

# ***MSG Ground Segment LRIT HRIT Mission Specific Implementation***

Doc.No. : EUM/MSG/SPE/057  
Issue : v7 e-signed  
Date : 30 November 2015  
WBS :

EUMETSAT  
Eumetsat-Allee 1, D-64295 Darmstadt, Germany  
Tel: +49 6151 807-7  
Fax: +49 6151 807 555  
<http://www.eumetsat.int>

### **Document Change Record**

<b>Issue / Revision</b>	<b>Date</b>	<b>DCN. No</b>	<b>Summary of Changes</b>
1	12/12/1996		New.
2	02/04/1998		Update all sections.
2.1	04/05/1998		Update all sections.
3	16/11/1998		Update all sections.

<b>Issue / Revision</b>	<b>Date</b>	<b>DCN. No</b>	<b>Summary of Changes</b>
4	21/09/1999		Update sections 1.4.2, 4.3.2.9, 5.3.1, 5.3.2.1, D.2.1, D.2.2 and D.2.3.
4.1	09/03/2001		Update all sections.
4.2	25/03/2004		Update all sections.
5	04/02/2005		Author changed from J. Güttlich, N. Sinander and E. Schaffner to K. Dammann. Appendix E added. Document reformatted.
6	21/06/2006		Author changed to N. Coyne.  Distribution list modified. Section 1.1. Applicability of document clarified with regard to EUMETCast and direct dissemination via MSG-2. Section 1.3.2. List of applicable documents extended. Sections 4.2.2.1. and 4.2.2.2. Size of image segments for EUMETCast and direct dissemination described in detail. Section 4.2.2.3. List of products of Imagery Type updated. Section 4.2.2.4. List of products of OverlayType updated. Section 4.2.6.3. Scope of document regarding SAF products clarified. Section 4.2.8. Size and timeliness of File Type #130 defined as actual (subject to optimisation). Section 4.2.9.1. List of meteorological products of File Type #144 updated. Section 4.3.2.2. Description of Header Type 1 detailed for EUMETCast and direct dissemination. Section 4.3.2.5. Definition of Header Type #4 modified. Section 4.3.2.6 .

<b>Issue / Revision</b>	<b>Date</b>	<b>DCN. No</b>	<b>Summary of Changes</b>
			<p>Interpretation of contents of Header Type #5 modified.</p> <p>Annex 1.2.2.1. GOES-10 to GOES-12 added to Image Product Specific Header description.</p> <p>Annex 5.1. MPEF products overview updated.</p> <p>Annex 5.3. Examples for Data Definition Block of MPEF Products removed.</p> <p>Annex 6.1. For SAF Products overview, reference made to EUMETSAT Web Site.</p>
7	30/11/2015		<p><u>Section 4.3.2.5:</u> Disseminating spacecraft definition changed</p> <p>Annex A: Updated for GOES, MPEF and SAF</p> <p><u>Annex E:</u> Corrected typo. Reworked Annex. Corrected references to equations.</p>

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>8</b>
1.1	Purpose and Scope of the Document .....	8
1.2	Document Structure.....	9
1.3	Applicable and Reference Documentation .....	9
1.3.1	Applicable Documentation .....	9
1.3.2	Reference Documentation .....	9
1.4	Conventions.....	11
<b>2</b>	<b>Introduction to the MSG Specific OSI Reference Model.....</b>	<b>12</b>
2.1	Communication Concept .....	12
2.2	MSG Dissemination Time Management.....	14
2.3	List of acronyms.....	14
<b>3</b>	<b>Application Layer.....</b>	<b>17</b>
3.1	Input Data Streams.....	17
3.1.1	General .....	17
3.1.2	MSG SEVIRI Level 1.5 Data.....	18
3.1.3	Meteorological Data & Products from GTS .....	18
3.1.4	Meteorological Products from MPEF .....	18
3.1.5	DCP Data.....	18
3.1.6	External Data from SGS .....	19
3.1.7	Service Messages .....	19
3.2	Rearrangement of Input Data .....	19
<b>4</b>	<b>Presentation Layer.....</b>	<b>20</b>
4.1	Structure of LRIT/HRIT Files .....	20
4.2	LRIT/HRIT File Types and Data Field Descriptions .....	20
4.2.1	LRIT/HRIT File Types Overview.....	20
4.2.2	File Type #0 - Image Data .....	21
4.2.3	File Type #1 - GTS Message.....	25
4.2.4	File Type #2 - Alphanumeric Text File.....	26
4.2.5	File Type #3 - Encryption Key Message.....	26
4.2.6	File Type #128 - Repeat Cycle Prologue.....	27
4.2.7	File Type #129 - Repeat Cycle Epilogue .....	28
4.2.8	File Type #130 - DCP Message .....	29
4.2.9	File Type #144 - Binary File.....	29
4.3	LRIT/HRIT File Header Types .....	30
4.3.1	General .....	30
4.3.2	Definition of Header Types .....	31
4.3.3	File Type vs. Header Implementation.....	47
<b>5</b>	<b>Session Layer.....</b>	<b>48</b>
5.1	General .....	48
5.2	Input to Session Layer .....	48
5.3	Compression.....	48
5.3.1	General .....	48
5.3.2	MSG Mission Specific Implementation of Compression.....	51
5.4	Encryption.....	94
5.4.1	En-/Decryption Principle .....	94
5.4.2	PN Pattern Generation .....	99
5.4.3	Station Key Unit Functionality.....	100
5.5	Session Layer Output .....	102
<b>6</b>	<b>Transport Layer.....</b>	<b>103</b>
6.1	General .....	103
6.2	Source Packetisation .....	103
6.2.1	Source Packet Structure.....	103
6.2.2	Test Packet Generation .....	105
6.3	Transport Layer Output .....	105

<b>7</b>	<b>Network Layer</b> .....	<b>106</b>
7.1	Input to Network Layer .....	106
7.2	General .....	106
7.3	Network Layer Processing .....	106
7.4	Output of Network Layer .....	106
<b>8</b>	<b>Data Link Layer</b> .....	<b>107</b>
8.1	Input to Data Link Layer .....	107
8.2	General .....	107
8.3	VCLC Sublayer Processing .....	107
8.4	VCA Sublayer Processing .....	108
8.4.1	VCDU Assembly .....	108
8.4.2	'Fill VCDU' Generation .....	109
8.4.3	Reed-Solomon Coding .....	109
8.4.4	Randomisation .....	110
8.4.5	Sync Marker Attachment .....	110
8.4.6	Serialisation and Output of the Data Link Layer .....	110
<b>9</b>	<b>Physical Layer</b> .....	<b>111</b>
<b>Appendix A</b>	<b>Application Data Unit Definition of Miscellaneous LRIT/HRIT Data &amp; Products</b> .....	<b>112</b>
A.1	Foreign Satellite Data via LRIT/HRIT .....	112
A.1.1	FSD Overview .....	112
A.1.2	Definition of FSD Repeat Cycle Prologue Data Fields .....	112
A.1.3	Example of an FSD Image Data Function (Header #3) .....	116
A.1.4	FSD Spectral Channel Identifiers .....	116
A.2	GTS Data & Products via LRIT/HRIT .....	116
A.3	Compression Test Message Pattern .....	117
A.4	Encryption Test Message Pattern .....	117
A.5	MPEF Products via LRIT/HRIT .....	117
A.5.1	Data_Definition_Block for MPEF Products .....	117
A.5.2	MPEF Product Overview .....	117
A.5.3	Repeat Cycle Prologue Definition for MPEF Products .....	118
A.5.4	Data_Definition_Block for MPEF Products .....	126
A.6	DCP Messages via LRIT/HRIT .....	127
<b>Appendix B</b>	<b>LRIT/HRIT Data Formatting Structured According OSI Reference Model</b> .....	<b>131</b>
<b>Appendix C</b>	<b>LRIT/HRIT Encryption Scheme</b> .....	<b>132</b>
C.1	Overview .....	132
C.2	Public Key Entry Representation in an Encryption Key Message File .....	132
C.3	Definition of the Seed Expansion .....	133
C.4	Example for Encryption Validation .....	133
C.5	Key Group Changes: The Bank Concept .....	134
C.5.1	Example of a Key Group Change as seen by a User Station/SKU .....	134
C.6	Example of Data Communication Between User Station and SKU .....	135
C.6.1	Protocol Naming Conventions .....	136
C.6.2	Protocol Definition .....	136
C.6.3	Protocol Scheme .....	137
C.6.4	Example: Command "Write PBK" .....	138
<b>Appendix D</b>	<b>LRIT/HRIT Physical Layer Details</b> .....	<b>140</b>
D.1	Nominal Coverage Area .....	140
D.1.1	HRIT G/T ISO-Contours .....	140
D.2	LRIT/HRIT Space-to-Ground Interface .....	142
D.2.1	Modulation Properties .....	142
D.2.2	HRIT Physical Layer Characteristics .....	142
D.2.3	LRIT Physical Layer Characteristics .....	144
<b>Appendix E</b>	<b>Derivation of the Navigation Coefficients</b> .....	<b>147</b>
E.1	Introduction .....	147
E.2	Derivation of CFAC and LFAC .....	149
E.3	Derivation of COFF and LOFF .....	150
E.3.1	LOFF of Segments .....	150
E.3.2	COFF of Segments .....	151

E.4	Lines Reduced Format – Rapid Scans.....	151
E.5	Summary .....	152
<b>Appendix F</b>	<b>List of TBDs and TBCs .....</b>	<b>153</b>

## 1 INTRODUCTION

### 1.1 Purpose and Scope of the Document

A Global Specification for Low Rate and High Rate Information Transmission (LRIT/HRIT) [AD.1] has been agreed by the Co-ordination Group of Meteorological Satellites (CGMS). The global specification bases on the CCSDS recommendations of Advanced Orbiting Systems (AOS) [RD.1] and the ISO standard 7498 (OSI Reference Model) [RD.2]. It defines the structure and the formatting of the LRIT/HRIT files and the processing and the transport protocols of all OSI layers applicable to all geostationary meteorological spacecraft.

The purpose of this document, the LRIT/HRIT Mission Specific Implementation, is the specification of the more detailed communication structure applied to the dissemination service of the Meteosat Second Generation (MSG).

It defines the formatting from the view of the transmitting site and includes the functionality of:

- The Data Acquisition and Dissemination Facility (DADF) within the MSG Mission Control Centre (MCC), and,
- The relevant parts of the baseband equipment of the MSG Primary Ground Station (PGS).

It further implies functionality from the receiving side (User Stations) point of view. This is in principle a reverse mechanism of the formatting defined in this document.

This document forms part of the MSG User Interface Documentation available to the potential MSG user station manufacturers and end-users.

#### **Important Note:**

This document was originally written purely for the HRIT/LRIT dissemination via MSG's on-board transponders. A failure of a Solid State Power Amplifier onboard MSG-1 a few weeks after launch meant that an alternative dissemination method was needed. EUMETCast was thus developed and is described in TD 15 [RD.25]. The communication structure implemented on the transmitting side by the DADF and PGS systems, has not been used for MSG-1, but is in use with the subsequent MSG satellites. This is referred to as "DIRECT Dissemination" throughout this document.

**Such discrepancies are not addressed further in the related sections: it is expected that the reader is in a position to establish a comprehensive and consistent specification of communication structure and formats by the exploitation of the applicable sections of this document in conjunction with the EUMETCast document TD 15 [RD.25]**

## 1.2 Document Structure

A brief description of the contents of each of the sections is given below:

Section 2	Provides an overview of the OSI layer reference model and its particular functionality w.r.t. the MSG dissemination service. Includes a glossary of terms and abbreviations used in this document.
Section 3	Presents the data flows and the external interfaces of the dissemination system on application level.
Section 4	Introduces to the LRIT/HRIT file structure in general and defines the mission specific file types and secondary headers.
Section 5	Contains the required details about the compression and encryption algorithms.
Sections 6 to 8	Summarise the mechanisms of formatting the data into source packets and transfer frames.
Section 9	Defines the MSG mission-specific parameters of the physical layer.
Appendix A	Llists all foreign satellite data, meteorological data and products disseminated via the LRIT/HRIT dissemination channels.
Appendix B	Displays the complete LRIT/HRIT formatting according to the OSI layer model.
Appendix C	Shows details of the LRIT/HRIT encryption scheme.
Appendix D	Presents the LRIT/HRIT Space-to-Ground Interface parameters.
Appendix E	Describes how to derive the LRIT/HRIT navigation parameters.
Appendix F	Contains lists of TBCs, TBDs.

## 1.3 Applicable and Reference Documentation

### 1.3.1 Applicable Documentation

[AD.1] EUMETSAT ‘LRIT/HRIT Global Specification’, CGMS 03.

### 1.3.2 Reference Documentation

- [RD.1] MSG Ground Segment Design Specification (GSDS) - Volume F: Data Types and Encoding Rules
- [RD.2] (not assigned)
- [RD.3] CCSDS ‘Advanced Orbiting Systems, Networks and Data Links: Architectural Specification’, CCSDS Recommendation 701.0-B2, Blue Book ([http://ccsds.org/blue\\_books.html](http://ccsds.org/blue_books.html))
- [RD.4] ITU-T T.4 Recommendation, Terminal Equipments and Protocols for Telematic Services, Standardization of Group 3 Facsimile Apparatus for Document Transmission, International Telecommunication Union, Geneva
- [RD.5] Meteosat Second Generation Level 1.5 Image Data Format Description, EUM/MSG/ICD/105.
- [RD.6] Overview of the Meteosat Second Generation Ground Segment, MSG/TEN/093

- [RD.7] GMS User's Guide, 2<sup>nd</sup> Edition, Appendix C (Transmission Characteristics of GMS S-VISSR Data), Meteorological Satellite Centre, Tokyo, Japan, March 1989
- [RD.8] Revision of GMS Stretched-VISSR Data Format, Japan Meteorological Agency, October 1993
- [RD.9] MTSAT HiRID, Technical Information, Japan Meteorological Agency, Issue 2, 1/7/1998 ([http://www.wmo.ch/hinsman/APT\\_WEFAXstatus.html](http://www.wmo.ch/hinsman/APT_WEFAXstatus.html))
- [RD.10] WMO Publications:
- WMO Manual on Codes, Volume 1, International Codes, Parts A (Alphanumeric Codes), B (Binary Codes) & C (Common Features to Binary Codes and Alphanumeric Codes), Publication number 306
  - WMO Manual on the Global Telecommunications System, Publication number 386
- [RD.11] Information technology - Digital compression and coding of continuous-tone still images, Part 1: Requirements and guidelines, ISO/IEC 10918-1
- [RD.12] Data Encryption Standard (DES), Federal Information Processing Standard (FIPS) PUB 46-2), U.S. Dept. of Commerce, National Institute of Standards and Technology  
(<http://www.itl.nist.gov/div897/pubs/fip46-2.htm>)
- [RD.13] DES Modes of Operation, FIPS PUB 81, U.S. Dept. of Commerce, National Bureau of Standards
- [RD.14] Telemetry Coding Standard, Blue Book, CCSDS 101.0-B-3  
([http://ccsds.org/blue\\_books.html](http://ccsds.org/blue_books.html))
- [RD.15] Radio Frequency and Modulation Standard, ESA PSS-04-105, Issue 1, December 1989
- [RD.16] DADF Interface Control Document Station Key Unit, MSG/ICD/102
- [RD.17] GOES IJK/LM Operation Ground Equipment (OGE), DRL 504-02, Part 1 of 2, Interface Specification, Sect. 3.0 GVAR Transmission Format, NOAA/NESDIS, December 1997
- [RD.18] Russian Geostationary Operational Meteorological Satellite GOMS/ELECTRO, High Resolution Direct (HRD) raw digital data transmission, SRC "Planeta", June 1997
- [RD.19] NOAA KLM User's Guide, Draft August 1997, NOAA/NESDIS,  
(<http://perigee.ncdc.noaa.gov/docs/intro.htm>)
- [RD.21] Pointers to Cryptographic Software (<http://www.cs.hut.fi/crypto/software.html> or <ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/des>)
- [RD.22] An Image Multiresolution Representation for Lossless and Lossy Compression, Amir Said, William A. Pearlman. SPIE Symposium on Visual Communications and Image Processing, Cambridge, MA, Nov. 1993
- [RD.23] Arithmetic Coding for Data Compression; Witten, Neal & Cleary; Commun. ACM, vol. 30, pp 520-540, June 1987
- [RD.24] The Meteosat Archive; Format Guide No. 1; Basic Imagery: OpenMTP Format; EUM FG 1; Rev 2.1; April 2000
- [RD.25] TD 15 - EUMETCast - Broadcast System for Environmental Data

## **1.4 Conventions**

All data types and encoding rules given in this document follow the specifications of [RD.1].

## **2 INTRODUCTION TO THE MSG SPECIFIC OSI REFERENCE MODEL**

### **2.1 Communication Concept**

The MSG dissemination service performs the acquisition, reformatting, compression and encryption of all or a sub-set of the mission data for distribution to the user community via two physical channels of the MSG space segment.

In principle, the data streams of both physical channels follow the same specification. One channel contains the Low Rate Information Transmission (LRIT), the other the High Rate Information Transmission (HRIT). Differences in the channel bandwidth will make a diversification between LRIT and HRIT necessary w.r.t. the contents and the resolution of the disseminated data.

The HRIT channel aims at Primary Data Users who wish to receive and process a comprehensive set of information, such as European meteorological centres, regional area forecast centres, research laboratories. The LRIT channel will only contain sub-sets of data or data sets compressed by higher factors. These will usually satisfy Secondary Data Users with limited capability in terms of acquisition and processing, such as smaller National Met. Services, universities, commercial companies, individuals etc.

This document conforms to [AD.1], which is based on the CCSDS AOS, Network and Data Links, Architectural Specification [RD.3]. The following sections will repeat certain concepts already presented in the above documentation up to a level necessary to understand the overall concept.

Table 2-1 presents the defined ISO/OSI layers from top to bottom and the equivalent functionality included in the LRIT/HRIT communication model from the view of the transmission service.

The complete required LRIT/HRIT formatting is performed by various MSG GS elements as listed in the most right column of the table. The DADF and the PGS are geographically separated from each other. Consequently, there will be a communication link (with its own protocol) between these two sites. The VCDUs generated by the data link layer processing of the DADF will form the application data units of this DADF - PGS communication link. It can be assumed that the chosen communication protocol for this link is transparent to the 'LRIT/HRIT application data units' and, therefore, is irrelevant for this document.

OSI Layer	Layer Functionality	MSG GS elements involved
Application layer	- acquisition of application data from the various MSG ground segment facilities	data flow from IMPF, MPEF, GTS, PGS, SGS, service message input to DADF
Presentation layer	- image segmentation - formatting to LRIT/HRIT file structure	DADF
session layer	- compression (if required) - encryption (if required)	DADF
Transport layer	- determination of APID - sequencing of transport files according to timeliness requirements - split of files into source packets - generation of 'idle packets'	DADF
Network layer	- determination of VC-ID	DADF
data link layer	- assembly of source packet into M_PDUs - multiplexing - assembly of VCDUs	DADF
communication link between DADF and PGS		
data link layer (c'tnd)	- generation of 'idle frames' - Reed-Solomon coding - Randomisation - attachment of sync marker	PGS
Physical layer	- serialisation - Viterbi coding - modulation	PGS

***Table 2-1 – LRIT/HRIT ISO/OSI Layer Functionality***

## 2.2 MSG Dissemination Time Management

The MSG image dissemination is no longer governed by fixed time slots as in the MOP/MTP concept. Instead, the time management in the MSG system is based on a flexible repeat cycle concept.

A MSG repeat cycle is defined by the time interval between two successive starts of the SEVIRI radiometer scan. The repeat cycle is not bound to absolute time references. Start and end of a repeat cycle will be defined by records within image segment header and trailer files.

The nominal dissemination service will maintain a regular, periodic distribution of the full Earth disc for 11 spectral channels and a reduced section of the Earth disc for the HRV channel.

Foreign satellite data, Meteorological Products and DCP Messages having different periodicity will be interleaved with the SEVIRI level 1.5 data according to a priority scheme based on timeliness requirements defined by the MSG end-user community. The timeliness will be achieved by sequencing the data flows in accordance to given priorities.

## 2.3 List of acronyms

<i>Acronym</i>	<i>Definition</i>
<b>AMV</b>	Atmospheric Motion Vector (MPEF product, BUFR coded)
<b>AOS</b>	Advanced Orbiting Systems
<b>APID</b>	Application Process Identifier
<b>AVHRR</b>	Advanced Very High Resolution Radiometer
<b>BER</b>	Bit Error Rate
<b>BPSK</b>	Binary PSK
<b>BUFR</b>	WMO standard for the coding of binary information
<b>C-VCDU</b>	Coded VCDU
<b>CCSDS</b>	Consultative Committee for Space Data Systems
<b>CGMS</b>	Coordination Group for Meteorological Satellites
<b>CLA</b>	Cloud Analysis (MPEF product, BUFR coded)
<b>CLAI</b>	Cloud Analysis (MPEF product in imagery form)
<b>CLM</b>	Cloud Mask
<b>CSR</b>	Clear Sky Radiance (MPEF product)
<b>CTH</b>	Cloud Top Height (MPEF product)
<b>DADF</b>	Data Acquisition and Dissemination Facility
<b>DCP</b>	Data Collection Platform
<b>DCT</b>	Discrete Cosine Transformation
<b>DEC</b>	Decryption Process
<b>DEC3</b>	Triple Decryption Process
<b>DES</b>	Data Encryption Standard
<b>DHT</b>	Define Huffman Table (JPEG marker)

<i>Acronym</i>	<i>Definition</i>
<b>DQT</b>	Define Quantisation Table (JPEG marker)
<b>DRI</b>	Define Restart Interval (JPEG marker)
<b>Eb/No</b>	Bit Energy / Noise Density
<b>ENC</b>	Encryption Process
<b>ENC3</b>	Triple Encryption Process
<b>EUMETSAT</b>	European Meteorological Organisation for the Exploitation of Meteorological Satellites
<b>FEC</b>	Forward Error Correction
<b>FSD</b>	Foreign Satellite Data
<b>GII</b>	Global Instability Index
<b>GMS</b>	Geostationary Meteorological Satellite (operated by JMA, Japan)
<b>GOES-E</b>	Geostationary Operational Environmental Satellite - Eastern location (operated by NOAA, U.S.)
<b>GOES-W</b>	Geostationary Operational Environmental Satellite - Western location (operated by NOAA, U.S.)
<b>GOMS</b>	Geostationary Operational Meteorological Satellite (operated by the Russian Federation)
<b>GRIB</b>	WMO standard for the coding of binary information
<b>GRID</b>	WMO standard for the coding of grid files
<b>GTS</b>	Global Telecommunications System
<b>HRIT</b>	High Rate Information Transmission
<b>HRUS</b>	HRIT User Station
<b>HRV</b>	High Resolution Visible (SEVIRI spectral channel)
<b>ICD</b>	Interface Control Document
<b>IMPF</b>	Image Processing Facility
<b>ISO</b>	International Organisation for Standardisation
<b>JPEG</b>	Joint Photographic Expert Group
<b>KMC</b>	Key Management Centre
<b>LRIT</b>	Low Rate Information Transmission
<b>LRUS</b>	LRIT User Station
<b>LSB</b>	Least Significant Bit
<b>MCC</b>	Mission Control Centre
<b>MCP</b>	Mission Communication Package
<b>METOP</b>	Meteorological Operational Satellite
<b>MPEF</b>	Meteorological Product Extraction Facility
<b>MSB</b>	Most Significant Bit
<b>MSG</b>	Meteosat Second Generation
<b>MSK</b>	Master Key
<b>MUBM</b>	MSG User Station Baseband Module
<b>NOAA</b>	National Oceanic and Atmospheric Administration
<b>NRZ-L</b>	NRZ-Level (Non-Return-to-zero)
<b>OFB</b>	Output Feedback (DES mode)
<b>PBK</b>	Public Key (encryption)
<b>PCM</b>	Pulse Code Modulation
<b>PFL</b>	Probability of Frame Loss
<b>PGS</b>	Primary Ground Station

<i>Acronym</i>	<i>Definition</i>
<b>PN</b>	Pseudo Noise
<b>PNK</b>	Pseudo Noise Key (encryption)
<b>QPSK</b>	Quarternary PSK
<b>RF</b>	Radio Frequency
<b>RST</b>	Restart Marker (JPEG)
<b>RTH</b>	Regional Telecommunications Hub
<b>RX</b>	Receiver
<b>S/C</b>	Spacecraft
<b>SEVIRI</b>	Spinning Enhanced Visible and Infra-Red Imager
<b>SGICD</b>	Space-to-Ground ICD
<b>SGS</b>	Support Ground Segment
<b>SHIP</b>	WMO standard for the coding of synoptic messages from ships
<b>SKU</b>	Station key Unit
<b>SOF</b>	Start of Frame (JPEG marker)
<b>SOS</b>	Start of Scan (JPEG marker)
<b>SYNOP</b>	WMO standard for the coding of synoptic messages
<b>T.4</b>	ITU-T standard (Group 3 facsimile)
<b>TBC</b>	To Be Confirmed
<b>TBD</b>	To Be Defined
<b>TH</b>	Tropospheric Humidity
<b>TOZ</b>	Total Ozone
<b>TX</b>	Transmitter
<b>VCDU</b>	Virtual Channel Data Unit
<b>WMO</b>	World Meteorological Organisation

### 3 APPLICATION LAYER

The application layer provides specific services between application processes. In the context of the MSG, the application data flow consists of received ground segment packets via external interfaces. This section presents an overview of the application data streams of the various Ground Segment Facilities to the DADF.

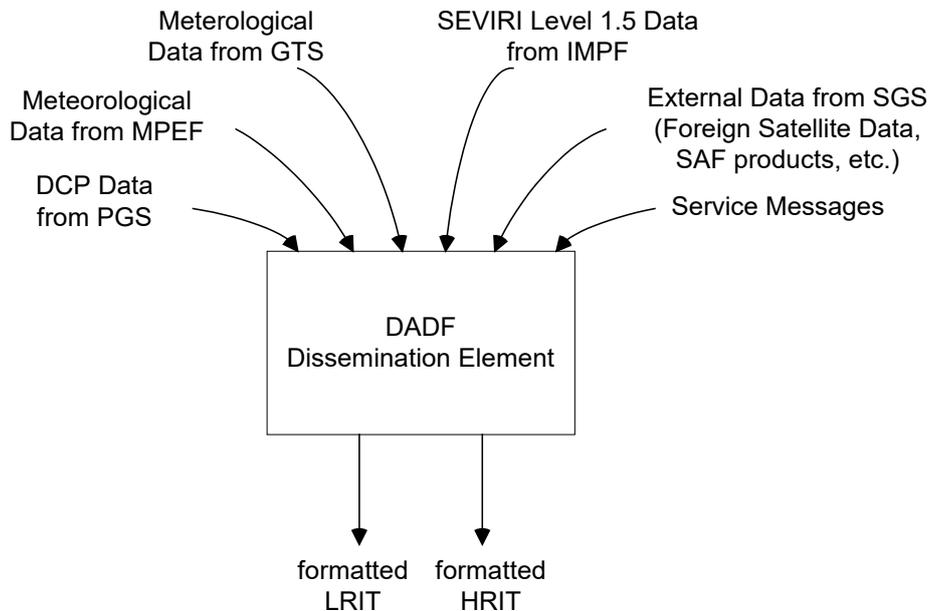
#### 3.1 Input Data Streams

##### 3.1.1 General

The DADF receives a number of data streams via its external interfaces. These interfaces may differ from each other in terms of physical and logical parameters (e.g. data rate, communication protocols, protocol, data structure etc.).

The data flows that will be received by the DADF (see Figure 3-1) are:

- SEVIRI Level 1.5 Data from the Image Processing Facility (IMPF)
- Meteorological Products from the Meteorological Product Extraction Facility (MPEF)
- Meteorological Data & Products from the Global Telecommunication System (GTS)
- Data Collection Platforms (DCP) Data received by the PGS
- External Data from Support Ground Segments (SGS)
- Service Messages generated within the DADF



**Figure 3-1 – Data Stream to MSG DADF Dissemination Element**

For further information about the data flow in the MSG Ground Segment the reader should refer to [RD.6].

The DADF will acquire the data in full size and their original resolution, check their consistency, extract and forward them to the presentation layer for further processing.

A brief description of the application data can be found in following sections 3.1.2 to 3.1.7.

### **3.1.2 MSG SEVIRI Level 1.5 Data**

Level 1.5 image data corresponds to the geolocated and radiometrically pre-processed image data, ready for further processing, e.g. the extraction of meteorological products. Any S/C-specific effects have been removed, and in particular, linearisation and equalisation of the image radiometry has been performed for all channels. The on-board blackbody data has been processed. Both radiometric and geometric quality control information is included. The DADF will receive the SEVIRI level 1.5 image data line-wise from the IMPF. Under normal conditions the repeat cycle will be 15 minutes. The image data of each repeat cycle will be accompanied by a header containing the start conditions and concluded by a trailer providing detailed repeat cycle quality information.

### **3.1.3 Meteorological Data & Products from GTS**

The meteorological data & products acquired from the GTS via the Regional Telecommunication Hub (RTH) Offenbach will contain GTS bulletins listed in Appendix A.2. This service is a continuation of the successful Meteosat Data Distribution (MDD) which serves as a GTS data relay aiming at countries with a poor telecommunication infrastructure.

In addition any products generated by external EUMETSAT application ground segments (e.g. Satellite Application Facilities (SAF) operated by National Met. Services) can be received via this interface.

### **3.1.4 Meteorological Products from MPEF**

The MPEF generates various meteorological products and distributes them via the GTS and the LRIT/HRIT dissemination. The meteorological products destined for dissemination are received by the DADF in the form of:

- WMO conform coded bulletins
- imagery type of representation
- overlay type of representation

An MPEF product list and format structure is contained in Appendix A.5.

### **3.1.5 DCP Data**

Observations and environmental data collected from Data Collection Platforms (DCP) which are located within the field of view of MSG will be relayed via the S/C, received by the PGS, processed and then routed to the DADF for dissemination.

After data processing, the DCP data will be retransmitted via LRIT in the form of concatenated DCP messages. See section A.6.

### 3.1.6 External Data from SGS

External data are by definition received via external dedicated links to various Support Ground Segments (SGS). The following external data types can be identified:

- Foreign Satellite Data (FSD)

The MSG dissemination system will provide a relay of foreign satellite data comprising the current geostationary satellite systems (GOES-E, GOES-W, GMS/MTSAT). A near global coverage will be achieved several times a day.

A list of the FSD contained in the LRIT/HRIT dissemination is given in Appendix A.1.

### 3.1.7 Service Messages

Service messages are data which are generated within the MSG GS to provide the end-users with regular operational information (e.g. administrative and encryption key messages) and more irregular mission specific support data (e.g. test patterns, grid/coastlines and algorithm updates).

The service messages will contain either text orientated or image type of data.

The reference test messages for compression and encryption are defined in the Appendices A.3 and A.4.

## 3.2 Rearrangement of Input Data

Due to the timeliness requirement defined by the end-user community and allowing for a better management of the LRIT/HRIT channel occupation and flexibility concerning the usage of compression schemes, the complete earth's disk image data will be divided into segments. Each of them forms a separate LRIT/HRIT file.

These files will be called *image segment files* from now on. They will contain a fixed number of lines. As a baseline, all image segment files provided via LRIT/HRIT will contain 464 lines. This value will in principle be configurable but it is the intention to determine an optimum value at the start of MSG operation and will then be kept stable. SEVIRI level 1.5 data, FSD and MPEF products (of overlay and imagery type) will have to follow the same rules.

Other data types (e.g. DCP/GTS or key messages) will either be concatenated or segmented not to exceed or fall below certain file size limits. Services messages will be of a size not requiring any rearrangement.

Further information about data segmentation and concatenation can be found in Section 4.

## 4 PRESENTATION LAYER

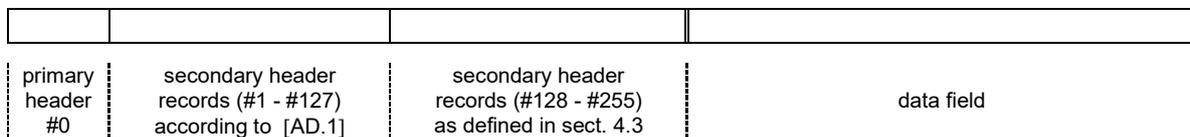
The presentation layer defines the uniform formatting of data. This layer receives the data streams as defined in chapter 3 from the application layer. The transfer mechanism of the MSG dissemination service is based upon the transfer of data units that are called **LRIT/HRIT files**. These files are the output of the presentation layer and their structure is explained in the following.

### 4.1 Structure of LRIT/HRIT Files

Each application data unit to be distributed via MSG dissemination will be formatted to a LRIT/HRIT file. An LRIT/HRIT file consists of one or more header records and one data field.

The primary header record is mandatory and defines the file type and the sizing of the complete LRIT/HRIT file. Depending on the file type, one or more secondary headers may be required to provide ancillary file information (see Section 4.3).

The file type number identifies the data contained in the data field. The global definitions of file types from #0 to #127 are defined in [AD.1]. Mission specific file type extensions are all file types from #128 and #255. For MSG, three additional file types (#128, #129 and #130) are specified. The description of LRIT/HRIT files and their data fields is contained in Section 4.2.



**Figure 4-1 – LRIT/HRIT File Structure**

## 4.2 LRIT/HRIT File Types and Data Field Descriptions

### 4.2.1 LRIT/HRIT File Types Overview

The ‘global’ file types (0 ... 127) are defined in [AD.1]. In addition, the mission specific file types (128 ... 130) are required for the MSG dissemination service to cover all data and information to be provided via the LRIT/HRIT dissemination. The file types (131 ... 255) are available for future expansion.

Table 4-1 specifies the ‘spread’ of all application data types identified and described in previous sections over the various LRIT/HRIT file types.

File type code	file type	Application data types/subtypes contained in the data field
<b>Global LRIT/HRIT file types</b>		
0	Image Data	Image data segments as described in sect. 4.2.2 - SEVIRI level 1.5 - FSD - Meteorological Products (imagery type/overlay type) - Service Messages of the type: Compression and Encryption Test Message Overlay files (grids/coastlines/political boundaries)
1	GTS Message	- MPEF Meteorological Products - SAF Meteorological Products - Meteorological Data & Products from GTS - DCP bulletins coded as GTS messages
2	Alphanumeric Text	Service Messages of the type: Administrative messages, newsletters, dissemination timetables, algorithm updates (alphanumeric source code)
3	Encryption Key Message	Support of MSG encryption scheme
4 ... 127	Reserved	(for further global usage)
<b>Mission specific LRIT/HRIT file types</b>		
128	Repeat Cycle Prologue	This file type provides additional information about the satellite/ image processing status known at the start of a SEVIRI level 1.5 data / FSD repeat cycle or processing information of meteorological products (see sect. 4.2.6 for further information).
129	Repeat Cycle Epilogue	This file type contains status information known at the end of a repeat cycle of SEVIRI level 1.5 data. (see sect. 4.2.7 for further information)
130	DCP Message	re-transmission of DCP messages (unprocessed) (see sect. 4.2.8 for further information)
131 ... 143	Reserved	(for further mission specific use)
144	Binary File	Support of binary file transmission (array of bytes)
145 ... 255	Reserved	(for further mission specific use)

**Table 4-1 – LRIT/HRIT File Types**

#### **4.2.2 File Type #0 - Image Data**

File type #0 will be used for all SEVIRI level 1.5, FSD, Meteorological Products (imagery type) and overlay data (grid and coastlines). The LRIT/HRIT data field of file type #0 contains bitmap data in accordance with the specifications in [AD.1].

The dissemination via MSG will only distribute images of a pixel resolution of 1, 8, 10 or 12 bit per pixel.

Due to the timeliness requirement, the image data files will only contain image segments (see Section 4.2.2.1 for further details).

The file type #0 may contain compressed and/or encrypted data. Further information about the algorithms applied to the image data content due to data compression and encryption can be found in Section 5 (session layer processing).

#### 4.2.2.1 SEVIRI Level 1.5 Data

The complete Earth's disk of MSG SEVIRI level 1.5 images will have a size of:

- 5568 x 11136 pixel for the HRV channel
- 3712 x 3712 pixel for the other 11 SEVIRI channels

For EUMETCast dissemination the following baseline applies:

With an image segment size of 464 lines , one complete Earth image will consist of:

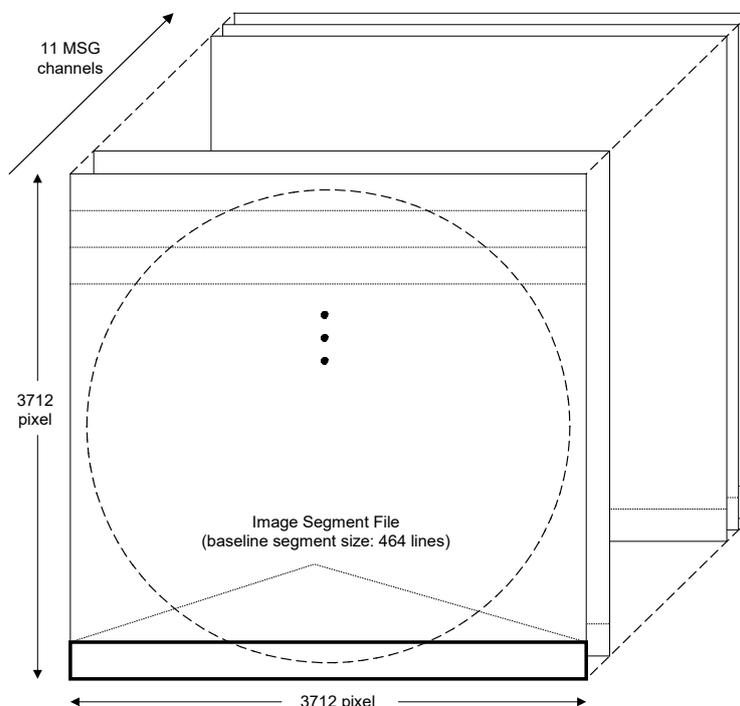
- 24 image segment files in the HRV channel (464 lines of 11136 pixels)
- and
- 8 image segment files for any other spectral channel (464 lines of 3712 pixel)

For direct dissemination the following baseline applies:

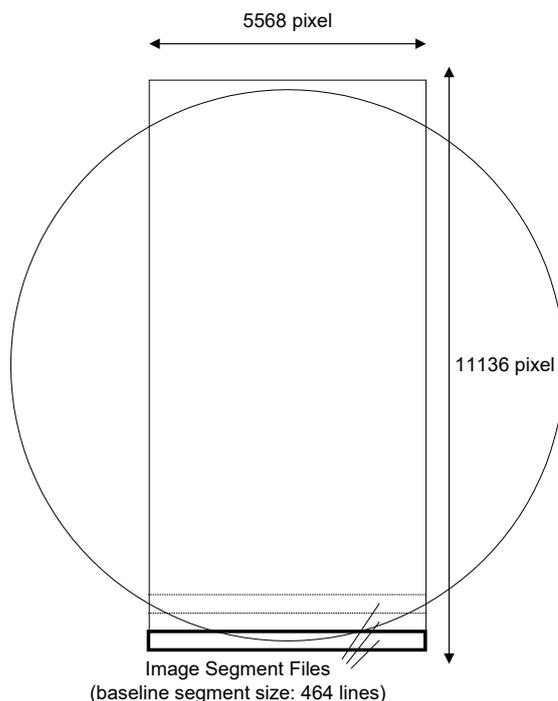
With an image segment size of 64 lines , one complete Earth image will consist of:

- 174 image segment files in the HRV channel (64 lines of 11136 pixels)
- and
- 58 image segment files for any other spectral channel (64 lines of 3712 pixel)

The line and column numbering of the image segment files will be identified by the Header Type #2, Image Navigation.



**Figure 4-2 – MSG LRIT/HRIT Image Data File Structure of 11 VIS, IR and near IR SEVIRI Channels**

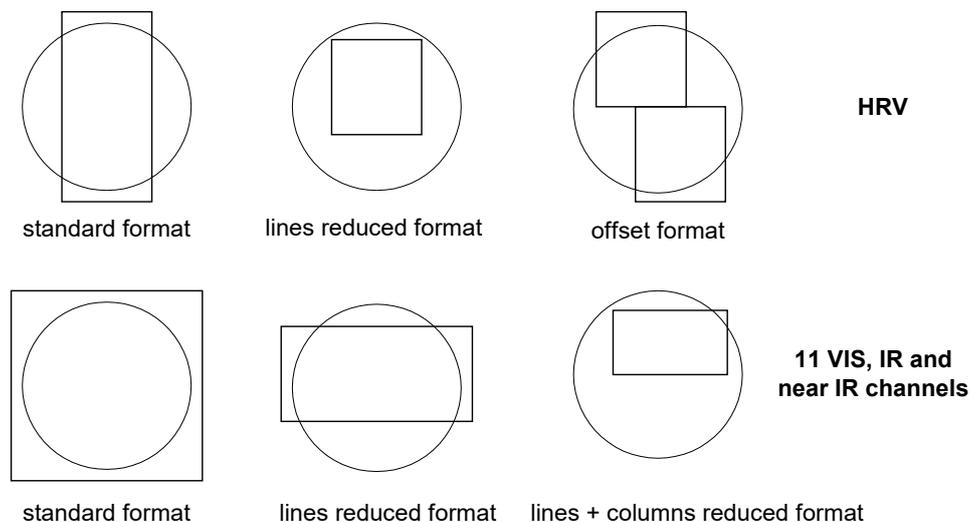


**Figure 4-3 – MSG LRIT/HRIT Image Data File Structure of the HRV SEVIRI Channel**

The MSG S/C supports ‘reduced line’ scans to allow for the dissemination at shorter intervals than the nominal repeat cycle. As a baseline, the resulting dissemination formats of the reduced line of offset scans may have any offset of multiples of 464 lines for EUMETCast dissemination and 64 lines for direct dissemination. In general this mechanism allows the DADF to create dissemination formats forming geographical subsets of the acquired input data (of either full Earth disk or ‘reduced line’ scans).

The image segments will be numbered. A fixed relationship between the image segment number (identified via header type #128 – image segment identification, see Section 4.3.2.9) and the line offset (header type #2 - image navigation, see Section 4.3.2.3) will be established. In addition, the Segment Identification header will define the first and the last segment number of the image segments forming part of a repeat cycle. The image segment numbering direction will be in line with the radiometer scan direction.

In the case of HRV offset scans the upper and the lower boundaries of the dissemination formats may suffer from either overlap or missing data due to the rectification process applied to the scanned lines.



**Figure 4-4 – Image Repeat Cycle Dissemination Formats**  
*(the 'lines reduced' or 'lines + columns reduced' dissemination formats will only be used in exceptional cases)*

As a baseline, the image data will contain the space area, which will artificially be set to zero. Small segments of 'real space data' may be included. Their existence and position will be defined in the Repeat Cycle Prologue (file type #128).

For further information on radiometric,geometric properties and image line structure of SEVIRI 1.5 data the reader should refer to [RD.5].

#### 4.2.2.2 Foreign Satellite Data

The DADF will treat the FSD with the same segmentation and segment numbering strategy as explained in the previous section. The size of the FSD will be multiples of 464 lines for EUMETCast dissemination and 64 lines for direct dissemination. Various parameters of secondary headers related to image size, pixel resolution, navigation and calibration will depend on the properties of the Foreign Satellites.

In addition to the image data files, Repeat Cycle Prologues (file type #128) will be used to inform the users about the calibration status of the contained image data. A definition of the Repeat Cycle Prologues vs. Foreign Satellite Data types is listed in Appendix A.1.

For Foreign Satellite Data the following reference documents apply: GOES [RD 17], OpenMTP [RD 24], MTSAT [RD 9]

#### 4.2.2.3 Meteorological Products (Imagery Type)

Currently no products of this type are disseminated.

#### 4.2.2.4 Meteorological Products (Overlay Type)

Currently no products of this type are disseminated

#### 4.2.2.5 Service Messages (Overlay Type)

The overlay information (containing coastlines/grid/political boundaries) is handled in the same manner as image data. The corresponding projection and coverage information are defined by the Image Navigation header #2.

As an overlay type of image contains 1 bit/pixel of information only, a dedicated lossless compression mechanism will be used (see Section 5.3.1.4).

Overlay-type images (1 bit/pixel) will be segmented not to exceed a certain LRIT/HRIT file size. The maximum segment size in number of lines will be configurable, but will apply to all overlay-type images independent of their data type.

#### 4.2.3 File Type #1 - GTS Message

The file type ‘GTS Message’ will contain data coded on conformance to [RD.10] (e.g. SYNOP, SHIP, GRID, BUFR, GRIB, T.4 coded fax charts, etc.) received from MPEF, GTS, SGS or DCPs.

(Note: In the context of this document the expression ‘GTS Message’ is used to describe data sets which are coded in accordance with [RD.10]. This file type is intended to deliver ‘GTS-type’ information via the MSG dissemination but it will not be used for distribution via the GTS.)

The file type #1 may contain a single GTS Message or a concatenation of several messages. The following concatenation mechanism is used within the data field of the LRIT/HRIT file:

```
GTS_Messages ::= RECORD
{ Messages VARIABLE ARRAY SIZE (1..65535) OF
    VARIABLE ARRAY SIZE (1..15000) OF BYTE
}
```

**Messages** is a variable length array of messages which are themselves variable arrays of bytes forming a WMO message coded according to [RD.10].

The DADF may rearrange (i.e. separate out and/or concatenate) the GTS messages received from other Ground Segment facilities to produce LRIT/HRIT files not exceeding or falling below a certain size.

The starting line of each GTS message will contain a transmission sequence number ‘nnn’ as defined in [RD.10]. Due to the possible rearrangement of GTS messages within the dissemination element a correct sequence of the transmission sequence number will not be guaranteed. Consequently, the user shall not use the transmission sequence number to check neither the sequence nor the completeness of received GTS messages. For these purposes counters on lower communication layers should be used.

The products originating from MPEF, GTS and SAF making use of the file type #1 are included in Appendices A.2, A.5 and A.6.

In addition DCP messages coded as SYNOP bulletins or other WMO formats are distributed via this file type.

#### **4.2.4 File Type #2 - Alphanumeric Text File**

This file type would mainly be used for the regular distribution of text-type service messages.

These messages have to be generated and updated within the MSG GS. Each message will have to contain a unique sequence number which allows to discard messages by the user which have already previously been received.

The annotation record (header type #4) will provide the means to distinguish between different files of this type.

The following types of text-type service messages are defined:

##### **Administrative Messages**

Regular dissemination of operational information (e.g. spacecraft status and events, dissemination performance statistics, etc.).

##### **Dissemination Tables**

These tables define the dissemination baseline in a structured form, The dissemination table parameter set will include information about the planned dissemination element configuration for LRIT and HRIT (e.g. compression type, planned start of repeat cycles, timeliness, etc.).

##### **Newsletters**

An irregular dissemination of information of general interest is foreseen via the use of newsletter-type of messages.

##### **Algorithm Updates**

The algorithm updates are an irregular distribution of descriptive text or software code examples for new dissemination or product algorithms.

Alphanumeric Type of Messages will be not be segmented.

#### **4.2.5 File Type #3 - Encryption Key Message**

The Encryption Key Message will contain a complete set of encrypted Message Keys (so-called Public Keys) of all Key Numbers for all authorised user stations. The Encryption Key Message may be segmented into several with each of them containing a subset of the total number of public keys in order not to exceed a certain LRIT/HRIT file size.

In the case of Encryption Key Message segmentation, the segment borders will be set that the public keys (PBK) pertaining to one User\_Station\_Number will be kept in one Encryption Key Message.

Encryption_Key_Message	::=	VARIABLE ARRAY SIZE (1 .. Max_Number_Key_Messages) OF RECORD
{		
User_Station_Number		UNSIGNED SHORT
Key_Number		UNSIGNED BYTE
Public_Key		UNSIGNED SIZE (192)
Public_Key_CRC		UNSIGNED SHORT
}		
Max_Number_Key_Messages	=	65536
<i>Note: Although the maximum number of key messages contained in file type #3 can theoretically go up to 65536, the DADF will be able to limit the number to a lower value not to exceed a certain user configurable maximum file size.</i>		

**User\_Station\_Number** is a number uniquely assigned to each authorised user station equipped with a Station Key Unit (SKU). The User\_Station\_Number is identical to the number displayed on the SKU.

**Key\_Number** is an index number for a particular Message Key used for encryption. The Key\_Number is also used to identify the Message Key used in encrypted LRIT/HRIT files via the Key Header type #7 (see Section 4.3.2.8).

**Public\_Key** contains the Message Key of Key\_Number encrypted against the Master Key belonging to User\_Station\_Number.

**Public\_CRC** contains a check sum in accordance with the definition given in Appendix C.2.

For further details on the definition of the encryption algorithm and related parameters the reader should refer to Section 5.4.

#### 4.2.6 File Type #128 - Repeat Cycle Prologue

The Repeat Cycle Prologue will be disseminated before or at an early stage of a repeat cycle of SEVIRI L1.5 images, Foreign Satellite Data and Met. Products. These include files of types #0 and #1. The repeat cycle prologue will contain information about the satellite status and/or the image/product processing known at the start of that particular repeat cycle. The repeat cycle prologues contain a version number which allows to uniquely identify their structure per data type. The definition of a data type is identical to the one of ProductID(1) in the annotation (header type #4, see Section 4.3.2.5)

The Repeat Cycle Prologue files will not be segmented.

##### 4.2.6.1 SEVIRI Level 1.5 Data Prologue Definition

The data field of the Repeat Cycle Prologue for SEVIRI 1.5 will contain the following subset of 15HEADER records defined in [RD.5]:

- SatelliteStatus
- ImageAcquisition

- CelestialEvents
- ImageDescription
- RadiometricProcessing
- GeometricProcessing

For SEVIRI Level 1.5 images, the repeat cycle will also include a repeat cycle epilogue (file type #129, see Section 4.2.7).

#### **4.2.6.2 Foreign Satellite Data Prologue Definition**

For all Foreign Satellite Data, the Repeat Cycle Prologue data field will contain an SGS\_Common\_Header record and an SGS\_Product\_Specific\_Header record as defined in Appendix A.1.

A Foreign Satellite Data repeat cycle will not be concluded by a repeat cycle epilogue.

#### **4.2.6.3 Meteorological Products Prologue Definition**

For the Meteorological Products originating from MPEF, the Repeat Cycle Prologue data field will contain an MPEF\_Product\_Header record and an MPEF\_ProductSpecific\_Header record as defined in Appendix A.5.2.

For the Meteorological Products originating from SAFs, the Repeat Cycle Prologue data field will contain the SGS\_Common\_Header and the SGS\_Product\_Specific\_Header records as defined in Appendix A.6. Current SAF products disseminated by EUMETCast are not covered by this document.

A Meteorological Product repeat cycle will not be concluded by a repeat cycle epilogue.

(Note: In the MSG context a Meteorological Product repeat cycle is identical to one product extraction time.

#### **4.2.7 File Type #129 - Repeat Cycle Epilogue**

The repeat cycle epilogue will be disseminated at a late stage or after the end of a SEVIRI 1.5 image repeat cycle. The use of this file type is limited to the case of the SEVIRI Level 1.5 image data. The image repeat cycle epilogue will contain information known to the MSG ground segment at the end of an image repeat cycle. The data field of the Repeat Cycle Epilogue will contain all records of the 15TRAILER as defined in [RD.5].

These are:

- 15TRAILERVersion
- ImageProductionStats\_Record
- NavigationExtractionResults\_Record
- RadiometricQuality\_Record
- GeometricQuality\_Record
- TimelinessAndCompleteness\_Record

The Repeat Cycle Epilogue files will not be segmented.

#### 4.2.8 File Type #130 - DCP Message

DCP Messages can contain all sorts of data (any clear ASCII coded text, GTS conform messages or under certain condition even proprietary binary data). Therefore, the re-transmission of ‘unprocessed’ DCP Messages cannot make use of any global file type and requires a mission specific file type.

In addition to the DCP message contents, fixed length quality data (frequency offset, signal strength, modulation index) will be included.

The file type #130 may contain a single DCP message or a concatenation of several messages. The following concatenation mechanism is used within the data field of the LRIT/HRIT file:

<pre>DCP_MESSAGE_DATA_FIELD ::= VARIABLE ARRAY SIZE OF RECORD {     DCP_QUALITY     DCP_MESSAGE }</pre>
---

DCP Messages will be concatenated not to exceed or fall below a certain LRIT file size. The maximum file size is currently 100 Kbytes. If the accumulation of DCP messages in the currently filled file has not reached this size after 10 minutes, it will be closed and sent with the actual smaller size.

These parameters may be changed to optimise dissemination performance.

A detailed definition of the DCP\_QUALITY and DCP\_MESSAGE record structure is defined in Appendix A.6.

#### 4.2.9 File Type #144 - Binary File

The file type ‘Binary File’ will contain data coded in a transparent way for the dissemination, i.e. only the sender and the receiver need to agree on the content and structure of the file content. For the CLM product this is described in Appendix A.5.

The file type #144 may contain any type of information in clear form, user specific encryption or user specific compression in any data format selected by the producer of the file.

As mentioned in Section 5.3.1 compression is only allowed for file type#0, i.e. also the new file type is not compressed by the dissemination.

XRIT files of file type #144 will typically be split into segment files of configurable size by the dissemination. The segment file size is only technically limited by the implementation in using an unsigned double (in XRIT header 0) for the definition of the data field. Practically it is assumed that the file size per file segment will typically be in the range between 2 Mbytes and 10 Mbytes.

Binary File::= VARIABLE ARRAY SIZE (1...4 GB) OF BYTE

The products originating from MPEF making use of the file type #144 are included in Appendix A.5.

#### 4.2.9.1 Meteorological Products in Binary Form

The meteorological products disseminated in binary form are handled in the same manner as GTS messages, however grouping of several products in one XRIT file is not supported. The DADF will apply the same segmentation and segment numbering strategy as explained in Section 4.2.2.1 for meteorological products of imagery type. The size of these will be specified in bytes.

The following products fall into this category:

Product Origin	Product Name	# of segments via dissemination
MPEF Product	CLM	6
MPEF Product	CTH	2
MPEF Product	CLAI	3
MPEF Product	CRM	10 (average – variable)

Additional information about the products of this category, their related repeat cycle prologues and specific headers, is provided in Appendix A.5.

### 4.3 LRIT/HRIT File Header Types

#### 4.3.1 General

The dissemination service will use the header types #0 - #7 of the LRIT/HRIT Global Specification as defined in [AD.1] and the mission specific headers #128 - #129 (see Table 4-2).

Code	Header Record Type	Structure
<b>Headers as defined in LRIT/HRIT Global Specification [AD.1]</b>		
0	primary header	according to [AD.1], LRIT/HRIT Global Spec.
1	image structure	"
2	image navigation	"
3	image data function	"
4	Annotation	"
5	time stamp	"
6	ancillary text	"
7	key header	"
8-127	(reserved for further global usage)	"
<b>Mission Specific Headers</b>		
128	segment identification	see sect. 4.3.2.9
129	image segment line quality	see sect. 4.3.2.10
130 – 255	(reserved for further mission specific usage)	

*Table 4-2 – Adaptation of LRIT/HRIT Header Types*

### 4.3.2 Definition of Header Types

The fields `Header_Type` and `Header_Record_Length` used in the following subsections are used as defined in [AD.1].

#### 4.3.2.1 Header Type #0 - Primary Header

The structure of the primary header record is defined as:

<b>Title: Primary Header Record</b>	<b>Id: PRIMARY HEADER</b>	
PRIMARY HEADER ::=	RECORD	
{ Header_Type	UNSIGNED BYTE (0)	-- fixed value
Header_Record_Length	UNSIGNED SHORT (16)	-- fixed value
File_Type_Code	ENUMERATED BYTE	-- defines file type
	{ image data file (0),	
	GTS Message (1),	
	alphanumeric text file (2),	
	encryption key message (3),	
	repeat cycle prologue (128),	
	repeat cycle epilogue (129),	
	DCP message (130),	
	binary file (144)}	
Total_Header_Length	UNSIGNED	-- variable
		specifies total size of all header records.
Data_Field_Length	UNSIGNED DOUBLE	-- specifies total size of the LRIT/HRIT file data field in bits. For image data files, this parameter will be completed after compression of the data field.
}		

**Table 4-3 – Primary Header**

#### Explanations:

- `File_Type_Code`

The `File_Type_Code` specifies the formatting of the data to be transmitted via LRIT/HRIT files. The relationship between application data types and `File_Type_Code` is as defined in Table 4-1.

### 4.3.2.2 Header Type #1 - Image Structure

The structure of the image structure record is defined as:

Title: Image Structure Record	Id: IMAGE_STRUCTURE	
IMAGE_STRUCTURE	::=	RECORD
{ Header_Type		UNSIGNED BYTE (1) -- fixed value
Header_Record_Length		UNSIGNED SHORT (9) -- fixed value
NB		UNSIGNED BYTE -- number of bits per pixel
NC		UNSIGNED SHORT -- number of columns
NL		UNSIGNED SHORT -- number of lines
Compression_Flag		ENUMERATED BYTE -- compression method
		{ no compression (0),
		lossless compression (1),
		lossy compression (2) }
}		

**Table 4-4 – Image Structure**

#### Explanations:

- NB (number of bits per pixel)

The value for NB will be either 1, 8, 10 or 12 bit/pixel

(Note: Sub-images as described in [AD.1, Section 4.3] may be used for Meteorological Products and FSD to include quality information and pixel padding beside the image data. For the definition of such sub-image structures the reader shall refer to the image data function (header #3) in Section 4.3.2.4.

- NC (number of columns)

The value for NC will be:

5568

For SEVIRI HRV channel

3712

For SEVIRI other channels

Any multiple of 464 (baseline segment value) For other image data (met. products, FSD, ...)

- NL (number of lines)

Due to the image segmentation the value of NL will be:

For EUMETCast:

464 (baseline value)  
compression

For all imagery products using JPEG and Wavelet

For direct dissemination:

64 (baseline value)

For all imagery products using JPEG and Wavelet  
compression

Any value

For overlay type images using T.4  
compression (refer to Header Type  
#128/Data Representation field for the  
identification of the applied  
compression algorithm)

- Compression\_Flag

The *Compression\_Flag* defines the compression method (lossless or lossy). The applicable compression algorithm and its inherent data representation is defined by the *Data\_Representation* field of the header type #128.

### 4.3.2.3 Header Type #2 - Image Navigation

The structure of the image navigation record is defined as:

Title: Image Navigation Record	Id: IMAGE_NAVIGATION	
IMAGE_NAVIGATION ::=	RECORD	
{ Header_Type	UNSIGNED BYTE (2)	-- fixed value
Header_Record_Length	UNSIGNED SHORT (51)	-- fixed value
Projection_Name	CHARACTERSTRING SIZE (32)	-- projection names as defined in [AD.1] and mission specific extensions
	{“GEOS(<sub_lon>)”	
	“POLAR(<prj_dir>,<prj_lon>)”	
	“MERCATOR”	
	“PSD”}	
CFAC	INTEGER	-- column scaling factor as defined in [AD.1]
LFAC	INTEGER	-- line scaling factor as defined in [AD.1]
COFF	INTEGER	-- column offset as defined in [AD.1]
LOFF	INTEGER	-- line offset as defined in [AD.1]
}		

**Table 4-5 – Image Navigation**

#### Explanations:

- Projection\_Name

The MSG LRIT/HRIT dissemination will make use of the following **Projection Names**:

“GEOS(<sub_lon>)”	For SEVIRI 1.5 image and all geostationary foreign satellite data The parameter <sub_lon> will be provided in the representation '±###.#'.
“POLAR(<prj_dir>,<prj_lon>)”	For image data in polar-stereographic projection from polar orbiters
“PSD”	For polar satellite data in their original scan and instrument swath width representation
Any of the above projections	For the distribution of overlay type meteorological products and service messages

All unused characters will be set to ASCII ‘space’ (‘20’h).

- CFAC / LFAC

The column and line scaling factors (CFAC and LFAC) contain variable values which depend on the input data and their specific segmentation approach.

The sign of the CFAC / LFAC values will define the spacecraft's scan direction. For a further description of the navigation functions the reader shall refer to [AD.1].

#### COFF / LOFF

COFF and LOFF are projection specific offsets and define the position of an image segment file window within the projection area.

For a further description of the navigation function the reader shall refer to [AD.1].

#### 4.3.2.4 Header Type #3 - Image Data Function

The structure of the image data function record is defined as:

Title: Image Data Function Record	Id: IMAGE_DATA_FUNCTION	
IMAGE_DATA_FUNCTION ::=	RECORD	
{ Header_Type	UNSIGNED BYTE (3)	-- fixed value
Header_Record_Length	UNSIGNED SHORT ()	-- variable value
Data_Definition_Block	ARRAY (Data_Def_Block_Size) of	-- variable size and contents in accordance with the descriptive language defined in [AD.1]
}	CHARACTERSTRING	

Note: Data\_Def\_Block\_Size = Header\_Record\_Length - 3

**Table 4-6 – Image Data Function**

#### Explanations

- Data\_Definition\_Block

This character string allows to define complex image data structures. In the MSG context, it is used to define overlay-type images and images which require to establish a relationship between their pixel values and an engineering unit. This header type will only be included in files containing imagery type or overlay type of Meteorological Products, FSD or service messages. The following paragraphs summarise the scope for using this header.

##### Meteorological Products (imagery type):

Meteorological Products of imagery type will contain a header type #3 to the relationship between pixel values and physical units. Example definitions are given in Appendix A.5.3.

##### Meteorological Products (overlay type) and Overlay files:

These single bit-plane images will include a header type #3 to define the ‘polarity’ of the bit map. An example definition is given in Appendix A.5.3.

##### Foreign Satellite Data

Foreign Satellite Data will include a header type #3 to describe the data calibration to establish a relationship between pixel values and radiances/temperatures or albedo. An example definition is given in Appendix A.1.

Service Messages (overlay type)

These single bit-plane images will include a header type #3 to define the ‘polarity’ of the bit map.

**4.3.2.5 Header Type #4 - Annotation**

For direct dissemination the annotation record should be used to identify more precisely the product/data type and sub-type of the LRIT/HRIT file. It is assembled to allow for quick and easy detection of the most relevant file contents criteria and route it to the respective post-processing function. This annotation record has also been utilised as the file name for LRIT and HRIT products disseminated via EUMETCast.

It can be assumed that all operating system in use in the MSG GS and at the user station sites will support long file names. Therefore, it is proposed to use the annotation text as a default distinctive file name. Besides being used for file storage purposes, this header will contain all criteria for the DADF encryption process in one header record. The user station can use the annotation to apply filter, sorting and processing criteria.

The structure of the annotation record is defined as:

<b>Title: Annotation Record</b>	<b>Id: ANNOTATION</b>	
ANNOTATION	::=	RECORD
{ Header_Type		UNSIGNED BYTE (4) -- fixed value
Header_Record_Length		UNSIGNED SHORT (64)
Annotation_Text		RECORD
{ XRITchannelID		CHARACTERSTRING SIZE (1)
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
DisseminationID		CHARACTERSTRING SIZE (3) Value between '000' and 999
Field Separator		CHARACTERSTRING SIZE (1) '.'
DisseminatingS/C		CHARACTERSTRING SIZE (6)
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
ProductID1		CHARACTERSTRING SIZE (12) see Table 4-8
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
ProductID2		CHARACTERSTRING SIZE (9) see Table 4-8
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
ProductID3		CHARACTERSTRING SIZE (9) see Table 4-8
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
ProductID4		CHARACTERSTRING SIZE (12) see Table 4-8
FieldSeparator		CHARACTERSTRING SIZE (1) '.'
Flags		CHARACTERSTRING SIZE (2)
}		
}		

**Table 4-7 – Annotations**

## Explanations

- Annotation\_Text

The Annotation\_Text record contains the following fields of character strings:

- **LRIT/HRIT Channel ID**

‘L’ for LRIT dissemination channel, ‘H’ for HRIT dissemination channel. For the first character of LRIT and HRIT file names appearing in the MSG Ground Segment and on the reception/User Station side, this Channel ID is used.

- **DisseminationID**

This ID identifies the dissemination source, covering the disseminating S/C (a commercial communication S/C for EUMETCast/DVB or MSG-2/MSG-3/MSG-4) and S/W entities in the MSG Ground Segment. The value can be between 000 and 999. Current values used as a baseline are:

EUMETCast (DVB)

100 Prime

000 C

Direct – via MSG-2 or MSG-3

101 Prime

001 Contingency

- **Disseminating S/C ID**

This field will contain the name of the disseminating S/C in alphanumeric form in case of direct dissemination (use of the dissemination transponder on Meteosat-n). In case of dissemination via EUMETCast, this ID contains the name of the S/C having taken the image used for dissemination.

The following names will be used: ‘MSG1\_\_’, ‘MSG2\_\_’, ‘MSG3\_\_’, ‘MSG4\_\_’, ‘MTP\_\_’ or ‘MSG\_\_’.

- **Product ID (1) – (4)**

Table 4-8 defines the contents of these fields.

**MSG Ground Segment LRIT HRIT Mission Specific Implementation**

Product ID (1) <sup>1)</sup>	S <sup>2)</sup>	Product ID (2) <sup>1)</sup>	S	Product ID (3) <sup>1)</sup>	S	Product ID (4) <sup>1), 3)</sup>	LRIT/ HRIT file type <sup>4)</sup>	APID <sup>5)</sup> LRIT/HRIT
12 char.	-	9 char.	-	9 char.	-	12 char.		
all S/C name <sup>6)</sup>	-	9 digit spectral channel name <sup>7)</sup>	-	6 digit file segment number <sup>8)</sup>	-	acquisition start time	#0	0/32 6/38 ...11/43 (depending on S/C)
'MSGi'	-		-	'PRO'	-	acquisition start time	#128	0/32
all S/C name but 'MSGi'	-	6 digit spectral channel name	-	'PRO'	-	acquisition start time	#128	6/38 ...11/43 (depending on S/C)
'MSGi'	-		-	'EPI'	-	acquisition start time	#129	
'DCP'	-	'DCP'	-	6 digit sequence number of day <sup>9)</sup>	-	acquisition time of oldest DCP message contained	#130	1/33
'DCP'	-	'WMO'	-	6 digit sequence number of day	-	acquisition time of oldest GTS bulletin contained	#1	1/33
'GTS'	-		-	6 digit sequence number of day	-	acquisition time of oldest GTS bulletin contained	#1	2/34
'MPEF'	-	product name <sup>10)</sup>	-	6 digit file segment number	-	nominal product time	#1	3/35
'MPEF'	-	product name	-	6 digit file segment number	-	nominal product time	#0	3/35
'MPEF'	-	product name	-	'PRO'	-	nominal product time	#128	3/35
'MPEF'	-	product name	-	6 digit file segment number	-	nominal product time	#144	3/35
'SAF'	-	product name	-	6 digit file segment number	-	nominal product time	#1	4/36
'SAF'	-	product name	-	6 digit file segment number	-	nominal product time	#0	4/36
'SAF'	-	product name	-	'PRO'	-	nominal product time	#128	4/36
'SERVICE'	-	service message name <sup>11)</sup>	-	sequence/ version number <sup>12)</sup>	-	nominal dissemination time	#0 or #2	5/37
'SERVICE'	-	'EKM'	-	first User Station Number of Key Message contained in file (4 digits in Hex) 13) + '_' + last User Station Number of Key Message contained in file (4 digits in Hex)	-	nominal dissemination time	#3	5/37

**Table 4-8 – Product ID (1) – (4) Definitions vs. Data Types and File Types**

Notes to Table 4-8:

- 1) The defined characterstrings for the ProductIDs (1) – (4) will be left aligned. Any remaining space in the field will be filled with ASCII characters ‘\_’ (‘5F’h).
- 2) The ProductIDs will separated from each other by the ASCII character ‘-’ (‘2D’h).
- 3) All times used in ProductID (4) have the notation “YYYYMMDDhhmm”
  - “acquisition start time” will identical to TrueRepeatCycleStartTime as defined in [RD.5] but rounded to a full minute
  - “acquisition time of the oldest ... contained” represents the entry time of a DCP message or GTS message into the Dissemination Element rounded to a full minute
  - “nominal product time” represents the NominalTime contained in MPEF\_Product\_Header or NominalSGSProductHeader in SGS\_Common\_Header rounded to a full minute.
  - “nominal dissemination time” represents the scheduled service message time
- 4) The file type column is included for information and cross-reference only. The numbering is in accordance with Table 4-1.
- 5) The APID column is included for information and cross-reference only. The numbering is in accordance with Table 6-1.
- 6) The S/C name are identical to GP\_SC\_Name as defined in [RD.1].
- 7) The spectral channel names consist of the central wavelength and the orbital position of the satellite in the form XX\_X\_YYYY where:
  - XX\_X are 4 digits identifying the central wavelength (e.g. ‘01\_6’ for 1.6  $\mu\text{m}$ , ‘10\_8’ for 10.8  $\mu\text{m}$ , ‘\_\_\_\_\_’ for wideband channels) and
  - YYYY are 4 digits identifying the orbital position (e.g. 075W for CGMS 075W or 140E for CGMS 140E)
- 8) The file segment number contains Segm\_Seq\_No of header type #128 in a 6 digit representation.
- 9) The sequence number the files sequentially starting from ‘0’ at midnight.
- 10) The product names are as defined in Appendices A.5.1 and A.6.1.
- 11) The service message names of image-type, overlay-type and alphanumeric-type of representation will be defined during their preparation in the DADF offline environment. The following categories are currently foreseen:
 

‘ADMIN’	for administrative messages (service details for the previous calendar day)
‘ALGORITHM’	for algorithm updates
‘COMP_TST1’	for compression test messages
‘COMP_TST2’	for compression test messages
‘EKM’	for encryption key messages
‘ENC_TEST’	for encryption test messages
‘NEWS’	for newsletters (real-time problems, fixing of problems)
‘OVL_CL’	for coast lines

'OVL_GRID'	for grid overlays
'OVL_PB'	for political boundaries
'REF-TEST'	for reference test messages
'REG-RPT'	for regular reports (scheduled announcements for the following week)
'TIMTABNNN'	for differences to 'TIMTABNOM'
'TIMTABNOM'	for details about the dissemination scheme (products per channel ...)
'USER-DATA'	for overlay files not of type 'OVL-xxx'

12) The sequence/version number has the structure:

sssss\_nnn

where 'sssss' is a five digit sequence number, incremented by one for each distinct service message, allowing thus the monitoring of completeness of reception of all service messages. The three 'nnn' will contain the file segment number (i.e. '001' if the service message is not divided into several files).

13) Example: An encryption key message file containing public keys for the user stations from 255 to 283 will have a product ID(2) of '00FF\_011B'.

- **Flags**

The Flags field will consist of 2 ASCII characters.

The first character identifies whether the LRIT/HRIT data field contains compressed or uncompressed data:

'C' identifies compressed data

'\_' identifies uncompressed data

The second character identifies whether the LRIT/HRIT data field contains encrypted or unencrypted data:

'E' identifies encrypted data

'\_' identifies unencrypted data

- **FieldSeparator**

The FieldSeparator consists of the single ASCII character '-' ('2D'h).

Notes to Table 4-8:

14) The defined characterstrings for the ProductIDs (1) – (4) will be left aligned. Any remaining space in the field will be filled with ASCII characters '\_' ('5F'h).

15) The ProductIDs will separated from each other by the ASCII character '-' ('2D'h).

16) All times used in ProductID (4) have the notation "YYYYMMDDhhmm"

"acquisition start time" will identical to TrueRepeatCycleStartTime as defined in [RD.5] but rounded to a full minute

“acquisition time of the oldest ... contained” represents the entry time of a DCP message or GTS message into the Dissemination Element rounded to a full minute

“nominal product time” represents the NominalTime contained in MPEF\_Product\_Header or NominalSGSProductHeader in SGS\_Common\_Header rounded to a full minute.

“nominal dissemination time” represents the scheduled service message time

- 17) The file type column is included for information and cross-reference only. The numbering is in accordance with Table 4-1.
- 18) The APID column is included for information and cross-reference only. The numbering is in accordance with Table 6-1.
- 19) The S/C name are identical to GP\_SC\_Name as defined in [RD.1].
- 20) The spectral channel names consist of the central wavelength and the orbital position of the satellite in the form XX\_X\_YYYY where:  
 XX\_X are 4 digits identifying the central wavelength (e.g. ‘01\_6’ for 1.6  $\mu\text{m}$ , ‘10\_8’ for 10.8  $\mu\text{m}$ , ‘\_\_\_\_’ for wideband channels) and  
 YYYY are 4 digits identifying the orbital position (e.g. 075W for CGMS 075W or 140E for CGMS 140E)
- 21) The file segment number contains Segm\_Seq\_No of header type #128 in a 6 digit representation.
- 22) The sequence number the files sequentially starting from ‘0’ at midnight.
- 23) The product names are as defined in Appendices A.5.1 and A.6.1.
- 24) The service message names of image-type, overlay-type and alphanumeric-type of representation will be defined during their preparation in the DADF offline environment. The following categories are currently foreseen:

‘ADMIN’	for administrative messages (service details for the previous calendar day)
‘ALGORITHM’	for algorithm updates
‘COMP_TST1’	for compression test messages
‘COMP_TST2’	for compression test messages
‘EKM’	for encryption key messages
‘ENC_TEST’	for encryption test messages
‘NEWS’	for newsletters (real-time problems, fixing of problems)
‘OVL_CL’	for coast lines
‘OVL_GRID’	for grid overlays
‘OVL_PB’	for political boundaries
‘REF-TEST’	for reference test messages
‘REG-RPT’	for regular reports (scheduled announcements for the following week)
‘TIMTABNNN’	for differences to ‘TIMTABNOM’
‘TIMTABNOM’	for details about the dissemination scheme (products per channel ...)
‘USER-DATA’	for overlay files not of type ‘OVL-xxx’

25) The sequence/version number has the structure:

sssss\_nnn

where ‘sssss’ is a five digit sequence number, incremented by one for each distinct service message, allowing thus the monitoring of completeness of reception of all service messages. The three ‘nnn’ will contain the file segment number (i.e. ‘001’ if the service message is not divided into several files).

26) Example: An encryption key message file containing public keys for the user stations from 255 to 283 will have a product ID(2) of ‘00FF\_011B’.

- **Flags**

The Flags field will consist of 2 ASCII characters.

The first character identifies whether the LRIT/HRIT data field contains compressed or uncompressed data:

'C' identifies compressed data

'\_' identifies uncompressed data

The second character identifies whether the LRIT/HRIT data field contains encrypted or unencrypted data:

'E' identifies encrypted data

'\_' identifies unencrypted data

- **FieldSeparator**

The FieldSeparator consists of the single ASCII character ‘-’ (‘2D’h).

#### **4.3.2.6 Header Type #5 - Time Stamp**

In the original HRIT/LRIT scheme of direct dissemination via an MSG S/C, the time stamp record was written by the dissemination element after the end of the session layer processing (i.e. after compression and encryption processing). Therefore, it would provide the end-user with the visibility of the data delay introduced by the dissemination sequencing scheme, the subsequent lower layer formatting and communication link delays to the PGS.

With the baseline dissemination scheme being EUMETCast (after the failure of a power amplifier used for dissemination onboard MSG-1), this delay information became obsolete. Furthermore, it was considered to be beneficial to provide an image repeat cycle timing information additional to the one contained in the Annotation Header (Type #4) and in the segment file names. Both are specifying the start of the respective repeat cycle.

The additional timing information is the end of repeat cycle time: using the parameter “PlannedRepeatCycleEnd” received from the MSG IMPF, in the format TIME CDS

EXPANDED. The time stamp used in XRIT Header type #5 is in the format TIME CDS SHORT, thus the nanosecond part of the parameter TIME CDS EXPANDED is cut off.

The (unchanged) structure of the time stamp record is defined as:

Title: Time Stamp Record	Id: TIME_STAMP		
TIME_STAMP	::=	RECORD	
{ Header_Type		UNSIGNED BYTE (5)	-- fixed value
Header_Record_Length		UNSIGNED SHORT (10)	-- fixed value
CDS_P_Field		UNSIGNED BYTE (64)	-- P-Field fixed value
			according to CCSDS
			bit 0 (MSB) = '0'b
			bits 1-3 = '100'b
			bits 4-7 = '0000'b
CDS_T_Field		TIME CDS SHORT	-- 6 octets T-field according to CCSDS
}			

**Table 4-9 – Time Stamp**

### Explanations

- CDS\_T\_Field

As defined by the CDS\_P\_Field, the 6 octets CDS\_T\_Field consists of

2 bytes	counter of days starting from 1 January 1958
4 bytes	milliseconds of day

### 4.3.2.7 Header Type #6 - Ancillary Text

The structure of the ancillary text record is defined as:

Title: Ancillary Text Record	Id: ANCILLARY_TEXT		
ANCILLARY_TEXT	::=	RECORD	
{ Header_Type		UNSIGNED BYTE (6)	-- fixed value
Header_Record_Length		UNSIGNED SHORT ()	-- variable value, max. 65532
Ancillary_Text		CHARACTERSTRING SIZE ()	-- plain text
}			

**Table 4-10 – Ancillary Text**

### Explanations

- Ancillary\_Text

The Ancillary\_Text will contain descriptive text identifying the contents of the LRIT/HRIT file in cases where the other headers are not conclusive. As identified in Table 4-14 the Ancillary Text Record will only be used for Service Message type files (e.g. overlay files, test messages, alphanumeric messages).

### 4.3.2.8 Header Type #7 - Key Header

The structure of the key header record is defined as:

Title:Key Header Record	Id: IMAGE DATA FUNCTION	
IMAGE DATA FUNCTION ::=	RECORD	
{ Header_Type	UNSIGNED BYTE (7)	-- fixed value
Header_Record_Length	UNSIGNED SHORT (12)	-- fixed value
Key_Number	UNSIGNED BYTE	-- index of the used MGK for the LRIT/HRIT file encryption
Seed	UNSIGNED DOUBLE	-- variable value
}		

*Table 4-11 – Key Header*

#### Explanations

- Key\_Number

The MSG dissemination will make use of the following key numbers:

64 -127      Key group 1

192 - 255      Key group 2

The reason for identifying two key groups is that the MSG encryption scheme will make use of a system whereby the actively used key group will be swapped from one to the other in regular intervals.

The remaining key numbers (0 - 63 and 128 - 191) are reserved for use by other EUMETSAT ground segments.

- Seed

The *Seed* is a random value generated by DADF used as a start value for the PN encryption pattern generation. For further details on the use of the Seed in the MSG encryption scheme the reader should refer to Sections 5.4.2 and 5.4.3.

### 4.3.2.9 Header Type #128 - Segment Identification

The use of this header type is mandatory for:

- all files of type #0 and
- files of type #1 originating from MPEF or SAFs and
- files of type #144 (binary file)

The structure of the segment identification record is defined as:

Title:Segment Identification Record	Id: SEGMENT_ID	
SEGMENT_ID ::=	RECORD	
{ Header_Type	UNSIGNED BYTE (128)	-- fixed value
Header_Record_Length	UNSIGNED SHORT (13)	-- fixed value
GP_SC_ID	ENUMERATED SHORT	
Spectral_Channel_ID	ENUMERATED BYTE	
Segm_Seq_No	UNSIGNED SHORT	-- segment sequence number
Planned_Start_Segm_Seq_No	UNSIGNED SHORT	-- planned start segment sequence number
Planned_End_Segm_Seq_No	UNSIGNED SHORT	-- planned end segment sequence

Data_Field_Representation	ENUMERATED BYTE {no specific formatting (0), JPEG interchange format (1), T.4 coded file format (2) Wavelet coded file format (3)}	number -- defines the representation of the LRIT/HRIT data filed
}		

**Table 4-12 – Segment Identification**

**Explanations**

- GP\_SC\_ID

GP\_SC\_ID will be identical to the definition given in [RD.1]. In the case the segmented data does not contain satellite imagery data, this record will be set to 0.

- Spectral\_Channel\_ID

For SEVIRI 1.5 data, the Spectral\_Channel\_ID record is identical to GP\_SC\_CHAN\_ID in [RD.1]. For FSD, this record will be set to values defined in Appendix A.1.4.

In the case the LRIT/HRIT data field does not contain single spectral channel information (e.g. meteorological products), this record will be set to 0.

- Segm\_Seq\_No

Segmentation is applicable to the following application data types:

- file type #0 (image data)
  - SEVIRI level 1.5
  - Foreign Satellite Data
  - Meteorological Products (imagery type)
  - Meteorological Products (overlay type)
- file type #1 (GTS Message)
  - Meteorological Products from MPEF
  - Meteorological Products from SAF
- file type #144 (binary file)
  - Meteorological Products (binary form)

Segm\_Seq\_No identifies the segment sequence number of the above-mentioned application data types.

For file type #0 (image data), Segm\_Seq\_No establishes a fixed relationship to the geographical location of the data contents as explained in Section 4.2.2. The precise location of the image segment can be derived from COFF/LOFF in the image navigation record #2.

For file type #1 (GTS messages), Segm\_Seq\_No does provide a continuous numbering starting from '1' without any geographical relationship.

For application data types falling into the above categories but no segmentation is applied (e.g. for a compression test image which is a single compressed image segment file), Segm\_Seq\_No will be set to '1'.

- **Planned\_Start\_Segm\_Seq\_No**

This parameter represents the planned number of the first image/product segment to be disseminated based on the knowledge at the start of the repeat cycle. The value of this parameter figure will be kept stable until the end of the repeat cycle's dissemination independent of the actual segment numbers being disseminated.

For application data types falling into the above categories but no segmentation is applied (e.g. for a compression test image which is a single compressed image segment file), **Planned\_Start\_Segm\_Seq\_No** will be set to '1'.

- **Planned\_End\_Segm\_Seq\_No**

This parameter represents the planned number of the last image/product segment to be disseminated based on the knowledge at the start of the repeat cycle. The value of this parameter figure will be kept stable until the end of the repeat cycle's dissemination independent of the actual segment numbers being disseminated.

For application data types falling into the above categories but no segmentation is applied (e.g. for a compression test image which is a single compressed image segment file), **Planned\_End\_Segm\_Seq\_No** will be set to '1'.

- **Data\_Field\_Representation**

In the case data compression is applied to the data field of file type #0, **Compression\_Flag** in header type #1 (see sect. 4.3.2.2) will be used to identify the compression method.

If the **Compression\_Flag** is set to a non-zero value **Data\_Field\_Representation** is used to define more precisely the compression algorithm applied to data field of the LRIT/HRIT file.

For file type #1 (GTS messages) this field will always be set to '0'.

- No specific formatting (0)
- JPEG Interchange Format (1)
- The JPEG interchange format as defined in [RD.11] will be used as baseline to support lossy compression for imagery type of data with 2 or more bits per pixel. The application of either the lossless (option) or lossy (baseline) method is defined by the **Compression\_Flag** in Header #1. Further compression parameters are contained in the data field of the LRIT/HRIT file as defined in Section 5.3.2.
- T.4 coded file format (2)
- T.4 coding can only be used for the dissemination of 1 bit/pixel images (e.g. overlay type of images or service messages). For more information on the implementation of T.4 coding the reader shall refer to Section 5.3.2.3 and [RD.4].
- Wavelet Interchange Format (3)
- The Wavelet interchange format as defined in Section 5.3.2.1 will be used as baseline to support lossless compression for imagery type of data with 2 or more bits per pixel. The application of either the lossless (baseline) or lossy (option) method is defined by the **Compression\_Flag** in Header #1. Further compression

parameters are contained in the data field of the LRIT/HRIT file as defined in Section 5.3.2.

#### 4.3.2.10 Header Type #129 - Image Segment Line Quality

Title: Image Segment Line	Id: LINE_QUALITY	Area: LRIT/HRIT
<b>Quality Record</b>		
LINE_QUALITY ::= RECORD		
{ Header_Type	UNSIGNED BYTE (129)	--fixed value
Header_Record_Length	UNSIGNED SHORT ()	--variable value
Line_Quality_Entries		
ARRAY SIZE (1..NL) OF RECORD		-- NL as defined in header type #1
{ Line_Number_in_Grid	INTEGER	
Line_Mean_Acquisition	TIME CDS SHORT	
Line_Validity	ENUMERATED BYTE	
	{ Not derived (0)	
	Nominal (1),	
	Based on missing data (2),	
	Based on corrupted data (3),	
	Based on replaced or interpolated data (4) },	
Line_Radiometric_Quality	ENUMERATED BYTE	
	{ Not derived (0)	
	Nominal (1),	
	Usable (2),	
	Suspect (3),	
	Do not use (4) },	
Line_Geometric_Quality	ENUMERATED BYTE	
	{ Not derived (0)	
	Nominal (1),	
	Usable (2),	
	Suspect (3),	
	Do not use (4) }	
}		

**Table 4-13 – Image Segment Line Quality**

#### Explanations

- Line\_Quality\_Entries

These records reflect the information received from the IMPF, MPEF or the SAF/SGS w.r.t. the line quality.

**LineNumberinGrid** represents the line number in the grid.

**LineMeanAcquisitionTime** represents the mean acquisition time of the line. It is set to zero for the CLAI and CTH products.

**LineValidity** qualifies the line validity. It is set to zero for the CLAI and CTH products.

**LineRadiometricQuality** qualifies the line radiometric quality. It is set to zero for the CLAI and CTH products.

**LineGeometricQuality** qualifies the line geometric quality. It is set to zero for CLAI and CTH products.

### 4.3.3 File Type vs. Header Implementation

The global mandatory/optional use of LRIT/HRIT header records is specified in [AD.1]. Table 4-14 defines the MSG mission specific use of header record types within certain LRIT/HRIT file types.

‘MSG mandatory use’ means that the identified header record will always be used in the MSG LRIT/HRIT dissemination. ‘MSG optional use’ means that only certain LRIT/HRIT files contain such header record.

↓ file types	→ header record types									
	0	1	2	3	4	5	6	7	128	129
0 image data files										
- SEVIRI Level 1.5	G	G	X		X	X		(X)	X	X
- FSD	G	G	X	(X)	X	X		(X)	X	(X)
- MPEF Met. Products (imagery + overlay type)	G	G	X	(X)	X	X		(X)	X	(X)
- overlay files as service message	G	G	X	(X)	X	X	X	(X)	X	
- encryption test messages	G	G	(X)		X	X	X	X	X	
- compression test messages	G	G	(X)		X	X	X		X	
1 GTS message										
- MPEF Met. Products	G				X	X		(X)	X	
- SAF Met. Products	G				X	X		(X)	X	
- GTS re-transmission	G				X	X		(X)		
- DCP messages coded as GTS bulletins	G				X	X		(X)		
2 alphanumeric text file	G				X	X	X	(X)		
3 encryption key message	G				X	X				
128 image repeat cycle prologue	G				X	X		(X)		
129 image repeat cycle epilogue	G				X	X		(X)		
130 DCP message	G				X	X		(X)		
144 binary file	G				X	X		(X)	X	

Remarks:

- G as required by [AD.1]
- X MSG mandatory use
- (X) MSG optional use
- 0 primary header
- 1 image structure
- 2 image navigation
- 3 image data function
- 4 annotation
- 5 time stamp
- 6 ancillary text
- 7 key header
- 128 segment identification
- 129 image segment line quality record

**Table 4-14 – Use of Header Records vs. File Type**

## 5 SESSION LAYER

### 5.1 General

The session layer provides the means for interchange of data. In the MSG context, this layer includes the definition of **data compression** and **encryption**. As part of the compression algorithm, the session layer offers a mechanism to include synchronisation points in the data stream.

All data which are not subject to compression or encryption will pass the relevant data processing functionality unmodified, and will receive its primary header record completed with the known data field length and time stamped.

In the case the data field of the LRIT/HRIT files is compressed and/or encrypted, the session layer functionality will modify or insert the following headers/header fields after processing:

For compression:

header type #0, Data_Field_Length	to be modified	(sect. 4.3.2.1, Primary Header)
header type #1, Compression_Flag	to be modified	(sect. 4.3.2.2, Image Structure)
header type #4, Flags field	to be modified	(sect. 4.3.2.5, Annotation)
header type #128, Data_Representation	to be modified	(sect. 4.3.2.9, Segment Identification)

For encryption:

header type #4, Flags field	to be modified	(sect. 4.3.2.5, Annotation)
header type #7	to be inserted	(sect. 4.3.2.8, Key Header)

### 5.2 Input to Session Layer

The LRIT/HRIT files as shown in Figure 4-1 are the input to the Session Layer processing.

### 5.3 Compression

#### 5.3.1 General

Compression is required to allow for the transmission of a full data set via the HRIT channel and to maximise the data available in the LRIT channel.

The Wavelet Transform (WT) Compression has been chosen as the compression baseline to support the lossless scheme for the MSG dissemination service.

The ISO standard 10918 ‘Digital compression and coding of continuous-tone still images’ [RD.11] known as JPEG has been chosen as the compression baseline to support the lossy schemes for the MSG dissemination service. JPEG lossless compression has been implemented in the MSG Ground Segment for optional use.

Images containing only 1bit/pixel cannot use JPEG. T.4 coding in accordance with [RD.4] has been chosen to support these image types.

The selection of the compression method and related configuration parameters for the MSG dissemination service will be based on a schedule and will allow to adjust the LRIT/HRIT

channel occupation according to end-user requirements. As a minimum, the selected compression method and related configuration parameters will be kept stable over the period of one image repeat cycle.

Due to the dynamic behaviour of satellite images depending on spectral channel, diurnal and seasonal variations, different quantisation and coding tables may have to be changed operationally. Compressed LRIT/HRIT files will always be self-describing, i.e., they will include all relevant information for the decoding process.

Data compression will be able to operate on single LRIT/HRIT files. If the compression flag (CFLG) of the image structure record (header type #1) is set to 1 or 2, further information about used compression algorithm and mode will be made available via the image segment identification (header type #128).

Compression will only be applied to file type #0 (image data).

Other file types may contain compressed data due to coding which was applied by entities external to the DADF, e.g. T.4 coded or binary GTS messages.

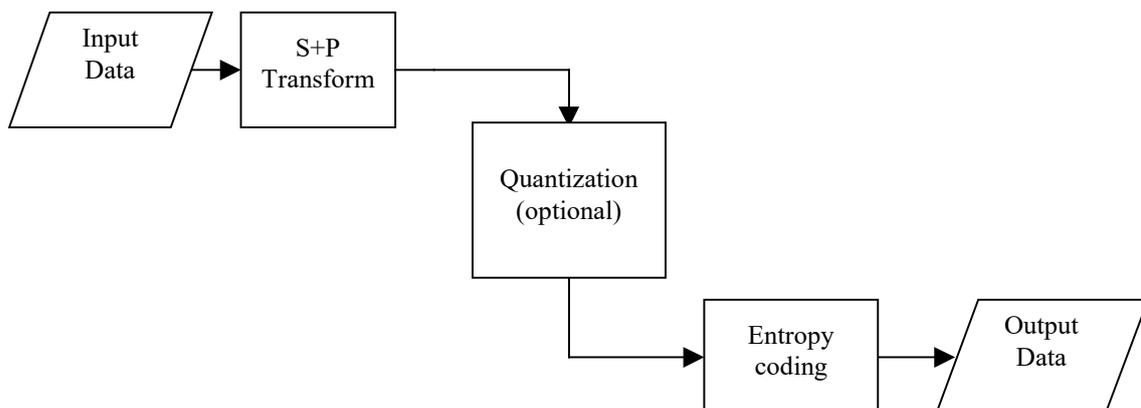
Compression will operate on the LRIT/HRIT file data field only. After compression, the LRIT/HRIT file will contain a compressed data field as depicted in Figure 5-1.



**Figure 5-1 – LRIT/HRIT File Structure with compressed data field**

### 5.3.1.1 Introduction to WT Compression

The WT (Wavelet Transform) compression handles any bit depth, smaller or equal to 16-bit per pixel. The input image is first transformed in a scale-space representation (the S-transform), followed by a prediction stage (P) to further remove eventual remaining redundancies. Several iterations of the S+P



**Figure 5-2 – WT compression scheme**

transform may be applied recursively in order to get a finer scale-space representation. The obtained coefficients are called the “Wavelet Transform coefficients”. If a lossy compression is to be performed, the Wavelet Transform coefficients are quantised. The resulting coefficients are then entropy coded using an algorithm similar to the JPEG Variable Length Coding (VLC) followed by Arithmetic Coding (AC). Figure 5-2 shows the principle of the Wavelet compression. The algorithm is further specified in Section 5.3.2.1.

### 5.3.1.2 Introduction to Lossy JPEG Compression

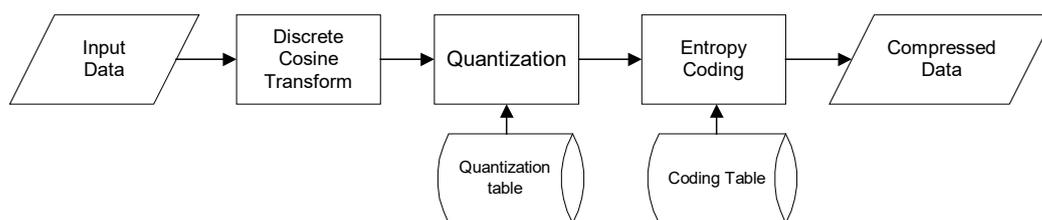
The JPEG lossy compression scheme supports 8-bit or 12-bit pixel resolution. Input to the lossy JPEG coder are data arrays sized 8x8 pixel. In a first step, a discrete cosine transform (DCT) turns each data array of intensity data into an array of frequency data. A following quantisation process sets the precision to which each of the frequency data values from the DCT are stored. The two-dimensional quantised DCT coefficients are then converted to a serial bit stream according to a zigzag algorithm and the results are entropy coded (compressed).

The following lossy DCT-based modes of compression exist:

- Baseline process (only 8-bit, only Huffman coding, only sequential scan)
- Extended process (8-bit or 12-bit, Huffman or arithmetic coding, sequential and progressive scans)
- Hierarchical process

Note: The modes that will be supported by MSG DADF are listed in Section 5.3.2.2.

Figure 5-3 shows the principle of lossy compression. For further information on the lossy compression algorithm, the reader shall refer to [RD.11].



**Figure 5-3 – Lossy JPEG Compression Scheme**

### 5.3.1.3 Introduction to Lossless JPEG Compression

This section provides a short introduction to the JPEG lossless compression.

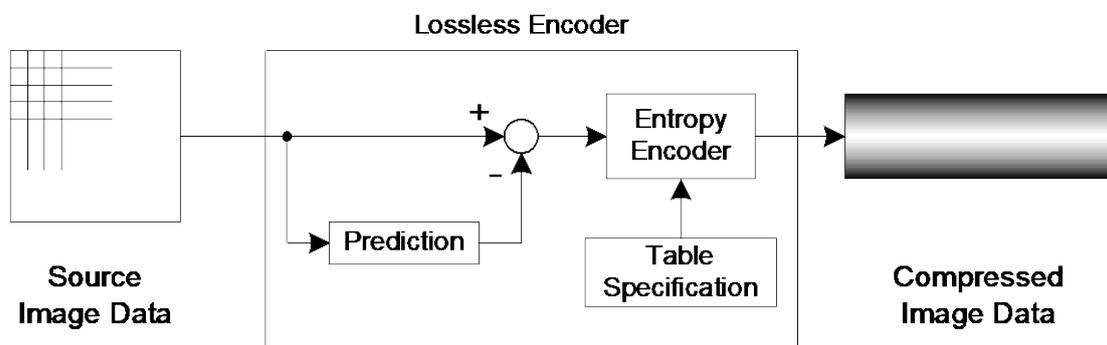
The lossless JPEG scheme is based on a prediction of the pixel value. The lossless JPEG encoder supports input precision of 2...16 bits per sample. A set of predictors is defined. The difference between the real pixel value and its predicted value forms the input to TBC is coded as a prediction and the difference of the pixel value to that prediction. No blocking

structure of the image data and no DCT-based encoding is used in the lossless compression mode.

Two lossless modes of compression are possible:

- Lossless process with Huffman coding
- Lossless process with arithmetic coding (not used by the MSG DADF)

Figure 5-4 shows the principle of lossless compression. For further information on the lossless compression algorithm, the reader shall refer to [RD.11, annex H].



*Figure 5-4 – JPEG Lossless Image Compression Scheme*

### 5.3.1.4 Introduction to T.4 Coding

For an introduction to T.4 coding the reader shall refer to [RD.4].

### 5.3.2 MSG Mission Specific Implementation of Compression

This section is split into two subsections that deal with the mission specific implementation of:

1. ‘Wavelet Transform compression’ for images with more than 1 bit/pixel
2. ‘JPEG compression’ for images with more than 1 bit/pixel
3. ‘T.4 type of compression’ for images with 1 bit/pixel

#### 5.3.2.1 Description of WT Coding

This section defines the WT coding algorithm and its mission specific WT implementation. It includes the definition of the used compression processes, the coding applied and the detailed marker segments.

##### 5.3.2.1.1 WT Compression Process

###### 5.3.2.1.1.1 Reference Documents

See Section 1.3.2.

### 5.3.2.1.1.2 General Requirements

The system used to implement the WT compression shall be able to perform arithmetic and logic operations on 32-bit signed integers (4 bytes). Unless a bit size is explicitly specified, all constants, internal state variables and temporary variables of the WT compression/decompression process shall be stored on 32-bit signed integers (minimum required).

It is also assumed that negative integer numbers are represented using their two-complement.

### 5.3.2.1.1.3 General Definitions

#### 5.3.2.1.1.3.1 Abbreviations and Symbols

<b>AASM(QQCD, CC)</b>	2D array of ASM indexed by QQCD and CC
<b>AC</b>	Arithmetic coding module
<b>ACBF</b>	Number of opposite bits to output with next bit
<b>ACLOW</b>	Lower bound of the arithmetic coding interval
<b>ACRANGE</b>	Arithmetic coding interval size
<b>ACRNB</b>	Number of bits used to represent ACRANGE/ADRANGE (31 bits)
<b>ACRTH</b>	ACRANGE/ADRANGE lower threshold
<b>AD</b>	Arithmetic decoding module
<b>ADRANGE</b>	Arithmetic decoding interval size
<b>ADVALUE</b>	Arithmetic decoding final interval value
<b>ALLPS</b>	All symbols except the most probable one (MPS) according to current ASM
<b>AND</b>	Bit-wise AND
<b>ASM</b>	Adaptive statistical model managing symbols probabilities, used with AC / AD
<b>BPP</b>	Input image pixel depth expressed in number of bits
<b>BS</b>	Bit state 0 or 1
<b>CC</b>	Conditional context when coding/decoding a symbol
<b>CMAV</b>	The maximum absolute value of the coefficients within a block
<b>CS</b>	Symbol to be entropy coded/decoded
<b>CUM(SI)</b>	Array of the current ASM holding cumulative frequency counts sorted by SI
<b>CUMTH</b>	Cumulative frequency threshold
<b>DPCM</b>	Difference Pulse Coded Modulation
<b>ECS</b>	Entropy coded segment
<b>FREQ(SI)</b>	Array of the current ASM holding the symbol frequency counts sorted by SI
<b>I2S(SI)</b>	Array of the current ASM holding the inverse of S2I
<b>IWT</b>	Inverse Wavelet transform
<b>LP</b>	Lossy parameter [0, 15]
<b>MPS</b>	Most probable symbol according to current ASM
<b>NBCMAV</b>	The number of significant bits needed to represent CMAV
<b>NBNBCMAV</b>	The number of significant bits needed to represent NBCMAV

<b>NBQCMVA</b>	The number of significant bits needed to represent QCMVA
<b>NBS</b>	Number of symbols managed by the current ASM
<b>PWTC</b>	Previous Wavelet transform coefficient, used with DPCM
<b>QQCD</b>	Quadrant quantised coefficients dynamic expressed in number of bits
<b>QCDS</b>	Quadrant coefficients dynamic shift expressed in number of bits
<b>QCMVA</b>	The maximum absolute value of the coefficients within a block quadrant
<b>Qmax</b>	Maximum quadrant number given W1
<b>QN</b>	Quadrant number
<b>S2I(CS)</b>	Array of the current ASM allowing symbol sorting according to frequency count
<b>SI</b>	Sorting index of the current symbol (CS)
<b>SLL</b>	Logical shift left
<b>SRL</b>	Logical shift right
<b>SWTC</b>	Shifted WTC according to QCDS
<b>VLBS</b>	Variable length bits sequence to be binary coded/decoded
<b>VLBSL</b>	Length in bits of the VLBS
<b>VLC</b>	Variable length coding module
<b>VLD</b>	Variable length decoding module
<b>W1</b>	Number of iterations of the Wavelet transform [3, 6]
<b>WT</b>	Wavelet transform
<b>WTC</b>	Wavelet transform coefficient

#### 5.3.2.1.1.3.2 Downwards Truncation

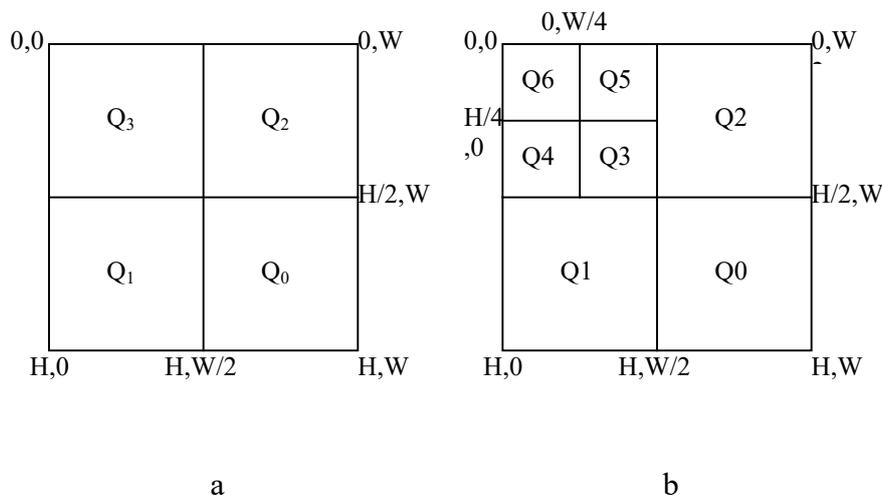
The downward truncation of a number  $x$  denoted by  $\lfloor x \rfloor$  is defined as the largest integer number smaller than  $x$ .

#### 5.3.2.1.1.3.3 Vector Indexing

If  $V$  is a vector of scalars,  $V_i$  is defined as the  $i^{\text{th}}$  element of  $V$ , where  $V_0$  is the left-most (in the case of a line vector) or top-most (in the case of a column vector) element.

#### 5.3.2.1.1.3.4 Coefficients Quadrants

If  $H$  and  $W$  respectively denote the number of lines and columns of a block of size  $H \times W$ , a quadrant is defined as a sub-block of size  $H/2 \times W/2$  composed of the  $W/2$  left or right-most samples of the  $H/2$  top or bottom-most lines of its parent block, as is illustrated in Figure 5-5 a. Quadrants can be nested in other quadrants as is illustrated in Figure 5-5 b where 4 new quadrants were defined in the last quadrant ( $Q_3$ ) of Figure 5-5 a.



**Figure 5-5 – Coefficients Quadrants**

The quadrants are numbered starting from zero, right to left, bottom to top, and recursively until all quadrants are numbered. This numbering scheme is illustrated in Figure 5-5 a and b. The number assigned to a quadrant is referred to as its index or quadrant number (QN).

Depending on the number of Wavelet transform iterations (WI), one then has the values of Qmax and QN range given in Table 5-1.

WI	QN	Qmax
3	0 to 9	9
4	0 to 12	12
5	0 to 15	15
6	0 to 18	18

**Table 5-1 – QN and Qmax as a Function of the Number of Wavelet Transform Iterations**

#### 5.3.2.1.1.4 Blocking

The processing is performed sequentially on a block-by-block basis, each block being coded/decoded and output before starting processing of the next block.

The following block sizes are supported: 16x16, 32x32, 64x64 and “whole image” (one single block). In any case, the blocks shall have a size that is an integer multiple of  $2^{WI}$  where WI is the number of resolution levels used in the Wavelet decomposition (number of iterations). All the blocks belonging to the same image frame shall have the same size. The Minimum Coded Unit (MCU) is defined as one block of contiguous samples.

The orientation of an image is as indicated on Figure A.1.b of [RD11].

The first block of the image is the top-left-most block, it contains the left-most samples of each of the top-most rows in the image. With this top-left-most block as the reference, the image is partitioned into contiguous MCUs to the right and to the bottom as shown in the left part of Figure A.4 of [RD11].

The order of the MCUs in a frame shall be left-to-right and top-to-bottom, as shown in Figure A.2 of [RD11].

If the size of the image is not an entire multiple of the block size, the encoding process shall extend the number of columns/rows as needed to complete the right-most columns/bottom-most rows of the block. The incomplete blocks are to be completed by replication of the right-most column and the bottom-most line of the image.

### 5.3.2.1.1.5 S+P Transform

One level of the 2D S+P transform is obtained first, by performing the 1D S+P transform (the scale-space transform (S) followed by the prediction (P)) on all the lines of the block. Then by performing the same 1D S+P transform on all the columns of the resulting coefficient block.

The block is supposed to be H lines high by W columns wide.

The transform process is recursive. The transformation is first applied to the whole block. It is then re-applied to the last quadrant ( $Q_3$ ) of the resulting coefficient block. If  $W_1$  denotes the number of Wavelet-transformed levels, the transform is recursively performed  $W_1$  times, hence the size of the smallest quadrant to which the transform was applied is  $H/2^{W_1-1}$  lines by  $W/2^{W_1-1}$  columns. Size constraints ensure the successive sub-blocks will always have even dimensions.

These operations shall be executed in integer arithmetic, and hence do not require any form of approximation.

#### 5.3.2.1.1.5.1 Wavelet Transform (S-transform)

The following equations specify the functional definition of the direct scale-space (S) transform for a 1D sequence of length N:

$$\begin{cases} T_n = \lfloor (O_{2n} + O_{2n+1}) / 2 \rfloor \\ T_{n+N/2} = O_{2n} - O_{2n+1} \end{cases}$$

for  $n = 0, \dots, N/2-1$  and where  $O_i$  and  $T_i$  denote the  $i^{\text{th}}$  pixel of the original respectively transformed 1D line/column.

The functional definition of the 1D inverse Wavelet (S) transform for a 1D sequence of length N is specified by the following equations

$$\begin{cases} O_{2n} = T_n + \lfloor (T_{n+N/2} + 1) / 2 \rfloor \\ O_{2n+1} = O_{2n} - T_{n+N/2} \end{cases}$$

for  $n=0, \dots, N/2-1$

The S transform increases the bit depth of the input coefficients by at most 1 bit per 1D operation, i.e. 2 bits for the 2-dimensional S transform (worst case). That bit increase only affects the second half of the transformed sequence ( $[T_{N/2}, T_{N-1}]$ ); the first half of the transformed sequence ( $[T_0, T_{N/2-1}]$ ) is not affected because it is just a low-pass filtered (mean) and sub-sampled version of the original sequence (bit depth remains equal to BPP or below).

Since the Wavelet transform is re-applied recursively to the first half of the transformed sequence only, the total bit increase due to the S transform, given any number of transform iterations, is still limited to 1 bit (2 bits for 2D S transform).

### 5.3.2.1.1.5.2 Prediction (P-transform)

If N denotes the length of the 1D sequence on which the prediction is to be performed, the prediction is only to be performed if  $N > 2$ .

The following equations specify the functional definition of the prediction process for a 1D sequence of length N:

$$\begin{cases} P_n = T_n & n = 0 \dots \frac{N}{2} - 1 \\ P_{N/2} = T_{N/2} - \lfloor \alpha_0 (T_0 - T_1) + \frac{1}{2} \rfloor \\ P_{1+N/2} = T_{1+N/2} - \lfloor \alpha_0 (T_0 - T_1) + \alpha_1 (T_1 - T_2) - \beta_1 T_{2+N/2} + \frac{1}{2} \rfloor \\ P_{n+N/2} = T_{n+N/2} - \lfloor \alpha_{-1} (T_{n-2} - T_{n-1}) + \alpha_0 (T_{n-1} - T_n) + \alpha_1 (T_n - T_{n+1}) - \beta_1 T_{n+1+N/2} + \frac{1}{2} \rfloor & n = 2 \dots \frac{N}{2} - 2 \\ P_{N-1} = T_{N-1} - \lfloor \alpha_0 (T_{N/2-2} - T_{N/2-1}) + \frac{1}{2} \rfloor \end{cases}$$

Where  $T_i$  and  $P_i$  denote the  $i^{\text{th}}$  pixel of the transformed respectively predicted 1D line/column. The values of the prediction coefficients,  $\alpha_j$  and  $\beta_1$ , are given in Table 5-2.

Predictor type	$\alpha_{-1}$	$\alpha_0$	$\alpha_1$	$\beta_1$
None	0	0	0	0
A	0	1/4	1/4	0
B	0	2/8	3/8	2/8
C	-1/16	4/16	8/16	6/16

**Table 5-2 – Predictor Coefficients as a Function of the Predictor Type**

The inverse prediction transform is given by:

$$\begin{cases} T_n = P_n & n = 0 \dots \frac{N}{2} - 1 \\ T_{N-1} = P_{N-1} + \lfloor \alpha_0 (T_{N/2-2} - T_{N/2-1}) + \frac{1}{2} \rfloor \\ T_{n+N/2} = P_{n+N/2} + \lfloor \alpha_{-1} (T_{n-2} - T_{n-1}) + \alpha_0 (T_{n-1} - T_n) + \alpha_1 (T_n - T_{n+1}) - \beta_1 T_{n+1+N/2} + \frac{1}{2} \rfloor & n = \frac{N}{2} - 2 \dots 2 \\ T_{1+N/2} = P_{1+N/2} + \lfloor \alpha_0 (T_0 - T_1) + \alpha_1 (T_1 - T_2) - \beta_1 T_{2+N/2} + \frac{1}{2} \rfloor \\ T_{N/2} = P_{N/2} + \lfloor \alpha_0 (T_0 - T_1) + \frac{1}{2} \rfloor \end{cases}$$

The prediction stage increases the bit depth of the input coefficients by at most 1 bit per 1D operation, i.e. 2 bits for the 2-dimensional prediction (worst case). That bit increase only affects the second half of the transformed sequence ( $[P_{N/2}, P_{N-1}]$ ); the first half of the transformed sequence ( $[P_0, P_{N/2-1}]$ ) remains unchanged ( $P_n = T_n$ ).

Since the Wavelet transform is re-applied recursively to the first half of the transformed sequence only, the total bit increase due to the prediction the stage, given any number of transform iterations, is still limited to 1 bit (2 bits for 2D prediction).

By combining the effects of the S transform and of the prediction stage, one can deduce that the maximum number of bits needed to store the 2D S+P transformed coefficients shall be limited to  $BPP + 4$ .

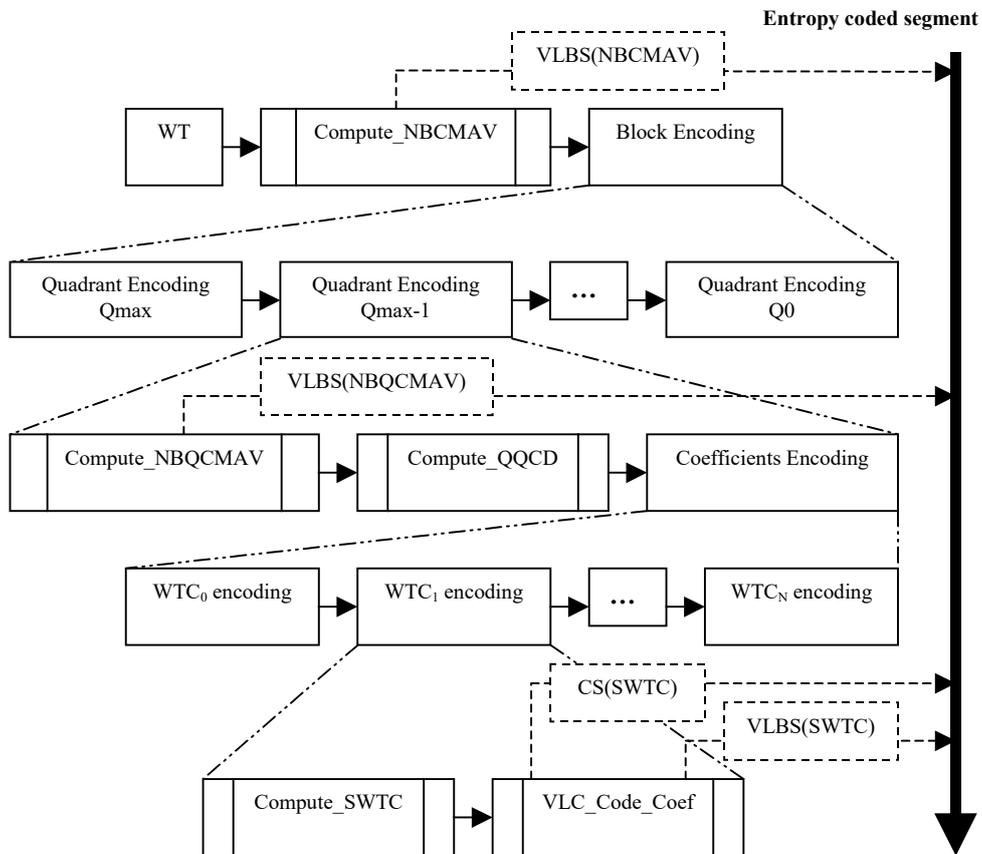
#### **5.3.2.1.1.6 Coding the Block**

As stated in Section 5.3.2.1.1.4, the processing is performed sequentially on a block-by-block basis.

First the forward Wavelet transform (WT) is applied on the block of pixels, producing a block of Wavelet transform coefficients (WTC). Before coding the block, the information about the WTC dynamic (NBCMAV) is computed and binary transmitted to the output stream (entropy coded segment, ECS).

The coding of the block is then sub-divided into the coding of each quadrant of the block (starting from Qmax and ending at Q0). The information about the quadrant WTC dynamic (NBQCMVA) is computed and binary transmitted to the output stream. Then according to the lossy parameter (LP), a quantisation process is applied producing shifted Wavelet transform coefficient (SWTC). Finally, the Variable Length Coding (VLC) module together with the Arithmetic Coding (AC) module is used to entropy code the SWTC and to feed the output stream.

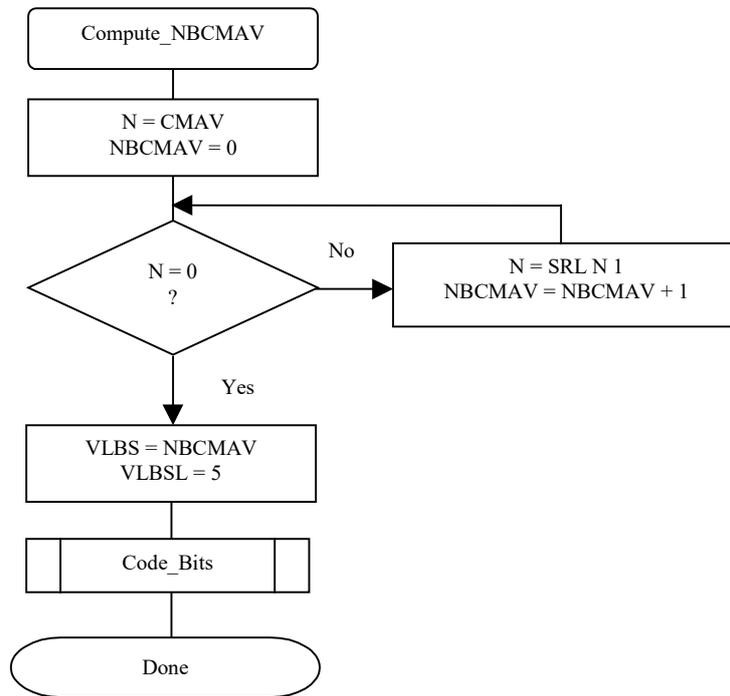
The encoding flow can be schematised as follow:



**Figure 5-6 – Encoding flow**

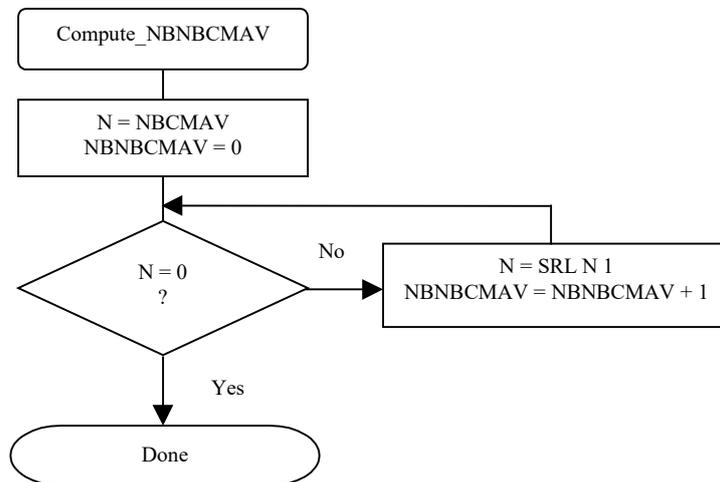
**5.3.2.1.1.6.1 Number of Bits Needed to Code the Coefficients**

The maximum absolute value of the coefficients within the block (CMAV) is searched for and the number of significant bits (NBCMAV) needed to represent that maximum is computed. NBCMAV shall be passed to AC module for binary coding as a sequence of 5 bits.



**Figure 5-7 – NBCMAV computation procedure**

The number of significant bits needed to represent NBCMAV is computed; that number is stored in NBNBCMAV:



**Figure 5-8 – NBNBCMAV computation procedure**

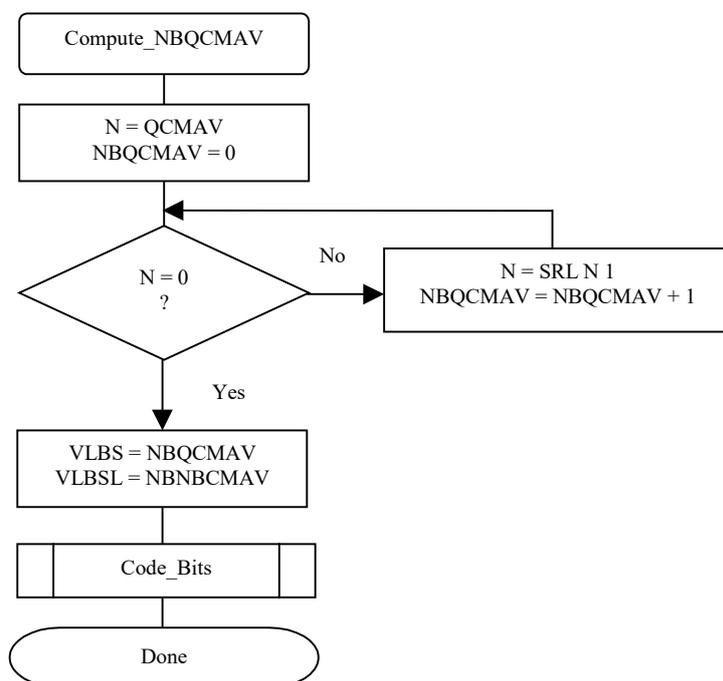
The Wavelet transform process (S transform and prediction stage) increases the bit depth of the produced coefficients by at most 1 bit per 1D operation, i.e. 2 bits for the S transform and 2 bits for the prediction stage (worst case). Since the images to be coded may have a pixel depth (BPP) ranging from 1bpp to 16bpp, NBCMAV shall be in the [0, 20] range (thus need 5 bits to be binary coded). NBNBCMAV shall be in the [0, 5] range (number of significant bits needed to represent NBQCMVA, see Section 5.3.2.1.1.6.3).

### 5.3.2.1.1.6.2 Quadrant Scanning Order

Quadrants are processed in decreasing QN order, i.e. beginning with the top-most; left-most quadrant (DC quadrant,  $QN=Q_{max}$ ) and ending with the bottom-most, right-most quadrant ( $QN=0$ ).

### 5.3.2.1.1.6.3 Number of Bits Needed to Code the Quadrant Coefficients

The maximum absolute value of the coefficients within the quadrant (QCMVA) is searched for and the number of significant bits (NBQCMVA) needed to represent that maximum is computed. NBQCMVA shall be passed to AC module for binary coding as a sequence of NBNBCMAV bits. Only quadrants having a non-zero NBQCMVA shall be coded.



**Figure 5-9 – NBQCMVA computation procedure**

NBQCMVA shall be in the  $[0, NBCMAV]$  range.

### 5.3.2.1.1.6.4 Coefficients Quantisation

In the case of lossy compression, the Wavelet transform coefficients (WTC) are quantised according to the lossy parameter (LP), to the quadrant (QN) the WTC belong to, and to the number of Wavelet transform iterations (WI).

The quantisation is performed by a right shift (division by a power of two). All coefficients belonging to the same quadrant are affected by the same shift, defined as the quadrant coefficients dynamic shift (QCDS).

#### 5.3.2.1.1.6.4.1 Computation of the Quadrant Quantised Coefficients Dynamic

The quadrant coefficients dynamic shift (QCDS) is the number of least significant bit planes that shall not be coded (because of the right shift). Its value is quadrant dependant (QN) and shall be derived using the following tables:

QN/LP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WI=3 0	0	1	2	2	2	3	3	3	3	3	4	4	4	4	4	4
1	0	0	0	1	1	1	2	2	2	2	2	3	3	3	3	3
2	0	0	0	1	1	1	2	2	2	2	2	3	3	3	3	3
3	0	0	0	0	1	1	1	2	2	2	2	2	3	3	3	3
4	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	2
5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	2
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

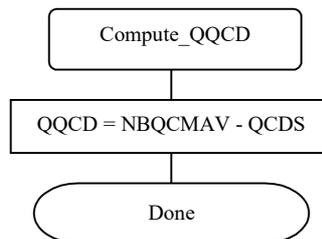
  

WI=4, 5 or 6	0...8	Same as WI=3															
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	10...19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 5-3 – QCDS Tables**

In the case of lossless coding (LP=0), the QCDS is zero for all quadrants. QCDS is always zero for the DC quadrant (QN=Qmax) as well as for all quadrants above Q9.

The QQCD represents the quantised WT coefficients dynamic expressed in significant number of bits. Using the QCDS, the QQCD is computed as follows:



**Figure 5-10 – QQCD Computation Procedure**

QQCD shall be used to select the appropriate statistical model family (see Section 5.3.2.1.1.6.5.2.3). Quadrants having a QQCD equal to zero shall not be coded.

#### 5.3.2.1.1.6.4.2 Computation of the Shifted WT Coefficients

For all quadrants except the DC quadrant (Qmax), the shifted WT coefficients (SWTC) are computed from the WT coefficients (WTC) by applying the quadrant coefficients dynamic shift (QCDS) as indicated on Figure 5-10.

This operation acts like a coefficient quantisation using integer division by power of two ( $2^{QCDS}$ ). After the WT, the WTC absolute value shall be in the  $[0, 2^{NBQCMAV}]$  range. After computation, the SWTC absolute value shall be in the  $[0, 2^{QQCD}]$  range.

Given the possible range for the WI and LP parameters, one can deduce that the DC quadrant (QN=Qmax) coefficients are always coded without any loss (QCDS is always 0, see Table 5-3).

Also, due to the definition of the WT, all coefficients of the DC quadrant are positive (sub-sampled mean) and have roughly the same distribution as the original pixels of the image ( $NBQCMAV \leq BPP$ ).

Before VLC coding the DC quadrant coefficients, a DPCM operation is applied in order to prepare the coefficients for the entropy coding. The difference between the current coefficient (WTC) and the previous one (PWTC) replaces each coefficient. For the first coefficient, the PWTC value shall be equal to  $2^{QQCD-1}$ . The scanning of the coefficient during DPCM is the same as during VLC coding (see Section 5.3.2.1.1.6.5.1).

Since the difference operation may increase the coefficient dynamic by one bit, the QQCD is increased by one after the DPCM phase. The coefficients of the DC quadrant have roughly the same bit depth as the original image ( $NBQCMAV \leq BPP$ ). After the DPCM phase, the DC quadrant coefficients are thus still below the limit fixed by the S+P transform ( $BPP+4$ , see Section 5.3.2.1.1.5)

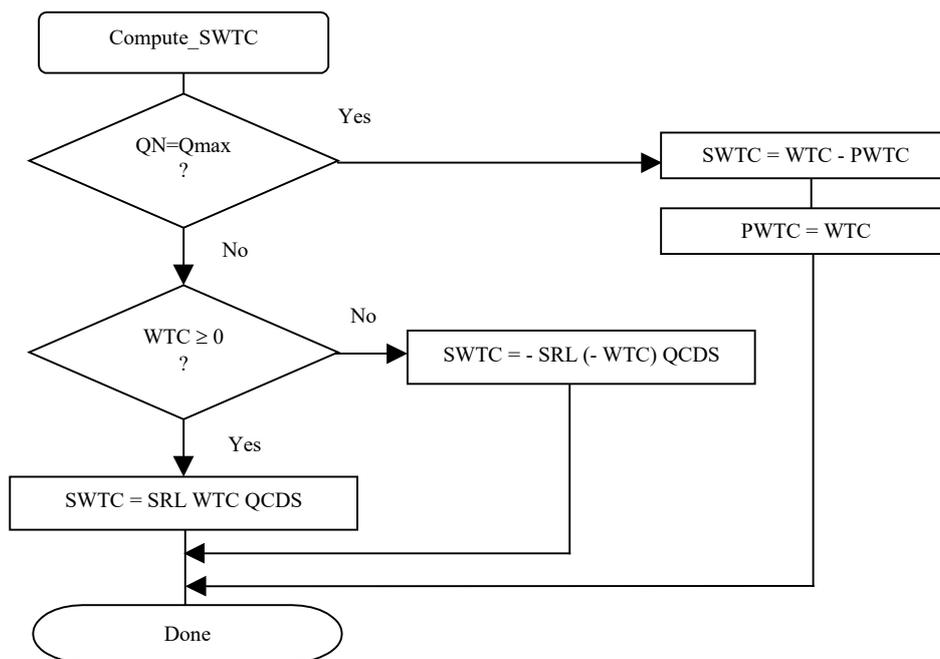


Figure 5-11 – SWTC Computation Procedure

NOTE: a non-null SWTC shall keep the sign of its WTC parent (QN≠Qmax only).

**5.3.2.1.1.6.5 Variable Length Coding**

The Variable Length Coding (VLC) module of the coder takes as input a block of shifted wavelet transform integer coefficients (SWTC) and produces a stream of symbols and bits sequences to be transferred to the Arithmetic Coding (AC) module.

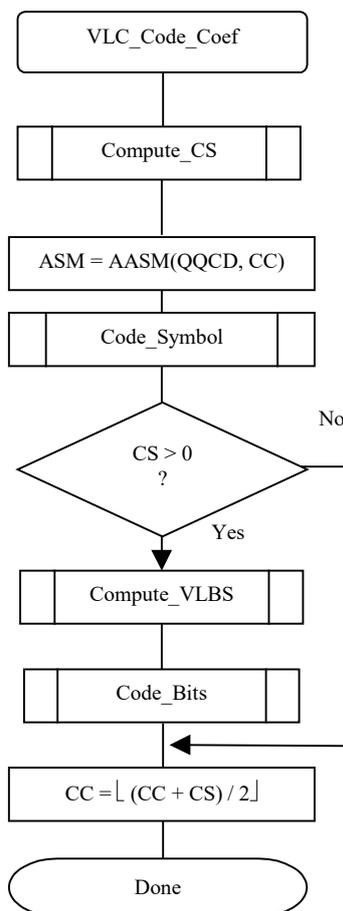
**5.3.2.1.1.6.5.1 Coefficients Scanning Within a Quadrant**

A quadrant can be seen as a two-dimensional array of H lines with W coefficients each. Line indexes range from 0 to H-1 and column indexes range from 0 to W-1.

Within a quadrant the lines are scanned from the top-most to the bottom-most line. For even index lines a left to right columns scanning is performed and for odd index lines a right to left columns scanning is done. The coefficients scanning thus starts from the top-most left-most coefficient and ends at the bottom-most, left-most coefficient if H is odd; or it ends at the bottom-most right-most coefficient if H is even.

**5.3.2.1.1.6.5.2 VLC Coding of a Coefficient**

From the SWTC, a symbol (CS) and a variable length bits sequence (VLBS) are derived and transmitted to the AC module for entropy and binary coding respectively. The VLC coding of a coefficient can be schematised as follows:



**Figure 5-12 – Coefficient VLC Coding Procedure**

The entropy coding of the derived symbol (CS) shall be performed using a context sensitive adaptive statistical model (ASM). The context is made of the quadrant quantised coefficient dynamic (QQCD, global context for all coefficients of the quadrant) and of a local context (CC). The local context (CC) for the current symbol is equal to the mean of {CC, CS} of the

previously coded coefficient in the same quadrant. When coding the first coefficient of a quadrant CC shall be set to QQCD.

#### 5.3.2.1.1.6.5.2.1 Computation of the Coefficient Symbol

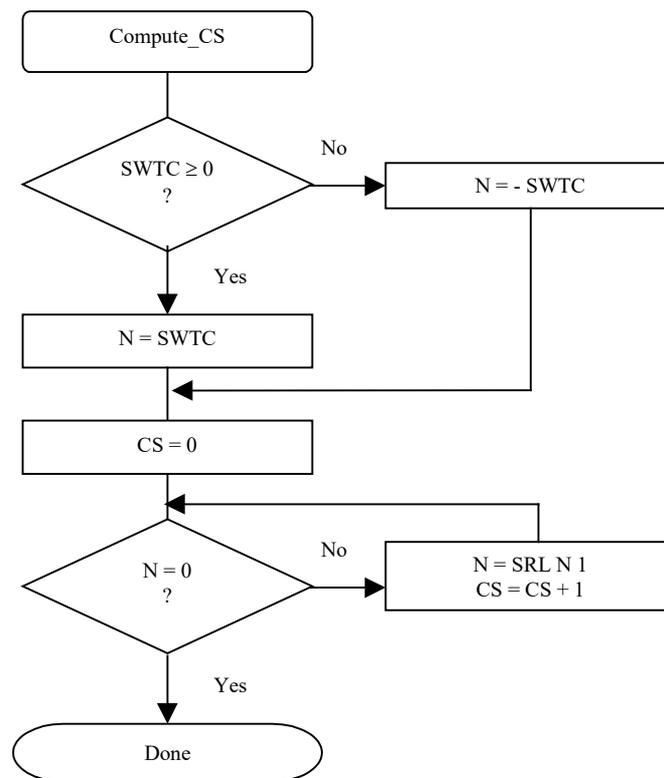
The coefficient symbol (CS) value is defined as the number of significant bits needed to represent in binary the absolute value of the SWTC; it shall be in the  $[0, QQCD]$  range and shall be passed to the AC module for entropy coding (Code\_Symbol procedure of the AC module). CS carries information about the magnitude of the SWTC. From CS, the SWTC magnitude is known to a limited precision: fine precision for low SWTCs (most probable coefficients) and coarse for high ones (less probable coefficients). The SWTC to CS operation allows grouping the coefficients into QQCD+1 classes.

CS	SWTC ranges
0	0
1	$\pm 1$
2	$[-3, -2], [2, 3]$
3	$[-7, -7], [4, 7]$
4	$[-15, -8], [8, 15]$
5	$[-31, -16], [16, 31]$
6	$[-63, -32], [32, 63]$
7	$[-127, -64], [64, 127]$
8	$[-255, -128], [128, 255]$
...	...
QQCD	$[-2^{QQCD-1}, -2^{QQCD-1}], [2^{QQCD-1}, 2^{QQCD-1}]$

**Table 5-4 – SWTC-to-CS Classes**

The information relative to the eventual sign of the SWTC and its position within the class shall be carried by the variable length binary sequence (see Section 5.3.2.1.1.6.5.2.2).

From SWTC, CS can be computed as follows:



**Figure 5-13 – CS Computation Procedure**

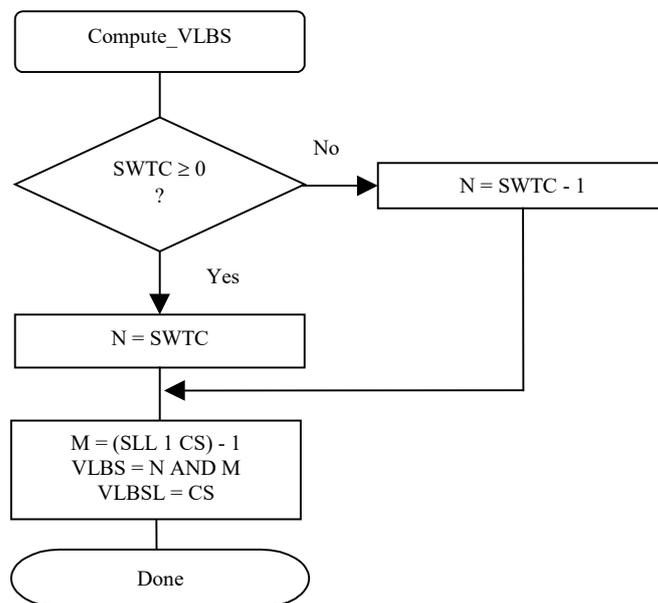
**5.3.2.1.1.6.5.2.2 Computation of the Variable Length Binary Sequence**

If CS is greater than 0, a variable length bits sequence (VLBS) is computed. It is a binary sequence of CS bits that is passed to the AC module for binary coding (Code\_Bits procedure of the AC module). The SWTC derived VLBS carries information about the sign of the coded SWTC and its position within the CS-derived coefficient class.

For a positive SWTC, the VLBS consists in the CS least significant bits of the SWTC. From the definition of CS (number of significant bits), we know in advance that the VLBS most significant bit is set to 1.

For a negative SWTC, the VLBS consists in the CS least significant bits of the one's complement of the SWTC. From the definition of CS, we know in advance that the VLBS most significant bit is set to 0.

The VLBS is computed as follows:



**Figure 5-14 – VLBS Computation Procedure**

Only the VLBSL least significant bits of VLBS define the bits sequence; the other bits shall be set to zero.

**5.3.2.1.1.6.5.2.3 Adaptive Statistical Models**

When entropy coding CS, the AC module makes use of an adaptive statistical model (ASM) managing the adaptive probabilities of occurrences of the different symbols. It is the responsibility of the VLC module to pass the appropriate ASM to the AC module depending on the quadrant quantised coefficients dynamic (QQCD) and on the local conditional context (CC).

According to the range of the QQCD and according to the range of the CC, the VLC module shall provide several ASM:

QQCD	Possible CC	Number of ASM	Number of symbols per ASM (NBS)
1	0, 1	2	2
2	0, 1, 2	3	3
3	0, ..., 3	4	4
4	0, ..., 4	5	5
...	...	...	...
N	0, ..., N	N + 1	N + 1

**Table 5-5 – ASM Needed by the Coder**

AASM(QQCD, CC) denotes the set of models and is indexed by QQCD and CC.

### 5.3.2.1.1.6.6 Arithmetic Coding

The Arithmetic Coding (AC) module of the coder takes as input a symbol (CS) that must be entropy coded using an adaptive statistical model (ASM). It also takes as input a variable length bits sequence (VLBS) that must be binary coded. The outputs of the AC module are sequence of bits that are pushed into the output stream.

#### 5.3.2.1.1.6.6.1 Adaptive Statistical Model

The role of the adaptive statistical model (ASM) is to manage adaptively: frequency counts, cumulative frequency counts and symbols sorting according to frequency counts.

This is implemented using a set of arrays:

- S2I(CS): holds the sorting indexes (SI) of each possible CS according to decreasing frequency count. A value of 1 denotes the most probable symbol (MPS) and a value of NBS denotes the least probable symbol. ALLLPS denotes all symbols that are not the MPS.
- I2S(SI): holds the symbols associated with each SI. This array implements the inverse of S2I(CS).
- FREQ(SI): holds the frequency counts sorted by SI.
- CUM(SI): holds the cumulative frequency counts of the FREQ(SI) array:

$$CUM(SI) = \sum_{i=SI+1}^{NBS} FREQ(i)$$

The size of these arrays is NBS+1, where NBS denotes the number of symbols managed by the ASM.

#### 5.3.2.1.1.6.6.1.1 Arrays Initialisation

At the start of the coding of the image and each time a restart marker is output all ASM provided by the VLC module shall be reset to their initial state. When coding successive WT blocks (not separated by restart markers), the VLC module shall not reset the ASM to benefit from previous blocks statistical learning.

The ASM arrays are initialised as follows:

CS	S2I(CS)
0	1
1	2
2	3
...	...
N	N+1
...	...
NBS-1	NBS

**Table 5-6 – S2I Array Initialisation**

SI	I2S(SI)	FREQ(SI)	CUM(SI)
0		0	NBS

1	0	1	NBS-1
2	1	1	NBS-2
...	...	1	...
N	N-1	1	NBS-N
...	...	1	...
NBS	NBS-1	1	0

**Table 5-7 – I2S, FREQ and CUM Array Initialisation**

Note: At any time CUM(0) holds the sum of all symbols frequency counts. Except for FREQ(0), the FREQ array entries shall always be greater than zero.

The probability of occurrence of each symbol CS is given by:

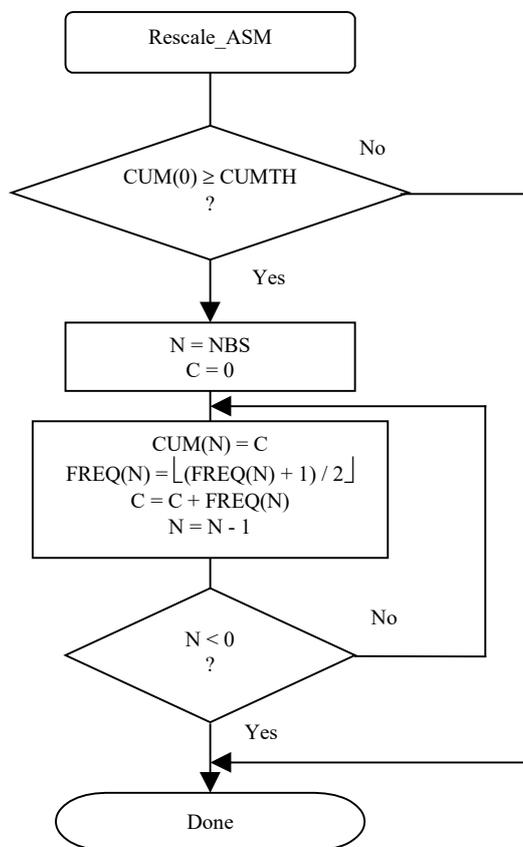
$$p(CS) = \frac{FREQ(S2I(CS))}{CUM(0)}$$

#### 5.3.2.1.1.6.6.1.2 Frequency Re-scaling

For practical AC reasons and for a beneficial compression effect, the symbols frequency counts are periodically re-scaled down. This re-scaling allows to weight more heavily the most recent events relative to older events (symbol occurrence).

The periodicity is determined by the value of CUM(0). Each time CUM(0) is equal to a pre-determined value (CUMTH), the re-scaling takes place. Since CUM(0) is always equal to the number of symbols already coded plus NBS (initial value), the re-scaling takes place at regular interval in the input stream of symbols. CUMTH shall be equal to  $32NBS$ .

The re-scaling is performed as follows:

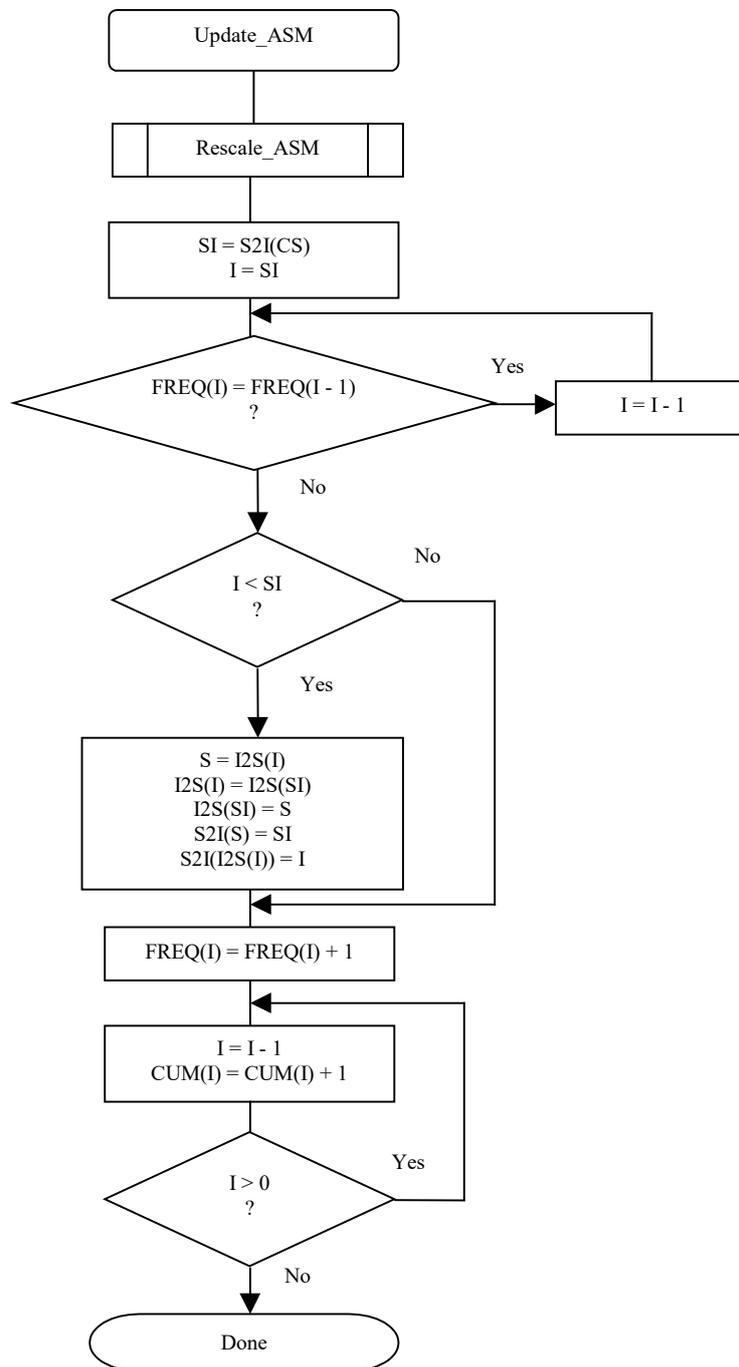


**Figure 5-15 – Model Re-scaling Computation Procedure**

**5.3.2.1.1.6.6.1.3 Model Update**

Each time a symbol is entropy coded the ASM shall be updated to adapt itself to the input stream. The frequency count of the symbol is incremented by one, the sorting index of the symbol is derived and the four arrays are adapted accordingly.

The ASM update is done as follows:



**Figure 5-16 – ASM Update Computation Procedure**

### 5.3.2.1.1.6.6.2 Arithmetic Coding Process

The entropy coding of symbols is performed by the Code\_Symbol procedure; it takes as parameters the symbol to be coded (CS) and the current ASM. The entropy coding is performed thanks to multi-symbols arithmetic coding (AC). The binary coding of variable length bits sequence is performed by the Code\_Bits procedure; it takes as parameters a bits sequence (VLBS) and the sequence length (VLBSL). The arithmetic coder is used to perform the binary coding thanks to an implicit equi-probable symbols ASM.

**5.3.2.1.1.6.6.2.1 Introduction**

The basis of AC is recursive interval subdivision based on the probability of occurrence of symbols. The only output of the AC is represented by a floating-point value inside the final interval. Practically, this floating point value ( $[0, 1[$ ) is represented by its binary representation (sum of negative powers of two) and most significant bits are output at the time they are unambiguously known.

The AC coding range (ACRANGE) represents the current coding interval size and the lower bound of this interval is denoted by ACLOW. ACRANGE and ACLOW are represented by large unsigned integer values (at least 32 bits). The number of significant bits used to represent the unsigned integer ACRANGE is given by ACRNB. The theoretical range  $[0, 1[$  is implemented using the integer range  $[0, 2^{ACRNB-1}[$ . ACRNB shall be set to 32.

Each time a symbol is entropy coded, ACRANGE is reduced according to the symbol probability (ACRANGE decrease is minimal when coding the most probable symbol, MPS) and ACLOW is updated to point the start of the new range.

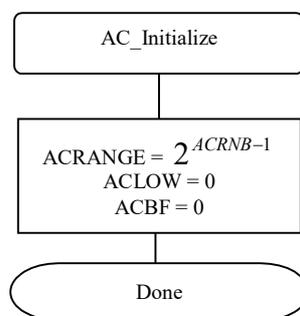
Each time a bits sequence is binary coded, ACRANGE is reduced according to the number of bits of the sequence (VLBS) and ACLOW is updated to point the start of the new range.

If ACRANGE falls below a given threshold (ACRTH), one or more bits are output and ACRANGE is doubled until it is above the threshold. ACLOW is then updated to point the start of the new range. The ACRTH value shall be equal to  $2^{ACRNB-3}$

When outputting bits, the arithmetic coder makes use of the Bit\_Plus\_Follow procedure that takes as parameter the state of the first bit (0 or 1, BS) and the number of opposite bits (opposite state) that shall be output. The number of opposite bits is given by the ACBF unsigned integer value (at least 32 bits long).

**5.3.2.1.1.6.6.2.2 Initialisation**

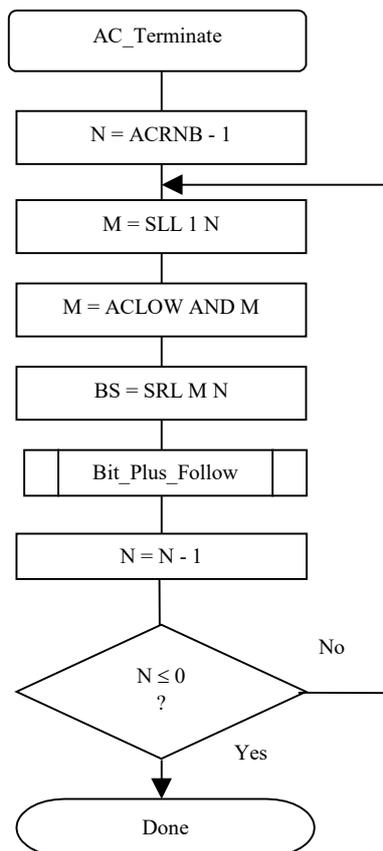
At the start of the coding of the image and each time a restart marker is output the AC module shall be reset to its initial state. The arithmetic coder is initialised as follows:



**Figure 5-17 – AC Initialisation Procedure**

**5.3.2.1.1.6.6.2.3 Termination**

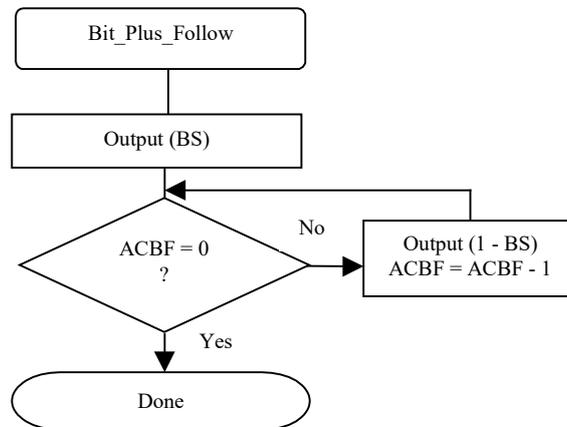
At the end of the coding of the image and before a restart marker is output the AC module shall be stopped. When the arithmetic coder is stopped, it shall flush remaining bits as well as the ACLOW value. This is done as follows:



**Figure 5-18 – AC Termination Procedure**

**5.3.2.1.1.6.6.2.4 Bits Output**

Every time the AC module writes data to the output stream, it uses the Bit\_Plus\_Follow procedure. It takes as parameter the bit state (BS) and outputs one or more bits depending on the ACBF value:

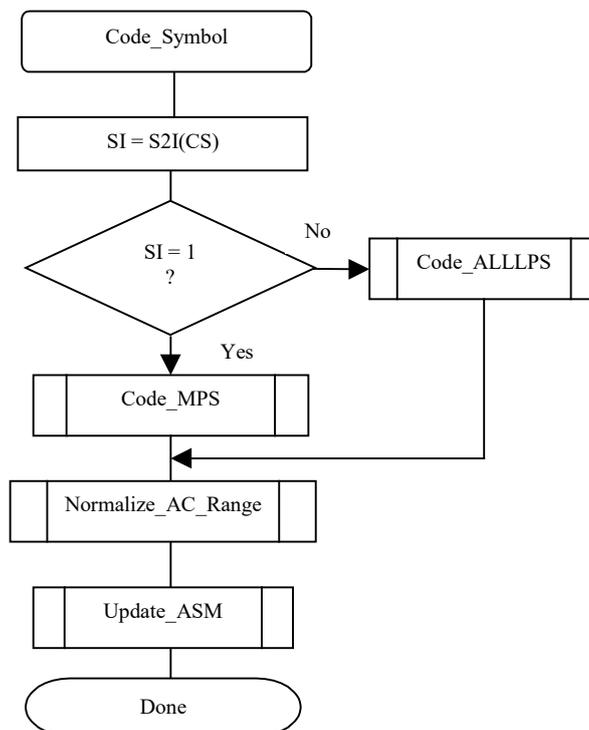


**Figure 5-19 – Bit\_Plus\_Follow Procedure**

The Output procedure writes sequentially the specified bit in the output stream (entropy coded segment). This procedure shall guarantee that any byte aligned sequence of 8 bits set to 1 (0xFF) is followed by a zero byte (0x00) in order to distinguish that bits sequence from any marker.

**5.3.2.1.1.6.6.2.5 Encoding Symbols**

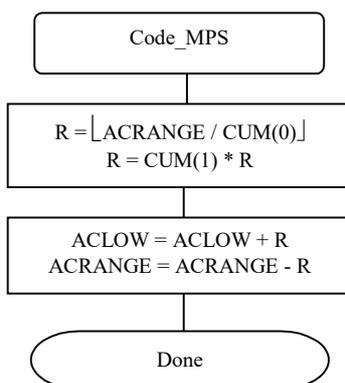
The AC module uses entropy coding to code symbols; it takes as parameters a symbol (CS) and a model (ASM). The ASM allows dividing the current range (ACRANGE) into NBS sub-ranges. Each sub-range is associated with a symbol and its size is proportional to the symbol probability. The AC entropy coder separates the symbol coding into two cases: The coding of the MPS and the coding of one of the ALLLPS.



**Figure 5-20 – Code\_Symbol Procedure**

**5.3.2.1.1.6.6.2.6 Encoding the Most Probable Symbol**

When coding the MPS the AC module selects the sub-range at the top of ACRANGE; the sub-range size is proportional to the MPS probability according to the ASM. The probability of the MPS is estimated as  $(1 - p_{ALLPS})$ ,  $p_{ALLPS}$  being the cumulated probabilities of the least probable symbols (ALLPS):

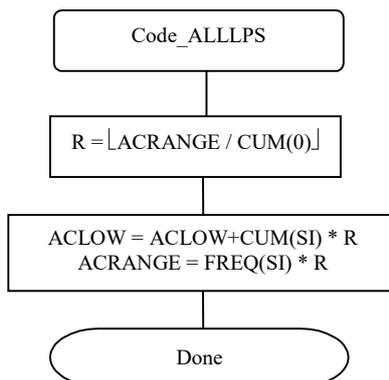


**Figure 5-21 – Code\_MPS Procedure**

NOTE: Multiplication and division operations are done with unsigned integer arithmetic. The Rescale\_ASM and Normalize\_AC\_Range procedures guarantee that  $ACRANGE \geq CUM(0)$ .

**5.3.2.1.1.6.6.2.7 Encoding Least Probable Symbols**

When coding one of the ALLPS the AC module selects one of the (NBS-1) sub-ranges at the bottom of ACRANGE; the first (bottom) sub-range is associated with the least probable symbol and the last one is associated with the second most probable symbol. The sub-range size is proportional to the symbol probability according to the ASM:



**Figure 5-22 – Code\_ALLPS Procedure**

NOTE: Multiplication and division operations are done with unsigned integer arithmetic. The Rescale\_ASM and Normalize\_AC\_Range procedures guarantee that  $ACRANGE \geq CUM(0)$ .

**5.3.2.1.1.6.6.2.8 Arithmetic Coding Range Normalisation**

Each time a symbol is entropy coded, ACRANGE is reduced. When it falls below a given threshold ACRTH, a range normalisation occurs and eventually one or more bits are output. ACRTH value is set at  $2^{ACRNB-3}$ . The normalisation is done as follow:

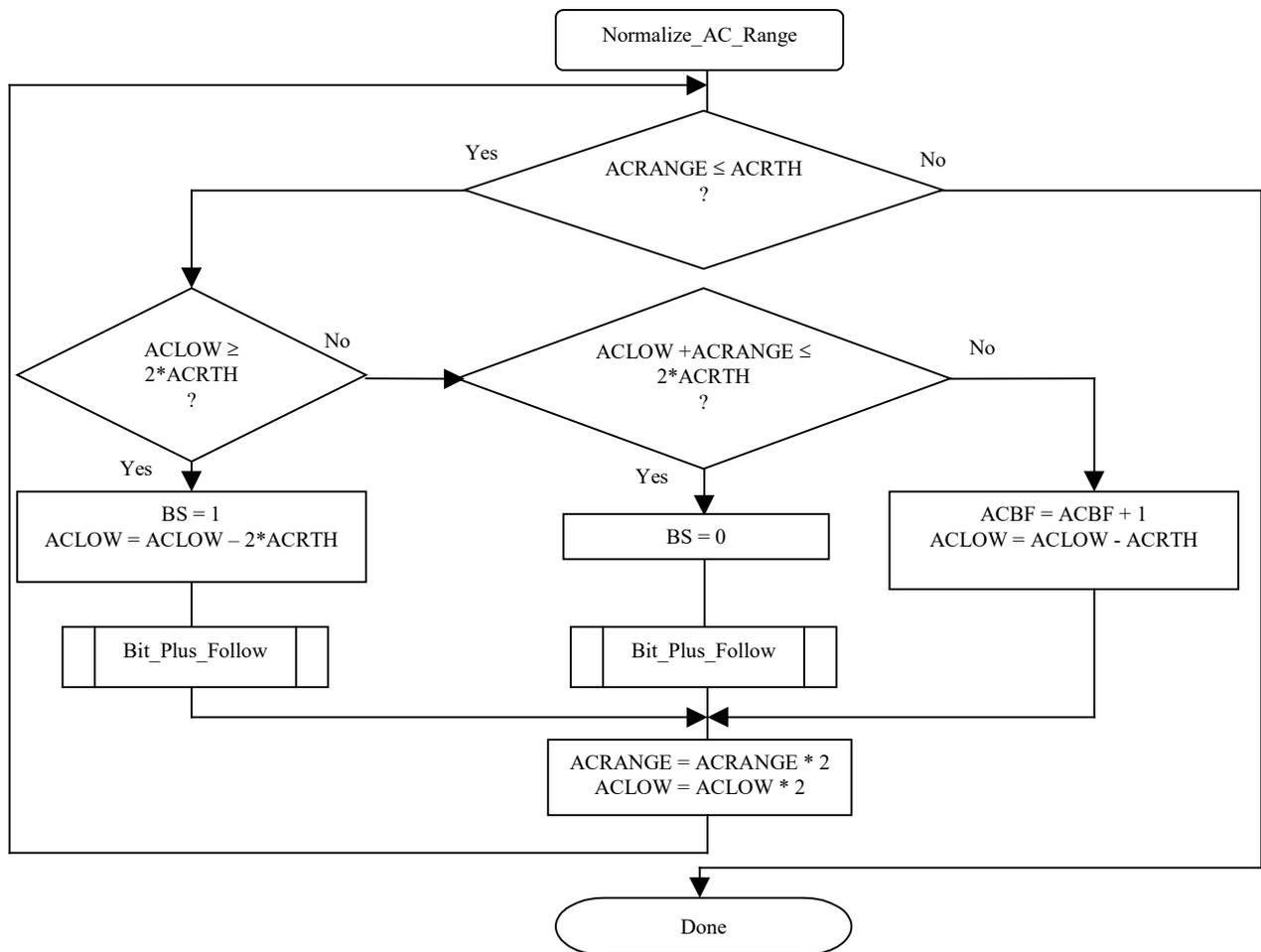


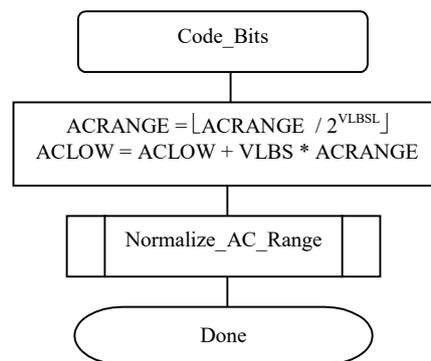
Figure 5-23 – Normalise\_AC\_Range Procedure

#### 5.3.2.1.1.6.6.2.9 Encoding Bits Sequences

When coding a variable length bits sequence (VLBS), the AC coder considers an implicit ASM of  $2^{VLBSL}$  equi-probable symbols. The ACRANGE is divided into  $2^{VLBSL}$  sub-ranges of equal length. The bottom sub-range is associated with a full zero VLBS, the top sub-range is associated with a full one VLBS.

The VLBS is represented by an unsigned integer value where only the VLBSL least significant bits define the sequence; other bits shall be set to zero.

The coding of the bit sequence is done as follows:



**Figure 5-24 – Code\_Bits Procedure**

The Normalise\_AC\_Range procedure guarantees that  $ACRANGE > ACRTH$ . VLBSL shall be in the  $[1, ACRNB-3]$  range.

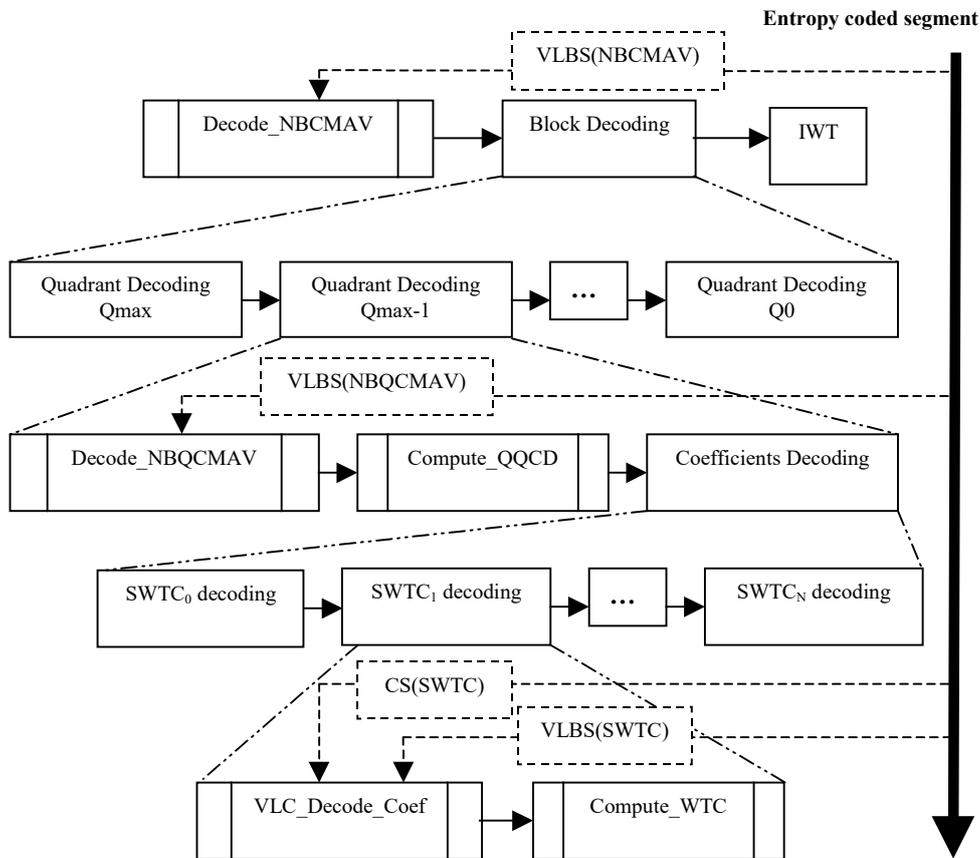
### 5.3.2.1.1.7 Decoding the Block

Just like the block coding (see Section 5.3.2.1.1.6), the decoding processing is performed sequentially on a block-by-block basis.

First, the information about the WTC dynamic (NBCMAV) is binary decoded from the input stream (entropy coded segment, ECS). The decoding of the block is then sub-divided into the decoding of each quadrant of the block (starting from Qmax and ending at Q0).

The information about the quadrant WTC dynamic (NBQCMAV) is binary decoded from the input stream. The shifted Wavelet coefficients (SWTC) are then entropy decoded using the Variable Length Decoding (VLD) and the Arithmetic Decoder (AD) modules. Then according to the lossy parameter, a de-quantisation process is applied to produce Wavelet transform coefficients (WTC). Finally, the inverse Wavelet transform (IWT) is applied on the block of coefficients, producing a block of image pixels.

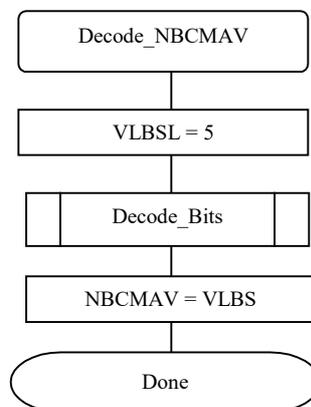
The decoding flow can be schematised as follow:



**Figure 5-25 – Decoding Flow**

**5.3.2.1.1.7.1 Number of Bits Needed to Represent the Coefficients**

The number of significant bits needed to represent the maximum absolute value coefficient within the block (NBCMAV) is decoded by the AD module as a 5 bits binary sequence. Only blocks having a non-zero NBCMAV shall be decoded; others shall have all their coefficients set to zero.



**Figure 5-26 – NBCMAV Decoding Procedure**

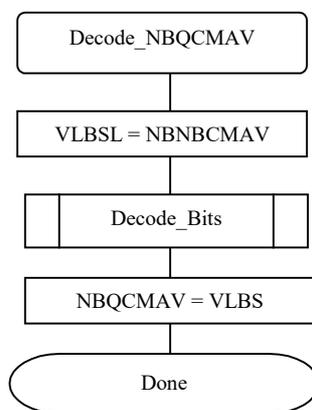
The number of significant bits needed to represent NBCMAV is computed; that value is given by NBNBCMAV (see Figure 5-27).

**5.3.2.1.1.7.2 Quadrant Scanning Order**

Quadrants are processed in decreasing QN order, i.e. beginning with the top-most; left-most quadrant (DC quadrant, QN=Qmax) and ending with the bottom-most, right-most quadrant (QN=0).

**5.3.2.1.1.7.3 Number of Bits Needed to Code the Quadrant Coefficients**

The number of significant bits needed to represent the maximum absolute value coefficient within a quadrant (NBQCMVA) is decoded by the AC module as a bits sequence of NBNBCMAV bits. NBQCMVA shall be in the [0, NBCMAV] range. Only quadrants having a non-zero NBQCMVA shall be decoded, others shall have all their coefficients set to zero.



**Figure 5-27 – NBQCMVA Decoding Procedure**

**5.3.2.1.1.7.4 Computation of the Quadrant Quantised Coefficients Dynamic**

According to the lossy parameter (LP), the quadrant number (QN) and the number of wavelet transform iterations (WI), the quadrant coefficients dynamic shift (QCDS) is derived using Table 5-3.

The quadrant quantised coefficients dynamic QQCD is then computed as described in Section 5.3.2.1.1.6.4.1

**5.3.2.1.1.7.5 Variable Length Decoding**

The Variable Length Decoding (VLD) module of the decoder takes as input a stream of symbols and bits sequences from the Arithmetic Decoding (AD) module and produces a block of shifted Wavelet transform coefficients (SWTC).

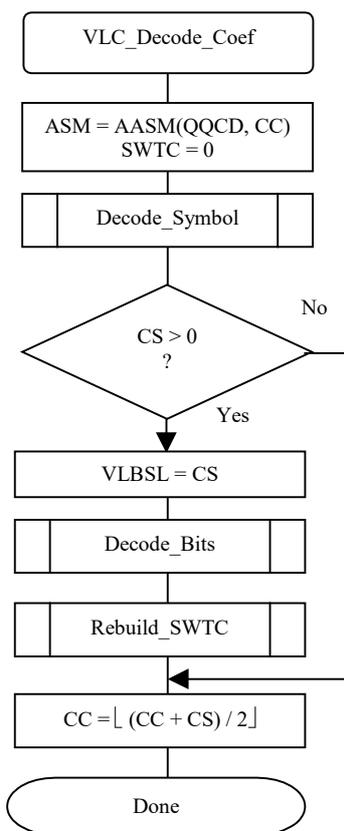
The coefficients scanning within a quadrant is the same as in the VLC module (see Section 5.3.2.1.1.6.5.1).

### 5.3.2.1.1.7.5.1 VLC Decoding of a Coefficient

From a symbol (CS) entropy decoded by the AD module (Decode\_Symbol procedure of the AD module) and from a variable length bits sequence (VLBS) binary decoded by the AD module (Decode\_Bits procedure of the AD module), a SWTC is rebuilt.

The adaptive statistical models (ASM) are managed the same way as in the coder (see Sections 5.3.2.1.1.6.5.2.3 and 5.3.2.1.1.6.6.1).

The VLC decoding of a coefficient can be schematised as follows:

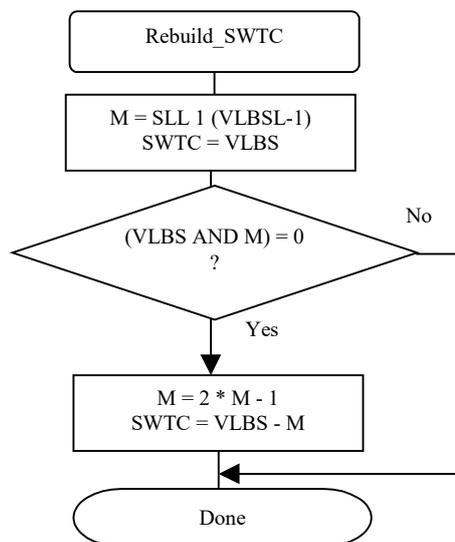


**Figure 5-28 – Coefficient VLC Decoding Procedure**

The conditional context used to select the ASM is the same as with the coder (see Section 5.3.2.1.1.6.5.2).

#### 5.3.2.1.1.7.5.1.1 Shifted Wavelet Coefficient Rebuild

When a non-zero symbol is decoded, the SWTC is rebuilt taking into account the symbol value (CS) and the variable length bits sequence (VLBS). The length (VLBSL) of the variable length bits sequence shall be equal to CS.



**Figure 5-29 – Rebuild\_SWTC Procedure**

#### 5.3.2.1.1.7.6 Arithmetic Decoding

The Arithmetic Decoding (AD) module of the decoder takes as input a sequence of bits provided by the input stream. From these bits, the AD entropy decodes a symbol (CS) using an adaptive statistical model (ASM). Eventually, it also binary decodes a variable length bits sequence (VLBS).

##### 5.3.2.1.1.7.6.1 Arithmetic Decoding Process

The entropy decoding of symbols is performed by the Decode\_Symbol procedure; it takes as parameter the current ASM. The entropy decoding is performed thanks to multi-symbols arithmetic decoding (AD). The binary decoding of a variable length bits sequence is performed by the Decode\_Bits procedure; it takes as parameter the sequence length in bits (VLBSL). The arithmetic decoder is used to perform the binary decoding thanks to an implicit equi-probable symbols ASM.

##### 5.3.2.1.1.7.6.1.1 Introduction

See Section 5.3.2.1.1.6.6.2.1 for more details.

The AD decoding range (ADRANGE) represents the current decoding interval size and the final interval value (ADVALUE) represents the final interval value relatively to the base of the current interval. The ADVALUE is known to limited precision (most significant bits). ADRANGE and ADVALUE are represented by large unsigned integer values (at least 32 bits).

The number of significant bits used to represent the unsigned integer ADRANGE is given by ACRNB (see Section 5.3.2.1.1.6.6.2.1).

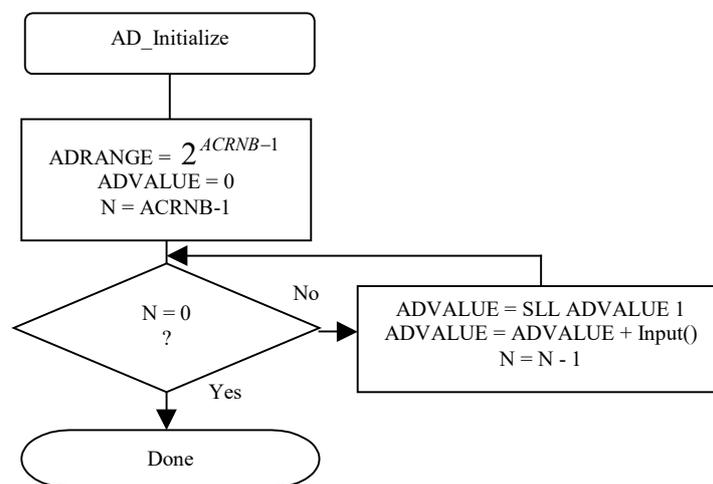
Each time a symbol is entropy decoded, ADRANGE is reduced according to the symbol probability (ADRANGE decrease is minimal when decoding the MPS) and ADVALUE is updated relatively to the start of the new range.

Each time a bits sequence is binary decoded, ADRANGE is reduced according to the number of bits of the sequence (VLBS) and ADVVALUE is updated relatively to the start of the new range.

If ADRANGE falls below a given threshold (ACRTH), one or more bits are input and pushed into the least significant bits of ADVVALUE. ADRANGE is then doubled until it is above the threshold. The ACRTH value shall be equal to  $2^{ACRNB-3}$

### 5.3.2.1.1.7.6.1.2 Initialisation

At the start of the decoding of the image and each time a restart marker is input the AD module shall be reset to its initial state. The arithmetic decoder is initialised as follows:

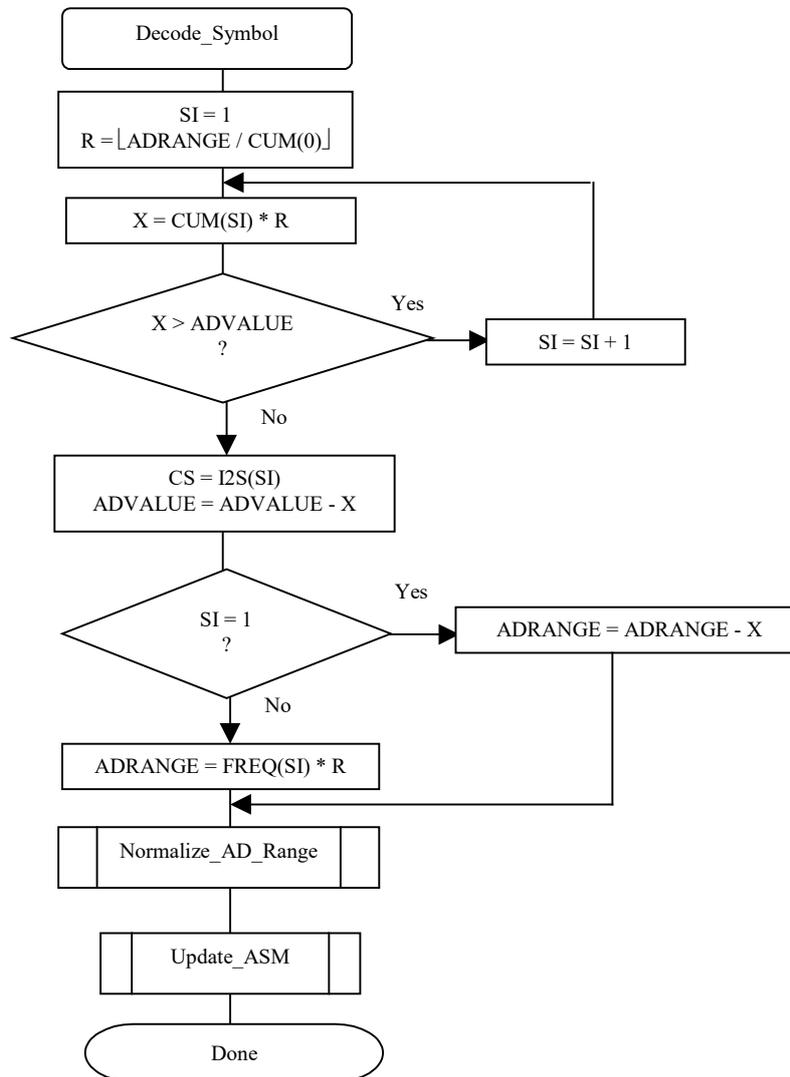


**Figure 5-30 – AD Initialisation Procedure**

The Input procedure provides the next bit from the input stream. This procedure shall guarantee that any byte aligned sequence of 8 bits set to 1 (0xFF), followed by a zero byte (0x00) will be seen as a 0xFF byte (0x00 byte is skipped). If the Input procedure is asked to return the state of the first bit (most significant bit) of a byte aligned sequence of 8 bits set to 1 (0xFF) that is not followed by a zero byte (0x00), the read from the input stream is cancelled. A cancelled read shall indicate an error in the entropy-coded segment (ECS).

### 5.3.2.1.1.7.6.1.3 Decoding Symbols

The AD module uses entropy decoding to decode symbols; it takes as parameter a model (ASM). The ASM allows dividing the current range (ADRANGE) into NBS sub-ranges. Each sub-range is associated with a symbol and its size is proportional to the symbol probability. The current final interval value (ADVVALUE) allows finding the symbol that was encoded at this stage by the AC module:



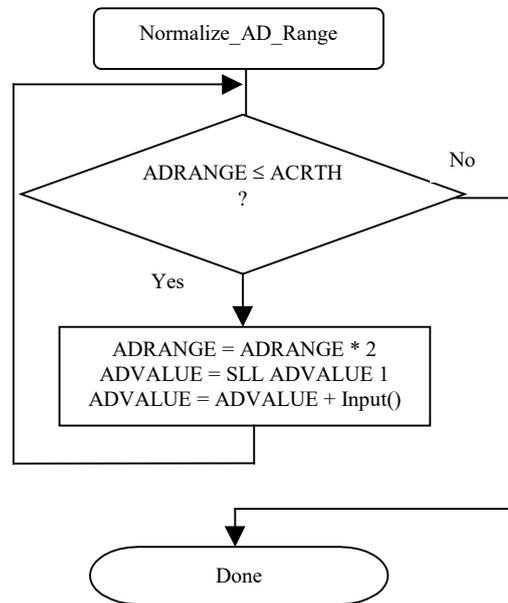
**Figure 5-31 – Decode\_symbol procedure**

NOTE: Multiplication and division operations are done with unsigned integer arithmetic. The Rescale\_ASM and Normalize\_AD\_Range procedures guarantee that  $ADRANGE \geq CUM(0)$

**5.3.2.1.1.7.6.1.4 Arithmetic Decoding Range Normalisation**

Each time a symbol is entropy decoded, ADRANGE is reduced. When it falls below a given threshold ACRTH, a range normalisation occurs and eventually one or more bits are input and pushed into the least significant bits of ADVALUE.

ACRTH value is set at  $2^{ACRNB-3}$ . The normalisation is done as follow:

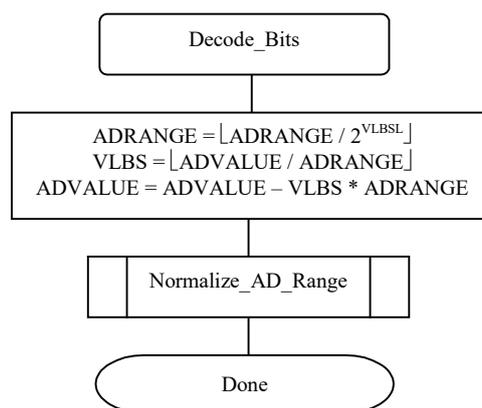


**Figure 5-32 – Normalise\_AD\_Range procedure**

**5.3.2.1.1.7.6.1.5 Decoding Bits Sequences**

When decoding a variable length bits sequence (VLBS), the AD module considers an implicit ASM of  $2^{VLBSL}$  equi-probable symbols. The current interval ADRANGE is divided into  $2^{VLBSL}$  sub-ranges of equal length. The bottom sub-range is associated with a full zero VLBS, the top sub-range is associated with a full one VLBS.

The decoding of a binary sequence is done as follow:



**Figure 5-33 – Decode\_Bits procedure**

NOTE: Multiplication and division operations are done with unsigned integer arithmetic. The Normalize\_AD\_Range procedure guarantees that  $ADRANGE > ACRTH$ . VLBSL shall be in the  $[1, ACRNB-3]$  range.

#### **5.3.2.1.1.7.7 Coefficients De-quantisation**

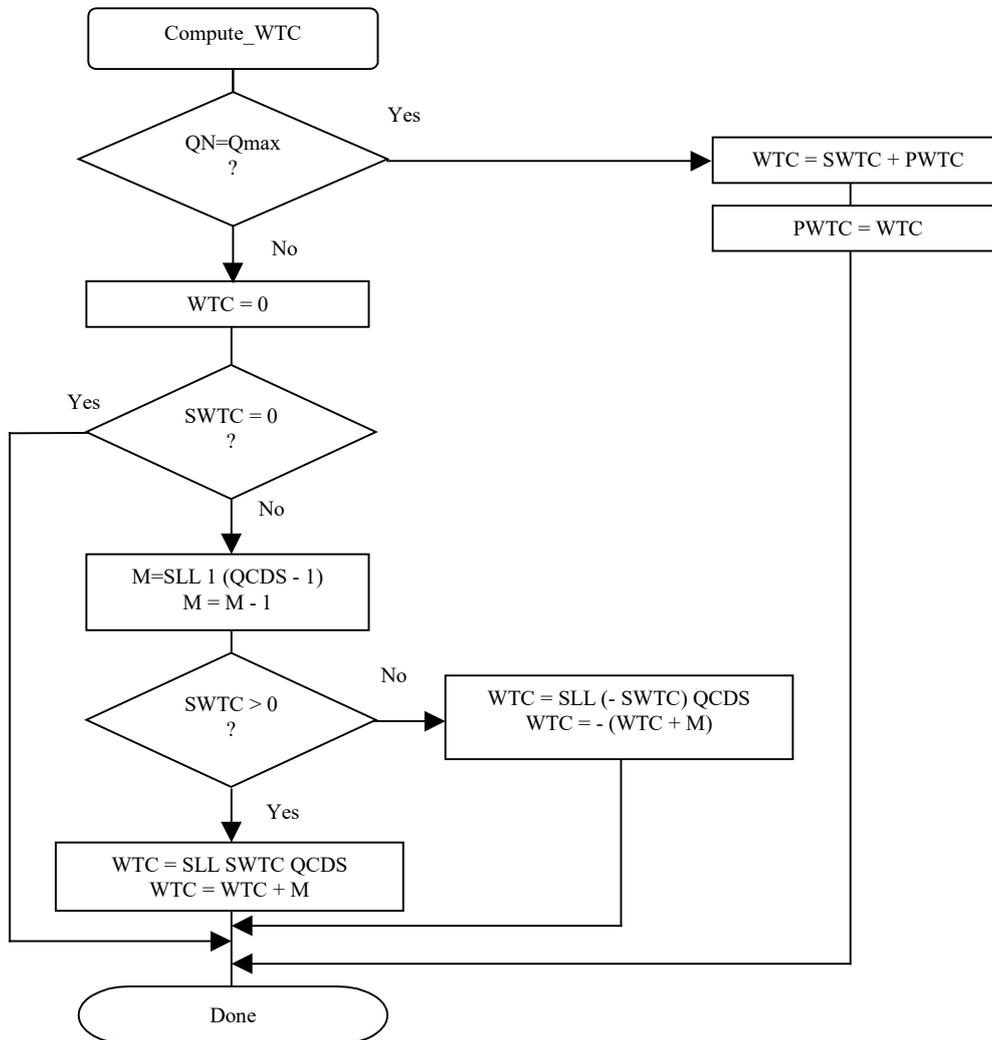
The Wavelet transform coefficients (WTC) are computed from the shifted coefficients (SWTC) by applying the quadrant coefficients dynamic shift (QCDS). The de-quantisation acts like a multiplication of the SWTC by a positive power of two ( $2^{QCDS}$ ).

##### **5.3.2.1.1.7.7.1 Computation of the WT Coefficients**

After de-quantisation, adding a constant to every non-null WT coefficient reduces the error interval due to the lossy compression; the constant is set to  $2^{QCDS-1} - 1$ . After VLD, the SWTC absolute value shall be in the  $[0, 2^{QQCD}]$  range; after computation, the WTC absolute value shall be in the  $[0, 2^{NBQCMAY}]$  range.

Given the possible range for the W1 and LP parameters, one can deduce that the DC quadrant (QN=Qmax) coefficients are always coded without any loss (QCDS is always 0). During the coding, a DPCM operation has been applied in order to prepare the coefficients for the entropy coding. The difference between the current coefficient (WTC) and the previous one (PWTC) replaced each coefficient to be coded. For the first coefficient, the PWTC value shall be equal to  $2^{QQCD-1}$ . The scanning of the coefficient during DPCM was the same as during VLC coding (see Section 5.3.2.1.1.6.5.1).

Since the difference operation may have increased the DC quadrant coefficient dynamic by one bit, the QQCD is increased by one before the VLD phase.



**Figure 5-34 – WTC computation procedure**

**5.3.2.1.2 WT Compression Data Format Specification**

**5.3.2.1.2.1 General Aspects**

Structurally, the compressed data format consists of an ordered collection of parameters, markers and entropy-coded data segments.

**5.3.2.1.2.1.1 Constituent parts**

See §B1.1 of [RD11], were table B.1 is replaced by Table 5-8.

Code Assignment	Symbol	Description
General markers		
X'FF01'	Header	Start of Image

X'FF02'	Data	End of the frame header
X'FF03'	Footer	End of Image
Restart interval termination		
X'FFE0' through X'FFEF'	WRST <sub>m</sub>	Restart with modulo 16 count "m"

**Table 5-8 – Marker Code Assignments**

### 5.3.2.1.2.2 General Syntax

The conventions used for the syntax figures are defined in §B.1.3 of [RD11].

#### 5.3.2.1.2.2.1 High-level Syntax

Figure 5-35 specifies the order of the high-level constituent parts of the compressed data format for the WT scheme.

The four markers shown in Figure 5-35 are defined as follows:

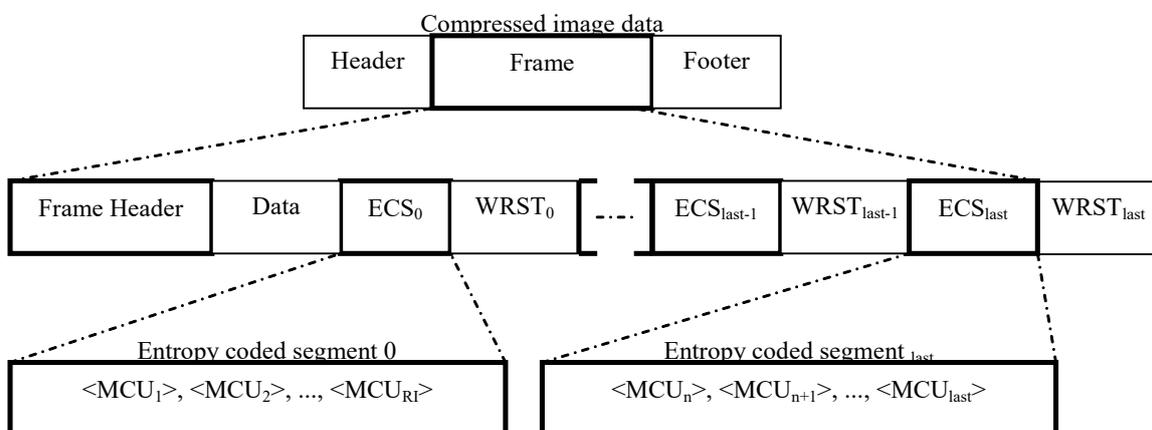
**Header:** Header marker – Marks the start of a compressed image represented in the compressed data format.

**Data:** Data marker – Marks the end of the header and the start of the entropy coded segments.

**Footer:** Footer marker – Marks the end of a compressed image represented in the compressed data format.

**WRST<sub>m</sub>:** Restart marker – A conditional marker which is placed between entropy-coded segments only if restart is enabled. There are 16 unique restart markers (m=0 ... 15) which repeat in sequence from 0 to 15, starting with 0 for each frame, to provide a modulo 16 restart interval count.

The top level of Figure 5-35 specifies that the compressed data format shall begin with a Header marker, shall contain one frame and shall end with a Footer marker.



**Figure 5-35 – Compressed data format syntax**

The second level of Figure 5-35 specifies that a frame shall begin with a frame header followed by a data marker and shall contain one or more entropy-coded data segments. If restart is not enabled, there shall be only one entropy-coded segment (the one labelled “last”), and no restart markers shall be present. If restart is enabled, a restart marker shall follow each complete entropy-coded segment.

The third level of Figure 5-35 specifies that each entropy-coded segment is comprised of a sequence of entropy-coded MCUs. If restart is enabled and the restart interval is defined to be  $R_i$ , each entropy-coded segment except the last one shall contain  $R_i$  MCUs and hence be complete. The last one shall contain whatever number of MCUs completes the frame.

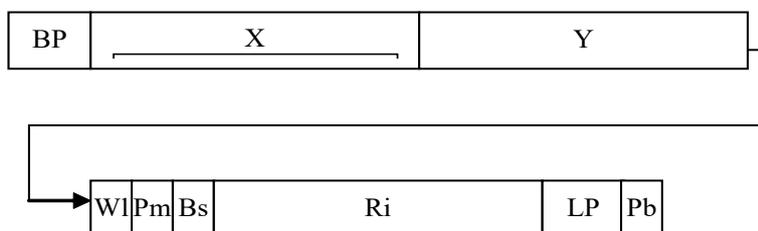
**5.3.2.1.2.2.2 Frame Header Syntax**

Figure 5-36 specifies the frame header that shall be present at the start of a frame. This header specifies the image characteristics and encoder parameters.

The Restart interval will be a value to represent 8, 16, 32 or 64 lines of the input data.

The supported pixel resolution will be between 1 and 16bpp, 16 included.

Images supported will have maximum 16384 lines and maximum 16384 columns.



**Figure 5-36 – Frame Header Syntax**

<b>Title: Frame Header</b>	<b>Id: FRAME_HEADER</b>	
FRAME_HEADER	::=	RECORD
{		
BPP	BITSTRING SIZE (4)	-- pixels resolution of the input image (bpp) modulo 16. The pixel resolution ranges from 1 to 16.
X	UNSIGNED SHORT	-- number of samples per line, variable length, covers normal scan and reduced scan sizes, value will be an integer multiple of 8, with a maximum of 16384.
Y	UNSIGNED SHORT	-- number of lines variable (464 lines are the current baseline for LRIT/HRIT image segment files)

W1	BITSTRING SIZE (2)	-- number of Wavelet transform level with 0 means 3 levels 1 means 4 levels 2 means 5 levels 3 means 6 levels
Pm	BITSTRING SIZE (2)	-- Prediction mode 0 means no prediction 1 means predictor A 2 means predictor B 3 means predictor C
Bs	BITSTRING SIZE (2)	-- Block size 0 means 16x16 1 means 32x32 2 means 64x64 3 means whole image
RI	UNSIGNED SHORT	-- restart interval (number of MCUs between two restart markers, 0 means restart is disabled)
LP	BITSTRING SIZE (4)	-- lossy parameter. 0 means lossless compression, any other value <16 means lossy
Pb	BITSTRING SIZE (2)	-- 2 pad bits to have a frame header size that is a multiple of 8 bits
}		

**Table 5-9 – Frame Header Structure**

Note: not all W1 values can be combined with a given Bs value.

### 5.3.2.2 Mission Specific JPEG Structure and Supported Modes

This section defines the mission specific JPEG implementation. It includes the definition of the overall structure, the used compression processes, the coding applied and the detailed marker segments.

The JPEG compression encoding process will transform the data field of an LRIT/HRIT image segment file into one *JPEG image* in accordance with the data format definitions given in [RD.11]. All symbol naming conventions used in [RD.11] have been transcribed to the conventions used in [RD.1]. The record structures provided in the following sub-sections are by definition identical to the ones defined in [RD.11]. Any parameter or structural limitations applicable to LRIT/HRIT are noted in the last column of the tables defining the various record structures. For explanations going beyond the given detail in this document the reader should refer to [RD.11].

Figure 5-37 shows the JPEG compressed image data structure closely following these definitions and the used terminology. The figure already includes certain assumptions about the JPEG modes used for the MSG dissemination concept.

The MSG dissemination will only make use of the following JPEG modes:

- sequential mode (as opposed to progressive → no multi-scans)
- non-interleaved mode (single spectrum, no multi-components)
- non-hierarchical mode (non-differential coding → no multi-frames)

For MSG, the following compression modes vs. input data resolution will be supported:

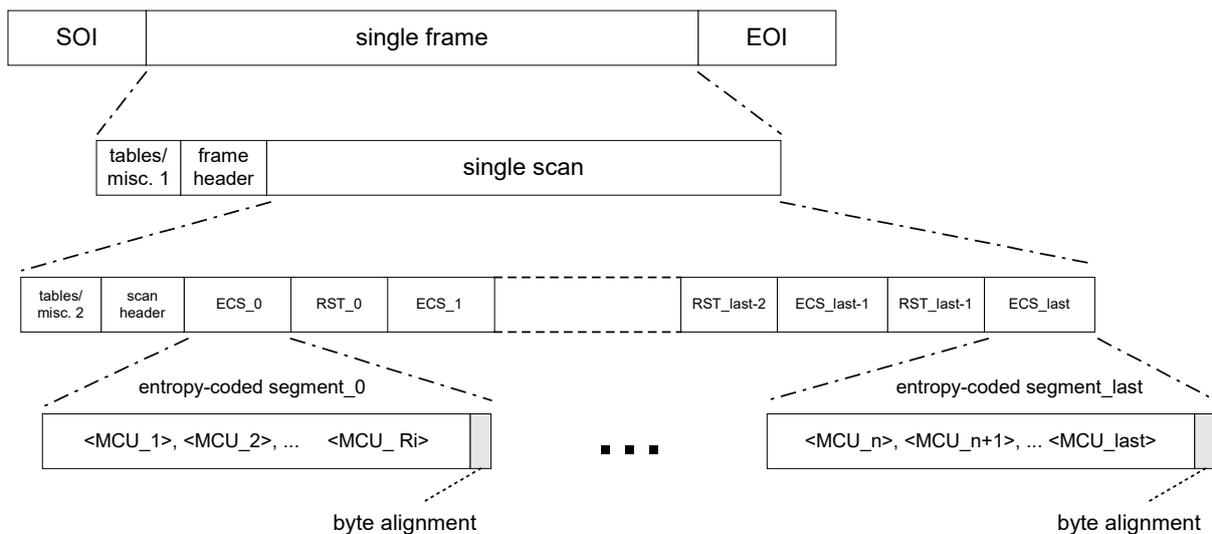
Input Data Resolution	JPEG Compression Mode	
	Lossy	Lossless
2 ... 16 bit	Not applicable	Predictive process on input data samples (Huffman coding)
8 bit	8 bit baseline process (Huffman coding)	
10 bit	8 bit baseline process (Huffman coding)	
10 bit	12 bit extended process (Huffman coding)	
12 bit	12 bit extended process (Huffman coding)	

**Table 5-10 – JPEG Compression Modes vs. Input Data Resolution**

Consequently, one JPEG image contains one frame with only one scan embedded between start of image (SOI) and end of image (EOI) markers. A JPEG scan will contain a number of entropy coded segments (ECS) of one component. The ECS are separated from each other by restart markers of (RST) to allow for re-synchronisation within an error-prone communication system.

An ECS contains minimum coded units (MCU). In the case of DCT-based (lossy) processes an MCU originates from an 8x8 pixel array. For lossless processes, an ECS consists of at least a pixel row. Huffman entropy coding will form the baseline for the MSG dissemination service.

The MSG DADF will not use arithmetic coding. The output of the JPEG process creates a byte aligned output as described in [RD.11, Section B.1.1.5]



**Figure 5-37 – MSG Mission Specific JPEG structure of compressed image data**

SOI            Start of Image Marker  
 EOI            End of Image Marker  
 ECS            Entropy coded segment  
 RST<sub>m</sub>        Restart Marker  
 MCU            Minimum coded unit

Ri Restart Interval

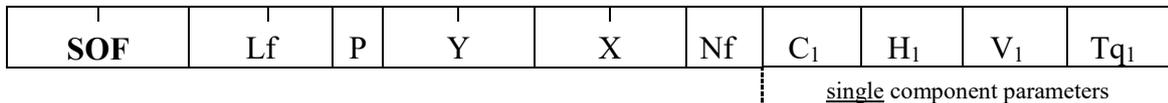
**5.3.2.2.1 JPEG Frame Header Structure**

The JPEG frame header directly follows the ‘tables/misc. 1’ field. It specifies the applied JPEG encoding process via its ‘Start of Frame’ marker (SOF) and provides information about sizing and component structure. Figure 5-38 and Table 5-11 provide all details of the MSG mission specific JPEG implementation.

The following restrictions apply:

- Only a sub-set of all possible SOF will be used
- Only parameters of a single component will be contained

The structure of a JPEG Frame Header is:



**Figure 5-38 – SOF structure**

<b>Title: Frame Header</b>	<b>Id: FRAME_HEADER</b>	
FRAME_HEADER	::=	RECORD
{SOFn		ENUMERATED SHORT -- start of frame marker
		{ baseline DCT sequential SOF <sub>0</sub> (Huffman coding)
		('FFC0'h),
		extended DCT sequential SOF <sub>1</sub> (Huffman coding)
		('FFC1'h),
		spatial sequential lossless SOF <sub>3</sub> (Huffman coding)
		('FFC3'h)
		}
Lf		UNSIGNED SHORT (11) -- frame header length, fixed value
P		UNSIGNED BYTE -- sample precision, variable value
Y		UNSIGNED SHORT -- number of lines
		variable (464 lines are the current baseline for LRIT/HRIT image segment files)
X		UNSIGNED SHORT -- number of samples per line, variable length, covers normal scan and reduced scan sizes, value will be an integer multiple of 8
Nf		UNSIGNED BYTE (1) -- number of image components, fixed value (only single component is supported)
C <sub>1</sub>		UNSIGNED BYTE (0) -- component identifier, fixed default value
H <sub>1</sub>		BITSTRING SIZE (4) -- horizontal sampling factor, fixed value (1)
V <sub>1</sub>		BITSTRING SIZE (4) -- vertical sampling factor, fixed value (1)
Tq <sub>1</sub>		UNSIGNED BYTE (0) -- quantisation table selector, fixed default value
}		

**Table 5-11 – Frame Header Structure**

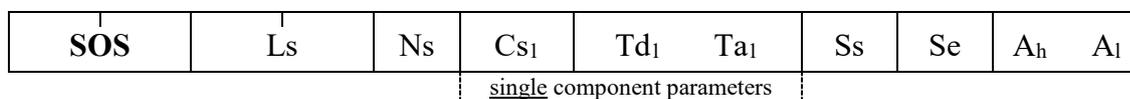
### 5.3.2.2.2 JPEG Scan Header Structure

The JPEG scan header directly follows the ‘tables/misc. 2’ field. It specifies further component specific parameters, selects entropy coding tables and their start values. Figure 5-38 and Table 5-12 provide all details of the MSG mission specific JPEG implementation.

The following restrictions apply:

- Only a SOF sub-set will be used
- Only parameters of a single component will be contained

The scan header will have the following structure:



**Figure 5-39 – SOS Structure**

<b>Title: Scan Header</b>		<b>Id: SCAN_HEADER</b>	
SCAN_HEADER	::=	RECORD	
{ SOS		UNSIGNED SHORT (‘FFDA’h)	-- start of scan marker
Ls		UNSIGNED SHORT (9)	-- scan header length, fixed length
Ns		UNSIGNED BYTE (1)	-- number of image components, fixed to 1
Cs <sub>1</sub>		UNSIGNED BYTE (0)	-- scan component selector, fixed to 0
Td <sub>1</sub>		BITSTRING SIZE (4)	-- DC entropy coding table selector, fixed to 0 (table 0)
Ta <sub>1</sub>		BITSTRING SIZE (4)	-- AC entropy coding table selector, fixed to 0, i.e. table 0 for lossy compression N/A (0) for lossless compression
Ss		UNSIGNED BYTE	-- start of spectral or predictor selection 0 for lossy processes, 1-7 according predictor table (see [RD.11, annex H])
Se		UNSIGNED BYTE	-- end of spectral selection 63 for lossy processes 0 for lossless processes
A <sub>h</sub>		BITSTRING SIZE (4)	-- successive approximation bit position high, fixed to 0 (no progressive mode used)
A <sub>l</sub>		BITSTRING SIZE (4)	-- successive approximation bit position low or point transform 0 for lossy processes 0..15 for lossless processes
}			

**Table 5-12 – Scan Header Structure**

### 5.3.2.2.3 Structure of Table and Miscellaneous Marker Segments

The MSG mission specific JPEG implementation will use the following tables and miscellaneous marker segments:

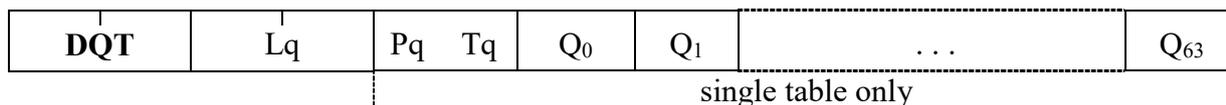
- Define quantisation table(s) (DQT)
- Define Huffman table(s) (DHT)
- Define restart interval (DRI)
- **Define quantisation Table Marker** (for lossy JPEG compression only)

In the case of DCT-based encoding processes (lossy compression), the ‘Define quantisation Table’ Marker will be contained in the ‘tables/misc. 1’ field(s) preceding the ‘Start of Frame’ Marker. Its syntax will follow the specification given in [RD.11, Section B.2.4.1].

Only one quantisation table will be contained in one JPEG image.

The update interval of the quantisation table elements  $Q_k$  in terms of image segment files/repeat cycle/spectral channel will depend on the dynamic behaviour of the satellites spectral channels.

The structure of the quantisation table marker (DQT) is:



**Figure 5-40 – Quantisation table structure**

Title: Quantisation Table Structure	Id: QUANT_TABLE_STRUCT	
QUANT_TABLE_STRUCT ::=	RECORD	
{ DQT	UNSIGNED SHORT ('FFDB'h)	-- fixed value
Lq	UNSIGNED SHORT (67 or 131)	-- quantisation table length, fixed value due to single table use
Pq	ENUMERATED SIZE (4)	-- quantisation table element precision
	{ baseline DCT (0), extended DCT (1) }	
Tq	ENUMERATED SIZE (4)	-- quantisation table identifier
	{ table 0 (0), table 1 (1), table 2 (2), table 3 (3) }	Only one table will be used at a time, the default value will be '0'
Qk	ARRAY SIZE (64) OF CHOICE	-- quantisation table elements
}	{ Pq=0 UNSIGNED BYTE, Pq=1 UNSIGNED SHORT }	value range for baseline DCT (1..255), value range for extended DCT (1..65535)

**Table 5-13 – DQT Marker**

- **Define Huffman Table Marker**

The ‘Define Huffman Table’ Marker syntax will follow the specification given in [RD.11, sect. B.2.4.2]. This marker will be contained in the ‘Tables/Misc. 2’ field directly following the ‘SOF’ marker. Not more than two ‘DHT’ markers (one DC table 0 and one AC table 0) will be contained per JPEG image.

The update interval of the HUFFVAL list parameters in terms of image segment files/repeat cycle/spectral channel will depend on the dynamic behaviour of the satellites spectral

channels. The dissemination element will allow to use different Huffman tables for all file types #0 but disregarding any segmentation. I.e., the granularity will either be a S/C spectral name, a met. product name or a service message name.

The structure of one ‘Define Huffman Table’ marker (DHT) is:

<b>DHT</b>	Lh	Tc	Th	L <sub>1</sub>	L <sub>2</sub>	...	L <sub>16</sub>	HUFFVAL_list
------------	----	----	----	----------------	----------------	-----	-----------------	--------------

**Figure 5-41 – Huffman table structure**

<b>Title:</b> Huffman Table Structure	<b>Id:</b> HUFF_TABLE_STRUCT
HUFF_TABLE_STRUCT ::=	RECORD
{ DHT	UNSIGNED SHORT ('FFC4'h) -- fixed value
Lh	UNSIGNED SHORT -- Huffman table definition length, variable value
Tc	ENUMERATED SIZE (4) -- table class
Th	{ DC Table (0), AC Table (1)} ENUMERATED SIZE (4) -- Huffman table identifier
Li	{ table 0 (0), table 1 (1), table 2 (2), table 3 (3), } ARRAY SIZE (16) -- number of Huffman codes of length i
HUFFVAL_list	OF UNSIGNED BYTE ARRAY SIZE (MT) -- list of values associated with each Huffman code of length I according to the Huffman coding model
}	OF UNSIGNED BYTE

**Table 5-14 – DHT Marker**

Definition of MT:

$$MT = \sum_{i=1}^{16} Li(t)$$

• **Define Restart Interval Marker**

The ‘Define Restart Interval’ Marker syntax will follow the specification given in [RD.11, sect. B.2.4.4]. This marker will be contained in the ‘Tables/Misc. 2’ field following the ‘SOF’ marker.

The structure of the ‘Define Restart Interval’ marker (DRI) is:

<b>DRI</b>	Lr	Cs
------------	----	----

**Figure 5-42 – Restart Interval Definition Structure**

<b>Title:</b> Define Restart Interval Structure	<b>Id:</b> RESTRT_INTER_STRUCT
RESTRT_INTER_STRUCT ::=	RECORD
{ DRI	UNSIGNED SHORT ('FFDD'h) -- fixed value
Lr	UNSIGNED SHORT (4) -- restart interval segment length, fixed

Ri	UNSIGNED SHORT	value -- restart interval (for value see Table 5-16)
----	----------------	---

**Table 5-15 – DRI Marker**

	value of Ri	
	DCT based lossy compression	Lossless compression
LRIT/HRIT	The number of MCUs will be of a value to represent 8, 16 or 32 lines of the input data.	The number of the n x MCUR will be of a value to represent 1, 2, 4 or 8 lines of the input data.

**Table 5-16 – LRIT/HRIT Restart Intervals**

### 5.3.2.3 Mission Specific Implementation of T.4 Coding

Only one dimensional coding will be used.

The LRIT/HRIT file data field will be compressed as defined in [RD.4, Section 4.1]. The fill bit insertion as specified in [RD.4, Section 4.1.3] will not be supported.

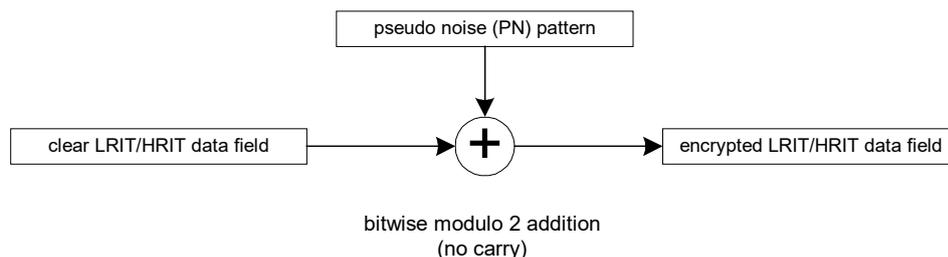
## 5.4 Encryption

The dissemination service includes a mechanism to control the access to HRIT and LRIT in accordance with the EUMETSAT data policy. The same encryption principle will be used for LRIT and HRIT but separate encryption processing will be performed. The selection of products to be encrypted and the use of the relevant key numbers will differ between LRIT and HRIT.

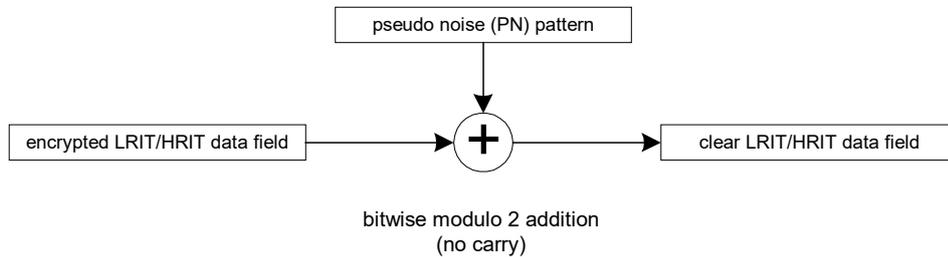
### 5.4.1 En-/Decryption Principle

#### 5.4.1.1 Description

The encryption algorithm will only operate on the data fields of LRIT/HRIT files and leave all header records unmodified. The encryption principle bases on the generation of a pseudo noise (PN) pattern which will be added to the clear data via a bit-by-bit modulo 2 addition (logical exclusive-OR, no carry) to the data field by the MSG DADF as shown in Figure 5-43. The user stations will need to apply the same principle to decrypt the encrypted data field accordingly (see Figure 5-44). A Station Key Unit (SKU) is mandatory for the creation of the Pseudo Noise Key being one value required for the PN pattern generation.



**Figure 5-43 – Encryption Principle at the Dissemination Element**



**Figure 5-44 – Decryption at the User Stations**

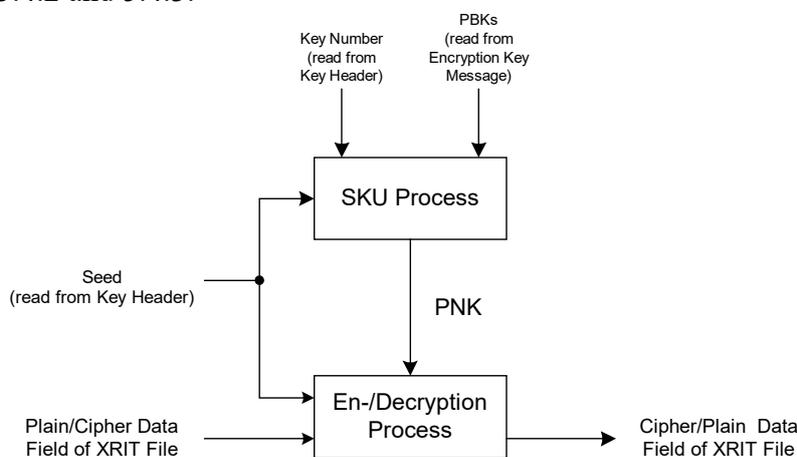
The PN pattern generation is based on DES3 which is a triple implementation of the Data Encryption Standard (DES) [RD.12]. The DES and DES3 conventions (e.g. graphical representation, numbering scheme) used in the MSG context are defined in Sections 5.4.1.2 and 5.4.1.3.

If encryption is applied to the LRIT/HRIT data field, a key header (header type #7) as defined in sect. 4.3.2.8 will be contained in the LRIT/HRIT file.

The key header defines which Key\_Number and Seed are used to create a Pseudo Noise Key (PNK). The PNK and the Seed are then used to generate the PN pattern. For the first processing step, the PNK creation, the DADF and the User Stations require a Station Key Unit (SKU). SKUs will be made available to end-users under the terms of a license agreement. Each SKU will have a unique station number.

For each station number and key number a Public Key (PBK) will be distributed via the file type #3 (encryption key message, see sect. 4.2.5) to the end-users. The PBKs will have to be stored in the SKUs and to be updated each time new keys are received. A functional block diagram of the PN generation is shown in Figure 5-45.

Further details about the SKU functionality and the PN generation can be found in Sections 5.4.2 and 5.4.3.



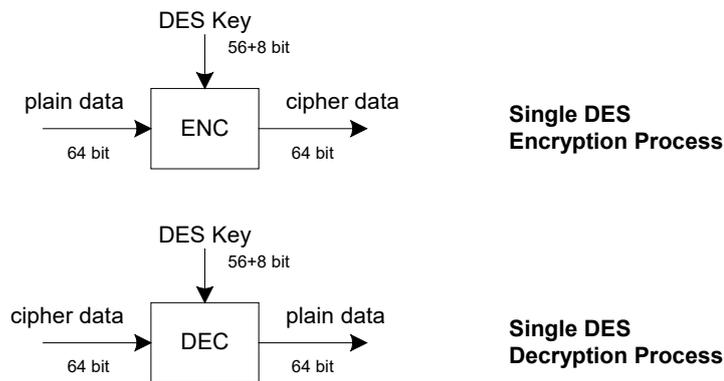
**Figure 5-45 – SKU Process and PN Generator**

A validation example for all defined algorithms can be found in a subsection of Appendix C.4.

**5.4.1.2 Triple DES Definition**

The MSG Encryption System makes use of a *triple* implementation of the Data Encryption Standard (DES). A *single* DES encryption /decryption process is defined by [RD.12].

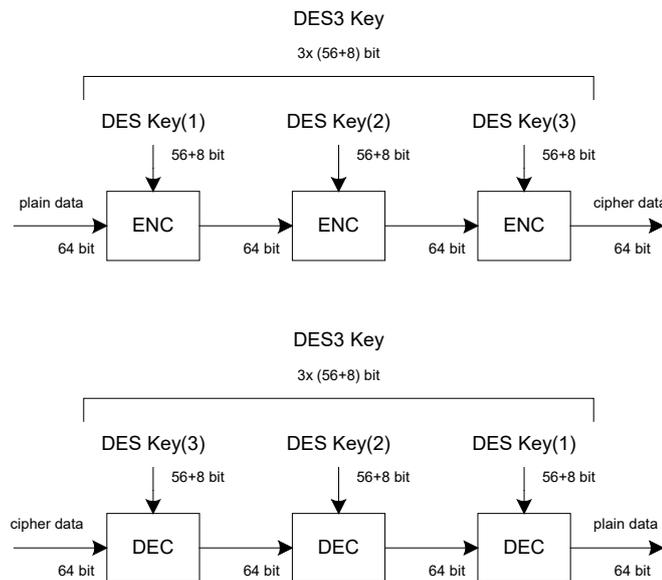
The following graphical symbols are used in this document to represent the *single* DES processes together with their input and output data:



**Figure 5-46 – Graphical Representation of single DES Processes**

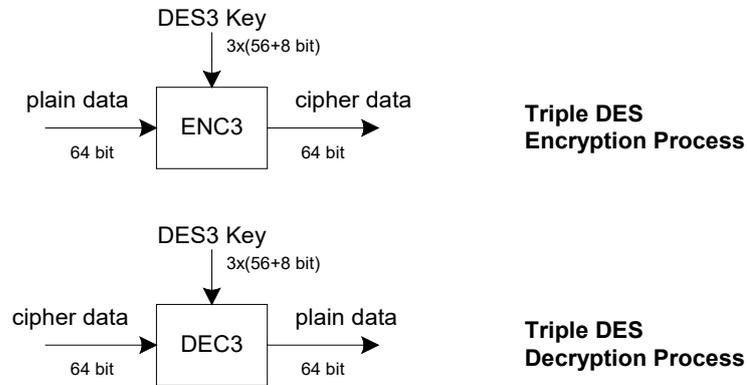
A *triple* DES process (DES3) is defined as a concatenation of three *single* DES processes. A DES3 **decryption** process is called **DEC3** and, accordingly, a DES3 **encryption** process is named **ENC3**.

Figure 5-47 explains such concatenation of the *single* DES processes to *triple* processes:



**Figure 5-47 – Concatenation of single DES Processes to triple DES Processes**

In accordance with the above naming convention, the concatenated graphical symbols in Figure 5-47 can be replaced by the **ENC3** or **DEC3** symbols shown in Figure 5-48:



**Figure 5-48 – Graphical Representation of triple DES Processes**

**5.4.1.3 DES Key Representation**

The DES3 keys used in the previous section consist of a concatenation of three ‘single’ DES keys (Key(1), Key(2) and Key(3)). Each ‘single’ DES key consists of 64 bit, 56 of which are used by the algorithm (forming the active key) and 8 of which are parity bits and used to detect errors in the key.

The DES3 key numbering convention used in Figure 5-49 conforms with [RD.12]. The 64 bit per ‘single’ DES key are numbered from left to right. Bits (8, 16, 24, ..., 64) are used for the parity checking of each 8-bit byte. The parity bits are set to the complement of the modulo 2 sum of the previous seven bits.

<b>DES3 Key</b>		
<b>DES Key(1)</b> 56 bit key(1) + 8 bit parity K(1,1), K(1,2), K(1,3), ... , K(1,64)	<b>DES Key(2)</b> 56 bit key(2) + 8 bit parity K(2,1), K(2,2), K(2,3), ... , K(2,64)	<b>DES Key(3)</b> 56 bit key(3) + 8 bit parity K(3,1), K(3,2), K(3,3), ... , K(3,64)

notation of K(a,b): a = DES Key number, b = DES Key bit number

In the case keys are distributed via communication means, K(1,1) equals by definition to the MSB (CCSDS Bit 0) and is transmitted first.

**Figure 5-49 – DES3 Key decomposition**

All DES3 keys used in the MSG encryption scheme, namely the MSKs and MGKs will follow this convention and contain parity bits as defined in [RD.12].

PBK and the PNK contents will follow the same naming and numbering conventions. But as the PBK is a ‘data’ output of a DES encryption process the PBK parity bit locations contain data instead of parities.

#### **5.4.1.4 MSG Key Definition**

The MSG encryption infrastructure requires four types of keys:

- Master Keys (MSK)
- Message Keys (MGK)
- Public Keys (PBK)
- PN Keys (PNK)

##### **5.4.1.4.1 Master Keys - MSK**

Master Keys are secret elements which are fixed (opposed to Message Keys or Public Keys which are considered to be static only for a limited period of time).

Master Keys are used:

- At the EUMETSAT Key Management Centre (KMC) for the Public Key generation (Message Key encryption against the relevant Master Key)
- At the MSG DADF and the user station locations for the Public Key decryption (i.e. Message Key recovery )

The KMC generates for the DADF and each user station a specific MSK. Each MSK is programmed into a Station Key Unit (SKU) which forms part of the DADF and every user station. The SKU avoids direct access to the secret MSK.

MSKs contain valid parity bits.

##### **5.4.1.4.2 Message Keys - MGK**

Message Keys are secret elements generated by the KMC and which are to be considered static. MGK sets are updated periodically according to operational issues. Each MGK is responsible for the encryption/decryption of more or less application data type (sub-)groups. MGKs are transmitted to the user stations via dissemination in encrypted form as so-called Public Keys. An MGK in clear form does only exist in the KMC and within the SKU process.

MGKs contain valid parity bits.

##### **5.4.1.4.3 Public Keys - PBK**

Public Keys are non-secret elements generated by the KMC and distributed to the DADF and the user stations. They are the result of an encryption of MGKs as data input with MSKs acting as keys. As being derived from MGKs, the PBKs are static and are updated periodically, too.

PBKs do not contain valid parity bits. The bits positioned at the locations of parity bits must not be modified as they are required for a precise MGK reproduction within the SKU.

**5.4.1.4.4 PN Keys - PNK**

PN Keys are dynamic keys which are used to generate the PN pattern for LRIT/HRIT data field encryption/decryption. PNKeys are generated for each LRIT/HRIT file and are a function of the seed and the MGK. The PNK forms the output of an SKU process and is the input to a PN Pattern Generation process.

PNKs do not contain valid parity bits.

**5.4.2 PN Pattern Generation**

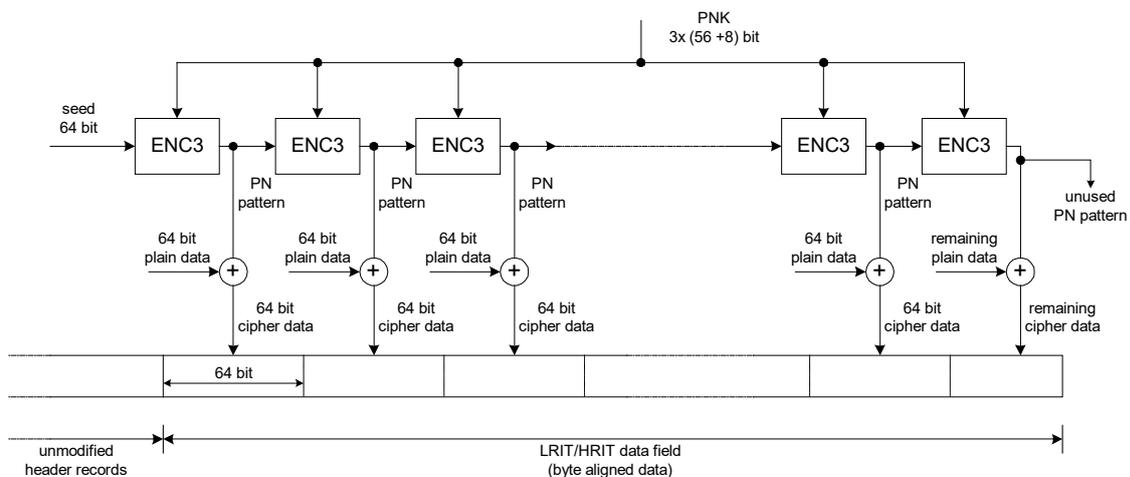
The PN pattern is a result of repeated triple DES encryption (ENC3) in an output feedback mode (OFB) with full feedback. The OFB is defined by [RD.13] and is graphically represented in the upper part of Figure 5-50. This mode avoids error propagation in an error prone communication system. The input and output of an ENC3 process are 64 bit blocks. For a further decomposition of the ENC3 functionality the reader should refer to Section 5.4.1.2 and Figure 5-51.

The initial input to the chain of ENC3 processes is the ‘Seed’. It is a random value given by the key header #7 (see sect. 4.3.2.8) for each LRIT/HRIT file. The key used for the PN pattern generation is the so-called Pseudo-Noise Key (PNK) which is a stable value for one LRIT/HRIT file.

The PNK is a result of a station key unit (SKU) process as described in Section 5.4.3.

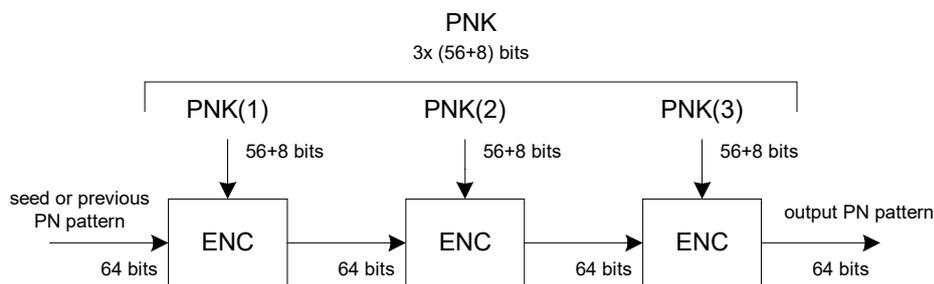
The LRIT/HRIT data field is byte aligned. As the PN pattern consists of blocks of 64 bit, a part of the last PN pattern block may remain unused. The number of ENC3 processes to be performed per LRIT/HRIT file varies with its data field length.

Figure 5-50 shows the principle of PN pattern generation and its application to the LRIT/HRIT data field.



**Figure 5-50 – PN pattern generation and application to LRIT/HRIT data field**

One ENC3 process in Figure 5-50 can be broken down to three consecutive single ENC processes as shown in Figure 5-51.



**Figure 5-51 – Single ENC3 PN Pattern Process**

The bit-wise modulo 2 addition (without carry) of the 64 bit output PN pattern has to be performed aligned to 64 bit blocks of the LRIT/HRIT data field in a way that the MSB of the output PN pattern (DES bit 1) is applied to the MSB (first transmitted one) of the LRIT/HRIT data field block.

The PN pattern process will be newly started for each LRIT/HRIT file. If the LRIT/HRIT data field is not a multiple of 64, any unused PN pattern shall be discarded.

### 5.4.3 Station Key Unit Functionality

The Station Key Unit (SKU) is a hardware device containing a microcontroller. The SKU will be programmed by the EUMETSAT KMC and will be given a unique Master Station Key (MSK) and station number before dispatch to the end-user.

The purpose of the Station Key Unit (SKU) is

- To safely store the MSK and protect it against read-out
- To integrity check and store the station PBKs as received via the dissemination
- To generate the PNK based on the PBK selected by Key\_Number and Seed

The DADF and the each MSG User Station has to interface to an SKU. The logical, electrical and mechanical SKU interfaces are defined in [RD.16]. For a data flow example for communication between a user station and an SKU the reader should refer to Appendix C.6.

Encryption Key Messages (file type #3, see sect. 4.2.5) consist of records of key numbers, station numbers and Public Keys (PBK). The PBKs are accompanied by a CRC field to allow for integrity checks. Encryption Key Messages are frequently made available via dissemination to the user stations.

Input to the SKU will be a Public Key (PBK), the corresponding CRC and the Seed.

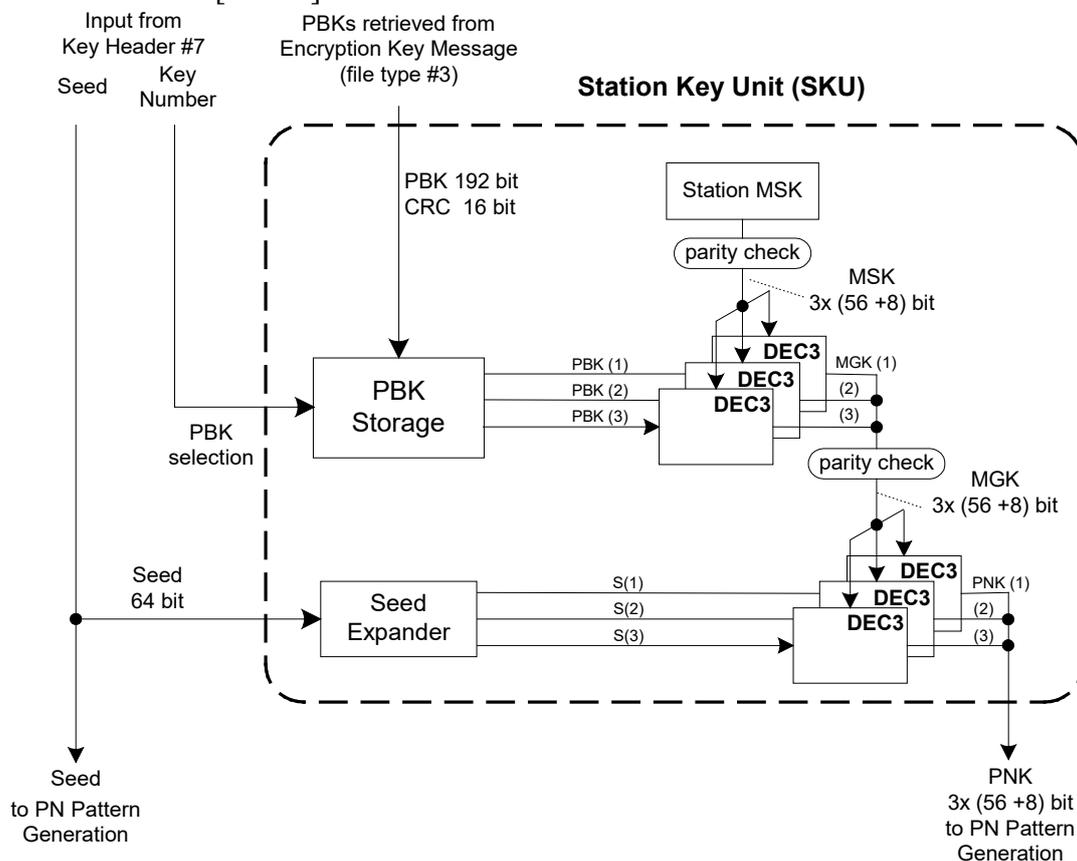
The SKU expands the Seed internally to a 192 bit value by the algorithm defined in Appendix C.3.

The MGK is an intermediate product within the SKU and will never appear at any output. The derivation of the PNK with the help of the SKU is identical for the DADF on the transmission side and the LRUS/HRUS on the receiving side (see Figure 5-52).

The following procedures have to be applied to reach an identical PNK at the DADF and the User Station side:

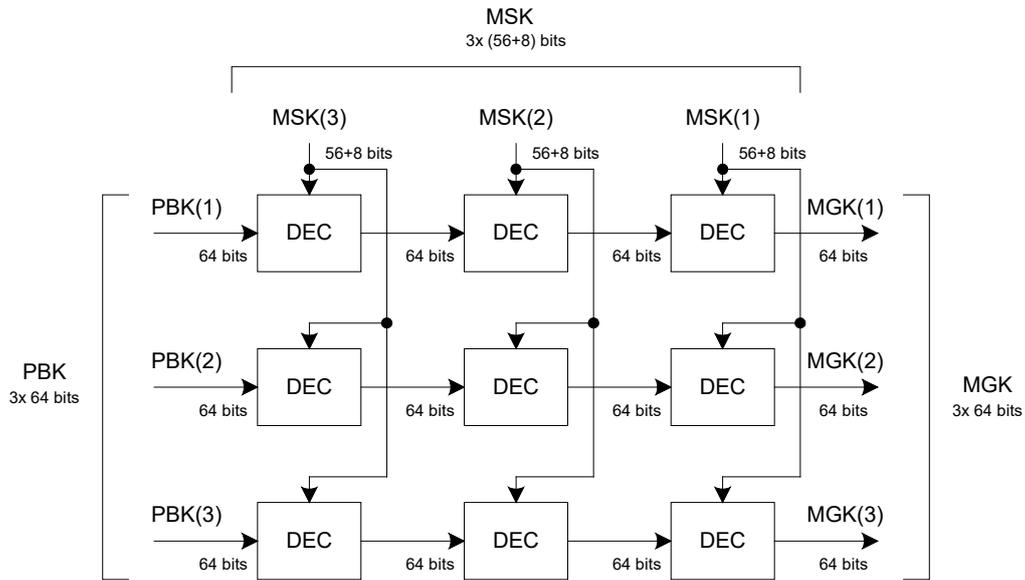
1. Retrieve the PBKs from the disseminated Encryption Key Messages (file type #3) with the relevant station number
2. Store the PBKs in the SKU (optional)
3. Enter the ‘expanded’ seed (‘original’ seed with SKU internal expansion optional) into the SKU
4. Retrieve the PNK from the SKU

The logical SKU interface to the DADF encryption functionality and/or the LRUS/HRUS MUBM is defined in [RD.16].



**Figure 5-52 – Station Key Unit (functional block diagram)**

The functionality of three times DEC3 as shown in Figure 5-52 can further be broken down as depicted in Figure 5-53. Each ‘DEC’ box represents a single DES decryption process.



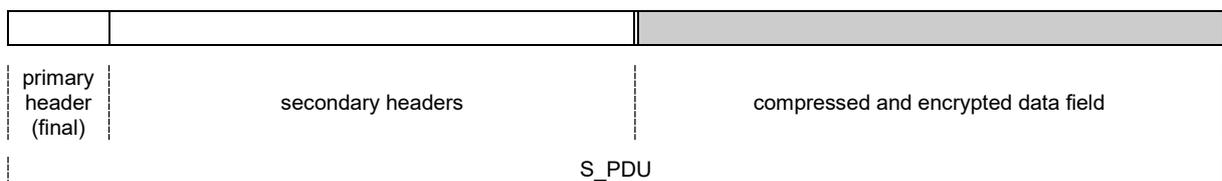
**Figure 5-53 – SKU triple DEC3 Functionality**

**5.5 Session Layer Output**

Output of the session layer to the transport layer is the session protocol data unit (S\_PDU) containing the variable length compressed and encrypted data field as shown in Figure 5-54.

If neither compression nor encryption has been applied, the session layer will leave the data field unmodified.

In any case the session layer processing will have to determine the data field length and fill into the primary header and to add the time stamp (if required) before it passes the complete data unit as S\_PDU to the transport layer.



**Figure 5-54 – LRIT/HRIT Session Protocol Data Unit (S\_PDU)**

The parameter PRIO as defined in [AD.1] will not be used. As an alternative the MSG DADF will implement buffering and sequencing mechanisms on LRIT/HRIT file level between the session and the transport layer to guarantee the specified product type timeliness requirements towards the end-user.

## 6 TRANSPORT LAYER

### 6.1 General

The transport layer receives the S\_PDUs as a transport service data unit (TP\_SDU) which is a variable length file as shown in Figure 5-54. The TP\_SDUs will be put into octet-aligned transport files, if such alignment has not already been performed previously.

The file counter within the transport header of the transport file structure (see [AD.1]) will restart from '0' after configuration changes (e.g. chain switch, S/W unit restart) in the Dissemination Element.

The transport files are split into one or more blocks of 8190 bytes size which form the user data field of the source packet. The last block may be shorter and contain 1 ... 8190 bytes only. Each user data field will be followed by a 2 octet Cyclic Redundancy Check (CRC).

### 6.2 Source Packetisation

#### 6.2.1 Source Packet Structure

This section defines in detail the source packet structure:

Source Packet Header (48 bits)							Packet Data Field (variable)	
Packet Identification				Packet Sequence Control		Packet length	User Data Field	
Version No	Type	Secondary Header Flag	APID	Sequence Flags	Packet Sequence Count		Application Data Field	Packet Error Control (CRC)
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits	16 bits	Variable	16 bits
2 octets				2 octets		2 octets	max. 8190 octets	2 octets

**Figure 6-1 – Source Packet Structure (TP\_PDU)**

**Version No** The version No. bits will be set to '000'b, identifying the Version-1 CCSDS packet.

**Type** Set to '0'b. Type is not used in CCSDS AOS.

**Secondary header flag** Set to '0'b. A Secondary Header is not used in the LRIT/HRIT Dissemination Service.

**APID** see Table 6-1.

Application Process Identifier (APID)		Application Names (in accordance with sect. 4.3.2.5)
LRIT	HRIT	
'0'	'32'	'MSGi' data
'1'	'33'	'DCP' data
'2'	'34'	'GTS' data
'3'	'35'	'MPEF' data
'4'	'36'	'SAF' data
'5'	'37'	'SERVICE' data
'6'	'38'	'MTPi' data
'7'	'39'	'METOPi' data
'8'	'40'	'GOESi' data
'9'	'41'	'GMSi' or 'MTSATi' data
'10'	'42'	'GOMSi' data
'11'	'43'	'NOAAi' data
'12' ... '30'	'44' ... '62'	for future extensions
'31'	63	Test Packet as defined in sect. 6.2.2
'64' ... '2031'		for future extensions
'2032' ... 2046'		Reserved by CCSDS (not be used for LRIT/HRIT)
'2047'		Fill Packets

**Table 6-1 – Application Process Identifiers**

Note: the above table assumes that all LRIT and all HRIT data are forwarded with an identical PRIO according to [AD.1]. The APIDs will not be distributed independent of the application data contents.

**Sequence Flag** as defined in [AD.1]

**Packet Sequence Count** 14-bit Packet Sequence Count, straight sequential count (modulo 16384) which number each source packet generated per APID. The Packet Sequence Count will restart from '0' after configuration changes in the Dissemination Element (e.g. chain switch, S/W unit restart).

**Packet Length** 16-bit binary count which expresses the length of the remainder of the source packet following this field.

**Application Data Field** contains up to 8190 octets of user data, i.e. a block of the TP\_SDU

**Packet Error Control** The 16 bit CRC forms the trailer of the user data field. It has to be derived as defined in [AD.1].

The CRC is computed over the entire application data field. The generator polynomial is

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

The encoder shall be initialised to 'all ones' for each application data field.

### 6.2.2 Test Packet Generation

In the case no user data is available or for test purposes, the transport layer processing of the MSG DADF can be configured to generate ‘full size’ test packets. The particular definitions of a fill source packet are:

<b>Version</b>	‘000’b
<b>Type</b>	‘0’b
<b>Secondary Header Flag</b>	‘0’b
<b>APID</b>	‘31’ for LRIT, ‘63’ for HRIT
<b>Sequence Flag</b>	‘11’b (unsegmented)
<b>Packet length</b>	8190 octets
<b>User Data Field</b>	‘all zeros’
<b>Packet Error Control</b>	as specified

### 6.3 Transport Layer Output

The transport layer output is the protocol data unit TP\_PDU which is identical to the source packet as depicted in Figure 6-1.

## **7 NETWORK LAYER**

### **7.1 Input to Network Layer**

The source packets as shown in Figure 6-1 are the CCSDS path service data units (CP\_SDU) forming the input to the Network Layer.

### **7.2 General**

The Network Layer represents the CCSDS AOS path layer. The only function to be provided with respect to the LRIT/HRIT dissemination service is the generation of a correct Virtual Channel Identifier (VC-ID). Otherwise the data received from the transport layer is transparently routed to the Data Link Layer.

### **7.3 Network Layer Processing**

Because the MSG LRIT/HRIT files have been properly sequenced on a higher layer, no real multiplexing will be performed in the data link layer. The MSG dissemination scheme will not make use of the underlying priorities of different VCs. Therefore, the determination of the VC-ID is kept as simple as possible.

The used VC-IDs are:

All LRIT data (using APIDs '0' ... '31') are mapped to VC 0.

All HRIT data (using APIDs '32' ... '63') are mapped to VC 1.

### **7.4 Output of Network Layer**

The CCSDS path protocol data unit (CP\_PDU) is identical to the initial CP\_SDU. The CP\_PDU will be forwarded to the Data Link Layer together with the determined VC-ID.

## 8 DATA LINK LAYER

### 8.1 Input to Data Link Layer

The Network Layer provides the CP\_PDUs as multiplexing service data units (M\_SDU) to the data link layer.

### 8.2 General

The Data Link Layer encompasses functionality of the CCSDS space link layer with its two sublayers:

- Virtual channel link control (VCLC) sublayer
- Virtual channel access (VCA) sublayer

As described in sect. 8.3 the VCLC sublayer processing provides the multiplexing service only. This includes filling of M\_SDUs into multiplexing protocol data units (M\_PDU). Fill packets may have to be generated for the completion of the M\_PDUs after time-out expiration.

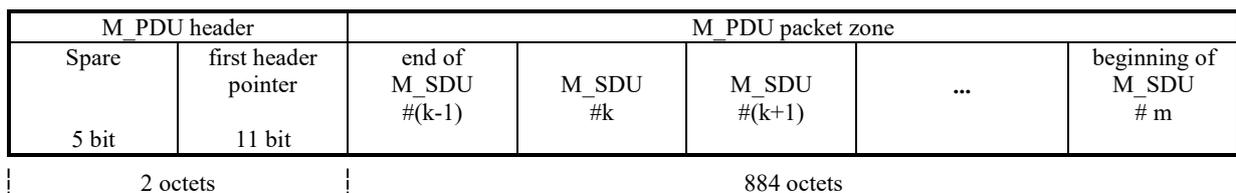
The VCA sublayer generates the virtual channel data units (VCDU), performs Reed-Solomon coding, data randomisation and attachment of synchronisation markers. The VCDU generation is still a functionality of the DADF, while after transmission of the VCDUs via a dedicated communication link the MSG PGS will be responsible for the final VCA sublayer processing. The PGS will generate ‘fill VCDUs as specified in sect. 8.3.2 to maintain continuous data delivery to the physical layer.

### 8.3 VCLC Sublayer Processing

The VCLC sublayer processing performs the multiplexing, the M\_PDU generation and the related fill packet generation in accordance with [AD.1].

*(Note: the fill packet generation in the VCLC Sublayer description of [AD.1] contains a mistake about the sizing of the packet in case M\_PDU incompleteness. The ‘remaining spare’ must be less than ‘seven’ instead of ‘eight’.)*

The M\_PDUs will consist of 886 octets of which 2 octets are the M\_PDU header and the 884 octets are the M\_PDU packet zone as shown in Figure 8-1.



**Figure 8-1 – M\_PDU Structure**

The M\_PDUs are then passed to the VCA sublayer service.

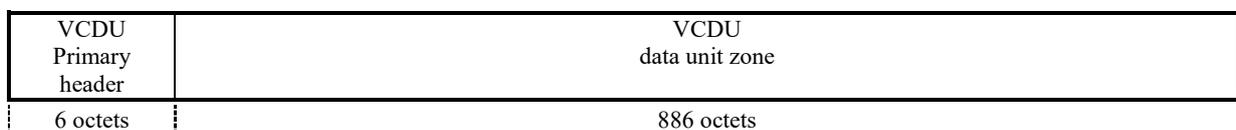
## 8.4 VCA Sublayer Processing

### 8.4.1 VCDU Assembly

The M\_PDUs from the VCLC layer are received as VCA\_SDUs from the VCLC sublayer and are used to assemble virtual channel data units (VCDU) according to [AD.1].

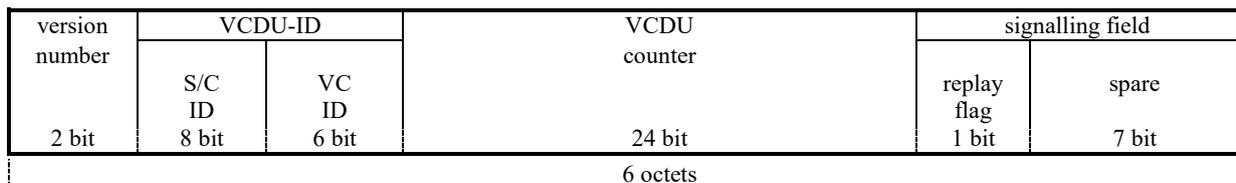
The VCDUs form the application data units which are transferred from the DADF to the PGS via a dedicated communication link (see also Section 2.1).

The VCDU structure is shown in Figure 8-2.



**Figure 8-2 – VCDU structure**

The decomposition of the VCDU header is given in Figure 8-3.



**Figure 8-3 – VCDU Primary Header**

#### Mission specific use:

**Version Number** '01'b

**VCDU-ID** The S/C IDs represent the disseminating spacecraft.

The following S/C IDs will be used:

MSG-1	'41'h
MSG-2	'42'h
MSG-3	'43'h
MSG-4	'44'h

The VC ID are as specified in sect. 7.3.

**VCDU Counter** as defined in [AD.1]  
 The VCDU Counter will restart from '0' after configuration changes in the Dissemination Element (e.g. chain switch, S/W unit restart).

**Signalling Field** 'all zeros'

### 8.4.2 ‘Fill VCDU’ Generation

Note: this is the first functionality being covered by the MSG PGS.

The VCA sublayer processing at the MSG PGS will automatically generate a ‘fill VCDU’ in the case no or not sufficient VCDUs (underflow condition) are received from the DADF via the dedicated communication link to maintain a continuous data flow at the specified packetised data rate to the physical layer.

The definition of a ‘Fill VCDU’ is:

<b>Version</b>	‘01’b	
<b>VCDU-ID</b>	S/C ID	depending on used S/C for dissemination (see list in Section 8.4.1)
	VC ID	63 (‘all ones’)
<b>VCDU Counter</b>	as defined in [AD.1]	
<b>VCDU Data Unit Zone</b>	fill pattern ‘all zeros’	

After ‘Fill VCDU’ generation the following constant ‘packetised data rate’ for the LRIT/HRIT dissemination channels will be achieved:

LRIT	128,000 bps
HRIT	1,000,000 bps

*(Note: the ‘packetised data rate’ is defined as the data stream after formatting and packetisation, and before FEC coding and transmission. More precisely, it is the data at the VCDU level excluding sync marker and R-S check symbols.)*

### 8.4.3 Reed-Solomon Coding

The LRIT/HRIT dissemination service is a Grade-2 service, therefore, the transmission of user data will be error controlled using Reed-Solomon coding as an outer code.

The used Reed-Solomon code is (255, 223) with an interleaving of  $I = 4$  according to [RD.14].

The VCDUs will be attached by 128 octets of Reed-Solomon check symbols to form a coded VCDU (C-VCDU).

VCDU primary header	VCDU data unit zone	Reed-Solomon Check Symbols
6 octets	886 octets	128 octets

**Figure 8-4 – C-VCDU Structure**

#### **8.4.4 Randomisation**

Randomisation is applied to all LRIT/HRIT C-VCDUs. It is a process by which a pseudo-random sequence is bitwise exclusive-ORed to all 8160 bits of the C-VCDU to ensure sufficient data transitions.

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

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

This sequence begins at the first bit of the C-VCDU and repeats after 255 bits, continuing repeatedly until the end of the C-VCDU. The sequence generator is then re-initialised to an all-ones state for the processing of the next C-VCDU.

The 255 bits of the pseudo-random sequence from the generator are shown below; the left-most bit is the first bit of the sequence to be exclusive-ORed with the first bit of the C-VCDU; the second bit of the sequence is exclusive-ORed with the second bit of the C-VCDU, and so on.

```
1111 1111 0100 1000 0000 1110 1100 0000 1001 1010 0000 1101 0111 0000 1011 1100
1000 1110 0010 1100 1001 0011 1010 1101 1010 0111 1011 0111 0100 0110 1100 1110
0101 1010 1001 0111 0111 1101 1100 1100 0011 0010 1010 0010 1011 1111 0011 1110
0000 1010 0001 0000 1111 0001 1000 1000 1001 0100 1100 1101 1110 1010 1011 000
and so on ...
```

For further information the reader shall refer to [RD.14].

#### **8.4.5 Sync Marker Attachment**

An attached synchronisation marker (ASM) will have to precede the randomised C-VCDU to allow for frame synchronisation. The 32 bit pattern can be represented in hexadecimal notation as:

‘1ACFFC1D’h

The ASM and together with the C-VCDU create the channel access data unit (CADU) of 1024 octets length.

#### **8.4.6 Serialisation and Output of the Data Link Layer**

As a final task the VCA sublayer performs the serialisation of the CADU and provides the serial bitstream to the physical layer.

## **9 PHYSICAL LAYER**

The physical layer w.r.t. the LRIT/HRIT transmission service performs the convolutional coding of the serialised data stream and its modulation onto the RF uplink signal.

The modulation schemes, the applied baseband filtering and the physical channels used are different for LRIT and HRIT. The RF uplink signals are received by the MSG on-board transponder and transmitted to the user stations via L-band downlinks. Various technological and propagation effects influence the signal properties. The complete parameter sets of the physical layer including the coverage areas of the LRIT/HRIT dissemination services are specified in Appendix E.

## APPENDIX A      APPLICATION      DATA      UNIT      DEFINITION      OF MISCELLANEOUS LRIT/HRIT DATA & PRODUCTS

The subsections of this appendix will define the product/data structure of the following application data units:

- Foreign Satellite Data
- GTS Data & Products
- Compression Test Message
- Encryption Test Message
- MPEF Products
- DCP Messages

### A.1      Foreign Satellite Data via LRIT/HRIT

#### A.1.1      FSD Overview

The following subsections provide further details on the Repeat Cycle Prologues and LRIT/HRIT header structure/parameter values of the Foreign Satellite Data.

#### A.1.2      Definition of FSD Repeat Cycle Prologue Data Fields

The Repeat Cycle Prologue data field (file type #128) for the Foreign Satellite Data will contain:

- An SGS\_Common\_Header and
- An SGS\_Product\_Specific\_Header record.

<pre> REPEAT_CYCLE_PROLOGUE ::= RECORD {   SGS_Common_Header           Common_Header   SGS_Product_Specific_Header CHOICE                                { Image_Product_Specific_Header,                                NonImage_Product_Specific_Header }         </pre>
--

The SGS\_Common\_Header is defined in Appendix A.1.2.1.

The SGS\_Product\_Specific\_Header can be either an Image\_Product\_Specific\_Header or a NonImage\_Product\_Specific\_Header. Currently, only Image\_Product\_Specific\_Headers are defined and are given in the tables and parameter descriptions below. They contain housekeeping information directly taken from the relevant foreign satellite's dissemination. Whenever necessary, references to other documentation are made.

Non-Imagery FSD Products are currently not part of the MSG LRIT/HRIT dissemination baseline.

### A.1.2.1 SGS\_Common\_Header

The Common\_Header record is defined as follows:

```
Common_Header ::= RECORD
```

<b>CommonHeaderVersion</b>	UNSIGNED BYTE (0),
<b>Pad1</b>	CHARACTERSTRING SIZE (3),
<b>NominalSGSProductTime</b>	TIME CDS SHORT,
<b>SGSProductQuality</b>	UNSIGNED BYTE (0..100),
<b>SGSProductCompleteness</b>	UNSIGNED BYTE (0..100),
<b>SGSProductTimeliness</b>	UNSIGNED BYTE (0..100),
<b>SGSProcessingInstanceId</b>	UNSIGNED BYTE,
<b>BaseAlgorithmVersion</b>	OCTETSTRING SIZE (16),
<b>ProductAlgorithmVersion</b>	OCTETSTRING SIZE (16)
}	

**CommonHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

**Pad1** contains characters with an ASCII value of zero used for alignment of the data on word boundaries.

**NominalSGSProductTime** represents the nominal product time.

**SGSProductQuality** is a measure of the overall reliability of the product.

**SGSProductCompleteness** is a measure of the coverage area of the product relative to the nominal coverage.

**SGSProductTimeliness** is a measure of the time of availability of the product relative to the nominal availability time.

**SGSProcessingInstanceId** identifies the processing instance which has been used to derive the product.

**BaseAlgorithmVersion:** for level 1 product, this is the version of the algorithm which produced it. For higher level product, this is the version of the algorithm that produced the level 1x product up which the present product is based.

**ProductAlgorithmVersion** identifies the version of the algorithm used to produce the present product. It is applicable only to level 2 or higher products.

### A.1.2.2 SGS\_Product\_Specific\_Header

The SGS\_Product Specific\_Header is part of the Repeat Cycle Prologue data field as defined in Appendix A.1.2.

### A.1.2.2.1 Image\_Product\_Specific\_Header

Image_Product_Specific_Header ::= RECORD	
{ImageProductSpecificHeaderVersion	UNSIGNED BYTE (0),
Pad1	CHARACTERSTRING SIZE (3),
Image_Product_Specific_Header_Length	UNSIGNED,
Image_Product_Specific_Data	CHOICE
{GOES	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	OCTETSTRING SIZE (16080)},
GMS/MTSAT	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	OCTETSTRING SIZE (22730)},
GOMS	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	OCTETSTRING SIZE (80)},
MTP	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	CHOICE
	{VIS OCTETSTRING SIZE (192999),
	IR OCTETSTRING SIZE (144515),
	WV OCTETSTRING SIZE (144515)}}}
Single_HRPT	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	Single_HRPT_Record},
HRPT_Composite	RECORD
{Version	UNSIGNED BYTE (0),
HeaderData	HRPT_Composite_Record}
}	
}	

**ImageProductSpecificHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

**Pad1** contains characters with an ASCII value of zero used for alignment of the data on word boundaries.

**Image\_Product\_Specific\_Header\_Length** contains the full length of the record.

**Version** is set to zero initially and is used to identify possible future upgrades of this record.

**HeaderData** contains information as specified below:

- For **GOES** it contains two sets of a “block 0 data section”. The first “block 0 data section” contains information at the start of the imager scan and the second “block 0 data section” contains information at the end of the scan. Each “block 0 data section” consists of 8040 bytes which are structured into six partitions containing
  1. Instrument and scan status;
  2. Instrument and attitude data;

3. Scan reference data;
4. Grid data;
5. Scan reference and calibration data;
6. Factory parameters;

The detailed structure of a “block 0 data section” is described in [RD.17, Section 3.3.4].

- For GMS/MTSAT, it contains a combination of information extracted from the documentation sectors of the GMS Stretched Visible and Infrared Spin Scan Radiometer (S-VISSR). The documentation sector is the first part of the information sectors within the S-VISSR data format. It is structured as follows:
  1. S/C and CDAS Status Block of the scan start           126 bytes
  2. Simplified Mapping Block 1 of scan start           64 bytes
  3. S/C and CDAS Status Block of the scan end       126 bytes
  4. Simplified Mapping Block 1 of scan end           64 bytes
  5. Simplified Mapping Block 2                       2500 bytes
  6. Orbit and Attitude Data Block                   3200 bytes
  7. Manual Amendment (MANAM) Block           10250 bytes
  8. Calibration Block                               6400 bytes

The detailed structure of the data blocks is described in [RD.7], [RD.8] and [RD.9].

The detailed structure of the above-mentioned synchronisation and telemetry records is described in [RD.18].

- For MTP, it consists of record 2 of the OpenMTP file format. This format is defined in Section 4.2 of [RD.24].

#### **A.1.2.2.2 NonImage\_Product\_Specific\_Header**

The NonImage\_Product\_Specific\_Header is defined as follows:

NonImage_Product_Specific_Header ::= RECORD	
{NonImageProductSpecificHeaderVersion	UNSIGNED BYTE (0),
Pad1	CHARACTERSTRING SIZE (3),
NonImage_Product_Specific_Header_Length	UNSIGNED,
NonImage_Product_Specific_Data	NULL
}	

**NonImageProductSpecificHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

**Pad1** contains characters with an ASCII value of zero used for alignment of the data on word boundaries.

**NonImage\_Product\_Specific\_Header\_Length** contains the full length of the record.

**NonImage\_Product\_Specific\_Data** contains the full SAF or SGS satellite data header for non-imagery products. Presently, there is no such data header defined.

### A.1.3 Example of an FSD Image Data Function (Header #3)

For FSD, the header type #3 will be used to establish a relationship between pixel values and physical units, e.g. to define a representation of calibrated image data. **Data\_Definition\_Block** (see Section 4.3.2.4) uses the self-descriptive language defined in [AD.1].

An example of a **Data\_Definition\_Block** for an infrared type of data calibration is given below:

```

$HALFTONE:=8<CR>>LF>
_NAME:=calibrated infrared<CR><LF>
_UNIT:=degree Kelvin<CR><LF>
0:=163<CR><LF>
79:=242<CR><LF>
255:=330<CR><LF>
    
```

### A.1.4 FSD Spectral Channel Identifiers

For FSD, the **Spectral\_Channel\_ID** value used in the Segment Identification header type #128 (see Section 4.3.2.9) is defined as follows:

<b>Spectral_Channel_ID</b>	<b>CHOICE</b>	<b>{GOES</b>	<b>ENUMERATED BYTE</b>		
			{VIS	00-7	(1),
		IR	03-9	(2),	
		WV	06-6	(6),	
		WV	06-8	(3),	
		IR	10-7	(4),	
		IR	12-0	(5)	
		IR	13-4	(7)},	
		<b>MTSAT</b>	<b>ENUMERATED BYTE</b>		
		{VIS	00-7	(1),	
		IR	03-8	(2),	
		WV	06-7	(3),	
		WV	06-8	(6),	
		IR	10-7	(7),	
		IR	10-8	(4),	
		IR	12-0	(5)},	
		<b>MTP</b>	<b>ENUMERATED BYTE</b>		
		{VIS	00-7	(1),	
		IR	11-5	(2),	
		WV	06-4	(3)},	
				}	

## A.2 GTS Data & Products via LRIT/HRIT

The MSG dissemination day 1 scenario for GTS Data & Products via LRIT/HRIT aims at the distribution of GTS data sets being identical to the MTP Meteorological Data Distribution (MDD) mission at the time of the start of MSG operations.

The MDD content is defined by a product schedule. The MDD product schedule is regularly reviewed. For the latest schedule available schedule the reader should refer to [RD.20]. The application data unit structure of the GTS Data & Products is defined in Section 4.2.3, [AD.1].

### **A.3 Compression Test Message Pattern**

The Compression “Test Message Pattern” would be used during the operational phase of the dissemination mission to as a means to check the compression function. This pattern has not been defined. It will be defined and announced as needed.

### **A.4 Encryption Test Message Pattern**

The Encryption Test Pattern would be used during the operational phase of the Dissemination Mission. This pattern has not been defined. It will be defined and announced as needed

### **A.5 MPEF Products via LRIT/HRIT**

The MPEF products described in the next subsections form a “super-set” for dissemination. This means, that not all of these products might appear in the operational dissemination schedules. But considering that this current baseline may be not the final one, these products may be reinserted later, and thus the respective descriptions are maintained.

#### **A.5.1 Data\_Definition\_Block for MPEF Products**

**Data\_Definition\_Block** is a record of the Image Data Function Header #3 as defined in Section 4.3.2.4.

For all MPEF imagery products (see list in Section A.5.1), the size of the character string is constant. Its length is 1024 characters.

#### **A.5.2 MPEF Product Overview**

The following subsections provide further details on the Repeat Cycle Prologues and LRIT/HRIT header structure/parameter values for the following MPEF products:

- 1) MPEF products (imagery type) disseminated as file type #0 (image data):  
Currently none
  
- 2) MPEF products disseminated as GTS message file type #1 (GTS Message):
  - AMV Atmospheric Motion Vector
  - CLA Cloud Analysis
  - CSR Clear Sky Radiance
  - GII Global Instability Index
  - RII Regional Instability Index
  - TH Tropospheric Humidity
  - TOZ Total Ozone

3) MPEF products disseminated as file type #144 (binary type):

ASR	All Sky Radiances
CLM	Cloud Mask
CLAI	Cloud Analysis Image
CTH	Cloud Top Height
CRM	Clear Sky Reflectance Map
DIV	Divergence
FIRC	Active Fire Monitoring CAP
FIRG	Active Fire Monitoring GRIB
MPEG	Multi-Sensor Precipitation Estimate GRIB
NDVI	Normalised Difference Vegetation Index
NDVD	Normalised Difference Vegetation Index Decadal
OCAE	Optimal Cloud Analysis
VOLC	Volcanic Ash Detection CAP
VOLE	Volcanic Ash Detection GRIB

### A.5.3 Repeat Cycle Prologue Definition for MPEF Products

The data field of the LRIT/HRIT file type #128 for MPEF products will contain the REPEAT\_CYCLE\_PROLOGUE record which is defined as follows:

REPEAT_CYCLE_PROLOGUE	::=	RECORD	
{	MPEF_Product_Header	RECORD	
	MPEF_Product_Specific_Header	CHOICE	
	{	AMV_Header	RECORD,
		CLA_Header	RECORD,
		CLAI_Header	RECORD,
		CLM_Header	RECORD,
		CSR_Header	RECORD,
		CRM_Header	RECORD,
		CTH_Header	RECORD,
		DIV_Header	RECORD,
		FIRC_Header	RECORD,
		FIRG_Header	RECORD,
		GII_Header	RECORD,
		MPEG_Header	RECORD,
		NDVI_Header	RECORD,
		NDVD_Header	RECORD,
		OCAE_Header	RECORD,
		RII_Header	RECORD,
		TH_Header	RECORD,
		TOZ_Header	RECORD,
		VOLC_Header	RECORD,
		VOLE_Header	RECORD }
	}		
}			

#### A.5.3.1 MPEF\_Product\_Header Record

The MPEF\_Product\_Header is defined as:

MPEF_Product_Header::=	RECORD
------------------------	--------

<b>{MPEF_File_Id</b>	INTEGER SHORT,	
<b>MPEF_header_version</b>	UNSIGNED BYTE (0),	
<b>ManualDissAuthRequested</b>	BOOLEAN BYTE,	
<b>ManualDisseminationAuth</b>	BOOLEAN BYTE,	
<b>DisseminationAuth</b>	BOOLEAN BYTE,	
<b>NominalTime</b>	TIME CDS SHORT,	
<b>ProductQuality</b>	UNSIGNED BYTE (0..100),	
<b>ProductCompleteness</b>	UNSIGNED BYTE (0..100),	
<b>ProductTimeliness</b>	UNSIGNED BYTE (0..100),	
<b>InstanceID</b>	ENUMERATED	BYTE
<b>ImagesUsed</b>	ImageDetails,	
<b>BaseAlgorithmVersion</b>	GP_CONFIG_ITEM_VERSION,	
<b>ProductAlgorithmVersion</b>	GP_CONFIG_ITEM_VERSION,	
<b>Filler</b>	CHARACTERSTRING SIZE(52)	
}		

**MPEF\_File\_Id** uniquely identifies the MPEF file type.

**MPEF\_header\_version** is set to zero initially and is used to identify possible future upgrades of this record.

**ManualDissAuthRequested** indicates whether the product requires manual dissemination authorisation.

**ManualDisseminationAuth** indicates whether dissemination authorisation has been given for products that require manual authorisation.

**DisseminationAuth** indicates whether dissemination authorisation has been given for products that do not require manual authorisation.

**NominalTime** Nominal Schedule Time taken from activity.

**ProductQuality** is a measure of the overall reliability of the product. It is measured between 0 and 100.

**ProductCompleteness** is a measure of the coverage area of the product relative to the nominal coverage. It is measured between 0 and 100.

**ProductTimeliness** is a measure of the time of availability of the product relative to the nominal availability time. It is measured between 0 and 100.

**InstanceId** identifies the processing instance that has been used to derive the product.

**ImagesUsed** identifies which repeat cycles have contributed to the product.

**BaseAlgorithmVersion** represents the version of the algorithm that produced the level 1.5 image up which the present product is based. It is taken from the **OverallConfiguration** field in the IMPFConfiguration record 1.5 image header.

**ProductAlgorithmVersion** identifies the version of the algorithm used to produce the product. It is taken from the SU version of the SU used to produce the product.

**Filler** contains characters with an ASCII value of zero and represents reserved space for later additions to the record.

ImageDetails is defined as:

ImageDetails::=	ARRAY SIZE (1..4) OF RECORD
{ <b>Pad1</b>	CHARACTERSTRING SIZE(2),
<b>ExpectedImageStart</b>	TIME CDS SHORT,
<b>ImageReceivedFlag</b>	BOOLEAN BYTE,
<b>Pad2</b>	CHARACTERSTRING SIZE(1),
<b>UsedImageStart</b>	TIME CDS SHORT,
<b>Pad3</b>	CHARACTERSTRING SIZE(2),
<b>UsedImageEnd</b>	TIME CDS SHORT,
}	
-- for ,, AMV, TOZ and GII, all 4 elements will be used.	
-- for other products, only the first element will be used; the last 3 being filled -- with zeros.	

**Pad1**, **Pad2**, and **Pad3** contain characters with an ASCII value of zero and represent padding to align the data.

**ExpectedImageStart** is the time of the expected image to be used to generate the product and is set to zero if the activity was scheduled to use the nearest repeat cycle.

**ImageReceivedFlag** is set to TRUE if the expected image was received. It is always false if the activity was scheduled to use the nearest repeat cycle.

**UsedImageStart** represents the start time of the image used to extract the product. It is taken from the TrueDateofRepeatCycleStart field in the ImageAcquisition record of the level 1.5 data.

**UsedImageEnd** represents the planned end time of the image used to extract the product. It is taken from the **PlannedForwardScanEnd** field in the ImageAcquisition record of the level 1.5 data.

### A.5.3.2 MPEF\_Product\_Specific\_Header

The MPEF\_Product\_Specific\_Header is a choice of the MPEF product name. The various header record definitions are given in the following subsections.

#### A.5.3.2.1 AMV\_Header

AMV\_Header is defined as:

{ <b>ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Pad1</b>	CHARACTERSTRING SIZE(3),

<b>ProcessingSegmentWidth</b>	UNSIGNED BYTE,
<b>ProcessingSegmentHeight</b>	UNSIGNED BYTE,
<b>NoVectorsInProduct</b>	UNSIGNED SHORT,
<b>NoVectorsPerBand</b>	ARRAY SIZE (1..12) OF UNSIGNED SHORT,
<b>NoVectorsPassAQC</b>	UNSIGNED SHORT,
<b>NoVectorsPassPerBand</b>	ARRAY SIZE (1..12) OF UNSIGNED SHORT,
<b>ProductVerified</b>	BOOLEAN BYTE,
<b>Pad2</b>	CHARACTERSTRING SIZE (3),
<b>NoGTSMessages</b>	UNSIGNED SHORT,
<b>MessageSizes</b>	UNSIGNED SHORT,
<b>Filler</b>	CHARACTERSTRING SIZE (96)
}	

**ProcessingSegmentWidth** represents the east/west size in pixels of each processing segment.

**ProcessingSegmentHeight** represents the north/south size in pixels of each processing segment.

**NoVectorsInProduct** represents the number of derived vectors from all spectral bands in the product.

**NoVectorsPerBand** represents the number of derived vectors per spectral band in the product.

**NoVectorsPassAQC** represents the number of derived vectors from all spectral bands having a quality better than the automatic quality control threshold.

**NoVectorsPassPerBand** represents the number of derived vectors per spectral band having a quality better than the automatic quality control threshold.

**ProductVerified** indicates whether the product has been verified against observed and forecast winds. Note that for all near-real-time AMV products disseminated from MPEF to DADF this field will be set to FALSE.

#### A.5.3.2.2 ASR\_Header

ASR\_Header is defined as:

{ <b>ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>ProcessingSegmentWidth</b>	UNSIGNED BYTE,
<b>ProcessingSegmentHeight</b>	UNSIGNED BYTE,
<b>Pad1</b>	CHARACTERSTRING SIZE(1),
<b>NoSegmentsInProduct</b>	UNSIGNED SHORT,
<b>NoGTSMessages</b>	UNSIGNED SHORT,
<b>MessageSizes</b>	UNSIGNED SHORT,
<b>Filler</b>	CHARACTERSTRING SIZE(96)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.3 CLA\_Header

The CLA\_Header record is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,	<b>ProcessingSegmentWidth</b>	UNSIGNED BYTE,
<b>ProcessingSegmentHeight</b>	UNSIGNED BYTE,		
<b>Pad1</b>	CHARACTERSTRING SIZE (1),		
<b>NoSegmentsInProduct</b>	UNSIGNED SHORT,		
<b>NoGTSMessages</b>	UNSIGNED SHORT,		
<b>MessageSizes</b>	UNSIGNED SHORT,		
<b>Filler</b>	CHARACTERSTRING SIZE (96)		
}			

**CLAProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.4 CLAI\_Header

CLAI\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**CLAIProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.5 CLM\_Header

CLM\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.6 CRM\_Header

CRM\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.7 CSR\_Header

The CSR\_Header record is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>ProcessingSegmentWidth</b>	UNSIGNED BYTE,
<b>ProcessingSegmentHeight</b>	UNSIGNED BYTE,
<b>Pad1</b>	CHARACTERSTRING SIZE(1),
<b>NoSegmentsInProduct</b>	UNSIGNED SHORT,
<b>NoGTSMessages</b>	UNSIGNED SHORT,
<b>MessageSizes</b>	UNSIGNED SHORT,
<b>Filler</b>	CHARACTERSTRING SIZE (96)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.8 CTH\_Header

CTH\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.9 DIV\_Header

DIV\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.10 FIRC\_Header

FIRC\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.11 FIRG\_Header

FIRG\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.12 GII\_Header

GII\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>SegmentWidth</b>	UNSIGNED BYTE,
<b>SegmentHeight</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (93)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.13 MPEG\_Header

MPEG\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.14 NDVI\_Header

NDVI\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.15 NDVD\_Header

NDVD\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.16 OCAE\_Header

OACE\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.17 RII\_Header

RII\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>SegmentWidth</b>	UNSIGNED BYTE,
<b>SegmentHeight</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
}	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.3.2.18 TH\_Header

TH\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Pad1</b>	CHARACTERSTRING SIZE (3),
<b>ProcessingSegmentWidth</b>	UNSIGNED BYTE,
<b>ProcessingSegmentHeight</b>	UNSIGNED BYTE,
<b>NoSegmentsInProduct</b>	UNSIGNED SHORT,
<b>ProductVerified</b>	UNSIGNED BYTE,
<b>Pad2</b>	CHARACTERSTRING SIZE (1),
<b>NoGTSMessages</b>	UNSIGNED SHORT,
<b>MessageSizes</b>	UNSIGNED SHORT,
<b>THVerificationSummary.NoCollsTH6_2</b>	UNSIGNED SHORT
<b>THVerificationSummary.MeanCalcTH6_2</b>	UNSIGNED BYTE,
<b>THVerificationSummary.MeanObsTH6_2</b>	UNSIGNED BYTE,

<b>THVerificationSummary.MeanDiffTH6_2</b>	UNSIGNED BYTE,
<b>THVerificationSummary.RMSDiffTH6_2</b>	UNSIGNED BYTE,
<b>THVerificationSummary.NoCollsTH7_3</b>	UNSIGNED SHORT,
<b>THVerificationSummary.MeanCalcTH7_3</b>	UNSIGNED BYTE,
<b>THVerificationSummary.MeanObsTH7_3</b>	UNSIGNED BYTE,
<b>THVerificationSummary.MeanDiffTH7_3</b>	UNSIGNED BYTE,
<b>THVerificationSummary.RMSDiffTH7_3</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (80)}

#### A.5.3.2.19 TOZ\_Header

TOZ\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>SegmentWidth</b>	UNSIGNED BYTE,
<b>SegmentHeight</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (93)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.20 VOLC\_Header

VOLC\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

#### A.5.3.2.21 VOLE\_Header

VOLE\_Header is defined as:

<b>{ProductHeaderVersion</b>	UNSIGNED BYTE,
<b>Filler</b>	CHARACTERSTRING SIZE (95)
<b>}</b>	

**ProductHeaderVersion** is set to zero initially and is used to identify possible future upgrades of this record.

### A.5.4 Data\_Definition\_Block for MPEF Products

**Data\_Definition\_Block** is a record of the Image Data Function Header #3 as defined in Section 4.3.2.4.



**FrequencyOffset** is the frequency difference between the received carrier and the nominal channel centre frequency. The units are in Hz. The precision of the measurement depends on the receiver design.

**CarrierLevel** is the level of the received signal carrier. The value indicated allows for a reliable relative indication of signal level between different receivers.

**Modulation Level** is an estimate of the modulation index of the received signal. The nominal modulation index for DCP messages is 60° with limits of +0% -10%.

**FramesyncErrors:** The DCP has a 15 bit frame sync word, transmitted immediately after the 250 bit preamble. This value indicates the number of bits in error in the sync word. If the receiver is configured for an error threshold of zero in the frame sync correlator, then this field will always be zero.

**AddressErrors:** The DCP Address field includes a BCH code. The code can be used to detect 1, 2 or 3 errors or to correct 1 or 2 bit errors in the 21 bit address value. This field indicates the number of detected or corrected bit errors. Note: the corrected address is placed in the DCP\_MESSAGE by the DCP receiver.

**EOTErrors:** The 31 bit EOT word is detectable with 0,1 or 2 bit errors. This value indicates the number of bits in error in the EOT word. If the receiver is configured for an error threshold of zero in the EOT correlator, then this field will always be zero.

**MessageLengthFlag:** The DCP message can have a maximum length of 5192 bits + EOT for “100bps Self Timed” and “100 bps Alert” DCPs and above for “100bps Long Message” DCPs (see **ChannelType** field). The receiver counts the message length and if the maximum length of data is detected without recognising an EOT, then this flag is set.

**EOTcorrect:** This flag is set false under 2 conditions:

1. If the MessageLengthFlag is set.
2. If the channel receiver unlocks (due to loss of signal), before EOT is detected.

**MessageDecodeFlag:** This flag shall be set permanently to TRUE. Note that future development of the DCP service will most likely provide for FEC coding with the DCP message and this flag is reserved for future DCP types.

**TimeFrameAlarm:** It is expected that the DCP receivers will have a reference time code input from the PGS master clock, to provide for accurate time tagging of the received messages. If for any reason the input time reference code to the receiver is not present or cannot be acquired, then this flag is set. However the receiver will continue to operate, even when no reference time code is available, by means of a free wheeling internal reference clock. The **TimeFrameAlarm** flag will indicate that the time tag word may be unreliable.

**AbortReason:** The DCP receiver will abort the acquisition process if the signal does not conform to DCP transmission formats. There are a number of possible reasons for the

acquisition process to be aborted. In the case an abort occurs the receiver will generate an Abort message and indicate the reason for the abort in this field.

**FrameSyncTime:** corresponds to the time of reception of the DCP message. This time is relative to the end of the sync pattern of the message in the received transmission.

The DCP\_MESSAGE record is defined as follows:

DCP_MESSAGE ::= RECORD	
{ <b>MessageLength</b>	UNSIGNED SHORT,
<b>DCPAddress</b>	UNSIGNED,
<b>DCPMessage</b>	OCTETSTRING,
<b>DCPEOT</b>	UNSIGNED
}	

The **MessageLength** field gives the length in bytes of the remaining fields in the record

The **DCPAddress** field contains a 31-bit Bose-Chaudhuri-Hocquegem (BCH) coded word plus one spare bit. The first 21 bits are the address itself, the following 10 bits are derived from the first 21 bits and serve as an error check. The PGS performs the **DCPAddress** check/correction and specifies the result in the DCP\_QUALITY record. In case of a test message, then the address is according to Section 4.6.6. In the case of an “abort message”, then the **DCPAddress** is set to zero.

**DCPMessage** either contains characters of an abridged version of the International Alphabet No. 5 (IA5) or any other (even private) binary coding as long as the EOT character is avoided. In the case of IA5 use the 8th bit forms an odd parity bit to the seven character bits (i.e. the parity bit equals to 'zero' in the case the previous 7 bits contain an odd number of 'ones'). The length of the **DCPMessage** field is given by (**MessageLength** – (2 x 4)).

Two different types of DCP messages exist. 'Self-timed' DCP messages contain a maximum of 5192 bits (649 bytes) (excluding DCPAddress and DCPEOT) and 'Alert' DCP messages can have a maximum of 184 bits (23 bytes) (excluding DCPAddress and DCPEOT).

When a DCP message is a test message (identifiable by its address) its contents are as given at the end of this section.

The **DCPEOT** field contains the 'End of Transmission' sequence, comprising 31 bits plus one spare bit. The first 8 bits correspond to the End of Text (EOT) character of IA5.

0010 0000 1011 1011 0101 0011 1100 011	+ spare bit
first	last
	transmitted bit

For packets containing “abort messages” (i.e. any packet with the **AbortReason** field set to anything other than success) the DCP\_MESSAGE record will contain zeros in all its fields except the **MessageLength** field. This will have a value of 8 indicating that the **DCPAddress** and **DCPEOT** are present but that the **DCPMessage** field has a zero length. **DCP Test Message.**

The MSG PGS includes equipment to generate DCP test messages. These messages are transmitted to the satellite like any other and received again by the PGS. Once received they are treated like any other DCP message and passed on to the DADF for dissemination.

These test messages can be identified by checking the DCP address which is either “2637F0D6“ (in hex) for EUM/MSG TEST 1 or „1217648C“ (in hex) for EUM/MSG TEST 2.

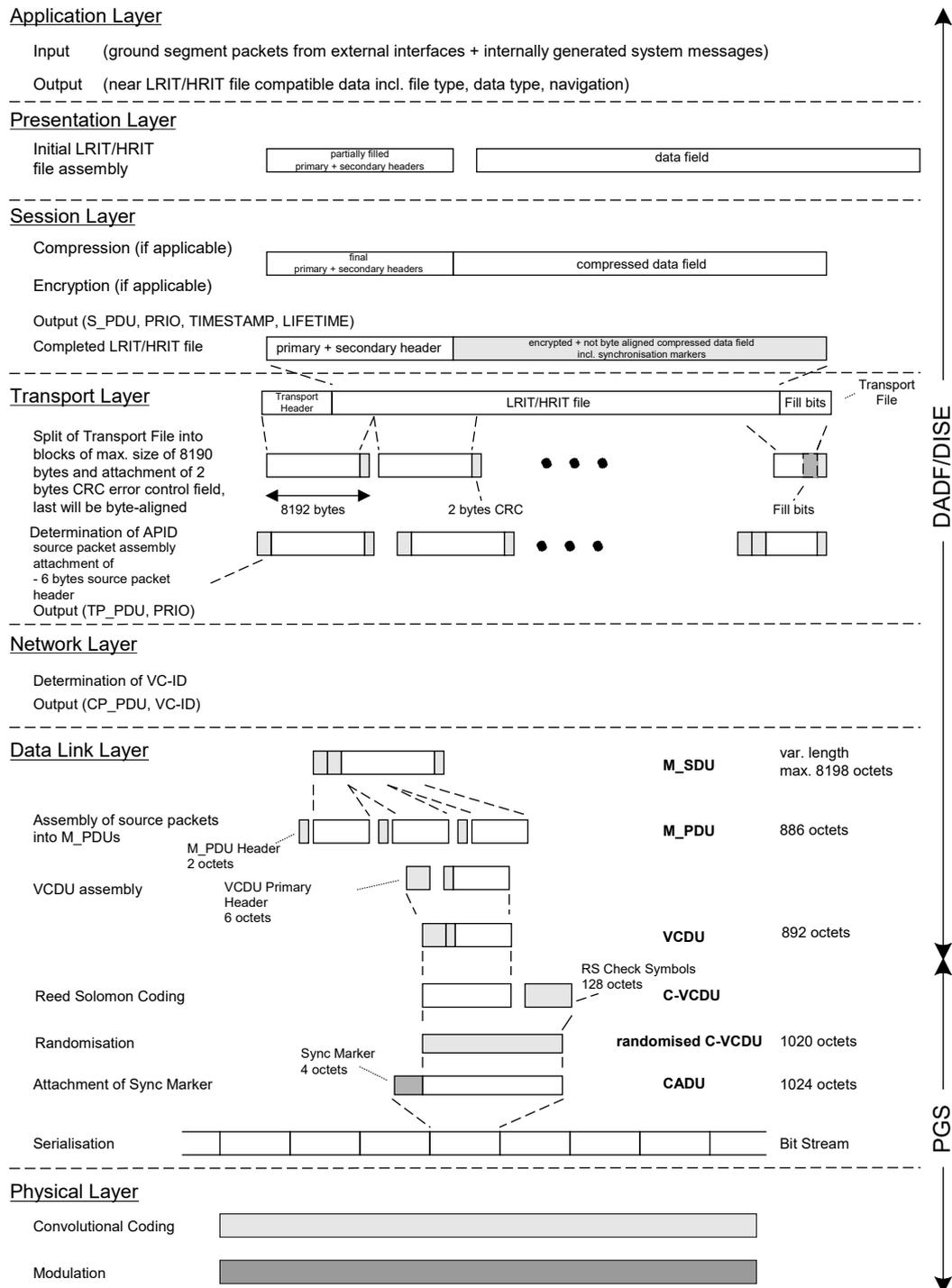
The DCP test message contents can be used for checks against a fixed pattern to detect any bit errors and to assess the link quality of the various DCP channels in use. The DCP test message contents are as follows (in hex):

```

000 : FC 0F 38 FC 8E 7C CC D7 34 49 00 81 44 30 59 48
010 : 85 D7 B5 0D 30 D8 0C B5 8E FD 88 E7 6D 01 85 56
020 : F1 3D 69 90 89 62 3B F0 B8 3F 61 B4 0B AB 79 DA
030 : 04 91 0C 34 CA CD D1 AF E8 D6 30 5B C1 E5 65 25
040 : 07 9F B3 17 D5 3E E4 E2 FA 96 10 4A 8D F3 37 C4
050 : 72 F2 B0 1B E3 7D 49 81 C5 74 69 11 CD 52 62 B8
060 : 3D E8 D4 B9 3B 73 75 6F 8A 6C 84 D3 A7 CC 54 F9
070 : 19 EB 59 CB 48 87 5E D5 BF A0 D2 A3 DE 95 9D 38
080 : FE 07 1C 7E 47 3E E6 6B 9A 24 80 40 22 98 2C A4
090 : C2 EB DA 06 18 6C 86 5A C7 7E C4 F3 B6 80 42 AB
0A0 : F8 9E 34 C8 44 B1 1D 78 DC 9F 30 DA 85 D5 3C 6D
0B0 : 82 48 06 1A E5 E6 E8 57 74 6B 98 AD E0 F2 B2 92
0C0 : 83 CF D9 8B 6A 1F 72 71 7D 4B 08 A5 C6 F9 1B 62
0D0 : 39 79 D8 8D F1 BE A4 C0 62 BA B4 88 66 29 31 DC
0E0 : 1E 74 EA DC 9D B9 BA 37 45 36 C2 E9 53 66 AA FC
0F0 : 8C F5 AC 65 A4 43 AF EA 5F 50 E9 51 EF CA 4E 1C
100 : FF 03 0E BF 23 1F F3 35 4D 12 40 20 11 4C 16 52
110 : E1 75 6D 03 0C 36 43 AD 63 3F E2 79 5B 40 A1 55
120 : 7C 4F 1A 64 A2 D8 0E 3C EE 4F 18 ED C2 6A 9E 36
130 : 41 24 03 8D 72 73 F4 2B BA 35 CC 56 70 79 59 C9
140 : C1 E7 EC 45 B5 0F B9 B8 BE 25 84 52 E3 FC 0D B1
150 : 9C 3C EC C6 78 5F 52 60 31 5D 5A 44 B3 94 18 6E
160 : 0F 3A 75 EE CE 5C DD 9B 22 1B E1 F4 29 33 55 7E
170 : C6 7A D6 32 D2 A1 57 F5 2F A8 F4 A8 77 65 27 8E
180 : FF 01 87 DF 91 8F F9 9A 26 09 20 90 08 26 0B A9
190 : F0 BA B6 01 06 9B A1 D6 B1 1F F1 BC 2D A0 D0 2A
1A0 : BE 27 0D 32 51 6C 07 1E F7 27 8C 76 61 35 4F 9B
1B0 : 20 92 81 46 B9 39 FA 15 DD 1A 66 2B B8 BC AC E4
1C0 : E0 73 F6 A2 DA 87 5C 5C DF 12 42 A9 71 FE 86 58
1D0 : 4E 1E 76 63 BC 2F 29 B0 98 2E 2D A2 59 4A 0C B7
1E0 : 07 9D 3A 77 67 AE EE 4D 91 8D 70 FA 94 99 2A 3F
1F0 : 63 3D 6B 19 E9 D0 AB FA 17 54 7A D4 BB B2 13 C7
200 : FF 80 C3 EF C8 C7 7C 4D 93 04 10 48 04 93 85 54
210 : 78 5D DB 00 83 CD 50 EB D8 8F 78 DE 16 50 68 15
220 : DF 93 06 99 28 B6 03 8F FB 13 46 BB B0 9A A7 4D
230 : 10 C9 40 A3 DC 1C FD 8A 6E 0D B3 15 5C 5E 56 72
240 : F0 39 7B 51 ED 43 2E AE 6F 09 A1 D4 38 7F 43 2C
250 : 27 0F BB 31 DE 97 14 58 4C 97 16 D1 2C 25 86 DB
260 : 83 4E 9D BB 33 57 F7 A6 C8 46 38 7D CA 4C 95 9F
270 : B1 9E B5 8C 74 E8 55 FD 0B 2A 3D EA 5D D9

```

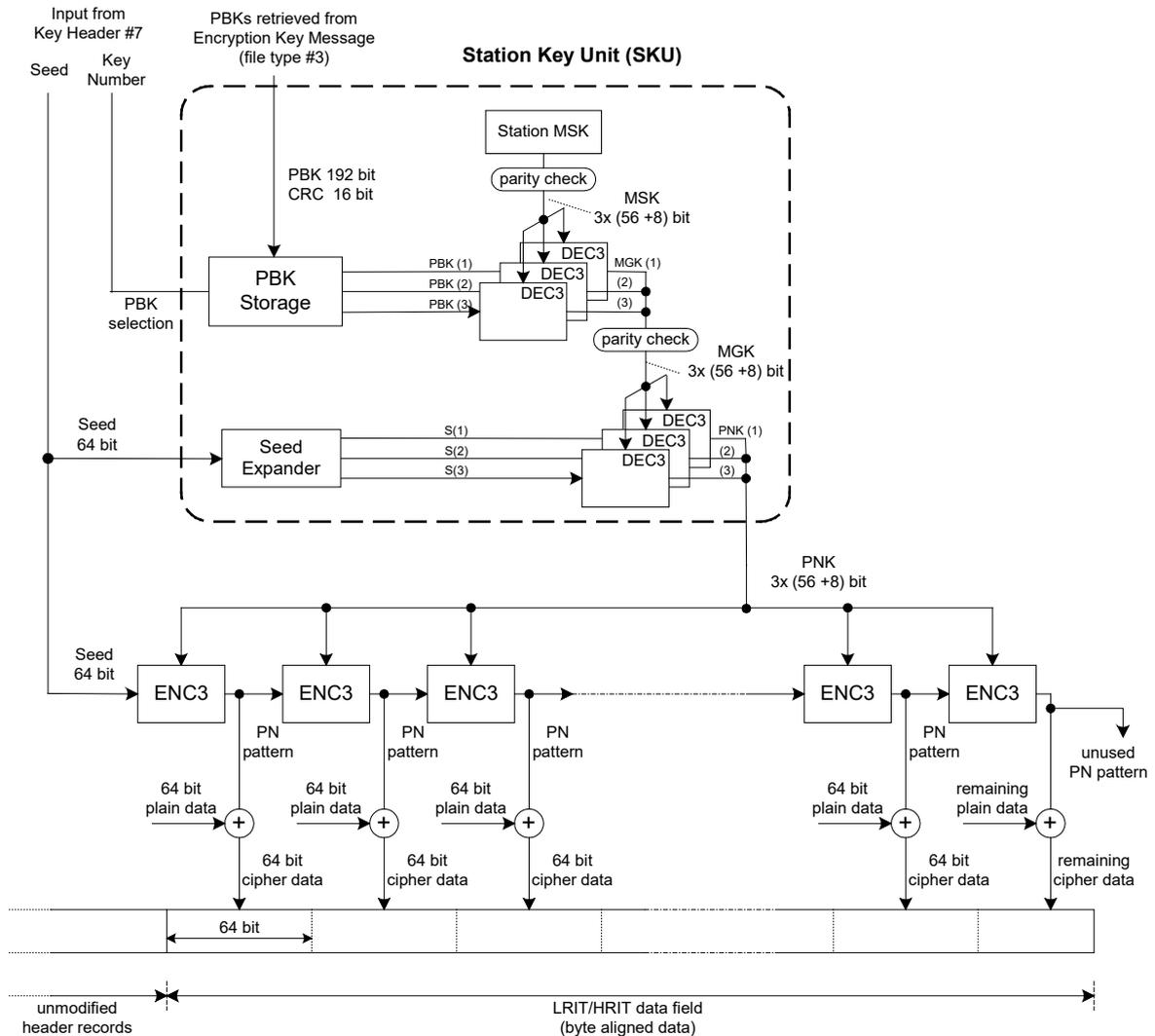
## APPENDIX B LRIT/HRIT DATA FORMATTING STRUCTURED ACCORDING OSI REFERENCE MODEL



**Figure B-1 – LRIT/HRIT Formatting according to OSI Reference Model**

**APPENDIX C LRIT/HRIT ENCRYPTION SCHEME**

**C.1 Overview**



**Figure C-1 – LRIT/HRIT Encryption Scheme**

**C.2 Public Key Entry Representation in an Encryption Key Message File**

The data field of an Encryption Key Message (file type #3) consists of a number of Public Key entries per user station and key number (see Section. 4.2.5). Each Public Key entry consists of:

- A 192 bit Public\_Key
- A 16 bit Public\_Key\_CRC

as shown in Figure C-2.

Public\_Key bit numbering conforms to the definitions given in Figure 5-49.

Public_key (192 bit) PBK(1,1) ... PBK(1,64), PBK(2,1) ... PBK(3,64)	Public_Key_CRC (16 bit)
---	----------------------------

**Figure C-2 – Public\_Key Entry of Encryption Key Message File**

The CRC is a 16 bit field which will be computed over the 24 bytes of the PBK starting with PBK(1,1) using the generator polynomial:

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

The polynomial generator shall be initialised to ‘all ones’ for each Key Message entry.

### C.3 Definition of the Seed Expansion

The Seed Expander is a functionality within the SKU. It will generate the following three 64 bit values based on the original Seed value provided by the LRIT/HRIT Key header (header type #7):

- S(1) = Seed
- S(2) = Seed value incremented by 1
- S(3) = Seed value incremented by 2

Example:

- Seed = '01 5C 4B 39 00 FF FF'h = S(1)
- S(2) = '01 5C 4B 39 01 00 00'h
- S(3) = '01 5C 4B 39 01 00 01'h

### C.4 Example for Encryption Validation

This example provides a complete set of encryption keys (MSK, PBK and PNK), a seed and a limited length of generated pseudo noise pattern in accordance with the definitions given in Sections 5.4.2 and 5.4.3.

The following values are assumed to be contained in the SKU:

- MSK(1) = '73 AE C1 46 20 57 13 BF'h
- MSK(2) = '8C F2 29 32 BA E3 DC 01'h
- MSK(3) = '4F 16 58 1C FB 89 A7 9B'h
  
- PBK(1) = '6F 74 15 E9 96 E1 20 59'h
- PBK(2) = '29 6A CC 8E C5 C9 76 3B'h
- PBK(3) = '68 C6 64 3B FD 88 84 E7'h

This leads to a Message Key result (internal to the SKU) of:

- MGK(1) = 'E6 15 7A 3B CE 52 F4 80'h
- MGK(2) = 'BF CB 0E 91 8C 2F D3 1F'h
- MGK(3) = '07 1A 67 FE C7 43 BA 51'h

Together with an assumed seed of:

```
S1 (Seed) = '0E 01 5C 4B 39 00 FF FF'h
S2        = '0E 01 5C 4B 39 01 00 00'h
S3        = '0E 01 5C 4B 39 01 00 01'h
```

The Pseudo Noise Key results in:

```
PNK(1)    = '20 5C DA 90 7D 95 E1 EA'h
PNK(2)    = '43 13 3A 71 1C 89 E2 84'h
PNK(3)    = 'CA 7E F1 19 01 69 56 BE'h
```

The following lists presents the first ten results of PN pattern generation:

```
'c9 a3 8f 2e f7 3a 16 04'h
'eb 42 9e bc e2 cc 4c b1'h
'b8 4b 24 69 31 ff 54 18'h
'54 8d d4 d2 71 47 61 cd'h
'75 55 7f 23 0d 2b 59 5d'h
'99 7e e3 32 92 c7 e0 dd'h
'd5 ae 08 16 34 eb 42 bd'h
'94 7c 32 13 2f 74 93 65'h
'64 5d f7 dd a3 ee a7 ad'h
'74 d2 58 09 59 92 b6 18'h
```

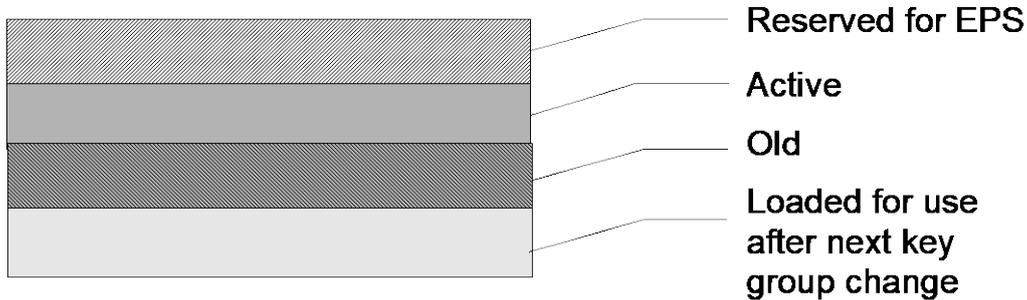
## **C.5 Key Group Changes: The Bank Concept**

In Section 4.3.2.8, Header Type #7 - Key Header, the concept of “key groups” is introduced, mentioning the swap of actively used key groups as an element of the MSG encryption scheme. For a better understanding of this scheme, its main elements are described in the following, addressing both the respective activities at the Dissemination/Key Centre Element of the MSG Ground Segment and the resulting effects on the User Stations/Station Key Units (SKUs).

For each pseudo key number (termed “Key\_Number” in “Header Type #7”) the Key Center generates two secret message keys. The encryption of the message keys against the secret master station key will generate two SKU specific public keys. Each public key belongs to a so-called key group. One key group contains the currently valid keys, the other key group contains those keys, which will be used for decryption after a key group change. If a key group change is executed, half of the public keys will be recalculated. The dissemination of Encryption Key Message files always includes both key groups.

### **C.5.1 Example of a Key Group Change as seen by a User Station/SKU**

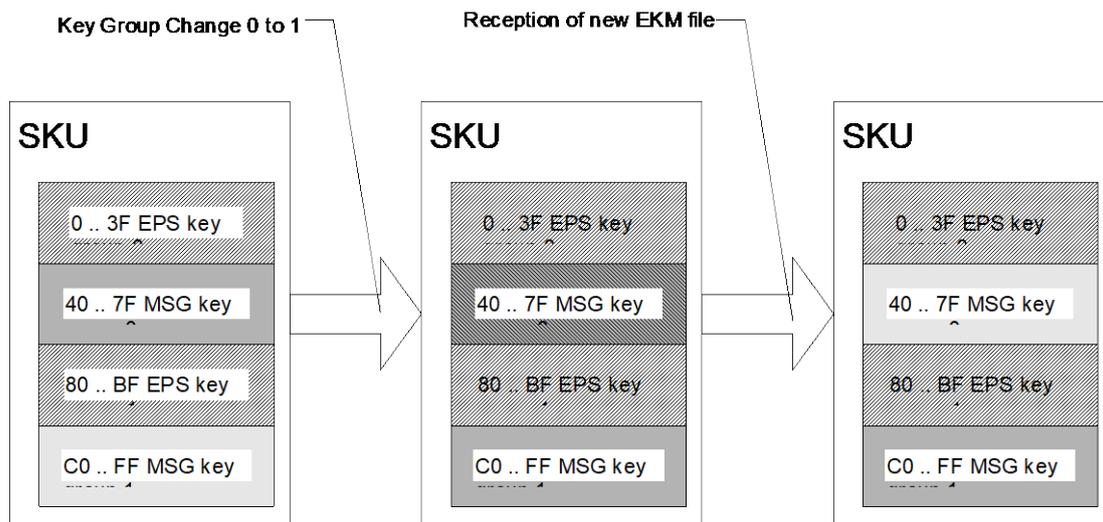
The following states of the storage of public keys in an SKU are identified:



**Figure C-3 – Legend for Public Key Storage Status**

It is assumed that the last change of the encryption key group was performed some time ago, and the User Station has loaded both key groups into the SKU public key messages storage. It is further assumed that key group 0 is the active one. After a key group change in the Dissemination Element, the encryption pre-processor is loaded with the new message keys (key group 1) and the subsequent messages to be disseminated will be encrypted using these new message keys.

User Stations can decrypt the received XRIT files, using the stored key group 1. As part of the key group change, the Dissemination Element generates and disseminates new Encryption Key Messages, containing the (currently active) key group 1 and a new key group 0. This new key group 0 is stored in the SKU, overwriting the old key group 0. That is the prerequisite for the next key group change, analogous to the one described above: The dissemination Element would encrypt using key group 0, and send out Encryption Key Messages with the active key group 0 and a new key group 1.



**Figure C-4 – Key Group Change: PBKs Storage in SKU**

**C.6 Example of Data Communication Between User Station and SKU**

A comprehensive and detailed description of the data communication between User Station and SKU is provided in the document *EUM/MSG/ICD/114 ICD SKU*. The following

sections describe the principles of the communication protocol and, as an example, the details of one command from the User Station to the SKU.

### C.6.1 Protocol Naming Conventions

Only a small number of instructions to the SKU and a small number of responses from the SKU are necessary to provide the required SKU functionality.

The following naming convention is used:

- The bits in one byte (8 bits) are numbered according to their binary weight, that means bit 0 is the least significant bit and bit 7 is the most significant bit.
- “A” designates the ASCII representation of the related bit pattern (A  $\hat{=}$  01000001)
- <CR> designates an ASCII control code, here carriage return ( <CR>  $\hat{=}$  00001101)

### C.6.2 Protocol Definition

The data passing the SKU interface constitute instructions to the SKU and answers from the SKU. All these data streams follow the same structure. The principal command structure for an instruction to the SKU is shown in the following table.

Each command to the SKU begins with a “.” as a start symbol, which can be used for synchronisation purposes. After that three bytes are provided to specify the command. These identifiers are followed by a separator. All these five bytes are ASCII coded and together form the so-called header of the command.

Afterwards the data are transferred, for example a key number, a public key and a public key CRC field etc. Each information byte is coded in two digit hexadecimal numbers (codes “0”...”9” and “A”...”F”). Leading zeros must neither be substituted by any other code nor be omitted. The sequence of digits is always starting with the most significant digit and ending with the least significant digit. The hexadecimal digits again are ASCII coded for the transmission.

The SKU command always ends with <CR> as a delimiter.

Byte-No.	Contents	Function
0	“.”	start symbol
1	“X”	} identifier
2	“Y”	
3	“Z”	
4	“_”	separator
5	“0”...”F”	MSD
6	“0”...”F”	LSD
7	“0”...”F”	MSD

•		
•		
•		
n-2	“0”...”F”	LSD
n-1	<CR>	stop symbol

**Table C-1 – Structure of SKU Commands**

The response string (answer) from the SKU has the same data structure. In addition a so-called result byte is inserted prior to the <CR> delimiter. The result byte is represented as two ASCII coded hex digits. The result byte provides information about potential errors which might have occurred during execution of a previously received instruction. The result byte is constructed such that errors for all implemented SKU commands can be coded by the eight bits of the byte. Therefore the same result byte can be used in all SKU response strings. The type of error is coded by the binary weight of the corresponding bit. The meaning of each individual bit within the result byte is shown in the following table.

Result Byte		DESCRIPTION
MSD	bit 3 =	1: time out error detected 0: OK
	bit 2 =	1: data error detected 0: OK
	bit 1 =	1: unknown command detected 0: OK
	bit 0 =	1: MSG parity error at PNK calculate access 0: MSG parity OK at PNK calculate access
LSD	bit 3 =	1: PBK CRC error at PNK calculate access 0: PBK CRC OK at PNK calculate access
	bit 2 =	1: CRC error at PBK read access 0: CRC OK at PBK read access
	bit 1 =	1: CRC error before PBK write access 0: CRC OK before PBK write access
	bit 0 =	1: CRC error after PBK write access 0: CRC OK after PBK write access

**Table C-2 – Result Byte Definition**

### C.6.3 Protocol Scheme

As already indicated above, the basic scheme of the communication protocol is as follows: The User Station sends an instruction to the SKU and the SKU reacts by outputting an appropriate answer. The synchronisation of incoming commands is performed with the help of the leading “.” of each instruction. If the SKU detects any instruction error, corrupted data or time out situations, a so-called error message will be returned.

### C.6.4 Example: Command “Write PBK”

The “Write PBK” command is used to store a public key in the SKUs non volatile memory. The detailed description of the command structure is given in the table below.

Byte-No.	Contents	Function
0	“.”	start symbol
1	“W”	} identifier
2	“R”	
3	“I”	
4	“ ” _	separator
5	“0”...”F”	} key number (ASCII coded hex)
6	“0”...”F”	
7	“0”...”F”	} public key (ASCII coded hex)
8	“0”...”F”	
.	.	
.	.	
54	“0”...”F”	LSD
55	“0”...”F”	} CRC field (ASCII coded hex)
56	“0”...”F”	
57	“0”...”F”	
58	“0”...”F”	
59	<CR>	stop symbol

**Table C-3 – Write PBK Command**

The command contains the key number, the public key and the public key CRC field in ASCII coded hexadecimal digits. The complete command consists of 60 bytes.

After performing a correct received “Write PBK” command the SKU response is as follows:

Byte-No.	Contents	Function
0	“.”	start symbol
1	“W”	} identifier
2	“R”	
3	“I”	
4	“ ” _	separator
5	“0”...”F”	} key number (ASCII coded hex)
6	“0”...”F”	

7	“0”...”F”	MS	} result (ASCII coded hex)
8	“0”...”F”	D LSD	
9	<CR>	stop symbol	

***Table C-4 – Response from SKU on "Write PBK"***

The message contains the key number and the result byte, both as ASCII coded hexadecimal digits. If the SKU detects a CRC error in the received public key, the write access will be rejected.

## APPENDIX D LRIT/HRIT PHYSICAL LAYER DETAILS

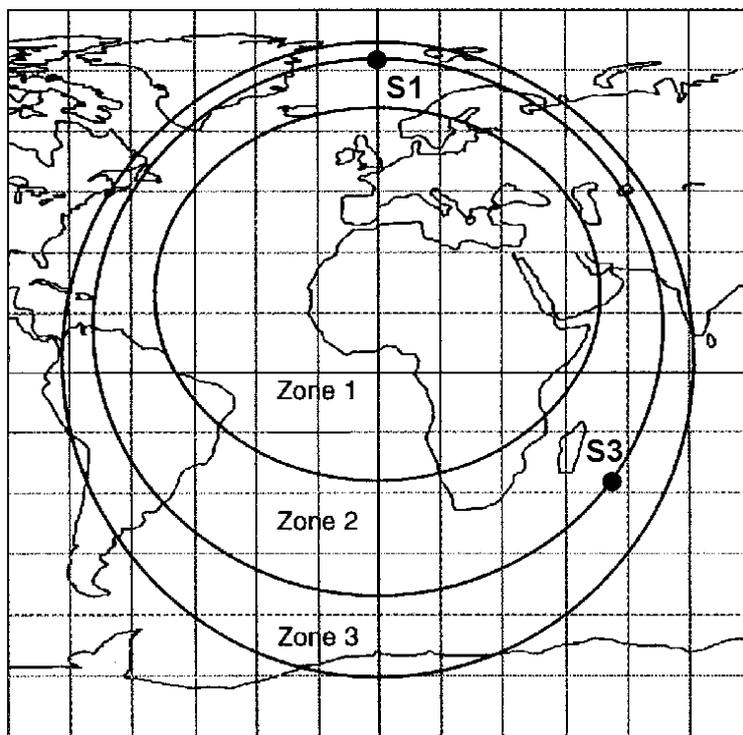
### D.1 Nominal Coverage Area

Zone 2 in Figure D-1 relates to the specified Nominal MSG Coverage Area. The coverage zones are identical for LRIT and HRIT. Please note that Figure D-1 does not intend to provide precise coverage boundaries and has to be replaced by a more precise drawing when test results of the MSG-1 Engineering Model are available.

User Station Front-end Specifications and allowed technical implementation losses related to the nominal coverage zone are given in Appendices D.2.2 and D.2.3.

A certain relaxation of these specifications can be expected in the central zone (zone 1) while more stringent requirements need to be applied if reception of MSG in the global zone (zone 3) is envisaged.

Please note that the locations S1 (most northerly point of the nominal coverage area) and S3 (most south-easterly point of the nominal coverage area) are used as references for certain space-to-ground parameters values.

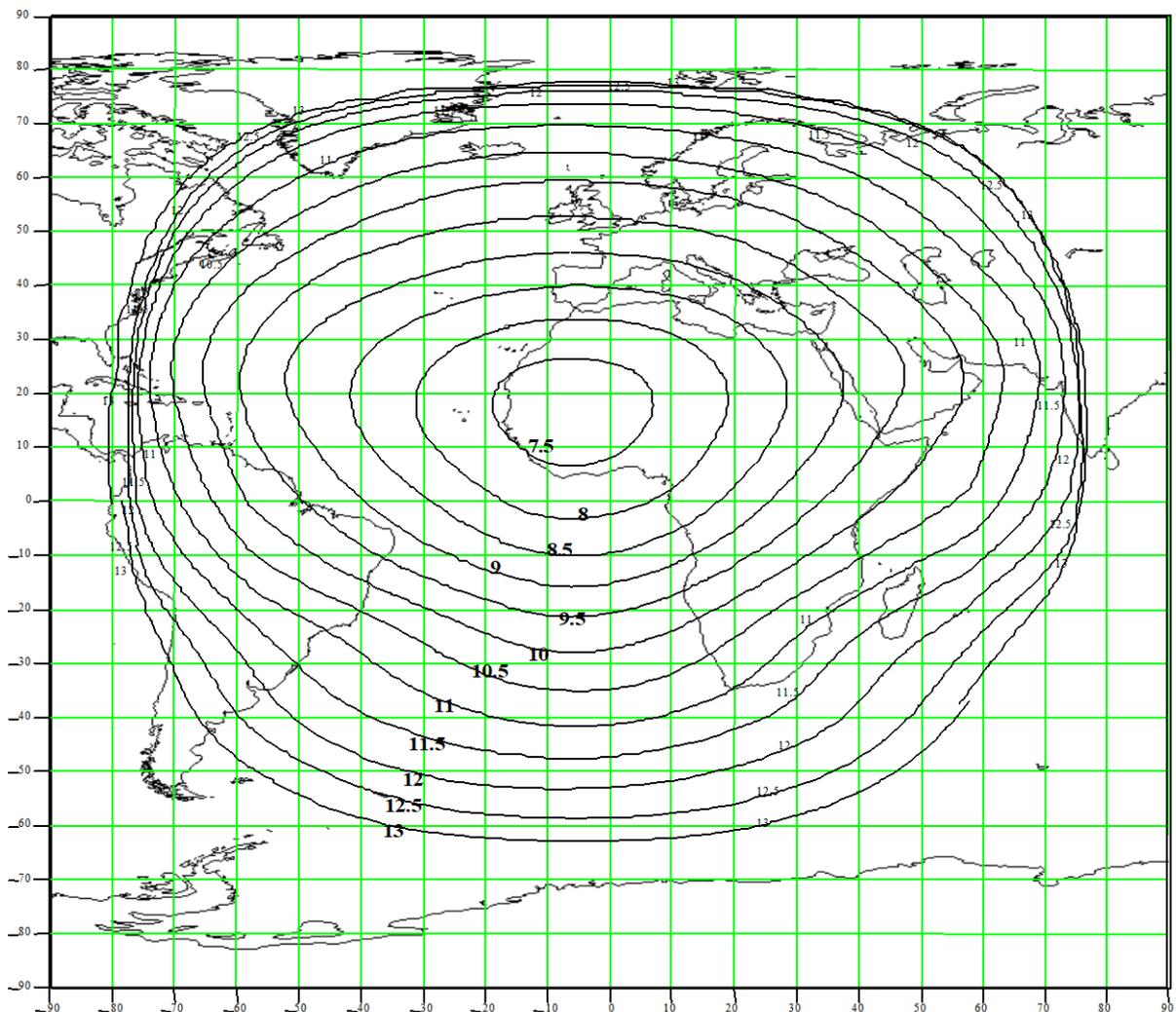


*Figure D-1 – MSG LRIT/HRIT Dissemination Coverage Zones*

#### D.1.1 HRIT G/T ISO-Contours

The G/T ISO-Contours in Figure D-2 show that the G/T requirement for an HRUS depends on the location of the station. For most areas of Africa and Europe a Front with a G/T of less

than 12 dB/K specified as nominal value looks sufficient. However, it must be pointed out, that these ISO values are for nominal S/C performance without any margin. Any under-performance or degradation of an MSG satellite would render these figures insufficient, causing the loss of reception for those stations which took full benefit from this relaxation. Thus, a margin of 2 to 3 dBs above the ISO figures for the respective locations is highly recommended. For an LRUS, similar considerations would apply, with an additional aspect: Lower G/T values achieved with smaller diameters of an LRUS antenna dish might be sufficient, but smaller dishes are more susceptible for picking up of interference, particularly through the temporary parallel operation of the MTP WEFAX mission.



**Figure D-2 – HRIT G/T ISO-Contours**

## D.2 LRIT/HRIT Space-to-Ground Interface

### D.2.1 Modulation Properties

Convolutional coding (R=0.5; K=7) is used, but no G2 path symbol inversion. No differential encoding is used.

For QPSK, the convolutional coding implementation shall be as follows:

- There is a single convolutional encoder,
- G1 and G2 symbols are placed on separate channels with G1 on I and G2 on Q.

The coding convention shall be as follows (only one-bit difference in adjacent phase states):

Carrier phase adv. (rad.)	Symbol values	
0	0	0
$\pi/2$	0	1
$\pi$	1	1
$3\pi/2$	1	0

The total length of the concatenated encoded structure is 2048 Bytes, containing concatenated (convolutional + Reed-Solomon) coding.

The physical characteristics of the HRIT and LRIT downlinks are given in the following subsections. The values contained are based on existing design documentation and will be updated once test results of the space and ground segment components are available:

### D.2.2 HRIT Physical Layer Characteristics

#### D.2.2.1 HRIT Down-link

Parameter	Nominal Case	Worst Case
Link general	HRIT down-link	
Link direction	satellite ---> ground	
Availability	on-station except - sun-satellite-PGS co-alignment, - non-operational satellite modes (stand-by)	
<b>Transmitting element</b>	1) Nominal case: L-band ESDA 2) Degraded case: L-band ESDA with one column failure	
Location	Satellite on geostationary orbit, within $\pm 0.5^\circ$ longitude accuracy 1) within $50^\circ$ W ... $50^\circ$ E at $\pm 1^\circ$ inclination 2) at $0^\circ$ longitude at $\pm 0.3^\circ$ inclination	
Tx distortion losses	0.3 dB	0.4 dB
Satellite EIRP	1) 19.4 dBW (S1) & 18.50 dBW (S3)	1) 17.3 dBW (S1) 16.1 dBW (S3) 2) 15.0 dBW

Parameter	Nominal Case	Worst Case
	2) 17.0 dBW	
Satellite EIRP variability over coverage	1) 4.8 dBpp 2) 7.7 dBpp	
Phase variability over one revolution	20°pp	
Satellite EIRP variability over one revolution	1) 4.5 dBpp 2) 7.8 dBpp	
Polarisation	linear horizontal, perpendicular to spin axis	
Cross-polar max level	- 20 dB	
<b>Link Characteristics</b>		
Centre frequency (F0)	1695.15 Mhz	
Frequency setting	± 2 ppm	
Long-term frequency stability	± 16 ppm over lifetime (7 years) and Qual temperature range.	
Useful Bandwidth (@ -1 dB)	1.960 MHz	
Packetised data rate	1.0 Mbps	
Total data rate	2.28 Mbps	
Modulation scheme	PCM / NRZ-L / QPSK	
Spurious modulation	Discrete: - 40 dBc ----- Random (phase noise): -49 dBc/Hz @ 10 Hz -76 dBc/Hz @ 100 Hz -90 dBc/Hz @ 1 kHz -90 dBc/Hz @ 10 kHz -105 dBc/Hz @ 100 kHz	
Gnd PLL BW (min)	BL > 100 Hz (one sided)	
Group delay variation (wrt up-link)	160 nspp	
Spurious output (@ transponder output)	RAB 1660 - 1670 MHz: -80 dBm/Hz --- 2066 - 2072 MHz: -105 dBm/1 MHz 2100 - 2110 MHz: -90 dBm/10 MHz 400 - 408 MHz: -137 dBm/Hz ----- -26 dBm/100Hz within [100 ; 10000] MHz, excluding useful bands and previous restrictions.	
Amplitude ripple	0.5 dBpp	
Pulse shaping	raised cosine with roll-off factor 0.7 and apportionment of 0.5 (between ground Tx & ground Rx filtering)	
Polarisation losses (Faraday rotation effect) with elevation of receiving element	0.5 dB (≥10°elevation) 0.1 dB (≥24° elevation)	2.0 dB (≥10°elevation)* 0.5 dB (≥24°elevation)* * arithmetic summation: 0.5 dB
Atmosph. + rain atten. losses (99.9% availability)	0.1 dB	0.226 dB
<b>Receiving Element</b>		
Location	1) Satellite field-of-view with min. elevation of 5° to satellite (Zone 3) 2) Reduced zone in ESDA degraded mode (approx. central zone)	
Polarisation	Linear horizontal (linear aligned to downlink polarisation)	
Polarisation mismatch losses	Not specified. no polarisation loss compensation (no polarisation auto-tracking)	
Pointing losses	0.92 dB	2.45 dB

Parameter	Nominal Case	Worst Case
	no auto-tracking system	
G/T (with location zone, see section )	17.0 dB/K in global zone 14.0 dB/K in nom. zone 12.0 dB/K in central zone	16.0 dB/K in global zone 12.0 dB/K in nom. zone 10.0 dB/K in central zone
On board C/I	30 dB	
Coding gain	9.4 dB	9.4 dB
Demodulation / Implementation losses	0.8 dB	1.0 dB
Implementation losses uncertainty	0.4 dB	0.4 dB
<b>Link Quality</b>		
Probability of frame loss (PFL)	$< 0.5 * 10^{-4}$ (Eb/No = 2.8 dB)	

**Table D-1 – HRIT Downlink - Physical Layer Characteristics**

## D.2.3 LRIT Physical Layer Characteristics

### D.2.3.1 LRIT Down-link

Parameter	Nominal Case	Worst Case
Link general	LRIT down-link	
Link direction	satellite ---> ground	
Availability	on-station except - sun-satellite-PGS co-alignment, - non-operational satellite modes (stand-by)	
<b>Transmitting Element</b>	1) Nominal case: L-band ESDA 2) Degraded case: L-band ESDA with one column failure	
Location	Satellite on geostationary orbit, within $\pm 0.5^\circ$ longitude accuracy 1) within $50^\circ$ W ... $50^\circ$ E at $\pm 1^\circ$ inclination 2) at $0^\circ$ longitude at $\pm 0.3^\circ$ inclination	
Transponder distortion losses	0.1 dB	0.2 dB
Satellite EIRP	1) 16.7 dBW (S1) 15.5 dBW (S3) 2) 14.4 dBW	1) 14.7 dBW (S1) 13.5 dBW (S3) 2) 12.4 dBW
Satellite EIRP variability over coverage	1) 5.1 dBpp 2) 7.8 dBpp	
Phase variability over one revolution	20°pp	
Satellite EIRP variability over one revolution	1) 4.8 dBpp 2) 7.9 dBpp	
Polarisation	linear horizontal, perpendicular to spin axis	
Cross polar max level	- 20 dB	
<b>Link Characteristics</b>		
Centre frequency (F0)	1691.00 Mhz	
Frequency setting	$\pm 2$ ppm	
Long-term frequency stability	$\pm 16$ ppm over lifetime (7 years) and Qual temp range	
Useful Bandwith (@ -1 dB)	0.660 MHz	

Parameter	Nominal Case	Worst Case
Packetised data rate	128 kbps	
Total data rate	290 kbps	
Modulation scheme	PCM / NRZ-L / BPSK	
Spurious modulation	Discrete: - 40 dBc ----- Random (phase noise): -49 dBc/Hz @ 10 Hz -76 dBc/Hz @ 100 Hz -90 dBc/Hz @ 1 kHz -100 dBc/Hz @ 10 kHz -105 dBc/Hz @ 100 kHz	
Gnd PLL BW (min)	BL > 100 Hz (one sided)	
Group delay variation (wrt up-link)	200 nspp	
Spurious output (@ transponder output)	RAB 1660 - 1670 MHz: -80 dBm/Hz ----- 2066 - 2072 MHz: -105 dBm/1 MHz 2100 - 2110 MHz: -90 dBm/10 MHz 400 - 408 MHz: -137 dBm/Hz ----- -26 dBm/100 Hz within [100 ; 10000] MHz, excluding useful bands and previous restrictions.	
Amplitude ripple	0.5 dBpp	
Pulse shaping	raised cosine with roll-off factor 1.0 and apportionment of 0.5 (between ground Tx & ground Rx filtering)	
Polarisation losses (Faraday rotation effect) with elevation of receiving element	0.5 dB ( $\geq 10^\circ$ elevation) 0.1 dB ( $\geq 24^\circ$ elevation)	2.0 dB ( $\geq 10^\circ$ elevation)* 0.5 dB ( $\geq 24^\circ$ elevation)* * arithmetic summation: 0.5 dB
Atmosph. + rain atten. Losses (99.9% availability)	0.1 dB	0.226 dB
<b>Receiving Element</b>	LRUS	
Location	1) Satellite field-of-view with min. elevation of $5^\circ$ to satellite (Zone 3) 2) Reduced zone in ESDA degraded mode (approx. central zone)	
Polarisation	Linear horizontal (linear aligned to downlink polarisation)	
Polarisation mismatch losses	Not specified no polarisation loss compensation (no polarisation auto-tracking)	
Pointing losses	0.61 dB	1.1 dB
	no auto-tracking system	
G/T (with location zone, see section )	10.0 dB/K in global zone 6.0 dB/K in nom. zone 4.0 dB/K in central zone	9.0 dB/K in global zone 5.0 dB/K in nom. zone 3.0 dB/K in central zone
On board C/I	40 dB	
Coding gain	9.4 dB	9.4 dB
Demodulation / Implementation losses	0.6 dB	1.0 dB
Implementation losses uncertainty	0.5 dB	0.5 dB
<b>Link Quality</b>		
Probability of frame loss (PFL)	$< 0.5 * 10^{-4}$ (Eb/No = 2.8 dB)	

**Table D-2 – LRIT Downlink - Physical Layer Characteristics**



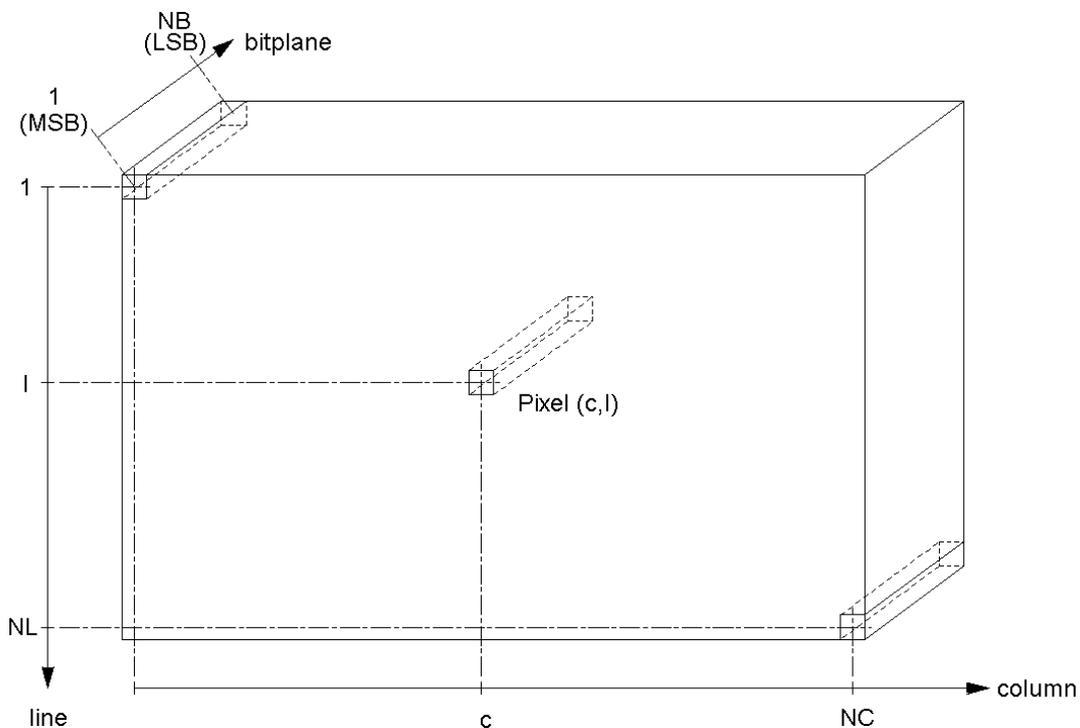
## APPENDIX E DERIVATION OF THE NAVIGATION COEFFICIENTS

### E.1 Introduction

This appendix describes the derivation of the LRIT/HRIT image navigation record coefficients CFAC, LFAC, COFF and LOFF that relate LRIT/HRIT image pixels to the intermediate coordinates or the geostationary scanning angles,  $x$  and  $y$ , as defined in [AD.1] and in the Image Navigation Record (Section 4.3.2.3 of this document).

The datum point of the level 1.5 images for the non-HRV channels in the operational scanning is the South-East corner pixel, which has the pixel coordinates (1,1). Essential for the derivation of the image navigation record coefficients are also the level 1.5 pixel coordinates of the sub-satellite point (nominally 0 degrees geographical latitude and longitude) and the pixel sampling (in angular units). These are specified in [RD.5] as pixel coordinates for the sub-satellite point to (1856,1856) and as pixel sampling to  $251.53/3 \mu\text{rad}$  for the non-HRV channels. The corresponding values for the HRV channel are (5566,5566) and  $251.53/9 \mu\text{rad}$ . The sub-satellite point coincides with the corresponding non-HRV and HRV pixel centres.

In the LRIT/HRIT file the image data field is ordered as a sequence of pixels, but in general it may not be the same as in Figure 4-2 in Section 4.2.2.1. Pixels are rather indexed according to Fig. E-1 which is shown below [AD.1].



**Figure E-1 – MSG LRIT/HRIT Image Structure**

In the following, pixel columns,  $c$ , and lines,  $l$ , always relate to LRIT/HRIT pixels and **not** to level 1.5 image pixel. The relation between pixel indices,  $c$  and  $l$ , and intermediate coordinates or geo-stationary scanning angles,  $x$  and  $y$ , is specified in [AD.1] as:

$$c = COFF + nint(x \cdot 2^{-16} \cdot CFAC) \quad (E.1)$$

$$l = LOFF + nint(y \cdot 2^{-16} \cdot LFAC) \quad (E.2)$$

where *nint* is the “nearest integer” operator. Equation E.1 is setting up the c-frame and E.2 is setting up the l-frame.

The coefficients COFF, LOFF, CFAC and LFAC are specified in the navigation record of the LRIT/HRIT file sent (see Section 4.3.2.3 of this document), so that everyone can unambiguously relate LRIT/HRIT pixel to geographical coordinates using the equation given in [AD.1]. Note, that the values of the geostationary scanning angle, *x* and *y*, that correspond to the centre of a level 1.5 image pixel change are in steps of  $\Delta = 83.84333 \mu\text{rad}$  ( $251.53/3 \mu\text{rad}$ ) for non-HRV channels and  $\Delta_{HRV} = 27.94778 \mu\text{rad}$  ( $251.53/9 \mu\text{rad}$ ) for the HRV channel. So their range is:

$$x = n \cdot \Delta \quad \text{where } n = -1856 \leq n \leq 1856 - 1 \quad \text{for non-HRV channels} \quad (E.3)$$

$$y = m \cdot \Delta \quad \text{where } m = -1856 \leq m \leq 1856 - 1 \quad (E.4)$$

$$x = n \cdot \Delta_{HRV} \quad \text{where } n = -5570 \leq n \leq 5566 - 1 \quad (E.5)$$

$$y = m \cdot \Delta_{HRV} \quad \text{where } m = -5570 \leq m \leq 5566 - 1 \quad \text{for the HRV channel} \quad (E.6)$$

Equation E.3 is setting up the n-frame and E.4 is setting up the m-frame.

		West	(image datum)					East
Non-HRV	c-frame	3712	...	1857	1856	1855	...	1
	n-frame	-1856	...	-1	0	1	...	1856-1
HRV	c-frame	11136	...	5567	5566	5565	...	1
	n-frame	-5570	...	-1	0	1	...	5566-1

**Table E-3 East-West Column numbering in the c-frame and the n-frame for Non-HRV and HRV channels.**

	Non-HRV l-frame	Non-HRV m-frame	HRV l-frame	HRV m-frame
North	3712	-1856	11136	-5570
	...	...	...	...
(image datum)	1857	-1	5567	-1
	1856	0	5566	0
	1855	1	5565	1
	...	...	...	...
South	1	1856-1	1	5566-1

**Table E-4 North-South Line numbering in the l-frame and the m-frame for the Non-HRV and HRV channels.**

The Tables E-1 and E-2 emphasize the relation of the counting of pixels in the different frames given by equations (E.3) to (E.6) for the Non-HRV and HRV channels and in relation

to the geographical location of the Earth when looking on a Full Earth Scanning Level 1.5 image reference grid which is defined in [RD.5].

This corresponds to an approximate scan range of  $\pm 9^\circ$  for  $x$  and  $y$ . In the operational scanning the positive  $x$ -direction is eastwards and the positive  $y$ -direction southwards, i.e. the most eastern pixel corresponds to  $n=1856-1$  and the most southern line to  $m=1856-1$ .

## E.2 Derivation of CFAC and LFAC

We first derive the scaling constants CFAC and LFAC (without any restrictions only for CFAC of non-HRV channels). Considering two adjacent pixels  $c$  and  $c+1$  we obtain, with substituting (E.3) into (E.1)

$$c = COFF + nint(n \cdot \Delta \cdot 2^{-16} \cdot CFAC) \quad (E.7)$$

$$c + 1 = COFF + nint((n - 1) \cdot \Delta \cdot 2^{-16} \cdot CFAC) \quad (E.8)$$

The minus sign appears because the  $n$ -frame and the  $x$ -frame are in different directions to each other.

We look for a solution to these equations while neglecting the  $nint$  Operator:

$$c = COFF + n \cdot \Delta \cdot 2^{-16} \cdot CFAC \quad (E.9)$$

$$c + 1 = COFF + (n - 1) \cdot \Delta \cdot 2^{-16} \cdot CFAC \quad (E.10)$$

Subtracting these two equations results in:

$$CFAC = -\frac{2^{16}}{\Delta} \quad (E.11)$$

With  $\Delta=83.84333 \mu\text{rad}$  for the non-HRV channels and after converting it into degrees, as defined in [AD.1], page 25, we obtain a value of  $CFAC = -13642336.642 \text{ deg}^{-1}$ .

As [AD.1] allows only integer values for CAFC/LAFC we apply the  $nint$ -operator to take the closest integer value, i.e.  $-13642337 \text{ deg}^{-1}$ , and verify that it satisfies the above equations.

$$c = COFF + nint(n \cdot \Delta_{\text{deg}} \cdot 2^{-16} \cdot -13642337) = COFF + nint(n \cdot -1.00000003) \quad (E.12)$$

$$c + 1 = COFF + nint((n + 1) \cdot \Delta_{\text{deg}} \cdot 2^{-16} \cdot -13642337) = COFF + nint((n + 1) \cdot -1.00000003) \quad (E.13)$$

As the range of  $n$  and  $m$  is limited to  $|n, m| \leq 1856$  the non-integer remainder never exceeds 1, so that the  $nint$ -operator can be omitted. Obviously a similar result is also obtained for LFAC. In combining both results we obtain:

$$c = COFF + nint(n \cdot -1.00000003) = COFF - n \quad (E.14)$$

$$l = LOFF + nint(m \cdot -1.00000003) = LOFF - m \quad (E.15)$$

In a similar way values for HRV can be derived, which is  $CFAC=LFAC=-40927010 \text{ deg}^{-1}$ .

For computational purposes it is useful to have CFAC/LFAC not in degrees<sup>-1</sup> but in radians<sup>-1</sup>. This could be easily derived using the above steps but omitting the radians to degrees conversion, resulting in the following values for the non-HRV channels  
 CFAC/LFAC=-781648343 rad<sup>-1</sup> and -2344944937 rad<sup>-1</sup> for the HRV channel.

A real-time processing of lines (pixel are stored as they are scanned and processed) results in it is advantageous to select a negative sign of CFAC/LFAC. This implies that the most southern level 1.5 line is the first line of the LRIT/HRIT file and the most eastern pixel is the first pixel every line to be stored in the file.

Once the sign and magnitude of CFAC/LFAC are defined, COFF/LOFF can be calculated.

### E.3 Derivation of COFF and LOFF

From Tables E-1 and E-2 we can read the COFF and LOFF values for the image datum in the intermediate coordinates  $x=0$  and  $y=0$ :

For  $x = 0$

$$\begin{aligned} c(0) &= COFF \Rightarrow \\ c(0) &= 1856 \\ c_{HRV}(0) &= 5566 \end{aligned}$$

For  $y = 0$

$$\begin{aligned} l(0) &= LOFF \Rightarrow \\ l(0) &= 1856 \\ l_{HRV}(0) &= 5566 \end{aligned}$$

#### E.3.1 LOFF of Segments

In the line direction the range of lines  $m$  is different for every image segment file. If  $k$  denote the image segment file number from South to North and  $NL$  is the number of lines in one image segment file (as given in HRIT Header Type #1, Section 4.3.2.2), the corresponding ranges are:

$$1855 - (k - 1) \cdot NL \leq m \leq 1855 - k \cdot NL + 1 \quad (E.16)$$

Note: The number of lines in an image segment file is currently set to  $NL=464$ . With that  $k=1, \dots, 8$  for non-HRV images.

Placing the most southern line value  $m$  first into the segment file implies:

$$l = 1 = LOFF - 1855 + (k - 1) \cdot NL \quad (E.17)$$

$$\Leftrightarrow LOFF = 1856 - (k - 1) \cdot NL \quad (E.18)$$

Therefore the LOFF value for the non-HRV channels change from 1856 for  $k=1$  to -1392 for  $k=8$ .

The situation for the LOFF values is quite similar as for the non-HRV channels except that the range of the HRV lines is different here:

$$5565 - (k - 1) \cdot M \leq m \leq 5565 - k \cdot NL + 1 \quad (\text{E.19})$$

Again,  $k$  refers to the image segment file (counting up from South to North) having a range from 1 to 24 (see Section 4.2.2.1 of this document). Hence we obtain:

$$l = 1 = LOFF - 5565 + (k - 1) \cdot NL \quad (\text{E.20})$$

$$\Leftrightarrow LOFF = 5566 - (k - 1) \cdot NL \quad (\text{E.21})$$

Therefore the LOFF value for the HRV channel changes from 5566 for  $k=1$  to  $-5106$  for  $k=24$ .

### E.3.2 COFF of Segments

Since the East/West extension of the HRIT Segments are always the full width of the Level 1.5 image COFF does not change.

For the HRV channel the situation is slightly different as lower or upper part of the level 1.5 image can have an offset depending on the Image Repeat Cycle Dissemination format (see Figure 4-4 of this document). Depending on the offset (specified by LowerEastColumnActual and UpperEastColumnActual, defined in [RD.5] Section 7.5.1) the range for the HRV channel is:

$$-1 - LowerEastColumnActual \leq n \leq 5566 - LowerEastColumnActual \quad (\text{E.22})$$

$$-1 - UpperEastColumnActual \leq n \leq 5566 - UpperEastColumnActual \quad (\text{E.23})$$

Putting again the most eastern pixel first yields:

$$c = 1 = COFF - 5566 + LowerEastColumnActual \quad (\text{E.24})$$

$$c = 1 = COFF - 5566 + UpperEastColumnActual \quad (\text{E.25})$$

Hence the corresponding COFF values are

$$COFF = 5567 - LowerEastColumnActual$$

Or (E.26)

$$COFF = 5567 - UpperEastColumnActual$$

### E.4 Lines Reduced Format – Rapid Scans

As stated in Section 4.2.2.1 MSG S/C supports ‘reduced lines’ scanning which is also known as “Rapid Scanning” or “Mini Scans”. The Segments which are not scanned are not disseminated. The disseminated segments have the segment number corresponding to the scanned region from a full Earth image. In other words if the Rapid Scan Region is the most northern part of the full image only the segments which cover this region are disseminated with their segment number accordingly. Please refer to Section 4.2.2.1 for further information.

## **E.5 Summary**

It must be emphasised that the above mentioned facts holds only for the operational scanning scheme which is from East to West and from South to North. With that the LRIT/HRIT image segment file is ordered with the most southern line first and within every line with the most eastern pixel first. We obtain the following values:

### **Non-HRV channel**

$$CFAC = LFAC = -13642337 \text{ deg}^{-1} = -781648343 \text{ rad}^{-1}$$

$$COFF = 1856$$

$$LOFF = 1856 - (k - 1) \cdot NL$$

### **HRV channel**

$$CFAC = LFAC = -40927010 \text{ deg}^{-1} = -2344944937 \text{ rad}^{-1}$$

$$COFF = 5567 - \text{LowerEastColumnActual} \mid \text{UpperEastColumnActual}$$

$$LOFF = 5566 - (k - 1) \cdot NL$$

**APPENDIX F            LIST OF TBDS AND TBCS****TBDs**

None

**TBCs**

None