Mod 3

SFOC0038-03-07-03

SFOC Software Interface Specification (SIS) Module	Name: SFOC-5-TIS-*DU-SFDU Publication Date: 09-06-89 CR 1695

MODULE TITLE: SFDUs Generated/Received by TIS

PURPOSE: Document describes the format of the TIS-encountered SFOC Standard Formatted Data Unit (SFDU) types, both received and generated.

DATA FLOW:

From	To	Data Object	Physical Medium
TIS	Any	SFDU	SFOC LAN

PREPARER(S): John Diehl, N. Pevyhouse, Tieh Ku, Jan Gohlke, Mike Tankenson

Approval:	Name	Date
Technical Group Supervisor	Jim Wilson	9/15/89
TIS System Engineer	Mike Tankenson	9/15/89
MGN TLM Subsystem Engineer	Betsy Wilson	9/15/89
JPL Control Authority	John Johnson	9/15/89
SFOC System Engineer	Jay Holladay	9/15/89
SFOC System Engineer	Gil Dawson	9/15/89

DOCUMENT LOG

Date	Page No.	Status
03-30-87	All	Draft
07-01-87	All	Third Draft
11-16-87	All	Prelim.
11-25-87	All	Draft
01-08-88	All	Prelim.
08-31-88	All	Final
03-16-89		Rev A to Final

Date	Page No.	Status

LIST OF TBD ITEMS

Page	Section	Item
4-10	4.2.1.5.1	GIF usage of the flag dsn-dodr SFDUs received via SFOC gateway.
4-11	4.2.1.5.1	GIF usage of the NASCOM seq_no field from the user header vs. the block serial number before the NASCOM trailer.

DISTRIBUTION

Dalaney, R.	301-260
Dawson, G.	264-728
Diehl, J.	301-260A
Haupt D.	301-260
Holladay, J.	264-728
Johnsen, J4.	264-728
Pevyhouse, N.	301-260
Tankenson, M.	301-260
Wilson, E.	301-260A
Wilson, J.	301-260A

cc:

R. Hall	171-232
K. Morrissett	T-1607
B. Polansky	234-728
A. Sacks	234-728

TABLE OF CONTENTS

CONTENTS (continued)

SECTION .1

GENERAL DESCRIPTION

1.1 PURPOSE OF DOCUMENT

This document describes the format of GIF-encountered and TIS-encountered (received and generated) SFOC Standard Formatted Data Unit (SFDU) types.

1.2 SCOPE

This document describes the format of SFDU primary labels and subheaders used in SFDUs that are received and produced by GIF and TIS. Only baseline SFOC structures are included. For mission-specific subheaders, refer to applicable document #4 for specific modules with the title SFOC-5-TIS-*DU-(mission) SFDU (mission = Mgn, MO, etc.). For a definition of the structures constituting the data portion of SFDUs, consult Applicable Document #4c.

For general information about SFDU structuring conventions, Compressed Header Data Objects (CHDOs), and the assignment of record identifiers (major type, minor type, format, and mission ID) to SFOC-generated SFDUs, refer to Applicable Document #4a.

Data structures are described in terms of Global Data Representation (GDR). See Applicable Document #4a for descriptions of these quantities.

1.3 APPLICABLE DOCUMENTS

1. VRM-2-280 — Venus Radar Mapper, Full Scale Development, Spacecraft System and Subsystem Design Book, Telemetry Measurements and Data Formats, March 1987.

2. Document 820-13 — DSN System Requirements, Detailed Interface Design

   a. Module TLM-3-17 — DSN Telemetry Interface with SFOC - Magellan, February 1, 1988.

   b. Module MON-5-12 — DSN Monitor and Control System Interface with Magellan Project Telecommunication Link Analysis, February 1, 1988.

3. JPL D-1672 — Mars Observer, Spacecraft Data Standards, MO 642-530, July 15, 1988.

4. JPL D-4675 SFOC0038-01-01-01, Space Flight Operations Center, Software Interface Specification, November 1987.

   a. Module, SFOC-5-SYS-*DU-NJPL, NJPL SFDU Global Definitions, Second Preliminary version, May 9, 1988.

   b. Module, SFOC-5-TIS-*DU-MgnSFDU, SFDUS Generated From TIS for Magellan, May 16, 1988.

   c. Module, SFOC-2-TIS-Any-Telem, Telemetry Input Subsystem Telemetry Products, February 29, 1988.

5. JPL D-4037 STIS-0004-02-01, Space Flight Operations Center, Telemetry Input Subsystem, Functional Requirement, March 3, 1988.

6. MGN SRD WE2-03 Magellan Spacecraft Project, Mission Operations System Requirements, Telemetry Processing Subsystem: MRO, DED, Flood, StarCal, Normal MRO Processing, SFOC-TISMGN-09, June 23, 1987.

1.4 FUNCTIONAL DESCRIPTION

1.4.1 Data Source, Destination, and Transfer Method

1.4.1.1 Data Sources

Real-time telemetry flows from the spacecraft via the DSN through the GIF subsystem to TIS. Telemetry that has been recorded at a ground station and is replayed via the DSN is treated by SFOC in the same way as that of real-time telemetry. A ground station is the source for monitor data.

For non-realtime data processing, the source can be a spooler file or a magnetic tape.

1.4.1.2 Destinations

The destination for any particular SFDU type can be selected by the user via user-control directives. Destinations may be spooler files and other SFOC subsystems reached via virtual circuits or broadcast channels over the SFOC LAN. Transfer methods utilize DTS services to transfer SFDUs over the SFOC LAN. CDA services are used to log and retrieve SFDUs to/from spooler files.

1.4.2 Pertinent Relationships with Other Interfaces

Changes to this SIS module may affect Applicable Document 4a, and changes to Applicable Document 4a may affect this module.

1.4.3 Labeling and Identification

SFDU elements are labeled and described in Sections 4.2 and

1.4.4 Assumptions and Constraints

All numeric values in this document are decimal unless stated otherwise.

SECTION .2

ENVIRONMENT

2.1 HARDWARE CHARACTERISTICS AND LIMITATIONS

Not Applicable.

2.2 INTERFACE MEDIUM AND CHARACTERISTICS

DTS and CDA services are the interface media for SFDU records over the SFOC LAN.

2.3 INPUT/OUTPUT PROTOCOLS

2.3.1 Device Addressing

Not Applicable.

2.3.2 Operating System Protocols

Not Applicable.

2.3.3 Deformatting

SFDUs are passed in a global data representation.

2.3.4 Calling Sequence

Not Applicable.

2.4 FAILURE PROTECTION, DETECTION, AND RECOVERY FEATURES

2.4.1 Backup Requirements

Depending on user-control directive action, SFDUs are logged to disk files and are therefore recoverable. In addition, a data stream can be replayed to recover lost data.

2.4.2 Security/Integrity Measures

Not Applicable.

2.5 END-OF-FILE (OR MEDIUM) CONVENTIONS

Not Applicable.

2.6 UTILITY PROGRAMS

Not Applicable.

SECTION .3

DATA FLOW CHARACTERISTICS

3.1 OPERATIONAL CHARACTERISTICS

3.1.1 Generation Method and Frequency

SFDUs are generated by the GIF and TIS subsystems within SFOC at a frequency that depends on a combination of the data rate and record length.

3.1.2 Time Span of Product

The time span of a telemetry SFDU record can be calculated by determining the SFDU record's Earth Received Time (time the first data bit was received by ground station), the data rate (bits per second), and the amount of data in the SFDU. The time span of non-telemetry SFDU records varies.

3.2 DATA VOLUME

Data volume depends entirely on the data rates used by specific missions. Refer to mission-specific SIS modules for detailed descriptions.

3.3 FLOW RATE

Not Applicable

SECTION .4

DETAILED DATA OBJECT DEFINITION

4.1 STRUCTURE AND ORGANIZATION OVERVIEW

An SFDU is the logical record structure used to contain telemetry data and SFOC system data for transfer between SFOC users. An SFDU consists of a header portion and a data portion. The header is the primary label that conforms to CCSDS standards. For SFOC-generated telemetry SFDUs, the data part is composed of compressed header data objects (CHDOs). For non-SFOC-generated SFDUs, the CHDO concept is not always used. For more information about compressed header data objects and the rules that govern the composition of SFDUs from these parts, refer to Applicable Document #4a.

The components of a SFOC-generated telemetry SFDU are:

- Primary label
- Subheader aggregation CHDO
- Primary header
- Secondary header (optional)
- Tertiary header (optional)
- Quaternary header (optional)
- Data CHDO (optional).

Telemetry SFDUs are identified by Data Definition Record (DDR) ID in the primary label and/or by major, minor, mission ID, and format types in the primary header. The combination of header types used in SFDU structures is identified in SFOC-5-SYS-*DU-NJPL (Applicable Document #4a).

The SFDU structure fields are described following each structure. All numeric values are in decimal unless otherwise indicated.

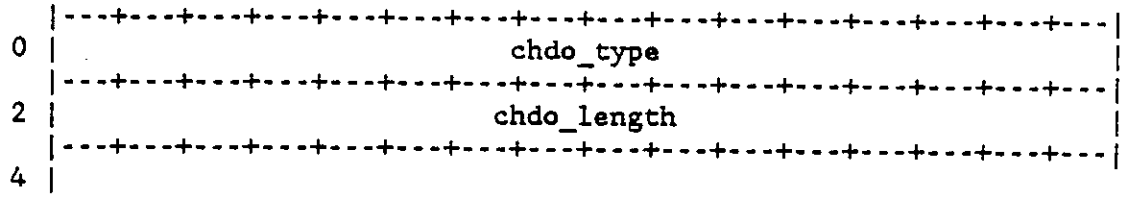
As specified in Applicable Document 4a, when CHDOs are present the subheader aggregation CHDO is mandatory. Its data are the combined subheaders. Its length thus provides identification and easy access to the data CHDO.

There is only one CHDO type that can appear in the primary header position. It must always be present in that position. It uniquely identifies the record and determines the CHDO composition of the remainder of the SFDU.

The number and types of additional subheaders vary. The terms Secondary Header, Tertiary Header, etc. refer to the position of a header CHDO within the SFDU. They do not imply that a given header CHDO must always appear in the same header position if it is used in more than one type of SFDU. It is permissible for a given header CHDO to appear, for example, as a secondary header in one SFDU and as a tertiary header in a different SFDU. Use of a null CHDO (described below) as a placeholder in the secondary, tertiary, and quaternary header positions is permissible.

A data CHDO is optional. If the SFDU requires no actual data, it may end with the last header. However, a placeholder CHDO may be provided. Programs that have to handle a multitude of SFDUs should test for the presence of a data CHDO by comparing the length fields of the primary label and the subheader aggregation CHDO.

The placeholder consists of a CHDO type and CHDO length as follows:



Byte Offset	Field Id	Description
0-1	chdo_type	Null CHDO type code. Indicates a header is omitted. (Value = 0).
2-3	chdo_length	Length of CHDO value field (remainder of CHDO). (Value = 0).

A more detailed description of the rules for composing SFOC standard CHDO-structured SFDUs is contained in Applicable Document #4a.

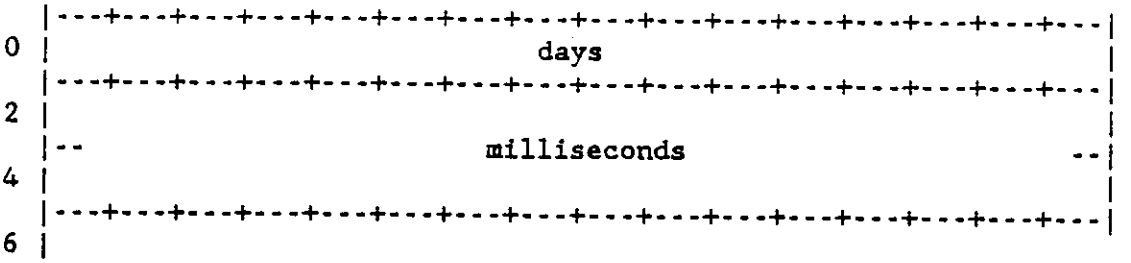
4.2 DATA FORMAT AND DEFINITION

4.2.1 Format

4.2.1.1 Externally Declared Structure Templates

The following structure is declared externally. See Applicable Document #4a for details.

time_dcl



Byte Offset	Field Id	Description
0-1	days	Days since January 1, 1958, starting with 0.
2-5	milliseconds	Milliseconds of current day.

4.2.1.2 JPL Primary SFDU label

This label is registered with the Control Authority in order to identify the SFDU; the label describes the structure and contents of subsequent information/data.

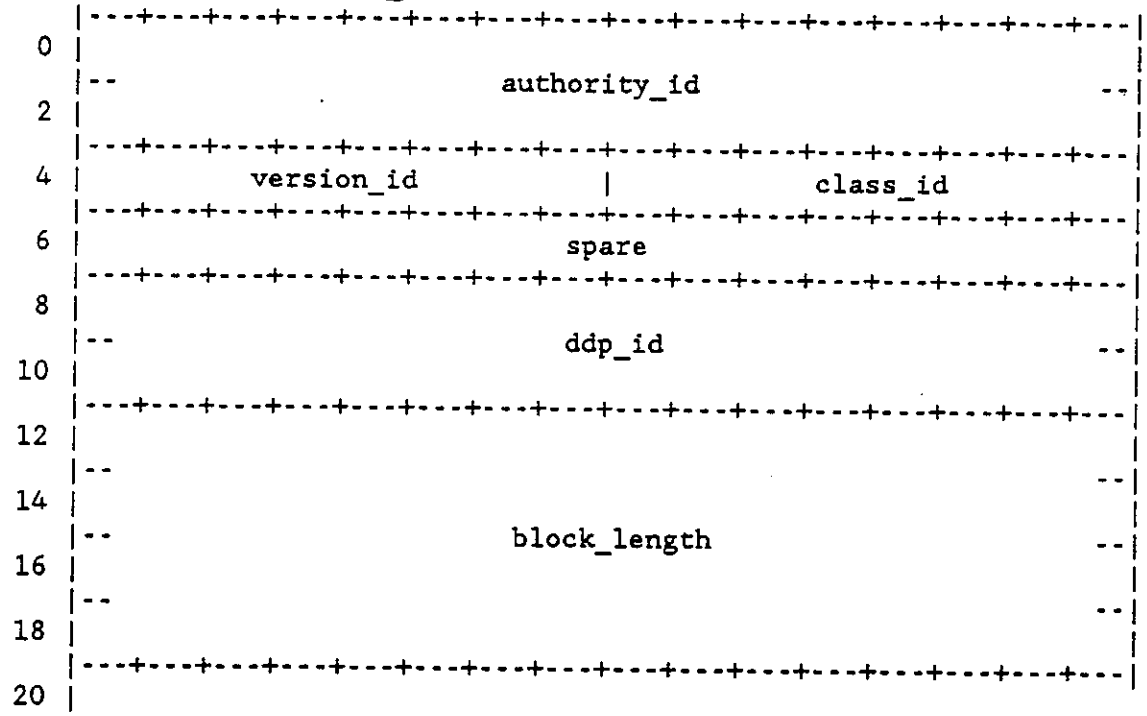
Some SFDUs encountered in the SFOC system will have only this label as a reference to identify the data that follows because the CHDO concept is not used. For such records a unique DDR-ID value will be used to distinguish the different records. Examples are SCLK/SCET coefficient records and orbit header records.

SFDUs which have CHDOs will always have a subheader aggregation CHDO and at least a primary header CHDO that allows further identification of the data by major type, minor type, format, and mission ID.

DSN-generated records have the same primary label structure but different label content. They also are composed of CHDOs, but the aggregation CHDO may not always be present. This module includes descriptions of DSN-generated primary labels and header CHDOs. To determine the composition of a complete SFDU from these parts, the reader should consult Applicable Document #4a.

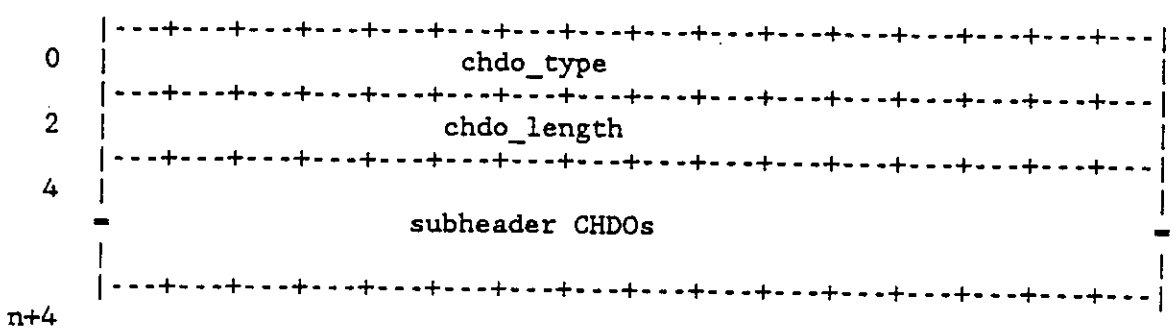
The primary label for all SFOC-generated and DSN-generated SFDU follows CCSDS standards and is structured as follows:

(structure primary_label)



Byte Offset	Field Id	Description
0-3	authority_id	JPL control authority identifier. (Value = CCSD or NJPL, Restricted ASCII (RA). (Restricted ASCII characters are the capital letters A-Z and the digits 0-9.)
4	version_id	Version ID for length field. Value = 1 (RA), which determines that block_length is an 8-character restricted ASCII field. Value = 2 (RA), which determines that block_length is a 64-bit unsigned integer.
5	class_id	Identifies the label class. Values in restricted ASCII (RA) that are known to SFOC are: Z - DSN telemetry primary label I - DSN monitor primary label I - SFOC primary label. K - SFOC keyword-value label.
6-7	Spare.	
8-11	ddp_id	Data Description Package identifier. Must be assigned by the JPL SFDU control authority. For a list of valid values, see Applicable Document #4a.
12-19	block_length	For version_id = 2, 64-bit unsigned integer indicating the length of the remainder (starting with byte 20) of this SFDU in bytes. Values are 0 - 32,746 and must be even. Therefore, bytes 12-17, the most-significant bit of byte 18, and the least-significant bit of byte 19 must all be zeroes. For version_id = 1, 8 characters (RA) indicating the length of the remainder (starting with byte 20) of this SFDU in bytes.

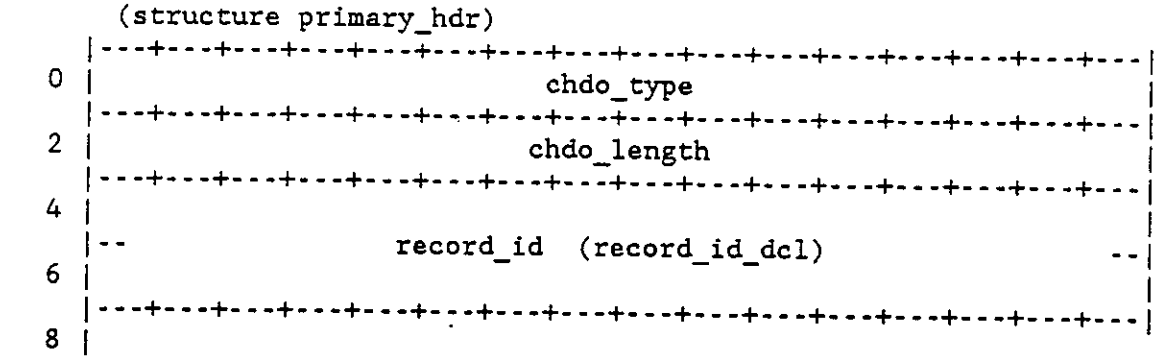
4.2.1.3 Subheader Aggregation CHDO



Byte Offset	Field Id	Description
0-1	chdo_type	Sub-header aggregation type code. (Value = 1).
2-3	chdo_length	Length of the combined size, n, of all subheaders that follow, in bytes, starting at the following byte. This provides an offset to the start of the data block (chdo_type).
4-n +3	subheader CHDOs	Combined subheader CHDOs occupying a total of n bytes.

4.2.1.4 Primary Header

The primary header is introduced by the DSN and is used for all SFOC-generated SFDUs.

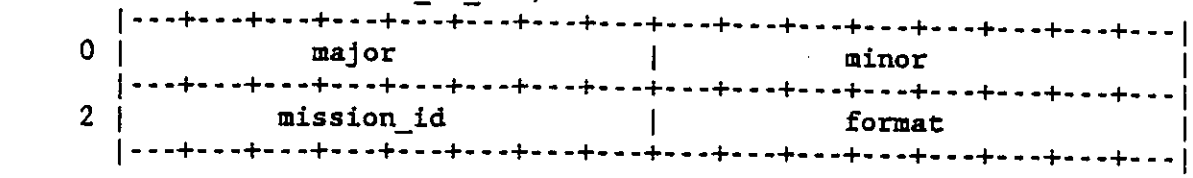


Byte Offset	Field Id	Description
0-1	chdo_type	Telemetry primary header CHDO type code. (Value = 2).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 4).

4-7 record_id Record identifier (described below in structure record_id_dcl, section 4.2.1.4.1).

4.2.1.4.1 Record_ID

(structure record_id_dcl)



Byte Offset	Field Id	Description
0	major	SFDU major type. Types 0-127 are reserved for common types. Specifies the major data categorization. For the complete list of assigned values refer to Applicable Document #4a.
1	minor	SFDU minor type. Types 0-127 are reserved for types that are defined and apply to more than one mission. Types 128-255 are used for mission-unique data. See Applicable Document #4a for a list of assigned types and guidelines for the assignment of identifiers.
2	mission_id	Mission identifier code. For a list of valid values, see Applicable Document #4a.
3	format	SFDU format type. This field is used in conjunction with major and minor to specifically define data types, which are mission-specific and defined in the mission-specific SFDU SIS module. Values are assigned as described for the minor type field above. See Applicable Document #4a for a list of assigned format values.

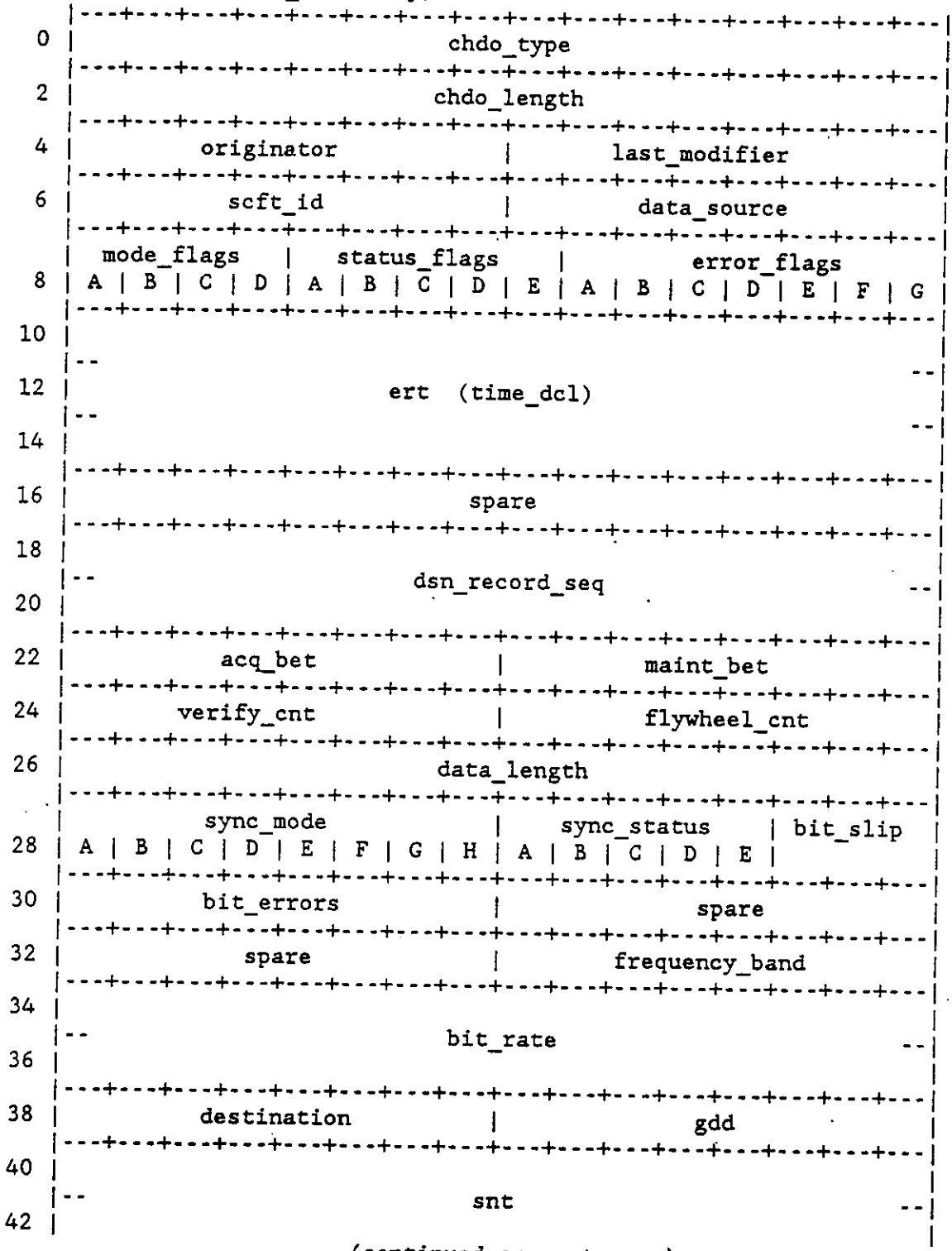
4.2.1.5 Secondary Headers

If a secondary header is present, it follows the primary header and has a structure which depends on the record ID.

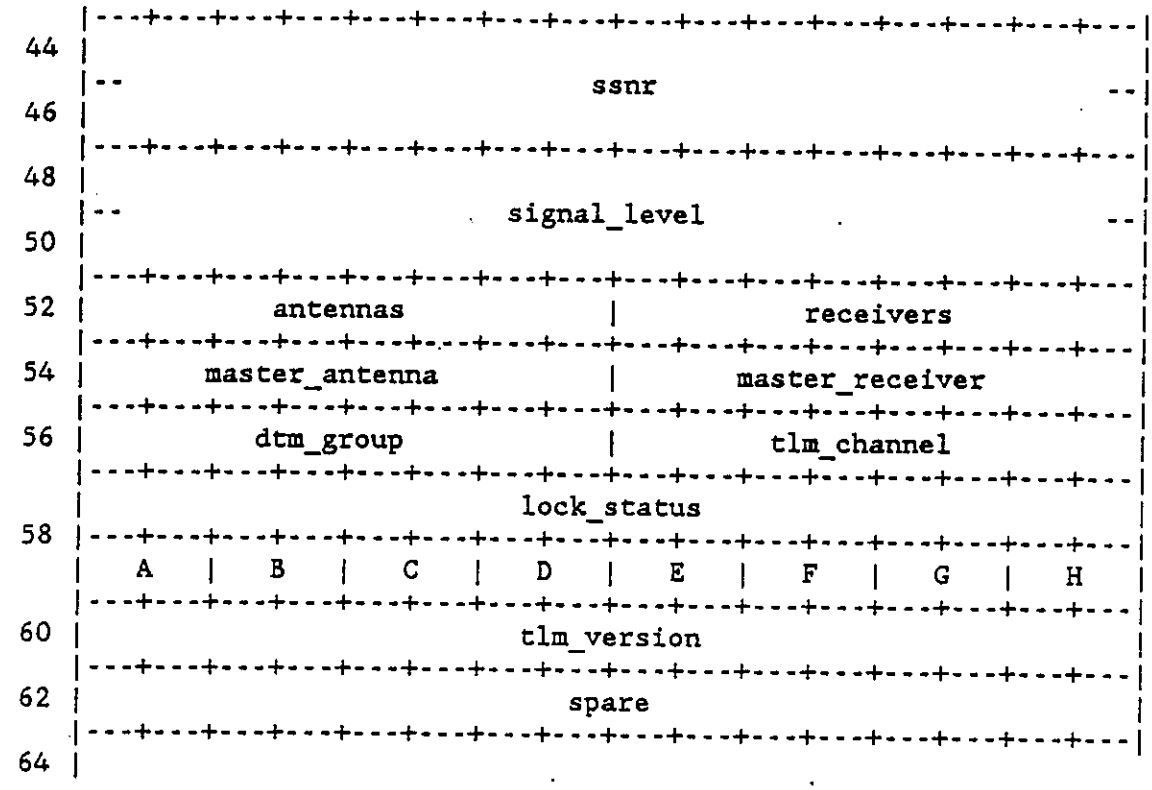
4.2.1.5.1 Secondary Header - DSN Telemetry

This SFDU header appears on all raw telemetry (synced and unsynced) received from the DSN. This header is reproduced here for completeness. The controlling document for the format of this header is Applicable Document #2a.

(structure dsn_secondary)



(continued on next page)



Most fields also occur in the same position in the GIF and TIS telemetry secondary headers. Some of these fields are not used by the DSN but are used later by GIF or TIS. Where this is the case, the use of this field along the path from DSN to TIS is explained.

Byte Offset	Field Id	Description
0-1	chdo_type	DSN telemetry secondary header CHDO type code. (Value = 3).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 60).
4	originator	Originator ID. Set to 48 by DSN.
5	last_modifier	Last modifier ID. Set to 48 by DSN. Changed to the GIF identifier in the GIF telemetry secondary header; changed to the TIS identifier in the TIS telemetry secondary header. For a list of valid values, see Applicable Document #4a.
6	scft_id	Spacecraft identifier (ref. 820-13 OPS 6-8 for code value).
7	data_source	DSN station ID of data source (ref. 820-13 OPS 6-8 for code value).

8	mode_flags	
	A pb_mode	Spacecraft realtime/playback flag. Set to 0 by DSN. Copied by GIF. Set by TIS for data from spacecraft tape recorder. Valid values are: 0 = Realtime or unknown 1 = Playback.
	B data_mode	Indicates whether data is real or simulated. Set to 0 by DSN. Set to 1 by SIM. Valid value are: 0 = Real or unknown 1 = Simulated.
	C test_mode	Indicates whether data is test- or flight-generated. Set to 1 by DSN. Valid value are: 0 = Test complex/GDS test generated 1 = Flight (MOS) generated.
	D replay_flag	SFOC realtime/replay flag. Set to 0 by DSN. Set to 1 by GIF when data is replayed from spooler files, IDR tapes, or SFDU tapes; otherwise set to 0. Set to 1 by TIS when data is replayed from spooler files or SFDU tapes; otherwise copied. Valid values are: 0 = SFOC realtime 1 = SFOC replay.
8-9	status_flags	
	A data_val	Data validity (0 = good data). Set to 0 by DSN. Set to 1 by GIF and TIS for anomaly records. Anomaly records have at least one flag in field anomaly_flags (see Secondary Header - GIF Telemetry) set, and their normal data CHDO is replaced by a null CHDO.

B dsn_val

Validity of antenna and DSN frame sync parameters. Set to 0 by DSN. Set to 0 by GIF when telemetry is received embedded in SFDUs, regardless whether or not those SFDUs are, in turn, embedded in GCF blocks. Set to 1 by GIF in the GIF secondary header when a record is constructed from GCF/NASCOM blocks containing only raw telemetry, i.e., no SFDU headers. Copied by TIS. In the GIF telemetry secondary header this flag, when set, means that the fields listed under a and b below are invalid and contain 0. In the TIS telemetry secondary header, group a fields are not present. Therefore, the flag applies to group b only:

a. acq_bet, main_bet, verify_count, flywheel_count, sync_mode, sync_status, bit_slip, bit_errors

b. antennas, receivers, master_antenna, master_receiver, dtm_group, tlm_channel, frequency_band.

C Spare

Set to 0 by DSN.

D ert_val

Earth received time (0 = ert is valid, 1 = ert is known to be bad).

E dsn_dodr

DSN DODR replay. Indicates whether this record was created as a result of a DSN DODR replay. Set to 0 by DSN. GIF uses this flag as follows:

a. For input received in GCF blocks (SFDUs or raw tlm), the flag is set/reset based on content of GDD-field in GCF block header.

GDD value	dsn_dodr value
0	0
1	1
3	0
4	1
6	0
7	0

b. For NASCOM blocks, the flag is not set.

c. For SFDUs received via SFOC gateway: TBD

TIS copies the flag. Valid values are:

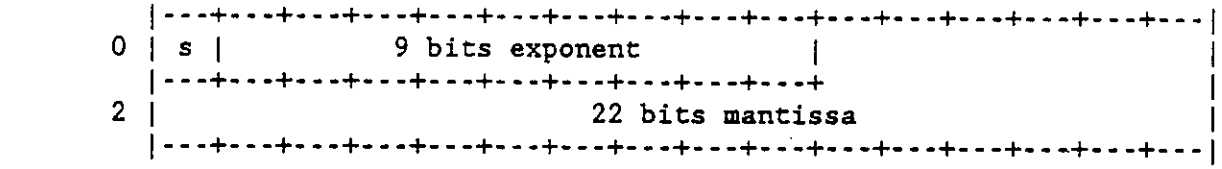
0 = No
1 = Yes.

9	error_flags	
	A checksum_err	Checksum error detected at FSS-DDR interface. Set by DSN. Valid values are: 0 = good data 1 = checksum error
	B-C GCF_block_err	Error status code from GCF block trailer. Set to 0 by DSN. When GIF receives GCF or NASCOM blocks, it puts the error-status code from the block trailer into this field. For other input, this field is copied. TIS just copies it. Valid value are: 00 - good data 1X - transmission error X1 - other GCF interface error.
	D-G spare.	
10-15	ert	Earth Received Time. DSN sets this field to the ERT of the last telemetry bit in the record. GIF adjusts the time for bit offset and data rate, so this field reflects ERT of the first bit. (In records received by GIF, ERT refers to the first bit in NASCOM blocks and the last bit in SFDUs and GCF blocks.) In TIS-created records this time always refers to the first bit of a minor frame, packet, or other block of telemetry.
16-17	spare	Set to 0 by DSN.
18-21	dsn_record_seq	Increments for every SFDU, starting at 1. Wraps to 0. This number is supplied in DSN-generated telemetry SFDUs and copied for GIF- and TIS-produced records. When GIF creates SFDUs from received GCF blocks that contain raw telemetry without SFDU headers, GIF puts the block serial number of the block (from which the first telemetry bit is extracted) into this field. Currently, for NASCOM blocks, GIF will map the header field 'seq_no' from the NASCOM user header field. Use of this field or the block serial number from the NASCOM trailer is TBD. For TIS frame-synced data, this number comes from the raw telemetry record supplying bit 1. For extracted frames, it comes from the parent frame supplying the first valid data bit.
22	acq_bet	DSN frame sync subassembly allowable bit error tolerance in search and verify modes (values = 0 - 15).

23	maint_bet	DSN frame sync subassembly allowable bit error tolerance in lock and flywheel modes (values = 0 - 15).
24	verify_cnt	DSN frame sync subassembly number of frames required in verify mode (values = 0 - 15).
25	flywheel_cnt	DSN frame sync subassembly number of frames allowed in flywheel mode (values = 0 - 15).
26-27	data_length	Number of received telemetry bits, i.e., number of valid bits in the data portion of the data CHDO. For properly synchronized data, this field will be numerically equal to the frame length.
28	sync_mode flags	(DSN frame sync sub-assembly)
	A fssa_resync	Forced resync.
	B fssa_med	Minimum error detection enabled.
	C spare.	
	D fssa_fly	In flywheel mode.
	E fssa_lock	In lock mode.
	F fssa_verify	In verify mode.
	G fssa_search	In search mode.
	H fssa_bypass	Bypass fssa (unsynced data).
29	sync_status flags	(DSN frame sync subassembly)
	A fssa_phase	Input data polarity (0 = normal, 1 = inverted). DSN uses this field to report data polarity but does not complement inverted data.
	B fssa_overflow	Output buffer overflow occurred prior to this block.
	C fssa_temp	High temp on FSS card.
	D bit_sequence	Indicates bit sequence of sync word. Valid values are: 0 = MSB first 1 = LSB first. Note: SFOC only receives this value in reset form (i.e. 0).
29	bit_slip	Bit slip values are: 010 = +2 001 = +1 000 = 0 101 = -1 110 = -2
30	bit_errors	Number of bit errors detected in the frame-sync word. (Value = 0 for out-of-sync or bypass.)
31 - 32	spare.	

33	frequency_band	Frequency band of the master receiver (with current DSN capability, all receivers are on the same band). Valid values 'S', 'X', or 'C'.
34-37	bit_rate	Commanded bit rate, set by the Telemetry Processor Assembly (TPA) operator (as contrasted with measured bit rate), in bits/sec. (MSB first, LSB = 1 b/s; binary unsigned 32-bit integer.
38	destination	Destination code. In the DSN telemetry secondary header, this field is "canned" and always set to CF. When GIF receives GCF or NASCOM blocks, it sets this field to the destination code contained in the block header. This also applies to GCF blocks that contain SFDUs.
39	gdd	Always set to 0 by DSN. GIF treats gdd like the destination code, above.
40-43	snt	System noise temperature. The format is 32-bit floating point with a sign bit, a 9-bit exponent, and a 22-bit mantissa. Units are in Kelvins with an accuracy of 0.1K.
44-47	ssnr	Symbol signal-to-noise ratio. Format is 32-bit floating point.
48-51	signal_level	Receiver signal level (master receiver in array mode). Format is 32-bit floating point. Signal level is provided in decibels relative to 1 milliwatt (dbm).

Modcomp 32 bit floating point format



52	antennas	Bits indicate with a 1 or 0 which antennas are being used. If array, multiple bits will be set (1 = in use, 0 = not in use). A - 26-meter (DSS 16, 46, 66). B - 34-meter t/r (DSS 12, 42, 61). C - 34-meter hef (DSS 15, 45). D - Unused (value = 0). E - 70-meter (DSS14, 43, 63). F - Unused (value = 0). G - Unused (value = 0). H - Unused (value = 0).

53	receivers	Bits indicate with a 1 or 0 which antennas are being used. If array, multiple bits will be set (1 = in use, 0 = not in use). A - Receiver #1. B - Receiver #2. C - Receiver #3. D - Receiver #4. E - Receiver #5. F - Receiver #6. G - Receiver #7. H - Receiver #8.
54	master_antenna	Master antenna number, (value range 1-8).
55	master_receiver	Master receiver number, (value range 1-8).
56	dtm_group	DTM group number, (value range 1-4).
57	tlm_channel	DSN telemetry channel number, (value range 1-2).
58-59	lock_status	Contains a two-bit status field for each of the following: A - Receiver (master receiver in array mode). B - Real time combiner - used only in array mode. C - Subcarrier demodulator. D - Symbol synchronizer. The Demodulator Synchronizer Assembly (DSA) is in lock if both the subcarrier demodulator and the symbol synchronizer are both in lock. E - Convolutional decoder. F - Frame synchronizer. The Frame Synchronizer is In Lock during Lock and Flywheel modes, Out of Lock during Search and Verify modes and Not in Use during Bypass mode. G-H spare. Each status field indicates the status for the respective equipment: 00 = Not in use 01 = Not used (invalid code) 10 = In lock 11 = Out of lock.
58	gdd	Gross Data Description. See 820-13 ops-6-8.
59	spare.	

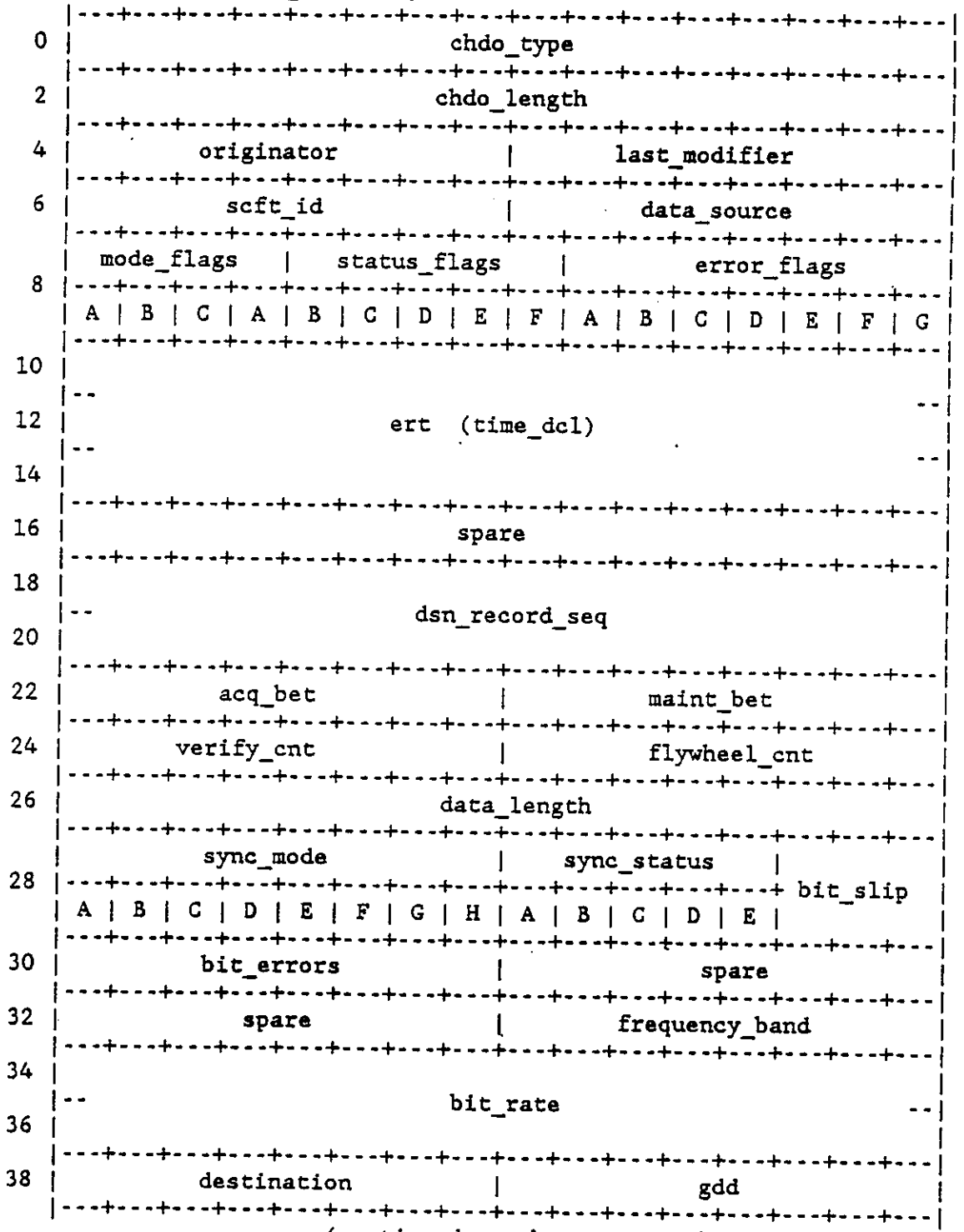
60-61	tlm_version	DSN telemetry software version number - two restricted ASCII characters. For GIF- and TIS-created SFDUs these two bytes are used to deposit the binary version and build numbers of the creating subsystem.

62-63 spare.

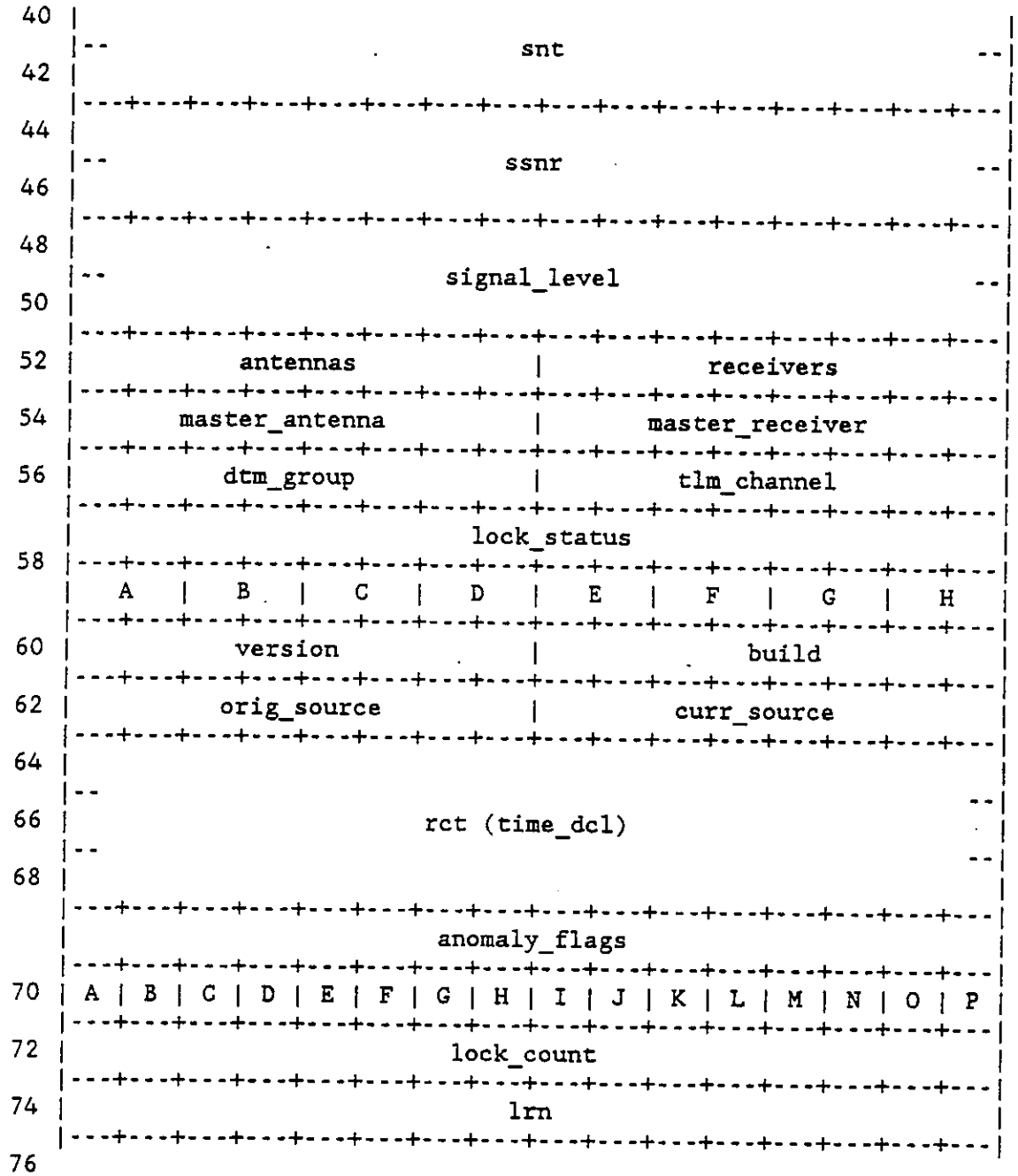
4.2.1.5.2 Secondary Header - GIF telemetry

This is the raw telemetry header that was originally received from the DSN, after it has been modified by the GIF subsystem. Note that the DSN header is first copied, then additional information is added by GIF. Some fields are modified.

(structure gif_secondary)



(continued on the next page)



The majority of information contained in the GIF DSN secondary header is copied from the DSN secondary header. The following describes only additional fields.

Byte Offset	Field Id	Description
0-1	chdo_type	GIF telemetry secondary header CHDO type code. (Value = 13).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 72).
5-59	see 4.2.1.5.1, Secondary Header - DSN telemetry	
60	version	Software version number (0-255) of the creating subsystem (GIF or TIS).
61	build	Software build number (0-255) of the creating subsystem (GIF or TIS).
62	orig_source	Indicates the original input path of the data that caused the creation of this record. This field is set by GIF when data is actually being received from one of these interfaces. It is copied by GIF from the input record during replay from a spooler file or SFDU tape. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F.
63	curr_source	Indicates the current input path of the data that caused the creation of this record. This field is set by the GIF (in the TIS telemetry secondary header by TIS) according to the current source of the input data. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F 6 = CDA spooler file 7 = SFDU tape. 8 = DTS virtual circuit 9 = CDA bytestream file 10 = UNIX bytestream file **Note:** If original_source and curr_source differ, then curr_source must indicate either CDA spooler file or SFDU tape, DTS virtual circuit, CDA bytestream file, or UNIX bytestream file.

64-69	rct	Record Creation Time. This field contains the system clock time, which is the SMC generated GMT time, at which the record was created by GIF or TIS. System clock time is maintained by SMC.
70-71	anomaly_flags	These flags are used to indicate an end in the sequence of normal sequential data of this record type. On normal good data, all flags are set to 0. An anomaly record is typically generated by copying the header of the last normal record and by setting the appropriate flags indicating the cause of the anomaly. Anomaly records contain a null data CHDO. When any of these flags are set, the data_val flag in field status_flags is also set.

A spare.

B upstream	Upstream anomaly - set only in conjunction with another anomaly flag. Indicates that the generating program received an anomaly record in its stream of input records and passed on the anomaly indication in its output stream.
C other	Any reason not identifiable by other anomaly flags.

D spare.

E format	Frame format ID has changed (applies to MRO, CMI, MSN parts of FID).
F forced	Operator forced loss-of-sync.
G phase	Change in data phase.
H break	Databreak. Sync code not found after flywheeling (sync lost).
I clock	New spacecraft clock or extraction reference.
J off	Generating processor turned off.
K timeout	Input timeout.
L sequence	Break or regression in input record sequence. GIF tests field **dsn_record_seq** for input received from the DSN. Sequence checks on SFOC-generated SFDUs are always performed on field **lrn**.
M overflow	Data was lost in real-time due to queue overflow.
N interface	An I/O error in an input interface caused loss of data.

O-P spare.

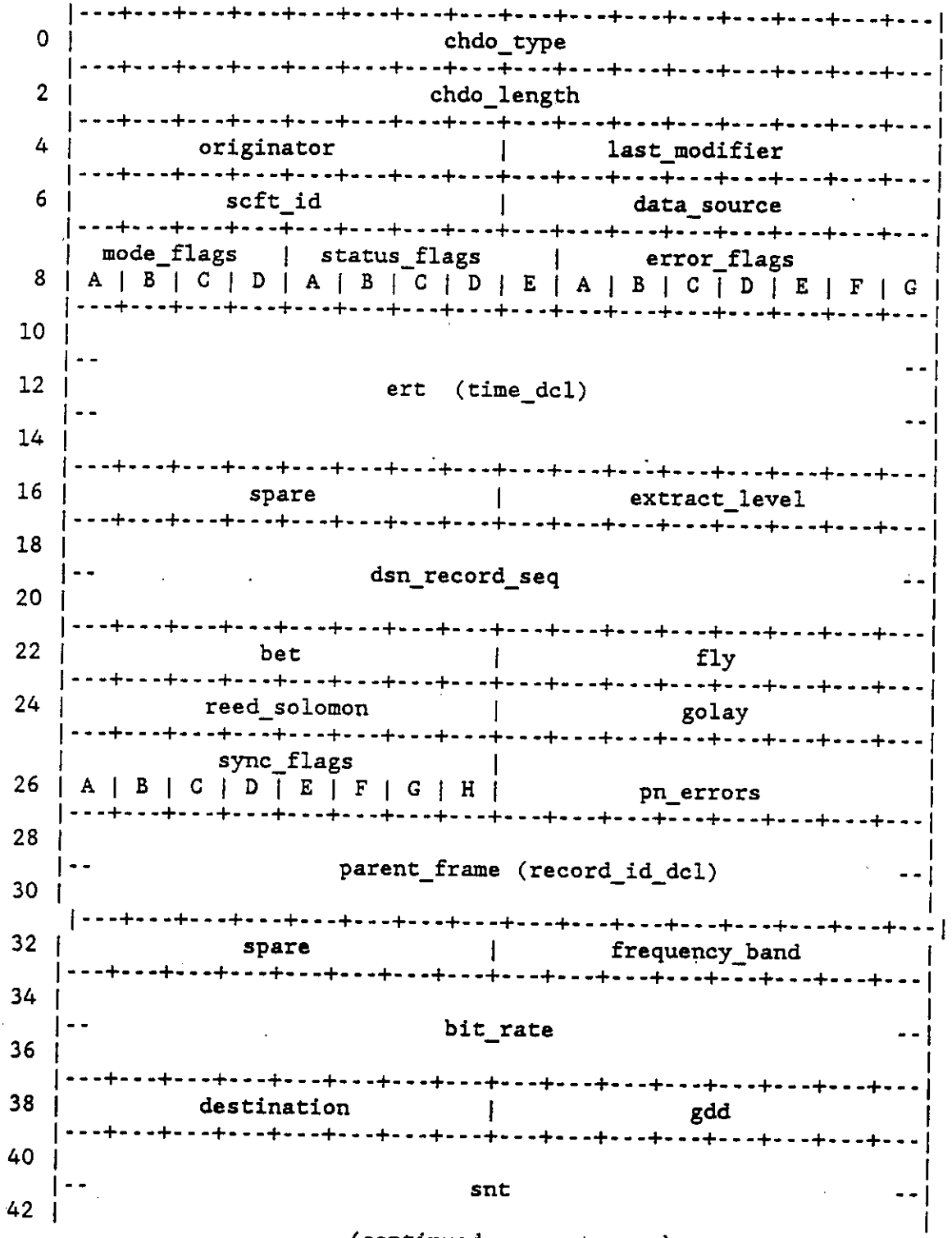
72-73	lock_count	This is an incrementing counter for all records of this type. The count starts at 1 and is incremented by 1 for each record. It resets to 0 upon overflow from 65,535. It also resets to 0 in the first normal data record that follows an anomaly record.

74-75	lrn	Logical record number. Set by the creating process. Contains the sequence number of all SFDUs of the same data type (same major / minor / format type). This counter starts with 1 for the first record of a given type after subsystem start-up, increments by 1 for each SFDU of the same type, and wraps to 0 on overflow from 65,535. After subsystem start-up, this counter is never reset by causes other than overflow.

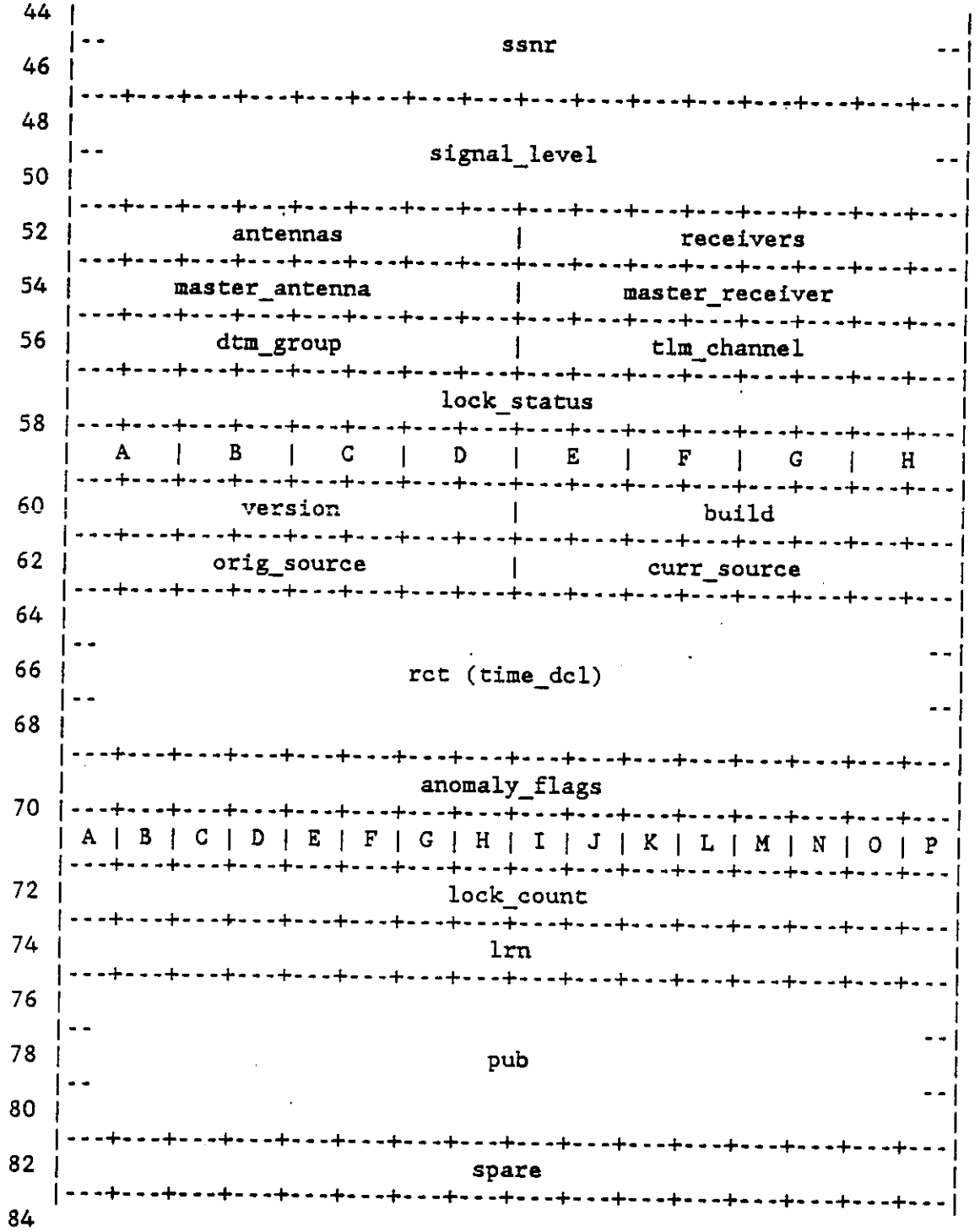
4.2.1.5.3 Secondary Header - TIS Telemetry

This header is added to frame-synced prime frames by the frame synchronizer and is carried through to all extracted and derived records (low/high rate science, engineering).

(structure tis_secondary)



(continued on next page)


44
ssnr
46
48
signal_level
50
antennas
receivers
52
master_antenna
master_receiver
54
dtm_group
tlm_channel
56
lock_status
58
A
B
C
D
E
F
G
H
version
build
60
orig_source
curr_source
62
64
rct (time_dcl)
66
68
anomaly_flags
70
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
lock_count
72
lrn
74
76
pub
78
80
spare
82
84

Byte Offset	Field Id	Description
0-1	chdo_type	TIS telemetry secondary header CHDO type code. (Value = 15).
2-3	chdo_length	Length of CHDO value field (remainder of header) in bytes. (Value = 80).

4-15 - see 4.2.1.5.2, Secondary Header - GIF Telemetry.

16	spare.	
17	extract_level	Extraction Level. For the frame-synced from raw telemetry, this field contains 0. For records extracted or derived from such a prime frame, this field contains the number of synchronous or asynchronous extractions that were performed to arrive at this record.

18-21 see 4.2.1.5.2, Secondary Header - GIF Telemetry

22	bet	Bit error tolerance applied to data synced by TIS.
23	fly	Flywheel factor applied to data synced by TIS.
24	reed_solomon	Reed-Solomon decode. Values are:

0 = No Reed-Solomon bit syndrome in this data type.

1 = Reed-Solomon bit syndrome exists in this data type but bit correction was not applied.

2 = Reed-Solomon bit syndrome has been removed (data compressed) but bit correction was not applied.

3 = The data has been Reed-Solomon decoded (Reed-Solomon syndrome bit correction has been applied) and the bit syndrome has been removed.

4 = The data has been Reed-Solomon decoded (Reed-Solomon syndrome bit correction has been applied), but the bit syndrome has not been removed.

25 golay — Golay decode. Values are:

- 0 = No Golay block-pairs exist in this data type.
- 1 = Golay block-pairs exist in this data type but the decode and correction algorithm was not applied.
- 2 = Golay symbols have been removed (data compressed) but the decode and correction algorithm was not applied.
- 3 = The data has been Golay decoded (decode and correction algorithm has been applied) and correction symbols have been removed.
- 4 = The data has been Golay decoded (decode and correction algorithm has been applied), but correction symbols have not been removed.

26 sync_flags - Frame synchronization status flags.

A phase — Indicates the phase of the input telemetry.
- 0 = True.
- 1 = Complemented. Bits in this output frame have been inverted.

B soft_sync — Indicates whether TIS synced data was produced without the aid of the frame-sync assist hardware.
- 0 = The frame-sync assist hardware was used
- 1 = Software frame-syncing only (no fsa).

C scid_force — Indicates whether the spacecraft ID was overridden by an operator-specified value.
- 0 = Not overridden
- 1 = The spacecraft identifier code has been overlaid with a new operator-entered spacecraft ID value.

D sclk_ref — Indicates whether two consecutive valid and properly incrementing occurrences of the SCLK have been encountered.
- 0 = SCLK reference established or not applicable
- 1 = SCLK reference not established.

E sclk_cor — Indicates whether SCLK correction has been disabled by operator directive.
- 0 = SCLK correction algorithm enabled
- 1 = SCLK correction disabled.

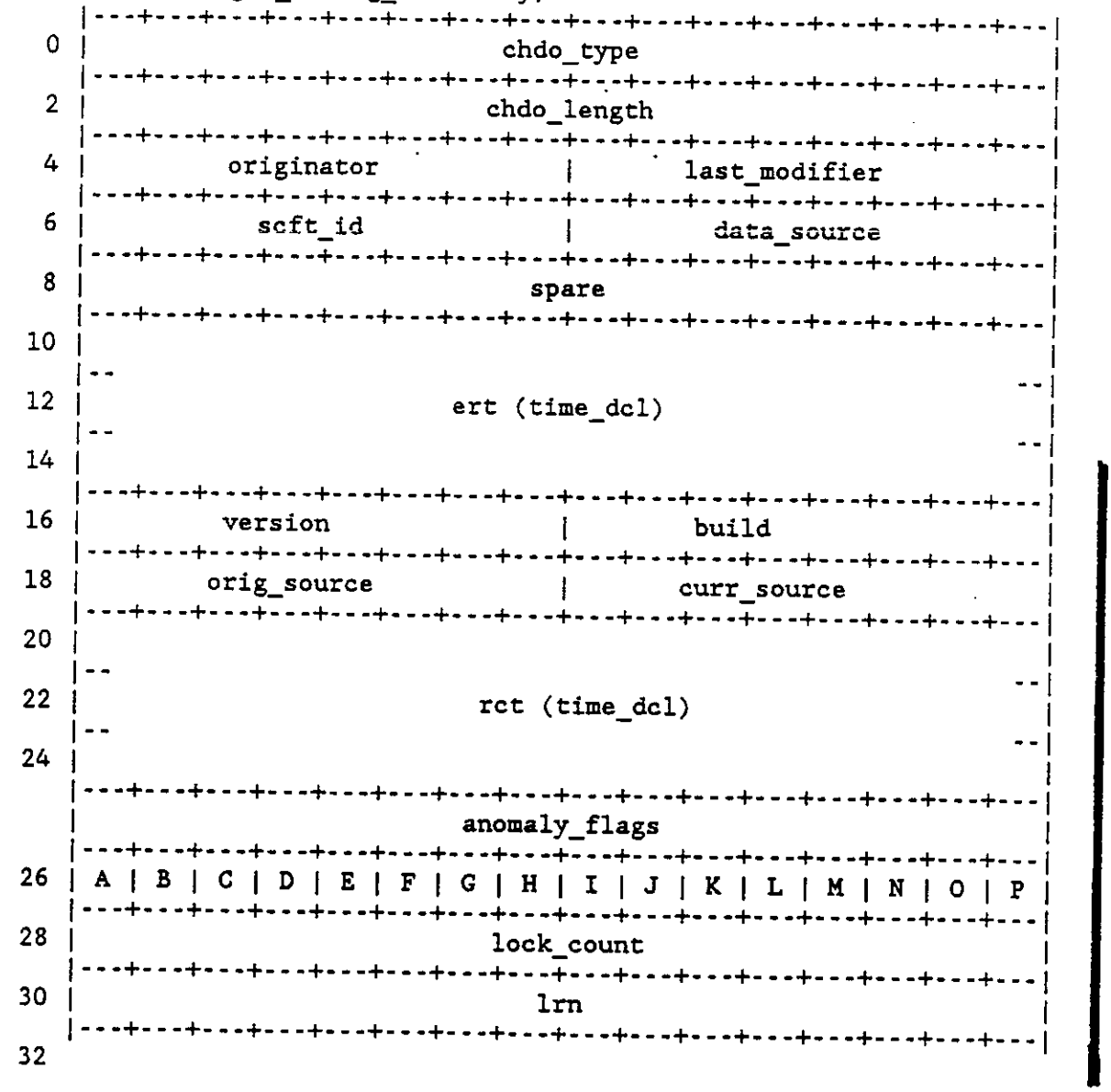
F-H — spare.

27	. pn_errors	Number of frame-sync code bit errors detected in parent frame by TIS. This field is copied to any derived frames.
28-31	parent_frame	Parent frame record ID. Identifies parent frame from which this record is derived/extracted. The creating process copies this field from the input frame primary header (record_id).

32-75 - see 4.2.1.5.2, Secondary Header-GIF Telemetry

76-81	pub	An ASCII string consisting of six project-unique bytes, entered into the TIS subsystem by user control directive. TIS places into these bytes whatever the user has entered.
82-83	spare.	

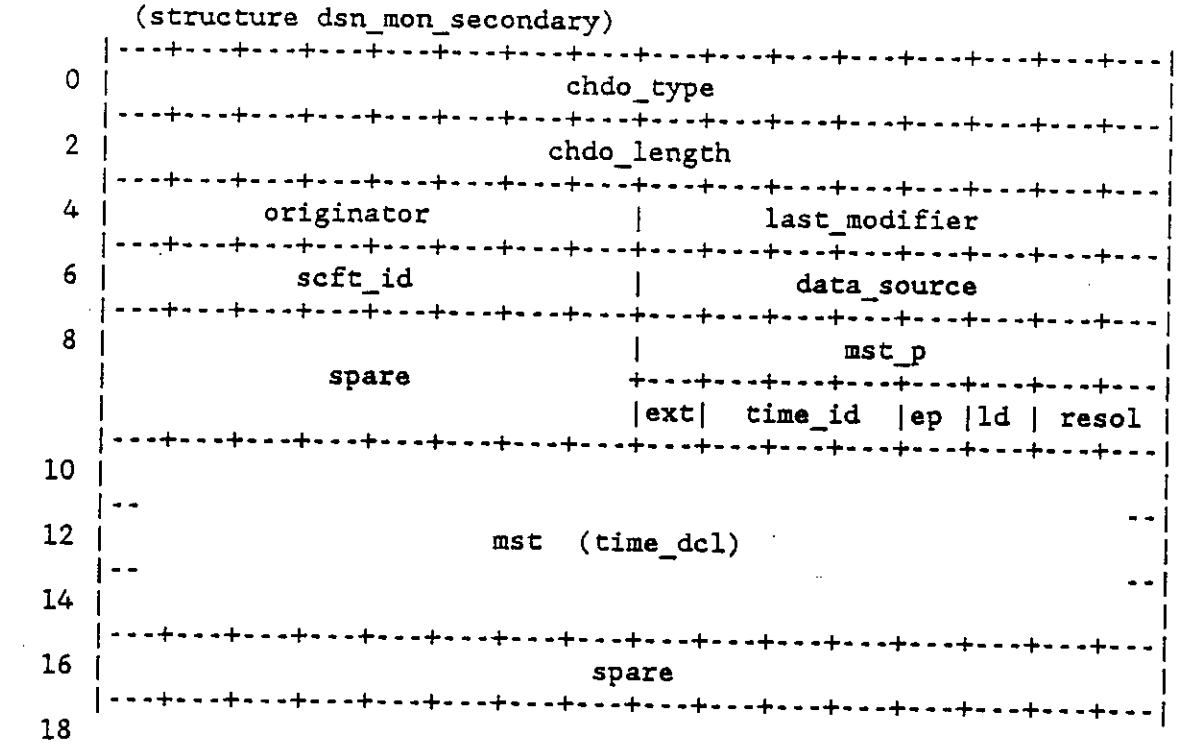
4.2.1.5.4 Secondary Header - GIF Raw Log
(structure gif_rawlog_secondary)



Byte Offset	Field Id	Description
0-1	chdo_type	GIF raw log secondary header CHDO type code. (Value = 18).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 24).
4	originator	Originator ID (Value = 108).
5	last_modifier	Last Modifier ID (Value = 108).
6	scft_id	Spacecraft ID. GIF copies this value from the GCF/NASCOM block header.
7	data_source	Source Station ID. GIF copies this value from the GCF/NASCOM block header.
8-9	spare.	
10-15	ert	Earth-Received Time. This time is taken from the GCF/NASCOM block header. For GCF blocks it is not yet adjusted to reflect time of first bit.

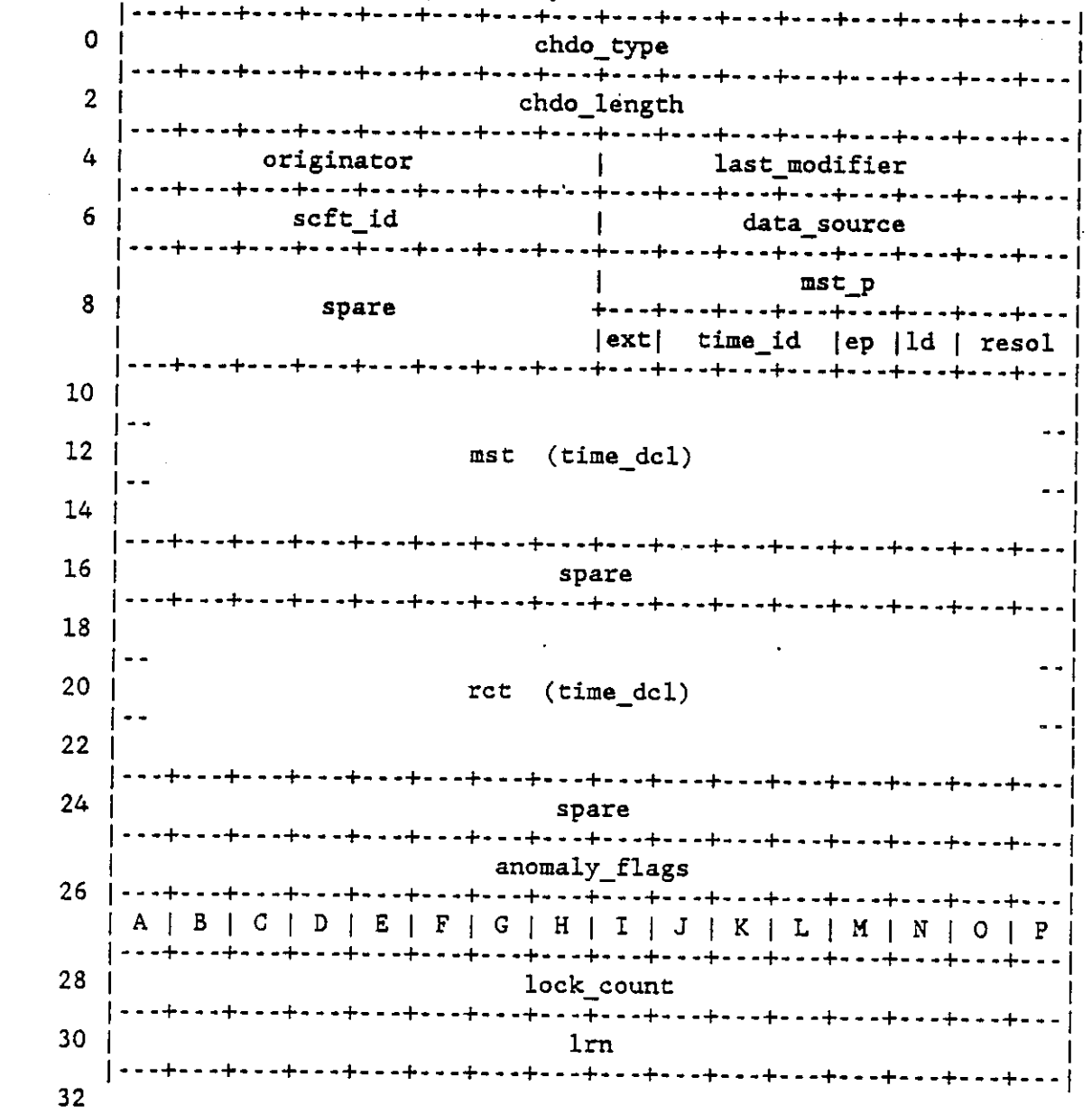
16-31 see 4.2.1.5.2, Secondary Header - GIF Telemetry, bytes 60-75.

4.2.1.5.5 Secondary Header - DSN Monitor



Byte Offset	Field Id	Description
0-1	chdo_type	DSN monitor secondary header CHDO type code. (Value = 12).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 14).
4	originator	Originator ID, assigned to DSN. (value = 48).
5	last_modifier	Last modifier ID, no modifier. (value=0).
6	scft_id	Spacecraft identifier (ref. 820-13 OPS 6-8 for code value).
7	data_source	DSN station source ID (ref. 820-13 OPS 6-8 for code value).
8	spare.	
9	mst_p - Time code identifier.	
	ext	Extension flag, always 0.
	time_id	Time code ID, always 4.
	ep	Epoch ID, always 1 for: 01/01/58.
	ld	Length of day segment, always 0 for: 16-bit segment.
	resol	Resolution, always 0 for: milliseconds.
10-15	mst	Monitor sample time. See 4.2.1.1 for format.
16-17	spare.	

4.2.1.5.6 Secondary Header - GIF Monitor
(structure gcf_mon_secondary)

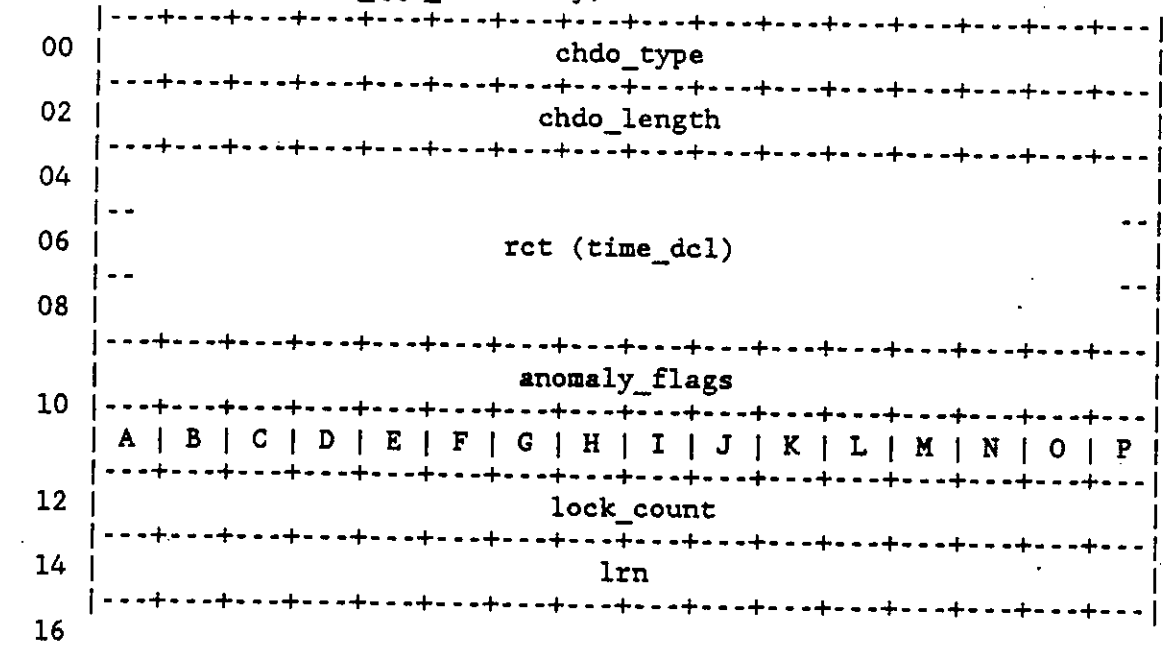


Byte Offset	Field Id	Description
0-1	chdo_type	GIF monitor secondary header CHDO type code. (Value = 121).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 28).
4	originator	Originator ID, assigned to DSN. (Value = 48)
5	last_modifier	Last modifier ID, assigned to GIF. (Value=108).

6	scft_id	Spacecraft identifier (ref. 820-13 OPS 6-8 for code value).
7	data_source	DSN station source ID (ref. 820-13 OPS 6-8 for code value).
8	spare.	
9	mst_p - Time code identifier.	
	ext	Extension flag, always 0.
	time_id	Time code ID, always 4.
	ep	Epoch ID, always 1 for: 01/01/58.
	ld	Length of day segment, always 0 for: 16-bit segment.
	resol	Resolution, always 0 for: milliseconds.
10-15	mst	Monitor sample time.
16-17	spare.	
18-23	rct	Record creation time; local time at which the record was created by GIF.
24-25	spare.	
26-27	anomaly_flags	See GIF/TIS secondary telemetry headers.
28-29	lock_count	See GIF/TIS secondary telemetry headers.
29-30	lrn	Logical record number - see GIF/TIS secondary telemetry headers.

4.2.1.5.7 Secondary Header - TIS QQC Data

(structure tis_qqc_secondary)



Byte Offset	Field Id	Description
00-01	chdo_type	TIS QQC secondary header CHDO type code. (Value = 122).
02-03	chdo_length	Length of CHDO value field (remainder of header) (Value = 12).
04-09	rct	Record creation time; local time at which TIS completes the record.
10-11	anomaly_flags	See GIF/TIS secondary telemetry headers. (The only anomaly that will be seen for this record type indicates Processor turned off.)
12-13	lock_count	See GIF/TIS secondary telemetry headers.
14-15	lrn	Logical record number - see TIS secondary telemetry header.

4.2.1.6 Tertiary Headers

Tertiary headers are mission-specific and are added by the TIS subsystem to the telemetry records it generates. Because they are mission-specific, they are defined and described in mission SIS modules.

4.2.1.7 Quaternary Headers

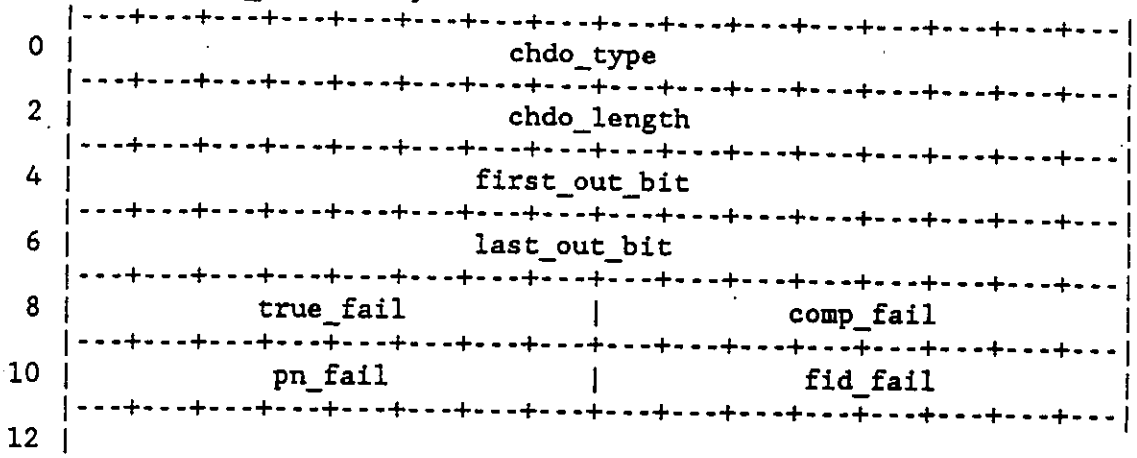
Quaternary headers that are mission-independent are defined and described in this SIS module. Mission-specific Quaternary headers are defined and described in mission SIS modules.

4.2.1.7.1 Quaternary Header - Out of Sync

The out-of-sync quaternary header provides the meta-data required for out-of-sync SFDUs created from input raw telemetry SFDUs that contain out-of-sync bits. Information includes the offset of first and last out-of-sync bits, as well as the number of failed sync attempts resulting from pns and fids in that block.

During acquisition, an input SFDU is kept until sync is found (or all sync attempts have failed). Each input SFDU that contains any bits that have not gone into a frame-synced frame is made into an out-of-sync record.

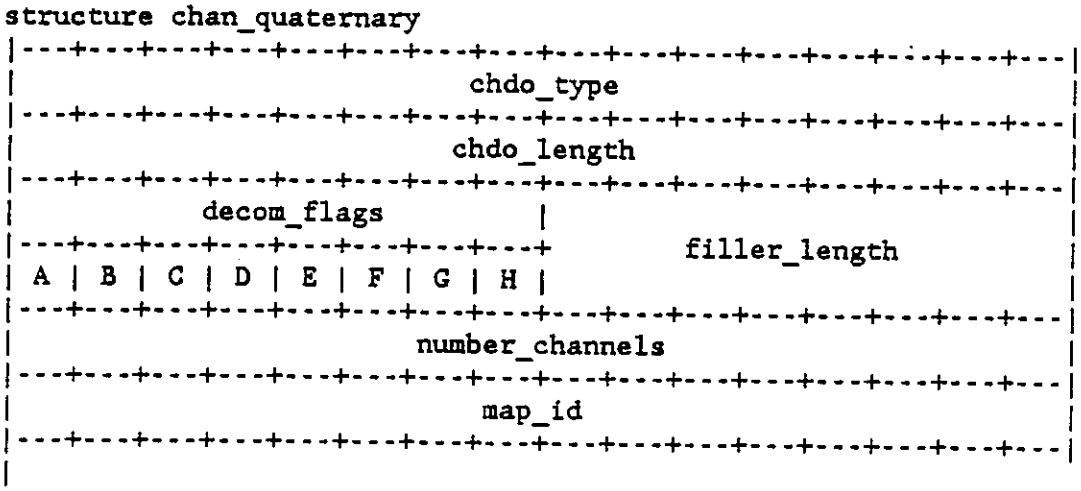
structure out_quaternary



Byte Offset	Field Id	Description
0-1	chdo_type	JPL telemetry quaternary type code. (Value = 20).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 8).
4-5	first_out_bit	Bit offset from first bit of data in the input SFDU to the first out-of-sync bit.
6-7	last_out_bit	Bit offset from first bit of data in the input SFDU to the last out-of-sync bit.
8	true_fail	Number of true pns passed over without finding sync in this SFDU.
9	comp_fail	Number of complemented pns passed over without finding sync in this SFDU.
10	pn_fail	Number of failed sync attempts due to bad trailing pn.
11	fid_fail	Number of failed sync attempts due to fid mismatch.

4.2.1.7.2 Quaternary Header - Channelized Data

This header accompanies all channelized data records.

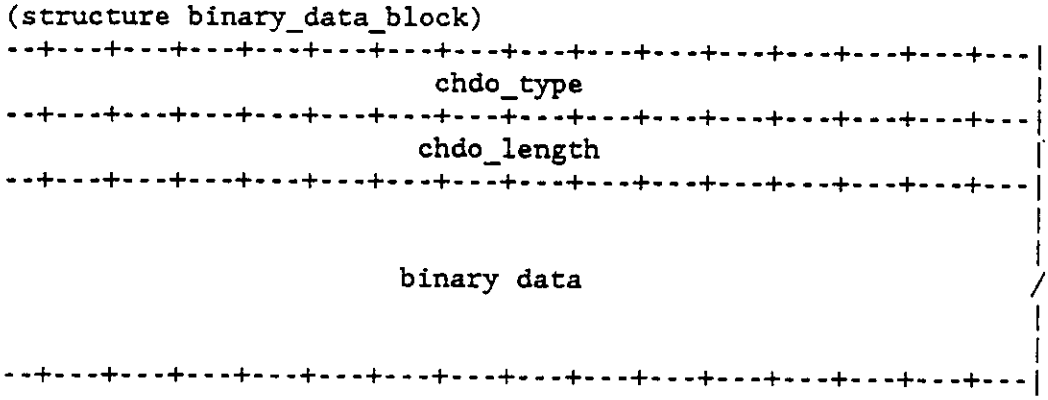


Byte Offset	Field_Id	Description
0-1	chdo_type	Channelized data quaternary header type code. (Value = 27).
2-3	chdo_length	Length of CHDO value field (remainder of header) (Value = 6).
4	decom_flags	
	A map_valid	Indicates whether the decommutation map used was a valid one. Rules for selecting a valid map are mission-specific. 0 = Valid map used 1 = Invalid map used.
	B-H spare.	
5	filler_length	This is the number of filler bits at the end of this SFDU record added in order to pad the data out to an even word length. Filler bits will follow the actual data area of the SFDU. Value may be 0 - 15.
6-7	number_channels	Number of channels in this SFDU.
8-9	map_id	Contains the map sequence number of the map used to perform decommutation for this SFDU. Values are 1 to 2^{16-1}.

4.2.1.8 Data Block Headers

4.2.1.8.1 Data Block Header - Standard Non-structured Binary

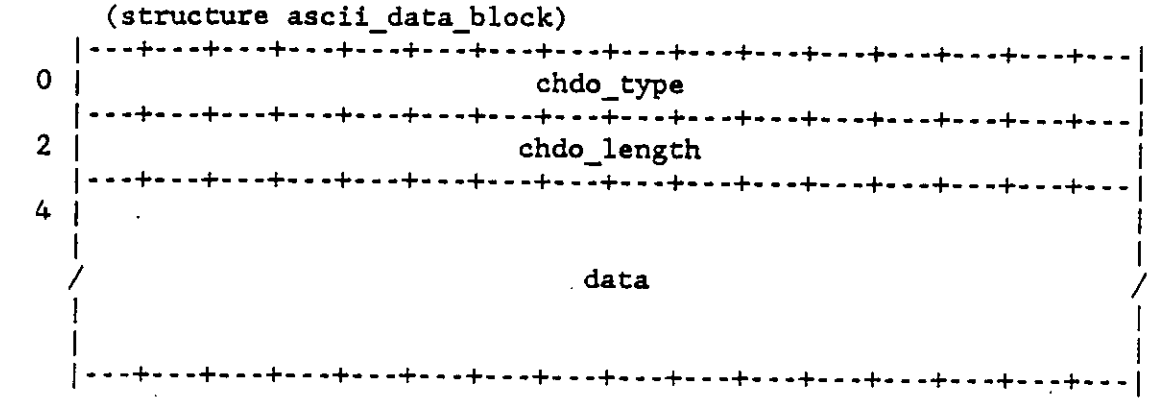
This data block type is used for all binary data that does not require any further breakdown into individual fields for processing purposes within SFOC. The data field may be of any length (within CHDO length constraints).



Byte Offset	Field Id	Description
0-1	chdo_type	Binary data block type code. (Value = 10).
2-3	chdo_length	Length of CHDO value field (data) in bytes, starting at the following byte. This value will always be an even number of bytes.
4-n	data	Data of variable length, depending on record_id (major, minor, mission ID, and format).

4.2.1.8.2 Data Block Header - Standard Non-structured ASCII

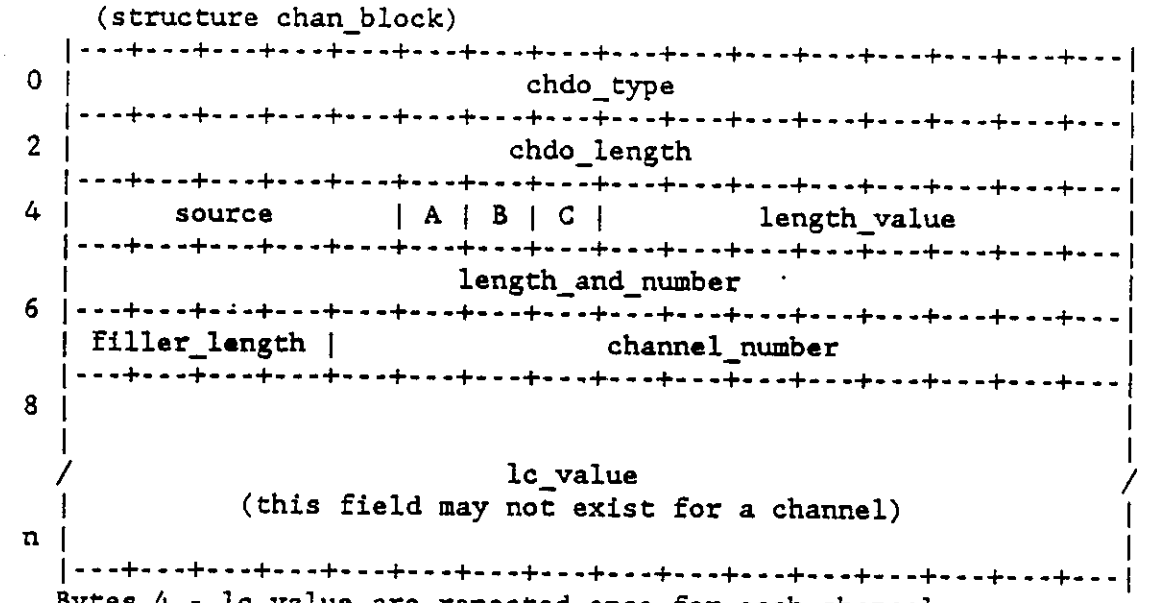
This data block type is used for strings of ASCII characters that require no further definition of structure. The data field may be of any length (within CHDO length constraints).



Byte Offset	Field Id	Description
0-1	chdo_type	ASCII data block type code. (Value = 123).
2-3	chdo_length	Length of CHDO value field (data) in bytes, starting at the following byte. This value will always be an even number of bytes.
4-n	data	Data of variable length, depending on record_id (major, minor, mission ID, and format).

4.2.1.9 Data block - Channelized Data

This data block is used for all types of channelized records.



Bytes 4 - lc_value are repeated once for each channel.

Byte Offset	Field_Id	Description
0-1	chdo_type	Channelized data block type code.(Value = 28).
2-3	chdo_length	Length of CHDO value field (data) in bytes, starting at the following byte. This value will always be an even number of bytes.
4	source	Specifies the alphabetic character used in the channel identifier. (Values = 1 - 26 (1=A, 2=B, etc.)
	A lv_flag	This flag identifies whether bits 8 through 15 (length_value) contain the actual value of the channel (if it will fit in a single byte), or the length of the actual value in words. 0 = length_value contains a length 1 = length_value contains a value and field lc_value does not exist for this channel.
	B bad_data	Indicates whether valid data or filler was decommutated. 0 = Valid data was used in decommutation process 1 = Filler data was decommutated.
	C error	Indicates whether an error occurred in the decommutation process for this channel. 0 = No error occurred 1 = Error occurred.
5	length_value	This field contains either the actual value of the channel or the length of the actual value (number of 16-bit words) stored in lc_value. Flag lv_flag determines which applies.
6-7	length_and_number	
	bits 0-3 filler_length	The number of bits preceding the first word of the actual channel value (in lc_value), since lc_value is right-justified. (Values are 0 - 15).
	4-15 channel_number	The numerical portion of the channel identifier (Values are 0 - 4095).
	8-n lc_value	Long channel value. If the value for a given channel cannot fit into a single byte, its value will then be stored here. The value should be stored in one to two hundred fifty-five 16-bit words. If this field contains a channel value, flag lv_flag must

be 0 and length_value must contain the number of 16-bit words used to store the channel value. If flag lv_flag = 1, then this field does not exist for this channel.

4.2.2 Data Representation Conventions

Most/least significant bit/byte conventions are as described in SIS module SFOC-5-SYS-*DU-NJPL (Applicable Document 4a).