INTERNATIONAL STANDARD

ISO

22641

Second edition
2012-07-15

Space data and information transfer systems — TM (telemetry) synchronization and channel coding

Systèmes de transfert des informations et données spatiales — Synchronisation TM (télémesure) et codage de canal

COPYRIGHT PROTECTED DOCUMENT

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 22641 was prepared by the Consultative Committee for Space Data Systems (CCSDS) (as CCSDS 131.0-B-2, August 2011) and was adopted (without modifications except those stated in Clause 2 of this International Standard) by Technical Committee ISO/TC 20, *Aircraft and space vehicles*, Subcommittee SC 13, *Space data and information transfer systems*.

This second edition cancels and replaces the first edition (ISO 22641:2005), which has been technically revised.

Space data and information transfer systems — TM (telemetry) synchronization and channel coding

1 Scope

1.1 This International Standard specifies synchronization and channel coding schemes for use with the TM Space Data Link Protocol (according to ISO 22645:2005) or the AOS Space Data Link Protocol (according to ISO 22666:2007), which are intended to be used over space-to-ground or space-to-space communications links by space missions.

1.2 This International Standard defines synchronization and channel coding schemes in terms of

a) the services provided to the users of this specification,

b) data formats, and

c) the procedures performed to generate and process the data formats.

1.3 This International Standard does not specify

a) individual implementations or products,

b) the methods or technologies required to perform the procedures, or

c) the management activities required to configure and control the system.

1.4 The scope and field of application are furthermore detailed in subclauses 1.2 and 1.3 of the enclosed CCSDS publication.

2 Requirements

Requirements are the technical recommendations made in the following publication (reproduced on the following pages), which is adopted as an International Standard:

CCSDS 131.0-B-2, August 2011, *TM synchronization and channel coding*.

For the purposes of international standardization, the modifications outlined below shall apply to the specific clauses and paragraphs of publication CCSDS 131.0-B-2.

Pages i to v

This part is information which is relevant to the CCSDS publication only.

Page 1-5

Add the following information to the reference indicated:

[1]   Document CCSDS 132.0-B-1, September 2003, is equivalent to ISO 22645:2005.

[2]   Document CCSDS 732.0-B-2, July 2006, is equivalent to ISO 22666:2007.

Page E-1

Add the following information to the reference indicated:

[E2]   Document CCSDS 101.0-B-6, October 2002, is equivalent to ISO 11754:2003[1)] .

3   Revision of publication CCSDS 131.0-B-2

It has been agreed with the Consultative Committee for Space Data Systems that Subcommittee ISO/TC 20/SC 13 will be consulted in the event of any revision or amendment of publication CCSDS 131.0-B-2. To this end, NASA will act as a liaison body between CCSDS and ISO.

---

1)   Withdrawn.

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

TM SYNCHRONIZATION AND CHANNEL CODING

RECOMMENDED STANDARD

CCSDS 131.0-B-2

BLUE BOOK
August 2011

AUTHORITY

Issue:	Recommended Standard, Issue 2
Date:	August 2011
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

o   Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.

o   Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:

--   The **standard** itself.

--   The anticipated date of initial operational capability.

--   The anticipated duration of operational service.

o   Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This document is a technical Recommended Standard for use in developing synchronization and channel coding systems and has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The synchronization and channel coding concept described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

This Recommended Standard establishes a common framework and provides a common basis for the synchronization and channel coding schemes to be used by space missions with the TM or AOS Space Data Link Protocol (references [1] or [2]) over space-to-ground and space-to-space communications links. This Recommended Standard was developed by consolidating the specifications regarding synchronization and channel coding in older CCSDS Recommended Standards (references [E2] and [E3]).

This Recommended Standard does not change the major technical contents defined in references [E2] and [E3], but the presentation of the specification has been changed so that:

a) these schemes can be used to transfer any data over any space link in either direction;

b) all CCSDS space link protocols are specified in a unified manner;

c) the layered model matches the Open Systems Interconnection (OSI) Basic Reference Model (reference [3]).

Together with the change in presentation, a few technical specifications in references [E2] and [E3] have been changed in order to define all Space Data Link Protocols in a unified way. Also, some technical terms in references [E2] and [E3] have been changed in order to unify the terminology used in all the CCSDS Recommended Standards that define space link protocols and to define these schemes as general communications schemes. These changes are listed in annex H of this Recommended Standard.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

7

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.

−   United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 131.0-B-1	TM Synchronization and Channel Coding, Issue 1	September 2003	Original issue
CCSDS 131.0-B-2	TM Synchronization and Channel Coding, Recommended Standard, Issue 2	August 2011	Current issue. Changes from the previous issue are summarized below.

Changes not compatible with previous issue:

  − adds LDPC Code.

Changes compatible with previous issue:

  − adds Security, SANA, and Patent Considerations annex;

  − corrects and updates erroneous and obsolete material;

  − updates all coding sections for uniformity of presentation; rephrases specifications in CCSDS terse style, using 'shall' to specify binding and verifiable requirements; and differentiates non-normative text from normative text (see 1.6.2);

  − clearly differentiates the terms 'Codeblock' and 'Codeword', with the term 'Codeblock' confined to Reed-Solomon coding (see definitions in annex D).

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

The purpose of this Recommended Standard is to specify synchronization and channel coding schemes used with the TM Space Data Link Protocol (reference [1]) or the AOS Space Data Link Protocol (reference [2]). These schemes are to be used over space-to-ground or space-to-space communications links by space missions.

1.2 SCOPE

This Recommended Standard defines synchronization and channel coding schemes in terms of:

a) the services provided to the users of this specification;

b) data formats; and

c) the procedures performed to generate and process the data formats.

It does not specify:

a) individual implementations or products;

b) the methods or technologies required to perform the procedures; or

c) the management activities required to configure and control the system.

1.3 APPLICABILITY

This Recommended Standard applies to the creation of Agency standards and to the future data communications over space links between CCSDS Agencies in cross-support situations. This Recommended Standard includes comprehensive specification of the data formats and procedures for inter-Agency cross support. It is neither a specification of, nor a design for, real systems that may be implemented for existing or future missions.

The Recommended Standard specified in this document is to be invoked through the normal standards programs of each CCSDS Agency, and is applicable to those missions for which cross support based on capabilities described in this Recommended Standard is anticipated. Where mandatory capabilities are clearly indicated in sections of this Recommended Standard, they must be implemented when this document is used as a basis for cross support. Where options are allowed or implied, implementation of these options is subject to specific bilateral cross support agreements between the Agencies involved.

1.4 RATIONALE

The CCSDS believes it is important to document the rationale underlying the recommendations chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions.

1.5 DOCUMENT STRUCTURE

This document is divided into eleven numbered sections and seven annexes:

a) section 1 presents the purpose, scope, applicability and rationale of this Recommended Standard and lists the conventions, definitions, and references used throughout the document;

b) section 2 provides an overview of synchronization and channel coding;

c) section 3 specifies convolutional coding;

d) section 4 specifies Reed-Solomon coding;

e) section 5 concatenated coding;

f) section 6 specifies Turbo coding;

g) section 7 specifies low-density parity-check coding;

h) section 8 specifies the frame synchronization scheme;

i) section 9 specifies the Pseudo-Randomizer;

j) section 10 specifies the allowed lengths of Transfer Frames;

k) section 11 lists the managed parameters associated with synchronization and channel coding;

l) annex A defines the service provided to the users;

m) annex B discusses security issues related to TM Channel Coding;

n) annex C provides the generator matrix circulant table applicable to rate-223/255 LDPC coding (7.3);

o) annex D lists acronyms and terms used within this document;

p) annex E provides a list of informative references;

q) annex F provides information on transformation between the Berlekamp (dual basis) and Conventional representations;

r) annex G provides information on Reed-Solomon coefficients;

s) annex H lists the changes from relevant previously published CCSDS Recommended Standards (references [E2] and [E3]).

1.6 CONVENTIONS AND DEFINITIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from the Open System Interconnection (OSI) Basic Reference Model

This Recommended Standard makes use of a number of terms defined in reference [3]. The use of those terms in this Recommended Standard shall be understood in a generic sense; i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

a) Data Link Layer;

b) Physical Layer;

c) service;

d) service data unit.

1.6.1.2 Definitions from OSI Service Definition Conventions

This Recommended Standard makes use of a number of terms defined in reference [4]. The use of those terms in this Recommended Standard shall be understood in a generic sense; i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

a) indication;

b) primitive;

c) request;

d) service provider;

e) service user.

1.6.1.3 Definition of Terms Used in This Recommended Standard

For the purposes of this Recommended Standard, the following definitions apply. Many other terms that pertain to specific items are defined in the appropriate sections.

Mission Phase: a period of a mission during which specified communications characteristics are fixed. The transition between two consecutive mission phases may cause an interruption of the communications services.

     17

Physical Channel: a stream of bits transferred over a space link in a single direction.

space link: a communications link between a spacecraft and its associated ground system or between two spacecraft. A space link consists of one or more Physical Channels in one or both directions.

1.6.2 NOMENCLATURE

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words 'shall' and 'must' imply a binding and verifiable specification;

b) the word 'should' implies an optional, but desirable, specification;

c) the word 'may' implies an optional specification;

d) the words 'is', 'are', and 'will' imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.6.3 CONVENTIONS

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'Bit 0', the following bit is defined to be 'Bit 1', and so on up to 'Bit *N*-1'. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., 'Bit 0' (see figure 1-1).

BIT 0   BIT *N*-1

N-BIT DATA FIELD

FIRST BIT TRANSMITTED = MSB
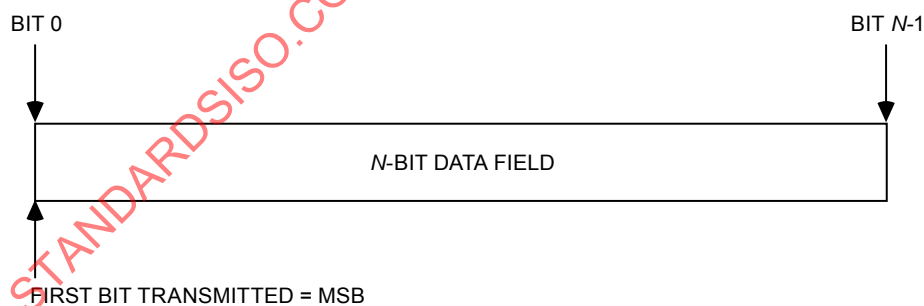
Figure 1-1: Bit Numbering Convention

In accordance with standard data-communications practice, data fields are often grouped into 8-bit 'words' which conform to the above convention. Throughout this Recommended Standard, such an 8-bit word is called an 'octet'.

The numbering for octets within a data structure starts with '0'.

1.7 PATENTED TECHNOLOGIES

The CCSDS draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning Turbo Coding (section 6) and Low-Density Parity-Check Coding (section 7).

The CCSDS takes no position concerning the evidence, validity, and scope of these patent rights.

The holders of these patent rights have assured the CCSDS that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with CCSDS. Information can be obtained from the CCSDS Secretariat at the address indicated on page i. Contact information for the holders of these patent rights is provided in annex B.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. The CCSDS shall not be held responsible for identifying any or all such patent rights.

1.8 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommended Standards.

[1] *TM Space Data Link Protocol.* Recommendation for Space Data Systems Standards, CCSDS 132.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[2] *AOS Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 732.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, July 2006.

[3] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1. 2nd ed. Geneva: ISO, 1994.

[4] *Information Technology—Open Systems Interconnection—Basic Reference Model— Conventions for the definition of OSI services*. International Standard, ISO/IEC 10731:1994. Geneva: ISO, 1994.

[5] *Radio Frequency and Modulation Systems—Part 1: Earth Stations and Spacecraft*. Recommendation for Space Data Systems Standards, CCSDS 401.0-B. Blue Book. Washington, D.C.: CCSDS, March 2003.

NOTE   –   Informative references are listed in annex E.

(Blank Page)

2   OVERVIEW

2.1   ARCHITECTURE

Figure 2-1 illustrates the relationship of this Recommended Standard to the Open Systems Interconnection reference model (reference [3]). Two sublayers of the Data Link Layer are defined for CCSDS space link protocols. The TM and AOS Space Data Link Protocols specified in references [1] and [2], respectively, correspond to the Data Link Protocol Sublayer, and provide functions for transferring data using the protocol data unit called the Transfer Frame. The Synchronization and Channel Coding Sublayer provides additional functions necessary for transferring Transfer Frames over a space link. These functions are error-control coding/decoding, Transfer Frame delimiting/synchronizing, and bit transition generation/removal.
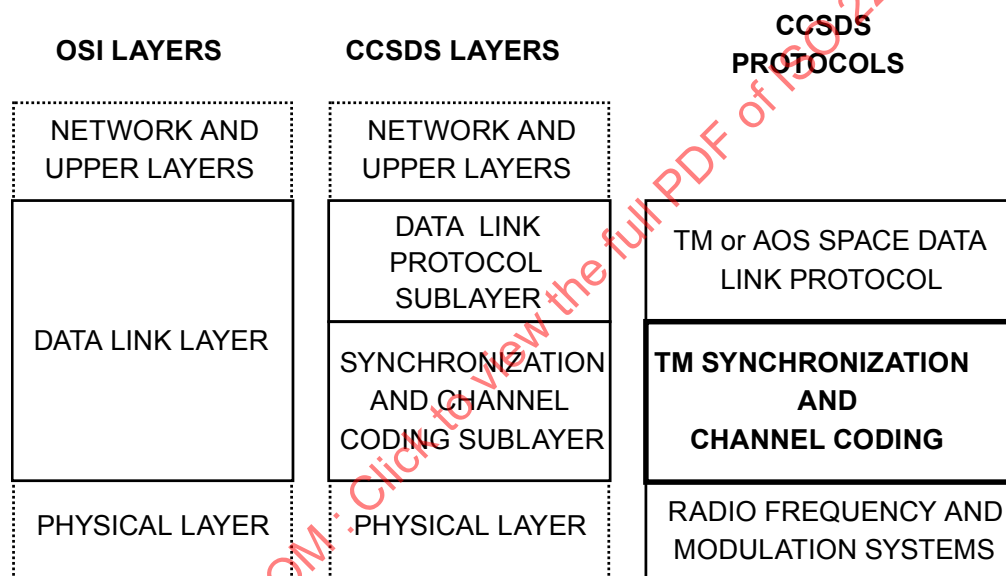
Figure 2-1: Relationship with OSI Layers

2.2   SUMMARY OF FUNCTIONS

2.2.1   GENERAL

The Synchronization and Channel Coding Sublayer provides the following three functions for transferring Transfer Frames over a space link:

    a) error-control coding, including frame validation;

    b) synchronization; and

    c) pseudo-randomizing.

2.2.2 ERROR-CONTROL CODING

This Recommended Standard specifies the following four types of error-control coding:

a) convolutional coding (section 3);

b) Reed-Solomon coding (section 4);

c) Turbo coding (section 5);

d) Low-Density Parity-Check (LDPC) coding (section 7).

One of the convolutional codes described in section 3 alone may be satisfactory depending on performance requirements.

For Physical Channels, which are bandwidth-constrained and cannot tolerate the increase in bandwidth required by the basic convolutional code specified in 3.3, the punctured convolutional codes specified in 3.4 have the advantage of smaller bandwidth expansion.

Alternatively, the Reed-Solomon codes and the high rate LDPC code specified in sections 4 and 7 also have the advantage of smaller bandwidth expansion and have the capability to indicate the presence of uncorrectable errors. Where a greater coding gain is needed than can be provided by a convolutional code or Reed-Solomon code alone, a concatenation of a convolutional code as the inner code with a Reed-Solomon code as the outer code may be used for improved performance.

The Turbo codes specified in section 5 or the LDPC codes specified in section 7 may be used to obtain even greater coding gain where the environment permits.

NOTES

1   In this Recommended Standard, the characteristics of the codes are specified only to the extent necessary to ensure interoperability and cross-support. The specification does not attempt to quantify the relative coding gain or the merits of each approach discussed, nor does it specify the design requirements for encoders or decoders.

2   The domains of applicability for the codes specified in this document are delineated in *Mission Profiles for TM Synchronization and Channel Coding* (reference [E5]).

2.2.3 FRAME VALIDATION

After decoding is performed, the upper layers at the receiving end also need to know whether or not each decoded Transfer Frame can be used as a valid data unit; i.e., an indication of the quality of the received frame is needed. This function is called Frame Validation. The Reed-Solomon and LDPC decoders can determine, with a very high probability, whether or not they can correctly decode a Transfer Frame. Therefore, the Reed-Solomon and LDPC codes are also

used for Frame Validation. When the Reed-Solomon or LDPC codes are not used, the Frame Error Control Field defined in references [1] or [2] is used for Frame Validation.

2.2.4 SYNCHRONIZATION

This Recommended Standard specifies a method for synchronizing Transfer Frames using an Attached Sync Marker (ASM) (see section 8).

The ASM may also be used for resolution of data ambiguity (sense of '1' and '0') if data ambiguity is not resolved by the modulation method used in the Physical Layer.

2.2.5 PSEUDO-RANDOMIZING

This Recommended Standard specifies a pseudo-randomizer to improve several aspects of the telemetry link that aid receiver acquisition, bit synchronization (see section 9), convolutional code synchronization, and proper Frame Validation (see 2.2.3).

2.3 INTERNAL ORGANIZATION OF SUBLAYER

2.3.1 SENDING END

Figure 2-2 shows the internal organization of the Synchronization and Channel Coding Sublayer of the sending end. This figure identifies functions performed by the sublayer and shows logical relationships among these functions. The figure is not intended to imply any hardware or software configuration in a real system. Depending on the options actually used for a mission, not all of the functions may be present in the sublayer.

At the sending end, the Synchronization and Channel Coding Sublayer accepts Transfer Frames of fixed length from the Data Link Protocol Sublayer (see figure 2-1), performs functions selected for the mission, and delivers a continuous and contiguous stream of channel symbols to the Physical Layer.

Data Link Protocol Sublayer

↓ Transfer Frames

Reed-Solomon, Turbo, or LDPC Encoding (optional)

↓ Codeblocks, Codewords,
or Transfer Frames

Pseudo-Random Sequence Generation (optional)

↓ (Randomized) Codeblocks,
Codewords, or Transfer Frames

Attachment of Attached Sync Marker

↓ Channel Access Data Units

Convolutional Encoding (optional)

↓ Channel Symbols

Physical Layer

Figure 2-2: Internal Organization of the Sublayer at the Sending End

2.3.2   RECEIVING END

Figure 2-3 shows the internal organization of the Synchronization and Channel Coding Sublayer of the receiving end. This figure identifies functions performed by the sublayer and shows logical relationships among these functions. The figure is not intended to imply any hardware or software configuration in a real system (e.g., some implementations perform frame synchronization before convolutional decoding when convolutional code rate 1/2 is used) . Depending on the options actually used for a mission, not all of the functions may be present in the sublayer.

At the receiving end, the Synchronization and Channel Coding Sublayer accepts a continuous and contiguous stream of channel symbols from the Physical Layer, performs functions selected for the mission, and delivers Transfer Frames to the Data Link Protocol Sublayer.

Data Link Protocol Sublayer

↑ Transfer Frames

| Reed-Solomon, Turbo, or LDPC Decoding (optional) |

↑ Codeblocks, Codewords, or
Transfer Frames

| Pseudo-Random Sequence Removal (optional) |

↑ (Randomized) Codeblocks,
Codewords, or Transfer Frames

| Frame Synchronization |

↑ Channel Access Data Units

| Convolutional Decoding (optional) |

↑ Channel Symbols

Physical Layer
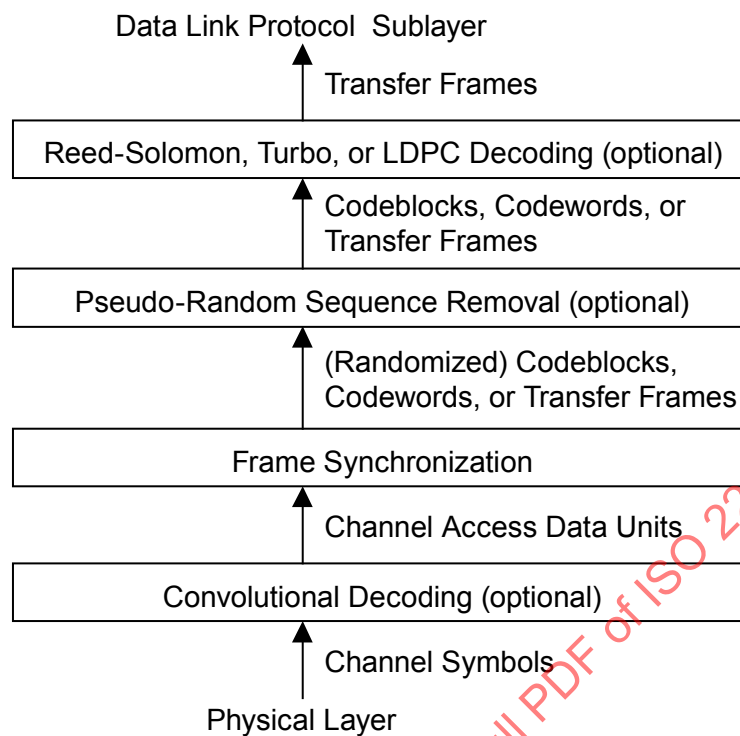
Figure 2-3: Internal Organization of the Sublayer at the Receiving End

(Blank Page)

3 CONVOLUTIONAL CODING

3.1 OVERVIEW

The basic convolutional code is a rate (r) 1/2, constraint-length (K) 7 transparent code which is well suited for channels with predominantly Gaussian noise. This code is defined in 3.3. When this code is punctured according to 3.4, higher code rates may be achieved although with lower error correcting performance.

Puncturing allows a single code rate of either 2/3, 3/4, 5/6 or 7/8 to be selected. The four different puncturing schemes allow selection of the most appropriate level of error correction and symbol rate for a given service or data rate.

3.2 GENERAL

3.2.1 ATTACHED SYNC MARKER

The Attached Sync Marker used with convolutional code shall be the 32-bit pattern specified in 8.2, and it shall always be inserted before performing convolutional encoding.

3.2.2 DATA RANDOMIZATION

The pseudo-randomizer defined in section 9 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE – An inverter is specified with the basic convolutional code to assure sufficient bit transitions to keep receiver symbol synchronizers in lock, when used with Binary Phase Shift Keying (BPSK) modulation. Sufficient bit transitions cannot be guaranteed by the inverter alone if some multiplexing schemes are used, e.g., with Quadrature Phase Shift Keying (QPSK) modulation, or if a punctured convolutional code is used. There are also data patterns for which convolutional code synchronization cannot be determined. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

3.2.3 FRAME VALIDATION

When TM or AOS Transfer Frames are used, the Frame Error Control Field (FECF) specified in references [1] and [2] shall be used to validate the Transfer Frame, unless the convolutional code is concatenated with an outer Reed-Solomon code (see section 4).

NOTE – If the decoder's correction capability is exceeded, undetected bursts of errors may appear in the output.

3.2.4   QUANTIZATION

Soft bit decisions with at least three-bit quantization should be used whenever constraints (such as complexity of decoder) permit.

3.3   BASIC CONVOLUTIONAL CODE SPECIFICATION

3.3.1   The basic convolutional code shall be the non-systematic code with the following characteristics:

(1) Nomenclature: Convolutional code with maximum-likelihood decoding.

(2) Code rate (r):   1/2 bit per symbol.

(3) Constraint length (K):   7 bits.

(4) Connection vectors: G1 = 1111001 (171 octal); G2= 1011011(133 octal).

(5) Symbol inversion: On output path of G2.

NOTE   –   An encoder block diagram is shown in figure 3-1. When a single encoder is used, G2 inversion provides no benefit to data randomization when even-order modulations higher than BPSK are used. G2 inversion does provide value when coding is done after channel splitting and with separate encoders on each channel.

3.3.2   The output symbol sequence shall be: $C_1(1)$, $\overline{C_2(1)}$, $C_1(2)$, $\overline{C_2(2)}$. . . .

3.3.3   When suppressed-carrier modulation systems are used:

a) Non-Return-to-Zero-Mark (NRZ-M) or Non-Return-to-Zero-Level (NRZ-L) may be used as a modulating waveform.

b) If the user contemplates differential encoding, i.e., conversion of his modulating waveform from NRZ-L to NRZ-M, such conversion should be performed at the input to the convolutional encoder.

NOTES

1   Since the convolutional codes are transparent, differential encoding can be used before the convolutional encoder to help phase ambiguity resolution and, correspondingly, the conversion at the receiving end from NRZ-M to NRZ-L should be performed at the output of the convolutional decoder. Differential encoding after the convolutional encoder is not advised because it introduces considerable loss of performance. It also would require differential detection, which is more complex with soft symbols.

2   When a fixed pattern (the fixed part of the convolutionally encoded Attached Sync Marker) in the symbol stream is used to provide node synchronization for the convolutional decoder, any modification of the pattern resulting from the modulating waveform conversion needs to be accounted for.
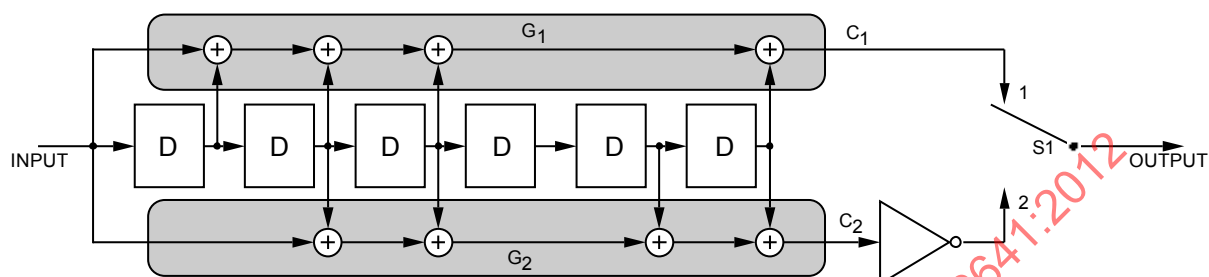
NOTES:

1. $\boxed{D}$ = SINGLE BIT DELAY.

2.   FOR EVERY INPUT BIT, TWO SYMBOLS ARE GENERATED BY COMPLETION OF A CYCLE FOR S1: POSITION 1, POSITION 2.

3.   S1 IS IN THE POSITION SHOWN (1) FOR THE FIRST SYMBOL ASSOCIATED WITH AN INCOMING BIT.

4. $\oplus$ = MODULO-2 ADDER.

5. $\longrightarrow\!\!\!\!\!\triangleright\!\!\circ\!-$ = INVERTER.

Figure 3-1: Basic Convolutional Encoder Block Diagram

29

3.4 PUNCTURED CONVOLUTIONAL CODES

3.4.1 The punctured convolutional code shall have the following characteristics:

(1) Nomenclature: Punctured convolutional code with maximum-likelihood decoding.

(2) Code rate (r): 1/2, punctured to 2/3, 3/4, 5/6 or 7/8.

(3) Constraint length (K): 7 bits.

(4) Connection vectors: G1 = 1111001 (171 octal); G2 = 1011011 (133 octal).

(5) Symbol inversion: None.

3.4.2 The puncturing patterns for each of the punctured convolutional code rates shall be as specified in table 3-1.

NOTE – Figure 3-2 depicts the punctured encoding scheme.
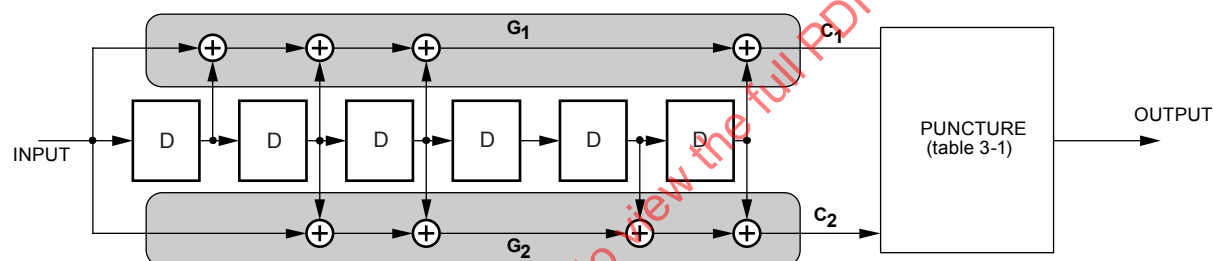
Figure 3-2: Punctured Encoder Block Diagram

Table 3-1: Puncture Code Patterns for Convolutional Code Rates

Puncturing Pattern 1 = transmitted symbol 0 = non-transmitted symbol	Code Rate	Output Sequence $C_1(t)$, $C_2(t)$ denote values at bit time t
C_1: 1 0 C_2: 1 1	2/3	$C_1(1)$ $C_2(1)$ $C_2(2)$ ...
C_1: 1 0 1 C_2: 1 1 0	3/4	$C_1(1)$ $C_2(1)$ $C_2(2)$ $C_1(3)$ ...
C_1: 1 0 1 0 1 C_2: 1 1 0 1 0	5/6	$C_1(1)$ $C_2(1)$ $C_2(2)$ $C_1(3)$ $C_2(4)$ $C_1(5)$ ...
C_1: 1 0 0 0 1 0 1 C_2: 1 1 1 1 0 1 0	7/8	$C_1(1)$ $C_2(1)$ $C_2(2)$ $C_2(3)$ $C_2(4)$ $C_1(5)$ $C_2(6)$ $C_1(7)$ ...

4 REED-SOLOMON CODING

4.1 OVERVIEW

The Reed-Solomon (R-S) codes defined in this section are powerful burst error correcting codes. One of two different error-correcting options may be chosen. For maximum performance (at the expense of accompanying overhead) the E=16 option can correct 16 R-S symbols in error per codeword. For lower overhead (with reduced performance) the E=8 option can correct 8 R-S symbols per codeword. The Reed-Solomon code may be used alone, and as such it provides an excellent forward error correction capability in a burst-noise channel. However, should the Reed-Solomon code alone not provide sufficient coding gain, it may be concatenated with the convolutional code defined in section 3. Used this way, the Reed-Solomon code is the *outer code*, while the convolutional code is the *inner code*.

4.2 GENERAL

4.2.1 DATA RANDOMIZATION

The pseudo-randomizer defined in section 9 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE – The recommended Reed-Solomon codes, by themselves, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Because of the quasi-cyclic nature of these codes, undetected decoding errors may result from incorrect codeblock synchronization. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

4.2.2 FRAME VALIDATION

The Frame Error Control Field (FECF) specified in references [1] and [2] is optional. The system designer may choose to use it for additional codeblock validation, particularly with the E=8 code.

NOTE – The Reed-Solomon code with E=16 has an extremely low undetected error rate, and that with E=8 has an undetected error rate low enough for some applications. Therefore the R-S decoder may be used alone to validate the codeblock, and consequently the contained TM Transfer Frame (reference [1]) or AOS Transfer Frame (reference [2]).

4.3 SPECIFICATION

4.3.1 PARAMETERS

The parameters of the selected Reed-Solomon (R-S) code are as follows:

a) *J* shall be 8 bits per R-S symbol.

b) *E* shall be 16 or 8 R-S symbols.

NOTE – *E* is the Reed-Solomon error correction capability, in symbols, within a R-S codeword.

4.3.2 GENERAL CHARACTERISTICS

The code shall conform to the following general characteristics:

a) *J*, *E*, and *I* (the depth of interleaving) are independent parameters.

b) $n = 2^J - 1 = 255$ symbols per R-S codeword.

c) $2E$ is the number of R-S symbols among *n* symbols of an R-S codeword representing parity checks.

d) $k = n - 2E$ is the number of R-S symbols among *n* R-S symbols of an R-S codeword representing information.

4.3.3 FIELD GENERATOR POLYNOMIAL

The field generator polynomial shall be:

$$F(x) = x^8 + x^7 + x^2 + x + 1$$

over GF(2).

4.3.4 CODE GENERATOR POLYNOMIAL

The code generator polynomial shall be:

$$g(x) = \prod_{j=128-E}^{127+E} (x - \alpha^{11j}) \qquad = \sum_{i=0}^{2E} G_i x^i$$

over GF(2^8), where F(α) = 0.

NOTES

1   It should be recognized that α^{11} is a primitive element in GF(2^8) and that F(*x*) and g(*x*) characterize a (255,223) Reed-Solomon code when *E* = 16 and a (255,239) Reed-Solomon code when *E* = 8.

2   The selected code is a systematic code. This results in a systematic codeblock.

4.3.5 SYMBOL INTERLEAVING

4.3.5.1   The allowable values of interleaving depth are *I*=1, 2, 3, 4, 5, and 8.

NOTE   –   *I*=1 is equivalent to the absence of interleaving.

4.3.5.2   The interleaving depth shall normally be fixed on a Physical Channel for a Mission Phase.

NOTE   –   Discussion of symbol interleaving is contained in 4.4.1.

4.3.6 MAXIMUM CODEBLOCK LENGTH

The maximum codeblock length, in R-S symbols, shall be determined by the following equation:

$$L_{\max} = nI = (2^J - 1)I = 255I$$

4.3.7 SHORTENED CODEBLOCK LENGTH

4.3.7.1   A shortened codeblock length may be used to accommodate frame lengths smaller than the maximum.

NOTE   –   However, since the Reed-Solomon code is a block code, the decoder must always operate on a full block basis.

4.3.7.2   To achieve a full codeblock, 'virtual fill' shall be added to make up the difference between the shortened block and the maximum codeblock length.

NOTES

1   The characteristics and limitations of virtual fill are covered in 4.3.8.2.

2   Since the virtual fill is not transmitted, both encoder and decoder need to be set to insert it with the proper length for the encoding and decoding processes to be carried out properly.

4.3.7.3   When an encoder (initially cleared at the start of a block) receives *kI*–*Q* symbols representing information (where *Q*, representing fill, is a multiple of *I*, and is less than *kI*), 2*EI* check symbols shall be computed over *kI* symbols, of which the leading *Q* symbols shall be treated as all-zero symbols.

NOTE   –   A (*nI*–*Q*, *kI*–*Q*) shortened codeblock results.

4.3.7.4   The leading *Q* symbols (all zeros) of the resulting shortened codeblock shall be neither entered into the encoder nor transmitted.

NOTE – Shortening the transmitted codeblock length in this way changes the overall performance to a degree dependent on the amount of virtual fill used. Since it incorporates no virtual fill, the maximum codeblock length allows full performance. In addition, as virtual fill in a codeblock is increased (at a specific bit rate), the number of codeblocks per unit time that the decoder must handle increases. Therefore, care should be taken so that the maximum operating speed of the decoder (codeblocks per unit time) is not exceeded.

4.3.8   REED-SOLOMON CODEBLOCK PARTITIONING AND VIRTUAL FILL

4.3.8.1   Parts of the partitioned Reed-Solomon codeblock (see figure 4-1) are defined as follows:

a) The **Reed-Solomon Check Symbols** shall consist of the trailing $2EI$ symbols ($2EIJ$ bits) of the codeblock.

NOTES

1   As an example, when $E = 16$ and $k = 223$, for $I=5$ this is always 1280 bits.

2   The **Transfer Frame** is defined by the TM Space Data Link Protocol (reference [1]) or the AOS Space Data Link Protocol (reference [2]). For constraints on the length of the Transfer Frame, see section 10.

b) The **Attached Sync Marker** used with R-S code

   1) shall be the 32-bit pattern specified in section 8;

   2) shall precede the Transmitted Codeblock.

NOTE –   Frame synchronizers should therefore be set to expect a marker at every Transmitted Codeblock + 32 bits.

c) The **Transmitted Codeblock** shall consist of the Transfer Frame (without the 32-bit sync marker) and R-S check symbols.

NOTES

1   The Transmitted Codeblock is the received data entity physically fed into the R-S decoder. (As an example, when $E = 16$ and $k = 223$, using $I=5$ and no virtual fill, the length of the transmitted codeblock will be 10,200 bits; if virtual fill is used, it will be incrementally shorter, depending on the amount used.)

2   The **Logical Codeblock** is the logical data entity operated upon by the R-S decoder. It can have a different length than the transmitted codeblock because it accounts for the amount of virtual fill that was introduced. (As an example, when $E = 16$ and $k = 223$, for $I=5$ the logical codeblock always appears to have exactly 10,200 bits in length.)

3   The R-S codeblock is partitioned as shown in figure 4-1.
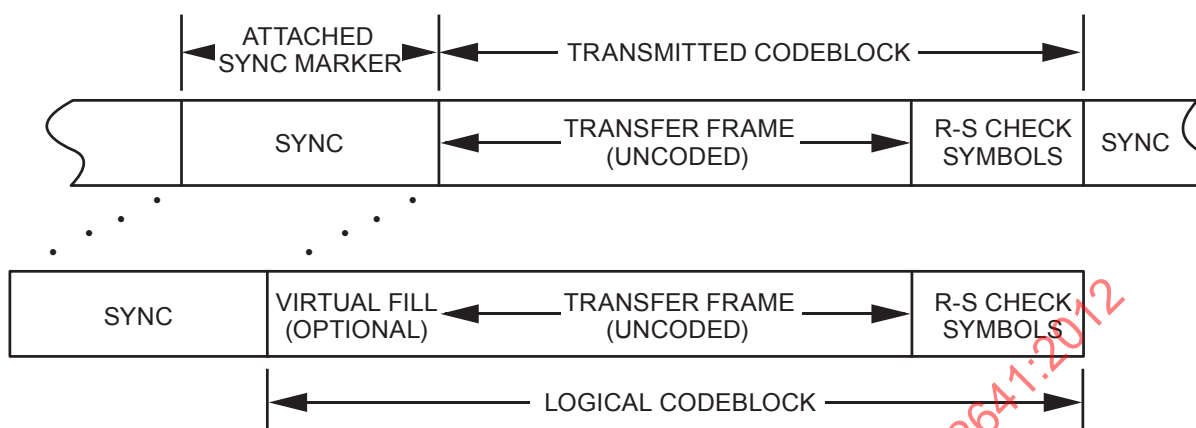
Figure 4-1: Reed-Solomon Codeblock Partitioning

4.3.8.2   Virtual fill shall be used to logically complete the codeblock. If used, virtual fill shall:

a)   consist of all zeros;

b)   not be transmitted;

c)   not change in length for a Mission Phase on a particular Physical Channel;

d)   be inserted only at the beginning of the codeblock (i.e., after the attached sync marker but before the beginning of the transmitted codeblock);

e)   be inserted only in integer multiples of $8I$ bits.

4.3.9   DUAL BASIS REPRESENTATION

4.3.9.1   Dual basis representation shall be used.

4.3.9.2   The order of transmission shall be **dual basis** eight-bit string $z_0, z_1, \ldots, z_7$ (i.e., with z_0 transmitted first).

4.3.9.3   The relationship between the two representations shall conform to the following two equations:

35

$$[z_0, \ldots, z_7] = [u_7, \ldots, u_0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

and

$$[u_7, \ldots, u_0] = [z_0, \ldots, z_7] \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

NOTES

1   Discussion of dual basis representation is contained in 4.4.2.

2   Further information relating the dual basis (Berlekamp) and conventional representations is given in annex F. Also included is a recommended scheme for permitting the symbols generated in a conventional encoder to be transformed to meet the symbol representation required by this document.

4.3.10 CODEBLOCK SYNCHRONIZATION

Codeblock synchronization of the Reed-Solomon decoder shall be achieved by synchronization of the Attached Sync Marker associated with each codeblock. (See section 8.)

NOTE –   At the receiving end, the ambiguity between true and complemented data must be resolved so that only true data is provided to the Reed-Solomon decoder. Data in NRZ-L form is normally resolved using the 32-bit Attached Sync Marker, while NRZ-M data is self-resolving.

4.4   DISCUSSION

4.4.1   SYMBOL INTERLEAVING

Symbol interleaving is accomplished in a manner functionally described with the aid of figure 4-2. (It should be noted that this functional description does not necessarily correspond to the physical implementation of an encoder.)
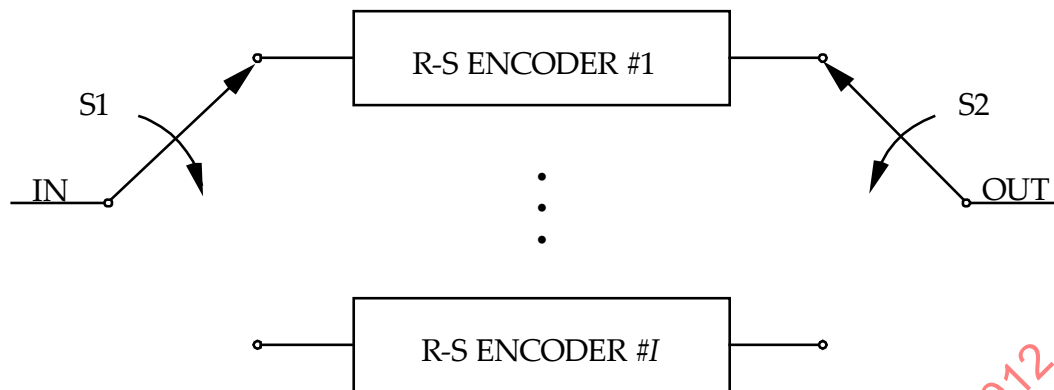
Figure 4-2: Functional Representation of R-S Interleaving

Data bits to be encoded into a single Reed-Solomon Codeblock enter at the port labeled 'IN'. Switches S1 and S2 are synchronized together and advance from encoder to encoder in the sequence $1,2, \ldots, I, 1,2, \ldots, I, \ldots$, spending one R-S symbol time (8 bits) in each position.

One codeblock will be formed from kI R-S symbols entering 'IN'. In this functional representation, a space of $2EI$ R-S symbols in duration is required between each entering set of kI R-S information symbols.

Because of the action of S1, each encoder accepts k of these symbols, with each symbol spaced I symbols apart (in the original stream). These k symbols are passed directly to the output of each encoder. The synchronized action of S2 reassembles the symbols at the port labeled 'OUT' in the same way as they entered at 'IN'.

Following this, each encoder outputs its $2E$ check symbols, one symbol at a time, as it is sampled in sequence by S2.

If, for $I=5$, the original symbol stream is

$$d_1^1 \ldots d_1^5 \, d_2^1 \ldots d_2^5 \ldots d_k^1 \ldots d_k^5 \quad [2E \times 5 \text{ spaces}]$$

then the output is the same sequence with the $[2E \times 5 \text{ spaces}]$ filled by the $[2E \times 5]$ check symbols as shown below:

$$p_1^1 \ldots p_1^5 \ldots p_{2E}^1 \ldots p_{2E}^5$$

where

$$d_1^i \, d_2^i \ldots d_K^i \, p_1^i \ldots p_{2E}^i$$

is the R-S codeword produced by the ith encoder. If q virtual fill symbols are used in each codeword, then replace k by $(k - q)$ in the above discussion.

With this method of interleaving, the original kI consecutive information symbols that entered the encoder appear unchanged at the output of the encoder with $2EI$ R-S check symbols appended.

4.4.2 DUAL BASIS SYMBOL REPRESENTATION AND ORDERING FOR TRANSMISSION

Each 8-bit Reed-Solomon symbol is an element of the finite field GF(256). Since GF(256) is a vector space of dimension 8 over the binary field GF(2), the actual 8-bit representation of a symbol is a function of the particular basis that is chosen.

One basis for GF(256) over GF(2) is the set ($1, \alpha^1, \alpha^2, \ldots, \alpha^7$). This means that any element of GF(256) has a representation of the form

$$u_7 \alpha^7 + u_6 \alpha^6 + \ldots + u_1 \alpha^1 + u_0 \alpha^0$$

where each u_i is either a zero or a one.

Another basis over GF(2) is the set ($1, \beta^1, \beta^2, \ldots, \beta^7$) where $\beta = \alpha^{117}$. To this basis there exists a so-called 'dual basis' ($\ell_0, \ell_1, \ldots, \ell_7$). It has the property that

$$\mathrm{Tr}(\ell_i \beta^j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

for each $j = 0, 1, \ldots, 7$. The function $\mathrm{Tr}(z)$, called the 'trace', is defined by

$$\mathrm{Tr}(z) = \sum_{K=0}^{7} z^{2^k}$$

for each element z of GF(256). Each Reed-Solomon symbol can also be represented as

$$z_0 \ell_0 + z_1 \ell_1 + \ldots + z_7 \ell_7$$

where each z_i is either a zero or a one.

5 CONCATENATED CODING

5.1 Concatenated codes shall consist of a combination of a Reed-Solomon code defined in section 4 with one of the convolutional codes defined in section 3.

5.2 The Reed-Solomon code shall be the outer code, and the convolutional code shall be the inner code.

(Blank Page)

6   TURBO CODING

6.1   OVERVIEW

Turbo codes are binary block codes with large codewords (hundreds or thousands of bits). Turbo codes may be used to obtain even greater coding gains than those provided by concatenated coding systems. They are systematic and inherently non-transparent.

6.2   GENERAL

6.2.1   DATA RANDOMIZATION

The pseudo-randomizer defined in section 9 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE –   The recommended Turbo codes, by themselves, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

6.2.2   FRAME VALIDATION

When Turbo codes are used with TM or AOS Transfer Frames, the Frame Error Control Field (FECF) specified in references [1] or [2], respectively, shall be used to validate the Transfer Frame.

NOTE –   While providing outstanding coding gain, Turbo codes may still leave some undetected errors in the decoded output.

6.3   SPECIFICATION

NOTE –   A Turbo encoder is a combination of two simple encoders. The input is a frame of k information bits. The two component encoders generate parity symbols from two simple recursive convolutional codes, each with a small number of states. The information bits are also sent uncoded. A key feature of Turbo codes is an interleaver, which permutes bit-wise the original k information bits before input to the second encoder.

The recommended Turbo code is a systematic code that shall conform to the following specifications:

a)   Code type shall be systematic parallel concatenated Turbo code.

b)   Number of component codes shall be two (plus an uncoded component to make the code systematic).

CCSDS 131.0-B-2   Page 6-1   August 2011

c) Type of component codes shall be recursive convolutional codes.

d) Number of states of each convolutional component code shall be 16.

e) Nominal code rates shall be $r = 1/2, 1/3, 1/4$, or $1/6$ (selectable).

NOTE – Because of 'trellis termination' symbols (see 6.3j)), the true code rates (defined as the ratios of the information block lengths to the codeblock lengths in table 6-2) are slightly smaller than the nominal code rates. In this Recommended Standard, the term 'code rate' always refers to the nominal code rates, $r = 1/2, 1/3, 1/4$, or $1/6$.

f) The specified information block lengths k shall be those specified in table 6-1. The corresponding codeblock lengths in bits, $n=(k+4)/r$, for the specified code rates shall be those specified in table 6-2.

NOTE – Information block lengths are chosen for compatibility with the corresponding Reed-Solomon interleaving depths, also shown in table 6-1.

Table 6-1: Specified Information Block Lengths

Information block length k, bits	Corresponding Reed-Solomon interleaving depth I	Notes
1784 (=223 × 1 octets)	1	For very low data rates or low latency
3568 (=223 × 2 octets)	2	
7136 (=223 × 4 octets)	4	
8920 (=223 × 5 octets)	5	For highest coding gain

Table 6-2: Codeword Lengths for Supported Code Rates (Measured in Bits)

Information block length k	Codeword length n			
	rate 1/2	rate 1/3	rate 1/4	rate 1/6
1784	3576	5364	7152	10728
3568	7144	10716	14288	21432
7136	14280	21420	28560	42840
8920	17848	26772	35696	53544

g) Turbo code permutation for each specified block length k shall conform to a particular reordering of the integers 1, 2, . . ., k as generated by the following algorithm.

NOTE – The interleaver is a fundamental component of the Turbo encoding and decoding process. The interleaver for Turbo codes is a fixed bit-by-bit permutation of the entire block of data. Unlike the symbol-by-symbol rectangular interleaver used with Reed-Solomon codes, the Turbo code permutation scrambles individual bits and resembles a randomly selected permutation in its lack of apparent orderliness.

1) k shall be expressed as $k=k_1 k_2$; the parameters k_1 and k_2 for the specified block sizes shall be selected from table 6-3.

Table 6-3: Parameters k_1 and k_2 for Specified Information Block Lengths

Information block length	k_1	k_2
1784	8	223
3568	8	223×2
7136	8	223×4
8920	8	223×5

2) The following operations shall be performed for $s=1$ to $s=k$ to obtain permutation numbers $\pi(s)$:

$$p_1 = 31;\ p_2 = 37;\ p_3 = 43;\ p_4 = 47;\ p_5 = 53;\ p_6 = 59;\ p_7 = 61;\ p_8 = 67$$

NOTE – In the equation below, $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, and p_q denotes one of the following eight prime integers:

$$m = (s-1) \bmod 2$$

$$i = \left\lfloor \frac{s-1}{2\,k_2} \right\rfloor$$

$$j = \left\lfloor \frac{s-1}{2} \right\rfloor - i\,k_2$$

$$t = (19i + 1) \bmod \frac{k_1}{2}$$

$$q = t \bmod 8 + 1$$

$$c = (p_q\,j + 21m) \bmod k_2$$

$$\pi(s) = 2\left(t + c\,\frac{k_1}{2} + 1\right) - m$$

The permutation numbers shall be interpreted such that the sth bit read out on line 'in b' in figure 6-2 is the $\pi(s)$th bit of the input information block, as shown in figure 6-1.
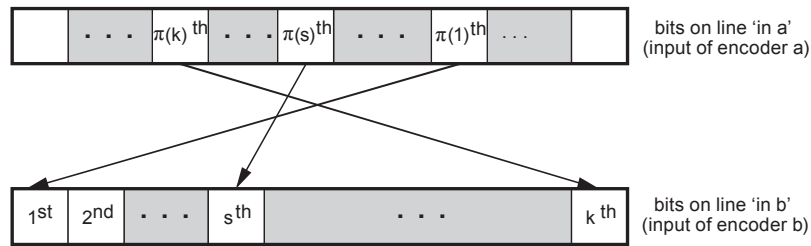
Figure 6-1: Interpretation of Permutation
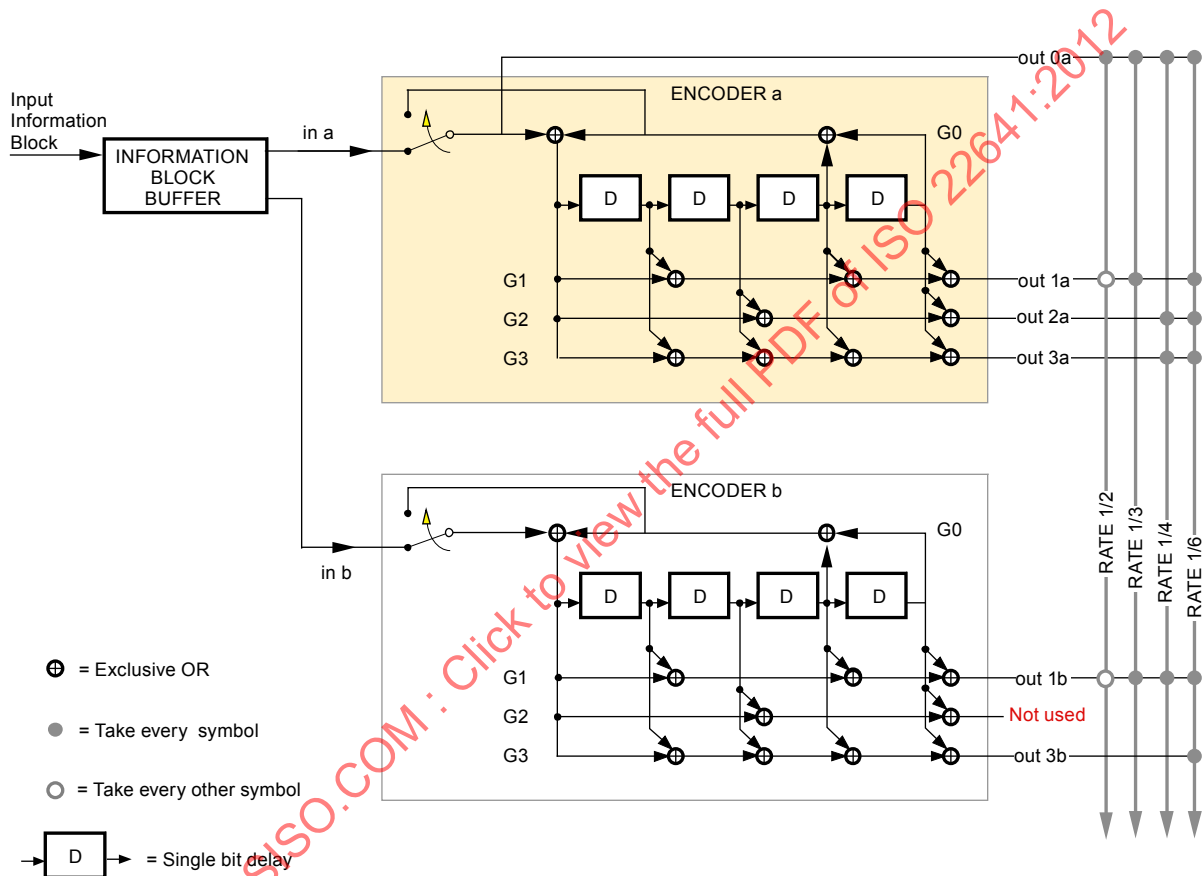
Figure 6-2: Turbo Encoder Block Diagram

h) Backward and forward connection vectors (see figure 6-2) shall be as follows:

1) Backward connection vector for both component codes and all code rates shall be G0 = 10011.

2) Forward connection vector for both component codes and rates 1/2 and 1/3 shall be G1 = 11011.

i) Puncturing of every other symbol from each component code shall be done for rate 1/2.

ii) No puncturing shall be done for rate 1/3.

45

    – Forward connection vectors for rate 1/4 shall be G2 = 10101, G3 = 11111 (1st component code); G1 = 11011 (2nd component code). No puncturing is done for rate 1/4.

    3) Forward connection vectors for rate 1/6 shall be G1 = 11011, G2 = 10101, G3 = 11111 (1st component code); G1 = 11011, G3 = 11111 (2nd component code). No puncturing shall done for rate 1/6.

i) Turbo encoder operation:

    1) Each input frame of k information bits shall be held in a frame buffer, and the bits in the buffer shall be read out in two different orders for the two component encoders.

    2) The first component encoder (a in figure 6-2) shall operate on the bits in unpermuted order ('in a'), while the second component encoder (b figure 6-2) shall receive the same bits permuted by the interleaver ('in b').

    NOTES

    1   The read-out addressing for 'in a' is a simple counter, while the addressing for 'in b' is specified by the Turbo code permutation described in 6.3g).

    2   The recommended encoder block diagram is shown in figure 6-2.

    3   The component encoders are recursive convolutional encoders realized by feedback shift registers as shown in figure 6-2. The circuits shown in this figure implement the backward connection vector, G0, and the forward connection vectors, G1, G2, G3, specified in 6.3h). A key difference between these convolutional component encoders and the standalone convolutional encoder recommended in section 3 is their recursiveness. In the figure this is indicated by the signal (corresponding to the backward connection vector G0) fed back into the leftmost adder of each component encoder.

j) Turbo Codeword specification:

    1) Both component encoders (see figure 6-2) shall be initialized with '0's in all registers, and both shall run for a total of $k+4$ bit times, producing an output Codeword of $(k+4)/r$ encoded symbols, where r is the nominal code rate.

    2) For the first k bit times, the input switches shall be in the lower position (as indicated in figure 6-2) to receive input data.

    3) For the final four bit times, these switches shall move to the upper position to receive feedback from the shift registers.

NOTE – This feedback cancels the same feedback sent (unswitched) to the leftmost adder and causes all four registers to become filled with zeros after the final 4 bit times. Filling the registers with zeros is called terminating the trellis.

4) During trellis termination the encoder shall continue to output nonzero encoded symbols. In particular, the 'systematic uncoded' output (line 'out 0a' in the figure) shall include an extra four bits from the feedback line in addition to the k information bits.

NOTE – In figure 6-2, the encoded symbols are multiplexed from top-to-bottom along the output line for the selected code rate to form the Turbo Codeword. For the rate 1/3 code, the output sequence is (out 0a, out 1a, out 1b); for rate 1/4, the sequence is (out 0a, out 2a, out 3a, out 1b); for rate 1/6, the sequence is (out 0a, out 1a, out 2a, out 3a, out 1b, out 3b). These sequences are repeated for $(k+4)$ bit times. For the rate 1/2 code, the output sequence is (out 0a, out 1a, out 0a, out 1b), repeated $(k+4)/2$ times. This pattern implies that out 1b is the first to be punctured, out 1a is the second, and so forth. The Turbo Codewords constructed from these output sequences are depicted in figure 6-3 for the four nominal code rates.
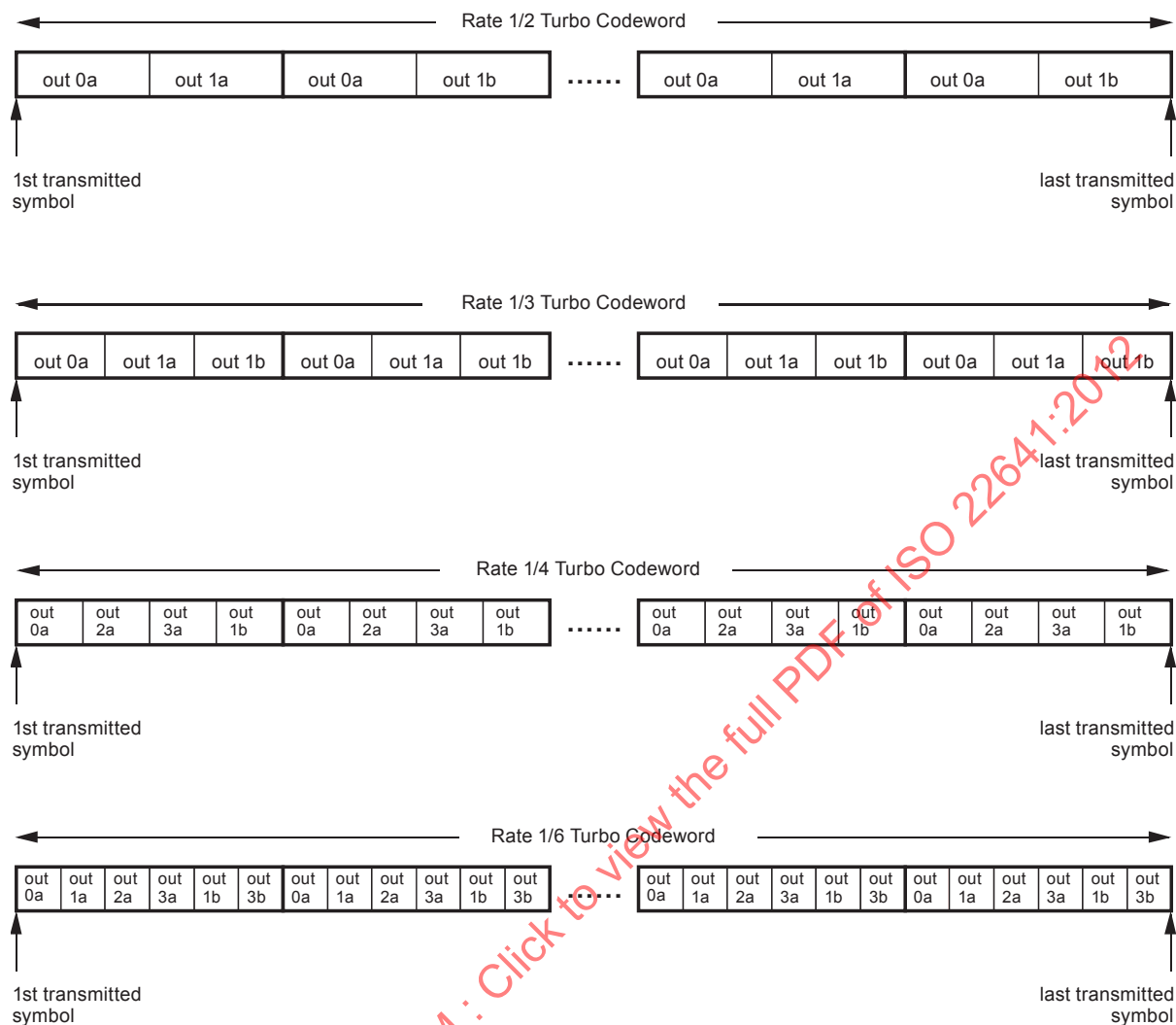
Figure 6-3: Turbo Codewords for Different Code Rates

k) Turbo Codeword synchronization:

An Attached Sync Marker (ASM) shall be associated with each Turbo Codeword (see section 8), and it shall precede the Turbo Codeword.

NOTES

1   The ASM is used to achieve Codeword synchronization of the Turbo decoder (i.e., frame synchronizers are normally set to expect a marker at a recurrence interval equal to the length of the ASM plus that of the Turbo Codeword).

2   Differential encoding does not provide benefits with Turbo Codes, and the ASM can also be used to resolve phase ambiguities. In fact, differential encoding before the Turbo encoder cannot be used because the Turbo codes recommended in this document are non-transparent, and differential encoding after the Turbo

encoder is not advised because it introduces considerable loss of performance. It also would require differential detection, which is more complex with soft symbols. This implies that phase ambiguities have to be detected and resolved before decoding.

3   A diagram of a Turbo Codeword with ASM is shown in figure 6-4. The length of the Turbo Codeword is inversely proportional to the nominal code rate *r*.
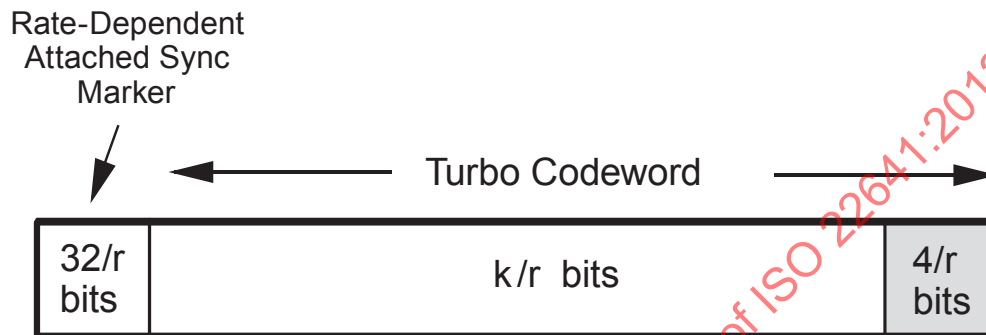
Figure 6-4: Turbo Codeword with Attached Sync Marker

49

(Blank Page)

7 LOW-DENSITY PARITY-CHECK CODING

7.1 OVERVIEW

Low-Density Parity-Check (LDPC) codes are binary block codes with large codewords (hundreds or thousands of bits). They may be used to obtain greater coding gains than those provided by concatenated coding systems.

An LDPC code is specified indirectly by a *v*-by-*w* parity check matrix H consisting of *v* linearly independent rows. Parity check matrices may include additional linearly dependent rows without changing the code. A coded sequence of *w* bits must lie in the *w*−*v* dimensional dual space of H; that is, it must satisfy all *v* parity check equations corresponding to the *v* rows of H. Alternatively, the code can be described through a (*w*−*v*)-by-*w* generator matrix G; such that a coded sequence lies in the *w*−*v* dimensional space of G. An encoder maps an input frame of *k* information bits uniquely into a codeword of *n* bits. LDPC codes may be shortened or expurgated so that *k*<*w*−*v*, and the remaining dimensions of the code remain unused. LDPC codes may also be extended or punctured to make *n* greater or less than *w*.

The distinguishing feature of LDPC codes is to have a low density of ones in the matrix H. Conversely, the generator matrix G is usually dense; that is, its density of ones is in the same order of that of zeros, at least for the non-systematic part of G.

Subsection 7.3 describes a code with a rate of 223/255 (approximately 7/8), and 7.4 describes a set of nine codes with rates 1/2, 2/3, and 4/5. These codes are systematic and non-transparent.

7.2 GENERAL

7.2.1 SYNCHRONIZATION

7.2.1.1 The (8160,7136) code defined in 7.3 shall be used with the 32-bit ASM shown in figure 8-1.

7.2.1.2 All of the nine codes with rates 1/2, 2/3, and 4/5, defined in 7.4, shall be used with the 64-bit ASM shown in figure 8-2.

NOTE – Differential encoding does not provide benefits with LDPC codes, and the ASM can also be used to resolve phase ambiguities. In fact, differential encoding before the LDPC encoder cannot be used because the LDPC codes recommended in this document are non-transparent, and differential encoding after the LDPC encoder is not advised because it introduces considerable loss of performance. It also would require differential detection, which is more complex with soft symbols. This implies that phase ambiguities have to be detected and resolved before decoding.

7.2.2   DATA RANDOMIZATION

The pseudo-randomizer defined in section 9 shall be used unless the system designer verifies that the concerns identified in the note below are resolved by other means.

NOTE   –   The recommended LDPC codes, by themselves, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Because of the quasi-cyclic nature of these codes, undetected decoding errors may result from incorrect codeword synchronization. The pseudo-randomizer is also used to aid signal acquisition and to mitigate spectral lines in the transmitted signal.

7.2.3   FRAME VALIDATION

7.2.3.1   The LDPC decoder may be used alone to validate the codeword, and consequently the contained TM Transfer Frame (reference [1]) or AOS Transfer Frame (reference [2]).

7.2.3.2   The Frame Error Control Field (FECF) specified in references [1] and [2] is optional, and the system designer may choose to use it for additional frame validation.

NOTE   –   The undetected frame and bit error rates of these LDPC codes lie several orders of magnitude below the corresponding detected error rates for any given operating signal-to-noise ratio.

7.3   LOW-DENSITY PARITY-CHECK CODE WITH RATE 223/255

7.3.1   OVERVIEW

The (8160,7136) recommended code is an expurgated, shortened, and extended version of a basic (8176,7156) LDPC code.

The recommended code has rate 223/255, and matches the length and dimension of the (255,223) I=4 Reed-Solomon code.

The basic code is transparent, although the modified version of this code is not, because of the sense of the fill bits.

Construction of the initial code is described in 7.3.2, expurgation and encoding are described in 7.3.4, and the shortening and extension that yield the recommended code are described in 7.3.5.

7.3.2   BASIC (8176,7156) LDPC CODE USED IN CONSTRUCTION

7.3.2.1   The parity check matrix for the (8176,7156) LDPC code shall be formed by using a 2×16 array of 511×511 square circulants.

NOTE – This creates a parity check matrix of size 1022×8176 and rank 1020.

7.3.2.2 The structure of the parity check base matrix shall be as shown in figure 7-1.

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} & A_{1,6} & A_{1,7} & A_{1,8} & A_{1,9} & A_{1,10} & A_{1,11} & A_{1,12} & A_{1,13} & A_{1,14} & A_{1,15} & A_{1,16} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} & A_{2,6} & A_{2,7} & A_{2,8} & A_{2,9} & A_{2,10} & A_{2,11} & A_{2,12} & A_{2,13} & A_{2,14} & A_{2,15} & A_{2,16} \end{bmatrix}$$

Figure 7-1: Base Parity Check Matrix of the Basic (8176,7156) LDPC Code

7.3.3   DISCUSSION

Each A $_{i,j}$ is a 511×511 circulant. The row weight of each of the 32 circulants is two; i.e., there are two '1's in each row. The total row weight of each row in the parity check matrix is $2 \times 16 = 32$. The position of the '1's in the first row of each circulant is defined in the second column of table 7-1; each subsequent row is given by a one-bit right cyclic shift of the preceding row. There are 511 possible positions, with position numbers between 0 and 510. The third column represents the absolute position of the '1's in the parity check matrix. There are 8176 possible positions; therefore these numbers are between 0 and 8175.

Table 7-1: Specification of Circulants

Circulant	'1's position in 1st row of circulant	Absolute '1's position in 1st row of Parity Check Matrix
$A_{1,1}$	0, 176	0, 176
$A_{1,2}$	12, 239	523, 750
$A_{1,3}$	0, 352	1022, 1374
$A_{1,4}$	24, 431	1557, 1964
$A_{1,5}$	0, 392	2044, 2436
$A_{1,6}$	151, 409	2706, 2964
$A_{1,7}$	0, 351	3066, 3417
$A_{1,8}$	9, 359	3586, 3936
$A_{1,9}$	0, 307	4088, 4395
$A_{1,10}$	53, 329	4652, 4928
$A_{1,11}$	0, 207	5110, 5317
$A_{1,12}$	18, 281	5639, 5902
$A_{1,13}$	0, 399	6132, 6531
$A_{1,14}$	202, 457	6845, 7100
$A_{1,15}$	0, 247	7154, 7401
$A_{1,16}$	36, 261	7701, 7926
$A_{2,1}$	99, 471	99, 471
$A_{2,2}$	130, 473	641, 984
$A_{2,3}$	198, 435	1220, 1457
$A_{2,4}$	260, 478	1793, 2011
$A_{2,5}$	215, 420	2259, 2464
$A_{2,6}$	282, 481	2837, 3036
$A_{2,7}$	48, 396	3114, 3462
$A_{2,8}$	193, 445	3770, 4022
$A_{2,9}$	273, 430	4361, 4518
$A_{2,10}$	302, 451	4901, 5050
$A_{2,11}$	96, 379	5206, 5489
$A_{2,12}$	191, 386	5812, 6007
$A_{2,13}$	244, 467	6376, 6599
$A_{2,14}$	364, 470	7007, 7113
$A_{2,15}$	51, 382	7205, 7536
$A_{2,16}$	192, 414	7857, 8079

7.3.4 ENCODING

NOTE – Two bits of information lie outside the structure of the quasi-cyclic encoder and increase complexity of the generator matrix for the basic (8176,7156) LDPC code. They are not included in this specification, which results in a generator matrix for a systematic (8176,7154) subcode that can be constructed entirely of circulants as shown in figure 7-2.

7.3.4.1 The generator matrix for the systematic (8176,7154) subcode shall be that illustrated in figure 7-2.

7.3.4.2 The left portion of the matrix shall be a 7154×7154 identity matrix, shown here as a block matrix, where I denotes the identity matrix of size 511×511, and 0 denotes the all-zero matrix of size 511×511.

7.3.4.3 The right portion of the matrix shall contain two columns of 511×511 circulants, denoted $B_{i,j}$, and constructed as follows:

1) $D = \begin{bmatrix} A_{1,15} & A_{1,16} \\ A_{2,15} & A_{2,16} \end{bmatrix}$ shall be defined from figure 7-1 and table 7-1.

NOTE – This equation describes a 1022×1022 matrix.

2) $u = (1\ 0\ 0\ 0\ ....\ 0)$ shall be the unit 511 tuple, i.e., a vector quantity of length 511 with a '1' at the leftmost position and '0's in the rest.

3) $z_i = (b_{i,1}\quad b_{i,2})$ shall be defined, where $i = 1, 2, ..., 14$ and the $b_{i,j}$s are the first rows of the $B_{i,j}$ circulants.

NOTE – For $i = 1, 2, ..., 14$, z_i is a vector with 1020 elements.

4) $M_i = \begin{bmatrix} A_{1,i} \\ A_{2,i} \end{bmatrix}$ shall be defined, where $i = 1, 2, ..., 14$.

NOTE – The parity check matrix can now be represented as: $[M_1\ M_2\ ...\ M_{14}\ D]$.

5) The 511th and 1022nd elements of z_i shall be set to zero and $M_i u^T + D z_i^T = 0$ shall be solved for z_i, where $i = 1, 2, ..., 14$ and T superscript represents matrix transpose.

NOTE – Since the rank of D is 1020 and not 1022, there are two linearly dependent columns. These columns can be taken to be the 511th and 1022nd.

6) The $b_{i,j}$s shall be extracted from the z_is.

NOTE – The $b_{i,j}$s are numerically tabulated in annex C.

55

$$
\begin{bmatrix}
I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{1,1} & B_{1,2} \\
0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{2,1} & B_{2,2} \\
0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{3,1} & B_{3,2} \\
0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{4,1} & B_{4,2} \\
0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{5,1} & B_{5,2} \\
0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{6,1} & B_{6,2} \\
0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{7,1} & B_{7,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & B_{8,1} & B_{8,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & B_{9,1} & B_{9,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & B_{10,1} & B_{10,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & B_{11,1} & B_{11,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & B_{12,1} & B_{12,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & B_{13,1} & B_{13,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & B_{14,1} & B_{14,2}
\end{bmatrix}
$$

Figure 7-2: Systematic Circulant Generator Matrix

7.3.5 RECOMMENDED (8160,7136) CODE

NOTE – The generator matrix in 7.3.4 describes an (8176,7154) subcode of the (8176,7156) code defined by the parity check matrix in 7.3.2.2.

The (8176,7154) subcode shall be shortened and extended as follows to form an (8160,7136) code with parameters that are multiples of 32.

1) The encoder shall accept as input a Telemetry Transfer Frame of 7136 bits (i.e. 892 octets matching the length and dimension of (255,223) I=4 Reed-Solomon),

2) 18 zeros shall be prefixed to the 7136-bit message to be encoded, yielding a 7154-element row vector.

3) This vector shall be multiplied by the generator matrix of section 6.2.3, yielding an 8176-element vector consisting of 18 zeros, 7136 systematic message symbols, and 1022 parity symbols.

4) From this vector, the 18 leading zeros shall be discarded, and two zeros shall be appended, yielding a codeword of 8160 symbols.

NOTE – The arrangement of these 18 virtual fill bits, information bits, parity bits, and two zero-fill bits is shown in figure 7-3.
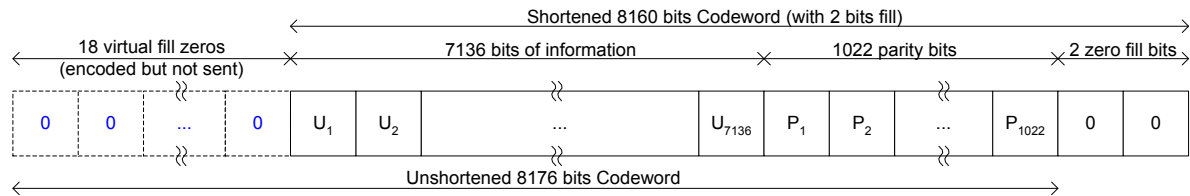
Figure 7-3: Shortened Codeword

7.4 LOW-DENSITY PARITY-CHECK CODE FAMILY WITH RATES 1/2, 2/3, AND 4/5

7.4.1 OVERVIEW

Nine punctured LDPC codes are specified, with information block lengths k=1024 bits, 4096 bits, and 16384 bits, and code rates r=1/2, 2/3, and 4/5. These parameters and the corresponding codeword lengths, $n=k/r$, are shown in table 7-5.

7.4.2 PARITY CHECK MATRICES

7.4.2.1 The parity check matrices shall be constructed from $M{\times}M$ submatrices, where the submatrix size is listed in table 7-2.

Table 7-2: Values of Submatrix Size M for Supported Codes

Information block length k	Submatrix size M		
	rate 1/2	rate 2/3	rate 4/5
1024	512	256	128
4096	2048	1024	512
16384	8192	4096	2048

7.4.2.2 The parity check matrices for the rate-1/2 codes shall satisfy the following equation:

$$H_{1/2} = \begin{bmatrix} 0_M & 0_M & I_M & 0_M & I_M \oplus \Pi_1 \\ I_M & I_M & 0_M & I_M & \Pi_2 \oplus \Pi_3 \oplus \Pi_4 \\ I_M & \Pi_5 \oplus \Pi_6 & 0_M & \Pi_7 \oplus \Pi_8 & I_M \end{bmatrix}$$

where I_M and 0_M are the $M{\times}M$ identity and zero matrices, respectively, and Π_1 through Π_8 are permutation matrices.

7.4.2.3 The parity check matrices for the rate-2/3 and rate-4/5 codes are specified with additional columns and permutation matrices and shall satisfy the following equations:

$$H_{2/3} = \begin{bmatrix} 0_M & 0_M & \\ \Pi_9 \oplus \Pi_{10} \oplus \Pi_{11} & I_M & \bigg| \ H_{1/2} \\ I_M & \Pi_{12} \oplus \Pi_{13} \oplus \Pi_{14} & \end{bmatrix}$$

$$H_{4/5} = \begin{bmatrix} 0_M & 0_M & 0_M & 0_M & \\ \Pi_{21} \oplus \Pi_{22} \oplus \Pi_{23} & I_M & \Pi_{15} \oplus \Pi_{16} \oplus \Pi_{17} & I_M & \bigg| \ H_{2/3} \\ I_M & \Pi_{24} \oplus \Pi_{25} \oplus \Pi_{26} & I_M & \Pi_{18} \oplus \Pi_{19} \oplus \Pi_{20} & \end{bmatrix}$$

7.4.2.4 The permutation matrix Π_K shall have non-zero entries in row i and column $\pi_K(i)$ for $i \in \{0,..., M\text{-}1\}$ and

$$\pi_k(i) = \frac{M}{4}\left(\left(\theta_k + \lfloor 4i / M \rfloor\right) \bmod 4\right) + \left(\phi_k\left(\lfloor 4i / M \rfloor, M\right) + i\right) \bmod \frac{M}{4}$$

where the functions θ_k and $\phi_k(j,M)$ are defined in table 7-3 and table 7-4.

NOTE – Values defined in tables 7-3 and 7-4 describe $\phi_k(j,M)$s using 7-tuples where consecutive positions in a tuple correspond to submatrix sizes from the set $M=\{128, 256, 512, 1024, 2048, 4096, 8192\}$.

7.4.2.5 For each of the parity check matrices, the code symbols corresponding to the last M columns shall be punctured (not transmitted).

Table 7-3: Description of $\phi_k(0,M)$ and $\phi_k(1,M)$

k	θ_k	$\phi_k(0,M)$ $M = 2^7 \dots 2^{13}$							$\phi_k(1,M)$ $M = 2^7 \dots 2^{13}$						
1	3	1	59	16	160	108	226	1148	0	0	0	0	0	0	0
2	0	22	18	103	241	126	618	2032	27	32	53	182	375	767	1822
3	1	0	52	105	185	238	404	249	30	21	74	249	436	227	203
4	2	26	23	0	251	481	32	1807	28	36	45	65	350	247	882
5	2	0	11	50	209	96	912	485	7	30	47	70	260	284	1989
6	3	10	7	29	103	28	950	1044	1	29	0	141	84	370	957
7	0	5	22	115	90	59	534	717	8	44	59	237	318	482	1705
8	1	18	25	30	184	225	63	873	20	29	102	77	382	273	1083
9	0	3	27	92	248	323	971	364	26	39	25	55	169	886	1072
10	1	22	30	78	12	28	304	1926	24	14	3	12	213	634	354
11	2	3	43	70	111	386	409	1241	4	22	88	227	67	762	1942
12	0	8	14	66	66	305	708	1769	12	15	65	42	313	184	446
13	2	25	46	39	173	34	719	532	23	48	62	52	242	696	1456
14	3	25	62	84	42	510	176	768	15	55	68	243	188	413	1940
15	0	2	44	79	157	147	743	1138	15	39	91	179	1	854	1660
16	1	27	12	70	174	199	759	965	22	11	70	250	306	544	1661
17	2	7	38	29	104	347	674	141	31	1	115	247	397	864	587
18	0	7	47	32	144	391	958	1527	3	50	31	164	80	82	708
19	1	15	1	45	43	165	984	505	29	40	121	17	33	1009	1466
20	2	10	52	113	181	414	11	1312	21	62	45	31	7	437	433
21	0	4	61	86	250	97	413	1840	2	27	56	149	447	36	1345
22	1	19	10	1	202	158	925	709	5	38	54	105	336	562	867
23	2	7	55	42	68	86	687	1427	11	40	108	183	424	816	1551
24	1	9	7	118	177	168	752	989	26	15	14	153	134	452	2041
25	2	26	12	33	170	506	867	1925	9	11	30	177	152	290	1383
26	3	17	2	126	89	489	323	270	17	18	116	19	492	778	1790

Table 7-4: Description of $\phi_k(2,M)$ and $\phi_k(3,M)$

k	θ_k	$\phi_k(2,M)$ $M = 2^7 \dots 2^{13}$							$\phi_k(3,M)$ $M = 2^7 \dots 2^{13}$						
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	12	46	8	35	219	254	318	13	44	35	162	312	285	1189
3	1	30	45	119	167	16	790	494	19	51	97	7	503	554	458
4	2	18	27	89	214	263	642	1467	14	12	112	31	388	809	460
5	2	10	48	31	84	415	248	757	15	15	64	164	48	185	1039
6	3	16	37	122	206	403	899	1085	20	12	93	11	7	49	1000
7	0	13	41	1	122	184	328	1630	17	4	99	237	185	101	1265
8	1	9	13	69	67	279	518	64	4	7	94	125	328	82	1223
9	0	7	9	92	147	198	477	689	4	2	103	133	254	898	874
10	1	15	49	47	54	307	404	1300	11	30	91	99	202	627	1292
11	2	16	36	11	23	432	698	148	17	53	3	105	285	154	1491
12	0	18	10	31	93	240	160	777	20	23	6	17	11	65	631
13	2	4	11	19	20	454	497	1431	8	29	39	97	168	81	464
14	3	23	18	66	197	294	100	659	22	37	113	91	127	823	461
15	0	5	54	49	46	479	518	352	19	42	92	211	8	50	844
16	1	3	40	81	162	289	92	1177	15	48	119	128	437	413	392
17	2	29	27	96	101	373	464	836	5	4	74	82	475	462	922
18	0	11	35	38	76	104	592	1572	21	10	73	115	85	175	256
19	1	4	25	83	78	141	198	348	17	18	116	248	419	715	1986
20	2	8	46	42	253	270	856	1040	9	56	31	62	459	537	19
21	0	2	24	58	124	439	235	779	20	9	127	26	468	722	266
22	1	11	33	24	143	333	134	476	18	11	98	140	209	37	471
23	2	11	18	25	63	399	542	191	31	23	23	121	311	488	1166
24	1	3	37	92	41	14	545	1393	13	8	38	12	211	179	1300
25	2	15	35	38	214	277	777	1752	2	7	18	41	510	430	1033
26	3	13	21	120	70	412	483	1627	18	24	62	249	320	264	1606

7.4.3 ENCODING

7.4.3.1 The encoder shall accept as input a Telemetry Transfer Frame of length k as per table 7-5.

7.4.3.2 Codewords consistent with the parity-check matrices in 7.4.2 shall be produced by performing matrix multiplication by block-circulant generator matrices.

NOTE – This family of codes supports rates $K/(K+2)$, where $K=2$ for a rate 1/2 code, $K=4$ for rate 2/3, and $K=8$ for rate 4/5. Corresponding generator matrices, G, have size $MK \times M(K+3)$ if punctured columns are described in the encoding, or $MK \times M(K+2)$ if punctured columns are omitted.

Table 7-5: Codeword Lengths for Supported Code Rates (Measured in Bits)

Telemetry Transfer Frame Length or Information block length k	Codeword length n		
	rate 1/2	rate 2/3	rate 4/5
1024	2048	1536	1280
4096	8192	6144	5120
16384	32768	24576	20480

7.4.3.3 The generator matrices shall be constructed as follows:

1) P shall be the $3M \times 3M$ submatrix of H consisting of the last $3M$ columns. Q shall be the $3M \times MK$ submatrix of H consisting of the first MK columns.

2) $W = (P^{-1}Q)^T$ shall be computed, where the arithmetic is performed modulo-2.

7.4.3.4 The matrix $G = \begin{bmatrix} I_{MK} & W \end{bmatrix}$ shall be constructed, where I_{MK} is the $MK \times MK$ identity matrix, and W is a dense matrix of circulants of size $MK \times 3M$.

7.4.3.5 The last M columns of G shall be punctured.

61

(Blank Page)

8   FRAME SYNCHRONIZATION

8.1   OVERVIEW

8.1.1   SYNCHRONIZATION

Frame or Codeblock synchronization is necessary for proper decoding of Reed-Solomon, Turbo, and LDPC Codewords, and subsequent processing of the Transfer Frames. Furthermore, it is necessary for synchronization of the pseudo-random generator, if used (see section 9). It is also useful in assisting the node synchronization process of the decoder for the convolutional code.

For a coding system using the basic convolutional code specified in 3.3, the ASM can be acquired either in the channel symbol domain (i.e., before any decoding) or in the domain of bits decoded by the convolutional decoder.

For a concatenated Reed-Solomon and convolutional coding system using the basic convolutional code specified in 3.3, the ASM can be acquired either in the channel symbol domain (i.e., before any decoding) or in the domain of bits decoded by the inner code (i.e., the code symbol domain of the Reed-Solomon code).

For a coding system using the punctured convolutional codes specified in 3.4, the ASM can only be acquired in the domain of bits decoded by the convolutional decoder. It cannot be acquired in the channel symbol domain (i.e., before any decoding).

For a concatenated Reed-Solomon and convolutional coding system using the punctured convolutional codes specified in 3.4, the ASM can only be acquired in the domain of bits decoded by the inner code (i.e., the code symbol domain of the Reed-Solomon code); i.e.; it cannot be acquired in the channel symbol domain (i.e., before any decoding).

For a Turbo or LDPC coding system, the ASM can only be acquired in the channel symbol domain (i.e., before any decoding in the code symbol domain of the Turbo or LDPC code).

8.1.2   CHANNEL ACCESS DATA UNIT

The data unit that consists of the ASM and the Transfer Frame (if the Physical Channel is not Reed-Solomon, Turbo, or LDPC coded) or the Reed-Solomon Codeblock or Turbo or LDPC Codeword (if the Physical Channel is Reed-Solomon, Turbo, or LDPC coded) is called the Channel Access Data Unit (CADU). The Transfer Frame, Codeword, or Codeblock in the CADU may or may not be randomized.

8.2   THE ATTACHED SYNC MARKER (ASM)

8.2.1.1   Reed-Solomon Codeblocks or Turbo or LDPC Codewords (or Transfer Frames, if the Physical Channel is not Reed-Solomon, Turbo, or LDPC coded) shall be synchronized by using a stream of fixed-length Codeblocks or Codewords (or Transfer Frames) with an Attached Sync Marker (ASM) between them.

NOTE   –   Synchronization is acquired on the receiving end by recognizing the specific bit pattern of the ASM in the Physical Channel data stream; synchronization is then verified by making further checks.

8.2.1.2   If convolutional code is used, the ASM shall be convolutionally encoded.

8.2.1.3   If an inner convolutional code is used in conjunction with an outer Reed-Solomon code, the ASM shall be encoded by the inner code but not by the outer code. (See section 3.)

8.3   ASM BIT PATTERNS

8.3.1   The ASM for data that is not Turbo or LDPC coded shall consist of a 32-bit (4-octet) marker with a pattern shown in figure 8-1.

8.3.2   The ASM for data that is Turbo coded with nominal code rate r = 1/2, 1/3, 1/4, or 1/6 shall consist of a 32/r-bit (4/r-octet) marker with bit patterns shown in figures 8-2 through 8-5.

8.3.3   The ASM for data that is LDPC coded with nominal code rate r = 7/8 shall consist of a 32-bit marker with bit pattern shown in figure 8-1.

8.3.4   The ASM for data that is LDPC coded with code rate r = 1/2, 2/3, or 4/5 shall consist of a 64-bit marker with bit pattern shown in figure 8-2.

NOTE   –   The ASM bit patterns are represented in hexadecimal notation as:

ASM for uncoded data, convolutional, Reed-Solomon, concatenated, and rate-7/8 LDPC coded data:	1ACFFC1D
ASM for rate-1/2 Turbo and rates 1/2, 2/3, and 4/5 LDPC coded data:	034776C7272895B0
ASM for rate-1/3 Turbo coded data:	25D5C0CE8990F6C9461BF79C
ASM for rate-1/4 Turbo coded data:	034776C7272895B0 FCB88938D8D76A4F
ASM for rate-1/6 Turbo coded data:	25D5C0CE8990F6C9461BF79C DA2A3F31766F0936B9E40863

FIRST TRANSMITTED BIT

(Bit 0)

↓

0001 1010 1100 1111 1111 1100 0001 1101

↑

LAST TRANSMITTED BIT

(Bit 31)

Figure 8-1: ASM Bit Pattern for Uncoded, Convolutional, Reed-Solomon, Concatenated, and Rate 7/8 LDPC Coded Data

FIRST TRANSMITTED BIT

(Bit 0)

↓

0000001101000111011101101100011100100111001010001001010110110000

↑

LAST TRANSMITTED BIT

(Bit 63)

Figure 8-2: ASM Bit Pattern for Rate 1/2 Turbo and Rates 1/2, 2/3, and 4/5 LDPC Coded Data

FIRST TRANSMITTED BIT

(Bit 0)

↓

001001011101010111000000011001110100010011001000011110110110010010100011000011011111101111100111100

↑

LAST TRANSMITTED BIT

(Bit 95)

Figure 8-3: ASM Bit Pattern for Rate 1/3 Turbo Coded Data

FIRST TRANSMITTED BIT

(Bit 0)

↓

0000001101000111011101101100011100100111001010001001010110110000
1111110010111000100010010011100011011000110101110110101001001111

↑

LAST TRANSMITTED BIT

(Bit 127)

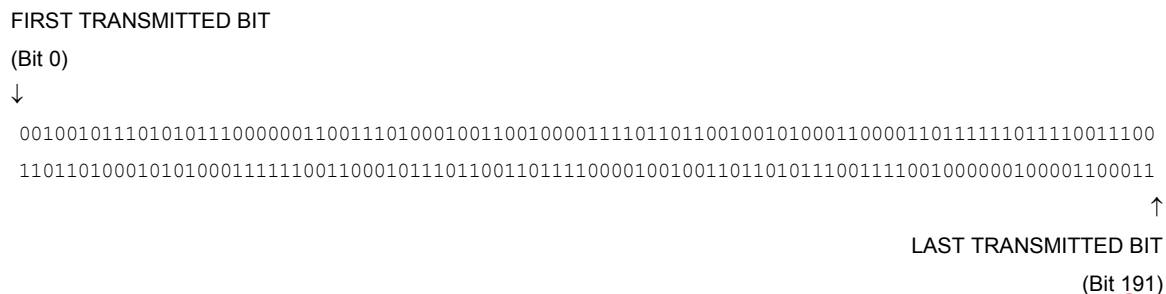**Figure 8-4: ASM Bit Pattern for Rate 1/4 Turbo Coded Data**

65

FIRST TRANSMITTED BIT

(Bit 0)

↓

001001011101010111000000110011101000100110010000111101101100100101000110000110111111011110011100
110110100010101000111111001100010111011001101111000010010011011010111001111001000000100001100011

↑

LAST TRANSMITTED BIT

(Bit 191)

Figure 8-5: ASM Bit Pattern for Rate 1/6 Turbo Coded Data

8.4 LOCATION OF ASM

8.4.1   The ASM shall be attached to (i.e., shall immediately precede) the codeblock (if Reed-Solomon encoded), the codeword (if Turbo or LDPC encoded), or the transfer frame (if convolutionally encoded only or uncoded).

8.4.2   The ASM shall immediately follow the end of the preceding codeblock (if Reed-Solomon encoded), the codeword (if Turbo or LDPC encoded), or the transfer frame (if convolutionally encoded only or uncoded); i.e., there shall be no intervening bits (data or fill) preceding the ASM.

8.5 RELATIONSHIP OF ASM TO REED-SOLOMON, TURBO, AND LDPC CODEBLOCKS AND CODEWORDS

8.5.1   The ASM shall NOT be a part of the encoded data space of the Reed-Solomon Codeblock, and it shall not be presented to the input of the Reed-Solomon encoder or decoder.

NOTE – This prevents the encoder from routinely regenerating a second, identical marker in the check symbol field under certain repeating data-dependent conditions (e.g., a test pattern of 01010101010 ... among others), which could cause synchronization difficulties at the receiving end. The relationship among the ASM, Reed-Solomon Codeblock, and Transfer Frame is illustrated in figure 4-1.

8.5.2   Similarly, the ASM shall not be presented to the input of the Turbo encoder or decoder. It shall be directly attached to the Turbo Codeword (see figure 6-4).

8.5.3   Similarly, the ASM shall not be presented to the input of the LDPC encoder or decoder. It shall be directly attached to the LDPC Codeword.

8.6 ASM FOR EMBEDDED DATA STREAM

NOTE – A different ASM pattern (see figure 8-6) may be required where another data stream (e.g., a stream of Transfer Frames played back from a tape recorder in the forward direction) is inserted into the data field of the Transfer Frame of the main stream appearing on the communications channel.

The ASM for the embedded data stream, to differentiate it from the main stream marker, shall consist of a 32-bit (4-octet) marker with a pattern as follows:

```
FIRST TRANSMITTED BIT
(Bit 0)
↓
  0011  0101  0010  1110  1111  1000  0101
                          0011
                                        ↑
                          LAST TRANSMITTED BIT
                          (Bit 31)
```

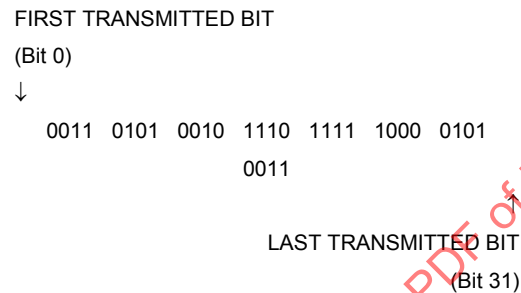**Figure 8-6: Embedded ASM Bit Pattern**

NOTE – This pattern is represented in hexadecimal notation as:

352EF853

(Blank Page)

9 PSEUDO-RANDOMIZER

9.1 OVERVIEW

In order for the receiver system to work properly, every data capture system at the receiving end requires that the incoming signal have sufficient bit transition density (see recommendation 2.4.9 in reference [5]), and allow proper synchronization of the decoder. The incoming signal must also be free of significant spectral lines, and be free of patterns that interfere with codeword synchronization and validation (see 2.2.2).

NOTE – Designers should note that the length-255 pseudo-randomizer may introduce spectral lines at 1/255 of the symbol rate, and these may be significant in some systems.

In order to ensure proper receiver operation, the data stream must be sufficiently random. The Pseudo-Randomizer defined in this section is the preferred method to ensure sufficient randomness for all combinations of CCSDS-recommended modulation and coding schemes. The Pseudo-Randomizer defined in this section is required unless the system designer verifies proper operation of the system if this Randomizer is not used.

NOTE – Problems with telemetry links have been encountered because this Pseudo-Randomizer was not used, and sufficient randomness was not ensured by other means and properly verified.

The presence or absence of pseudo-randomization is fixed for a Physical Channel and is *managed* (i.e., its presence or absence is not signaled in the transmitted data stream but must be known a priori) by the receiver.

9.2 PSEUDO-RANDOMIZER DESCRIPTION

9.2.1 The method for ensuring sufficient transitions is to exclusive-OR each bit of the Codeblock, Codeword, or Transfer Frame with a standard pseudo-random sequence.

9.2.2 If the pseudo-randomizer is used, on the sending end it shall be applied to the Codeblock, Codeword, or Transfer Frame after Reed-Solomon, Turbo, or LDPC encoding (if any of these are used), but before convolutional encoding (if used). On the receiving end, it shall be applied to de-randomize the data after convolutional decoding (if used) and codeblock or codeword synchronization but before Reed-Solomon, Turbo, or LDPC decoding (if any of these are used).

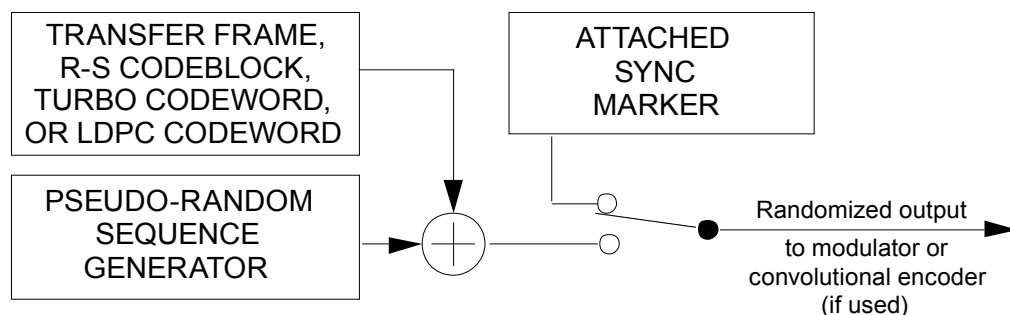NOTE – The configuration at the sending end is shown in figure 9-1.

```
┌────────────────────────┐      ┌──────────────┐
│  TRANSFER FRAME,       │      │  ATTACHED    │
│  R-S CODEBLOCK,        │      │  SYNC        │
│  TURBO CODEWORD,       │      │  MARKER      │
│  OR LDPC CODEWORD      │      │              │
└────────────────────────┘      └──────────────┘
┌────────────────────────┐
│  PSEUDO-RANDOM         │              Randomized output
│  SEQUENCE              │──(+)──o──●──→ to modulator or
│  GENERATOR             │       o      convolutional encoder
└────────────────────────┘              (if used)
```

Figure 9-1: Pseudo-Randomizer Configuration

9.3 SYNCHRONIZATION AND APPLICATION OF PSEUDO-RANDOMIZER

9.3.1 The Attached Sync Marker (ASM) shall be used for synchronizing the pseudo-randomizer.

NOTE – The Attached Sync Marker (ASM) is already optimally configured for synchronization purposes.

9.3.2 The pseudo-random sequence shall be applied starting with the first bit of the Codeblock, Codeword, or Transfer Frame. On the sending end, the Codeblock, Codeword, or Transfer Frame shall be randomized by exclusive-ORing the first bit of the Codeblock, Codeword, or Transfer Frame with the first bit of the pseudo-random sequence, followed by the second bit of the Codeblock, Codeword, or Transfer Frame with the second bit of the pseudo-random sequence, and so on.

9.3.3 On the receiving end, the original Codeblock, Codeword, or Transfer Frame shall be reconstructed (i.e., derandomized) using the same pseudo-random sequence.

9.3.4 After locating the ASM in the received data stream, the data immediately following the ASM shall be derandomized.

NOTES

1   The ASM was not randomized and is not derandomized.

2   Derandomization can be accomplished by performing exclusive-OR with hard bits or inversion with soft bits.

9.4 SEQUENCE SPECIFICATION

9.4.1 The pseudo-random sequence shall be generated using the following polynomial:

$$h(x) = x^8 + x^7 + x^5 + x^3 + 1$$

9.4.2 This sequence shall begin at the first bit of the Codeblock, Codeword, or Transfer Frame and shall repeat after 255 bits, continuing repeatedly until the end of the Codeblock,

Codeword, or Transfer Frame. The sequence generator shall be initialized to the all-ones state at the start of each Codeblock, Codeword, or Transfer Frame.

NOTE – The first 40 bits of the pseudo-random sequence from the generator are shown below. The leftmost bit is the first bit of the sequence to be exclusive-ORed with the first bit of the Codeblock, Codeword, or Transfer Frame; the second bit of the sequence is exclusive-ORed with the second bit of the Codeblock, Codeword, or Transfer Frame, and so on.

1111 1111 0100 1000 0000 1110 1100 0000 1001 1010 . . . .

9.5 LOGIC DIAGRAM

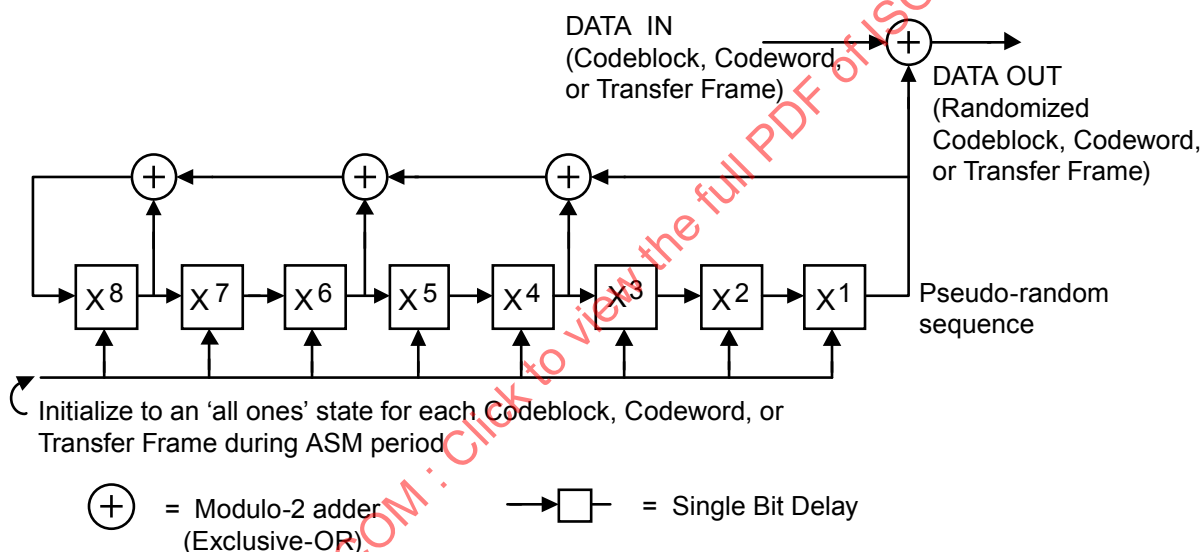NOTE – Figure 9-2 represents a possible generator for the specified sequence.

Figure 9-2: Pseudo-Randomizer Logic Diagram

(Blank Page)

10 TRANSFER FRAME LENGTHS

10.1 OVERVIEW

Neither the TM Space Data Link Protocol (reference [1]) nor the AOS Space Data Link Protocol (reference [2]) specifies the length of Transfer Frames because there are constraints on the Transfer Frame length depending on the selected coding options.

The constraints on Transfer Frame lengths specified in this section apply to both TM Transfer Frames and the AOS Transfer Frames.

10.2 GENERAL

10.2.1 Once selected, the Transfer Frame length shall be fixed for a Mission Phase on a particular Physical Channel.

NOTE – The Transfer Frame lengths shown here do not include the length of the Attached Sync Marker (ASM) specified in section 8.

10.3 CASE 1: UNCODED

The length of the Transfer Frames shall be any integer number of octets, as required by the using project, with a maximum of 2048 octets.

10.4 CASE 2: CONVOLUTIONAL ONLY

The length of the Transfer Frames shall be any integer number of octets, as required by the using project, with a maximum of 2048 octets.

10.5 CASE 3: REED-SOLOMON ONLY

NOTES

1   With the Reed-Solomon Codes specified in section 4, only certain specific lengths of Transfer Frames may be contained within the codeblock's data space. In some cases these lengths can be shortened at a small sacrifice in coding gain.

2   Since these R-S codes have a symbol length of 8 bits, the length of the codeblock (in octets) is a multiple of the interleaving depth, which provides 'octet compatibility'.

10.5.1 If necessary, Transfer Frame lengths shall be shortened in discrete steps by using virtual fill.

10.5.2 If high-speed efficiency is needed for '32-bit compatibility' (with 32-bit processors, for example), then the length of the codeblock shall be a combined multiple of 4 and the interleaving depth.

10.5.3 When only octet compatibility is required, lengths for Transfer Frames (L in octets) shall be determined using the following equation:

$$L = (255\text{-}2E\text{-}q)\,I$$

such that L is a positive integer,

where E = error correction capability,

q = number of virtual fill symbols per R-S codeword, and

I = interleaving depth.

10.5.4 When 32-bit compatibility is required, the Transfer Frame length must be chosen so that

a) it is expressed by the equation in 10.5.3; and

b) the codeblock length $(255\text{-}q)I$ (in octets) is a multiple of 4.

10.6 CASE 4: CONCATENATED

The allowable lengths of Transfer Frames when the concatenated (Reed-Solomon and convolutional) coding is used are the same as those for the Reed-Solomon–only case (Case 3) shown in 10.5.

10.7 CASE 5: TURBO

10.7.1 The Transfer Frame lengths shall be selected to match the information block lengths for the selected Turbo code.

NOTE – The Turbo Codes specified in section 5 of this Recommended Standard are block codes.

10.7.2 Only the following information block lengths shall be used (values are in octets):

a) 223;

b) 446;

c) 892;

d) 1115.

NOTE – Performance for only the above block lengths (i.e., Transfer Frame lengths) has been validated by CCSDS and approved for use (values are in octets).

10.8 CASE 6: LDPC

10.8.1 The Transfer Frame lengths must match the information block lengths for the selected LDPC code.

NOTE – The LDPC Codes specified in section 7 of this Recommended Standard are block codes.

10.8.2 When the rate-7/8 LDPC code is used, the only allowable Transfer Frame length is 892 octets.

10.8.3 When the 1/2-, 2/3-, and 4/5-rate LDPC codes are used, the allowable Transfer Frame lengths are 128 octets, 512 octets, or 2048 octets.

(Blank Page)

11 MANAGED PARAMETERS

11.1 OVERVIEW

In order to conserve bandwidth on the space link, some parameters associated with synchronization and channel coding are handled by management rather than by inline communications protocol. The managed parameters are those which tend to be static for long periods of time, and whose change generally signifies a major reconfiguration of the synchronization and channel coding systems associated with a particular mission. Through the use of a management system, management conveys the required information to the synchronization and channel coding systems.

In this section, the managed parameters used by synchronization and channel coding systems are listed. These parameters are defined in an abstract sense and are not intended to imply any particular implementation of a management system.

11.2 GENERAL

11.2.1 All the managed parameters specified in this section shall be fixed for all Transfer Frames on a Physical Channel during a given Mission Phase.

11.2.2 When the Reed-Solomon or LDPC codes are not used, the Frame Error Control Field defined in references [1] or [2] shall be present.

NOTE – The presence or absence of the Frame Error Control Field is established by the management of the relevant Data Link Protocol. When the Reed-Solomon or LDPC codes are used, the Frame Error Control Field can still be present but no check is required by the decoding system.

11.2.3 If present, the Frame Error Control Field shall occur within every Transfer Frame transmitted within the same Physical Channel throughout a Mission Phase.

11.3 MANAGED PARAMETERS FOR SELECTED OPTIONS

The managed parameters for a particular Physical Channel shall be those specified in table 11-1.

Table 11-1: Managed Parameters for Selected Options

Managed Parameter	Allowed Values
Randomizer	Present/Absent
Coding Method	None Convolutional Reed-Solomon Concatenated Code Turbo LDPC

11.4 MANAGED PARAMETERS FOR CONVOLUTIONAL CODE

The managed parameters for convolutional code shall be those specified in table 11-2.

Table 11-2: Managed Parameters for Convolutional Code

Managed Parameter	Allowed Values
Code rate (r)	1/2, 2/3, 3/4, 5/6, 7/8

11.5 MANAGED PARAMETERS FOR REED-SOLOMON CODE

The managed parameters for Reed-Solomon code shall be those specified in table 11-3.

Table 11-3: Managed Parameters for Reed-Solomon Code

Managed Parameter	Allowed Values
Error Correction Capability (E, symbols)	8, 16
Interleaving Depth (I)	1, 2, 3, 4, 5, 8
Virtual Fill Length (Q, symbols)	Integer

11.6 MANAGED PARAMETERS FOR TURBO CODE

The managed parameters for Turbo code shall be those specified in table 11-4.

Table 11-4: Managed Parameters for Turbo Code

Managed Parameter	Allowed Values
Nominal Code Rate (r)	1/2, 1/3, 1/4, 1/6
Information Block Length (k, bits)	1784, 3568, 7136, 8920

11.7 MANAGED PARAMETERS FOR LOW-DENSITY PARITY-CHECK CODE

The managed parameters for LDPC code shall be those specified in table 11-5.

Table 11-5: Managed Parameters for Low-Density Parity-Check Code

Managed Parameter	Allowed Values
Code Rate (r)	1/2, 2/3, 4/5, 7/8
Information Block Length (k, bits)	7136 (if r=7/8) 1024, 4096, 16384 (if r=1/2, 2/3, or 4/5)

11.8 MANAGED PARAMETERS FOR FRAME SYNCHRONIZATION

The managed parameters for frame synchronization shall be those specified in table 11-6.

Table 11-6: Managed Parameters for Frame Synchronization

Managed Parameter	Allowed Values
Transfer Frame Length (bits)	Integer
ASM Length (bits)	32, 64, 96, 128, 192

NOTE – The ASM length is determined by the selected coding schemes.

(Blank Page)

ANNEX A

SERVICE

(NORMATIVE)

A1   OVERVIEW

A1.1   BACKGROUND

This annex provides service definition in the form of primitives, which present an abstract model of the logical exchange of data and control information between the service provider and the service user.   The definitions of primitives are independent of specific implementation approaches.

The parameters of the primitives are specified in an abstract sense and specify the information to be made available to the user of the primitives. The way in which a specific implementation makes this information available is not constrained by this specification. In addition to the parameters specified in this annex, an implementation can provide other parameters to the service user (e.g., parameters for controlling the service, monitoring performance, facilitating diagnosis, and so on).

A2   OVERVIEW OF THE SERVICE

The TM Synchronization and Channel Coding provides unidirectional (one way) transfer of a sequence of fixed-length TM or AOS Transfer Frames at a constant frame rate over a Physical Channel across a space link, with optional error detection/correction.

Only one user can use this service on a Physical Channel, and Transfer Frames from different users are not multiplexed together within one Physical Channel.

A3   SERVICE PARAMETERS

A3.1   FRAME

A3.1.1   The Frame parameter is the service data unit of this service and shall be either a TM Transfer Frame defined in reference [1] or an AOS Transfer Frame defined in reference [2].

A3.1.2   The length of any Transfer Frame transferred on a Physical Channel must be the same, and is established by management.

CCSDS 131.0-B-2   Page A-1   August 2011

A3.2   QUALITY INDICATOR

The Quality Indicator parameter shall be used to notify the user at the receiving end of the service that there is an uncorrectable error in the received Transfer Frame.

A3.3   SEQUENCE INDICATOR

The Sequence Indicator parameter shall be used to notify the user at the receiving end of the service that one or more Transfer Frames of the Physical Channel have been lost as the result of a loss of frame synchronization.

A4   SERVICE PRIMITIVES

A4.1   GENERAL

A4.1.1   The service primitives associated with this service are:

   a) ChannelAccess.request;

   b) ChannelAccess.indication.

A4.1.2   The ChannelAccess.request primitive shall be passed from the service user at the sending end to the service provider to request that a Frame be transferred through the Physical Channel to the user at the receiving end.

A4.1.3   The ChannelAccess.indication shall be passed from the service provider to the service user at the receiving end to deliver a Frame.

A4.2   ChannelAccess.request

A4.2.1   Function

The ChannelAccess.request primitive is the service request primitive for this service.

A4.2.2   Semantics

The ChannelAccess.request primitive shall provide a parameter as follows:

   ChannelAccess.request   (Frame)

A4.2.3   When Generated

The ChannelAccess.request primitive is passed to the service provider to request it to process and send the Frame.

A4.2.4   Effect On Receipt

Receipt of the ChannelAccess.request primitive causes the service provider to perform the functions described in 2.3.1 and to transfer the resulting channel symbols.

A4.3   ChannelAccess.indication

A4.3.1   Function

The ChannelAccess.indication primitive is the service indication primitive for this service.

A4.3.2   Semantics

The ChannelAccess.indication primitive shall provide parameters as follows:

        ChannelAccess.indication   (Frame,
                                        Quality Indicator,
                                        Sequence Indicator)

A4.3.3   When Generated

The ChannelAccess.indication primitive is passed from the service provider to the service user at the receiving end to deliver a Frame.

A4.3.4   Effect On Receipt

The effect of receipt of the ChannelAccess.indication primitive by the service user is undefined.

(Blank Page)

ANNEX B

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

B1   SECURITY CONSIDERATIONS

B1.1   SECURITY BACKGROUND

It is assumed that security is provided by encryption, authentication methods, and access control to be performed at higher layers (application and/or transport layers). Mission and service providers are expected to select from recommended security methods, suitable to the specific application profile. Specification of these security methods and other security provisions is outside the scope of this Recommended Standard. The coding layer has the objective of delivering data with the minimum possible amount of residual errors. An LDPC, Reed-Solomon, or other code with CRC code must be used to insure that residual errors are detected and the frame flagged. There is an extremely low probability of additional undetected errors that may escape this scrutiny. These errors may affect the encryption process in unpredictable ways, possibly affecting the decryption stage and producing data loss, but will not compromise the security of the data.

B1.2   SECURITY CONCERNS

Security concerns in the areas of data privacy, authentication, access control, availability of resources, and auditing are to be addressed in higher layers and are not related to this Recommended Standard. The coding layer does not affect the proper functioning of methods used to achieve such protection at higher layers, except for undetected errors, as explained above.

The physical integrity of data bits is protected from channel errors by the coding systems specified in this Recommended Standard. In case of congestion or disruption of the link, the coding layer provides methods for frame re-synchronization.

B1.3   POTENTIAL THREATS AND ATTACK SCENARIOS

An eavesdropper can receive and decode the codewords, but will not be able to get to the user data if proper encryption is performed at a higher layer. An interferer could affect the performance of the decoder by congesting it with unwanted data, but such data would be rejected by the authentication process. Such interference or jamming must be dealt with at the Physical Layer and through proper spectrum regulatory entities.