ANSI/TIA-102.BAAA-A 2003
Approved: September 17, 2003

TIA STANDARD

Project 25

FDMA – Common Air Interface

New Technology Standards Project – Digital Radio Technical Standards

TIA-102.BAAA-A

Revision of TIA-102.BAAA)

TELECOMMUNICATIONS INDUSTRY ASSOCIATION

Representing the telecommunications industry in association with the Electronic Industries Alliance

NOTICE

NOTICE

TIA Engineering Standards and Publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for their particular need. The existence of such Standards and Publications shall not in any respect preclude any member or non-member of TIA from manufacturing or selling products not conforming to such Standards and Publications. Neither shall the existence of such Standards and Publications preclude their voluntary use by Non-TIA members, either domestically or internationally.

Standards and Publications are adopted by TIA in accordance with the American National Standards Institute (ANSI) patent policy. By such action, TIA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard or Publication.

This Standard does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this Standard to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

(From Standards Proposal No. 3-4080-RV1, formulated under the cognizance of the TIA TR-8.15 Subcommittee on Common Air Interface.)

Published by
©TELECOMMUNICATIONS INDUSTRY ASSOCIATION 2002
Standards and Technology Department
2500 Wilson Boulevard
Arlington, VA 22201 U.S.A.

Foreword

This foreword is not part of this standard.

This document has been submitted to APCO/NASTD/FED by the Telecommunications Industry Association (TIA), as provided for in a Memorandum of Understanding (MOU) dated December, 1993. That MOU provides that APCO/NASTD/FED will devise a Common System Standard for digital public safety communications (the Standard), and that TIA shall provide technical assistance in the development of documentation for the Standard.

This document has been developed with inputs from the TIA Project 25 Interface Committee (APIC), the APIC Format Task Group, and TIA Industry members. This document has been developed by the TR-8.15 common air interface subcommittee of TIA.

This document is being published because it is felt that there is an urgent need for technical information on the emerging digital techniques for Land Mobile Radio Service.

This document describes the access method, bandwidth, modulation, data rate, and message format for radios meeting the Project 25 requirements. These definitions are necessary to ensure a common air interface. This document includes a normative Annex A, to list the bits of a voice transmission in consecutive order.

This standard presents a design for a Common Air Interface which was recommended by TIA to APCO/NASTD/FED as being suitable for use as part of their standard for a digital public safety radio system (Project 25). The TIA has published a *Project 25 System and Standards Definition* document to list the remaining parts of the Project 25 suite of documents, which includes this common air interface.

Table of Contents

List of Tables

List of Figures

Patent Identification

The reader's attention is called to the possibility that compliance with this document may require the use of one or more inventions covered by patent rights.

By publication of this document no position is taken with respect to the validity of those claims or any patent rights in connection therewith. The patent holders so far identified have, we believe, filed with APCO/NASTD/FED statements of willingness to grant licenses under those rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such licenses. Details may be obtained from APCO/NASTD/FED.

The following patent holders and patents have been identified in accordance with the TIA intellectual property rights policy:

TIA shall not be responsible for identifying patents for which licenses may be required by this document or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Motorola Inc. -- Patent Nos. US 5,377,229; US 5,271,017; US 5,185,796; US 5,148,482; US 4,636,791; US 4,590,473

Ericsson Inc. -- Patent Nos. US 4,757,536

Abbreviations

For the purposes of this standard, the following abbreviations apply.

AFC	Automatic Frequency Control
ARQ	Automatic Retry Request to retry corrupted packets
BR	Base Radio, a reference designating a base station
C4FM	Compatible 4-level FM version of QPSK-c modulation
CQPSK	Compatible QPSK version of QPSK-c modulation
CRC	Cyclic Redundancy Checksum for data error detection
Dibit	2 bits grouped together to represent a 4-level symbol
ES	Encryption Synchronization information embedded in voice
FS	Frame Synchronization to mark the first information bit
GF	Galois Field to calculate parity checks for a RS code
Golay	Name of a standard error correction code
Hex Bit	6 bits grouped together to represent a Reed-Solomon code symbol
IMBE	Improved Multi-Band Excitation coder for voice
KID	Key Identifier to indicate the encryption key for the message
LC	Link Control information embedded in voice
LLC	Logical Link Control sublayer of the OSI Data Link Layer
LSD	Low Speed Data embedded in voice
MAC	Media Access Control sublayer of the OSI Data Link Layer
MI	Message Indicator to initialize encryption
MR	Mobile Radio, a reference designating a mobile or portable radio
NID	Network Identifier code word following the frame sync
Octal	Base 8 notation for numbers, also called radix 8
Octet	8 bits grouped together, also called a byte
OSI	Open Systems Interconnection reference model
QPSK	Quadrature Phase Shift Keying modulation for data
QPSK-c	Compatible Quadrature Phase Shift Keying family of modulations
RS	Reed-Solomon error correction code
SAP	Service Access Point, where a network provides a service
Trellis Code	Type of error correcting code for modulation
Tribit	3 bits grouped together into a symbol for a trellis code
Type 3	Type of encryption for non-classified communications
Type 4	Type of encryption for export from the US
Um	Common Air Interface reference point

1 Introduction

The objective of this document is to define the Common Air Interface for all reference configurations described in the general system model. In a first reference configuration, this is between mobile and portable subscriber units and the base stations of any RF subsystems, as illustrated here.

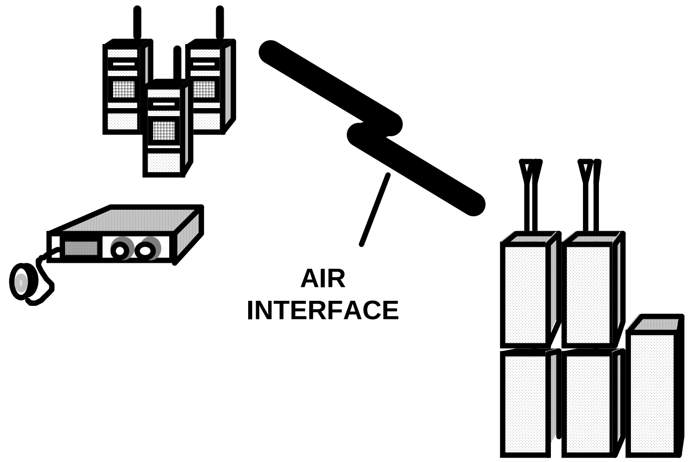

Figure 1-1 Example Repeater Illustration

According to the Project 25 General System Model, this is the Um interface between a mobile subscriber unit functional group, and one or more base radio functional groups at a site, at multiple sites within an RF subsystem, and within any RF subsystems where the subscriber unit might roam.

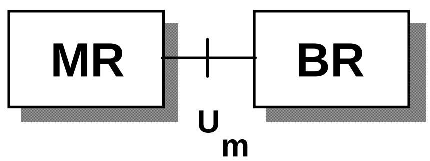

Figure 1-2 Repeater Reference [Model] Configuration

In a second reference configuration, this is directly between mobile and portable subscriber units in a talk-around configuration, as illustrated here.

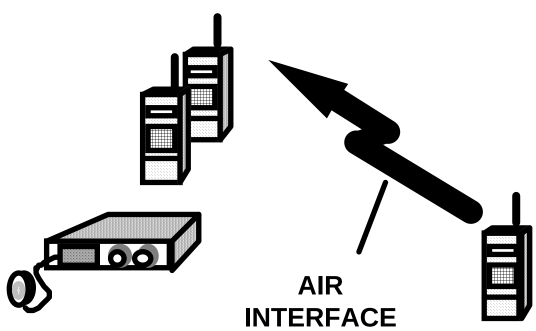

Figure 1-3 Example Talk-Around Illustration

In this reference configuration, this is the Um interface between a pair of mobile subscriber unit functional groups.

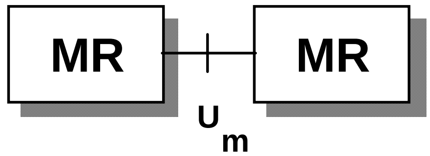

Figure 1-4 Talk-Around Reference Configuration

2 Scope

The Project 25 standard will cover all of the parts of a system for public safety land mobile radio communications. These systems have subscriber units (which include portable radios for hand held operation and mobile radios for vehicular operation), base stations (for fixed installations), and other fixed equipment (for wide-area operation and console operator positions), as well as computer equipment (for data communications). There are interfaces between each of these equipment items. The Common Air Interface allows these radios to send and receive digital information over a radio channel. This is shown in Figure 2-1.

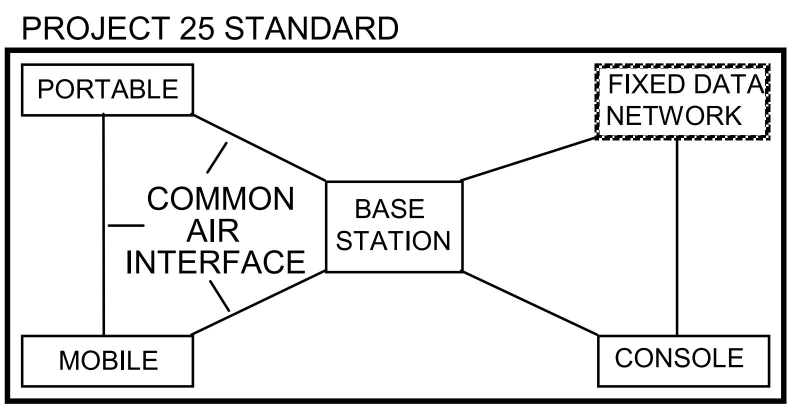

Figure 2-1 Common Air Interface in a Radio System

The Common Air Interface is part of an intended suite of standards necessary for interoperability. A general overview of the Project 25 standard is in the *Project 25 System and Standard Definition* given as reference [1]. The reader should have familiarity with this document. Included in this reference are the Project 25 Overview, General System Model, Standards Organization, Glossary, and the Statement of Requirements. The Standards Organization is a list of other applicable Project 25 standards. Sections of reference [1] are also listed as references. They are reference [1.1], *STATEMENT OF REQUIREMENTS*, and reference [1.2], *General System Model.*

This document directly addresses many of the requirements in the *STATEMENT OF REQUIREMENTS* and it uses the elements of the model given in the *General System Model* document. Specifically, this document addresses the Physical Layer and Data Link Layer of the Common Air Interface.

The objective of the Common Air Interface is to ensure that subscriber unit equipment that conforms to this document will be interoperable at the Physical

Layer and Data Link Layer with subscriber unit equipment from different manufacturers, and compatible with radio systems for different agencies. The full capability of Project 25 will be achieved when these subsequent specifications and Network Layer definitions are developed. This will allow effective and reliable intra-agency and inter-agency communications in an all-digital mode for voice and data. This is in conformance to the *STATEMENT OF REQUIREMENTS*.

This document will not provide the specification or operational detail for system implementations which include but are not limited to trunking, roaming, network management, vocoder, security, data, subsystems interfaces and data between private and public switched telephone networks. It will provide only the appropriate access requirements to ensure common air interface compatibility. This document will be updated as necessary to ensure Physical and Data Link Layer compatibility for the purpose of providing a common air interface as additional system implementation requirements are available.

The Common Air Interface defines a reference point at which communications between radios can take place. This reference point is designated Um. Communications through Um are done at a gross bit rate of 9.6 kbps. Several processes take place to convert information for transmission through the Um reference point. The Common Air Interface uses a standard voice coder (reference [2]) to convert speech to a digital format for communication through Um. This voice information is then protected with additional error correction coding to provide protection over the channel.

2.1 Revision History

September 1, 1992, P25.920901.3.0, contained the initial outline.

Version 0.2, September 23, 1992, P25.920901.3.1, was the first generally circulated version.

Version 0.2.1, November 7, 1992, P25.920901.3.2, included revisions to the encryption synchronization for Type 1 operation, clarifications of the description, and the inclusion of missing information.

Version 0.2.2, November 16, 1992, P25.920901.3.3, included naming convention for IMBE bits.

Version 0.2.3, December 9, 1992, P25.920901.3.4, final names selected for IMBE bits.

Version 0.2.4, April 7, 1993, P25.930901.3.5, altered the Hamming parity matrix and IMBE interleave table. Also added annex for transmit bit order.

Version 0.2.5, May 19, 1993, P25.930901.3.6, altered the section numbers and added a paragraph in the introduction.

Version 0.2.5 Rev B, June 2, 1993, altered to include Selective ARQ.

Version 0.2.5 Rev C, July 26, 1993, made numerous typographical changes.

Version 0.2.5 Rev D, August 16, 1993, made numerous typographical changes.

Version 0.2.5 Rev E, February 10, 1994, changed to incorporate comments on a written ballot. The talk-group ID expanded to 16 bits.

Version 0.2.5 Rev F, February 6, 1995, added some patents to the patent list, and symmetrically addressed ACKs.
Revision G, March 23, 1995, added another patent and referred deviation measurement to TSB-102.CAAA.
Revision H, February 9, 1998, added FDMA to the title and a paragraph in section 9.1 to mention DQPSK modulation.
Revision J, June 12, 1998, removed restriction to 12.5 kHz channels.
Revision K, September 6, 2002, added AFC for 700 MHz.
Revision L, February 18, 2003, added more AFC information.
Revision M, August 5, 2003, resolved ballot comments.

2.2 References

The following documents contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

[1] *Project 25 System and Standards Definition*, TIA, TSB102-A, November 1995. This contains within it references 1.1 and 1.2.
[1.1] *STATEMENT OF REQUIREMENTS*, *Project 25 System and Standards Definition*, Appendix C, 8/12/92.
[1.2] *APCO Project 25 General System Model*, *Project 25 System and Standards Definition*, Section 3.
[2] *Project 25 Vocoder Description*, TIA, TIA/EIA-102.BABA, May 1998.
[3] *Common Air Interface Operational Description for Conventional Channels*, TIA, TSB102.BAAD, October 1994.
[4] *Digital C4FM / CQPSK Transceiver Methods of Measurement*, TIA/EIA-102.CAAA, June 1999.
[5] *Digital C4FM/CQPSK Transceiver Performance Recommendations*, TIA/EIA-102.CAAB, November 2000.
[6] *Link Control Word Formats and Messages,* TIA, TSB-102.AABF, May 1996.
[7] *Common Air Interface Reserved Values,* TIA, TSB-102.BAAC-A-1, June 2001.

3 Description

In reference to the *General System Model* document, the Common Air Interface reference point is Um. This interface is between mobile, portable, and base station radios. The interface specifications are organized with the Open System Interface (OSI) reference model into seven layers. This document describes the formats for the bottom two layers of the Um interface. These are the Physical Layer and the Data Link Layer in the OSI reference model.

The Physical Layer is primarily concerned with modulation. This is described in Section 9 of this document. The Data Link Layer is further separated into a Media Access Control (MAC) sublayer and a Logical Link Control (LLC) sublayer.

The MAC sublayer is described in Section 8. The LLC sublayer is described in Sections 4, 5, 6, and 7. The arrangement of layers, sublayers, and interfaces is shown in Figure 3-1.

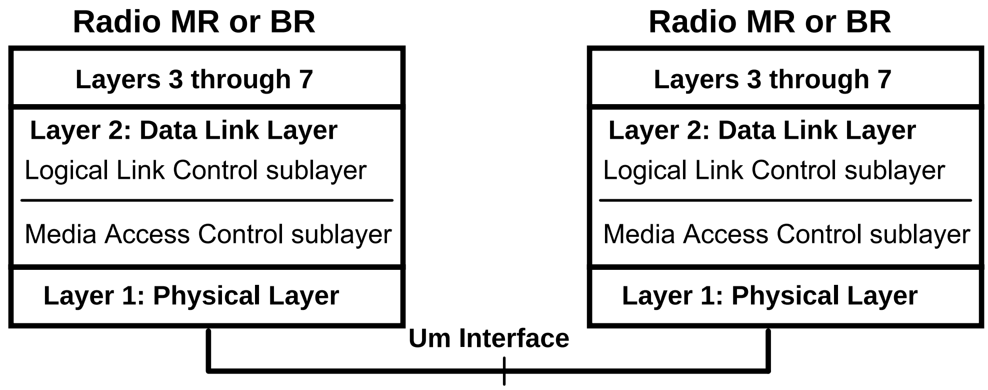

Figure 3-1 Reference Model Diagram

This document describes the following items that are necessary for basic digital communication.

Radio Channel Bandwidth
Radio Channel Bit Rate
Modulation
Channel Access
Frame Formats including codes for Error Detection and Correction
Voice Coder

Some additional parts of a Common Air Interface might include Trunking and Encryption. These additional parts are described in other documents which are listed in the reference section.

The Common Air Interface consists of several parts that are diagrammed in Figure 3-2. Voice is processed in a sequence that starts with a voice coder, proceeds to encryption, and then adds error correction. Other voice functions are also performed that are not included in the simple diagram of Figure 3-2, such as addressing and embedded signaling.

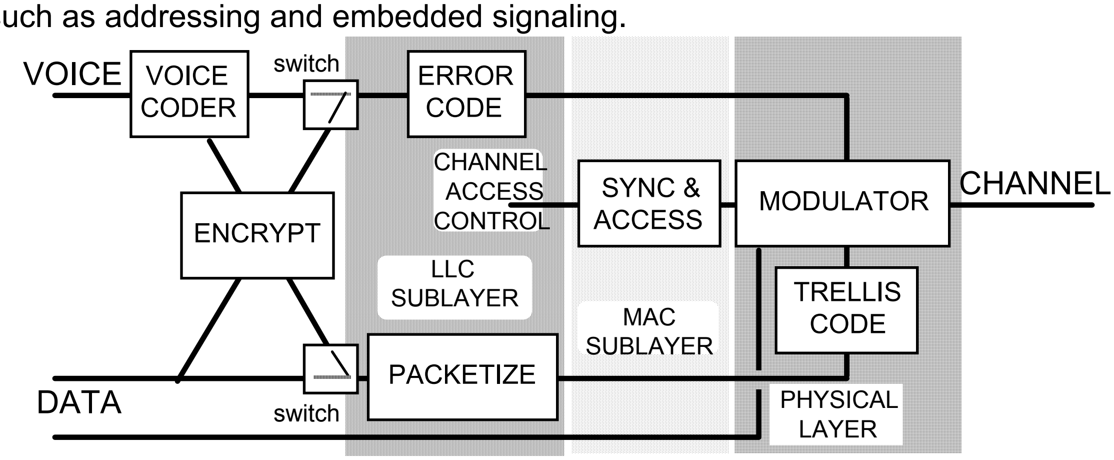

Figure 3-2 Simple Transmitter Diagram

Data is processed in either of two distinct ways. The simplest way is to directly modulate the transmitter without any other processing. The other way proceeds by first decomposing large data messages into small packets of information. These packets then have error correction added to them to tolerate noise and degradation on the channel. Other parts of the data process are not diagrammed, including any source coding of the data or encryption.

Voice or data messages are stored in the transmitter until the transmitter is ready to send the message. At that time, the message is gated to the radio modulator for transmission over the Common Air Interface. Each of these parts of the Common Air Interface are described in this document.

The order in which subjects in this document are presented corresponds approximately to the order in which information is processed by a transmitter. As a result, the subject of error correction is described before the subject of modulation. In the higher layers of the reference model of Figure 3-1, the information is portrayed as an abstract block of data that generally does not represent the way in which it is transmitted through the Common Air Interface. In the lower layers of Figure 3-1, the information is portrayed in a more physical manner that more accurately represents the order in which it is transmitted. In general, the logical representation is used in the LLC sublayer and above while the physical representation is used in the MAC sublayer and below.

4 Voice Coder

The voice coder for the Common Air Interface converts voice information into digital data at the continuous average rate of 4.4 kbps. This corresponds to a voice frame of 88 bits for every 20 ms of speech. The voice coder for this document is Improved Multi-Band Excitation or IMBE. The details of IMBE for this document are defined in reference [2], *Vocoder Description*.

An IMBE frame encodes 20 ms of speech into 88 bits of information. There are several fields of information which are briefly described here.

Quantized Pitch -- b_0 -- 8 bits -- This encodes the pitch period. The quantized pitch value is determined from the unquantized value in the transmitter by the following relation.

$$b\_0 = \mathrm{INT}[\ 16\ \mathrm{kHz} \cdot \mathrm{period}\ - 39\ ]$$ where the period is in seconds.

The receiver computes the period from:

$$\mathrm{period} = (b\_0 + 39.5) \cdot 62.5\ \mu s.$$

The pitch period allows the computation of the number of harmonics, L, and the number of sub-bands, K, as follows.

$$x = \mathrm{INT}[\ 4\ \mathrm{kHz} \cdot \mathrm{period} + 0.25\ ]$$
$$L = \mathrm{INT}[\ 0.9254\ x\ ]$$
$$K = \mathrm{INT}[\ (L+2)/3\ ] \quad \text{if } L \le 36$$
$$K = 12 \quad \text{if } L > 36$$

The quantized pitch is restricted to the range $0 \le b\_0 \le 207$ for normal speech. The values 216, 217, 218, and 219 are used to encode silence. Other values, 208 through 215 and 220 through 255, are reserved for future use.

Voiced / UnVoiced -- b_1 -- K bits -- The parameter K will vary from 3 to 12, with the value of 3 being used for higher pitched sounds (i.e., shorter pitch periods). Each bit in this vector indicates whether the corresponding sub-band is voiced or unvoiced.

Quantized Gain Level -- b_2 -- 6 bits -- This encodes the average amplitude of the speech frame. What is actually encoded is an amplitude residue, so an estimate of the actual speech amplitude requires several residues to be combined together from consecutive voice frames.

Quantized Gain Vector and DCT Coefficients -- 73–K bits -- this is an array of L–1 parameters which encode the amplitudes of the DCT coefficients of the residues. The IMBE encoder produces the encoded parameters b_3, b_4, ... b_L+1. The parameters are collected together into vectors for the error correcting codes. The precise way this is done is described in the *Vocoder Description*.

Sync -- 1 bit -- This bit alternates between zero and one in the transmitter and may be ignored in the receiver.

These fields are collected together into vectors with the symbols, u_0, u_1, ... u_7. The details are given in the *Vocoder Description* . The sync bit is the least significant bit of u_7. The construction of the most significant bits of u_0 and the least significant bits of u_7 are depicted below. The bits signified by periods (.) are dependent on the parameters K and L. In general, the most significant bits are depicted on the left and the least significant are on the right. The index number for a bit starts with number 0 as the least significant bit and counts up.

	b_0						b_2					
u_0 composition 12 bits	7	6	5	4	3	2	5	4	3	.	.	.

				b2	b_0		sync
u_7 composition 7 bits	.	.	.	0	1	0	

5 Voice Formats

5.1 General Description

Voice information is transmitted with additional encryption information, unit identification, and low speed data to fully utilize the 9.6 kbps of channel capacity in the Common Air Interface. These other information fields are described in Section 5.

Voice frames are protected with 56 parity check bits to make the overall size of the encoded voice frame 144 bits. The voice frames are then collected into larger data units of 9 frames each. Each data unit is then transmitted over-the-air in 180 ms. These voice data units are called Logical Link Data Units. Two consecutive data units are grouped together into a superframe. Each superframe contains 18 voice frames and takes 360 ms to transmit over-the-air. The superframes of information are repeated for the duration of the voice message after the header has been sent. The information content and code words of a superframe are diagrammed in Figure 5-1.

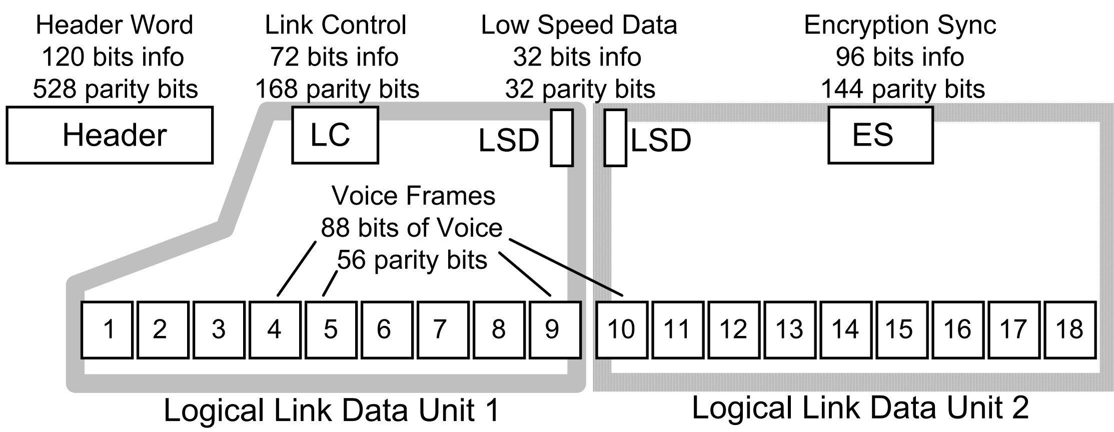

Figure 5-1 Voice Message Structure

A short header is sent at the beginning of the voice transmission to properly initialize any encryption function for the message. A Header consists of a single large code word of 648 bits with 120 bits of information. Additional encryption information is periodically interleaved throughout the voice message as described in Sections 5.4 and 8.2.2. A voice message also includes Link Control information as well as Low Speed Data, described in Sections 5.5 and 5.6 respectively.

Header Data Unit	Logical Link Data Unit 1	Logical Link Data Unit 2	Logical Link Data Unit 1	Logical Link Data Unit 2	Terminator Data Unit

SUPERFRAME
360 msec

Figure 5-2 Data Units for Voice Messages

The sequence of information during a voice transmission is shown in Figure 5-2. The voice message begins with a Header, and then continues with Logical Link Data Units or LDUs. The LDUs alternate until the end of the voice message. The end of the message is marked with a terminator. The terminator can follow any of the other voice data units. The detailed structure of the data units is given in Section 8.

5.1.1 Notation

The error correction for voice makes extensive use of Reed-Solomon codes over an extension Galois Field. The common notation for this type of code is:

RS = Reed-Solomon, as in "an RS code"

$GF(2^6)$ = extension Galois Field with 2^6=64 elements,
as in "$GF(2^6)$ arithmetic"

hex bit = 6-bit symbol for one of the elements of the $GF(2^6)$ field

Error correcting codes are usually denoted by their block length parameters, n, k, and d. The length of the code word block is n. The number of information symbols in the code word is k. The minimum Hamming distance between code words is d. The code is then denoted by the triplet (n,k,d) as in "(24,12,8) Golay code." Almost all the codes in this description use binary codes, where the parameters n, k, and d are in bits. The only exceptions are the Reed-Solomon codes where the parameters are for symbols of 6 bits each, i.e., hex bits. The reader can convert the RS code parameters to dimensions of bits by multiplying the n and k parameters by 6.

Systematic codes are used for all voice information. Each code word contains n symbols. The first k symbols in the left hand part of the code word contain the information. The last n–k symbols in the right hand part contain the parity checks for the code word.

5.1.2 Reserved Bits and Null Bits

In many places in the following formats, there are extra bits which have no assigned functions. These are labeled as reserved bits or sometimes as null bits. Reserved bits are reserved for future standard definitions. They are not intended to allow non-standard implementations, but to allow future revisions to the document. Transmitters which conform to the standard definitions should encode the reserved bits with nulls (zeros). Receivers should ignore these fields.

For some fields, not all of the available values are defined. For example, the Data Unit ID field in Section 8.5.1 has sixteen possible values, but not all of them are defined with a function. The remaining values are labeled as reserved to indicate that future versions of the document may utilize these values for some possible function.

In some places, additional bits are added as pads. This is especially prevalent at the end of data units described in Section 8.2. In these cases, the bits are referred to as null bits and are simply place holders to pad the unit of data to a convenient time boundary. These bits are not generally part of the message and the radio system does not need to transport them through the fixed network. However, these bits are required as part of the Common Air Interface and may not be used for any other function.

5.2 Header Word

The header word includes the following information fields.

Message Indicator (MI) -- 72 bits -- This is the initialization vector for a Type 1, Type 2, Type 3 or Type 4 encryption algorithm.

Manufacturer's ID (MFID) -- 8 bits -- This is asserted when non-standard features are included in the voice message by the manufacturer. This field has a standard value ($00) when all of the other information fields conform to the definitions of the Common Air Interface. It is a minimum requirement for a standard radio to be able to transmit or receive messages using the standard field definitions of the Common Air Interface, with the standard value for the MFID field. The minimum requirement for standard receivers is to ignore messages which do not contain the standard values for the MFID field.

Algorithm ID (ALGID) -- 8 bits -- This identifies the encryption algorithm in systems with multiple algorithms.

Key ID (KID) -- 16 bits -- This identifies the encryption key for systems with multiple encryption keys.

Talk-group ID (TGID) -- 16 bits -- This identifies the talk-group for the message.

These information fields are concatenated together into 120 bits. They are then separated into 20 symbols of 6 bits each. Each symbol is called a hex bit. These are encoded with a (36,20,17) Reed-Solomon code to yield 36 hex bits. These 36 hex bits are then in turn encoded with an (18,6,8) shortened Golay code. This yields 648 bits all together.

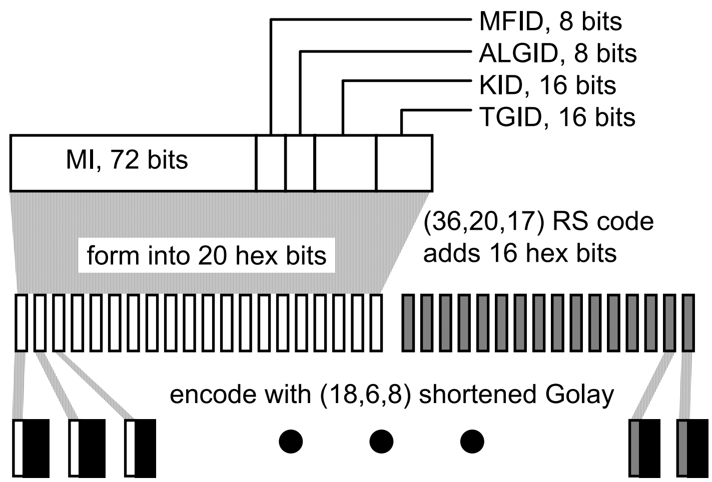

Figure 5-3 Diagram of Header Code Word Construction

5.3 Voice Frames

Voice frames consist of 8 information vectors, denoted u_0, u_1, ... u_7, described in Section 4. Voice frames are encoded into a 144 bit word. The IMBE bits have been subjectively rated according to their effect on audio quality and are protected using Golay and Hamming codes described in Sections 5.7 and 5.8. The 48 most important bits are protected against errors using four (23,12,7) Golay code words. The next 33 most significant bits are protected against errors using three (15,11,3) Hamming code words. The remaining 7 least significant bits are not protected against errors. The partition of the IMBE information bits into code words is given in Figure 5-4.

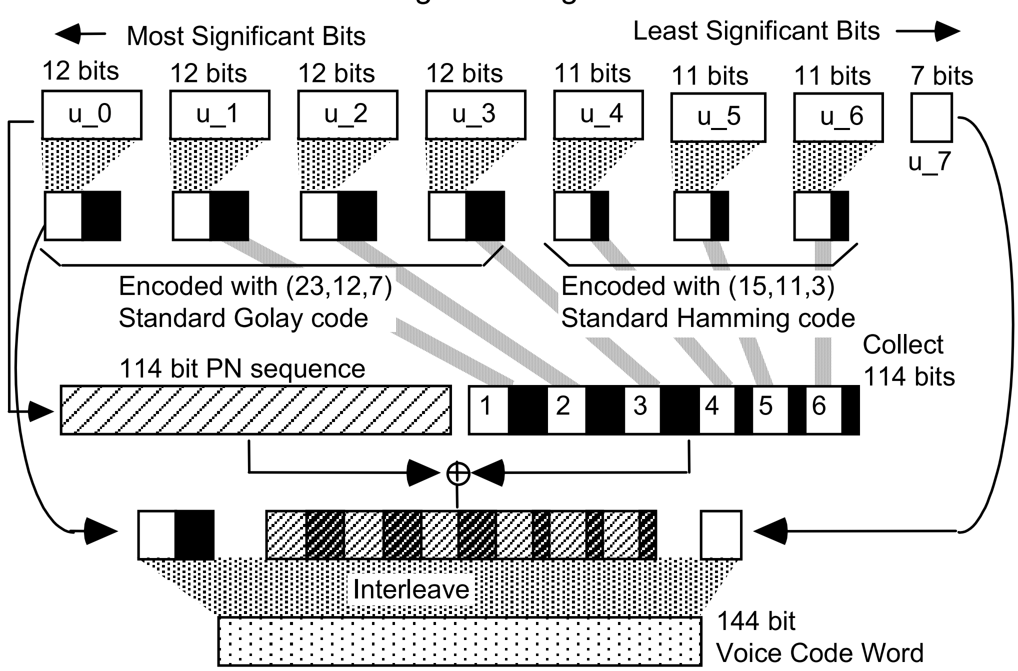

Figure 5-4 Diagram of Voice Code Word Construction

It must be remembered that the information words u_0, u_1, ... u_7 may be encrypted. For messages that are encrypted, the labels u_0, u_1, ... u_7 designate the encrypted cipher text and not the plain text which is processed by the voice coder.

After the Golay and Hamming words are constructed, a 114 bit pseudo random sequence (called a PN sequence) is generated from the 12 bits of u_0 in the following way. First, a 16 bit initial value called p_0 is generated from u_0 by shifting it to the left 4 bits. Then, a linear congruential pseudo random sequence is generated. Only the first 114 values are used, and only bit 15 (the most significant) is used for the 114 bit PN sequence. This is all described by the formulas below.

$$p_0 = 16\ u\_0$$
$$p_n = [\ 173\ p_{n-1} + 13849\ ] \bmod 65536 \qquad \text{for } 1 \le n \le 114$$

Let $p_n(15)$ represent bit 15 of p_n. Then:

m_1 = [$p_1(15)$, $p_2(15)$, $p_3(15)$, ... $p_{23}(15)$]	first 23 bits
m_2 = [$p_{24}(15)$, $p_{25}(15)$, $p_{26}(15)$, ... $p_{46}(15)$]	next 23 bits
m_3 = [$p_{47}(15)$, $p_{48}(15)$, $p_{49}(15)$, ... $p_{69}(15)$]	next 23 bits
m_4 = [$p_{70}(15)$, $p_{71}(15)$, $p_{72}(15)$, ... $p_{84}(15)$]	next 15 bits
m_5 = [$p_{85}(15)$, $p_{86}(15)$, $p_{87}(15)$, ... $p_{99}(15)$]	next 15 bits
m_6 = [$p_{100}(15)$, $p_{101}(15)$, $p_{102}(15)$, ... $p_{114}(15)$]	last 15 bits

These PN vectors are then exclusive-ored bit-wise with the Golay code words 1, 2, and 3, and Hamming code words 4, 5, and 6. The right most bit in the vector m_X is denoted m_X(0). The left most bit in vectors m_1, m_2, and m_3 is denoted m_X(22), and in general the bits are denoted with numbers in sequence from left to right as 22, 21, 20, ... 2, 1, 0. Similarly, the left most bit in vectors m_4, m_5, and m_6 is denoted m_X(14), and in general the bits are denoted with the numbers 14, 13, 12, ... 2, 1, 0 from left to right.

The result of the exclusive-or sum of the Golay and Hamming code words with vectors m_1, m_2, ... m_6 are called coset code words. The coset code words are denoted c_1, c_2, ... c_6 respectively, with c_1, c_2, and c_3 being Golay coset code words and c_4, c_5, and c_6 being Hamming coset code words. For convenience, the code word c_0 is defined to be Golay code word 0 and the code word c_7 is defined to be u_7. This is summarized in the following equations.

c_0 = Golay code word 0 = [u_0, Golay_0_parity]
c_1 = Golay coset code word 1 = [u_1, Golay_1_parity] ⊕ m_1
c_2 = Golay coset code word 2 = [u_2, Golay_2_parity] ⊕ m_2
c_3 = Golay coset code word 3 = [u_3, Golay_3_parity] ⊕ m_3
c_4 = Hamming coset code word 4 = [u_4, Hamming_4_parity] ⊕ m_4
c_5 = Hamming coset code word 5 = [u_5, Hamming_5_parity] ⊕ m_5
c_6 = Hamming coset code word 6 = [u_6, Hamming_6_parity] ⊕ m_6
c_7 = u_7

5.3.1 Interleaving Schedule

In order to resist fades, the code words are interleaved throughout the voice frame. The interleaving table is given in Table 5-1. The length of the voice code word is 144 bits, or 72 dibit symbols. Each dibit has a most significant bit (Bit 1) and a least significant bit (Bit 0), as shown. The code words are labeled c_X(i) where (i) refers to the index of the bit. Bit 0 is the least significant or right most bit. The symbols are transmitted starting with symbol 0 and ending with symbol 71. There are often other symbols interleaved within the voice frame as described in Section 8.2.

Table 5-1 Interleaving Schedule for Voice Word

Symbol	Bit 1	Bit 0	Symbol	Bit 1	Bit 0
0	c_0(22)	c_1(21)	36	c_0(10)	c_1(9)
1	c_2(20)	c_3(19)	37	c_2(8)	c_3(7)
2	c_4(10)	c_5(1)	38	c_5(13)	c_6(4)
3	c_1(20)	c_0(21)	39	c_1(8)	c_0(9)
4	c_3(18)	c_2(19)	40	c_3(6)	c_2(7)
5	c_5(0)	c_4(9)	41	c_6(3)	c_5(12)
6	c_0(20)	c_1(19)	42	c_0(8)	c_1(7)
7	c_2(18)	c_3(17)	43	c_2(6)	c_3(5)
8	c_4(8)	c_6(14)	44	c_5(11)	c_6(2)
9	c_1(18)	c_0(19)	45	c_1(6)	c_0(7)
10	c_3(16)	c_2(17)	46	c_3(4)	c_2(5)
11	c_6(13)	c_4(7)	47	c_6(1)	c_5(10)
12	c_0(18)	c_1(17)	48	c_0(6)	c_1(5)
13	c_2(16)	c_3(15)	49	c_2(4)	c_3(3)
14	c_4(6)	c_6(12)	50	c_5(9)	c_6(0)
15	c_1(16)	c_0(17)	51	c_1(4)	c_0(5)
16	c_3(14)	c_2(15)	52	c_3(2)	c_2(3)
17	c_6(11)	c_4(5)	53	c_7(6)	c_5(8)
18	c_0(16)	c_1(15)	54	c_0(4)	c_1(3)
19	c_2(14)	c_3(13)	55	c_2(2)	c_3(1)
20	c_4(4)	c_6(10)	56	c_5(7)	c_7(5)
21	c_1(14)	c_0(15)	57	c_1(2)	c_0(3)
22	c_3(12)	c_2(13)	58	c_3(0)	c_2(1)
23	c_6(9)	c_4(3)	59	c_7(4)	c_5(6)
24	c_0(14)	c_1(13)	60	c_0(2)	c_1(1)
25	c_2(12)	c_3(11)	61	c_2(0)	c_4(14)
26	c_4(2)	c_6(8)	62	c_5(5)	c_7(3)
27	c_1(12)	c_0(13)	63	c_1(0)	c_0(1)
28	c_3(10)	c_2(11)	64	c_4(13)	c_3(22)
29	c_6(7)	c_4(1)	65	c_7(2)	c_5(4)
30	c_0(12)	c_1(11)	66	c_0(0)	c_2(22)
31	c_2(10)	c_3(9)	67	c_3(21)	c_4(12)
32	c_4(0)	c_6(6)	68	c_5(3)	c_7(1)
33	c_1(10)	c_0(11)	69	c_2(21)	c_1(22)
34	c_3(8)	c_2(9)	70	c_4(11)	c_3(20)
35	c_6(5)	c_5(14)	71	c_7(0)	c_5(2)

5.4 Encryption Sync Word

The encryption information consists of 96 bits to convey the identification and synchronization necessary to support a multi-key encryption system. The information includes the **Message Indicator** (MI), **Algorithm ID** (ALGID) for the encryption algorithm, and the **Key ID** (KID) for the encryption key. This information field is expanded to 240 bits with an error correcting code, and interleaved with the voice information. The encoding is done by serializing the 96 bits of information into 16 hex bits. These are then encoded with a (24,16,9) RS code to yield 24 hex bits. These hex bits are then encoded with a (10,6,3) shortened Hamming code. This yields 24 shortened Hamming words for a total of 240 bits.

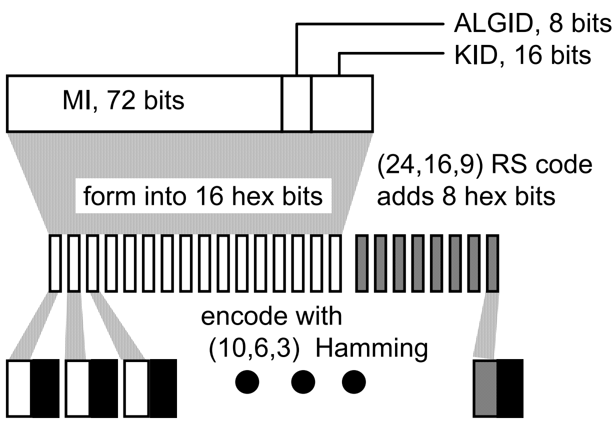

Figure 5-5 Diagram of Encryption Sync Code Word Construction

5.5 Link Control Word

The unit identification is encoded into a 72 bit field that is generically identified as Link Control or LC. The Link Control field includes a **Talk-group ID** (TGID), a **Source ID**, a **Destination ID**, an **Emergency** indicator, a **Manufacturer's ID** (MFID) and any other information necessary for identification of the call. There is too much information to be held in a code word with a fixed field format, so a variable format is used with a **Link Control Format** (LCF) to specify the word's information content. Two format examples are diagrammed in Figure 5-6. Other formats, as well as operational details, are provided in other documents. The reader is referred to reference [6].

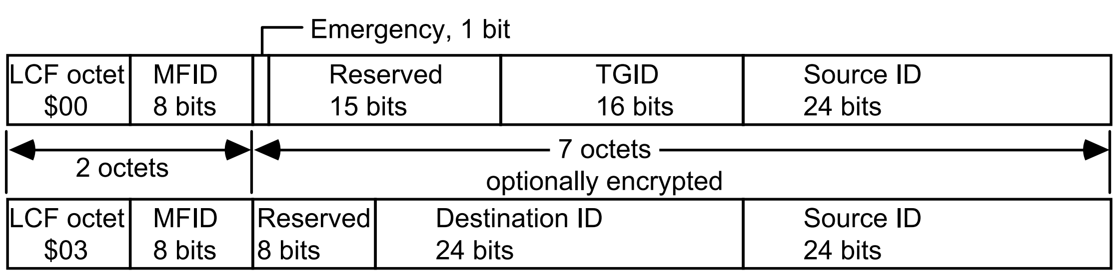

Figure 5-6 Link Control Formats

The LC code word is constructed by serializing the LC information into 12 hex bits. These are then encoded with a (24,12,13) RS code to yield 24 hex bits. These hex bits are then encoded with the (10,6,3) shortened Hamming code. This yields 24 shortened Hamming code words for a total of 240 bits.

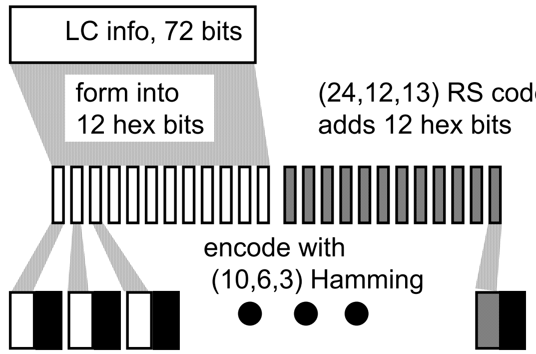

Figure 5-7 Diagram of Link Control Code Word Construction

5.5.1 Terminator Link Control

The Common Air Interface also provides a means for sending a Link Control code word without sending any voice information. This is done in a Terminator Data Unit to be defined in Section 8.2.3. In this case, the Link Control format remains the same but the code word is constructed slightly differently. The 24 hex bits of the Reed-Solomon code are paired into 12 pairs. Each pair is then encoded with a (24,12,8) extended Golay code. This yields a total of 288 bits. This is diagrammed in Figure 5-8.

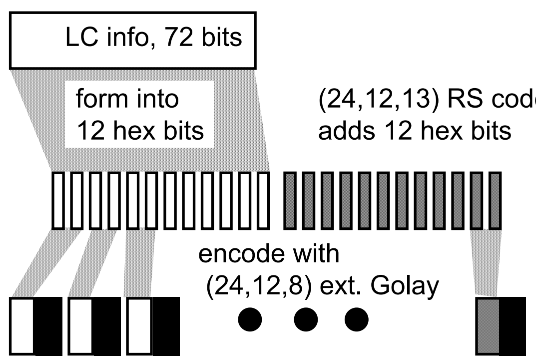

Figure 5-8 Diagram of Terminator Link Control Code Word

5.6 Low Speed Data Word

Low Speed Data is a serial stream of octets, or bytes, of information. This information is provided for user applications that are not defined in the Common Air Interface. There are 4 octets, or 32 bits, provided in every 360 ms superframe, for a total capacity of 88.89 bps. Each octet is encoded with a (16,8,5) shortened cyclic code. This yields 4 code words, or a total of 64 bits, for each superframe.

The generator polynomial for the code is given by:

$$g(x) = x^8 + x^5 + x^4 + x^3 + 1.$$

The generator matrix for this code is given in systematic form in Table 5-2.

Table 5-2 Generator Matrix of (16,8,5) code

Row	8 x 8 Identity	8 x 8 Parity
1	1 0 0 0 0 0 0 0	0 1 0 0 1 1 1 0
2	0 1 0 0 0 0 0 0	0 0 1 0 0 1 1 1
3	0 0 1 0 0 0 0 0	1 0 0 0 1 1 1 1
4	0 0 0 1 0 0 0 0	1 1 0 1 1 0 1 1
5	0 0 0 0 1 0 0 0	1 1 1 1 0 0 0 1
6	0 0 0 0 0 1 0 0	1 1 1 0 0 1 0 0
7	0 0 0 0 0 0 1 0	0 1 1 1 0 0 1 0
8	0 0 0 0 0 0 0 1	0 0 1 1 1 0 0 1

An example code word can be constructed for the USASCII letter "A" as follows.
"A" = `$41` encodes to `$41 1e = 01 00 00 01  00 01 11 10`

5.7 Golay Code Generator Matrices

There are two Golay codes used for the voice information. These are the (23,12,7) standard Golay code and the (18,6,8) shortened Golay code. Both of these codes are derived from the standard (23,12,7) Golay code that is generated by the polynomial g(x) given below. The variable x is used for elements in an extension Galois Field, which in the case of the Golay code is $GF(2^{23})$.

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 = 6165 \text{ in octal}$$

It is convenient to use a more compact notation for the matrices generated by this polynomial, so octal notation will be used. To convert to octal, we first express the polynomial in binary notation by evaluating the polynomial with x=2. This yields a number that can be converted to binary, octal or decimal on any standard hexadecimal calculator. Hence:

```
g(2) = 3189 rad 10 = 110001110101 rad 2 = 6165 rad 8
```

It is easy to see that the bits in the binary notation correspond to the coefficients in the polynomial notation.

The (24,12,8) extended Golay code is formed by appending a single parity check to the (23,12,7) standard Golay code. The (18,6,8) shortened Golay code is formed by deleting the left most 6 bits from the (24,12,8) extended Golay code. The k x n generator matrices are given in Table 5-3. Because the systematic form of the codes are used, the left hand k x k portion of the generator matrix is an identity matrix.

Table 5-3 Golay Generator Matrices

row	(24,12,8)	(18,6,8)
1	4000 6165	40 3315
2	2000 3073	20 1547
3	1000 7550	10 6706
4	0400 3664	04 5227
5	0200 1732	02 4476
6	0100 6631	01 4353
7	0040 3315	
8	0020 1547	
9	0010 6706	← the least significant bit
10	0004 5227	is not used for the
11	0002 4476	(23,12,7) standard Golay
12	0001 4353	

5.8 Hamming Code Generator Matrices

The generator matrix for the (15,11,3) standard Hamming code and the (10,6,3) shortened Hamming code are given in Table 5-4. They are derived from the code that is generated by the polynomial:

$$g(x) = x^4 + x + 1 = 23 \text{ in octal}$$

The shortened code is determined by deleting particular rows and columns from the standard Hamming code, not just the left hand columns and top rows. The reason for this is to improve the likelihood of error detection for the shortened code. The (15,11,3) standard Hamming code is derived from the cyclic version by arranging the rows of the four parity check columns in ascending order.

Table 5-4 Generator Matrices for Hamming Codes

row	(10,6,3) short code	(15,11,3) standard Hamming code
1	1 0 0 0 0 0 1 1 1 0	1 0 0 0 0 0 0 0 0 0 0 1 1 1 1
2	0 1 0 0 0 0 1 1 0 1	0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
3	0 0 1 0 0 0 1 0 1 1	0 0 1 0 0 0 0 0 0 0 0 1 1 0 1
4	0 0 0 1 0 0 0 1 1 1	0 0 0 1 0 0 0 0 0 0 0 1 1 0 0
5	0 0 0 0 1 0 0 0 1 1	0 0 0 0 1 0 0 0 0 0 0 1 0 1 1
6	0 0 0 0 0 1 1 1 0 0	0 0 0 0 0 1 0 0 0 0 0 1 0 1 0
7		0 0 0 0 0 0 1 0 0 0 0 1 0 0 1
8		0 0 0 0 0 0 0 1 0 0 0 0 1 1 1
9		0 0 0 0 0 0 0 0 1 0 0 0 1 1 0
10		0 0 0 0 0 0 0 0 0 1 0 0 1 0 1
11		0 0 0 0 0 0 0 0 0 0 1 0 0 1 1

5.9 Reed-Solomon Code Generator Matrices

There are three Reed-Solomon codes used for the voice information. These are the (36,20,17) code, the (24,16,9) code, and the (24,12,13) code. These codes are all shortened from codes of length 63 by deleting the left most information symbols of the longer code. For example, the (36,20,17) code is shortened from the (63,47,17) code by deleting the left most 27 symbols.

Each symbol in the RS code is an element of the extension Galois Field GF(2^6) that is generated by the primitive characteristic polynomial $\alpha^6+\alpha+1$. The elements of GF(2^6) are represented in polynomial form in the variable α with binary coefficients and degree less than 6. These polynomials are compactly and conveniently represented by a 2-digit octal number by evaluating the polynomial for α=2 and then converting the number to octal, just as was done for the Golay generator polynomial above. Some representative examples are given below.

Polynomial	Binary	Octal	Exponential
0	000 000	00	---
1	000 001	01	α^0
α	000 010	02	α^1
$\alpha+1$	000 011	03	α^6
α^2	000 100	04	α^2
...	...	...	...
$\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha$	111 110	76	α^{57}
$\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha+1$	111 111	77	α^{58}

It is useful to also represent the elements of the field in an exponential form as shown in the last column of the examples given above. This is done by reducing the powers of α modulo the primitive characteristic polynomial. Tables of all the elements of the field in octal and exponential form are given below.

$\alpha^e = b_5\, \alpha^5 + b_4\, \alpha^4 + \ldots\, b_1\, \alpha + b_0$
e = exponent expressed in decimal radix
b = octal representation of bits (b_5, b_4, ... b_1, b_0)

Exponential: $b = \alpha^e$

e	0	1	2	3	4	5	6	7
0	01	02	04	10	20	40	03	06
8	14	30	60	43	05	12	24	50
16	23	46	17	36	74	73	65	51
24	21	42	07	16	34	70	63	45
32	11	22	44	13	26	54	33	66
40	57	35	72	67	55	31	62	47
48	15	32	64	53	25	52	27	56
56	37	76	77	75	71	61	41	01

Logarithm: $e = \log(b)$

b	0	1	2	3	4	5	6	7
00	-	0	1	6	2	12	7	26
10	3	32	13	35	8	48	27	18
20	4	24	33	16	14	52	36	54
30	9	45	49	38	28	41	19	56
40	5	62	25	11	34	31	17	47
50	15	23	53	51	37	44	55	40
60	10	61	46	30	50	22	39	43
70	29	60	42	21	20	59	57	58

The generator polynomial of an (N,K,D) Reed-Solomon code is given by the following formula. Let R = N–K. Then:

$$g(x) = (x + \alpha)\,(x + \alpha^2)\,(x + \alpha^3)\ \ldots\ (x + \alpha^R)$$

This can be directly evaluated for all three codes with the help of the GF(2^6) exponential and logarithm tables given above. The results are given below. The coefficients are in octal while the exponents are in decimal.

$$g_{HDR}(x) = 60 + 73\,x + 46\,x^2 + 51\,x^3 + 73\,x^4 + 05\,x^5 + 42\,x^6 + 64\,x^7 + 33\,x^8 + 22\,x^9 + 27\,x^{10} + 21\,x^{11} + 23\,x^{12} + 02\,x^{13} + 35\,x^{14} + 34\,x^{15} + x^{16}$$

$$g_{LC}(x) = 50 + 41\,x + 02\,x^2 + 74\,x^3 + 11\,x^4 + 60\,x^5 + 34\,x^6 + 71\,x^7 + 03\,x^8 + 55\,x^9 + 05\,x^{10} + 71\,x^{11} + x^{12}$$

$$g_{ES}(x) = 26 + 06\,x + 24\,x^2 + 57\,x^3 + 60\,x^4 + 45\,x^5 + 75\,x^6 + 67\,x^7 + x^8$$

The generator matrix is easily constructed from the generator polynomial. In the case of the Reed-Solomon code, the natural expression for the generator matrix uses hex bit symbols. In the natural form, the generator matrix is a K by N matrix of GF(2^6) symbols. The matrices are expressed in systematic form so the left hand K by K matrix is an identity matrix. The generator matrices for the LC and ES codes are fully documented below. The Header code is large enough that the identity part of the matrix is left out, and only the right hand parity portion of the matrix is given.

row	$\mathbf{G}_{LC}$ matrix of (24,12,13) shortened Reed-Solomon code					
1	01 00 00 00	00 00 00 00	00 00 00 00	62 44 03 25	14 16 27 03	53 04 36 47
2	00 01 00 00	00 00 00 00	00 00 00 00	11 12 11 11	16 64 67 55	01 76 26 73
3	00 00 01 00	00 00 00 00	00 00 00 00	03 01 05 75	14 06 20 44	66 06 70 66
4	00 00 00 01	00 00 00 00	00 00 00 00	21 70 27 45	16 67 23 64	73 33 44 21
5	00 00 00 00	01 00 00 00	00 00 00 00	30 22 03 75	15 15 33 15	51 03 53 50
6	00 00 00 00	00 01 00 00	00 00 00 00	01 41 27 56	76 64 21 53	04 25 01 12
7	00 00 00 00	00 00 01 00	00 00 00 00	61 76 21 55	76 01 63 35	30 13 64 70
8	00 00 00 00	00 00 00 01	00 00 00 00	24 22 71 56	21 35 73 42	57 74 43 76
9	00 00 00 00	00 00 00 00	01 00 00 00	72 42 05 20	43 47 33 56	01 16 13 76
10	00 00 00 00	00 00 00 00	00 01 00 00	72 14 65 54	35 25 41 16	15 40 71 26
11	00 00 00 00	00 00 00 00	00 00 01 00	73 65 36 61	42 22 17 04	44 20 25 05
12	00 00 00 00	00 00 00 00	00 00 00 01	71 05 55 03	71 34 60 11	74 02 41 50

row	$\mathbf{G}_{ES}$ matrix of (24,16,9) shortened Reed-Solomon code					
1	01 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	51 45 67 15	64 67 52 12
2	00 01 00 00	00 00 00 00	00 00 00 00	00 00 00 00	57 25 63 73	71 22 40 15
3	00 00 01 00	00 00 00 00	00 00 00 00	00 00 00 00	05 01 31 04	16 54 25 76
4	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	73 07 47 14	41 77 47 11
5	00 00 00 00	01 00 00 00	00 00 00 00	00 00 00 00	75 15 51 51	17 67 17 57
6	00 00 00 00	00 01 00 00	00 00 00 00	00 00 00 00	20 32 14 42	75 42 70 54
7	00 00 00 00	00 00 01 00	00 00 00 00	00 00 00 00	02 75 43 05	01 40 12 64
8	00 00 00 00	00 00 00 01	00 00 00 00	00 00 00 00	24 74 15 72	24 26 74 61
9	00 00 00 00	00 00 00 00	01 00 00 00	00 00 00 00	42 64 07 22	61 20 40 65
10	00 00 00 00	00 00 00 00	00 01 00 00	00 00 00 00	32 32 55 41	57 66 21 77
11	00 00 00 00	00 00 00 00	00 00 01 00	00 00 00 00	65 36 25 07	50 16 40 51
12	00 00 00 00	00 00 00 00	00 00 00 01	00 00 00 00	64 06 54 32	76 46 14 36
13	00 00 00 00	00 00 00 00	00 00 00 00	01 00 00 00	62 63 74 70	05 27 37 46
14	00 00 00 00	00 00 00 00	00 00 00 00	00 01 00 00	55 43 34 71	57 76 50 64
15	00 00 00 00	00 00 00 00	00 00 00 00	00 00 01 00	24 23 23 05	50 70 42 23
16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 01	67 75 45 60	57 24 06 26

row	$\mathbf{P}_{HDR}$ matrix of (36,20,17) shortened Reed-Solomon code			
1	74 37 34 06	02 07 44 64	26 14 26 44	54 13 77 05
2	04 17 50 24	11 05 30 57	33 03 02 02	15 16 25 26
3	07 23 37 46	56 75 43 45	55 21 50 31	45 27 71 62
4	26 05 07 63	63 27 63 40	06 04 40 45	47 30 75 07
5	23 73 73 41	72 34 21 51	67 16 31 74	11 21 12 21
6	24 51 25 23	22 41 74 66	74 65 70 36	67 45 64 01
7	52 33 14 02	20 06 14 25	52 23 35 74	75 75 43 27
8	55 62 56 25	73 60 15 30	13 17 20 02	70 55 14 47
9	54 51 32 65	77 12 54 13	35 32 56 12	75 01 72 63
10	74 41 30 41	43 22 51 06	64 33 03 47	27 12 55 47
11	54 70 11 03	13 22 16 57	03 45 72 31	30 56 35 22
12	51 07 72 30	65 54 06 21	36 63 50 61	64 52 01 60
13	01 65 32 70	13 44 73 24	12 52 21 55	12 35 14 72
14	11 70 05 10	65 24 15 77	22 24 24 74	07 44 07 46
15	06 02 65 11	41 20 45 42	46 54 35 12	40 64 65 33
16	34 31 01 15	44 64 16 24	52 16 06 62	20 13 55 57
17	63 43 25 44	77 63 17 17	64 14 40 74	31 72 54 06
18	71 21 70 44	56 04 30 74	04 23 71 70	63 45 56 43
19	02 01 53 74	02 14 52 74	12 57 24 63	15 42 52 33
20	34 35 02 23	21 27 22 33	64 42 05 73	51 46 73 60

6 Data Packets

Data service is provided by the Common Air Interface by the transmission and reception of data packets. The data service operational detail is provided by additional specifications. The reader is referred to reference [1], *Project 25 System and Standard Definition.*

6.1 General Description

Data messages are transferred over the Common Air Interface with a packet technique. The data message is first split into fragments, where each fragment is the information contained in a packet. The message fragments are then formed into packets, and the packets are in turn split into a sequence of information blocks. Each block is protected by a trellis code, and the sequence of blocks is transferred over the Common Air Interface as a single packet. Part of the packet is a check sum to be used to verify the accuracy of the error correction in the receiver. If the packet is corrupted in confirmed data mode during reception, then an automatic retransmission request is generated (ARQ) to repeat parts of the packet. The receiver then reassembles the packets into a continuous message.

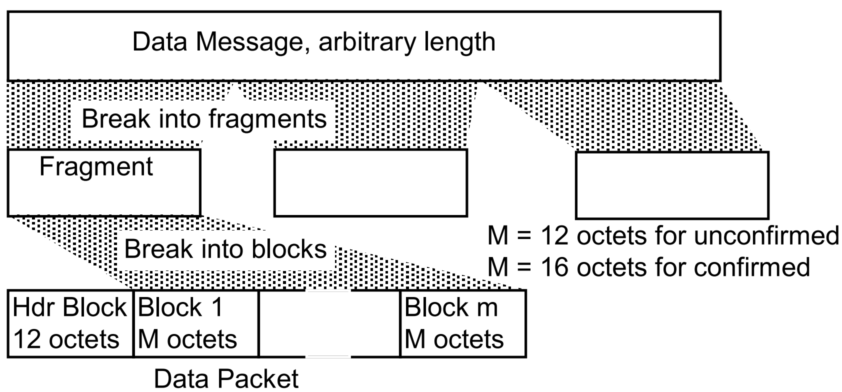

Figure 6-1 Decomposition of Data Message into Packets

The maximum number of fragments of a single data message is not bounded. The only restriction on the fragments is that they must not be longer than a maximum length. Each fragment is in turn split into blocks with each block containing either 16 or 12 octets of data, for confirmed and unconfirmed data transmission respectively. A special block called a header block is sent at the beginning of the packet. The maximum number of blocks in a packet, including the header block, is upper bounded by the maximum fragment length. The maximum fragment length sets a minimum storage limit on the BR and MR of a Project-25 system. The MR and BR shall have storage for a fragment of at least 512 octets. Larger fragments are possible if transmitters and receivers are designed for them. In such a circumstance, the fragments need not be limited to 512 octets.

6.2 Header Block Structure

The header block contains 10 octets of address and control information, followed by 2 octets of a header CRC error detection code (see Figure 6-2). The header CRC is calculated from the first 10 octets of address and control using the cyclical redundant coding procedure commonly referred to as CRC-CCITT. The particulars of the calculation follow.

Consider the 80 header bits as the coefficients of a polynomial M(x) of degree 79, associating the MSB of the zero-th header octet with x^{79} and the LSB of the ninth header octet with x^0. Define the generator polynomial, $G_H(x)$, and the inversion polynomial, $I_H(x)$.

$$G_H(x) = x^{16} + x^{12} + x^5 + 1 \qquad I_H(x) = x^{15} + x^{14} + x^{13} + \ldots + x^2 + x + 1$$

The header CRC polynomial, $F_H(x)$, is then computed from the formula,

$$F_H(x) = (x^{16} M(x) \bmod G_H(x)) + I_H(x) \qquad \text{modulo 2, i.e., in GF(2)}$$

The coefficients of $F_H(x)$ are placed in the CRC field with the MSB of the zero-th octet of the CRC corresponding to x^{15} and the LSB of the next octet of the CRC corresponding to x^0.

6.3 Data Block Structure

Data packets use two different structures for two different types of data. Data may be sent with either confirmed delivery or unconfirmed delivery. Confirmed delivery is used to require the recipient of the packet to send an acknowledgment of receipt. Unconfirmed delivery is used if the originator of the packet does not require an acknowledgment. The distinction between confirmed and unconfirmed is signified in the header block.

6.3.1 Unconfirmed Data Block Structure

In the case of unconfirmed delivery, the data packet has a 32 bit CRC over the data contents to allow the recipient to determine if the packet has been received without error. This CRC is positioned at the end of the packet, as the last four octets in the last block of the packet. If the required data does not exactly fill all the blocks then pad octets shall be added to the end of the data in the packet, but before the CRC, to extend the total packet length to exactly fill all of the blocks in the packet. Packets with unconfirmed delivery always put 12 octets in each block, and protect each block with a rate 1/2 trellis code. The number of blocks in the packet which follow the header block, the number of pad octets in the packet, and the fact that it uses unconfirmed delivery, are all indicated in the header block.

6.3.2 Confirmed Data Block Structure

In the case of confirmed delivery, the individual blocks in the packet each have separate CRC parity checks. Each block contains 16 octets of data, and two octets of numbering and CRC information. The number and CRC octets allow the recipient to distinguish those data blocks which were received correctly. When a data block is received with a detectable error, then the recipient shall send an acknowledgment back to the sender to request a retransmission of the corrupted block. This is called selective ARQ. Each block in the packet is protected with a rate 3/4 trellis code. As in the case of unconfirmed delivery, pad octets shall be used to exactly fill all of the blocks in the packet.

6.3.3 Packet CRC

The Last Block in a packet comprises several octets of user information and / or pad octets, followed by a 4-octet CRC parity check. This is referred to as the packet CRC.

The packet CRC is a 4-octet cyclic redundancy check coded over all of the data octets included in the Intermediate Blocks and the octets of user information of the Last Block. The specific calculation is as follows.

Let k be the total number of user information and pad bits over which the packet CRC is to be calculated. Consider the k message bits as the coefficients of a polynomial M(x) of degree k–1, associating the MSB of the zero-th message octet with x^{k-1} and the LSB of the last message octet with x^0. Define the generator polynomial, $G_M(x)$, and the inversion polynomial, $I_M(x)$.

$$G_M(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

$$I_M(x) = x^{31} + x^{30} + x^{29} + ... + x^2 + x + 1$$

The packet CRC polynomial, $F_M(x)$, is then computed from the following formula.

$$F_M(x) = (x^{32} M(x) \bmod G_M(x)) + I_M(x) \qquad \text{modulo 2, i.e., in GF(2)}$$

The coefficients of $F_M(x)$ are placed in the CRC field with the MSB of the zero-th octet of the CRC corresponding to x^{31} and the LSB of the third octet of the CRC corresponding to x^0.

6.4 Confirmed Data Packet Formats

6.4.1 Confirmed Data Packet Header Format

The confirmed data packet header block is shown in Figure 6-2. The bits are filled in for the fields with static definitions. Additional fields are labeled and described below. Reserved bits in the block are set to 0. This packet header is used for user data messages which require an acknowledgment upon receipt.

Header Block

	7	6	5	4	3	2	1	0
octet 0	0	A	IO	1	0	1	1	0
1	1	1	SAP Identifier					
2	Manuf. ID							
3	Logical Link ID							
4								
5								
6	F	Blocks to Follow						
7	0	0	0	Pad Octet Ct				
8	S	N(S)			FSNF			
9	0	0	Data Header Offset					
10	Header CRC							
octet 11								

Figure 6-2 Confirmed Data Packet Header Block

A/N (bit 6 of octet 0) = 1 to indicate that confirmation for this packet is desired.

I/O (bit 5 of octet 0) = 1 for outbound messages, 0 for inbound messages.

Format (bits 4 – 0 of octet 0) = %10110 to identify this message as a data packet with confirmed delivery.

SAP Identifier identifies the Service Access Point to which the data is directed. Defined SAP values are in a supplementary document.

Manufacturer's ID identifies the manufacturer for non-standard data functions. Defined values are in a supplementary document.

Logical Link ID identifies the MR (subscriber unit) which sent the packet for inbound messages, or the MR to which the packet is directed for outbound messages.

FMF (bit 7 of octet 6) is the Full Message Flag (FMF). The FMF = 1 on the first try for the complete packet and FMF = 0 on subsequent retries. The value of FMF=1 is used in the receiver to signify that the **Blocks to Follow** and **Pad Octet Count** fields indicate the amount of data being transported in the complete packet.

Blocks to Follow specifies the number of blocks in the packet not including the header block.

Pad Octet Count specifies the number of pad octets which have been appended to the data octets to form an integer number of blocks. The actual number of data octets is given by the formula below.

Number Data Octets = 16 x **Blocks to Follow** – 4 – **Pad Octet Count**

Syn (bit 7 of octet 8) is a flag used to re-synchronize the physical sublayer sequence numbers. The receiver accepts the N(S) and FSNF fields in this message if the Syn bit is asserted. This bit effectively disables the rejection of duplicate messages when it is asserted. It should only be asserted on specially defined registration messages. On all user data messages, it should be cleared.

N(S) is the sequence number of the packet. This is used to identify each request packet so that the receiver may correctly order the received message segments and eliminate duplicate copies. It is incremented modulo 8 for each new data packet that is transmitted.

The transmitter shall not increment this counter for an automatic retry. The receiver maintains a receiver variable V(R) which is the sequence number of the last valid packet to be received. The receiver accepts packets with N(S) = V(R) or V(R)+1. If N(S)=V(R), then the packet is a duplicate. If N(S) = V(R)+1, then the packet is the next one in the sequence.

FSNF is the Fragment Sequence Number Field. This is used to consecutively number message fragments that together make up a longer data message. The most significant bit shall be asserted for the last fragment in the chain, and shall be cleared otherwise. The three least significant bits correspond to the Fragment Sequence Number (FSN). They shall be set to %000 for the first fragment and shall be incremented for each subsequent fragment. When the number reaches %111 the next increment shall be %001 and not %000. A logical message consisting of a single fragment (or packet) shall have a value of %1000 for the FSNF.

Data Header Offset is an offset pointer that divides the data portion of the packet into two components, data header and data information. A data header offset value of 3 would specify that the first three octets in the user data comprise the data header and that the actual information begins in the fourth byte. The use of a data header is dependent on the application.

Header CRC is the CRC parity check described in Section 6.2 for the header block.

6.4.2 Confirmed Data Packet Data Block Format

The blocks of user data are diagrammed in Figure 6-3. The first two octets contain a serial number and a CRC. The remaining 16 octets may contain user data. In the case of the Last Block of the packet the 16 octets may contain some pad octets and a 4 octet CRC defined in Section 6.3.3. The next to last block in the packet may also contain some pad octets, if the **Pad Octet Count** in the Header Block exceeds 12.

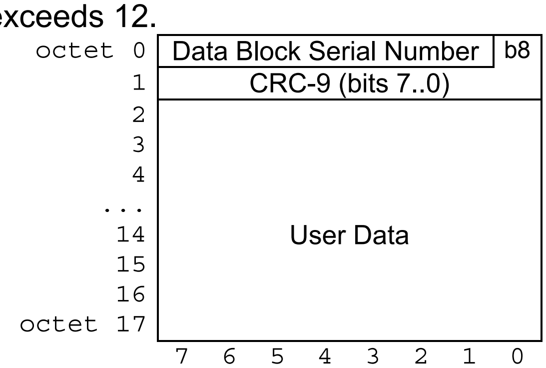

Figure 6-3 Confirmed Data Packet Data Block

Serial Number (bits 7..1 of octet 0) is the serial number for the block within the packet. On the first try these serial numbers start at 0 and increment up to M–1, where M is equal to the **Blocks To Follow** field in the Header Block. On subsequent retries, not all blocks are generally included, and these serial numbers allow the transmitter to indicate which blocks are being sent.

CRC-9 (octet 1 and bit 0 of octet 0) is the 9-bit CRC for the data block. The computation of this field is described below.

The transmitter computes the CRC-9 as follows. First, the 16 octets of user data and the seven bits of the serial number are arranged as 135 bits, with the serial number being the first seven bits. These are then considered to be the coefficients of a message polynomial, M(x), of degree 134, with

bit 6 of the Serial Number corresponding to a coefficient of the x^{134} term,
bit 5 of the Serial Number corresponding to the x^{133} term,
...
bit 0 of the Serial Number corresponding to the x^{128} term,
bit 7 of octet 2 corresponding to the x^{127} term,
bit 6 of octet 2 corresponding tot he x^{126} term,
...
bit 1 of octet 17 corresponding to the x^1 term, and
bit 0 of octet 17 corresponding to the x^0 term.

Define the generator polynomial, $G_9(x)$, and the inversion polynomial, $I_9(x)$.

$G_9(x) = x^9 + x^6 + x^4 + x^3 + 1$ $\qquad$ $I_9(x) = x^8 + x^7 + x^6 + \ldots x + 1$

The CRC-9 polynomial, $F_9(x)$, shall be computed from the formula,

$F_9(x) = (x^9 M(x) \bmod G_9(x)) + I_9(x)$ $\qquad$ modulo 2, i.e., in GF(2)

The coefficients of $F_9(x)$ are placed in the CRC-9 field with the MSB corresponding to bit 0 of octet 0, the next most significant bit corresponding to bit 7 of octet 1, and the LSB corresponding to bit 0 of octet 1.

6.4.3 Confirmed Data Packet Last Data Block Format

The last block of the packet is diagrammed in Figure 6-4. Up to 12 pad octets may be included in the last data block. If the message has more than 12 pad octets, then additional pad octets are included in the second to last data block. The last four octets of the last data block consist of the packet CRC, as described in Section 6.3.3.

octet	bits 7 6 5 4 3 2 1	bit 0
octet 0	Data Block Serial Number	b8
1	CRC-9 (bits 7..0)	
2 3 . . . 13	User Data followed by Pad Octets	
14 15 16 octet 17	Packet CRC	

Figure 6-4 Confirmed Data Packet Last Data Block

6.5 Response Packet Format

The response packet header block is shown in Figure 6-5. This packet may have data blocks following it as shown in Figure 6-6. It is used to confirm delivery for data packets sent with the A/N bit set, as shown in the Confirmed Data Packet format.

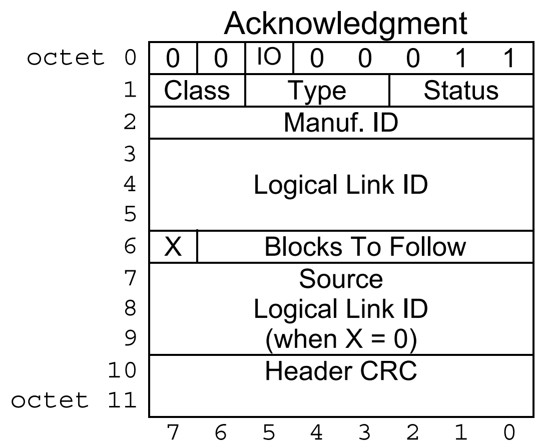

Figure 6-5 Response Packet Format

I/O (bit 5 of octet 0) indicates the direction of transmission. It is 1 for outbound messages and 0 for inbound messages.

Format (bits 4 .. 0 of octet 0) = %00011 to identify this message as a Response.

Class, **Type**, and **Status** specify the meaning of the response. An abbreviated table of the meanings is given in Table 6-1.

Table 6-1 Response Packet Class, Type, and Status Definitions

Class	Type	Status	Meaning
%00	%001	N(R)	ACK -- All blocks successfully received
%01	%000	N(R)*	NACK -- Illegal Format
%01	%001	N(R)	NACK -- Packet CRC (32-bit) parity check failure
%01	%010	N(R)	NACK -- Memory Full
%01	%011	FSN	NACK -- Out of logical sequence FSN
%01	%100	N(R)	NACK -- Undeliverable
%01	%101	V(R)	NACK -- Out of sequence, N(S) ≠ V(R) or V(R)+1
%01	%110	N(R)	NACK -- Invalid User disallowed by the system
%10	%000	N(R)	ACK -- Selective Retry for some blocks

* In the case of an illegal format, the N(R) may have no real meaning.

Logical Link ID (octets 3, 4, 5) has the same meaning as for the Confirmed Data Packet Format in Section 6.4. In the case of Enhanced Addressing used in Section 6.7, this field contains the Destination Logical Link ID.

X (bit 7 of octet 6) is set to %1 for Response Packets sent in response to the Confirmed Delivery packets defined in Section 6.4. The X bit is cleared to %0 for Response Packets sent in response to Enhanced Address Confirmed Delivery packets defined in Section 6.7.

Blocks To Follow indicates the number of additional blocks which contain flags for selective retries.

Source Logical Link ID (octets 7, 8, 9) is set to nulls when X=1. It is set to the address of the responding MR (or BR) when X=0.

Header CRC is computed with the method described in Section 6.2.

N(R) is the sequence number N(S) received in the confirmed data packet.

In the case of NACK Type %011, the FSN value fills the Status field. The FSN is the Fragment Sequence Number, namely the three least significant bits of the FSNF field. It indicates the FSN received in the packet that was unexpected by the receiver.

In the case where blocks are to be selectively retried, the Class field is set to %10, and subsequent blocks of additional information are appended to the header block. The number of blocks is indicated in the Blocks To Follow field. The format for subsequent blocks is given in Figure 6-6 for the case where only a single block follows. It contains selective retry flags for up to 64 blocks. If more flags are necessary, then two blocks may be used and flags for up to 127 blocks are sent. The maximum number of blocks in a packet is 127.

	7	6	5	4	3	2	1	0
octet 0	f7	f6	f5	f4	f3	f2	f1	f0
1	15		.	.	.		f9	f8
2	23		.	.	.			16
3	31		.	.	.			24
4							33	32
5								
6								
7								
8	CRC							
9								
10								
octet 11								

Figure 6-6 Response Packet Data Block

The Flag bits are set to 1 to indicate the receipt of the corresponding block, and they are set to 0 to indicate that the block should be retried. They are placed in the order shown in Figure 6-6. Unused flag bits, bits corresponding to blocks with numbers higher than are used in the packet, shall be set to 1.

The CRC is the 32 bit CRC defined in Section 6.3.3.

6.6 Unconfirmed Data Packet Header Format

The format of the Unconfirmed Data Packet Header is shown in Figure 6-7. It is the same as the Confirmed Data Packet Header, except that the **Syn**, **N(S)**, and **FSNF** fields are nulled out. The **A/N** bit (bit 6 of octet 0) is cleared to 0 to indicate unconfirmed delivery. This header is useful for data messages which are not to be acknowledged by this layer 2 protocol. This includes certain classes of Over-the-Air-Rekeying messages which are acknowledged in a delayed fashion, or which are acknowledged in an encrypted fashion. The meaning of the fields of this header can be found in the Confirmed Data Packet Header Format given in Section 6.4.

Header Block

octet	7	6	5	4	3	2	1	0
octet 0	0	A	IO	1	0	1	0	1
1	1	1	SAP Identifier					
2	Manuf. ID							
3, 4, 5	Logical Link ID							
6	1	Blocks to Follow						
7	0	0	0	Pad Octet Count				
8	reserved octet							
9	0	0	Data Header Offset					
10, octet 11	Header CRC							

Figure 6-7 Unconfirmed Data Packet Header Format

The data blocks which follow this Header Block consist of 12 octets of user data. There is no CRC or Serial Number in each block as in the case of the format for Confirmed Delivery in Section 6.4. The Last Block shall contain a packet CRC in the last four octets as defined in section 6.3.3. This is shown in Figure 6-8. The number of octets of user data is computed in a slightly different fashion than in the Confirmed Data case. The formula is:

Number Data Octets = 12 x **Blocks To Follow** – 4 – **Pad Octet Count**

octet	7 6 5 4 3 2 1 0
octet 0, 1, ..., 7	User Data followed by Pad Octets
8, 9, 10, octet 11	Packet CRC

Figure 6-8 Unconfirmed Data Packet Last Data Block

6.7 Enhanced Addressing Format

Enhanced Addressing is used when subscribers have to send data directly between each other without necessarily relying on fixed network equipment. This service requires both a source and a destination address on every packet of information. For this purpose, a special SAP Identifier, denoted as the "EA SAP" is used to signify that a second address is inserted in the packet before the user data. The general format of an Enhanced Addressing data message with Unconfirmed delivery is shown in Figure 6-9. The format for Confirmed delivery is given in Figure 6-10.

The header block conveys the destination ID, so the I/O bit in header octet 0 is set to %1 to signify a destination address. The A/N bit is set or cleared in the usual way to indicate confirmed or unconfirmed delivery. In the case of Unconfirmed delivery, the bit is cleared.

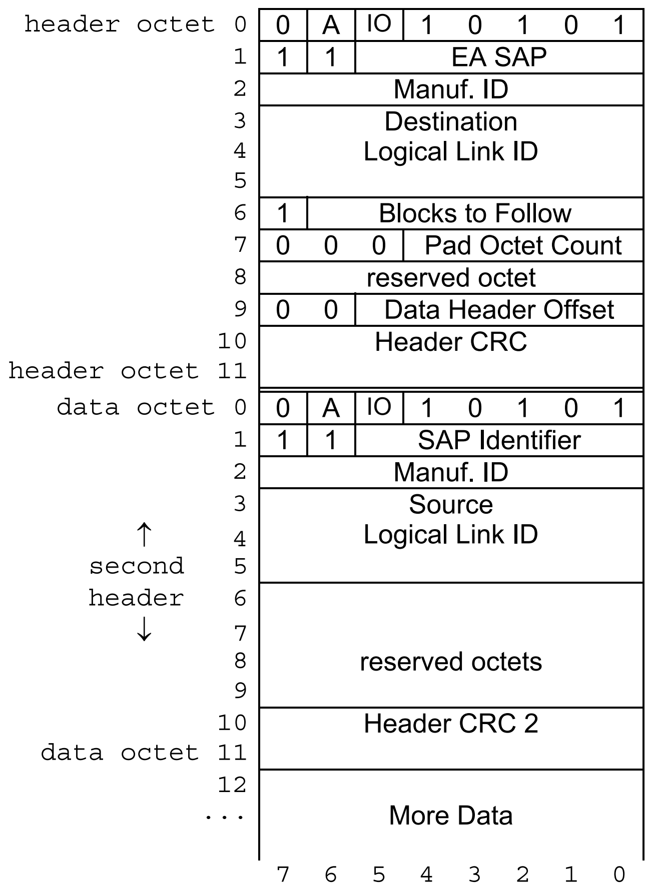

Figure 6-9 Unconfirmed Enhanced Addressing Packet Header Format

The first data block in the Unconfirmed packet conveys the source address of the message in a format that is identical to a header block. It is referred to as the second header. The I/O bit is cleared in data octet 0 to indicate that the address is a source address. The SAP Identifier for the message is also conveyed in this block in octet 1. Octets 6 through 9 are unused, and then the normal header CRC is calculated over this data block and put into octets 10 and 11.

The format for a Confirmed data packet with Enhanced Addressing is shown in Figure 6-10 below. In this packet the A/N bit in the Header Block is set to 1. The first data block now contains a reservation of four octets to convey the address of the Source as well as the SAP Identifier for the user data.

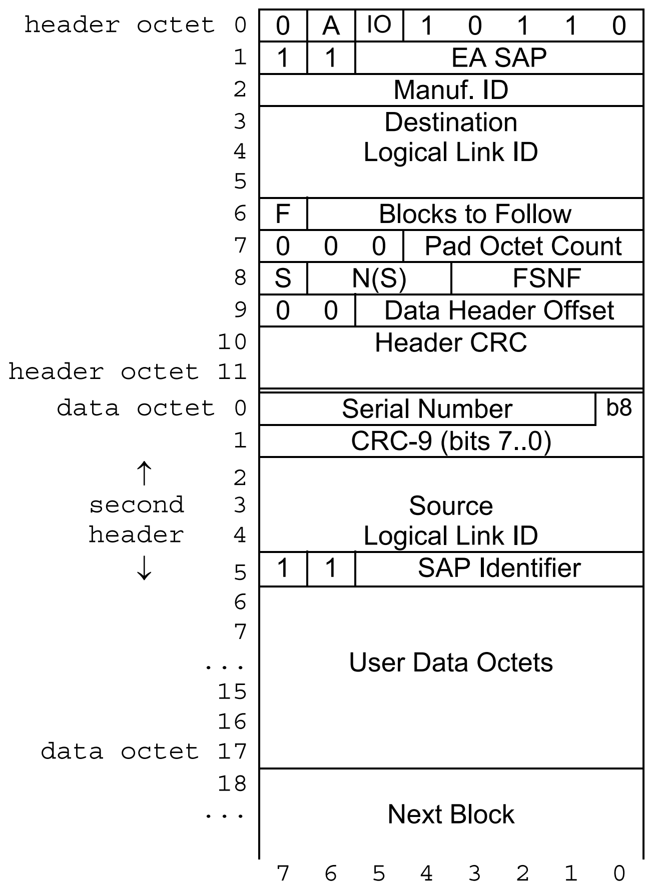

Figure 6-10 Confirmed Enhanced Addressing Packet Header Format

7 Data Error Correction

The data blocks for Unconfirmed data packets use a rate 1/2 trellis code while data blocks for Confirmed data packets use a rate 3/4 trellis code. The Header block for data packets always uses a rate 1/2 trellis code. The encoding process of the rate 3/4 code is diagrammed in Figure 7-1.

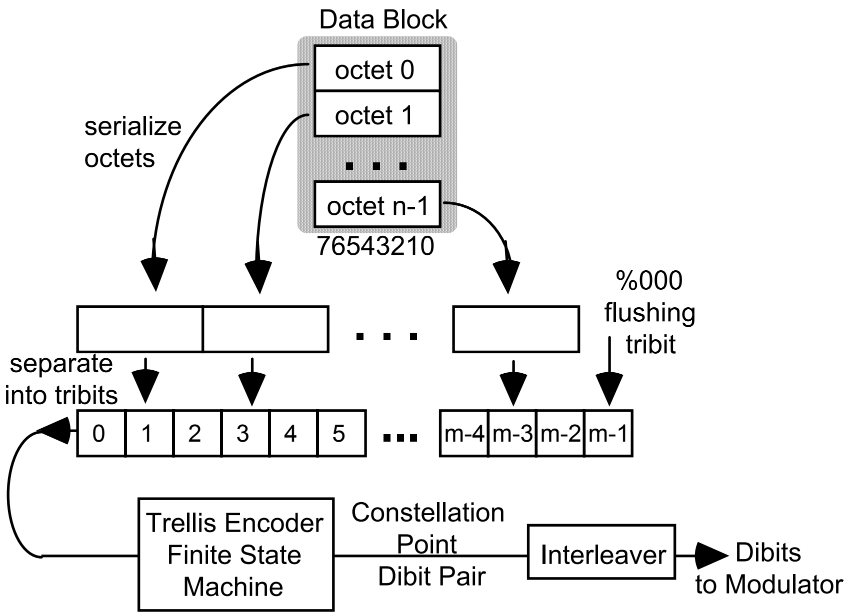

Figure 7-1 Rate 3/4 Trellis Encoder Overview

The encoding process begins by serializing a sequence of octets as shown, from left to right, and then separating the result into a serial stream of tribits. Each tribit contains three bits with the most significant bit to the left and the least significant bit to the right. Consequently, each tribit is represented by an octal number in the range 0 to 7. The tribit stream is applied to the trellis encoder, starting with tribit 0 and ending with tribit m–1.

The process for the rate 1/2 code is similar. The data octets are serialized in the same way, except that they are separated into dibits instead of tribits. The trellis encoder processes individual dibits on the input and produces a constellation point consisting of a dibit pair on the output. Hence, 2 bits go in and 4 bits go out, resulting in a rate 1/2.

Table 7-1 Trellis Code Word Sizes

	Rate 3/4	Rate 1/2
Input Size	48 tribits	48 dibits
Output Size	98 dibits	98 dibits
(n,k)	(196,144)	(196,96)

The trellis encoder is implemented as a finite state machine, or FSM. It appends a %00 dibit (or %000 tribit) at the end of the stream to flush out the final state.

The dibits on the output are mapped to ±1, ±3 amplitudes and then interleaved before being modulated.

7.1 Finite State Machine

The trellis encoder receives m dibits (or tribits) as input, and outputs 2m dibits. The encoding process is diagrammed below in Figure 7-2. The encoder is a 4-state finite state machine for the rate 1/2 code and an 8-state finite state machine (FSM) for the rate 3/4 code, with an initial state of 0. The FSM used in this particular implementation has the special property of having the current input as the next state. For each dibit (or tribit) input, there is a corresponding output constellation point which is represented as a dibit pair.

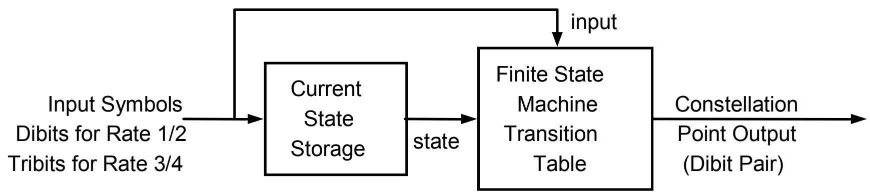

Figure 7-2 Trellis Encoder Block Diagram

The state transition table is shown in Table 7-2 below. The output of the state transition table is one of 16 constellation points. The constellation to dibit pair mapping is shown in Table 7-3 below.

Table 7-2 Trellis Encoder State Transition Tables

Rate 1/2	Input 0	1	2	3
FSM State 0	0	15	12	3
1	4	11	8	7
2	13	2	1	14
3	9	6	5	10

Rate 3/4	Input Tribit 0	1	2	3	4	5	6	7
FSM State 0	0	8	4	12	2	10	6	14
1	4	12	2	10	6	14	0	8
2	1	9	5	13	3	11	7	15
3	5	13	3	11	7	15	1	9
4	3	11	7	15	1	9	5	13
5	7	15	1	9	5	13	3	11
6	2	10	6	14	0	8	4	12
7	6	14	0	8	4	12	2	10

Table 7-3 Constellation to Dibit Pair Mapping

Constellation Point	Dibit 0	Dibit 1	Constellation Point	Dibit 0	Dibit 1
0	+1	–1	**8**	–3	+3
1	–1	–1	**9**	+3	+3
2	+3	–3	**10**	–1	+1
3	–3	–3	**11**	+1	+1
4	–3	–1	**12**	+1	+3
5	+3	–1	**13**	–1	+3
6	–1	–3	**14**	+3	+1
7	+1	–3	**15**	–3	+1

7.2 Interleaver

Interleaving is done for data blocks, both for the rate 3/4 code and the rate 1/2 code. The purpose of the interleaver is to spread burst errors due to Rayleigh fading over the 98 dibit block. In the interleaver, the dibit array is rearranged to form another dibit array according to the interleave table shown in Table 7-4 below. The interleaving table is the same for both codes.

Table 7-4 Interleave Table

INTERLEAVE TABLE							
Output Index	**Input Index**	**Output Index**	**Input Index**	**Output Index**	**Input Index**	**Output Index**	**Input Index**
0	0	**26**	2	**50**	4	**74**	6
1	1	**27**	3	**51**	5	**75**	7
2	8	**28**	10	**52**	12	**76**	14
3	9	**29**	11	**53**	13	**77**	15
4	16	**30**	18	**54**	20	**78**	22
5	17	**31**	19	**55**	21	**79**	23
18	72	**44**	74	**68**	76	**92**	78
19	73	**45**	75	**69**	77	**93**	79
20	80	**46**	82	**70**	84	**94**	86
21	81	**47**	83	**71**	85	**95**	87
22	88	**48**	90	**72**	92	**96**	94
23	89	**49**	91	**73**	93	**97**	95
24	96						
25	97						

8 Channel Access

8.1 General Description

This portion of the Common Air Interface specifies the method of synchronization for frames of information encoded as described in the other Sections. It also specifies a Network Identifier (NID) so that co-channel interference can be rejected by receivers. Finally, a method of signaling channel status to subscriber units is defined so that collisions between data messages are minimized.

Frame synchronization is provided by a special sequence of bits that marks the location of the first bit of the message. This frame synchronization is done at the beginning of every voice and data message, and it is periodically inserted every 180 ms throughout the voice message to allow receivers to pick up voice messages from the middle.

The Network Identifier provides a simple means of addressing radio networks or specific repeaters, depending on the radio system configuration. The NID also allows subscriber units to reject traffic from other radio networks that happen to interfere with desired communications. The NID also has a small field which identifies the type of message (e.g., voice or data) so that the proper error correction can be performed.

Access to the radio channel is done in such a way as to minimize collisions between data messages from different subscriber units, and also to minimize collisions between data and voice. The technique of Carrier Sense Multiple Access (CSMA) may be used for this purpose. On typical repeater channels, there is a radio frequency pair. One frequency is used for inbound messages to the repeater's receiver and another frequency is used for outbound messages from the repeater's transmitter. The repeater is full-duplex, so it can transmit simultaneously while it is receiving. While the repeater is transmitting, it can send status information to all the listening subscriber units about the status (idle or busy) of the inbound channel. When a subscriber unit wishes to transmit a data message, it shall wait until the inbound channel is idle before it transmits, unless the call is an emergency call. Operational detail of channel access may be found in additional specifications for trunked and conventional channel operation. These may be found in references [1] and [3].

8.2 Data Units

Voice and data messages are sent over-the-air as data units. Each data unit has a frame synchronization and Network Identifier word as a preamble. Throughout the data unit there are status symbols interleaved such that there is one status symbol of two bits after every 70 bits of information. There are six different data units, five for voice and one for data. These are diagrammed in Figure 8-1 below.

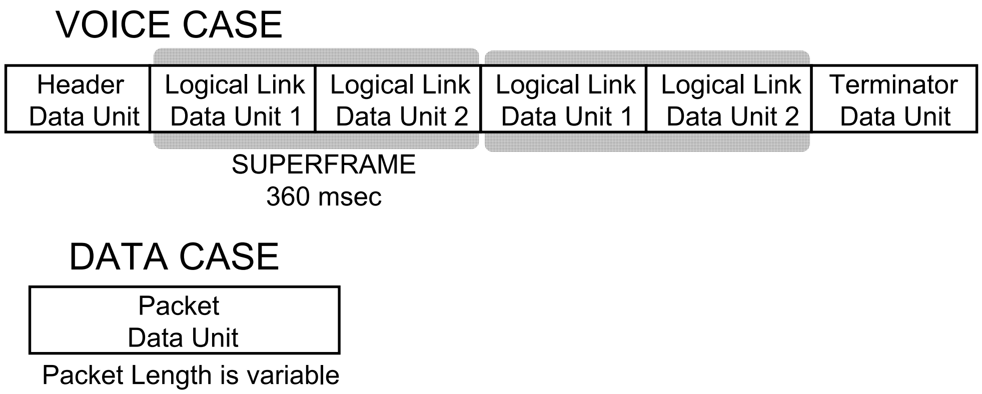

Figure 8-1 Data Units for Voice and Data Messages

8.2.1 Header Data Unit

A diagram of the header data unit is given in Figure 8-2. It consists of the frame synchronization (FS) signal, followed by the Network Identifier (NID), then the header code word of 648 bits. There are ten null bits appended to the end to fill the entire data unit out to 770 bits. This total is then expanded with eleven status symbols to 792 bits. This takes 82.5 ms of air time at 9.6 kbps.

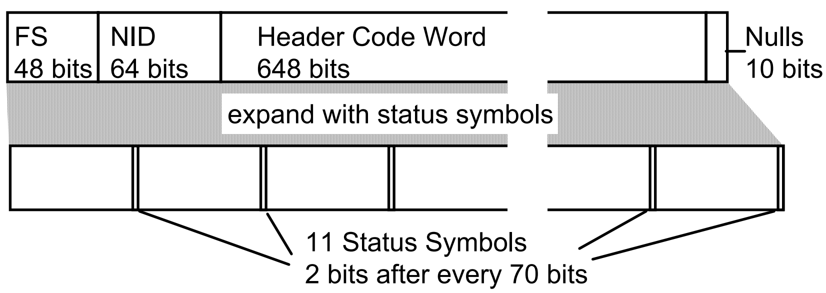

Figure 8-2 Header Data Unit

8.2.2 Superframe Data Units

A superframe is constructed from two data units. These are simply called Logical Link Data Unit 1 and Logical Link Data Unit 2, or LDU1 and LDU2. A diagram of the superframe data units is given in Figures 8-3 and 8-4. Each data unit conveys nine voice code words. Each voice code word is in a frame that spans an average of 20 ms of air time. The voice frames are numbered 1 through 18, with frames 1 through 9 in LDU1 and frames 10 through 18 in LDU2.

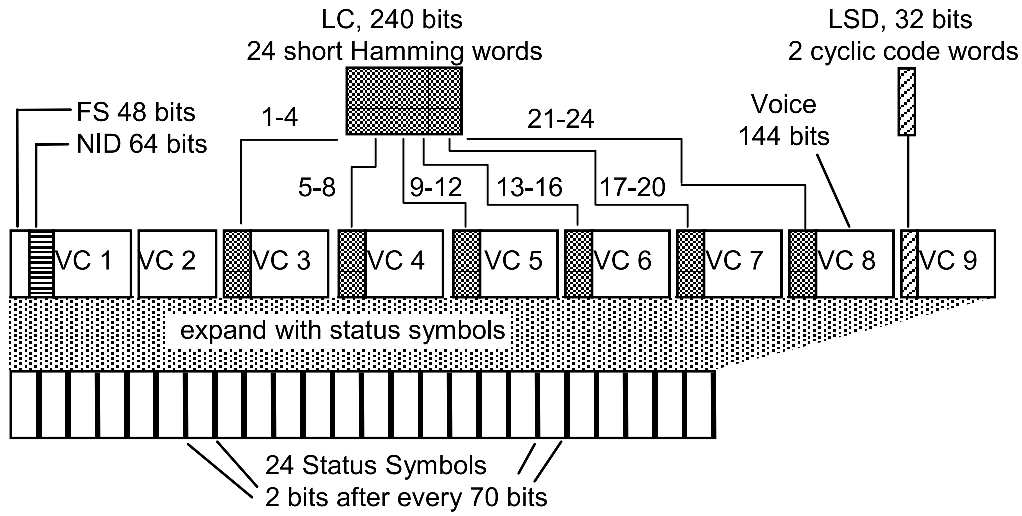

Figure 8-3 Logical Link Data Unit 1

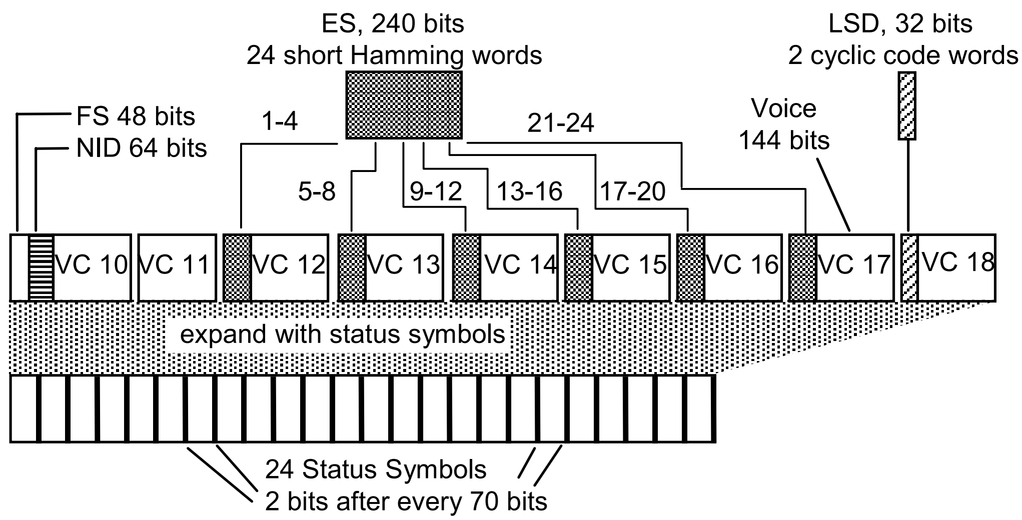

Figure 8-4 Logical Link Data Unit 2

8.2.3 Terminator Data Units

There are two terminating data units for voice messages. The simple one consists solely of a frame sync and Network ID. A more elaborate terminator adds a Link Control word. These are diagrammed in Figures 8-5 and 8-6.

The simple terminating data unit is intended for simple operation. At the end of a voice message, the transmitter sustains the transmission until the Link Data Unit of Section 8.2.2 is completed. This is done by encoding silence for the voice. At the end of the Link Data Unit, the transmitter then sends the simple terminating data unit to signify the end of the message. The terminating data unit may follow either LDU1 or LDU2.

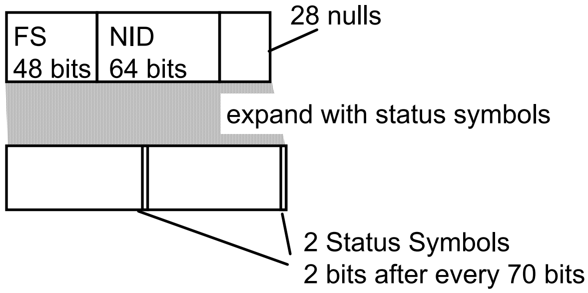

Figure 8-5 Simple Terminator Data Unit

The Common Air Interface also provides a means of sending a Link Control word at the end of a voice message. This is an expanded terminating data unit diagrammed in Figure 8-6. The Link Control word is protected with the Golay inner code instead of the Hamming inner code that is used in the Link Data Unit for voice. This is explained in Section 5.5.1. The use of the Terminator Data Units may be found in the specifications for trunked and conventional channel operation. These may be found in reference [1].

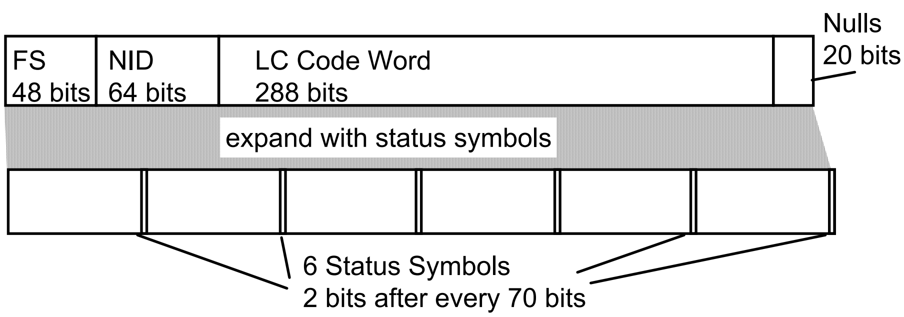

Figure 8-6 Terminator Data Unit with Link Control

8.2.4 Packet Data Unit

Packets of data are sent as variable length data units. The length of the data packet is enclosed in the header block of the data packet. Trailing nulls are sent at the end of the data packet to fill out to the next status symbol.

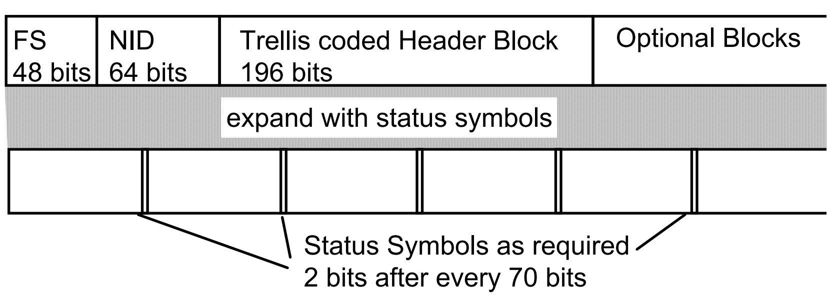

Figure 8-7 Packet Data Unit

8.3 Frame Sync Word

The sync word consists of the bit sequence listed in Table 8-1. This is transmitted in the order from left to right as shown. Only 24 bits are listed in the first vector, while the sync signal actually consists of 48 bits. Each bit is consequently expanded to represent two bits which are then encoded into a single di-bit symbol. The expanded sequence is listed in the second vector.

Table 8-1 Frame Sync Word Sequence

```
transmitted first                          transmitted last
   ↓                                                    ↓
   0000     0100     1100     1111     0101     1111
     0        4        C        F        5        F
   where "1" = di-bit (11)  and "0" = di-bit (01)
   The expanded vector is:
   01010101  01110101  11110101  11111111  01110111  11111111
     5   5     7   5     F   5     F   F     7   7     F   F
```

8.4 Status Symbols

Status symbol codes are given in Table 8-2. The Usage column indicates radio transmitters that may transmit the corresponding Status Symbol value. Subscribers shall not transmit Status Symbol values 01 or 11. Subscribers may determine the presence of infrastructure transmissions by the periodic presence of those status symbol values (01 or 11) interleaved with the Unknown Status Symbol value of 10. Subscribers may gate an AFC function with the detection of infrastructure transmissions so as to obtain nearly the same reference oscillator stability as provided by the infrastructure.

Table 8-2 Status Symbol Codes

Status Symbol	Meaning	Usage
01	Inbound Channel is Busy	Repeater
00	Unknown, use for talk-around	Subscriber
10	Unknown, use for inbound or outbound	Repeater or subscriber
11	Inbound Channel is Idle	Repeater

8.5 Network Identifier

The Network Identifier encodes 16 bits of information. The 16 bits of information are separated into a 12-bit Network Access Code and a 4-bit Data Unit ID as shown in Table 8-3. This information is protected with a (63,16,23) primitive BCH code. A single parity bit is appended to the end to round out the NID code word to 64 bits.

Table 8-3 Network Identifier Information

↓ transmitted first — transmitted last ↓

Network Access Code												Data Unit ID			
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	S3	S2	S1	S0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

8.5.1 Data Unit ID

The codes for the 6 different data units for this document are given in Table 8-4. The P bit is the last (64-th) parity bit in the code word. The remaining 10 values not given in Table 8-4 are reserved for use in trunking or other systems. The reader is referred to reference [7].

Table 8-4 Data Unit Identifier Values

Data Unit ID	P	Data Unit Usage
%0000	0	Header Data Unit
%0011	0	Terminator without subsequent Link Control
%0101	1	Logical Link Data Unit 1
%1010	1	Logical Link Data Unit 2
%1100	0	Packet Data Unit
%1111	0	Terminator with subsequent Link Control

8.5.2 NID Code

The (63,16,23) BCH code used for the NID is generated using the extension Galois Field as for the Reed-Solomon codes used for voice information. The generator polynomial for the code is of 47-th degree with 27 non-zero terms. It is given below in octal notation using the same conventions described for the RS code.

g(x) = 6331 1413 6723 5453 in octal

The generator matrix for the full code, after adding the 64-th bit, is given below in octal notation. This is a systematic code, so the left hand 16 columns form an identity matrix and the right hand 48 columns form a parity check matrix.

row	16 x 16 Identity		16 x 48 Parity Check			
1	10	0000	6331	1413	6723	5452
2	04	0000	5265	5216	1472	3276
3	02	0000	4603	7114	6116	4164
4	01	0000	2301	7446	3047	2072
5	00	4000	7271	6230	7300	0466
6	00	2000	5605	6507	5263	5660
7	00	1000	2702	7243	6531	6730
8	00	0400	1341	3521	7254	7354
9	00	0200	0560	5650	7526	3566
10	00	0100	6141	3337	5170	4220
11	00	0040	3060	5557	6474	2110
12	00	0020	1430	2667	7236	1044
13	00	0010	0614	1333	7517	0422
14	00	0004	6037	1146	1164	1642
15	00	0002	5326	5070	6351	5373
16	00	0001	4662	3027	5647	3127

9 Modulation

9.1 General Description

The modulation is a form of differential Quadrature Phase Shift Keying (QPSK) where each successive symbol is shifted in phase from its predecessor by 45 degrees (π/4 radians). The receiver of this modulation is intended to be compatible with any transmitter which accomplishes this phase shift of the carrier, of which at least two types are available. A transmitter which modulates the phase but keeps the amplitude of the carrier constant will generate a constant envelope frequency modulated waveform which will be denoted C4FM. A transmitter which modulates the phase and simultaneously modulates the carrier amplitude to minimize the width of the emitted spectrum will generate an amplitude modulated waveform which will be denoted CQPSK.

Other forms of differential Quadrature Phase Shift Keying modulation (for example π/4 DQPSK) will be studied for possible inclusion in this document.

9.2 Symbols

The modulation sends 4800 symbols/sec with each symbol conveying 2 bits of information. The mapping between symbols and bits is given in Table 9-1.

Table 9-1 Dibit Symbol Mapping to Modulation Phase or Deviation

Information Bits	Symbol	CQPSK Phase Change	C4FM Deviation
01	+3	+135 degrees	+1.80 kHz
00	+1	+45 degrees	+0.60 kHz
10	-1	-45 degrees	-0.60 kHz
11	-3	-135 degrees	-1.80 kHz

9.3 Nyquist Raised Cosine Filter

The modulation symbols are filtered with a Raised Cosine Filter which satisfies the Nyquist criterion minimizing intersymbol interference. The input to this filter consists of a series of impulses, scaled according to paragraph 9.2, and separated in time by 208.33 microseconds (1/4800 sec). The group delay of the filter is flat over the passband for |f| < 2880 Hz. The magnitude response of the filter is given approximately by the following formula.

f = frequency in hertz

$|H(f)|$ = magnitude response of the Nyquist Raised Cosine Filter

$$|H(f)| = 1 \quad \text{for } |f| < 1920 \text{ Hz}$$

$$|H(f)| = 0.5 + 0.5 \cos(2 \pi f / 1920) \quad \text{for } 1920 \text{ Hz} < |f| < 2880 \text{ Hz}$$

$$|H(f)| = 0 \quad \text{for } |f| > 2880 \text{ Hz}$$

9.4 C4FM Modulator

The C4FM modulator consists of a Nyquist Raised Cosine Filter, cascaded with a Shaping Filter, cascaded with a frequency modulator. The Nyquist Raised Cosine Filter is described in paragraph 9.3. The Shaping Filter has a flat group delay over the passband for |f| < 2880 Hz. The magnitude response of the

Shaping Filter is given approximately by the following formula. The response of the filter above 2880 Hz is not specified because the filter H(f) should cut off above 2880 Hz.

$|P(f)|$ = magnitude response of the Shaping Filter

$$|P(f)| = (\pi f/4800) / \sin(\pi f/4800) \quad \text{for } |f| < 2880 \text{ Hz}$$

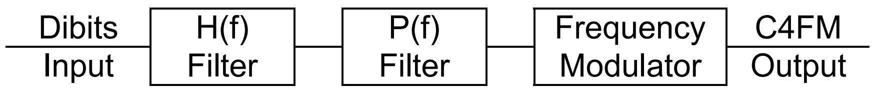

Figure 9-1 C4FM Modulator

9.4.1 C4FM Deviation

The C4FM modulator must have the deviation set to provide the proper carrier phase shift for each modulated symbol. The deviation is set with a test signal consisting of the following symbol stream.

```
... 01 01 11 11 01 01 11 11 ...
```

This test signal is processed by the modulator to create a C4FM signal equivalent to a 1.2 kHz sine wave modulating an FM signal with a peak deviation equal to: $\pi/2$ x 1800 Hz = 2827 Hz. The method of measurement for this test signal and the tolerance on the deviation are specified in references [4] and [5], respectively.

9.5 CQPSK Modulator

The CQPSK modulator consists of In Phase and Quadrature Phase (I and Q) amplitude modulators which modulate two carriers with the Q phase delayed from the I phase by 90 degrees. The I and Q modulators are driven by the filtered output of a 5-level signal which is derived from the Information Bits by the lookup table given in Table 9-2. The lookup table uses a state variable which represents the current phase of the carrier, ranging from 0 to represent 0 degrees to 7 to represent 315 degrees.

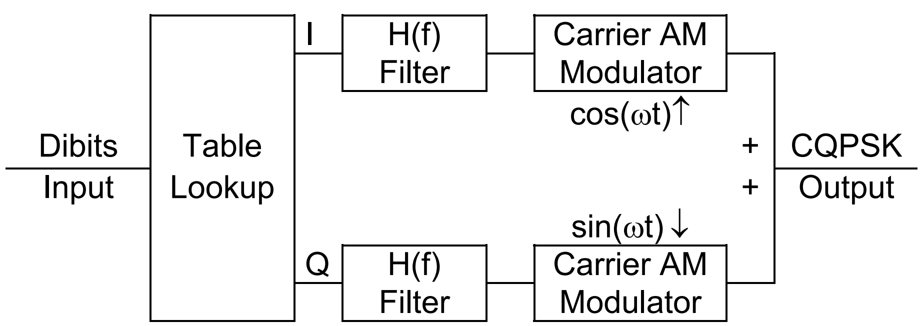

Figure 9-2 CQPSK Modulator

Table 9-2 CQPSK I and Q Lookup Table

Current Phase State	Information Bits	Next Phase State	I Level	Q Level
0	00	1	0.7071	0.7071
0	01	3	-0.7071	0.7071
0	11	5	-0.7071	-0.7071
0	10	7	0.7071	-0.7071
1	00	2	0.0	1.0
1	01	4	-1.0	0.0
1	11	6	0.0	-1.0
1	10	0	1.0	0.0
2	00	3	-0.7071	0.7071
2	01	5	-0.7071	-0.7071
2	11	7	0.7071	-0.7071
2	10	1	0.7071	0.7071
3	00	4	-1.0	0.0
3	01	6	0.0	-1.0
3	11	0	1.0	0.0
3	10	2	0.0	1.0
4	00	5	-0.7071	-0.7071
4	01	7	0.7071	-0.7071
4	11	1	0.7071	0.7071
4	10	3	-0.7071	0.7071
5	00	6	0.0	-1.0
5	01	0	1.0	0.0
5	11	2	0.0	1.0
5	10	4	-1.0	0.0
6	00	7	0.7071	-0.7071
6	01	1	0.7071	0.7071
6	11	3	-0.7071	0.7071
6	10	5	-0.7071	-0.7071
7	00	0	1.0	0.0
7	01	2	0.0	1.0
7	11	4	-1.0	0.0
7	10	6	0.0	-1.0

The Information Bits are processed by the lookup table to yield a 5-level I signal and a 5-level Q signal. The I and Q signals are filtered with the Nyquist Raised Cosine Filter described in paragraph 9.3. The I signal is then multiplied by the carrier and the Q signal is multiplied by the carrier after it has been delayed by 90 degrees. The modulated I and Q carriers are then summed together to yield the modulator output.

9.6 QPSK Demodulator

The QPSK demodulator is capable of receiving a signal from either the C4FM modulator or the CQPSK modulator. It consists of a frequency modulation detector, followed by an Integrate and Dump Filter and then a stochastic gradient clock recovery device. The frequency modulation detector can be a standard discriminator output provided by an FM receiver. The deviation sensitivity of the discriminator must be fixed within 10%.

The Integrate and Dump Filter has a flat group delay over the passband for $|f| < 2880$ Hz. The magnitude response of the Integrate and Dump Filter is given approximately by the following formula. Only the response for $|f| < 2880$ Hz is specified in this document.

```
      f = frequency in hertz
  |D(f)| = magnitude response of the Integrate and Dump Filter
  |D(f)| = sin(π f/4800) / (π f/4800)
```

The stochastic gradient clock recovery device provides clock recovery for the waveform at the output of the Integrate and Dump Filter. Several techniques are available for this process. The details of the clock recovery are not specified by this document.

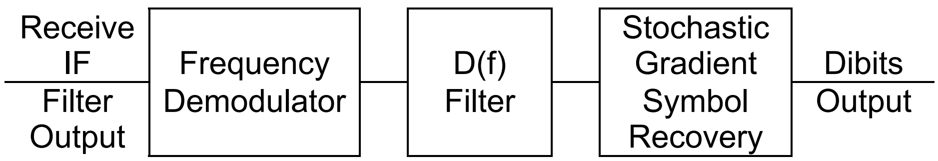

Figure 9-3 QPSK Demodulator

10 Annex for Transmit Bit Order

This normative annex lists all of the bits that are transmitted for voice messages in consecutive order as they shall be transmitted according to this document.

10.1 Conventions

There are a large number of bits that are transmitted for voice. Some simple conventions have been adopted in naming and numbering them so that the bits can be understood in the context of the *Common Air Interface*.

First of all, the transmitted message consists of a series of data units, namely a

Header Data Unit, followed by a
Logical Link Data Unit 1 (LDU 1), followed by a
Logical Link Data Unit 2 (LDU 2), followed by
additional LDU 1 and LDU 2 units until the end of the message, then a
Terminator Data Unit is sent.

There are two kinds of terminator data units, and either one can be sent. One contains a Link Control (LC) word, and one does not. The Terminator can follow either LDU 1 or LDU 2. A description and picture of the data units are given in the *Common Air Interface*.

The transmitter modulation consists of a sequence of dibit symbols that are serially transmitted. Each dibit consists of 2 bits of information as discussed in the *Common Air Interface*. The tables for the data units consist of a sequence of dibit symbols, starting with symbol 0 and numbered in the order of transmission. The first symbol transmitted shall be symbol 0.

The transmitted signal consists of a sequence of fields of information. Each field is in turn decomposed into bits. For example, the Network Access Code, or `NAC`, consists of 12 bits which are numbered 11, 10, 9, ... 1, 0. The least significant bit is always numbered 0 in a field. Generally, the least significant bit is always transmitted last. The least significant bit is always portrayed as the right-most bit in the *Common Air Interface* description. The number of the bit is always enclosed in parenthesis, for example `NAC(11)` refers to the most significant bit of the Network Access Code.

After most of the information fields there is a parity check field for the error correcting code. The name of the code is always used to denote the parity check field. For example, a BCH code is used to protect the Network Identifier, so the parity check field is named `BCH_parity(x)` where `x` varies from 47 down to 0. Bit 0 is always the least significant bit of the parity check field, and is always portrayed as the right-most bit in the *Common Air Interface*.

The following information field names are used in the tables.

NAME	MEANING
NAC	Network Access Code, part of the Network Identifier
DUID	Data Unit Identifier, part of the Network Identifier
LC_format	Link Control Format, part of the Link Control word
MFID	Manufacturer's Identifier, part of the Link Control word
LC_information	remaining information in the Link Control word
LSD_info	Low Speed Data information, in octets
TGID	Talk Group ID, part of the Header word
MI	Message Indicator for encryption synchronization
ALGID	Algorithm ID for encryption algorithm
KID	Key ID for the encryption key
SS	Status Symbol, also known as busy bits
c_x	code word 'x' for an IMBE voice frame
BCH_parity	(64,16,23) BCH parity checks for the Network ID
RS_parity	Reed-Solomon parity check hex bit
Short_Hamm_parity	(10,6,3) Shortened Hamming parity check
Short_Golay_parity	(18,6,8) Shortened Golay parity check
Extend_Golay_parity	(24,12,8) Extended Golay parity check
cyclic_parity	(16,8,5) shortened cyclic code parity check

In some cases an extension digit is used to follow the field name. This occurs for the IMBE code words, as in c_0, c_1, ... c_7, as well as the Reed-Solomon parity checks as in RS_parity_11. In the case of the IMBE code words, the extension digit is defined in the *Common Air Interface*. In the case of the Reed-Solomon parity checks, the extension digit represents the index of the parity check hex bit. As in the case of the binary codes, the hex bit 0 represents the right-most or least significant hex bit in the Reed-Solomon code. It also represents the 0-th degree polynomial coefficient when the Reed-Solomon code word is represented as a polynomial.

10.2 Header Data Unit

The next few pages give the order in which symbols shall be transmitted for the Header Data Unit. The Header Data Unit is transmitted at the beginning of a voice message.

Symbol	Field	Bit 1	Bit 0
0	Frame	0	1
1	Sync	0	1
2		0	1
3		0	1
4		0	1
5		1	1
6		0	1
7		0	1
8		1	1
9		1	1
10		0	1
11		0	1
12		1	1
13		1	1
14		1	1
15		1	1
16		0	1
17		1	1
18		0	1
19		1	1
20		1	1
21		1	1
22		1	1
23		1	1
24	Network	NAC(11)	NAC(10)
25	ID	NAC(9)	NAC(8)
26		NAC(7)	NAC(6)
27		NAC(5)	NAC(4)
28		NAC(3)	NAC(2)
29		NAC(1)	NAC(0)
30		DUID(3)	DUID(2)
31		DUID(1)	DUID(0)
32		BCH_parity(47)	BCH_parity(46)
33		BCH_parity(45)	BCH_parity(44)
34		BCH_parity(43)	BCH_parity(42)
35	Status 1	SS(1)	SS(0)
36		BCH_parity(41)	BCH_parity(40)
37		BCH_parity(39)	BCH_parity(38)
38		BCH_parity(37)	BCH_parity(36)
39		BCH_parity(35)	BCH_parity(34)
40		BCH_parity(33)	BCH_parity(32)
41		BCH_parity(31)	BCH_parity(30)
42		BCH_parity(29)	BCH_parity(28)
43		BCH_parity(27)	BCH_parity(26)
44		BCH_parity(25)	BCH_parity(24)
45		BCH_parity(23)	BCH_parity(22)
46		BCH_parity(21)	BCH_parity(20)
47		BCH_parity(19)	BCH_parity(18)
48		BCH_parity(17)	BCH_parity(16)
49		BCH_parity(15)	BCH_parity(14)
50		BCH_parity(13)	BCH_parity(12)
51		BCH_parity(11)	BCH_parity(10)
52		BCH_parity(9)	BCH_parity(8)
53		BCH_parity(7)	BCH_parity(6)
54		BCH_parity(5)	BCH_parity(4)
55		BCH_parity(3)	BCH_parity(2)
56		BCH_parity(1)	BCH_parity(0)
57	Header	MI(71)	MI(70)
58	Code	MI(69)	MI(68)
59	Word	MI(67)	MI(66)
60		Short_Golay_parity(11)	Short_Golay_parity(10)

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

133		MI(19)	MI(18)
134		Short_Golay_parity(11)	Short_Golay_parity(10)
135		Short_Golay_parity(9)	Short_Golay_parity(8)
136		Short_Golay_parity(7)	Short_Golay_parity(6)
137		Short_Golay_parity(5)	Short_Golay_parity(4)
138		Short_Golay_parity(3)	Short_Golay_parity(2)
139		Short_Golay_parity(1)	Short_Golay_parity(0)
140		MI(17)	MI(16)
141		MI(15)	MI(14)
142		MI(13)	MI(12)
143	Status 4	SS(1)	SS(0)
144		Short_Golay_parity(11)	Short_Golay_parity(10)
145		Short_Golay_parity(9)	Short_Golay_parity(8)
146		Short_Golay_parity(7)	Short_Golay_parity(6)
147		Short_Golay_parity(5)	Short_Golay_parity(4)
148		Short_Golay_parity(3)	Short_Golay_parity(2)
149		Short_Golay_parity(1)	Short_Golay_parity(0)
150		MI(11)	MI(10)
151		MI(9)	MI(8)
152		MI(7)	MI(6)
153		Short_Golay_parity(11)	Short_Golay_parity(10)
154		Short_Golay_parity(9)	Short_Golay_parity(8)
155		Short_Golay_parity(7)	Short_Golay_parity(6)
156		Short_Golay_parity(5)	Short_Golay_parity(4)
157		Short_Golay_parity(3)	Short_Golay_parity(2)
158		Short_Golay_parity(1)	Short_Golay_parity(0)
159		MI(5)	MI(4)
160		MI(3)	MI(2)
161		MI(1)	MI(0)
162		Short_Golay_parity(11)	Short_Golay_parity(10)
163		Short_Golay_parity(9)	Short_Golay_parity(8)
164		Short_Golay_parity(7)	Short_Golay_parity(6)
165		Short_Golay_parity(5)	Short_Golay_parity(4)
166		Short_Golay_parity(3)	Short_Golay_parity(2)
167		Short_Golay_parity(1)	Short_Golay_parity(0)
168		MFID(7)	MFID(6)
169		MFID(5)	MFID(4)
170		MFID(3)	MFID(2)
171		Short_Golay_parity(11)	Short_Golay_parity(10)
172		Short_Golay_parity(9)	Short_Golay_parity(8)
173		Short_Golay_parity(7)	Short_Golay_parity(6)
174		Short_Golay_parity(5)	Short_Golay_parity(4)
175		Short_Golay_parity(3)	Short_Golay_parity(2)
176		Short_Golay_parity(1)	Short_Golay_parity(0)
177		MFID(1)	MFID(0)
178		ALGID(7)	ALGID(6)
179	Status 5	SS(1)	SS(0)
180		ALGID(5)	ALGID(4)
181		Short_Golay_parity(11)	Short_Golay_parity(10)
182		Short_Golay_parity(9)	Short_Golay_parity(8)
183		Short_Golay_parity(7)	Short_Golay_parity(6)
184		Short_Golay_parity(5)	Short_Golay_parity(4)
185		Short_Golay_parity(3)	Short_Golay_parity(2)
186		Short_Golay_parity(1)	Short_Golay_parity(0)
187		ALGID(3)	ALGID(2)
188		ALGID(1)	ALGID(0)
189		KID(15)	KID(14)
190		Short_Golay_parity(11)	Short_Golay_parity(10)
191		Short_Golay_parity(9)	Short_Golay_parity(8)
192		Short_Golay_parity(7)	Short_Golay_parity(6)
193		Short_Golay_parity(5)	Short_Golay_parity(4)
194		Short_Golay_parity(3)	Short_Golay_parity(2)
195		Short_Golay_parity(1)	Short_Golay_parity(0)
196		KID(13)	KID(12)
197		KID(11)	KID(10)
198		KID(9)	KID(8)
199		Short_Golay_parity(11)	Short_Golay_parity(10)
200		Short_Golay_parity(9)	Short_Golay_parity(8)
201		Short_Golay_parity(7)	Short_Golay_parity(6)
202		Short_Golay_parity(5)	Short_Golay_parity(4)
203		Short_Golay_parity(3)	Short_Golay_parity(2)
204		Short_Golay_parity(1)	Short_Golay_parity(0)

205		KID(7)	KID(6)
206		KID(5)	KID(4)
207		KID(3)	KID(2)
208		Short_Golay_parity(11)	Short_Golay_parity(10)
209		Short_Golay_parity(9)	Short_Golay_parity(8)
210		Short_Golay_parity(7)	Short_Golay_parity(6)
211		Short_Golay_parity(5)	Short_Golay_parity(4)
212		Short_Golay_parity(3)	Short_Golay_parity(2)
213		Short_Golay_parity(1)	Short_Golay_parity(0)
214		KID(1)	KID(0)
215	Status 6	SS(1)	SS(0)
216		TGID(15)	TGID(14)
217		TGID(13)	TGID(12)
218		Short_Golay_parity(11)	Short_Golay_parity(10)
219		Short_Golay_parity(9)	Short_Golay_parity(8)
220		Short_Golay_parity(7)	Short_Golay_parity(6)
221		Short_Golay_parity(5)	Short_Golay_parity(4)
222		Short_Golay_parity(3)	Short_Golay_parity(2)
223		Short_Golay_parity(1)	Short_Golay_parity(0)
224		TGID(11)	TGID(10)
225		TGID(9)	TGID(8)
226		TGID(7)	TGID(6)
227		Short_Golay_parity(11)	Short_Golay_parity(10)
228		Short_Golay_parity(9)	Short_Golay_parity(8)
229		Short_Golay_parity(7)	Short_Golay_parity(6)
230		Short_Golay_parity(5)	Short_Golay_parity(4)
231		Short_Golay_parity(3)	Short_Golay_parity(2)
232		Short_Golay_parity(1)	Short_Golay_parity(0)
233		TGID(5)	TGID(4)
234		TGID(3)	TGID(2)
235		TGID(1)	TGID(0)
236		Short_Golay_parity(11)	Short_Golay_parity(10)
237		Short_Golay_parity(9)	Short_Golay_parity(8)
238		Short_Golay_parity(7)	Short_Golay_parity(6)
239		Short_Golay_parity(5)	Short_Golay_parity(4)
240		Short_Golay_parity(3)	Short_Golay_parity(2)
241		Short_Golay_parity(1)	Short_Golay_parity(0)
242		RS_parity_15(5)	RS_parity_15(4)
243		RS_parity_15(3)	RS_parity_15(2)
244		RS_parity_15(1)	RS_parity_15(0)
245		Short_Golay_parity(11)	Short_Golay_parity(10)
246		Short_Golay_parity(9)	Short_Golay_parity(8)
247		Short_Golay_parity(7)	Short_Golay_parity(6)
248		Short_Golay_parity(5)	Short_Golay_parity(4)
249		Short_Golay_parity(3)	Short_Golay_parity(2)
250		Short_Golay_parity(1)	Short_Golay_parity(0)
251	Status 7	SS(1)	SS(0)
252		RS_parity_14(5)	RS_parity_14(4)
253		RS_parity_14(3)	RS_parity_14(2)
254		RS_parity_14(1)	RS_parity_14(0)
255		Short_Golay_parity(11)	Short_Golay_parity(10)
256		Short_Golay_parity(9)	Short_Golay_parity(8)
257		Short_Golay_parity(7)	Short_Golay_parity(6)
258		Short_Golay_parity(5)	Short_Golay_parity(4)
259		Short_Golay_parity(3)	Short_Golay_parity(2)
260		Short_Golay_parity(1)	Short_Golay_parity(0)
261		RS_parity_13(5)	RS_parity_13(4)
262		RS_parity_13(3)	RS_parity_13(2)
263		RS_parity_13(1)	RS_parity_13(0)
264		Short_Golay_parity(11)	Short_Golay_parity(10)
265		Short_Golay_parity(9)	Short_Golay_parity(8)
266		Short_Golay_parity(7)	Short_Golay_parity(6)
267		Short_Golay_parity(5)	Short_Golay_parity(4)
268		Short_Golay_parity(3)	Short_Golay_parity(2)
269		Short_Golay_parity(1)	Short_Golay_parity(0)
270		RS_parity_12(5)	RS_parity_12(4)
271		RS_parity_12(3)	RS_parity_12(2)
272		RS_parity_12(1)	RS_parity_12(0)
273		Short_Golay_parity(11)	Short_Golay_parity(10)
274		Short_Golay_parity(9)	Short_Golay_parity(8)
275		Short_Golay_parity(7)	Short_Golay_parity(6)
276		Short_Golay_parity(5)	Short_Golay_parity(4)

277		Short_Golay_parity(3)	Short_Golay_parity(2)	
278		Short_Golay_parity(1)	Short_Golay_parity(0)	
279		RS_parity_11(5)	RS_parity_11(4)	
280		RS_parity_11(3)	RS_parity_11(2)	
281		RS_parity_11(1)	RS_parity_11(0)	
282		Short_Golay_parity(11)	Short_Golay_parity(10)	
283		Short_Golay_parity(9)	Short_Golay_parity(8)	
284		Short_Golay_parity(7)	Short_Golay_parity(6)	
285		Short_Golay_parity(5)	Short_Golay_parity(4)	
286		Short_Golay_parity(3)	Short_Golay_parity(2)	
287	Status 8	SS(1)	SS(0)	
288		Short_Golay_parity(1)	Short_Golay_parity(0)	
289		RS_parity_10(5)	RS_parity_10(4)	
290		RS_parity_10(3)	RS_parity_10(2)	
291		RS_parity_10(1)	RS_parity_10(0)	
292		Short_Golay_parity(11)	Short_Golay_parity(10)	
293		Short_Golay_parity(9)	Short_Golay_parity(8)	
294		Short_Golay_parity(7)	Short_Golay_parity(6)	
295		Short_Golay_parity(5)	Short_Golay_parity(4)	
296		Short_Golay_parity(3)	Short_Golay_parity(2)	
297		Short_Golay_parity(1)	Short_Golay_parity(0)	
298		RS_parity_9(5)	RS_parity_9(4)	
299		RS_parity_9(3)	RS_parity_9(2)	
300		RS_parity_9(1)	RS_parity_9(0)	
301		Short_Golay_parity(11)	Short_Golay_parity(10)	
302		Short_Golay_parity(9)	Short_Golay_parity(8)	
303		Short_Golay_parity(7)	Short_Golay_parity(6)	
304		Short_Golay_parity(5)	Short_Golay_parity(4)	
305		Short_Golay_parity(3)	Short_Golay_parity(2)	
306		Short_Golay_parity(1)	Short_Golay_parity(0)	
307		RS_parity_8(5)	RS_parity_8(4)	
308		RS_parity_8(3)	RS_parity_8(2)	
309		RS_parity_8(1)	RS_parity_8(0)	
310		Short_Golay_parity(11)	Short_Golay_parity(10)	
311		Short_Golay_parity(9)	Short_Golay_parity(8)	
312		Short_Golay_parity(7)	Short_Golay_parity(6)	
313		Short_Golay_parity(5)	Short_Golay_parity(4)	
314		Short_Golay_parity(3)	Short_Golay_parity(2)	
315		Short_Golay_parity(1)	Short_Golay_parity(0)	
316		RS_parity_7(5)	RS_parity_7(4)	
317		RS_parity_7(3)	RS_parity_7(2)	
318		RS_parity_7(1)	RS_parity_7(0)	
319		Short_Golay_parity(11)	Short_Golay_parity(10)	
320		Short_Golay_parity(9)	Short_Golay_parity(8)	
321		Short_Golay_parity(7)	Short_Golay_parity(6)	
322		Short_Golay_parity(5)	Short_Golay_parity(4)	
323	Status 9		SS(1)	SS(0)
324		Short_Golay_parity(3)	Short_Golay_parity(2)	
325		Short_Golay_parity(1)	Short_Golay_parity(0)	
326		RS_parity_6(5)	RS_parity_6(4)	
327		RS_parity_6(3)	RS_parity_6(2)	
328		RS_parity_6(1)	RS_parity_6(0)	
329		Short_Golay_parity(11)	Short_Golay_parity(10)	
330		Short_Golay_parity(9)	Short_Golay_parity(8)	
331		Short_Golay_parity(7)	Short_Golay_parity(6)	
332		Short_Golay_parity(5)	Short_Golay_parity(4)	
333		Short_Golay_parity(3)	Short_Golay_parity(2)	
334		Short_Golay_parity(1)	Short_Golay_parity(0)	
335		RS_parity_5(5)	RS_parity_5(4)	
336		RS_parity_5(3)	RS_parity_5(2)	
337		RS_parity_5(1)	RS_parity_5(0)	
338		Short_Golay_parity(11)	Short_Golay_parity(10)	
339		Short_Golay_parity(9)	Short_Golay_parity(8)	
340		Short_Golay_parity(7)	Short_Golay_parity(6)	
341		Short_Golay_parity(5)	Short_Golay_parity(4)	
342		Short_Golay_parity(3)	Short_Golay_parity(2)	
343		Short_Golay_parity(1)	Short_Golay_parity(0)	
344		RS_parity_4(5)	RS_parity_4(4)	
345		RS_parity_4(3)	RS_parity_4(2)	
346		RS_parity_4(1)	RS_parity_4(0)	
347		Short_Golay_parity(11)	Short_Golay_parity(10)	
348		Short_Golay_parity(9)	Short_Golay_parity(8)	

349		Short_Golay_parity(7)	Short_Golay_parity(6)
350		Short_Golay_parity(5)	Short_Golay_parity(4)
351		Short_Golay_parity(3)	Short_Golay_parity(2)
352		Short_Golay_parity(1)	Short_Golay_parity(0)
353		RS_parity_3(5)	RS_parity_3(4)
354		RS_parity_3(3)	RS_parity_3(2)
355		RS_parity_3(1)	RS_parity_3(0)
356		Short_Golay_parity(11)	Short_Golay_parity(10)
357		Short_Golay_parity(9)	Short_Golay_parity(8)
358		Short_Golay_parity(7)	Short_Golay_parity(6)
359	Status 10	SS(1)	SS(0)
360		Short_Golay_parity(5)	Short_Golay_parity(4)
361		Short_Golay_parity(3)	Short_Golay_parity(2)
362		Short_Golay_parity(1)	Short_Golay_parity(0)
363		RS_parity_2(5)	RS_parity_2(4)
364		RS_parity_2(3)	RS_parity_2(2)
365		RS_parity_2(1)	RS_parity_2(0)
366		Short_Golay_parity(11)	Short_Golay_parity(10)
367		Short_Golay_parity(9)	Short_Golay_parity(8)
368		Short_Golay_parity(7)	Short_Golay_parity(6)
369		Short_Golay_parity(5)	Short_Golay_parity(4)
370		Short_Golay_parity(3)	Short_Golay_parity(2)
371		Short_Golay_parity(1)	Short_Golay_parity(0)
372		RS_parity_1(5)	RS_parity_1(4)
373		RS_parity_1(3)	RS_parity_1(2)
374		RS_parity_1(1)	RS_parity_1(0)
375		Short_Golay_parity(11)	Short_Golay_parity(10)
376		Short_Golay_parity(9)	Short_Golay_parity(8)
377		Short_Golay_parity(7)	Short_Golay_parity(6)
378		Short_Golay_parity(5)	Short_Golay_parity(4)
379		Short_Golay_parity(3)	Short_Golay_parity(2)
380		Short_Golay_parity(1)	Short_Golay_parity(0)
381		RS_parity_0(5)	RS_parity_0(4)
382		RS_parity_0(3)	RS_parity_0(2)
383		RS_parity_0(1)	RS_parity_0(0)
384		Short_Golay_parity(11)	Short_Golay_parity(10)
385		Short_Golay_parity(9)	Short_Golay_parity(8)
386		Short_Golay_parity(7)	Short_Golay_parity(6)
387		Short_Golay_parity(5)	Short_Golay_parity(4)
388		Short_Golay_parity(3)	Short_Golay_parity(2)
389		Short_Golay_parity(1)	Short_Golay_parity(0)
390	Nulls	0	0
391		0	0
392		0	0
393		0	0
394		0	0
395	Status 11	SS(1)	SS(0)

10.3 Logical Link Data Unit 1

The next several pages give the order in which symbols shall be transmitted for the first Logical Link Data Unit for voice. This data unit spans a time interval of 180 ms. It sends 9 frames of IMBE information.

Symbol	Field	Bit 1	Bit 0
0	Frame	0	1
1	Sync	0	1
2		0	1
3		0	1
4		0	1
5		1	1
6		0	1
7		0	1
8		1	1
9		1	1
10		0	1
11		0	1
12		1	1
13		1	1
14		1	1
15		1	1
16		0	1
17		1	1
18		0	1
19		1	1
20		1	1
21		1	1
22		1	1
23		1	1
24	Network	NAC(11)	NAC(10)
25	ID	NAC(9)	NAC(8)
26		NAC(7)	NAC(6)
27		NAC(5)	NAC(4)
28		NAC(3)	NAC(2)
29		NAC(1)	NAC(0)
30		DUID(3)	DUID(2)
31		DUID(1)	DUID(0)
32		BCH_parity(47)	BCH_parity(46)
33		BCH_parity(45)	BCH_parity(44)
34		BCH_parity(43)	BCH_parity(42)
35	Status 1	SS(1)	SS(0)
36		BCH_parity(41)	BCH_parity(40)
37		BCH_parity(39)	BCH_parity(38)
38		BCH_parity(37)	BCH_parity(36)
39		BCH_parity(35)	BCH_parity(34)
40		BCH_parity(33)	BCH_parity(32)
41		BCH_parity(31)	BCH_parity(30)
42		BCH_parity(29)	BCH_parity(28)
43		BCH_parity(27)	BCH_parity(26)
44		BCH_parity(25)	BCH_parity(24)
45		BCH_parity(23)	BCH_parity(22)
46		BCH_parity(21)	BCH_parity(20)
47		BCH_parity(19)	BCH_parity(18)
48		BCH_parity(17)	BCH_parity(16)
49		BCH_parity(15)	BCH_parity(14)
50		BCH_parity(13)	BCH_parity(12)
51		BCH_parity(11)	BCH_parity(10)
52		BCH_parity(9)	BCH_parity(8)
53		BCH_parity(7)	BCH_parity(6)
54		BCH_parity(5)	BCH_parity(4)
55		BCH_parity(3)	BCH_parity(2)
56		BCH_parity(1)	BCH_parity(0)
57	IMBE 1	c_0(22)	c_1(21)
58		c_2(20)	c_3(19)
59		c_4(10)	c_5(1)
60		c_1(20)	c_0(21)

61		c_3(18)	c_2(19)
62		c_5(0)	c_4(9)
63		c_0(20)	c_1(19)
64		c_2(18)	c_3(17)
65		c_4(8)	c_6(14)
66		c_1(18)	c_0(19)
67		c_3(16)	c_2(17)
68		c_6(13)	c_4(7)
69		c_0(18)	c_1(17)
70		c_2(16)	c_3(15)
71	Status 2	SS(1)	SS(0)
72		c_4(6)	c_6(12)
73		c_1(16)	c_0(17)
74		c_3(14)	c_2(15)
75		c_6(11)	c_4(5)
76		c_0(16)	c_1(15)
77		c_2(14)	c_3(13)
78		c_4(4)	c_6(10)
79		c_1(14)	c_0(15)
80		c_3(12)	c_2(13)
81		c_6(9)	c_4(3)
82		c_0(14)	c_1(13)
83		c_2(12)	c_3(11)
84		c_4(2)	c_6(8)
85		c_1(12)	c_0(13)
86		c_3(10)	c_2(11)
87		c_6(7)	c_4(1)
88		c_0(12)	c_1(11)
89		c_2(10)	c_3(9)
90		c_4(0)	c_6(6)
91		c_1(10)	c_0(11)
92		c_3(8)	c_2(9)
93		c_6(5)	c_5(14)
94		c_0(10)	c_1(9)
95		c_2(8)	c_3(7)
96		c_5(13)	c_6(4)
97		c_1(8)	c_0(9)
98		c_3(6)	c_2(7)
99		c_6(3)	c_5(12)
100		c_0(8)	c_1(7)
101		c_2(6)	c_3(5)
102		c_5(11)	c_6(2)
103		c_1(6)	c_0(7)
104		c_3(4)	c_2(5)
105		c_6(1)	c_5(10)
106		c_0(6)	c_1(5)
107	Status 3	SS(1)	SS(0)
108		c_2(4)	c_3(3)
109		c_5(9)	c_6(0)
110		c_1(4)	c_0(5)
111		c_3(2)	c_2(3)
112		c_7(6)	c_5(8)
113		c_0(4)	c_1(3)
114		c_2(2)	c_3(1)
115		c_5(7)	c_7(5)
116		c_1(2)	c_0(3)
117		c_3(0)	c_2(1)
118		c_7(4)	c_5(6)
119		c_0(2)	c_1(1)
120		c_2(0)	c_4(14)
121		c_5(5)	c_7(3)
122		c_1(0)	c_0(1)
123		c_4(13)	c_3(22)
124		c_7(2)	c_5(4)
125		c_0(0)	c_2(22)
126		c_3(21)	c_4(12)
127		c_5(3)	c_7(1)
128		c_2(21)	c_1(22)
129		c_4(11)	c_3(20)
130	END of 1	c_7(0)	c_5(2)
131	IMBE 2	c_0(22)	c_1(21)
132		c_2(20)	c_3(19)

133		c_4(10)	c_5(1)
134		c_1(20)	c_0(21)
135		c_3(18)	c_2(19)
136		c_5(0)	c_4(9)
137		c_0(20)	c_1(19)
138		c_2(18)	c_3(17)
139		c_4(8)	c_6(14)
140		c_1(18)	c_0(19)
141		c_3(16)	c_2(17)
142		c_6(13)	c_4(7)
143	Status 4	SS(1)	SS(0)
144		c_0(18)	c_1(17)
145		c_2(16)	c_3(15)
146		c_4(6)	c_6(12)
147		c_1(16)	c_0(17)
148		c_3(14)	c_2(15)
149		c_6(11)	c_4(5)
150		c_0(16)	c_1(15)
151		c_2(14)	c_3(13)
152		c_4(4)	c_6(10)
153		c_1(14)	c_0(15)
154		c_3(12)	c_2(13)
155		c_6(9)	c_4(3)
156		c_0(14)	c_1(13)
157		c_2(12)	c_3(11)
158		c_4(2)	c_6(8)
159		c_1(12)	c_0(13)
160		c_3(10)	c_2(11)
161		c_6(7)	c_4(1)
162		c_0(12)	c_1(11)
163		c_2(10)	c_3(9)
164		c_4(0)	c_6(6)
165		c_1(10)	c_0(11)
166		c_3(8)	c_2(9)
167		c_6(5)	c_5(14)
168		c_0(10)	c_1(9)
169		c_2(8)	c_3(7)
170		c_5(13)	c_6(4)
171		c_1(8)	c_0(9)
172		c_3(6)	c_2(7)
173		c_6(3)	c_5(12)
174		c_0(8)	c_1(7)
175		c_2(6)	c_3(5)
176		c_5(11)	c_6(2)
177		c_1(6)	c_0(7)
178		c_3(4)	c_2(5)
179	Status 5	SS(1)	SS(0)
180		c_6(1)	c_5(10)
181		c_0(6)	c_1(5)
182		c_2(4)	c_3(3)
183		c_5(9)	c_6(0)
184		c_1(4)	c_0(5)
185		c_3(2)	c_2(3)
186		c_7(6)	c_5(8)
187		c_0(4)	c_1(3)
188		c_2(2)	c_3(1)
189		c_5(7)	c_7(5)
190		c_1(2)	c_0(3)
191		c_3(0)	c_2(1)
192		c_7(4)	c_5(6)
193		c_0(2)	c_1(1)
194		c_2(0)	c_4(14)
195		c_5(5)	c_7(3)
196		c_1(0)	c_0(1)
197		c_4(13)	c_3(22)
198		c_7(2)	c_5(4)
199		c_0(0)	c_2(22)
200		c_3(21)	c_4(12)
201		c_5(3)	c_7(1)
202		c_2(21)	c_1(22)
203		c_4(11)	c_3(20)
204	END of 2	c_7(0)	c_5(2)

205		LC_format(7)	LC_format(6)
206		LC_format(5)	LC_format(4)
207		LC_format(3)	LC_format(2)
208		Short_Hamm_parity(3)	Short_Hamm_parity(2)
209		Short_Hamm_parity(1)	Short_Hamm_parity(0)
210		LC_format(1)	LC_format(0)
211		MFID(7)	MFID(6)
212		MFID(5)	MFID(4)
213		Short_Hamm_parity(3)	Short_Hamm_parity(2)
214		Short_Hamm_parity(1)	Short_Hamm_parity(0)
215	Status 6	SS(1)	SS(0)
216		MFID(3)	MFID(2)
217		MFID(1)	MFID(0)
218		LC_information(55)	LC_information(54)
219		Short_Hamm_parity(3)	Short_Hamm_parity(2)
220		Short_Hamm_parity(1)	Short_Hamm_parity(0)
221		LC_information(53)	LC_information(52)
222		LC_information(51)	LC_information(50)
223		LC_information(49)	LC_information(48)
224		Short_Hamm_parity(3)	Short_Hamm_parity(2)
225		Short_Hamm_parity(1)	Short_Hamm_parity(0)
226	IMBE 3	c_0(22)	c_1(21)
227		c_2(20)	c_3(19)
228		c_4(10)	c_5(1)
229		c_1(20)	c_0(21)
230		c_3(18)	c_2(19)
231		c_5(0)	c_4(9)
232		c_0(20)	c_1(19)
233		c_2(18)	c_3(17)
234		c_4(8)	c_6(14)
235		c_1(18)	c_0(19)
236		c_3(16)	c_2(17)
237		c_6(13)	c_4(7)
238		c_0(18)	c_1(17)
239		c_2(16)	c_3(15)
240		c_4(6)	c_6(12)
241		c_1(16)	c_0(17)
242		c_3(14)	c_2(15)
243		c_6(11)	c_4(5)
244		c_0(16)	c_1(15)
245		c_2(14)	c_3(13)
246		c_4(4)	c_6(10)
247		c_1(14)	c_0(15)
248		c_3(12)	c_2(13)
249		c_6(9)	c_4(3)
250		c_0(14)	c_1(13)
251	Status 7	SS(1)	SS(0)
252		c_2(12)	c_3(11)
253		c_4(2)	c_6(8)
254		c_1(12)	c_0(13)
255		c_3(10)	c_2(11)
256		c_6(7)	c_4(1)
257		c_0(12)	c_1(11)
258		c_2(10)	c_3(9)
259		c_4(0)	c_6(6)
260		c_1(10)	c_0(11)
261		c_3(8)	c_2(9)
262		c_6(5)	c_5(14)
263		c_0(10)	c_1(9)
264		c_2(8)	c_3(7)
265		c_5(13)	c_6(4)
266		c_1(8)	c_0(9)
267		c_3(6)	c_2(7)
268		c_6(3)	c_5(12)
269		c_0(8)	c_1(7)
270		c_2(6)	c_3(5)
271		c_5(11)	c_6(2)
272		c_1(6)	c_0(7)
273		c_3(4)	c_2(5)
274		c_6(1)	c_5(10)
275		c_0(6)	c_1(5)
276		c_2(4)	c_3(3)

277		c_5(9)	c_6(0)
278		c_1(4)	c_0(5)
279		c_3(2)	c_2(3)
280		c_7(6)	c_5(8)
281		c_0(4)	c_1(3)
282		c_2(2)	c_3(1)
283		c_5(7)	c_7(5)
284		c_1(2)	c_0(3)
285		c_3(0)	c_2(1)
286		c_7(4)	c_5(6)
287	Status 8	SS(1)	SS(0)
288		c_0(2)	c_1(1)
289		c_2(0)	c_4(14)
290		c_5(5)	c_7(3)
291		c_1(0)	c_0(1)
292		c_4(13)	c_3(22)
293		c_7(2)	c_5(4)
294		c_0(0)	c_2(22)
295		c_3(21)	c_4(12)
296		c_5(3)	c_7(1)
297		c_2(21)	c_1(22)
298		c_4(11)	c_3(20)
299	END of 3	c_7(0)	c_5(2)
300		LC_information(47)	LC_information(46)
301		LC_information(45)	LC_information(44)
302		LC_information(43)	LC_information(42)
303		Short_Hamm_parity(3)	Short_Hamm_parity(2)
304		Short_Hamm_parity(1)	Short_Hamm_parity(0)
305		LC_information(41)	LC_information(40)
306		LC_information(39)	LC_information(38)
307		LC_information(37)	LC_information(36)
308		Short_Hamm_parity(3)	Short_Hamm_parity(2)
309		Short_Hamm_parity(1)	Short_Hamm_parity(0)
310		LC_information(35)	LC_information(34)
311		LC_information(33)	LC_information(32)
312		LC_information(31)	LC_information(30)
313		Short_Hamm_parity(3)	Short_Hamm_parity(2)
314		Short_Hamm_parity(1)	Short_Hamm_parity(0)
315		LC_information(29)	LC_information(28)
316		LC_information(27)	LC_information(26)
317		LC_information(25)	LC_information(24)
318		Short_Hamm_parity(3)	Short_Hamm_parity(2)
319		Short_Hamm_parity(1)	Short_Hamm_parity(0)
320	IMBE 4	c_0(22)	c_1(21)
321		c_2(20)	c_3(19)
322		c_4(10)	c_5(1)
323	Status 9	SS(1)	SS(0)
324		c_1(20)	c_0(21)
325		c_3(18)	c_2(19)
326		c_5(0)	c_4(9)
327		c_0(20)	c_1(19)
328		c_2(18)	c_3(17)
329		c_4(8)	c_6(14)
330		c_1(18)	c_0(19)
331		c_3(16)	c_2(17)
332		c_6(13)	c_4(7)
333		c_0(18)	c_1(17)
334		c_2(16)	c_3(15)
335		c_4(6)	c_6(12)
336		c_1(16)	c_0(17)
337		c_3(14)	c_2(15)
338		c_6(11)	c_4(5)
339		c_0(16)	c_1(15)
340		c_2(14)	c_3(13)
341		c_4(4)	c_6(10)
342		c_1(14)	c_0(15)
343		c_3(12)	c_2(13)
344		c_6(9)	c_4(3)
345		c_0(14)	c_1(13)
346		c_2(12)	c_3(11)
347		c_4(2)	c_6(8)
348		c_1(12)	c_0(13)

349		c_3(10)	c_2(11)
350		c_6(7)	c_4(1)
351		c_0(12)	c_1(11)
352		c_2(10)	c_3(9)
353		c_4(0)	c_6(6)
354		c_1(10)	c_0(11)
355		c_3(8)	c_2(9)
356		c_6(5)	c_5(14)
357		c_0(10)	c_1(9)
358		c_2(8)	c_3(7)
359	Status 10	SS(1)	SS(0)
360		c_5(13)	c_6(4)
361		c_1(8)	c_0(9)
362		c_3(6)	c_2(7)
363		c_6(3)	c_5(12)
364		c_0(8)	c_1(7)
365		c_2(6)	c_3(5)
366		c_5(11)	c_6(2)
367		c_1(6)	c_0(7)
368		c_3(4)	c_2(5)
369		c_6(1)	c_5(10)
370		c_0(6)	c_1(5)
371		c_2(4)	c_3(3)
372		c_5(9)	c_6(0)
373		c_1(4)	c_0(5)
374		c_3(2)	c_2(3)
375		c_7(6)	c_5(8)
376		c_0(4)	c_1(3)
377		c_2(2)	c_3(1)
378		c_5(7)	c_7(5)
379		c_1(2)	c_0(3)
380		c_3(0)	c_2(1)
381		c_7(4)	c_5(6)
382		c_0(2)	c_1(1)
383		c_2(0)	c_4(14)
384		c_5(5)	c_7(3)
385		c_1(0)	c_0(1)
386		c_4(13)	c_3(22)
387		c_7(2)	c_5(4)
388		c_0(0)	c_2(22)
389		c_3(21)	c_4(12)
390		c_5(3)	c_7(1)
391		c_2(21)	c_1(22)
392		c_4(11)	c_3(20)
393	END of 4	c_7(0)	c_5(2)
394		LC_information(23)	LC_information(22)
395	Status 11	SS(1)	SS(0)
396		LC_information(21)	LC_information(20)
397		LC_information(19)	LC_information(18)
398		Short_Hamm_parity(3)	Short_Hamm_parity(2)
399		Short_Hamm_parity(1)	Short_Hamm_parity(0)
400		LC_information(17)	LC_information(16)
401		LC_information(15)	LC_information(14)
402		LC_information(13)	LC_information(12)
403		Short_Hamm_parity(3)	Short_Hamm_parity(2)
404		Short_Hamm_parity(1)	Short_Hamm_parity(0)
405		LC_information(11)	LC_information(10)
406		LC_information(9)	LC_information(8)
407		LC_information(7)	LC_information(6)
408		Short_Hamm_parity(3)	Short_Hamm_parity(2)
409		Short_Hamm_parity(1)	Short_Hamm_parity(0)
410		LC_information(5)	LC_information(4)
411		LC_information(3)	LC_information(2)
412		LC_information(1)	LC_information(0)
413		Short_Hamm_parity(3)	Short_Hamm_parity(2)
414		Short_Hamm_parity(1)	Short_Hamm_parity(0)
415	IMBE 5	c_0(22)	c_1(21)
416		c_2(20)	c_3(19)
417		c_4(10)	c_5(1)
418		c_1(20)	c_0(21)
419		c_3(18)	c_2(19)
420		c_5(0)	c_4(9)

421		c_0(20)	c_1(19)
422		c_2(18)	c_3(17)
423		c_4(8)	c_6(14)
424		c_1(18)	c_0(19)
425		c_3(16)	c_2(17)
426		c_6(13)	c_4(7)
427		c_0(18)	c_1(17)
428		c_2(16)	c_3(15)
429		c_4(6)	c_6(12)
430		c_1(16)	c_0(17)
431	Status 12	SS(1)	SS(0)
432		c_3(14)	c_2(15)
433		c_6(11)	c_4(5)
434		c_0(16)	c_1(15)
435		c_2(14)	c_3(13)
436		c_4(4)	c_6(10)
437		c_1(14)	c_0(15)
438		c_3(12)	c_2(13)
439		c_6(9)	c_4(3)
440		c_0(14)	c_1(13)
441		c_2(12)	c_3(11)
442		c_4(2)	c_6(8)
443		c_1(12)	c_0(13)
444		c_3(10)	c_2(11)
445		c_6(7)	c_4(1)
446		c_0(12)	c_1(11)
447		c_2(10)	c_3(9)
448		c_4(0)	c_6(6)
449		c_1(10)	c_0(11)
450		c_3(8)	c_2(9)
451		c_6(5)	c_5(14)
452		c_0(10)	c_1(9)
453		c_2(8)	c_3(7)
454		c_5(13)	c_6(4)
455		c_1(8)	c_0(9)
456		c_3(6)	c_2(7)
457		c_6(3)	c_5(12)
458		c_0(8)	c_1(7)
459		c_2(6)	c_3(5)
460		c_5(11)	c_6(2)
461		c_1(6)	c_0(7)
462		c_3(4)	c_2(5)
463		c_6(1)	c_5(10)
464		c_0(6)	c_1(5)
465		c_2(4)	c_3(3)
466		c_5(9)	c_6(0)
467	Status 13	SS(1)	SS(0)
468		c_1(4)	c_0(5)
469		c_3(2)	c_2(3)
470		c_7(6)	c_5(8)
471		c_0(4)	c_1(3)
472		c_2(2)	c_3(1)
473		c_5(7)	c_7(5)
474		c_1(2)	c_0(3)
475		c_3(0)	c_2(1)
476		c_7(4)	c_5(6)
477		c_0(2)	c_1(1)
478		c_2(0)	c_4(14)
479		c_5(5)	c_7(3)
480		c_1(0)	c_0(1)
481		c_4(13)	c_3(22)
482		c_7(2)	c_5(4)
483		c_0(0)	c_2(22)
484		c_3(21)	c_4(12)
485		c_5(3)	c_7(1)
486		c_2(21)	c_1(22)
487		c_4(11)	c_3(20)
488	END of 5	c_7(0)	c_5(2)
489		RS_parity_11(5)	RS_parity_11(4)
490		RS_parity_11(3)	RS_parity_11(2)
491		RS_parity_11(1)	RS_parity_11(0)
492		Short_Hamm_parity(3)	Short_Hamm_parity(2)

493		Short_Hamm_parity(1)	Short_Hamm_parity(0)
494		RS_parity_10(5)	RS_parity_10(4)
495		RS_parity_10(3)	RS_parity_10(2)
496		RS_parity_10(1)	RS_parity_10(0)
497		Short_Hamm_parity(3)	Short_Hamm_parity(2)
498		Short_Hamm_parity(1)	Short_Hamm_parity(0)
499		RS_parity_9(5)	RS_parity_9(4)
500		RS_parity_9(3)	RS_parity_9(2)
501		RS_parity_9(1)	RS_parity_9(0)
502		Short_Hamm_parity(3)	Short_Hamm_parity(2)
503	Status 14	SS(1)	SS(0)
504		Short_Hamm_parity(1)	Short_Hamm_parity(0)
505		RS_parity_8(5)	RS_parity_8(4)
506		RS_parity_8(3)	RS_parity_8(2)
507		RS_parity_8(1)	RS_parity_8(0)
508		Short_Hamm_parity(3)	Short_Hamm_parity(2)
509		Short_Hamm_parity(1)	Short_Hamm_parity(0)
510	IMBE 6	c_0(22)	c_1(21)
511		c_2(20)	c_3(19)
512		c_4(10)	c_5(1)
513		c_1(20)	c_0(21)
514		c_3(18)	c_2(19)
515		c_5(0)	c_4(9)
516		c_0(20)	c_1(19)
517		c_2(18)	c_3(17)
518		c_4(8)	c_6(14)
519		c_1(18)	c_0(19)
520		c_3(16)	c_2(17)
521		c_6(13)	c_4(7)
522		c_0(18)	c_1(17)
523		c_2(16)	c_3(15)
524		c_4(6)	c_6(12)
525		c_1(16)	c_0(17)
526		c_3(14)	c_2(15)
527		c_6(11)	c_4(5)
528		c_0(16)	c_1(15)
529		c_2(14)	c_3(13)
530		c_4(4)	c_6(10)
531		c_1(14)	c_0(15)
532		c_3(12)	c_2(13)
533		c_6(9)	c_4(3)
534		c_0(14)	c_1(13)
535		c_2(12)	c_3(11)
536		c_4(2)	c_6(8)
537		c_1(12)	c_0(13)
538		c_3(10)	c_2(11)
539	Status 15	SS(1)	SS(0)
540		c_6(7)	c_4(1)
541		c_0(12)	c_1(11)
542		c_2(10)	c_3(9)
543		c_4(0)	c_6(6)
544		c_1(10)	c_0(11)
545		c_3(8)	c_2(9)
546		c_6(5)	c_5(14)
547		c_0(10)	c_1(9)
548		c_2(8)	c_3(7)
549		c_5(13)	c_6(4)
550		c_1(8)	c_0(9)
551		c_3(6)	c_2(7)
552		c_6(3)	c_5(12)
553		c_0(8)	c_1(7)
554		c_2(6)	c_3(5)
555		c_5(11)	c_6(2)
556		c_1(6)	c_0(7)
557		c_3(4)	c_2(5)
558		c_6(1)	c_5(10)
559		c_0(6)	c_1(5)
560		c_2(4)	c_3(3)
561		c_5(9)	c_6(0)
562		c_1(4)	c_0(5)
563		c_3(2)	c_2(3)
564		c_7(6)	c_5(8)

565		c_0(4)	c_1(3)
566		c_2(2)	c_3(1)
567		c_5(7)	c_7(5)
568		c_1(2)	c_0(3)
569		c_3(0)	c_2(1)
570		c_7(4)	c_5(6)
571		c_0(2)	c_1(1)
572		c_2(0)	c_4(14)
573		c_5(5)	c_7(3)
574		c_1(0)	c_0(1)
575	Status 16	SS(1)	SS(0)
576		c_4(13)	c_3(22)
577		c_7(2)	c_5(4)
578		c_0(0)	c_2(22)
579		c_3(21)	c_4(12)
580		c_5(3)	c_7(1)
581		c_2(21)	c_1(22)
582		c_4(11)	c_3(20)
583	END of 6	c_7(0)	c_5(2)
584		RS_parity_7(5)	RS_parity_7(4)
585		RS_parity_7(3)	RS_parity_7(2)
586		RS_parity_7(1)	RS_parity_7(0)
587		Short_Hamm_parity(3)	Short_Hamm_parity(2)
588		Short_Hamm_parity(1)	Short_Hamm_parity(0)
589		RS_parity_6(5)	RS_parity_6(4)
590		RS_parity_6(3)	RS_parity_6(2)
591		RS_parity_6(1)	RS_parity_6(0)
592		Short_Hamm_parity(3)	Short_Hamm_parity(2)
593		Short_Hamm_parity(1)	Short_Hamm_parity(0)
594		RS_parity_5(5)	RS_parity_5(4)
595		RS_parity_5(3)	RS_parity_5(2)
596		RS_parity_5(1)	RS_parity_5(0)
597		Short_Hamm_parity(3)	Short_Hamm_parity(2)
598		Short_Hamm_parity(1)	Short_Hamm_parity(0)
599		RS_parity_4(5)	RS_parity_4(4)
600		RS_parity_4(3)	RS_parity_4(2)
601		RS_parity_4(1)	RS_parity_4(0)
602		Short_Hamm_parity(3)	Short_Hamm_parity(2)
603		Short_Hamm_parity(1)	Short_Hamm_parity(0)
604	IMBE 7	c_0(22)	c_1(21)
605		c_2(20)	c_3(19)
606		c_4(10)	c_5(1)
607		c_1(20)	c_0(21)
608		c_3(18)	c_2(19)
609		c_5(0)	c_4(9)
610		c_0(20)	c_1(19)
611	Status 17	SS(1)	SS(0)
612		c_2(18)	c_3(17)
613		c_4(8)	c_6(14)
614		c_1(18)	c_0(19)
615		c_3(16)	c_2(17)
616		c_6(13)	c_4(7)
617		c_0(18)	c_1(17)
618		c_2(16)	c_3(15)
619		c_4(6)	c_6(12)
620		c_1(16)	c_0(17)
621		c_3(14)	c_2(15)
622		c_6(11)	c_4(5)
623		c_0(16)	c_1(15)
624		c_2(14)	c_3(13)
625		c_4(4)	c_6(10)
626		c_1(14)	c_0(15)
627		c_3(12)	c_2(13)
628		c_6(9)	c_4(3)
629		c_0(14)	c_1(13)
630		c_2(12)	c_3(11)
631		c_4(2)	c_6(8)
632		c_1(12)	c_0(13)
633		c_3(10)	c_2(11)
634		c_6(7)	c_4(1)
635		c_0(12)	c_1(11)
636		c_2(10)	c_3(9)

637		c_4(0)	c_6(6)
638		c_1(10)	c_0(11)
639		c_3(8)	c_2(9)
640		c_6(5)	c_5(14)
641		c_0(10)	c_1(9)
642		c_2(8)	c_3(7)
643		c_5(13)	c_6(4)
644		c_1(8)	c_0(9)
645		c_3(6)	c_2(7)
646		c_6(3)	c_5(12)
647	Status 18	SS(1)	SS(0)
648		c_0(8)	c_1(7)
649		c_2(6)	c_3(5)
650		c_5(11)	c_6(2)
651		c_1(6)	c_0(7)
652		c_3(4)	c_2(5)
653		c_6(1)	c_5(10)
654		c_0(6)	c_1(5)
655		c_2(4)	c_3(3)
656		c_5(9)	c_6(0)
657		c_1(4)	c_0(5)
658		c_3(2)	c_2(3)
659		c_7(6)	c_5(8)
660		c_0(4)	c_1(3)
661		c_2(2)	c_3(1)
662		c_5(7)	c_7(5)
663		c_1(2)	c_0(3)
664		c_3(0)	c_2(1)
665		c_7(4)	c_5(6)
666		c_0(2)	c_1(1)
667		c_2(0)	c_4(14)
668		c_5(5)	c_7(3)
669		c_1(0)	c_0(1)
670		c_4(13)	c_3(22)
671		c_7(2)	c_5(4)
672		c_0(0)	c_2(22)
673		c_3(21)	c_4(12)
674		c_5(3)	c_7(1)
675		c_2(21)	c_1(22)
676		c_4(11)	c_3(20)
677	END of 7	c_7(0)	c_5(2)
678		RS_parity_3(5)	RS_parity_3(4)
679		RS_parity_3(3)	RS_parity_3(2)
680		RS_parity_3(1)	RS_parity_3(0)
681		Short_Hamm_parity(3)	Short_Hamm_parity(2)
682		Short_Hamm_parity(1)	Short_Hamm_parity(0)
683	Status 19	SS(1)	SS(0)
684		RS_parity_2(5)	RS_parity_2(4)
685		RS_parity_2(3)	RS_parity_2(2)
686		RS_parity_2(1)	RS_parity_2(0)
687		Short_Hamm_parity(3)	Short_Hamm_parity(2)
688		Short_Hamm_parity(1)	Short_Hamm_parity(0)
689		RS_parity_1(5)	RS_parity_1(4)
690		RS_parity_1(3)	RS_parity_1(2)
691		RS_parity_1(1)	RS_parity_1(0)
692		Short_Hamm_parity(3)	Short_Hamm_parity(2)
693		Short_Hamm_parity(1)	Short_Hamm_parity(0)
694		RS_parity_0(5)	RS_parity_0(4)
695		RS_parity_0(3)	RS_parity_0(2)
696		RS_parity_0(1)	RS_parity_0(0)
697		Short_Hamm_parity(3)	Short_Hamm_parity(2)
698		Short_Hamm_parity(1)	Short_Hamm_parity(0)
699	IMBE 8	c_0(22)	c_1(21)
700		c_2(20)	c_3(19)
701		c_4(10)	c_5(1)
702		c_1(20)	c_0(21)
703		c_3(18)	c_2(19)
704		c_5(0)	c_4(9)
705		c_0(20)	c_1(19)
706		c_2(18)	c_3(17)
707		c_4(8)	c_6(14)
708		c_1(18)	c_0(19)

709		c_3(16)	c_2(17)
710		c_6(13)	c_4(7)
711		c_0(18)	c_1(17)
712		c_2(16)	c_3(15)
713		c_4(6)	c_6(12)
714		c_1(16)	c_0(17)
715		c_3(14)	c_2(15)
716		c_6(11)	c_4(5)
717		c_0(16)	c_1(15)
718		c_2(14)	c_3(13)
719	Status 20	SS(1)	SS(0)
720		c_4(4)	c_6(10)
721		c_1(14)	c_0(15)
722		c_3(12)	c_2(13)
723		c_6(9)	c_4(3)
724		c_0(14)	c_1(13)
725		c_2(12)	c_3(11)
726		c_4(2)	c_6(8)
727		c_1(12)	c_0(13)
728		c_3(10)	c_2(11)
729		c_6(7)	c_4(1)
730		c_0(12)	c_1(11)
731		c_2(10)	c_3(9)
732		c_4(0)	c_6(6)
733		c_1(10)	c_0(11)
734		c_3(8)	c_2(9)
735		c_6(5)	c_5(14)
736		c_0(10)	c_1(9)
737		c_2(8)	c_3(7)
738		c_5(13)	c_6(4)
739		c_1(8)	c_0(9)
740		c_3(6)	c_2(7)
741		c_6(3)	c_5(12)
742		c_0(8)	c_1(7)
743		c_2(6)	c_3(5)
744		c_5(11)	c_6(2)
745		c_1(6)	c_0(7)
746		c_3(4)	c_2(5)
747		c_6(1)	c_5(10)
748		c_0(6)	c_1(5)
749		c_2(4)	c_3(3)
750		c_5(9)	c_6(0)
751		c_1(4)	c_0(5)
752		c_3(2)	c_2(3)
753		c_7(6)	c_5(8)
754		c_0(4)	c_1(3)
755	Status 21	SS(1)	SS(0)
756		c_2(2)	c_3(1)
757		c_5(7)	c_7(5)
758		c_1(2)	c_0(3)
759		c_3(0)	c_2(1)
760		c_7(4)	c_5(6)
761		c_0(2)	c_1(1)
762		c_2(0)	c_4(14)
763		c_5(5)	c_7(3)
764		c_1(0)	c_0(1)
765		c_4(13)	c_3(22)
766		c_7(2)	c_5(4)
767		c_0(0)	c_2(22)
768		c_3(21)	c_4(12)
769		c_5(3)	c_7(1)
770		c_2(21)	c_1(22)
771		c_4(11)	c_3(20)
772	END of 8	c_7(0)	c_5(2)
773		LSD_info_1(7)	LSD_info_1(6)
774		LSD_info_1(5)	LSD_info_1(4)
775		LSD_info_1(3)	LSD_info_1(2)
776		LSD_info_1(1)	LSD_info_1(0)
777		cyclic_parity(7)	cyclic_parity(6)
778		cyclic_parity(5)	cyclic_parity(4)
779		cyclic_parity(3)	cyclic_parity(2)
780		cyclic_parity(1)	cyclic_parity(0)

781		LSD_info_2(7)	LSD_info_2(6)
782		LSD_info_2(5)	LSD_info_2(4)
783		LSD_info_2(3)	LSD_info_2(2)
784		LSD_info_2(1)	LSD_info_2(0)
785		cyclic_parity(7)	cyclic_parity(6)
786		cyclic_parity(5)	cyclic_parity(4)
787		cyclic_parity(3)	cyclic_parity(2)
788		cyclic_parity(1)	cyclic_parity(0)
789	IMBE 9	c_0(22)	c_1(21)
790		c_2(20)	c_3(19)
791	Status 22	SS(1)	SS(0)
792		c_4(10)	c_5(1)
793		c_1(20)	c_0(21)
794		c_3(18)	c_2(19)
795		c_5(0)	c_4(9)
796		c_0(20)	c_1(19)
797		c_2(18)	c_3(17)
798		c_4(8)	c_6(14)
799		c_1(18)	c_0(19)
800		c_3(16)	c_2(17)
801		c_6(13)	c_4(7)
802		c_0(18)	c_1(17)
803		c_2(16)	c_3(15)
804		c_4(6)	c_6(12)
805		c_1(16)	c_0(17)
806		c_3(14)	c_2(15)
807		c_6(11)	c_4(5)
808		c_0(16)	c_1(15)
809		c_2(14)	c_3(13)
810		c_4(4)	c_6(10)
811		c_1(14)	c_0(15)
812		c_3(12)	c_2(13)
813		c_6(9)	c_4(3)
814		c_0(14)	c_1(13)
815		c_2(12)	c_3(11)
816		c_4(2)	c_6(8)
817		c_1(12)	c_0(13)
818		c_3(10)	c_2(11)
819		c_6(7)	c_4(1)
820		c_0(12)	c_1(11)
821		c_2(10)	c_3(9)
822		c_4(0)	c_6(6)
823		c_1(10)	c_0(11)
824		c_3(8)	c_2(9)
825		c_6(5)	c_5(14)
826		c_0(10)	c_1(9)
827	Status 23	SS(1)	SS(0)
828		c_2(8)	c_3(7)
829		c_5(13)	c_6(4)
830		c_1(8)	c_0(9)
831		c_3(6)	c_2(7)
832		c_6(3)	c_5(12)
833		c_0(8)	c_1(7)
834		c_2(6)	c_3(5)
835		c_5(11)	c_6(2)
836		c_1(6)	c_0(7)
837		c_3(4)	c_2(5)
838		c_6(1)	c_5(10)
839		c_0(6)	c_1(5)
840		c_2(4)	c_3(3)
841		c_5(9)	c_6(0)
842		c_1(4)	c_0(5)
843		c_3(2)	c_2(3)
844		c_7(6)	c_5(8)
845		c_0(4)	c_1(3)
846		c_2(2)	c_3(1)
847		c_5(7)	c_7(5)
848		c_1(2)	c_0(3)
849		c_3(0)	c_2(1)
850		c_7(4)	c_5(6)
851		c_0(2)	c_1(1)
852		c_2(0)	c_4(14)

853		c_5(5)	c_7(3)
854		c_1(0)	c_0(1)
855		c_4(13)	c_3(22)
856		c_7(2)	c_5(4)
857		c_0(0)	c_2(22)
858		c_3(21)	c_4(12)
859		c_5(3)	c_7(1)
860		c_2(21)	c_1(22)
861		c_4(11)	c_3(20)
862	END of 9	c_7(0)	c_5(2)
863	Status 24	SS(1)	SS(0)

10.4 Logical Link Data Unit 2

The next several pages give the order in which symbols shall be transmitted for the second Logical Link Data Unit for voice. This data unit spans a time interval of 180 ms. It sends 9 frames of IMBE information.

Symbol	Field	Bit 1	Bit 0
864	Frame	0	1
865	Sync	0	1
866		0	1
867		0	1
868		0	1
869		1	1
870		0	1
871		0	1
872		1	1
873		1	1
874		0	1
875		0	1
876		1	1
877		1	1
878		1	1
879		1	1
880		0	1
881		1	1
882		0	1
883		1	1
884		1	1
885		1	1
886		1	1
887		1	1
888	Network	NAC(11)	NAC(10)
889	ID	NAC(9)	NAC(8)
890		NAC(7)	NAC(6)
891		NAC(5)	NAC(4)
892		NAC(3)	NAC(2)
893		NAC(1)	NAC(0)
894		DUID(3)	DUID(2)
895		DUID(1)	DUID(0)
896		BCH_parity(47)	BCH_parity(46)
897		BCH_parity(45)	BCH_parity(44)
898		BCH_parity(43)	BCH_parity(42)
899	Status 25	SS(1)	SS(0)
900		BCH_parity(41)	BCH_parity(40)
901		BCH_parity(39)	BCH_parity(38)
902		BCH_parity(37)	BCH_parity(36)
903		BCH_parity(35)	BCH_parity(34)
904		BCH_parity(33)	BCH_parity(32)
905		BCH_parity(31)	BCH_parity(30)
906		BCH_parity(29)	BCH_parity(28)
907		BCH_parity(27)	BCH_parity(26)
908		BCH_parity(25)	BCH_parity(24)
909		BCH_parity(23)	BCH_parity(22)
910		BCH_parity(21)	BCH_parity(20)
911		BCH_parity(19)	BCH_parity(18)
912		BCH_parity(17)	BCH_parity(16)
913		BCH_parity(15)	BCH_parity(14)
914		BCH_parity(13)	BCH_parity(12)
915		BCH_parity(11)	BCH_parity(10)
916		BCH_parity(9)	BCH_parity(8)
917		BCH_parity(7)	BCH_parity(6)
918		BCH_parity(5)	BCH_parity(4)
919		BCH_parity(3)	BCH_parity(2)
920		BCH_parity(1)	BCH_parity(0)
921	IMBE 10	c_0(22)	c_1(21)
922		c_2(20)	c_3(19)
923		c_4(10)	c_5(1)
924		c_1(20)	c_0(21)

925		c_3(18)	c_2(19)
926		c_5(0)	c_4(9)
927		c_0(20)	c_1(19)
928		c_2(18)	c_3(17)
929		c_4(8)	c_6(14)
930		c_1(18)	c_0(19)
931		c_3(16)	c_2(17)
932		c_6(13)	c_4(7)
933		c_0(18)	c_1(17)
934		c_2(16)	c_3(15)
935	Status 26	SS(1)	SS(0)
936		c_4(6)	c_6(12)
937		c_1(16)	c_0(17)
938		c_3(14)	c_2(15)
939		c_6(11)	c_4(5)
940		c_0(16)	c_1(15)
941		c_2(14)	c_3(13)
942		c_4(4)	c_6(10)
943		c_1(14)	c_0(15)
944		c_3(12)	c_2(13)
945		c_6(9)	c_4(3)
946		c_0(14)	c_1(13)
947		c_2(12)	c_3(11)
948		c_4(2)	c_6(8)
949		c_1(12)	c_0(13)
950		c_3(10)	c_2(11)
951		c_6(7)	c_4(1)
952		c_0(12)	c_1(11)
953		c_2(10)	c_3(9)
954		c_4(0)	c_6(6)
955		c_1(10)	c_0(11)
956		c_3(8)	c_2(9)
957		c_6(5)	c_5(14)
958		c_0(10)	c_1(9)
959		c_2(8)	c_3(7)
960		c_5(13)	c_6(4)
961		c_1(8)	c_0(9)
962		c_3(6)	c_2(7)
963		c_6(3)	c_5(12)
964		c_0(8)	c_1(7)
965		c_2(6)	c_3(5)
966		c_5(11)	c_6(2)
967		c_1(6)	c_0(7)
968		c_3(4)	c_2(5)
969		c_6(1)	c_5(10)
970		c_0(6)	c_1(5)
971	Status 27	SS(1)	SS(0)
972		c_2(4)	c_3(3)
973		c_5(9)	c_6(0)
974		c_1(4)	c_0(5)
975		c_3(2)	c_2(3)
976		c_7(6)	c_5(8)
977		c_0(4)	c_1(3)
978		c_2(2)	c_3(1)
979		c_5(7)	c_7(5)
980		c_1(2)	c_0(3)
981		c_3(0)	c_2(1)
982		c_7(4)	c_5(6)
983		c_0(2)	c_1(1)
984		c_2(0)	c_4(14)
985		c_5(5)	c_7(3)
986		c_1(0)	c_0(1)
987		c_4(13)	c_3(22)
988		c_7(2)	c_5(4)
989		c_0(0)	c_2(22)
990		c_3(21)	c_4(12)
991		c_5(3)	c_7(1)
992		c_2(21)	c_1(22)
993		c_4(11)	c_3(20)
994	END of 10	c_7(0)	c_5(2)
995	IMBE 11	c_0(22)	c_1(21)
996		c_2(20)	c_3(19)

997		c_4(10)	c_5(1)
998		c_1(20)	c_0(21)
999		c_3(18)	c_2(19)
1000		c_5(0)	c_4(9)
1001		c_0(20)	c_1(19)
1002		c_2(18)	c_3(17)
1003		c_4(8)	c_6(14)
1004		c_1(18)	c_0(19)
1005		c_3(16)	c_2(17)
1006		c_6(13)	c_4(7)
1007	Status 28	SS(1)	SS(0)
1008		c_0(18)	c_1(17)
1009		c_2(16)	c_3(15)
1010		c_4(6)	c_6(12)
1011		c_1(16)	c_0(17)
1012		c_3(14)	c_2(15)
1013		c_6(11)	c_4(5)
1014		c_0(16)	c_1(15)
1015		c_2(14)	c_3(13)
1016		c_4(4)	c_6(10)
1017		c_1(14)	c_0(15)
1018		c_3(12)	c_2(13)
1019		c_6(9)	c_4(3)
1020		c_0(14)	c_1(13)
1021		c_2(12)	c_3(11)
1022		c_4(2)	c_6(8)
1023		c_1(12)	c_0(13)
1024		c_3(10)	c_2(11)
1025		c_6(7)	c_4(1)
1026		c_0(12)	c_1(11)
1027		c_2(10)	c_3(9)
1028		c_4(0)	c_6(6)
1029		c_1(10)	c_0(11)
1030		c_3(8)	c_2(9)
1031		c_6(5)	c_5(14)
1032		c_0(10)	c_1(9)
1033		c_2(8)	c_3(7)
1034		c_5(13)	c_6(4)
1035		c_1(8)	c_0(9)
1036		c_3(6)	c_2(7)
1037		c_6(3)	c_5(12)
1038		c_0(8)	c_1(7)
1039		c_2(6)	c_3(5)
1040		c_5(11)	c_6(2)
1041		c_1(6)	c_0(7)
1042		c_3(4)	c_2(5)
1043	Status 29	SS(1)	SS(0)
1044		c_6(1)	c_5(10)
1045		c_0(6)	c_1(5)
1046		c_2(4)	c_3(3)
1047		c_5(9)	c_6(0)
1048		c_1(4)	c_0(5)
1049		c_3(2)	c_2(3)
1050		c_7(6)	c_5(8)
1051		c_0(4)	c_1(3)
1052		c_2(2)	c_3(1)
1053		c_5(7)	c_7(5)
1054		c_1(2)	c_0(3)
1055		c_3(0)	c_2(1)
1056		c_7(4)	c_5(6)
1057		c_0(2)	c_1(1)
1058		c_2(0)	c_4(14)
1059		c_5(5)	c_7(3)
1060		c_1(0)	c_0(1)
1061		c_4(13)	c_3(22)
1062		c_7(2)	c_5(4)
1063		c_0(0)	c_2(22)
1064		c_3(21)	c_4(12)
1065		c_5(3)	c_7(1)
1066		c_2(21)	c_1(22)
1067		c_4(11)	c_3(20)
1068	END of 11	c_7(0)	c_5(2)

1069		MI(71)	MI(70)
1070		MI(69)	MI(68)
1071		MI(67)	MI(66)
1072		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1073		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1074		MI(65)	MI(64)
1075		MI(63)	MI(62)
1076		MI(61)	MI(60)
1077		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1078		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1079	Status 30	SS(1)	SS(0)
1080		MI(59)	MI(58)
1081		MI(57)	MI(56)
1082		MI(55)	MI(54)
1083		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1084		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1085		MI(53)	MI(52)
1086		MI(51)	MI(50)
1087		MI(49)	MI(48)
1088		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1089		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1090	IMBE 12	c_0(22)	c_1(21)
1091		c_2(20)	c_3(19)
1092		c_4(10)	c_5(1)
1093		c_1(20)	c_0(21)
1094		c_3(18)	c_2(19)
1095		c_5(0)	c_4(9)
1096		c_0(20)	c_1(19)
1097		c_2(18)	c_3(17)
1098		c_4(8)	c_6(14)
1099		c_1(18)	c_0(19)
1100		c_3(16)	c_2(17)
1101		c_6(13)	c_4(7)
1102		c_0(18)	c_1(17)
1103		c_2(16)	c_3(15)
1104		c_4(6)	c_6(12)
1105		c_1(16)	c_0(17)
1106		c_3(14)	c_2(15)
1107		c_6(11)	c_4(5)
1108		c_0(16)	c_1(15)
1109		c_2(14)	c_3(13)
1110		c_4(4)	c_6(10)
1111		c_1(14)	c_0(15)
1112		c_3(12)	c_2(13)
1113		c_6(9)	c_4(3)
1114		c_0(14)	c_1(13)
1115	Status 31	SS(1)	SS(0)
1116		c_2(12)	c_3(11)
1117		c_4(2)	c_6(8)
1118		c_1(12)	c_0(13)
1119		c_3(10)	c_2(11)
1120		c_6(7)	c_4(1)
1121		c_0(12)	c_1(11)
1122		c_2(10)	c_3(9)
1123		c_4(0)	c_6(6)
1124		c_1(10)	c_0(11)
1125		c_3(8)	c_2(9)
1126		c_6(5)	c_5(14)
1127		c_0(10)	c_1(9)
1128		c_2(8)	c_3(7)
1129		c_5(13)	c_6(4)
1130		c_1(8)	c_0(9)
1131		c_3(6)	c_2(7)
1132		c_6(3)	c_5(12)
1133		c_0(8)	c_1(7)
1134		c_2(6)	c_3(5)
1135		c_5(11)	c_6(2)
1136		c_1(6)	c_0(7)
1137		c_3(4)	c_2(5)
1138		c_6(1)	c_5(10)
1139		c_0(6)	c_1(5)
1140		c_2(4)	c_3(3)

1141		c_5(9)	c_6(0)
1142		c_1(4)	c_0(5)
1143		c_3(2)	c_2(3)
1144		c_7(6)	c_5(8)
1145		c_0(4)	c_1(3)
1146		c_2(2)	c_3(1)
1147		c_5(7)	c_7(5)
1148		c_1(2)	c_0(3)
1149		c_3(0)	c_2(1)
1150		c_7(4)	c_5(6)
1151	Status 32	SS(1)	SS(0)
1152		c_0(2)	c_1(1)
1153		c_2(0)	c_4(14)
1154		c_5(5)	c_7(3)
1155		c_1(0)	c_0(1)
1156		c_4(13)	c_3(22)
1157		c_7(2)	c_5(4)
1158		c_0(0)	c_2(22)
1159		c_3(21)	c_4(12)
1160		c_5(3)	c_7(1)
1161		c_2(21)	c_1(22)
1162		c_4(11)	c_3(20)
1163	END of 12	c_7(0)	c_5(2)
1164		MI(47)	MI(46)
1165		MI(45)	MI(44)
1166		MI(43)	MI(42)
1167		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1168		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1169		MI(41)	MI(40)
1170		MI(39)	MI(38)
1171		MI(37)	MI(36)
1172		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1173		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1174		MI(35)	MI(34)
1175		MI(33)	MI(32)
1176		MI(31)	MI(30)
1177		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1178		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1179		MI(29)	MI(28)
1180		MI(27)	MI(26)
1181		MI(25)	MI(24)
1182		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1183		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1184	IMBE 13	c_0(22)	c_1(21)
1185		c_2(20)	c_3(19)
1186		c_4(10)	c_5(1)
1187	Status 33	SS(1)	SS(0)
1188		c_1(20)	c_0(21)
1189		c_3(18)	c_2(19)
1190		c_5(0)	c_4(9)
1191		c_0(20)	c_1(19)
1192		c_2(18)	c_3(17)
1193		c_4(8)	c_6(14)
1194		c_1(18)	c_0(19)
1195		c_3(16)	c_2(17)
1196		c_6(13)	c_4(7)
1197		c_0(18)	c_1(17)
1198		c_2(16)	c_3(15)
1199		c_4(6)	c_6(12)
1200		c_1(16)	c_0(17)
1201		c_3(14)	c_2(15)
1202		c_6(11)	c_4(5)
1203		c_0(16)	c_1(15)
1204		c_2(14)	c_3(13)
1205		c_4(4)	c_6(10)
1206		c_1(14)	c_0(15)
1207		c_3(12)	c_2(13)
1208		c_6(9)	c_4(3)
1209		c_0(14)	c_1(13)
1210		c_2(12)	c_3(11)
1211		c_4(2)	c_6(8)
1212		c_1(12)	c_0(13)

1213		c_3(10)	c_2(11)
1214		c_6(7)	c_4(1)
1215		c_0(12)	c_1(11)
1216		c_2(10)	c_3(9)
1217		c_4(0)	c_6(6)
1218		c_1(10)	c_0(11)
1219		c_3(8)	c_2(9)
1220		c_6(5)	c_5(14)
1221		c_0(10)	c_1(9)
1222		c_2(8)	c_3(7)
1223	Status 34	SS(1)	SS(0)
1224		c_5(13)	c_6(4)
1225		c_1(8)	c_0(9)
1226		c_3(6)	c_2(7)
1227		c_6(3)	c_5(12)
1228		c_0(8)	c_1(7)
1229		c_2(6)	c_3(5)
1230		c_5(11)	c_6(2)
1231		c_1(6)	c_0(7)
1232		c_3(4)	c_2(5)
1233		c_6(1)	c_5(10)
1234		c_0(6)	c_1(5)
1235		c_2(4)	c_3(3)
1236		c_5(9)	c_6(0)
1237		c_1(4)	c_0(5)
1238		c_3(2)	c_2(3)
1239		c_7(6)	c_5(8)
1240		c_0(4)	c_1(3)
1241		c_2(2)	c_3(1)
1242		c_5(7)	c_7(5)
1243		c_1(2)	c_0(3)
1244		c_3(0)	c_2(1)
1245		c_7(4)	c_5(6)
1246		c_0(2)	c_1(1)
1247		c_2(0)	c_4(14)
1248		c_5(5)	c_7(3)
1249		c_1(0)	c_0(1)
1250		c_4(13)	c_3(22)
1251		c_7(2)	c_5(4)
1252		c_0(0)	c_2(22)
1253		c_3(21)	c_4(12)
1254		c_5(3)	c_7(1)
1255		c_2(21)	c_1(22)
1256		c_4(11)	c_3(20)
1257	END of 13	c_7(0)	c_5(2)
1258		MI(23)	MI(22)
1259	Status 35	SS(1)	SS(0)
1260		MI(21)	MI(20)
1261		MI(19)	MI(18)
1262		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1263		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1264		MI(17)	MI(16)
1265		MI(15)	MI(14)
1266		MI(13)	MI(12)
1267		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1268		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1269		MI(11)	MI(10)
1270		MI(9)	MI(8)
1271		MI(7)	MI(6)
1272		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1273		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1274		MI(5)	MI(4)
1275		MI(3)	MI(2)
1276		MI(1)	MI(0)
1277		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1278		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1279	IMBE 14	c_0(22)	c_1(21)
1280		c_2(20)	c_3(19)
1281		c_4(10)	c_5(1)
1282		c_1(20)	c_0(21)
1283		c_3(18)	c_2(19)
1284		c_5(0)	c_4(9)

1285		c_0(20)	c_1(19)
1286		c_2(18)	c_3(17)
1287		c_4(8)	c_6(14)
1288		c_1(18)	c_0(19)
1289		c_3(16)	c_2(17)
1290		c_6(13)	c_4(7)
1291		c_0(18)	c_1(17)
1292		c_2(16)	c_3(15)
1293		c_4(6)	c_6(12)
1294		c_1(16)	c_0(17)
1295	Status 36	SS(1)	SS(0)
1296		c_3(14)	c_2(15)
1297		c_6(11)	c_4(5)
1298		c_0(16)	c_1(15)
1299		c_2(14)	c_3(13)
1300		c_4(4)	c_6(10)
1301		c_1(14)	c_0(15)
1302		c_3(12)	c_2(13)
1303		c_6(9)	c_4(3)
1304		c_0(14)	c_1(13)
1305		c_2(12)	c_3(11)
1306		c_4(2)	c_6(8)
1307		c_1(12)	c_0(13)
1308		c_3(10)	c_2(11)
1309		c_6(7)	c_4(1)
1310		c_0(12)	c_1(11)
1311		c_2(10)	c_3(9)
1312		c_4(0)	c_6(6)
1313		c_1(10)	c_0(11)
1314		c_3(8)	c_2(9)
1315		c_6(5)	c_5(14)
1316		c_0(10)	c_1(9)
1317		c_2(8)	c_3(7)
1318		c_5(13)	c_6(4)
1319		c_1(8)	c_0(9)
1320		c_3(6)	c_2(7)
1321		c_6(3)	c_5(12)
1322		c_0(8)	c_1(7)
1323		c_2(6)	c_3(5)
1324		c_5(11)	c_6(2)
1325		c_1(6)	c_0(7)
1326		c_3(4)	c_2(5)
1327		c_6(1)	c_5(10)
1328		c_0(6)	c_1(5)
1329		c_2(4)	c_3(3)
1330		c_5(9)	c_6(0)
1331	Status 37	SS(1)	SS(0)
1332		c_1(4)	c_0(5)
1333		c_3(2)	c_2(3)
1334		c_7(6)	c_5(8)
1335		c_0(4)	c_1(3)
1336		c_2(2)	c_3(1)
1337		c_5(7)	c_7(5)
1338		c_1(2)	c_0(3)
1339		c_3(0)	c_2(1)
1340		c_7(4)	c_5(6)
1341		c_0(2)	c_1(1)
1342		c_2(0)	c_4(14)
1343		c_5(5)	c_7(3)
1344		c_1(0)	c_0(1)
1345		c_4(13)	c_3(22)
1346		c_7(2)	c_5(4)
1347		c_0(0)	c_2(22)
1348		c_3(21)	c_4(12)
1349		c_5(3)	c_7(1)
1350		c_2(21)	c_1(22)
1351		c_4(11)	c_3(20)
1352	END of 14	c_7(0)	c_5(2)
1353		ALGID(7)	ALGID(6)
1354		ALGID(5)	ALGID(4)
1355		ALGID(3)	ALGID(2)
1356		Short_Hamm_parity(3)	Short_Hamm_parity(2)

1357		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1358		ALGID(1)	ALGID(0)
1359		KID(15)	KID(14)
1360		KID(13)	KID(12)
1361		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1362		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1363		KID(11)	KID(10)
1364		KID(9)	KID(8)
1365		KID(7)	KID(6)
1366		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1367	Status 38	SS(1)	SS(0)
1368		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1369		KID(5)	KID(4)
1370		KID(3)	KID(2)
1371		KID(1)	KID(0)
1372		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1373		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1374	IMBE 15	c_0(22)	c_1(21)
1375		c_2(20)	c_3(19)
1376		c_4(10)	c_5(1)
1377		c_1(20)	c_0(21)
1378		c_3(18)	c_2(19)
1379		c_5(0)	c_4(9)
1380		c_0(20)	c_1(19)
1381		c_2(18)	c_3(17)
1382		c_4(8)	c_6(14)
1383		c_1(18)	c_0(19)
1384		c_3(16)	c_2(17)
1385		c_6(13)	c_4(7)
1386		c_0(18)	c_1(17)
1387		c_2(16)	c_3(15)
1388		c_4(6)	c_6(12)
1389		c_1(16)	c_0(17)
1390		c_3(14)	c_2(15)
1391		c_6(11)	c_4(5)
1392		c_0(16)	c_1(15)
1393		c_2(14)	c_3(13)
1394		c_4(4)	c_6(10)
1395		c_1(14)	c_0(15)
1396		c_3(12)	c_2(13)
1397		c_6(9)	c_4(3)
1398		c_0(14)	c_1(13)
1399		c_2(12)	c_3(11)
1400		c_4(2)	c_6(8)
1401		c_1(12)	c_0(13)
1402		c_3(10)	c_2(11)
1403	Status 39	SS(1)	SS(0)
1404		c_6(7)	c_4(1)
1405		c_0(12)	c_1(11)
1406		c_2(10)	c_3(9)
1407		c_4(0)	c_6(6)
1408		c_1(10)	c_0(11)
1409		c_3(8)	c_2(9)
1410		c_6(5)	c_5(14)
1411		c_0(10)	c_1(9)
1412		c_2(8)	c_3(7)
1413		c_5(13)	c_6(4)
1414		c_1(8)	c_0(9)
1415		c_3(6)	c_2(7)
1416		c_6(3)	c_5(12)
1417		c_0(8)	c_1(7)
1418		c_2(6)	c_3(5)
1419		c_5(11)	c_6(2)
1420		c_1(6)	c_0(7)
1421		c_3(4)	c_2(5)
1422		c_6(1)	c_5(10)
1423		c_0(6)	c_1(5)
1424		c_2(4)	c_3(3)
1425		c_5(9)	c_6(0)
1426		c_1(4)	c_0(5)
1427		c_3(2)	c_2(3)
1428		c_7(6)	c_5(8)

1429		c_0(4)	c_1(3)
1430		c_2(2)	c_3(1)
1431		c_5(7)	c_7(5)
1432		c_1(2)	c_0(3)
1433		c_3(0)	c_2(1)
1434		c_7(4)	c_5(6)
1435		c_0(2)	c_1(1)
1436		c_2(0)	c_4(14)
1437		c_5(5)	c_7(3)
1438		c_1(0)	c_0(1)
1439	Status 40	SS(1)	SS(0)
1440		c_4(13)	c_3(22)
1441		c_7(2)	c_5(4)
1442		c_0(0)	c_2(22)
1443		c_3(21)	c_4(12)
1444		c_5(3)	c_7(1)
1445		c_2(21)	c_1(22)
1446		c_4(11)	c_3(20)
1447	END of 15	c_7(0)	c_5(2)
1448		RS_parity_7(5)	RS_parity_7(4)
1449		RS_parity_7(3)	RS_parity_7(2)
1450		RS_parity_7(1)	RS_parity_7(0)
1451		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1452		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1453		RS_parity_6(5)	RS_parity_6(4)
1454		RS_parity_6(3)	RS_parity_6(2)
1455		RS_parity_6(1)	RS_parity_6(0)
1456		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1457		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1458		RS_parity_5(5)	RS_parity_5(4)
1459		RS_parity_5(3)	RS_parity_5(2)
1460		RS_parity_5(1)	RS_parity_5(0)
1461		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1462		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1463		RS_parity_4(5)	RS_parity_4(4)
1464		RS_parity_4(3)	RS_parity_4(2)
1465		RS_parity_4(1)	RS_parity_4(0)
1466		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1467		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1468	IMBE 16	c_0(22)	c_1(21)
1469		c_2(20)	c_3(19)
1470		c_4(10)	c_5(1)
1471		c_1(20)	c_0(21)
1472		c_3(18)	c_2(19)
1473		c_5(0)	c_4(9)
1474		c_0(20)	c_1(19)
1475	Status 41	SS(1)	SS(0)
1476		c_2(18)	c_3(17)
1477		c_4(8)	c_6(14)
1478		c_1(18)	c_0(19)
1479		c_3(16)	c_2(17)
1480		c_6(13)	c_4(7)
1481		c_0(18)	c_1(17)
1482		c_2(16)	c_3(15)
1483		c_4(6)	c_6(12)
1484		c_1(16)	c_0(17)
1485		c_3(14)	c_2(15)
1486		c_6(11)	c_4(5)
1487		c_0(16)	c_1(15)
1488		c_2(14)	c_3(13)
1489		c_4(4)	c_6(10)
1490		c_1(14)	c_0(15)
1491		c_3(12)	c_2(13)
1492		c_6(9)	c_4(3)
1493		c_0(14)	c_1(13)
1494		c_2(12)	c_3(11)
1495		c_4(2)	c_6(8)
1496		c_1(12)	c_0(13)
1497		c_3(10)	c_2(11)
1498		c_6(7)	c_4(1)
1499		c_0(12)	c_1(11)
1500		c_2(10)	c_3(9)

1501		c_4(0)	c_6(6)
1502		c_1(10)	c_0(11)
1503		c_3(8)	c_2(9)
1504		c_6(5)	c_5(14)
1505		c_0(10)	c_1(9)
1506		c_2(8)	c_3(7)
1507		c_5(13)	c_6(4)
1508		c_1(8)	c_0(9)
1509		c_3(6)	c_2(7)
1510		c_6(3)	c_5(12)
1511	Status 42	SS(1)	SS(0)
1512		c_0(8)	c_1(7)
1513		c_2(6)	c_3(5)
1514		c_5(11)	c_6(2)
1515		c_1(6)	c_0(7)
1516		c_3(4)	c_2(5)
1517		c_6(1)	c_5(10)
1518		c_0(6)	c_1(5)
1519		c_2(4)	c_3(3)
1520		c_5(9)	c_6(0)
1521		c_1(4)	c_0(5)
1522		c_3(2)	c_2(3)
1523		c_7(6)	c_5(8)
1524		c_0(4)	c_1(3)
1525		c_2(2)	c_3(1)
1526		c_5(7)	c_7(5)
1527		c_1(2)	c_0(3)
1528		c_3(0)	c_2(1)
1529		c_7(4)	c_5(6)
1530		c_0(2)	c_1(1)
1531		c_2(0)	c_4(14)
1532		c_5(5)	c_7(3)
1533		c_1(0)	c_0(1)
1534		c_4(13)	c_3(22)
1535		c_7(2)	c_5(4)
1536		c_0(0)	c_2(22)
1537		c_3(21)	c_4(12)
1538		c_5(3)	c_7(1)
1539		c_2(21)	c_1(22)
1540		c_4(11)	c_3(20)
1541	END of 16	c_7(0)	c_5(2)
1542		RS_parity_3(5)	RS_parity_3(4)
1543		RS_parity_3(3)	RS_parity_3(2)
1544		RS_parity_3(1)	RS_parity_3(0)
1545		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1546		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1547	Status 43	SS(1)	SS(0)
1548		RS_parity_2(5)	RS_parity_2(4)
1549		RS_parity_2(3)	RS_parity_2(2)
1550		RS_parity_2(1)	RS_parity_2(0)
1551		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1552		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1553		RS_parity_1(5)	RS_parity_1(4)
1554		RS_parity_1(3)	RS_parity_1(2)
1555		RS_parity_1(1)	RS_parity_1(0)
1556		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1557		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1558		RS_parity_0(5)	RS_parity_0(4)
1559		RS_parity_0(3)	RS_parity_0(2)
1560		RS_parity_0(1)	RS_parity_0(0)
1561		Short_Hamm_parity(3)	Short_Hamm_parity(2)
1562		Short_Hamm_parity(1)	Short_Hamm_parity(0)
1563	IMBE 17	c_0(22)	c_1(21)
1564		c_2(20)	c_3(19)
1565		c_4(10)	c_5(1)
1566		c_1(20)	c_0(21)
1567		c_3(18)	c_2(19)
1568		c_5(0)	c_4(9)
1569		c_0(20)	c_1(19)
1570		c_2(18)	c_3(17)
1571		c_4(8)	c_6(14)
1572		c_1(18)	c_0(19)

1573		c_3(16)	c_2(17)
1574		c_6(13)	c_4(7)
1575		c_0(18)	c_1(17)
1576		c_2(16)	c_3(15)
1577		c_4(6)	c_6(12)
1578		c_1(16)	c_0(17)
1579		c_3(14)	c_2(15)
1580		c_6(11)	c_4(5)
1581		c_0(16)	c_1(15)
1582		c_2(14)	c_3(13)
1583	Status 44	SS(1)	SS(0)
1584		c_4(4)	c_6(10)
1585		c_1(14)	c_0(15)
1586		c_3(12)	c_2(13)
1587		c_6(9)	c_4(3)
1588		c_0(14)	c_1(13)
1589		c_2(12)	c_3(11)
1590		c_4(2)	c_6(8)
1591		c_1(12)	c_0(13)
1592		c_3(10)	c_2(11)
1593		c_6(7)	c_4(1)
1594		c_0(12)	c_1(11)
1595		c_2(10)	c_3(9)
1596		c_4(0)	c_6(6)
1597		c_1(10)	c_0(11)
1598		c_3(8)	c_2(9)
1599		c_6(5)	c_5(14)
1600		c_0(10)	c_1(9)
1601		c_2(8)	c_3(7)
1602		c_5(13)	c_6(4)
1603		c_1(8)	c_0(9)
1604		c_3(6)	c_2(7)
1605		c_6(3)	c_5(12)
1606		c_0(8)	c_1(7)
1607		c_2(6)	c_3(5)
1608		c_5(11)	c_6(2)
1609		c_1(6)	c_0(7)
1610		c_3(4)	c_2(5)
1611		c_6(1)	c_5(10)
1612		c_0(6)	c_1(5)
1613		c_2(4)	c_3(3)
1614		c_5(9)	c_6(0)
1615		c_1(4)	c_0(5)
1616		c_3(2)	c_2(3)
1617		c_7(6)	c_5(8)
1618		c_0(4)	c_1(3)
1619	Status 45	SS(1)	SS(0)
1620		c_2(2)	c_3(1)
1621		c_5(7)	c_7(5)
1622		c_1(2)	c_0(3)
1623		c_3(0)	c_2(1)
1624		c_7(4)	c_5(6)
1625		c_0(2)	c_1(1)
1626		c_2(0)	c_4(14)
1627		c_5(5)	c_7(3)
1628		c_1(0)	c_0(1)
1629		c_4(13)	c_3(22)
1630		c_7(2)	c_5(4)
1631		c_0(0)	c_2(22)
1632		c_3(21)	c_4(12)
1633		c_5(3)	c_7(1)
1634		c_2(21)	c_1(22)
1635		c_4(11)	c_3(20)
1636	END of 17	c_7(0)	c_5(2)
1637		LSD_info_3(7)	LSD_info_3(6)
1638		LSD_info_3(5)	LSD_info_3(4)
1639		LSD_info_3(3)	LSD_info_3(2)
1640		LSD_info_3(1)	LSD_info_3(0)
1641		cyclic_parity(7)	cyclic_parity(6)
1642		cyclic_parity(5)	cyclic_parity(4)
1643		cyclic_parity(3)	cyclic_parity(2)
1644		cyclic_parity(1)	cyclic_parity(0)

1645		LSD_info_4(7)	LSD_info_4(6)
1646		LSD_info_4(5)	LSD_info_4(4)
1647		LSD_info_4(3)	LSD_info_4(2)
1648		LSD_info_4(1)	LSD_info_4(0)
1649		cyclic_parity(7)	cyclic_parity(6)
1650		cyclic_parity(5)	cyclic_parity(4)
1651		cyclic_parity(3)	cyclic_parity(2)
1652		cyclic_parity(1)	cyclic_parity(0)
1653	IMBE 18	c_0(22)	c_1(21)
1654		c_2(20)	c_3(19)
1655	Status 46	SS(1)	SS(0)
1656		c_4(10)	c_5(1)
1657		c_1(20)	c_0(21)
1658		c_3(18)	c_2(19)
1659		c_5(0)	c_4(9)
1660		c_0(20)	c_1(19)
1661		c_2(18)	c_3(17)
1662		c_4(8)	c_6(14)
1663		c_1(18)	c_0(19)
1664		c_3(16)	c_2(17)
1665		c_6(13)	c_4(7)
1666		c_0(18)	c_1(17)
1667		c_2(16)	c_3(15)
1668		c_4(6)	c_6(12)
1669		c_1(16)	c_0(17)
1670		c_3(14)	c_2(15)
1671		c_6(11)	c_4(5)
1672		c_0(16)	c_1(15)
1673		c_2(14)	c_3(13)
1674		c_4(4)	c_6(10)
1675		c_1(14)	c_0(15)
1676		c_3(12)	c_2(13)
1677		c_6(9)	c_4(3)
1678		c_0(14)	c_1(13)
1679		c_2(12)	c_3(11)
1680		c_4(2)	c_6(8)
1681		c_1(12)	c_0(13)
1682		c_3(10)	c_2(11)
1683		c_6(7)	c_4(1)
1684		c_0(12)	c_1(11)
1685		c_2(10)	c_3(9)
1686		c_4(0)	c_6(6)
1687		c_1(10)	c_0(11)
1688		c_3(8)	c_2(9)
1689		c_6(5)	c_5(14)
1690		c_0(10)	c_1(9)
1691	Status 47	SS(1)	SS(0)
1692		c_2(8)	c_3(7)
1693		c_5(13)	c_6(4)
1694		c_1(8)	c_0(9)
1695		c_3(6)	c_2(7)
1696		c_6(3)	c_5(12)
1697		c_0(8)	c_1(7)
1698		c_2(6)	c_3(5)
1699		c_5(11)	c_6(2)
1700		c_1(6)	c_0(7)
1701		c_3(4)	c_2(5)
1702		c_6(1)	c_5(10)
1703		c_0(6)	c_1(5)
1704		c_2(4)	c_3(3)
1705		c_5(9)	c_6(0)
1706		c_1(4)	c_0(5)
1707		c_3(2)	c_2(3)
1708		c_7(6)	c_5(8)
1709		c_0(4)	c_1(3)
1710		c_2(2)	c_3(1)
1711		c_5(7)	c_7(5)
1712		c_1(2)	c_0(3)
1713		c_3(0)	c_2(1)
1714		c_7(4)	c_5(6)
1715		c_0(2)	c_1(1)
1716		c_2(0)	c_4(14)

1717		c_5(5)	c_7(3)
1718		c_1(0)	c_0(1)
1719		c_4(13)	c_3(22)
1720		c_7(2)	c_5(4)
1721		c_0(0)	c_2(22)
1722		c_3(21)	c_4(12)
1723		c_5(3)	c_7(1)
1724		c_2(21)	c_1(22)
1725		c_4(11)	c_3(20)
1726	END of 18	c_7(0)	c_5(2)
1727	Status 48	SS(1)	SS(0)

10.5 Simple Terminator Data Unit

The next few pages give the order in which symbols shall be transmitted for the simple Terminator Data Unit. The simple Terminator Data Unit is transmitted at the end of a voice message.

Symbol	Field	Bit 1	Bit 0
0	Frame	0	1
1	Sync	0	1
2		0	1
3		0	1
4		0	1
5		1	1
6		0	1
7		0	1
8		1	1
9		1	1
10		0	1
11		0	1
12		1	1
13		1	1
14		1	1
15		1	1
16		0	1
17		1	1
18		0	1
19		1	1
20		1	1
21		1	1
22		1	1
23		1	1
24	Network	NAC(11)	NAC(10)
25	ID	NAC(9)	NAC(8)
26		NAC(7)	NAC(6)
27		NAC(5)	NAC(4)
28		NAC(3)	NAC(2)
29		NAC(1)	NAC(0)
30		DUID(3)	DUID(2)
31		DUID(1)	DUID(0)
32		BCH_parity(47)	BCH_parity(46)
33		BCH_parity(45)	BCH_parity(44)
34		BCH_parity(43)	BCH_parity(42)
35	Status 1	SS(1)	SS(0)
36		BCH_parity(41)	BCH_parity(40)
37		BCH_parity(39)	BCH_parity(38)
38		BCH_parity(37)	BCH_parity(36)
39		BCH_parity(35)	BCH_parity(34)
40		BCH_parity(33)	BCH_parity(32)
41		BCH_parity(31)	BCH_parity(30)
42		BCH_parity(29)	BCH_parity(28)
43		BCH_parity(27)	BCH_parity(26)
44		BCH_parity(25)	BCH_parity(24)
45		BCH_parity(23)	BCH_parity(22)
46		BCH_parity(21)	BCH_parity(20)
47		BCH_parity(19)	BCH_parity(18)
48		BCH_parity(17)	BCH_parity(16)
49		BCH_parity(15)	BCH_parity(14)
50		BCH_parity(13)	BCH_parity(12)
51		BCH_parity(11)	BCH_parity(10)
52		BCH_parity(9)	BCH_parity(8)
53		BCH_parity(7)	BCH_parity(6)
54		BCH_parity(5)	BCH_parity(4)
55		BCH_parity(3)	BCH_parity(2)
56		BCH_parity(1)	BCH_parity(0)
57	Nulls	0	0
58		0	0
59		0	0
60		0	0

61		0	0
62		0	0
63		0	0
64		0	0
65		0	0
66		0	0
67		0	0
68		0	0
69		0	0
70		0	0
71	Status 2	SS(1)	SS(0)

10.6 Terminator Data Unit with Link Control

The next few pages give the order in which symbols shall be transmitted for the Terminator Data Unit with a Link Control word. The Terminator Data Unit is transmitted at the end of a voice message or when a Link Control word is intended to be sent without any voice information.

Symbol	Field	Bit 1	Bit 0
0	Frame	0	1
1	Sync	0	1
2		0	1
3		0	1
4		0	1
5		1	1
6		0	1
7		0	1
8		1	1
9		1	1
10		0	1
11		0	1
12		1	1
13		1	1
14		1	1
15		1	1
16		0	1
17		1	1
18		0	1
19		1	1
20		1	1
21		1	1
22		1	1
23		1	1
24	Network	NAC(11)	NAC(10)
25	ID	NAC(9)	NAC(8)
26		NAC(7)	NAC(6)
27		NAC(5)	NAC(4)
28		NAC(3)	NAC(2)
29		NAC(1)	NAC(0)
30		DUID(3)	DUID(2)
31		DUID(1)	DUID(0)
32		BCH_parity(47)	BCH_parity(46)
33		BCH_parity(45)	BCH_parity(44)
34		BCH_parity(43)	BCH_parity(42)
35	Status 1	SS(1)	SS(0)
36		BCH_parity(41)	BCH_parity(40)
37		BCH_parity(39)	BCH_parity(38)
38		BCH_parity(37)	BCH_parity(36)
39		BCH_parity(35)	BCH_parity(34)
40		BCH_parity(33)	BCH_parity(32)
41		BCH_parity(31)	BCH_parity(30)
42		BCH_parity(29)	BCH_parity(28)
43		BCH_parity(27)	BCH_parity(26)
44		BCH_parity(25)	BCH_parity(24)
45		BCH_parity(23)	BCH_parity(22)
46		BCH_parity(21)	BCH_parity(20)
47		BCH_parity(19)	BCH_parity(18)
48		BCH_parity(17)	BCH_parity(16)
49		BCH_parity(15)	BCH_parity(14)
50		BCH_parity(13)	BCH_parity(12)
51		BCH_parity(11)	BCH_parity(10)
52		BCH_parity(9)	BCH_parity(8)
53		BCH_parity(7)	BCH_parity(6)
54		BCH_parity(5)	BCH_parity(4)
55		BCH_parity(3)	BCH_parity(2)
56		BCH_parity(1)	BCH_parity(0)
57	LC	LC_format(7)	LC_format(6)
58	Code	LC_format(5)	LC_format(4)

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

131		RS_parity_11(5)	RS_parity_11(4)
132		RS_parity_11(3)	RS_parity_11(2)
133		RS_parity_11(1)	RS_parity_11(0)
134		RS_parity_10(5)	RS_parity_10(4)
135		RS_parity_10(3)	RS_parity_10(2)
136		RS_parity_10(1)	RS_parity_10(0)
137		Extend_Golay_parity(11)	Extend_Golay_parity(10)
138		Extend_Golay_parity(9)	Extend_Golay_parity(8)
139		Extend_Golay_parity(7)	Extend_Golay_parity(6)
140		Extend_Golay_parity(5)	Extend_Golay_parity(4)
141		Extend_Golay_parity(3)	Extend_Golay_parity(2)
142		Extend_Golay_parity(1)	Extend_Golay_parity(0)
143	Status 4	SS(1)	SS(0)
144		RS_parity_9(5)	RS_parity_9(4)
145		RS_parity_9(3)	RS_parity_9(2)
146		RS_parity_9(1)	RS_parity_9(0)
147		RS_parity_8(5)	RS_parity_8(4)
148		RS_parity_8(3)	RS_parity_8(2)
149		RS_parity_8(1)	RS_parity_8(0)
150		Extend_Golay_parity(11)	Extend_Golay_parity(10)
151		Extend_Golay_parity(9)	Extend_Golay_parity(8)
152		Extend_Golay_parity(7)	Extend_Golay_parity(6)
153		Extend_Golay_parity(5)	Extend_Golay_parity(4)
154		Extend_Golay_parity(3)	Extend_Golay_parity(2)
155		Extend_Golay_parity(1)	Extend_Golay_parity(0)
156		RS_parity_7(5)	RS_parity_7(4)
157		RS_parity_7(3)	RS_parity_7(2)
158		RS_parity_7(1)	RS_parity_7(0)
159		RS_parity_6(5)	RS_parity_6(4)
160		RS_parity_6(3)	RS_parity_6(2)
161		RS_parity_6(1)	RS_parity_6(0)
162		Extend_Golay_parity(11)	Extend_Golay_parity(10)
163		Extend_Golay_parity(9)	Extend_Golay_parity(8)
164		Extend_Golay_parity(7)	Extend_Golay_parity(6)
165		Extend_Golay_parity(5)	Extend_Golay_parity(4)
166		Extend_Golay_parity(3)	Extend_Golay_parity(2)
167		Extend_Golay_parity(1)	Extend_Golay_parity(0)
168		RS_parity_5(5)	RS_parity_5(4)
169		RS_parity_5(3)	RS_parity_5(2)
170		RS_parity_5(1)	RS_parity_5(0)
171		RS_parity_4(5)	RS_parity_4(4)
172		RS_parity_4(3)	RS_parity_4(2)
173		RS_parity_4(1)	RS_parity_4(0)
174		Extend_Golay_parity(11)	Extend_Golay_parity(10)
175		Extend_Golay_parity(9)	Extend_Golay_parity(8)
176		Extend_Golay_parity(7)	Extend_Golay_parity(6)
177		Extend_Golay_parity(5)	Extend_Golay_parity(4)
178		Extend_Golay_parity(3)	Extend_Golay_parity(2)
179	Status 5	SS(1)	SS(0)
180		Extend_Golay_parity(1)	Extend_Golay_parity(0)
181		RS_parity_3(5)	RS_parity_3(4)
182		RS_parity_3(3)	RS_parity_3(2)
183		RS_parity_3(1)	RS_parity_3(0)
184		RS_parity_2(5)	RS_parity_2(4)
185		RS_parity_2(3)	RS_parity_2(2)
186		RS_parity_2(1)	RS_parity_2(0)
187		Extend_Golay_parity(11)	Extend_Golay_parity(10)
188		Extend_Golay_parity(9)	Extend_Golay_parity(8)
189		Extend_Golay_parity(7)	Extend_Golay_parity(6)
190		Extend_Golay_parity(5)	Extend_Golay_parity(4)
191		Extend_Golay_parity(3)	Extend_Golay_parity(2)
192		Extend_Golay_parity(1)	Extend_Golay_parity(0)
193		RS_parity_1(5)	RS_parity_1(4)
194		RS_parity_1(3)	RS_parity_1(2)
195		RS_parity_1(1)	RS_parity_1(0)
196		RS_parity_0(5)	RS_parity_0(4)
197		RS_parity_0(3)	RS_parity_0(2)
198		RS_parity_0(1)	RS_parity_0(0)
199		Extend_Golay_parity(11)	Extend_Golay_parity(10)
200		Extend_Golay_parity(9)	Extend_Golay_parity(8)
201		Extend_Golay_parity(7)	Extend_Golay_parity(6)
202		Extend_Golay_parity(5)	Extend_Golay_parity(4)

203		Extend_Golay_parity(3)	Extend_Golay_parity(2)
204		Extend_Golay_parity(1)	Extend_Golay_parity(0)
205	Nulls	0	0
206		0	0
207		0	0
208		0	0
209		0	0
210		0	0
211		0	0
212		0	0
213		0	0
214		0	0
215	Status 6	SS(1)	SS(0)

TIA