magellium

S2MRCPBG

S2MRCPBG-S2QL
USER MANUAL

S2MRCPBG-S2QL
USER MANUAL

	Name	Company	Date	Signature
Written by:	Grégory Cazanave	Magellium	14/12/2012	
Verified by:	Grégory Cazanave	Magellium	14/12/2012	

Document reference:	S2MRCPBG-MU-019-MAG
Issue.revision:	1.9
Issue date:	14/12/2012
Customer:	ESA/ESTEC
ITT reference:	4000103130/11/NL/JA

Distribution list

	Name	Company	Copies
Recipients:	Philippe Martimort Riccardo Duca Olivier Colin Ferran Gascon Borja Lopez Fernandez Enrico Cadau	ESA/ESTEC ESA/ESTEC ESA/ESRIN ESA/ESRIN ESA/ESRIN ESA/ESRIN	1 digital 1 digital 1 digital 1 digital 1 digital 1 digital
Internal copy:	Customer folder Project leader Development team	Magellium Magellium Magellium	1 1 (electronic) 1 (electronic)

Document change record

Issue	Rev.	Date	Nature of change	Observations
1	0	29/05/2012	Document creation for S2MCPBG, taking account of KO presentation, PM1, PM2, PM3, PM4, Change Request 1, SRR & TRR RIDS. Result of the merge of 2 (now) obsolete documents: S2MRCPBG-MU-012-MAG-MU S2QUICKLOOK-2.1 & S2MRCPBG-MU-014-MAG-MU S2MRCPBG-1.1	
1	1	06/07/2012	Update with PM6 remarks and TRR RIDS TRR-TECH-21: Appendix F TRR-TECH-20: §5.3 TRR-TECH-18: §5.1.4 TRR-TECH-17: §5.4 TRR-TECH-14: §7.2 TRR-TECH-13: §4.3 TRR-TECH-12: §1.4 & §4.2 TRR-TECH-11: Appendix G TRR-TECH-10: §6.1.2 TRR-TECH-9: §4.3 & 7.1.3 TRR-TECH-8: Appendix D TRR-TECH-7: full document TRR-TECH-6: §5.2 §5.5 §6.2 §7.2 Deletion of pipe parameter.	

1	2	25/07/2012	Update with mail additional remarks: Quality mask output: §5.4 p29 'described in APPENDIX D too) Output RAW & LUM format: Appendix A & B ENVI header & Quality Mask : Appendix C qualityMask option 2 value: Appendix F & §5.1.2 French words: §5.4 §6.1.1	
1	3	14/09/2012	Update with remarks: Precision about unsigned short. Separate Quality Mask file naming. Precisions in appendix A, B, D. Precision about naming procedure (<sourcepacketlength>). English translation. Wording in Appendix A,B.	
1	4	28/09/2012	Update with remarks from the QR colocation meeting: Specification of the test environment delivery. Update of Appendix A: precision on quality mask granularity and encoding size. Extension ".bin" mandatory for input named pipe and DECOMP directory has to be created for output pipe. Update the behaviour of firstBDL parameter and specification of the recommended values. Update error/warning messages. Add configuration files of all S2 bands. Case sensitiveness update of option firstScene and qualityMask. Installation procedure update	
1	5	12/10/2012	Update with delta-QR remarks: Update error messages to take into account behaviour modification about checks on firstbdl and firsScene parameters. Update with the exit code of S2QL and the mirror option.	
1	6	02/11/2012	Update final comments from delta QR and modifications about EBBLNC management	

			(confirm_ebbln_generique parameter value set by default to 4)	
1	7	15/11/2012	Acronyms: added CRC definition Update description of the following parameters: D, ByPassPond, Filter53, SPKSHLN, P0-P9 Miroir: moved to fixed parameters Mode_big_endian: fixed parameter, description “reserved”	
1	8	27/11/2012	Update with mirror option configuration	
1	9	14/12/2012	Update scene_time_period parameter value in defaut.conf configuration file	

Table of Contents

1. Introduction

1.1. Purpose

- S2MRCPBG (full resolution decompression software) and S2QL (quicklook resolution decompression software) are compliant with linux platforms and validated on (TBD) with respect to the PDGS Reference platform; the command line are compliant with the csh profile and can be encapsulated in a csh script.
- This document is the User Manual of S2MRCPBG and S2QL decompression software dedicated for the Sentinel 2 mission. It aims to
  - Give an overview of the software,
  - Highlights matches important information of the global Sentinel2 mission,
  - Describe how to use the software in the Sentinel 2 context to generate unsigned short decompressed images. This encoding mode is the only one accessible for the S2 decompressed image.

Remark: “unsigned short” refers to 16 bits encoding in the entire document.

1.2. Organization of the document

Besides this introduction, this document includes the following chapters:

- Chapter 2 describes the installation procedure,
- Chapter 3 gives a global overview of the software components,
- Chapter 4 gives global Sentinel 2 information related to compression/decompression,
- Chapter 5 deals with the parameters for Sentinel 2 mission listing fixed parameters and explaining deeper those that can be modified.
- Chapter 6 describes command lines.
- Chapter 7 deals with error messages.
- Chapter 8 lists outputs.
- Appendix A describes RAW format image.
- Appendix B describes LUM format image.
- Appendix C describes ENVI like header format image,
- Appendix D describes the quality mask generation process.

1.3. Associated documents

1.3.1. Applicable Documents

Id.	Reference	Title
AD 01	**RfQ 3-12978 SCOT Appendix 3**	Special Conditions of Tender

AD 02	**RfQ 3-12978 SOW Appendix 1 - S2-SW-ESA-SY-112**	Statement Of Work
AD 03	**MAG-10-PTF-008-ALE-MRCPBG**	Magellium Technical Proposal
AD 04	**MAG-10-PTF-008-ALE-MRCPB**	Magellium Management Proposal
AD 05	**RfQ 3-12978**	Contract
AD 06	**S2-CR-ESA-SY-0020**	ESA Change Request 1
AD 07	**CCN1**	Magellium proposal to S2-CR-ESA-SY-0020package: S2MRCPBG-CCN ESA-CR1 Annex A - Work Description CCN1 Annex B - PSS A2 CCN 1

1.3.2. Reference Documents

Id.	Reference	Title
RD 01	**GS2.ICD.ASF.MSI.00008**	MSI Mission Data ICD (Astrium document)
RD 02	**GS2.RP.ASF.MSI.00006**	MSI Technical Description (Astrium document)
RD 03	**S2MRCPBG-NOT-008-MAG**	Global processing description
RD 04	**MIMU_MRCPBG_v6.0_English**	MRCPBG v9.0 Installation instructions and user manual
RD 05	**MAG-PRD-P.MGTSQ-014**	Magellium's rules of management of the documentation

1.4. Acronyms

Acronym	Description
BF	Low frequency/ies
bpp	Bits Per Pixel
CM	Configuration Management
CR	Change Request
CRC	Cyclic Redundancy Check
DACO	Ancillary compression data
DCF	Descriptive Configuration File
DWT	Discrete Wavelet Transform
EBB	EmBeded Bit stream
ESA	European Space Agency
ESRIN	European Space Research Institute
FS	Source format = Packet
FTP	File Transfer Protocol
HF	High frequency/ies
LD	Decompression software
LUM	Extensions of CNES image files (LUMinance) (See Appendix B)
PSC	Packet Sequence Count
PH	Primary Header
QL	Quicklook
RAW	Basic format for image files (See Appendix A)
S	Step

SB	Sub-Band
SH	Secondary Header
SPKLN	Source packet length corresponding to the compression ratio; name use for the information encoded in the primary header of the source packet.
SS	Sub Step
STRIP	*Block of 16 lines*
SYFS	Synchro source format
TBC	To Be Confirmed
TBD	To Be Defined
TO / TO^{-1}	Wavelet transform / Inverse wavelet transform
TPM	Third Party Mission
TBU	To Be Updated
TBW	To Be Written

2. Installation procedure

This chapter explains how to install the S2MRCPBG and S2QL software.

2.1. S2MRCPBG S2QL Installation procedure

This chapter deals with the installation of S2MRCPBG and S2QL software.
Delivery contains two different elements:

- Nominal delivery with software, test data and associated documentation,
- Test environment delivery with software, test data set, validation script.

2.1.1. S2MRCPBG S2QL Nominal delivery

This delivery is the one that can be directly diffused to end users.
Installation steps to follow are:

- Uncompress the "deliveryX.Z_DDMMYYYY.tar.gz" delivered file:

```
tar zxvf  deliveryX.Z_DDMMYYYY.tar.gz
```

- Check the uncompressed directory:
  - Main Directory: S2MRCPBG_X.Z,
  - Sub directory S2MRCPBG_X.Z/bin_gcc containing:
    - "mrcpbg_sol" binary executable file compiled with gcc; if not executable, use the command:

```
chmod u+x mrcpbg_sol
```

    - "libQL.so" binary file compiled with gcc,
    - Add the path that leads to the file "libQL.so" to the environment variable "LD_LIBRARY_PATH". If, for instance, the library is stored next to the executable "ExtractionAlbum", the current directory can be added to the environment variable as follows:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
```

    - "ExtractionAlbum" binary executable file compiled with gcc; if not executable, use the command:

```
chmod u+x ExtractionAlbum
```

    - "script_check_install.sh": this script loads a decompression test to validate the installation of the binary,
    - "bit stream_check_install.bin": bit stream to check the installation with the previous script,
  - Sub directory S2MRCPBG_X.Z/bin_icc containing:
    - "mrcpbg_sol" binary executable file compiled with icc; if not executable, use the command:

```
chmod u+x mrcpbg_sol
```

    - "libQL.so" binary file compiled with icc,
    - Add the path that leads to the file "libQL.so" to the environment variable "LD_LIBRARY_PATH". If, for instance, the library is stored next to the executable

"ExtractionAlbum", the current directory can be added to the environment variable as follows:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
```

    - "ExtractionAlbum" binary executable file compiled with icc; if not executable, use the command:

```
chmod u+x ExtractionAlbum
```

    - "script_check_install.sh": this script loads a decompression test to validate the installation of the binary,
    - "bit stream_check_install.bin": bit stream to check the installation with the previous script,
  - Sub directory S2MRCPBG_X.Z/CONF: sub directory containing configuration files for the installation test,
    - "defaut.apid",
    - "defaut.ih_10m",
    - "defaut.ih_20m",
    - "defaut.ih_60m",
    - "defaut.conf",
    - "defaut.fpft".
- Launch the installation script test "script_check_install.sh" from the "bin_gcc" and "bin_icc" directory:

```
"./script_check_install.sh"
```

- Check the creation of the 2 directories: "DECOMP", "RESULT",
- Check the "log" and the "error" files in the created directory "RESULT": no error processing.
- Check the full resolution decompressed image in the directory "DECOMP".
- Check the quicklook image in the local directory.

2.1.2. S2MRCPBG S2QL test environment delivery

This delivery is the one that allows validating software behaviour on your platform following the "User Acceptance Tes Report" document.

Installation steps to follow are:

- Uncompress the "TEST_S2MRCPBG-S2QL.tar.gz" delivered file:

```
tar zxvf  TEST_S2MRCPBG-S2QL.tar.gz
```

- Check the uncompressed directory:
  - Main Directory: S2MRCPBG_test_environment,
    - bin_gcc directory,
    - bin_icc directory,
    - bit streams directory,
    - CONF directory.

```
cd S2MRCPBG_test_environment ; ls –l bin_gcc bin_icc
```

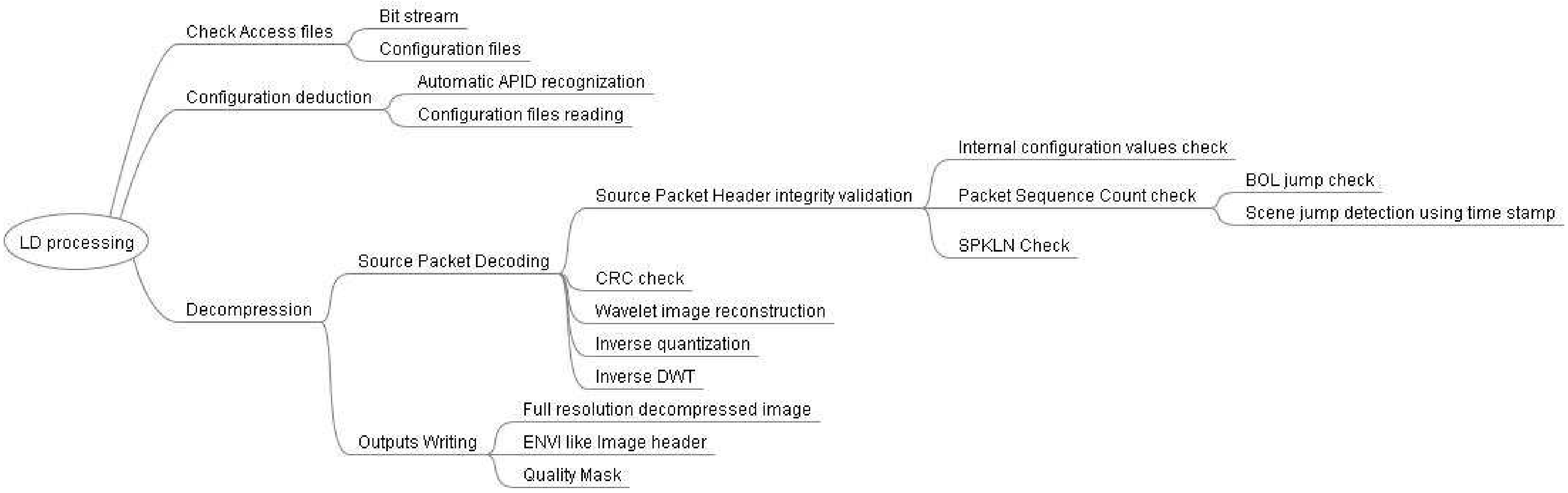
  - Following files are present:
    - mrcpbg_sol,
    - Extraction album,
    - libQL.so,

- script_validation_S2MRCPBG.sh,
- script_validation_S2QL.sh.

3. Global overview of the software

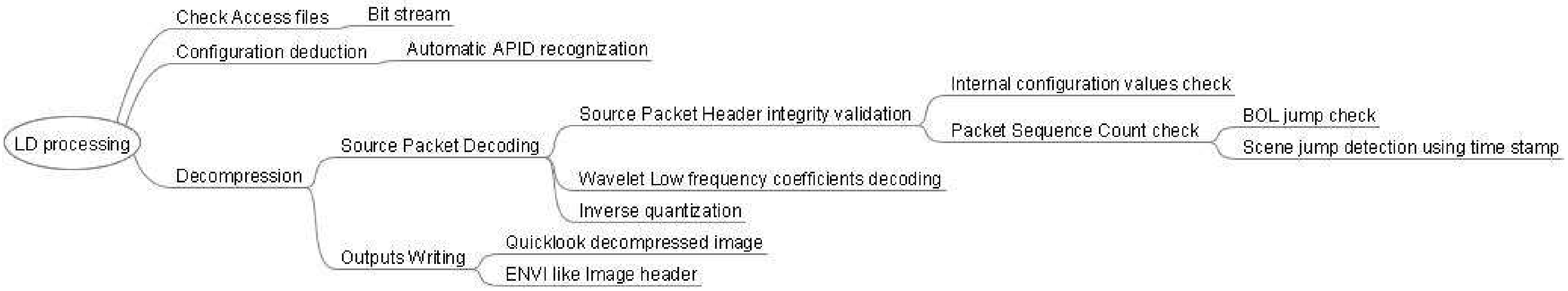
3.1. S2MRCPBG Decompression Steps

The following scheme describes the overall processing executed by S2MRCPBG to decompress:


LD processing
Check Access files
Bit stream
Configuration files
Configuration deduction
Automatic APID recognization
Configuration files reading
Decompression
Source Packet Decoding
Source Packet Header integrity validation
Internal configuration values check
Packet Sequence Count check
BOL jump check
Scene jump detection using time stamp
SPKLN Check
CRC check
Wavelet image reconstruction
Inverse quantization
Inverse DWT
Outputs Writing
Full resolution decompressed image
ENVI like Image header
Quality Mask

3.2. S2QL Decompression Steps

The following scheme describes the overall processing executed by S2QL to decompress:


LD processing
Check Access files
Bit stream
Configuration deduction
Automatic APID recognization
Decompression
Source Packet Decoding
Source Packet Header integrity validation
Internal configuration values check
Packet Sequence Count check
BOL jump check
Scene jump detection using time stamp
Wavelet Low frequency coefficients decoding
Inverse quantization
Outputs Writing
Quicklook decompressed image
ENVI like Image header

3.3. Explanation of main steps

The following table explains the main objective of the main steps presented in the previous schemes:

Step/Sub-step	Description
S - Configuration deduction	Initialization of all the parameters of the software with the Sentinel 2 context. **S2MRCPBG** configuration is fixed using four configurations files. Some of these parameters can be changed on the command line which has the priority over the configuration files. In this step, the APID of the first source packet is used to automatically configure the parameters included in the configuration file "defaut.ih". The association mechanism between "defaut.ih" file and APID is controlled by the configuration file "defaut.apid". Other parameters are initialized by reading the two other configuration files "defaut.fpft", "defaut.conf". All the configuration files and parameters are described in chapter 5.1. As for **S2QL** configuration, it is automatically deduced from APID and S2 context.
SS – Source Packet Decoding	This sub step realizes both integrity check of the bit stream to be decompressed and the decompression. The integrity check consists here in validating the sequence of PSC (Appendix G) using its number in the header of the source packet, the maximal number of STRIP for Sentinel 2 scenes, the time stamp of each source packet and the time period to acquire source packets (control of scene jumps). The other steps are dedicated to the decompression process of the compressed data. Sub processes are directly symmetric with the compression process. It includes decoding, inverse quantization and inverse DWT. **S2MRCPG** applies the entire decoding process. **S2QL** only decodes low frequency coefficients.
SS – Outputs writing	This sub step is related to the output produced by the software. In the Sentinel 2 context, **S2MRCPBG** outputs are the full resolution decompressed image in the RAW or LUM format (see Appendix D), the associated quality mask (separately or inside the decompressed image as the first column) and optionally the ENVI like header which simplify image visualisation. As for **S2QL**, outputs are the quicklook image (low resolution image) and optionally the ENVI like header which simplify image visualisation

4. Sentinel2 Mission information

4.1. Generic information about the Sentinel 2 mission

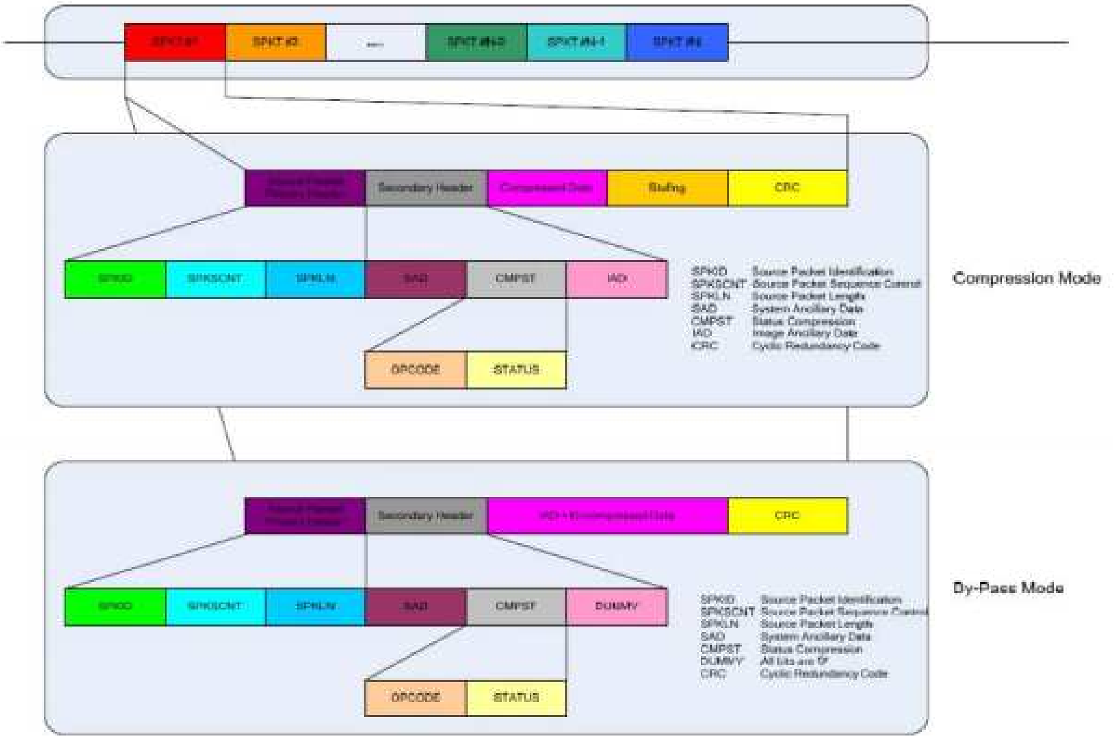
The 13 bands implemented in MSI are described in the following table:

Band (implementation)	Detection type	SSD
B2	VNIR	10m
B8		10m
B3		10m
B4		10m
B5		20m
B6		20m
B7		20m
B8A		20m
B1		60m
B9		60m
B10	SWIR	60m
B11		20m
B12		20m

Formatted CCSDS source packets contain the following:

- Primary Header,
- Secondary Header,
- Data: compressed,
- Stuffing Bytes (with compressed data),
- CRC.

An overview of CCSDS source packet implementation on MSI for both data transmission (compressed or uncompressed) is described in the next figure:


Secondary Header
Stufing
CRC
SPKSCNT
CMPST
IAD
OPCODE
STATUS
SPKID Source Packet Identification
SPKSCNT Source Packet Sequence Control
SPKLN Source Packet Length
SAD System Ancillary Data
CMPST Status Compression
IAD Image Ancillary Data
CRC Cyclic Redundancy Code
Compression Mode
Secondary Header
CRC
SPKSCNT
SAD
CMPST
DUMMY
OPCODE
STATUS
SPKID Source Packet Identification
SPKSCNT Source Packet Sequence Control
SPKLN Source Packet Length
SAD System Ancillary Data
CMPST Status Compression
DUMMY All bits are 0
CRC Cyclic Redundancy Code
By-Pass Mode

A strip is a set of 16 contiguous lines corresponding to one detector and one spectral band. A CCSDS packet corresponds to data of a strip. One strip per detector and per band is transmitted in one CCSDS source packet. The instrument transmits complete source packets only and S2MRCPBG can only decompress complete CCSDS source packets. If S2MRCPBGware encounters an incomplete packet, there will be a CRC check error and a lost of synchronisation to continue the decompression. An incomplete packet has to be detected before decompression processing; then it has to be:

- Either deleted: Black STRIP generation level at decompression,
- Either 0 stuffed: CRC error at decompression but the decompression can go on and the quality mask will contains that the source packet is wrong.

Each scene consists of a deterministic number of CCSDS source packets as specified in the next table:

SSD	**Number of packets (strips) per detector and band**	**Number of detectors**	**Number of bands**	**Number of CCSDS packets**	**Bands**
10m	144	12	4	6912	B2, B8, B3, B4
20m	72	12	6	5184	B5, B6, B7, B8a, B11, B12
60m	24	12	3	864	B1, B9, B19

4.2. Primary header useful information for the LD

The following table highlights information extracted from the source packet PH at decompression time:

Name	Remarks
APID	This Application identifier permits to identify a band and its compression parameters.
Sequence Flag	"11" unsegmented
Packet Sequence Count	The number of the first packet for each band is 0. For an image acquisition, this count is reset on first strip of each scene because of the non overlapping compression mode. The allowed range is: - 0… 143 for the 10m band, - 0…71 for the 20m band, - 0…23 for the 60m band.
Packet Length	Number of octets in the packet data field minus one.

4.3. Secondary header useful information for the LD

The unique useful information extracted from the source packet SH at decompression level is scene_start_time tag. It is used conjointly with the scene_time_period parameter to check the consistency of source packets and scenes taking into account that:

- Each source packet has the same scene_start_time inside a scene,
- The difference between the scene_start_time of two consecutives scenes is equal to scene_time_period parameter value.

4.4. Data filed of compressed data

Compressed Data	Stuffing	CRC
The Compressed Data Field contains the Embedded Bit stream (EBB) of one strip (16 lines), which includes all and only the compressed data information. The EBB contains the compressed information to describe an image area up to the requested image quality as set by minimum quantization step. The EBB data is	The Stuffing Field is necessary, if the number of compressed data is lower than the target size. In this case, Stuffing Bits will be added, to reach exactly the target size (a).	The CRC calculation is performed by the following formula: - The generator polynomial is: x16 + x12 + x5 + 1 - The encoder is initialized to the "all ones" state for each packet.

organized in a fixed order attempting to place the most important information. The EBB of each rate regulated image area is truncated at the target EBB length, thereby loosing least important information.		
Note (a): Target size is defined by the target compressed bit rate and the number of pixels for each strip. It is both an Upper limit and a lower limit. Stuffing bits are inserted in strips with very low complexity such as desert or oceans. Stuffing allows memory management with deterministic file sizes and telemetry with deterministic file transmission durations.		

Stuffing bits are checked at decompression level; they must have a 0 value.

CRC is recalculated at decompression time to validate the global integrity of each source packet.

5. Sentinel 2 configuration

5.1. S2MRCPBG Configuration files description

5.1.1. Format of configuration files

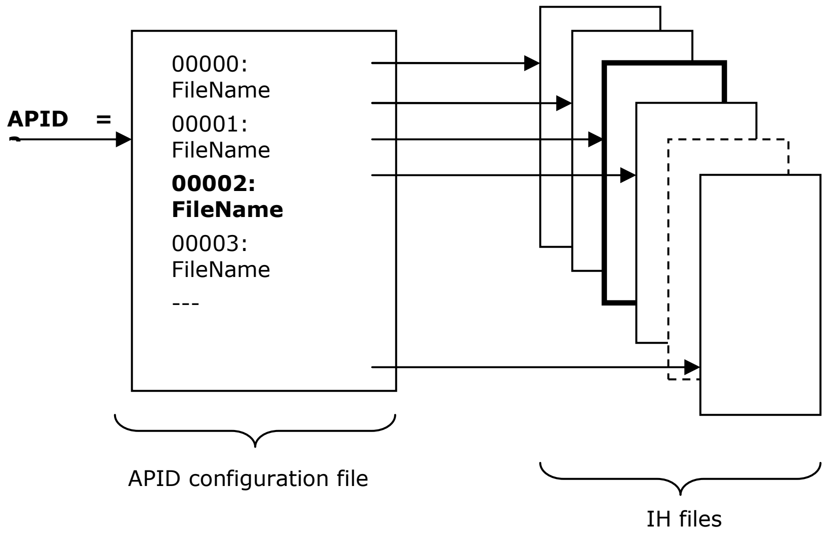
The four parameter files used to configure S2MRCPBG are in text format and are based on the following principles:

- Everything that follows the "#" character is considered to be commentary and is not read (whether a whole line or part of a line),
- The file type is specified by a keyword:
  - "fic_conf" for the .conf file,
  - "fic_apid" for the .apid file,
  - "fic_ih" for the .ih file,
  - "fic_fpft" for the .fpft file,
- The keyword "fin" marks the end of the parameter file,
- Parameter specification adheres to the following rule:

"keyword" "space/tab" "value".

5.1.2. APID and IH file association

The APID number is a field in each header of the source formats to be decompressed.
The association between the IH file and APID number is as shown in the following diagram:



5.1.3. S2MRCPBG configuration files access

For Sentinel-2 usage, S2MRCPBG uses 4 configuration files:

- Named “defaut.conf”, “defaut.ih”, “defaut.fpft”, “defaut.apid”;
- Located in the directory “../CONF” relatively to the directory of the S2MRCPBG executable “mrcpbg_sol”.

To answer, multiple configurations for one global case, it is possible to specify which are the configuration files to use specifying their full path:

Representation	`--fconf <`**`path/file.conf`**`> in the command line`
Signification	Definition of the configuration file “.conf” to use
Possible choices	A “defaut.conf” like text file
Defined in	`Only accessible on the command line`
Examples	`--fconf /users/s2mrcpbg/CONF/defaut_s2mrcpbg.conf`

Representation	`--fih <`**`path/file.ih`**`> in the command line`
Signification	Definition of the configuration file “.ih” to use
Possible choices	A “defaut.ih” like text file
Defined in	`Only accessible on the command line`
Examples	`--fih /users/s2mrcpbg/CONF/defaut_s2mrcpbg.ih`

Representation	`--fpft <`**`path/file.fpft`**`> in the command line`
Signification	Definition of the configuration file “.fpft” to use
Possible choices	A “defaut.fpft” like text file
Defined in	`Only accessible on the command line`
Examples	`--fpft /users/s2mrcpbg/CONF/defaut_s2mrcpbg.fpft`

Representation	`--fapid <`**`path/file.apid`**`> in the command line`
Signification	Definition of the configuration file “.apid” to use
Possible choices	A “defaut.apid” like text file
Defined in	`Only accessible on the command line`
Examples	`--fapid /users/s2mrcpbg/CONF/defaut_s2mrcpbg.apid`

5.2. S2QL Configuration files description

There is no configuration file to use S2QL software.

The full set of parameters is deduced using APID information and global S2 mission context.

S2QL reads the APID in the primary header of the first source packet to deduce the parameters for one band and one detector.

QL decompression is data driven, the bit stream is decompressed reading the source packet length written in each primary header of each source packet. Unlike S2MRCPBG full resolution decompression software, S2QL do not apply any check on the read packet length.

Remarks:

- The number of columns passed to the QuickLook function is also computed from the APID field.
- The zero stuffing of missing STRIPs is always activated.
- The secondary header length passed to the QL function is fixed to 54 bytes (Sentinel2 default parameter).
- The length of the scenes in number of STRIPs is computed from the APID field in the bit stream.
- The scenes duration is set to its default value for Sentinel2: 3.5s

Note: There is no synchronization mechanism in the software, therefore if a packet size, read in the bit stream, is different from the actual size of the packet, the behaviour of the software is undefined.

Note 2: Stuffing consists in injecting 0 values in the wavelet coefficients used by the inverse wavelet transform (and not directly putting 0 values in the output image). That implies that stuffed STRIPs do not contain only 0 values in the decompressed image.

5.3. Sentinel2 fixed parameters

This chapter lists the fixed parameters for the Sentinel 2 mission and gives a brief explanation on them.

For more details on this parameters please refer on the generic user manual of MRCPBG software (RD 04) and the specification of the compression process from Astrium (RD 01 & RD 02).

- ".conf" file fixed parameters:

Keyword	Value	Description
valRecentrage	0	Reserved
mode_LD	0	Reserved
vcrc	1	CRC Verification : 0 inactive, 1 active
confirm_premier_crc	1	In case of CRC error 1 value is to continue the decompression process ignoring the CRC error 2 value is to generate a black strip corresponding to this source packet ion the output image 0 value is to set interactive mode action when CRC error is encountered. Choice 1 or 2 is

		proposed to the user at each CRC error.
confirm_crc	1	Reserved. This parameter is useful to choose which action decompression process apply when a crc is wrong (others crc than the first one in case of multiple crc). 1 value for S2 forces to use the information coded in the source packet to decompress even if crc is false. A warning is emitted in case of false CRC. 2 value discard the information contained from the erroneous part covered by the wrong crc till the last part of the source packet. **Not used in S2 context, because only one CRC.**
confirm_depassement_pS	1	In case of error in entropy decoder 0 ask user; 1 continue ; 2 stop decompression
confirm_ebbln_generique	4 (to decompress in a data driven mode)	In case EBBLNC and SPKLN not consistent 0 ask user; 1 stop ; 2 force SPKLN; 3 force EBBLNC ; 4 force SPKLN and deactivate warning dump in case iof difference between SPKLN and EBBLNC
confirm_integrite	1	In case of format integrity error: 0 ask user;1 continue
image_quicklook	0	Generation of decompressed image files 0 image only, 1 quicklook only, 2 image+quicklook 0 image only, 1 quicklook only, 2 image+quicklook
type_quantif	0	Reserved
mode_big_endian	0	Output format : 0: generates the output image with the endianness of the computer or 1: forces to generate the output image in big endian mode.
left_crop	0	Left crop of the image
right_crop	0	Right crop of the image
PSC_jump_stuffing	1	Zero stuffing of missing STRIP
english	1	Output messages language (0 = french ; 1 = english)
scene_time_period	3.608064	Scene time duration in seconds

mode_overlap	0	0 = mode non overlap; 1 = mode overlap
miroir	0 (TBC)	Reserved

- ".fpft" file fixed parameters:

Keyword	Value	Description
ModeBit stream	0	Reserved
IHP	0	Reserved
IHRP	0	Reserved
TailleSynchro	0	Reserved
ValeurSynchro	0	Reserved
SPKSHLN	54	Secondary Header Length in words of 2 bytes
NCRC	1	Number of CRC per source packet: 0 no CRC; 1 one CRC
PHP	1	Reserved

- ".ih" file fixed parameters:

Keyword	Value	Description
INBDL	0	Reserved
D	12	Dynamic of pixel in output image in bits
EDP	0	Reserved
Filter53	0	Wavelet filter: 0 Wavelet transform 97 filters; 1 Wavelet Transform53 filters
Filter97J2K	1	Reserved
ByPassPond	1	Reserved
ByPassCentre	1	Reserved
SeuilMask_HV	0	Reserved
ModeMask_HV	0	Reserved
SeuilMask_D	0	Reserved
ModeMask_D	0	Reserved
LoiAdapt	1	Reserved
ModeEBB	0	Reserved
ModeRegulation	3	Regulation mode: 0 non regulated; 2 limited; 3 regulated
P9	255.0	Reserved

P8	255.0	Reserved
P7	255.0	Reserved
P6	255.0	Reserved
P5	255.0	Reserved
P4	255.0	Reserved
P3	255.0	Reserved
P2	255.0	Reserved
P1	255.0	Reserved
P0	255.0	Reserved

5.4. S2MRCPBG Sentinel2 configurable parameters

This chapter lists the configurable parameters accessible for Sentinel 2 mission users and gives a complete explanation on them.

APID mechanism permits to automatically choose a specific band configuration fixing configurable parameters for each band and detector case.

APID Correspondence Configuration:

- ".conf" file configurable parameters:

Keyword	Value	Description
FirstBDL	1	Number of the first strip to decompress 1: decompress from beginning of file >=3: decompress from specified strip number
nbBDLout	-1	Number of strips to be decompressed -1: decompress until end of file <(NumberOfStripsInFile - FirstBDL)
rep_travail	.	Name of working directory
firstScene	1	First scene to be decompressed (used if max_boundary_band is set)
qualityMask	1	Quality Mask generation.
silent	1	Silent mode
genlog	1	Generation of log file
generr	1	Generation of error file

raw	1	Output format of FR and QL images (0 = lum ; 1 = raw)
rawHeader	1	Image header generation if raw format set (0 = no ; 1 = yes)
qualityMask	1	Generate a quality mask (0=no,1=first column of image,2= separated file)

Representation	--firstbdl <**no_STRIP**> in the command line
Signification	First STRIP (= Source Packet) to be decompressed (numbering starts at 1); relative to the beginning of a scene if firstScene is set ; i.e, firstbdl strips are jumped from the beginning of firstScene.
Possible choices	Strictly positive integer. Recommendation: specify firstbdl value as lesser or equal to max_boudary_band to only jump STRIPs inside a scene. If firstbdl value is specified greater than max_boundary_band it is a relative jump in number of strips from the first STRIP of firstScene. In the S2 context, max_boundary_band is always active due to non-overlap mode.
Defined in	“file.conf” by the keyword “FirstBDL”.
Examples	--firstbdl 3

Representation	--nbdldec <**nb_STRIP**> in the command line
Signification	Number of STRIPs to be decompressed starting with firstbdl.
Possible choices	-1=all / strictly positive integer.
Defined in	“file.conf” by the keyword “nbBDLout”.
Examples	-- nbdldec 10

Representation	-wdir **<directory>** in the command line
Signification	Name of directory in which the DECOMP, RESULT directories will be created
Possible choices	Indefinite.
Defined in	“file.conf” by the keyword “rep_travail”.
Examples	--wdir /home/comp/mrcpbg/simus

Representation	`--qualityMask <`**`0,1,2`**`>`
Signification	Generation of a quality mask at line granularity.
Possible choices	0=no / 1=yes, as the first column of output image(same coding system as image pixels=unsigned short 16 bits) / 2=yes in a separated file (one byte per line).
Defined in	`« file.conf » by the keyword « qualityMask ».`

Examples	`--qualityMask 0`
	`--qualityMask 1`

- ".ih" file configurable parameters:

Keyword	Value	Description
INCOL	1296 or 2592	Number of columns; can also be forced by command line
max_boundary_band	144, 72, 24 for 10m, 20m, 60m	Number of source packets for each band contained in one scene

Representation	`--ncol <`**`number_columns_image`**`> in the command line`
Signification	Number of columns (width) in the image to be decompressed.
Possible choices	Positive integers. 1296 or 2592 for Sentinel 2 mission.
Defined in	`"file.ih" by the keyword "INCOL".`
Examples	`--ncol 1528`

Representation	`--max_boundary_band <`**`nb STRIPs`**`> in the command line`
Signification	Size of the scenes in number of STRIPs
Possible choices	Positive integer > 3 and < 2^14-1. 144, 72, 24 for 10m, 20m, 60m for Sentinel 2 mission.
Defined in	`« file.ih » by the keyword « max_boundary_band ».`
Examples	`--max_boundary_band 144`

- Command line configurable parameters:

Synoptique	`--basenom <`**`basename`**`>` « basename » is the name keyword you specify to replace <basename> in the naming procedure of S2MRCPBG (for example <BaseName>_1str_to97_cons0.log considering the name of the log file).
Signification	Basename of the compression software outputs.
Possible choices	String.
Defined in	--basenom <basename_without_any_estension>
Examples	`--basenom simu_23-06-2012 so the log file name will be simu_23-06-2012_1str_to97_cons0.log`

Remark:

- This parameter is only accessible on the command line as other ones but it is the only one which shall be used.
- Suffix "_0" is used because EBBLNC value is fixed to 0 in the Sentinel-2 configuration file.

5.5. S2QL parameters

In order to work properly, ExtractionAlbum requires six parameters:

- Input file (S2QL can only decompress one band detector file at a time, exactly like S2MRCPBG software),
- Output file,
- First scene to be decompressed,
- First STRIP to be decompressed,
- Number of STRIPs to be decompressed,
- Mirror flag value.

"Input file" is the path to the bit stream file to be decompressed.

"Output file" is the path to the output image, its file extension has to be one of the following 3:

- lum: to produce a LUM formatted image (Warning: this format cannot be used to write the output image in an Unix pipe file).
- raw: to produce a RAW formatted image,
- hdr: to produce a RAW formatted image and an associated HDR/ENVI file.

"First scene to be decompressed" is the first scene to be decompressed, it starts at 1.

"First STRIP to be decompressed" is the number of the first STRIP in the scene that will be decoded by the QuickLook function, it starts at 1.

"Number of STRIPs to be decompressed" is the number of STRIPs that will be decoded by the QuickLook function (special value -1 means to decode all the STRIPs).

"Mirror flag" allows the user to flip the output image. This activation flag permits to produce a vertically mirrored image as output (0= mirror is inactive, 1 = mirror is set).

6. Command Lines

6.1. S2MRCPBG command line

To introduce this chapter, notice that input file based mode and input pipe based mode are managed internally by S2MRCPBG software. This means that these 2 possibilities are transparent for the end user considering the command line execution.

6.1.1. Shell command line execution

In Sentinel 2 context, the representation of the command line is as follows:

```
mrcpbg_sol <bit stream.bin> [--Option <Parameters>]
```

Items between <> are to be instantiated with real names or numbers.

6.1.2. Unix/Linux pipe use

Unix/Linux Pipe are only compatible with the raw output format.

In computing, a named pipe (also known as a FIFO for its behaviour) is an extension to the traditional pipe concept on Unix and Unix-like systems, and is one of the methods of inter-process communication. A traditional pipe is "unnamed" because it exists anonymously and persists only for as long as the process is running. A named pipe is system-persistent and exists beyond the life of the process and must be deleted once it is no longer being used. Processes generally attach to the named pipe (usually appearing as a file) to perform inter-process communication.

Instead of a conventional, unnamed, shell pipeline, a named pipeline makes use of the filesystem. It is explicitly created using mkfifo() or mknod(), and two separate processes can access the pipe by name — one process can open it as a reader, and the other as a writer.

To use Unix pipes as input with S2MRCPBG, one can create a pipe with the exact name of the input file of "mrcpbg" to read the binary stream from it ; the extension ".bin" is mandatory because this extension is checked on the input by S2MRCPBG:

```
mkfifo KC300_00_01.bin
```

and run mrcpbg:

```
mrcpbg KC300_00_01.bin
```

In a separate process shell, independently, one could then dump the compressed data into the pipe **with complete source packets**:

```
cat source_packet_XXX.bin > KC300_00_01.bin
```

Remark: the run order of the two last commands doesn't impact the possibility to decompress. If mrcpbg is run before, it will wait for data to decompress else if cat is called first it will wait for pipe reading by mrcpbg when the pipe is full before going on with the cat process himself.

To use Unix pipes as output with S2MRCPBG, one can create a pipe with the exact name of the output file of "mrcpbg" to decompress the image into it ; by default the OUTPUT directory is ./DECOMP, it is so necessary to create the directory before if needed and create the output pipe inside:

```
if [ ! -d ./DECOMP ]; then mkdir ./DECOMP; fi
mkfifo ./DECOMP/KC300_00_01_1str_to97_dcmp_cons5841.raw
```

and run mrcpbg normally:

```
mrcpbg KC300_00_01.bin
```

In a separate process shell, independently, one could then retrieve the decompressed data from the pipe:

```
cat KC300_00_01_1str_to97_dcmp_cons5841.raw > my_image.raw
```

The named pipe can be deleted just like any file:

```
rm KC300_00_01_1str_to97_dcmp_cons5841.raw
```

6.1.3. Parallel execution

Parallel executions of S2MRCPBG on the same directory on different input bit streams/pipes or output pipes are possible. In case of parallel partial extractions on the same input bit stream (only in the file based mode; it is not possible with pipe because of their FIFO behavior), the workdir directories have to be different not to overwrite the output. Partial extraction output name does not contain a specific suffix; i.e two different partial extractions on the same input bit stream produce the same name for the output.

6.1.4. Mirror option configuration

Due to focal planes organization, mirror option has to be treated by the encapsulation program of S2MRCPBG software.

The focal plane configurations highlight that:

- Even VNIR detectors (2, 4, 6, 8, 10, 12) have to be mirrored (pixels index shall be inverted: the last pixel is now number 1),
- Odd SWIR detectors (1, 3, 5, 7, 9, 11) have to be mirrored (pixel index shall be inverted: the last pixel is now number 1).

Mirror option on the command line is **mandatory** in the Sentinel 2 context following the previous rules. This means that:

- To decompress bit streams relative to even VNIR detectors and odd SWIR detectors, it is mandatory to specify on the command line "--miroir 1":

```
mrcpbg_sol <bit stream.bin> --miroir 1
```

- To decompress all other bit streams (odd VNIR and even SWIR detectors), it is mandatory to specify on the command line "--miroir 0":

```
mrcpbg_sol <bit stream.bin> --miroir 0
```

6.2. S2QL command line

6.2.1. Schell command line execution

In Sentinel 2 context, the representation of the command line is as follows:

```
 ExtractionAlbum <input_file.bin> <output_file.lum|raw|hdr>
<First_Scene> <First_STRIP> <number_of_STRIPs> <mirror>
```

Items between <> are to be instantiated with real names or numbers.

Detailed parameters description is given in chapter 5.5.

- **input_file.bin** : Input bit stream to decompress,
- **output_file.lum|raw|hdr**: Output file,
- **First_Scene**: First scene to be decompressed,
- **First_STRIP**: First STRIP to be decompressed,
- **number_of_STRIPs**: Number of STRIPs to be decompressed,
- **mirror: flag to activate the output image vertical flip**.

6.2.2. Mirror flag configuration

As mentioned in the S2MRCPB chapter (6.1.4); mirror flag is driven by band number and detector number:

- Even VNIR detectors (2, 4, 6, 8, 10, 12) have to be mirrored (pixels index shall be inverted: the last pixel is now number 1),
- Odd SWIR detectors (1, 3, 5, 7, 9, 11) have to be mirrored (pixel index shall be inverted: the last pixel is now number 1).

Mirror option on the command line has to be set following the previous rules. This means that:

- To decompress bit streams relative to even VNIR detectors and odd SWIR detectors, it is mandatory to set mirror flag on the command line to 1:

```
ExtractionAlbum <input_file.bin> <output_file.lum|raw|hdr>
<First_Scene> <First_STRIP> <number_of_STRIPs> 1
```

- To decompress all other bit streams (odd VNIR and even SWIR detectors), it is mandatory to set mirror flag on the command line to 0:

```
ExtractionAlbum <input_file.bin> <output_file.lum|raw|hdr>
<First_Scene> <First_STRIP> <number_of_STRIPs> 0
```

7. Software outputs

7.1. S2MRCPBG outputs

Execution of S2MRCPBG:

- Sends the Unix/Linux shell a return code corresponding to the following cases:
  - 0: no error detected during entire execution.
  - 1: one error occurred which shut down the software.
  - 2: one or more warnings appeared during execution.
  - 3: one error tied to the software parameterization occurred. Error file might be empty. Run the same configuration in silent mode 0 in order to get more details.
  - 4: software silent mode is incompatible with interactive mode of operations whereby some questions are raised to the user on the terminal. This return code should however never happen in the Sentinel-2 specified configuration..
- Displays information on the screen (see below).
- Generates a decompressed image in the wdir/DECOMP directory,
- Generates a set of information files in the wdir/RESULT directory, (wdir is the work directory defined in the software’s configuration).

The above-mentioned on-screen information and generated files are detailed in the following paragraphs.

7.1.1. Screen

```
mrcpbg_sol BIN/KC300_00_01.bin
```

With the above command line and considering that “silent” mode is not activated in the “conf” configuration file, the display is as follows:

```
---------------------------------------------------------------
              GENERIC DECOMPRESSION SOFTWARE (LD)
                  (v9.0  -  December 09th 2011)
         Copyright (c) 2006 Cnes. All rights reserved.
---------------------------------------------------------------

LD parameters :
---------------
 Working mode                           : generic
 Decompression base                     : string
 Configuration File                     : ./../CONF/defaut.conf
 Transmit Format Parameters File        : ./../CONF/defaut.fpft
 Image Header File                      : ./../CONF/defaut.ih
 APID/Image Header Files matching File  : ./../CONF/defaut.apid

Image Parameters :
------------------
 Input Bit stream                        : BIN/KC300_00_01.bin
 Number of columns / Dynamic            : 1296 / 12 bpp
 Decompressed output Image              : .//DECOMP/KC300_00_01_1str_to97_dcmp_cons5841.raw

Reverse Wavelet Transform parameters :
----------------------------------------------------
 DWT : 9/7 JPEG2K

Decoding parameters :
---------------------
 Reverse Quantization by centroid       : no
 Visual Mask HV / D                     : no / no
 Coder adaptation table                 : 1

Information Files :
------------------------
 Execution summary                      :
 error log                              :

Initialization ........................................

Processing STRIP :   288 (185.24 STRIP/s)
End of bit stream : BIN/KC300_00_01.bin

End of decompression ...............................

-----------------------------------------------------------
```

The screen display is split into sections informing the user of the software’s current options. Each section contains all the files and parameters used and is displayed only if it is used.

If “silent” mode is activated, nothing is written on the screen.

7.1.2. Files produced in S2 context

- The name of the unsigned short full resolution and full decompressed image is formed as follows:

  <BaseName>_1str_to97_dcmp_cons0.lum|.raw

- The name of the error file is formed as follows:

  < BaseName >_1str_to97_cons0.err

- The name of the log file is formed as follows:

  < BaseName >_1str_to97_cons0.log

- The name of the quality mask independent file is formed as follows (value option qualityMask 2):

  < BaseName >.mask

With:

BaseName: basename(bit stream to be treated) or specified with the --basenom option

con0: this text corresponds to "cons" juxtaposed with the value of EBBLNC (fixed to 0 in the default Sentinel 2 configuration file).

_1str: fixed for Sentinel2 mission; means only one bit stream has be decompressed = mono ASIC mode

_to97: fixed for Sentinel2 mission; means that we use 9-7 wavelet filters for the DWT

_dcmp:fixed string (always added and never modified)

7.1.2.1..log file format

ASCII file containing a copy of the screen information *(See 7.1.1*) when the option "genlog" is activated (1 value). In case of deactivation of this parameter (0 value), no log file is produced.

In case of silent mode is activated, nothing is written on the console. If log file is required, the content corresponds to the console information if it was required. Log file generation is compatible with silent mode because "silent mode" only controls information written in the console.

7.1.2.2..err file format

ASCII file containing any error messages or warnings (as described in 7.1.3) when the option "generr" is activated (1 value). In case of deactivation of the parameter (0 value), no error file is produced.

Here, it is an example of a generated ".err" file:

```
     0        WarningSTRIPSize                Warning         Read size (5857) diff.
Regul. Size 5858  (string 0)        answer : 2
     1        WarningSTRIPSize                Warning         Read size (5857) diff.
Regul. Size 5858  (string 0)        answer : 2
```

The first item is related to the source packet number.

The second item is a generic short keyword highlighting where the problem occurred.

The third one classifies the message as error or warning.

The fourth one explains briefly the encountered problem.
The last one specifies the action done when the problem occurs.

The following table lists all the keyword used in the ".err file":

Keyword	Correspondence with listed messages below
`SystemError`	SO – Generic error when calling the LD
`CLIParameter`	SO - Erroneous command Line
`SoftwareConfiguration`	SO – Configuration file(s) not found
`Bit streamFile`	M001
`STRIPout`	M002
`PSCError`	M007
`ErrorCRCfailed`	M004 – M005
`Packet/MissingFS`	M012
`WarningStuffing`	M010
`ErrorEBBLN`	M006
`CrcSH/DACO`	M008
`WarningCRC`	M004 – M005
`WarningRICE`	M009
`WrongPSC`	M007
`WarningSTRIPSize`	M011
`WarningAlignment`	M010
`UnknownApid`	M003

7.1.3. Software Error messages

Errors are detected at the earliest:
on entering parameters, an unexpected value triggers display of the LD's mode of use,
during execution.
Two types of messages appear on the standard output (stdout):

```
-------------------------------------------------------------
the following error made the software stop :
%s                          (*)
SYSTEM ERROR : %s   (**)
-------------------------------------------------------------
```

or:

```
-------------------------------------------------------------
l'erreur the following error made the software stop :
%s                          (*)
INTERNAL ERROR : %s         (***)
-------------------------------------------------------------
```

(*) optional message specifying the error context.
(**) system message relating to the error detected (e.g. file not found).
(***) specific message relating to the error detected (see message).

An additional message is sent to stderr:

```
Program aborted !
```

All of these messages are also written in the error file. A distinction is made between errors that shutdown the software and warnings that allow it to continue operating.

ID M	Message	Description	Step/SUB step related
FRM001	Cannot open %s for reading	Appears when it is impossible to open the bit stream in read mode.	Check Access files – Bit Stream
FRM002	Number of STRIP requested is not coherent. Extracting %d STRIP.	Appears when the number of requested STRIP is out of range (outside of the bit stream).	Check Access files – Bit Stream
FRM003	APID %d not found in APID file	Appears when S2MRCPBG cannot associate the APID read in the bit stream with its configuration.	Check Access files – Configuration files
FRM004	Computed CRC %d (%04X) diff. CRC in bit stream string %d (%04X)	Appears when a CRC error is detected.	Souce Packet decoding - CRC check
FRM005	First CRC fragment is not valid in the bit stream string %d. Would you force the STRIP %d to 0 ? (y/n)	Appears when the first CRC of the block (BLD) is false.	Source Packet decoding - CRC check
FRM006	Read size (%d) diff. Regul. Size %d (string %d)	Appears in the generic branch when the SPKLN read does not correspond to the setting defined in the parameter in the regulated or non-regulated mode.	Source Packet decoding – Source Packet Header integrity validation - SPKLN Check
FRM007	Read PSC (%d) diff. (%d) (string %d)	Appears when S2MRCPBG detects a PSC jump in the bit stream using scene_start_time and scene_time_period information.	Source Packet decoding – Source Packet Header integrity validation – PSC check – STRIP jump check
FRM008	CRC in a SH+DACO (string %d)	Appears when S2MRCPBG detects the presence of a CRC in a SH+DACO block.	Source Packet decoding - CRC check
FRM009	S path overflow	Appears when a RICE event to be decoded exceeds the size of the block (STRIP).	Source Packet decoding – Wavelet image reconstruction
FRM010	Decod. align. Not empty	Appears when S2MRCPBG	Source Packet decoding –

	(STRIP end)	detects non-null alignment bits at the end of a block (STRIP) in the bit stream.	Wavelet image reconstruction
FRM011	Too much alignment bits (STRIP end)	Appears when S2MRCPBG calculates that it has more than 15 alignment bits for a block (STRIP).	Source Packet decoding – Wavelet image reconstruction
FRM012	decoded start scene (%f) diff. (%f) (string %d)	Appears when S2MRCPBG detects an error in the currently decoded scene time stamp (depending on the previously decoded ones and the duration of scenes parameter)	Source Packet decoding – Packet Sequence Count Check
FRM012	`Cannot set "qualityMask"` WITHOUT "zero stuffing of missing STRIP"	Appears when you try to generate the quality mask and stuffing option is not set	S - Configuration deduction
FRM013	`Combination of parameters firstbdl and firstScene is out of range`	Appears when the combination of parameters "--firstScene" and "--firstbdl".	S - Configuration deduction

7.2. S2QL outputs

The return value of the ExtractionAlbum software is 1 if any error stopped the execution, 0 in case of success.

Execution of S2QL:

- Sends the Unix/Linux shell a return code corresponding to the following cases:
  - 0: no error detected during entire execution.
  - 1: one error occurred which shut down the software.
  - 2: one or more warnings appeared during execution.
  - 3: one error tied to the software parameterization occurred.
  -
- Generates a decompressed quicklook image in the execution directory,
- Displays error message in the screen.

The above-mentioned on-screen information and generated files are detailed in the following paragraphs.

7.2.1. Screen

```
ExtractionAlbum KC300_00_01.bin KC300_00_01.lum 1 1 -1 0
```

There is no "stdout" write in nominal mode (no accessible verbose mode as S2MRCPBG).Only error messages are dumped into "stdout" and readable on the screen console.

7.2.2. Files produced in S2 context

- The name of the quicklook image (unsigned short type) is formed as follows and doesn't use any suffix (different naming strategy from the S2MRCPBG one):

  <BaseName>.lum|.raw

- No error or log output file is created; error message are only written in the console.

7.2.3. Error messages

List array of S2QL console error messages:

ID M	Message	Description	Step/SUB step related
QLM001	Error : bin file opening failed	Appears when it is impossible to open the bit stream in read mode.	Check Access files – Bit Stream
QLM002	Error : output file opening failed	Appears when S2QL cannot create output image file.	Outputs writing
QLM003	Error : parameter First_Scene	Appears when parameter first_Scene is negative or null.	Input Parameters
QLM004	Error : parameter First_STRIP	Appears when parameter first_STRIP is negative or null.	Input Parameters
QLM005	Error : parameter number_of_STRIPs	Appears when parameter number_of_STRIPs is negative or null.	Input Parameters
QLM006	Error : Invalid APID in first packet	Appears when S2QL cannot associate the APID read in the bit stream with a S2 configuration.	Configuration deduction
QLM007	Error : parameter First_STRIP is greater than the number of STRIPs in a scene	Appears when parameter first_STRIP is greater than the number of strips present in a scene.	Configuration deduction & Input Parameters
QLM008	Warning : Invalid PSC	Warning emission when the encountered PSC is not the expected one.	Source Packet decoding – Packet Sequence Count check
QLM009	Error : Incomplete or corrupted bit stream	Appears when S2QL cannot reach the next STRIP.	Source Packet decoding – Wavelet low frequency coefficients decoding
QLM010	Warning : Invalid scene time stamp	Appears when the read time stamp is no the expected one.	Source Packet decoding – Source Packet Header integrity validation
QLM011	Warning : Invalid bit stream : error type %u	Appears when S2QL encountered an error inside the quicklook function. (see	Source Packet decoding – Wavelet low frequency coefficients decoding

		Appendix E for details)	
QLM012	Error : the first STRIP to be decompressed is greater than the number of STRIP in the bit stream file	Appears when parameter first_STRIP is greater than the number of STRIPs in a scene.	Input Parameters
QLM013	Warning : the number of STRIP to be decompressed is greater than the number of STRIP in the bit stream file from the first STRIP selected	Appears when the number of STRIP to be decompressed is greater than the number of STRIP in the bit stream file from the first STRIP selected	Input Parameters
QLM014	Error : output HDR file opening failed	Appears when it is impossible to write the ENVI like header	Outputs writing
QLM015	Error : mirror parameter	Appears when the value of mirror parameter is different from the two possible ones: 0 or 1.	Input Parameters

Appendix A - Generalized RAW format Specification

Generalized RAW format allows to format images encoded with either:

- 8 to 16 bits per pixel;
- 32 bits Floating point per pixel.

It contains uncompressed image data and optionally quality mask information (value option qualityMask 1).

Image area format:

- Pixel coded on 8 bits : 1 byte per pixel (char type) ;
- Pixel coded on 9 to 16 bits : 2 bytes per pixel (short type) :
  - Byte 1 (first in the file): (N-8) most significant bits of the pixel, where N is the number of bits per pixel. Possible values for the byte: [0 ; 2N-8 - 1],
  - Byte 2 (second in the file): 8 less significant bits. Possible values for the byte: [0 ; 255] ;

  Remark: swap byte 1 and byte 2 descriptions if endianness is Little Endian
- Pixel coded on 32 bits floats: 4 bytes per pixel (float type);
- Option: quality mask value of the current line included as the first pixel line image with the same encoding system.

“Sentinel-2 RAW format Specification" as specific case of the above general RAW format:

When required, quality mask is generated at line granularity level (and not at pixel one), i.e one value per line.

The encoding size on each value per line is different depending on the generation option used:

- Each value per line is encoded as a byte value for separate quality mask in the S2 context (i.e 8 bits encoding size),
- Each value has to be coded like the other pixels of the image when the quality mask is added as the first column of the output image. Each quality mask value is then encoded as an unsigned short (i.e, 16 bits encoding size) in the S2 context.

Appendix B - Generalized LUM format Specification

Generalized LUM format allows to format images encoded with either:

- 8 to 16 bits per pixel ;
- 32 bits Floating point per pixel.

LUM Format corresponds to raw format with an additional line as header containing information about the image and optionally quality mask information (value option qualityMask 1).

Header area format (size = 1 line of the image) :

- 1st 32 bits word: number of columns (32 bits integer) ;
- 2nd 32 bits word: number of lines (32 bits integer) ;
- 3rd 32 bits word: "pixel encoding type" field, in ASCII. Possible values are:
  - Big Endian :
    - 08BI : pixels are coded on 8 bits [0 ; 255],
    - 09BI : pixels are coded on 9 bits [0 ; 511],
    - ...,
    - 16BI : pixels are coded on 16 bits [0 ; 65535],
    - FLOA : pixels are coded on 32 bit floats;
  - Little Endian :
    - 08LI : pixels are coded on 8 bits [0 ; 255],
    - 09LI : pixels are coded on 9 bits [0 ; 511],
    - ...,
    - 16LI : pixels are coded on 16 bits [0 ; 65535],
    - FLOL : pixels are coded on 32 bit floats;
- The remaining space of the header may contain random stuff.

Image area format :

- Pixel coded on 8 bits : 1 byte per pixel (char type) ;
- Pixel coded on 9 to 16 bits : 2 bytes per pixel (short type) :
  - Byte 1 (first in the file): (N-8) most significant bits of the pixel, where N is the number of bits per pixel. Possible values for the byte: [0 ; 2N-8 - 1],
  - Byte 2 (second in the file): 8 less significant bits. Possible values for the byte: [0 ; 255] ;

  Remark: swap byte 1 and byte 2 descriptions if endianness is Little Endian
- Pixel coded on 32 bits floats: 4 bytes per pixel (float type);
- Option: quality mask value of the current line included as the first pixel line image with the same encoding system.

“Sentinel-2 LUM format Specification" as specific case of the above general LUM format:

- 8 bits per pixel is used for separate quality mask in the S2 context,
- 16 bits per pixel is used for the decompressed image in the S2 context.

Appendix C - Generalized ENVI LIKE HEADER format Specification

This appendix describes the ENVI header format:

```
ENVI
description = {
File Imported into ENVI. }
samples = 1624
lines   = 1616
bands   = 1
header offset = 0
file type = ENVI Standard
data type = 12
interleave = bsq
sensor type = Unknown
byte order = 0
wavelength units = Unknown
```

This example is related to an image of:

- 1624 columns(“sample”),
- 1616 lines(“lines”),
- containing only one spectral band (“bands”),
- with no additional information before the image data (“header offset”),
- pixels are coded on unsigned short (“data type”) and in little endian (“byte order”).

Excepted the number of columns and lines, this example is fully compliant with the header for Sentinel 2 mission.

Optionally (value option qualityMask 1), the output image can contain quality mask information. This additional data correspond exactly to one additional column in the image data. In this case the ENVI header column information is increased of 1.

Appendix D - Quality MASK Generation Process description

This appendix presents the algorithm for defining quality index on each output line (synchronized with the output image lines). The line quality index generation is strip based. Each strip impacts the quality of his own 16 lines and the 27 margin lines of its previous and following strips.

Here are listed the possible values for a line:

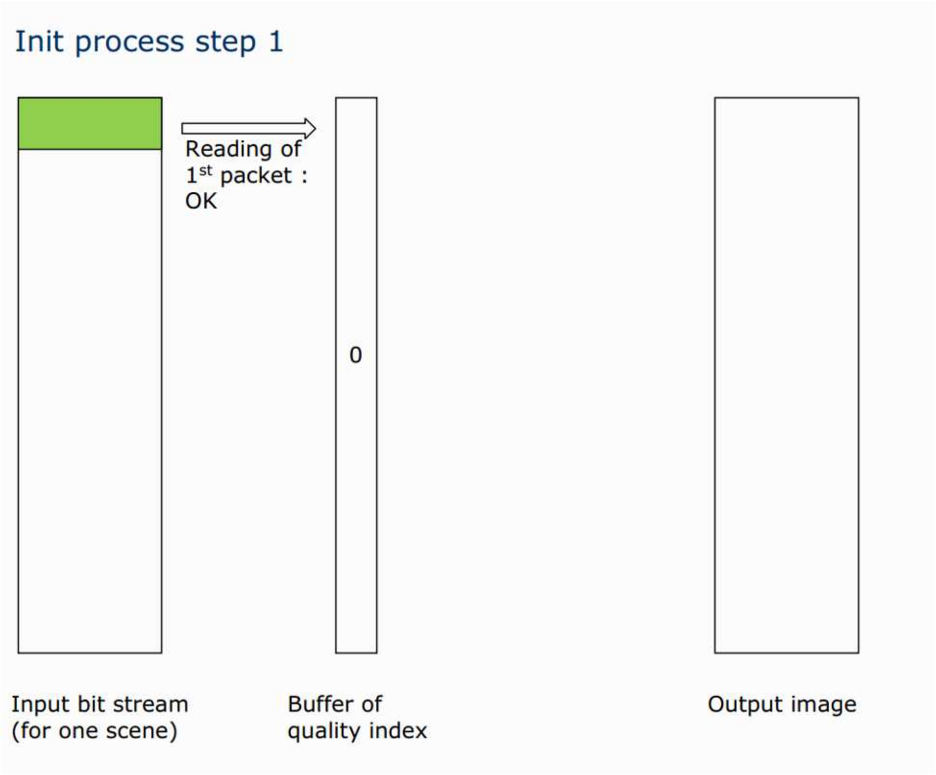
- 0: decompression is OK
- 1: decompression is not OK - error decoding
- 3: decompression is not OK - missing packets
- 2: decompression is OK, degraded line from previous errors

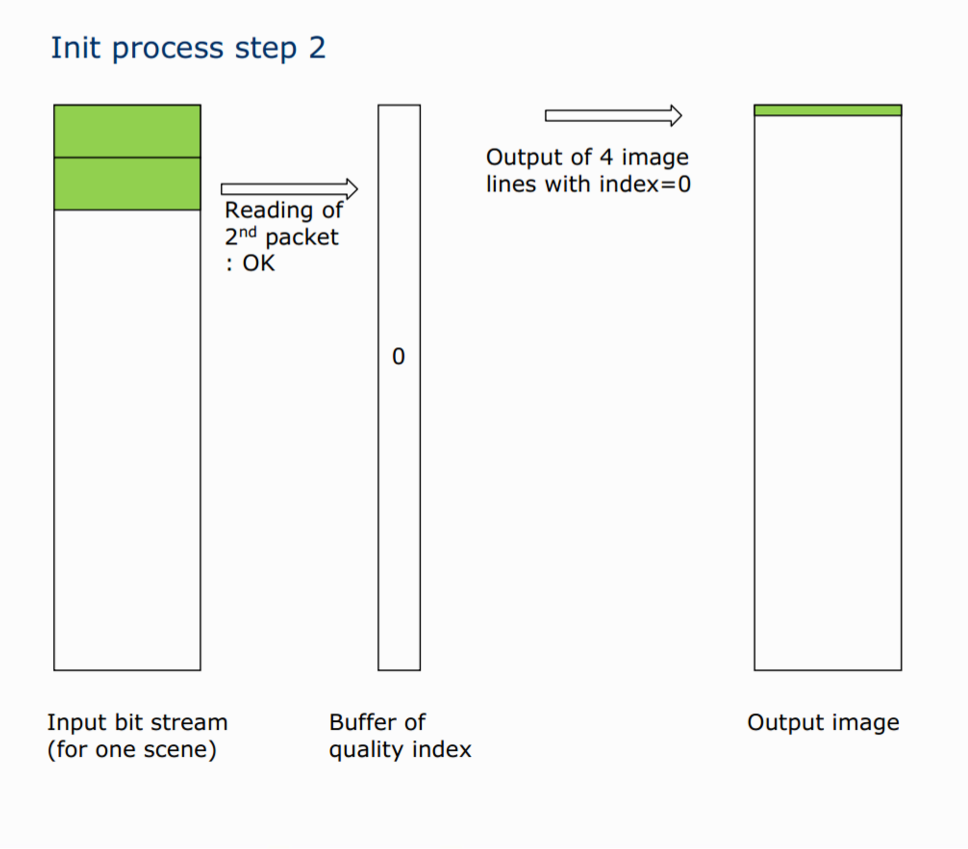
The generation process of the quality mask is described below:

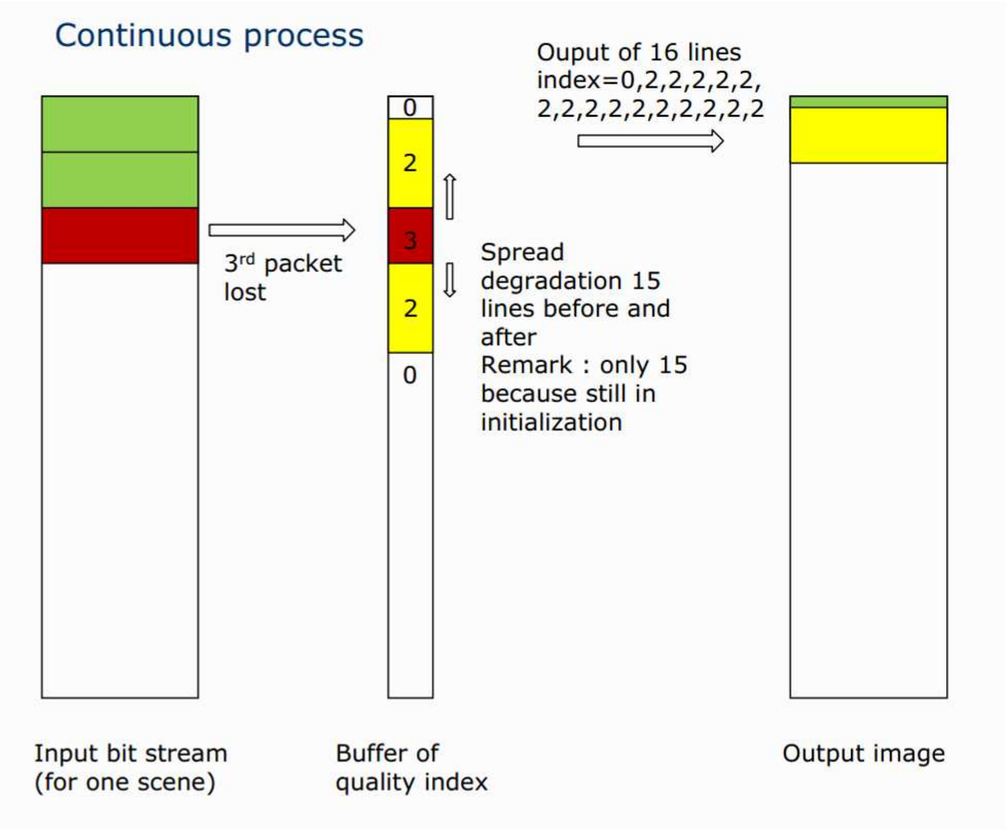
- Definition of buffer quality indexes for the scene (one index for each line) initialized at 0.
- On each input packet, the buffer is filled with:
  - 16 index not changed if decoding is OK
  - 16 index at 1 or 3 if decoding is NOK (1) or packet is lost (3)
    - => 27 indexes before the packet set to 2 (degraded) if actual value is 0
    - => 27 indexes after the packet set to 2.
- On each output of line, the corresponding line index is retrieved.

The quality mask can be integrated in the output decompressed image as the first column of the image with the pixel format (raw unsigned short) or in a separate binary file in raw unsigned byte format with one value by image output line.

Here it is an example if the quality mask generation process:


Init process step 1
Reading of 1st packet : OK
0
Input bit stream (for one scene)
Buffer of quality index
Output image


Init process step 2
Reading of 2nd packet : OK
0
Output of 4 image lines with index=0
Input bit stream (for one scene)
Buffer of quality index
Output image



Continuous process

Ouput of 16 lines
index=0,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2

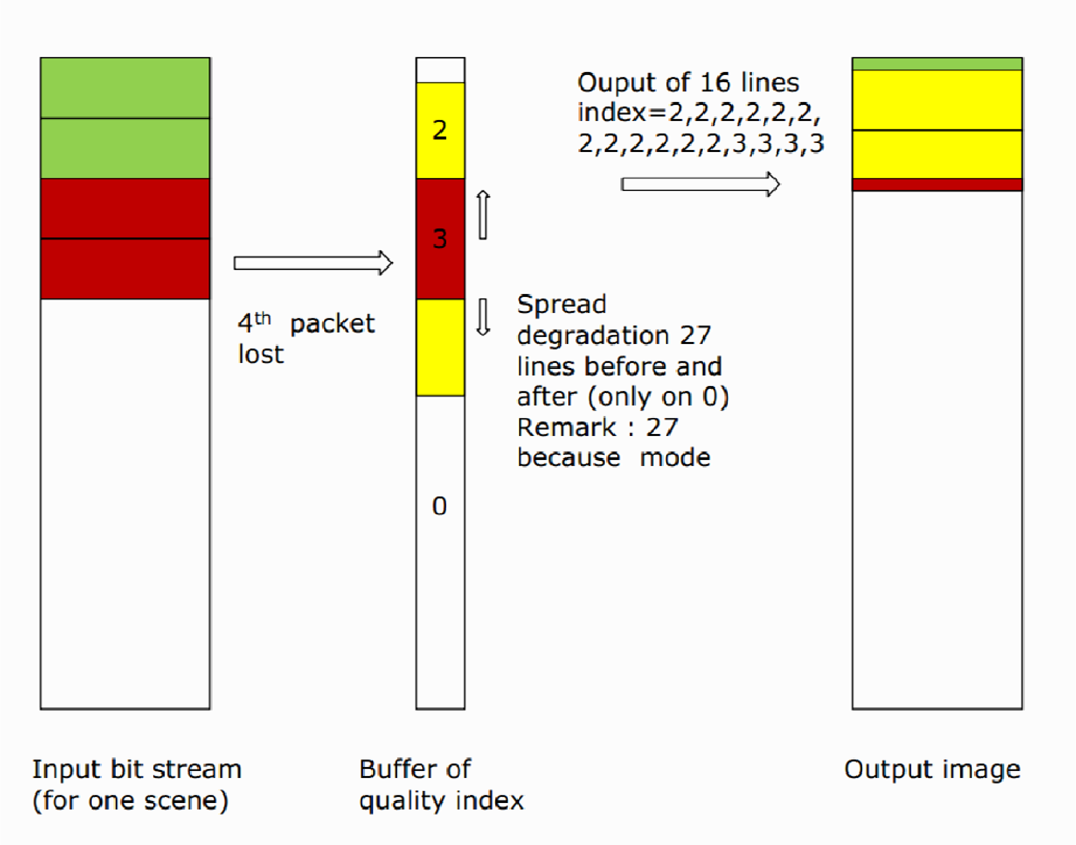
3rd packet
lost

Spread
degradation 15
lines before and
after
Remark : only 15
because still in
initialization

Input bit stream
(for one scene)

Buffer of
quality index

Output image



Ouput of 16 lines
index=2,2,2,2,2,2,
2,2,2,2,2,2,3,3,3,3

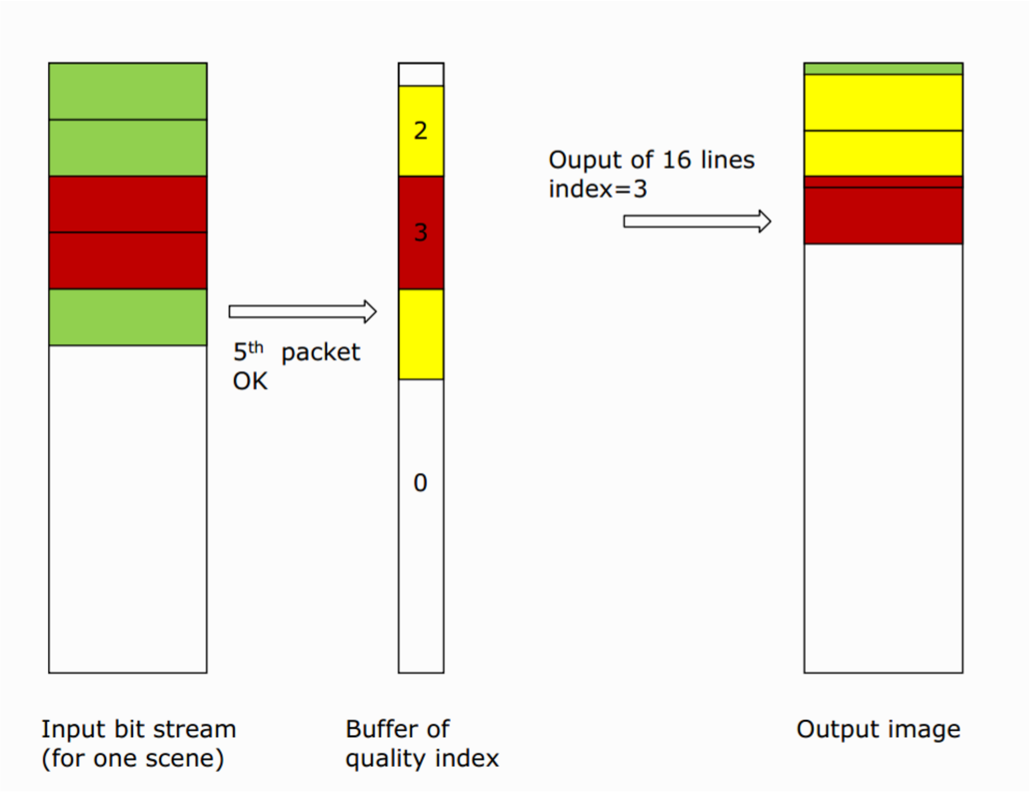
4th packet
lost

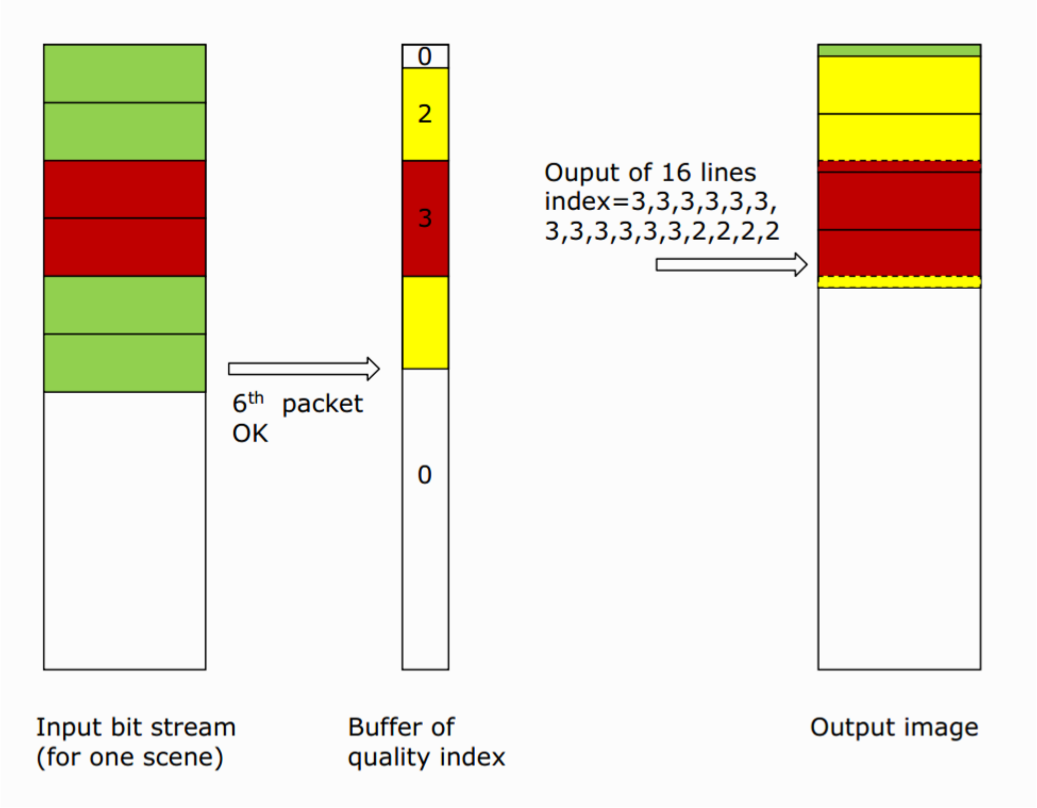
Spread
degradation 27
lines before and
after (only on 0)
Remark : 27
because mode

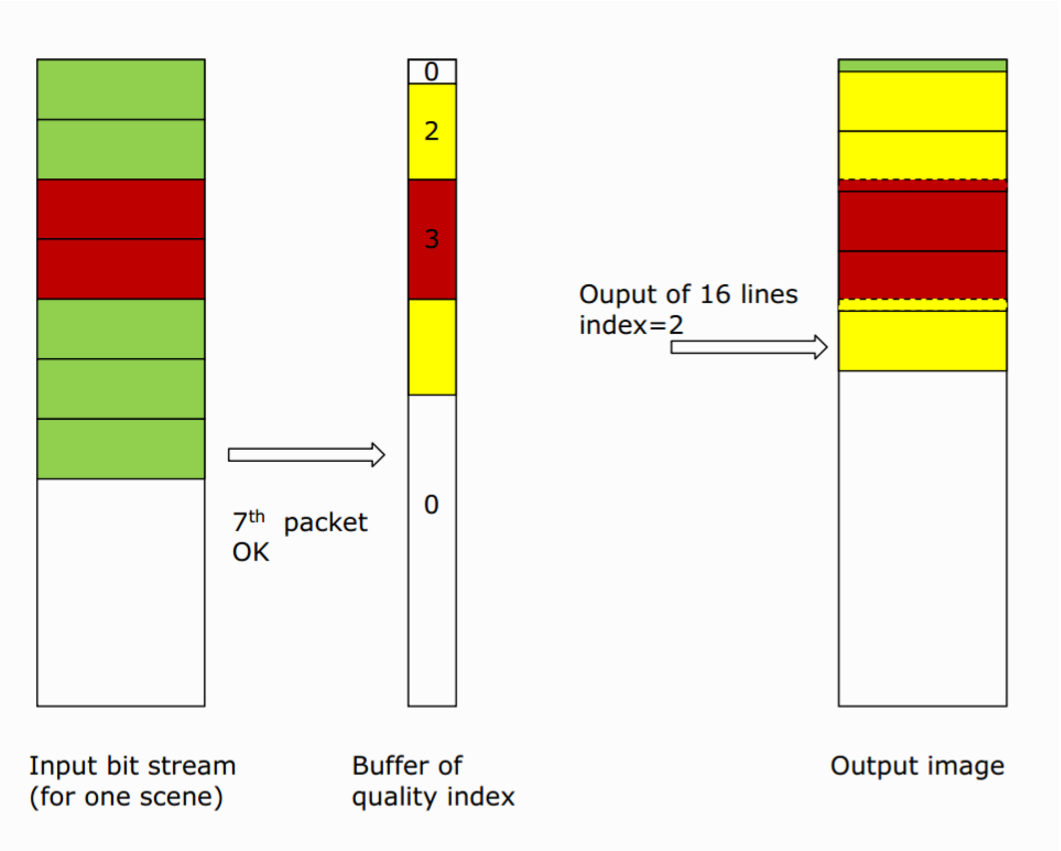
Input bit stream
(for one scene)

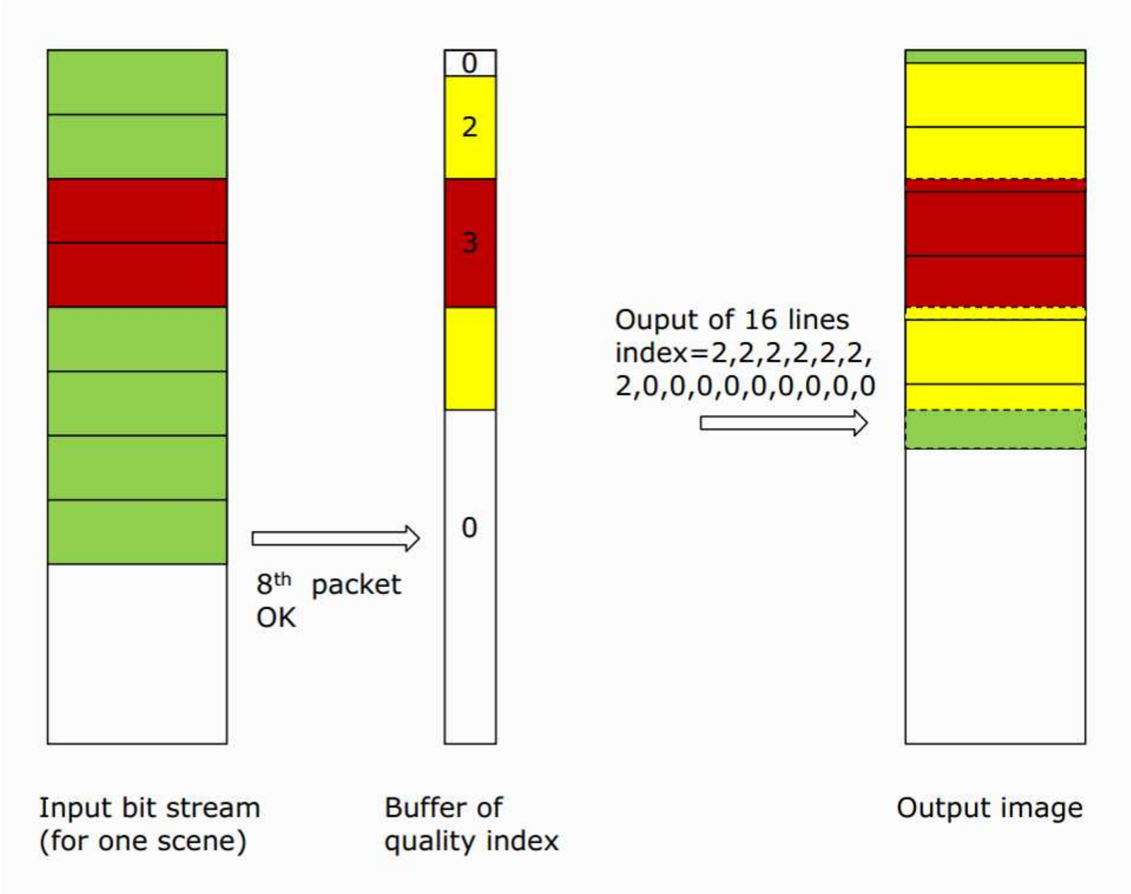
Buffer of
quality index

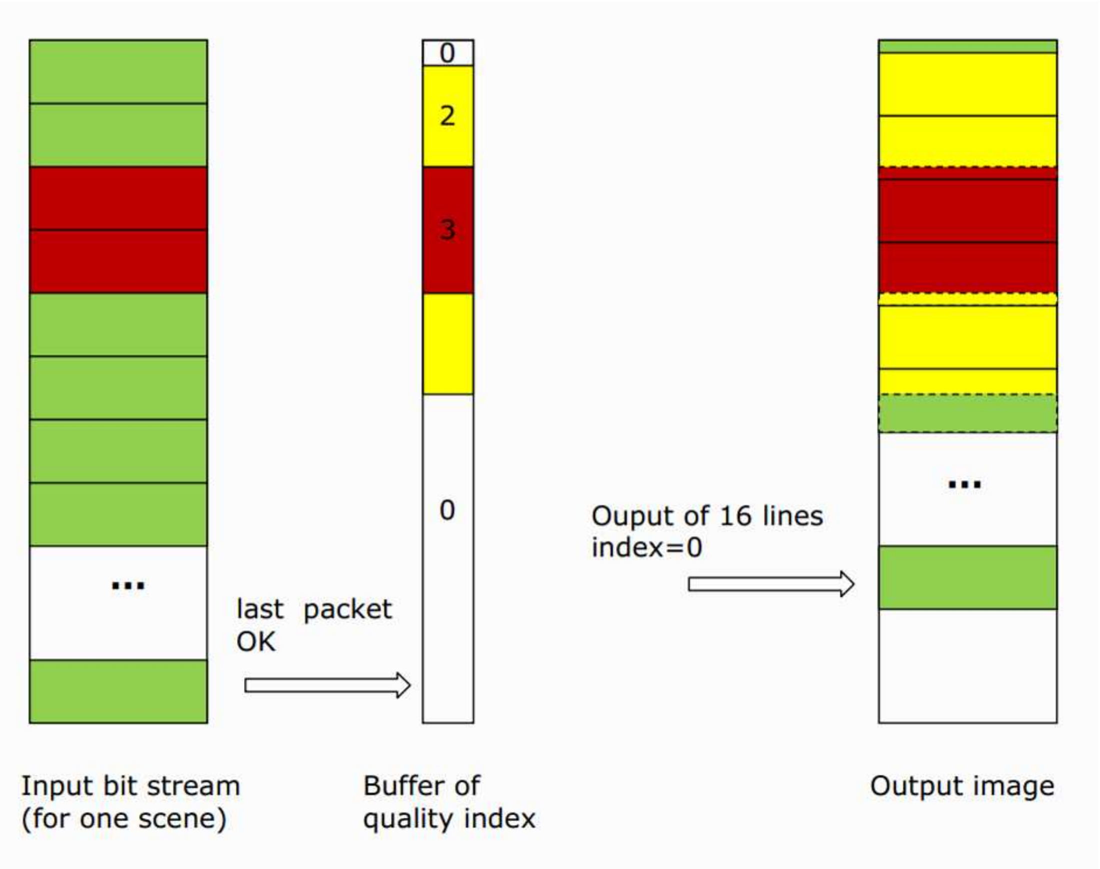
Output image


2
3
0
5th packet
OK
Ouput of 16 lines
index=3
Input bit stream
(for one scene)
Buffer of
quality index
Output image

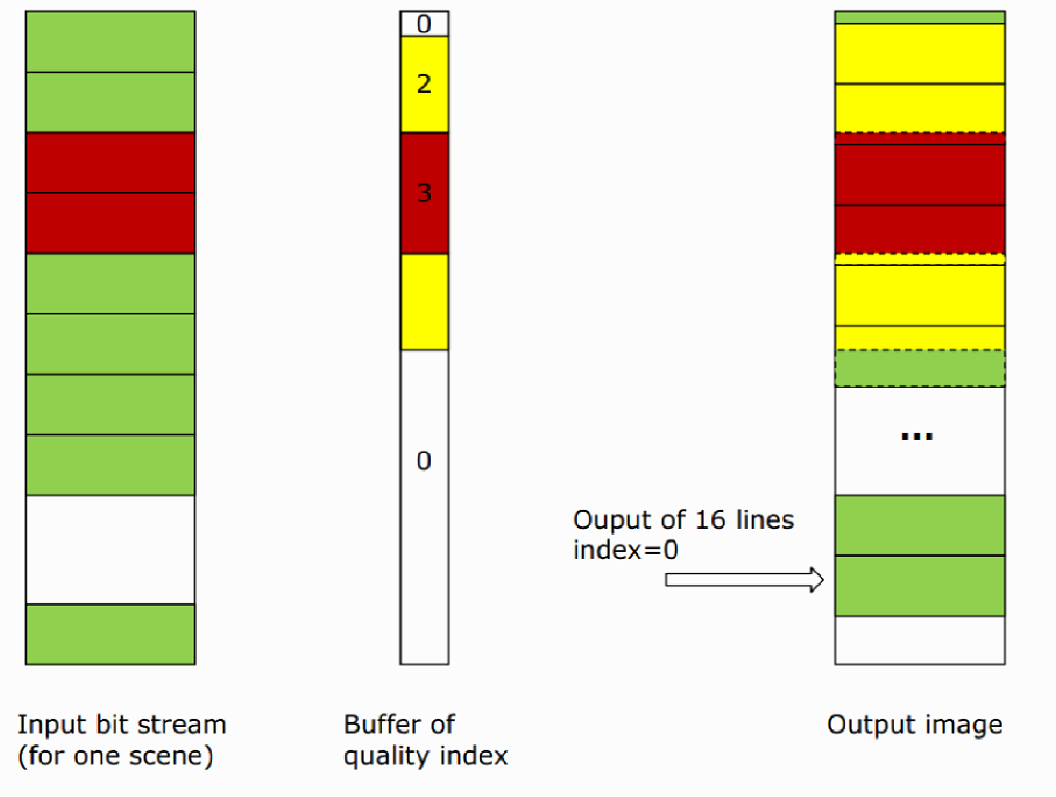

0
2
3
0
6th packet
OK
Ouput of 16 lines
index=3,3,3,3,3,3,
3,3,3,3,3,3,2,2,2,2
Input bit stream
(for one scene)
Buffer of
quality index
Output image


0
2
3
0
Ouput of 16 lines
index=2
7th packet
OK
Input bit stream
(for one scene)
Buffer of
quality index
Output image

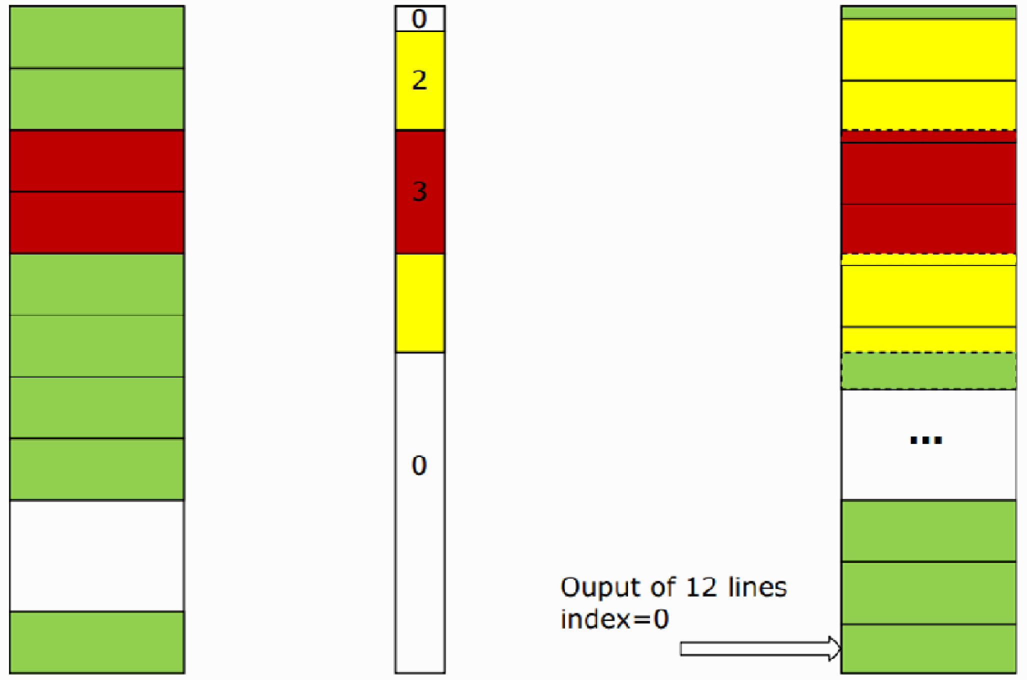

0
2
3
0
Ouput of 16 lines
index=2,2,2,2,2,2,
2,0,0,0,0,0,0,0,0
8th packet
OK
Input bit stream
(for one scene)
Buffer of
quality index
Output image



« buffers emptying » process step 1



« buffers emptying » process step 2



Ouput of 12 lines
index=0

Input bit stream
(for one scene)

Buffer of
quality index

Output image

Appendix E – Quicklook Function

This appendix explains how to use the QuickLook function, located in the dynamic library "libQL.so":

```
      extern int QuickLook (unsigned char *Packet_In, unsigned int * QL_Out, int
nbcol, int shln, int mirror);
```

In order to work properly, the QuickLook function requires five parameters:

- unsigned char * Packet_In,
- unsigned int * QL_Out,
- int nbcol,
- int shln,
- int mirror.

"Packet_In" contains a complete data packet, i.e.:

- the "primary header" (Packet Identification, Packet Sequence Count, Packet Length),
- the "secondary header",
- the bit stream data,
- the CRC.

Each packet from a given bit stream must therefore be retrieved and stored in "Packet_In" (as an unsigned char) and applied to the QuickLook function.

"QL_Out" then contains the output of the QuickLook function.

The array has to be previously allocated by the caller with a size of 2*nbcol/8 integers (representing two lines of nbcol/8 columns of the full resolution image).

Remark: Two lines of nbcol/8 columns is because the QuickLook function decompresses only a part of the BF to produce the QL_Out image. The compressed image uses 3 level DWT with 16 lines STRIPs. The reducing factor at each level is 2, so 16/2^3 = 2 lines and nbcol/2^3 (nbcol/8) columns.

The "QL_Out" table contains the pixels on a 12-bit dynamic range, with one pixel per integer, in the storage format of the machine executing the program (little or big endian).

Therefore, it is necessary to know the endianness of the machine executing the program (little endian for Xeon) in order to save the data correctly. (note that the ExtractionAlbum program takes care of this).

The "nbcol" parameter refers to the number of columns of the full resolution image (the full resolution image is the image which was compressed resulting in the bit stream file). It depends on the treated band of the Sentinel2 mission.

The "shln" parameter refers to the size of the secondary header in words (a word is equivalent to two bytes). Value "54" is expected for all the bands of the Sentinel 2 mission.

The "mirror" parameter is a flag (value 0 for false or 1 for true) to produce a vertically mirrored output image.

The return value of the QuickLook function is an integer that can take these different values:

- 0: successful execution,
- 1: error detected in the field CesureBF,
- 3: the sum of NPN+CesureBF is greater than 17 (NPN is the number of 0 value bit plane from the most significant bit),
- 4: the capacity of the number of S bits generated is exceeded by a type 2 event.

To give high level information about theses return codes, we can say that:

- 1 & 3 return values are error on retrieval of coefficient for decompression,
- 4: linked with the sequence of bit for coefficient to be encoded.

Appendix F – S2MRCPBG configuration files

These configuration files are based on APID mechanism and allows to treat every bands and every detector.

- defaut.conf:

```
###############################################################################

# CONF configuration file for Sentinel 2 mission

################################################################################


fic_conf

# Sentinel-2 fixed configuration

valRecentrage           0     # refocusing value
mode_LD                 0     # LD mode : 0 = GENERIC MODE ; 1 : PLEIADES MODE
miroir                 0(TBC) # output images are mirrored
confirm_premier_crc     1     # forces to continue the decompression if the first fragment of a
source packet covered by a CRC is in error
                              # 1 = forces to continue
                              # 0 = question user
                              # 2 = Black STRIP generation forced
confirm_crc             1     # forces to continue the decompression if a fragment different from
the first one of a source packet covered by a CRC is in error
                              # 1 = forces to continue
                              # 0 = question user
                              # 2 = bits from the first one included in the wrong source packet
fragment till the last one of the source packet are unused to decompress
confirm_depassement_pS  1     # forces to continue the decompression even if a problem of a S route
is encountered during decoding (too many S bits generated to reconstruct wavelet coefficients)
                              # 1 = forces to continue
                              # 0 = question user
                              # 2 = LD stops
confirm_ebbln_generique 4     # forces to stop the decompression when an error is encountered on the
EBBLNC (bitrate check)
                              # 1 = forces to stop decompression at he STRIP in error
                              # 0 = question user
                              # 2 = use SPKLN read in the header of the source packet to decompress
                              # 3 = use the EBBLNC specified in the configuration file or in the
command line to decompress
                              # 4 = use SPKLN and deactivate EBBLNC SPKLN check
image_quicklook       0       # QL image generation
fichier_psh           0       # activate SH decoding in an output text file
type_quantif          0       # activate inverse quantization using centroid method

mode_big_endian       0 # forces output image in big endian, even on a little endian platform
left_crop             0 # crop on the left of the image (left column cutting)
right_crop            0 # crop on the right of the image (right column cutting)
PSC_jump_stuffing     1 # activates black STRIP generation when a PSC jump is detectedenglish
1 # Software language (0 = francais ; 1 = anglais)
scene_time_period     3.608064 # scene time stamp in seconds
mode_overlap          0 # overlap mode management; 0 = mode non overlap ; 1 = mode overlap
```

```
# Sentinel-2 configurable parameters with their default value

silent                   0    # silent mode
FirstBDL                 1    # first STRIP to be decompressed (index begin at 0)
nbBDLout                -1    # number of STRIPs to be decompressed (-1 = decompress all the STRIP
present in the bit stream)
firstScene            1 # first scene to be decompressed; only used if max_boundary_band different
from 0
raw                   1 # outout format of the FR and QL generated images (0 = lum ; 1 = raw)
rawHeader             1 # ENVI like header generated if raw format sopecified (0 =  non ; 1 = oui)
qualityMask           1 # quality mask production (0 =  no production ; 1 = production in the output
image ; 2 = production as an independent file)
genlog                1 # output log file generation in RESULT Directory
generr               1 # output error file generation in RESULT Directory
rep_travail            .      # working directory name

fin
```

defaut.ih for 10m bands:

```
##############################################################################

# IH 10m configuration file for Sentinel 2 mission

###############################################################################


fic_ih

# Sentinel-2 fixed configuration

INBDL                         0      # nummber of STRIPS of the output image (0 means decompress
                              all)
D                             12     # dynamic of the pixels
EDP                           0      # exponent divisor pixel
Filter53                      0      # filters TO 5/3 (0 = use 9/7 filters; 1 = uses 5/3 ones)
Filter97J2K                   0      # filters 9/7 management (0 = ASIC ones, 1 = JP2K ones)
ByPassPond                    1      # ponderation Bypass
ByPassCentre                  1      # refocusing Bypass
SeuilMask_HV                  0      # masking treshold for H and V wavelet coefficients
                                     # 0 = no masking, treshold masking 1/2/3
ModeMask_HV                   1      # activates HV masking
                                     # 0 = 1 bit masking / 1 = 2 bits masking
SeuilMask_D                   0      # masking treshold for D wavelet coefficients
                                      # 0 = no masking, treshold masking 1/2/3
ModeMask_D                    0      # activates D masking
                                     # 0 = 1 bit masking / 1 = 2 bits masking
LoiAdapt                      1      # adaptation law for the Run Length Rice coding
ModeEBB                       0      # activates EBBImage mode ( 0 = STRIP mode ; 1 = Full Image
                              mode)
ModeRegulation                 3     # regulation mode of the source packet (1 : no regulation ; 2 =
                              limited mode ; 3 = regulated mode using EBBLNC)
EBBLNC                        10455  # Dummy fixed value for Sentinel 2
P9                            256.0  # ponderation value P9
P8                            256.0  # ponderation value P8
P7                            256.0  # ponderation value P7
P6                            256.0  # ponderation value P6
P5                            256.0  # ponderation value P5
P4                            256.0  # ponderation value P4
P3                            256.0  # ponderation value P3
P2                            256.0  # ponderation value P2
```

```
P1                           256.0   # ponderation value P1
P0                           256.0   # ponderation value P0

# Sentinel-2 configurable parameters with their default value

INCOL                        2592    # number of column of the output image
max_boundary_band 144 # Scene lenght in number of STRIP (management of the non overlap mode)


fin
```

defaut.ih for 20m bands:

```
###############################################################################

# IH 20m configuration file for Sentinel 2 mission

################################################################################


fic_ih

# Sentinel-2 fixed configuration

INBDL   0      # nummber of STRIPS of the output image (0 means decompress all)
D              12     # dynamic of the pixels
EDP            0      # exponent divisor pixel
Filter53       0      # filters TO 5/3 (0 = use 9/7 filters; 1 = uses 5/3 ones)
Filter97J2K    0      # filters 9/7 management (0 = ASIC ones, 1 = JP2K ones)
ByPassPond     1      # ponderation Bypass
ByPassCentre   1      # refocusing Bypass
SeuilMask_HV   0      # masking treshold for H and V wavelet coefficients
                      # 0 = no masking, treshold masking 1/2/3
ModeMask_HV           1      # activates HV masking
                      # 0 = 1 bit masking / 1 = 2 bits masking
SeuilMask_D    0      # masking treshold for D wavelet coefficients                            # 0
= no masking, treshold masking 1/2/3
ModeMask_D     0      # activates D masking
                      # 0 = 1 bit masking / 1 = 2 bits masking
LoiAdapt       1      # adaptation law for the Run Length Rice coding
ModeEBB        0      # activates EBBImage mode ( 0 = STRIP mode ; 1 = Full Image mode)
ModeRegulation                3      # regulation mode of the source packet (1 : no regulation ; 2 =
limited mode ; 3 = regulated mode using EBBLNC)
EBBLNC         5841   # Dummy fixed value for Sentinel 2

P9             256.0  # ponderation value P9
P8             256.0  # ponderation value P8
P7             256.0  # ponderation value P7
P6             256.0  # ponderation value P6
P5             256.0  # ponderation value P5
P4             256.0  # ponderation value P4
P3             256.0  # ponderation value P3
P2             256.0  # ponderation value P2
P1             256.0  # ponderation value P1
P0             256.0  # ponderation value P0


# Sentinel-2 configurable parameters with their default value

INCOL          1296   # number of column of the output image
max_boundary_band 72 # Scene lenght in number of STRIP (management of the non overlap mode)
```

```
fin
```

defaut.ih for 60m bands:

```
###############################################################################
# IH 60m configuration file for Sentinel 2 mission
################################################################################

fic_ih
# Sentinel-2 fixed configuration

INBDL  0       # nummber of STRIPS of the output image (0 means decompress all)
D              12     # dynamic of the pixels
EDP            0      # exponent divisor pixel
Filter53       0      # filters TO 5/3 (0 = use 9/7 filters; 1 = uses 5/3 ones)
Filter97J2K    0      # filters 9/7 management (0 = ASIC ones, 1 = JP2K ones)
ByPassPond     1      # ponderation Bypass
ByPassCentre   1      # refocusing Bypass
SeuilMask_HV   0      # masking treshold for H and V wavelet coefficients
                      # 0 = no masking, treshold masking 1/2/3
ModeMask_HV           1       # activates HV masking
                      # 0 = 1 bit masking / 1 = 2 bits masking
SeuilMask_D    0      # masking treshold for D wavelet coefficients                            # 0
= no masking, treshold masking 1/2/3
ModeMask_D     0      # activates D masking
                      # 0 = 1 bit masking / 1 = 2 bits masking
LoiAdapt       1      # adaptation law for the Run Length Rice coding
ModeEBB        0      # activates EBBImage mode ( 0 = STRIP mode ; 1 = Full Image mode)
ModeRegulation                3      # regulation mode of the source packet (1 : no regulation ; 2 =
limited mode ; 3 = regulated mode using EBBLNC)
EBBLNC         5841   # Dummy fixed value for Sentinel 2
P9             256.0  # ponderation value P9
P8             256.0  # ponderation value P8
P7             256.0  # ponderation value P7
P6             256.0  # ponderation value P6
P5             256.0  # ponderation value P5
P4             256.0  # ponderation value P4
P3             256.0  # ponderation value P3
P2             256.0  # ponderation value P2
P1             256.0  # ponderation value P1
P0             256.0  # ponderation value P0

# Sentinel-2 configurable parameters with their default value

INCOL          1296   # number of column of the output image
max_boundary_band 24 # Scene lenght in number of STRIP (management of the non overlap mode)


fin
```

defaut.fpft:

```
##############################################################################

# FPFT APID configuration file for Sentinel 2 mission

###############################################################################


fic_fpft

ModeBit stream 0      # formatting mode of the bit stream
IHP            0      # image header presence flag
IHRP           0      # image header repetition presence flag
TailleSynchro  0      # synchronization pattern length when bit stream mode is activated
ValeurSynchro  00     # synchronization pattern value
SPKSHLN        54     # secondary header length
NCRC            1     # number of CRC by STRIP
PHP             1     # primary header poresence flag (source packet header)

Fin
```

- defaut.apid:

This configuration file contains the link between APID and each detector of each band.

```
##############################################################################

# APID configuration file for Sentinel 2 mission

###############################################################################

fic_apid

# 60m bands configuration
#B1
0      defaut.ih_60m
16     defaut.ih_60m
32     defaut.ih_60m
48     defaut.ih_60m
64     defaut.ih_60m
80     defaut.ih_60m
256    defaut.ih_60m
272    defaut.ih_60m
288    defaut.ih_60m
304    defaut.ih_60m
320    defaut.ih_60m
336    defaut.ih_60m

#B9
9      defaut.ih_60m
25     defaut.ih_60m
41     defaut.ih_60m
57     defaut.ih_60m
73     defaut.ih_60m
89     defaut.ih_60m
265    defaut.ih_60m
281    defaut.ih_60m
297    defaut.ih_60m
313    defaut.ih_60m
329    defaut.ih_60m
345    defaut.ih_60m

#B10
10     defaut.ih_60m
26     defaut.ih_60m
```

```
42      defaut.ih_60m
58      defaut.ih_60m
74      defaut.ih_60m
90      defaut.ih_60m
266     defaut.ih_60m
282     defaut.ih_60m
298     defaut.ih_60m
314     defaut.ih_60m
330     defaut.ih_60m
346     defaut.ih_60m

# 10m bands configuration
#B2
1       defaut.ih_10m
17      defaut.ih_10m
33      defaut.ih_10m
49      defaut.ih_10m
65      defaut.ih_10m
81      defaut.ih_10m
257     defaut.ih_10m
273     defaut.ih_10m
289     defaut.ih_10m
305     defaut.ih_10m
321     defaut.ih_10m
337     defaut.ih_10m

#B3
2       defaut.ih_10m
18      defaut.ih_10m
34      defaut.ih_10m
50      defaut.ih_10m
66      defaut.ih_10m
82      defaut.ih_10m
258     defaut.ih_10m
274     defaut.ih_10m
290     defaut.ih_10m
306     defaut.ih_10m
322     defaut.ih_10m
338     defaut.ih_10m

#B4
3       defaut.ih_10m
19      defaut.ih_10m
35      defaut.ih_10m
51      defaut.ih_10m
67      defaut.ih_10m
83      defaut.ih_10m
259     defaut.ih_10m
275     defaut.ih_10m
291     defaut.ih_10m
307     defaut.ih_10m
323     defaut.ih_10m
339     defaut.ih_10m

#B8
7       defaut.ih_10m
23      defaut.ih_10m
39      defaut.ih_10m
55      defaut.ih_10m
71      defaut.ih_10m
87      defaut.ih_10m
263     defaut.ih_10m
279     defaut.ih_10m
295     defaut.ih_10m
311     defaut.ih_10m
327     defaut.ih_10m
343     defaut.ih_10m

# 20m bands configuration
```

```
#B5
4       defaut.ih_20m
20      defaut.ih_20m
36      defaut.ih_20m
52      defaut.ih_20m
68      defaut.ih_20m
84      defaut.ih_20m
260     defaut.ih_20m
276     defaut.ih_20m
292     defaut.ih_20m
308     defaut.ih_20m
324     defaut.ih_20m
340     defaut.ih_20m

#B6
5       defaut.ih_20m
21      defaut.ih_20m
37      defaut.ih_20m
53      defaut.ih_20m
69      defaut.ih_20m
85      defaut.ih_20m
261     defaut.ih_20m
277     defaut.ih_20m
293     defaut.ih_20m
309     defaut.ih_20m
325     defaut.ih_20m
341     defaut.ih_20m

#B7
6       defaut.ih_20m
22      defaut.ih_20m
38      defaut.ih_20m
54      defaut.ih_20m
70      defaut.ih_20m
86      defaut.ih_20m
262     defaut.ih_20m
278     defaut.ih_20m
294     defaut.ih_20m
310     defaut.ih_20m
326     defaut.ih_20m
342     defaut.ih_20m

#B8a
8       defaut.ih_20m
24      defaut.ih_20m
40      defaut.ih_20m
56      defaut.ih_20m
72      defaut.ih_20m
88      defaut.ih_20m
264     defaut.ih_20m
280     defaut.ih_20m
296     defaut.ih_20m
312     defaut.ih_20m
328     defaut.ih_20m
344     defaut.ih_20m

#B11
11      defaut.ih_20m
27      defaut.ih_20m
43      defaut.ih_20m
59      defaut.ih_20m
75      defaut.ih_20m
91      defaut.ih_20m
267     defaut.ih_20m
283     defaut.ih_20m
299     defaut.ih_20m
315     defaut.ih_20m
331     defaut.ih_20m
347     defaut.ih_20m
```

```
#B12
12      defaut.ih_20m
28      defaut.ih_20m
44      defaut.ih_20m
60      defaut.ih_20m
76      defaut.ih_20m
92      defaut.ih_20m
268     defaut.ih_20m
284     defaut.ih_20m
300     defaut.ih_20m
316     defaut.ih_20m
332     defaut.ih_20m
348     defaut.ih_20m

fin
```

Appendix G – S2MRCPBG-S2QL User Manual

This appendix deals with the MRCPBG processing of packet sequence count error. The processing is based on stuffing missing source packets.

*** Detection

During the decompression, MRCPBG executable checks if the PSC read is the expected one:

PSC == previous_PSC + 1

If we consider that, in S2 context:

- there's a mirror at each beginning and end of scene (mirror is included in the algortihm applied by the WICOM at compression level),
- each scene begins with a PSC equal to 0

MRCPBG executable has to check that PSC is less than the max boundary (depending of the processed subband) and that the first PSC has the good value:

SCENE_BEGIN == TRUE && PSC == 0
PSC <= MAX_BOUNDARY_BAND

*** Error processing

There are 2 kinds of error:

- error detection and stuffing process: This process is used when we can consider that a source packet is missing. In this case, MRCPBG executable gives out a warning about the wrong PSC encountered and stuffs virtually the decompression buffers to fill in the encountered PSC jump. Virtually means that MRCPBG executable injects as many zero wavelet coefficients as needed to perform the inverse wavelet transform of the missing source packets. This allows introducing as many zero blocs of lines as needed in the decompressed image. Practically, because of dependency between the adjacent blocs, there are non-zero values in the first lines and last lines of the added blocs.

    - case 1 : first PSC different from 0
    - case 2 : PSC jump in the scene

- error detection and warning emission only: This process is only used when MRCPBG executable encounters a PSC not in the range for the considered subband. The only solution is to consider an error in the PSC value. MRCPBG executable only gives out a warning relative to the wrong PSC and follows the decompression as in the nominal case.

    - case 3 : PSC greater than the max boundary for the considered subband

<u>Note:</u> an error on a bloc of line can impact 2 additional blocs of lines.

<u>10m Band Example:</u>

MAX_BOUNDARY_BAND=144
Previous_PSC=132
Current_PSC=2

MRCPBG executable detects error and scene transition.
=>12 black strips generated at the end of the current scene
=> 1 black strip generated for the next scene

(144 for 10m, 72 for 20 m and 24 for 60m)