**EUROPEAN TELECOMMUNICATION STANDARD**

**ETS 300 401**

**February 1995**

Source: EBU/ETSI JTC

Reference: DE/JTC-DAB

ICS: 33.060

**Key words:** Broadcasting, radio, DAB, audio, data broadcasting, mobile/portable/fixed receiver, terrestrial/satellite broadcasting

European Broadcasting Union — Union Européenne de Radio-Télévision

# Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers

## ETSI

European Telecommunications Standards Institute

**ETSI Secretariat**

**Postal address:** F-06921 Sophia Antipolis CEDEX - FRANCE
**Office address:** 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
**X.400:** c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

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

# Contents

## Foreword

This European Telecommunication Standard (ETS) has been produced under the authority of the Joint Technical Committee (JTC) of the European Broadcasting Union (EBU) and the European Telecommunications Standards Institute (ETSI).

NOTE: This EBU/ETSI JTC was established in 1990 to co-ordinate the drafting of ETSs in the specific field of radio, television and data broadcasting.

The EBU is a professional association of broadcasting organizations whose work includes co-ordination of its Members activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 Countries in the European Broadcasting area; its headquarters is in Geneva *.

* European Broadcasting Union
Case Postale 67
CH-1218 GRAND SACONNEX (Geneva)
Switzerland

Tel: +41 22 717 2111
Fax: +41 22 798 5897

This ETS on Digital Audio Broadcasting (DAB)[1)] was prepared by the Project Team PT-20V. The work of the Project Team was based on the studies carried out by the Joint EBU/Eureka 147 Task Force on DAB System Definition. This joint industry/broadcasters group provided the necessary guidance on all relevant technical matters to the Project Team.

The Project Team consisted of members of European broadcasting organizations and the consumer electronics industry.

The ETS on DAB is based on the overall system and service requirements adopted by the ITU-R Recommendations 774 [1] and 789 [2]. The audio coding algorithm used by the DAB system has been subject to the standardization process within the ISO/Moving Pictures Expert Group (MPEG), see ISO/IEC 11172-3 [3]. The layered ISO open system interconnect model ISO 7498 [4] has been used to the extent possible, and interfacing to information technology equipment and communications networks has been taken into account where applicable.

This ETS defines the nature and content of the transmitted DAB signal with reference to the conceptual emission part. The emphasis is given to the normative elements; informative elements are included only to the extent necessary to provide interpretative guidance to the DAB system users and equipment manufacturers.

The DAB system is a novel sound broadcasting system intended to supersede the existing analogue amplitude and frequency modulation systems. It is a rugged, yet highly spectrum and power efficient sound and data broadcasting system. It has been designed for terrestrial and satellite as well as for hybrid and mixed delivery. The DAB system has been publicly demonstrated on a number of occasions during its development. It has been subject to extensive field tests and computer simulations in Europe, USA and Canada.

---

**1)** DAB is a registered trademark owned by one of the Eureka 147 partners.

Blank page

# 1 Scope

This European Telecommunication Standard (ETS) establishes a broadcasting standard for the Digital Audio Broadcasting (DAB) system designed for delivery of high-quality digital audio programme and data services for mobile, portable and fixed reception from terrestrial or satellite transmitters in the Very High Frequency (VHF)/Ultra High Frequency (UHF) frequency bands as well as for distribution through cable networks. The DAB system is designed to provide spectrum and power efficient techniques in terrestrial transmitter network planning, known as the Single Frequency Network (SFN) and the gap-filling technique. The DAB system is suitable for satellite as well as hybrid/mixed terrestrial/satellite broadcasting, using a simple, nearly omni-directional receiving antenna. The DAB system meets the required sharing criteria with other radiocommunication services.

This ETS defines the DAB transmission signal. It includes the coding algorithms for audio, multiplexing of audio programme and data services, channel coding and modulation. A limited range of supplementary services associated with programme services is defined. Provision is also made for transmission of additional data services which may be programme related or not, within the limit of the total system capacity. The ETS provides information on the system configuration which includes information about the ensembles, services, service components and linking of them. Provision is made for a compatible cross-reference to existing Frequency Modulation (FM) services.

This ETS describes the nominal characteristics of the emitted DAB signal. The aspects related to the receiver design are outside the scope of this ETS. Hardware implementation considerations are not covered.

# 2 Normative references

This ETS incorporates, by dated and undated references, provisions from other publications. These normative references are cited at the appropriated places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1] ITU-R Recommendation BS.774 (March 1994): "Digital sound broadcasting to vehicular, portable and fixed receivers using terrestrial transmitters in the VHF/UHF bands".

[2] ITU-R Recommendation BO.789 (March 1994): "Digital sound broadcasting to vehicular, portable and fixed receivers for BSS (sound) in the frequency range 500 - 3000 MHz".

[3] ISO/IEC 11172-3 (March 1993): "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1,5 Mbit/s - Audio Part".

[4] ISO 7498 (1984): "Open Systems Interconnection (OSI) Basic Reference Model".

[5] EN 50067 (April 1992): "Specification of the Radio Data System (RDS)".

[6] ITU-T Recommendation X.25 (1993): "Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit".

[7] ETS 300 250: "Television systems; ETSI/EBU Joint Technical Committee (JTC) Specification of the D2-MAC/packet".

[8] ETS 300 174 (1992): "Network Aspects (NA); Digital coding of component television signals for contribution quality applications in the range 34-45 Mbits/s".

[9] ISO 3901 (1986): "International Standard Recording Code (ISRC)".

[10] Norwegian Telecom, Issue 2 (July 20, 1989): "NR - MSK Access Control System".

[11] EN 50094 (1992): "Access control system for the MAC/packet family: Eurocrypt".

[12] IEC 958 (1989), Amendment 1 (1993)(AES/EBU): "Digital Audio Interface".

[13] CCIR Recommendation 562-3 ITU Radiocommunications Sector, Volume X, (1990): "Subjective assessment of sound quality".

[14] ISO/IEC DIS 13818-3 (March 1994): "Generic coding of moving pictures and associated audio - Audio part".

[15] prEN 797: "Bar coding - Symbology specifcations Universal Product Code/European Article Number (UPC/EAN)".

# 3 Definitions, abbreviations, symbols and conventions

## 3.1 Definitions

For the purposes of this ETS, the following definitions apply:

**Alias component:** Mirrored signal component resulting from sub-Nyquist sampling.

**Announcement cluster:** A group of services which share the same announcement interruption privileges.

**Audio bit stream:** A sequence of consecutive audio frames.

**Audio frame:** A frame of 24 ms duration which contains information of an ISO/IEC 11172-3 [3] Layer II encoded audio signal, corresponding to 1 152 consecutive audio samples at a 48 kHz sampling frequency. It is the smallest part of the audio bit stream which is decodable on its own.

**Audio mode:** The audio coding system provides single channel, dual channel, stereo and joint stereo audio modes. In each mode, the complete audio signal is encoded as one audio bit stream.

**Auxiliary Information Channel (AIC):** All or part of sub-channel 63, used to carry information redirected from the Fast Information Channel.

**Bark:** See "Critical band".

**Bit Allocation (BAl):** A time-varying assignment of bits to samples in different sub-bands according to a psychoacoustic model.

**Bound:** The lowest sub-band in which Intensity stereo coding is used, in the case of Joint stereo mode.

**Capacity Unit (CU):** The smallest addressable unit (64 bits) of the Common Interleaved Frame (CIF).

**Common Interleaved Frame (CIF):** The serial digital output from the main service multiplexer which is contained in the Main Service Channel part of the transmission frame. It is common to all transmission modes and contains 55 296 bits (i.e. 864 CUs).

**Conditional Access (CA):** A mechanism by which the user access to service components can be restricted.

**Convolutional coding:** The coding procedure which generates redundancy in the transmitted data stream in order to provide ruggedness against transmission distortions.

**Critical band:** A psychoacoustic measure in the frequency domain which corresponds to the frequency selectivity of the human ear. The unit of this psychoacoustic measure is called Bark. The Bark scale is a non-linear mapping of the frequency scale over the entire audio frequency range.

**DAB audio frame:** The same as audio frame, but includes all specific DAB audio-related information.

**DAB transmission signal:** The transmitted radio frequency signal.

**Data service:** A service which comprises a non-audio Primary service component and optionally additional Secondary service components.

**Dual channel mode:** The audio mode, in which two audio channels with independent programme contents (e.g. bilingual) are encoded within one audio bit stream. The coding process is the same as for the Stereo mode.

**Energy dispersal:** An operation involving deterministic selective complementing of bits in the Logical frame, intended to reduce the possibility that systematic patterns result in unwanted regularity in the transmitted signal.

**Ensemble:** The transmitted signal, comprising a set of regularly and closely-spaced orthogonal carriers. The ensemble is the entity which is received and processed. In general, it contains programme and data services.

**Ensemble Identifier (EId):** A unique 16-bit code, allocated to an ensemble and intended to allow unambiguous world-wide identification of that ensemble. The individual bits of the code are not individually interpretable.

**Extended Programme Associated Data (X-PAD):** The extended part of the PAD carried towards the end of the DAB audio frame, immediately before the Scale Factor Cyclic Redundancy Check (CRC). Its length is variable.

**Fast Information Block (FIB):** A data burst of 256 bits. The sequence of FIBs is carried by the Fast Information Channel. The structure of the FIB is common to all transmission modes.

**Fast Information Channel (FIC):** A part of the transmission frame, comprising the Fast Information Blocks, which contains the multiplex configuration information together with optional service Information and data service components.

**Fast Information Data Channel (FIDC):** The dedicated part of the Fast Information Channel which is available for non-audio related data services, such as paging.

**Fast Information Group (FIG):** A package of data used for one application in the Fast Information Channel. Eight different types are available to provide a classification of the applications.

**Fixed Programme Associated Data (F-PAD):** The fixed part of the PAD contained in the last two bytes of the DAB audio frame.

**Intensity stereo coding:** A method of exploiting stereo irrelevance or redundancy in stereophonic audio programmes. It is based on retaining only the energy envelope of the right and left channels at high frequencies. At low frequencies, the fine structure of the left and right channel of a stereophonic signal is retained.

**Joint stereo mode:** The audio mode in which two channels forming a stereo pair (left and right) are encoded within one bit stream and for which stereophonic irrelevance or redundancy is exploited for further bit reduction. The method used in the DAB system is Intensity stereo coding.

**Logical frame:** A data burst, contributing to the contents of a sub-channel, during a time interval of 24 ms. For example, data bursts at the output of an audio encoder, a Conditional Access scrambler and a convolutional encoder are referred to as Logical frames. The number of bits contained in a specific Logical frame depends on the stage in the encoding process and the bit rate associated with the sub-channel.

**Logical frame count:** The value of the CIF counter corresponding to the first CIF which carries data from the Logical frame.

**Main Service Channel (MSC):** A channel which occupies the major part of the transmission frame and which carries all the digital audio service components, together with possible supporting and additional data service components.

**Masking:** Property of the human auditory system by which an audio signal cannot be perceived in the presence of another audio signal.

**Masking threshold:** A function of frequency and time, specifying the sound pressure level below which an audio signal cannot be perceived by the human auditory system.

**MSC data group:** A package of data used for one application in the Main Service Channel. MSC data groups are transported in a series of one or more packets.

**Multiplex Configuration Information (MCI):** Information defining the configuration of the multiplex. It contains the current (and in the case of an imminent re-configuration, the forthcoming) details about the services, service components and sub-channels and the linking between these objects. It is carried in the FIC in order that a receiver may interpret this information in advance of the service components carried in the Main Service Channel. It also includes identification of the ensemble itself and a date and time marker.

**Null symbol:** The first Orthogonal Frequency Division Multiplex (OFDM) symbol of the transmission frame.

**OFDM symbol:** The transmitted signal for that portion of time when the modulating phase state is held constant on each of the equi-spaced, equal amplitude carriers in the ensemble. Each carrier is four-phase differentially modulated from one symbol to another, giving a gross bit rate of two bits per carrier per symbol.

**Packet mode:** The mode of data transmission in which data are carried in addressable blocks called packets. Packets are used to convey MSC data groups within a sub-channel.

**Polyphase filter bank:** A set of equal-bandwidth filters with special phase relationship, allowing for efficient implementation of a filter bank.

**Primary service component:** The first and mandatory component of a service. It can be used as a default selection in the receiver.

**Programme:** A time-slice of a programme service, corresponding to an entry in a programme schedule.

**Programme Associated Data (PAD):** Information which is related to the audio data in terms of contents and synchronization. The PAD field is located at the end of the DAB audio frame.

**Programme item:** A time-slice of a programme, for example, a piece of music or a news report.

**Programme service:** A service which comprises an audio Primary service component and optionally additional Secondary service components.

**Protection level:** A level specifying the degree of protection, provided by the convolutional coding, against transmission errors.

**Protection profile:** Defines the scheme of convolutional coding applied.

**Psychoacoustic model:** A mathematical model of the masking behaviour of the human auditory system.

**Scale Factor (ScF):** A factor by which a set of values is scaled before quantization. The numerical code for the Scale Factor is called the Scale Factor Index.

**Scale Factor Select Information (ScFSI):** A 2-bit code which indicates for each sub-band how many Scale Factors are coded within the audio frame.

**Secondary service component:** In case a service contains more than the Primary service component, the additional service components are Secondary service components.

**Service:** The user-selectable output which can be either a programme service or a data service.

**Service component:** A part of a service which carries either audio (including PAD) or data. The service components of a given service are linked together by the Multiplex Configuration Information. Each service component is carried either in a sub-channel or in the Fast Information Data Channel.

**Service Identifier (SId):** A 16-bit code used to identify a particular service.

**Service Information (SI):** Auxiliary information about services, such as service labels and programme type codes.

**Service label:** Alphanumeric characters associated with a particular service and intended for display in a receiver.

**Side information:** Information in the encoded audio bit stream which is necessary for controlling the audio decoder. This information includes Bit Allocation, Scale Factor Select Information and Scale Factors.

**Single channel mode:** The audio mode, in which a monophonic audio programme is encoded within one bit stream.

**Single Frequency Network (SFN):** A network of DAB transmitters sharing the same radio frequency to achieve a large area coverage.

**Specific Service Component:** A part of a service which is specifically used for specialist data components.

**Stereo mode:** The audio mode, in which two channels forming a stereo pair (left and right) are encoded within one bit stream and for which the coding process is the same as for the Dual channel mode.

**Stream mode:** The mode of data transmission within the Main Service Channel in which data are carried transparently from source to destination. Data are carried in Logical frames.

**Stuffing:** One or more bits which may be inserted into the audio bit stream. Stuffing bits are ignored by the audio decoding process. The purpose is to fill up a data field when required.

**Sub-band:** A subdivision of the audio frequency range. In the audio coding system, 32 sub-bands of equal bandwidth are used.

**Sub-band samples:** The sub-band filter bank in the audio encoder creates a filtered and sub-sampled representation of the input audio signal. The filtered samples are called sub-band samples. From 384 consecutive input audio samples, 12 consecutive sub-band samples are generated for each of the 32 sub-bands.

**Sub-channel:** A part of the Main Service Channel which is individually convolutionally encoded and comprises an integral number of Capacity Units per Common Interleaved Frame.

**Synchronization channel:** A part of the transmission frame providing a phase reference.

**Syncword:** A 12-bit code embedded in the ISO/IEC 11172-3 [3] bit stream that identifies the beginning of an audio frame.

**Transmission frame:** The actual transmitted frame, specific to the three transmission modes, conveying the Synchronization channel, the Fast Information Channel and the Main Service Channel.

**Transmission mode:** A specific set of transmission parameters (e.g. number of carriers, OFDM symbol duration). Three transmission modes (i.e. I, II, III) are defined to allow the system to be used for different network configurations and a range of operating frequencies.

**Unequal Error Protection (UEP):** An error protection procedure which allows the bit error characteristics to be matched with the bit error sensitivity of the different parts of the bit stream.

### 3.2 Abbreviations

For the purposes of this ETS, the following abbreviations apply:

| | |
|---|---|
| ACS | Access Control System |
| AIC | Auxiliary Information Channel |
| ASCTy | Audio Service Component Type |
| BAl | Bit Allocation |
| CA | Conditional Access |
| CAId | Conditional Access Identifier |
| CCIR | Comité Consultatif International des Radiocommunications |
| CI | Contents Indicator |
| CIF | Common Interleaved Frame |
| C/N | Current/Next |
| CRC | Cyclic Redundancy Check |
| CU | Capacity Unit |
| CW | Control Word |
| D-QPSK | Differential QPSK |
| DAB | Digital Audio Broadcasting |
| DGCA | Data Group Conditional Access |
| DRC | Dynamic Range Control |
| DSCTy | Data Service Component Type |
| EAN | European Article Number |
| ECC | Extended Country Code |
| ECM | Entitlement Checking Message |
| EId | Ensemble Identifier |
| EMM | Entitlement Management Message |
| ETS | European Telecommunication Standard |
| EWS | Emergency Warning Systems |
| F-PAD | Fixed Programme Associated Data |
| FF | Flagfield Format |
| FFT | Fast Fourier Transform |
| FI | Frequency Information |
| FIB | Fast Information Block |
| FIC | Fast Information Channel |
| FIDC | Fast Information Data Channel |
| FIDCId | Fast Information Data Channel Identifier |
| FIG | Fast Information Group |
| FM | Frequency Modulation |
| IM | Initialization Modifier |
| ISO | International Organization for Standardization |
| ISRC | International Standard Recording Code |
| ITTS | Interactive Text Transmission System |
| IW | Initialization Word |
| LFN | Logical Frame Number |
| LSb | Least Significant bit |
| LSB | Least Significant Byte |
| LTO | Local Time Offset |
| MainId | Main Identifier of a Transmitter |
| MCI | Multiplex Configuration Information |
| MJD | Modified Julian Date |
| MPEG | Moving Pictures Expert Group |
| M/S | Music/Speech |
| MSb | Most Significant bit |
| MSB | Most Significant Byte |
| MSC | Main Service Channel |
| NCC | Number of Coarse Codes |
| NFC | Number of Fine Codes |
| OE | Other Ensemble |
| OFDM | Orthogonal Frequency Division Multiplex |
| OSI | Open Systems Interconnection |
| PAD | Programme Associated Data |
| PCM | Pulse Coded Modulation |

| | |
|---|---|
| PIN | Programme Item Number (RDS) |
| PNum | Programme Number |
| PRBS | Pseudo-Random Binary Sequence |
| PTy | Programme Type |
| QPSK | Quadrature Phase Shift Keying |
| RDS | Radio Data System |
| Rfa | Reserved for future addition |
| Rfu | Reserved for future use |
| SCCA | Service Component Conditional Access |
| ScF | Scale Factor |
| ScF-CRC | audio Scale Factor - Cyclic Redundancy Check (error check) |
| ScFSI | Scale Factor Select Information |
| SCId | Service Component Identifier |
| SCTy | Service Component Type |
| S/D | Static/Dynamic |
| SFN | Single Frequency Network |
| SI | Service Information |
| SId | Service Identifier |
| SMR | Signal-to-Mask Ratio |
| SSCTy | Specific Service Component Type |
| SubCh | Sub-channel of the Main Service Channel |
| SubChId | Sub-channel Identifier |
| SubId | Sub-Identifier of a Transmitter |
| TCId | Type Component Identifier |
| TII | Transmitter Identification Information |
| TMC | Traffic Message Channel |
| TMId | Transport Mechanism Identifier |
| UEP | Unequal Error Protection |
| UHF | Ultra High Frequency |
| UPC | Universal Product Code |
| UTC | Co-ordinated Universal Time |
| VHF | Very High Frequency |
| X-PAD | Extended Programme Associated Data |

## 3.3 Mathematical symbols

For the purposes of this ETS, the following mathematical symbols apply:

### 3.3.1 Arithmetic operators

| | |
|---|---|
| ^ | Power |
| / | Integer division with truncation of the result toward zero. For example, 7/4 and -7/-4 are truncated to 1 and -7/4 and 7/-4 are truncated to -1 |
| Q(a/b) | Q(a/b) is the quotient part of the division of a by b (a and b positive integers) |
| R(a/b) | R(a/b) is the remainder of the division of a by b |

mod(a,b) (b positive integer) $\mathrm{mod}(a,b) = \begin{cases} \mathrm{R}(a/b) & \text{if } a \text{ is a positive integer} \\ \mathrm{R}((b - \mathrm{R}(-a/b))/b) & \text{if } a \text{ is a negative integer} \end{cases}$

| | |
|---|---|
| (mod p) | Modulo p operation |

### 3.3.2 Logical and set operators

| | |
|---|---|
| max [,...,] | The maximum value in the argument list |
| min [,...,] | The minimum value in the argument list |
| $\oplus$ | Exclusive or |
| $\cap$ | Set intersection |
| $\cup$ | Set union |
| \ | Set exclusion: {-3, -2, ...,3} \ {0} is the set of integers {-3, -2, -1, 1, 2, 3} |

### 3.3.3 Functions

| | |
|---|---|
| sin | Sine |
| cos | Cosine |
| exp | Exponential |
| $e^{(.)}$ | Exponential function |
| $\sqrt{\ }$ | Square root |
| log10 | Logarithm to base 10 |
| *j* | Imaginary unit, $j^2=-1$ |
| Rect | $\mathrm{Rect}(x) = \begin{cases} 1 & \text{if } 0 \le x < 1 \\ 0 & \text{elsewhere} \end{cases}$ |
| δ | Kronecker symbol $\delta(\mathrm{i},\mathrm{j}) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \ne j \end{cases}$ |

### 3.3.4 Constants

| | |
|---|---|
| **π** | 3,14159265359... |
| **e** | 2,71828182846... |

## 3.4 C-language mathematical symbols

### 3.4.1 Arithmetic operators

| | |
|---|---|
| + | Addition |
| - | Subtraction (as a binary operator) or negation (as a unary operator) |
| ++ | Increment |
| - - | Decrement |
| * | Multiplication |
| DIV | Integer division with truncation of the result toward -∞ |
| % | Modulo operator. Defined only for positive numbers |
| $\log_{10}$ | Logarithm to base 10 |

### 3.4.2 Logical operators

| | |
|---|---|
| \|\| | Logical OR |

### 3.4.3 Relational operators

| | |
|---|---|
| > | Greater than |
| ≥ | Greater than or equal to |
| < | Less than |
| ≤ | Less than or equal to |
| == | Equal to |
| != | Not equal to |

### 3.4.4 Assignment

| | |
|---|---|
| = | Assignment operator |

### 3.4.5 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded DAB audio bit-stream:

| | |
|---|---|
| **bslbf** | Bit string, left bit first, where "left" is the order in which bit strings are written in the ETS. Bit strings are written as a string of 1s and 0s within single quote marks, e.g. "1000 0001". Blanks within a bit string are for ease of reading and have no significance. |
| **bound** | Number of first sub-band in joint stereo mode. |
| **ch** | Channel. If ch has the value 0 the left channel of a stereo signal or the first of two independent audio signals is indicated. |
| **chlimit** | No. of channels. |
| **dscf** | Difference between two Scale Factors. |
| **gr** | Granule of three sub-band samples per sub-band. |
| **nbal** | Number of allocated bits per sub-band sample. |
| **nch** | Number of channels; equal to 1 for single channel mode, 2 in other modes. |
| **rpchof** | Remainder polynomial coefficients, highest order first. |
| **sb** | Sub-band. |
| **sblimit** | The number of the lowest sub-band for which no bits are allocated. |
| **scfsi** | Scale Factor selection information. |
| **uimsbf** | Unsigned integer, most significant bit first. |

The byte order of multi-byte words is most significant byte first.

### 3.4.6 Method of describing bit stream syntax

The bit stream described in clause 7 is the bit stream that exists in the DAB-receiver at the interface between channel decoder and audio decoder. The bit stream is described using the 'C' software language which is used to program the processor which assembles the programme audio and associated data for channel coding. Each data item in the bit stream is in bold type. It is described by its name, its length in bits, and a mnemonic for its type and order of transmission.

The action caused by a decoded data element in a bit stream depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and definition of the state variables used in their decoding are described in annex B. The following constructs are used to express the conditions when data elements are present, and are in normal type.

NOTE: This syntax uses the 'C'-code convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true.

| | |
|---|---|
| while ( condition ) {<br>   **data_element**<br>      **. . .**<br>} | If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true. |
| do {<br>   **data_element**<br>      **. . .** | The data element always occurs at least once. |
| } while ( condition ) | The data element is repeated until the condition is not true. |
| if ( condition)  {<br>   **data_element**<br>      **. . .**<br>} | If the condition is true, then the first group of data elements occurs next in the data stream. |
| else  {<br>   **data_element**<br>      **. . .**<br>} | If the condition is not true, then the second group of data elements occurs next in the data stream. |
| for (expr1; expr2; expr3)  {<br>   **data_element**<br>      **. . .**<br>} | expr1 is an expression specifying the initialization of the loop. Normally it specifies the initial state of the counter. expr2 is a condition specifying a test made before each iteration of the loop. The loop terminates when the condition is not true. expr3 is an expression that is performed at the end of each iteration of the loop, normally it increments a counter. |

NOTE: The most common usage of this construct is as follows.

| | |
|---|---|
| for (  i = 0; i < n; i++)  {<br>   **data_element**<br>      **. . .**<br>} | The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to one for the second occurrence, and so forth. |

As noted, the group of data elements may contain nested conditional constructs. For compactness, the {} may be omitted when only one data element follows.

| | |
|---|---|
| **data_element []** | data_element [] is an array of data. The number of data elements is indicated by the context. |
| **data_element [n]** | data_element [n] is the (n+1)th element of an array of data. |
| **data_element [m][n]** | data_element [m][n] is the (m+1),(n+1)th element of a two-dimensional array of data. |
| **data_element [l][m][n]** | data_element [l][m][n] is the (l+1),(m+1),(n+1)th element of a three-dimensional array of data. |
| **data_element [m..n]** | is the inclusive range of bits between bit m and bit n in the data_element. |

### 3.5 Convention

Unless otherwise stated, the following notation, regarding the order of bits within each step of processing is used:

- in figures, the bit shown in the left hand position is considered to be first;

- in tables, the bit shown in the left hand position is considered to be first;

- in byte fields, the Most Significant bit (MSb) is considered to be first and denoted by the higher number. For example, the MSb of a single byte is denoted '$b_7$' and the Least Significant bit (LSb) is denoted '$b_0$';

- in vectors (mathematical expressions), the bit with the lowest index is considered to be first.

  NOTE: Due to time-interleaving, this order of bits is not the true transmission order.

# 4 Basic DAB system description

The conceptual block diagram of the emission part of the DAB system is given in figure 1. Each block is labelled in order to indicate the function it performs.

This ETS gives the description of the individual blocks in terms of their input/output transfer functions, as appropriate. The sequence of clauses in this ETS generally follows the information flow in the left-to-right direction. The functions of the DAB system related to general transport mechanisms and multiplex control are given in clauses 5 and 6, respectively. Clause 7 describes a main function of the DAB system, i.e. audio coding. This is followed by the description of the available data features, in clause 8. Clause 9 specifies the Conditional Access mechanisms of the DAB System. Clauses 10 to 15 give the description of the transmission-related functions, as shown in the block diagram.

## 4.1 Transport mechanisms

General transport mechanisms used in the DAB system for transmission of digital audio programme and data services are described in clause 5.

Two mechanisms for transporting the data are defined: the FIC and the MSC.

The primary function of the FIC, which is made up of Fast Information Blocks (FIB), is to carry control information necessary to interpret the configuration of the MSC. The essential part of this control information is the Multiplex Configuration Information (MCI), which contains information on the multiplex structure and, when necessary, its re-configuration. Other types of information which can be included in the FIC represent the Service Information (SI), the Conditional Access (CA) management information and Fast Information Data Channel (FIDC). In order to allow a rapid and safe response to the MCI, the FIC is transmitted without time interleaving, but with a high level of protection against transmission errors.

The MSC is made up of a sequence of Common Interleaved Frames (CIF). A CIF is a data field of 55 296 bits, transmitted every 24 ms. The smallest addressable unit of the CIF is the Capacity Unit (CU), the size of which is 64 bits. Integral number of CUs are grouped together to constitute the basic transport unit of the MSC, called sub-channel. The MSC constitutes therefore a multiplex of sub-channels.

For service components in the MSC, two different transport modes are defined, the stream mode and the packet mode.

The stream mode provides a transparent transmission from source to destination at a fixed bit rate in a given sub-channel.

The packet mode is defined for the purpose of conveying several data service components into a single sub-channel. Each sub-channel may carry one or more service components.

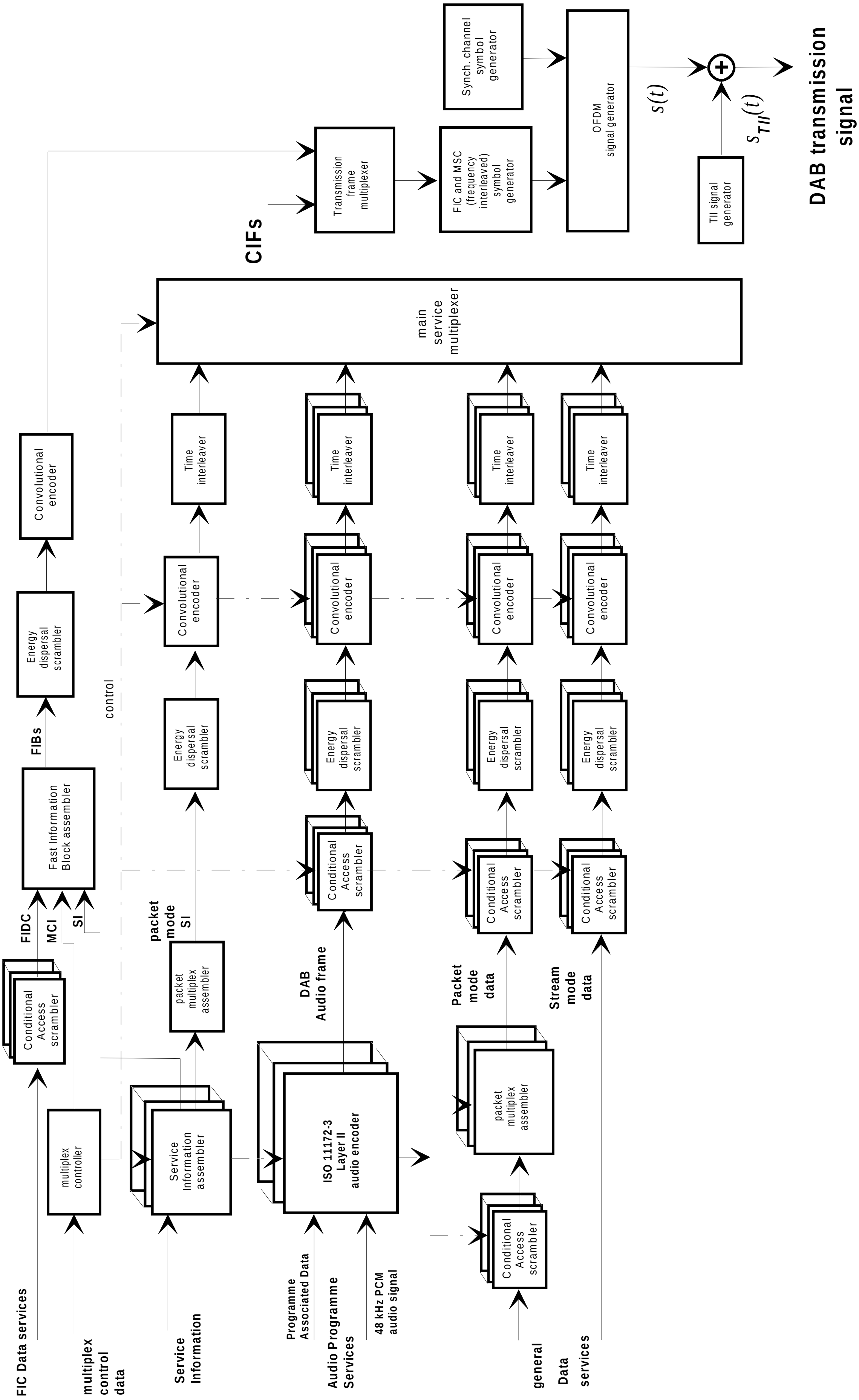


**Figure 1: Conceptual DAB emission block diagram**

## 4.2 Multiplex Configuration Information (MCI)

The Multiplex Configuration Information (MCI) of the system is described in clause 6.

The MCI is carried in the FIC. The MCI basically describes how the DAB multiplex is organized. Specifically, it provides the following information:

a) defines the sub-channel organization;

b) lists the services available in the ensemble;

c) establishes links between services and service components;

d) establishes links between sub-channels and service components;

e) manages the multiplex re-configuration.

## 4.3 Audio coding

The coding algorithm applied to the audio signals as well as the structure of the encoded bit stream are described in clause 7. Additional information on the operations implemented at the encoder and at the decoder is given in the annexes A, B and C.

The DAB system uses Layer II of ISO/IEC 11172-3 [3], suitably formatted for DAB transmission.

The encoder processes the input Pulse Coded Modulation (PCM) audio signal, sampled at 48 kHz, and produces the compressed audio bit stream of different bit rates ranging from 32 kbit/s to 192 kbit/s per mono or twice this bit rate for the stereo mode.

Four audio modes are provided:

a) single channel (i.e. mono) mode;

b) dual channel (i.e. two mono channels) mode;

c) stereo mode;

d) joint stereo mode.

The Layer II of ISO/IEC 11172-3 [3] contains the basic filtering of the digital audio input into 32 sub-bands, fixed segmentation to format the data into blocks, a psychoacoustic model to determine the Bit Allocation (BAI), and quantization using block companding and frame coding. It also provides coding of the BAI, ScFs and audio sample data. The psychoacoustic model is not rigidly determined and may use various estimations of the auditory masking thresholds; nevertheless, the DAB audio frame shall conform to the provisions of clause 7.

Each audio frame contains a number of bytes which may be used to carry the Programme Associated Data (PAD), i.e. the information that is related to the audio in terms of contents and synchronization. The PAD contains two bytes of Fixed PAD (F-PAD), and an optional extension called the Extended PAD (X-PAD). Functions available for the PAD include Dynamic Range Control (DRC), music/speech indication, programme-related text, etc.

## 4.4 Data features

The provision of data features which can be carried either in the MSC or in the FIC, is described in clause 8.

The Service Information (SI) features which may be carried in the FIC include the following: Service component language, Service linking, time and country identifier, Programme Number (PNum), Programme Type (PTy), announcements, Regional identification, Local service area, Other Ensembles, FM services information, Frequency Information (FI), Transmitter Identification Information (TII), ensemble and service labels, etc. It is also possible to redirect certain features to the MSC.

The Fast Information Data Channel (FIDC) carried in the FIC includes provision for paging, Traffic Message Channel (TMC), Emergency Warning System (EWS), etc. It is generally considered that, due to the limited capacity of the FIC, priority should be given to the system-control related information, i.e. the MCI.

## 4.5 Conditional Access (CA)

General provisions for Conditional Access (CA) are given in clause 9.

The purpose of CA is to permit the service and/or service components to be made incomprehensible to unauthorized users.

This ETS describes the scrambling procedures for Stream and Packet modes and for the Fast Information Data Channel (FIDC). The parameters used to provide signalling and synchronization for the CA are described.

The mechanisms to send dedicated messages called Entitlement Checking Messages (ECM) and Entitlement Management Messages (EMM) are also described. The MCI includes the appropriate parameters to indicate whether service components are scrambled or not, and how to find the parameters necessary for descrambling.

## 4.6 Energy dispersal

Clause 10 describes the energy dispersal of the DAB signal. The purpose is to avoid the transmission of signal patterns which might result in an unwanted regularity in the transmitted signal.

## 4.7 Convolutional coding

The process of convolutional coding is applied at the output of each energy dispersal scrambler. The convolutional coding process described in clause 11 consists of generating redundancy as part of the error protection mechanism required to combat adverse propagation conditions.

The convolutional coding parameters depend on the type of service carried, the net bit rate, and the desired level of error protection. In the case of audio service components, the coding rate varies during each DAB audio frame according to an Unequal Error Protection (UEP) profile which is pre-determined for that service component.

## 4.8 Time interleaving

The process of time interleaving described in clause 12 is applied at the output of each convolutional encoder contributing to the sub-channels in the MSC. It is not applied to the FIC.

### 4.9 Common Interleaved Frame

Clause 13 describes how the convolutionally-encoded and time-interleaved Logical frames constituting the sub-channels, are combined into a structure called Common Interleaved Frame (CIF). A CIF consists of 55 296 bits, grouped into 864 Capacity Units (CU) and is transmitted every 24 ms. The configuration of the CIF is signalled by the Multiplex Configuration Information (MCI) carried in the FIC.

### 4.10 DAB transmission signal

The description of the DAB transmission signal in the temporal domain is given in clause 14. In order to allow the DAB system to be used in different transmission network configurations and over a wide range of operating frequencies, three transmission modes are defined. The transmitted signal has a frame structure of 96 ms duration (Transmission mode I) and 24 ms (Transmission modes II and III) and consists of consecutive Orthogonal Frequency Division Multiplex (OFDM) symbols. The OFDM symbols are generated from the output of the multiplexer which combines the CIFs and the convolutionally encoded FIBs. Their generation involves the processes of Differential Quadrature Phase Shift Keying (D-QPSK), frequency interleaving, and D-QPSK symbols frequency multiplexing (OFDM generator).

The transmission frame consists of a sequence of three groups of OFDM symbols: synchronization channel symbols, Fast Information Channel symbols and Main Service Channel symbols. The synchronization channel symbols comprise the null symbol and the phase reference symbol.

The null symbols are also used to allow a limited number of OFDM carriers to convey the Transmitter Identification Information (TII).

### 4.11 Radio frequency parameters

Clause 15 specifies the permitted values of the central frequency of the DAB ensemble and indicates the frequency limits under which the three DAB transmission modes are designed to operate. Elements on time and spectrum characteristics of the emitted signal are also given.

### 4.12 Main DAB system characteristics

The main DAB system characteristics are summarized in table 1.

**Table 1: Main DAB system characteristics**

| | **Main DAB system elements** | **Features** | | **Clause** |
|---|---|---|---|---|
| 1 | DAB System inputs | audio Programme services | audio input format: PCM audio samples<br><br>sampling rate: 48 kHz<br><br>input resolution: up to 22 bits/sample | annex A.1 |
| | | Programme Associated Data<br><br>Service Information<br><br>Multiplex Configuration Information<br><br>FIC Data services<br><br>general data services | | |
| 2 | Audio coding standard ISO/IEC 11172-3 Layer II [3] | Audio modes | single channel<br>stereo<br>dual channel<br>joint stereo | 7.2.1.3<br>annex A.2 |
| | | audio bit rates | 32, 48, 56, 64, 80, 96, 112, 128, 160, 192 kbit/s for single channel<br><br>64, 96, 112, 128, 160, 192, 224, 256, 320, 384 kbit/s for stereo, joint stereo and dual channel | |
| | | Audio frame duration | 24 ms | |
| 3 | Main user information contained in audio header | Audio mode<br>copyright<br>original/copy | | 7.2.1.3 |

**(continued)**

**Table 1: Main DAB system characteristics (continued)**

| | | | | |
|---|---|---|---|---|
| 4 | Programme Associated Data services (PAD) | fixed PAD<br>available data capacity: 667 bit/s<br><br>extended PAD | dynamic range control<br>music/speech indication<br>command channel<br>ISRC and UPC/EAN<br><br>programme related text (ITTS)<br>table of contents<br>dynamic label segment<br>in-house information | 7.4<br>annex A.3 and A.4 |
| 5 | Protection mechanisms for audio | Audio side information CRC<br>Scale Factor CRC | | 7.3.1.4<br>7.3.2.9<br>annex E |
| 6 | Fast Information Block (FIB) | Data field of size 256 bits carrying the FIC information; independent on the transmission mode | Different types of data can be contained in a FIB (see rows 10,11, 12, 14) | 5.2 |
| 7 | Transport modes in the MSC | Stream mode<br><br>Packet mode | one service component per sub-channel can be transmitted transparently at a fixed bit rate<br><br>The packet structure supports multiplexing of various service components in one sub-channel; a sequence of packets is conveyed in a sub-channel | 5.3.1<br><br>5.3.2<br>5.3.3 |
| 8 | General data transport mechanisms | Main Service Channel (MSC) - time and frequency interleaved<br><br>Fast Information Channel (FIC) - frequency interleaved | MSC carries audio and data service components<br><br>FIC carries the Multiplex Configuration Information (MCI), SI and optionally data service components; allows for rapid access of information by the receiver | 5 |
| 9 | Multiplex Configuration Information (MCI) | The MCI provides repetitive information about ensembles, services, service components, sub-channels and linking of them | The MCI of a future multiplex configuration is sent in advance to allow for continuity of services | 6<br>annex D |

**(continued)**

**Table 1: Main DAB system characteristics (continued)**

| | | | | |
|---|---|---|---|---|
| 10 | Service Information (SI) | Service component language<br>Service linking<br>Time and country identifier<br>Programme Number<br>Programme type<br>Announcements<br>Service trigger<br>Frequency Information<br>Other ensembles<br>FM services<br>Transmitter identification<br>Regional identification<br>Local service area<br>Ensemble label<br>Service label | | 8.1 |
| 11 | Auxiliary Information Channel (AIC) | A part of sub-channel 63 in the MSC is used to carry data which cannot be accommodated in the FIC | Packet address 1023 | 5.4<br>8.1.12 |
| 12 | Fast Information Data Channel (FIDC) | Fast access information carried in the FIC | Paging<br>Traffic Message Channel<br>Emergency Warning System | 8.2 |
| 13 | Conditional Access (CA) | The CA may be applied to audio and data service components | scrambling/descrambling<br>entitlement checking<br>entitlement management | 9 |
| 14 | Energy dispersal | Energy dispersal is applied to avoid unwanted regularity in the transmitted signal | | 10 |
| 15 | Channel protection mechanisms | Convolutional coding is applied. | audio:<br>5 protection levels with Unequal Error Protection (UEP);<br><br>data:<br>4 protection levels with equal error protection | 11 |
| 16 | Time interleaving | interleaving depth:<br>16 Logical frames<br>(384 ms) | | 12 |
| 17 | Common Interleaved Frame (CIF) | Data field of size 55 296 bits, carrying the MSC information; independent of transmission mode | | 13 |

**(continued)**

**Table 1: Main DAB system characteristics (concluded)**

| | | | | |
|---|---|---|---|---|
| 18 | Frequency interleaving | distributes the convolutionally encoded data over the bandwidth of 1,5 MHz | | 14.6 |
| 19 | DAB Transmission frame | comprises the following OFDM symbols:<br>- Null symbol<br>-Phase Reference symbol<br>- FIC symbols<br>- MSC symbols | | 14.1 |
| 20 | Transmission modes | Transmission mode I: intended for SFN in Bands I, II and III<br><br>Transmission mode II: intended for local services in Bands I, II, III, IV, V and L-band<br><br>Transmission mode III: intended for frequencies below 3 GHz and cable | no. of carriers: 1536<br>carrier spacing: 1 kHz<br>symbol duration: 1,246 ms<br>guard interval: 246 µs<br><br>no. of carriers: 384<br>carrier spacing: 4 kHz<br>symbol duration: 312 µs<br>guard interval: 62 µs<br><br>no. of carriers: 192<br>carrier spacing: 8 kHz<br>symbol duration: 156 µs<br>guard interval: 31 µs | 14.2 |
| 21 | Modulation | differentially encoded Quadrature Phase Shift Keying (QPSK) | | 14.7 |
| 22 | RF characteristics | Time and spectrum characteristics | | 15 |

# 5 Transport mechanisms

## 5.1 Introduction

The DAB system is designed to carry several digital audio signals together with data signals. Audio and data signals are considered to be service components which can be grouped together to form services (see clause 6). This subclause describes the main transport mechanisms available in the DAB multiplex.

The DAB transmission system combines three channels (see also subclause 14.1):

1) **Main Service Channel (MSC):** used to carry audio and data service components. The MSC is a time-interleaved (see clause 12) data channel divided into a number of sub-channels which are individually convolutionally coded, with equal or unequal error protection (see subclause 11.3). Each sub-channel may carry one or more service components. The organisation of the sub-channels and service components is called the multiplex configuration;

2) **Fast Information Channel (FIC):** used for rapid access of information by a receiver. In particular it is used to send the Multiplex Configuration Information (MCI) (see clause 6) and optionally Service Information and data services (see clause 8). The FIC is a non-time-interleaved data channel with fixed equal error protection (see subclause 11.2);

3) **Synchronization channel:** used internally within the transmission system for basic demodulator functions, such as transmission frame synchronization, automatic frequency control, channel state estimation, and transmitter identification. The synchronization channel is described in subclause 14.3 and no further details are given here.

Each channel supplies data from different sources and these data are provided to form a transmission frame (see also figure 1: general block diagram, clause 4). A more detailed description is given in subclause 14.2.

Both the organization and length of a transmission frame depend on the transmission mode (see clause 14 and subclause 15.1). The Fast Information Block (FIB) and the Common Interleaved Frame (CIF) are introduced in order to provide transmission mode independent data transport packages associated with the FIC and MSC respectively (see figure 2).

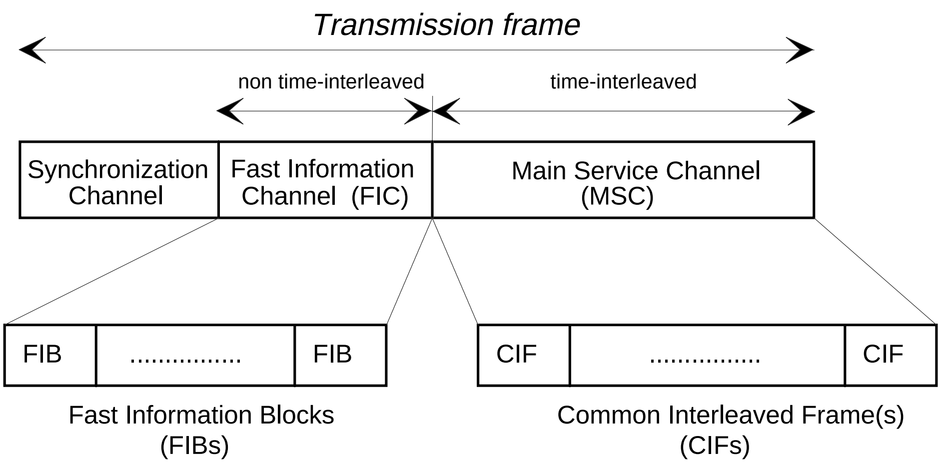


Figure 2: Transmission mode independent description of the FIC and MSC

Table 2 gives the transmission frame duration and the number of FIBs and CIFs which are associated with each transmission frame for the three transmission modes.

**Table 2: General transport characteristics of the transmission frame**

| Transmission mode | Duration of transmission frame | Number of FIBs per transmission frame | Number of CIFs per transmission frame |
|---|---|---|---|
| I | 96 ms | 12 | 4 |
| II | 24 ms | 3 | 1 |
| III | 24 ms | 4 | 1 |

In transmission mode I, the 12 FIBs contributing to one transmission frame shall be divided into four groups which are each assigned to one of the CIFs contributing to the same transmission frame. The information contained in the first three FIBs shall refer to the first CIF, the information contained in the fourth, fifth and sixth FIB to the second CIF, and so on. All FIBs contributing to a transmission frame, in transmission modes II and III, shall be assigned to the one CIF associated with that transmission frame.

The following subclauses describe the formation of the FIC and MSC.

## 5.2 The Fast Information Channel (FIC)

The FIC is made up of FIBs.

### 5.2.1 Fast Information Block (FIB)

The general structure of the FIB is shown in figure 3, for a case when the useful data does not occupy the whole of a FIB data field. The FIB contains 256 bits and comprises an FIB data field and a CRC.

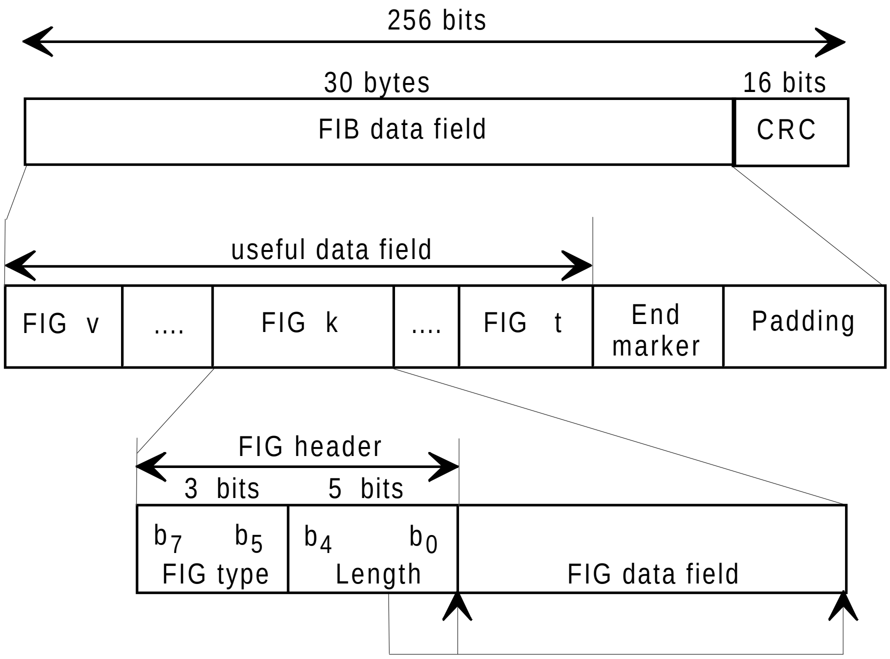


**Figure 3: Structure of the FIB**

**FIB data field:** the FIB data field shall be organized in bytes allocated to useful data, an end marker and padding in the following way:

- the useful data occupy the whole 30 bytes of the FIB data field. In this case, there shall be no end marker and no padding bytes;
- the useful data occupy 29 bytes of the FIB data field. In this case, there shall be an end marker but no padding bytes;
- the useful data occupy less than 29 bytes. In this case, there shall be both an end marker and padding bytes;
- there is no useful data. In this case, the FIB data field shall begin with an end marker and the rest of the FIB data field contains padding bytes.

The FIB data field is described as follows:

- **Useful data field:** this contains one or more Fast Information Groups (FIGs) (see subclause 5.2.2);
- **End marker:** is a special FIG and shall have a FIG header field (111 11111) and no FIG data field;
- **Padding:** this field shall contain the bytes required to complete the FIB data field. The padding byte field shall contain all zeroes.

**CRC:** a 16-bit Cyclic Redundancy Check word is calculated on the FIB data field and shall be generated according to the procedure defined in annex E. The generation shall be based on the polynomial $G(x) = x^{16} + x^{12} + x^{5} + 1$ (ITU-T Recommendation X.25 [6]).

At the beginning of each CRC word calculation, all shift register stage contents shall be initialized to "1". The CRC word shall be complemented (1s complement) prior to transmission.

### 5.2.2 Fast Information Group (FIG)

The FIG shall comprise the FIG header and the FIG data field (see figure 3). The following definitions apply:

**FIG header**: shall contain the FIG type and the length:

- **FIG type**: is a 3-bit field which shall indicate the type of data contained in the FIG data field. The assignment of FIG types is given in table 3;

- **Length**: this 5-bit field shall represent the length in bytes of the FIG data field and is expressed as an unsigned binary number (MSb first) in the range 1 - 29. Values 0, 30 and 31 shall be reserved for future use of the FIG data field except for 31 (11111) when used with FIG type 7 (111) which is used for the end marker.

**FIG data field:** this field is described in subclauses 5.2.2.1 to 5.2.2.4 and 6.2 to 6.4 and clause 8.

**Table 3: List of FIG types**

| FIG type number | FIG type | FIG application |
|---|---|---|
| 0 | 000 | MCI and part of the SI |
| 1 | 001 | Labels, etc. (part of the SI) |
| 2 | 010 | Reserved |
| 3 | 011 | Reserved |
| 4 | 100 | Reserved |
| 5 | 101 | FIC Data Channel (FIDC) |
| 6 | 110 | Conditional Access (CA) |
| 7 | 111 | In house (except for Length 31) |

Generally, FIGs may be arranged in any order except where special operational requirements dictate otherwise. FIGs shall not be split between FIBs. FIG types 0, 1, 5 and 6 are defined in subclauses 5.2.2.1 to 5.2.2.4.

### 5.2.2.1 FIG type 0 data field

The FIG type 0 is used to signal the current and future multiplex configuration, a multiplex re-configuration, time and date and other basic Service Information. The structure of the FIG type 0 data field is shown in figure 4.

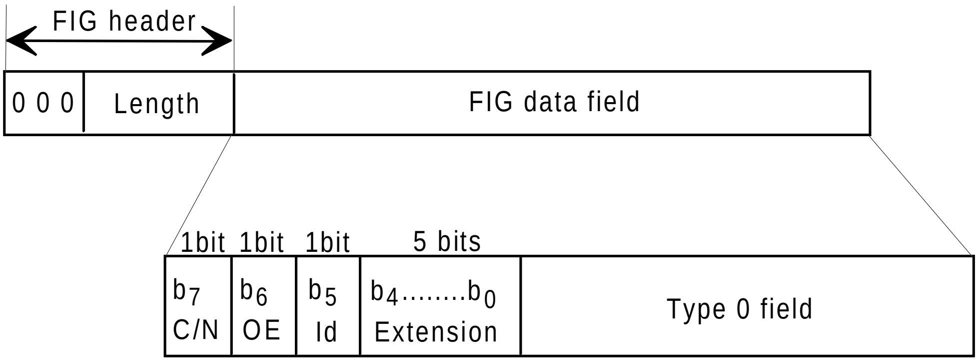


**Figure 4: Structure of the FIG type 0 data field**

The following definitions apply:

**C/N** (Current/Next): this flag shall indicate one of two situations, as follows:

a) the type 0 field applies to the current or next version of the multiplex configuration, as follows:

0 : current configuration;
1 : next configuration;

b) the version number of the type 0 field.

The C/N flag applies to situation (a) for the Extension containing multiplex configuration features: these are Extensions 1 to 4 (see also subclause 6.1). The C/N flag applies to situation (b) for Extensions 11, 18, 21, 24, 25, 27 and 30 (see also subclauses 8.1.16.1, 8.1.6.1, 8.1.8, 8.1.10.2, 8.1.10.5.1, 8.1.11.2.1 and 8.1.18.1). For those Extensions which do not use this flag, the bit $b_7$ shall be reserved for future use of the Type 0 field. This Reserved for future use (Rfu) bit is set to "0" for the currently specified Extension field and Type 0 field.

**OE (**Other Ensemble**)**: this flag shall indicate whether the information is related to this or another ensemble, as follows:

0 : this ensemble;
1 : other ensemble.

The OE flag is used for Extensions 16, 17, 21, 24 and 30 (see subclauses 8.1.4, 8.1.5, 8.1.8, 8.1.10.2 and 8.1.18.1). For those extensions which do not use this flag, the bit $b_6$ shall be reserved for future use of the Type 0 field. This Rfu bit is set to "0" for the currently specified Extension field and Type 0 field.

**Id**: this 1-bit flag shall indicate whether the Service Identifiers (SIds), associated with some data features, are in the 16-bit or 32-bit format and, in the case of the service organisation, the type of service components, as follows:

0 : 16-bit SId;
1 : 32-bit SId.

The Id flag is used for Extensions 2, 3, 23 and 24 (see subclauses 6.3.1, 6.3.2, 8.1.10.2 and 8.1.17). When the Id flag is not used, the Service Identifier (SId) takes the 16-bit format. For those Extensions which do not use this flag, the bit $b_5$ shall be reserved for future use of the Type 0 field. This Rfu bit is set to "0" for the currently specified Extension field and Type 0 field.

NOTE: 16-bit and 32-bit Service Identifiers may not be mixed in the same Type 0 field.

**Extension**: this 5-bit field, expressed as an unsigned binary number, identifies one of 32 interpretations of the FIG type 0 field (see subclauses 6.2, 6.3, 6.4 and 8.1). Those extensions, which are not defined, are reserved for future use.

### 5.2.2.2 FIG type 1 data field

The FIG type 1 is used to signal labels for display and other information defining labels. The structure of the FIG type 1 data field is shown in figure 5.

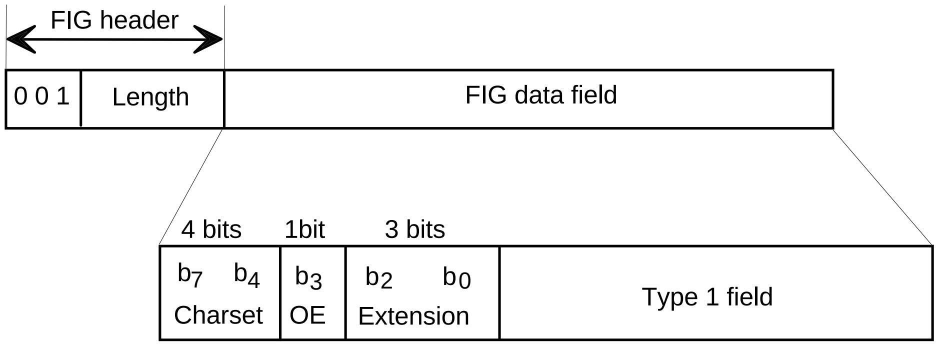


**Figure 5: Structure of the FIG type 1 data field**

The following definitions apply:

**Charset**: this 4-bit field shall identify a character set to qualify the character information contained in the FIG type 1 field. The following character sets are defined (see EN 50067 [5]):

$b_7$ $b_4$
0 0 0 0 : complete EBU Latin based repertoire;
0 0 0 1 : EBU Latin based common core, Cyrillic, Greek;
0 0 1 0 : EBU Latin based core, Arabic, Hebrew, Cyrillic and Greek.

The remaining codes are reserved for future definition.

**OE:** this flag shall indicate whether the information is related to this or another ensemble, as follows:

0 : this ensemble;
1 : other ensemble.

The OE flag is used for Extensions 0 and 1 (see subclauses 8.1.13 and 8.1.14). For those extensions which do not use this flag, the bit $b_3$ shall be reserved for future use. The Rfu bit is set to "0" for the currently specified extension field and FIG type 1 field.

**Extension**: this 3-bit field, expressed as an unsigned binary number, shall identify one of 8 interpretations of the FIG type 1 field (see subclause 8.1). Those extensions, which are not defined, are reserved for future use.

### 5.2.2.3 FIG type 5 data field

The FIG type 5 is used for the Fast Information Data Channel (FIDC). The structure of the FIG type 5 data field is shown in figure 6.

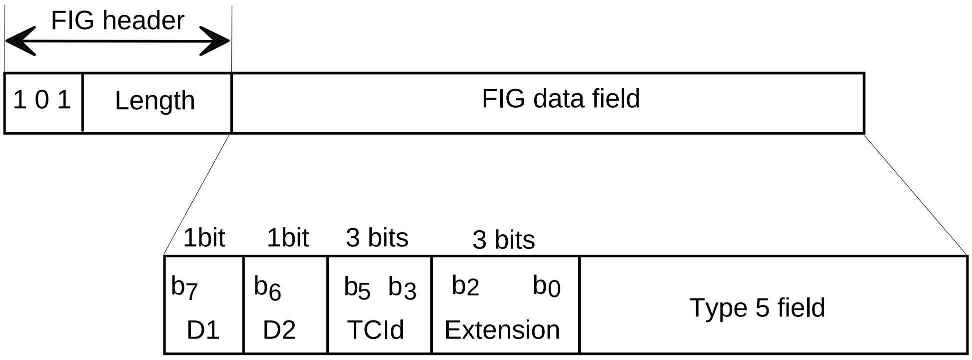


**Figure 6: Structure of the FIG type 5 data field**

The following definitions apply:

**D1, D2**: these 1-bit flags are defined for each extension individually (see subclause 8.2).

**TCId** (Type Component Identifier): this 3-bit field shall identify one of eight different service components which may be carried using the same extension number.

**Extension**: this 3-bit field, expressed as an unsigned binary number, shall identify one of 8 interpretations of the FIG type 5 field (see sublause 8.2). Those extensions, which are not defined, are reserved for future use.

### 5.2.2.4 FIG type 6 data field

The FIG type 6 is used to send the control and management information about a scrambled service component. This information is referred to as CA messages (see clause 9). The structure of the FIG type 6 data field is shown in figure 7.

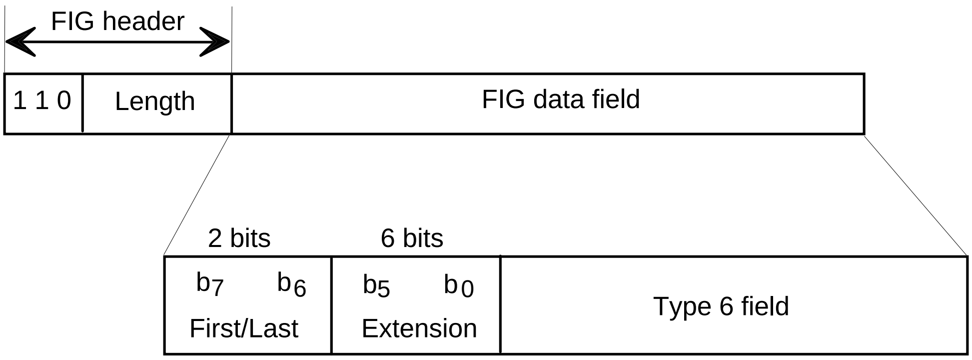


**Figure 7: Structure of the FIG type 6 data field**

The following definitions apply:

**First/Last**: this 2-bit field shall indicate how CA messages are managed if they have to be split into more than one FIG Type 6 field. The flags are set according to table 4.

**Extension**: this 6-bit field, expressed as an unsigned binary number, shall identify one of 64 interpretations of the FIG type 6 field (see subclause 9.3.2.2). Those extensions, which are not defined, are reserved for future use.

**Table 4: First/Last flags for FIG type 6 data fields**

| First $b_7$ | Last $b_6$ | The FIG type 6 data field is the: |
|---|---|---|
| 0 | 0 | intermediate FIG type 6 data field of a series |
| 0 | 1 | last FIG type 6 data field of a series |
| 1 | 0 | first FIG type 6 data field of a series |
| 1 | 1 | one and only one FIG type 6 data field |

## 5.3 The Main Service Channel (MSC)

The MSC is made up of Common Interleaved Frames (CIFs). The CIF contains 55 296 bits. The smallest addressable unit of the CIF is the Capacity Unit (CU), comprising 64 bits. Therefore, the CIF contains 864 CUs, which shall be identified by the CU addresses 0 to 863. The bit structure of the CIF is described in clause 13.

The MSC is divided into sub-channels. Each sub-channel shall occupy an integral number of consecutive CUs and is individually convolutionally encoded. Each CU may only be used for one sub-channel. A service component is a part of a service which carries either audio or general data. The DAB service structure is explained in subclause 6.1.

The data, carried in the MSC, shall be divided at source into regular 24 ms bursts corresponding to the sub-channel data capacity of each CIF. Each burst of data constitutes a logical frame. Each logical frame is associated with a corresponding CIF. Succeeding CIFs are identified by the value of the CIF counter, which is signalled in the MCI (see subclause 6.4). The logical frame count is a notional count which shall be defined as the value of the CIF counter corresponding to the first CIF which carries data from the logical frame.

There are two transport modes in the MSC: one is called the stream mode and the other the packet mode.

### 5.3.1 Stream mode

The Stream mode allows a service application to accept and deliver data transparently from source to destination. At any one time, the data rate of the application shall be fixed in multiples of 8 kbit/s. The application must either supply information on demand, or include a method of handling data asynchronously at a lower rate. Data shall be divided into logical frames.

Only one service component shall be carried in one sub-channel.

### 5.3.2 Packet mode - network level

The packet mode allows different data service components to be carried within the same sub-channel. The permissible data rates shall be multiples of 8 kbit/s, subject only to the total available sub-channel data capacity.

A packet shall be identified by an address. Packets with different addresses may be sent in any order in a sub-channel. However, the sequence of packets with the same address shall be maintained.

Packets shall have a fixed length and four standard packet length types are allowed (see table 5). It is permissible to mix packet types of several lengths in a sub-channel provided that there is an integral number of packets per Logical frame. Padding packets shall be used, if necessary to adjust the data rate to the required multiple of 8 kbit/s.

The links between the service component and the packet address are given in the MCI (see subclause 6.3.2).

A packet shall consist of a Packet header, a Packet data field and a Packet CRC (see figure 8).

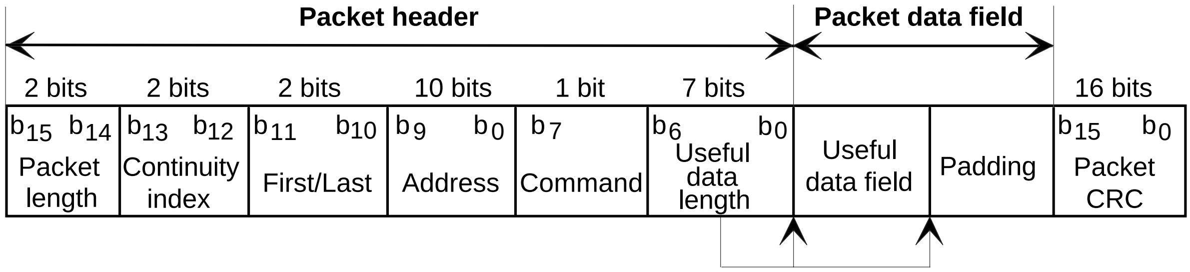


**Figure 8: Packet structure**

### 5.3.2.1 Packet header

The packet header has a length of 3 bytes and it shall comprise the following parameters:

- **Packet length:** four different packet data field lengths are allowed, see table 5;

**Table 5: Packet length**

| Packet Length Indicator $b_{15}$ $b_{14}$ | Packet length (in bytes) | Packet data field length (in bytes) |
|---|---|---|
| 0 0 | 24 | 19 |
| 0 1 | 48 | 43 |
| 1 0 | 72 | 67 |
| 1 1 | 96 | 91 |

- **Continuity index**: this 2-bit, modulo-4 counter shall be incremented by one for each successive packet in a series having the same address. It provides the link between successive packets, carrying the same service component, regardless of length;

- **First / Last**: these flags shall be used to identify particular packets which form a succession of packets, carrying data groups of the same service component (see subclause 5.3.3). The flags shall be assigned as in table 6;

**Table 6: First/Last flags for packet mode**

| First $b_{11}$ | Last $b_{10}$ | The packet is the: |
|---|---|---|
| **0** | **0** | **intermediate packet of a series** |
| **0** | **1** | **last packet of a series** |
| **1** | **0** | **first packet of a series** |
| **1** | **1** | **the one and only packet** |

- **Address:** a 10-bit address code which shall identify packets carrying a particular service component within a sub-channel. Address 0 shall be used for padding packets and shall not be assigned to any service component. Up to 1 023 service components may be carried simultaneously in a sub-channel;

- **Command:** this bit shall indicate whether the packet is used for general data or for special commands (for example, in conjunction with conditional access - see subclauses 9.2.3 and 9.2.4.2) as follows:

  0 : data packet;
  1 : command packet.

- **Useful data length**: the useful data length represents the length in bytes of the associated useful data field in the inclusive range 0 to 91 and shall be coded as an unsigned 7-bit binary number.

### 5.3.2.2 Packet data field

This field contains the useful data field and padding.

**Useful data field**: this field shall contain the useful service component data, assembled as MSC data groups (see subclause 5.3.3).

**Padding**: this field shall comprise the bytes required to complete the packet data field according to the number of bytes given in table 5. The padding byte field shall contain all zeroes.

### 5.3.2.3 Packet CRC

The packet CRC shall be a 16-bit CRC word calculated on the packet header and the packet data field. It shall be generated according to the procedure defined in annex E. The generation shall be based on the polynomial $G(x) = x^{16} + x^{12} + x^5 + 1$ (ITU-T Recommendation X.25 [6]).

At the beginning of each CRC word calculation, all shift register stage contents shall be initialised to "1". The CRC word shall be complemented (1s complement) prior to transmission.

### 5.3.3 Packet mode - transport level

Service component information shall be structured into MSC data groups for transport in one or more packets. A MSC data group shall contain a data group header, an optional session header, a data group data field and an optional data group CRC. The structure of the data group is shown in figure 9.

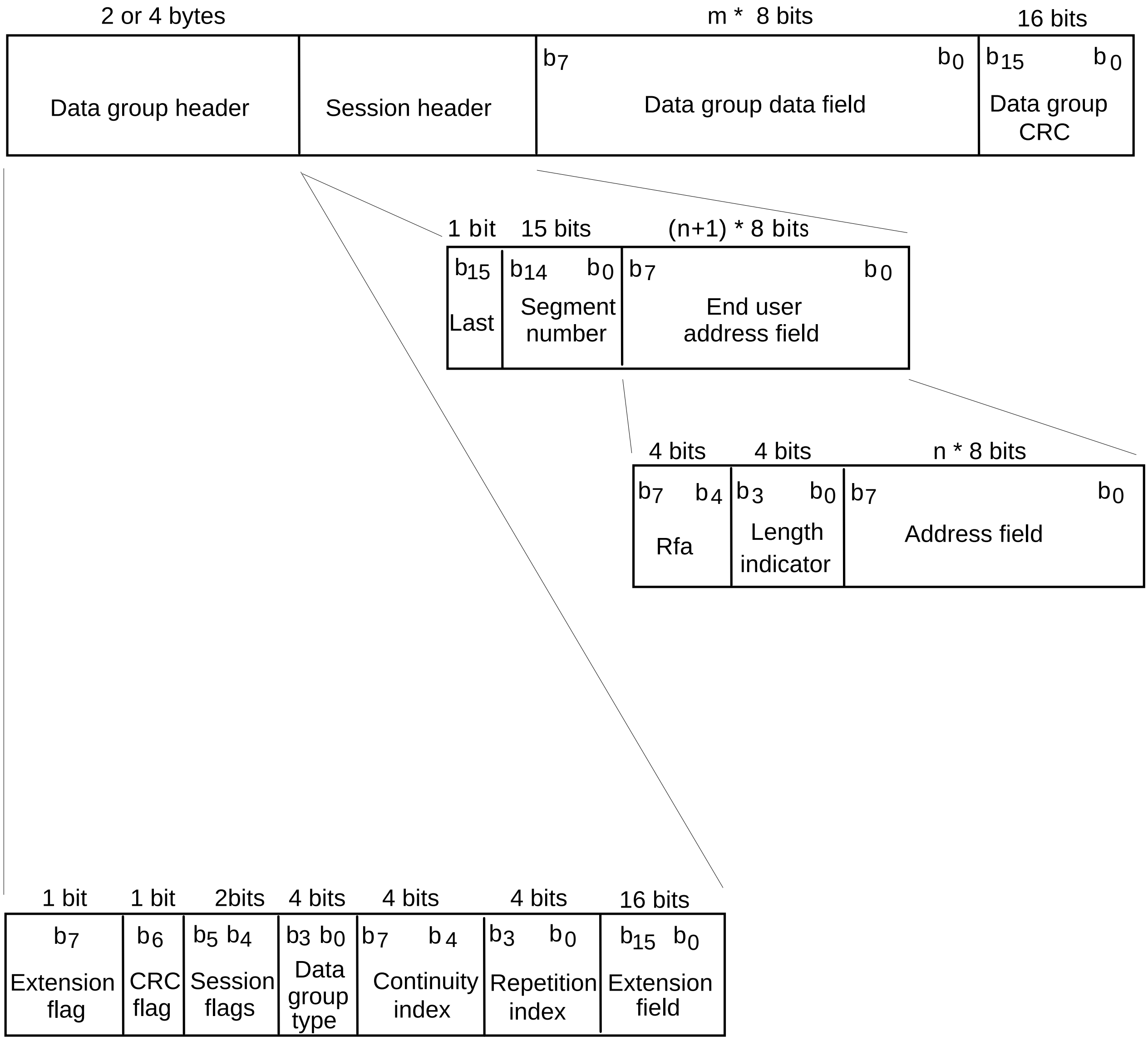


**Figure 9: Structure of the MSC data group**

#### 5.3.3.1 Data group header

The following definitions apply:

- **Extension flag**: this 1-bit flag shall indicate whether the extension field is present, or not, as follows:

  0 : no extension;
  1 : extension.

- **CRC flag**: this 1-bit flag shall indicate whether there is a CRC at the end of the data group as follows:

  0 : no data group CRC;
  1 : data group CRC present.

- **Session flags**: this 2 bits flag field shall indicate whether the segment number and/or the end user address field is present, or not, as follows:

  $b_5$ $b_4$
  0 0 : no session header;
  0 1 : no last flag, no segment number; end user address field present;
  1 0 : last flag and segment number present; no end user address field;
  1 1 : last flag, segment number and end user address field present.

- **Data group type**: this 4-bit field shall define the type of data carried in the data group data field. The following types are defined (the remaining types are reserved for future definition):

  $b_3$ $b_0$
  0 0 0 0 : General data;
  0 0 0 1 : CA;
  0 0 1 0 : General data and CA;
  0 0 1 1 : Reserved for file transfer description.

- **Continuity index**: the binary value of this 4-bit field shall be incremented each time a data group of a particular type, with a content different from that of the immediately preceding data group of the same type, is transmitted.

- **Repetition index**: the binary value of this 4-bit field shall signal the remaining number of repetitions of a data group with the same data content, occurring in successive data groups of the same type. Exceptionally, the code "1111" shall be used to signal that the repetition continues for an undefined period.

- **Extension field:** this 16-bit field shall be used to carry the Data Group Conditional Access (DGCA) when general data uses conditional access (Data group type 0010) (see subclause 9.2.3). The DGCA contains the Initialisation Modifier (IM) and additional conditional access (CA) information. For other Data group types, the Extension field is reserved for future additions to the Data group header.

#### 5.3.3.2 Session header

- **Last:** this 1-bit flag shall indicate whether the segment number field is the last or whether there are more to be transmitted, as follows:

  0 : more segments to follow;
  1 : last segment.

- **Segment number**: this field, coded as an unsigned binary number, shall indicate the segment number in the range 0 to 32 767.

  NOTE: The first segment is numbered 0 and the segment number is incremented by one at each new segment.

- **Rfa** (Reserved for future addition): this 4-bit field is reserved for future additions.

- **Length indicator**: this 4-bit field shall indicate the length in bytes of the address field. It is coded as an unsigned binary number in the range 0 to 15.

- **Address field**: this field shall indicate the address of the end user. It is coded as an unsigned binary number in the range 0 to 15.

#### 5.3.3.3 Data group data field

The data group data field shall contain an integral number of bytes, with a maximum of 8 191 bytes.

#### 5.3.3.4 Data group CRC

The data group CRC shall be a 16-bit CRC word calculated on the data group header, the session header and the data group data field and generated according to the procedure defined in annex E. The generation shall be based on the polynomial $G(x) = x^{16} + x^{12} + x^{5} + 1$ (ITU-T Recommendation X.25 [6]).

At the beginning of each CRC word calculation, all shift register stage contents shall be initialised to "1". The CRC word shall be complemented (1s complement) prior to transmission.

### 5.3.4 Interrelation of network and transport level in packet mode

The information associated with one MSC data group shall be transmitted in one or more packets, sharing the same address. All packets may contain padding bytes. Figure 10 shows the situation when a data group is spread across several packets, sharing the same address j. The settings of the First/Last flags are given.

The data field of the first packet shall begin with the data group header. The data field of the last packet shall end with the data group CRC, if any, and padding bytes, if necessary.

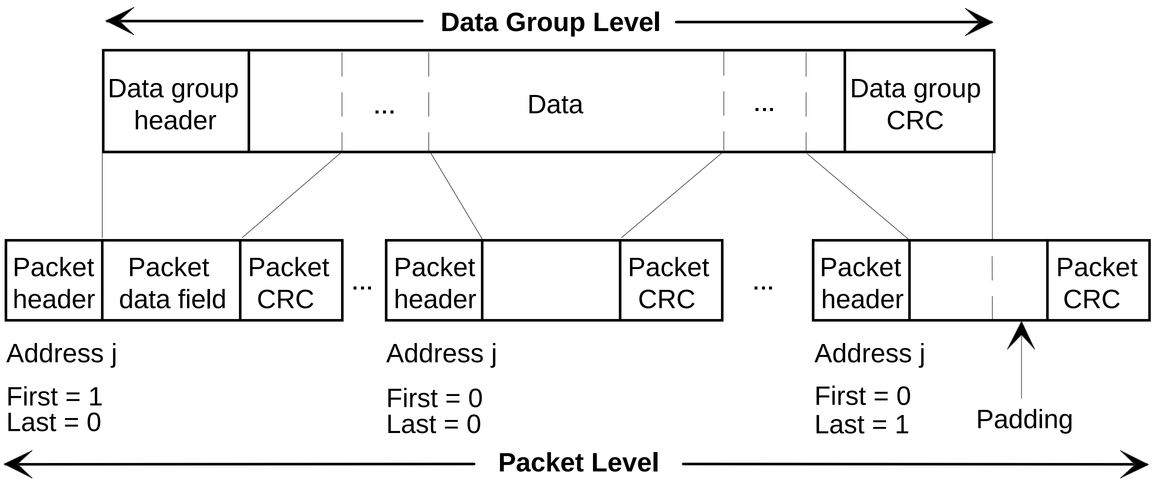


**Figure 10: Relationship between a MSC data group and a sequence of packets**

## 5.4 Transport of Service Information in the Auxiliary Information Channel

This subclause describes the mechanism for redirecting information from the FIC to the Auxiliary Information Channel (AIC). This mechanism may be used for certain extensions of FIC data group types 0 and 1 (see subclause 8.1.12). The AIC is formed using sub-channel 63 and packet address 1 023. The following conditions shall also apply when the AIC is used:

- the MSC data group (see subclause 5.3.3) shall be organised as shown in figure 11. It shall contain FIGs as defined in subclause 5.2.2;

- different FIG types may be carried in one MSC data group data field;

- the data group type list in the MSC data group shall be set to "General data";
- the maximum length of the MSC data group data field shall be 512 bytes;
- the MSC data group shall contain a data group CRC.

| | | | | |
|---|---|---|---|---|
| MSC Data group header | FIG i | ............. | FIG k | MSC Data group CRC |

**Figure 11: MSC data group structure for carrying FIGs**

# 6 Multiplex Configuration Information (MCI)

## 6.1 Introduction

The principal method of user access to the service components carried in the Multiplex is by selecting a service. Several services may be accessible within one ensemble, and each service contains one or more service components.

The essential service component of a service is called the Primary service component. Normally this would carry the audio, but data service components can be primary as well. All other service components are optional and are called Secondary service components.

An example of a service structure is shown in figure 12. In this example, the DAB ensemble is recognized by the associated ensemble label ("DAB ENSEMBLE ONE") and carries several services which can be accessed directly by the user. Three of these services are described.

The first service (identified by the service label "ALPHA 1 RADIO") comprises three service components: a primary audio component and two secondary components which are used for a Traffic Message Channel (ALPHA-TMC) and Service Information (ALPHA-SI). The audio component and SI are carried in separate sub-channels in the Main Service Channel, whereas the TMC is carried in the Fast Information Data Channel (FIDC) (see subclause 8.2) within the FIC. The SI is carried in Packet mode within the AIC (see also subclause 5.4).

The second service (identified by the service label "BETA RADIO") comprises two service components. In this case, there is a secondary audio component.

The linking arrangement allows service components to be shared by different services. It also allows the service structure to be changed so that a service may change its service component. The third service (identified by the service label "ALPHA 2 RADIO") shares the ALPHA-TMC service component with "ALPHA 1 RADIO". Also, at times, it shares the audio service component with "ALPHA 1 RADIO", as indicated by the switch.

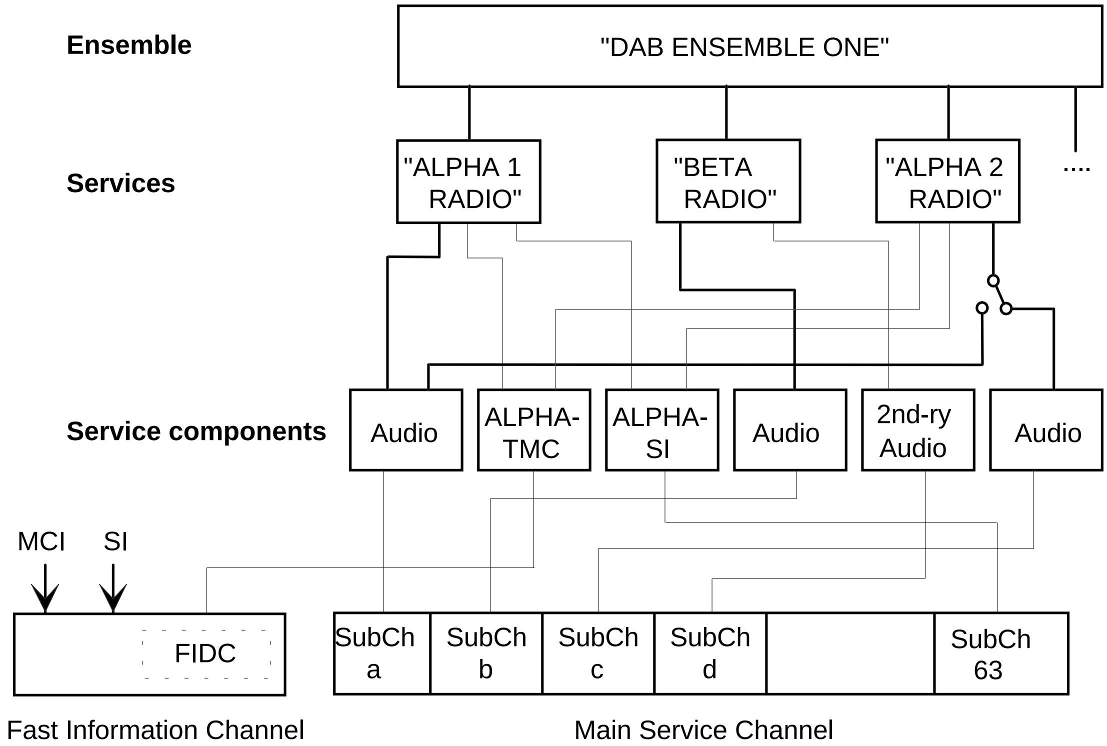


**Figure 12: An example of the DAB service structure**

The organisation of the sub-channels, services and service components in an ensemble is managed by the MCI. The MCI serves five principal functions:

a) to define the organisation of the sub-channels in terms of their position and size in the CIF and their error protection;

b) to list the services available in the ensemble;

c) to establish the links between service and service components;

d) to establish the links between service components and sub-channels;

e) to signal a multiplex re-configuration.

The details of coding the MCI in the FIC are described in subclauses 6.2 to 6.4. The dynamic re-configuration of the multiplex is described in subclause 6.5.

The MCI is coded in FIG type 0 (see subclause 5.2.2) using Extensions 0, 1, 2, 3 and 4. The C/N flag is used in Extensions 1, 2, 3 and 4 to distinguish between the current and a future configuration. The whole of the MCI shall be carried in the first FIB of successive 24 ms periods, corresponding to CIFs. Additionally, the first FIB of successive 24 ms periods may carry SI and FIDC information. MCI may also be repeated in other FIBs.

## 6.2 Sub-channel organisation

The sub-channel organisation defines the position and size of the sub-channels in the CIF and the error protection employed. It is coded in Extension 1 of FIG type 0 (FIG 0/1) as shown in figure 13. Each sub-channel is described explicitly by its start address (in the range 0 to 863 CUs) and (either explicitly or implicitly) by the size of the sub-channel and the error coding protection mechanism employed. Up to 64 sub-channels may be addressed in a multiplex using a sub-channel Identifier which takes values 0 to 63. The values are not related to the sub-channel position in the MSC.

Two forms of signalling the sub-channel size and error protection are used. The first is a shorter form which is used for audio service components employing the Unequal Error Protection (UEP) profiles given in subclause 11.3.1. In this case, the UEP index is signalled explicitly and the sub-channel size can be implicitly derived from it. The second form requires the sub-channel size and error protection to be signalled explicitly. In this case, eight options for defining these parameters are allowed; only the first is defined and it is used for equal error protection according to subclause 11.3.2.1.

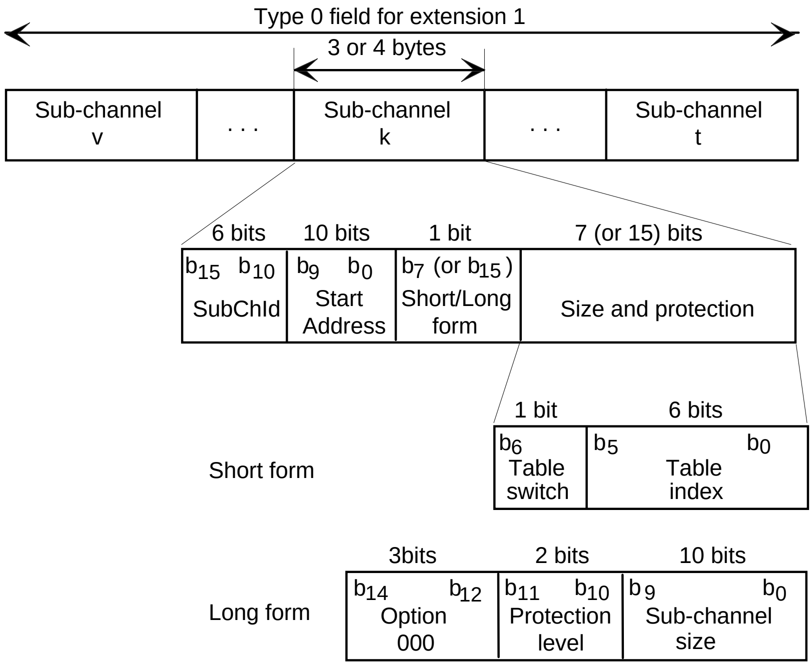


**Figure 13: Structure of the sub-channel organisation field**

The following definitions apply:

**SubChId** (Sub-channel Identifier): this 6-bit field, coded as an unsigned binary number, shall identify a sub-channel.

**Start Address**: this 10-bit field, coded as an unsigned binary number (in the range 0 to 863), shall address the first Capacity Unit (CU) of the sub-channel.

**Short/Long form**: this 1-bit flag shall indicate whether the short or the long form of the size and protection field is used, as follows:

0 : short form;
1 : long form.

- **Short form:**

  - **Table switch**: this field shall indicate a short-form definition of the sub-channel size, protection level and audio bit rate as follows:

    0 : table 7;
    1 : reserved for future use of the table index field.

**Table 7: Sub-channel size for audio service components as a function of the audio bit rate and the protection level (short-form application)**

| Index | Sub-Channel size (CU) | Protection level | Bit rate | Index | Sub-Channel size (CU) | Protection level | Bit rate |
|---|---|---|---|---|---|---|---|
| 0 | 16 | 5 | 32 | 33 | 64 | 5 | 128 |
| 1 | 21 | 4 | 32 | 34 | 84 | 4 | 128 |
| 2 | 24 | 3 | 32 | 35 | 96 | 3 | 128 |
| 3 | 29 | 2 | 32 | 36 | 116 | 2 | 128 |
| 4 | 35 | 1 | 32 | 37 | 140 | 1 | 128 |
| 5 | 24 | 5 | 48 | 38 | 80 | 5 | 160 |
| 6 | 29 | 4 | 48 | 39 | 104 | 4 | 160 |
| 7 | 35 | 3 | 48 | 40 | 116 | 3 | 160 |
| 8 | 42 | 2 | 48 | 41 | 140 | 2 | 160 |
| 9 | 52 | 1 | 48 | 42 | 168 | 1 | 160 |
| 10 | 29 | 5 | 56 | 43 | 96 | 5 | 192 |
| 11 | 35 | 4 | 56 | 44 | 116 | 4 | 192 |
| 12 | 42 | 3 | 56 | 45 | 140 | 3 | 192 |
| 13 | 52 | 2 | 56 | 46 | 168 | 2 | 192 |
| | x | | | 47 | 208 | 1 | 192 |
| 14 | 32 | 5 | 64 | 48 | 116 | 5 | 224 |
| 15 | 42 | 4 | 64 | 49 | 140 | 4 | 224 |
| 16 | 48 | 3 | 64 | 50 | 168 | 3 | 224 |
| 17 | 58 | 2 | 64 | 51 | 208 | 2 | 224 |
| 18 | 70 | 1 | 64 | 52 | 232 | 1 | 224 |
| 19 | 40 | 5 | 80 | 53 | 128 | 5 | 256 |
| 20 | 52 | 4 | 80 | 54 | 168 | 4 | 256 |
| 21 | 58 | 3 | 80 | 55 | 192 | 3 | 256 |
| 22 | 70 | 2 | 80 | 56 | 232 | 2 | 256 |
| 23 | 84 | 1 | 80 | 57 | 280 | 1 | 256 |
| 24 | 48 | 5 | 96 | 58 | 160 | 5 | 320 |
| 25 | 58 | 4 | 96 | 59 | 208 | 4 | 320 |
| 26 | 70 | 3 | 96 | | x | | |
| 27 | 84 | 2 | 96 | 60 | 280 | 2 | 320 |
| 28 | 104 | 1 | 96 | | x | | |
| 29 | 58 | 5 | 112 | 61 | 192 | 5 | 384 |
| 30 | 70 | 4 | 112 | | x | | |
| 31 | 84 | 3 | 112 | 62 | 280 | 3 | 384 |
| 32 | 104 | 2 | 112 | | x | | |
| | x | | | 63 | 416 | 1 | 384 |

- **Table index**: this field contains an index which shall identify one of the 64 options available for the sub-channel size and protection level. The index shall be coded as an unsigned binary number. The net data rate associated with each index is also given. Six of the possible combinations of protection level and rate are not provided and are indicated by an X.

- **Long form:**

  - **Option**: this 3-bit field shall indicate the option. One option (000) is defined for data with equal error protection as defined in subclause 11.3.2.1. The remaining options are reserved for future use.

    In the case of option 000, the following parameters are defined (see subclause 11.3.2.1):

- **Protection level**: this 2-bit field shall indicate the protection level as follows:

  00 : protection level 1;
  01 : protection level 2;
  10 : protection level 3;
  11 : protection level 4.

  The associated convolutional coding rate is given in table 8.

- **Sub-channel size**: this 10-bit field, coded as an unsigned binary number, shall define the number of Capacity Units occupied by the sub-channel. The number of capacity units shall be coded as an unsigned binary number (in the range 1 to 864). Table 8 shows the number of CUs required for all permissible data rates, in multiples of 8 kbit/s, for the four protection levels defined.

**Table 8: Sub-channel size for data at different coding rates, as a function of the data rate 8 n kbit/s (where n is an integer ≥ 1)**

| **Protection level** | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Convolutional coding rate** | 1/4 | 3/8 | 1/2 | 3/4 |
| **Sub-channel size (CUs)** | 12 n | 8 n | 6 n | 4 n |

### 6.3 Service organisation

The service organisation defines the services and service components carried in the ensemble. It is coded in the Extensions 2, 3 and 4 of FIG type 0. Each service shall be identified by a Service Identifier which, when used in conjunction with an Extended Country Code, is unique world-wide. Each service component shall be uniquely identified within the ensemble. When a service component is transported in the MSC in Stream mode, the basic service organisation information is coded in FIG 0/2 (see subclause 6.3.1). Service components, carried in the Packet mode, require additional signalling of the sub-channel and packet address. Extension 3 is used for this purpose (see subclause 6.3.2). Also, when service components are scrambled (see clause 9), the Service Component Conditional Access (SCCA) field (see subclause 9.2.2) is signalled in Extension 3, for data in Packet mode, and in Extension 4 (see subclause 6.3.3) for data carried in the Stream mode or in the FIC.

### 6.3.1 Basic service and service component definition

The Extension 2 of FIG type 0 (i.e. FIG 0/2) defines the basic service organisation. All the service descriptions applying to a service shall be contained within one field (service k) carried in a single FIG. Figure 14 shows the structure.

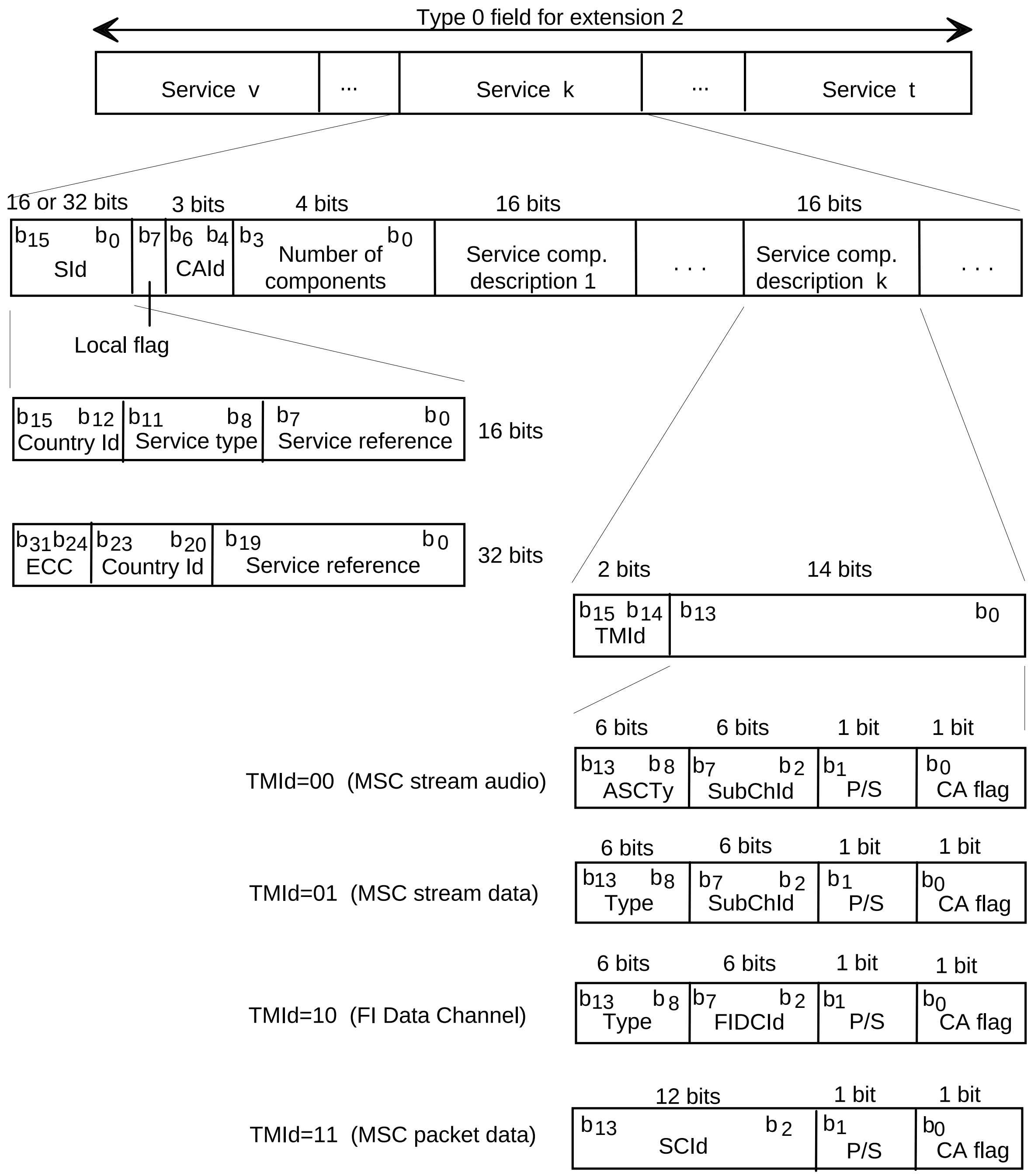


**Figure 14: Structure of the service organisation field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit or 32-bit field shall identify the service. The 16-bit Service identifier shall be structured in the same way as the Programme Identification (PI) code as defined in annex D of the RDS specification EN 50067 [5]. The 32-bit Service identifier shall be structured as shown in figure 14.

**Service Identifier description**:

- **Country Id** (Identification): this 4-bit field shall be defined as for the Programme Identification (PI) code - see annex D of the RDS specification EN 50067 [5].

- **Service type**: this 4-bit field indicates the service area coverage and shall be defined as for the Programme Identification (PI) code - see annex D of the RDS specification EN 50067 [5].

- **Service reference**: this field shall indicate the number of the service. For the 16-bit description this is an 8-bit field which shall be defined as for the Programme Identification (PI) code - see annex D of the RDS specification EN 50067 [5]. For the 32-bit description this is a 20-bit field which shall identify the specific data service.

- **ECC** (Extended Country Code): this 8-bit field shall be defined as in annex D of the RDS specification EN 50067 [5].

**Local flag**: this 1-bit flag shall indicate whether the service is available over the whole, or only a partial area served by the ensemble, as follows:

0 : whole ensemble service area;
1 : partial ensemble service area.

**CAId** (Conditional Access Identifier): this 3-bit field shall identify the Access Control System (ACS) used for the service. If no ACS is used for the service, CAId is set to zero. The following access control systems are defined. The remaining types are reserved for future use:

$b_6$ $b_4$
0 0 0 : No access control for all the components of the service;
0 0 1 : NR-MSK [10];
0 1 0 : Eurocrypt EN 50094 [11].

**Number of components**: this 4-bit field, coded as an unsigned binary number, shall indicate the number of service components (maximum 12), associated with the service. Each component shall be coded, according to the transport mechanism used.

**Service component description:**

- **TMId** (Transport Mechanism Identifier): this 2-bit field shall indicate the transport mechanism used:

  $b_{15}$ $b_{14}$
  0 0 : MSC - Stream mode - audio;
  0 1 : MSC - Stream mode - data;
  1 0 : FIDC;
  1 1 : MSC - Packet mode - data.

- **ASCTy** (Audio Service Component Type): this 6-bit field shall indicate the type of the audio service component. The following types are defined (the remaining types are reserved for future use):

  $b_{13}$ $b_8$
  0 0 0 0 0 0 : foreground sound;
  0 0 0 0 0 1 : background sound.

- **SubChId** (Sub-channel Identifier): this 6-bit field shall identify the sub-channel in which the service component is carried.

- **P/S** (Primary/Secondary): this 1-bit flag shall indicate whether the service component is the primary one, as follows:

  0 : not primary (secondary);
  1 : primary.

- **CA flag**: this 1-bit field flag shall indicate whether the service component is scrambled, as follows:

  0 : no scrambling;
  1 : scrambling.

- **Type**: this 6-bit field shall identify the type of the service component as follows:

  16 bit SId : Data Service Component Type (DSCTy);
  32 bit SId : Specific Service Component Type (SSCTy).

  - **DSCTy** (Data Service Component Type): this 6-bit field shall indicate the type of the data service component. The following types are defined (the remaining types are reserved for future use):

    $b_{13}$ $b_8$
    0 0 0 0 0 0 : Paging;
    0 0 0 0 0 1 : Traffic Message Channel (TMC), see bibliography;
    0 0 0 0 1 0 : Emergency Warning Systems (EWS,) EN 50067 [5];
    0 0 0 0 1 1 : Interactive Text Transmission System (ITTS), see annex F, bibliography.

- **SSCTy** (Specific Service Component Type): this 6-bit field shall indicate the type of specific service component. These types are defined by the specific service.

- **FIDCId** (Fast Information Data Channel Identifier): this 6-bit field shall identify the service component carried in the FIDC. It is organised as shown in figure 15 and the TCId and extension fields are described in subclause 5.2.2.3.

| | 3bits | 3bits |
|---|---|---|
| FIDCId | $b_7$ $b_5$<br>TCId | $b_4$ $b_2$<br>Ext |

**Figure 15: Structure of the FIDCId**

- **SCId** (Service Component Identifier): this 12-bit field shall uniquely identify the service component within the ensemble.

### 6.3.2 Service component in Packet mode with or without Conditional Access

The Extension 3 of FIG type 0 (FIG 0/3) gives additional information about the service component description in Packet mode. Figure 16 shows the structure.

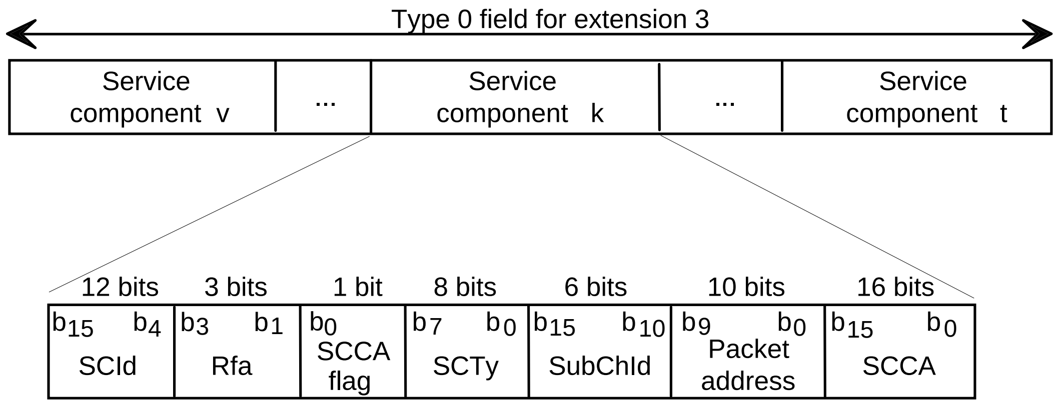


**Figure 16: Structure of the service component field in Packet mode**

The following definitions apply:

**SCId** (Service Component Identifier): see subclause 6.3.1.

**Rfa**: this 3-bit field shall be reserved for future additions. All bits shall be set to zero until they are defined.

**SCCA flag**: this 1-bit flag shall indicate whether the Service Component Conditional Access (SCCA) field is present, or not, as follows:

0 : no SCCA field;
1 : SCCA field is present.

**SCTy (Service Component Type)**: this 8-bit field shall indicate the type of the service component. If the 16-bit SId is used (Id = 0), the first four types are defined in the description of the DSCTy (see subclause 6.3.1) with the additional requirement that the first two bits of the SCTy field ($b_7$, $b_6$) shall be set to "00". If the 32-bit SId is used (Id = 1), these types are defined by the specific service.

**SubChId** (Sub-channel Identifier): see subclause 6.3.1.

**Packet address**: this 10-bit field shall define the address of the packet in which the service component is carried.

**SCCA (Service Component Conditional Access)**: this 16-bit field shall contain the descrambling parameters for accessing scrambled service components (see subclause 9.2.2).

### 6.3.3 Service component with conditional access in stream mode or FIC

The Extension 4 of FIG type 0 (FIG 0/4) gives additional information about the service component description for components with CA and carried in Stream mode or in the FIC. Figure 17 shows the structure.

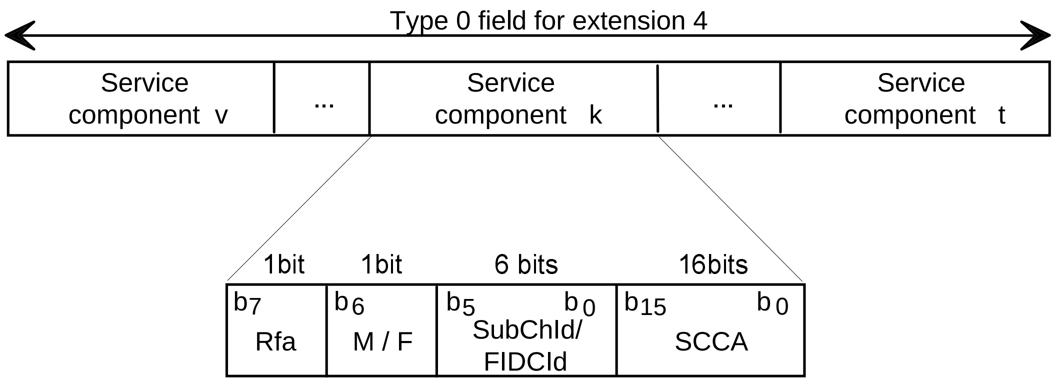


**Figure 17: Structure of the service component field in Stream mode or FIC**

The following definitions apply:

**Rfa**: this 1-bit field is reserved for future additions. The bit shall be set to zero until it is defined.

**M/F**: this 1-bit flag shall indicate whether the service component is carried in the MSC or in the FIC and whether the component is identified using the SubChId or the FIDCId, as follows:

0 : MSC and SubChId;
1 : FIC and FIDCId.

**SubChId** (Sub-channel Identifier): see subclause 6.3.1.

**FIDCId** (Fast Information Data Channel Identifier): see subclause 6.3.1.

**SCCA** (Service Component Conditional Access): see subclause 6.3.2.

### 6.3.4 Service component "hooks"

For any future service component related SI, table 9 shows how service components can be identified for a particular transport mechanism. The description of these identifiers is given in subclause 6.3.1.

**Table 9: Service component identifiers**

| Transport mechanism | Service component identifier |
|---|---|
| Audio or data in Stream mode | SubChId |
| FIDC | FIDCId |
| Data in packets | SCId |

## 6.4 Ensemble information

The ensemble information contains SI and control mechanisms which are common to all services contained in the ensemble. It is specifically used to provide an alarm flag and CIF counter (24 ms increments) for use with the management of a multiplex re-configuration. Three different levels of multiplex re-configuration are distinguished:

a) change of sub–channel organisation only (see subclause 6.2);

b) change of service organisation only (see subclause 6.3);

c) change of sub–channel and service organisation.

The ensemble information is coded in Extension 0 of FIG type 0 (FIG 0/0) as shown in figure 18.

Type 0 field for extension 0

| 16 bits | 2bits | 1bit | 5 + 8 bits | | 8 bits |
|---|---|---|---|---|---|
| $b_{15}$ $b_0$ | $b_{15}$ $b_{14}$ | $b_{13}$ | $b_{12}$ $b_8$ | $b_7$ $b_0$ | $b_7$ $b_0$ |
| EId | Change flag | Al flag | 0-19 | 0-249 | Occurrence change 0-249 |
| | | | CIF Count | | |

**Figure 18: Structure of the ensemble information field**

The following definitions apply:

**EId** (Ensemble Identifier): a unique 16-bit code, shall be allocated to the ensemble and allows unambiguous identification of the ensemble.

**Change flag**: this 2-bit field shall be used to indicate whether there is to be a change in the sub-channel or service organisation, or both, as follows:

$b_{15}$ $b_{14}$
0 0 : no change, no occurrence change field present;
0 1 : sub-channel organisation only;
1 0 : service organisation only;
1 1 : sub-channel organisation and service organisation.

**Al flag** (Alarm flag): this 1-bit flag shall be used to signal that alarm messages are accessible within the ensemble, as follows:

0 : alarm messages not accessible;
1 : alarm messages accessible.

The alarm messages are presented as an alarm announcement which interrupts the reception of programme services carried in the ensemble (see subclause 8.1.6.1).

**CIF count**: this modulo-5 000 binary counter shall be arranged in two parts and is incremented by one at each successive CIF. The higher part is a modulo-20 counter (0 to 19) and the lower part is a modulo-250 counter (0 to 249).

**Occurrence change**: this 8-bit field shall indicate the value of the lower part of the CIF counter from which the new configuration applies.

NOTE: In any 96 ms period, the FIG 0/0 should be transmitted in a fixed time position. In transmission mode I, this should be the first FIB (of the three) associated with the first CIF (of the four) in the transmission frame (see subclause 5.1). In transmission modes II and III, this should be the first FIB of every fourth transmission frame.

### 6.5 Multiplex re-configuration

The ensemble information (see subclause 6.4) provides the required mechanisms for changing the multiplex configuration whilst maintaining continuity of services. Such a multiplex re-configuration is achieved by sending the MCI of the future multiplex configuration in advance as well as the MCI for the current configuration. Accordingly, every MCI message includes a C/N flag signalling whether its information applies to the current or to the next multiplex configuration (see subclause 5.2.2).

Service continuity requires the signalling of the exact instant of time, from which a multiplex re-configuration is to be effective. The time boundary between two CIFs is used for this purpose. Every CIF is addressable by the value of the CIF counter. The occurrence change field, which comprises the lower part of the CIF count, is used to signal the instant of the multiplex re-configuration. It permits a multiplex re-configuration to be signalled within an interval of up to six seconds in advance. A multiplex configuration shall remain stable for at least six seconds (250 CIFs).

NOTE: It is expected that the MCI for a new configuration will be signalled at least three times in the six-second period immediately before the change occurs.

A multiplex re-configuration requires a careful co-ordination of the factors which affect the definition of the sub-channels. These factors include the source audio/data bit rate and convolutional encoding/decoding. The timing of changes made to any of these factors can only be made in terms of logical frames. However the logical frame count is related to the CIF count (see subclause 5.3) and this provides the link for co-ordinating these activities.

In general, whenever a multiplex re-configuration occurs at a given CIF count *n* (i.e. the new configuration is valid from this time), then each of the actions related to the sub-channels, affected by this re-configuration, shall be changed at the Logical frame with the corresponding Logical frame count. There is only one exception to this rule: if the number of CUs allocated to a sub-channel decreases at the CIF count *n*, then all the corresponding changes made in that sub-channel, at the Logical frame level, shall occur at CIF count (*n - 15*) which is fifteen 24 ms bursts in advance. This is a consequence of the time interleaving process, described in clause 12.

Additional information related to multiplex re-configuration is given in annex D.

# 7 Audio coding

The coding technique for high quality audio signals uses the properties of human sound perception by exploiting the spectral and temporal masking effects of the ear. This technique allows a bit rate reduction from 768 kbit/s down to about 100 kbit/s per mono channel, while preserving the subjective quality of the digital studio signal for any critical source material (see reference CCIR Recommendation 562-3 [13]).

An overview of the principal functions of the audio coding scheme is shown in the simplified block diagram of the DAB audio encoder (see figure 19). The main characteristics of the audio coding system, like audio modes, bit rates and audio frame length are given in annex A, clause A.2, whereas the characteristics of the input audio signal are given in annex A, clause A.1.

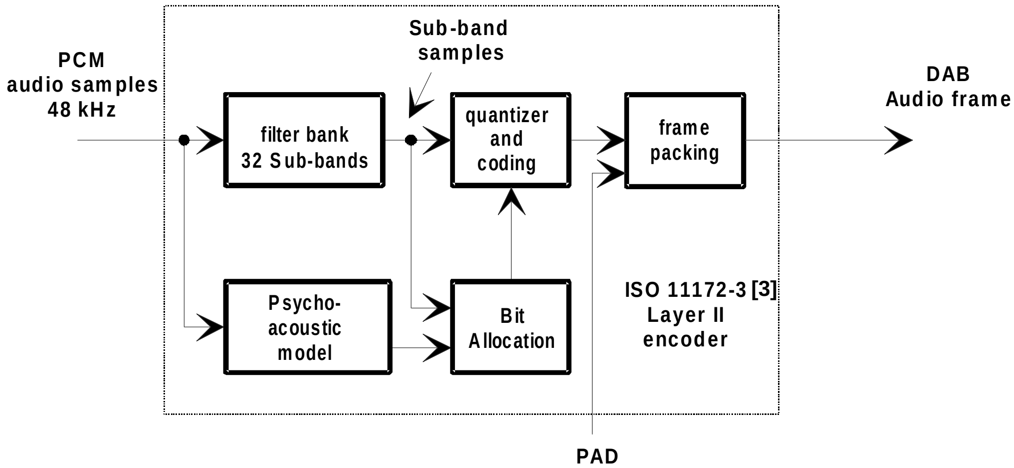


**Figure 19: Simplified block diagram of the DAB audio encoder**

The input PCM audio samples are fed into the audio encoder. A filter bank creates a filtered and sub-sampled representation of the input audio signal. The filtered samples are called sub-band samples. A psychoacoustic model of the human ear should create a set of data to control the quantizer and coding. These data can be different depending on the actual implementation of the encoder. An estimation of the masking threshold can be used to obtain these quantizer control data. The quantizer and coding block shall create a set of coding symbols from the sub-band samples. The frame packing block shall assemble the actual audio bit stream from the output data of the previous block, and shall add other information, such as header information, CRC words for error detection and Programme Associated Data (PAD), which are intimately related with the coded audio signal. The resulting audio frame corresponds to 24 ms duration of audio and shall comply with the ISO/IEC 11172-3 Layer II format [3]. The formatting of the DAB audio frame shall be done in such a way that the structure of the DAB audio frame conforms to the audio bit stream syntax described in subclause 7.3.

The simplified block diagram of the audio decoder in the receiver, shown in figure 20, accepts the DAB audio frame in the syntax defined in subclause 7.3.2 which is a conformant subset of the ISO/IEC 11172-3 [3] Layer II Bit stream syntax defined in subclause 7.3.1. This allows the use of an ISO/IEC 11172-3 [3] Layer II decoder. The ISO audio frame shall be fed into the audio decoder, which unpacks the data of the frame to recover the various elements of information. The reconstruction block shall reconstruct the quantized sub-band samples. An inverse filter bank shall transform the sub-band samples back to produce digital PCM audio signals at 48 kHz sampling frequency according to annex B.

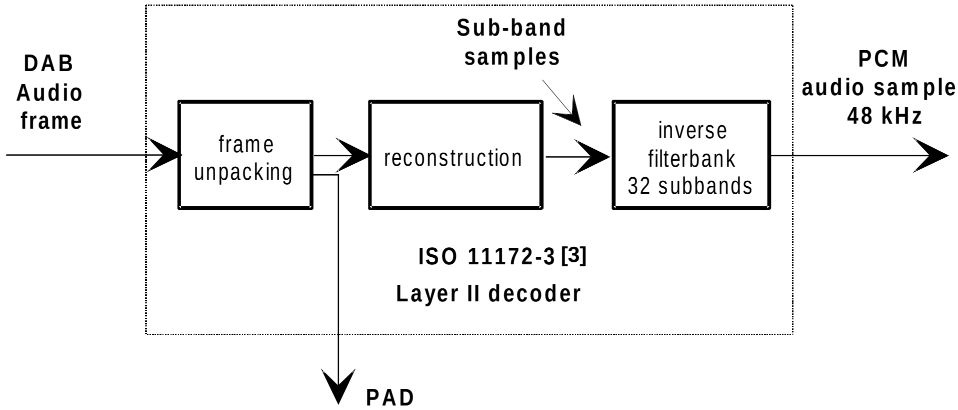


**Figure 20: Simplified block diagram of the DAB audio decoder**

## 7.1 Audio encoding

The source encoder for the DAB system is the ISO/IEC 11172-3 [3] Layer II encoder with restrictions on some parameters and some additional protection against transmission errors. In the ISO/IEC 11172-3 [3] only the encoded audio bit stream, rather than the encoder, and the decoder are specified. In subsequent clauses, both normative and informative parts of the encoding technique are described. An example of one complete suitable encoder with the corresponding flow diagram (figure 21) is given in the following subclauses.

The DAB source coding algorithm is based on a perceptual coding technique. The six primary parts of such an audio encoding technique are:

1) analysis sub-band filter (subclause 7.1.1 and annex C, clause C.1);

2) Scale Factor calculation (subclauses 7.1.2 to 7.1.4);

3) psychoacoustic model (subclause 7.1.5 and annex C, clause C.2);

4) Bit Allocation (BAl) procedure (subclause 7.1.6 and annex C, clause C.3);

5) quantizing and coding (subclauses 7.1.7 and 7.1.8);

6) bit stream formatter (subclause 7.1.9).

### 7.1.1 Analysis sub-band filter

An analysis filter should be used to split the broadband audio signal with sampling frequency $f_S$ into 32 equally spaced sub-bands, each with a sampling frequency of $f_S/32$. This filter, called a poly-phase analysis filter bank, is critically sampled (i.e. there are as many samples in the sub-band domain as there are in the time domain). A detailed description of a suitable analysis sub-band filter bank with the appropriate formulae, coefficients and flow charts is described in annex C, clause C.1.

The encoding algorithm provides a frequency response down to 0 Hz. However, in applications where this is not desirable, a high-pass filter should be included at the audio input of the encoder. The application of such a high-pass filter avoids an unnecessarily high bit rate requirement for the lowest sub-band and may increase the overall audio quality. The cut-off frequency should be in the range of 2 to 10 Hz.

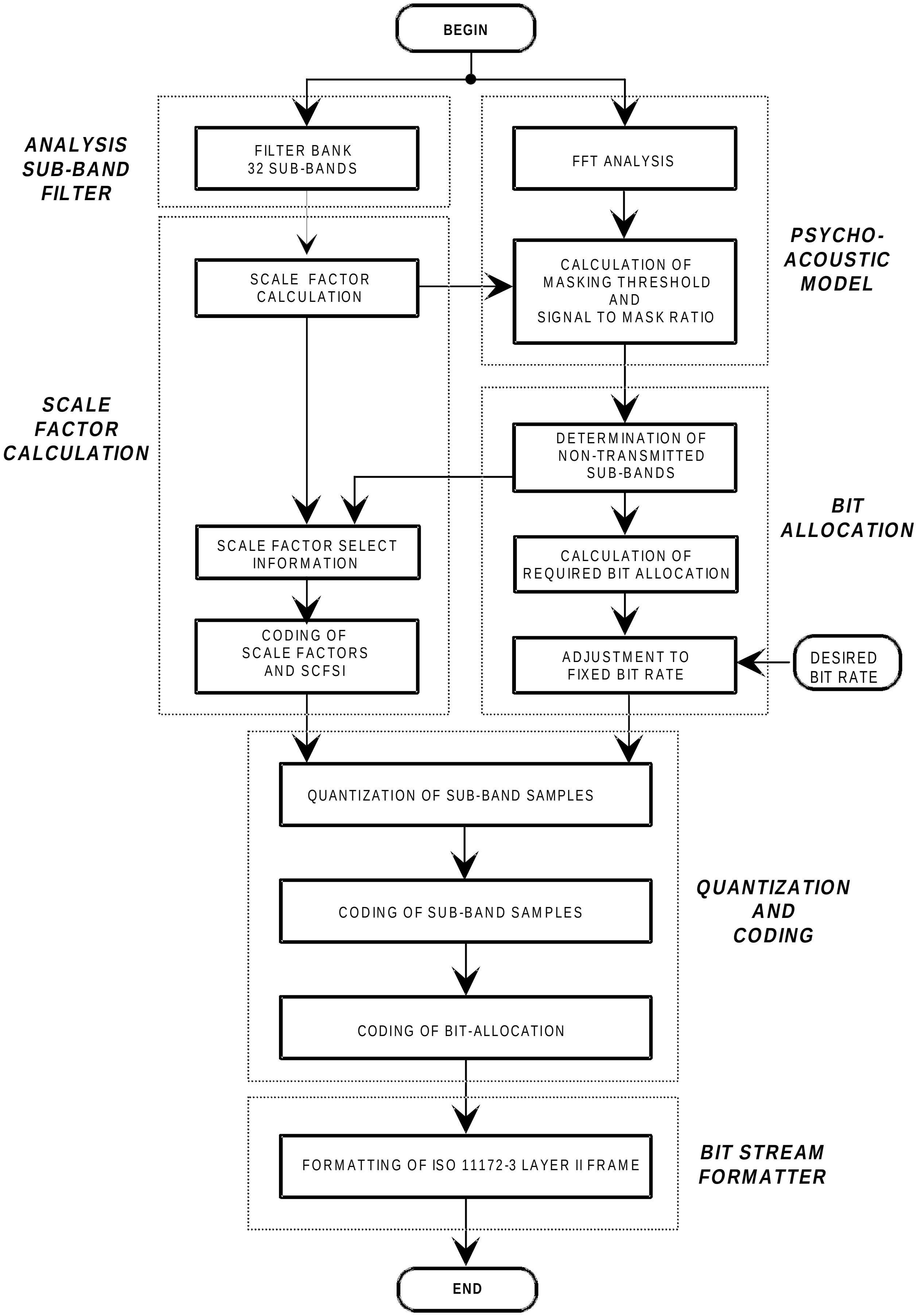


**Figure 21: Flow diagram of the ISO/IEC 11172-3 [3] Layer II audio encoder**

### 7.1.2 Scale Factor calculation

In each sub-band, 36 samples shall be grouped for processing. Before quantization, the output samples of the filter bank should be normalized. The calculation of the Scale Factor (ScF) for each sub-band shall be performed every 12 sub-band samples. The maximum of the absolute value of these 12 samples shall be determined. The lowest value, given by the column "Scale Factor" in table 10, which is larger than this maximum shall be used as the ScF of the 12 sub-band samples.

**Table 10: Scale Factors**

| Index iscf | Scale Factor ScF | Index iscf | Scale Factor Scf |
|---|---|---|---|
| 0 | 2,00000000000000 | 32 | 0,00123039165029 |
| 1 | 1,58740105196820 | 33 | 0,00097656250000 |
| 2 | 1,25992104989487 | 34 | 0,00077509816991 |
| 3 | 1,00000000000000 | 35 | 0,00061519582514 |
| 4 | 0,79370052598410 | 36 | 0,00048828125000 |
| 5 | 0,62996052494744 | 37 | 0,00038754908495 |
| 6 | 0,50000000000000 | 38 | 0,00030759791257 |
| 7 | 0,39685026299205 | 39 | 0,00024414062500 |
| 8 | 0,31498026247372 | 40 | 0,00019377454248 |
| 9 | 0,25000000000000 | 41 | 0,00015379895629 |
| 10 | 0,19842513149602 | 42 | 0,00012207031250 |
| 11 | 0,15749013123686 | 43 | 0,00009688727124 |
| 12 | 0,12500000000000 | 44 | 0,00007689947814 |
| 13 | 0,09921256574801 | 45 | 0,00006103515625 |
| 14 | 0,07874506561843 | 46 | 0,00004844363562 |
| 15 | 0,06250000000000 | 47 | 0,00003844973907 |
| 16 | 0,04960628287401 | 48 | 0,00003051757813 |
| 17 | 0,03937253280921 | 49 | 0,00002422181781 |
| 18 | 0,03125000000000 | 50 | 0,00001922486954 |
| 19 | 0,02480314143700 | 51 | 0,00001525878906 |
| 20 | 0,01968626640461 | 52 | 0,00001211090890 |
| 21 | 0,01562500000000 | 53 | 0,00000961243477 |
| 22 | 0,01240157071850 | 54 | 0,00000762939453 |
| 23 | 0,00984313320230 | 55 | 0,00000605545445 |
| 24 | 0,00781250000000 | 56 | 0,00000480621738 |
| 25 | 0,00620078535925 | 57 | 0,00000381469727 |
| 26 | 0,00492156660115 | 58 | 0,00000302772723 |
| 27 | 0,00390625000000 | 59 | 0,00000240310869 |
| 28 | 0,00310039267963 | 60 | 0,00000190734863 |
| 29 | 0,00246078330058 | 61 | 0,00000151386361 |
| 30 | 0,00195312500000 | 62 | 0,00000120155435 |
| 31 | 0,00155019633981 | | |

### 7.1.3 Coding of Scale Factors

This subclause is partly of informative, and partly of normative nature. The index 'iscf' in table 10 is represented by 6 bits, MSb first. The ScF of a certain sub-band shall be transmitted only if a non-zero number of bits has been allocated to this sub-band.

A DAB audio frame corresponds to 36 sub-band samples and therefore contains three ScFs per sub-band. Some may not be transmitted. This subclause gives information about which ScFs should be transmitted, and how they shall be encoded.

The two differences $dscf_1$ and $dscf_2$ of the successive ScF indices $iscf_1$, $iscf_2$ and $iscf_3$ shall be calculated as follows:

$dscf_1 = iscf_1 - iscf_2$;
$dscf_2 = iscf_2 - iscf_3$.

Five classes of ScF difference shall be defined. The class of each of the differences should be determined by the following table 11.

**Table 11: ScF difference classes**

| class | dscf |
|---|---|
| 1 | dscf ≤ -3 |
| 2 | -3 < dscf < 0 |
| 3 | dscf = 0 |
| 4 | 0 < dscf < 3 |
| 5 | dscf ≥ 3 |

**Table 12: ScF transmission patterns**

| $Class_1$ | $Class_2$ | Transmission Pattern | Scale Factor Select. Information (scfsi) | Code |
|---|---|---|---|---|
| 1 | 1 | 1 2 3 | 0 | 00 |
| 1 | 2 | 1 2 2 | 3 | 11 |
| 1 | 3 | 1 2 2 | 3 | 11 |
| 1 | 4 | 1 3 3 | 3 | 11 |
| 1 | 5 | 1 2 3 | 0 | 00 |
| 2 | 1 | 1 1 3 | 1 | 01 |
| 2 | 2 | 1 1 1 | 2 | 10 |
| 2 | 3 | 1 1 1 | 2 | 10 |
| 2 | 4 | 4 4 4 | 2 | 10 |
| 2 | 5 | 1 1 3 | 1 | 01 |
| 3 | 1 | 1 1 1 | 2 | 10 |
| 3 | 2 | 1 1 1 | 2 | 10 |
| 3 | 3 | 1 1 1 | 2 | 10 |
| 3 | 4 | 3 3 3 | 2 | 10 |
| 3 | 5 | 1 1 3 | 1 | 01 |
| 4 | 1 | 2 2 2 | 2 | 10 |
| 4 | 2 | 2 2 2 | 2 | 10 |
| 4 | 3 | 2 2 2 | 2 | 10 |
| 4 | 4 | 3 3 3 | 2 | 10 |
| 4 | 5 | 1 2 3 | 0 | 00 |
| 5 | 1 | 1 2 3 | 0 | 00 |
| 5 | 2 | 1 2 2 | 3 | 11 |
| 5 | 3 | 1 2 2 | 3 | 11 |
| 5 | 4 | 1 3 3 | 3 | 11 |
| 5 | 5 | 1 2 3 | 0 | 00 |

The pair of difference classes shall indicate the entry point in the table 12. For each pair of difference classes the actual transmission pattern of Scale Factors and the actual Scale Factor Selection Information (ScFSI) shall be determined from table 12.

Only the Scale Factors indicated in the "transmission pattern" shall be transmitted. A "1", "2" or "3" means that the first, second or third Scale Factor, respectively, is transmitted within an audio frame. A "4" means that the maximum of the three Scale Factors is transmitted. If two or three of the Scale Factors are the same, not all Scale Factors should be transmitted for a certain sub-band within one audio frame. The information describing the number and the position of the Scale Factors in each sub-band is called "Scale Factor Select Information" (ScFSI).

### 7.1.4 Coding of Scale Factor Selection Information

The ScFSI shall be coded by an unsigned two bit binary word, MSb first, which is also to be found in the table 12, showing the Scale Factor transmission patterns. Only the ScFSI for the sub-bands which will have a non-zero Bit Allocation shall be transmitted.

### 7.1.5 Psychoacoustic model

A psychoacoustic model should calculate a just-noticeable noise-level for each sub-band in the filter bank. This noise level should be used in the Bit Allocation procedure to determine the actual quantizer for each sub-band. The final output of the model is a Signal-to-Mask Ratio (SMR) for each sub-band. For a high coding efficiency, it is recommended to use a psychoacoustic model with an appropriate frequency analysis. An example of a reference psychoacoustic model is presented in annex C, clause C.2.

### 7.1.6 Bit Allocation

A Bit Allocation procedure shall be applied. Different strategies for allocating the bits to the sub-band samples of the individual sub-bands are possible. A reference model of the Bit Allocation procedure is described in annex C, clause C.3. The principle used in this allocation procedure is minimisation of the total noise-to-mask ratio over the audio frame with the constraint that the number of bits used does not exceed the number of bits available for that DAB audio frame. The bit allocater should consider both the output samples from the filter bank and the Signal-to-Mask-Ratios (SMRs) from the psychoacoustic model, and assigns a number of bits to each sample (or group of samples) in each sub-band, in order to simultaneously meet both the bit rate and masking requirements. At low bit rates, when the demand derived from the masking threshold cannot be met, the bit allocater attempts to spread bits in a psychoacoustically inoffensive manner among the sub-bands.

After determining, how many bits should be distributed to each sub-band signal, the resulting number shall be used to code the sub-band samples, the ScFSI and the ScFs. Only a limited number of quantizations is allowed for each sub-band. Tables 13 and 14 indicate for every sub-band the number of quantization steps which shall be used to quantize the sub-band samples. Table 13 shall be used for bit rates of 56 to 192 kbit/s in Single channel mode as well as for 64 kbit/s and 96 to 384 kbit/s in all other audio modes. The number of the lowest sub-band for which no bits are allocated, called "sblimit", equals 27, and the total number of bits used for the Bit Allocation per audio frame is defined by the sum of "nbal". If "sblimit" is equal to 27, the sum of 'nbal' is equal to 88 for Single channel mode, whereas the sum of 'nbal' is equal to 176 for Dual channel or Stereo mode. This number is smaller, if the Joint stereo mode is used. Table 14 shall be used for bit rates of 32 and 48 kbit/s in Single channel mode. The number of bits required to represent these quantized sub-band samples shall be derived from the two last columns of table 16. In this case "sblimit" is equal to 8, and the total number of bits used for the Bit Allocation per audio frame, i.e. sum of "nbal", is 26.

### 7.1.7 Bit Allocation coding

In order to increase the coding efficiency, only a limited number of possible quantizations are permitted. Both the number and the quantizations may be different from one sub-band (denoted as "sb" in tables 13 and 14) to another. Only the index with word length "nbal" given in tables 13 and 14, which depends on the bit rate and audio mode, shall be transmitted, MSb first.

**Table 13: Bit Allocation and possible quantization per sub-band**

Bit rates: 56, 64, 80, 96, 112, 128, 160, and 192 kbits/s (single channel mode)

Bit rates: 64, 96, 112, 128, 160, 192, 224, 256, 320 and 384 kbits/s (all other audio modes)

**index --->**

| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **sb** | **nbal** | **nlevels** | | | | | | | | | | | | | | | |
| 0 | 4 | - | 3 | 7 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 16383 | 32767 | 65535 |
| 1 | 4 | - | 3 | 7 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 16383 | 32767 | 65535 |
| 2 | 4 | - | 3 | 7 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 16383 | 32767 | 65535 |
| 3 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 4 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 5 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 6 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 7 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 8 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 9 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 10 | 4 | - | 3 | 5 | 7 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 65535 |
| 11 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 12 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 13 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 14 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 15 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 16 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 17 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 18 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 19 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 20 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 21 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 22 | 3 | - | 3 | 5 | 7 | 9 | 15 | 31 | 65535 | | | | | | | | |
| 23 | 2 | - | 3 | 5 | 65535 | | | | | | | | | | | | |
| 24 | 2 | - | 3 | 5 | 65535 | | | | | | | | | | | | |
| 25 | 2 | - | 3 | 5 | 65535 | | | | | | | | | | | | |
| 26 | 2 | - | 3 | 5 | 65535 | | | | | | | | | | | | |
| 27 | 0 | - | | | | | | | | | | | | | | | |
| 28 | 0 | - | | | | | | | | | | | | | | | |
| 29 | 0 | - | | | | | | | | | | | | | | | |
| 30 | 0 | - | | | | | | | | | | | | | | | |
| 31 | 0 | - | | | | | | | | | | | | | | | |

**Table 14: Bit Allocation and possible quantization per sub-band**

Bit rates: 32 and 48 kbits/s (Single channel mode)

**index --->**

| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **sb** | **nbal** | **nlevels** | | | | | | | | | | | | | | | |
| **0** | **4** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 16383 | 32767 |
| **1** | **4** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | 255 | 511 | 1023 | 2047 | 4095 | 8191 | 16383 | 32767 |
| **2** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **3** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **4** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **5** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **6** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **7** | **3** | - | 3 | 5 | 9 | 15 | 31 | 63 | 127 | | | | | | | | |
| **8** | **0** | - | | | | | | | | | | | | | | | |
| **9** | **0** | - | | | | | | | | | | | | | | | |
| **10** | **0** | - | | | | | | | | | | | | | | | |
| **11** | **0** | - | | | | | | | | | | | | | | | |
| **12** | **0** | - | | | | | | | | | | | | | | | |
| **13** | **0** | - | | | | | | | | | | | | | | | |
| **14** | **0** | - | | | | | | | | | | | | | | | |
| **15** | **0** | - | | | | | | | | | | | | | | | |
| **16** | **0** | - | | | | | | | | | | | | | | | |
| **17** | **0** | - | | | | | | | | | | | | | | | |
| **18** | **0** | - | | | | | | | | | | | | | | | |
| **19** | **0** | - | | | | | | | | | | | | | | | |
| **20** | **0** | - | | | | | | | | | | | | | | | |
| **21** | **0** | - | | | | | | | | | | | | | | | |
| **22** | **0** | - | | | | | | | | | | | | | | | |
| **23** | **0** | - | | | | | | | | | | | | | | | |
| **24** | **0** | - | | | | | | | | | | | | | | | |
| **25** | **0** | - | | | | | | | | | | | | | | | |
| **26** | **0** | - | | | | | | | | | | | | | | | |
| **27** | **0** | - | | | | | | | | | | | | | | | |
| **28** | **0** | - | | | | | | | | | | | | | | | |
| **29** | **0** | - | | | | | | | | | | | | | | | |
| **30** | **0** | - | | | | | | | | | | | | | | | |
| **31** | **0** | - | | | | | | | | | | | | | | | |

### 7.1.8 Quantization and coding of sub-band samples

A quantization process of the sub-band samples shall be applied. The following description of this process is informative, but the coding of the sub-band samples has to follow normative rules.

Each of the 12 consecutive sub-band samples, which are grouped together for the scaling process, should be normalized by dividing its value by the Scale Factor to obtain a value denoted $X$ and quantized using the following procedure:

- calculate $A * X + B$;
- take the N most significant bits;
- invert the MSb.

**Table 15: Quantization coefficients**

| No. of steps | *A* | *B* |
|---|---|---|
| 3 | 0,750000000 | -0,250000000 |
| 5 | 0,625000000 | -0,375000000 |
| 7 | 0,875000000 | -0,125000000 |
| 9 | 0,562500000 | -0,437500000 |
| 15 | 0,937500000 | -0,062500000 |
| 31 | 0,968750000 | -0,031250000 |
| 63 | 0,984375000 | -0,015625000 |
| 127 | 0,992187500 | -0,007812500 |
| 255 | 0,996093750 | -0,003906250 |
| 511 | 0,998046875 | -0,001953125 |
| 1 023 | 0,999023438 | -0,000976563 |
| 2 047 | 0,999511719 | -0,000488281 |
| 4 095 | 0,999755859 | -0,000244141 |
| 8 191 | 0,999877930 | -0,000122070 |
| 16 383 | 0,999938965 | -0,000061035 |
| 32 767 | 0,999969482 | -0,000030518 |
| 65 535 | 0,999984741 | -0,000015259 |

The quantization coefficients $A$ and $B$ can be found in table 15. The number N of bits per codeword, given in table 16, represents the number of bits necessary to encode the number of quantization steps. The inversion of the MSb shall be done in order to avoid the all "1" code that is used for the synchronization word in the ISO/IEC 11172-3 [3] header.

Three consecutive sub-band samples, called a granule, shall be considered for coding. Table 16 gives the number of quantization steps that the samples will be quantized to. The same table specifies, whether grouping of a granule shall be used or not. If grouping is not required, the three samples shall be coded with three individual codewords.

If grouping of a granule is required, which depends on the number of quantization steps m (m = 3, 5 or 9), the three consecutive sub-band samples shall be coded with one codeword. Only one value $v_m$, Most Significant Byte (MSB) first, shall be transmitted for this grouped granule. The relationship between the coded value $v_m$ and the three samples $x, y, z$ of a granule shall be one of the following:

$$v_3 = 9z + 3y + x \qquad (v_3 \text{ in } 0\ldots 26)$$

$$v_5 = 25z + 5y + x \qquad (v_5 \text{ in } 0\ldots 124)$$

$$v_9 = 81z + 9y + x \qquad (v_9 \text{ in } 0\ldots 728)$$

**Table 16: Classes of quantization**

| No. of steps | C | D | Grouping | Samples per codeword | N Bits per codeword |
|---|---|---|---|---|---|
| 3 | 1,33333333333 | 0,50000000000 | yes | 3 | 5 |
| 5 | 1,60000000000 | 0,50000000000 | yes | 3 | 7 |
| 7 | 1,14285714286 | 0,25000000000 | no | 1 | 3 |
| 9 | 1,77777777777 | 0,50000000000 | yes | 3 | 10 |
| 15 | 1,06666666666 | 0,12500000000 | no | 1 | 4 |
| 31 | 1,03225806452 | 0,06250000000 | no | 1 | 5 |
| 63 | 1,01587301587 | 0,03125000000 | no | 1 | 6 |
| 127 | 1,00787401575 | 0,01562500000 | no | 1 | 7 |
| 255 | 1,00392156863 | 0,00781250000 | no | 1 | 8 |
| 511 | 1,00195694716 | 0,00390625000 | no | 1 | 9 |
| 1 023 | 1,00097751711 | 0,00195312500 | no | 1 | 10 |
| 2 047 | 1,00048851979 | 0,00097656250 | no | 1 | 11 |
| 4 095 | 1,00024420024 | 0,00048828125 | no | 1 | 12 |
| 8 191 | 1,00012208522 | 0,00024414063 | no | 1 | 13 |
| 16 383 | 1,00006103888 | 0,00012207031 | no | 1 | 14 |
| 32 767 | 1,00003051851 | 0,00006103516 | no | 1 | 15 |
| 65 535 | 1,00001525902 | 0,00003051758 | no | 1 | 16 |

### 7.1.9 Formatting of the Audio bit stream

The frame formatter of the audio encoder shall take the Bit Allocation, ScFSI, ScF and the quantized sub-band samples together with header information and a few code words used for error detection to format the ISO/IEC 11172-3 [3] Layer II bit stream. It shall further divide this bit stream into audio frames, each corresponding to 1152 PCM audio samples, which is equivalent to a duration of 24 ms. The principal structure of such an ISO/IEC 11172-3 [3] Layer II audio frame with its correspondence to the DAB audio frame can be seen in figure 22.

Each audio frame starts with a header, consisting of a syncword and audio system related information. A Cyclic Redundancy Check (CRC), following the header protects a part of the header information, the Bit Allocation, and the ScFSI fields. After the CRC follows Bit Allocation, ScFSI and Scale Factors. The sub-band samples, which will be used by the decoder to reconstruct the PCM audio signal, are the last audio data part in the ISO/IEC 11172-3 [3] Layer II audio frame before the ancillary data field. This ancillary data field, which is of variable length, is located at the end of the ISO/IEC 11172-3 [3] Layer II audio frame. The details of the content of the audio frame can be found in subclause 7.3.

An adaptation of the ISO/IEC 11172-3 [3] Layer II audio frame to the DAB audio frame is performed in order to introduce:

- specific DAB Scale Factor Error Check (ScF-CRC);

- a fixed and a variable field of Programme Associated Data (F-PAD and X-PAD).

The lower part of figure 22 indicates how this additional specific information, necessary for DAB, shall be inserted into the ancillary data field of the ISO/IEC 11172-3 [3] Layer II audio frame.

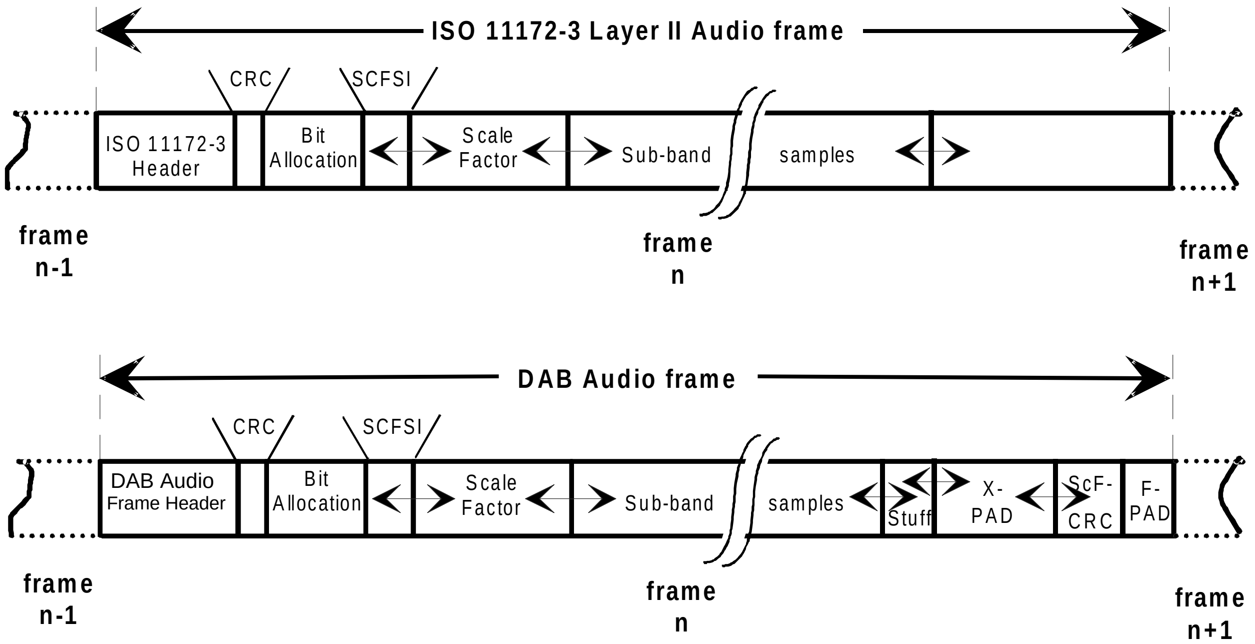


**Figure 22: Frame structure of ISO/IEC 11172-3 [3] Layer II and corresponding DAB audio bit stream**

The first four bytes of the DAB audio frame contain the ISO/IEC 11172-3 [3] header. This header carries information for the audio decoder. In the DAB system, some of this information is currently defined as static information. This is:

- **syncword:** set to external synchronization of the audio decoder;
- **ID:** set to ISO/IEC 11172-3 [3] (ID = MPEG Audio);
- **Layer:** set to Layer II (layer = Layer II);
- **protection_bit:** set to CRC protection on.

### 7.2 Semantics of the audio bit stream

The following subclauses are describing the specific semantic meaning of the ISO/IEC 11172-3 [3] Layer II bit stream (subclause 7.2.1) and its correspondence to the DAB audio bit stream (subclause 7.2.2).

#### 7.2.1 ISO/IEC 11172-3 Layer II bit stream

##### 7.2.1.1 Audio sequence

The DAB audio coding system uses the ISO/IEC 11172-3 [3] Layer II format. A graphic representation of an audio frame in ISO/IEC 11172-3 [3] Layer II format is given in the upper part of figure 22.

**Audio_frame:** part of the bit stream that is decodable by itself. It contains information for 1 152 samples. It starts with a syncword, and ends just before the next syncword. It consists of an integer number of slots. A slot contains one byte.

#### 7.2.1.2 Audio frame

**header:** part of the bit stream containing synchronisation and state information.

**error_check:** part of the bit stream containing information for error detection of significant audio side information.

**audio_data:** part of the bit stream containing information on the audio samples.

**ancillary_data:** part of the bit stream that may be used for ancillary data.

#### 7.2.1.3 Audio frame header

The first 32 bits (four bytes) are header information.

**syncword:** the bit string "1111 1111 1111".

**ID:** one bit to indicate the ID of the algorithm. Equals "1" for MPEG audio, "0" for future use ("0" is used for MPEG-2 Audio ISO/IEC DIS 13818-3 [14]).

**Layer:** 2 bits to indicate which layer is used, according to the following table 17.

**Table 17: Indication of ISO/MPEG-Audio Layer**

| Code | Layer |
|---|---|
| "11" | not used in DAB |
| "10" | Layer II |
| "01" | not used in DAB |
| "00" | reserved |

**protection_bit:** one bit to indicate whether redundancy has been added in the audio bit stream to facilitate error detection and concealment. Equals "0" because redundancy shall be added for DAB application.

**bit_rate_index:** indicates the bit rate. The bit_rate_index is an index to a specified bit rate, shown in table 18.

**Table 18: Specified total bit rates per audio programme**

| bit_rate_index | bit rate specified |
|---|---|
| "0000" | not used in DAB |
| "0001" | 32 kbit/s |
| "0010" | 48 kbit/s |
| "0011" | 56 kbit/s |
| "0100" | 64 kbit/s |
| "0101" | 80 kbit/s |
| "0110" | 96 kbit/s |
| "0111" | 112 kbit/s |
| "1000" | 128 kbit/s |
| "1001" | 160 kbit/s |
| "1010" | 192 kbit/s |
| "1011" | 224 kbit/s |
| "1100" | 256 kbit/s |
| "1101" | 320 kbit/s |
| "1110" | 384 kbit/s |
| "1111" | forbidden |

The bit_rate_index indicates the total bit rate irrespective of the mode (stereo, joint_stereo, dual_channel, single_channel). The total bit rate includes all bits in an audio frame, i.e. all bits necessary for header, audio signal, PAD and error detection information.

NOTE: In order to provide the smallest possible delay and complexity, the decoder is not required to support a continuously variable bit rate. However, the bit rate may change from time to time during continuing service. The smallest resolution for changing the bit rate is 6 s (see subclause 6.5 on Multiplex re-configuration).

The encoder in the DAB transmitter should support at least one of these bit rates given in table 18, whereas the audio decoder must be capable of working at all these bit rates. Not all combinations of total bit rates and audio modes are allowed. Table 19 shows the audio modes which can be chosen, dependent on the bit rate.

**Table 19: Combinations of total bit rates per audio programme and audio modes**

| total bit rate | Audio modes |
|---|---|
| 32 kbit/s | single_channel |
| 48 kbit/s | single_channel |
| 56 kbit/s | single_channel |
| 64 kbit/s | all modes |
| 80 kbit/s | single_channel |
| 96 kbit/s | all modes |
| 112 kbit/s | all modes |
| 128 kbit/s | all modes |
| 160 kbit/s | all modes |
| 192 kbit/s | all modes |
| 224 kbit/s | stereo, intensity_stereo, dual_channel |
| 256 kbit/s | stereo, intensity_stereo, dual_channel |
| 320 kbit/s | stereo, intensity_stereo, dual_channel |
| 384 kbit/s | stereo, intensity_stereo, dual_channel |

**sampling_frequency:** indicates the sampling frequency, according to table 20. The DAB system uses the value of "01", indicating a sampling frequency of 48 kHz.

**Table 20: Specified sampling frequencies per PCM audio input/output signal**

| sampling_frequency | frequency specified |
|---|---|
| "00" | not used in DAB |
| "01" | 48 kHz |
| "10" | reserved for future use (32 kHz used in ISO/IEC 11172-3) [3] |
| "11" | reserved for future use |

**padding_bit**: fixed value of "0". No padding is necessary for 48 kHz sampling frequency.

**private_bit:** bit for private use. This bit will not be used in the future by ISO/IEC 11172-3 [3], and is not interpreted by an ISO/IEC 11172-3 [3] decoder.

**mode:** indicates the audio mode according to table 21. In Layer II the joint_stereo mode is intensity_stereo.

**Table 21: Audio modes which can be selected in the audio encoder**

| mode | Audio mode specified |
|---|---|
| "00" | stereo |
| "01" | joint_stereo (intensity_stereo) |
| "10" | dual_channel |
| "11" | single_channel |

In all audio modes, except joint_stereo, the value of bound equals sblimit. In joint_stereo mode the bound is determined by the mode_extension.

**mode_extension:** these bits are used in joint_stereo mode, and indicate which sub-bands are in intensity_stereo. All other sub-bands are coded in stereo mode. The figures are given by table 22.

**Table 22: Sub-bands in intensity stereo mode, indicated by the mode_extension**

| mode_extension | Sub-bands in intensity_stereo | Bound |
|---|---|---|
| "00" | 4-31 | bound == 4 |
| "01" | 8-31 | bound == 8 |
| "10" | 12-31 | bound == 12 |
| "11" | 16-31 | bound == 16 |

**copyright:** if this bit equals "0" there is no copyright on the ISO/IEC 11172-3 [3] audio bit stream; "1" means copyright protected.

**original/copy:** this bit equals "0" if the bit stream is a copy, "1" if it is an original.

**emphasis:** indicates the type of de-emphasis that shall be used by an ISO/IEC 11172-3 [3] decoder. The DAB system shall use a fixed value of "00", indicating "no emphasis", and therefore the use of pre/de-emphasis is excluded.

**Table 23: emphasis of the input PCM audio signal**

| emphasis | emphasis specified |
|---|---|
| "00" | no emphasis |
| "01" | not used in DAB |
| "10" | not used in DAB |
| "11" | not used in DAB |

### 7.2.1.4 Error check

**crc_check:** a 16 bit parity check word used for error detection of the most error-sensitive part of the audio information within the encoded audio bit stream. This information includes the third and fourth bytes of the ISO/IEC 11172-3 [3] header, Bit Allocation and Scale Factor Select Information (see for more details annex B, clause B.2).

### 7.2.1.5 Audio data

**allocation[ch][sb]:** contains information concerning the quantizers used for the samples in sub-band sb in channel ch, whether the information on three consecutive samples of a granule has been grouped to one code, and on the number of bits used to code the samples. The meaning and length of this field depends on the number of the sub-band, the bit rate, and the sampling frequency. The bits in this field form an unsigned integer used as an index to the relevant Bit Allocation table 13 or 14, which gives the number of levels `nlevels` used for quantization. For sub-bands in intensity_stereo mode the bit-stream contains only one allocation data element per sub-band.

**Table 24: Transmission of Scale Factors dependent on ScFSI**

| scfsi [ch][sb] | action |
|---|---|
| "00" | three ScFs transmitted, for parts 0,1,2 respectively. |
| "01" | two ScFs transmitted, first one valid for parts 0 and 1, second one for part 2. |
| "10" | one ScF transmitted, valid for all three parts. |
| "11" | two ScFs transmitted, first one valid for part 0, the second one for parts 1 and 2. |

**scfsi[ch][sb]:** Scale Factor selection information. This gives information on the number of Scale Factors transferred for sub-band sb in channel ch and for which parts of the signal in this frame they are valid (see table 24). The frame is divided into three equal parts of 12 sub-band samples each per sub-band.

**scalefactor[ch][sb][p]:** indicates the factor by which the re-quantized samples of sub-band sb in channel ch and of part p of the frame should be multiplied. The six bits constitute an unsigned integer, index to table 10, showing the Scale Factors.

**grouping[ch][sb]:** is a function that determines, whether grouping is in effect for coding of samples in sub-band sb of channel ch. Grouping means, that three consecutive samples of the current sub-band sb in channel ch which form the granule gr are coded and transmitted using one common codeword and not using three separate codewords. Grouping[ch][sb] is true, if in the Bit Allocation table currently in use (see either table 13 or 14) the value found under sb (first row) and allocation[ch][sb] (column) is either 3, 5, or 9. Otherwise it is false. For sub-bands in Intensity stereo mode the grouping is valid for both channels.

**samplecode[ch][sb][gr]:** coded representation of the three consecutive samples in the granule gr in sub-band sb of channel ch. For sub-bands in Intensity stereo mode the coded representation of the samplecode is valid for both channels.

**sample[ch][sb][s]:** coded representation of the s-th sample in sub-band sb of channel ch. For sub-bands in Intensity stereo mode the coded representation of the sample is valid for both channels.

#### 7.2.1.6 Ancillary data

**ancillary_bit:** user definable.

The number of ancillary bits (no_of_ancillary_bits) equals the available number of bits in an audio frame minus the number of bits actually used for header, error check and audio data. The no_of_ancillary_bits corresponds to the distance between the end of the audio data in an ISO/IEC 11172-3 [3] Layer II audio frame, and the beginning of the header of the next audio frame.

### 7.2.2 DAB audio bit stream

The DAB system uses the ISO/IEC 11172-3 [3] Layer II format with additional specific information, necessary for the DAB application.

The field for this additional specific information is defined in DAB to contain the DAB fields Extended Programme Associated Data (X-PAD), Audio Scale Factor Error Check (ScF-CRC) and Fixed Programme Associated Data (F-PAD) (see subclauses 7.2.2.7, 7.2.2.8 and 7.2.2.9).

#### 7.2.2.1 DAB audio sequence

A detailed graphic representation of the content and the structure of a DAB audio frame is given in figures 23 and 24.

**DAB_audio_frame:** part of the bit stream that is decodable by itself. Besides the information for 1 152 audio samples, it contains all specific DAB audio information (see also definition given in subclause 7.2.1.1).

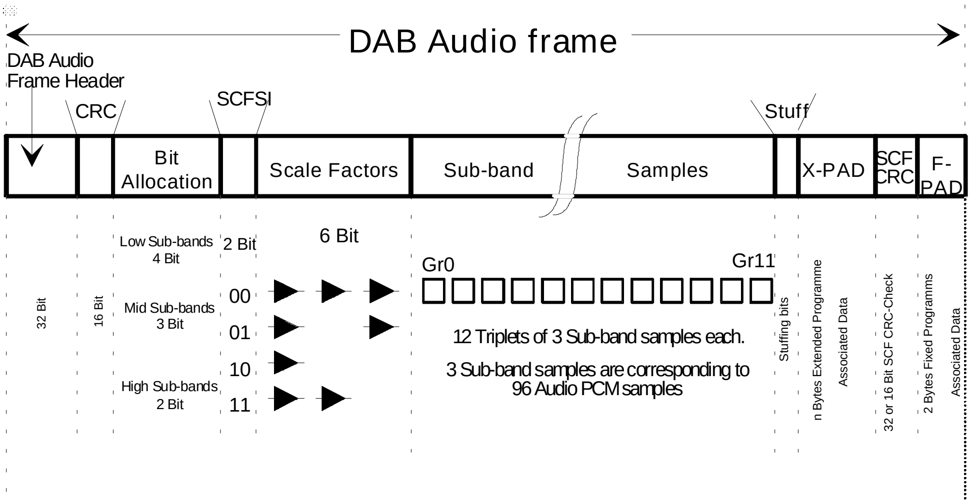


**Figure 23: Structure of the DAB audio frame**

### 7.2.2.2 DAB audio frame

**DAB_audio_frame_header:** part of the bit stream (the first 32 bits) containing relevant state information for the DAB audio decoder.

**error_check:** see definition given in subclause 7.2.1.2.

**audio_data:** see definition given in subclause 7.2.1.2.

**audio_stuffing_bits:** number of stuff bits inserted between the end of audio_data and the beginning of x_prog_ass_data.

**x_prog_ass_data:** part of the DAB audio frame with variable length in multiples of bytes, that may be used for Programme Associated Data.

**scf_error_check:** part of the DAB audio frame containing information for error detection of ScFs.

**f_prog_ass_data:** part of the frame with constant length of two bytes, that may be used for Programme Associated Data.

### 7.2.2.3 DAB audio frame header

The DAB audio frame header is identical to the ISO/IEC 11172-3 [3] header.

For the semantic meaning of the following parameters:

- bitrate_index;
- sampling_frequency;
- padding_bit;
- private_bit;
- mode;
- mode_extension;
- copyright;
- original/copy;

- copyright;
- original/copy;
- emphasis.

See definitions given in subclause 7.2.1.3.

#### 7.2.2.4 Error check

See definitions given in subclause 7.2.1.4.

#### 7.2.2.5 Audio data

See definitions given in subclause 7.2.1.5.

#### 7.2.2.6 Audio stuffing bits

The total number of bits available for audio_data per DAB audio frame equals (bit_rate * 0,024) minus bits used by DAB_audio_frame_header(), error_check(), x_prog_ass_data(), scf_error_check(), and f_prog_ass_data(). The number of bits actually used by audio_data may be less. In that case a number of stuffing bits are inserted between the end of audio_data and the beginning of x_prog_ass_data().

**stuff_bit:** single bit without useful information. This bit is not defined in DAB. Stuffing bits fill the space from the start of the ISO/IEC 11172-3 [3] ancillary data field up to the beginning of the X-PAD field.

#### 7.2.2.7 Extended Programme Associated Data (X-PAD)

**x-pad_byte[i]:** $i^{th}$ byte of X-PAD. A variable number of bytes, no_of_x-pad_bytes is used for X-PAD, i.e. the length of this field is variable in multiples of bytes. The actual value is given in subclause 7.4.

#### 7.2.2.8 Scale Factor Error Check (ScF-CRC)

**scf-crc_check[i]:** $i^{th}$ word used for Cyclic Redundancy Check (CRC) words, each protecting the Scale Factors of a group of sub-bands of the following DAB audio frame. The actual number of CRC words, no_of_scf_error_checks depends upon the bit-rate and audio mode (see annex B, clause B.3). Either two or four 8-bit cyclic redundancy check words are used for error detection of two or four different spectral groups of Scale Factors within the encoded bit stream.

#### 7.2.2.9 Fixed Programme Associated Data (F-PAD)

**f-pad_byte[i]:** $i^{th}$ byte of F-PAD. A fixed number of two bytes for Fixed Programme Associated Data (F-PAD), comprising the last two bytes of the DAB audio frame, is used.

**Frame Structure of coded bitstream: valid for 1152 PCM Audio Input Samples (Stereo Mode)**
**Duration of a Frame: 24 ms**

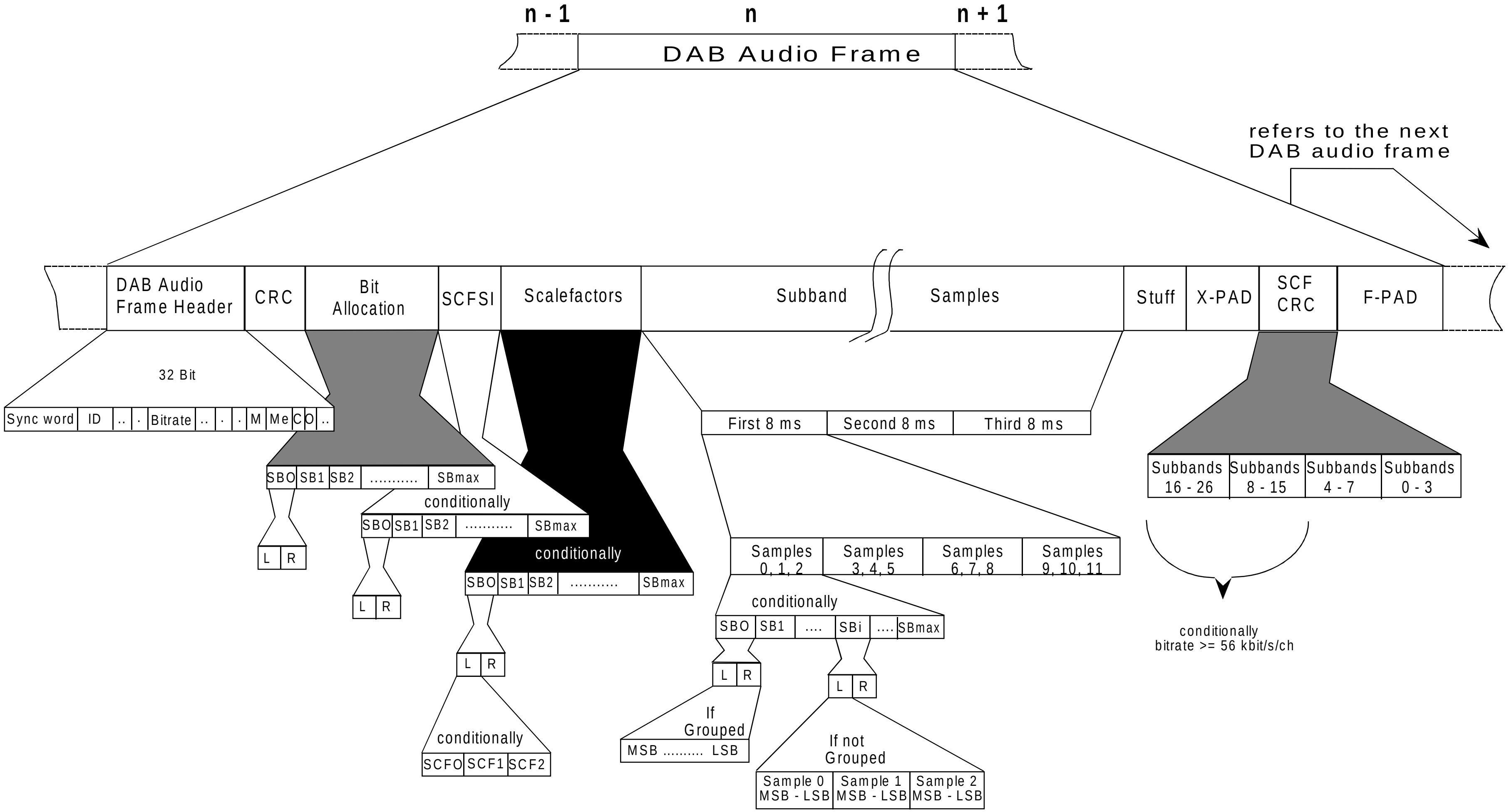


Figure 24: DAB Audio Frame structure

## 7.3 Audio bit stream syntax

The details of the audio frame can be found in this subclause, which describes the ISO/IEC 11172-3 [3] Layer II bit stream syntax (subclause 7.3.1), and the DAB audio bit stream syntax (subclause 7.3.2). A detailed structure of the DAB audio frame is given in figure 24.

### 7.3.1 ISO/IEC 11172-3 Layer II bit stream syntax

This syntax is valid at the output of an ISO/IEC 11172-3 [3] Layer II audio encoder and at the input of an ISO/IEC 11172-3 [3] Layer II decoder.

#### 7.3.1.1 Audio sequence

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| audio sequence() | | |
| { | | |
| while (nextbits()==syncword) { | | |
| frame() | | |
| } | | |
| } | | |

#### 7.3.1.2 Audio frame

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| frame() | | |
| { | | |
| header() | | |
| error_check() | | |
| audio_data() | | |
| ancillary_data() | | |
| } | | |

#### 7.3.1.3 Header

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| header() | | |
| { | | |
| **syncword** | **12** | **bslbf** |
| **ID** | **1** | **bslbf** |
| **layer** | **2** | **bslbf** |
| **protection_bit** | **1** | **bslbf** |
| **bitrate_index** | **4** | **bslbf** |
| **sampling_frequency** | **2** | **bslbf** |
| **padding_bit** | **1** | **bslbf** |
| **private_bit** | **1** | **bslbf** |
| **mode** | **2** | **bslbf** |
| **mode_extension** | **2** | **bslbf** |
| **copyright** | **1** | **bslbf** |
| **original/copy** | **1** | **bslbf** |
| **emphasis** | **2** | **bslbf** |
| } | | |

### 7.3.1.4 Error check

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| error_check() | | |
| { | | |
| if (protection_bit==0) | | |
| **crc_check** | **16** | **rpchof** |
| } | | |

### 7.3.1.5 Audio data

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| audio_data() | | |
| { | | |
| for (sb=0; sb<bound; sb++) | | |
| for (ch=0; ch<nch; ch++) | | |
| **allocation[ch][sb]** | **2...4** | **uimsbf** |
| for (sb=bound; sb<sblimit; sb++) { | | |
| **allocation[0][sb]** | **2...4** | **uimsbf** |
| allocation[1][sb]=allocation\|0]\|sb] | | |
| } | | |
| for (sb=0; sb<sblimit; sb++) | | |
| for (ch=0; ch<nch; ch++) | | |
| if (allocation[ch][sb]!=0) | | |
| **scfsi[ch][sb]** | **2** | **bslbf** |
| for (sb=0; sb<sblimit; sb++) | | |
| for (ch=0; ch<nch; ch++) | | |
| if (allocation[ch][sb]!=0) { | | |
| if (scfsi[ch][sb]==0) { | | |
| **scalefactor[ch][sb][0]** | **6** | **uimsbf** |
| **scalefactor[ch][sb][1]** | **6** | **uimsbf** |
| **scalefactor[ch][sb][2]** | **6** | **uimsbf** |
| } | | |
| if ((scfsi[ch][sb]==1) \|\| (scfsi[ch][sb]==3)){ | | |
| **scalefactor[ch][sb][0]** | **6** | **uimsbf** |
| **scalefactor[ch][sb][2]** | **6** | **uimsbf** |
| } | | |
| if (scfsi[ch][sb]==2) | | |
| **scalefactor[ch][sb][0]** | **6** | **uimsbf** |
| } | | |
| for (gr=0; gr<12; gr++) { | | |
| for (sb=0; sb<bound; sb++) | | |
| for (ch=0; ch<nch; ch++) | | |
| if (allocation[ch][sb]!=0) { | | |
| if (grouping[ch][sb]) | | |
| **samplecode[ch][sb][gr]** | **5...10** | **uimsbf** |
| else | | |
| for (s=0; s<3; s++) | | |
| **sample[ch][sb][3*gr+s]** | **3..16** | **uimsbf** |
| } | | |
| for (sb=bound; sb<sblimit; sb++) | | |
| if (allocation[0][sb]!=0) { | | |
| if (grouping[0][sb]) | | |
| **samplecode[0][sb][gr]** | **5...10** | **uimsbf** |
| else | | |
| for (s=0; s<3; s++) | | |
| **sample[0][sb][3*gr+s]** } | **3...16** | **uimsbf** |
| } | | |
| } | | |
| } | | |

#### 7.3.1.6 Ancillary data

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| ancillary_data() | | |
| { | | |
| if ((layer == 1) \|\| (layer == 2)) | | |
| for (b=0; b<no_of_ancillary_bits; b++) | | |
| **ancillary_bit** | **1** | **bslbf** |
| } | | |

### 7.3.2 DAB audio bit stream syntax

This syntax is valid at the input of the Conditional Access Scrambler (see figure 1) in the DAB transmitter.

#### 7.3.2.1 DAB audio sequence

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| dab_audio sequence() | | |
| { | | |
| while true { | | |
| dab_audio_frame() | | |
| } | | |
| } | | |

#### 7.3.2.2 DAB audio frame

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| dab_audio_frame() | | |
| { | | |
| dab_audio_frame_header() | | |
| error_check() | | |
| audio_data() | | |
| audio_stuffing_bits() | | |
| x_prog_ass_data() | | |
| scf_error_check() | | |
| f_prog_ass_data() | | |
| } | | |

### 7.3.2.3 DAB audio frame header

The DAB audio frame header is defined identically to the header defined in subclause 7.3.1.3.

### 7.3.2.4 Error check

| Syntax | No. of bits | Mnemonic |
|---|---|---|

See subclause 7.3.1.4.

### 7.3.2.5 Audio data

| Syntax | No. of bits | Mnemonic |
|---|---|---|

See subclause 7.3.1.5.

### 7.3.2.6 Audio stuffing bits

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| audio_stuffing_bits() | | |
| { | | |
| while (bitsum < (bit_rate * 0.024 - no_of_x-pad_bytes * 8 | | |
| - 2* 8 - no_of_scf_error_checks * 8)) { | | |
| **stuff_bit** | **1** | **bslbf** |
| bitsum++ | | |
| } | | |
| } | | |

### 7.3.2.7 Extended Programme Associated Data

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| x_prog_ass_data() | | |
| { | | |
| for (i=0; i<no_of_x-pad_bytes; i++) | | |
| **x-pad_byte(i)** | **8** | **bslbf** |
| } | | |

### 7.3.2.8 Scale Factor error check

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| scf_error_check() | | |
| { | | |
| for (i=no_of_scf_error_checks-1; i≥0; i--) | | |
| **scf-crc_check(i)** | **8** | **rpchof** |
| } | | |

### 7.3.2.9 Fixed Programme Associated Data

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| f_prog_ass_data() | | |
| { | | |
| for (i=0; i<2; i++) | | |
| **f-pad_byte(i)** | **8** | **bslbf** |
| } | | |

## 7.4 Programme Associated Data (PAD)

Each DAB audio frame contains a number of bytes which may carry Programme Associated Data (PAD). PAD is information which is synchronous to the audio and its contents may be intimately related to the audio. The PAD bytes in successive audio frames constitute the PAD channel. The functions provided by PAD are given in annex A, clause A.4.

In each DAB audio frame there are two bytes called the fixed PAD (F-PAD) field. The F-PAD field is intended to carry control information with a strong real-time character and data with a very low bit rate. The PAD channel may be extended using an extended PAD (X-PAD) field, intended to carry information providing additional functions to the listener, such as programme related text. The length of the X-PAD field is chosen by the service provider.

The use of PAD is optional. Therefore, it is not mandatory to send any information in the F-PAD field. If no information is sent, all bytes in the F-PAD field shall be set to zero, which also implies that no X-PAD field is present.

The PAD carried in the DAB audio frame n shall be associated with the audio carried in the following frame, n+1.

If functions in PAD are used in Dual channel mode, they shall apply to channel 0 unless otherwise signalled by the application.

Figure 25 shows the location of the F-PAD and X-PAD fields within the DAB audio frame.

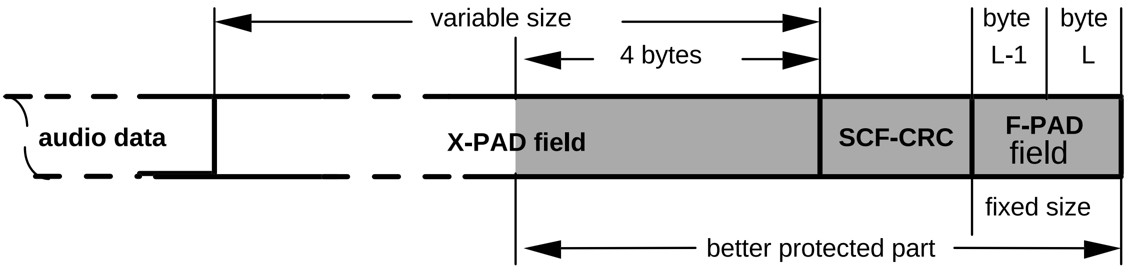


**Figure 25: Location of the PAD bytes within the DAB audio frame**

The two bytes of the F-PAD field (byte L-1 and byte L ) are located at the end of the DAB audio frame, following the Scale Factor CRC (ScF-CRC). The X-PAD field is located just before the ScF-CRC. The audio data must terminate before the beginning of the X-PAD field.

The F-PAD channel carries a two-bit field, "X-PAD Ind", which indicates one of three possibilities for the length of the X-PAD field:

1) No X-PAD: only the F-PAD field is available. All bits in the frame up to the ScF-CRC may be filled with audio data.

2) Short X-PAD: in this case the length of the X-PAD field is four bytes in every DAB audio frame, and the entire X-PAD field lies in the better protected part of the DAB audio frame (i.e. is as well protected as the ScF-CRC). In total, 6 bytes carry PAD.

3) Variable size X-PAD: in this case the length of the X-PAD field may vary from frame to frame. The length of the X-PAD field in the current DAB audio frame can be deduced from the contents information carried within the X-PAD field. Only a part (4 bytes) of the X-PAD field is as well protected as the ScF-CRC. The remainder has a lower protection. Application data carried in the X-PAD channel may require further error protection.

### 7.4.1 Coding of F-PAD

Figure 26 shows the structure of the F-PAD field. The information in byte L-1 is carried time-multiplexed in one-byte groups that are individually identified by their "F-PAD type". The information carried in byte L-1 shall be valid for the current and forthcoming audio frames, until it is replaced by new information.

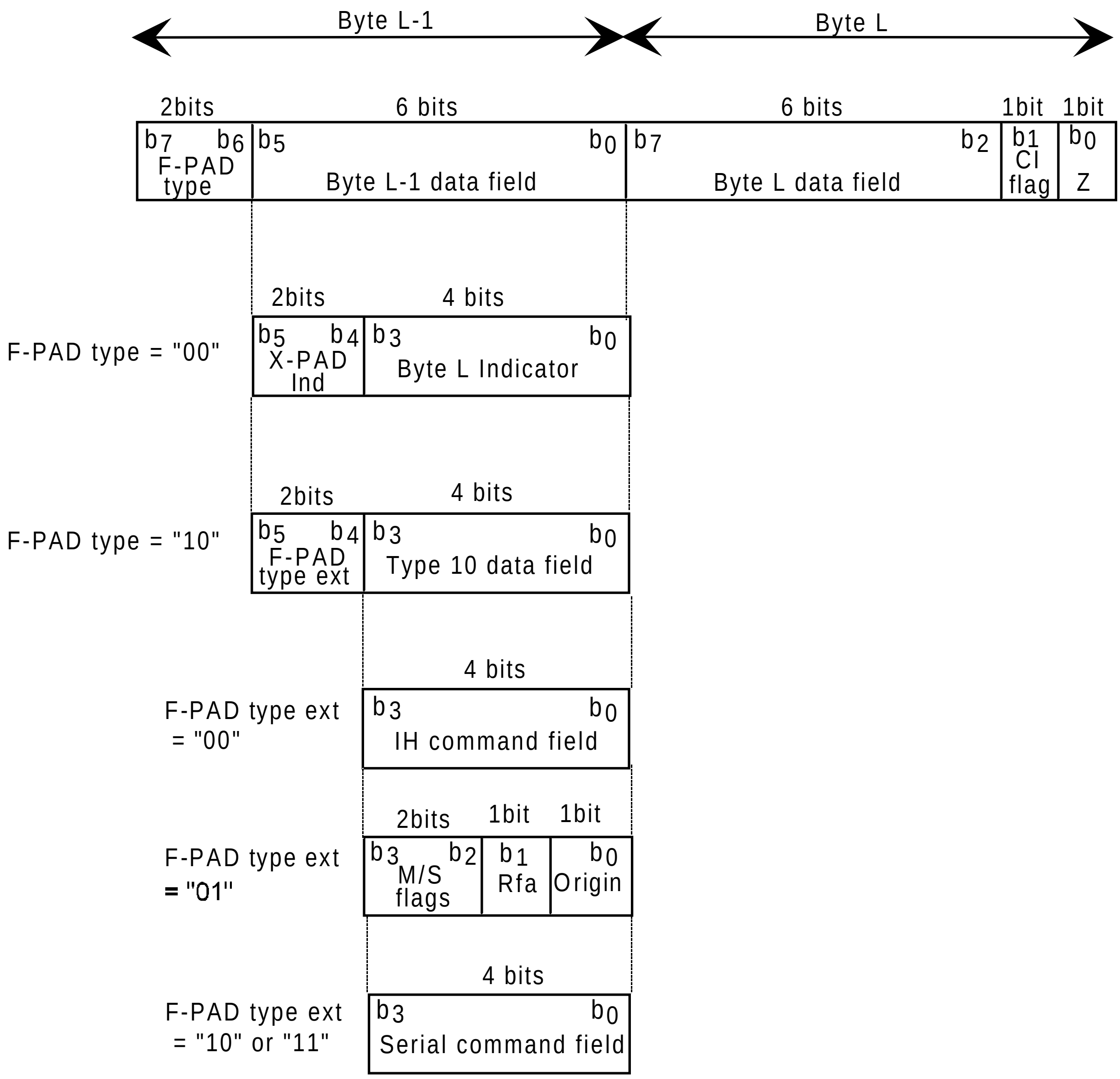


**Figure 26: Structure of the F-PAD field**

The following definitions apply for byte L-1:

**F-PAD type**: this 2-bit field shall indicate the content of the byte L-1 data field. The values "01" and "11" are reserved for future use of the byte L-1 data field.

**F-PAD type "00":**

- **X-PAD Ind** (X-PAD Indicator): this 2-bit field shall indicate the presence and length of the X-PAD field, as follows:

  $b_5$ $b_4$
  0 0 : no X-PAD;
  0 1 : short X-PAD;
  1 0 : variable size X-PAD;
  1 1 : reserved for future use.

- **Byte L Indicator**: this 4-bit field shall indicate the data content of the byte L data field, as follows:

  $b_3$ $b_0$
  0 0 0 0 : In-house information, or no information;
  0 0 0 1 : DRC (Dynamic Range Control) data.

  The remaining values are reserved for future use of the byte L data field.

  The coding of the in-house information is not subject to standardization. The coding details of the DRC signal are specified in subclause 7.4.1.1.

**F-PAD type "10":**

- **F-PAD type ext** (F-PAD type extension)**:** this 2-bit field shall indicate the contents of the type 10 data field, as follows:

  $b_5$ $b_4$
  0 0 : in-house real-time commands;
  0 1 : Music/Speech indication and Origin;
  1 0 : serial command channel (start);
  1 1 : serial command channel (continuation).

- **F-PAD type extension "00":** the in-house real-time commands are not subject to standardization.

- **F-PAD type extension "01":**

  - **M/S flags** (Music/Speech flags)**:** This 2-bit flag field shall indicate whether the audio consists of music or speech, or that no music/speech indication is given, as follows:

    $b_3$ $b_2$
    0 0 : Music/Speech is not signalled;
    0 1 : Music;
    1 0 : speech;
    1 1 : reserved for future use.

  - **Rfa:** This 1-bit field shall be reserved for future additions. The bit shall be set to "0" until it is defined.

  - **Origin:** This 1-bit field should carry one bit of the ISRC (see ISO 3901 [9]) or EN 797 [15]). The information shall be carried one bit at the time in a packet structure.

    A packet shall consist of a sync word, a data identifier and a data field.

    The sync word shall consist of the following 9 bits, in the order of transmission: "1111 1111 0".
    The data identifier shall comprise 3 bits, as follows (in the order of transmission):

    0 0 1 : ISRC (ISO 3901 [9]);
    0 1 0 : UPC/EAN (EN 797 [15]).

All other identifier codes are reserved for future use.

In the case of ISRC, the data field shall consist of 58 bits, representing 5 letter codes (of 6 bits each) followed by 7 digit codes (of 4 bits each).

In the case of UPC/EAN, the data field shall consist of 52 bits, representing 13 digit codes (of 4 bits each).

Each letter or digit code shall be transmitted with MSb first.

If both ISRC and UPC/EAN are transmitted, their packets shall be conveyed alternately. If neither ISRC nor UPC/EAN is transmitted, then the origin bit shall be set to zero.

- **F-PAD type extensions "10" and "11":**

  - **Serial command field:** this 4-bit field is reserved for future use.

The following definitions apply for byte L:

**Byte L data field**: the contents of this 6-bit field depend on the byte L indicator. The coding details for DRC are given in subclause 7.4.1.1. The in-house information is not subject to standardization.

**CI** (Contents Indicator) **flag:** this 1-bit flag shall signal whether the X-PAD field in the current DAB audio frame includes a contents indicator, as follows:

0 : no contents indicator;
1 : contents indicator(s) present.

**Z**: this bit shall be set to "0" for synchronization purposes in serial communication links.

### 7.4.1.1 Dynamic Range Control data

The DRC data is carried in the byte L data field as described above. The DRC data can be used in the receiver to set the gain of a variable gain amplifier. The DRC data shall be coded as follows:

$b_7$ $b_2$ : DRC signal;
0 0 0 0 0 0 : 0 dB;
0 0 0 0 0 1 : +0,25 dB;
0 0 0 0 1 0 : +0,50 dB;
↓ ↓ ↓ ↓ ↓ ↓
↓ ↓ ↓ ↓ ↓ ↓ : continuous steps of +0,25 dB;
↓ ↓ ↓ ↓ ↓ ↓
1 1 1 1 1 1 : +15,75 dB.

When DRC is used, the current DRC data shall be conveyed in every DAB audio frame. The control information in byte L-1 is time-multiplexed which allows for an independent operation of DRC data and the Music/Speech flags.

### 7.4.2 Structure of X-PAD

Application data shall be structured into X-PAD data groups for transport within the X-PAD channel. The structure and content of X-PAD data groups are defined in subclause 7.4.5.

The X-PAD field contains at least one X-PAD data subfield, and possibly a number of contents indicators. A contents indicator signals the application type for the data carried in the associated X-PAD data subfield, and when appropriate also the length of the subfield. The contents indicators may be omitted under certain conditions. Further details on the allowed number of X-PAD data subfields and the use of contents indicators are specified in subclauses 7.4.2.1, 7.4.2.2 and 7.4.4.

Data carried in the X-PAD field is defined in logical order. Before transmission, the order of the bytes within each X-PAD field shall be reversed. The reversed order applies only to the byte sequence; the bit order within each byte shall not be reversed; MSb shall be transmitted first. This means that the application data is "transmitted before" the contents indicator(s), and that the contents indicator(s) are carried just before the ScF-CRC.

Figures 27 and 28 illustrate two examples of how application data are transported in the X-PAD channel. In the first example the X-PAD data group extends over a number of X-PAD fields (i.e. a number of DAB audio frames), and in the second example three X-PAD data groups are carried in one X-PAD field (i.e. one DAB audio frame).

The X-PAD indicator (X-PAD Ind, transported in the F-PAD channel) signals whether no X-PAD, short X-PAD or variable size X-PAD is used. The CI flag signals whether the X-PAD field contains one or more contents indicators (CI) or only application data.

In the first example (figure 27) a single contents indicator is required to indicate the beginning of the X-PAD data group. In the second example (figure 27), three contents indicators are required, one for each data subfield. In addition, a fourth contents indicator, CI 4, is used to terminate the contents indicator list (see subclause 7.4.3).

NOTE: In figures 27 and 28 the logical order of the information is depicted (the transmission order within each DAB audio frame is reversed, as described above).

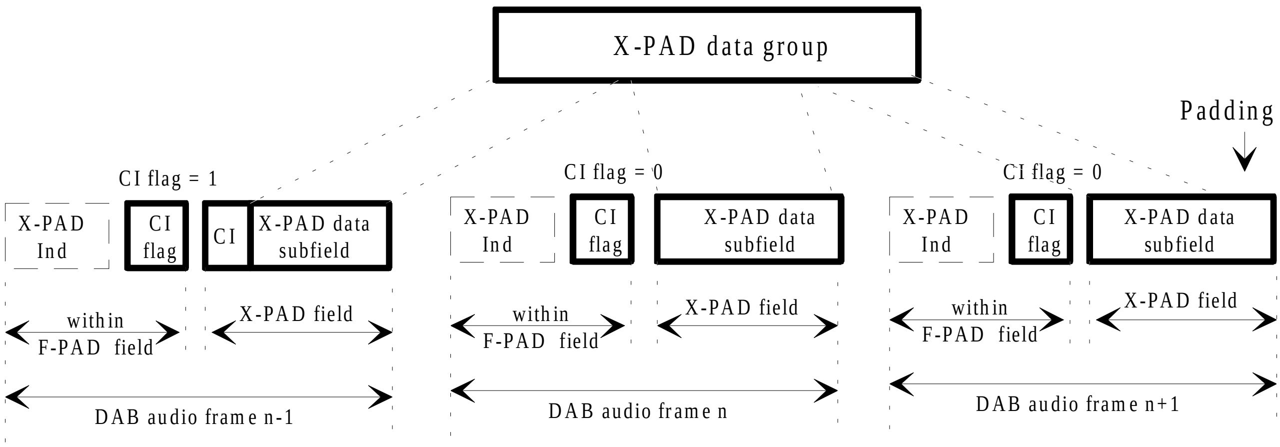


**Figure 27: An X-PAD data group extending over three consecutive X-PAD fields**

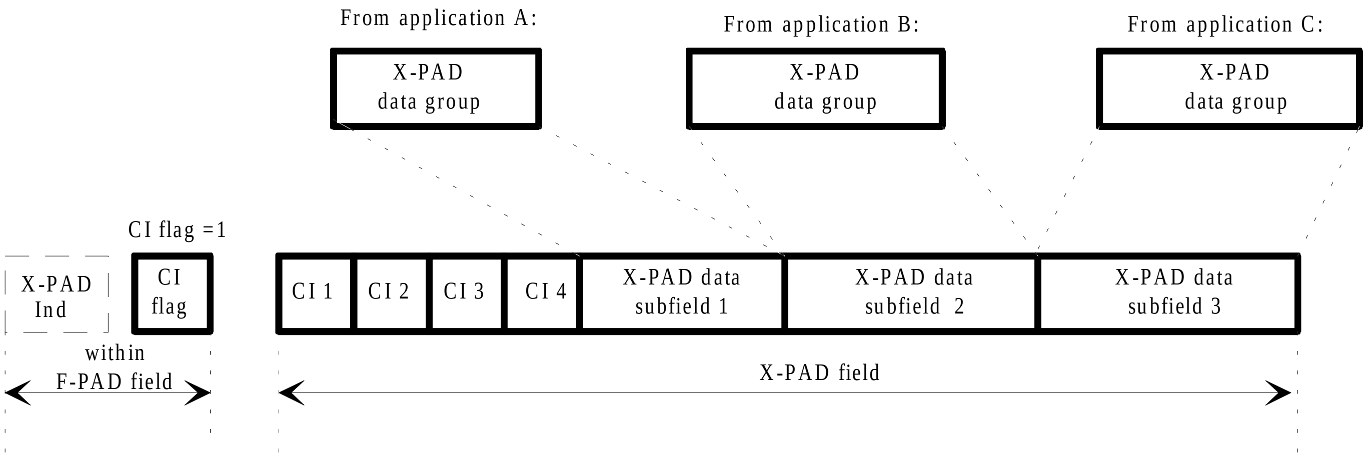


**Figure 28: Three X-PAD data groups carried in one X-PAD field**

An X-PAD data subfield may either contain a complete X-PAD data group or a part of a data group. An X-PAD data group may be interrupted by an X-PAD data group from a different application, and resumed later. In this way a time-critical application can be transported punctually, even if another application requires many X-PAD fields to transport one X-PAD data group.

### 7.4.2.1 Short X-PAD

The length of the X-PAD field shall be 4 bytes. Each X-PAD field shall comprise either one contents indicator and one X-PAD data subfield of 3 bytes, or one X-PAD data subfield of 4 bytes. The contents indicator shall signal the application type.

A contents indicator is required for the following two situations:

- when the X-PAD data subfield contains the start of an X-PAD data group;
- when the X-PAD data subfield contains data from an X-PAD data group that has been interrupted and now is being resumed.

The contents indicator may be omitted if the X-PAD data subfield (4 bytes long) contains a continuation of the X-PAD data group carried in the previous DAB audio frame.

If the final part of an X-PAD data group does not entirely fill the X-PAD data subfield in which it is transported, padding bits shall be appended to the X-PAD data group. Padding bits shall be set to zero.

The contents indicator flag, transported in the F-PAD field, shall signal for each DAB audio frame, whether the X-PAD field contains a contents indicator or not.

### 7.4.2.2 Variable size X-PAD

The length of the X-PAD field may vary from one DAB audio frame to the next.

The contents indicators shall, when present, be assembled in a contents indicator list in the beginning (logical meaning) of the X-PAD field. Each X-PAD field shall comprise either one X-PAD data subfield or a number of X-PAD data subfields, together with a contents indicator list.

Each contents indicator shall signal the application type for the data in the associated X-PAD data subfield and the length of the subfield. The total length of the X-PAD field may therefore be derived from the contents indicators.

The maximum number of data subfields within one X-PAD field is four. When more than one data subfield is included there shall be a contents indicator associated with each subfield. The order of the contents indicators within the list shall be the same as the order of the X-PAD data subfields, i.e. the first contents indicator is associated to the first X-PAD data subfield , the second contents indicator to the second X-PAD data subfield, etc.

The contents indicators are either 1 byte or 2 bytes long, depending on the application (details are given in subclause 7.4.4.2). The contents indicator list shall comprise up to 4 bytes, thereby allowing for a maximum number of X-PAD data subfields within one X-PAD field being one of the following:

- four X-PAD data subfields with 1-byte contents indicators;

- two X-PAD data subfields with 1-byte contents indicators and one X-PAD data subfield with a 2-byte contents indicator;

- two X-PAD data subfields with 2-bytes contents indicators.

If the contents indicator list is shorter than four bytes, an end marker, consisting of a contents indicator of application type 0, shall be used to terminate the list, see subclause 7.4.3.

A contents indicator list shall be included whenever any of the following situations apply:

- when there is more than one X-PAD data subfield in the X-PAD field;

- when an X-PAD data subfield contains the start of an X-PAD data group;

- when an X-PAD data subfield contains data from an X-PAD data group that has been interrupted and is now being resumed;

- when the capacity of the X-PAD channel is changed;

- when there is only one X-PAD data subfield which does not fill the X-PAD field.

The contents indicator list may be omitted if both of the following conditions apply:

- the length of the X-PAD field is the same as in the previous DAB audio frame;

- the X-PAD field comprises a single data subfield containing a continuation of the X-PAD data group carried in the last (logical meaning) X-PAD data subfield of the previous DAB audio frame.

If the final part of an X-PAD data group does not entirely fill the X-PAD data subfield in which it is transported, padding bits shall be appended to the X-PAD data group. Padding bits shall be set to zero.

The contents indicator flag, transported in the F-PAD field, shall signal for each DAB frame, whether the X-PAD field contains contents indicators or not.

### 7.4.3 Application types

There are a maximum of 287 application types available. The application types are arranged in an application type table consisting of three parts:

1) the first part comprises the application types 0 to 30;

2) the second part comprises the application types 31 to 255;

3) the third part comprises the application types 256 to 286.

For applications that may generate long X-PAD data groups, two application data types are defined: one is used to indicate the start of an X-PAD data group and the other is used to indicate the continuation of a data group after an interruption.

Application types 0, 2 to 11, 32 and 33 are defined in table 25. Application type 0 has a special meaning (see below). The remaining application types are reserved for future definition.

Table 25: X-PAD Application types

| Application type | Description |
|---|---|
| 0 | End marker |
| 2 | Dynamic label segment, start of X-PAD data group |
| 3 | Dynamic label segment, continuation of X-PAD data group |
| 4 | ITTS, start of X-PAD data group |
| 5 | ITTS, continuation of X-PAD data group |
| 6 | In-house information, start of X-PAD data group |
| 7 | In-house information, continuation of X-PAD data group |
| 8 | Closed user-group channel, start of X-PAD data group |
| 9 | Closed user-group packet channel, continuation of X-PAD data group |
| 10 | Table of contents, start of X-PAD data group |
| 11 | Table of contents, continuation of X-PAD data group |
| 32 | Closed user-group stream channel, start of X-PAD data group |
| 33 | Closed user-group stream channel, continuation of X-PAD data group |

The end marker shall be used for three different purposes:

1) to terminate the contents indicator list when, for the variable size X-PAD, there is a contents indicator list shorter than four bytes;

2) to signal that the X-PAD field contains no data;

3) to terminate the list of applications within an X-PAD data group used for the table of contents (see subclause 7.4.5.1).

### 7.4.4 Contents indicator

The format of the contents indicator depends on whether short X-PAD or variable size X-PAD is used, as described in the following subclauses.

#### 7.4.4.1 Contents indicator in short X-PAD

The contents indicator shall specify an application type from the first or second part of the application type table (255 application types). It is encoded as an unsigned binary number, in a single byte.

The third part of the application type table cannot be addressed, and hence these application types cannot be used for short X-PAD.

#### 7.4.4.2 Contents indicator in variable size X-PAD

The coding of the contents indicator in the case of variable size X-PAD is shown in figure 29.

| 3 bits | 5 bits | 8 bits |
|---|---|---|
| $b_7$ $b_5$ Length | $b_4$ $b_0$ AppTy | $b_7$ $b_0$ AppTy ext |

**Figure 29: Contents indicator for variable size X-PAD, shown in logical order. (The transmission order of the bytes is reversed)**

The following definitions apply:

**Length**: this 3-bit field shall indicate the length of the associated X-PAD data subfield (in the current audio frame), as follows:

| $b_7$ $b_5$ | |
|---|---|
| 0 0 0 : | 4 bytes; |
| 0 0 1 : | 6 bytes; |
| 0 1 0 : | 8 bytes; |
| 0 1 1 : | 12 bytes; |
| 1 0 0 : | 16 bytes; |
| 1 0 1 : | 24 bytes; |
| 1 1 0 : | 32 bytes; |
| 1 1 1 : | 48 bytes. |

**AppTy** (Application Type): this 5-bit field shall either specify an application type from the first part of the application type table, or contain the extension identifier. Application types shall be coded as unsigned binary numbers. The value "'11111" is the extension identifier indicating that the application type extension field is present and that an application type from the second or third part of the application type table is being addressed.

**AppTy ext** (Application Type extension): this 8-bit field shall be used to address application types from the second or third part of the application type table. This field, expressed as an unsigned binary number, shall signal the application type number minus 31. This field is present only when the application type field contains "11111".

### 7.4.5 Applications in X-PAD

This subclause contains the coding details for the X-PAD data groups of the standardized applications.

All the applications described shall use the same kind of Cyclic Redundancy Check for error detection. The CRC shall be generated according to the procedure defined in annex E. The generation shall be based on the polynomial,

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

The CRC word shall be complemented (1s complement) prior to transmission. At the beginning of each CRC word calculation, all register stages shall be initialized to "1'".

One or more padding bytes which shall be set to zero may be included after the CRC to fill up the X-PAD data subfield carrying the last part of the X-PAD data group.

#### 7.4.5.1 Table of contents

The table of contents provides a list of the applications carried in the X-PAD channel. The list should include application type numbers of applications that are provided at present, or will be provided in the near future.

Figure 30 shows the structure of the X-PAD data group for the Table of contents application.

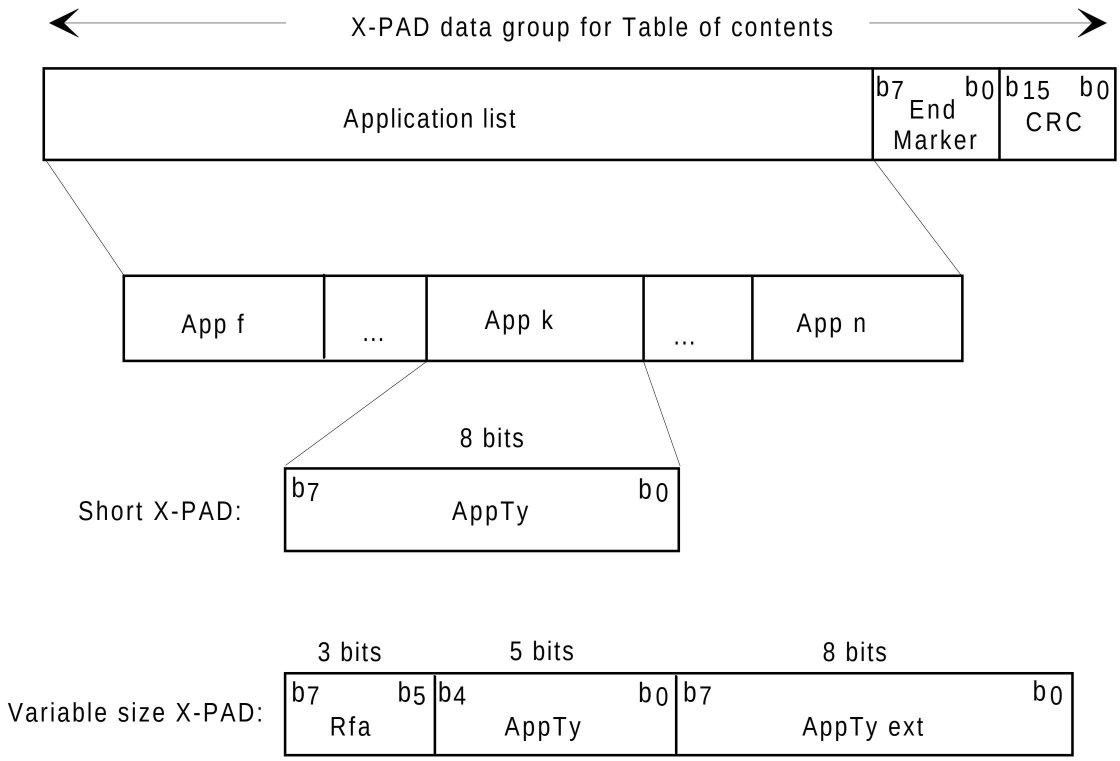


**Figure 30: Structure of the X-PAD data group for the Table of contents application**

The following definitions apply:

**Application list**: this list shall indicate all the applications contained in the table of contents.

**App** (Application): this field shall identify an application. The coding depends on whether short X-PAD or variable size X-PAD is used.

**Short X-PAD**:

> **AppTy** (Application Type): this 8-bit field shall specify an application type from the first or second part of the type table, coded as an unsigned binary number.

**Variable size X-PAD**:

> **Rfa**: this 3-bit field is reserved for future addition. All bits shall be set to zero until they are defined.
>
> **AppTy** (Application Type): see subclause 7.4.4.2.
>
> **AppTy ext** (Application Type extension): see subclause 7.4.4.2.

**End marker**: this 8-bit field shall be set to "0000 0000", see subclause 7.4.3.

**CRC** (Cyclic Redundancy Check): this CRC shall be calculated on the application list and end marker.

#### 7.4.5.2 Dynamic label segment

A dynamic label may comprise up to 8 segments, each consisting of up to 16 characters. Each segment is carried in one X-PAD data group. Figure 31 shows the structure of the X-PAD data group for the Dynamic Label segment.

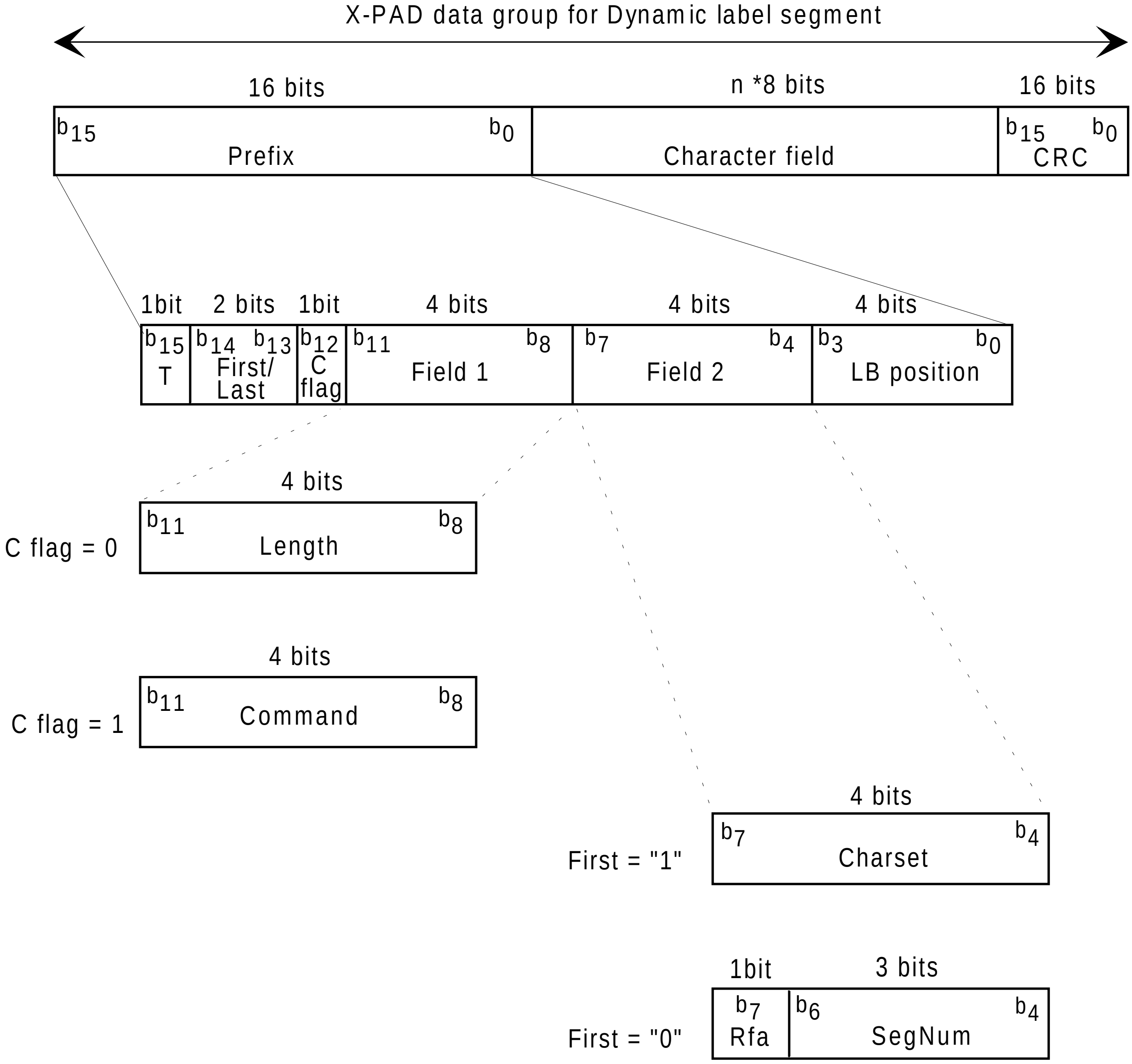


**Figure 31: Structure of the X-PAD data group for the Dynamic Label segment**

The following definitions apply:

**Prefix:**

> **T** (Toggle bit) **:** this bit shall be maintained in the same state as long as segments from the same label are being transmitted. When a segment from a different dynamic label is sent for the first time, this bit shall be inverted with respect to its previous state. If a dynamic label, which may consist of several segments, is repeated, then this bit shall remain unchanged.

**First/Last:** These flags shall be used to identify particular segments which form a succession of segments in a dynamic label. The flags shall be assigned as follows:

| **First flag $b_{14}$** | **Last flag $b_{13}$** | **The segment is:** |
|---|---|---|
| 0 | 0 | : an intermediate segment; |
| 0 | 1 | : the last segment; |
| 1 | 0 | : the first segment; |
| 1 | 1 | : the one and only segment. |

**C** (Command) **flag:** this 1-bit flag shall signal whether Field 1 contains the length of the label segment or a special command, as follows:

0: Field 1 signals the length of the label segment;
1: Field 1 contains a special command.

**Field 1:**

- **Length:** this 4-bit field, expressed as an unsigned binary number, shall specify the number of character codes in the character field minus 1.

- **Command:** this 4-bit field shall contain a special command, as follows (all other codes are reserved for future use):

$b_{11}$ $b_8$
0 0 0 1 : "the label shall be removed from the display".

**Field 2:** this 4-bit field shall contain either a character set identifier or a segment number, depending on the value of the First flag (b14).

**First flag = "1":**

**Charset:** see subclause 5.2.2.2.

**First flag = "0":**

**Rfa:** this 1-bit field is reserved for future additions. The bit shall be set to zero until it is defined.

**SegNum** (Segment number)**:** this 3-bit field, expressed as an unsigned binary number, shall specify the sequence number of the current segment minus 1. (The second segment of a label corresponds to SegNum=1, the third segment to SegNum=2, etc.) The value 0 is reserved for future use.

**LB** (Line-break) **position**: this 4-bit field, expressed as an unsigned binary number, shall indicate after which character position a label segment may be divided to suit a short display (the first character corresponds to position 1). The value "0000" shall be used to indicate that no information is signalled.

**Character field**: this field shall define the Dynamic label segment. It shall be coded as a string of characters (maximum 16) which are chosen from an 8-bit character set signalled by Charset field in the prefix. The first character code after the prefix shall correspond to the leftmost character on the display. This field shall be omitted when the C flag = "1" (special command).

**CRC** (Cyclic Redundancy Check): this CRC shall be calculated on the prefix and the character field.

#### 7.4.5.3 ITTS packets

The X-PAD data group shall consist of an ITTS packet, followed by a CRC word, calculated on all 48 bytes of the ITTS packet. For further details about ITTS, see annex F, bibliography.

#### 7.4.5.4 In-house information

The X-PAD data group contains in-house information. The coding of the in-house information is not subject to standardization.

#### 7.4.5.5 Closed user-group packet channel

The X-PAD data group shall consist of a closed user-group packet. The structure of a closed user-group packet shall conform to the description in subclauses 5.3.2 and 5.3.3.

#### 7.4.5.6 Closed user-group stream channel

The X-PAD data group shall contain a closed user group data stream.

# 8 Data features

This clause describes the coding of Service Information (SI) and non-audio data applications.

## 8.1 Service Information (SI)

### 8.1.1 Introduction

SI provides supplementary information about services, both audio programme and data. It does not include Multiplex Configuration Information (MCI) which is treated separately (see clause 6). The following subclauses describe the SI features. Service-related features include announcements, the service trigger and Frequency Information (FI). The language feature allows the language associated with a service component to be signalled. Programme-related features include Programme Number and programme type. The services, Programme Number (PNum), programme type, FI and the announcement features associated with other ensembles are signalled separately. Provision is made to signal the radio frequencies and announcements associated with FM services. Labels are provided for the ensemble and individual services. Also, there are features to give the time and country identifiers and to associate transmitter identification codes with geographical locations.

The coding of these features in the Fast Information Channel (FIC) is given by reference to their FIG type and extension (see subclause 5.2). Generally, the SI is encoded in FIG Types 0 and 1: some extensions are reserved for future use. Also, certain features may be transported in the Auxiliary Information Channel (within the MSC) using a re-direction signalling mechanism defined in subclause 8.1.12.

### 8.1.2 Service component language

The service component language feature is encoded in Extension 5 of FIG type 0 (FIG 0/5). Figure 32 shows the structure of the service component language field which is part of the Type 0 field (see also figure 4).

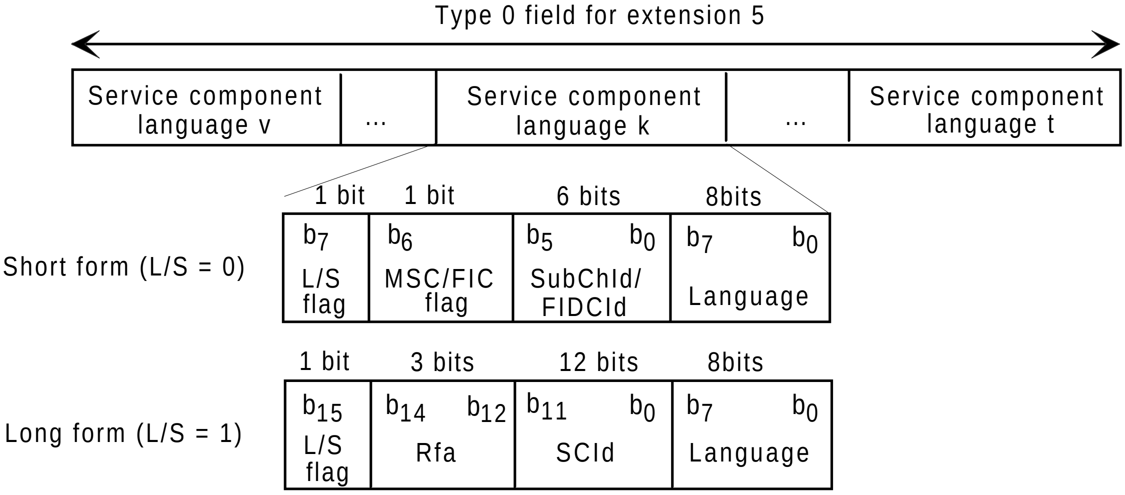


**Figure 32: Structure of the service component language field**

The following definitions apply:

**L/S flag**: this 1-bit flag shall indicate whether the service component identifier takes the short or the long form, as follows:

0 : short form;
1 : long form.

**Short form:**

- **MSC/FIC flag**: this 1-bit flag shall indicate whether the component is carried in the MSC in Stream mode or in the FIC, and whether the following field specifies the SubChId or the FIDCId (see subclause 6.3.1), as follows:

  0 : MSC in Stream mode and SubChId identifies the sub-channel;
  1 : FIC and FIDCId identifies the component carried in the FIC.

- **SubChId** (Sub-channel Identifier): this 6-bit field shall identify the sub-channel in which the service component is carried;

- **FIDCId** (Fast Information Data Channel Identifier): this 6-bit flag shall identify the service component carried in the FIDC (see subclause 6.3.1);

- **Language**: this 8-bit field shall define the service component language, see ETS 300 250 [7].

**Long form:**

- **Rfa**: this 3-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined;

- **SCId**: this 12-bit field shall identify the service component;

- **Language**: see above.

### 8.1.3 Time and country identifier

The Extension 10 of FIG type 0 (FIG 0/10) defines the date, time, local time offset, extended country code and the International Table. Figure 33 shows the structure of the time and country identifier field which is part of the Type 0 field (see also figure 4). The time reference shall be defined by the synchronization channel (see subclause 14.3.3).

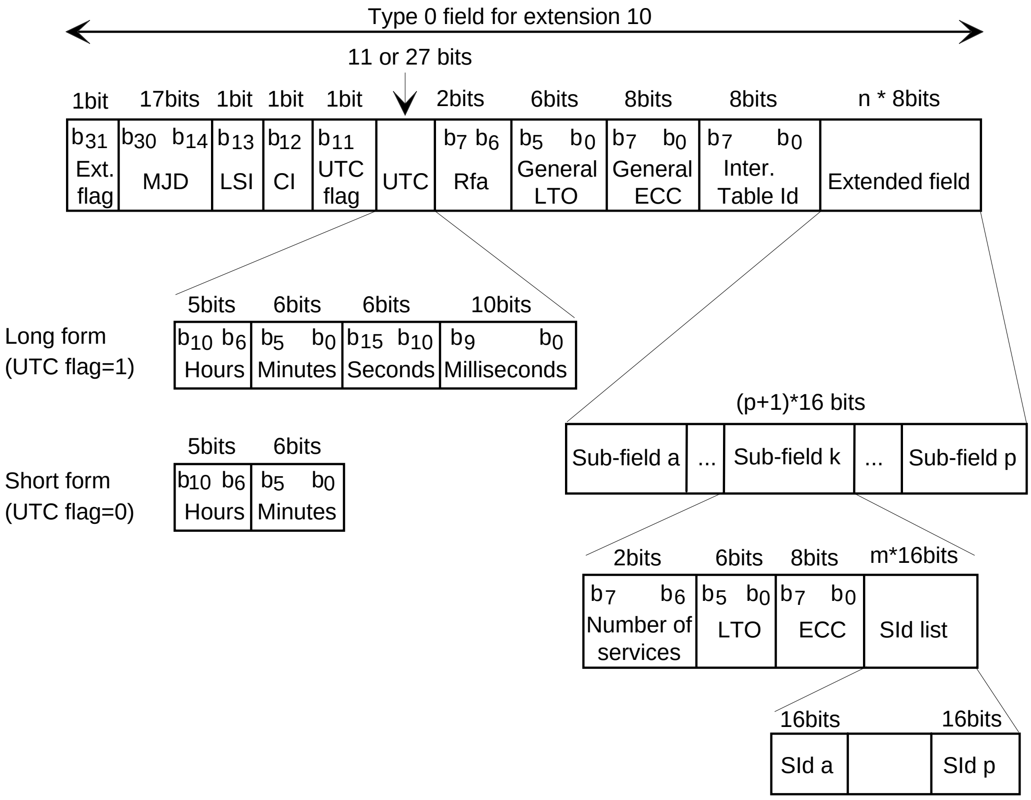


**Figure 33: Structure of the time and country identifier field**

The following definitions apply:

**Ext.** (Extension) **flag**: this 1-bit field shall indicate whether the optional extended field is present or not, as follows:

0 : no extended field;
1 : extended field present.

**MJD** (Modified Julian Date): this 17-bit binary number shall define the current date according to the Modified Julian coding strategy (EN 50067 [5]). This number increments daily at 0000 Co-ordinated Universal Time (UTC) and extends over the range 0-99 999. As an example, MJD 50 000 corresponds to 1995 October 10.

**LSI** (Leap Second Indicator): this 1-bit flag shall be set to "1" throughout a UTC day containing a leap second.

**CI** (Confidence Indicator): this 1-bit field shall be set to "1" when the timing information is within an agreed tolerance (for example, ± 0,2 ms at first ground fall).

**UTC flag**: this 1-bit field shall indicate whether the UTC (see below) takes the short form or the long form, as follows:

0 : UTC short form;
1 : UTC long form.

**UTC** (Co-ordinated Universal Time): two forms are available depending upon the state of the UTC flag. They shall be defined as follows:

- **short form**: this 11-bit field contains two sub-fields, which are both coded as unsigned binary numbers. The first sub-field is a 5-bit field which shall define the hours and the other sub-field is a 6-bit field which shall define the minutes;

- **long form**: in addition to the hours and minutes fields defined in the short form, this 27-bit field shall contain two further sub-fields, both of which shall be coded as unsigned binary numbers. The first is a 6-bit field which shall define the seconds and the other is a 10-bit field which shall define the milliseconds.

**Rfa**: this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**General LTO** (Local Times Offset): this 6-bit field shall give the Local Times Offset (LTO) for the majority of services contained in the ensemble. It is expressed in multiples of half hours in the range -12 hours to +12 hours. Bit $b_5$ shall give the sense of the LTO, as follows:

0 : positive offset;
1 : negative offset.

**General ECC** (Extended Country Code): this 8-bit field shall indicate which geographical area is associated with the majority of services in the ensemble. The ECC shall be defined as in EN 50067 [5], table D1, page 67.

**Inter** (International) **Table Id**: this 8-bit field shall be used to select an international table. The following international table identifiers are defined (other table identifiers shall be reserved for future use):

$b_7$ $b_0$
00000001 : EBU table of PTY (see EN 50067 [5]) and table of announcement types (see note);
00000010 : North American (US) table (see annex F, bibliography).

NOTE: The table of announcement types is under consideration by the EBU.

**Extended field**: this n*8-bit field shall contain one or more sub-fields, which define those services for which either their associated LTO or ECC, or both, differ from that generally defined for other services within the ensemble. The following definitions apply to each sub-field:

- **Number of services**: this 2-bit field, expressed as an unsigned binary number, shall indicate the number of services (in the range 1 to 3) for which the following LTO and ECC parameters differ from the general cases defined above;

- **LTO** (Local Times Offset): this 6-bit field shall be coded in the same way as the general LTO;

- **ECC** (Extended Country Code): this 8-bit field shall be coded in the same way as the general ECC;

- **SId** (Service Identification) **list**: this field of m SIds shall identify the services.

### 8.1.4 Programme Number

The Programme Number (PNum) feature is encoded in Extension 16 of FIG type 0 (FIG 0/16). Figure 34 shows the structure of the Programme Number field which is part of the Type 0 field (see also figure 4).

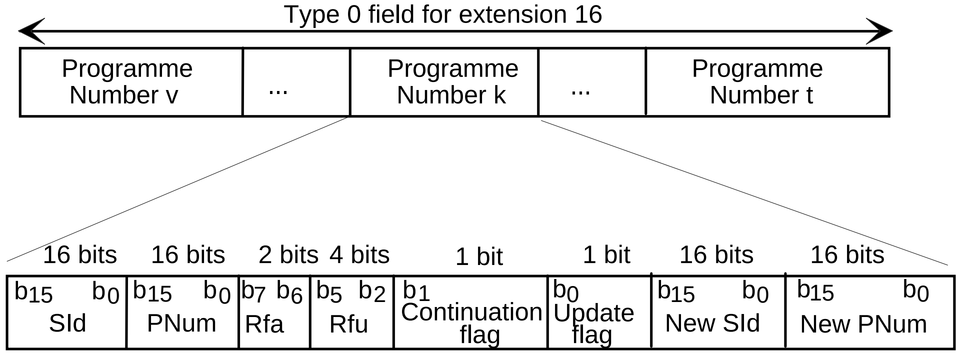


**Figure 34: Structure of Programme Number field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**PNum** (Programme Number): this 16-bit field shall define the date and time at which a programme begins. This field is coded in the same way as the RDS "Programme Item Number (PIN)" feature (EN 50067 [5]).

**Rfa**: this 2-bit field shall be reserved for future additions. The bits shall be set to "0" until they are defined.

**Rfu**: this 4-bit field shall be reserved for future use of the associated New SId and New PNum fields. The Rfu bits shall be set to zero for the currently specified definition of these fields.

**Continuation flag**: this 1-bit flag shall indicate that there will be a planned interruption to the programme , but the programme will be continued later. It is coded as follows:

0 : the programme will not be subject to a planned interruption;
1 : the programme will be interrupted but continued later.

**Update flag**: this 1-bit flag shall indicate a re-direction to a different service and time, as follows:

0 : no re-direction;
1 : re-direction.

- In the case of a re-direction, the New SId and New PNum shall be appended:

**New SId**: this 16-bit field shall contain the SId of the target service of the re-direction.

**New PNum**: this 16-bit field shall specify a new time of emission or a time at which the programme will be continued. It shall be coded in the same way as PNum.

NOTE: Special codes are allowed when the date part of the PNum field signals date="0". In this case, the hours and minutes part of the field shall contain a special code, as follows:

| Date | Hours | Minutes | Code description |
|---|---|---|---|
| 0 | 0 | 0 | Status code: no meaningful PNum is currently provided. |
| 0 | 0 | 1 | Blank code: the current programme is not worth recording. |
| 0 | 0 | 2 | Interrupt code: the interrupt is unplanned (for example a traffic announcement). |

### 8.1.5 Programme Type

The Programme Type (PTy) feature allows programme contents to be categorised according to their intended audience. There are two levels of categorisation which are referred to as "coarse" codes and "fine" codes. All codes belong to either an international or national set. An international set of 32 fixed codes (in Europe, this includes the EBU set used for RDS, see EN 50067 [5]) constitutes half the complement of coarse codes. A further 32 coarse codes are downloadable and may be chosen at an international or national level. Fine codes are chosen from a further independent set. There can be a maximum of 256 fine codes which are downloadable and may be chosen at an international or national level.

Each fine code has an associated coarse code which can allow a receiver to default to searching at a coarser level if no fine code can be found. However, to preserve compatibility with receivers responding to coarse codes only, the corresponding coarse code should be broadcast.

#### 8.1.5.1 Programme Type coding

The Programme Type feature is encoded in Extension 17 of FIG type 0 (FIG 0/17). It consists of at least one code from the international set of fixed coarse codes followed by zero or one additional coarse codes and zero, one or two fine codes. The Programme Type classification applies to the Primary service component. All the information applying to a service shall be contained within one field (programme type k) and carried in a single FIG. Figure 35 shows the structure of Programme Type field which is part of the Type 0 field (see also figure 4).

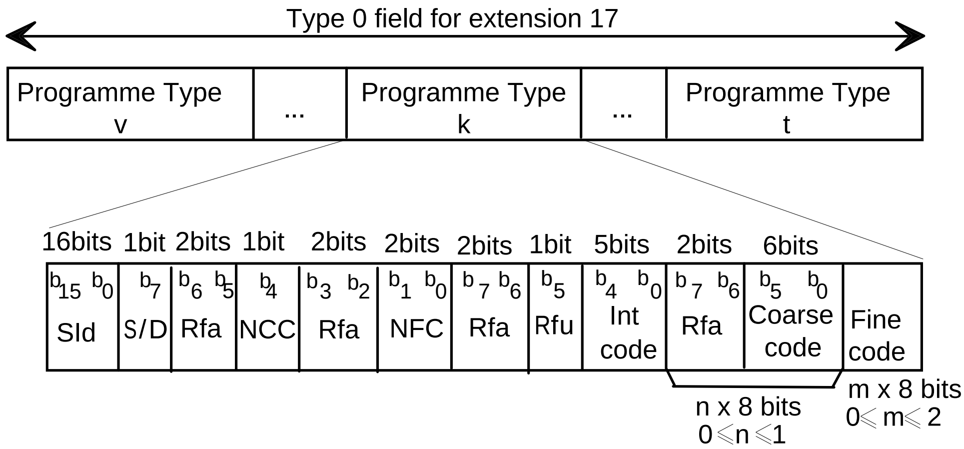


**Figure 35: Structure of the Programme Type field**

The following definitions apply:

**SId:** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**S/D:** (Static/Dynamic): this 1-bit flag shall indicate that the Programme Type codes, signalled in this programme field, represent the current programme contents, as follows:

0 : Programme Type codes may not represent the current programme contents;
1 : Programme Type codes represent the current programme contents.

**Rfa:** this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**NCC:** (Number of Coarse Codes): this 1-bit field, expressed as an unsigned binary number, shall specify the number of coarse codes in the range 0 to 1.

**Rfa:** this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**NFC** (Number of Fine Codes): this 2-bit field, expressed as an unsigned binary number, shall specify the number of fine codes in the range 0 to 2.

**Rfa:** this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Rfu:** this 1-bit field shall be reserved for extending the international code field to the full coarse code range. The bit shall bit set to "0" for the currently specified international code field.

**International code:** this 5-bit field shall specify the basic Programme Type (PTy) category. This code is chosen from an international table (see subclause 8.1.3).

**Rfa:** this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Coarse code:** this 6-bit field shall specify the coarse PTy code which should be chosen from an international or national table.

**Fine code:** this 8-bit field shall specify the fine PTy code.

### 8.1.5.2 Programme Type downloading

The Extension 2 of FIG type 1 (FIG 1/2) is used to dynamically define the meaning of a PTy coarse code or a PTy fine code. It is also used to establish a link between a coarse code and a fine code. Several fine codes may be linked with the same coarse code but each fine code can be linked only with one coarse code. Figure 36 shows the structure of the PTy downloading field which is part of the Type 1 field (see also figure 5).

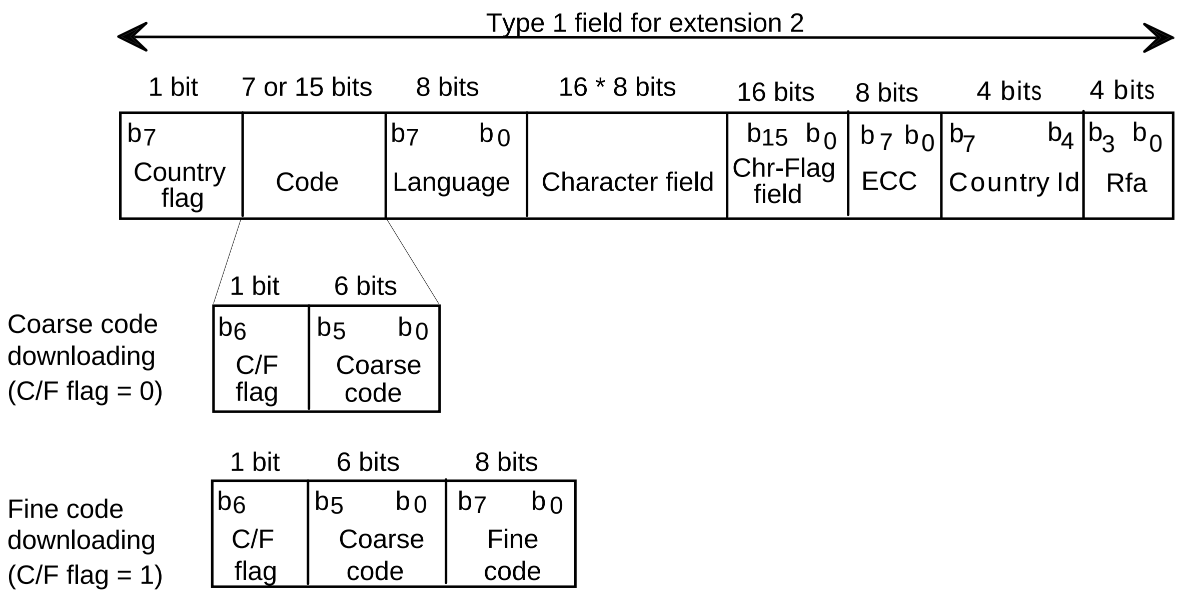


**Figure 36: Structure of the Programme Type downloading field**

The following definitions apply:

**Country flag:** this 1-bit flag shall indicate whether the last two bytes of the Programme Type downloading field (ECC, Country Id and Rfa) are present, or not, as follows:

0 : last two bytes absent;
1 : last two bytes present.

**Code:**

- **C/F flag** (Coarse/Fine flag): this 1-bit flag shall indicate whether a coarse code or a fine code is defined, as follows:

  0 : coarse code defined in the rest of the field;
  1 : fine code defined in rest of field.

Coarse code downloading:

- **Coarse code:** this 6-bit field shall give the number of the Coarse Programme Type being defined;

Fine code downloading:

- **Coarse code:** this 6-bit field shall give the number of the Coarse Programme Type to which the Fine Programme Type being defined is linked;

- **Fine code:** this 8-bit field shall give the number of the Fine Programme Type being defined.

**Language:** this 8 bit-field shall indicate the language of the Programme Type label. It shall be coded according to ETS 300 250 [7].

**Character field:** this 16-byte field shall define the Programme Type label. It shall be coded as a string of 16 characters, which are chosen from an 8-bit character set signalled by the Charset field in the first byte of the FIG type 1 data field (see subclause 5.2.2.2).

**Chr** (Character)**-Flag-field:** this 16-bit flag field shall indicate which of the 16 characters of the Character Field are to be displayed in an abbreviated form of the label, as follows:

$b_i$ (i = 1, ... , 16);
0 : not to be displayed in abbreviated label;
1 : to be displayed in abbreviated label.

NOTE: Not more than 8 of the $b_i$ may be set to "1".

**ECC** (Extended Country Code): this 8-bit field shall be used in combination with the Country Id field to identify a geographical area over which the Programme Type definition is valid. The ECC coding shall be as defined in EN 50067 [5] (see also subclause 8.1.3).

**Country Id:** this 4-bit field is defined in subclause 6.3.1.

**Rfa:** this 4-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

### 8.1.5.3 Programme Type preview

The extension 12 of FIG type 0 (FIG 0/12) is used to provide a preview of Programme Type codes of programmes which are planned to be broadcast in the future. Figure 37 shows the structure of the Programme Type preview field which is part of the Type 0 field (see also figure 4).

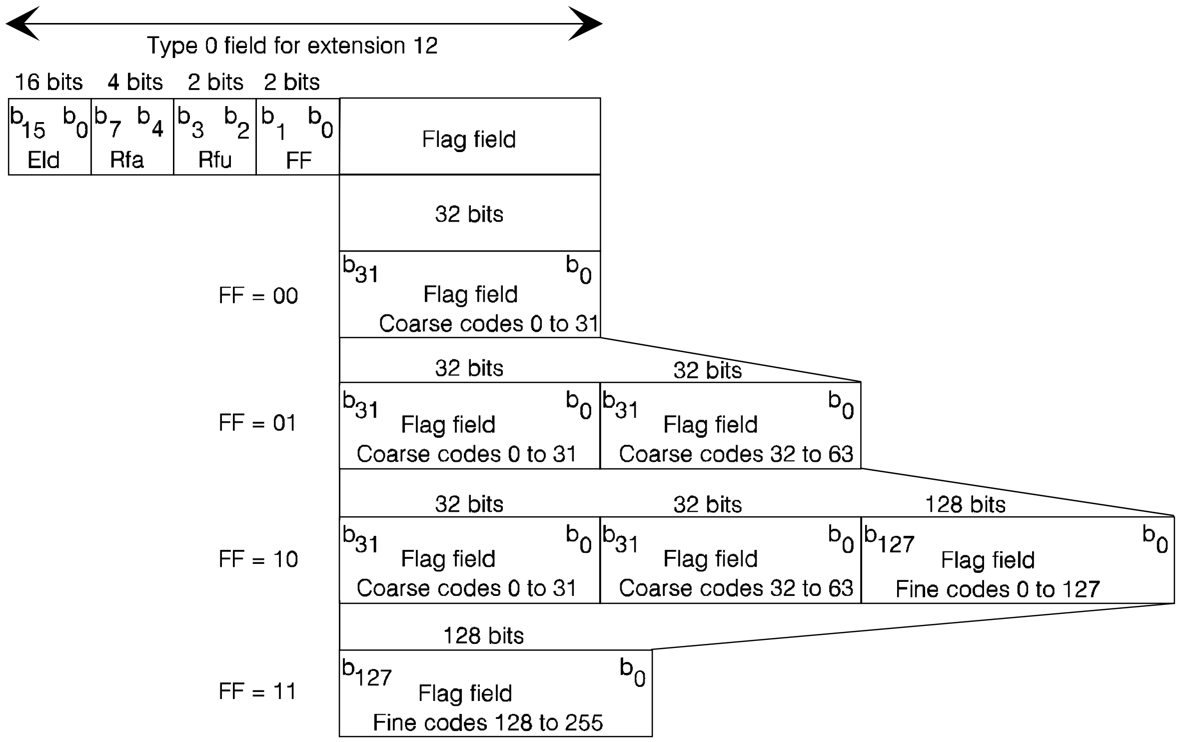


**Figure 37: Structure of the Programme Type preview field**

The following definitions apply:

**EId:** (Ensemble Identifier): this 16-bit field shall identify the ensemble. This field is absent when the OE flag is set to "0" (see subclause 5.2.2.1).

**Rfa:** this 4-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Rfu:** this 2-bit field shall be reserved for future use of the Flagfield Format (FF) and flag field.

**FF:** (Flagfield Format): this 2-bit field shall identify the contents of the flag field, as follows (see figure 37):

00 : Coarse codes 0 to 31;
01 : Coarse codes 0 to 31 and coarse codes 32 to 63;
10 : Coarse codes 0 to 31 and coarse codes 32 to 63 and fine codes 0 to 127;
11 : Fine codes 128 to 255.

**Flag field:**

**Flag field coarse codes 0 to 31:** this 32-bit field shall identify the coarse codes intended to be used in the future, as follows:

$b_i$ : (i = 0, to 31);
0 : Coarse code i not planned to be broadcast;
1 : Coarse code i planned to be broadcast.

**Flag field coarse codes 32 to 63:** this 32-bit field shall identify the coarse codes intended to be used in the future, as follows:

$b_i$ : (i = 0, to 31);
0 : Coarse code (32 + i) not planned to be broadcast;
1 : Coarse code (32 + i) planned to be broadcast.

**Flag field fine codes 0 to 127:** this 128-bit field shall identify the fine codes intended to be used in the future, as follows:

$b_i$ : (i = 0, to 127);
0 : Fine code i not planned to be broadcast;
1 : Fine code i planned to be broadcast.

**Flag field fine codes 128 to 255:** this 128-bit field shall identify the fine codes intended to be used in the future, as follows:

$b_i$ : (i = 0, to 127);
0 : Fine code (128 + i) not planned to be broadcast;
1 : Fine code (128 + i) planned to be broadcast.

### 8.1.6 Announcements

The announcement feature is effected in two stages. The announcement support assigns, to a service, the types of announcements by which the service may be interrupted and the links to other services which share the same interruption privileges. This support information is relatively static. The announcement switching provides the dynamic signal to allow a vectored interruption of the service by another carrying an announcement.

#### 8.1.6.1 Announcement support

The announcement support feature is encoded in Extension 18 of FIG type 0 (FIG 0/18). Figure 38 shows the structure of announcement support field which is part of the Type 0 field (see also figure 4).

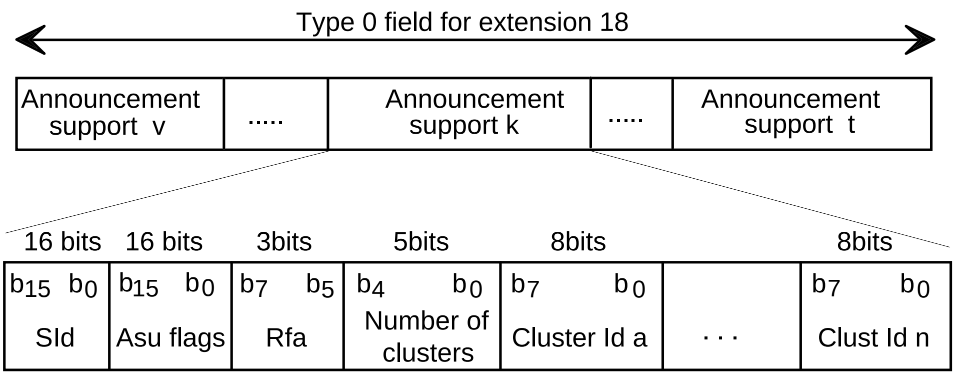


**Figure 38: Structure of announcement support field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**Asu** (Announcement support) **flags**: this 16-bit flag field shall specify the type(s) of announcements by which a service may be interrupted. The interpretation of the flags shall be as in table 26.

**Table 26: Allocation of announcement types**

| Bit flag | Announcement type |
|---|---|
| $b_0$ | Alarm |
| $b_1$ | Traffic |
| $b_2$ - $b_{15}$ | to be defined (see subclause 8.1.3) |

The flags shall be coded as follows:

$b_i$ : (i = 0, ... , 15);
0 : Announcement type not supported;
1 : Announcement type supported.

**Rfa**: this 3-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Number of clusters**: this 5-bit field shall specify the number of the following Cluster Ids.

**Cluster Id**: this 8-bit field shall identify the announcement cluster. From the service point of view the cluster identifies a group of services which share the same announcement interruption privileges. Several cluster Ids may be included in the announcement support field. Cluster Id = "0000 0000" and Cluster Id = "1111 1111" are pre-defined and shall not be signalled in the announcement support field (see subclause 8.1.6.2).

In the case of "Alarm" announcements, the cluster Id = "1111 1111" shall be used.

### 8.1.6.2 Announcement switching

The announcement switching description is encoded in Extension 19 of FIG type 0 (FIG 0/19). Figure 39 shows the structure of the announcement switching field which is part of the Type 0 field (see also figure 4).

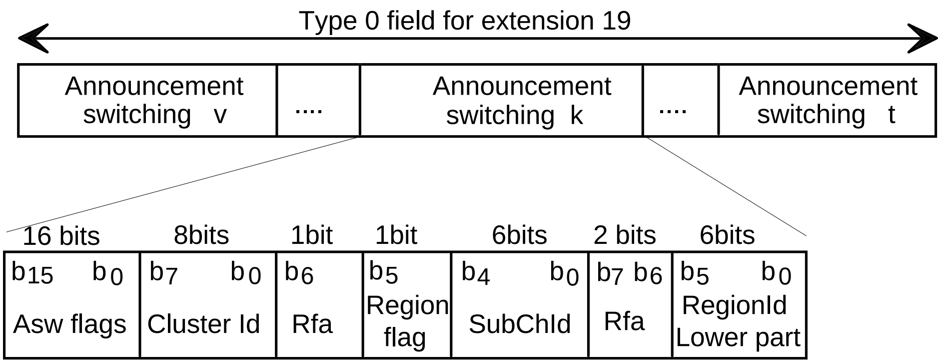


**Figure 39: Structure of announcement switching field**

The following definitions apply:

**Asw** (Announcement switching) **flags**: this 16-bit field shall specify the announcement types which apply to the announcement. The individual bits indicate whether or not a particular announcement type is signalled. The interpretation of the flags shall be as in table 26. The flags shall be coded as follows:

$b_i$ : (i = 0, ... , 15);
0 : Announcement type not valid;
1 : Announcement type valid.

**Cluster Id**: this 8-bit field shall identify the announcement cluster. Cluster Id = "0000 0000" shall be used for signalling an announcement which occurs within the same service. Cluster Id = "1111 1111" shall be used for Alarm announcements and shall be directed to all programme services carried in the ensemble. It shall not be used for any other announcement type.

**Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to "0" until it is defined.

**Region flag**: this 1-bit flag shall indicate whether a region is specified or not, as follows:

0 : last byte (the 2-bit Rfa field and the RegionId Lower part field) absent. The announcement concerns the whole service area;
1 : last byte (the 2-bit Rfa field and the RegionId Lower part field) present.

**SubChId**: this 6-bit field shall identify the Sub-channel which contains the audio service component carrying the announcement.

**Rfa**: this 2-bit field shall be reserved for future additions. These bits shall be set to zero until they are defined.

**RegionId Lower part**: this 6-bit field shall identify the region using the lower part of the Region Identifier. The upper part of the Region Identifier shall be set to "0" (see subclause 8.1.16).

### 8.1.7 Service trigger

The service Trigger feature is encoded in Extension 20 of FIG type 0 (FIG 0/20). It provides a general mechanism to signal that a new service will be emitted. Figure 40 shows the structure of the service trigger field which is part of the type 0 field (see also figure 4).

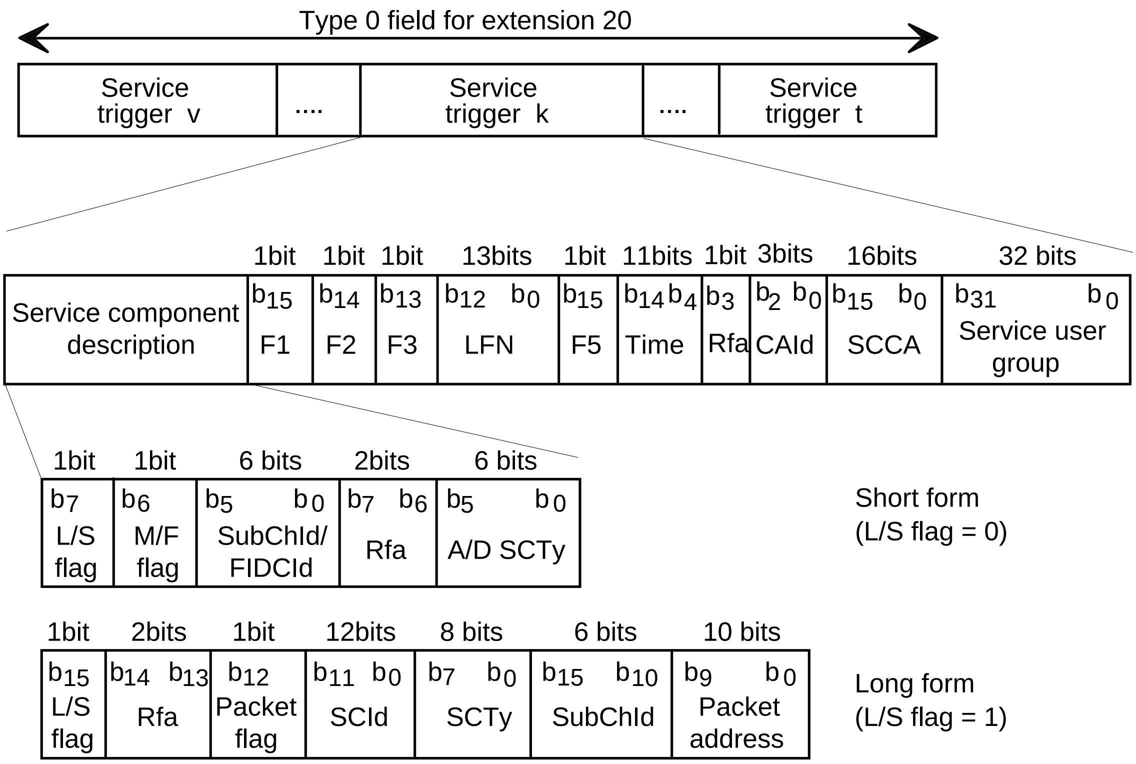


**Figure 40: Structure of the service trigger field**

The following definitions apply:

**Service component description**: this 16-, 24- or 40-bit field shall identify the component, as follows:

**L/S flag**: this 1-bit flag shall indicate whether the service component description takes the short form or the long form, as follows:

0 : short form;
1 : long form.

**Short form:**

- **M/F flag**: this 1-bit flag shall indicate whether the service component is carried in the MSC or in the FIC and whether the component is identified using the SubChId, as follows:

  0 : MSC and SubChId;
  1 : FIC and FIDCId.

- **SubChId** (Sub-channel Identifier): this 6-bit field shall identify the sub-channel in which the service component is carried (see also subclause 6.3.1);

- **FIDCId** (Fast Information Data Channel Identifier): this 6-bit field shall identify the service component carried in the FIDC (see also subclause 6.3.1);

- **Rfa**: this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined;

- **A/D SCTy** (Audio/Data Service Component Type): this 6-bit field shall indicate the service component, as defined in subclause 6.3.1.

**Long form:**

- **Rfa**: this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined;

- **Packet flag**: this 1-bit flag shall indicate whether the last three bytes of the long form field, containing the SCTy, SubChId and Address fields, are present, or not, as follows:

  0 : last three bytes of the long form field absent;
  1 : last three bytes of the long form field present.

  If the Packet flag is set to "0", the packet information shall be provided with a Data group type 0, Extension 3 (see subclause 6.3.2).

- **SCId** (Service Component Identifier): this 12-bit field shall uniquely identify the service component within the ensemble (see subclause 6.3.1);

- **SCTy** (Service Component Type): this 8-bit field shall indicate the type of the service component as defined in subclause 6.3.2;

- **SubChId** (Sub-channel Identifier): this 6-bit field shall identify the sub-channel in which the service component is carried (see also subclause 6.3.1);

- **Packet address**: this 10-bit field shall define the address of the packet in which the service component is carried.

**F1**: this 1-bit flag shall indicate whether the following F5 flag, time, CAId and SCCA fields are present, or not, as follows:

0 : F5 flag, time, CAId and SCCA fields absent;
1 : F5 flag, time, CAId and SCCA fields present.

**F2**: this 1-bit flag shall indicate whether the following time field and Logical Frame Number (LFN) applies to the beginning or the end of the service broadcast:

0 : beginning of the service broadcast;
1 : end of the service broadcast.

**F3**: this 1-bit flag shall indicate whether the following user group field is present, or not, as follows:

0 : user group field absent;
1 : user group field present.

**LFN** (Logical Frame Number): this 13-bit field, expressed as an unsigned binary number, shall contain the Logical frame count which indicates the number of the Logical frame from which the broadcast of the service component will start or end.

The value 1FFF (hex) shall be reserved to indicate that the LFN is not effective.

**F5**: this 1-bit flag shall indicate whether the time field, if present, is effective, as follows:

0 : Time field effective;
1 : Time field not effective.

**Time**: this 11-bit field shall indicate the time from which the emission of the service component will start or end. The time field is coded in the short form of the UTC as described in subclause 8.1.3.

**Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to zero until it is defined.

**CAId** (Conditional Access Identifier): this 3-bit field shall identify the Access Control System (ACS) used for the service. If no ACS is used for the service, CAId is set to "0" (see subclause 6.3.1).

**SCCA** (Service Component Conditional Access): this 16-bit field shall contain the descrambling parameters for accessing the service component (see subclause 9.2.2).

**Service user group**: this 32-bit field shall be used to identify the receivers to which the new service is directed.

### 8.1.8 Frequency Information

The Extension 21 of FIG type 0 (FIG 0/21) is assigned to providing radio Frequency Information (FI). This may apply to the whole ensemble, an individual service or a particular Primary service component, both within and outside the coverage area of the ensemble. In this case, the OE flag shall be set to "0" in the FIG type 0 sub-header (see subclause 5.2.2.1).

FIG 0/21 is also used to provide lists of frequencies of other ensembles and FM services which can be received in the coverage area of the ensemble. In this case, the OE flag shall be set to "1" in the FIG type 0 sub-header (see subclause 5.2.2.1), the "local" and "continuity" flags in the FI list header and the control field (see later in this subclause) are meaningless and set to "0". The R&M parameter (see later in this subclause) is restricted to the values "000" and "100".

Figure 41 shows the structure of the FI field which is part of the Type 0 field (see also figure 4).

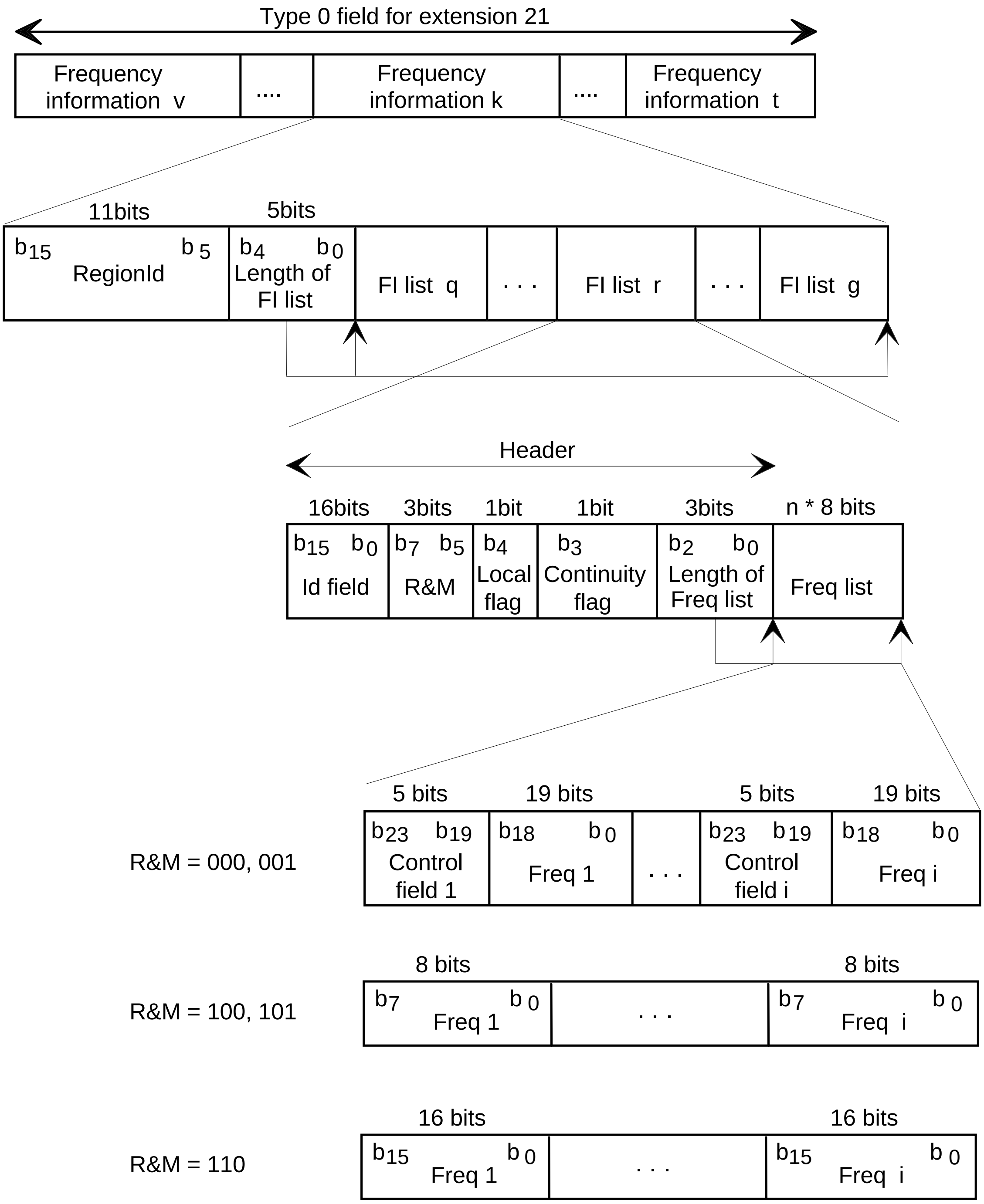


**Figure 41: Structure of the Frequency Information field**

The following definitions apply:

**RegionId**: this 11-bit field shall identify the region for which the frequency information applies (see subclause 8.1.16).

If the RegionId (see subclause 8.1.16) is "0000 0000 000", no area is specified and the FI shall be valid in the whole Ensemble area.

**Length of FI list**: this 5-bit field, expressed as an unsigned binary number, shall represent the length in bytes of the FI list field.

**FI** (Frequency Information) **list:**

- **Id field** (Identifier field): this 16-bit field shall depend on the following R&M field, as follows:

  If R&M = 000 Id field = EId (see subclause 6.4);

  If R&M = 001, 100, 101, 110 Id field = SId (see subclause 6.3.1).

- **R&M** (Range & Modulation): this 3-bit field shall define the range and modulation parameters which affect the structure of the alternative frequencies list. The entries marked "Rfu" shall be reserved for future use of the Freq list field. The coding is as follows:

  0 0 0 : DAB ensemble;
  0 0 1 : DAB service;
  0 1 0 : Rfu;
  0 1 1 : Rfu;
  1 0 0 : FM;
  1 0 1 : AM (MW & LW);
  1 1 0 : AM (SW);
  1 1 1 : Rfu.

- **Local flag**: this 1-bit flag shall indicate whether, or not, the alternative reception possibilities carry the same service when the service is split into two or more local services, as follows:

  0 : same service;
  1 : different service.

- **Continuity flag**: this 1-bit flag shall depend on the R&M field, as follows:

  If R&M = 000 or 001, the continuity flag shall signal that:

  0 : continuous output not expected;
  1 : continuous output possible.

  If R&M = 100, 101 or 110 the continuity flag shall indicate whether, or not, there is an appropriate time delay on the audio signal of an alternative service source on FM/AM to compensate the decoding time delay of DAB.

  0 : no compensating time delay on AM/FM audio signal;
  1 : compensating time delay on AM/FM audio signal.

- **Length of Freq list**: this 3-bit field, expressed as an unsigned binary number, shall represent the length in bytes of the following Freq list field;

- **Freq** (frequency) **list:**

  The structure of the frequencies list depends on R&M. In the following descriptions, the parameter Freq shall be treated as an unsigned binary number.

  **R&M = 000, 001:**

  - **Control field**: this 5-bit field shall be used to qualify the following Frequency (Freq). The following functions are defined (the remainder shall be reserved for future use of the Freq field);

    $b_{23}$ - $b_{19}$
    0 0 0 0 0 : geographically adjacent area;
    0 0 0 0 1 : not geographically adjacent area.

- **Freq** (Frequency): this 19-bit field shall represent the frequency associated with the alternative service source or other service. The centre frequency of the alternative ensemble shall be given by Freq* 16 kHz.

**R&M = 100, 101:**

- R&M = 100 (FM): the carrier frequency of the FM transmission is given by:

  87,5 MHz + (Freq * 100 kHz);

- R&M = 101 (AM): the carrier frequency of the AM transmission is given by:

  144 kHz + (Freq * 9 kHz) if Freq < 16;

  387 kHz + (Freq * 9 kHz) if Freq $\geq$ 16.

**R&M = 110 (SW):**

- the carrier frequency of the SW transmission is given by: Freq * 5 kHz;
- this option may also be used for other frequency bands using a 5 kHz grid.

### 8.1.9 Transmitter Identification Information (TII)

The Extension 22 of FIG type 0 (FIG 0/22) provides the cross reference between the transmitter identifiers (see subclause 14.8) and the geographic locations and relative time delays of the transmitters. Transmitter identifiers are separated into two groups which are called Main Identifiers (MainId) and Sub-Identifiers (SubId) respectively. Sub-Identifiers 1 to 23 shall be used for terrestrial transmitters. Sub-Identifier 0 shall be used for satellite transmitters. Figure 42 shows the structure of the TII field which is part of the Type 0 field (see also figure 4).

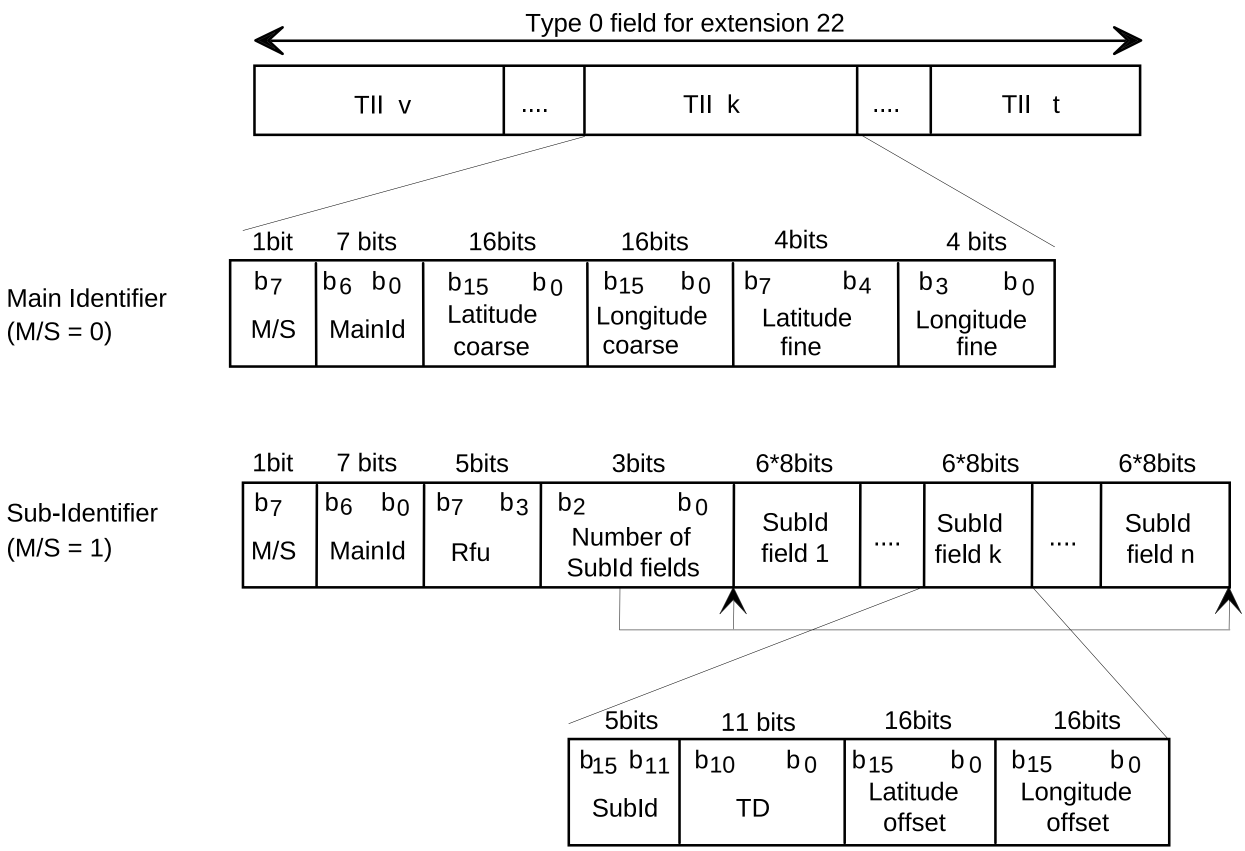


**Figure 42: Structure of the Transmitter Identification Information (TII) field**

The following definitions apply:

**M/S** (Main/Sub): this 1-bit flag shall indicate whether the remainder of the TII field refers to a Main Identifier or to a Sub-Identifier, as follows:

0 : Main identifier;
1 : Sub-identifier.

**Main Identifier (M/S = 0):**

- **MainId**: this 7-bit field, expressed as an unsigned binary number, shall specify the Main Identifier. The coding range shall be 0 to 69 for transmission modes I and II and 0 to 5 for transmission mode III;

- **Latitude coarse**: this 16-bit field, coded as a two's complement number, shall specify the coarse latitude. It shall be calculated by multiplying the value of the Latitude coarse field by ($90°/2^{15}$). Southern latitudes shall be considered negative and northern latitudes positive;

- **Longitude coarse**: this 16-bit field, coded as a two's complement number, shall specify the coarse latitude. It shall be calculated by multiplying the value of the Latitude coarse field by ($180°/2^{15}$). West of Greenwich shall be considered negative and east of Greenwich positive;

- **Latitude fine**: this 4-bit field shall specify the latitude offset. This unsigned binary number multiplied by ($90°/2^{19}$) shall be added to the coarse latitude to calculate the latitude with full precision;

- **Longitude fine**: this 4-bit field shall specify the longitude offset. This unsigned binary number multiplied by ($180°/2^{19}$) shall be added to the coarse longitude to calculate the longitude with full precision.

**Sub-Identifier (M/S = 1):**

- **MainId**: this 7-bit field, coded as an unsigned binary number, shall specify the Main Identifier;

- **Rfu**: this 5-bit field shall be reserved for future use of the field containing the list of SubId fields (as identified by the Number of SubId fields field). The Rfu bits shall be set to zero for the currently specified definition of this associated field;

- **Number of SubId fields**: this 3-bit field, coded as an unsigned binary number, shall give the total number of SubId fields following (48 bit each). The information corresponds to successive SubIds;

  **SubId field:**

  - **SubId**: this 5-bit field, coded as an unsigned binary number, shall specify the SubId to which the following data applies. The value zero shall not be used in this field and the coding range of this field is restricted to the values 1 to 23;

  - **TD** (Time Delay): this 11-bit field shall specify the time delay in microseconds. It is coded as an unsigned binary number in the range 0 to 2 047;

  - **Latitude offset**: this 16-bit field shall specify the latitude offset of the transmitter from the reference associated with the same MainId. It shall be coded as a two's complement number. The value of the latitude shall be calculated by adding or subtracting the value of the latitude offset field multiplied by ($90°/2^{19}$) to/from the latitude of the reference;

  - **Longitude offset**: this 16-bit field shall specify the longitude offset of the transmitter from the reference associated with the same MainId. It shall be coded as a two's complement number. The value of the longitude shall be calculated by adding or subtracting the value of the longitude offset field multiplied by ($180°/2^{19}$) to/from the latitude of the reference.

### 8.1.10 Other ensembles

The following subclauses describe the other ensembles features.

### 8.1.10.1 Frequencies

The other ensembles' frequencies are encoded in Extension 21 of FIG type 0 (see subclause 8.1.8). The OE flag (see subclause 5.2.2.1) shall be set to "1".

### 8.1.10.2 Services

The Extension 24 of FIG type 0 (FIG 0/24) is used to identify the services carried in other DAB ensembles. Figure 43 shows the structure of the other ensembles services field which is part of the Type 0 field (see also figure 4).

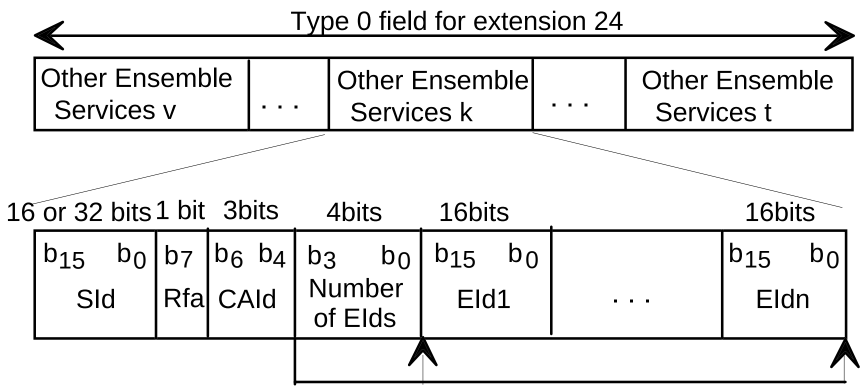


**Figure 43: Structure of the other ensembles services field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit or 32-bit field shall identify a service. The coding details are given in subclause 6.3.1.

**Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to zero until it is defined.

**CAId** (Conditional Access Identifier): this 3-bit field shall identify the Access Control System (ACS) used for the service. If no ACS is used for the service, CAId is set to "0" (see subclause 6.3.1).

**Number of EIds**: this 4-bit field shall indicate the number of EIds signalled in the following list.

**EId** (Ensemble identifier): this 16-bit code shall identify the ensemble. The coding details are given in subclause 6.4.

### 8.1.10.3 Programme Number

The other ensembles Programme Number (PNum) feature is encoded in Extension 16 of the FIG type 0 (FIG 0/16) as described in subclause 8.1.4. The OE flag (see subclause 5.2.2.1) shall be set to "1".

### 8.1.10.4 Programme Type

The other ensembles Programme Type feature is encoded in Extension 17 of the FIG type 0 (FIG 0/17) as described in subclause 8.1.5. The OE flag (see subclause 5.2.2) shall be set to "1".

### 8.1.10.5 Announcements

The other ensembles announcements feature is encoded in Extensions 25 and 26 of the FIG type 0. (FIG 0/25 and FIG 0/26). Extension 25 is used for the announcement support and Extension 26 is used for the announcement switching (see subclause 8.1.6).

#### 8.1.10.5.1 Announcement support

Figure 44 shows the structure of the other ensembles Announcement support field which is part of the Type 0 field (see also figure 4).

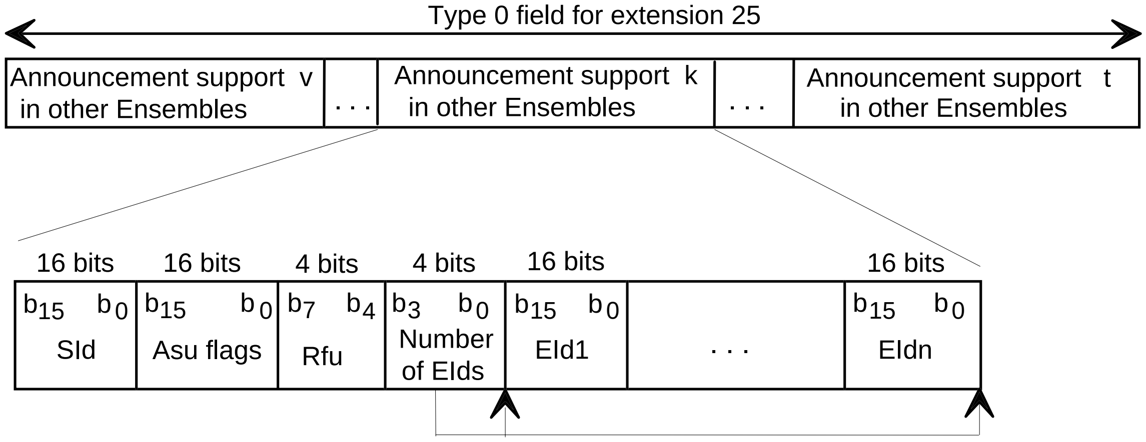


**Figure 44: Structure of other ensembles announcement support field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**Asu** (Announcement support) **flags:** this 16-bit field shall specify the type of announcements by which a service may be interrupted. The possible values are given in table 26. The coding of these values is given in subclause 8.1.6.

**Rfu**: this 4-bit field shall be reserved for future use of the associated field containing the list of EIds (as identified by the Number of EIds field). The Rfu bits shall be set to zero for the currently specified definition of this associated field.

**Number of EIds**: this 4-bit field shall indicate the number of EIds carried in the following list.

**EId** (Ensemble Identifier): this 16-bit field shall identify the other ensemble. The coding details are given in subclause 6.4.

### 8.1.10.5.2 Announcement switching

Figure 45 shows the structure of the other ensembles announcement switching field which is part of the Type 0 field (see also figure 4).

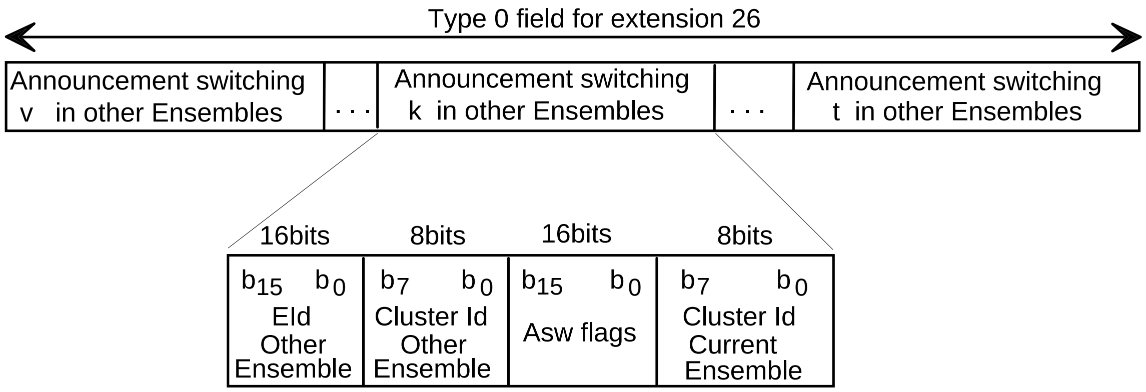


**Figure 45: Structure of announcement switching field**

The following definitions apply:

**EId Other Ensemble**: this 16-bit field shall identify the other ensemble. The coding details are given in subclause 6.4.

**Cluster Id Other Ensemble**: this 8-bit field shall identify the announcement cluster to which the announcement is directed in the other ensemble.

**Asw** (Announcement switching) **flags**: this 16-bit field shall specify the announcement types which apply to the announcement. The individual bits indicate whether or not a particular announcement type is signalled The possible values are given in table 26. The coding of these values is given in subclause 8.1.6.

**Cluster Id Current Ensemble**: this 8-bit field shall identify the announcement cluster, in the current ensemble, to which the announcement is relayed.

#### 8.1.10.6 Text labels

Ensemble and service labels associated with other ensembles are encoded in extensions 0 and 1 respectively of FIG type 1 (FIGs 1/0 and 1/1) as described in subclauses 8.1.13 and 8.1.14. In each case the OE flag (see subclause 5.2.2.2) shall be set to "1".

#### 8.1.10.7 Satellite database

The satellite database of other ensembles broadcast from a satellite network is encoded in extension 30 of FIG type 0 (FIG 0/30) as described in subclause 8.1.18.2. The OE flag (see subclause 5.2.2.1) shall be set to "1".

### 8.1.11 FM services

The radio frequencies of FM services may be signalled and announcements carried on FM services may be allowed to interrupt DAB services. These features are described in this subclause.

#### 8.1.11.1 Frequencies

The frequencies on which FM services can be found are encoded in Extension 21 of the FIG type 0 (FIG 0/21) as described in subclause 8.1.8. The OE flag (see subclause 5.2.2.1) shall be set to "1".

#### 8.1.11.2 Announcements

The FM services announcements feature is encoded in Extension 27 and 28 of the FIG type 0. (FIG 0/27 and FIG 0/28). Extension 27 is used for the announcement support and Extension 28 is used for the announcement switching (see subclause 8.1.6).

##### 8.1.11.2.1 Announcement support

Figure 46 shows the structure of the FM services announcement support field which is part of the Type 0 field (see also figure 4).

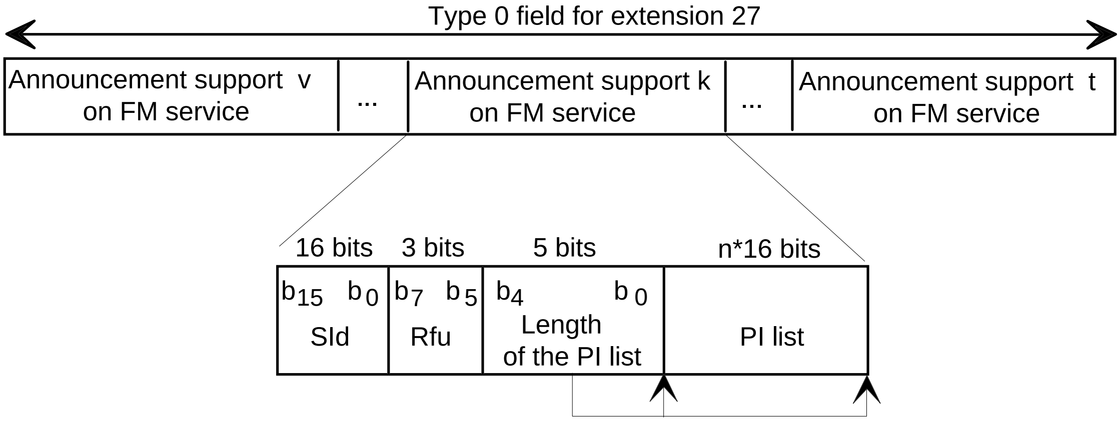


**Figure 46: Structure of the FM services announcement support field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**Rfu**: this 3-bit field shall be reserved for future use of the associated PI list. The Rfu bits shall be set to zero for the currently specified definition of this associated field.

**Length of the PI list**: this 5-bit field, expressed as an unsigned binary number, shall represent the length in bytes of the PI list.

**PI list:** this list contains n PI codes. Each 16-bit PI code field shall identify a service (programme identification code) carried in the FM channel (see EN 50067 [5]).

#### 8.1.11.2.2 Announcement switching

Figure 47 shows the structure of the FM services announcement switching field which is part of the Type 0 field (see also figure 4).

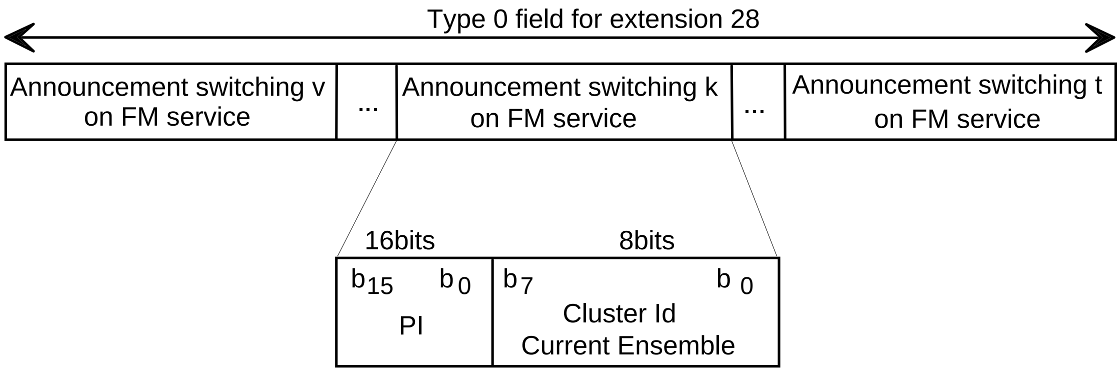


**Figure 47: Structure of the FM services announcement switching field**

The following definitions apply:

**PI:** this 16-bit field shall identify a service (programme identification code) carried in the FM channel (see EN 50067 [5]).

**Cluster Id Current Ensemble**: this 8-bit field shall identify the announcement cluster, in the current ensemble, to which the announcement is relayed.

### 8.1.12 FIC re-direction

The Extension 31 of FIG type 0 (FIG 0/31) is used to signal which data features, coded in FIG types 0 and 1, are carried in the Auxiliary Information Channel (AIC). Figure 48 shows the structure of the FIC overflow signalling field which is part of the Type 0 field (see also figure 4).

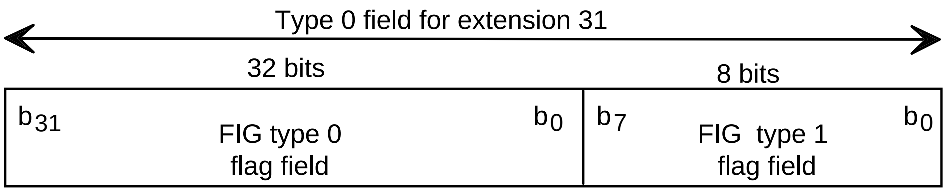


**Figure 48: FIC re-direction field**

The following definitions apply:

**FIG type 0 flag field**: this 32-bit field shall indicate which of the 32 Extensions of FIG type 0 are carried in the AIC, as follows:

$b_i$ : (i = 0, ... , 31);
0 : extension i is not carried in the AIC;
1 : extension i is carried in the AIC.

For the FIG type 0 flag field, $b_0$ to $b_5$, $b_{10}$, $b_{29}$ and $b_{31}$ shall be set to "0" because the corresponding information is always carried in the FIC.

**FIG type 1 flag field**: this 8-bit field shall indicate which of the 8 extensions of FIG type 1 are carried in the AIC, as follows:

$b_i$ : (i = 0, ... , 7);
0 : extension i is not carried in the AIC;
1 : extension i is carried in the AIC.

### 8.1.13 Ensemble label

The ensemble label feature is encoded in Extension 0 of the FIG type 1 (FIG 1/0). Figure 49 shows the structure of the ensembles label field which is part of the Type 1 field (see also figure 5). The OE flag (see subclause 5.2.2.2) shall be set to "0" for the current ensemble and to "1" for other ensembles.

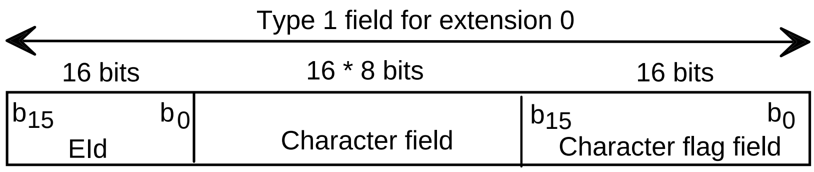


**Figure 49: Structure of the ensemble label field**

The following definitions apply:

**EId** (Ensemble Identifier): this 16-bit field shall identify the ensemble. The coding details are given in subclause 6.4.

**Character field**: this 16-byte field shall define the ensemble label. It shall be coded as a string of 16 characters which are chosen from an 8-bit character set signalled by the Charset field in the first byte of the FIG type 1 data field (see subclause 5.2.2.2).

**Character flag field**: this 16-bit field shall indicate which of the 16 characters of the character field are to be displayed in an abbreviated form of the label, as follows:

$b_i$ : (i = 0, ... , 15);
0 : not to be displayed in abbreviated form;
1 : to be displayed in abbreviated form.

NOTE: Not more than 8 of the $b_i$ may be set to "1".

### 8.1.14 Service label

The service label feature is encoded in Extension 1 of the FIG type 1 (FIG 1/1). Figure 50 shows the structure of the service label field which is part of the Type 1 field (see also figure 5). The OE flag (see subclause 5.2.2.2) shall be set to "0" for the services carried in the current ensemble and set to "1" for services carried in other ensembles.

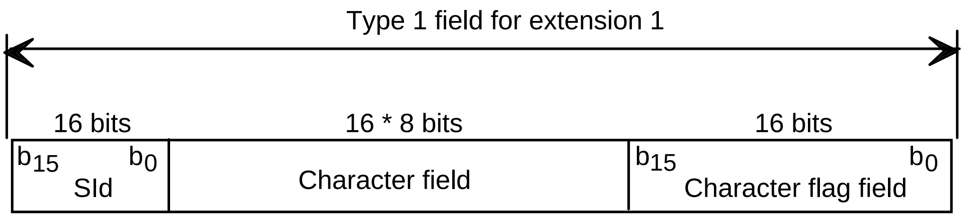


**Figure 50: Structure of the service label field**

The following definitions apply:

**SId** (Service Identifier): this 16-bit field shall identify the service (see subclause 6.3.1).

**Character field**: this 16-byte field shall define the service label. It shall be coded as a string of 16 characters which are chosen from an 8-bit character set signalled by Charset field in the first byte of the FIG type 1 data field (see subclause 5.2.2.2).

**Character flag field**: this 16-bit field shall indicate which of the 16 characters of the character field are to be displayed in an abbreviated form of the label, as follows:

$b_i$ : (i = 0, ... , 15);
0 : not to be displayed in abbreviated form;
1 : to be displayed in abbreviated form.

NOTE: Not more than 8 of the $b_i$ may be set to "1".

### 8.1.15 Service linking information

The extension 6 of FIG type 0 (FIG 0/6) provides service linking information for use when services carry the same programme. Figure 51 shows the structure of the service linking field which is part of the Type 0 field (see also figure 4).

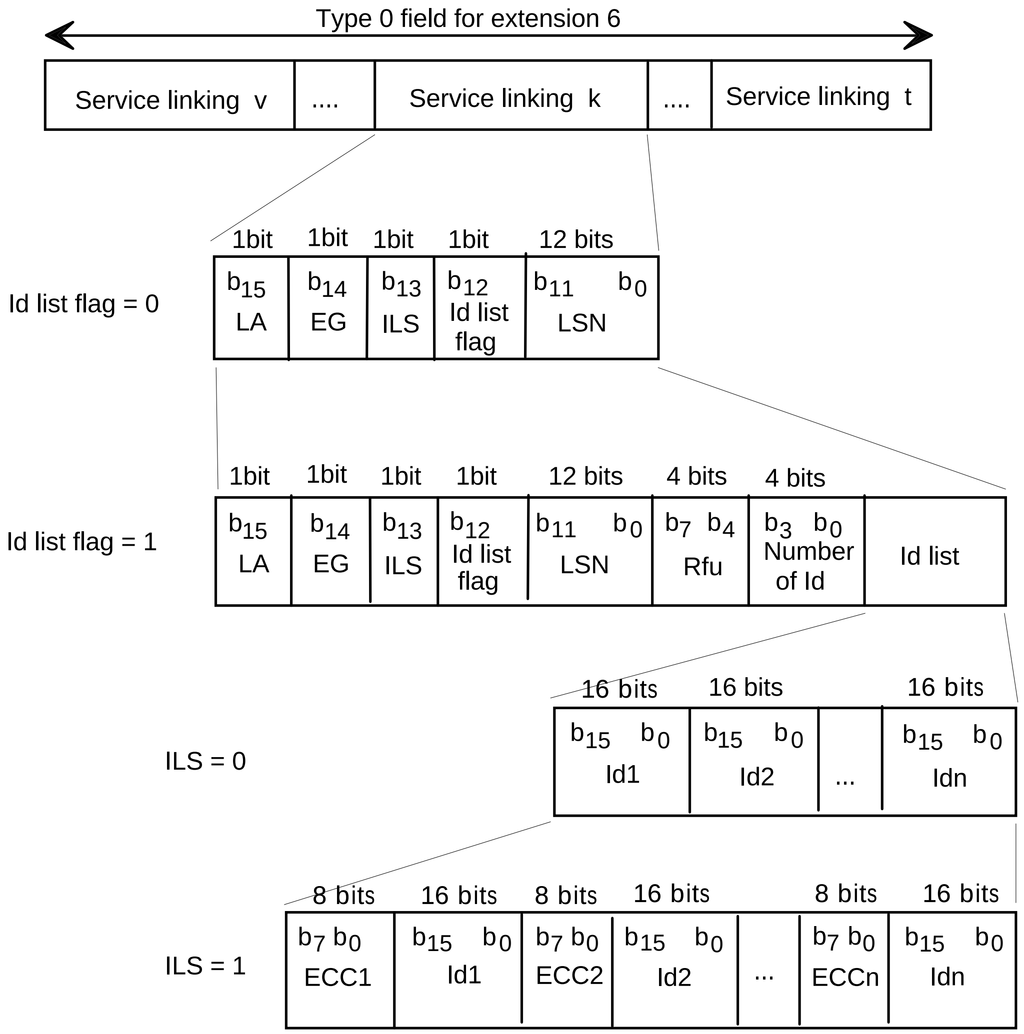


**Figure 51: Structure of the Service linking field**

The following definitions apply:

**LA** (Linkage Actuator): this 1-bit flag shall indicate whether the link is active or inactive, as follows:

0 : potential future link or deactivated link;
1 : active link.

**EG** (Extended Generic indicator): this 1-bit flag shall indicate whether the set of linked services includes generic sets or not, as follows:

0 : no generic set(s);
1 : generic set.

A generic set of services is characterised by services having SIds which are identical except for the Service type (see subclause 6.3.1). Generic Services have a service type in the range "0100" to "1111".

**ILS** (International linkage Set indicator): this 1-bit flag shall indicate whether the link is national or international, as follows:

0 : national link;
1 : international link.

**Id list flag**: this 1-bit flag shall indicate whether the list of service identifiers together with the Rfu field and the Number of Id field, is present or not, as follows:

0 : no Id list, no Rfu field, no Number of Id field;

1 : Id list, Rfu field and Number of Id field present.

**LSN** (Linkage set number): this 12-bit flag represents a number which shall be common to all Services linked together as a set. It is unique to the set of linked Services and shall be agreed by appropriate national or international bodies.

**Rfu**: this 4-bit field shall be reserved for future use of the field containing the Id list. The Rfu bits shall be set to zero for the currently specified definition of this associated field;

**Number of Id**: this 4-bit field shall specify the number of Service identifiers (maximum 12 when ILS = 0, maximum 8 when ILS = 1), expressed as an unsigned binary number, in the Id list.

**Id list:**

- **Id** (Identifier of Service): this 16-bit field shall identify the service by means of the SId (see subclause 6.3.1). For generically linked Services, one of the SIds from the generic set shall be chosen.

- **ECC** (Extended country code): this 8-bit field shall identify the country to which the Service belongs (see subclause 8.1.3).

### 8.1.16 Regional identification

#### 8.1.16.1 Region definition

The extension 11 of FIG type 0 (FIG 0/11) defines the geographical area by providing the cross reference between a Region Identifier and a TII list or geographical co-ordinates. The geographical area shall always be defined in terms of a TII list (GATy = "0000"), regardless of whatever other means of definition is also signalled (GATy ≠ "0000"). Figure 52 shows the structure of the Regional identification field which is part of the Type 0 field (see also figure 4).

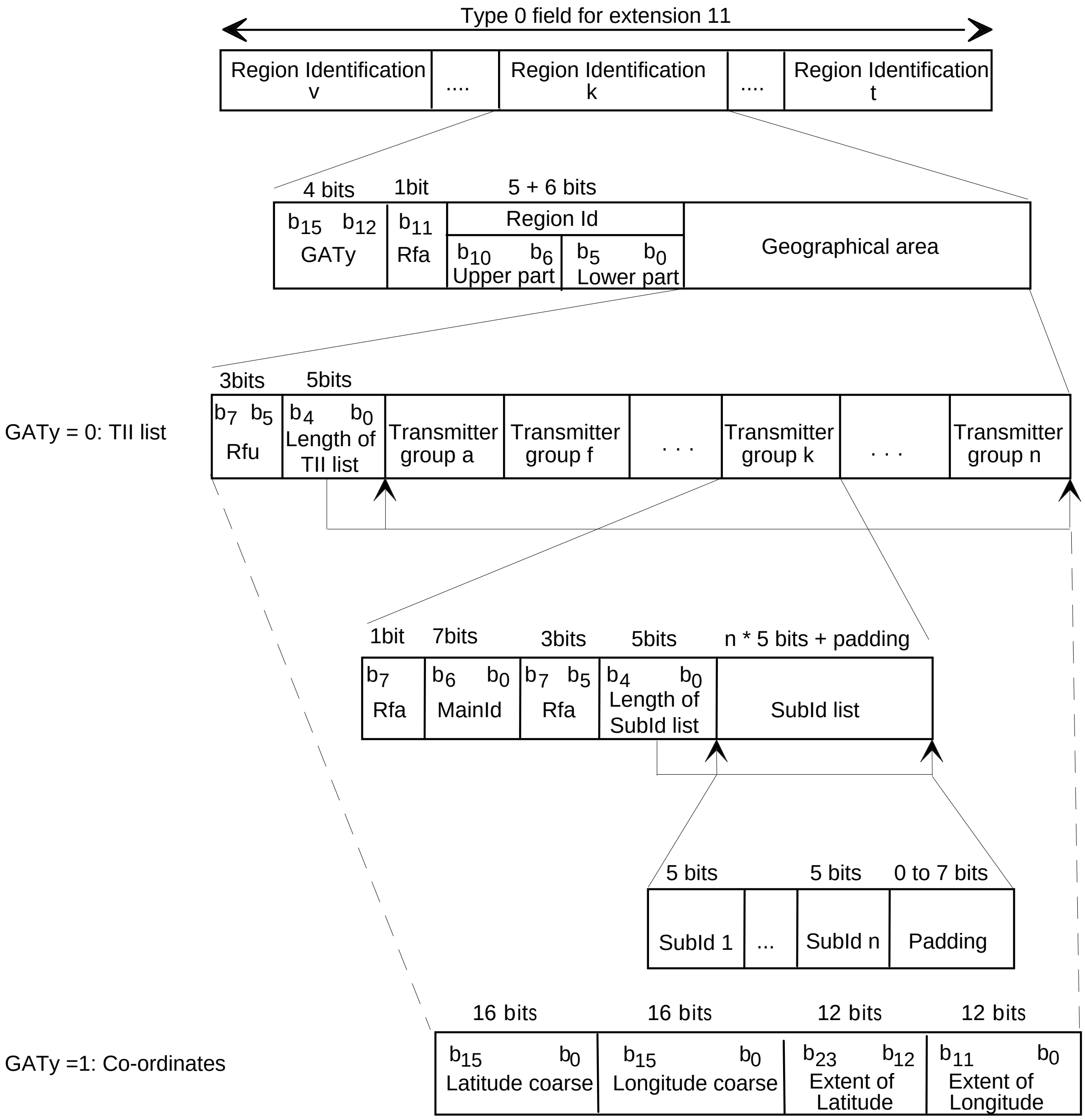


**Figure 52: Structure of regional identification information**

The following definitions apply:

**GATy:** this 4-bit field shall indicate the coding method for the Geographical area. The following values of the GATy field are defined (the remaining values are reserved for future use of the Geographical area field):

| $b_{15}$ $b_{12}$ | |
|---|---|
| 0 0 0 0 : | Geographical area defined by an TII list; |
| 0 0 0 1 : | Geographical area defined as a spherical rectangle by the geographical co-ordinates of one corner and its latitude and longitude extents. |

**Rfa:** this 1-bit field shall be reserved for future additions. The bit shall be set to "0" until it is defined.

**RegionId:** this 11-bit field, organised as a 5-bit **upper part** and a 6-bit **lower part**, shall identify the region. The value RegionId = "0" is reserved (see subclause 8.1.8).

**Geographical area:** this field defines the regional area in terms of a TII list or a rectangle based on co-ordinates.

**TII list**: this field defines the geographical area in terms of the service area of all transmitters identified in the list:

- **Rfu**: this 3-bit field shall be reserved for future use of the associated geographical area field excluding the first byte. The Rfu bit shall be set to "0" for the currently specified definition of this associated field;

- **Length of TII list**: this 5-bit field, expressed as an unsigned binary number, shall represent the length in bytes of the TII list field, excluding the byte in which this length parameter is carried.

The definitions below apply to one transmitter group sharing a main identifier:

- **Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to "0" until it is defined;

- **MainId** (Identifier): this 7-bit field, expressed as an unsigned binary number, shall identify a group of transmitters (see subclause 8.1.9);

- **Rfa**: this 3-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined;

- **Length of SubId list**: this 5-bit field, expressed as an unsigned binary number, shall represent the length in bytes of the SubId list field. The length 0 shall be reserved for future use;

- **SubId list**: this list shall contain a number of SubIds. Padding bits (0 to 7) shall be added for byte alignment;

- **SubId**: this 5-bit field, expressed as an unsigned binary number, shall signal a Sub-identifier associated with a transmitter (see subclause 8.1.9);

- **Padding**: this field shall contain sufficient bits in the range 1 ... 7 to make up the overall length of each SubId list field to an integral number of bytes. Padding bits shall be set to "0".

**Co-ordinates:** this field defines the geographical area in terms of a rectangle based on co-ordinates.

- **Latitude coarse:** this 16-bit field, coded as a two's complement number, shall specify the coarse latitude of a corner of the spherical rectangle that defines the area. It shall be calculated by multiplying the value of the Latitude coarse field by ($90°/2^{15}$). Southern latitudes shall be considered negative and northern latitudes positive;

- **Longitude coarse:** this 16-bit field, coded as a two's complement number, shall specify the coarse longitude of a corner of the spherical rectangle that defines the area. It shall be calculated by multiplying the value of the Longitude coarse field by ($180°/2^{15}$). West of Greenwich shall be considered negative and east of Greenwich positive;

- **Extent of Latitude:** this 12-bit field, coded as a unsigned binary number, shall specify the extent of latitude of the spherical rectangle that defines the area. It shall be calculated by multiplying the value of the extent of Latitude field by ($90°/2^{15}$);

- **Extent of Longitude:** this 12-bit field, coded as a unsigned binary number, shall specify the extent of longitude of the spherical rectangle that defines the area. It shall be calculated by multiplying the value of the extent of Longitude field by ($180°/2^{15}$).

#### 8.1.16.2 Region label

The region label feature is encoded in extension 3 of FIG type 1 (FIG 1/3). Figure 53 shows the structure of the region label field which is part of the Type 1 field (see also figure 5).

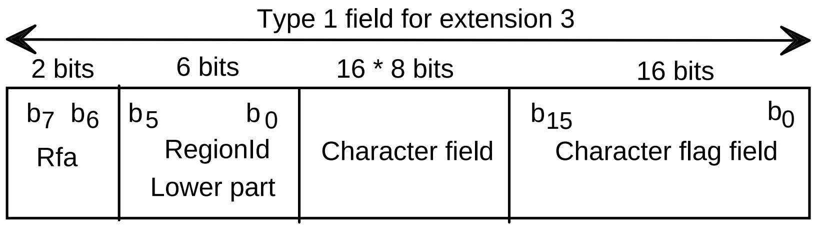


**Figure 53: Structure of the region label field**

The following definitions apply:

**Rfa:** this 2-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**RegionId Lower part**: this 6-bit field shall indicate the lower part of the Region Identifier (see subclause 8.1.16.1). The RegionId Upper part shall be set to "00000".

**Character field**: this 16-byte field shall define the region label. It is coded as a string of 16 characters which are chosen from an 8-bit character set signalled by the Charset field in the first byte of the FIG type 1 data field (see subclause 5.2.2.2).

**Character flag field**: this 16-bit field shall indicate which of the 16 characters of the Character field are to be displayed in an abbreviated form of the label, as follows:

$b_i$: (i=0, ..., 15);
0: not to be displayed in abbreviated form;
1: to be displayed in abbreviated form.

NOTE: Not more than 8 of the $b_i$ may be set to "1".

### 8.1.17 Local service area

The Local service area feature is encoded in extension 23 of FIG type 0 (FIG 0/23). Figure 54 shows the structure of the local service area field which is part of the Type 0 field (see also figure 4).

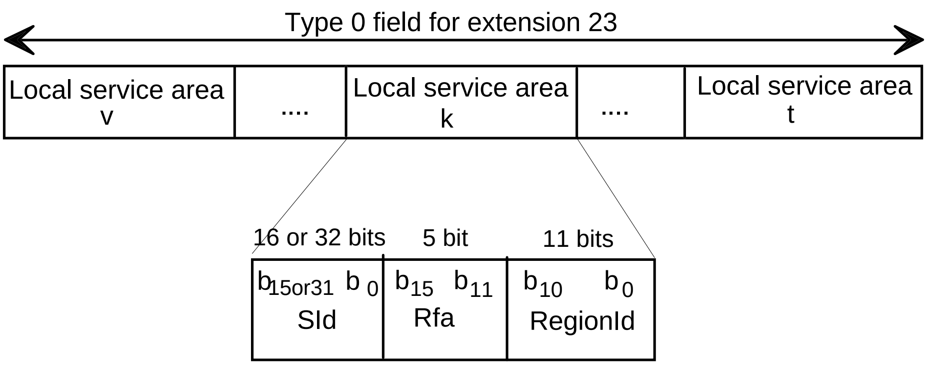


**Figure 54: Structure of Local service area field**

The following definitions apply:

**SId** (Service Identifier): this 16 or 32-bit field shall identify the service (see subclause 6.3.1).

**Rfa**: this 5-bit field shall be reserved for future additions. These bits shall be set to zero until they are defined.

**RegionId**: this 11-bit field shall identify the service area (see subclause 8.1.16).

### 8.1.18 Satellite assistance

Satellite assistance is provided, particularly for satellites in Highly inclined Elliptical Orbits (HEOs), in the form of a database of Doppler shift and time delay differences. A separate handover mechanism allows databases to be exchanged at a precise time.

#### 8.1.18.1 Satellite database

The Satellite database provides the information needed to assist service continuation at satellite handover or when switching to a satellite is performed.

The Satellite database is coded in extension 30 of FIG type 0 (0/30) as shown in figure 55:

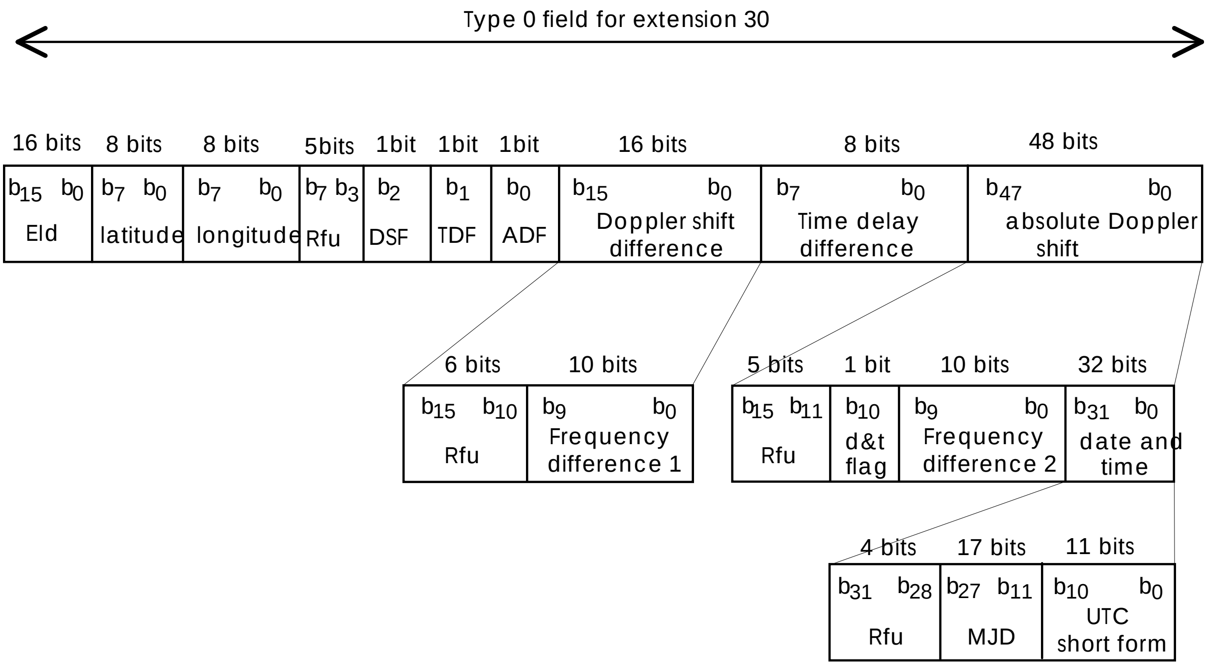


**Figure 55: Structure of the satellite database field**

The following definitions apply:

**EId**: this 16-bit field shall identify the Ensemble.

**Latitude**: this 8-bit field, coded as a two's complement number, shall specify the reference latitude for this field. It shall be calculated by multiplying the value of the latitude field by ($90°/2^7$). Southern latitudes shall be considered negative and northern latitudes positive.

**Longitude:** this 8-bit field, coded as a two's complement number, shall specify the reference longitude for this field. It shall be calculated by multiplying the value of the longitude field by ($180°/2^7$). West of Greenwich shall be considered negative and east of Greenwich positive.

**Rfu**: this 5-bit field shall be reserved for future use of the remainder of the type 0 field for extension 30. The Rfu bits shall be set to zero for the currently specified definition.

**DSF** (Doppler shift difference flag): this 1 bit flag shall indicate whether or not the Doppler shift difference field is present:

0 : Doppler shift difference field not present;
1 : Doppler shift difference field present.

**TDF** (Time Delay difference Flag): this 1 bit flag shall indicate whether or not the time delay difference field is present:

0 : Time delay difference field not present;
1 : Time delay difference field present.

**ADF** (Absolute Doppler shift Flag): this 1 bit flag shall indicate whether or not the Absolute Doppler shift field is present:

0 : Absolute Doppler shift field not present;
1 : Absolute Doppler shift field present.

**Doppler shift difference:**

- **Rfu**: this 6-bit field shall be used for future use of the frequency difference 1 field. The Rfu bits shall be set to zero for the currently specified definition of this field.

- **Frequency difference 1**: this 10-bit field, coded as a two's complement number, shall specify the frequency difference between the signal from the ascending satellite after handover and the signal from the descending satellite before handover at the position specified by latitude and longitude at the time of the next satellite handover. The frequency difference shall be calculated by multiplying the value of this field by 0,1 kHz.

**Time delay difference:** this 8-bit field, coded as a two's complement number, shall specify the difference in round trip time delay between the signal from the ascending satellite after handover and the signal from the descending satellite before handover at the position specified by latitude and longitude at the time of the next imminent satellite handover. The time delay difference shall be calculated by multiplying the value of this field by 0,5 ms.

**Absolute Doppler shift:**

- **Rfu**: this 5-bit field shall be used for future use of the frequency difference 2 field. The Rfu bits shall be set to zero for the currently specified definition of the d&t flag, frequency difference 2 and the date and time fields.

- **d&t** (date and time) **flag**: this 1 bit flag shall indicate whether or not the date and time field is present:

  0 : date and time field not present;
  1 : date and time field present.

- **Frequency difference 2**: this 10-bit field, coded as a two's complement number, shall specify the frequency difference between the centre frequency of the received signal at the position specified by latitude and longitude and at the time specified by the UTC field (if present) or at handover time (if UTC field not present) and the nominal centre frequency of the Ensemble. The frequency difference shall be calculated by multiplying the value of this field by 0,1 kHz.

- **date and time**:

    - **Rfu**: this 4 bit field shall be used for future use of the MJD and UTC fields. The Rfu bits shall be set to zero for the currently specified definition of these fields.

    - **MJD** (Modified Julian Date): see subclause 8.1.3.

    - **UTC, short form:** see subclause 8.1.3.

#### 8.1.18.2 Satellite handover

The Satellite handover information contains a control mechanism for the handover from the descending to the ascending satellite broadcasting a DAB Ensemble. It provides the value of the CIF count (see subclause 6.4) at which the handover occurs.

The satellite handover information is coded in extension 29 of FIG type 0 (0/29) as shown in figure 56.

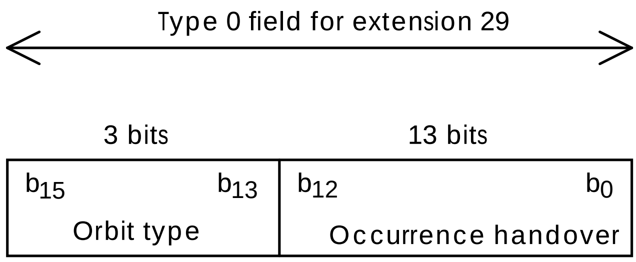


**Figure 56: Structure of the satellite handover field**

The following definitions apply:

**Orbit type**: this 3 bit field shall be used to identify the orbit type of the satellite system used for broadcasting the DAB Ensemble. The following satellite orbit type is defined. The remaining types are reserved for future use.

$b_{15}$ $b_{13}$
0 0 0 : HEO type (ESA).

**Occurrence handover**: this 13-bit field shall indicate the higher ($b_{12}...b_8$) and lower ($b_7...b_0$) parts of the CIF counter from which the ascending satellite broadcasts the ensemble.

## 8.2 Fast Information Data Channel

This subclause defines the data service components which may be transported in the Fast Information Data Channel (FIDC) within the FIC. The FIDC uses FIG type 5 (see figure 6).

### 8.2.1 Paging

The paging feature is encoded in Extension 0 of FIG type 5 (FIG 5/0). A pointer mechanism is used to indicate where the paging information may be carried in the Main Service Channel. Figure 57 shows the structure of the paging field which is part of the Type 5 field (see also figure 6).

The following definitions apply to the flags $D_1$ and $D_2$ defined in subclause 5.2.2.3:

**D1**: this 1-bit flag shall signal two definitions of the Type 5 field (see figure 6), as follows:

0 : paging messages in an integral number of bytes;
1 : pointer - see below.

**D2**: this 1-bit flag shall signal whether the paging messages are scrambled, as follows:

0 : not scrambled according to subclause 9.1.4.4, Configuration 2;
1 : scrambled according subclause 9.1.4.4, Configuration 2.

| | Type 5 field for extension 0 |
|---|---|
| (D1 = 0) | Paging Messages |

| | 6bits | 10bits | 1bit | 1bit | 1bit | 13 bits | 1bit | 11bits | 1bit | 3bits | 16 bits | 32 bits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pointer (D1 = 1) | $b_{15}$ $b_{10}$ SubChId | $b_9$ $b_0$ Packet address | $b_{15}$ F1 | $b_{14}$ F2 | $b_{13}$ Rfa | $b_{12}$ $b_0$ LFN | $b_{15}$ F3 | $b_{14}$ $b_4$ Time | $b_3$ Rfa | $b_2$ $b_0$ CAId | $b_{15}$ $b_0$ SCCA | $b_{31}$ $b_0$ Paging user group |

**Figure 57: Structure of the paging field**

The following definitions apply:

**Paging Messages**: this field shall contain the paging messages.

**SubChId** (Sub-channel Identifier**)**: this 6-bit field shall identify the sub-channel to which the paging information is directed (see subclause 6.2).

**Packet address**: this 10-bit field shall identify the address of the packet, carrying the paging data.

**F1**: this 1-bit flag shall indicate whether the following $F_3$ flag, time, CAId and SCCA fields are present, or not, as follows:

0 : $F_3$ flag, time, CAId and SCCA fields absent;
1 : $F_3$ flag, time, CAId and SCCA fields present.

**F2**: this 1-bit flag shall indicate whether the following time field and LFN applies to the beginning or the end of the service broadcast:

0 : begin of the service broadcast;
1 : end of the service broadcast.

**Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to "0" until it is defined.

**LFN** (Logical Frame Number): this 13-bit field, expressed as an unsigned binary number, shall contain the Logical frame count which indicates the number of the Logical frame from which the broadcast of the service component will start or end.

The value 1FFF (hex) shall be reserved to indicate that the LFN is not effective.

**F3**: this 1-bit flag shall indicate whether the time field, if present, is effective, or not, as follows:

0 : time field effective;
1 : time field not effective.

**Time**: this 11-bit field shall indicate the time from which the emission of the service component will start or end. The time field is coded in the short form of the UTC as described in subclause 8.1.3.

**Rfa**: this 1-bit field shall be reserved for future additions. The bit shall be set to zero until it is defined.

**CAId** (Conditional Access Identifier): this 3-bit field shall identify the Access Control System (ACS) used for the service. If no ACS is used for the service, CAId is set to zero (see subclause 6.3.1).

**SCCA** (Service Component Conditional Access): this 16-bit field shall contain the descrambling parameters for accessing the service component (see subclause 9.2.2).

**Paging user group**: this 32-bit field shall be used to identify the receivers to which the new paging service is directed.

### 8.2.2 Traffic Message Channel (TMC)

Traffic messages are encoded in Extension 1 of FIG type 5 (FIG 5/1) (see note). Figure 58 shows the structure of the TMC message field which is part of the Type 5 field (see also figure 6).

NOTE: TMC messages should be encoded according to the Alert C protocol. However, at the time of writing this ETS, the Alert C protocol has not yet been standardized. A reference to an Alert C preliminary draft standard is given in annex F, bibliography.

The following definitions apply to the flags $D_1$ and $D_2$ (see subclause 5.2.2.3):

**D1**: this 1-bit flag shall be reserved for future use of the Type 5 field (see figure 6);

**D2**: this 1-bit flag shall be reserved for future use of the Type 5 field (see figure 6).

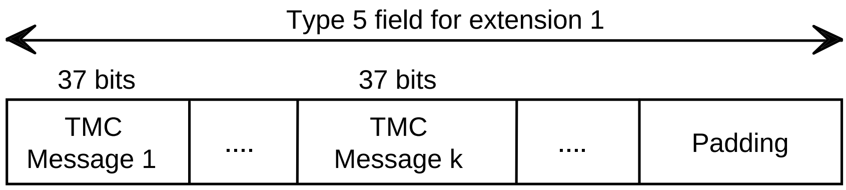


**Figure 58: Structure of Traffic Message Channel field**

The following definitions apply:

**TMC Message**: this 37-bit field shall be reserved for a TMC message see Bibliography.

**Padding**: this field shall contain sufficient bits in the range 1 ... 7 to make up the length to an integral number of bytes. The padding bits shall be set to "0".

### 8.2.3 Emergency Warning Systems (EWS)

The Emergency Warnings Systems (EWS) feature (reference EN 50067 [5]) is encoded in Extension 2 of FIG type 5 (FIG 5/2). Figure 59 shows the structure of the EWS field which is part of the Type 5 field (see also figure 6).

The following definitions apply to the flags D1 and D2 (see subclause 5.2.2.3):

**D1**: this 1-bit flag shall be reserved for future use of the Type 5 field (see figure 6).

**D2**: this 1-bit flag shall signal whether the Type 5 field (see figure 6) contains receiver control information or messages, as follows:

0 : control information;
1 : messages.

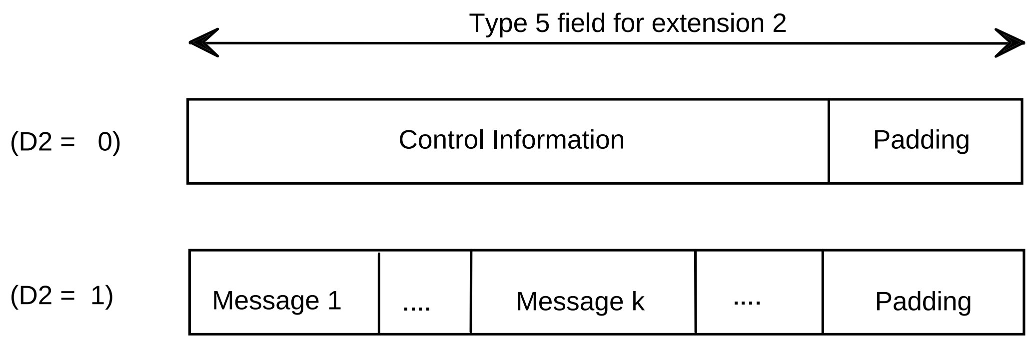


**Figure 59: Structure of Emergency Warning System field**

The following definitions apply:

**Control Information**: this field shall contain control information for EWS receivers.

**Padding**: this field shall contain sufficient bits in the range 1 ... 7 to make up the length to an integral number of bytes. The padding bits shall be set to "0".

**Message**: this field shall contain an EWS message.

# 9 Conditional Access (CA)

The CA system used in the DAB system includes three main functions: scrambling/descrambling, entitlement checking and entitlement management.

The scrambling/descrambling function aims to make the service incomprehensible to unauthorised users. Descrambling can be achieved by any receiver having an appropriate descrambler and holding a secret Control Word (CW). Scrambling can be applied to service components, either using a common Control Word or using separate Control Words for each component.

The entitlement checking function consists of broadcasting the conditions required to access a service, together with encrypted secret codes to enable the descrambling for authorized receivers. These codes are sent inside dedicated messages called Entitlement Checking Messages (ECMs) and these are carried in the ensemble.

The entitlement management function consists of distributing entitlements to receivers. There are several kinds of entitlements matching different means of subscribing to a service: subscription per theme, level or class, pre-booked pay-per-programme or impulse pay-per-programme, per service or per time. This information is sent inside dedicated messages called Entitlement Management Messages (EMMs) and these may be carried in the same ensemble as the scrambled services or may not.

The control and management functions require the use of secret keys and cryptographic algorithms.

This clause describes the mechanisms available to control access to service components sent in the DAB multiplex. Subclause 9.1 describes the scrambling/descrambling procedures for data in Stream and Packet modes. These procedures are completely independent of any other scrambling procedures that may also be performed on the signal (for example energy dispersal scrambling). Subclause 9.2 describes the parameters which are used to provide signalling and synchronization for access control. Subclause 9.3

describes the different possibilities that can be used to send the access control messages (ECMs and EMMs).

## 9.1 Scrambling audio and data

### 9.1.1 Introduction

For each service component, a Conditional Access flag (CA flag) and/or a Conditional Access Identifier (CAId, see subclause 9.2.1) shall be used to indicate whether or not the service component uses Conditional Access mechanisms and, if so, which kind of mechanism is used.

When no CA mechanism is used, the CA scrambling of the service component shall not be used.

When Conditional Access mechanisms are used, the service component shall be sent in one of these three different scrambling modes:

a) unscrambled;

b) scrambled with a specific Control Word (CW), called "local Control Word", which is permanently installed in the receiver;

c) scrambled with a Control Word which is changed regularly. The new value of the CW is sent encrypted to receivers in the Entitlement Checking Messages (ECMs).

In scrambling modes a) and b), no subscription is needed. The service component is said to be in **free access mode**.

In scrambling mode c), a subscription is required to recover the encrypted Control Word. The component is said to be in **controlled access mode**.

### 9.1.2 Description of the audio and data scrambling processes

To scramble audio and data, a Pseudo-Random Binary Sequence (PRBS) shall be added modulo 2 to the audio or data bytes, that shall be scrambled according to the mechanism described in subclauses 9.1.4.2, 9.1.4.3 and 9.1.4.4. The particular bytes which shall remain unscrambled are also defined. During this period the PRBS generator is not activated. The PRBS generator shall be the same as defined in ETS 300 174 [8], subclause 12.2.

### 9.1.3 Generating scrambling and descrambling sequences

An Initialization Word (IW) shall be used to initialize the PRBS generator. The IW bytes shall be inserted in the PRBS, most significant byte first, byte by byte. In this subclause, the formation of the IW is defined and phasing considerations are described.

#### 9.1.3.1 Generation of the initialization word

The Initialization Word is a bit string which shall be used to initialize the PRBS generator. It contains two parts, the Initialization Modifier (IM) and the Control Word (CW):

a) the Initialization Modifier (IM) varies very often (every Logical Frame or every MSC data group) and is used to modify the Initialization Word at each new initialization of the PRBS generator. The PRBS generator is reinitialized very often to allow fast (re)synchronization of the scrambler and the descramblers, and to prevent the output of very long scrambling/descrambling sequences. The Initialization Modifier comprises a number (Logical frame count, MSC data group counter value, notional packet counter value) and sometimes a service component Identifier. This last parameter should be used to prevent two service components using the same ECMs and being scrambled with the same scrambling sequences;

b) the Control Word (CW) is changed less often and provides the "secret key" used to scramble and descramble the service component. The Control Word shall be 64 bits long. In free access mode, the Control Word shall be fixed, it shall have all 64 bits set to "1". In controlled access mode, the Control Word shall be provided by the Access Control System (ACS).

#### 9.1.3.2 Phasing

The period during which a CW is valid is called a **phase**. Each phase shall be allocated a parity (even or odd), which toggles for each new phase. A phase parity flag shall be used to indicate the parity of the current phase.

### 9.1.4 Scrambling/descrambling processes

This subclause specifies three different Conditional Access (CA) signalling configurations and the way Conditional Access is incorporated into the different data transport mechanisms (see subclauses 5.3 and 5.2.2.3 for audio data, data in Stream and Packet mode and for the FIDC, respectively).

#### 9.1.4.1 Conditional Access signalling configurations

Three different signalling configurations are summarized in table 27.

The first is suitable for all data transport mechanisms which are synchronized to the CIF counter; the second is suitable only for data in Packet mode and for data sent in the FIDC; the third is suitable only for data in Packet mode.

**Table 27: CA signalling locations**

| CA signalling configuration | CA data in FIC or sub-channel 63 | CA data with service component |
|---|---|---|
| 1 | IM (Logical frame count)<br>Flags (parameter SCCA)<br>ECM in FIG 6 or sub-channel 63 | |
| 2 | ECM in FIG 6 or sub-channel 63 | IM<br>Flags |
| 3 | | ECM<br>IM<br>Flags |

**Configuration 1**

In this case, all the parameters which are necessary to descramble a service component shall be carried separately from the service component:

- the IM and the parity flag shall be derived from the Logical frame count (see subclause 5.3), the phase parity shall be changed every 250 Logical frames and so the parity flag shall be signalled using bit $b_8$ of the Logical frame count and the IM using bits $b_7$ ... $b_0$ of the Logical frame count;

- the scrambling mode and the updating bits of the service component shall be sent in the parameter SCCA in the FIC;

- the ECMs containing the Control Words shall be sent either in the FIG type 6 or in sub-channel 63.

**Configuration 2**

- the initialization modifier, the phase parity, the scrambling mode and the updating bits shall be sent at the beginning of each MSC data group in the scrambled sub-channel or of each FIG type 5;
- the ECMs containing the Control Words are sent either in the FIG type 6 or in sub-channel 63.

**Configuration 3**

- the Initialization Modifier, the phase parity, the scrambling mode and the updating bits shall be sent at the beginning of each MSC data group in the scrambled sub-channel or of each FIG type 5;
- the ECMs containing the Control Words shall be sent in command packets inserted inside the Packet stream of the service component.

### 9.1.4.2 Scrambling/descrambling of the service components in Stream mode

For audio data, scrambling shall be performed before energy dispersal scrambling (see figure 60).

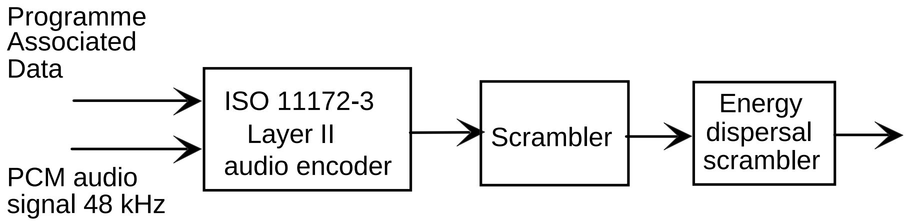


**Figure 60: Scrambling of audio in Stream mode**

For general data, scrambling is performed before energy dispersal scrambling (see figure 61).

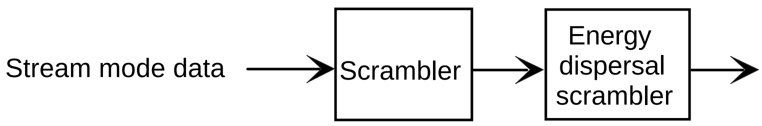


**Figure 61: Scrambling of general data in Stream mode**

In both cases, at each new logical frame, the PRBS generator is initialized with the following Initialization Word (MSB first):

- (8 LSbits of Logical frame count),(2 bits "00"),(6 bits SubChId),(64 bits CW).

The 10 bytes of IW shall be inserted in the PRBS generator, most significant byte first, byte per byte.

### 9.1.4.3 Scrambling/descrambling of the service components in Packet mode

For service components in Packet mode, all the three CA signalling configurations are possible.

**Configuration 1**

When configuration 1 is chosen, scrambling shall be performed after the packet multiplex assembler and before the energy dispersal scrambler as shown in figure 62.

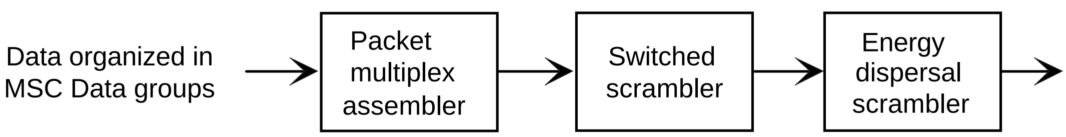


**Figure 62: Scrambling in the Packet mode in configuration 1**

The PRBS generator shall be initialized at the beginning of each packet with the following Initialization Word (MSB first):

- (8 LSbits of Logical frame count),(2 bits "00"),(6 bits SubChId),(8 bits number of the packet in the Logical frame of the sub-channel),(64 bits CW).

At each new Logical frame, the number of the first packet sent in the sub-channel shall be zero. This packet number is incremented (modulo 256) at each new packet in the Logical frame of the sub-channel (independently of its address).

Padding packets, padding bytes (if any), packet headers and the packet CRC shall not be scrambled. The packet CRC shall be calculated on the unscrambled packet header and the unscrambled data field.

The 11 bytes of IW shall be inserted in the PRBS generator, most significant byte first, byte per byte.

**Configurations 2 and 3**

In these two configurations, data, already organized in MSC data group data fields shall be scrambled as shown in figure 63. The Initialization Modifier, the phase parity, the scrambling mode and the updating bits shall be sent at the beginning of each of these MSC data groups in a parameter called DGCA.

Scrambling is performed on the data group data field only. The data group header and the session header (see figure 9) are not scrambled. The data group CRC is performed on the unscrambled data group header, the unscrambled DGCA field, the optional unscrambled segment number, the optional unscrambled EUA field and the scrambled data group data field.

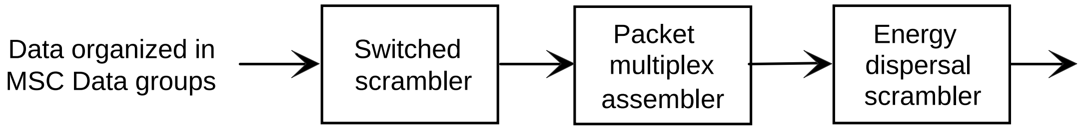


**Figure 63: Scrambling in the Packet mode in configurations 2 and 3**

The PRBS generator shall be initialized at the beginning of the MSC data group with the following Initialization Word (MSbit first):

- (10 bits of Initialization Modifier),(6 bits "000000"),(64 bits CW).

### 9.1.4.4 Scrambling/descrambling of the service components sent in FIDC

For service components sent in FIDC, only CA signalling configurations 1 and 2 are possible. Scrambling is performed before the Fast Information Block assembler.

**Configuration 1**

In configuration 1, scrambling shall be performed on data already organized in the FIG type 5 format (see figure 64).

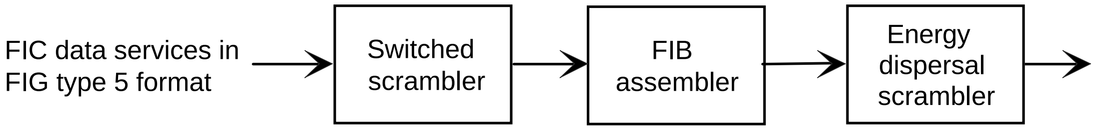


**Figure 64: Scrambling in the FIDC in configuration 1**

The 8 LSbits of the CIF counter shall be used as the IM for all scrambled FIGs sent in FIBs, which are assigned to the same CIF.

The PRBS generator shall be initialized, for each new FIG, with the following Initialization Word (MSbit first):

- (IM),(2 bits "00"),(6 bits FIDCId),(8 bits number of the FIG type 5 in the FIC),(64 bits CW).

For every new IM, the first FIG type 5 field shall have a number equal to zero. This FIG type 5 number shall be incremented by 1 (modulo 256) at each new FIG type 5 field (independently of its Extension field and TCId).

Only the type 5 field is scrambled: the FIG type 5 header and the following byte shall always be unscrambled.

The 11 bytes of IW shall be inserted in the PRBS generator, most significant byte first, byte per byte.

The FIB CRC shall be calculated on all FIGs, scrambled or unscrambled, contained in the FIB data field.

**Configuration 2**

In this configuration, scrambling is performed individually on each FIC data service, before data is organized in the FIG type 5 format (see figure 65).

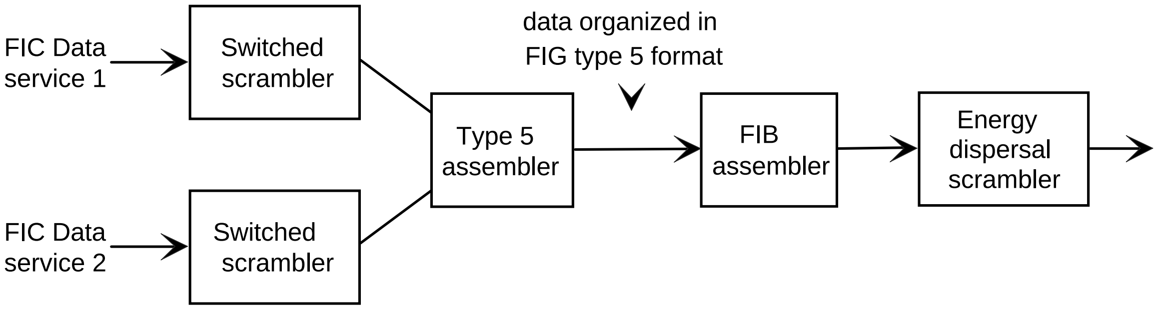


**Figure 65: Scrambling in the FIDC in configuration 2**

The Initialization Modifier, the phase parity, the scrambling mode and the updating bits shall be sent in a parameter called FIDCCA or FIDCCA_Ext. These bits are not scrambled. The situation after scrambling and after the FIG type 5 assembler is shown in figure 66.

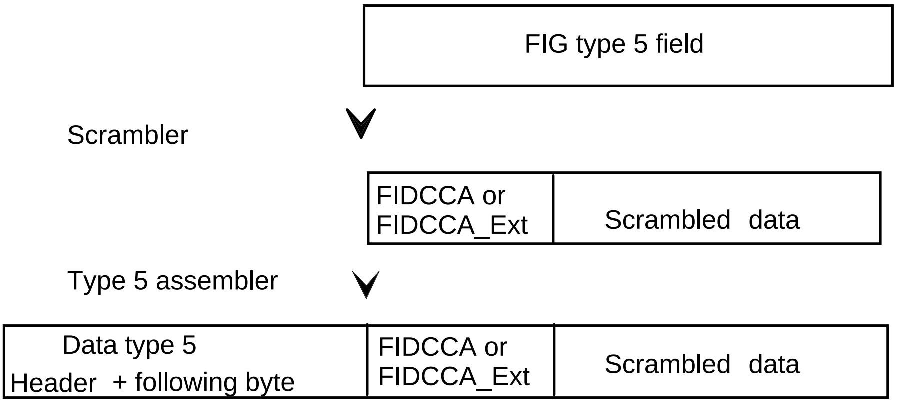


**Figure 66: Insertion of FIDCCA in the FIDC in configuration 2**

The PRBS generator shall be initialized, for each new FIG, with the following Initialization Word (MSB first):

- (10 bits of Initialization Modifier),(6 bits "000000"),(64 bits CW).

The FIB CRC shall be calculated on all FIGs, scrambled or unscrambled, contained in the FIB data field.

## 9.2 Signalling and synchronizing data

This subclause describes all the Access Control parameters which are used to provide signalling and synchronization for Conditional Access.

### 9.2.1 Conditional Access Identifier (CAId)

This 3-bit field shall identify the Conditional Access system used for all the service components of a service (see subclause 6.3.1).

### 9.2.2 Service Component Conditional Access (SCCA)

For each access controlled service component, the SCCA contains the parameters necessary for descrambling. The SCCA comprises two bytes as described in the following two subclauses.

#### 9.2.2.1 First byte of SCCA

- FIC/SC = 0 (CA signalled indirectly).

The CA may be signalled indirectly for Service components in Packet mode or Service components sent in FIDC. The 7 bits ($b_6$ to $b_0$) shall be stuffing bits which are set to "0" as shown on figure 67.

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| FIC/SC = 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 67: First byte of SCCA, when not used to signal CA directly**

- FIC/SC = 1 (CA signalled directly)

The 7 bits ($b_6$ to $b_0$) are defined in figure 68. They contain scrambling flags which shall indicate whether or not the Service component is scrambled and which shall describe which scrambling mode is used. These flags also contain the signalling bits which shall control the switching from one scrambling mode to another and the synchronization of the scrambler and descrambler.

| $b_7$ FIC/SC = 1 | $b_6$ Scr | $b_5$ Scr | $b_4$ Rp | $b_3$ Upd | $b_2$ Upd | $b_1$ Upd | $b_0$ Upd |
|---|---|---|---|---|---|---|---|

**Figure 68: First byte of SCCA, when used to signal CA directly**

The following definitions apply:

**Scr:** this 2-bit field shall identify the scrambling mode as follows:

$b_6$ $b_5$
0 0 : not allowed;
0 1 : unscrambled;
1 0 : scrambled with a local Control Word;
1 1 : scrambled with a Control Word regularly transmitted and changed with ECMs.

**Rp:** this 1-bit flag shall indicate replacement operations as follows:

0 : replacement is inactive;
1 : replacement is active and the receiver has to take into account the replacement characteristics given by the Access Control System (ACS).

It indicates to the receiver when to take into account the replacement characteristics given by the ACS if it is in a blackout state.

**Upd:** this 4-bit field shall be used for managing CA updates. There are three CA update possibilities:

1) Update "mode": this update shall indicate a change in the scrambling mode. The future scrambling mode is described by the two bits $b_1$ and $b_0$ and should be taken into account when the four least significant bits of the Logical frame count are zero:

$b_3$ $b_0$
0 0 0 0 : no update;
x x 0 1 : update "mode"; future mode is "unscrambled";
x x 1 0 : update "mode"; future mode is "scrambled with a local Control Word";
x x 1 1 : Update "mode"; future mode is "scrambled with a Control Word regularly transmitted and changed with ECMs.

2) Update "ECM": this update shall indicate a change in the ECM transmission and makes the descrambler read the next ECM:

$b_3$ $b_0$
x 1 x x : update ECM.

3) Update "access": this update shall indicate a change in the access conditions. The new access conditions should be taken into account when the four least significant bits of the Logical frame count are zero:

$b_3$ $b_0$
1 x x x : update "access".

### 9.2.2.2 Second byte of SCCA

The second byte of SCCA shall indicate how to find the ECMs and the possible EMMs of the access controlled service component.

The ECMs shall be sent in either the FIC in FIG type 6 or in sub-channel 63 or in the same sub-channel as the service component itself. This last option is possible only for service components sent in Packet mode.

The EMM shall be sent together with the ECM or in the sub-channel 63.

Table 28 summarizes all the possible configurations.

**Table 28: Allowed ECM/EMM transport mechanisms inside the ensemble**

| **EMM** | **ECM** | | |
|---|---|---|---|
| | **FIG type 6** | **Component** | **Sub-channel 63** |
| **FIG type 6** | Case 1 | not allowed | not allowed |
| **Component** | not allowed | Case 4 | not allowed |
| **Sub-channel 63** | Case 2 | Case 5 | Case 3 |

For the cases indicated in table 28, the following definitions apply:

**Case 1:** FIC/Sub-channel of the Main Service Channel (SubCh) 63/SC = 00.

| $b_7$ $b_6$ | $b_5$ $b_0$ |
|---|---|
| FIC/SubCh63/SC = "00" | FIC_ECMId |

**Figure 69: Second byte of SCCA for Case 1**

In this case, the ECMs and the EMMs shall be sent in the FIC in the FIG type 6.

The bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (FIC_ECMId) which is used to identify the structure containing the ECM message in the FIG type 6. The value "000000" is not allowed.

**Case 2:** FIC/SubCh 63/SC = 01.

| $b_7$ $b_6$ | $b_5$ $b_0$ |
|---|---|
| FIC/SubCh63/SC = "01" | FIC_ECMId |

**Figure 70: Second byte of SCCA for Case 2**

In this case, the ECMs shall be sent in the FIC in the FIG type 6 and the EMMs shall be sent in the MSC in the sub-channel identified by the SubChId 63.

The bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (FIC_ECMId) which is used to identify the structure containing the ECM message in the FIG type 6. The value "000000" is not allowed.

**Case 3:** FIC/Subch 63/SC = 10.

| $b_7$ $b_6$ | $b_5$ $b_0$ |
|---|---|
| FIC/SubCh63/SC = "10" | Pk_ECMId |

**Figure 71: Second byte of SCCA for Case 3**

In this case, the ECM and the EMM shall be sent in the MSC in the ECM/EMM sub-channel identified by the SubChId 63.

The bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (Pk_ECMId) which is used to identify the structure containing the ECM message (6 least significant bits of the address of the packets transporting these ECMs). The value "000000" is not allowed.

**Cases 4 and 5:** FIC/Subch 63/SC = 11.

| $b_7$ … $b_6$ | $b_5$ | $b_4$ … $b_0$ |
|---|---|---|
| FIC/SubCh63/SC = "11" | EMM flag | 0 0 0 0 0 |

**Figure 72: Second byte of SCCA for Cases 4 and 5**

In these cases, the ECMs shall be sent in the same sub-channel as the service component. This can be achieved for service components in Packet mode only.

The transmission of the EMM shall depend on the value of EMM flag, as follows:

EMM flag = 0 : the EMMs are sent in the same sub-channel as the service component (Case 4);

EMM flag = 1 : the EMMs are sent in sub-channel 63 (Case 5).

The bits $b_4$ to $b_0$ are stuffing bits, which shall be set to "0".

### 9.2.3 Data Group Conditional Access (DGCA)

This 16-bit parameter is used to transport the Initialization Modifier (IM) and the scrambling flags in the headers of the MSC data groups carrying the service component.

This parameter shall be carried in the extension field of data groups with type = "0010". Consequently, the extension flag for the data group header is set to "1". The command bit of packet headers shall be set to "0" (data) (see subclauses 5.3.2 and 5.3.3).

The coding of DGCA is described in figure 73:

| 10 bits | 1 bit | 1 bit | 1 bit | 2 bits | 1 bit |
|---|---|---|---|---|---|
| $b_{15}$ … $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ … $b_1$ | $b_0$ |
| IM | Rfa | Rp | Parity flag | Scrambling mode | Update ECM |

**Figure 73: Coding of the Data Group Conditional Access (DGCA) field**

**Initialization Modifier (IM):** this 10-bit parameter shall be used together with the Control Word to form the initialization word used to initialize the PRBS generator.

**Rfa:** this bit shall be reserved for future additions. The bit shall be set to "0" until it is defined.

**Rp:** this 1-bit flag shall indicate replacement operations as follows:

0 : replacement is inactive;
1 : replacement is active and the receiver has to take into account the replacement characteristics given by the Access Control System.

It indicates to the receiver when to take into account the replacement characteristics given by the ACS if it is in a blackout state.

**Parity flag:** this 1-bit flag shall be used to indicate the parity of the control word to be used for the current data group, as follows:

0 : even parity;
1 : odd parity.

**Scrambling mode:** this 2-bit field shall specify the scrambling mode, as follows:

$b_2$ $b_1$
0 0 : not allowed;
0 1 : unscrambled;
1 0 : free access (i.e. scrambled with a local Control Word);
1 1 : controlled access (i.e. scrambled with a Control Word regularly transmitted and changed with ECMs).

**Update ECM: t**his 1-bit flag shall indicate a change in the ECM transmission and make the descrambler read the next ECM:

0 : no update;
1 : update ECM. Next ECM shall be sent to the ACS.

### 9.2.4 Fast Information Data Channel Conditional Access (FIDCCA and FIDCCA_Ext)

#### 9.2.4.1 FIDCCA

FIDCCA is a 16-bit parameter which is used to transport the Initialization Modifier (IM) and some scrambling flags at the start of the FIG type 5 field (see figure 66) transporting the service component. This parameter shall exist if the CA flag of the service component is set to "1" and/or if CAId is not equal to zero.

The coding of FIDCCA is described in figure 74:

| 10 bits | 1 bit | 1 bit | 1 bit | 2 bits | 1 bit |
|---|---|---|---|---|---|
| $b_{15}$ $b_6$ IM | $b_5$ Ext. flag | $b_4$ Rp | $b_3$ Parity flag | $b_2$ $b_1$ Scrambling mode | $b_0$ Update ECM |

**Figure 74: Coding of the Fast Information Data Channel Conditional Access (FIDCCA) field without extension**

**Initialization Modifier:** this 10-bit field shall be used together with the Control Word to form the initialization word used to initialize the PRBS generator.

**Ext. flag:** this 1-bit flag shall distinguish between FIDCCA and FIDCCA_Ext:

0 : FIDCCA;
1 : FIDCCA_Ext.

**Rp:** this 1-bit flag shall indicate replacement operations, as follows:

0 : replacement is inactive;
1 : replacement is active and the receiver has to take into account the replacement characteristics given by the Access Control System (ACS).

It should indicate to the receiver when to take into account the replacement characteristics given by the ACS if it is in a blackout state.

**Parity flag:** this 1-bit flag shall be used to indicate the parity of the control word to be used for the current FIG type 5, as follows:

0 : even parity;
1 : odd parity.

**Scrambling mode:** this 2-bit field describes the scrambling mode as follows:

$b_2$ $b_1$
0 0 : not allowed;
0 1 : unscrambled;
1 0 : free access (i.e. scrambled with a local Control Word);
1 1 : controlled access (i.e. scrambled with a Control Word regularly transmitted and changed with ECMs).

**Update ECM:** this 1-bit flag shall indicate a change in the ECM transmission and it makes the descrambler read the next ECM:

0 : no update;
1 : update ECM. Next ECM shall be sent to the ACS.

### 9.2.4.2 FIDCCA_Extended

FIDCCA_Extended is a 24-bit parameter consisting of FIDCCA and the second byte of SCCA indicating where the ECMs of the service component can be found.

The coding of FIDCCA_Ext is described in figure 75.

| 10 bits | 1 bit | 1 bit | 1 bit | 2 bits | 1 bit | 2 bits | 6 bits |
|---|---|---|---|---|---|---|---|
| $b_{15}$ $b_6$ IM | $b_5$ Ext. flag | $b_4$ Rp | $b_3$ Parity flag | $b_2$ $b_1$ Scrambling mode | $b_0$ Update ECM | $b_7$ $b_6$ FIC/SubCh63 /SC | $b_5$ $b_0$ ECM Identifier |

**Figure 75: Coding of the Fast Information Data Channel Conditional Access - Extended field**

The first two bytes are the same as for FIDCCA (see subclause 9.2.4.1). The remaining parameters are defined as follows:

- FIC/SubCh 63/SC = 00:

  - in this case, the ECMs and the EMMs shall be sent in the FIG type 6;

  - the bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (FIC_ECMId) which is used to identify the structure containing the ECM message in FIG type 6. The value "000000" is not allowed.

- FIC/SubCh 63/SC = 01:

  - in this case, the ECMs shall be sent in the FIC in the FIG type 6 and the EMMs shall be sent in sub-channel 63;

  - the bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (FIC_ECMId) which is used to identify the structure containing the ECM message in the FIG type 6. The value "000000" is not allowed.

- FIC/SubCh 63/SC = 10:

  - in this case, the ECM and the EMM shall be sent in sub-channel 63;

  - the bits $b_5$ to $b_0$ shall indicate the value of the ECM Identifier (Pk_ECMId) which is used to identify the structure containing the ECM message (6 least significant bits of the address of the packets transporting these ECMs). The value "000000" is not allowed.

- FIC/SubCh 63/SC = 11: this case shall be reserved for future use.

## 9.3 ECM and EMM transmission

ECMs (Entitlement Checking Messages) give information about the conditions required to access a service. EMMs (Entitlement Management Messages) transport new entitlements and management data to customers. This subclause describes the coding of ECMs and EMMs and their transport mechanisms.

### 9.3.1 General description

All access control messages shall begin with a parameter CAId identifying the Access Control System which can interpret and process the messages. The receiver only sends to the ACS the messages which the ACS can interpret and process.

#### 9.3.1.1 ECM coding

An ECM identifier (ECMId) shall be used to point to a specific ECM. The ECM is coded as shown in figure 76:

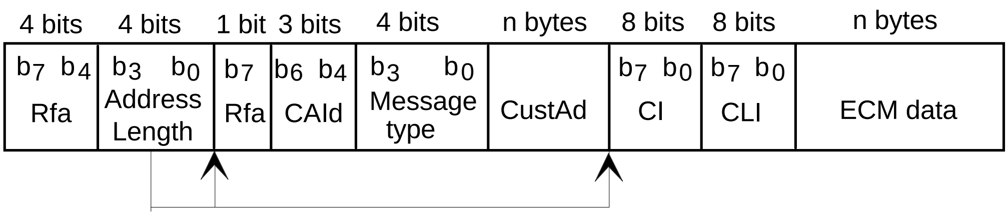


**Figure 76: ECM coding field**

The following definitions apply:

**Rfa:** this 4-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Address length**: this 4-bit field, expressed as an unsigned binary number in the range 1 to 15, shall specify the length in bytes of the field comprising the following Rfa, CAId, message type and CustAd. The length "1" signals that there is no CustAd field.

**Rfa:** this bit shall be reserved for future additions. The bit shall be set to zero until it is defined.

**CAId:** see subclause 9.2.1.

**Message type** (type of message): this 4-bit field shall specify the type of message, as follows (the remaining types are used for EMMs, see subclause 9.3.1.2):

| $b_3$ $b_0$ | |
|---|---|
| 0 0 0 0 | : ECM; |
| 0 0 0 1 | : reserved for specific ECM; |
| 0 0 1 x | : reserved for specific ECM. |

**CustAd** (Customer Address): this parameter is optional for ECMs. The length of CustAd shall be defined as follows:

- 40 bits: CustAd should also be called UA (Unique Address);
- 24 bits: CustAd should also be called SA (Shared Address);
- 16 bits: CustAd should also be called CCA (Collective Address).

**CI** (Command Identifier): this 8-bit field shall specify the format of the parameter field and the crypto-algorithm type (see subclause 9.3.1.3).

**CLI** (Command Length Indicator): this 8-bit field (expressed as an unsigned binary number) shall indicate the number of bytes of the ECM data field.

### 9.3.1.2 EMM coding

All EMMs shall be sent inside structures containing at least the parameters shown in figure 77:

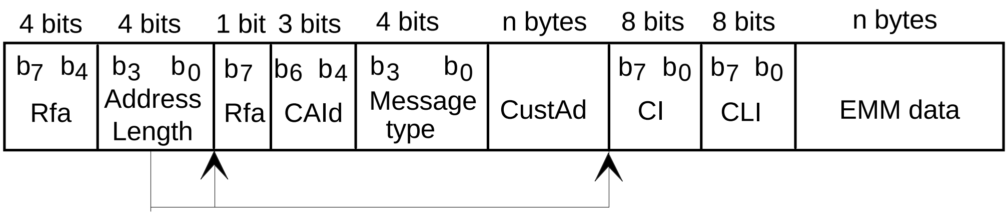


**Figure 77: EMM coding field**

The following definitions apply:

**Rfa:** this 4-bit field shall be reserved for future additions. The bits shall be set to zero until they are defined.

**Address length**: see subclause 9.3.1.1.

**Rfa:** this bit shall be reserved for future additions. The bit shall be set to zero until it is defined.

**CAId:** see subclause 9.2.1.

**Message type** (type of message): this 4-bit field shall specify the type of message, as follows (the remaining types are reserved for future use of EMM message types):

| $b_3$ $b_0$ | |
|---|---|
| 0 0 x x | : values not allowed (reserved for ECMs - see subclause 9.3.1.1); |
| 0 1 0 0 | : EMM for a unique customer (EMM-U); |
| 0 1 0 1 | : EMM for small groups of customers (EMM-S); |
| 0 1 1 0 | : EMM for large groups of customers (EMM-C); |
| 0 1 1 1 | : EMM for the entire audience (EMM-G). |

**CustAd (**Customer Address)**:** this parameter shall exist in all EMMs, except EMM-G. The length of CustAd shall be defined as follows:

- 40 bits for EMM-U. In this case, CustAd should also be called UA (Unique Address);

- 24 bits for EMM-S. In this case, CustAd should also be called SA (Shared Address);

- 16 bits for EMM-C. In this case, CustAd should also be called CCA (Collective Address).

**CI** (Command Identifier): see subclause 9.3.1.1.

**CLI** (Command Length Indicator): this 8-bit field, expressed as an unsigned binary number, shall indicate the number of bytes of the EMM data field.

### 9.3.1.3 Command Identifier (CI) coding

The CI describes the format used for the parameter field and the type of cryptographic algorithm used for decryption. It shall be included in all EMMs and ECMs. Its content is described figure 78.

| 6 bits | 1 bit | 1 bit |
|---|---|---|
| $b_7$ … $b_2$ | $b_1$ | $b_0$ |
| Type of crypto-algorithm | Rfa | T |

**Figure 78: Coding of the Command Identifier field**

The following definitions apply:

**Type of crypto-algorithm:** this 6-bit field shall be used to identify one of 64 types of crypto-algorithms.

**Rfa:** this bit shall be reserved for future additions. The bit shall be set to "0" until it is defined.

**T:** the toggle bit. It shall be maintained in the same state as long as the content of the message has not changed. It shall be used in EMM-G and in ECM to indicate a change in the information content of these messages. It has no meaning for the EMM-U, EMM-C and EMM-S. The toggle bit is attached to a given crypto-algorithm type ; therefore, if ECMs or EMM-G corresponding to two different types of crypto-algorithm are sent, the corresponding toggle bits are kept separate.

### 9.3.2 Transport

ECMs and EMMs can be sent in the FIC, sub-channel 63 in the MSC or in the same sub-channel as the service component.

### 9.3.2.1 MSC

The ECM is carried in an MSC data group as shown in figure 79 (see also figure 9):

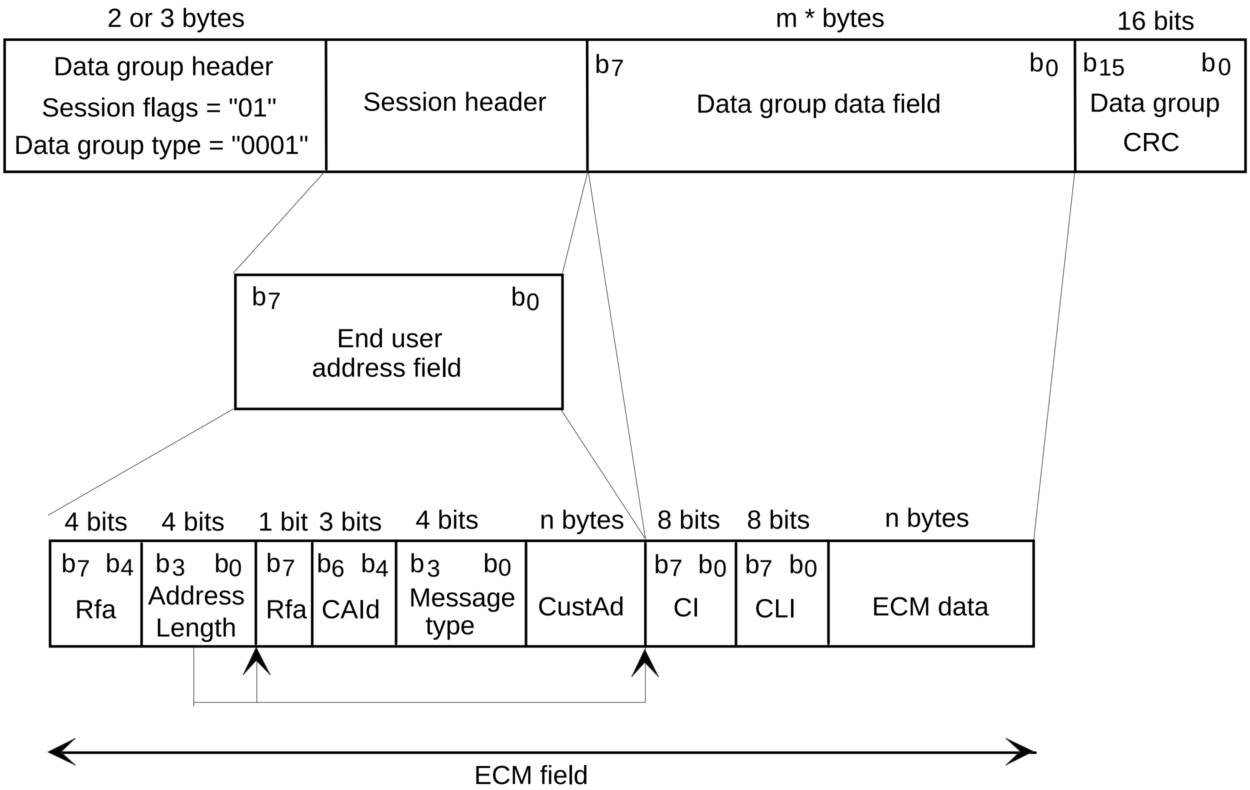


**Figure 79: Data group structure containing an ECM**

The following definitions apply:

**Data group header**: see subclause 5.3.3.1. The session flags shall be set to indicate no segment number but the end user address field present ("01"). The data group type shall be set to "CA" ("0001");

**Session header**: the Last flag and Segment number are absent, only the end user address field remains - see subclause 5.3.3.2;

**Data group data field**: see subclause 5.3.3.3;

**CRC**: see subclause 5.3.3.4;

**Rfa, Address length, CAId, Message type, CustAd, CI, CLI, ECM data**: see subclause 9.3.1.1.

The EMM is carried in an MSC data group as shown in figure 80 (see also figure 9):

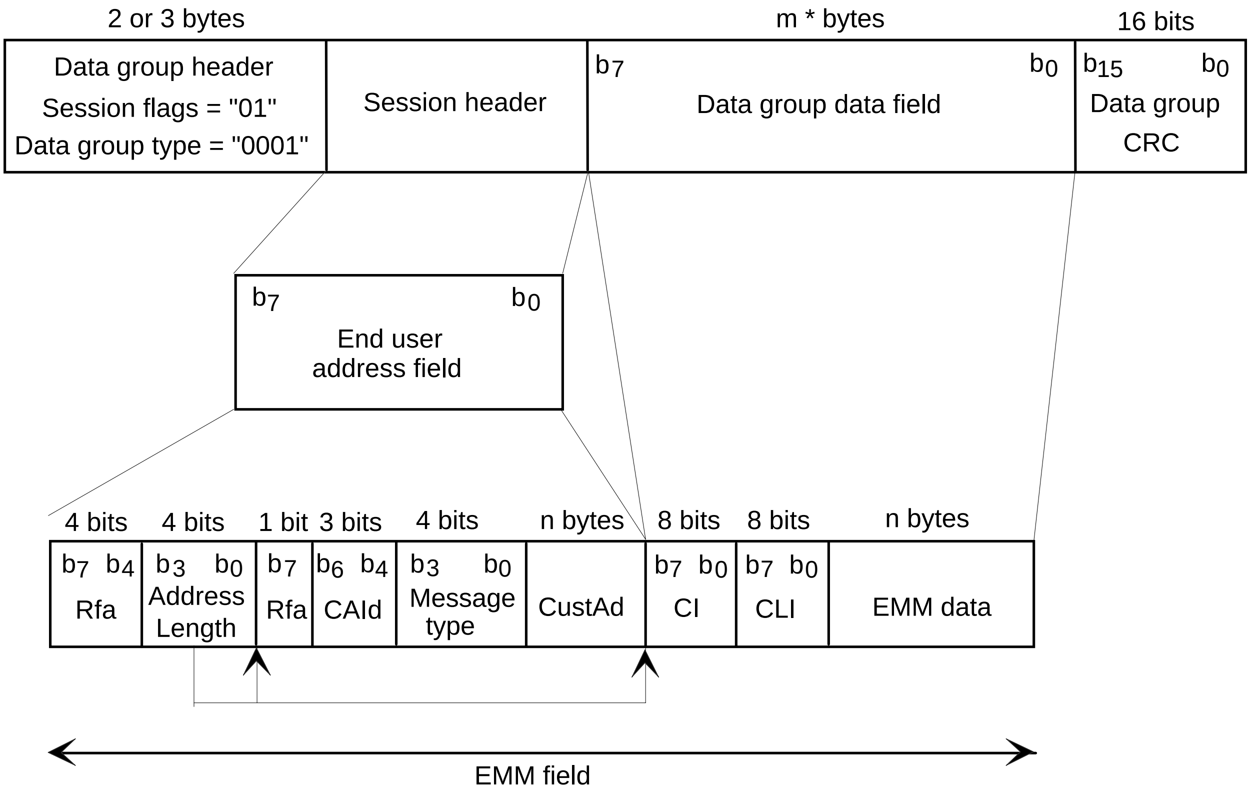


**Figure 80: Data group structure containing an EMM**

The following definitions apply:

**Data group header**: see subclause 5.3.3.1. The session flags shall be set to indicate no segment number but the end user address field present ("01"). The data group type shall be set to "CA" ("0001").

**Session header**: the Last flag and Segment number are absent, only the end user address field remains - see subclause 5.3.3.2.

**Data group data field**: see subclause 5.3.3.3.

**CRC**: see subclause 5.3.3.4.

**Rfa, Address length, CAId, Message type, CustAd, CI, CLI, EMM data**: see subclause 9.3.1.2.

At the network level, each MSC data group containing one ECM or one EMM shall be carried in one or several packets having the same address (see subclause 5.3.2).

The EMMs of all the access controlled service components shall be carried in packets having the same address (see table 29).

The ECMs of each access controlled service component shall be carried in packets with addresses described in table 29.

**Table 29: Packet address for ECMs and EMMs -**

| Type of message | Packet address (10 bits) | |
|---|---|---|
| | *MSb* | *LSb* |
| | $b_9$ $b_6$ | $b_5$ $b_0$ |
| ECM | 0 0 0 1 | Pk_ECMId (6 bits) |
| EMM | 0 0 0 1 | 0 0 0 0 0 0 |

### 9.3.2.2 FIC

The ECM and the EMM shall be carried in FIG type 6 as shown in figure 81 (see also figure 7):

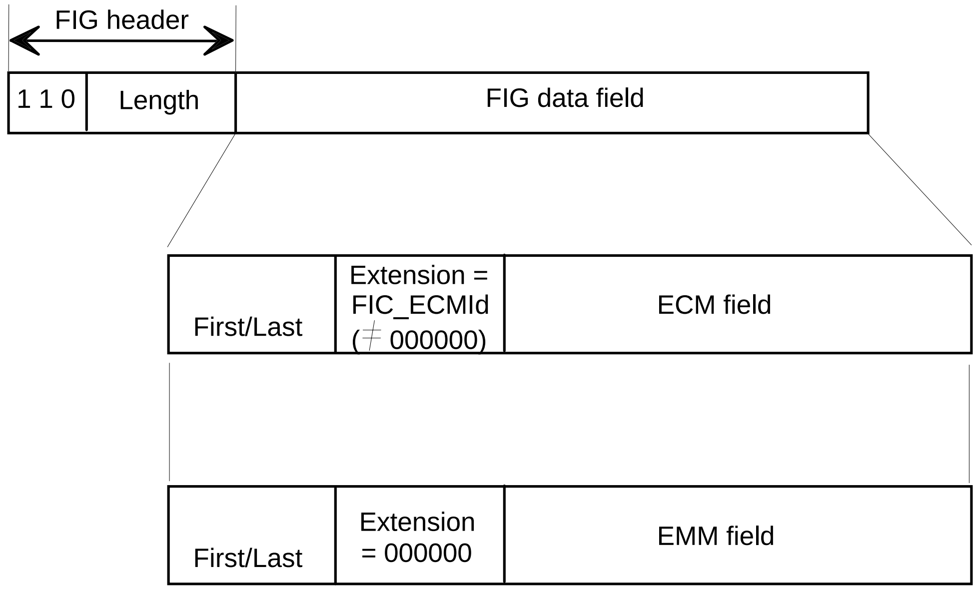


**Figure 81: FIG type 6 containing ECM and EMM**

The following definitions apply:

**FIG header**, **FIG data field**: see subclause 5.2.2.

**First/Last**, **Extension**: see subclause 5.2.2.4.

**FIC_ECMId** (FIC ECM Identifier): this 6-bit field shall identify the portion of the ECM data carried in the FIG;

**ECM field**: this field shall contain all or part of the ECM data (see subclause 9.3.1.1 and figure 79). The extension field contains all or part of one ECM identified by FIC_ECMId.

**EMM field**: this field shall contain all or part of one EMM (see subclause 9.3.1.2 and figure 80).

### 9.3.2.3 Together with service component

The ECMs and EMMs shall be coded in the same way as that described for the MSC in subclause 9.3.2.1.

At the network level, each MSC data group, containing one ECM or one EMM, shall be carried in one or several command packets having the same address as the service component.

# 10 Energy dispersal

## 10.1 General procedure

In order to ensure appropriate energy dispersal in the transmitted signal, the individual inputs of the energy dispersal scramblers shown in figure 1 shall be scrambled by a modulo-2 addition with a pseudo-random binary sequence (PRBS), prior to convolutional encoding.

The PRBS shall be defined as the output of the feedback shift register of figure 82. It shall use a polynomial of degree 9, defined by:

$$P(X) = X^9 + X^5 + 1$$

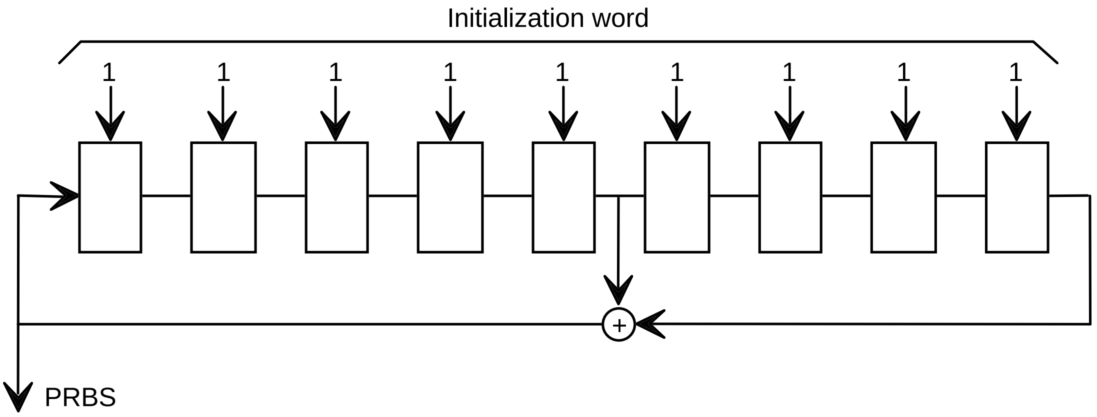


**Figure 82: PRBS generator**

The initialization word shall be applied in such a way that the first bit of the PRBS is obtained when the outputs of all shift register stages are set to value "1"; the first 16 bits of the PRBS are given in table 30:

**Table 30: First 16 bits of the PRBS**

| bit index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bit value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

## 10.2 Energy dispersal as applied in the Fast Information Channel

The FIBs defined in subclause 5.2.1 shall be processed by the energy dispersal scrambler as follows:

**Transmission modes I and II:**

The 3 FIBs corresponding to one CIF shall be grouped together at the output of the Fast Information Block assembler to form a 768-bit vector. This vector shall be a juxtaposition of the 3 FIBs, the first bit of this vector being the first bit of the first FIB.

This vector shall be scrambled with the PRBS, the first bit of the vector being added modulo 2 to the PRBS bit of index 0.

**Transmission mode III:**

The 4 FIBs corresponding to one CIF shall be grouped together at the output of the Fast Information Block (FIB) assembler to form a 1 024-bit vector. This vector shall be a juxtaposition of the 4 FIBs, the first bit of this vector being the first bit of the first FIB.

This vector shall be scrambled with the PRBS, the first bit of the vector being added modulo 2 to the PRBS bit of index 0.

### 10.3 Energy dispersal as applied in the Main Service Channel.

The content of each Logical frame at the output of the SI packet multiplex assembler, and at the output of each CA scrambler, as shown in figure 1, shall be scrambled in such a way that the first bit of each Logical frame associated with a given sub-channel shall be added modulo 2 to the PRBS bit of index 0.

## 11 Convolutional Coding

The channel encoding process is based on punctured convolutional coding, which allows both equal and Unequal Error Protection (UEP), matched to bit error sensitivity characteristics.

This process is applied to the output of each energy dispersal scrambler. The output of a scrambler is denoted as a vector $\left(a_i\right)_{i=0}^{I-1}$ of $I$ bits during any given Logical frame.

Subclause 11.1 defines the general encoding procedure. Subclauses 11.2 and 11.3 define the particular application of the encoding procedure in the FIC and in the MSC.

### 11.1 Convolutional code

#### 11.1.1 Mother code

The channel coding is based on a convolutional code with constraint length 7. The mother convolutional encoder generates from the vector $\left(a_i\right)_{i=0}^{I-1}$ a codeword $\left\{\left(x_{0,i}, x_{1,i}, x_{2,i}, x_{3,i}\right)\right\}_{i=0}^{I+5}$. This codeword shall be defined by:

$$x_{0,i} = a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6};$$

$$x_{1,i} = a_i \oplus a_{i-1} \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-6};$$

$$x_{2,i} = a_i \oplus a_{i-1} \oplus a_{i-4} \oplus a_{i-6};$$

$$x_{3,i} = a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6};$$

for $i = 0, 1, 2, ..., I+5$.

When $i$ does not belong to the set $\{0,1,2,\ldots,I-1\}$, $a_i$ shall be equal to zero by definition.

The encoding can be achieved using the convolutional encoder presented in figure 83.

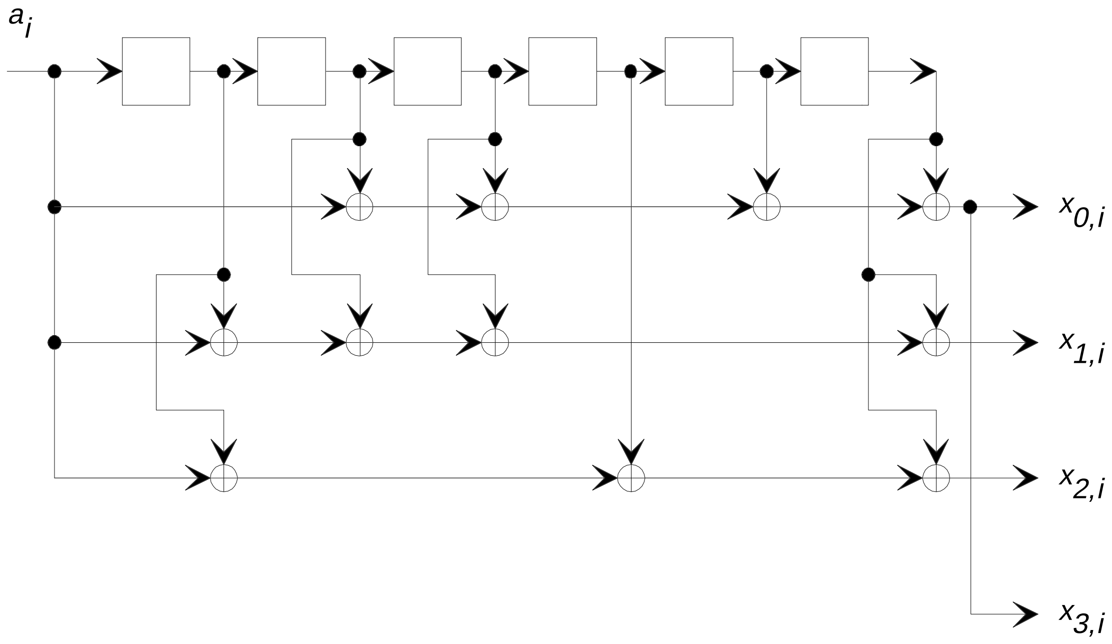


**Figure 83: Convolutional encoder**

The octal forms of the generator polynomials are 133, 171, 145 and 133, respectively.

The vector $(a_{-6}, a_{-5}, a_{-4}, a_{-3}, a_{-2}, a_{-1})$ corresponds to the all-zero initial state of the shift register and the vector $(a_I, a_{I+1}, a_{I+2}, a_{I+3}, a_{I+4}, a_{I+5})$ corresponds to the all-zero final state of the shift register.

Each codeword $\left\{(x_{0,i}, x_{1,i}, x_{2,i}, x_{3,i})\right\}_{i=0}^{I+5}$ is ordered as a serial mother codeword $\boldsymbol{U}$ with:

$$\mathbf{U} = (u_0, u_1, u_2, \ldots, u_{4I+23});$$

and

$$u_i = x_{R\left(\frac{i}{4}\right), Q\left(\frac{i}{4}\right)} \quad \text{for} \quad i = 0, 1, 2, \ldots, 4I+23.$$

The functions R and Q, denoting the remainder and the quotient of a division, respectively, are defined in subclause 3.3.

### 11.1.2 Puncturing procedure

Punctured convolutional coding shall be applied; some predefined codebits generated by the mother code are not transmitted.

The first $4I$ bits $(u_0, u_1, u_2, \ldots, u_{4I-1})$ of the serial mother codeword $\boldsymbol{U}$ shall be split into consecutive blocks of 128 bits.

Each block shall be divided into four consecutive sub-blocks of 32 bits. All sub-blocks belonging to the same block shall be punctured using the same rule, given by the value of the puncturing index $PI$.

Each index $PI$ corresponds to a puncturing vector $\boldsymbol{V_{PI}}$, denoted by:

$$\mathbf{V_{PI}} = (v_{PI,0}, v_{PI,1}, \ldots, v_{PI,i}, \ldots, v_{PI,31}).$$

The $(i + 1)^{th}$ bit $(i = 0, 1, 2, ..., 31)$ in each sub-block is processed according to the value of the element $v_{PI,i}$ of the puncturing vector $\boldsymbol{V_{PI}}$, in the following way:

- for $v_{PI,i} = 0$, the corresponding bit shall be taken out of the sub-block and shall not be transmitted;

- for $v_{PI,i} = 1$, the corresponding bit shall be retained in the sub-block and shall be transmitted.

The values of the puncturing vectors are given in table 31.

In this table, the value of the code rate (equal to $\frac{8}{8+PI}$) is also indicated.

**Table 31: Puncturing vectors**

| | $(v_{PI,0}$ ,... ..,$v_{PI,31})$ |
|---|---|
| PI=1:<br>code rate: 8/9 | 1100 1000 1000 1000 1000 1000 1000 1000 |
| PI=2:<br>code rate: 8/10 | 1100 1000 1000 1000 1100 1000 1000 1000 |
| PI=3:<br>code rate: 8/11 | 1100 1000 1100 1000 1100 1000 1000 1000 |
| PI=4:<br>code rate: 8/12 | 1100 1000 1100 1000 1100 1000 1100 1000 |
| PI=5:<br>code rate: 8/13 | 1100 1100 1100 1000 1100 1000 1100 1000 |
| PI=6:<br>code rate: 8/14 | 1100 1100 1100 1000 1100 1100 1100 1000 |
| PI=7:<br>code rate: 8/15 | 1100 1100 1100 1100 1100 1100 1100 1000 |
| PI=8:<br>code rate: 8/16 | 1100 1100 1100 1100 1100 1100 1100 1100 |
| PI=9:<br>code rate: 8/17 | 1110 1100 1100 1100 1100 1100 1100 1100 |
| PI=10:<br>code rate: 8/18 | 1110 1100 1100 1100 1110 1100 1100 1100 |
| PI=11:<br>code rate: 8/19 | 1110 1100 1110 1100 1110 1100 1100 1100 |
| PI=12:<br>code rate: 8/20 | 1110 1100 1110 1100 1110 1100 1110 1100 |
| PI=13:<br>code rate: 8/21 | 1110 1110 1110 1100 1110 1100 1110 1100 |
| PI=14:<br>code rate: 8/22 | 1110 1110 1110 1100 1110 1110 1110 1100 |
| PI=15:<br>code rate: 8/23 | 1110 1110 1110 1110 1110 1110 1110 1100 |
| PI=16:<br>code rate: 8/24 | 1110 1110 1110 1110 1110 1110 1110 1110 |
| PI=17:<br>code rate: 8/25 | 1111 1110 1110 1110 1110 1110 1110 1110 |
| PI=18:<br>code rate: 8/26 | 1111 1110 1110 1110 1111 1110 1110 1110 |
| PI=19:<br>code rate: 8/27 | 1111 1110 1111 1110 1111 1110 1110 1110 |
| PI=20:<br>code rate: 8/28 | 1111 1110 1111 1110 1111 1110 1111 1110 |
| PI=21:<br>code rate: 8/29 | 1111 1111 1111 1110 1111 1110 1111 1110 |
| PI=22:<br>code rate: 8/30 | 1111 1111 1111 1110 1111 1111 1111 1110 |
| PI=23:<br>code rate: 8/31 | 1111 1111 1111 1111 1111 1111 1111 1110 |
| PI=24:<br>code rate: 8/32 | 1111 1111 1111 1111 1111 1111 1111 1111 |

The last 24 bits of the serial mother codeword, $(u_{4I}, u_{4I+1}, u_{4I+2}, \ldots, u_{4I+23})$, shall be punctured using the puncturing vector given by:

$$V_T = (1100\,1100\,1100\,1100\,1100\,1100).$$

The resulting 12 bits are called tail bits.

The four punctured sub-blocks originating from each 128-bit block shall then be grouped together into a block of length $4(8+PI)$ bits. All these blocks shall then be grouped together and the tail bits shall be appended to the last block. The order of the sub-blocks and blocks shall be retained.

The resulting word is called a punctured codeword.

To ensure a word length of a multiple of 64 bits at the output of the encoding process, it shall be necessary for certain puncturing schemes to append "zero value" padding bits at the end of the punctured codeword, as specified in the following subclauses.

The resulting word is called a convolutional codeword.

## 11.2 Coding in the Fast Information Channel

This subclause defines the encoding procedure applied in the Fast Information Channel (FIC).

### 11.2.1 Transmission modes I and II

Each 768-bit vector $(a_i)_{i=0}^{767}$ at the output of the energy dispersal scrambler (see subclause 10.2) shall be processed as defined in subclause 11.1.1.

The first $4I = 3072$ bits of the serial mother codeword $\boldsymbol{U}$ shall be split into 24 consecutive blocks of 128 bits, as defined in subclause 11.1.2.

The first 21 blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI = 16$.

The remaining 3 blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI = 15$.

This corresponds to a code rate of approximately 1/3.

Finally, the last 24 bits of the serial mother codeword shall be punctured as defined in subclause 11.1.2. No padding bits shall be added.

The resulting convolutional codeword is denoted $(b_i)_{i=0}^{2303}$.

### 11.2.2 Transmission mode III

Each 1 024-bit vector $(a_i)_{i=0}^{1023}$ at the output of the energy dispersal scrambler (see subclause 10.2) shall be processed as defined in subclause 11.1.1.

The first $4I = 4096$ bits of the serial mother codeword $\boldsymbol{U}$ shall be split into 32 consecutive blocks of 128 bits, as defined in subclause 11.1.2.

The first 29 blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI = 16$.

The remaining 3 blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI = 15$.

This corresponds to a code rate of approximately 1/3.

Finally, the last 24 bits of the serial mother codeword shall be punctured as defined in subclause 11.1.2. No padding bits shall be added.

The resulting convolutional codeword is denoted $(b_i)_{i=0}^{3071}$.

### 11.3 Coding in the Main Service Channel

In this subclause the details of the puncturing procedure are specified in terms of protection profiles and protection levels. A protection profile associates the various blocks of a mother codeword with a collection of puncturing indices. For each of the allowed values of the audio or data bitrate a number of permissible protection profiles are defined. The set of protection profiles allows for audio and data broadcasting over radio frequency channels or cable networks with a level of protection suited to the requirements of the transmission channel.

Each protection profile is associated with a protection level indicating the relative level of protection provided. Protection level 1 indicates the highest level of protection. The protection levels defined in subclauses 11.3.1 and 11.3.2 are independent of each other.

#### 11.3.1 Coding for a sub-channel conveying an audio service component

Each Logical frame at the output of the energy dispersal scrambler (according to subclause 10.3) corresponding to the processing of an audio service component, consists of a $I$-bit vector $(a_i)_{i=0}^{I-1}$, where $I$ is a function of the audio bit rate. The possible values for the audio bit rate are defined in subclause 7.2.1.

Each vector $(a_i)_{i=0}^{I-1}$ shall be processed as defined in subclause 11.1.1. The first $4I$ bits of the serial mother codeword $\boldsymbol{U}$ are split into $L$ consecutive blocks of 128 bits, as defined in subclause 11.1.2.

The value of *L* for each possible audio bit rate shall comply with table 32.

**Table 32: Correspondence between the audio bit rates and the parameters *I* and *L***

| audio bit rate (kbit/s) | $I$ | $L$ |
|---|---|---|
| 32 | 768 | 24 |
| 48 | 1 152 | 36 |
| 56 | 1 344 | 42 |
| 64 | 1 536 | 48 |
| 80 | 1 920 | 60 |
| 96 | 2 304 | 72 |
| 112 | 2 688 | 84 |
| 128 | 3 072 | 96 |
| 160 | 3 840 | 120 |
| 192 | 4 608 | 144 |
| 224 | 5 376 | 168 |
| 256 | 6 144 | 192 |
| 320 | 7 680 | 240 |
| 384 | 9 216 | 288 |

The first $L_1$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_1$.

The next $L_2$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_2$.

The next $L_3$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_3$.

The remaining $L_4$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_4$.

Finally, the last 24 bits of the serial mother codeword shall be punctured as described in subclause 11.1.2.

Each quadruple $(L_1, L_2, L_3, L_4)$ associated to a quadruple $(PI_1, PI_2, PI_3, PI_4)$, defines a protection profile.

Five protection levels $P$ $(P = 1, 2, 3, 4, 5)$ are defined.

The permissible protection profiles are specified as a function of the audio bit rate and the protection level *P* in table 33 (see also table 7 of clause 6). To ensure a word length of a multiple of 64 bits at the output of the encoding process, a certain number of "zero value" padding bits shall be appended at the end of the punctured codeword for certain protection profiles, as specified in table 33.

The resulting convolutional codeword for a given Logical frame is denoted $(b_i)_{i=0}^{M-1}$.

**Table 33: Audio service component protection profiles**

| audio bit rate (kbit/s) | $P$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $PI_1$ | $PI_2$ | $PI_3$ | $PI_4$ | number of padding bits |
|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 5 | 3 | 4 | 17 | 0 | 5 | 3 | 2 | - | 0 |
| 32 | 4 | 3 | 3 | 18 | 0 | 11 | 6 | 5 | - | 0 |
| 32 | 3 | 3 | 4 | 14 | 3 | 15 | 9 | 6 | 8 | 0 |
| 32 | 2 | 3 | 4 | 14 | 3 | 22 | 13 | 8 | 13 | 0 |
| 32 | 1 | 3 | 5 | 13 | 3 | 24 | 17 | 12 | 17 | 4 |
| 48 | 5 | 4 | 3 | 26 | 3 | 5 | 4 | 2 | 3 | 0 |
| 48 | 4 | 3 | 4 | 26 | 3 | 9 | 6 | 4 | 6 | 0 |
| 48 | 3 | 3 | 4 | 26 | 3 | 15 | 10 | 6 | 9 | 4 |
| 48 | 2 | 3 | 4 | 26 | 3 | 24 | 14 | 8 | 15 | 0 |
| 48 | 1 | 3 | 5 | 25 | 3 | 24 | 18 | 13 | 18 | 0 |
| 56 | 5 | 6 | 10 | 23 | 3 | 5 | 4 | 2 | 3 | 0 |
| 56 | 4 | 6 | 10 | 23 | 3 | 9 | 6 | 4 | 5 | 0 |
| 56 | 3 | 6 | 12 | 21 | 3 | 16 | 7 | 6 | 9 | 0 |
| 56 | 2 | 6 | 10 | 23 | 3 | 23 | 13 | 8 | 13 | 8 |
| 64 | 5 | 6 | 9 | 31 | 2 | 5 | 3 | 2 | 3 | 0 |
| 64 | 4 | 6 | 9 | 33 | 0 | 11 | 6 | 5 | - | 0 |
| 64 | 3 | 6 | 12 | 27 | 3 | 16 | 8 | 6 | 9 | 0 |
| 64 | 2 | 6 | 10 | 29 | 3 | 23 | 13 | 8 | 13 | 8 |
| 64 | 1 | 6 | 11 | 28 | 3 | 24 | 18 | 12 | 18 | 4 |
| 80 | 5 | 6 | 10 | 41 | 3 | 6 | 3 | 2 | 3 | 0 |
| 80 | 4 | 6 | 10 | 41 | 3 | 11 | 6 | 5 | 6 | 0 |
| 80 | 3 | 6 | 11 | 40 | 3 | 16 | 8 | 6 | 7 | 0 |
| 80 | 2 | 6 | 10 | 41 | 3 | 23 | 13 | 8 | 13 | 8 |
| 80 | 1 | 6 | 10 | 41 | 3 | 24 | 17 | 12 | 18 | 4 |
| 96 | 5 | 7 | 9 | 53 | 3 | 5 | 4 | 2 | 4 | 0 |
| 96 | 4 | 7 | 10 | 52 | 3 | 9 | 6 | 4 | 6 | 0 |
| 96 | 3 | 6 | 12 | 51 | 3 | 16 | 9 | 6 | 10 | 4 |
| 96 | 2 | 6 | 10 | 53 | 3 | 22 | 12 | 9 | 12 | 0 |
| 96 | 1 | 6 | 13 | 50 | 3 | 24 | 18 | 13 | 19 | 0 |
| 112 | 5 | 14 | 17 | 50 | 3 | 5 | 4 | 2 | 5 | 0 |
| 112 | 4 | 11 | 21 | 49 | 3 | 9 | 6 | 4 | 8 | 0 |
| 112 | 3 | 11 | 23 | 47 | 3 | 16 | 8 | 6 | 9 | 0 |
| 112 | 2 | 11 | 21 | 49 | 3 | 23 | 12 | 9 | 14 | 4 |
| 128 | 5 | 12 | 19 | 62 | 3 | 5 | 3 | 2 | 4 | 0 |
| 128 | 4 | 11 | 21 | 61 | 3 | 11 | 6 | 5 | 7 | 0 |
| 128 | 3 | 11 | 22 | 60 | 3 | 16 | 9 | 6 | 10 | 4 |
| 128 | 2 | 11 | 21 | 61 | 3 | 22 | 12 | 9 | 14 | 0 |
| 128 | 1 | 11 | 20 | 62 | 3 | 24 | 17 | 13 | 19 | 8 |
| 160 | 5 | 11 | 19 | 87 | 3 | 5 | 4 | 2 | 4 | 0 |
| 160 | 4 | 11 | 23 | 83 | 3 | 11 | 6 | 5 | 9 | 0 |
| 160 | 3 | 11 | 24 | 82 | 3 | 16 | 8 | 6 | 11 | 0 |
| 160 | 2 | 11 | 21 | 85 | 3 | 22 | 11 | 9 | 13 | 0 |
| 160 | 1 | 11 | 22 | 84 | 3 | 24 | 18 | 12 | 19 | 0 |
| 192 | 5 | 11 | 20 | 110 | 3 | 6 | 4 | 2 | 5 | 0 |
| 192 | 4 | 11 | 22 | 108 | 3 | 10 | 6 | 4 | 9 | 0 |
| 192 | 3 | 11 | 24 | 106 | 3 | 16 | 10 | 6 | 11 | 0 |
| 192 | 2 | 11 | 20 | 110 | 3 | 22 | 13 | 9 | 13 | 8 |
| 192 | 1 | 11 | 21 | 109 | 3 | 24 | 20 | 13 | 24 | 0 |
| 224 | 5 | 12 | 22 | 131 | 3 | 8 | 6 | 2 | 6 | 4 |
| 224 | 4 | 12 | 26 | 127 | 3 | 12 | 8 | 4 | 11 | 0 |
| 224 | 3 | 11 | 20 | 134 | 3 | 16 | 10 | 7 | 9 | 0 |
| 224 | 2 | 11 | 22 | 132 | 3 | 24 | 16 | 10 | 15 | 0 |

(continued)

**Table 33: Audio service component protection profiles (concluded)**

| audio bit rate (kbit/s) | $P$ | $L_1$ $L_2$ $L_3$ $L_4$ | $PI_1$ $PI_2$ $PI_3$ $PI_4$ | number of padding bits |
|---|---|---|---|---|
| 224 | 1 | 11 24 130 3 | 24 20 12 20 | 4 |
| 256 | 5 | 11 24 154 3 | 6 5 2 5 | 0 |
| 256 | 4 | 11 24 154 3 | 12 9 5 10 | 4 |
| 256 | 3 | 11 27 151 3 | 16 10 7 10 | 0 |
| 256 | 2 | 11 22 156 3 | 24 14 10 13 | 8 |
| 256 | 1 | 11 26 152 3 | 24 19 14 18 | 4 |
| 320 | 5 | 11 26 200 3 | 8 5 2 6 | 4 |
| 320 | 4 | 11 25 201 3 | 13 9 5 10 | 8 |
| 320 | 2 | 11 26 200 3 | 24 17 9 17 | 0 |
| 384 | 5 | 11 27 247 3 | 8 6 2 7 | 0 |
| 384 | 3 | 11 24 250 3 | 16 9 7 10 | 4 |
| 384 | 1 | 12 28 245 3 | 24 20 14 23 | 8 |

Table 34 gives the approximate value of the resulting average code rate as a function of the audio bit rate and the protection level $P$. In this table, the options denoted by X are not provided.

**Table 34: Average code rate as a function of the audio bit rate and the protection level *P***

| *P* / audio bit rate (kbit/s) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 32 | 0,34 | 0,43 | 0,50 | 0,57 | 0,75 |
| 48 | 0,34 | 0,43 | 0,51 | 0,62 | 0,75 |
| 56 | X | 0,40 | 0,50 | 0,60 | 0,72 |
| 64 | 0,34 | 0,43 | 0,50 | 0,57 | 0,75 |
| 80 | 0,36 | 0,43 | 0,51 | 0,58 | 0,75 |
| 96 | 0,34 | 0,43 | 0,51 | 0,62 | 0,75 |
| 112 | X | 0,40 | 0,50 | 0,60 | 0,72 |
| 128 | 0,34 | 0,43 | 0,50 | 0,57 | 0,75 |
| 160 | 0,36 | 0,43 | 0,51 | 0,58 | 0,75 |
| 192 | 0,34 | 0,43 | 0,51 | 0,62 | 0,75 |
| 224 | 0,35 | 0,40 | 0,50 | 0,60 | 0,72 |
| 256 | 0,34 | 0,43 | 0,50 | 0,57 | 0,75 |
| 320 | X | 0,43 | X | 0,58 | 0,75 |
| 384 | 0,34 | X | 0,51 | X | 0,75 |

### 11.3.2 Coding for a sub-channel conveying data service components

#### 11.3.2.1 Equal error protection coding

Each Logical frame at the output of the energy dispersal scrambler (according to subclause 10.3), corresponding to the processing of one or more data service components (Packet mode), or a single data service component (Stream mode), consists of $I$-bit vector $(a_i)_{i=0}^{I-1}$, where $I$ is a function of the data bit rate. The possible bit rates are multiples of 8 kbit/s as specified in subclauses 5.3.1 and 5.3.2.

Each vector $(a_i)_{i=0}^{I-1}$ shall be processed as defined in subclause 11.1.1. The first $4I$ bits of the serial mother codeword $\boldsymbol{U}$ are split into $L$ consecutive blocks of 128 bits, as defined in subclause 11.1.2.

The value of $L$ for each possible bit rate shall comply with table 35.

**Table 35: Correspondence between the possible bit rates and the parameters *I* and *L***

| Data bit rate (kbit/s) | $I$ | $L$ |
|---|---|---|
| 8 | 192 | 6 |
| : | : | : |
| : | : | : |
| 8n | 192n | 6n |
| : | : | : |
| : | : | : |
| 1 728 | 41 472 | 1 296 |

The first $L_1$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_1$.

The remaining $L_2$ blocks shall be punctured as defined in subclause 11.1.2, according to the puncturing index $PI_2$.

Finally, the last 24 bits of the serial mother codeword shall be punctured as defined in subclause 11.1.2. No padding bits shall be added.

Each pair $(L_1, L_2)$ associated to a pair $(PI_1, PI_2)$ defines a protection profile.

Four protection levels are defined. These four protection levels *P*(*P* = 1, 2, 3, 4), correspond to the code rates 1/4, 3/8, 1/2 and 3/4 respectively.

The permissible protection profiles are specified as a function of the bit rate, and the protection level $P$, in table 36.

The resulting convolutional codeword for a given Logical frame is denoted $(b_i)_{i=0}^{M-1}$.

**Table 36: Equal error protection profile**

| data bit rate (kbit/s) | $P$ | $L_1$ | $L_2$ | $PI_1$ | $PI_2$ |
|---|---|---|---|---|---|
| 8n | 4 | 4n-3 | 2n+3 | 3 | 2 |
| 8n | 3 | 6n-3 | 3 | 8 | 7 |
| 8<br>8n (n>1) | 2 | 5<br>2n-3 | 1<br>4n+3 | 13<br>14 | 12<br>13 |
| 8n | 1 | 6n-3 | 3 | 24 | 23 |

### 11.3.2.2 Future error protection coding

There is provision for new error protection coding schemes to be applied to Stream mode data, as indicated in subclause 6.2. This may be required for new applications different from those defined in clauses 7 and 8. However, the encoding procedures specified in subclause 11.1 shall apply.

## 12 Time interleaving

Time interleaving shall be applied to the output of each convolutional encoder for all sub-channels of the Main Service Channel (MSC). It shall not be applied to the FIC.

The output of each individual convolutional encoder is a sequence of convolutional codewords denoted $\left(\boldsymbol{B_r}\right)_{r=-\infty}^{\infty}$ where $r$ is defined as the time index ranging from $-\infty$ to $+\infty$, whose value taken modulo 5 000 (mod($r$,5 000)) is equal to the Logical frame count of the corresponding Logical frame. The time index is introduced for the purpose of describing system properties over several Logical frames independently of the absolute time. It is notionally related to the logical frame count in such a way that they are aligned at r equals 0.

The convolutional codeword $\boldsymbol{B_r}$ is denoted as a vector:

$$\boldsymbol{B_r} = \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,M_r-1}\right)$$

of length $M_r$ consisting of bits $b_{r,i_r}$.

The length $M_r$ of these vectors depends on the index $r$, because it may change from one logical frame to the next, if a multiplex re-configuration occurs.

However, due to the limitation on the minimum time between two successive multiplex re-configurations (see clause 6), the value of $M_r$ is not allowed to change more than once during a period of 250 Logical frames.

The output of the interleaver is denoted as a sequence $\left(\boldsymbol{C_r}\right)_{r=-\infty}^{\infty}$ of vectors,

$$\boldsymbol{C_r} = \left(c_{r,0}, c_{r,1}, \ldots, c_{r,i_r}, \ldots, c_{r,N_r-1}\right)$$

of length $N_r$ consisting of bits $c_{r,i_r}$. The sequence of vectors $\left(\boldsymbol{C_r}\right)_{r=-\infty}^{\infty}$ constitutes the content of a sub-channel.

As long as the multiplex configuration remains stable, the length of the vector $\boldsymbol{C_r}$ shall be equal to the length of the vector $\boldsymbol{B_r}$ , i.e. $N_r = M_r$. During a time period of 15 Logical frames after a multiplex re-configuration event, the length $N_r$ may be larger than $M_r$ for some sub-channels. The relation between $N_r$ and $M_r$ is defined later in this clause.

The time interleaving shall be performed according to the following relation:

$$c_{r,i_r} = \begin{cases} b_{r',i_r} & \text{if} \quad i_r \le M_{r'} - 1 \\ 0 & \text{if not} \end{cases}$$

for $i_r = 0, 1, 2, \ldots, N_r - 1$ and all integers $r$.

The relationship between the indices $r'$, $r$ and $i_r$ is specified in table 37, where $r'$ is given as a function of $r$ for each of the possible values of $i_r$ modulo 16.

**Table 37: Relationship between the indices $r'$, $r$ and $i_r$**

| $R(i_r/16)$ | $r'(r, i_r)$ |
|---|---|
| 0 | $r$ |
| 1 | $r$-8 |
| 2 | $r$-4 |
| 3 | $r$-12 |
| 4 | $r$-2 |
| 5 | $r$-10 |
| 6 | $r$-6 |
| 7 | $r$-14 |
| 8 | $r$-1 |
| 9 | $r$-9 |
| 10 | $r$-5 |
| 11 | $r$-13 |
| 12 | $r$-3 |
| 13 | $r$-11 |
| 14 | $r$-7 |
| 15 | $r$-15 |

The following shall apply for the relationship between the values of $M_r$ and $N_r$.

a) If $M_r$ has been constant over the last 16 frames, i.e:

$$M_r = M_{r-1} = \cdots = M_{r-15};$$

then

$$N_r = M_r.$$

b) If $M_r$ has increased during this period, i.e:

$$M_r > M_{r-15};$$

then

$$N_r = M_r.$$

c) If $M_r$ has decreased over the last 16 frames, i.e:

$$M_r < M_{r-15};$$

then

$$N_r = M_{r-15}.$$

NOTE: There is a one-to-one correspondence between $M_r$ and $N_r$ which may be expressed by the following two equivalent equations:

$$N_r = \max(M_r, M_{r-15});$$

or equivalently

$$M_r = \min(N_r, N_{r+15}).$$

The time interleaving rule is illustrated by the three following examples:

**Example 1:** Time interleaving of a data service component with a bit rate of 8 kbit/s protected by protection level 4.

It is supposed that no multiplex re-configuration occurs.

The convolutional codeword $\boldsymbol{B_r}$ is:

$$\boldsymbol{B_r} = \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,255}\right).$$

In this example, $M_r = N_r = 256$ for all values of $r$.

Figure 84 illustrates the time interleaving rule.

**Example 2:** Time interleaving of a data service component with a bit rate of 8 kbit/s protected by protection level 4, and re-configured to a data service component with a bit rate of 8 kbit/s protected by protection level 3.

The convolutional codeword $\boldsymbol{B_r}$ is:

$$\mathbf{B_r} = \begin{cases} \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,255}\right) \text{ for } r \le r_0 - 1 \\ \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,383}\right) \text{ for } r \ge r_0 \end{cases}$$

In this example, $M_r = \begin{cases} 256 & \text{for } r \le r_0 - 1 \\ 384 & \text{for } r \ge r_0 \end{cases}$

The value of $N_r$ is given by:

$$N_r = \begin{cases} 256 & \text{for } r \le r_0 - 1 \\ 384 & \text{for } r \ge r_0 \end{cases}$$

Figure 84 illustrates the time interleaving rule.

**Example 3:** Time interleaving of a data service component with a bit rate of 8 kbit/s protected by protection level 3, and re-configured to a data service component with a bit rate of 8 kbit/s protected by protection level 4.

The convolutional codeword $\boldsymbol{B_r}$ is:

$$\mathbf{B_r} = \begin{cases} \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,383}\right) \text{ for } r \le r_0 - 1 \\ \left(b_{r,0}, b_{r,1}, \ldots, b_{r,i_r}, \ldots, b_{r,255}\right) \text{ for } r \ge r_0 \end{cases}$$

In this example, $M_r = \begin{cases} 384 & \text{for } r \le r_0 - 1 \\ 256 & \text{for } r \ge r_0 \end{cases}$

The value of $N_r$ is given by:

$$N_r = \begin{cases} 384 & \text{for } r \le r_0 + 14 \\ 256 & \text{for } r \ge r_0 + 15 \end{cases}$$

Figure 86 illustrates the time interleaving rule.

Time interleaver input

| . | r-1 | r |
|---|---|---|
| . | $a_{r-1,0}$ | $a_{r,0}$ |
| . | $a_{r-1,1}$ | $a_{r,1}$ |
| . | $a_{r-1,2}$ | $a_{r,2}$ |
| . | $a_{r-1,3}$ | $a_{r,3}$ |
| . | $a_{r-1,4}$ | $a_{r,4}$ |
| . | $a_{r-1,5}$ | $a_{r,5}$ |
| . | $a_{r-1,6}$ | $a_{r,6}$ |
| . | $a_{r-1,7}$ | $a_{r,7}$ |
| . | $a_{r-1,8}$ | $a_{r,8}$ |
| . | $a_{r-1,9}$ | $a_{r,9}$ |
| . | $a_{r-1,10}$ | $a_{r,10}$ |
| . | $a_{r-1,11}$ | $a_{r,11}$ |
| . | $a_{r-1,12}$ | $a_{r,12}$ |
| . | $a_{r-1,13}$ | $a_{r,13}$ |
| . | $a_{r-1,14}$ | $a_{r,14}$ |
| . | $a_{r-1,15}$ | $a_{r,15}$ |
| . | $a_{r-1,16}$ | $a_{r,16}$ |
| . | $a_{r-1,17}$ | $a_{r,17}$ |
| . | $a_{r-1,18}$ | $a_{r,18}$ |
| . | .... | .... |
| . | .... | .... |
| . | .... | .... |
| . | $a_{r-1,255}$ | $a_{r,255}$ |

Time interleaver output

time →

| r | r+1 | r+2 | r+3 | r+4 | r+5 | r+6 | r+7 | r+8 | r+9 | r+10 | r+11 | r+12 | r+13 | r+14 | r+15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_{r,0}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-8,1}$ | $a_{r-7,1}$ | $a_{r-6,1}$ | $a_{r-5,1}$ | $a_{r-4,1}$ | $a_{r-3,1}$ | $a_{r-2,1}$ | $a_{r-1,1}$ | $a_{r,1}$ | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-4,2}$ | $a_{r-3,2}$ | $a_{r-2,2}$ | $a_{r-1,2}$ | $a_{r,2}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-12,3}$ | $a_{r-11,3}$ | $a_{r-10,3}$ | $a_{r-9,3}$ | $a_{r-8,3}$ | $a_{r-7,3}$ | $a_{r-6,3}$ | $a_{r-5,3}$ | $a_{r-4,3}$ | $a_{r-3,3}$ | $a_{r-2,3}$ | $a_{r-1,3}$ | $a_{r,3}$ | .... | .... | .... |
| $a_{r-2,4}$ | $a_{r-1,4}$ | $a_{r,4}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-10,5}$ | $a_{r-9,5}$ | $a_{r-8,5}$ | $a_{r-7,5}$ | $a_{r-6,5}$ | $a_{r-5,5}$ | $a_{r-4,5}$ | $a_{r-3,5}$ | $a_{r-2,5}$ | $a_{r-1,5}$ | $a_{r,5}$ | .... | .... | .... | .... | .... |
| $a_{r-6,6}$ | $a_{r-5,6}$ | $a_{r-4,6}$ | $a_{r-3,6}$ | $a_{r-2,6}$ | $a_{r-1,6}$ | $a_{r,6}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-14,7}$ | $a_{r-13,7}$ | $a_{r-12,7}$ | $a_{r-11,7}$ | $a_{r-10,7}$ | $a_{r-9,7}$ | $a_{r-8,7}$ | $a_{r-7,7}$ | $a_{r-6,7}$ | $a_{r-5,7}$ | $a_{r-4,7}$ | $a_{r-3,7}$ | $a_{r-2,7}$ | $a_{r-1,7}$ | $a_{r,7}$ | .... |
| $a_{r-1,8}$ | $a_{r,8}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-9,9}$ | $a_{r-8,9}$ | $a_{r-7,9}$ | $a_{r-6,9}$ | $a_{r-5,9}$ | $a_{r-4,9}$ | $a_{r-3,9}$ | $a_{r-2,9}$ | $a_{r-1,9}$ | $a_{r,9}$ | .... | .... | .... | .... | .... | .... |
| $a_{r-5,10}$ | $a_{r-4,10}$ | $a_{r-3,10}$ | $a_{r-2,10}$ | $a_{r-1,10}$ | $a_{r,10}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-13,11}$ | $a_{r-12,11}$ | $a_{r-11,11}$ | $a_{r-10,11}$ | $a_{r-9,11}$ | $a_{r-8,11}$ | $a_{r-7,11}$ | $a_{r-6,11}$ | $a_{r-5,11}$ | $a_{r-4,11}$ | $a_{r-3,11}$ | $a_{r-2,11}$ | $a_{r-1,11}$ | $a_{r,11}$ | .... | .... |
| $a_{r-3,12}$ | $a_{r-2,12}$ | $a_{r-1,12}$ | $a_{r,12}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-11,13}$ | $a_{r-10,13}$ | $a_{r-9,13}$ | $a_{r-8,13}$ | $a_{r-7,13}$ | $a_{r-6,13}$ | $a_{r-5,13}$ | $a_{r-4,13}$ | $a_{r-3,13}$ | $a_{r-2,13}$ | $a_{r-1,13}$ | $a_{r,13}$ | .... | .... | .... | .... |
| $a_{r-7,14}$ | $a_{r-6,14}$ | $a_{r-5,14}$ | $a_{r-4,14}$ | $a_{r-3,14}$ | $a_{r-2,14}$ | $a_{r-1,14}$ | $a_{r,14}$ | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-15,15}$ | $a_{r-14,15}$ | $a_{r-13,15}$ | $a_{r-12,15}$ | $a_{r-11,15}$ | $a_{r-10,15}$ | $a_{r-9,15}$ | $a_{r-8,15}$ | $a_{r-7,15}$ | $a_{r-6,15}$ | $a_{r-5,15}$ | $a_{r-4,15}$ | $a_{r-3,15}$ | $a_{r-2,15}$ | $a_{r-1,15}$ | $a_{r,15}$ |
| $a_{r,16}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-8,17}$ | $a_{r-7,17}$ | $a_{r-6,17}$ | $a_{r-5,17}$ | $a_{r-4,17}$ | $a_{r-3,17}$ | $a_{r-2,17}$ | $a_{r-1,17}$ | $a_{r,17}$ | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-4,18}$ | $a_{r-3,18}$ | $a_{r-2,18}$ | $a_{r-1,18}$ | $a_{r,18}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| $a_{r-15,255}$ | $a_{r-14,255}$ | $a_{r-13,255}$ | $a_{r-12,255}$ | $a_{r-11,255}$ | $a_{r-10,255}$ | $a_{r-9,255}$ | $a_{r-8,255}$ | $a_{r-7,255}$ | $a_{r-6,255}$ | $a_{r-5,255}$ | $a_{r-4,255}$ | $a_{r-3,255}$ | $a_{r-2,255}$ | $a_{r-1,255}$ | $a_{r,255}$ |

**Figure 84: Illustration of time interleaving for example 1**

Time interleaver input                                    time →

| $r_0$-3 | $r_0$-2 | $r_0$-1 | $r_0$ | $r_0$+1 | $r_0$+2 | $r_0$+3 | $r_0$+4 | $r_0$+5 | $r_0$+6 | $r_0$+7 | $r_0$+8 | $r_0$+9 | $r_0$+10 | $r_0$+11 | $r_0$+12 | $r_0$+13 | $r_0$+14 | $r_0$+15 | $r_0$+16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .... | .... | $a_{r0-1,0}$ | $a_{r0,0}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-1,1}$ | $a_{r0,1}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-1,255}$ | $a_{r0,255}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | $a_{r0,383}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |

Time interleaver output

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .... | .... | $a_{r0-1,0}$ | $a_{r0,0}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+14,0}$ | $a_{r0+15,0}$ | .... |
| .... | .... | $a_{r0-9,1}$ | $a_{r0-8,1}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+6,1}$ | $a_{r0+7,1}$ | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-16,255}$ | $a_{r0-15,255}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0-1,255}$ | $a_{r0,255}$ | .... |
| | | | $a_{r0,256}$ | $a_{r0+1,256}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+14,256}$ | $a_{r0+15,256}$ | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+6,257}$ | $a_{r0+7,257}$ | .... |
| | | | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0,263}$ | $a_{r0+1,263}$ | .... |
| | | | 0 | $a_{r0,264}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+13,264}$ | $a_{r0+14,264}$ | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+5,265}$ | $a_{r0+6,265}$ | .... |
| | | | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | $a_{r0,271}$ | .... |
| | | | $a_{r0,272}$ | $a_{r0+1,272}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+14,272}$ | $a_{r0+15,272}$ | .... |
| | | | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | 0 | $a_{r0,376}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+13,376}$ | $a_{r0+14,376}$ | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+5,377}$ | $a_{r0+6,377}$ | .... |
| | | | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+7,382}$ | $a_{r0+8,382}$ | .... |
| | | | 0 | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | $a_{r0,383}$ | .... |

**Figure 85: Illustration of time interleaving for example 2**

Time interleaver input                                    time →

| $r_0$-3 | $r_0$-2 | $r_0$-1 | $r_0$ | $r_0$+1 | $r_0$+2 | $r_0$+3 | $r_0$+4 | $r_0$+5 | $r_0$+6 | $r_0$+7 | $r_0$+8 | $r_0$+9 | $r_0$+10 | $r_0$+11 | $r_0$+12 | $r_0$+13 | $r_0$+14 | $r_0$+15 | $r_0$+16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .... | .... | $a_{r0-1.0}$ | $a_{r0.0}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-1.1}$ | $a_{r0.1}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-1.255}$ | $a_{r0.255}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | .... | | | | | | | | | | | | | | | | | |
| .... | .... | $a_{r0-1.383}$ | | | | | | | | | | | | | | | | | |

Time interleaver output

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .... | .... | $a_{r0-1.0}$ | $a_{r0.0}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+13.0}$ | $a_{r0+14.0}$ | $a_{r0+15.0}$ | .... |
| .... | .... | $a_{r0-9.1}$ | $a_{r0-8.1}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0+5.1}$ | $a_{r0+6.1}$ | $a_{r0+7.1}$ | .... |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... |
| .... | .... | $a_{r0-16.255}$ | $a_{r0-15.255}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0-2.255}$ | $a_{r0-1.255}$ | $a_{r0.255}$ | .... |
| .... | .... | $a_{r0-1.256}$ | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | $a_{r0-9.257}$ | $a_{r0-8.257}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | | |
| .... | .... | $a_{r0-15.263}$ | $a_{r0-14.263}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0-1.263}$ | 0 | | |
| .... | .... | $a_{r0-2.264}$ | $a_{r0-1.264}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | | |
| .... | .... | $a_{r0-16.271}$ | $a_{r0-15.271}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0-2.271}$ | $a_{r0-1.271}$ | | |
| .... | .... | $a_{r0-1.272}$ | 0 | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | $a_{r0-9.273}$ | $a_{r0-8.273}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | 0 | 0 | | |
| .... | .... | $a_{r0-16.383}$ | $a_{r0-15.383}$ | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | .... | $a_{r0-2.383}$ | $a_{r0-1.383}$ | | |

**Figure 86: Illustration of time interleaving for example 3**

# 13 Common Interleaved Frame

This clause specifies the bit structure of the Common Interleaved Frame (CIF). Each vector $\boldsymbol{C_r}$ at the output of a time interleaver (specified in clause 12) contains a multiple of 64 bits, and therefore, every sub-channel occupies an integral number of CUs in the CIF.

Using the notation of clause 12, the required number of CUs for a sub-channel during the CIF of time index *r*, is equal to $N_r/64$.

The vectors $\boldsymbol{C_r}$ for the various sub-channels shall be multiplexed in such a way that every sub-channel shall occupy an integral number of consecutive CUs.

The address of the CU assigned to the first bit of a vector $\boldsymbol{C_r}$ is called the start address.

The CIF bits shall be assigned consecutively so that the first bit of each vector $\boldsymbol{C_r}$ at the output of a time interleaver shall be assigned to the first bit of the CU of start address, and the last bit of each vector $\boldsymbol{C_r}$ shall be assigned to the last bit of the last CU assigned to that sub-channel.

If the set of sub-channels do not fill the whole CIF, all unassigned CUs shall be filled with padding bits. The value of the padding bits shall be defined as follows:

If the $(i+1)^{th}$ bit of the CIF belongs to a CU containing padding bits, it shall take the value of the $(i+1)^{th}$ bit of the PRBS defined in clause 10.

These rules are illustrated in figure 87.

| sub-channel SubChId = p | sub-channel SubChId = m | Padding | sub-channel SubChId = k | sub-channel SubChId = l | Padding |
|---|---|---|---|---|---|

CIF

| 0 | 1 | 2 | .. | 34 | ........................ | .............. | .................. | ................... | 862 | 863 |
|---|---|---|---|---|---|---|---|---|---|---|

CU addresses

**Figure 87: Example of a CIF structure**

The $(i+1)^{th}$ bit of the CIF of index r shall be denoted by $d_{r,i}$ (*i=0, 1, 2, ..., 55 295*).

The index *r* taken modulo 5 000 (mod(*r*,5 000)) is equal to the CIF count defined in subclause 5.3.

The structure of the CIF is signalled by the MCI, as defined in subclause 6.2.

# 14 DAB transmission signal

## 14.1 General principles

The transmitted signal is built up around a transmission frame structure corresponding to the juxtaposition in time of the Synchronization channel, the FIC and the MSC (see also subclause 5.1).

The transmission frame duration is denoted by $T_F$.

The structure of the transmission frame is shown in figure 88.

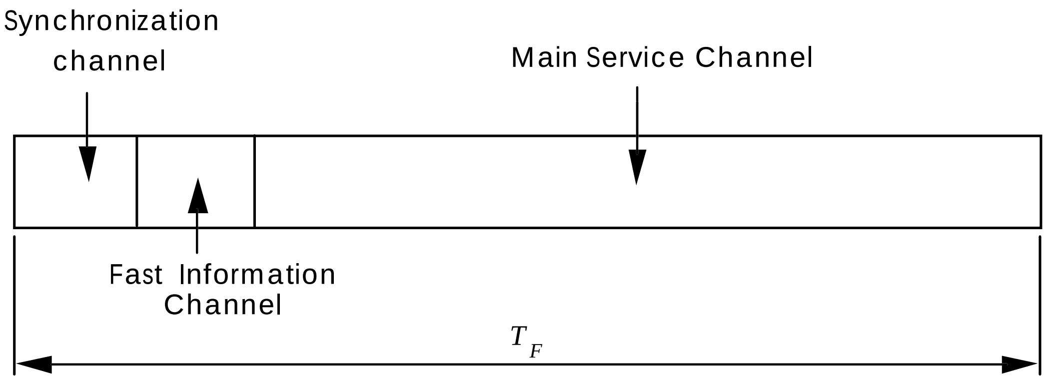


**Figure 88: Transmission frame structure**

Three transmission modes are defined, each having its particular set of parameters. The use of these transmission modes depends on the network configuration and operating frequencies, as defined in clause 15.

Each transmission frame is divided into a sequence of OFDM symbols, each symbol consisting of a number of carriers. The DAB transmission signal is defined as the sum of two signals; the main signal $s(t)$ and an optional signal $s_{TII}(t)$ as illustrated in figure 1. Subclause 14.2 define the main signal. The subclauses 14.3 to 14.7 specify the content of the Synchronization Channel as well as the processing applied to the convolutionally encoded FIBs and the CIFs in order to generate the main signal $s(t)$. Subclause 14.8 specifies the generation of the signal $s_{TII}(t)$.

## 14.2 Structure of the main signal

Each transmission frame shall consist of consecutive OFDM symbols. The number of OFDM symbols in a transmission frame is dependent on the transmission mode. The Synchronization channel in any transmission mode shall occupy the first two OFDM symbols of each transmission frame.

The first OFDM symbol of the transmission frame shall be the Null symbol of duration $T_{NULL}$. The remaining part of the transmission frame shall be a juxtaposition of OFDM symbols of duration $T_S$.

Each of these OFDM symbols shall consist of a set of equally-spaced carriers, with a carrier spacing equal to $1/T_u$ . The main signal $s(t)$ shall be defined using the following formula:

$$s(t) = \mathrm{Re}\left\{ e^{2j\pi f_c t} \sum_{m=-\infty}^{+\infty} \sum_{l=0}^{L} \sum_{k=-K/2}^{K/2} z_{m,l,k} \cdot g_{k,l}\left(t - mT_F - T_{NULL} - (l-1)T_S\right) \right\}$$

with,

$$g_{k,l}(t) = \begin{cases} 0 & \text{for} \quad l = 0 \\ e^{2j\pi k(t-\Delta)/T_U} \cdot \mathrm{Rect}\left(t/T_S\right) & \text{for} \quad l = 1,2,\ldots,L \end{cases}$$

and $T_S = T_U + \Delta$.

The various parameters and variables are defined as follows:

$L$ is the number of OFDM symbols per transmission frame (the Null symbol being excluded);

$K$ is the number of transmitted carriers;

$T_F$ is the transmission frame duration;

$T_{NULL}$ is the Null symbol duration;

$T_S$ is the duration of OFDM symbols of indices $l=1, 2, 3,..., L$;

$T_U$ is the inverse of the carrier spacing;

$\Delta$ is the duration of the time interval called guard interval;

$z_{m,l,k}$ is the complex D-QPSK symbol associated to carrier $k$ of OFDM symbol $l$ during transmission frame $m$. Its values are defined in the following subclauses. For $k = 0$, $z_{m,l,k} = 0$, so that the central carrier is not transmitted;

$f_c$ is the central frequency of the signal. The possible values of $f_c$ are given in clause 15.

These parameters are specified in table 38 for transmission modes I, II and III. The values of the various time-related parameters are given in multiples of the elementary period T = 1/2 048 000 seconds, and approximately in seconds.

**Table 38: Definition of the parameters for transmission modes I, II and III**

| Parameter | Transmission mode I | Transmission mode II | Transmission mode III |
|---|---|---|---|
| $L$ | *76* | *76* | *153* |
| $K$ | *1 536* | *384* | *192* |
| $T_F$ | *196 608 T*<br>*96 ms* | *49 152 T*<br>*24 ms* | *49 152 T*<br>*24 ms* |
| $T_{NULL}$ | *2 656 T*<br>*~1,297 ms* | *664 T*<br>*~324 µs* | *345 T*<br>*~168 µs* |
| $T_S$ | *2 552 T*<br>*~1,246 ms* | *638 T*<br>*~312 µs* | *319 T*<br>*~156 µs* |
| $T_u$ | *2 048 T*<br>*1 ms* | *512 T*<br>*250 µs* | *256 T*<br>*125 µs* |
| $\Delta$ | *504 T*<br>*~246 µs* | *126 T*<br>*~62 µs* | *63 T*<br>*~31 µs* |

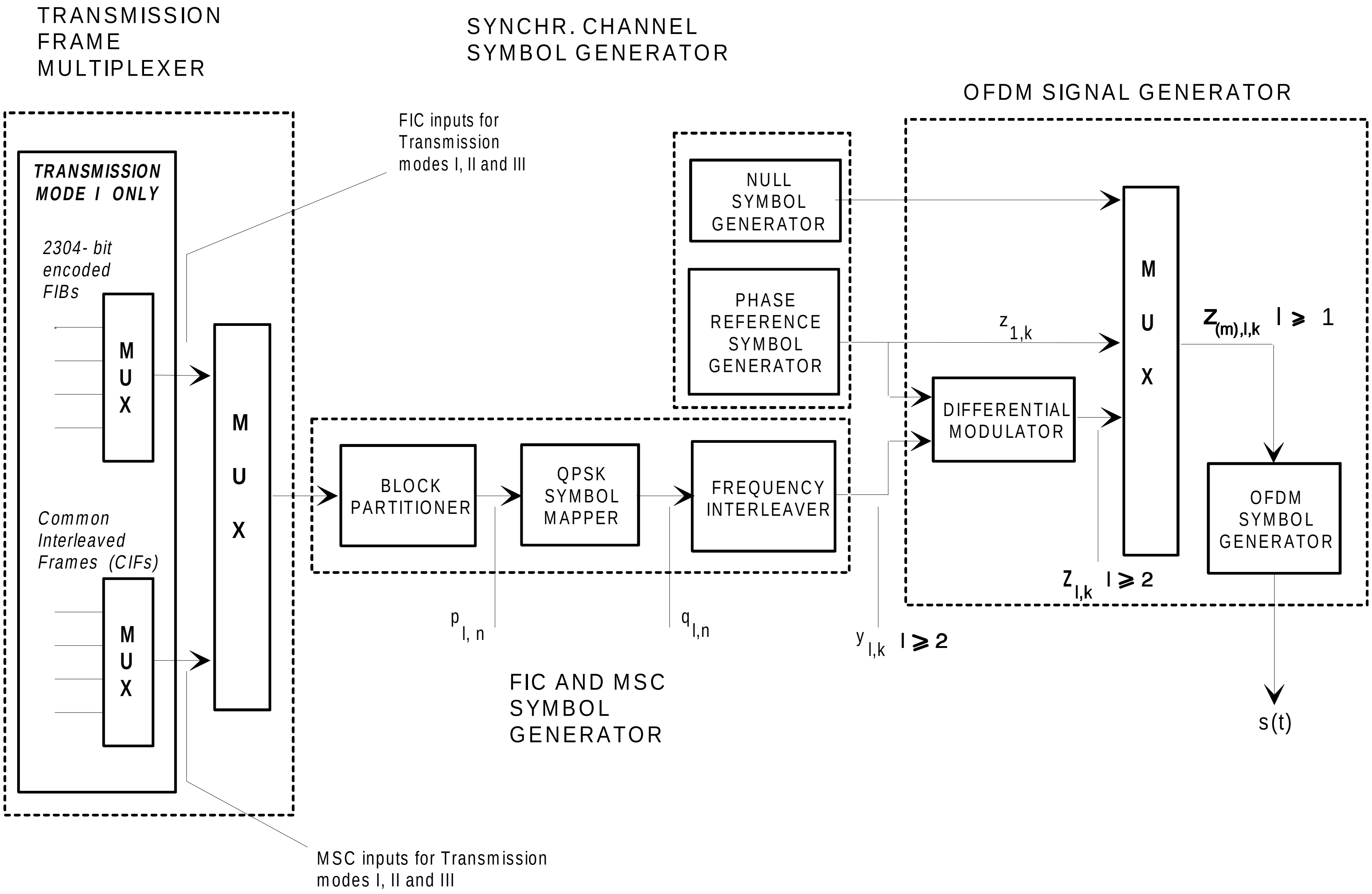


**Figure 89: Generation of the main signal**

Subclauses 14.4, 14.5, 14.6, and 14.7 describe the generation of the complex D-QPSK symbols $z_{m, l, k}$ constituting a transmission frame. Since the same processing is applied for every transmission frame, the index $m$ will be suppressed when appropriate.

The generation of the main signal $s(t)$ is described in the conceptual block diagram of figure 89.

### 14.3 Synchronization channel

This subclause specifies the characteristics of the synchronization channel which consists of the first two ODFM symbols of each transmission frame. This channel may also be used for carrying optional Transmitter Identification Information (TII) (see subclause 14.8), by adding to the Null symbol defined in subclause 14.2, the TII signal.

#### 14.3.1 Null symbol

As previously described, the first OFDM symbol of the transmission frame is the Null symbol. During the time interval $[0, T_{NULL}[$, the main signal $s(t)$ shall be equal to 0.

#### 14.3.2 Phase reference symbol

The second OFDM symbol of the transmission frame is the phase reference symbol. It constitutes the reference for the differential modulation for the next OFDM symbol. The Phase reference symbol is defined by the values of $z_{l, k}$ for $l = 1$:

$$z_{1,k} = \begin{cases} e^{j\varphi_k} & for\ -\frac{K}{2} \le k < 0\ and\ 0 < k \le \frac{K}{2} \\ 0 & for\ k = 0 \end{cases}$$

The values of $\varphi_k$ shall be obtained from the following formula:

$$\varphi_k = \frac{\pi}{2}\left(h_{i,k-k'} + n\right)$$

The indices $i$, $k'$ and the parameter $n$ are specified as functions of the carrier index $k$ for the three transmission modes in tables 40, 41 and 42.

The values of the parameter $h_{i, j}$ as a function of its indices $i$ and $j$, are specified in table 42.

**Table 39: Relation between the indices *i*, *k'* and *n* and the carrier index *k* for transmission mode I**

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| - 768 | - 737 | -768 | 0 | 1 |
| - 736 | - 705 | -736 | 1 | 2 |
| - 704 | - 673 | -704 | 2 | 0 |
| - 672 | - 641 | -672 | 3 | 1 |
| - 640 | - 609 | -640 | 0 | 3 |
| - 608 | - 577 | -608 | 1 | 2 |
| - 576 | - 545 | -576 | 2 | 2 |
| - 544 | - 513 | -544 | 3 | 3 |
| - 512 | - 481 | -512 | 0 | 2 |
| - 480 | - 449 | -480 | 1 | 1 |
| - 448 | - 417 | -448 | 2 | 2 |
| - 416 | - 385 | -416 | 3 | 3 |
| - 384 | - 353 | -384 | 0 | 1 |
| - 352 | - 321 | -352 | 1 | 2 |
| - 320 | - 289 | -320 | 2 | 3 |
| - 288 | - 257 | -288 | 3 | 3 |
| - 256 | - 225 | -256 | 0 | 2 |
| - 224 | - 193 | -224 | 1 | 2 |
| - 192 | - 161 | -192 | 2 | 2 |
| - 160 | - 129 | -160 | 3 | 1 |
| - 128 | - 97 | -128 | 0 | 1 |
| - 96 | - 65 | -96 | 1 | 3 |
| - 64 | - 33 | -64 | 2 | 1 |
| - 32 | - 1 | -32 | 3 | 2 |

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| 1 | 32 | 1 | 0 | 3 |
| 33 | 64 | 33 | 3 | 1 |
| 65 | 96 | 65 | 2 | 1 |
| 97 | 128 | 97 | 1 | 1 |
| 129 | 160 | 129 | 0 | 2 |
| 161 | 192 | 161 | 3 | 2 |
| 193 | 224 | 193 | 2 | 1 |
| 225 | 256 | 225 | 1 | 0 |
| 257 | 288 | 257 | 0 | 2 |
| 289 | 320 | 289 | 3 | 2 |
| 321 | 352 | 321 | 2 | 3 |
| 353 | 384 | 353 | 1 | 3 |
| 385 | 416 | 385 | 0 | 0 |
| 417 | 448 | 417 | 3 | 2 |
| 449 | 480 | 449 | 2 | 1 |
| 481 | 512 | 481 | 1 | 3 |
| 513 | 544 | 513 | 0 | 3 |
| 545 | 576 | 545 | 3 | 3 |
| 577 | 608 | 577 | 2 | 3 |
| 609 | 640 | 609 | 1 | 0 |
| 641 | 672 | 641 | 0 | 3 |
| 673 | 704 | 673 | 3 | 0 |
| 705 | 736 | 705 | 2 | 1 |
| 737 | 768 | 737 | 1 | 1 |

**Table 40: Relation between the indices *i*, *k'* and *n* and the carrier index *k* for transmission mode II**

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| - 192 | - 161 | - 192 | 0 | 2 |
| - 160 | - 129 | - 160 | 1 | 3 |
| - 128 | - 97 | - 128 | 2 | 2 |
| - 96 | - 65 | - 96 | 3 | 2 |
| - 64 | - 33 | - 64 | 0 | 1 |
| - 32 | - 1 | - 32 | 1 | 2 |

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| 1 | 32 | 1 | 2 | 0 |
| 33 | 64 | 33 | 1 | 2 |
| 65 | 96 | 65 | 0 | 2 |
| 97 | 128 | 97 | 3 | 1 |
| 129 | 160 | 129 | 2 | 0 |
| 161 | 192 | 161 | 1 | 3 |

**Table 41: Relation between the indices *i*, *k'* and *n* and the carrier index *k* for transmission mode III**

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| - 96 | - 65 | - 96 | 0 | 2 |
| - 64 | - 33 | - 64 | 1 | 3 |
| - 32 | - 1 | - 32 | 2 | 0 |

| *k* in the range of min | max | *k'* | *i* | *n* |
|---|---|---|---|---|
| 1 | 32 | 1 | 3 | 2 |
| 33 | 64 | 33 | 2 | 2 |
| 65 | 96 | 65 | 1 | 2 |

**Table 42: Time-Frequency-Phase parameter *h* values**

| $j$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $h_{0,j}$ | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 2 | 2 | 1 | 1 |
| $h_{1,j}$ | 0 | 3 | 2 | 3 | 0 | 1 | 3 | 0 | 2 | 1 | 2 | 3 | 2 | 3 | 3 | 0 |
| $h_{2,j}$ | 0 | 0 | 0 | 2 | 0 | 2 | 1 | 3 | 2 | 2 | 0 | 2 | 2 | 0 | 1 | 3 |
| $h_{3,j}$ | 0 | 1 | 2 | 1 | 0 | 3 | 3 | 2 | 2 | 3 | 2 | 1 | 2 | 1 | 3 | 2 |

| $j$ | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $h_{0,j}$ | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 2 | 2 | 1 | 1 |
| $h_{1,j}$ | 0 | 3 | 2 | 3 | 0 | 1 | 3 | 0 | 2 | 1 | 2 | 3 | 2 | 3 | 3 | 0 |
| $h_{2,j}$ | 0 | 0 | 0 | 2 | 0 | 2 | 1 | 3 | 2 | 2 | 0 | 2 | 2 | 0 | 1 | 3 |
| $h_{3,j}$ | 0 | 1 | 2 | 1 | 0 | 3 | 3 | 2 | 2 | 3 | 2 | 1 | 2 | 1 | 3 | 2 |

### 14.3.3 Time reference

The synchronization channel shall serve as the reference for the time information carried in the FIC (see subclause 8.1.3). The time information carried in the FIC shall be taken to be the time of transmission of the start of the Null symbol in the transmission frame carrying the time information.

## 14.4 Block partitioning and association of blocks to OFDM symbols

This subclause defines the process applied to the sequence of convolutionally encoded FIBs and to the sequence of CIFs, to constitute the blocks of data which will be associated to OFDM symbols. This process is transmission mode dependent.

### 14.4.1 Block partitioning and association of blocks to OFDM symbols in the Fast Information Channel

#### 14.4.1.1 Transmission mode I

In transmission mode I, four groups of convolutionally encoded FIBs shall be transmitted within each transmission frame, as indicated in subclause 5.1.

Four convolutional codewords, defined in subclause 11.2.1 corresponding to four consecutive groups of FIBs shall be multiplexed to form a vector, which shall then be divided into three blocks to be transmitted on three OFDM symbols.

The bits $b_i$ of the convolutional codeword $(b_i)_{i=0}^{2303}$ defined in subclause 11.2.1 will, in this subclause, be indexed by the time index $r$, and will be denoted $b_{r,i}$. The index $r$ is defined in such a way that its value modulo 5 000 (mod($r$,5 000)) is equal to the CIF count defined in subclause 5.3. This relationship follows from the association of FIBs to CIFs, see subclause 5.1.

The multiplexing of four consecutive convolutional codewords into one vector $\boldsymbol{B'}$ is defined by the following relation:

$$b'_{i'} = b_{r,i}$$

and

$$i' = i + 2304 \cdot \mathrm{mod}(r,4) \qquad \text{for } i = 0,1,2,\ldots,2303 \text{ and for any value of } r.$$

where:

$b'_{i'}$ denotes the $(i'+1)^{th}$ bit of the vector $B'$;

$b_{r,\,i}$ denotes the $(i+1)^{th}$ bit of the $(r+1)^{th}$ convolutional codeword.

This means that the arrangement of convolutionally encoded FIBs in a transmission frame shall be such that convolutionally encoded FIBs of CIF counts 0, 1, 2 and 3 are transmitted in the same transmission frame, those of CIF counts 4, 5, 6 and 7 in the next transmission frame, and so on.

The vector $\left(b'_{i'}\right)_{i'=0}^{9215}$ shall be divided into three consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 2, 3, 4$ respectively.

Each block $\boldsymbol{P_l}$ is a vector $\left(p_{l,n}\right)_{n=0}^{3071}$, the bits $p_{l,\,n}$ being defined by:

$$p_{l,n} = b'_{i'}$$

and

$$l = \mathrm{Q}\left(i'/3072\right) + 2 \qquad i' = 0, 1, 2, \ldots, 9215$$

$$n = \mathrm{R}\left(i'/3072\right) \qquad i' = 0, 1, 2, \ldots, 9215$$

The principle of this block partitioning is shown in figure 90, for $r$ = 0, 1, 2 and 3.

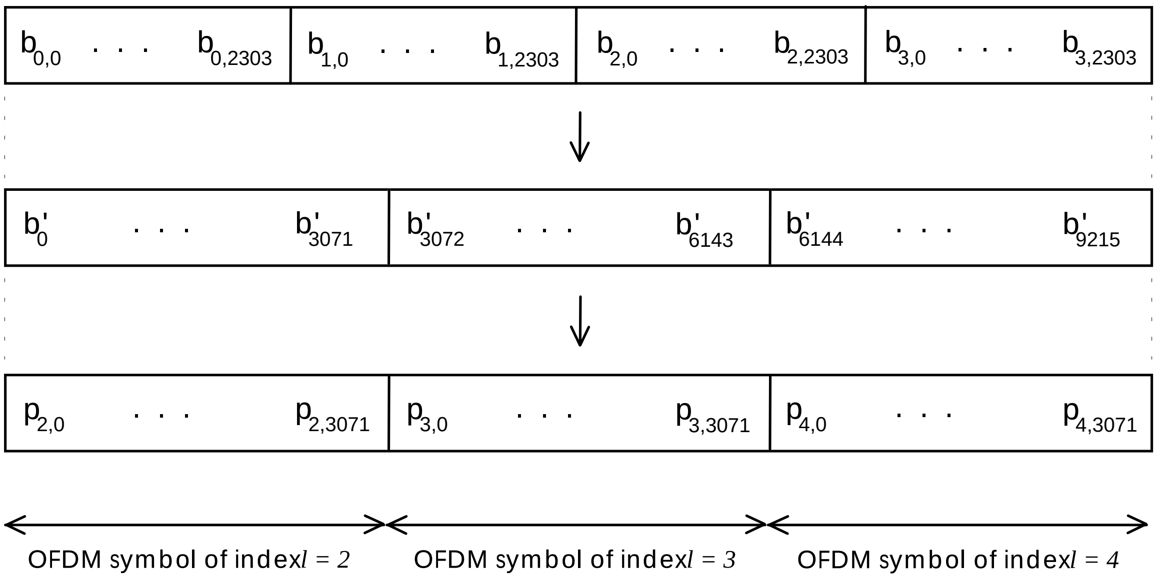


**Figure 90: Block partitioning in the FIC for transmission mode I**

### 14.4.1.2 Transmission mode II

The convolutional codeword $\left(b_i\right)_{i=0}^{2303}$ defined in subclause 11.2.1 shall be divided into three consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 2, 3, 4$ respectively.

Each block $\boldsymbol{P_l}$ is a vector $\left(p_{l,n}\right)_{n=0}^{767}$, the bits $p_{l,n}$ being defined by:

$$p_{l,n} = b_i$$

and

$$l = \mathrm{Q}(i/768) + 2 \qquad i = 0,1,2,\ldots,2303$$

$$n = \mathrm{R}(i/768) \qquad i = 0,1,2,\ldots,2303$$

The principle of this block partitioning is shown in figure 91.

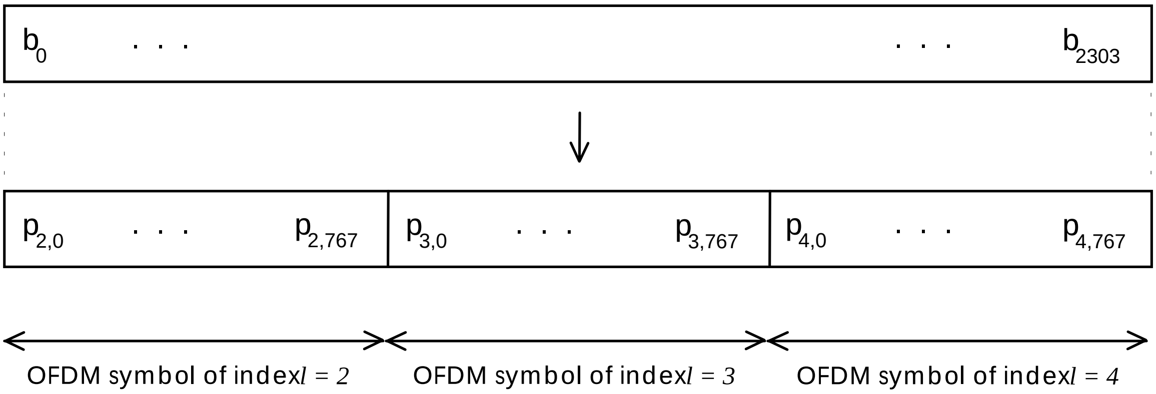


**Figure 91: Block partitioning in the FIC for transmission mode II**

#### 14.4.1.3 Transmission mode III

The convolutional codeword $\left(b_i\right)_{i=0}^{3071}$ defined in subclause 11.2.2 shall be divided into eight consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 2, 3, 4, \ldots, 9$ respectively.

Each block $\boldsymbol{P_l}$ is a vector $\left(p_{l,n}\right)_{n=0}^{383}$, the bits $p_{l,n}$ being defined by:

$$p_{l,n} = b_i$$

and

$$l = \mathrm{Q}(i/384) + 2 \qquad i = 0,1,2,\ldots,3071$$

$$n = \mathrm{R}(i/384) \qquad i = 0,1,2,\ldots,3071$$

The principle of this block partitioning is shown in figure 92.

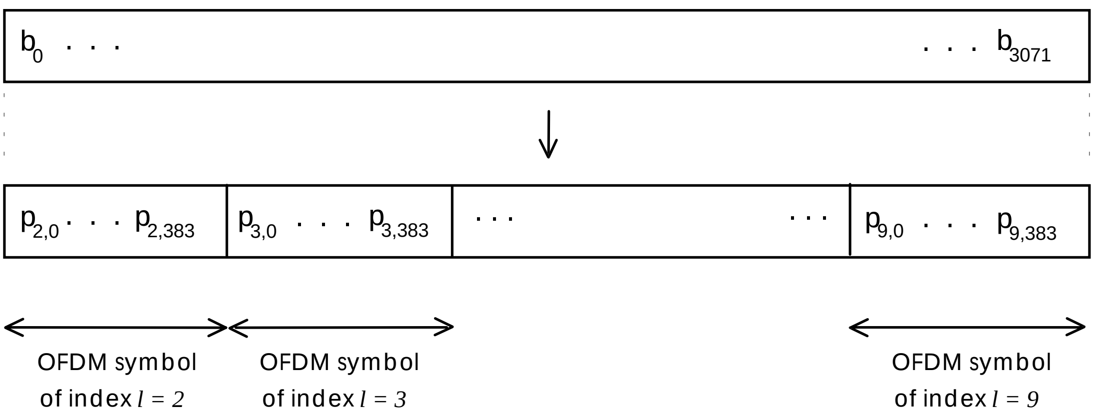


**Figure 92: Block partitioning in the FIC for transmission mode III**

### 14.4.2 Block partitioning and association of blocks to OFDM symbols in the Main Service Channel

#### 14.4.2.1 Transmission mode I

In transmission mode I, four CIFs shall be transmitted within each transmission frame as indicated in subclause 5.1.

Four consecutive CIFs defined in clause 13 shall be multiplexed to form a vector, which shall then be divided into 72 blocks to be transmitted on 72 OFDM symbols.

The multiplexing of four consecutive CIFs into one vector $\boldsymbol{D'}$ is defined by the following relation:

$$d'_{i'} = d_{r,i}$$

and

$$i' = i + 55296 \cdot \mathrm{mod}(r,4) \qquad \text{for } i = 0,1,2,\ldots,55295 \text{ and for any value of } r$$

where:

$d'_{i'}$ denotes the $(i'+1)^{th}$ bit of the vector $\boldsymbol{D'}$;

$d_{r,\,i}$ denotes the $(i+1)^{th}$ bit of the $(r+1)^{th}$ CIF.

As defined in clause 13, the index $r$ taken modulo 5 000 (mod($r$,5 000)) is equal to the CIF count defined in subclause 5.3.

This means that the arrangement of CIFs in a transmission frame shall be such that CIFs of CIF count 0, 1, 2 and 3 are transmitted in the same transmission frame, those of CIF count 4, 5, 6 and 7 in the next transmission frame, and so on.

The vector $\left(d'_{i'}\right)_{i'=0}^{221183}$ shall be divided into 72 consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 5, 6, 7, ..., 76$ respectively.

Each block $\boldsymbol{P_l}$ is a vector $\left(p_{l,n}\right)_{n=0}^{3071}$ , the bits $p_{l,\,n}$ being defined by:

$$p_{l,n} = d'_{i'}$$

and

$$l = \mathrm{Q}\left(\mathrm{i}'/3072\right) + 5 \qquad i' = 0,1,2,\ldots,221183$$
$$n = \mathrm{R}\left(i'/3072\right) \qquad i' = 0,1,2,\ldots,221183$$

The principle of this block partitioning is shown in figure 93, for $r$ = 0, 1, 2 and 3.

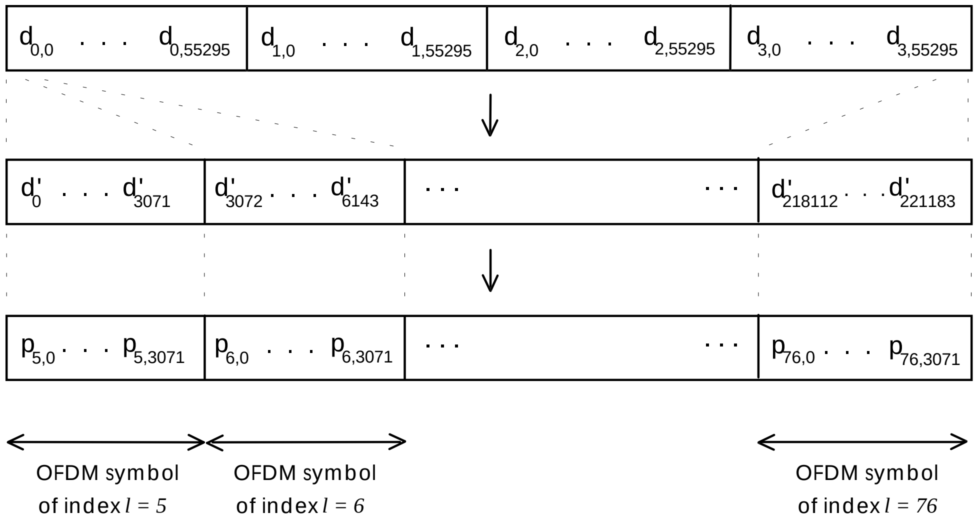


**Figure 93: Block partitioning in the MSC for transmission mode I**

### 14.4.2.2 Transmission mode II

The CIF constituted by the vector $(d_i)_{i=0}^{55295}$ defined in clause 13 shall be divided into 72 consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 5, 6, 7, \ldots, 76$ respectively.

The index $r$ of $d_{r,i}$ is omitted here because the process is not depending on the value of $r$.

Each block $\boldsymbol{P_l}$ is a vector $(p_{l,n})_{n=0}^{767}$, the bits $p_{l,n}$ being defined by:

$$p_{l,n} = d_i$$

and

$$l = \mathrm{Q}(i/768) + 5 \qquad i = 0,1,2,\ldots,55295$$

$$n = \mathrm{R}(i/768) \qquad i = 0,1,2,\ldots,55295$$

The principle of this block partitioning is shown in figure 94.

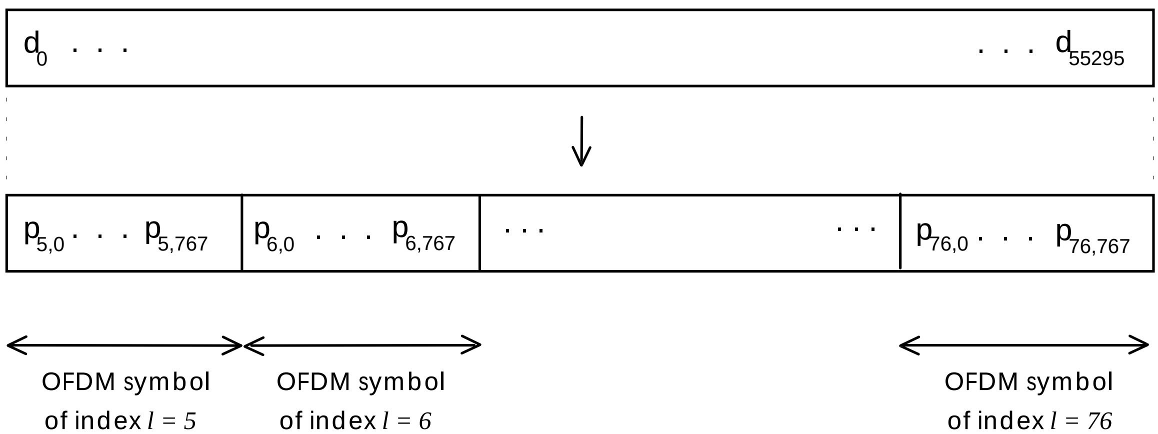


**Figure 94: Block partitioning in the MSC for transmission mode II**

### 14.4.2.3 Transmission mode III

The CIF constituted by the vector $(d_i)_{i=0}^{55295}$ defined in clause 13 shall be divided into 144 consecutive blocks $\boldsymbol{P_l}$, each block containing the bits to be transmitted in the OFDM symbol of index $l = 10, 11, 12, ..., 153$ respectively.

The index $r$ of $d_{r,i}$ is omitted here because the process is not depending on the value of $r$.

Each block $\boldsymbol{P_l}$ is a vector $(p_{l,n})_{n=0}^{383}$, the bits $p_{l,n}$ being defined by:

$$p_{l,n} = d_i$$

and

$$l = \mathrm{Q}(i/384) + 10 \qquad i = 0,1,2,\ldots,55\,295$$

$$n = \mathrm{R}(i/384) \qquad i = 0,1,2,\ldots,55\,295$$

The principle of this block partitioning is shown in figure 95.

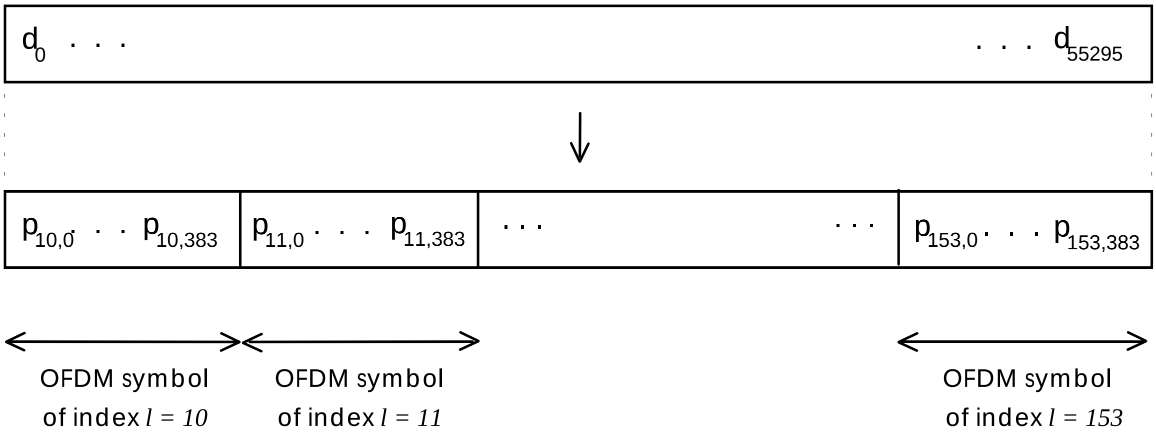


**Figure 95: Block partitioning in the MSC for transmission mode III**

## 14.5 QPSK symbol mapper

For any of the OFDM symbols of index $l = 2, 3, 4, ..., L$, the $2K$-bit vector $(p_{l,n})_{n=0}^{2K-1}$, whose elements $p_{l,n}$ are defined in subclauses 14.4.1 and 14.4.2, shall be mapped on the $K$ complex QPSK symbols $q_{l,n}$ according to the following relation:

$$q_{l,n} = \frac{1}{\sqrt{2}}\left[(1 - 2p_{l,n}) + j(1 - 2p_{l,n+K})\right] \qquad \text{for} \quad n = 0,1,2,\ldots,K-1.$$

## 14.6 Frequency interleaving

This subclause defines the correspondence between the index $n$ of the QPSK symbols $q_{l,n}$ and the carrier index $k$ $(-K/2 \le k < 0 \text{ and } 0 < k \le K/2)$.

The QPSK symbols shall be re-ordered according to the following relation:

$$y_{l,k} = q_{l,n} \quad \text{for } l = 2,3,4,\cdots,L$$

with $k = F(n)$, where $F$ is a function defined in the following subclauses for the three transmission modes.

### 14.6.1 Transmission mode I

Let $\Pi(i)$ be a permutation in the set of integers $i = 0, 1, 2,..., 2047$ obtained from the following congruential relation:

$$\Pi(i) = 13\ \Pi(i\text{-}1) + 511 \pmod{2048} \text{ and } \Pi(0) = 0;$$

$$\text{for } i = 1, 2, ..., 2047.$$

$\Pi(i)$ defines a permutation of the ordered set $(0, 1, 2, ..., 2047)$, resulting in the set:

$$A = \{\Pi(0), \Pi(1), \Pi(2), ..., \Pi(2047)\}.$$

Let $D$ be the set $D = \{d_0, d_1, d_2, ..., d_{1535}\}$, containing $1536$ elements and defined as being the subset of $A$ with the same element ordering, comprising all the elements of $A$ higher than or equal to $256$ and lower than or equal to $1792$, excluding $1024$. Thus, if $\Pi(i)$ is the $(n+1)$th element of $A$ in the range $[256, 1792]$ excluding $1024$, $d_n = \Pi(i)$.

The correspondence between the index $n \in \{0,1,2,\ldots,1535\}$ of the QPSK symbol $q_{l,\ n}$ and the frequency index $k \in \{-768,-767,-766,\ldots,768\} \setminus \{0\}$, shall be given by:

$$k = F(n) = d_n\text{-}1024.$$

The function $F$ is a one-to-one mapping between the sets $\{0, 1, 2, ..., 1535\}$ and

$\{-768, -767, -766, ..., 768\} \setminus \{0\}$.

The interleaving rule is illustrated in table 43.

**Table 43: Frequency interleaving for transmission mode I**

| $i$ | $\Pi(i)$ | $d_n$ | $n$ | $k$ |
|---|---|---|---|---|
| 0 | 0 | | | |
| 1 | 511 | 511 | 0 | -513 |
| 2 | 1 010 | 1 010 | 1 | -14 |
| 3 | 1 353 | 1 353 | 2 | 329 |
| 4 | 1 716 | 1 716 | 3 | 692 |
| 5 | 291 | 291 | 4 | -733 |
| 6 | 198 | | | |
| 7 | 1 037 | 1 037 | 5 | 13 |
| 8 | 1 704 | 1 704 | 6 | 680 |
| 9 | 135 | | | |
| 10 | 218 | | | |
| 11 | 1 297 | 1 297 | 7 | 273 |
| 12 | 988 | 988 | 8 | -36 |
| 13 | 1 076 | 1 067 | 9 | 43 |
| 14 | 46 | | | |
| 15 | 1 109 | 1 109 | 10 | 85 |
| 16 | 592 | 592 | 11 | -432 |
| 17 | 15 | | | |
| 18 | 706 | 706 | 12 | -318 |
| : | : | : | | |
| 2 044 | 1 676 | 1 676 | 1 533 | 652 |
| 2 045 | 1 819 | | | |
| 2 046 | 1 630 | 1 630 | 1 534 | 606 |
| 2 047 | 1 221 | 1 221 | 1 535 | 197 |

### 14.6.2 Transmission mode II

Let $\Pi(i)$ be a permutation in the set of integers $i = 0, 1, 2,..., 511$ obtained from the following congruential relation:

$$\Pi(i) = 13\ \Pi(i\text{-}1) + 127 \text{ (mod } 512) \text{ and } \Pi(0) = 0;$$

$$\text{for } i = 1, 2, ..., 511.$$

$\Pi(i)$ defines a permutation of the ordered set $(0, 1, 2, ..., 511)$, resulting in the set:

$$A = \{\Pi(0), \Pi(1), \Pi(2), ..., \Pi(511)\}.$$

Let $D$ be the set $D = \{d_0, d_1, d_2, ..., d_{383}\}$, containing $384$ elements and defined as being the subset of $A$ with the same element ordering, comprising all the elements of $A$ higher than or equal to $64$ and lower than or equal to $448$, excluding $256$. Thus, if $\Pi(i)$ is the *(n+1)*th element of $A$ in the range $[64, 448]$ excluding $256$, $d_n = \Pi(i)$.

The correspondence between the index $n \in \{0,1,2,\ldots,383\}$ of the QPSK symbol $q_{l,\ n}$ and the frequency index $k \in \{-192,-191,-190,\ldots,192\} \setminus \{0\}$, shall be given by:

$$k = F(n) = d_n\text{-}256.$$

The function $F$ is a one-to-one mapping between the sets $\{0, 1, 2, ..., 383\}$ and

$\{-192, -191, -190, ..., 192\} \setminus \{0\}$.

The interleaving rule is illustrated in table 44.

**Table 44: Frequency interleaving for transmission mode II**

| $i$ | $\Pi(i)$ | $d_n$ | $n$ | $k$ |
|---|---|---|---|---|
| 0 | 0 | | | |
| 1 | 127 | 127 | 0 | - 129 |
| 2 | 242 | 242 | 1 | - 14 |
| 3 | 201 | 201 | 2 | - 55 |
| 4 | 180 | 180 | 3 | - 76 |
| 5 | 419 | 419 | 4 | 163 |
| 6 | 454 | | | |
| 7 | 397 | 397 | 5 | 141 |
| 8 | 168 | 168 | 6 | - 88 |
| 9 | 263 | 263 | 7 | 7 |
| 10 | 474 | | | |
| 11 | 145 | 145 | 8 | - 111 |
| 12 | 476 | | | |
| 13 | 171 | 171 | 9 | - 85 |
| 14 | 302 | 302 | 10 | 46 |
| 15 | 469 | | | |
| 16 | 80 | 80 | 11 | - 176 |
| 17 | 143 | 143 | 12 | - 113 |
| 18 | 450 | | | |
| : | : | : | | |
| 508 | 140 | 140 | 380 | - 116 |
| 509 | 411 | 411 | 381 | 155 |
| 510 | 350 | 350 | 382 | 94 |
| 511 | 69 | 69 | 383 | - 187 |

### 14.6.3 Transmission mode III

Let $\Pi(i)$ be a permutation in the set of integers $i = 0, 1, 2,..., 255$ obtained from the following congruential relation:

$$\Pi(i) = 13\ \Pi(i\text{-}1) + 63 \text{ (mod } 256) \text{ and } \Pi(0) = 0;$$

$$\text{for } i = 1, 2, ..., 255.$$

$\Pi(i)$ defines a permutation of the ordered set $(0, 1, 2, ..., 255)$, resulting in the set:

$$A = \{\Pi(0), \Pi(1), \Pi(2), ..., \Pi(255)\}.$$

Let $D$ be the set $D = \{d_0, d_1, d_2, ..., d_{191}\}$, containing $192$ elements and defined as being the subset of $A$ with the same element ordering, comprising all the elements of $A$ higher than or equal to $32$ and lower than or equal to $224$, excluding $128$. Thus, if $\Pi(i)$ is the $(n+1)$th element of $A$ in the range $[32, 224]$ excluding $128$, $d_n = \Pi(i)$.

The correspondence between the index $n \in \{0,1,2,\ldots,191\}$ of the QPSK symbol $q_{l,\ n}$ and the frequency index $k \in \{-96,-95,-94,\ldots,96\} \setminus \{0\}$, shall be given by:

$$k = F(n) = d_n\text{-}128.$$

The function $F$ is a one-to-one mapping between the sets $\{0, 1, 2, ..., 191\}$ and

$\{-96, -95-, -94, ..., 96\} \setminus \{0\}$.

The interleaving rule is illustrated in table 45.

**Table 45: Frequency interleaving for transmission mode III**

| $i$ | $\Pi(i)$ | $d_n$ | $n$ | $k$ |
|---|---|---|---|---|
| 0 | 0 | | | |
| 1 | 63 | 63 | 0 | - 65 |
| 2 | 114 | 114 | 1 | - 14 |
| 3 | 9 | | | |
| 4 | 180 | 180 | 2 | 52 |
| 5 | 99 | 99 | 3 | - 29 |
| 6 | 70 | 70 | 4 | - 58 |
| 7 | 205 | 205 | 5 | 77 |
| 8 | 168 | 168 | 6 | 40 |
| 9 | 199 | 199 | 7 | 71 |
| 10 | 90 | 90 | 8 | - 38 |
| 11 | 209 | 209 | 9 | 81 |
| 12 | 220 | 220 | 10 | 92 |
| 13 | 107 | 107 | 11 | - 21 |
| 14 | 174 | 174 | 12 | 46 |
| 15 | 21 | | | |
| 16 | 80 | 80 | 13 | - 48 |
| 17 | 79 | 79 | 14 | - 49 |
| 18 | 66 | 66 | 15 | - 62 |
| : | : | : | | |
| 252 | 140 | 140 | 188 | 12 |
| 253 | 91 | 91 | 189 | - 37 |
| 254 | 222 | 222 | 190 | 94 |
| 255 | 133 | 133 | 191 | 5 |

## 14.7 Differential modulation

Differential modulation shall be applied to the QPSK symbols on each carrier. The differential modulation is defined by the following rule:

$$z_{l,k} = z_{l-1,k} \cdot y_{l,k}$$

for $l = 2,3,4,\ldots,L$

and $-\frac{K}{2} \leq k \leq \frac{K}{2}$

This means that each carrier is modulated using a $\pi/4$-shift D-QPSK. All together, they form the main signal defined in subclause 14.2.

As indicated in subclause 14.2, the generation of the complex D-QPSK symbols $z_{m,l,k}$ does not depend on the transmission frame index $m$, which appears on the formula defining the main signal $s(t)$.

The main signal $s(t)$ is therefore defined for all values of $t$.

## 14.8 Transmitter Identification Information

The TII is conveyed in the Synchronization channel. It provides unambiguous identification of each transmitter in a DAB network. The implementation of TII is optional.

When used in conjunction with Service Information as defined in subclause 8.1.9, the TII feature may provide an estimate of the geographical position of the receiver. In particular, TII is used with the Frequency Information (FI) facility in a Single Frequency Network (SFN).

The TII signal shall fill the Null symbol of each transmission frame comprising the CIFs of CIF count 0,1,2,3 modulo 8 (transmission mode I) and comprising the CIF of CIF count 0 modulo 2 (transmission modes II and III).

The TII signal consists of a certain number of pairs of adjacent carriers of an OFDM symbol; the actual selection of those carriers present in the TII symbol identifies the transmitter.

The selection of the carriers is defined by assigning two numbers to each transmitter; the pattern number p and the comb number c. These numbers are the Main Identifier and sub-Identifier of a transmitter, respectively, and are used in the TII feature described in subclause 8.1.9.

The TII signal $s_{TII}(t)$ associated with a given transmitter shall be:

$$s_{TII}(t) = \mathrm{Re}\left\{ e^{2j\pi f_c t} \sum_{m=-\infty}^{+\infty} \sum_{k=-K/2}^{K/2} z_{m,0,k} \cdot g_{TII,k}(t - mT_F) \right\}$$

where:

$$g_{TII,k}(t) = e^{2\pi jk(t-T_{NULL}+T_U)/T_U} \cdot \mathrm{Rect}(t/T_{NULL})$$

The parameters $T_U$, $T_{NULL}$ and $f_c$ are defined in subclause 14.2; $z_{m,0,k}$ is the complex number associated to carrier $k$ of the Null symbol. It is equal to zero during the transmission frame m when the TII signal is not transmitted. Its values, for the transmission frame m where the TII signal is transmitted, shall be derived from the values of p and c.

The following relation is defined:

$$z_{m,0,k} = A_{c,p}(k) \cdot e^{j\varphi_k} + A_{c,p}(k-1) \cdot e^{j\varphi_{k-1}}$$

The values of $\varphi_k$ are defined in subclause 14.3.2. The values of $A_{c,p}(k)$ are specified in the following subclauses.

### 14.8.1 Transmission modes I and II

The following formulae shall apply:

**Transmission mode I:**

$$A_{c,p}(k) = \begin{cases} \sum_{b=0}^{7} \delta(k, -768 + 2c + 48b) \cdot a_b(p) & \text{for } -768 \le k < -384 \\ \sum_{b=0}^{7} \delta(k, -384 + 2c + 48b) \cdot a_b(p) & \text{for } -384 \le k < 0 \\ \sum_{b=0}^{7} \delta(k, 1 + 2c + 48b) \cdot a_b(p) & \text{for } 0 < k \le 384 \\ \sum_{b=0}^{7} \delta(k, 385 + 2c + 48b) \cdot a_b(p) & \text{for } 384 < k \le 768 \end{cases}$$

and $A_{c,p}(0) = A_{c,p}(-769) = 0$.

This formula shall apply for $0 \le c \le 23$.

$a_b(p)$ is defined in table 46. $\delta$ is the Kronecker symbol defined in subclause 3.3.

**Transmission mode II:**

$$A_{c,p}(k) = \sum_{b=0}^{3} \delta(k, -192 + 2c + 48b) \cdot a_b(p) + \sum_{b=4}^{7} \delta(k, -191 + 2c + 48b) \cdot a_b(p)$$

and $A_{c,p}(-193) = 0$.

This formula shall apply for $0 \leq c \leq 23$.

$a_b(p)$ is defined in table 46. $\delta$ is the Kronecker symbol defined in subclause 3.3.

**Table 46: TII pattern for transmission modes I and II**

| ***p*** | $a_b(p)$ ***b*=0,1,2,3,4,5,6,7** | ***p*** | $a_b(p)$ ***b*=0,1,2,3,4,5,6,7** | ***p*** | $a_b(p)$ ***b*=0,1,2,3,4,5,6,7** |
|---|---|---|---|---|---|
| 0 | 0 0 0 0 1 1 1 1 | 24 | 0 1 0 1 1 1 0 0 | 48 | 1 0 1 0 1 0 0 1 |
| 1 | 0 0 0 1 0 1 1 1 | 25 | 0 1 1 0 0 0 1 1 | 49 | 1 0 1 0 1 0 1 0 |
| 2 | 0 0 0 1 1 0 1 1 | 26 | 0 1 1 0 0 1 0 1 | 50 | 1 0 1 0 1 1 0 0 |
| 3 | 0 0 0 1 1 1 0 1 | 27 | 0 1 1 0 0 1 1 0 | 51 | 1 0 1 1 0 0 0 1 |
| 4 | 0 0 0 1 1 1 1 0 | 28 | 0 1 1 0 1 0 0 1 | 52 | 1 0 1 1 0 0 1 0 |
| 5 | 0 0 1 0 0 1 1 1 | 29 | 0 1 1 0 1 0 1 0 | 53 | 1 0 1 1 0 1 0 0 |
| 6 | 0 0 1 0 1 0 1 1 | 30 | 0 1 1 0 1 1 0 0 | 54 | 1 0 1 1 1 0 0 0 |
| 7 | 0 0 1 0 1 1 0 1 | 31 | 0 1 1 1 0 0 0 1 | 55 | 1 1 0 0 0 0 1 1 |
| 8 | 0 0 1 0 1 1 1 0 | 32 | 0 1 1 1 0 0 1 0 | 56 | 1 1 0 0 0 1 0 1 |
| 9 | 0 0 1 1 0 0 1 1 | 33 | 0 1 1 1 0 1 0 0 | 57 | 1 1 0 0 0 1 1 0 |
| 10 | 0 0 1 1 0 1 0 1 | 34 | 0 1 1 1 1 0 0 0 | 58 | 1 1 0 0 1 0 0 1 |
| 11 | 0 0 1 1 0 1 1 0 | 35 | 1 0 0 0 0 1 1 1 | 59 | 1 1 0 0 1 0 1 0 |
| 12 | 0 0 1 1 1 0 0 1 | 36 | 1 0 0 0 1 0 1 1 | 60 | 1 1 0 0 1 1 0 0 |
| 13 | 0 0 1 1 1 0 1 0 | 37 | 1 0 0 0 1 1 0 1 | 61 | 1 1 0 1 0 0 0 1 |
| 14 | 0 0 1 1 1 1 0 0 | 38 | 1 0 0 0 1 1 1 0 | 62 | 1 1 0 1 0 0 1 0 |
| 15 | 0 1 0 0 0 1 1 1 | 39 | 1 0 0 1 0 0 1 1 | 63 | 1 1 0 1 0 1 0 0 |
| 16 | 0 1 0 0 1 0 1 1 | 40 | 1 0 0 1 0 1 0 1 | 64 | 1 1 0 1 1 0 0 0 |
| 17 | 0 1 0 0 1 1 0 1 | 41 | 1 0 0 1 0 1 1 0 | 65 | 1 1 1 0 0 0 0 1 |
| 18 | 0 1 0 0 1 1 1 0 | 42 | 1 0 0 1 1 0 0 1 | 66 | 1 1 1 0 0 0 1 0 |
| 19 | 0 1 0 1 0 0 1 1 | 43 | 1 0 0 1 1 0 1 0 | 67 | 1 1 1 0 0 1 0 0 |
| 20 | 0 1 0 1 0 1 0 1 | 44 | 1 0 0 1 1 1 0 0 | 68 | 1 1 1 0 1 0 0 0 |
| 21 | 0 1 0 1 0 1 1 0 | 45 | 1 0 1 0 0 0 1 1 | 69 | 1 1 1 1 0 0 0 0 |
| 22 | 0 1 0 1 1 0 0 1 | 46 | 1 0 1 0 0 1 0 1 | | |
| 23 | 0 1 0 1 1 0 1 0 | 47 | 1 0 1 0 0 1 1 0 | | |

Figure 96 illustrates the result of this procedure in transmission mode II for c=4 and p=16.

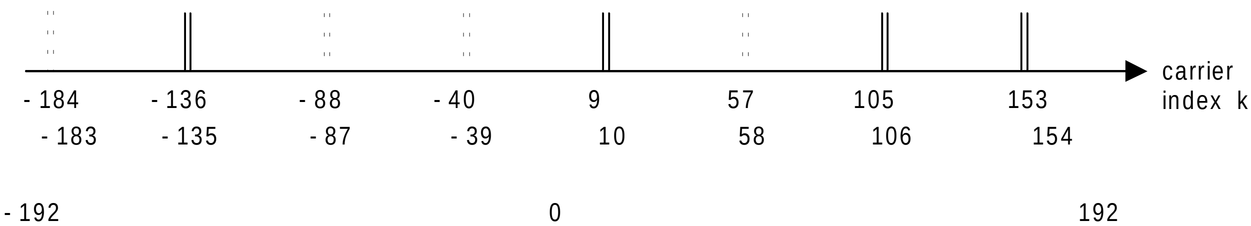


**Figure 96: Example of TII signal in transmission mode II**

In this figure, all the pairs of carriers shown belong to the comb of number c=4. Only these shown by a full line are actually transmitted, because they belong to the set of carriers defined by the pattern number p=16.

### 14.8.2 Transmission mode III

The following formula shall apply:

$$A_{c,p}(k) = \sum_{b=0}^{1} \delta(k,-96+2c+48b) \cdot a_b(p) + \sum_{b=2}^{3} \delta(k,-95+2c+48b) \cdot a_b(p)$$

and $A_{c,p}(-97) = 0$ .

This formula shall apply for $0 \le c \le 23$ .

$a_b(p)$ is defined in table 47. $\delta$ is the Kronecker symbol defined in subclause 3.3.

**Table 47: TII pattern for transmission mode III**

| ***p*** | $a_b(p)$ ***b*=0,1,2,3** |
|---|---|
| 0 | 0 0 1 1 |
| 1 | 0 1 0 1 |
| 2 | 0 1 1 0 |
| 3 | 1 0 0 1 |
| 4 | 1 0 1 0 |
| 5 | 1 1 0 0 |

# 15 Radio frequency characteristics

This clause defines the preferred choice of the transmission mode as a function of the system operating conditions.

It also indicates the nominal characteristics of the DAB transmission signal at the radio frequency level, and specifies the permitted values of the central frequency. The consequences of pre-filtering and non-linear amplification on the time and spectrum characteristics of the signal, as well as on the performance of the system, are not included here.

### 15.1 Use of the transmission modes

The preferred choice of the transmission modes is dependent on the system operating conditions.

Transmission mode I is intended to be used for terrestrial Single Frequency Networks (SFN) and local-area broadcasting in Bands I, II and III.

Transmission mode II is intended to be used for terrestrial local broadcasting in Bands I, II, III, IV, V and in the 1 452 - 1 492 MHz frequency band (i.e. L-Band). It can also be used for satellite-only and hybrid satellite-terrestrial broadcasting in L-Band.

Transmission mode III is intended to be used for terrestrial, satellite and hybrid satellite-terrestrial broadcasting below 3 000 MHz.

For cable distribution, transmission mode III is the preferred mode because it can be used at any frequency available on cable. However, transmission modes I and II may also be used, depending on the chosen frequency band.

### 15.2 Time characteristics

The DAB transmission signal consists of a succession of consecutive transmission frames of 96 ms duration for transmission mode I, and 24 ms duration for transmission modes II and III.

The Synchronization channel occupies the first 5 208 elementary periods (approximately 2,543 ms) for transmission mode I, 1 302 elementary periods (approximately 0,636 ms) for transmission mode II, and

664 elementary periods (approximately 0,324 ms) for transmission mode III. The elementary period is 1/2 048 000 seconds (see subclause 14.2).

The modulated OFDM symbols, corresponding to the FIC and the MSC, occupy the remaining portion of the transmission frame. These are approximately 93,457 ms for transmission mode I, approximately 23,364 ms for transmission mode II, and approximately 23,676 ms for transmission mode III.

The synchronization channel conveys a fixed pattern as defined in clause 14.

The modulated OFDM symbols, as a sum of equally-spaced orthogonal carriers with independent phases, exhibit a Gaussian-like amplitude distribution.

### 15.3 Spectrum characteristics

The synchronization channel, repeated at the transmission frame rate, constitutes a fixed pattern described in clause 14, during which the transmitted signal is the juxtaposition of equally-spaced orthogonal carriers, with fixed amplitudes and phases.

The modulated OFDM symbols constitutes a juxtaposition of equally-spaced orthogonal carriers, with constant amplitude and time varying independent phases, resulting from the modulation procedure described in clause 14.

The power spectral density $P_k(f)$ of each carrier at frequency $f_k = f_c + k/T_u$ .

($-K/2 \leq k < 0$ and $0 < k \leq K/2$) is defined by the following expression:

$$P_k(f) = \left[\frac{\sin\pi(f - f_k)T_s}{\pi(f - f_k)T_s}\right]^2$$

The overall power spectral density of the modulated symbols is the sum of the power spectral densities of all the carriers. Because the OFDM symbol duration is larger than the inverse of the carrier spacing, the main lobe of the power spectral density of each carrier is narrower than twice the carrier spacing. The theoretical DAB transmission signal spectra are illustrated in figures 97, 98 and 99 for transmission modes I, II and III respectively.

The level of the signal at frequencies outside the nominal 1,536 MHz bandwidth can be reduced by applying an appropriate filtering. The degree of suppression required of the side lobes shown in figures 97, 98 and 99 will depend on the network configuration chosen and frequency co-ordination criteria with other transmissions.

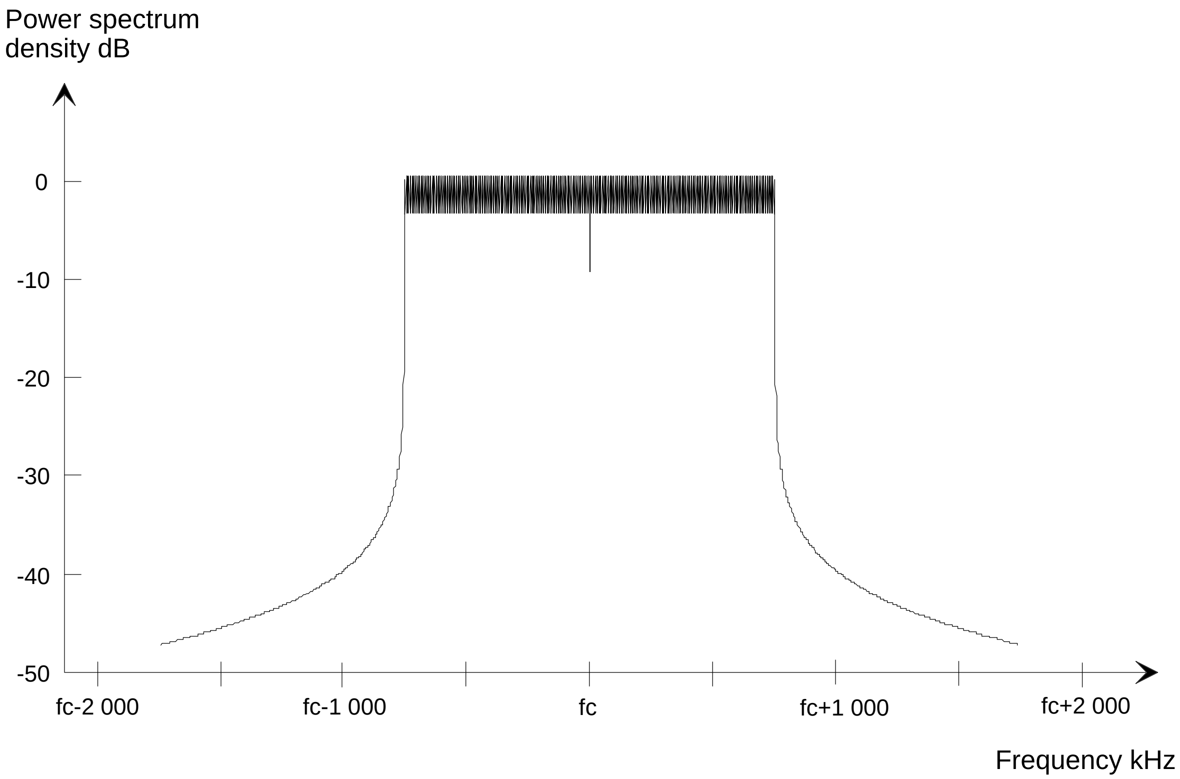


**Figure 97: Theoretical DAB transmission signal spectrum for transmission mode I**

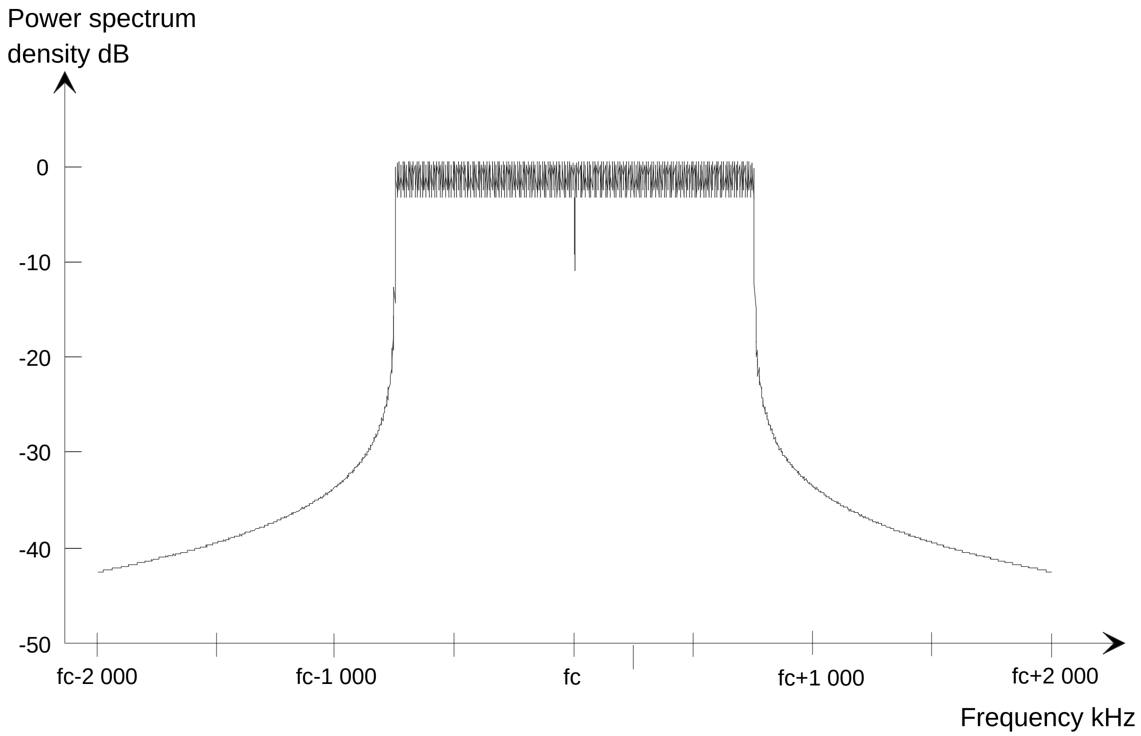


**Figure 98: Theoretical DAB transmission signal spectrum for transmission mode II**

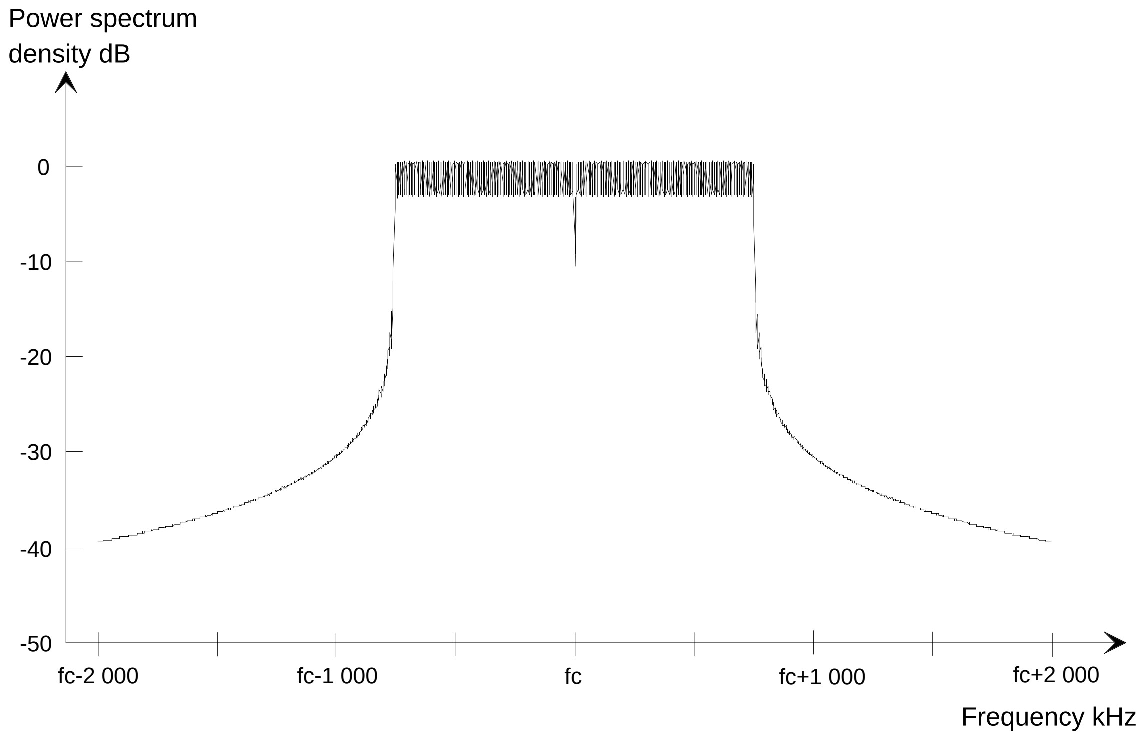


**Figure 99: Theoretical DAB transmission signal spectrum for transmission mode III**

## 15.4 Spectrum mask

The out-of-band radiated signal spectrum in any 4 kHz band shall be constrained by one of the masks defined in figure 100.

The solid line mask shall apply to VHF transmitters in critical areas for adjacent channel interference. The dotted line mask shall apply to VHF transmitters in other circumstances and to UHF transmitters in critical cases for adjacent channel interference.

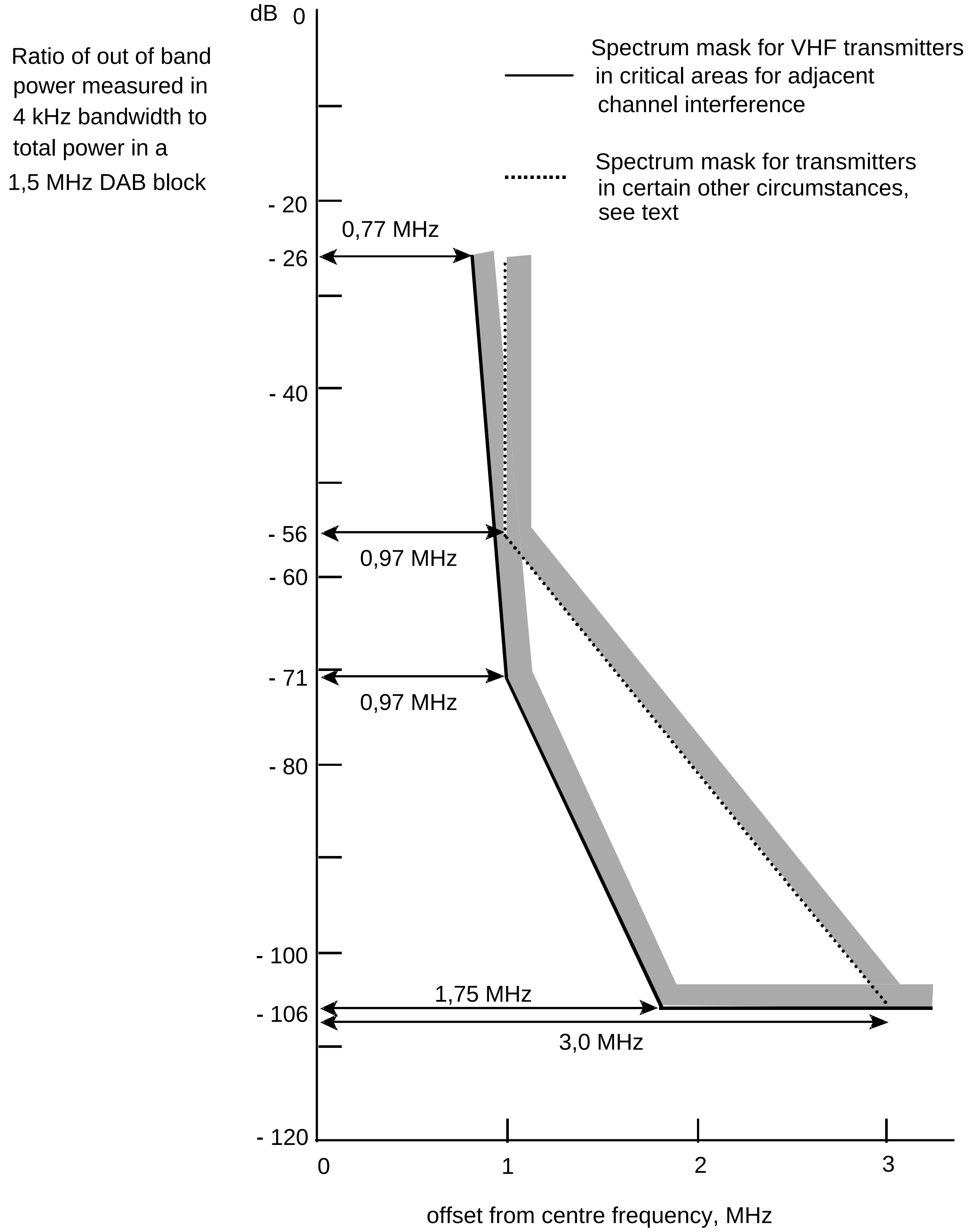


**Figure 100: Out-of-band spectrum mask for DAB transmission signal (all transmission modes)**

### 15.5 Permitted values of the central frequency

The nominal central frequency $f_c$ shall be an exact multiple of 16 kHz.

The actual central frequency may be offset by up to ± 1/2 carrier spacing ($1/T_u$) in any transmission mode, where necessary, to improve spectrum sharing.

# Annex A (informative): Main characteristics of the audio coding system

## A.1 Input audio signal characteristics

The main characteristics of the input audio signal are:

- **audio bandwidth:** the input audio signal can cover the whole audio frequency range up to 20 kHz;
- **audio Interface:** the digital input signal may conform to the AES/EBU interface specification (see IEC 958 [12]);
- **input resolution:** the system can support any input resolution up to 22 bits/sample;
- **sampling frequency:** the sampling frequency of the input audio signal at the DAB transmitter and of the output audio signal at the DAB receiver is 48 kHz.

## A.2 Audio coding characteristics

The main characteristics of the audio coding system are:

- **audio modes:** four Audio modes are provided:
    - single channel mode (one monophonic audio programme);
    - stereo mode (left and right channels of a stereophonic audio programme);
    - dual channel mode. In this mode, the two audio channels can be either bilingual, or two mono channels;
    - joint stereo mode. In this mode, the encoder exploits redundancy and irrelevancy of stereo signals for further data reduction, using Intensity stereo coding.
- **Bit rate:** the permitted bit rates of the encoded audio signal in Single channel mode are as follows: 32, 48, 56, 64, 80, 96, 112, 128, 160, and 192 kbit/s. The Stereo, Dual channel and Joint stereo modes use twice the bit rate of the Single channel mode;
- **DAB Audio frame length:** the length of a DAB Audio frame is always 24 ms, corresponding to 1 152 PCM audio samples.

## A.3 Audio associated data characteristics

### Programme Associated Data

Each DAB audio frame contains a number of bytes specifically for carrying Programme Associated Data (PAD). At the end of the DAB Audio frame, a capacity of at least two bytes, called Fixed Programme Associated Data (F-PAD), equivalent to a bit rate of 0,667 kbit/s, shall be provided, irrespective of the bit rate or the Audio mode (i.e. Single channel, Stereo and Dual channel modes will all have the same capacity of F-PAD). But the broadcaster may choose to extend this capacity, called Extended Programme Associated Data (X-PAD) in order to transmit more audio related data.

These PAD shall comprise mainly information which are intimately associated with the audio signal, and which would become useless if delayed in a queue with other data, or if removed from the channel-coded Audio bit stream and sent in a separate data service. By reserving the limited capacity available for the PAD for information satisfying these criteria, it is possible to make the most effective use of such a data channel, which is strongly linked with the encoded audio signal. Although some capacity of X-PAD can also be provided for Programme service information, further capacity can be provided elsewhere in the DAB multiplex (or ensemble) to carry additional information, such as text, relating to the various programmes in the ensemble which may require this, or some similar, facility.

### Error protection of PAD

The F-PAD and some parts of X-PAD are more strongly protected by the convolutional code of the transmission system than most of the other parts of the DAB audio bit stream (see clause 11). These fields are protected with different code-rates due to the Unequal Error Protection (UEP). Compared to the audio sub-band samples, a higher protection applies to F-PAD and four bytes of X-PAD. Depending on the requirement of the different types of PAD, this protection may be supplemented by additional protection schemes.

## A.4 Functions provided by PAD

The following functions are provided by F-PAD and X-PAD:

- **Dynamic Range Control:** with the help of the Dynamic Range Control (DRC) the receiver may reduce the dynamic range of the audio signal. The purpose of this is to adapt the dynamic range of the audio signal to listening in a noisy environment.

  In order to provide the dynamic range compression facility for receivers requiring the DRC signal, the audio programme is examined in a compression processor on the broadcasters premises, which derives the DRC signal, but does not apply any compression to the audio. The DRC signal, which controls a variable-gain amplifier or an equivalent digital processor in the receiver, is send to the receiver in the Fixed Programme Associated Data (F-PAD), see subclause 7.4.1.1.

- **Music/Speech indication:** two bits representing the Music/Speech flags indicate whether the transmitted sound consists of music or speech. The receiver may use this information to control any sound processing circuitry. One special combination of the flags signals that no indication is given. The Music/Speech indication should be repeated about 10 times per second.

- **Command channel:** a channel can be provided to convey, synchronously to the music, special commands to the decoder. Such commands may be used, for instance, to trigger the read out of a picture from a buffer memory that was filled, asynchronously, in advance. This channel is able to carry a few bytes within 0,2 - 0,5 second, at irregular intervals.

- **ISRC and UPC/EAN:** ISO 3901 [9] and the Universal Product Code / European Article Number (see EN 797 [15]) are provided by (some) digital carriers of pre-recorded software. Transmission of ISRC and UPC/EAN requires 10 bit per second.

- **Programme related text:** to elucidate the transmitted audio signal - a song, a programme item - coded text may be carried together with the audio (see annex F, bibliography). This text may be made on-site by the programme provider, it may be read from digital pre-recorded software and relayed more or less transparently, or various sources can be combined. The channel capacity required for text is dependent on how comprehensive and attractive the service is made.

- **In-house information:** channels can be provided for both short, synchronous commands and for long strings of asynchronous data. The meaning of these commands and data is not subject to standardisation, as it is intended for internal use within the broadcast chain only.

# Annex B (normative): Audio decoding

## B.1 General

The first action is synchronization of the decoder to the incoming Audio bit stream. Just after start-up this is done by using an external hardware synchronization signal, which is provided by the COFDM channel-demodulator.

In the DAB application, some parts of the ISO/IEC 11172-3 [3] header information, which are still kept in the DAB audio frame header, are already known to the decoder and need not to be decoded. These are layer, protection_bit, sampling_frequency, padding_bit, private bit and emphasis.

In addition to the ID bit, bitrate_index bits, copyright bit and original/copy bit, the decoder shall read the mode bits, and if these equal "01" also the mode_extension bits. The mode_extension bits set the 'bound' as shown in subclause 7.2 and thus indicate which sub-bands are coded in the Intensity stereo mode.

## B.2 CRC Check for audio side information

A CRC-check word for detecting errors within the significant side information of a DAB audio frame has been inserted in the bit stream just after the DAB audio frame header. The error detection method used is 'CRC-16' whose generator polynomial is:

$$G_1(X) = X^{16} + X^{15} + X^2 + 1$$

The bits included into the CRC-check are:

- 16 bits of DAB_audio_frame_header( ), starting with bit_rate_index and ending with emphasis;
- a number of bits of audio_data( ), starting with the first bit. These bits include BA and ScFSI.

The method for the calculation of the CRC word in the decoder is described in annex E. The initial state of the shift register is "1111 1111 1111 1111". If the final output of the shift register and the CRC-check word in the DAB audio frame are not identical, a transmission error has occurred in the protected field of the audio bit stream.

## B.3 CRC check for Scale Factors

For detection of errors within the three MSb´s of the Scale Factors, CRC-check words shall be inserted in the DAB audio bit stream just in front of the F-PAD field of the preceding DAB audio frame. The CRC-check words are covering the Scale Factors of the following sub-bands:

If the bit rate per channel is greater than or equal to 56 kbit/s (i.e. bit rate $\geq$ 56 kbit/s for single channel mode, bit rate $\geq$ 112 kbit/s for all other modes):

- ScF-CRC0: Sub-bands 0 to 3 (sub-band group 0);
- ScF-CRC1: Sub-bands 4 to 7 (sub-band group 1);
- ScF-CRC2: Sub-bands 8 to 15 (sub-band group 2);
- ScF-CRC3: Sub-bands 16 to 26 (sub-band group 3).

If the bit rate per channel is less than 56 kbit/s (i.e. bit rate < 56 kbit/s for Single channel mode, bit rate < 112 kbit/s for other audio modes):

- ScF-CRC0: Sub-bands 0 to 3 (sub-band group 0);
- ScF-CRC1: Sub-bands 4 to 7 (sub-band group 1).

To keep the position of the ScF-CRC-check words 1 and 2 independent of the bit rate, the ScF-CRC-check words are put in reverse order in the bit stream:

- ScF-CRC3, ScF- CRC2, ScF-CRC1, ScF-CRC0 bitrate ≥ 56 kbit/s/ch;
- ScF-CRC1, ScF-CRC0 bitrate < 56 kbit/s/ch.

The error detection method used is 'CRC-8', whose generator polynomial is:

$$G_2(X) = X^8 + X^4 + X^3 + X^2 + 1$$

The bits included in the CRC-check are the 3 MSbs of all Scale Factors of the sub-band group, according to their order in the bit stream.

The method for the calculation of the ScF-CRC word is the same as for the CRC word in clause B.2, and is described in annex E. If the output of the shift register and the transmitted ScF-CRC-check words not identical, a transmission error has occurred in the three MSbs of one of the Scale Factors of this special sub-band group.

## B.4 Decoding of ISO/IEC 11172-3 Layer II bit stream

The principles of the decoding process are given in ISO/IEC 11172-3 [3], Chapter 2.4.3.3.

# Annex C (informative): Audio encoding

## C.1 Analysis sub-band filter

The first step in the encoding process of a broadband PCM audio signal should be the filtering into 32 equally spaced sub-bands, each of which is down-sampled by a factor of $f_s/32$. The flow chart of this iterative process with the appropriate formulas is given in figure C.1. The analysis sub-band filtering includes the following steps:

- input 32 PCM audio samples;

- build an input sample vector $X$ of 512 elements, so that the 512 most recent PCM audio samples are stored in the vector $X$. In each iteration 32 PCM audio samples are shifted in at positions 0 to 31, the most recent on at position 0, and the 32 oldest samples are shifted out. Position 0 of the vector $X$ always contains the most recent sample, and position 511 the oldest one;

- vector $X$ is windowed by vector $C$. The coefficients $C_i$ are to be found in the table C.1;

- calculate the 64 intermediate values $Y_i$ according to the formula given in the analysis filter flow chart;

- the 32 sub-band samples $S_i$ are calculated by matrixing. The coefficients for the matrix $M$ can be calculated by the following formula:

$$M_{ik} = \cos [(2i + 1)(k - 16)\pi/64] \quad 0 \le i \le 31, \; 0 \le k \le 63$$

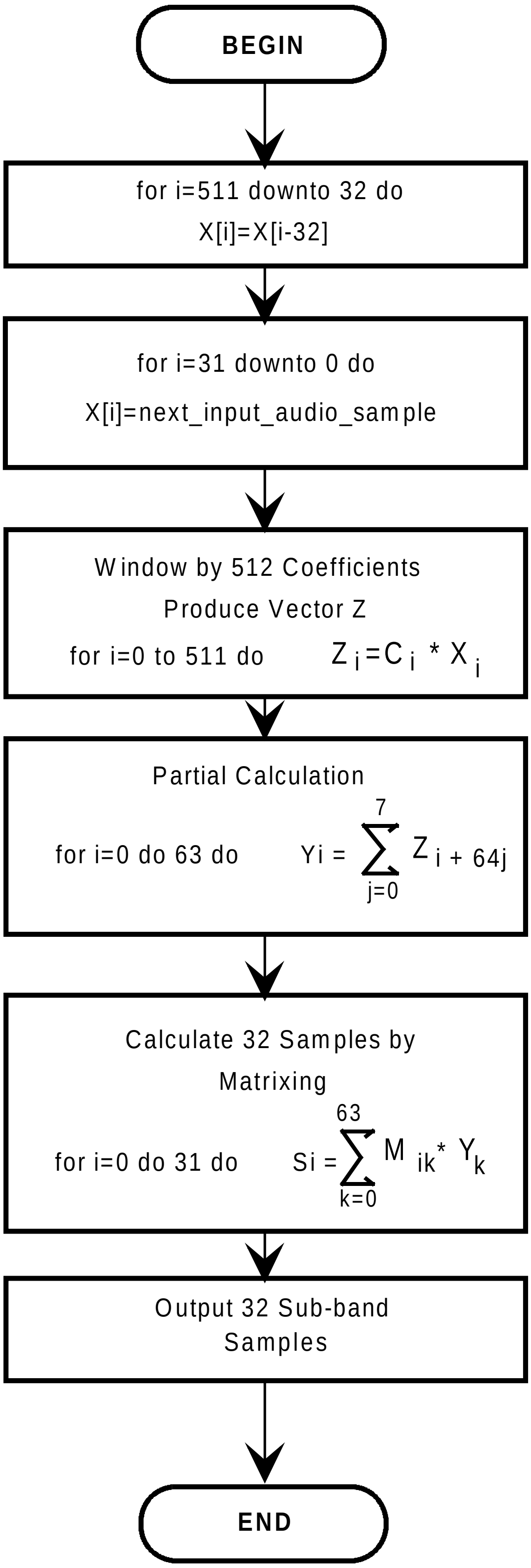


**Figure C.1: Analysis sub-band filter flow chart**

**Table C.1: Coefficients $C_i$ of the analysis window**

| | | | |
|---|---|---|---|
| C[ 0]= 0,000000000 | C[ 1]=-0,000000477 | C[ 2]=-0,000000477 | C[ 3]=-0,000000477 |
| C[ 4]=-0,000000477 | C[ 5]=-0,000000477 | C[ 6]=-0,000000477 | C[ 7]=-0,000000954 |
| C[ 8]=-0,000000954 | C[ 9]=-0,000000954 | C[ 10]=-0,000000954 | C[ 11]=-0,000001431 |
| C[ 12]=-0,000001431 | C[ 13]=-0,000001907 | C[ 14]=-0,000001907 | C[ 15]=-0,000002384 |
| C[ 16]=-0,000002384 | C[ 17]=-0,000002861 | C[ 18]=-0,000003338 | C[ 19]=-0,000003338 |
| C[ 20]=-0,000003815 | C[ 21]=-0,000004292 | C[ 22]=-0,000004768 | C[ 23]=-0,000005245 |
| C[ 24]=-0,000006199 | C[ 25]=-0,000006676 | C[ 26]=-0,000007629 | C[ 27]=-0,000008106 |
| C[ 28]=-0,000009060 | C[ 29]=-0,000010014 | C[ 30]=-0,000011444 | C[ 31]=-0,000012398 |
| C[ 32]=-0,000013828 | C[ 33]=-0,000014782 | C[ 34]=-0,000016689 | C[ 35]=-0,000018120 |
| C[ 36]=-0,000019550 | C[ 37]=-0,000021458 | C[ 38]=-0,000023365 | C[ 39]=-0,000025272 |
| C[ 40]=-0,000027657 | C[ 41]=-0,000030041 | C[ 42]=-0,000032425 | C[ 43]=-0,000034809 |
| C[ 44]=-0,000037670 | C[ 45]=-0,000040531 | C[ 46]=-0,000043392 | C[ 47]=-0,000046253 |
| C[ 48]=-0,000049591 | C[ 49]=-0,000052929 | C[ 50]=-0,000055790 | C[ 51]=-0,000059605 |
| C[ 52]=-0,000062943 | C[ 53]=-0,000066280 | C[ 54]=-0,000070095 | C[ 55]=-0,000073433 |
| C[ 56]=-0,000076771 | C[ 57]=-0,000080585 | C[ 58]=-0,000083923 | C[ 59]=-0,000087261 |
| C[ 60]=-0,000090599 | C[ 61]=-0,000093460 | C[ 62]=-0,000096321 | C[ 63]=-0,000099182 |
| C[ 64]= 0,000101566 | C[ 65]= 0,000103951 | C[ 66]= 0,000105858 | C[ 67]= 0,000107288 |
| C[ 68]= 0,000108242 | C[ 69]= 0,000108719 | C[ 70]= 0,000108719 | C[ 71]= 0,000108242 |
| C[ 72]= 0,000106812 | C[ 73]= 0,000105381 | C[ 74]= 0,000102520 | C[ 75]= 0,000099182 |
| C[ 76]= 0,000095367 | C[ 77]= 0,000090122 | C[ 78]= 0,000084400 | C[ 79]= 0,000077724 |
| C[ 80]= 0,000069618 | C[ 81]= 0,000060558 | C[ 82]= 0,000050545 | C[ 83]= 0,000039577 |
| C[ 84]= 0,000027180 | C[ 85]= 0,000013828 | C[ 86]=-0,000000954 | C[ 87]=-0,000017166 |
| C[ 88]=-0,000034332 | C[ 89]=-0,000052929 | C[ 90]=-0,000072956 | C[ 91]=-0,000093937 |
| C[ 92]=-0,000116348 | C[ 93]=-0,000140190 | C[ 94]=-0,000165462 | C[ 95]=-0,000191212 |
| C[ 96]=-0,000218868 | C[ 97]=-0,000247478 | C[ 98]=-0,000277042 | C[ 99]=-0,000307560 |
| C[100]=-0,000339031 | C[101]=-0,000371456 | C[102]=-0,000404358 | C[103]=-0,000438213 |
| C[104]=-0,000472546 | C[105]=-0,000507355 | C[106]=-0,000542164 | C[107]=-0,000576973 |
| C[108]=-0,000611782 | C[109]=-0,000646591 | C[110]=-0,000680923 | C[111]=-0,000714302 |
| C[112]=-0,000747204 | C[113]=-0,000779152 | C[114]=-0,000809669 | C[115]=-0,000838757 |
| C[116]=-0,000866413 | C[117]=-0,000891685 | C[118]=-0,000915051 | C[119]=-0,000935555 |
| C[120]=-0,000954151 | C[121]=-0,000968933 | C[122]=-0,000980854 | C[123]=-0,000989437 |
| C[124]=-0,000994205 | C[125]=-0,000995159 | C[126]=-0,000991821 | C[127]=-0,000983715 |
| C[128]= 0,000971317 | C[129]= 0,000953674 | C[130]= 0,000930786 | C[131]= 0,000902653 |
| C[132]= 0,000868797 | C[133]= 0,000829220 | C[134]= 0,000783920 | C[135]= 0,000731945 |
| C[136]= 0,000674248 | C[137]= 0,000610352 | C[138]= 0,000539303 | C[139]= 0,000462532 |
| C[140]= 0,000378609 | C[141]= 0,000288486 | C[142]= 0,000191689 | C[143]= 0,000088215 |
| C[144]=-0,000021458 | C[145]=-0,000137329 | C[146]=-0,000259876 | C[147]=-0,000388145 |
| C[148]=-0,000522137 | C[149]=-0,000661850 | C[150]=-0,000806808 | C[151]=-0,000956535 |
| C[152]=-0,001111031 | C[153]=-0,001269817 | C[154]=-0,001432419 | C[155]=-0,001597881 |
| C[156]=-0,001766682 | C[157]=-0,001937389 | C[158]=-0,002110004 | C[159]=-0,002283096 |
| C[160]=-0,002457142 | C[161]=-0,002630711 | C[162]=-0,002803326 | C[163]=-0,002974033 |
| C[164]=-0,003141880 | C[165]=-0,003306866 | C[166]=-0,003467083 | C[167]=-0,003622532 |
| C[168]=-0,003771782 | C[169]=-0,003914356 | C[170]=-0,004048824 | C[171]=-0,004174709 |
| C[172]=-0,004290581 | C[173]=-0,004395962 | C[174]=-0,004489899 | C[175]=-0,004570484 |

**(continued)**

**Table C.1: Coefficients $C_i$ of the analysis window (continued)**

| | | | |
|---|---|---|---|
| C[176]=-0,004638195 | C[177]=-0,004691124 | C[178]=-0,004728317 | C[179]=-0,004748821 |
| C[180]=-0,004752159 | C[181]=-0,004737377 | C[182]=-0,004703045 | C[183]=-0,004649162 |
| C[184]=-0,004573822 | C[185]=-0,004477024 | C[186]=-0,004357815 | C[187]=-0,004215240 |
| C[188]=-0,004049301 | C[189]=-0,003858566 | C[190]=-0,003643036 | C[191]=-0,003401756 |
| C[192]= 0,003134727 | C[193]= 0,002841473 | C[194]= 0,002521515 | C[195]= 0,002174854 |
| C[196]= 0,001800537 | C[197]= 0,001399517 | C[198]= 0,000971317 | C[199]= 0,000515938 |
| C[200]= 0,000033379 | C[201]=-0,000475883 | C[202]=-0,001011848 | C[203]=-0,001573563 |
| C[204]=-0,002161503 | C[205]=-0,002774239 | C[206]=-0,003411293 | C[207]=-0,004072189 |
| C[208]=-0,004756451 | C[209]=-0,005462170 | C[210]=-0,006189346 | C[211]=-0,006937027 |
| C[212]=-0,007703304 | C[213]=-0,008487225 | C[214]=-0,009287834 | C[215]=-0,010103703 |
| C[216]=-0,010933399 | C[217]=-0,011775017 | C[218]=-0,012627602 | C[219]=-0,013489246 |
| C[220]=-0,014358521 | C[221]=-0,015233517 | C[222]=-0,016112804 | C[223]=-0,016994476 |
| C[224]=-0,017876148 | C[225]=-0,018756866 | C[226]=-0,019634247 | C[227]=-0,020506859 |
| C[228]=-0,021372318 | C[229]=-0,022228718 | C[230]=-0,023074150 | C[231]=-0,023907185 |
| C[232]=-0,024725437 | C[233]=-0,025527000 | C[234]=-0,026310921 | C[235]=-0,027073860 |
| C[236]=-0,027815342 | C[237]=-0,028532982 | C[238]=-0,029224873 | C[239]=-0,029890060 |
| C[240]=-0,030526638 | C[241]=-0,031132698 | C[242]=-0,031706810 | C[243]=-0,032248020 |
| C[244]=-0,032754898 | C[245]=-0,033225536 | C[246]=-0,033659935 | C[247]=-0,034055710 |
| C[248]=-0,034412861 | C[249]=-0,034730434 | C[250]=-0,035007000 | C[251]=-0,035242081 |
| C[252]=-0,035435200 | C[253]=-0,035586357 | C[254]=-0,035694122 | C[255]=-0,035758972 |
| C[256]= 0,035780907 | C[257]= 0,035758972 | C[258]= 0,035694122 | C[259]= 0,035586357 |
| C[260]= 0,035435200 | C[261]= 0,035242081 | C[262]= 0,035007000 | C[263]= 0,034730434 |
| C[264]= 0,034412861 | C[265]= 0,034055710 | C[266]= 0,033659935 | C[267]= 0,033225536 |
| C[268]= 0,032754898 | C[269]= 0,032248020 | C[270]= 0,031706810 | C[271]= 0,031132698 |
| C[272]= 0,030526638 | C[273]= 0,029890060 | C[274]= 0,029224873 | C[275]= 0,028532982 |
| C[276]= 0,027815342 | C[277]= 0,027073860 | C[278]= 0,026310921 | C[279]= 0,025527000 |
| C[280]= 0,024725437 | C[281]= 0,023907185 | C[282]= 0,023074150 | C[283]= 0,022228718 |
| C[284]= 0,021372318 | C[285]= 0,020506859 | C[286]= 0,019634247 | C[287]= 0,018756866 |
| C[288]= 0,017876148 | C[289]= 0,016994476 | C[290]= 0,016112804 | C[291]= 0,015233517 |
| C[292]= 0,014358521 | C[293]= 0,013489246 | C[294]= 0,012627602 | C[295]= 0,011775017 |
| C[296]= 0,010933399 | C[297]= 0,010103703 | C[298]= 0,009287834 | C[299]= 0,008487225 |
| C[300]= 0,007703304 | C[301]= 0,006937027 | C[302]= 0,006189346 | C[303]= 0,005462170 |
| C[304]= 0,004756451 | C[305]= 0,004072189 | C[306]= 0,003411293 | C[307]= 0,002774239 |
| C[308]= 0,002161503 | C[309]= 0,001573563 | C[310]= 0,001011848 | C[311]= 0,000475883 |
| C[312]=-0,000033379 | C[313]=-0,000515938 | C[314]=-0,000971317 | C[315]=-0,001399517 |
| C[316]=-0,001800537 | C[317]=-0,002174854 | C[318]=-0,002521515 | C[319]=-0,002841473 |
| C[320]= 0,003134727 | C[321]= 0,003401756 | C[322]= 0,003643036 | C[323]= 0,003858566 |
| C[324]= 0,004049301 | C[325]= 0,004215240 | C[326]= 0,004357815 | C[327]= 0,004477024 |
| C[328]= 0,004573822 | C[329]= 0,004649162 | C[330]= 0,004703045 | C[331]= 0,004737377 |
| C[332]= 0,004752159 | C[333]= 0,004748821 | C[334]= 0,004728317 | C[335]= 0,004691124 |
| C[336]= 0,004638195 | C[337]= 0,004570484 | C[338]= 0,004489899 | C[339]= 0,004395962 |
| C[340]= 0,004290581 | C[341]= 0,004174709 | C[342]= 0,004048824 | C[343]= 0,003914356 |
| C[344]= 0,003771782 | C[345]= 0,003622532 | C[346]= 0,003467083 | C[347]= 0,003306866 |
| C[348]= 0,003141880 | C[349]= 0,002974033 | C[350]= 0,002803326 | C[351]= 0,002630711 |
| C[352]= 0,002457142 | C[353]= 0,002283096 | C[354]= 0,002110004 | C[355]= 0,001937389 |

**(continued)**

**Table C.1: Coefficients $C_i$ of the analysis window (concluded)**

| | | | |
|---|---|---|---|
| C[356]= 0,001766682 | C[357]= 0,001597881 | C[358]= 0,001432419 | C[359]= 0,001269817 |
| C[360]= 0,001111031 | C[361]= 0,000956535 | C[362]= 0,000806808 | C[363]= 0,000661850 |
| C[364]= 0,000522137 | C[365]= 0,000388145 | C[366]= 0,000259876 | C[367]= 0,000137329 |
| C[368]= 0,000021458 | C[369]=-0,000088215 | C[370]=-0,000191689 | C[371]=-0,000288486 |
| C[372]=-0,000378609 | C[373]=-0,000462532 | C[374]=-0,000539303 | C[375]=-0,000610352 |
| C[376]=-0,000674248 | C[377]=-0,000731945 | C[378]=-0,000783920 | C[379]=-0,000829220 |
| C[380]=-0,000868797 | C[381]=-0,000902653 | C[382]=-0,000930786 | C[383]=-0,000953674 |
| C[384]= 0,000971317 | C[385]= 0,000983715 | C[386]= 0,000991821 | C[387]= 0,000995159 |
| C[388]= 0,000994205 | C[389]= 0,000989437 | C[390]= 0,000980854 | C[391]= 0,000968933 |
| C[392]= 0,000954151 | C[393]= 0,000935555 | C[394]= 0,000915051 | C[395]= 0,000891685 |
| C[396]= 0,000866413 | C[397]= 0,000838757 | C[398]= 0,000809669 | C[399]= 0,000779152 |
| C[400]= 0,000747204 | C[401]= 0,000714302 | C[402]= 0,000680923 | C[403]= 0,000646591 |
| C[404]= 0,000611782 | C[405]= 0,000576973 | C[406]= 0,000542164 | C[407]= 0,000507355 |
| C[408]= 0,000472546 | C[409]= 0,000438213 | C[410]= 0,000404358 | C[411]= 0,000371456 |
| C[412]= 0,000339031 | C[413]= 0,000307560 | C[414]= 0,000277042 | C[415]= 0,000247478 |
| C[416]= 0,000218868 | C[417]= 0,000191212 | C[418]= 0,000165462 | C[419]= 0,000140190 |
| C[420]= 0,000116348 | C[421]= 0,000093937 | C[422]= 0,000072956 | C[423]= 0,000052929 |
| C[424]= 0,000034332 | C[425]= 0,000017166 | C[426]= 0,000000954 | C[427]=-0,000013828 |
| C[428]=-0,000027180 | C[429]=-0,000039577 | C[430]=-0,000050545 | C[431]=-0,000060558 |
| C[432]=-0,000069618 | C[433]=-0,000077724 | C[434]=-0,000084400 | C[435]=-0,000090122 |
| C[436]=-0,000095367 | C[437]=-0,000099182 | C[438]=-0,000102520 | C[439]=-0,000105381 |
| C[440]=-0,000106812 | C[441]=-0,000108242 | C[442]=-0,000108719 | C[443]=-0,000108719 |
| C[444]=-0,000108242 | C[445]=-0,000107288 | C[446]=-0,000105858 | C[447]=-0,000103951 |
| C[448]= 0,000101566 | C[449]= 0,000099182 | C[450]= 0,000096321 | C[451]= 0,000093460 |
| C[452]= 0,000090599 | C[453]= 0,000087261 | C[454]= 0,000083923 | C[455]= 0,000080585 |
| C[456]= 0,000076771 | C[457]= 0,000073433 | C[458]= 0,000070095 | C[459]= 0,000066280 |
| C[460]= 0,000062943 | C[461]= 0,000059605 | C[462]= 0,000055790 | C[463]= 0,000052929 |
| C[464]= 0,000049591 | C[465]= 0,000046253 | C[466]= 0,000043392 | C[467]= 0,000040531 |
| C[468]= 0,000037670 | C[469]= 0,000034809 | C[470]= 0,000032425 | C[471]= 0,000030041 |
| C[472]= 0,000027657 | C[473]= 0,000025272 | C[474]= 0,000023365 | C[475]= 0,000021458 |
| C[476]= 0,000019550 | C[477]= 0,000018120 | C[478]= 0,000016689 | C[479]= 0,000014782 |
| C[480]= 0,000013828 | C[481]= 0,000012398 | C[482]= 0,000011444 | C[483]= 0,000010014 |
| C[484]= 0,000009060 | C[485]= 0,000008106 | C[486]= 0,000007629 | C[487]= 0,000006676 |
| C[488]= 0,000006199 | C[489]= 0,000005245 | C[490]= 0,000004768 | C[491]= 0,000004292 |
| C[492]= 0,000003815 | C[493]= 0,000003338 | C[494]= 0,000003338 | C[495]= 0,000002861 |
| C[496]= 0,000002384 | C[497]= 0,000002384 | C[498]= 0,000001907 | C[499]= 0,000001907 |
| C[500]= 0,000001431 | C[501]= 0,000001431 | C[502]= 0,000000954 | C[503]= 0,000000954 |
| C[504]= 0,000000954 | C[505]= 0,000000954 | C[506]= 0,000000477 | C[507]= 0,000000477 |
| C[508]= 0,000000477 | C[509]= 0,000000477 | C[510]= 0,000000477 | C[511]= 0,000000477 |

## C.2 Psychoacoustic model

For each frame, corresponding to 1 152 input samples, a bit allocation shall be determined. The bit allocation of the 32 sub-bands should be calculated on the basis of the signal-to-mask ratios of all the sub-bands. Therefore it is necessary to determine, for each sub-band the maximum signal level and the minimum masking threshold in dB. The minimum masking threshold is derived from an Fast Fourier Transform (FFT) of the input PCM signal, followed by a psychoacoustic model calculation.

The FFT in parallel with the sub-band filter compensates for the lack of spectral selectivity obtained at low frequencies by the sub-band filter bank. This technique provides both a sufficient time resolution for the coded audio signal (Polyphase filter with optimized window for minimal pre-echoes) and a sufficient spectral resolution for the calculation of the masking thresholds.

The frequencies and levels of aliasing distortions can be calculated. This is necessary for calculating a minimum bit rate for those sub-bands which need some bits to cancel the aliasing components in the decoder. The additional complexity to calculate the better frequency resolution is necessary only in the encoder, and introduces no additional delay or complexity in the decoder.

The calculation of the signal-to-mask-ratio (SMR) is based on the following steps:

**Step 1** calculation of the FFT for time to frequency conversion;

**Step 2** determination of the sound pressure level in dB in each sub-band;

**Step 3** determination of the threshold in quiet (absolute threshold);

**Step 4** finding of the tonal (more sinusoid-like) and non-tonal (more noise-like) components of the audio signal;

**Step 5** decimation of the maskers, to obtain only the relevant maskers;

**Step 6** calculation of the individual masking thresholds;

**Step 7** determination of the global masking threshold;

**Step 8** determination of the minimum masking threshold in each sub-band;

**Step 9** calculation of the signal-to-mask ratio in each sub-band.

The following gives further details on the above steps.

**Step 1:** FFT Analysis.

The masking threshold is derived from an estimate of the power density spectrum that is calculated by a 1 024-point FFT. The FFT is calculated directly from the input PCM signal, windowed by a Hann window.

For a coincidence in time between the bit allocation and the corresponding sub-band samples, the PCM-samples entering the FFT have to be delayed:

1) the delay of the analysis sub-band filter is 256 samples, corresponding to 5,3 ms at 48 kHz sampling rate. A window shift of 256 samples is required to compensate for the delay in the analysis sub-band filter;

2) the Hann window shall coincide with the sub-band samples of the frame. This requires an additional window shift of minus 64 samples.

Technical data of the FFT:

- transform length N 1 024 samples;
- Window size 21,3 ms;
- Frequency resolution 46,875 Hz;
- Hann window, h(i): $h(i) = \sqrt{8/3} * 0,5 * \{1 - \cos[2 * \pi * i/N]\} \quad 0 \le i \le N\text{-}1$;
- power density spectrum $X(k)$:

$$X(k) = 10 * \log_{10} \left| \frac{1}{N} \sum_{l=0}^{N-1} h(l) * s(l) * e^{(-j * k * l * 2 * \pi/N)} \right|^2 dB \quad 0 \le k \le N/2$$

where $s(l)$ is the input signal.

A normalization to the reference level of 96 dB SPL (Sound Pressure Level) should be done in such a way that the maximum value corresponds to 96 dB.

**Step 2:** Determination of the Sound Pressure Level.

The SPL $L_{sb}$ in sub-band $n$ should be computed by:

$$L_{sb}(n) = MAX[\ X(k), 20 * \log_{10}(scf_{max}(n) * 32\ 768) - 10\ ]\ dB$$

$X(k)$ in sub-band $n$

where $X(k)$ is the Sound Pressure Level of the spectral line with index $k$ of the FFT with the maximum amplitude in the frequency range corresponding to sub-band $n$. The expression $scf_{max}(n)$ is the maximum of the three Scale Factors of sub-band $n$ within a frame. The "$-10\ dB$" term corrects for the difference between peak and rms level. The Sound Pressure Level $L_{sb}(n)$ is computed for every sub-band $n$.

The following alternative method of calculating $L_{sb}(n)$ offers a potential for better encoder performance, but this technique has not been subjected to a formal audio quality test.

The alternative SPL $L_{sb}(n)$ in sub-band $n$ should be computed by:

$$L_{sb}(n) = MAX[\ X_{spl}(n), 20 * \log_{10}(scf_{max}(n) * 32\ 768)-10\ ]\ dB;$$

with,

$$X_{spl}(n) = 10\log\left(\sum_{k(n)}^{k(n+1)} 10^{X(k)/10}\right)\ dB,$$

*where* $k(n) = n\ N/64$

*and* $X_{spl}(n)$ is the alternative Sound Pressure Level corresponding to sub-band $n$.

**Step 3:** Considering the threshold in quiet.

The threshold in quiet $LT_q(k)$, also called absolute threshold, is available in the table C.2. Values are available for each sample in the frequency domain where the masking threshold is calculated.

An offset depending on the overall bit rate should be used for the absolute threshold. This offset is - 12 dB for bit rates ≥ 96 kbit/s and 0 dB for bit rates < 96 kbit/s per channel.

**Step 4:** Finding of tonal and non-tonal components.

The tonality of a masking component has an influence on the masking threshold. For this reason, it is worthwhile to discriminate between tonal and non-tonal components. For calculating the global masking threshold it is necessary to derive the tonal and the non-tonal components from the FFT spectrum.

This step should start with the determination of local maxima, continued by extracting tonal components (sinusoids) and calculating the intensity of the non-tonal components within a bandwidth of a Critical band. The boundaries of the Critical bands are given in the table B.3.

The bandwidth of the Critical bands varies with the centre frequency with a bandwidth of about only 0,1 kHz at low frequencies and with a bandwidth of about 4 kHz at high frequencies. It is known from psychoacoustic experiments that the ear has a better frequency resolution in the lower than in the higher frequency region. To determine if a local maximum may be a tonal component a frequency range $df$ around the local maximum should be examined. The frequency range $df$ is given by:

$$df = 93{,}75 \text{ Hz} \quad 0{,}0 \text{ kHz} < f \leq 3{,}0 \text{ kHz};$$

$$df = 140{,}63 \text{ Hz} \quad 3{,}0 \text{ kHz} < f \leq 6{,}0 \text{ kHz};$$

$$df = 281{,}25 \text{ Hz} \quad 6{,}0 \text{ kHz} < f \leq 12{,}0 \text{ kHz};$$

$$df = 562{,}50 \text{ Hz} \quad 12{,}0 \text{ kHz} < f \leq 24{,}0 \text{ kHz}.$$

To make lists of the spectral lines $X(k)$ that are tonal or non-tonal, the following three operations are performed:

**a) Labelling of local maxima:**

- a spectral line $X(k)$ is labelled as a local maximum if;

$$X(k) > X(k-1) \quad \text{and} \quad X(k) \geq X(k+1)$$

**b) Listing of tonal components and calculation of the Sound Pressure Level:**

- a local maximum is put in the list of tonal components if;

$$X(k) - X(k+j) \geq 7 \text{ dB},$$

where $j$ is chosen according to:

| | | | |
|---|---|---|---|
| j = - 2, +2 | for | 2 < k < | 63; |
| j = - 3, -2, +2, +3 | for | 63 ≤ k < | 127; |
| j = - 6,..., -2, +2,..., +6 | for | 127 ≤ k < | 255; |
| j = - 12,..., -2, +2,..., +12 | for | 255≤ k ≤ | 500. |

If $X(k)$ is found to be a tonal component, then the following parameters are listed:

- index number $k$ of the spectral line;
- SPL $X_{tm}(k)=10 * \log_{10} (10^{X(k-1)/10} + 10^{X(k)/10} + 10^{X(k+1)/10})$ dB;
- tonal flag.

Next, all spectral lines within the examined frequency range are set to -∞ dB.

**c) Listing of non-tonal components and calculation of the power**

The non-tonal (noise) components are calculated from the remaining spectral lines. To calculate the non-tonal components from these spectral lines $X(k)$, the Critical bands $z(k)$ are determined using the table C.3. For 48 kHz sampling frequency 26 Critical bands are considered. Within each Critical band, the power of the spectral lines (remained after the tonal components have been zeroed) are summed to form the Sound Pressure Level of the new non-tonal component $X_{nm}(k)$ corresponding to that Critical band.

The following parameters are listed:

- index number $k$ of the spectral line nearest to the geometric mean of the Critical band;
- SPL $X_{nm}(k)$ in dB;
- non-tonal flag.

**Step 5:** Decimation of tonal and non-tonal masking components.

Decimation is a procedure that is used to reduce the number of maskers which are considered for the calculation of the global masking threshold:

a) Tonal $X_{tm}(k)$ or non-tonal components $X_{nm}(k)$ are considered for the calculation of the masking threshold only if:

- $X_{tm}(k) \geq LT_q(k) \quad or \quad X_{nm}(k) \geq LT_q(k).$

In this expression, $LT_q(k)$ is the absolute threshold (or threshold in quiet) at the frequency of index $k$. These values are given in the table C.2.

b) Decimation of two or more tonal components within a distance of less then 0,5 Bark: The component with the highest power should be kept, and the smaller component(s) should be removed from the list of tonal components. For this operation, a sliding window in the Critical band domain should be used with a width of 0,5 Bark.

In the following, the index $j$ is used to indicate the relevant tonal or non-tonal masking components from the combined decimated list.

**Step 6:** Calculation of individual masking thresholds.

Of the original 512 frequency domain samples, indexed by $k$, only a subset of the samples, indexed by $i$, are considered for the global masking threshold calculation. The samples used are shown in table C.2.

For the frequency lines corresponding to the frequency region which is covered by the first three sub-bands no sub-sampling is used. For the frequency region which is covered by next three sub-bands every second spectral line is considered. For the frequency region corresponding to the next six sub-bands every fourth spectral line is considered. Finally, in the remaining sub-bands every eighth spectral line is considered up to 20 kHz (see also table C.2). The number of samples, $i$, in the sub-sampled frequency domain is 126.

To every tonal and non-tonal component the index $i$ in the sub-sampled frequency domain is assigned, which is closest in frequency to the original spectral line $X(k)$. This index $i$ is given also in table C.2.

The individual masking thresholds of both tonal and non-tonal components are given by the following expression:

$$LT_{tm}[z(j),z(i)] \;=\; X_{tm}[z(j)] + av_{tm}[z(j)] + vf[z(j),z(i)] \quad dB;$$

$$LT_{nm}[z(j),z(i)] \;=\; X_{nm}[z(j)] + av_{nm}[z(j)] + vf[z(j),z(i)] \quad dB.$$

In this formula $LT_{tm}$ and $LT_{nm}$ are the individual masking thresholds at Critical band rate $z$ in Bark of the masking component at the Critical band rate $z_m$ in Bark. The values in dB can be either positive or negative. The term $X_{tm}[z(j)]$ is the Sound Pressure Level of the masking component with the index number $j$ at the corresponding Critical band rate $z(j)$. The term $av$ is called the masking index and $vf$ the masking function of the masking component $X_{tm}[z(j)]$. The masking index $av$ is different for tonal and non-tonal masker ($av_{tm}$ and $av_{nm}$).

For tonal maskers it is given by:

$$av_{tm} = -1{,}525 - 0{,}275 * z(j) - 4{,}5 \; dB,$$

and for non-tonal maskers:

$$av_{nm} = -1{,}525 - 0{,}175 * z(j) - 0{,}5 \; dB.$$

The masking function $vf$ of a masker is characterized by different lower and upper slopes, which depend on the distance in Bark $dz = z(i) - z(j)$ to the masker. In this expression $i$ is the index of the spectral line at which the masking function is calculated and $j$ that of the masker. The Critical band rates $z(j)$ and $z(i)$ can be found in table C.2. The masking function, which is the same for tonal and non-tonal maskers, is given by:

$$vf = 17 * (dz + 1) - (0{,}4 * X[z(j)] + 6) \quad dB \qquad \text{for } -3 \le dz < -1 \text{ Bark};$$

$$vf = (0{,}4 * X[z(j)] + 6) * dz \quad dB \qquad \text{for } -1 \le dz < 0 \text{ Bark};$$

$$vf = -17 * dz \quad dB \qquad \text{for } 0 \le dz < 1 \text{ Bark};$$

$$vf = -(dz - 1) * (17 - 0{,}15 * X[z(j)]) - 17 \quad dB \qquad \text{for } 1 \le dz < 8 \text{ Bark}.$$

In these expressions $X[z(j)]$ is the Sound Pressure Level of the j'th masking component in dB. For reasons of implementation complexity, the masking should no longer considered ($LT_{tm}$ and $LT_{nm}$ are set to $-\infty$ dB outside this range) if $dz < -3$ Bark, or $dz \ge 8$ Bark.

**Step 7:** Calculation of the global masking threshold $LT_g$.

The global masking threshold $LT_g(i)$ at the i'th frequency sample is derived from the upper and lower slopes of the individual masking threshold of each of the $j$ tonal and non-tonal maskers, and in addition from the threshold in quiet $LT_q(i)$, which is also given in table C.2. The global masking threshold is found by summing the powers corresponding to the individual masking thresholds and the threshold in quiet.

$$LT_g(i) = 10\log_{10}(10^{LT_q(i)/10} + \sum_{j=1}^{m} 10^{LT_{tm}[z(j),z(i)]/10} + \sum_{j=1}^{n} 10^{LT_{nm}[z(j),z(i)]/10} \ dB$$

The total number of tonal maskers is given by $m$, and the total number of non-tonal maskers is given by $n$. For a given $i$, the range of $j$ can be reduced to just encompass those masking components that are within - 8 to +3 Bark from i. Outside of this range $LT_{tm}$ and $LT_{nm}$ are -∞ dB.

**Step 8:** Determination of the minimum masking threshold.

The minimum masking level $LT_{min}(n)$ in sub-band $n$ is determined by the following expression:

$$LT_{min}(n) \quad = \quad MIN[\ LT_g(i)\ ] \ \ dB$$

$f(i)$ in sub-band n

where $f(i)$ is the frequency of the i'th frequency sample. The $f(i)$ are tabulated in the table C.2.

A minimum masking level $LT_{min}(n)$ is computed for every sub-band.

**Step 9:** Calculation of the Signal-to-Mask-Ratio

The Signal-to-Mask Ratio

$$SMR_{sb}(n) \quad = \quad L_{sb}(n)\text{-}LT_{min}(n) \quad dB$$

is computed for every sub-band $n$.

**Table C.2: Frequencies, Critical band rates and absolute threshold**

| Index Number i | Frequency Hz | Critical Band Rate z | Absolute Threshold dB | Index Number i | Frequency Hz | Critical Band Rate z | Absolute Threshold dB |
|---|---|---|---|---|---|---|---|
| 1 | 46,88 | ,463 | 42,10 | 39 | 1 828,13 | 12,518 | 0,49 |
| 2 | 93,75 | ,925 | 24,17 | 40 | 1 875,00 | 12,684 | 0,29 |
| 3 | 140,63 | 1,385 | 17,47 | 41 | 1 921,88 | 12,845 | 0,09 |
| 4 | 187,50 | 1,842 | 13,87 | 42 | 1 968,75 | 13,002 | - 0,11 |
| 5 | 234,38 | 2,295 | 11,60 | 43 | 2 015,63 | 13,154 | - 0,32 |
| 6 | 281,25 | 2,742 | 10,01 | 44 | 2 062,50 | 13,302 | - 0,54 |
| 7 | 328,13 | 3,184 | 8,84 | 45 | 2 109,38 | 13,446 | - 0,75 |
| 8 | 375,00 | 3,618 | 7,94 | 46 | 2 156,25 | 13,586 | - 0,97 |
| 9 | 421,88 | 4,045 | 7,22 | 47 | 2 203,13 | 13,723 | - 1,20 |
| 10 | 468,75 | 4,463 | 6,62 | 48 | 2 250,00 | 13,855 | - 1,43 |
| 11 | 515,63 | 4,872 | 6,12 | 49 | 2 343,75 | 14,111 | - 1,88 |
| 12 | 562,50 | 5,272 | 5,70 | 50 | 2 437,50 | 14,354 | - 2,34 |
| 13 | 609,38 | 5,661 | 5,33 | 51 | 2 531,25 | 14,585 | - 2,79 |
| 14 | 656,25 | 6,041 | 5,00 | 52 | 2 625,00 | 14,807 | - 3,22 |
| 15 | 703,13 | 6,411 | 4,71 | 53 | 2 718,75 | 15,018 | - 3,62 |
| 16 | 750,00 | 6,770 | 4,45 | 54 | 2 812,50 | 15,221 | - 3,98 |
| 17 | 796,88 | 7,119 | 4,21 | 55 | 2 906,25 | 15,415 | - 4,30 |
| 18 | 843,75 | 7,457 | 4,00 | 56 | 3 000,00 | 15,602 | - 4,57 |
| 19 | 890,63 | 7,785 | 3,79 | 57 | 3 093,75 | 15,783 | - 4,77 |
| 20 | 937,50 | 8,103 | 3,61 | 58 | 3 187,50 | 15,956 | - 4,91 |
| 21 | 984,38 | 8,410 | 3,43 | 59 | 3 281,25 | 16,124 | - 4,98 |
| 22 | 1 031,25 | 8,708 | 3,26 | 60 | 3 375,00 | 16,287 | - 4,97 |
| 23 | 1 078,13 | 8,996 | 3,09 | 61 | 3 468,75 | 16,445 | - 4,90 |
| 24 | 1 125,00 | 9,275 | 2,93 | 62 | 3 562,50 | 16,598 | - 4,76 |
| 25 | 1 171,88 | 9,544 | 2,78 | 63 | 3 656,25 | 16,746 | - 4,55 |
| 26 | 1 218,75 | 9,805 | 2,63 | 64 | 3 750,00 | 16,891 | - 4,29 |
| 27 | 1 265,63 | 10,057 | 2,47 | 65 | 3 843,75 | 17,032 | - 3,99 |
| 28 | 1 312,50 | 10,301 | 2,32 | 66 | 3 937,50 | 17,169 | - 3,64 |
| 29 | 1 359,38 | 10,537 | 2,17 | 67 | 4 031,25 | 17,303 | - 3,26 |
| 30 | 1 406,25 | 10,765 | 2,02 | 68 | 4 125,00 | 17,434 | - 2,86 |
| 31 | 1 453,13 | 10,986 | 1,86 | 69 | 4 218,75 | 17,563 | - 2,45 |
| 32 | 1 500,00 | 11,199 | 1,71 | 70 | 4 312,50 | 17,688 | - 2,04 |
| 33 | 1 546,88 | 11,406 | 1,55 | 71 | 4 406,25 | 17,811 | - 1,63 |
| 34 | 1 593,75 | 11,606 | 1,38 | 72 | 4 500,00 | 17,932 | - 1,24 |
| 35 | 1 640,63 | 11,800 | 1,21 | 73 | 4 687,50 | 18,166 | - 0,51 |
| 36 | 1 687,50 | 11,988 | 1,04 | 74 | 4 875,00 | 18,392 | 0,12 |
| 37 | 1 734,38 | 12,170 | 0,86 | 75 | 5 062,50 | 18,611 | 0,64 |
| 38 | 1 781,25 | 12,347 | 0,67 | 76 | 5 250,00 | 18,823 | 1,06 |

(continued)

**Table C.2: Frequencies, Critical band rates and absolute threshold (concluded):**

| Index Number i | Frequency Hz | Critical Band Rate z | Absolute Threshold dB |
|---|---|---|---|
| 77 | 5 437,50 | 19,028 | 1,39 |
| 78 | 5 625,00 | 19,226 | 1,66 |
| 79 | 5 812,50 | 19,419 | 1,88 |
| 80 | 6 000,00 | 19,606 | 2,08 |
| 81 | 6 187,50 | 19,788 | 2,27 |
| 82 | 6 375,00 | 19,964 | 2,46 |
| 83 | 6 562,50 | 20,135 | 2,65 |
| 84 | 6 750,00 | 20,300 | 2,86 |
| 85 | 6 937,50 | 20,461 | 3,09 |
| 86 | 7 125,00 | 20,616 | 3,33 |
| 87 | 7 312,50 | 20,766 | 3,60 |
| 88 | 7 500,00 | 20,912 | 3,89 |
| 89 | 7 687,50 | 21,052 | 4,20 |
| 90 | 7 875,00 | 21,188 | 4,54 |
| 91 | 8 062,50 | 21,318 | 4,91 |
| 92 | 8 250,00 | 21,445 | 5,31 |
| 93 | 8 437,50 | 21,567 | 5,73 |
| 94 | 8 625,00 | 21,684 | 6,18 |
| 95 | 8 812,50 | 21,797 | 6,67 |
| 96 | 9 000,00 | 21,906 | 7,19 |
| 97 | 9 375,00 | 22,113 | 8,33 |
| 98 | 9 750,00 | 22,304 | 9,63 |
| 99 | 10 125,00 | 22,482 | 11,08 |
| 100 | 10 500,00 | 22,646 | 12,71 |
| 101 | 10 875,00 | 22,799 | 14,53 |

| Index Number i | Frequency Hz | Critical Band Rate z | Absolute Threshold dB |
|---|---|---|---|
| 102 | 11 250,00 | 22,941 | 16,54 |
| 103 | 11 625,00 | 23,072 | 18,77 |
| 104 | 12 000,00 | 23,195 | 21,23 |
| 105 | 12 375,00 | 23,309 | 23,94 |
| 106 | 12 750,00 | 23,415 | 26,90 |
| 107 | 13 125,00 | 23,515 | 30,14 |
| 108 | 13 500,00 | 23,607 | 33,67 |
| 109 | 13 875,00 | 23,694 | 37,51 |
| 110 | 14 250,00 | 23,775 | 41,67 |
| 111 | 14 625,00 | 23,852 | 46,17 |
| 112 | 15 000,00 | 23,923 | 51,04 |
| 113 | 15 375,00 | 23,991 | 56,29 |
| 114 | 15 750,00 | 24,054 | 61,94 |
| 115 | 16 125,00 | 24,114 | 68,00 |
| 116 | 16 500,00 | 24,171 | 68,00 |
| 117 | 16 875,00 | 24,224 | 68,00 |
| 118 | 17 250,00 | 24,275 | 68,00 |
| 119 | 17 625,00 | 24,322 | 68,00 |
| 120 | 18 000,00 | 24,368 | 68,00 |
| 121 | 18 375,00 | 24,411 | 68,00 |
| 122 | 18 750,00 | 24,452 | 68,00 |
| 123 | 19 125,00 | 24,491 | 68,00 |
| 124 | 19 500,00 | 24,528 | 68,00 |
| 125 | 19 875,00 | 24,564 | 68,00 |
| 126 | 20 250,00 | 24,597 | 68,00 |

## C.3 Bit allocation procedure

Before adjustment to a fixed bit rate, the number of bits, *"adb"*, that are available for coding the samples and the Scale Factors must be determined. This number can be obtained by subtracting from the total number of available bits *"cb"*, the number of bits needed for Bit Allocation *"bbal"*, and the number of bits *"banc"* required for ancillary data:

$$adb = cb - (bbal + banc).$$

The resulting number should be used to code the sub-band samples and the Scale Factors. The principle used in the allocation procedure is minimization of the total Noise-to-Mask Ratio over the DAB audio frame with the constraint that the number of bits used does not exceed the number of bits available for that frame. Use is made of tables 13 and 14 in subclause 7.1, that indicate for every sub-band the number of steps that may be used to quantize the samples. The number of bits required to represent these quantized samples can be derived from table 16 of subclause 7.1.

**Table C.3: Critical band boundaries**

| no | index of table F&CB | frequency Hz | Bark z |
|---|---|---|---|
| 0 | 1 | 46,875 | 0,463 |
| 1 | 2 | 93,750 | 0,925 |
| 2 | 3 | 140,625 | 1,385 |
| 3 | 5 | 234,375 | 2,295 |
| 4 | 7 | 328,125 | 3,184 |
| 5 | 9 | 421,875 | 4,045 |
| 6 | 12 | 562,500 | 5,272 |
| 7 | 14 | 656,250 | 6,041 |
| 8 | 17 | 796,875 | 7,119 |
| 9 | 20 | 937,500 | 8,103 |
| 10 | 24 | 1 125,000 | 9,275 |
| 11 | 27 | 1 265,625 | 10,057 |
| 12 | 32 | 1 500,000 | 11,199 |
| 13 | 37 | 1 734,375 | 12,170 |
| 14 | 42 | 1 968,750 | 13,002 |
| 15 | 49 | 2 343,750 | 14,111 |
| 16 | 53 | 2 718,750 | 15,018 |
| 17 | 59 | 3 281,250 | 16,124 |
| 18 | 65 | 3 843,750 | 17,032 |
| 19 | 73 | 4 687,500 | 18,166 |
| 20 | 77 | 5 437,500 | 19,028 |
| 21 | 82 | 6 375,000 | 19,964 |
| 22 | 89 | 7 687,500 | 21,052 |
| 23 | 97 | 9 375,000 | 22,113 |
| 24 | 103 | 11 625,000 | 23,072 |
| 25 | 113 | 15 375,000 | 23,991 |
| 26 | 126 | 20 250,000 | 24,597 |

NOTE: The frequencies represent the top end of each critical band.

The allocation procedure is an iterative procedure where, in each iteration step the number of levels of the sub-band that has the greatest benefit is increased.

First the Mask-to-Noise Ratio *"MNR"* for each sub-band should be calculated by subtracting from the Signal-to-Noise-Ratio *"SNR"* the Signal-to-Mask-Ratio *"SMR"*:

$$MNR = SNR - SMR$$

The *SNR* can be found in the informative table C.4. The *SMR* is the output of the psychoacoustic model.

**Table C.4: Signal-to Noise-Ratios**

| No. of steps | SNR dB |
|---|---|
| 0 | 0,00 |
| 3 | 7,00 |
| 5 | 11,00 |
| 7 | 16,00 |
| 9 | 20,84 |
| 15 | 25,28 |
| 31 | 31,59 |
| 63 | 37,75 |
| 127 | 43,84 |
| 255 | 49,89 |
| 511 | 55,93 |
| 1 023 | 61,96 |
| 2 047 | 67,98 |
| 4 095 | 74,01 |
| 8 191 | 80,03 |
| 16 383 | 86,05 |
| 32 767 | 92,01 |
| 65 535 | 98,01 |

Then zero bits should be allocated to the sub-band samples and the Scale Factors. The number of bits for the sub-band samples *bspl* and the number of bits for the Scale Factors *bscf* are set to zero. Next an iterative procedure should be started. Each iteration loop should contain the following steps:

- determination of the minimal *MNR* of all sub-bands;
- the accuracy of the quantization of the sub-band with the minimal MNR should be increased by using the next higher entry in the relevant tables 13 and 14 of subclause 7.1;
- the new *MNR* of this sub-band should be calculated;
- *bspl* should be updated according to the additional number of bits required. If a non-zero number of bits is assigned to a sub-band for the first time, *bsel* has to be updated, and *bscf* has to be updated according to the number of Scale Factors required for this sub-band. Then *adb* should be calculated again using the formula:
    - adb = cb-(bbal + bsel + bscf + bspl + banc).

The iterative procedure should be repeated as long as *adb* is not less than any possible increase of *bspl*, *bsel* and *bscf* within one loop.

## C.4 Bit sensitivity to errors

This part of the annex indicates the sensitivity of individual bits to random errors if application-specific error protection is needed. This sensitivity for each bit is given in table C.5 by a value from 0 to 5, indicating the amount of degradation resulting from one isolated error:

- 5 catastrophic;
- 4 very annoying;
- 3 annoying;
- 2 slightly annoying;
- 1 audible;
- 0 insensitive.

The values are not the results of precise measurements, rather they rely upon knowledge of the coding scheme. They assume that the error detection scheme is not in use. The DAB audio frame header and error check information defined in subclauses 7.3.2.3 and 7.3.2.4 are considered to have the highest sensitivity.

Some fields in the DAB audio frame do not have a fixed length. All bits in this fields are rated for error sensitivity, even if not in use.

**Table C.5: Bit sensitivity of DAB audio frame bits**

| Parameters | Number of bits | sensitivity |
|---|---|---|
| Bit Allocation | all bits | 5 |
| ScFSI | all bits | 5 |
| ScFs | 5 (msb) | 4 |
| | 4 | 4 |
| | 3 | 4 |
| | 2 | 3 |
| | 1 | 2 |
| | 0 (lsb) | 1 |
| Sub-band samples (*) | 8 -16(msb) | 3 |
| | 5 - 7 | 2 |
| | 3,4 | 1 |
| | (lsb)0 - 2 | 0 |

NOTE: (*) according to the bit allocation.

## C.5 Error concealment

A feature of the coded bit stream is the CRC word which provides some error detection facility to the decoder, described in annex B, clause B.2. The Hamming distance of this error detection code is d = 4, which allows for the detection of up to 3 single bit errors or for the detection of one error burst of up to 16 bit length. The amount and the position of the protected bits within one encoded DAB audio frame generally depends on the mode and the bit rate.

The CRC word should be used to control an error concealment strategy in order to avoid severe impairments of the reconstructed audio signal due to errors in the most sensitive information.

Some basic techniques may be used for concealment, for instance information substitution, or muting. A simple substitution technique consists, when an erroneous frame occurs, of replacing it by the previous one (if error free).

In addition to the error protection facilities provided by the ISO/IEC 11172-3 [3] audio coding standard, facilities for an error check of the Scale Factors have been provided in an ISO compatible manner. The exact method is described in annex B, clause B.3. To avoid audible distortions, evoked by erroneous Scale Factors, the application of a concealment technique, either muting of those Scale Factors where an error was detected, as a rather simple method, or a repetition of the previously received Scale Factors, which did not show an error in the 3 MSBs, as a more advanced method, is recommended.

## C.6 Joint stereo coding

The optional Joint stereo coding method used is Intensity stereo coding. Intensity stereo coding can be used to increase the audio quality and/or reduce the bit rate for stereophonic signals. The gain in bit rate is typically about 10 to 30 kbit/s. It requires negligible additional decoder complexity. The increase of encoder complexity is small. The encoder and decoder delay is not affected.

Psychoacoustic results indicate that, at high frequencies (above about 2 kHz), the localization of the stereophonic image within a Critical band is determined by the temporal envelope and not by the temporal fine structure of the audio signal.

The basic idea for Intensity stereo coding is that for some sub-bands, instead of transmitting separate left and right sub-band samples, only the sum-signal should be transmitted, but with Scale Factors for both the left and right channels, thus preserving the stereophonic image.

Flow diagrams of a stereo encoder and decoder, including intensity stereo mode, are shown in figures C.2 and C.3. First, an estimation should be made of the required bit rate for both left and right channel. If the required bit rate exceeds the available bit rate, the required bit rate should be decreased by setting a number of sub-bands to Intensity stereo mode. Depending on the bit rate needed, sub-bands:

16 to 31;
12 to 31;
8 to 31; or
4 to 31.

can be set to Intensity stereo mode. For the quantization of such combined sub-bands, the higher of the BAs for left and right channel should be used.

The left and right sub-band signals of the sub-bands in Joint stereo mode should be added. These new sub-band signals should be scaled in the normal way, but the originally determined Scale Factors of the left and right sub-band signals should be transmitted according to the bitstream syntax. Quantization and coding of common sub-band samples, and coding of common Bit Allocation should be performed in the same way as in independent coding of the left and right channel of a stereophonic programme.

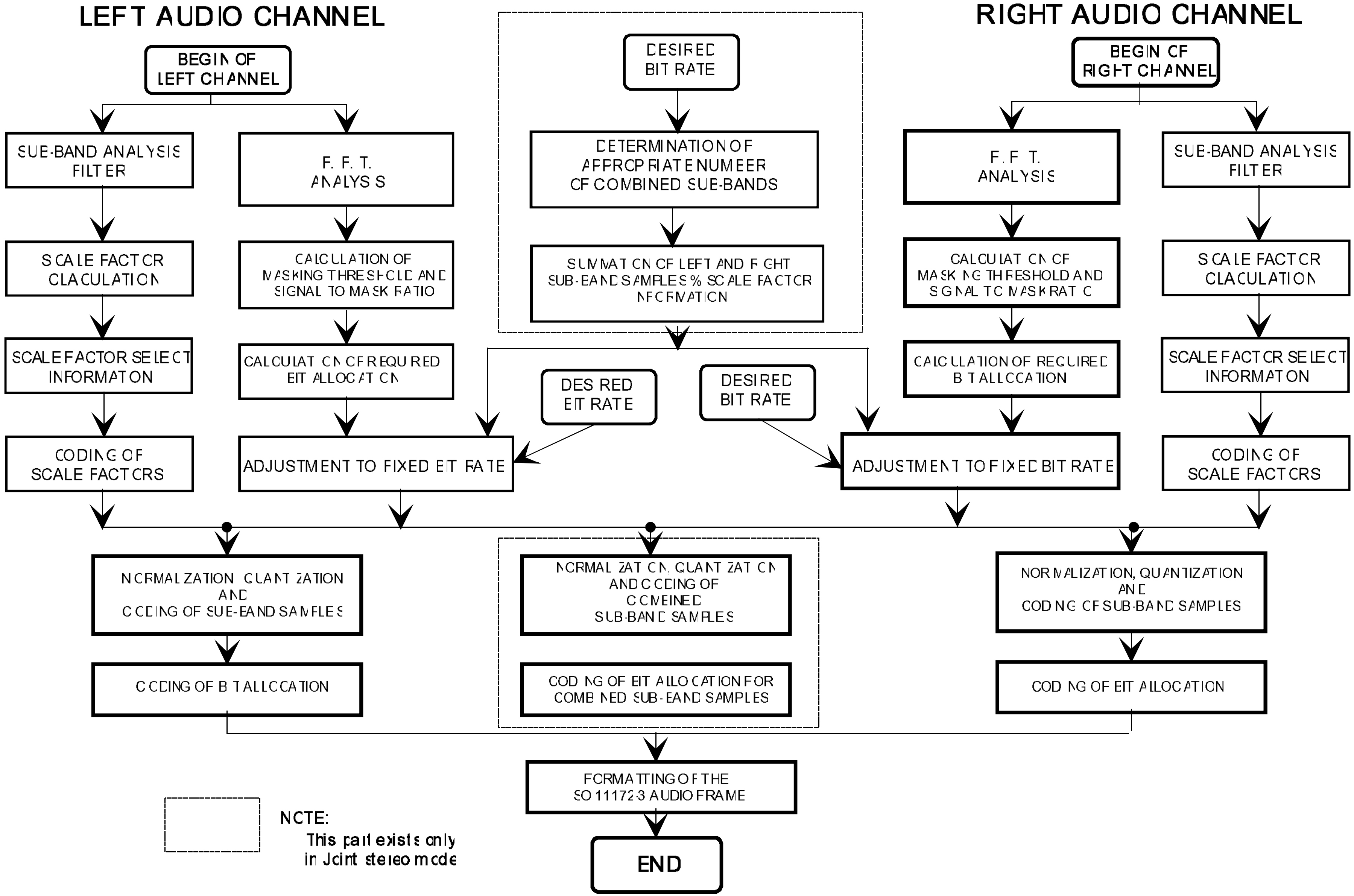


**Figure C.2: General ISO/IEC 11172-3 [3] Layer II stereo encoder flow chart**

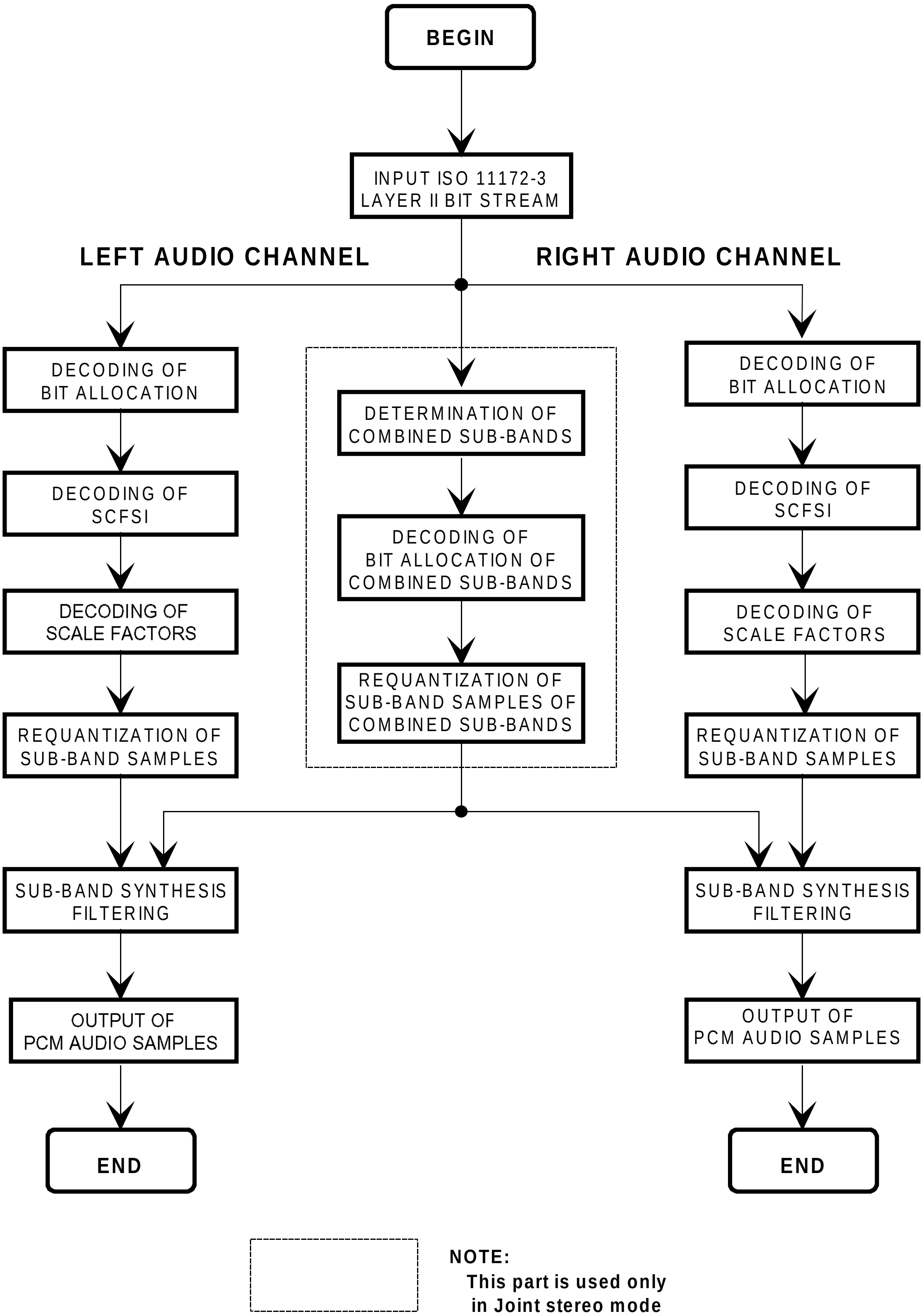


Figure C.3: General ISO/IEC 11172-3 [3] Layer II stereo decoder flow chart

## Annex D (informative): Multiplex re-configuration

The data associated with each Logical frame is spread over 16 CIFs as a result of the time interleaving process. This has to be taken into account, when the multiplex is re-configured.

A CU may only be allocated to one sub-channel at a time. Consequently, when the data rate of a sub-channel is changed at the input of the time interleaver, the allocation of CUs to the sub-channel is affected in the following way:

- when the data bit rate is increased, the additional CUs need to be allocated immediately, because some of the bits have zero delay through the time interleaver. During the following 15 CIF periods, only half, on average, of the newly allocated CUs are filled with valid data. The remaining portion contains zero value padding bits;

- when the data bit rate is decreased, the original number of CUs needs to remain allocated to the original sub-channel for another 15 CIF periods, because some of the bits are delayed by 15 CIF periods in the time interleaver. During this period, only half, on average, of the CUs, to be released following the recombination process, are filled with valid data. The remaining portion contains zero value padding bits.

The combination of time interleaving and de-interleaving results in a constant delay of 15 CIF periods. Therefore, every change of bit rate at the input of the time interleaver should be followed by an equivalent change at the output of the de-interleaver 15 CIF periods later.

A multiplex re-configuration is signalled as an event occurring at a given time instant. For sub-channels affected by the re-configuration, the changes of bit rate at the inputs of the time interleavers have to be co-ordinated. The instant of the re-configuration, signalled by the occurrence change, defines the CIF count from which the reallocation of CUs is effective.

As an illustration of the rules, defined in subclause 6.5, for co-ordinating the changes in the bit rate and the instant of re-configuration, the following examples are given. In all cases, it is assumed that the multiplex re-configuration occurs between CIFs of time index $r_0$-1 and $r_0$. Only a change of error protection profile in the convolutional encoders is considered in these examples:

a) **change in sub-channel position only**: the convolutional encoder is not affected;

b) **new sub-channel defined**: if a new sub-channel appears at $r=r_0$ (which did not exist at $r=r_0-1$) then the convolutional encoder uses the corresponding Protection profile for $r \geq r_0$;

c) **sub-channel removed**: if a sub-channel disappears at $r=r_0$ (which did exist at $r=r_0-1$) then the convolutional encoder ceases encoding at $r=r_0-15$;

d) **sub-channel capacity increased**: if a sub-channel increases its number of CUs between CIF of time index $r=r_0-1$ and $r=r_0$, then the convolutional encoder uses the new Protection profile for $r \geq r_0$;

e) **sub-channel capacity decreased**: if a sub-channel decreases its number of CUs between CIF of time index $r=r_0-1$ and $r=r_0$, then the convolutional encoder uses the new Protection profile for $r \geq r_0-15$;

f) **change in protection profile**: if the Protection profile changes between $r=r_0-1$ and $r=r_0$, but the number of CUs remains unchanged, then the convolutional encoder uses the new Protection profile for $r \geq r_0$.

As a further illustration, figure D.1 shows how two sub-channels exchange capacity during two multiplex reconfigurations (cases d and e above) At the first re-configuration, sub-channel 1 increases from 4 to 6 CUs and sub-channel 2 decreases from 6 to 4 CUs. The second re-configuration restores the original situation.

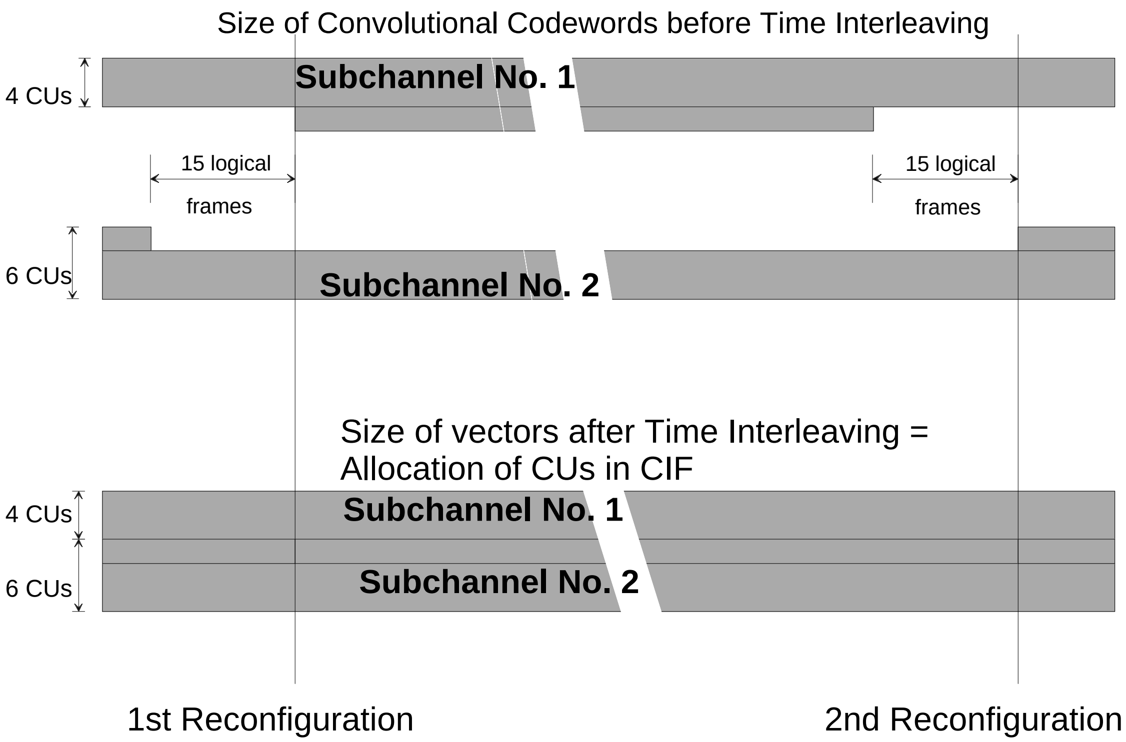


**Figure D.1: Example of multiplex re-configuration**

# Annex E (normative): Calculation of the CRC word

The implementation of cyclic redundancy check codes (CRC-codes) for audio and data transmission allows the detection of transmission errors at the receiver side. For this purpose CRC words shall be included in the transmitted data. These CRC words shall be defined by the result of the procedure described in this annex.

A CRC code is defined by a polynomial of degree $n$:

$$G(x) = x^n + g_{n-1}x^{n-1} + \ldots + g_2 x^2 + g_1 x + 1$$

with $n \geq 1$

and $g_i \in \{0,1\}$ , $i = 1.....n-1$

The CRC calculation may be performed by means of a shift register containing $n$ register stages, equivalent to the degree of the polynomial (see figure E.1). The stages are denoted by $b_0$ ... $b_{n-1}$, where $b_0$ corresponds to 1, $b_1$ to $x$, $b_2$ to $x^2$, ..., $b_{n-1}$ to $x^{n-1}$. The shift register is tapped by inserting XORs at the input of those stages, where the corresponding coefficients $g_i$ of the polynomial are "1".

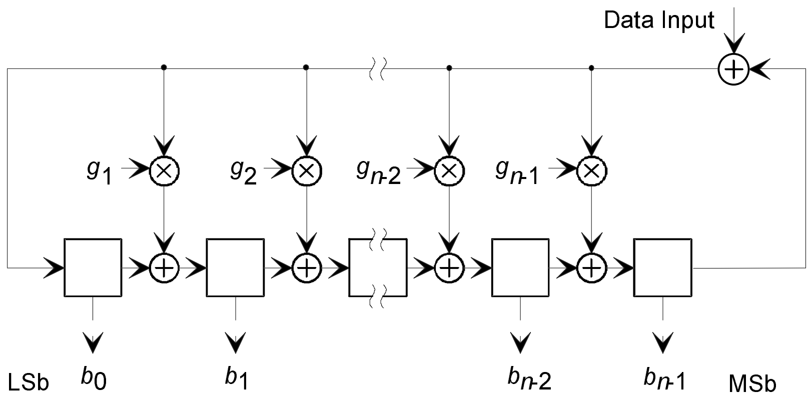


**Figure E.1: General CRC block diagram**

At the beginning of the CRC calculation, all register stage contents are initialised as specified in the respective subclauses, either to all ones or to all zeros. After applying the first bit of the data block (MSb first) to the input, the shift clock causes the register to shift its content by one stage towards the MSb stage ($b_{n-1}$), while loading the tapped stages with the result of the appropriate XOR operations. The procedure is then repeated for each data bit. Following the shift after applying the last bit (LSb) of the data block to the input, the shift register contains the CRC word which is then read out. Data and CRC word are transmitted with MSb first.

The CRC codes used in the DAB system shall be based on the following polynomials:

- $G(x) = x^{16} + x^{12} + x^5 + 1;$
- $G(x) = x^{16} + x^{15} + x^2 + 1;$
- $G(x) = x^8 + x^4 + x^3 + x^2 + 1.$

The assignment of the polynomials to the respective applications are given in subclauses 5.2.1, 5.3.2.3, 5.3.3.3, B2, B3 and annex D. These subclauses also indicate the size and the content of the associated data blocks, the initialisation of the shift register and a possible inversion (1s complement) of the CRC word prior to transmission.

## Annex F (informative): Bibliography

1) EBU Recommendation R79 (1994): "Recommended system for digital sound broadcasting to mobile, portable and fixed receivers in the appropriate frequency bands in the range 30 MHz to 3 GHz".

2) European Community DRIVE Programme, Proposed Pre-Standard (November 1990): "RDS ALERT C (Advice and Problem Location for European Road Traffic)".

3) National Radio Systems Committee (NRSC) (January 8, 1993): "Radio Broadcast Data System - RBDS", Published jointly by EIA and NAB.

4) EACEM Technical Report No 7 (July 1994): "Interactive Text Transmission System (ITTS)".

5) ITU-R Draft New Recommendation Doc. 10/76 rev., Working Party 10/B, March 1994, Geneva: "System for terrestrial digital sound broadcasting to vehicular, portable and fixed receivers in the frequency range 30 - 3 000 MHz".

6) ITU-R Draft New Recommendation, Doc10/75, Working Party 10 - 11S, March 1994, Geneva: "System for digital sound broadcasting to vehicular, portable and fixed receivers for BSS (sound) bands in the frequency range 1 000 - 3 000 MHz".

## History

| Document history | |
|---|---|
| February 1995 | First Edition |
| January 1996 | Converted into Adobe Acrobat Portable Document Format (PDF) |
| | |
| | |
| | |

ISBN 2-7437-0000-9
Dépôt légal : Février 1995