Consultative Committee for Space Data Systems

RECOMMENDATION FOR SPACE DATA SYSTEM STANDARDS

LOSSLESS DATA COMPRESSION

CCSDS 121.0-B-1

BLUE BOOK

May 1997

AUTHORITY

Issue:	Blue Book, Issue 1
Date:	May 1997
Location:	São José dos Campos São Paulo, Brazil

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 Recommendations is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems* (reference [B1]), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This Recommendation is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code MG)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of member space Agencies. 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 **Recommendations** and are not considered binding on any Agency.

This **Recommendation** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

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

- Whenever an Agency establishes a CCSDS-related **standard**, the Agency will provide other CCSDS member Agencies with the following information:

  -- The **standard** itself.

  -- The anticipated date of initial operational capability.

  -- The anticipated duration of operational service.

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

No later than five years from its date of issuance, this **Recommendation** 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 **Recommendation** is issued, existing CCSDS-related Agency standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each Agency to determine when such standards or implementations are to be modified. Each Agency is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommendation.

FOREWORD

This Recommendation establishes a common framework and provides a common basis for a Lossless data compression algorithm applicable to several different types of data.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommendation 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* (reference [B1]). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/ccsds/

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

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

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Industry Canada/Communications Research Centre (CRC)/Canada.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 121.0-B-1	Lossless Data Compression, Blue Book, Issue 1	May 1997	Original Issue

CONTENTS

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a Recommendation for source-coding data-compression algorithm applied to digital data and to specify how these compressed data shall be inserted into source packets for retrieval and decoding.

Source coding for data compression is a method utilized in data systems to reduce the volume of digital data to achieve benefits in areas including, but not limited to,

a) reduction of transmission channel bandwidth;

b) reduction of the buffering and storage requirement;

c) reduction of data-transmission time at a given rate.

1.2 SCOPE

The characteristics of source codes are specified only to the extent necessary to ensure multi-mission support capabilities. The specification does not attempt to quantify the relative bandwidth reduction, the merits of each approach discussed, or the design requirements for coders and associated decoders. Some performance information is included in reference [B2].

This Recommendation addresses only Lossless source coding, which is applicable to a wide range of digital data, both imaging and non-imaging, where the requirement is for a moderate data-rate reduction constrained to allow no distortion to be added in the data compression/decompression process. The decompression process is not addressed. See reference [B2] for an outline of an implementation.

1.3 APPLICABILITY

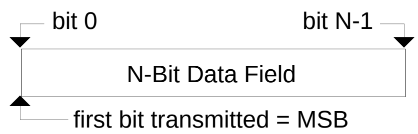
This Recommendation applies to data compression applications of space missions anticipating packetized telemetry cross support. In addition, it serves as a guideline for the development of compatible CCSDS Agency standards in this field, based on good engineering practice.

1.4 RATIONALE

The concept and rationale for the Lossless source coding for data compression algorithm described herein may be found in reference [B2].

1.5 BIT NUMBERING CONVENTION AND NOMENCLATURE

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’.



In accordance with modern data communications practice, spacecraft data fields are often grouped into 8-bit ‘words’ which conform to the above convention. Throughout this Recommendation, the following nomenclature is used to describe this grouping:

8-Bit Word = ‘Octet’

1.6 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommendation 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 Recommendations.

[1] *Telemetry Channel Coding*. Recommendation for Space Data Systems Standards, CCSDS 101.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, May 1992.

[2] *Packet Telemetry*. Recommendation for Space Data Systems Standards, CCSDS 102.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, November 1995.

[3] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data Systems Standards, CCSDS 701.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, November 1992.

2 OVERVIEW

2.1 GENERAL

This Recommendation defines for standardization a particular adaptive source coding algorithm that has widespread applicability to many forms of digital data. In particular, the science data from many types of imaging or non-imaging instruments are well suited for the application of this algorithm.

There are two classes of source coding methods: Lossless and Lossy.

A Lossless source coding technique preserves source data accuracy and removes redundancy in the data source. In the decoding process, the original data can be reconstructed from the compressed data by restoring the removed redundancy; the decompression process adds no distortion. This technique is particularly useful when data integrity cannot be compromised. The price it pays is generally a lower Compression Ratio, which is defined as the ratio of the number of original uncompressed bits to the number of compressed bits including overhead bits necessary for signaling parameters.

On the other hand, a Lossy source coding method removes some of the source information content along with the redundancy. The original data cannot be fully restored and data distortion occurs. However, if some distortion can be tolerated, Lossy source coding generally achieves a higher compression ratio. By controlling the amount of acceptable distortion and compression, this technique may enable acquisition and dissemination of mission data within a critical time span.

This Recommendation addresses only Lossless source coding and does not attempt to explain the theory underlying the operation of the algorithm.

2.2 THE SOURCE CODER

The Lossless source coder consists of two separate functional parts: the preprocessor and the adaptive entropy coder, as shown in figure 2-1.

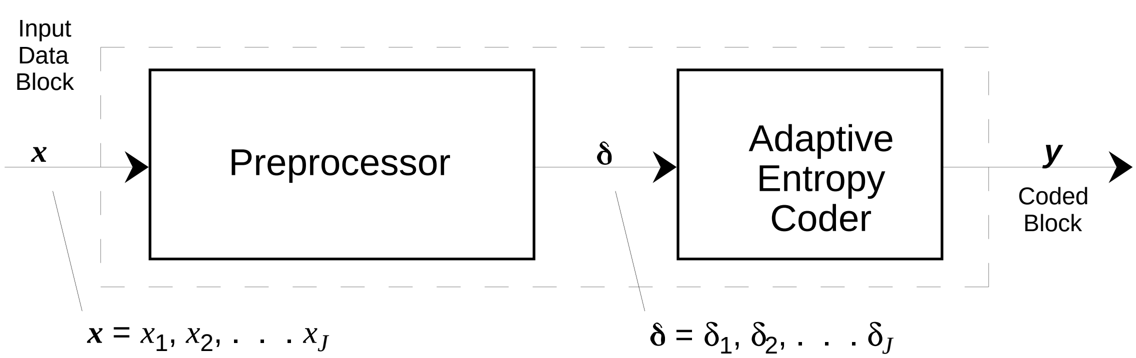


Figure 2-1: Schematic of the Source Coder

The inputs to the source coder are

$$\boldsymbol{x} = x_1, x_2, \ldots x_J$$

which is a block of J n-bit samples, where n is a constant value.

Preprocessor:

The preprocessor applies a reversible function to input data samples $\boldsymbol{x}$, to produce a 'preferred source':

$$\boldsymbol{\delta} = \delta_1, \delta_2, \ldots \delta_i, \ldots \delta_J$$

where each δ_i is an n-bit integer, $0 \le \delta_i \le (2^n–1)$. For an ideal preprocessing stage, $\boldsymbol{\delta}$ will have the following properties:

a) the $\{\delta_i\}$ is statistically independent and identically distributed;

b) the preferred probability, p_m, that any sample δ_i will take on integer value m is a nonincreasing function of value m, for $m = 0, 1, \ldots (2^n–1)$.

The preprocessor function is a reversible operation, and, in general, the best Lossless preprocessor will meet the above conditions and produce the lowest entropy, which is a measure of the smallest average number of bits that can be used to represent each sample.

This Recommendation does not attempt to explain methods for choosing a preprocessing stage. This Recommendation does provide the definition of a basic preprocessing stage that may be suitable for many applications. However, it is important that the user carefully address this issue since careful selection of an appropriate preprocessing stage is essential for efficient compression and depends on the source-data characteristics. Interested users should refer to reference [B2].

Adaptive Entropy Coder:

The function of the Adaptive Entropy Coder is to calculate uniquely decipherable, variable-length codewords corresponding to each block of samples input from the preprocessor. The entropy coder incorporates multiple coding options, each exhibiting efficient performance over different yet overlapping ranges of entropy. The coder selects the coding option that gives the highest compression ratio among the various options on the same block of J samples. A code-option 'identifier', requiring only a few bits, is attached before the first codeword bit in a coded block to signal the coding option to the decoder for proper decompression. Since the block size J is small (16 or fewer samples) and a new code option is selected for each block, the overall coding can be adjusted to adapt to rapid changes in data statistics.

2.3 PACKETIZATION OF CODED DATA

The variable-length encoded bit stream representing a *J*-sample block forms a Coded Data Set (CDS). CCSDS telemetry source packet structure is recommended to transport the CDSes, which will be contained in the source data field of the packet. The information related to, for example, the sensor, mission, time, and other mission-specific details necessary for the routing and accounting of the packets, will be contained in the Packet Primary Header and (if present) in the Packet Secondary Header (see references [2] and [3]).

2.4 ERROR CONTROL

Individual channel bit errors have greater consequences when data are compressed. Even then, the consequences need not be catastrophic. For this reason, to limit error propagation when utilizing the source coding algorithm described in this document, the following is recommended:

a) use telemetry channel coding as described in reference [1];

b) use packetized telemetry as described in references [2] and [3].

3 ADAPTIVE ENTROPY CODER

3.1 CODE SPECIFICATION

3.1.1 Figure 3-1 represents the general-purpose Adaptive Entropy Coder with a preprocessor. Basically, such a coder chooses one of a set of code options to use to represent an incoming block of preprocessed data samples, **δ**. A unique identifier (ID) bit sequence is attached to the code block to indicate to the decoder which decoding option to use.

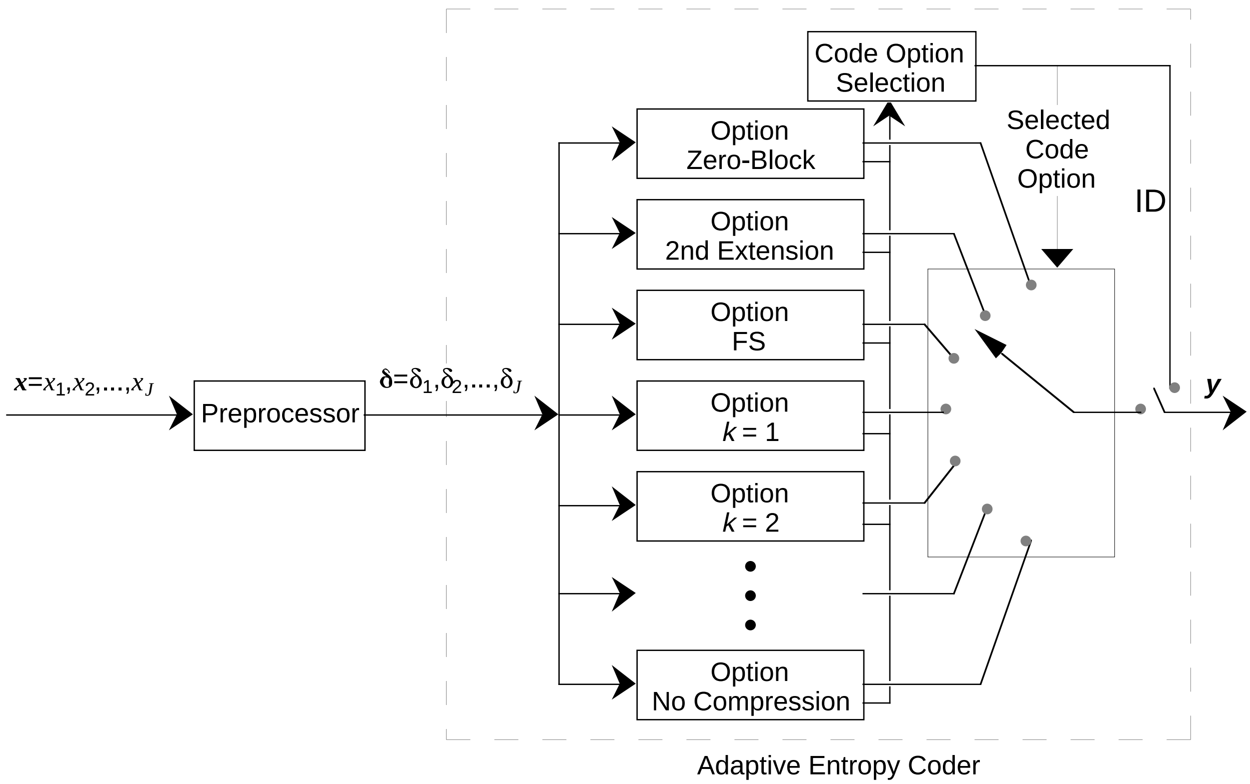


Figure 3-1: The Adaptive Entropy Coder with a Preprocessor

NOTE – Figure 3-1 illustrates the principle of the Adaptive Entropy Coder with a preprocessor; it does not illustrate an implementation.

3.1.2 The basic code selected is a variable-length code that utilizes Rice's adaptive coding technique (refer to reference [B2]). In Rice's coding technique, several algorithms are concurrently applied to a block of J consecutive preprocessed samples. The algorithm option that yields the shortest codeword sequence for the current block of data is selected for transmission. The zero-block option is a special case in that a single codeword sequence represents one or more consecutive blocks of J preprocessed samples (see 3.4.3). In all other options, the codeword sequence represents a single block of J consecutive preprocessed samples.

3.1.3 The following variables are required by Rice's adaptive coding technique:

– block size, J;

– resolution, n (number of input bits/sample);

– the ID bit sequence of the selected code option.

3.1.4 The following constraints shall apply to the Entropy Coder's variable-length adaptive coding scheme:

J	=	8 or 16 samples per block;
n	=	resolution with a maximum of 32 bits per sample with digital signal values from 0 to 2^n-1, or from -2^{n-1} to 2^{n-1}-1.

3.2 FUNDAMENTAL SEQUENCE

The most basic option is a variable-length Fundamental Sequence (FS) codeword, which consists of m zeros followed by a one when preprocessed sample $\delta_i = m$. Table 3-1 illustrates the FS codewords. A Fundamental Sequence is the concatenation of J FS codewords.

Table 3-1: Fundamental Sequence Codewords As a Function of the Preprocessed Samples

Preprocessed Sample Values, δ_i	FS Codeword
0	1
1	01
2	001
.	.
.	.
.	.
2^n–1	$\underbrace{0000 \ldots 00001}_{(2^n-1 \text{ zeros})}$

3.3 SAMPLE SPLITTING

3.3.1 The kth split-sample option is obtained by removing the k least-significant bits (LSBs) from the binary representation of each preprocessed sample, δ_i, and encoding the remaining bits with an FS codeword (see figure 3-2). This produces a varying codeword length. The FS codewords for the current block of J preprocessed samples are transmitted along with the removed LSBs, preceded by an ID field indicating the value of k. This process enables the adaptation of codeword length to source-data statistics.

3.3.2 The FS option described in 3.2 is a special case of sample splitting where $k = 0$.

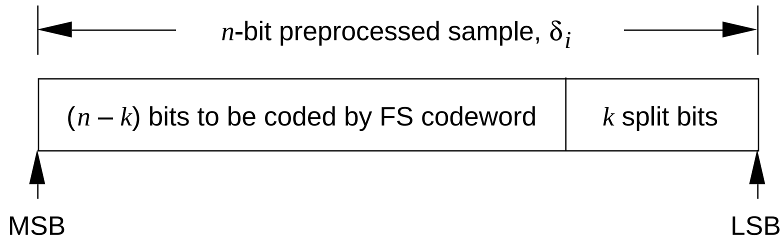


Figure 3-2: Split-Sample Format

3.4 LOW ENTROPY OPTIONS

3.4.1 GENERAL

The two code options, the Second-Extension option[1] and the Zero-Block option, provide more efficient coding than other options when the preprocessed data are highly compressible.

3.4.2 THE SECOND-EXTENSION OPTION

When the Second-Extension option is selected, each pair of preprocessed samples in a J-sample block is transformed and encoded using an FS codeword. Let δ_i and δ_{i+1} be adjacent pairs of samples from a J-sample preprocessed data block. They are transformed into a single new symbol γ by the following equation.

$$\gamma = (\delta_i + \delta_{i+1})(\delta_i + \delta_{i+1} + 1)/2 + \delta_{i+1}$$

The $J/2$ transformed symbols in a block are encoded using the FS codeword of table 3-1. The above process requires J to be an even integer which the recommended values in 3.1.4 obey ($J = 8$ or 16).

3.4.3 ZERO-BLOCK OPTION

3.4.3.1 The Zero-Block option is selected when one or more blocks of preprocessed samples are all zeros. In this case, a single codeword may represent several blocks of preprocessed samples, unlike other options where an FS codeword represents only one or two preprocessed samples.

[1] The first extension of a preprocessed sample is the preprocessed sample itself.

3.4.3.2 The set of blocks between consecutive reference samples, *r*, as described in 4.3, is partitioned into one or more segments. Each segment, except possibly the last, contains *s* blocks. The recommended value of *s* is 64.

3.4.3.3 Within each segment, each group of adjacent all-zeros blocks is encoded by the FS codewords, specified in table 3-2, which identify the length of each group. The Remainder-Of-Segment (ROS) codeword in table 3-2 is used to denote that the remainder of a segment consists of five or more all-zeros blocks.

Table 3-2: Zero-Block Fundamental Sequence Codewords As a Function of the Number of Consecutive All-Zeros Blocks

Number of All-Zeros Blocks	FS Codeword
1	1
2	01
3	001
4	0001
ROS	00001
5	000001
6	0000001
7	00000001
8	000000001
.	.
.	.
.	.
63	0000 . . . 0000000001 (63 0s and a 1)

3.5 NO COMPRESSION

The last option is not to apply any data compression. If it is the selected option, the preprocessed block of samples receives an attached identification field but is otherwise unaltered.

3.6 CODE SELECTION

The Adaptive Entropy Coder includes a code selection function, which selects the coding option that performs best on the current block of samples. The selection is made on the basis of the number of bits that the selected option will use to code the current block of samples. An ID bit sequence specifies which option was used to encode the accompanying set of codewords. The ID bit sequences are shown in table 5-1.

4 PREPROCESSOR

4.1 PREPROCESSOR FUNCTION

4.1.1 Two of the factors contributing to the performance measure in the coding bit rate (bits/sample) of a Lossless data compression technique are the amount of correlation removed among data samples in the preprocessing stage, and the coding efficiency of the entropy coder. The function of the preprocessor is to decorrelate data and reformat them into non-negative integers with the preferred probability distribution. There are situations when a preprocessor is not necessary (see reference [B2]), and may be bypassed to provide better compression performance.

4.1.2 A preprocessor contains two functions, prediction and mapping, as shown in figure 4-1. The preprocessor subtracts the predicted value, $\hat{x}_i$, from the current data value, x_i. The resultant $(n+1)$-bit prediction error, Δ_i, is then mapped to an n-bit integer value, δ_i, based on the predictor value, $\hat{x}_i$. When a predictor is properly chosen, the prediction error tends to be small, and for some sources, has a probability distribution approaching Laplacian, for which the Adaptive Entropy Coder is optimal. There are several preprocessing techniques, of which only one, the Unit-Delay Predictor as described in 4.2, is presented in this Recommendation (see reference [B2] for predictor examples).

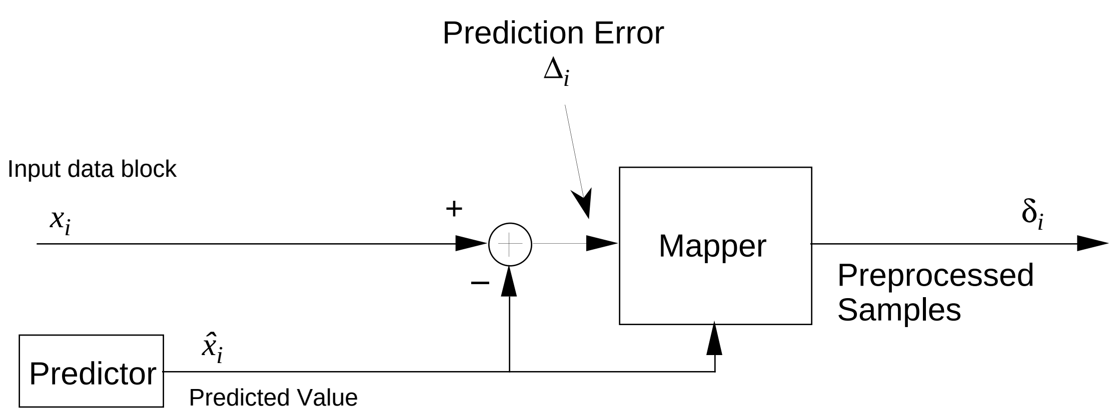


Figure 4-1: A Preprocessor

4.2 PREDICTORS

4.2.1 PREDICTION TECHNIQUES

Several preprocessing techniques can be used with the Adaptive Entropy Coder. One technique, using the Unit-Delay Predictor, is specified in 4.2.2 below. An application-specific predictor may be used instead of the unit-delay predictor, but such a predictor is unique and not specified in this Recommendation.

4.2.2 UNIT-DELAY PREDICTOR

The unit-delay predictor technique illustrated in figure 4-2 uses the one-sample delayed input data signal as the predictor for the current data signal, and the prediction error is passed to the following mapper along with the predictor value for mapping to an integer.

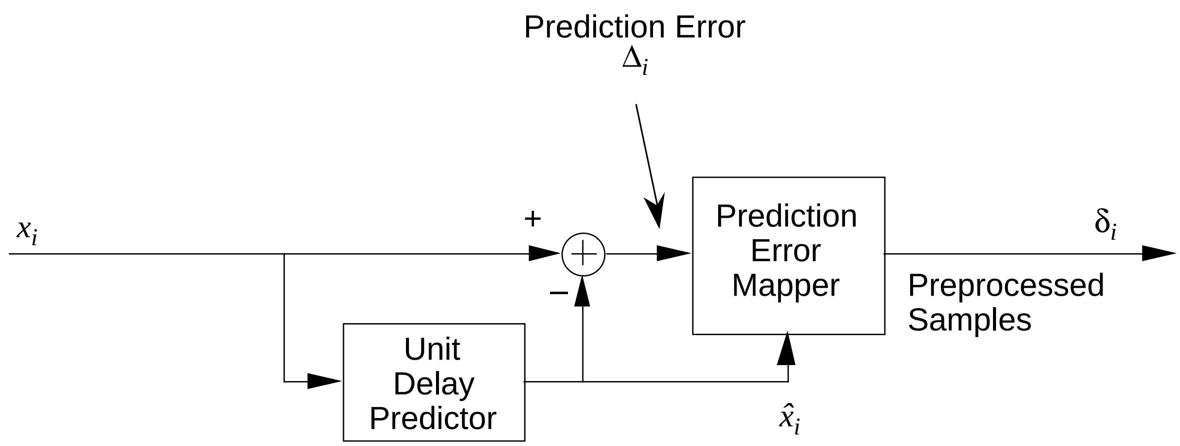


Figure 4-2: Preprocessor Using a Unit-Delay Predictor

4.3 REFERENCE SAMPLE

A reference sample is an unaltered input data sample upon which succeeding sample prediction is based. When a unit-delay predictor or other higher-order predictors that use the previous data signal in prediction are used, references are required by the decoder to recover the sample values from decoded predictor errors. When the reference sample is inserted, there are $J - 1$ preprocessed samples in the CDS. The user must determine how often to insert references. The reference sample interval, r, is limited to a maximum value of 256 CDSes (e.g., 4096 samples when $J = 16$). When reference is not required in the preprocessor, parameter r serves to define an interval of input data sample blocks that will be further segmented in the zero-block option described in 3.4.3.2.

4.4 PREDICTION ERROR MAPPER

The Prediction Error mapper takes the prediction error values and maps them into non-negative integers suitable for the Adaptive Entropy Coder. The prediction error Δ_i resulting from taking the difference between a signal value, x_i, and a predictor value, $\hat{x}_i$, both n-bit integers, will have an (n+1)-bit dynamic range of $[-2^n+1, 2^n-1]$. However, for every predictor value, there are only 2^n possible prediction error values. With a properly chosen predictor, the most probable prediction error value is zero, followed by +l and –1, +2 and –2, ..., etc. The smallest prediction error value is the difference between the minimum signal value, x_{min}, and the predictor value, $\hat{x}_i$:

$x_{min} - \hat{x}_i$. The largest prediction error value is the difference between the maximum signal value, x_{max}, and the predictor value, $\hat{x}_i$: $x_{max} - \hat{x}_i$. To map the possible 2^n prediction error values into non-negative integers, the following equation is used:

$$\delta_i = \begin{cases} 2\Delta_i & 0 \le \Delta_i \le \theta \\ 2\,|\Delta_i| - 1 & -\theta \le \Delta_i < 0 \\ \theta + |\Delta_i| & \text{otherwise} \end{cases}$$

where

$$\theta = minimum(\hat{x}_i - x_{min}, x_{max} - \hat{x}_i);$$

and for signed n-bit signal value,

$$x_{min} = -2^{n-1},\ x_{max} = 2^{n-1}-1;$$

for non-negative n-bit signal value,

$$x_{min} = 0,\ x_{max} = 2^n-1.$$

5 DATA FORMAT

5.1 LOSSLESS DATA STRUCTURES

5.1.1 GENERAL

Several parameters are required in order to transfer the adaptive variable-length losslessly coded data between the coder and the telemetry channel packet formatter.

5.1.2 OPTION IDENTIFICATION

5.1.2.1 The ID Field specifies which of the options was used for the accompanying set of samples. The ID-code keys for each of the options are shown in table 5-1.

Table 5-1: Selected Code Option Identification Key

Option	$n \leq 8$	$8 < n \leq 16$	$16 < n$
Zero Block	0000	00000	000000
Second Extension	0001	00001	000001
FS	001	0001	00001
$k = 1$	010	0010	00010
$k = 2$	011	0011	00011
$k = 3$	100	0100	00100
$k = 4$	101	0101	00101
$k = 5$	110	0110	00110
$k = 6$	—	0111	00111
$k = 7$	—	1000	01000
$k = 8$	—	1001	01001
$k = 9$	—	1010	01010
$k = 10$	—	1011	01011
$k = 11$	—	1100	01100
$k = 12$	—	1101	01101
$k = 13$	—	1110	01110
$k = 14$	—	—	01111
$k = 15$	—	—	10000
⋮	⋮	⋮	⋮
$k = 29$	—	—	11110
no compression	111	1111	11111
NOTE – '—' indicates no applicable value.			

5.1.2.2 For applications not requiring the full entropy range of performance provided by the specified options, a subset of the options at the source may be implemented. The ID is always required, even if a subset of the options is used.

5.1.3 REFERENCE SAMPLE

When the preprocessor is present and the reference sample is required, the first CDS of the Source Packet Data Field shall contain a reference sample. References shall then be inserted in the Source Packet Data Field at least every 256 CDSes as specified in 4.3. When the preprocessor is absent, or it does not require a reference sample, the reference sample shall not be inserted in the CDS.

5.1.4 CODED DATA SET FORMAT

5.1.4.1 The CDS format when a sample-splitting option is selected is shown in figure 5-1. Figure 5-1a shows the case where there is a reference sample; figure 5-1b shows the format when no reference sample is present. The CDS has the following structure when a sample-splitting option is selected: 1) ID bit sequence optionally followed by an n-bit reference sample, 2) compressed data, and 3) concatenated k least-significant bits from each sample.

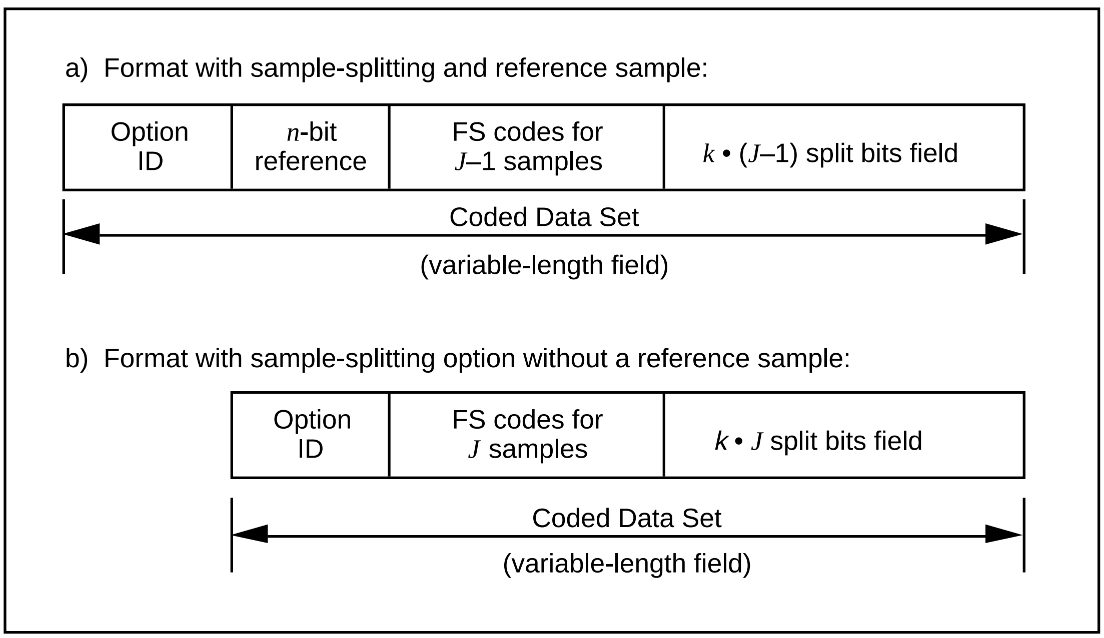


Figure 5-1: CDS Format When Sample-Splitting Option Is Selected

5.1.4.2 When the no-compression option is selected, the CDS is fixed length containing the option ID field, optionally followed by an n-bit reference sample, and J preprocessed samples. The case where a reference is present is shown in figure 5-2a; the non-reference case is shown in figure 5-2b.

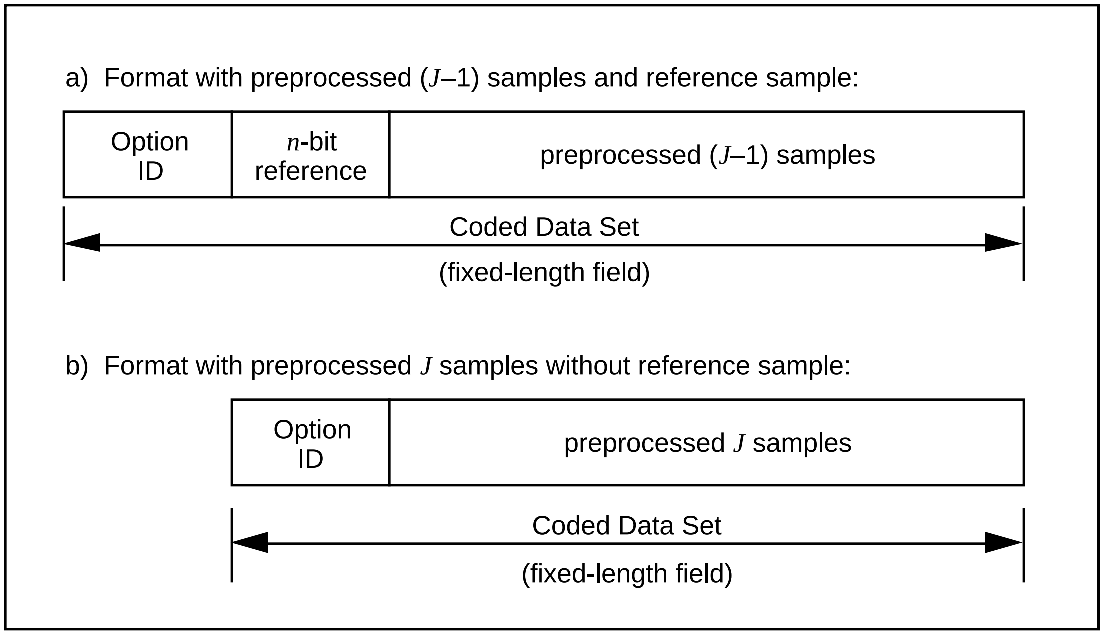


Figure 5-2: CDS Format When No-Compression Option Is Selected

5.1.4.3 When the Zero-Block option is selected, the CDS contains the option ID field, optionally followed by an *n*-bit reference sample, and a required FS codeword specifying the number of concatenated zero valued blocks or the ROS condition as described in 3.4.3. The case where a reference is present is shown in figure 5-3a; the non-reference case is shown in figure 5-3b.

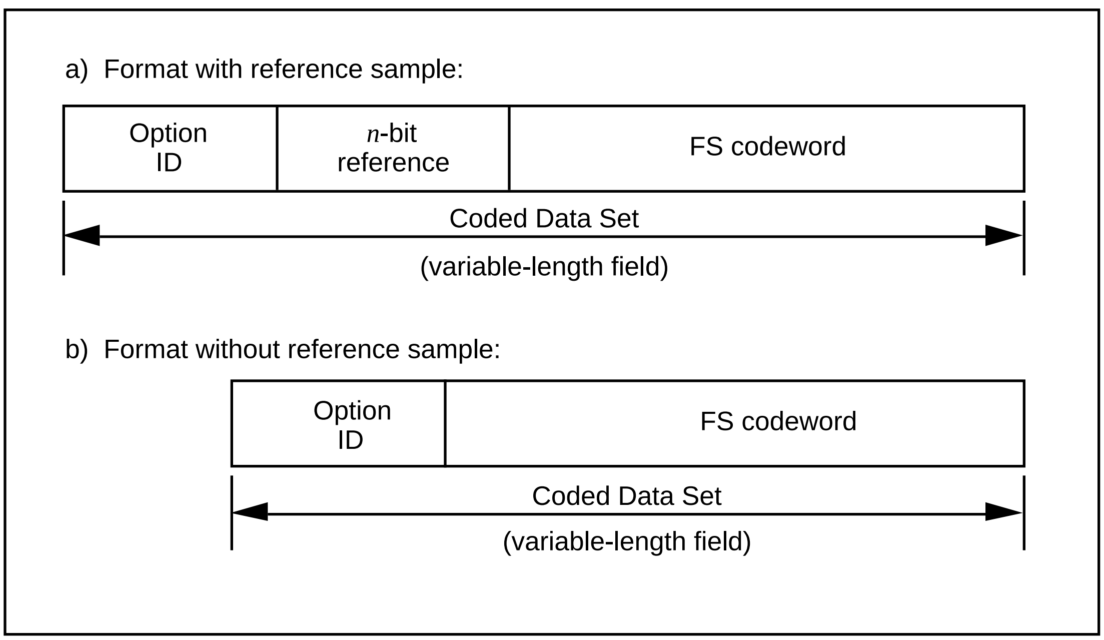


Figure 5-3: CDS Format When Zero-Block Option Is Selected

5.1.4.4 When the Second Extension option is selected, the CDS contains the option ID field, optionally followed by an *n*-bit reference sample, and required FS codewords for $\frac{J}{2}$ transformed samples. The case where a reference is present is shown in figure 5-4a; the non-reference case is shown in figure 5-4b. In the case when a reference is inserted, a '0' sample is added in front of the J–1 preprocessed samples, so $\frac{J}{2}$ samples are produced after the transformation.

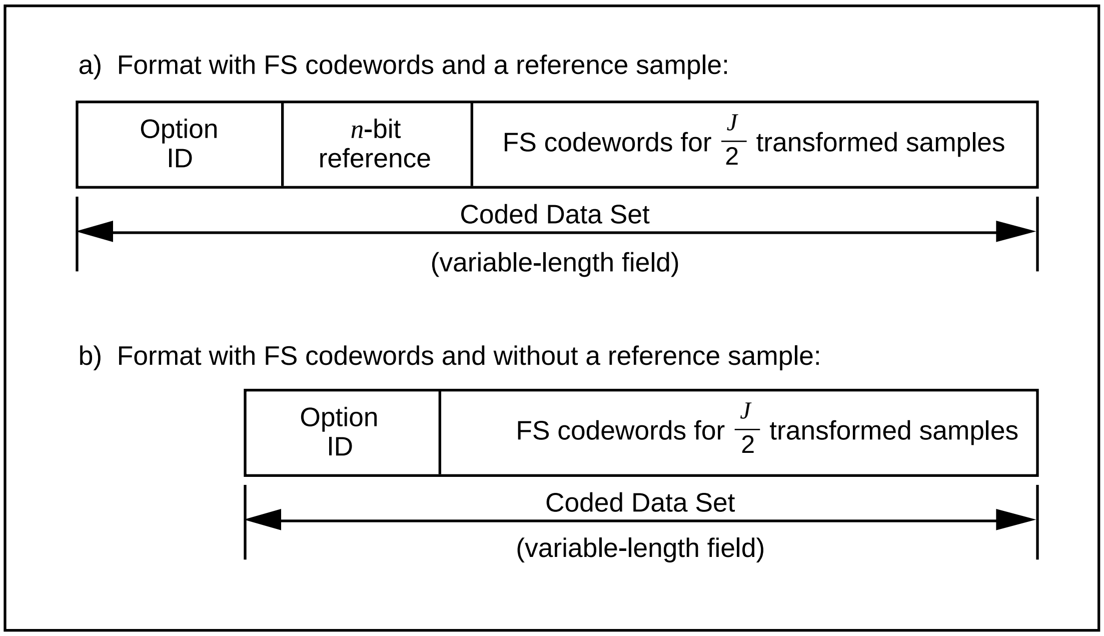


Figure 5-4: CDS Format When the Second-Extension Option Is Selected

5.2 PACKET FORMAT

5.2.1 LOSSLESS PACKET FORMAT

Lossless data compression packets shall be formatted as shown in figure 5-5 (see references [2] and [3]). The packet formatter uses the parameter provided by the source data coder to form one or more CDSes to determine the packet size in octets. Fill bits of zero value may be needed to force the packet to end on an octet boundary.

Packet Primary Header	Secondary Header (optional)	CDS #1 (with *n*-bit reference)	. . .	CDS #*l*	Fill Bits

Figure 5-5: Packet Format for *l* CDSes

5.2.2 PACKET REQUIREMENTS

5.2.2.1 A Source Packet Data Field must meet the following requirements:

– a CDS within a packet must meet the format requirements defined in 5.1;

– when the reference sample is used, the Source Packet Data Field shall begin with a CDS that contains this reference, followed by one or more CSDes; when the reference sample is not required in the preprocessor, or the preprocessor is absent, a reference sample shall not be inserted in the first CDS in the Source Packet Data Field;

– several CDSes can be put in sequence within a source packet;

– fill bits are allowed only at the end of the Source Packet Data field, not within the body of compressed data;

– each packet must end on an octet boundary.

NOTE – Some implementations may require an adequate number of fill bits be added in order to end a packet on an even-numbered octet boundary.

5.2.2.2 Unless the option to use the CIP is chosen (see section 6), in order to decode packets that may include fill bits, several pieces of information must be communicated to the decoder a priori. This information will be mission specific and fixed for a given Application Process Identifier (APID) per mission:

– l, the number of CDSes that are in a packet;

– r, the reference sample interval, equaling the number of CDSes counted from one CDS containing a reference sample up to but not including the next consecutive CDS containing a reference sample;

– n, the resolution;

– J, the number of samples per block.

5.2.2.3 A Packet Secondary Header is optional and can be used, for example, to relate observation time and position information to the user (see references [2] and [3]).

5.2.2.4 The use of the Grouping Flags in the Packet Sequence Control Field is optional and can be used, for example, to signal a group of compressed data packets. Their use is governed by reference [2].

6 COMPRESSION IDENTIFICATION PACKET (OPTIONAL)

6.1 COMPRESSION IDENTIFICATION PACKET STRUCTURE

6.1.1 When the compressed data are transmitted as grouped source packets, a Compression Identification Packet (CIP) is an optional packet that, if used, shall precede and provide configuration information for a group of compressed application data packets. The CIP will be transmitted from an application process in space to one or several sink processes on the ground.

6.1.2 The CIP shall be the first packet of the group.

6.1.3 The CIP shall consist of two major fields positioned contiguously in the following sequence: Packet Primary Header and the Packet Data Field. See figure 6-1.

6.1.4 The CIP shall contain information that would allow the decompressor to be automatically configured to acquire a group of compressed application data packets without the need for managing a-priori information. The CIP shall be utilized to configure the decompressor automatically only if there is a reliable system for file transfer.

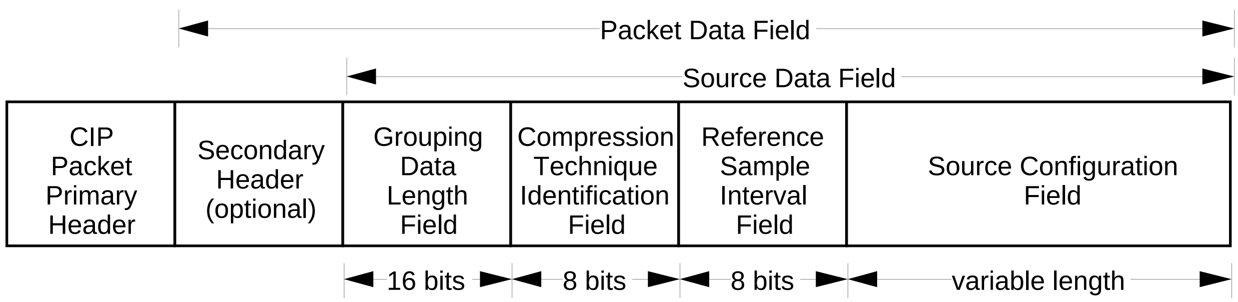


Figure 6-1: Compression Identification Packet Structure

6.2 CIP PRIMARY HEADER

6.2.1 GENERAL

The Packet Primary Header is mandatory for the CIP and its structure shall conform to the CCSDS Packet Telemetry Blue Book, reference [2]. The CIP Packet Primary Header Field shall contain the source data APID. The use of the CIP will be mission specific and fixed for a given APID.

6.2.2 GROUPING FLAGS

6.2.2.1 The Grouping Flags are in the packet Sequence Control field, as specified in reference [2]. The field is located in the Packet Primary Header of packets encapsulating compressed user data. As indicated below, the field is always ‘01’ for the CIP Primary Header.

6.2.2.2 The Grouping Flags shall be set as follows:

- ‘01’ for the group’s first packet, which is the CIP;
- ‘00’ for the continuing source packets with compressed data of the group;
- ‘10’ for the last source packet with compressed data of the group.

6.2.2.3 For a source packet not belonging to a group of source packets with compressed data, the Grouping Flags shall be set to ‘11’.

6.3 PACKET DATA FIELD

6.3.1 GENERAL

The Packet Data Field of a CIP shall consist of two fields positioned contiguously in the following sequence: Packet Secondary Header (optional) and the Source Data Field.

6.3.2 SECONDARY HEADER (OPTIONAL)

The Secondary Header is a means for placing ancillary data such as time and spacecraft position/attitude information with the CIP.

6.3.3 SOURCE DATA FIELD

6.3.3.1 General

The Source Data Field for the CIP shall consist of four fields positioned contiguously in the following sequence:

	Length (bits)
The Grouping Data Length	16
Compression Technique Identification	8
Reference Sample Interval	8
Source Configuration	(Variable)

6.3.3.2 Grouping Data Length

The Grouping Data Length is a 16-bit field of which the first four bits are reserved. The remaining 12 bits of the field shall contain a binary number equal to the number of packets containing compressed data within the group minus one, with the number of packets containing compressed data ranging from 1 to 4096. The number of packets in the group with the CIP included shall range from 2 to 4097.

6.3.3.3 Compression Technique Identification Field

6.3.3.3.1 The Compression Technique Identification (CTI) field shall signal the compression technique in use for the group of source packets identified by the CIP.

6.3.3.3.2 When no compression technique for the current group is used, the CTI field shall be set to all zeros.

6.3.3.3.3 Only the Lossless data compression technique is currently defined, and is signaled by the value 1 in the CTI field. Other values are reserved for future use by CCSDS and are not permitted.

6.3.3.4 Reference Sample Interval Field

6.3.3.4.1 The reference sample interval, r, equals the number of CDSes counted from one CDS containing a reference sample up to but not including the next consecutive CDS containing a reference sample. When the preprocessor is absent, or it does not require a reference sample, the reference sample shall not be inserted in the CDS; nevertheless, parameter r serves to define the interval of input data sample blocks for the zero-block option as described in 4.3.

6.3.3.4.2 The 8-bit r field shall contain a binary number equal to $r - 1$, with the value of r ranging from 1 to 256.

6.3.3.5 Source Configuration

6.3.3.5.1 Subfield Partitions

6.3.3.5.1.1 The Source Configuration field shall be partitioned into three subfields: Preprocessor, Entropy Coder, and Instrument Configuration (see figure 6-2). The Preprocessor and Entropy Coder subfields are required, whereas the Instrument Configuration subfield is optional. Each of these Compressor Configuration subfields shall have a header as the first two bits to identify the subfield type.

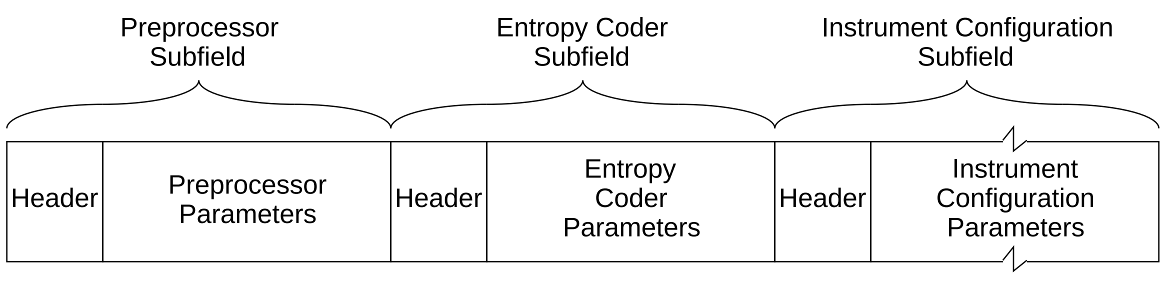


Figure 6-2: Source Configuration Field

6.3.3.5.1.2 The subfield type flags shall be set as follows:

00 – Preprocessor
01 – Entropy Coder
10 – Instrument Configuration
11 – Reserved

6.3.3.5.2 Preprocessor

6.3.3.5.2.1 The length of the Preprocessor subfield shall be two octets, the first two bits of which shall be the header as described in 6.3.3.5.1.2.

6.3.3.5.2.2 The Preprocessor parameters for the Lossless data compressor shall be partitioned into six areas and shall be positioned contiguously following the 2-bit Preprocessor header. See 3.1 and section 4 for preprocessor parameter definitions. The six areas are

a) Preprocessor Status (1 bit)

0 – absent
1 – present

b) Predictor type (3 bits); ignore if preprocessor status is '0':

000 – bypass predictor
001 – unit delay predictor
111 – application-specific predictor
All other codes are reserved by CCSDS for future preprocessing options.

c) Mapper type (2 bits); ignore if preprocessor status is '0':

00 – Prediction Error mapper described in 4.4
01 – reserved
10 – reserved
11 – application-specific mapper

d) Block size (J) (2 bits):

	Number Sample/Block
00	8
01	16
10	reserved
11	application specific

e) Data sense (l bit):

0 – two's complement
1 – positive; mandatory if preprocessor is bypassed

f) Input data sample resolution (*n*) (5 bits):

The 5-bit Input Data Sample field shall contain a binary number equal to the input data sample resolution minus one, with the data sample resolution ranging from 1 to 32.

6.3.3.5.3 Entropy Coder

6.3.3.5.3.1 The length of the Entropy Coder subfield shall be two octets, the first two bits of which shall be the header as described in 6.3.3.5.1.2.

6.3.3.5.3.2 The Entropy Coder parameters subfield shall be partitioned into two areas and shall be positioned contiguously following the 2-bit Entropy Coder header. The two areas are

a) Entropy Coder option (2 bits):

A maximum of 32 options are permitted (see table 5-1 for option identification keys):

00 — Spare
01 — for $n \le 8$
10 — for $8 < n \le 16$
11 — for $16 < n \le 32$

b) Number of CDSes per packet, *l* (12 bits):

The 12-bit field indicating the number of CDSes per packet (*l*) shall contain a binary number equal to $l - 1$.

6.3.3.5.4 Instrument Configuration

The Instrument Configuration Subfield (ICS) is an instrument-unique field used to address unique instrument configuration parameters. The contents of this field are mission specific and are beyond the scope of this Recommendation. If used, the first two bits shall be the header as specified in 6.3.3.5.1.2.

ANNEX A

GLOSSARY OF ACRONYMS AND TERMS

(This annex **is not** part of the Recommendation.)

A1 PURPOSE

This annex defines key acronyms and terms that are used throughout this Recommendation to describe source coding for data compression.

A2 ACRONYMS

Acronym	Definition
CDS	coded data set
CIP	compression identification packet
CTI	compression technique identification
FS	fundamental sequence
ICS	instrument configuration subfield
LSB	least significant bit
MSB	most significant bit
ROS	remainder of segment

A3 TERMS

ADAPTIVE ENTROPY CODER: An entropy coder codes the source samples with uniquely decodable codewords that, upon decoding, reconstruct the source samples. With an Adaptive Entropy Coder, the average codeword length also follows closely the information content of the source.

ENTROPY: Entropy is a quantitative measure of the average amount of information per source sample, expressed in bits/sample.

FUNDAMENTAL SEQUENCE: The variable-length Fundamental Sequence (FS) code represents the non-negative integer m with a binary codeword of m zeros followed by a 1. Application of the FS code to a block of J samples produces a sequence of J concatenated codewords called the Fundamental Sequence.

RICE’S ADAPTIVE CODING: The basic Rice adaptive coding algorithm chooses the best of several code options to use on a block of data. These options are targeted to be efficient over different ranges of data activity. The options are implemented using a combination of FS coding and the splitting of preprocessed samples into their most-significant and least-significant bit parts.

SAMPLE SPLITTING: Sample splitting is a procedure for separating the binary representation of a sample into two groups of adjacent bits, one for lower-order bits, the other for higher-order bits.

SPLIT BITS: Split bits are the lower-order bits separated by sample splitting from the binary representation of a sample.

ANNEX B

INFORMATIVE REFERENCES

(This annex **is not** part of the Recommendation.)

[B1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-7. Yellow Book. Issue 7. Washington, D.C.: CCSDS, November 1996 or later issue.

[B2] *Lossless Data Compression.* Report Concerning Space Data Systems Standards, CCSDS 120.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, May 1997.

[B3] *Telemetry Summary of Concept and Rationale*. Report Concerning Space Data Systems Standards, CCSDS 100.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, December 1987.