 | 1040 | 219 | 370 | 797 | 945 | 584 | 732 | 1159 | 65 | 489 | 640 | 276 | 427 |
| 27 | 854 | 1002 | 184 | 335 | 1222 | 122 | 546 | 697 | 1124 | 27 | 917 | 1065 | 241 | 392 | 816 | 967 |
| 28 | 609 | 760 | 1187 | 84 | 511 | 662 | 304 | 455 | 879 | 1030 | 206 | 354 | 781 | 1244 | 147 | 574 |
| 29 | 725 | 1149 | 49 | 476 | 624 | 1087 | 269 | 417 | 844 | 995 | 168 | 319 | 1206 | 112 | 539 | 687 |
| 30 | 1114 | 14 | 901 | 1052 | 231 | 382 | 809 | 957 | 596 | 744 | 1171 | 77 | 501 | 652 | 288 | 439 |
| 31 | 866 | 1014 | 196 | 347 | 1234 | 134 | 558 | 709 | 1136 | 39 | 929 | 1077 | 253 | 404 | 828 | 979 |
| 32 | 161 | 621 | 772 | 1199 | 96 | 523 | 674 | 1098 | 4 | 467 | 891 | 1042 | 218 | 366 | 793 | 944 |
| 33 | 586 | 737 | 1161 | 61 | 488 | 636 | 281 | 429 | 856 | 1007 | 180 | 331 | 1218 | 124 | 551 | 699 |
| 34 | 1126 | 26 | 913 | 1064 | 243 | 394 | 821 | 969 | 608 | 756 | 1183 | 89 | 513 | 664 | 300 | 451 |
| 35 | 878 | 1026 | 208 | 359 | 783 | 1246 | 146 | 570 | 721 | 1148 | 51 | 478 | 629 | 1089 | 265 | 416 |
| 36 | 840 | 991 | 173 | 321 | 1211 | 108 | 535 | 686 | 1110 | 16 | 903 | 1054 | 230 | 378 | 805 | 956 |
| 37 | 598 | 749 | 1173 | 73 | 500 | 648 | 293 | 441 | 868 | 1019 | 192 | 343 | 1230 | 136 | 563 | 711 |
| 38 | 1138 | 38 | 925 | 1076 | 255 | 406 | 833 | 981 | 157 | 620 | 768 | 1195 | 101 | 525 | 676 | 1103 |
| 39 | 6 | 457 | 896 | 1032 | 226 | 365 | 801 | 940 | 576 | 739 | 1154 | 69 | 484 | 647 | 283 | 422 |
| 40 | 858 | 997 | 191 | 327 | 1217 | 126 | 541 | 704 | 1116 | 34 | 921 | 1060 | 248 | 384 | 823 | 962 |
| 41 | 604 | 767 | 1179 | 91 | 506 | 666 | 311 | 447 | 886 | 1025 | 210 | 349 | 788 | 1236 | 154 | 569 |
| 42 | 729 | 1144 | 56 | 468 | 631 | 1082 | 273 | 412 | 851 | 987 | 175 | 314 | 1201 | 119 | 531 | 694 |
| 43 | 1109 | 18 | 908 | 1044 | 238 | 377 | 813 | 952 | 588 | 751 | 1166 | 81 | 496 | 659 | 295 | 434 |
| 44 | 870 | 1009 | 203 | 339 | 1229 | 138 | 553 | 716 | 1128 | 46 | 933 | 1072 | 260 | 396 | 835 | 974 |
| 45 | 165 | 616 | 779 | 1191 | 103 | 518 | 678 | 1093 | 11 | 459 | 898 | 1037 | 222 | 361 | 790 | 936 |
| 46 | 581 | 741 | 1156 | 68 | 480 | 643 | 285 | 424 | 863 | 999 | 187 | 326 | 1213 | 131 | 543 | 706 |
| 47 | 1121 | 30 | 920 | 1056 | 250 | 389 | 825 | 964 | 600 | 763 | 1178 | 93 | 508 | 671 | 307 | 446 |
| 48 | 882 | 1021 | 215 | 351 | 790 | 1241 | 150 | 565 | 728 | 1140 | 58 | 473 | 633 | 1084 | 272 | 408 |
| 49 | 847 | 986 | 177 | 316 | 1203 | 115 | 530 | 690 | 1105 | 23 | 910 | 1049 | 234 | 373 | 812 | 948 |
| 50 | 593 | 753 | 1168 | 80 | 492 | 655 | 297 | 436 | 875 | 1011 | 199 | 338 | 1225 | 143 | 555 | 718 |
| 51 | 1133 | 42 | 932 | 1068 | 262 | 401 | 837 | 976 | 164 | 612 | 775 | 1190 | 105 | 520 | 683 | 1095 |
| 52 | 1 | 464 | 888 | 1039 | 221 | 369 | 796 | 947 | 583 | 734 | 1158 | 64 | 491 | 639 | 278 | 426 |
| 53 | 853 | 1004 | 183 | 334 | 1221 | 121 | 548 | 696 | 1123 | 29 | 916 | 1067 | 240 | 391 | 818 | 966 |
| 54 | 611 | 759 | 1186 | 86 | 510 | 661 | 303 | 454 | 881 | 1029 | 205 | 356 | 780 | 1243 | 149 | 573 |
| 55 | 724 | 1151 | 48 | 475 | 626 | 1086 | 268 | 419 | 843 | 994 | 170 | 318 | 1208 | 111 | 538 | 689 |
| 56 | 1113 | 13 | 900 | 1051 | 233 | 381 | 808 | 959 | 595 | 746 | 1170 | 76 | 503 | 651 | 290 | 438 |
| 57 | 865 | 1016 | 195 | 346 | 1233 | 133 | 560 | 708 | 1135 | 41 | 928 | 1079 | 252 | 403 | 830 | 978 |
| 58 | 160 | 623 | 771 | 1198 | 98 | 522 | 673 | 1100 | 3 | 466 | 893 | 1041 | 217 | 368 | 792 | 943 |
| 59 | 585 | 736 | 1163 | 60 | 487 | 638 | 280 | 431 | 855 | 1006 | 182 | 330 | 1220 | 123 | 550 | 701 |
| 60 | 1125 | 25 | 912 | 1063 | 245 | 393 | 820 | 971 | 607 | 758 | 1182 | 88 | 515 | 663 | 302 | 450 |
| 61 | 877 | 1028 | 207 | 358 | 785 | 1245 | 145 | 572 | 720 | 1147 | 53 | 477 | 628 | 1091 | 264 | 415 |
| 62 | 842 | 990 | 172 | 323 | 1210 | 110 | 534 | 685 | 1112 | 15 | 905 | 1053 | 229 | 380 | 804 | 955 |
| 63 | 597 | 748 | 1175 | 72 | 499 | 650 | 292 | 443 | 867 | 1018 | 194 | 342 | 1232 | 135 | 562 | 713 |
| 64 | 1137 | 37 | 924 | 1075 | 257 | 405 | 832 | 983 | 156 | 619 | 770 | 1194 | 100 | 527 | 675 | 1102 |
| 65 | 8 | 456 | 895 | 1034 | 225 | 364 | 803 | 939 | 578 | 738 | 1153 | 71 | 483 | 646 | 282 | 421 |
| 66 | 860 | 996 | 190 | 329 | 1216 | 128 | 540 | 703 | 1118 | 33 | 923 | 1059 | 247 | 386 | 822 | 961 |
| 67 | 603 | 766 | 1181 | 90 | 505 | 668 | 310 | 449 | 885 | 1024 | 212 | 348 | 787 | 1238 | 153 | 568 |

| | | | | | | | | | | | | | | | | |
|----|------|------|------|------|------|------|-----|------|------|------|------|------|------|------|-----|------|
| 68 | 731 | 1143 | 55 | 470 | 630 | 1081 | 275 | 411 | 850 | 989 | 174 | 313 | 1200 | 118 | 533 | 693 |
| 69 | 1108 | 20 | 907 | 1046 | 237 | 376 | 815 | 951 | 590 | 750 | 1165 | 83 | 495 | 658 | 294 | 433 |
| 70 | 872 | 1008 | 202 | 341 | 1228 | 140 | 552 | 715 | 1130 | 45 | 935 | 1071 | 259 | 398 | 834 | 973 |
| 71 | 167 | 615 | 778 | 1193 | 102 | 517 | 680 | 1092 | 10 | 461 | 897 | 1036 | 224 | 360 | 799 | 938 |
| 72 | 580 | 743 | 1155 | 67 | 482 | 642 | 287 | 423 | 862 | 1001 | 186 | 325 | 1212 | 130 | 545 | 705 |
| 73 | 1120 | 32 | 919 | 1058 | 249 | 388 | 827 | 963 | 602 | 762 | 1177 | 95 | 507 | 670 | 306 | 445 |
| 74 | 884 | 1020 | 214 | 353 | 789 | 1240 | 152 | 564 | 727 | 1142 | 57 | 472 | 635 | 1083 | 271 | 410 |
| 75 | 846 | 985 | 179 | 315 | 1205 | 114 | 529 | 692 | 1104 | 22 | 909 | 1048 | 236 | 372 | 811 | 950 |
| 76 | 592 | 755 | 1167 | 79 | 494 | 654 | 299 | 435 | 874 | 1013 | 198 | 337 | 1224 | 142 | 557 | 717 |
| 77 | 1132 | 44 | 931 | 1070 | 261 | 400 | 839 | 975 | 163 | 614 | 774 | 1189 | 107 | 519 | 682 | 1097 |

This table describes the interleaving applied to MCS-5 and MCS-6

$$d_i(j') = d_c(k') \text{ for } k' = 0, 1, \dots, 1223$$

$$k' = 16 \cdot m + n$$

The value of j' for a given k is in the cell located in the row m and in the column n .

Annex A (informative): Summary of Channel Types

| | |
|--------------|---|
| TCH/EFS: | enhanced full rate speech traffic channel |
| TCH/FS: | full rate speech traffic channel |
| TCH/HS: | half rate speech traffic channel |
| TCH/AFS: | adaptive multirate full rate speech traffic channel |
| TCH/AFS12.2 | adaptive multirate full rate speech, 12.2 kbit/s |
| TCH/AFS10.2 | adaptive multirate full rate speech, 10.2 kbit/s |
| TCH/AFS7.95 | adaptive multirate full rate speech, 7.95 kbit/s |
| TCH/AFS7.4 | adaptive multirate full rate speech, 7.5 kbit/s |
| TCH/AFS6.7 | adaptive multirate full rate speech, 6.7 kbit/s |
| TCH/AFS5.9 | adaptive multirate full rate speech, 5.9 kbit/s |
| TCH/AFS5.15 | adaptive multirate full rate speech, 5.15 kbit/s |
| TCH/AFS4.75 | adaptive multirate full rate speech, 4.75 kbit/s |
| TCH/AHS: | adaptive multirate half rate speech traffic channel |
| TCH/AHS7.95 | adaptive multirate half rate speech, 7.95 kbit/s |
| TCH/AHS7.4 | adaptive multirate half rate speech, 7.5 kbit/s |
| TCH/AHS6.7 | adaptive multirate half rate speech, 6.7 kbit/s |
| TCH/AHS5.9 | adaptive multirate half rate speech, 5.9 kbit/s |
| TCH/AHS5.15 | adaptive multirate half rate speech, 5.15 kbit/s |
| TCH/AHS4.75 | adaptive multirate half rate speech, 4.75 kbit/s |
| E-TCH/F43.2: | 43.2 kbit/s full rate data traffic channel |
| E-TCH/F32.0: | 32.0 kbit/s full rate data traffic channel |
| E-TCH/F28.8: | 28.8 kbit/s full rate data traffic channel |
| TCH/F14.4 | 14.4 kbit/s full rate data traffic channel |
| TCH/F9.6: | 9.6 kbit/s full rate data traffic channel |
| TCH/F4.8: | 4.8 kbit/s full rate data traffic channel |
| TCH/H4.8: | 4.8 kbit/s half rate data traffic channel |
| TCH/F2.4: | 2.4 kbit/s full rate data traffic channel |
| TCH/H2.4: | 2.4 kbit/s half rate data traffic channel |
| SACCH: | slow associated control channel |
| FACCH/F: | fast associated control channel at full rate |
| FACCH/H: | fast associated control channel at half rate |

| | |
|------------|--|
| E-FACCH/F: | enhanced circuit switched fast associated control channel at full rate |
| SDCCH: | stand-alone dedicated control channel |
| BCCH: | broadcast control channel |
| PCH: | paging channel |
| AGCH | access grant channel |
| RACH: | random access channel |
| SCH: | synchronization channel |
| CBCH: | cell broadcast channel |
| CTSBCH-SB: | CTS beacon channel (synchronisation burst) |
| CTSPCH: | CTS paging channel |
| CTSARCH: | CTS access request channel |
| CTSAGCH: | CTS access grant channel |
| PDTCH | packet data traffic channel |
| PACCH | packet associated control channel |
| PBCCH | packet broadcast control channel |
| PAGCH | packet access grant channel |
| PPCH | packet paging channel |
| PNCH | packet notification channel |
| PTCCH | packet timing advance control channel |
| PRACH | packet random access channel |
| CFCCH | Compact Frequency Correction Channel |
| CPAGCH | Compact Packet Access Grant Channel |
| CPBCCH | Compact Packet Broadcast Control Channel |
| CPCCCH | Compact Packet Common Control Channel |
| CPNCH | Compact Packet Notification Channel (for PTM-M on CPCCCH) |
| CPPCH | Compact Packet Paging Channel |
| CPRACH | Compact Packet Random Access Channel |
| CSCH | Compact Synchronization Channel |

Annex B (informative): Summary of Polynomials Used for Convolutional Codes

| | |
|---------------------------------------|---|
| $G_0 = 1 + D^3 + D^4$ | TCH/FS, TCH/EFS, TCH/AFS, TCH/AHS, TCH/F14.4, TCH/F9.6, TCH/H4.8, SDCCH, BCCH, PCH, SACCH, FACCH, E-FACCH, AGCH, RACH, SCH, CSCH, CTSBCH-SB, CTSPCH, CTSARCH, CTSAGCH, PDTCH (CS-1, CS-2, CS3, CS-4), PACCH, PBCCH, PAGCH, PPCH, PNCH, PTCCH, PRACH, CPBCCH, CPAGCH, CPPCH, CPNCH |
| $G_1 = 1 + D + D^3 + D^4$ | TCH/FS, TCH/EFS, TCH/AFS, TCH/AHS, TCH/F14.4, TCH/F9.6, TCH/H4.8, SACCH, FACCH, E-FACCH, SDCCH, BCCH, PCH, AGCH, RACH, SCH, TCH/F4.8, TCH/F2.4, TCH/H2.4, PDTCH (CS-1, CS-2, CS-3, CS-4), PACCH, PBCCH, PAGCH, PPCH, PNCH, PTCCH, PRACH, CPBCCH, CPAGCH, CPPCH, CPNCH, CPNCH |
| $G_2 = 1 + D^2 + D^4$ | TCH/AFS, TCH/F4.8, TCH/F2.4, TCH/H2.4 |
| $G_3 = 1 + D + D^2 + D^3 + D^4$ | TCH/AFS, TCH/F4.8, TCH/F2.4, TCH/H2.4 |
| $G_4 = 1 + D^2 + D^3 + D^5 + D^6$ | TCH/HS, TCH/AFS, TCH/AHS, E-TCH/F43.2, E-TCH/F32.0, E-TCH/F28.8, PDTCH (MCS-1, MCS-2, MCS-3, MCS-4, MCS-5, MCS-6, MCS-7, MCS-8, MCS-9) |
| $G_5 = 1 + D + D^4 + D^6$ | TCH/HS, TCH/AFS, TCH/AHS, PDTCH (MCS-1, MCS-2, MCS-3, MCS-4, MCS-5, MCS-6, MCS-7, MCS-8, MCS-9) |
| $G_6 = 1 + D + D^2 + D^3 + D^4 + D^6$ | TCH/HS, TCH/AFS, TCH/AHS |
| $G_7 = 1 + D + D^2 + D^3 + D^6$ | E-TCH/F43.2, E-TCH/F32.0, E-TCH/F28.8, PDTCH (MCS-1, MCS-2, MCS-3, MCS-4, MCS-5, MCS-6, MCS-7, MCS-8, MCS-9) |

Annex C (informative): Change control history

| SPEC | SMG | CR | PHA | VERS | NEW_VE | SUBJECT |
|-------|------|------|-----|-------|--------|---|
| 05.03 | s25 | A015 | R97 | 6.0.0 | 6.1.0 | 14.4kbps Data Service |
| 05.03 | s27 | | R97 | 6.1.0 | 6.1.2 | Change of status to EN |
| 05.03 | s28 | A017 | R97 | 6.1.2 | 6.2.0 | Clarification on the definition of USF precoding |
| 05.03 | s28 | A016 | R98 | 6.2.0 | 7.0.0 | Introduction of CTS in 05.03 |
| 05.03 | s28 | | R98 | 7.0.0 | 7.0.1 | Correction to Figure 1 |
| 05.03 | s29 | A021 | R98 | 7.0.1 | 7.1.0 | Introduction of AMR |
| 05.03 | s29 | A022 | R99 | 7.1.0 | 8.0.0 | Introduction of ECSD/EDGE |
| 05.03 | s30 | A023 | R99 | 8.0.0 | 8.1.0 | Introduction of Fast power Control for ECSD in 05.03 |
| 05.03 | s30 | A025 | R99 | 8.0.0 | 8.1.0 | EGPRS Channel Coding |
| 05.03 | s30 | A026 | R99 | 8.0.0 | 8.1.0 | AMR Channel Coding |
| 05.03 | s30 | A027 | R99 | 8.0.0 | 8.1.0 | EDGE Compact logical channels |
| 05.03 | s30 | A029 | R99 | 8.0.0 | 8.1.0 | Correction of several small bugs in the AMR section / Optimization of the transmission of the in-band parameter Mode Indication |
| 05.03 | s30 | A030 | R99 | 8.0.0 | 8.1.0 | E-FACCH/F interleaving |
| 05.03 | s30 | A032 | R99 | 8.0.0 | 8.1.0 | Introduction of RATSCCH for AMR |
| 05.03 | s30b | A033 | R99 | 8.1.0 | 8.2.0 | Correction of EGPRS channel coding |
| 05.03 | s31 | A035 | R99 | 8.2.0 | 8.3.0 | Correction concerning SID_FIRST and clarification concerning bit order of codec mode code words |
| 05.03 | s31 | A036 | R99 | 8.2.0 | 8.3.0 | Editorial correction for ECSD channel coding |
| 05.03 | s31 | A037 | R99 | 8.2.0 | 8.3.0 | Correction for EGPRS Channel Coding |
| 05.03 | S31b | A039 | R99 | 8.3.0 | 8.4.0 | Fast inband signalling: E-IACCH |

History

| Document history | | | |
|-------------------------|------------|-----------------------------|--|
| V8.3.0 | April 2000 | One-step Approval Procedure | OAP 20000825: 2000-04-26 to 2000-08-25 |
| V8.4.0 | May 2000 | One-step Approval Procedure | OAP 20000929: 2000-05-31 to 2000-09-29 |
| | | | |
| | | | |
| | | | |