

ESA PSS-04-151 Issue 1  
September 1992

# Telecommand decoder specification

Prepared by:  
The Standards Approval Board (STAB)  
for Space Data Communications

Approved by:  
The Inspector General, ESA

**european space agency / agence spatiale européenne**  
8-10, rue Mario-Nikis, 75738 PARIS CEDEX, France

Published by ESA Publications Division,  
ESTEC, Noordwijk, The Netherlands.

Printed in the Netherlands.

Price: E2

ISSN 0379 - 4059

Copyright © 1993 by European Space Agency

## **SPACE DATA COMMUNICATIONS PROCEDURES, SPECIFICATIONS & STANDARDS**

Space Data Communications is the subject of the PSS-04 branch of the ESA Procedures, Specifications & Standards (PSS) series. This branch is further divided into two subbranches:

- the **Space Link Standards and Protocols** subbranch (document reference nos.: ESA PSS-04-1XX)
- the **Spacecraft Data Interfaces and Protocols** subbranch (document reference nos.: ESA PSS-04-2XX)

The purpose of these Space Data Communications PSS documents is to ensure the compatibility of spacecraft TT & C subsystems with the relevant ESA infrastructure (i.e. the ESA (ESOC) tracking and data-communication network and the ESA (ESTEC) satellite check-out facilities).

**DOCUMENT CHANGE RECORD**

<b>Issue number and date</b>	<b>Sections affected</b>	<b>Remarks</b>
Issue 1, September 1993	New document	Complies with (and completes) Reference [2]

## REFERENCES

- [1] **Radio Frequency and Modulation Standard (ESA PSS-04-105)**, Issue 1, 1989, or later compatible issue<sup>(\*)</sup>, European Space Agency.
- [2] **Packet Telecommand Standard (ESA PSS-04-107)**, Issue 2, 1992, or later compatible issue<sup>(\*)</sup>, European Space Agency.
- [3] **Packet Telemetry Standard (ESA PSS-04-106)**, Issue 1, 1988, or later compatible issue<sup>(\*)</sup>, European Space Agency.

---

NOTE(\*) The expression "**or later compatible issue**" is meant to cover the following possibility:

It may happen that any of the documents given as references have been published under a new issue number since the date of publishing of this issue of ESA PSS-04-151. In such a case, the changes that caused any such document to be re-issued have been recognised to be of no consequence to this issue of ESA PSS-04-151, which has therefore remained unchanged.

**PAGE INTENTIONALLY LEFT BLANK**

## TABLE OF CONTENTS

<b>SECTION 1</b>	<b>PURPOSE AND SCOPE</b>	<b>1</b>
	1.1 PURPOSE	1
	1.2 SCOPE	1
<b>SECTION 2</b>	<b>APPLICABLE DOCUMENTS</b>	<b>3</b>
<b>SECTION 3</b>	<b>OVERVIEW OF THE TELECOMMAND SUBSYSTEM REQUIREMENTS</b>	<b>5</b>
3.1	SCOPE OF THE REQUIREMENTS	5
3.2	OPERATIONAL REQUIREMENTS	5
	3.2.1 Dual Access and Redundancy	5
	3.2.2 Monitoring via Telemetry	6
	3.2.3 Readiness and Accessibility	6
	3.2.4 Automated Sequence-Controlled Service	7
	3.2.5 Multiplexing of Data Streams	8
	3.2.6 Authentication of Command Data	8
3.3	DATA COMMUNICATIONS REQUIREMENTS	9
	3.3.1 Probability of Errors	9
	3.3.2 Maximum Uplink Symbol Rate	10
3.4	IMPLEMENTATION REQUIREMENTS	11
	3.4.1 Physical Layer (Symbol Stream) Interface Capability	11
	3.4.2 Telemetry Interface Capability	11
	3.4.3 MAP Interface Capability	12
	3.4.4 Mission-Specific Data Programming Capability	12
	3.4.5 Immunity to Changes of State	13
	3.4.6 Physical Distribution of Major Functions	13
	3.4.7 Cross-Coupling of Major Functions with MAP Interfaces	14
	3.4.8 Technology and Electrical Characteristics	15
<b>SECTION 4</b>	<b>PHYSICAL LAYER</b>	<b>17</b>
<b>SECTION 5</b>	<b>CODING LAYER</b>	<b>19</b>
5.1	SPECIFICATION	19
	5.1.1 TC Codeblock Length	19

	5.1.2	Additional Information on the Coding Layer Procedures	19
	5.1.3	First and Last Data Transfer in the DECODE State	20
	5.1.4	Polynomial Forms of the BCH Code	21
	5.1.5	Possible Realisation of a (63,56) Modified Hamming Decoder	22
	5.2	DESIGN REQUIREMENTS	24
<b>SECTION 6</b>		<b>TRANSFER LAYER</b>	<b>29</b>
	6.1	SPECIFICATION	29
	6.2	DESIGN REQUIREMENTS	29
	6.2.1	Spacecraft Identifier	30
	6.2.2	Virtual Channel Identifier	30
	6.2.3	FARM Sliding Window Width	30
<b>SECTION 7</b>		<b>SEGMENTATION LAYER</b>	<b>31</b>
	7.1	SPECIFICATION	31
	7.2	DESIGN REQUIREMENTS	31
	7.2.1	MAP Interface Implementation Capability	31
	7.2.2	Selection of Authenticated MAPs (Authenticated MAP ID Pointer)	32
<b>SECTION 8</b>		<b>AUTHENTICATION LAYER</b>	<b>35</b>
	8.1	GENERAL SPECIFICATION	35
	8.2	THE AUTHENTICATION PROCESSOR	37
	8.2.1	Functional Concept	37
	8.2.2	The Hashing Function	37
	8.2.3	The Hard Knapsack	40
	8.2.4	The Deletion Box	42
	8.2.5	The Signature Comparator	42
	8.2.6	The Authentication Key	42
	8.3	THE SUPERVISOR	43
	8.3.1	Functional Concept	43
	8.3.2	The LAC Registers	44
	8.3.3	The Final Authorisation Function	45
	8.3.4	The Control Command Processor	45
	8.3.5	The Deletion Function	45
	8.4	FORMATS OF THE AUTHENTICATED TC SEGMENTS	46



	8.4.1	General Format	46
	8.4.2	Specific Formats	46
	8.4.3	"Dummy Segment" Control Command	47
	8.4.4	"Select Key" Control Commands	49
	8.4.5	"Load Fixed Key In Programmable Key Memory" Control Command	49
	8.4.6	"Set New LAC Count Value" Control Command	50
	8.4.7	"Change Programmable Key Block" Control Commands	50
8.5		OPERATIONAL PROCEDURES	51
	8.5.1	Introduction	51
	8.5.2	Procedures for Changing the Programmable Key Contents	51
	8.5.3	Recovery Procedures	55
8.6		DESIGN REQUIREMENTS	58
<b>SECTION 9</b>		<b>COMMAND PULSE DISTRIBUTION UNIT</b>	<b>61</b>
	9.1	GENERAL REQUIREMENTS	61
	9.2	SPECIFICATION	62
		9.2.1 Checking the TC Segment	62
		9.2.2 Checking the CPDU-specific TC Packet	63
		9.2.3 Processing the Application Data	64
9.3		DESIGN REQUIREMENTS	65
	9.3.1	Application Identifier	65
	9.3.2	Maximum Capability of the CPDU	65
	9.3.3	Execution of Command Instructions	66
	9.3.4	Data Flow Control	66
	9.3.5	Output Waveforms	67
<b>SECTION 10</b>		<b>TELEMETRY REPORTING</b>	<b>69</b>
	10.1	GENERAL REQUIREMENTS	69
		10.1.1 Status Data	69
		10.1.2 Survey Data	70
		10.1.3 Data Report Storage before Readout	70
		10.1.4 "Cold Start" Data	71
10.2		CLCW STATUS REPORT	71
	10.2.1	Specification	71
	10.2.2	Design Requirements	71
10.3		CPDU STATUS REPORT	72
	10.3.1	Specification	72

	10.3.2	Design Requirements	72
10.4		AU STATUS REPORT	73
	10.4.1	Specification	73
	10.4.2	Design Requirements	73
10.5		FRAME ANALYSIS REPORT (FAR)	74
	10.5.1	Specification	74
	10.5.2	Design Requirements	77
<b>SECTION 11 INTERFACES</b>			<b>79</b>
11.1		INTRODUCTION	79
11.2		PHYSICAL LAYER (SYMBOL STREAM) INTERFACE	79
	11.2.1	General	79
	11.2.2	Symbol Clock Signal	79
	11.2.3	Symbol Stream Signal	79
	11.2.4	"Channel Active Indication" Signal	80
	11.2.5	Maximum Symbol Rate	80
11.3		MAP INTERFACE	80
	11.3.1	General	80
	11.3.2	Clock Out (CKOUT)	83
	11.3.3	Data Set Ready (DSR)	83
	11.3.4	Data Terminal Ready (DTR)	84
	11.3.5	Segment Data (DATA)	85
	11.3.6	Abort Data Transfer (ADT)	85
11.4		TELEMETRY INTERFACE	86
	11.4.1	General	86
	11.4.2	Sampling (SAMPLING)	88
	11.4.3	Clock In (CKIN)	88
	11.4.4	Data (DATA)	89
11.5		COMMAND PULSE OUTPUTS	89
	11.5.1	Terminology	89
	11.5.2	Specification	90

## ILLUSTRATIONS

Figure 5.1	(63,56) MODIFIED HAMMING DECODER	23
Figure 8.1	FUNCTIONAL LAYOUT OF THE AUTHENTICATION UNIT	36

Figure 8.2	FUNCTIONAL DIAGRAM OF THE AUTHENTICATION PROCESSOR	38
Figure 8.3	POSSIBLE REALISATION OF THE HASHING FUNCTION	39
Figure 8.4	CONCEPTUAL DIAGRAM OF THE KNAPSACK	41
Figure 8.5	FORMATS OF AUTHENTICATION CONTROL COMMANDS (FULL TC SEGMENT)	48
Figure 8.6	ORGANISATION OF THE PROGRAMMABLE KEY MEMORY	54
Figure 11.1	MAP INTERFACE WAVEFORMS: EXAMPLE OF DATA TRANSFER WITHOUT DATA-FLOW CONTROL	81
Figure 11.2	MAP INTERFACE WAVEFORMS: EXAMPLE OF DATA TRANSFER WITH DATA-FLOW CONTROL	82
Figure 11.3	MAP INTERFACE WAVEFORMS: EXAMPLE OF AN ABORTED DATA TRANSFER (CAUSED BY BD FRAME ARRIVAL)	82
Figure 11.4	TELEMETRY INTERFACE WAVEFORMS	87
Figure A.1	EXAMPLE OF A TELECOMMAND SUBSYSTEM CONFIGURATION	92

## TABLES

Table 5.1	DECODING STRATEGY	25
Table 8.1	LIST OF AUTHENTICATION CONTROL COMMANDS	47

## APPENDICES

A	TELECOMMAND SUBSYSTEM CONFIGURATION ON BOARD AN ESA SPACECRAFT	91
B	VERIFICATION OF COMPLIANCE	93
C	DIFFERENCES WITH EARLY IMPLEMENTATIONS	105
D	GLOSSARY OF ACRONYMS	113

**PAGE INTENTIONALLY LEFT BLANK**

## 1. PURPOSE AND SCOPE

### 1.1 PURPOSE

The purpose of this Specification is to establish uniform requirements for the implementation of spacecraft telecommand systems to be flown on board ESA spacecraft.

### 1.2 SCOPE

This Specification belongs to the Space Link Standards and Protocols sub-branch of the ESA PSS branch for Space Data Communications.

The ESA telecommand concept embraces multiple layers of data communication protocol which are specified in a set of two separate documents:

- (a) The **Radio Frequency and Modulation Standard** (Reference [1]), which specifies the "physical" characteristics of the space link (uplink as well as downlink).
- (b) The **Packet Telecommand Standard** (Reference [2]), which specifies the "**data link**" characteristics of the space uplink, as well as some of the "**transport**" characteristics required for the orderly delivery of the telecommand data units proper (Telecommand Packets).

The **Packet Telemetry Standard** (Reference [3]), which specifies the data structures used on the downlink, complements the first two documents.

This Telecommand Decoder Specification is essentially a **functional design specification** which conforms to Reference [2].

References [1] and [3] are only necessary to obtain a full view of the complete space data communication system and of its operation<sup>(\*)</sup>.

---

NOTE(\*) The reader is expected to have a good knowledge of all References, and especially Reference [2].

**PAGE INTENTIONALLY LEFT BLANK**

## 2. APPLICABLE DOCUMENTS

The document applicable to the functional design of the Telecommand Decoder described in this Specification is:

- Reference [2].

Most of the technical data required for the design of the Telecommand Decoder is to be found in Reference [2]. The telecommand system layers of specific relevance are:

- the Coding Layer;
- the Transfer Layer;
- the Segmentation Layer;
- the Authentication Layer.

The Packetisation Layer is only applicable to the Command Pulse Distribution Unit (CPDU) which, although it is not related to the "data link" functions of the TC Decoder proper, is specified as a part of it – from an implementation standpoint – because of its critical role during emergency operations.

This Specification essentially deals with elements which are not addressed in Reference [2]. Many of these elements are out of the scope of Reference [2], such as some of the operational requirements (e.g. dual access and redundancy), and most of the implementation and design requirements (e.g. interfaces). Some other elements are simply missing from Reference [2], sometimes deliberately (e.g. the full specification of the Authentication Layer).

All other documents listed as **References** are mainly required at system level. They may also contain important information on interface signals. For example: PCM waveforms (e.g. NRZ-L) are specified in Reference [1].

Finally, the reader is reminded that the conventions used in this Specification are those found throughout all ESA Space Data Communications standards, and notably in Reference [2]. Particular caution should be observed when interpreting the **Bit Numbering Convention** defined in Section 3.1 of Reference [2].

**PAGE INTENTIONALLY LEFT BLANK**



### **3. OVERVIEW OF THE TELECOMMAND SUBSYSTEM REQUIREMENTS**

#### **3.1 SCOPE OF THE REQUIREMENTS**

The following telecommand subsystem requirements apply to free-flying spacecraft. From a space data communications standpoint, these spacecraft are also referred to as "conventional" spacecraft, as opposed to "Advanced Orbiting Systems" (AOS), a term that denotes the class of spacecraft making up an Earth-orbiting infrastructure (with space stations, space planes and data relay satellites). Requirements concerning AOS spacecraft are out of the scope of this Specification.

Free-flying spacecraft are typically equipped with a fully redundant Telemetry, Tracking and Command (TT&C) subsystem. The requirements governing the capability of the (tele)Command part of that TT&C subsystem are the subject of the next sections.

#### **3.2 OPERATIONAL REQUIREMENTS**

##### **3.2.1 Dual Access and Redundancy**

(Although this requirement can also be viewed as an essential subsystem implementation requirement, it is primarily an operational requirement.)

The on-board telecommand subsystem shall be fundamentally redundant, so as to offer access to the spacecraft via – typically – two separate, distinct telecommand chains, at least for what concerns the "front-end" part, that is:

- the Telecommand (TC) Decoder, as described in this document, including the Authentication Unit when this option is selected for the mission;
- the Command Pulse Distribution Unit (CPDU), for the reconfiguration of vital spacecraft functions, as described in this document.

An illustration of the **Telecommand System Configuration on board an ESA Spacecraft** can be found in **Appendix A**.

It should be noted that, if so required by a particular mission, more than two telecommand chains can be envisaged.

### 3.2.2 Monitoring via Telemetry

Constant, real-time monitoring of the Command Link Control Word (CLCW) data of the telecommand chain being used shall always be provided via telemetry. Whenever possible, constant, real-time monitoring of the CLCW data of **all** telecommand chains shall be preferred.

**Appendix C** of Reference [2], on **Data Link Management and Monitoring**, fully covers the requirements for CLCW multiplexing on the telemetry downlink. For other status data belonging to the "front-end" subsystem, such as:

- CPDU status data,
- Authentication Unit status data (when the authentication option is selected for the mission),

constant monitoring is expected to be provided for both telecommand chains, at least during the nominal operational phases.

Section 10, on Telemetry Reporting, covers the telemetry-related functions and data structures for all layers. Section 11 covers the Telemetry Interface.

### 3.2.3 Readiness and Accessibility

The on-board telecommand subsystem shall always be in a constant state of readiness and able to provide its data-link services immediately through either of its two access channels (Virtual Channels). More specifically:

- (a) Both parts of the redundant "front-end" subsystem shall be **functional at all times**: both Virtual Channels are permanently **open** and able to accept telecommand messages from ground.

**IMPORTANT NOTE:** Special "wake-up" or "switch-on" procedures designed to bring an otherwise non-functional telecommand chain in operation are **not allowed**.

- (b) The telecommand subsystem shall be able to receive, transfer and execute vital (and simple) command instructions at all times, and more particularly during critical operational phases: launching of the

spacecraft, failure of a vital subsystem, loss of the telemetry return link (the so-called "blind telecommanding" capability requirement), etc.

- (c) In the event of a temporary spacecraft power loss, the telecommand subsystem shall automatically set itself into a pre-defined state as soon as the power returns ("cold start" initialisation state).
- (d) The telecommand subsystem shall be designed to greatly minimise sensitivity to events causing erroneous changes of state. If and when the current state of the subsystem cannot be maintained, the "cold start" initialisation state shall apply. (See also Section 3.4.5 on Implementation Requirements.)

In Reference [2], the essential purpose of the Expedited Service of the Transfer Layer is to allow both (a) and (b) requirements to be fully met. As regards the Sequence-Controlled Service of the Transfer Layer, requirement (a) is fully met, whereas requirement (b) can only be partially satisfied (e.g. it can be met as long as the telemetry return link is available or the TC Decoder is not locked into a permanent "Wait" state).

#### **3.2.4 Automated Sequence-Controlled Service**

The nominal data-link service (i.e. the Sequence-Controlled Service) shall meet the following operational requirements:

- (a) The service shall guarantee that the command data (i.e. in this case, the TC Segments) are delivered in the same sequential order in which they were generated at the sending end, possibly with some variable delay between TC Segments. No TC Segment shall contain an error, be lost or duplicated. (The data link performance probabilities required for this quality of service are covered in Section 3.3 on Data Communications Requirements.)
- (b) The transmission (and retransmission) process shall be fully automated. (The data link performance probabilities required for the practical operation of such an automated system are covered in Section 3.3 on Data Communications Requirements.)
- (c) It shall be possible to reduce to a minimum the variable delay mentioned in (a) by systematically retransmitting the complete sequence of TC Transfer Frames without waiting for any effective telemetry acknowledgement (CLCW) to reach the telecommand process on ground. Such a transmission mode is intended to help

reduce access times for the loading of command data during critical operational phases, and in particular:

- during launch or immediately after, such as during injection of a spacecraft from a low Earth orbit into a geostationary Earth orbit;
- during deep-space operations.

Section 9 of Reference [2] discusses this requirement.

### **3.2.5 Multiplexing of Data Streams**

The on-board telecommand system shall have the ability to demultiplex distinct streams of TC Packets for the following mission-specific purposes:

- implementation of separate subsystem data interfaces;
- implementation of redundant data interfaces for a given subsystem;
- implementation of operational access mechanisms (e.g. time slots) on the uplink, for bandwidth or priority management, each MAP offering effectively a "virtual-channel-like" capability;
- implementation of Packet Assembly Controller (PAC) data and control lines when packet re-assembly is required.

This is provided by the Segmentation Layer (TC Segments, with MAPs). Operability of the system relies on the availability of the Sequence-Controlled Service (automation and service guarantee).

### **3.2.6 Authentication of Command Data**

Authentication of command data shall meet the following requirements:

- (a) Authentication shall not involve the data link layers (Physical, Coding, Transfer), so as to be transparent to these layers.
- (b) Authentication shall be offered preferably as a centralised service, both on the ground and on board the spacecraft, for reasons of improved security (e.g. key management), ease of operation (e.g. maximum automation for the loading of secret keys) and implementation efficiency (e.g. standardisation).

(It is for these reasons that, in Reference [2], authentication of TC Segments is preferred to authentication of TC Packets.)

- (c) In the event of a temporary power loss, the requirements of Paragraph (c) of Section 3.2.3 (Readiness and Accessibility) are supplemented by the following:

When the spacecraft power is restored, it shall be possible to set the on-board Authentication Unit (AU) back to its full previous state, from ground. The procedure shall strictly minimise the risk of an attack, particularly that based on using previous recordings of authenticated telecommand messages. The preferred solution is to equip the AU with non-volatile memory devices that will preserve enough communication protocol data for the ground to safely reload the previous states.

- (d) When no telemetry is available (e.g. during a critical spacecraft recovery phase, with little on-board power and a "blind" telecommanding situation), it shall be possible to send a single, simple (mission-specified) authenticated command to disconnect either of the redundant AUs. Reconnection of an AU, after full recovery, shall use a similar (but unauthenticated, this time) command.

### **3.3 DATA COMMUNICATIONS REQUIREMENTS**

#### **3.3.1 Probability of Errors**

The bit error rate (BER) recommended by ESA as the worst acceptable value to operate the uplink is  $1 \times 10^{-5}$ .

The baseline performance criteria for telecommand were established by the CCSDS (Reference [2] is directly derived from a CCSDS Recommendation on Telecommand). Two criteria must be simultaneously met by the overall telecommand system. They are discussed in the next two paragraphs.

##### **(a) Probability of detected (uncorrectable) errors**

The first criterion concerns the probability of frame rejection (PSFR), which is essentially due to the detection of uncorrectable errors occurring on the uplink. For a maximum-length TC Transfer Frame (worst case), CCSDS specifies that the PSFR shall be smaller (better) than one frame for every 1000 frames. This requirement can only be met with a TC Codeblock decoder operating in the single-error correction and double-error detecting

(SEC) mode. By contrast, the triple-error detection (TED) mode (the other possible decoding mode) has a poor PSFR, although it guarantees a good probability of undetected errors (see Paragraph (b) hereafter).

Reference [2], which specifies that SEC be used, shows that the PSFR so obtained is more than two orders of magnitude better (about  $1 \times 10^{-5}$ ) at a BER of  $1 \times 10^{-5}$ .

### **(b) Probability of undetected errors**

The second of the two performance criteria concerns the probability that a frame error occurring on the uplink is not detected: this is the probability of undetected frame error (PFU).

CCSDS specifies that the PFU must be smaller than one frame for every  $10^9$  frames under all conditions where the bit error rate provided by the layer below is better than  $1 \times 10^{-5}$ . This requirement can be met comfortably in the TED mode ( $1 \times 10^{-14}$ ), but only very marginally in the SEC mode. Also, a PFU of  $1 \times 10^{-14}$  (or better) is favoured by ESA as a more realistic criterion.

The solution is to improve the poor PFU of the SEC decoder by making use of the additional error detection capability of the TC Transfer Frame (CRC in the Frame Error Control Field), with no degradation of the PSFR.

Reference [2] recommends that a BER of  $10^{-5}$  be guaranteed as the highest (worst) value on the uplink channel. For that value of BER, the actual PFU of the ESA telecommand system is better than  $1 \times 10^{-19}$ .

### **3.3.2 Maximum Uplink Symbol Rate**

The maximum uplink data rate is directly constrained by the maximum value of the uplink symbol rate. This is specified in Reference [1] to be 4 kHz. However, it is known that:

- new requirements for higher uplink symbol rates are expected to materialise within the next decade, with values in the region of 20 kHz;
- the current state of the art in microelectronics technology makes it possible to achieve symbol rates several times greater than 4 kHz with no particular difficulty.

Unless otherwise specified, the design and technology trade-offs for a given implementation shall support the maximum feasible symbol rate.

### **3.4 IMPLEMENTATION REQUIREMENTS**

This section deals with subsystem implementation requirements which could not appear (or be developed) under Sections 3.2 and 3.3.

As a general rule, design requirements specific to a given layer are treated (under the title "Design Requirements") in the section related to that layer.

#### **3.4.1 Physical Layer (Symbol Stream) Interface Capability**

The telecommand subsystem of a free-flying spacecraft will normally be connected to several radio-frequency transponders and demodulators (Physical Layer), the exact number varying with the mission. As a consequence of this, and unless otherwise specified, a minimum of 4 separate TC symbol stream input interfaces shall be provided for each TC Decoder. Whenever feasible, it is recommended that up to 6 separate input interfaces per TC Decoder be provided.

Section 5 of this document specifies an automatic mechanism for the reliable selection of an active symbol stream input. This mechanism is expected to facilitate the design of the cross-coupling systems found on free-flying spacecraft upstream of the TC Decoders.

See also Section 11 on Interfaces.

#### **3.4.2 Telemetry Interface Capability**

The capability of the telemetry interface is essentially governed by the operational requirement of Section 3.2.2 (Monitoring via Telemetry).

In this document, an additional capability is specified, which concerns the acquisition of survey data on the performance of the data link as seen from the receiving end. These survey data are organised in a 32-bit data structure called "Frame Analysis Report" and specified in Section 10.

The capability to telemeter the "Frame Analysis Report" (FAR) shall always be provided by the TC Decoder. However, the decision to downlink the FAR data is an operational requirement which is left to the particular mission authority.

See also Appendix C of Reference [2], and Section 11 of this document (Interfaces).

### 3.4.3 MAP Interface Capability

The capability of the MAP interface is fundamentally governed by the characteristics of the Segmentation Layer, and in particular by:

- the basic 64-data-output capacity of the MAP Identifier;
- the possibility to associate MAP connections by pairs when packet re-assembly is required, with one pair of MAPs for each implemented PAC (one data MAP associated to one control MAP).

However, the final configuration and implementation characteristics of the MAP interface required on board each spacecraft is mission-specific.

The TC Decoder specified in this document shall provide the basic interface (inputs and outputs) from which any number of single-MAP and/or multi-MAP interfaces could be implemented to satisfy the (mission-specific) operational requirement of Section 3.2.5 (Multiplexing of Data Streams).

For the detailed specification of the basic MAP interface mechanisms and waveforms, see Section 11 on Interfaces.

### 3.4.4 Mission-Specific Data Programming Capability

Throughout this document, the term "Mission-Specific Data" applies to data that are specifically assigned to each of the redundant TC Decoders of a particular spacecraft. For practical reasons, it is required to be able to program or re-program on ground (e.g. some time shortly before launch, as could be the case with the Fixed Key of the authentication system) the "Mission-Specific Data" of each TC Decoder.

More specifically, it is recommended that the flight units of the TC Decoder subsystem be easy to re-program once they are mounted on the spacecraft. Such re-programming shall always be possible after the units have been demounted from the spacecraft. A conventional solution consists of providing removable, easy-to-access plug-ins (i.e. wired-up connectors, ROMs etc.) that can be tested and qualified in advance.

Typical "Mission-Specific Data" are:

- Spacecraft ID (10 bits)
- VC ID (6 bits)



- FARM Sliding Window width (PW and NW) (2 x 8 bits)
- Authenticated MAP ID Pointer (5 bits)
- Fixed Key of the Authentication Unit (2940 bits)
- Application Identifier (11 bits) of the CPDU

### 3.4.5 Immunity to Changes of State

The operational requirements of Paragraph (d) of Section 3.2.3 (Readiness and Accessibility) specify that the TC Decoder "shall be designed to greatly minimise sensitivity to events causing erroneous changes of state."

Therefore, erroneous changes of state shall be detected, and appropriate action shall be taken to return to an operationally valid state. In the worst case, this state shall be the "cold start" initialisation state, as already specified in Paragraph (d) of Section 3.2.3.

The design must ensure that the incidental corruption of some status data will never set the TC Decoder into a permanent, irrecoverable state that would be fatal to the system. An example of an **incorrect** design would be that in which the Spacecraft ID is read-out from a PROM at switch-on and stored in a register prior to the validation of all subsequent TC Transfer Frames. If corruption of the data stored in the register is allowed to take place without being detected and corrected, the TC Decoder will reject all incoming frames thereof, with no foreseeable chances of recovery.

### 3.4.6 Physical Distribution of Major Functions

This Section is complemented by **Appendix A: Telecommand Subsystem Configuration on board an ESA Spacecraft**.

The physical distribution of the major functions making up a TC Decoder subsystem is left to the discretion of the implementer. For instance, it is quite valid to implement the complete TC Decoder system (that is, the "basic" TC Decoder – as shown in Figure A-1 of Appendix A – complete with the Authentication Unit and the CPDU) as a single chip.

However, in all cases, the "basic" TC Decoder chip shall provide the interface necessary to connect an external Authentication Unit circuit. The reason for this is that future security requirements may foster the development of other authentication algorithms. A new Authentication Unit could be designed that would retain the same interface (as justified by Reference [2], which does not specify the authentication algorithm).

### 3.4.7 Cross-Coupling of Major Functions with MAP Interfaces

This Section is complemented by **Appendix A: Telecommand System Configuration on board an ESA Spacecraft**.

Figure A-1 of Appendix A shows the TC Decoder subsystem as a typical set of major functions (where the Authentication Unit is not visible).

The telecommand subsystem is fundamentally redundant (operational requirement of Section 3.2.1), with all units always functional (operational requirement of Section 3.2.3). The spacecraft Data Management Subsystem (DMS), which is concerned with telemetry/command data handling and distribution, is also (usually) foreseen as a redundant subsystem. However, in such a case, one of the two DMS subsystems is usually not functional. If and when the functional DMS part fails, one can only rely on the TC Decoder subsystem for switching 'off' the failed DMS part and switching 'on' the other DMS part (this being done through one of the two TC-Decoder-dedicated CPDUs).

The MAP interface connections between each TC Decoder and each DMS unit are doubled, effectively "cross-coupling" each of these four functions. Such cross-coupling can be both **advantageous** and **practical** because one of the two DMS functions is **unpowered**. This fact makes the design of the necessary MAP interfaces **less critical** from a failure-mode standpoint.

This is **not the case** for the cross-coupling of TC Decoders and CPDUs. In Figure A-1, such cross-coupling connections are not shown. However, it can be required that cross-coupling of TC Decoders and CPDUs be implemented on board a given spacecraft, which implies a certain amount of additional risk. The risk comes from the fact that the four functions (two "basic" TC Decoders and two CPDUs) are **always powered**, with the consequence that one chain can effectively be "corrupted" by the failure of the other one if the interface used for the cross-coupling connections has not been specifically designed for that purpose.

One possible technique for the design of a fail-safe MAP interface that should meet such a requirement is to enable some critical MAP signal(s) at some particular time by means of "time-out windows".

In the end, the choice of the design concept is left to the designer, who will have to achieve his goal while keeping the complexity of the interface sufficiently low to ensure that its intrinsic reliability will not be reduced. Unless the contrary is proven by appropriate study of the detailed behaviour of the

cross-coupling circuits, ESA recommends that "basic" TC Decoders and CPDUs should **not** be cross-coupled.

### 3.4.8 Technology and Electrical Characteristics

This Specification is a functional design specification. Therefore, the problems associated with the technology selected for the implementation of the TC Decoder functions are not within its scope. Specific requirements concerning the technology and its electrical characteristics may have to be supplemented for the relevant development contract.

However, the following guidelines should be considered in the light of such development contracts:

- (a) The current state-of-the-art in microcircuitry makes ASICs the preferred candidates for implementing the TC Decoder functions.
- (b) CMOS technology appears to be the current universal choice for the design of such ASICs.
- (c) Therefore, electrical characteristics are expected to be those of the CMOS technology. More particularly, this fact should dominate the design of the various interfaces.
- (d) In this document, the specification of interfaces is limited to their **functional** aspect, i.e. to a description of the various signal waveforms, with their logical states and essential timing. Extrapolation of this information for the realisation of ASICs should not create any particular difficulty, the final subsystem interface being outside of the scope of this Specification.
- (e) Very low sensitivity of the selected technology to harmful radiation shall always be considered in view of the critical importance of the TC Decoder subsystem for the survivability of the spacecraft.

**PAGE INTENTIONALLY LEFT BLANK**

## 4. PHYSICAL LAYER

The Physical Layer is the subject of Reference [1] and provides the information required for a broader view of the overall telecommand system. Otherwise, the Physical Layer is of no relevance to the design of the TC Decoder, except for its interface with the Coding Layer.

In Reference [2], the Physical Layer is discussed in Section 4. The subsections of particular interest to this specification are:

- Subsection 4.3.3.2 for what concerns the interface between Physical Layer and Coding Layer at the receiving end;
- Subsection 4.3.4.2 for what concerns the actions at the receiving end.

**PAGE INTENTIONALLY LEFT BLANK**

## 5. CODING LAYER

### 5.1 SPECIFICATION

#### 5.1.1 TC Codeblock Length

Section 5 of Reference [2] is fully dedicated to the Coding Layer.

In Reference [2], four standard lengths are specified for the TC Codeblock, with one recommended Codeblock length (8 octets). The essential reason for allowing a choice of four Codeblock lengths is related to the design of TC Decoders, for which it may be found that a particular design is easier to optimise with one Codeblock length (say, 7 octets) rather than with the recommended one (8 octets).

- **In this Specification, unless otherwise agreed by ESA for a particular implementation (after proper justification), the length of the TC Codeblock shall be 8 octets.**

#### 5.1.2 Additional Information on the Coding Layer Procedures

In the current issue (Issue 2) of Reference [2], some information on the Coding Layer Procedures is missing. (It is intended to insert all such additional information in Reference [2] when it is next reviewed.) The information concerns essentially:

- the DECODE State of the CLTU decoder: this is given in Section 5.1.3 hereafter;
- the BCH code polynomial, from a decoding performance standpoint: this is given in Section 5.1.4;
- the TC Codeblock decoding procedure, with a model of the BCH decoder: this is given in Section 5.1.5.

### 5.1.3 First and Last Data Transfer in the DECODE State

When in the DECODE state (S3), each candidate Codeblock is decoded in the SEC<sup>(\*)</sup> mode:

- (a) When the first "Candidate Codeblock" (i.e. "Candidate Codeblock" 0, which follows Event 3 (E3): START SEQUENCE FOUND) is found to be error free, or if it contained an error which has been corrected, its information octets (i.e. 7 octets) are transferred to the layer above. At the same time, a "Candidate Frame Arrived Indication" signal shall be set TRUE, indicating the beginning of a transfer of a block of octets that will make up a "Candidate Frame".

There are now two cases:

#### CASE 1

When an Event 4 – (E4): CODEBLOCK REJECTION – occurs for any of the 37 possible "Candidate Codeblocks" that can follow Codeblock 0 (i.e. from "Candidate Codeblock" 1 to "Candidate Codeblock" 37, this last candidate being – most probably – the Tail Sequence), the decoder returns to the SEARCH state (S2), with the following actions:

- the Codeblock is abandoned (erased);
- no information octets from that Codeblock are transferred to the layer above;
- the "Candidate Frame Arrived Indication" signal is set FALSE, indicating to the layer above the end of the transfer of a block of octets that makes up a "Candidate Frame".

#### CASE 2

When an Event 2 – (E2): CHANNEL DEACTIVATION – occurs which affects any of the 37 possible "Candidate Codeblocks" that can follow Codeblock 0, the decoder returns to the INACTIVE state (S1), with the following actions:

---

NOTE(\*) The acronym "SEC", as used in Reference [2] and in this Specification, is short for "SEC & DED", which is the full-length acronym for "Single-Error Correction and Double-Error Detection".



- the actions specified for CASE 1 also apply here.
- (b) When an Event 4 (E4), or an Event E2 (E2), occurs which affects the first "Candidate Codeblock" ("Candidate Codeblock" 0), the CLTU shall be ABANDONED. No "Candidate Frame" octets have been transferred to the layer above.
- (c) If and when more than 37 Codeblocks have been accepted in one CLTU, the decoder returns to the SEARCH state (S2). The CLTU is effectively aborted (this being required to avoid any sort of "lock-up" situation, whether accidental or not); it shall be reported as ABANDONED and a signal sent to the Transfer Layer indicating that the entire block of octets making up the "Candidate Frame" must be erased.

#### 5.1.4 Polynomial Forms of the BCH Code

The BCH code specified in Reference [2] is a Hamming code. Coding experts use the term "Hamming" for codes that correct single errors, and reserve the term "BCH" for codes that correct more than one error. Therefore, a Hamming code is a BCH code that corrects single errors. It is also a "perfect code". Furthermore:

- The BCH code of Reference [2] is a "modified" Hamming code. To be more specific: it is an "expurgated" Hamming code.
- **n** is the length of the code, which can take one of four values: 63, 55, 47, 39. (The specified, "preferred" value is 63, which is that used for the discussion below. However, the performance of the code remains the same for any of the four lengths.)
- **k** is the number of information bits.
- **m = n – k** is the number of parity check bits.
- **d** is the hamming distance.
- The basic (63,57) Hamming code has a generator polynomial:

$$g_1(x) = x^6 + x + 1$$

- We have:  $n = 63$ ,  $k = 57$ ,  $m = 6$ ,  $d = 3$  (single-error correcting, or double-error detecting, but **not both**).
- The above Hamming code has been first "shortened" to (62,56), and then "extended" to (63,56). These two operations are equivalent to "expurgating" the original Hamming code.
- The expurgated Hamming code has a generator polynomial:

$$g(x) = g_1(x) (x - 1)$$

where an information bit has been replaced by a check bit. Hence the factor  $(x - 1)$ .

- We have:  $n = 63$ ,  $k = 56$ ,  $m = 7$ ,  $d = 4$ . This signifies that the code is either a triple-error detecting (TED) or a single-error correcting **and** double-error detecting (SEC) code, but **not both**.
- If one performs the above multiplication in modulo 2 arithmetic, the result is:

$$g(x) = (x^6 + x + 1) (x + 1) = x^7 + x^6 + x^2 + 1$$

which is the form used to express the generator polynomial of the BCH code in Reference [2].

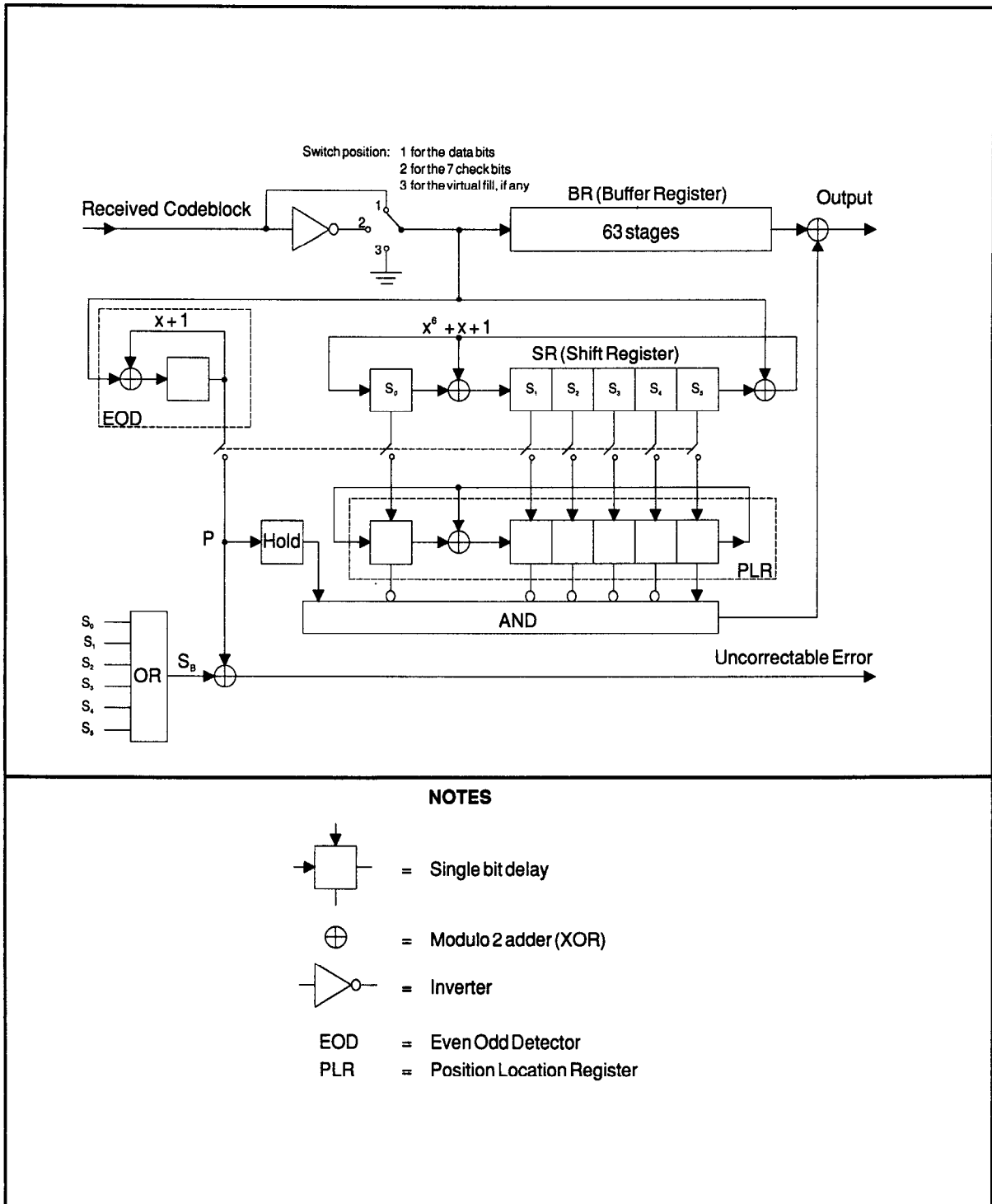
- Finally, a dummy bit (Filler Bit) is added to make the codeblock structure a multiple of eight (e.g.  $63 + 1 = 64$ ).

The non-factored form of  $g(x)$  is used to design a TED decoder.

For SEC (i.e. single-error correction **and** double-error detection), the implementation of the Modified Hamming Decoder – described in Section 5.1.5 and shown in Figure 5.1 – is best explained by using the factored form of  $g(x)$ . The  $(x - 1)$  factor is the one-stage parity-bit decoder called Even-Odd Detector (EOD) on the diagram, and  $g_1(x)$  is the 6-stage Hamming decoder.

### 5.1.5 Possible Realisation of a (63,56) Modified Hamming Decoder

Figure 5.1 shows a possible arrangement for decoding the modified (63,56) BCH code with the aid of shift registers.



**NOTES**

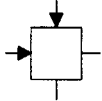

-  = Single bit delay
- $\oplus$  = Modulo 2 adder (XOR)
-  = Inverter
- EOD = Even Odd Detector
- PLR = Position Location Register

Figure 5.1 (63,56) MODIFIED HAMMING DECODER

The implementer is not required to realise a Hamming Decoder that strictly complies with the arrangement of Figure 5.1, but is free to optimise his design by selecting any other suitable arrangement.

However, it is required that functional compliance between the implementer's design and the Hamming Decoder of Figure 5.1 be proven before the design is frozen.

The following information will help the reader to understand the operation of the decoder shown in Figure 5.1:

- The EOD and the shift register SR are set to all zeros before receiving each TC Codeblock.
- The Filler Bit of the TC Codeblock is removed by simply ignoring it: all registers are stopped during the last bit period (i.e. during the 64th bit for a full-length TC Codeblock).
- The contents of the upper shift register SR are transferred to the bottom shift register PLR (Position Location Register) after the  $n$ th shift by momentarily closing the switches ( $n = 64 - j$ , with  $j = 1, 9, 17$  or  $25$ ; for a 64-bit TC Codeblock,  $n = 63$ ).
- The contents of registers BR and PLR are shifted at the same time, so that a **1** at the output of the AND gate may correct (by means of the XOR gate) any single-bit error at the output of register BR. This correction occurs when the contents of the PLR register are **000001** and the HOLD output is **1**.
- The decoding strategy is summarised in Table 5.1, where the binary value SB (for S Binary) of the error syndrome is either **1** or **0**, when the value of the error syndrome  $S$  ( $S = S_0, S_1, S_2, S_3, S_4, S_5$ ) is either **0** or  $> 0$ .

## 5.2 DESIGN REQUIREMENTS

Subsection 3.4.1 specifies that the TC Decoder subsystem must feature a minimum of 4 Physical Layer (symbol stream) input interfaces per "basic" TC Decoder, with a recommended number of 6 input interfaces. The symbol streams are provided by the spacecraft radio-frequency and modulation subsystem – typically by telecommand subcarrier demodulators. Each

EOD OUTPUT (P)	BINARY SYNDROME VALUE (SB)	SYNDROME VALUE (S)	FILLER BIT VALUE	DECISION
0	0	0	IGNORED	<p>NO ERRORS</p> <p>The output of the AND gate is 0: NO CORRECTION. The output of the XOR gate is 0: the Codeblock is accepted (NO EVENT E4).</p>
0	1	>0	IGNORED	<p>DETECTION OF EVEN ERRORS</p> <p>The output of the AND gate is 0: NO CORRECTION. The output of the XOR gate is 1: the Codeblock is rejected (EVENT E4).</p>
1	0	0	IGNORED	<p>DETECTION OF ODD ERRORS IN AN APPARENTLY CORRECT CODEBLOCK</p> <p>The output of the AND gate is 0: NO CORRECTION. The output of the XOR gate is 1: the Codeblock is rejected (EVENT E4).</p>
1	1	>0	0	<p>DETECTION OF ODD ERRORS IN AN INCORRECT CODEBLOCK</p> <p>The output of the AND gate is 1 at position where the contents of PLR are 000001: CORRECTION OF SINGLE ERROR. The output of the XOR gate is 0: the Codeblock is accepted (NO EVENT E4).</p>
1	1	>0	1	<p>DETECTION OF ODD ERRORS IN AN INCORRECT CODEBLOCK</p> <p>This is the same as above, but, since the Filler Bit value is 1, the Codeblock must be rejected (EVENT E4).</p>

Table 5.1 DECODING STRATEGY

symbol stream interface (this interface is also called "TC channel" or "input channel" interface) consists of three input lines:

- Symbol Stream signal (NRZ-L)
- Symbol Clock signal
- "Channel Active Indication" signal

When the Channel Active Indication signal of one input interface becomes TRUE (Event E1), the CLTU decoder goes from the INACTIVE state (S1) to the SEARCH state (S2). Sustained arrival of CLTUs on the Symbol Stream line will keep the decoder in the SEARCH state until Event E3 (START SEQUENCE FOUND) occurs, sending the decoder into the DECODE state (S3). The system will remain in the DECODE state until – typically – it fails to decode the Tail Sequence (Event E4, CODEBLOCK REJECTION), thus going back to the SEARCH state (S2), and so on.

Since the symbol stream interface is provided with a symbol clock signal, it is expected that the implementer will use this clock signal not only to acquire each TC Codeblock, but also to drive the circuitry of the Modified Hamming Decoder. In such a case, it is accepted that an Event E2 (CHANNEL DEACTIVATION), when it occurs at a time when a TC Codeblock is clocked in, also affects the Codeblock that was acquired immediately before (because the clock signal disappeared and the Hamming Decoder stopped).

It is normally intended to operate spacecraft with only one input channel ACTIVE at any one time. However, the TC Decoder must also be able to successfully select one input when it is in the SEARCH state (S2) and several inputs are ACTIVE simultaneously.

Therefore, when in the SEARCH state, the TC Decoder shall search for a CLTU Start Sequence on **all** ACTIVE inputs simultaneously, symbol by symbol, until one Start Sequence is found in one of the symbol streams, and the corresponding input selected for the rest of that particular CLTU decoding process. This selection shall be performed with no particular, pre-determined preference.

When Event E4 occurs (CODEBLOCK REJECTION, most of the time caused by the Tail Sequence), the decoder returns to the SEARCH state and the search for a Start Sequence on **all** ACTIVE inputs shall be resumed, and so on.

The TC Decoder shall also be able to recover automatically from the following faulty situation:

- it is in State S3 (DECODE), and
- the Channel Active Indication signal provided by the selected input interface is still TRUE (i.e. indicating that the channel is ACTIVE), whether by failure or not, and
- the Symbol Clock signal provided by the selected input interface has disappeared, whether by failure or not.

A timer is suggested as a possible solution for the required fail-safe system, with a time-out period valid for the full range of uplink bit rate (a value of about one second is suggested for that period).

**PAGE INTENTIONALLY LEFT BLANK**



## 6. TRANSFER LAYER

### 6.1 SPECIFICATION

Section 6 of Reference [2] fully specifies the functions of the Transfer Layer.

Concerning Subsection 6.3.3.2 of Reference [2], Paragraph (x): Buffer Administration Variables; the following clarification is added:

- In the FARM-1 State Table (Table 6.3), Event E10 occurs every time the "back-end" buffer is released, i.e. every time it is emptied of the data it contains, whether this was caused by the normal readout of segment data by the Higher Layer, or by a segment data erasure preceding the arrival of BD segment data.

In Subsection 6.3.3.4 of Reference [2], the service primitive specified for "From FARM-1 to Segmentation Layer" as the "TC Segment" is deemed to implicitly include information on the number of octets that were transferred and that make up that particular TC Segment.

Also in Reference [2], the Lockout State (S3) of the FARM-1 State Table is characterised by the state of the Lockout Flag (Main Feature of State: Lockout Flag is on [ = 1]); the states of the two other flags (Retransmit Flag and Wait Flag) may vary. When the TC Decoder is brought under power, whether deliberately or after a temporary loss of power, the FARM-1 shall be set to a "cold start" initialisation state that will be the Lockout State (S3), with:

- Retransmit Flag = 0
- Wait Flag = 0
- FARM-B Counter = 0
- V(R) = 0

### 6.2 DESIGN REQUIREMENTS

The following design requirements concern "Mission-Specific Data" particular to the Transfer Layer, as already introduced in Subsection 3.4.4 on "Mission-Specific Data Programming Capability". They are repeated herebelow for the sake of completeness.

### **6.2.1 Spacecraft Identifier**

It shall be possible to easily program (or re-program) the 10-bit Spacecraft Identifier of each "basic" TC Decoder, on ground, after it has been delivered as a flight unit.

### **6.2.2 Virtual Channel Identifier**

It shall be possible to easily program (or re-program) the 6-bit Virtual Channel Identifier of each "basic" TC Decoder, as for the Spacecraft Identifier.

### **6.2.3 FARM Sliding Window Width**

It shall be possible to easily program (or re-program) the width of the FARM Sliding Window of each "basic" TC Decoder, as for the Spacecraft Identifier. More specifically: it shall be possible to program any 8-bit value for "PW" and "NW" (one value for each), as specified in Reference [2].

## 7. SEGMENTATION LAYER

### 7.1 SPECIFICATION

Section 7 of Reference [2] fully specifies the Segmentation Layer. However, only some of the Segmentation Layer functions reside in the "basic" TC Decoder. These functions are:

- The back-end buffer for the accepted TC Segment. The back-end buffer is "shared" between the Transfer Layer and the Segmentation Layer. For this reason, it is already discussed in Section 6 of Reference [2], and in particular in Item (x) of Paragraph (b) of Subsection 6.3.3.2 on COP-1 Variables, as well as in Subsection 6.3.3.4 on COP-1 Sublayer Interface at the Receiving End.
- The MAP interface. The specification of the MAP interface proper (lines, signals and waveforms) is the subject of Section 11. MAP interface design requirements are specified in the next section.

### 7.2 DESIGN REQUIREMENTS

#### 7.2.1 MAP Interface Implementation Capability

The MAP interface circuitry, as provided by the "basic" TC Decoder, shall allow a particular flight unit to be equipped with a variable number of MAP interfaces, as already introduced in Subsection 3.4.3.

The effective limit is set by the MAP Identifier, with a maximum of 64 MAPs. In view of the fact that MAP 63 is specifically assigned to the control of the Authentication Unit (Control Commands of the AU; see Section 8 for more details), it shall be possible to implement one separate interface for:

- (a) each of the 64 MAP Identifiers, as required, when no AU is used;
- (b) each of the first 63 MAPs (from MAP 0 to MAP 62), as required, when the AU is used .

(However, in the first case (a), the designer may choose to restrict the use of MAP 63 to the exclusive operation of the AU circuitry if this can help simplify his implementation.)

In such an instance, each MAP interface will only respond to one particular MAP Identifier. It shall also be possible to implement separate interfaces for pre-defined groups of MAP Identifiers.

(This last requirement covers, in particular, the possibility to associate MAP connections in pairs – as defined in Reference [2] – when packet re-assembly is required.)

In Figure A-1 of Appendix A, each "basic" TC Decoder features two pre-defined, separate MAP interfaces:

- one separate interface for both MAP 1 and MAP 2;
- one separate interface for both MAP 3 and MAP 4.

The interfaces shown in Figure A-1 are an attempt to illustrate the demultiplexing of two interleaved streams of (unsegmented) TC Packets. Requirements for such multiplexing are mission-specific, and likely to be of an operational nature.

Note that, in Figure A-1, the CPDU is serviced by an interface responding to a single MAP Identifier (MAP 0): there is no cross-coupling between "basic" TC Decoders and CPDUs. Design requirements for cross-coupling major spacecraft data system functions by means of MAP interfaces have already been specified in Subsection 3.4.7.

As a general rule, and in order to assist testing on ground, subsystems incorporating the TC Decoder shall provide an interface that allows monitoring of all TC Segments, whatever the value of the MAP Identifier.

### **7.2.2 Selection of Authenticated MAPs (Authenticated MAP ID Pointer)**

It shall be possible to select those MAPs that are deemed to carry authenticated TC Segments. This selection must take into account the possibility to associate MAP IDs in pairs when packet re-assembly is required (i.e. when segmentation of TC Packets is required).

Therefore, authenticated MAPs shall be selected by pairs. To do this, the implementer shall use the five LSBs of the MAP Identifier field (i.e. Bits 3 through 7 of the TC Segment Header). The selection mechanism shall be such that it will "point" at the last pair of MAP Identifiers (counting upwards from MAP 0) that carries authenticated TC Segments. The value identifying

this particular pair of identifiers is called the "Authenticated MAP ID Pointer". For example:

- Selecting ("pointing to") MAP 0 means that the first pair of MAPs (i.e. MAPs 0 and 32) is expected to carry authenticated TC Segments.
- Selecting MAP 4 means that the first 4 pairs of MAPs (i.e. MAPs 0 and 32, 1 and 33, 2 and 34, 3 and 35) are expected to carry authenticated TC Segments.
- Selecting MAP 31 means that the 64 MAPs are expected to carry authenticated TC Segments. (Note that, as specified in Section 8 on the Authentication Layer, the 64th MAP (MAP 63) is always used for the exclusive control of the Authentication Unit – that is, when the Authentication option is used by the particular mission – and that, consequently, MAP 31 can only be used for data streams that do not require packet re-assembly.)

Only those TC Segments that are routed via the MAPs selected in this way will be processed by the Authentication Unit, as specified in Section 8.

Note that when authentication is not used at all, the Authentication Unit will simply be disabled (See Subsection 8.6 on Design Requirements for the Authentication Layer).

The MAP Identifier Pointer belongs to the "Mission-Programmable Data" already introduced in Subsection 3.4.4. It shall be possible to easily program (or re-program) the selected value of the 5-bit MAP Identifier Pointer for each "basic" TC Decoder.

**PAGE INTENTIONALLY LEFT BLANK**

## 8. AUTHENTICATION LAYER

### 8.1 GENERAL SPECIFICATION

Section 10 of Reference [2] introduces the concept of encrypted authentication of TC Segments and specifies:

- the general layout of the onboard authentication part of the TC Decoder, i.e. the Authentication Unit (AU), with its major functions and interfaces;
- the major elements of the authentication protocol, i.e. the Authentication Tail of the authenticated TC Segment, with its Logical Authentication Channels (LAC Identifiers and LAC Counters) and the Authentication Signature.

The detailed specification of the full authentication system is provided in the following sections of this chapter, i.e.:

- the Authentication Processor;
- the Supervisor;
- the various formats of authenticated TC Segments;
- the operational procedures for loading the secret keys and for recovery;
- the design requirements.

Figure 10.1 of Reference [2] only provides a conceptual illustration of the AU layout. Figure 8.1 of this section shows a more functional version of the same layout.

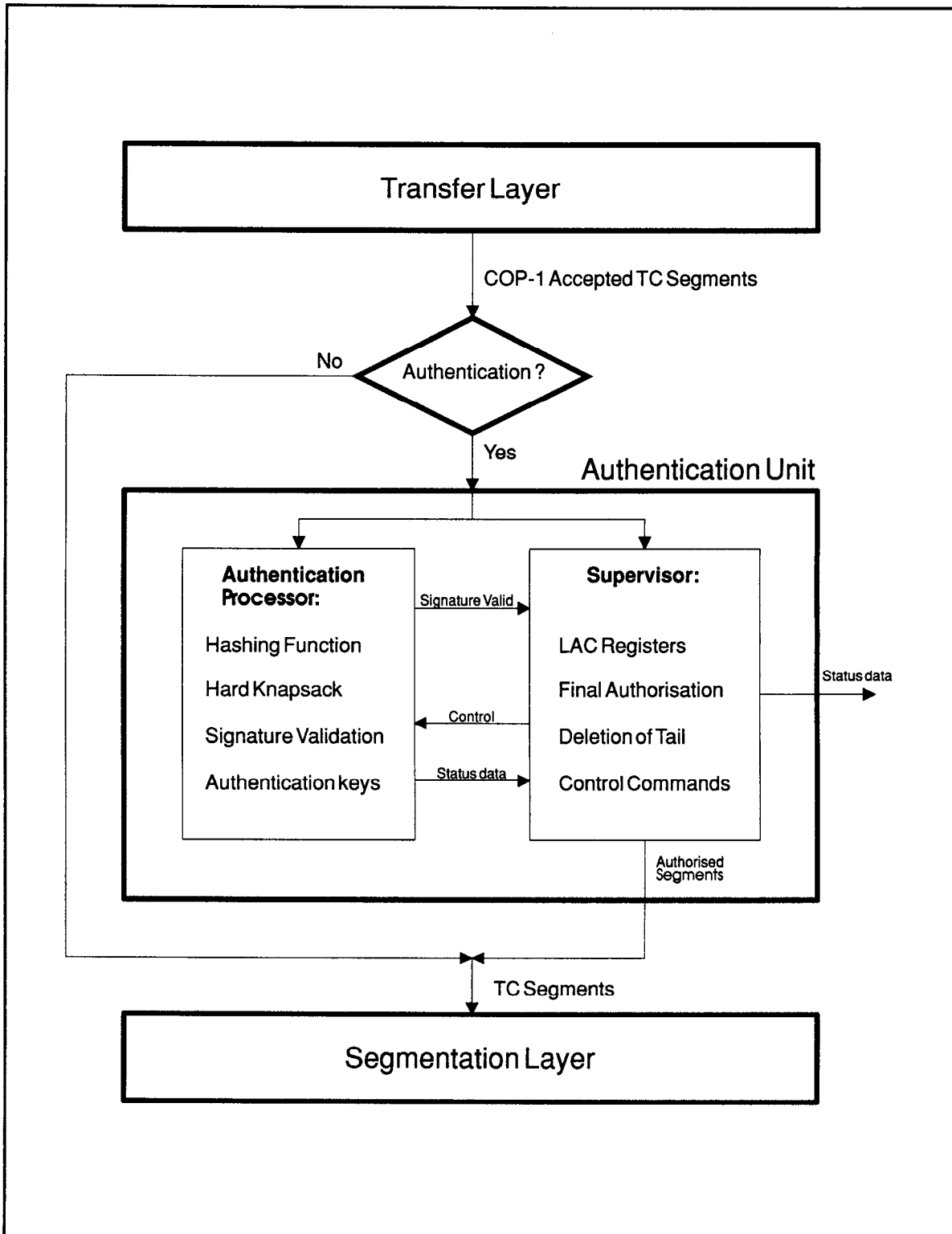


Figure 8.1 FUNCTIONAL LAYOUT OF THE AUTHENTICATION UNIT



## **8.2 THE AUTHENTICATION PROCESSOR**

### **8.2.1 Functional Concept**

The authentication method selected by ESA belongs to the Explicit Signature Systems (plain text message with appended signature). It is based on a provably complex one-way function called the "hard knapsack". The method consists of generating a 40-bit digital signature using a transformation under a secret key applied to the message (in the present case: the TC Segment). This "authentication signature" guarantees to the recipient that the TC Segment is authentic with respect to its sender and its contents.

The receiving end authenticates an incoming TC Segment by performing the same transformation made by the transmitting end, and by comparing the received signature with the onboard-generated one. Figure 8.2 shows a functional diagram of the Authentication Processor. There are four main parts:

- the Hashing Function;
- the Hard Knapsack;
- the Deletion Box;
- the Signature Comparator.

They are described in the next four subsections. Not apparent on the functional diagram of Figure 8.2 is the organisation of the secret Authentication Keys stored in the Authentication Processor. This is described in Subsection 8.2.6.

At the transmitting end, the system is identical to that of Figure 8.2, except for the Signature Comparator which only exists at the receiving end.

### **8.2.2 The Hashing Function**

One immediately apparent purpose of the Hashing Function, as shown in Figure 8.2, is to compress the variable amount of data bits constituted by the "extended" message  $x$  into a pre-signature  $P$  of fixed length (60 bits). But the most important purpose of the Hashing Function is to keep the pre-signature  $P$  secret, so that there is as much uncertainty about the input to the Hard Knapsack as possible.

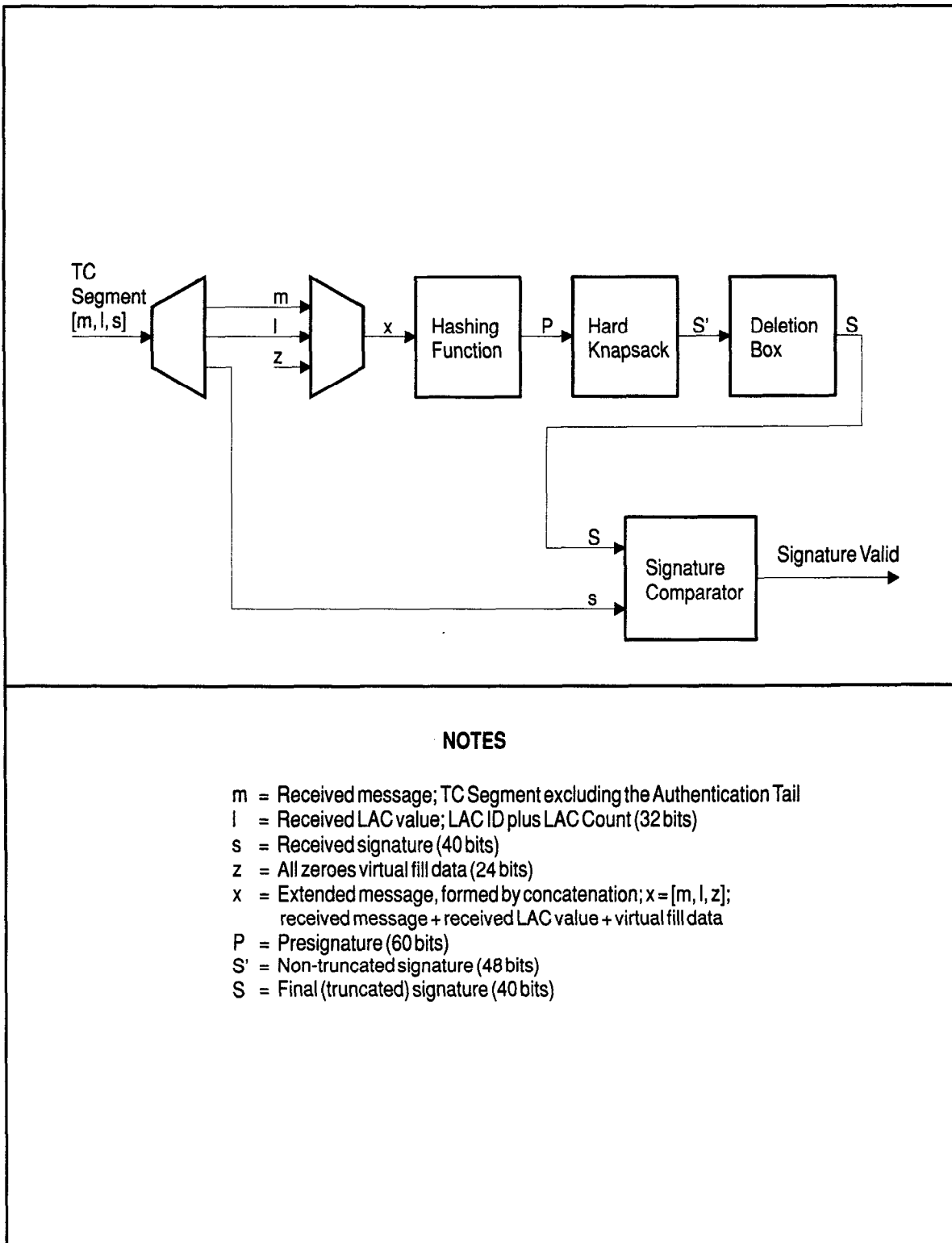


Figure 8.2 FUNCTIONAL DIAGRAM OF THE AUTHENTICATION PROCESSOR

Consequently, the Hashing Function cannot be public, but must be a secret transformation. Therefore, some part of the secret Authentication Key must reside in the Hashing Function.

The device realising the specified Hashing Function is a 60-bit linear feedback shift register (LFSR), as shown in Figure 8.3. The 60 feedback coefficients  $C_0, C_1, \dots, C_{59}$  are secret and part of the Authentication Key.

The LFSR must be initialised to the 60-bit value  $P' = 1000\dots000$  (where Bit  $P'_0 = 1$ ) before the process of each authenticated TC Segment begins.  $P$  will be the value in the LFSR after the last bit of the variable-length extended message  $x$  has been shifted in.

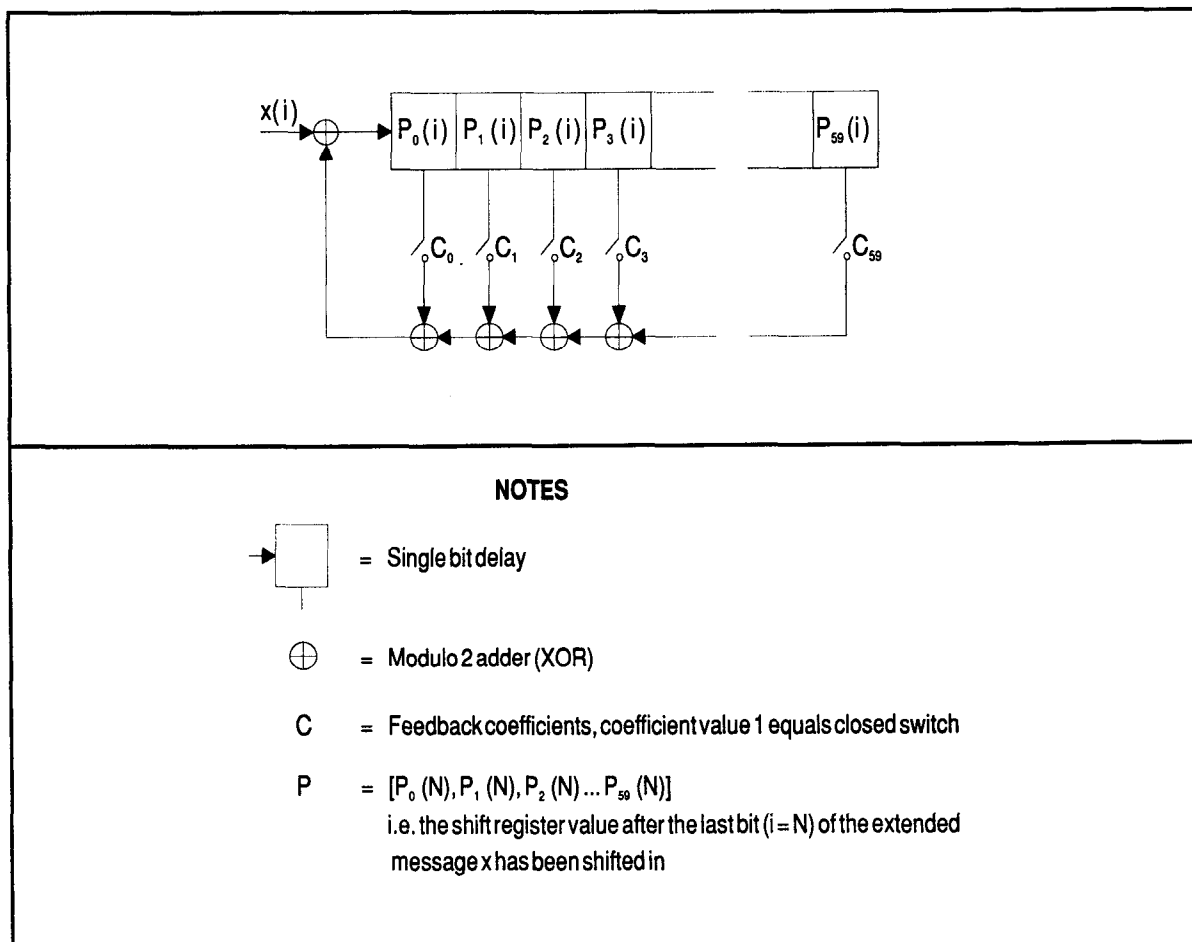


Figure 8.3 POSSIBLE REALISATION OF THE HASHING FUNCTION

The extended message  $\mathbf{x}$  ( $\mathbf{x} = [\mathbf{m}, \mathbf{l}, \mathbf{z}]$ ) consists of the following data elements, placed one after the other in that order:

- the received message  $\mathbf{m}$ , i.e. the TC Segment (variable from 1 to 240 octets) without the Authentication Tail;
- the received LAC value  $\mathbf{l}$ , i.e. 4 octets (2 bits of LAC ID, plus 30 bits of LAC Count);
- three octets of virtual fill  $\mathbf{z}$ , consisting of 24 zeros.

The purpose of the 24 bits of virtual fill is to ensure, at the sending end of the process as well as in the TC Decoder, that the Hashing Function is provided with a minimum of data bits. The required minimum is 60 bits. Since the smallest length allowed for a TC Segment is 1 octet, adding only 4 octets of LAC value  $\mathbf{l}$  would give a minimum of 5 octets (40 bits) for  $\mathbf{x}$ . The systematic addition of 24 bits of  $\mathbf{z}$  (all zeros) to  $[\mathbf{m}, \mathbf{l}]$  guarantees that  $\mathbf{x}$  will never be smaller than 64 bits.

The 24 bits of virtual fill  $\mathbf{z}$  are not transmitted by the sending end. Therefore, they must be re-generated at the receiving end.

Note that since  $\mathbf{m}$  (i.e. the TC Segment) cannot be equal to zero, the total length of an authenticated TC Segment (i.e.  $[\mathbf{m}, \mathbf{l}, \mathbf{s}]$ ) cannot be smaller than 10 octets. Anything smaller than 10 octets shall be rejected and reported accordingly (see Section 10.5, on Frame Analysis Report).

### 8.2.3 The Hard Knapsack

The purpose of the Hard Knapsack, as shown in Figure 8.2, is to make the overall system nonlinear, so that there is no easy way to replace the signature system by some equivalent transformation, and to serve as a true one-way function, so that it is not possible to deduce (by computation) the presignature  $\mathbf{P}$  from the value of signature  $\mathbf{S}$ .

The Hard Knapsack hides the 60-bit presignature  $\mathbf{P}$  from the output world (signature  $\mathbf{S}$ ) in the same way the Hashing Function hides it from the input world (extended message  $\mathbf{x}$ ).

The incompatible number systems on which the Hashing Function and the Hard Knapsack are based ensure that the secret presignature  $\mathbf{P}$  is inaccessible to anyone trying to break the system.

The Hard Knapsack is based on the concept of the modular knapsack shown in Figure 8.4.

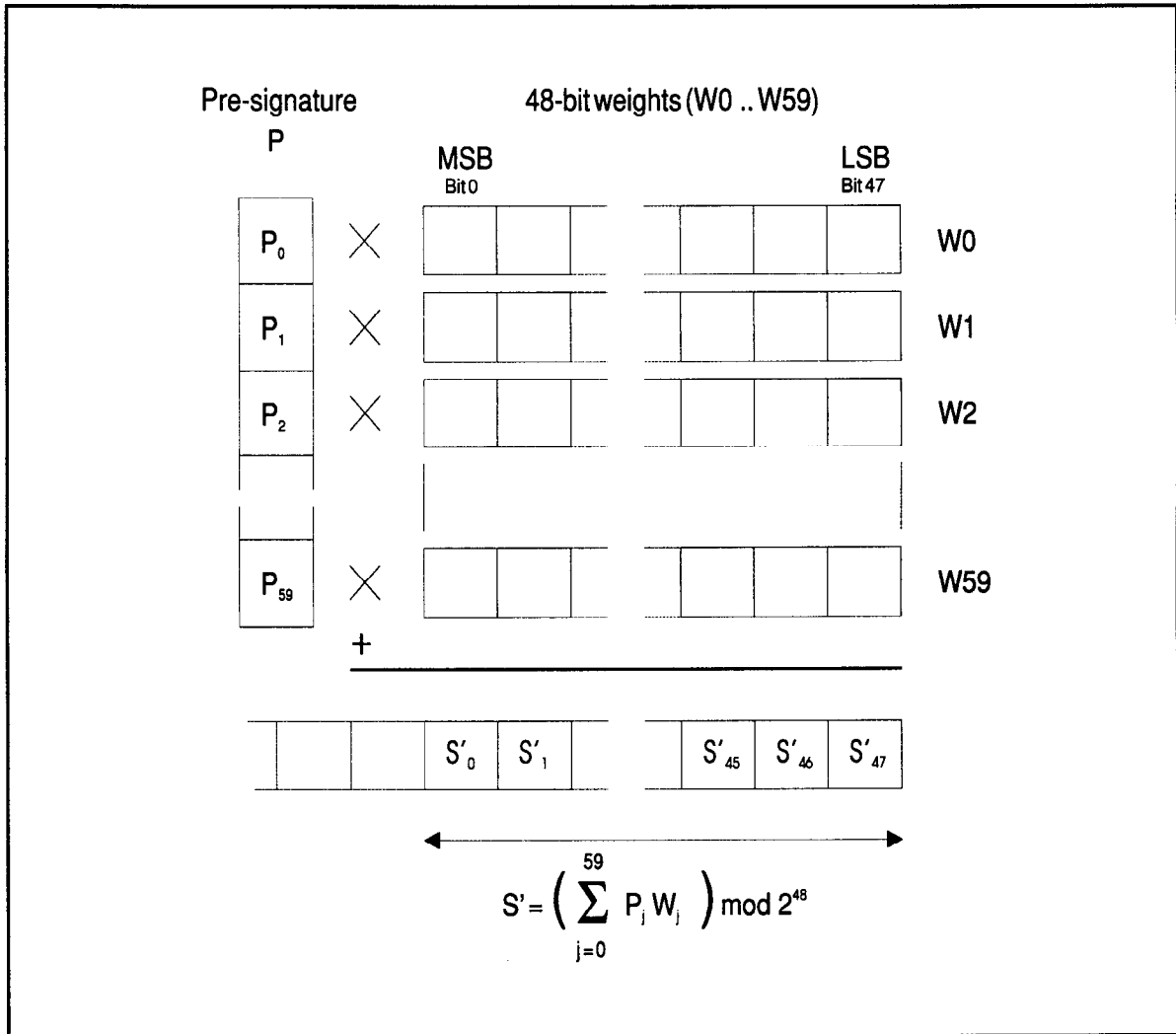


Figure 8.4 CONCEPTUAL DIAGRAM OF THE KNAPSACK

It consists of 60 weights (numbered from  $W_0$  to  $W_{59}$ , each weight  $W$  being 48 bits long) and is defined by the following transformation:

$$S' = \left( \sum_{j=0}^{j=59} P_j W_j \right) \text{mod } Q$$

where:

- $Q = 2^{48}$
- the bits  $P_j$  of the presignature  $\mathbf{P}$  select the corresponding weights ( $W_j$ ) of the knapsack to form the integral sum modulo  $Q$

The result is the 48-bit knapsack sum  $\mathbf{S}'$ . The most significant bit of the sum is called Bit  $S'0$ .

#### 8.2.4 The Deletion Box

The Deletion Box deletes the 8 least significant bits of the 48-bit knapsack sum  $\mathbf{S}'$ , i.e. bits  $S'40$  through  $S'47$ . (It has been shown that the least significant bits are unacceptably weak from a cryptanalysis standpoint.) The result is the 40-bit authentication signature  $\mathbf{S}$  (numbered from Bit 0 to Bit 39, as for signature  $\mathbf{s}$ ).

#### 8.2.5 The Signature Comparator

The Signature Comparator merely compares the received 40-bit signature  $\mathbf{s}$  with the onboard generated 40-bit signature  $\mathbf{S}$ . The resulting validation signal is sent to the Supervisor for final authorisation of the TC Segment (comparison of received LAC contents with onboard LAC registers).

#### 8.2.6 The Authentication Key

The Authentication Key (which must be kept secret) consists of:

60 x 48-bit Hard Knapsack Weights	=	2880 bits
60 x 1-bit Hashing Function coefficients	=	60 bits
<hr/>		
Full Authentication Key	=	2940 bits

2880 bits represent (exactly)	:	360 octets
60 bits represent (not exactly)	:	8 octets
<hr/>		
2940 bits represent (not exactly)	:	368 octets

The system includes two such 2940-bit keys:

- a fixed, mission-unique Authentication Key, called the Fixed Key;
- an in-flight programmable Authentication Key, called the Programmable Key.

**(a) Fixed Key**

The Fixed Key is required for start-up and emergency (recovery) operations, and shall, normally, only be used during these phases. The contents of the Fixed Key are permanently stored in the AU (ROM) as part of the "Mission-Specific Data" already introduced in Subsection 3.4.4.

**(b) Programmable Key**

The Programmable Key is required for all normal operations. The contents of the Programmable Key reside in the AU (RAM), where they can be modified by means of Authentication Control Commands specifically defined for that purpose. The format of these Control Commands ("Change Programmable Key Block" Control Commands, which are specified in Section 8.4) allows any 5-octet block to be modified starting at any of 370 (74 x 5) octet boundaries. The Programmable Key memory shall therefore be designed accordingly.

The operational procedure for the modification of the Programmable Key is given in Section 8.5.

## **8.3 THE SUPERVISOR**

### **8.3.1 Functional Concept**

The Supervisor consists of four main parts:

- the Logical Authentication Channel (LAC) Registers;
- the Final Authorisation Function;
- the Control Command Processor;
- the Deletion Function.

They are briefly described in the next four subsections.

### 8.3.2 The LAC Registers

The concept of the three Logical Authentication Channels (LACs), with the function of the three LAC Counters and the format of the Authentication Tail, are introduced and specified in Section 10 of Reference [2].

Three LAC Registers shall be provided:

- one "Principal LAC" register (LAC ID = **00**);
- one "Auxiliary LAC" register (LAC ID = **01**);
- one "Recovery LAC" register (LAC ID = **10**).

Each register is, in principle, a 32-bit memory. However, the 2 bits of LAC ID (Bits 0 and 1) are fixed. The purpose of the LAC ID is to identify the LAC used by the sending end and, consequently, the LAC Register to be selected for the final authorisation of a TC Segment. As regards the 30 bits of LAC Count (Bits 2 through 31, where the LSB is Bit 31), they shall be implemented as follows:

- For the two first registers (Principal and Auxiliary), the 30 bits shall be fully in-flight programmable by means of the appropriate Authentication Control Commands (for details, see Sections 8.4 and 8.5) and maintained by the Supervisor after each successful authentication (one increment of the selected Counter after each authorisation).

The "cold start" initialisation state shall be: all 30 bits set to **1**.

- For the third register (Recovery), the original requirement is that the state of the 30 bits be permanently stored, even in the event of loss of power (duration of the event: unspecified). This requirement implies "non-volatility" of the data, which is a severe requirement from an implementation standpoint. Therefore, only the first 8 LSBs (Bits 24 through 31) shall be maintained (i.e. incremented and programmed in flight). The remaining 22 bits (Bits 2 through 23) shall be permanently set to **1**. This limitation is consistent with the operational procedures of Section 8.5, which restrict the use of the Recovery LAC.



### 8.3.3 The Final Authorisation Function

When the received signature **s** of a TC Segment is found to be identical with the onboard-generated signature **S**, the signal "Signature Valid" is sent to the Final Authorisation Function, where the contents of the received LAC Count field shall be compared with the contents of the indicated LAC Register. If the two contents are found to be equal, there are two cases:

- The TC Segment was transferred on an "authenticated" MAP with a MAP ID that can range from 0 to 62. In this case, the TC Segment shall be authorised for transfer to the Segmentation Layer.
- The TC Segment was transferred on MAP 63 (i.e. MAP 111111), which is DEDICATED to the transfer of Authentication Control Commands. In this case, the Control Command Processor is authorised to further process the TC Segment, which will never be transferred to the Segmentation Layer.

In both cases, the contents of the indicated LAC Register shall be incremented by 1.

### 8.3.4 The Control Command Processor

The function of the Control Command Processor is to process the special TC Segments called "Authentication Control Commands" (which have been authorised by the Final Authorisation Function) and execute the instructions they contain.

The formats of the various Authentication Control Commands are specified in Section 8.5.

Any TC Segment not conforming to the specified formats (i.e. both in length and in contents) shall be rejected and reported as "not executable" by means of the Frame Analysis Report (see Section 10, on Telemetry Reporting).

### 8.3.5 The Deletion Function

The Deletion Function deletes the Authentication Tail of all TC Segments that have been authorised by the Final Authorisation Function.

(The complete authentication process is meant to be transparent to an observer located at the receiving end of the Segmentation Layer.)

## 8.4 FORMATS OF THE AUTHENTICATED TC SEGMENTS

### 8.4.1 General Format

The general format of an authenticated TC Segment is specified in Reference [2]. This Specification provides the following additional requirements:

- (a) The length of the signature field of the Authentication Tail shall be 5 octets ( $N = 5$ ).
- (b) Therefore, the length of the Authentication Tail shall be 9 octets; the maximum length of the TC Segment shall be 240 octets, and its minimum length 10 octets.

### 8.4.2 Specific Formats

As specified in Reference [2], it is necessary to differentiate TC Segments containing the Authentication Control Commands required to reconfigure the AU. This is done by allocating the TC Segment Header contents "all ones" to these particular segments, i.e.:

- Sequence Flags set to **11** (Unsegmented)
- MAP ID set to **111111** (MAP 63)

The formats of the Authentication Control Commands are organised in three "Groups", as follows:

- One octet of TC Segment Header for all three Groups.
- One octet following the Segment Header to specify the Control Command Identifier, with Identifiers specific to each Group.
- Zero, four or eight octets of Control Command Data Field, depending on the selected Group.

Table 8.1 gives the complete list of Authentication Control Commands, with Group numbers, Control Command IDs and Command Names. Figure 8.5 shows the format of the TC Segment for each Group, complete with Authentication Tail.

Each Control Command is specified in the following subsections.

**8.4.3 "Dummy Segment" Control Command**

The purpose of this Group 1 Control Command is to serve as a "blank" or "NOP" (No Operation) for testing purposes, or as required during operations.

When authenticated, the TC Segment (which consists of two octets, the first containing all ones and the second all zeros) is deemed to be delivered to the AU where it will vanish and achieve nothing. However, since the AU has effectively authenticated the Dummy Segment, the contents of the LAC Register used during the authorisation process have been incremented and a telemetry report prepared accordingly ("authorised dummy segment received" report of the Frame Analysis Report).

GROUP	CONTROLCOMMAND IDENTIFIER (8 BITS)	COMMAND NAME
GROUP 1	0000 0000	DUMMY SEGMENT
	0000 0101	SELECT FIXED KEY
	0000 0110	SELECT PROGRAMMABLE KEY
	0000 0111	LOAD FIXED KEY IN PROGRAMMABLE KEY MEMORY
GROUP 2	0000 1001	SET NEW LAC COUNT VALUE
GROUP 3	0000 1010	CHANGE PROGRAMMABLE KEY BLOCK "A"
	0000 1011	CHANGE PROGRAMMABLE KEY BLOCK "B"

Table 8.1 LIST OF AUTHENTICATION CONTROL COMMANDS

1 octet	1 octet	9 octets
Segment Header	Control Command Identifier	Authentication Tail
11111111	00000***	LAC + Signature

### Group 1 Control Command

1 octet	1 octet	4 octets		9 octets
Segment Header	Control Command Identifier	LAC value to be set		Authentication Tail
11111111	00001001	LAC ID 2 bits	LAC Count 30 bits	LAC + Signature

### Group 2 Control Command

1 octet	1 octet	1 octet	7 octets	9 octets
Segment Header	Control Command Identifier	Start Address of new 40-bit Keyblock	Key-specific pattern (to be encoded)	Authentication Tail
11111111	0000101*			LAC + Signature

### Group 3 Control Command

Figure 8.5 FORMATS OF AUTHENTICATION CONTROL COMMANDS (FULL TC SEGMENT)

#### 8.4.4 "Select Key" Control Commands

Two such Group 1 Control Commands are provided for the selection of one of the two Authentication Keys:

##### (a) "Select Fixed Key"

This Control Command shall **mandatorily** be encrypted with the Fixed Key. The AU, which monitors the Control Command Identifier octet of each Authentication Control Command, shall select the Fixed Key **prior to authenticating** the TC Segment:

- If authentication is successful, the Fixed Key shall remain selected.
- If authentication is unsuccessful, the key previously in use shall remain selected.

##### (b) "Select Programmable Key"

This Control Command shall **mandatorily** be encrypted with the Programmable Key. The AU shall select the Programmable Key **prior to authenticating** the TC Segment:

- If authentication is successful, the Programmable Key shall remain selected.
- If authentication is unsuccessful, the key previously in use shall remain selected.

#### 8.4.5 "Load Fixed Key In Programmable Key Memory" Control Command

The purpose of this Group 1 Control Command is to reload the Fixed Key set of 2940 bits (which can be relied upon at all times) in the Programmable Key memory with a single command instruction (see also the Recovery Procedures of Section 8.5).

The key used for encrypting the TC Segment containing the Control Command will be whatever key was selected in the AU at the time the command was transmitted.

#### **8.4.6 "Set New LAC Count Value" Control Command**

The purpose of this Group 2 Control Command is to set the value of either of the three programmable LAC Registers (i.e. "Principal", "Auxiliary" and "Recovery"). Therefore, this Control Command shall only be implemented for LAC Identifiers **00**, **01** and **10**.

As soon as the TC Segment is authorised by the authentication process, the specified LAC Count value shall be forced into the selected LAC Register. The LAC specified in the Authentication Tail of the TC Segment can be any of the three LACs.

Note that the 22 MSBs of the 30-bit Recovery LAC Register are permanently set to all ones. Therefore, those same bits in a "Set New Recovery LAC Count Value" Control Command shall be ignored by the AU.

The key used for encrypting the TC Segment containing the Control Command will be whatever key was selected in the AU at the time the command was transmitted.

#### **8.4.7 "Change Programmable Key Block" Control Commands**

Two such Group 3 Control Commands are provided to cover the full size of the 2940-bit Programmable Key. This is because the START ADDRESS specified by the third octet of the Control Command allows ANY 5-octet block to be modified at ANY of 368 octet boundaries, as follows:

- Command "A" concerns the first 256 octet boundaries.
- Command "B" concerns the last 112 octet boundaries.

It shall be possible to load a 5-octet (40 bits) block starting from any of the 368 octet boundaries. In practice, the operational procedure shall consist of loading the 74 consecutive blocks of 40 bits required for the complete 2940-bit Programmable Key (this is specified in detail in Section 8.5).

Any transmission using the unused boundaries of Command "B" (from 113 to 256) shall be ignored and reported as "non-executable".

The key used for encrypting the TC Segments containing the sequence of Control Commands will be whatever key was selected in the AU at the time each Control Command was transmitted. (The consequences of using the Programmable Key are discussed in Section 8.5.)

Once the TC Segment has been authorised by the authentication process, the TC Segment, minus the 40-bit signature **s** (i.e. [m,l]) shall be **complemented** and passed once more through the signature-building process, i.e. through the Authentication Processor. The 24 bits of virtual fill **z** shall be inserted as before, i.e. they shall not be complemented, but shall remain "all zeros". The result of the process is a 40-bit "pseudo-signature" which, instead of being sent to the Signature Comparator, shall be loaded in the Programmable Key memory, starting at the octet location indicated by the START ADDRESS field, as follows:

- Bits 32 through 39 of "pseudo-signature" at the indicated octet location;
- Bits 24 through 31 of "pseudo-signature" at the next location (START ADDRESS + 1);
- And so on, until Bits 0 through 7 are loaded at location START ADDRESS + 4.

Thus, 40 contiguous bits of the Programmable Key have been changed.

## **8.5 OPERATIONAL PROCEDURES**

### **8.5.1 Introduction**

This section on operational procedures covers:

- The standard set of procedures required to change the contents of the Programmable Key.
- The recovery procedures that have been envisaged to cope with the various emergency situations and which directly involve one of the redundant TC Decoders complete with its AU.

These are discussed in the next two subsections.

### **8.5.2 Procedures for Changing the Programmable Key Contents**

#### **(a) Procedural aspects specific to the authentication system**

Cryptanalysis studies have shown that each Authentication Key (i.e. 2940 bits) must be selected according to severe cryptographic criteria. The

process used to select each new Authentication Key can provide any number of keys, as required for the operations of a particular mission. Each selected key is fully validated from a cryptographic standpoint.

Therefore, although the specification of the AU makes it mandatory to change the contents of the Programmable Key by an amount as small as a block of 40 bits (5 octets), starting from **any** of 368 octet boundaries (as specified by the START ADDRESS in the Control Command), this shall not (normally) be done in practice. Instead, the entire contents of the "old" key shall be modified by re-loading all 2940 bits making up the new key.

### **(b) Specification of the loading method**

The loading of a new 2940-bit Programmable Key shall be done with the smallest amount of Authentication Control Commands, and in a standard way. The aim is to standardise the operational procedures and automate the ground system, thus improving the efficiency and reliability of spacecraft operations.

Therefore, only 74 Control Commands shall be uplinked, starting with:

- Change Programmable Key Block "A" Control Command:
  - counting up 52 Control Commands from START ADDRESS (decimal) 0, 5, 10, ..., etc., until 255, for the loading of the first 260 octets (52 x 5 octets) of the 368-octet memory.
  
- Change Programmable Key Block "B" Control Command:
  - counting up 22 Control Commands from START ADDRESS (decimal) 4, 9, 14, ..., etc., until 109, for the loading of the last 108 octets of the 368-octet memory (22 x 5 = 110 octets, where the last 16 bits (Bits 24 through 39) of the 40-bit "pseudo-signature" loaded in the memory as a result of the last Control Command being authorised by the AU are NOT USED).

The first key bits to be loaded shall belong to the Hard Knapsack weights. The first Control Command (Change Programmable Key Block "A" Command, with START ADDRESS "0") will result in the last 40 bits (Bits 47 to 8) of the first weight (W0) to be loaded in the memory octet locations with addresses 0,1,2,3 and 4: Bit 32 of the "pseudo-signature" shall be loaded



as Bit 40 of W0, Bit 39 as Bit 47 of W0, Bit 24 as Bit 32 of W0, Bit 31 as Bit 39 of W0, and so on.

The last key bits to be loaded shall be the 60 one-bit coefficients of the Hashing Function.

The 73rd Control Command shall load the C24 to C59 coefficients as follows:

- bit 32 to 35 of the "pseudo-signature" are **not significant**;
- bit 36 of the "pseudo-signature" shall be loaded as coefficient C59;
- bit 39 of the "pseudo-signature" shall be loaded as coefficient C56;
- bit 24 of the "pseudo-signature" shall be loaded as coefficient C55; and so on, until
- bit 7 of the "pseudo-signature" shall be loaded as coefficient C24.

The 74th Control Command shall load the remaining coefficients (C0 to C23) as follows:

- bit 32 of the "pseudo-signature" shall be loaded as coefficient C23;
- bit 39 of the "pseudo-signature" shall be loaded as coefficient C16; and so on, until
- bit 23 of the "pseudo-signature" shall be loaded as coefficient C0. The remaining bits of the "pseudo-signature" (bit 0 to 15) are **not significant**.

Figure 8.6 shows the organisation of the Programmable Key memory and its contents. The bits of no significance shall **conventionally** be set to **zeros** for what concerns the process at the sending end.

(This is for the benefit of the authentication process at the sending end, where the values of the unused bits **must be known** at all times during the encryption process.)

There are now two possible cases:

- The loading operations are made with the Fixed Key.
- The loading operations are made with the Programmable Key.

The first case is straightforward, since the encrypting key on ground remains the **same** during the entire operation. However, although it is a requirement

Address (decimal)			Command rank		
Bank A	000	40	W0(40 to 47)	47	1 <sup>st</sup>
	001	32	W0(32 to 39)	39	
	002	24	W0(24 to 31)	31	
	003	16	W0(16 to 23)	23	
	004	8	W0( 8 to 15)	15	
	005	0	W0( 0 to 7)	7	
	006	40	W1(40 to 47)	47	
007	32	W1(32 to 39)	39		
Bank A	255	16	W42(16 to 23)	23	
Bank B	000	8	W42( 8 to 15)	15	52 <sup>nd</sup>
	103	0	W59( 0 to 7)	7	72 <sup>nd</sup>
	104	59	C(59 to 56)	56	
	105	55	C(55 to 48)	48	
	106	47	C(47 to 40)	40	73 <sup>rd</sup>
	107	39	C(39 to 32)	32	
	108	31	C(31 to 24)	24	
	109	23	C(23 to 16)	16	
	110	15	C(15 to 8)	8	74 <sup>th</sup>
	111	7	C( 7 to 0)	0	

Figure 8.6 ORGANISATION OF THE PROGRAMMABLE KEY MEMORY

that the Fixed Key be selectable for the encryption of the uploading Control Commands, this will not be the preferred mode of operation.

The second case is more complex, since the contents of the Programmable key are **changed** after each "Change Programmable Key Block" Control Command has been successfully accepted by the AU. However, the process is completely predictable and can therefore be automated.

### 8.5.3 Recovery Procedures

#### (a) Overview of the typical emergency situations

This subsection covers the recovery procedures envisaged during various emergency situations. The scenarios described are provided for guidance. These scenarios were evolved as requirements for the study and specification of the authentication system. However, the final definition on what the recovery procedures will effectively be lies with each particular mission project.

The emergency situations likely to be encountered can be listed as follows:

**Emergency A:** One TC Decoder fails to deliver command data because (in the case of interest here) of a malfunction in the AU.

**Emergency B:** A power-down event occurred on board the spacecraft which erased/destroyed the contents of the AU's volatile memory. In this scenario, the contents of the Programmable Key and of the two programmable LAC Registers (Principal and Auxiliary) have now been replaced by their "cold start" values (i.e. by the contents of the Fixed Key for the first, and the value "all ones" for the last two).

**Emergency C:** A telemetry subsystem failure on board the spacecraft has resulted in the telemetry downlink being interrupted.

**Emergency D:** The contents of the Programmable Key have been corrupted (i.e. they are not known any more). Therefore, a new value of the Programmable Key must be loaded in the best conditions of secrecy.

Each situation is discussed in the next paragraphs.

**(b) Emergency A**

In this situation, the redundant TC Decoder must be used.

It is important to note that for the redundant TC Decoder to fully achieve its role (which is to allow only authorised TC Segments to access the spacecraft Data Management System), it must be ensured that the TC Decoder has failed in a permanent "Unauthorised" state (i.e. the failed TC Decoder does not allow **anything** to go through).

**(c) Emergency B**

The provisions made to cover this type of emergency situation are suitable to other situations (such as Emergency D).

In the case of a temporary power loss on board the spacecraft, all volatile memory contents are expected to be lost. In such a case, a provision must be made in the on-board AU which ensures that, when power fails, the contents of at least one LAC Register are preserved (i.e. permanently memorised): this is the Recovery LAC Register.

When power returns, the AU shall:

- be in Fixed Key mode of operation;
- load the Fixed Key contents into the Programmable Key memory;
- set the contents of the two "non-permanent" LAC Registers to "all ones".

The procedure for re-configuring the AU to its original settings is as follows:

- A "SET NEW LAC COUNTER VALUE" Control Command is sent to set one of the two "nonpermanent" LAC Registers (e.g. the Principal LAC Register) to its correct value. This is done by means of the Fixed Key and the Recovery LAC Register (the contents of the latter are assumed to be reliably known).
- The "SELECT PROGRAMMABLE KEY" Control Command is sent, using the contents of the Programmable Key (which are, in this particular case, those of the Fixed Key) and of the newly set-up Principal LAC Register.

- The required number of "CHANGE PROGRAMMABLE KEY BLOCK" Control Commands is sent to change the contents of the Programmable Key to the value of a new key, still using the Principal LAC.
- The remaining (Auxiliary) LAC Register is then set to its correct value, using the newly loaded Programmable Key and the Principal LAC.

#### **(d) Emergency C**

When the telemetry system has failed on board the spacecraft and no downlink is available (i.e. no CLCW, no telemetry Source Packets), the first step will be to send one reconfiguration command.

This command (which may be provided by the CPDU) is to be foreseen by the particular mission project. There are several possible options. The most likely to be selected is to send a command that will disconnect the AU of one of the two TC Decoders.

The command (possibly a short CPDU TC Packet, with one Command Pulse instruction) will have to be sent "blind", using the Expedited Service of COP-1 (BD Transfer Frame). The LAC used shall be the Recovery LAC.

The command will be transmitted several times, for safety, as there is no means – in principle – of verifying its actual execution.

Assuming the command has been executed, the next step will be to send whatever commands were foreseen to re-establish the telemetry link. The transmission mode still being "blind", BD Transfer Frames shall be used.

If these attempts have been unsuccessful, the first step must be repeated, followed by the next step, until they are successful. If the commands persist in not being executed, the redundant TC Decoder must be selected for the same procedure to be repeated.

A possible additional feature may consist in generating the AU-disconnecting command automatically on board the spacecraft, after a certain time without telecommand contact from ground has elapsed.

#### **(e) Emergency D**

This scenario should not be confused with that of Emergency B.

If the contents of the Programmable Key have been corrupted by errors (e.g. errors due to single-event upset), the "new" contents are now unknown. In such cases, a useful command is the "LOAD FIXED KEY IN PROGRAMMABLE KEY MEMORY" Control Command, which forces the Fixed Key PROM contents into the Programmable Key RAM. However, since the Programmable Key cannot be used anymore (it is unknown), it may be necessary to select the Fixed Key ("SELECT FIXED KEY" Control Command) before sending the "LOAD FIXED KEY IN PROGRAMMABLE KEY MEMORY" Control Command.

The next step will be to send a "SELECT PROGRAMMABLE KEY" Control Command, followed by a standard reload of a complete new key.

Another possibility is to send the "SELECT FIXED KEY" Control Command and reload a new Programmable Key with the Fixed Key. However, in this case, the Fixed key shall be used 74 times for the reload, whereas it was used only once during the reload operation of the first scenario (because the contents of the Programmable Key change constantly during the entire loading).

## 8.6 DESIGN REQUIREMENTS

Because of the particular characteristics of the AU, most design requirements have already been covered in the previous sections. The following is a list of design requirements of note, whether they have already been mentioned or not:

- (a) It shall be possible to disconnect the AU by means of (typically) a Command Pulse. Therefore, a pin shall be foreseen that disables the AU.
- (b) At switch-on time, or after a temporary loss of power, the AU shall return to a "cold start" initialisation state which can be recapitulated as follows (see also Section 10, on Telemetry Reporting):
  - Key in use: Fixed Key
  - Contents of Programmable Key: Undefined. (It is suggested that the Fixed Key contents be loaded at that time, if this can be done in a short time and without any inconvenience.)
  - Contents of Principal and Auxiliary LAC Registers: all ones

- (c) The 8 LSBs of the Recovery LAC Register must be maintained and permanently memorised, even in case of loss of power.
- (d) Once the AU has started an authentication process, it shall not be possible to interrupt or abort this process. This applies, in particular, to the arrival of a new TC Segment, whether it was transferred by means of an AD or a BD frame: such an arrival shall not disturb the on-going authentication process.
- (e) The AU shall not cause the Wait Flag of FARM-1 to be set on (i.e. to be set to 1) when TC Segments to be authenticated and passed to the Segmentation Layer are transmitted by means of the COP-1 Sequence-Controlled Service (AD Service).

However, during AU control operations, the AU will be required to perform the complex authentication processes that characterise certain Authentication Control Commands (such as the "Change Programmable Key Block" Control Command). It can be assumed that such operations will only be performed when the Sequence-Controlled Service of COP-1 is available.

As a consequence, the implementer is allowed (after proper justification and agreement to proceed) to make use of the COP-1 data flow control mechanism (i.e. of the FARM-1 Wait Flag) for dealing with this specific type of problem, in order to simplify the design of the TC Decoder. It is emphasised that this allowance only applies to TC Segments carrying AU Control Commands (MAP 63).

The implementer shall provide all necessary engineering data concerning the delays caused by the AU during the transfer of TC Segments through the uplink. This shall include any operational limitation to be observed when the Recovery LAC Register is used (e.g. data rate limitation when electromechanical devices are used to implement the nonvolatile memory).

- (f) The design of the AU shall be such as to have, in the event of a failure in the AU itself, the highest probability to fail in a permanently "unauthorised" state. This means that, unless the AU is disconnected by a reconfiguration command, the consequence of the failure is that all TC Segments are prevented from reaching the Segmentation Layer.

- (g) The design of the VLSI TC Decoder shall provide an interface for an external AU circuit (see Implementation Requirement of Subsection 3.4.6, on Physical Distribution of Major Functions). This interface can be based on the MAP interface, if this is suitable to the implementer.



## 9. COMMAND PULSE DISTRIBUTION UNIT

### 9.1 GENERAL REQUIREMENTS

The requirements that lead to the definition of the CPDU are determined by the operational environment of the free-flying spacecraft: it is an automated vehicle that cannot be reached by human crews and repaired by them in case of failure or severe anomaly.

In most instances, the human operators on ground may rely on the nominal telemetry and telecommand functions of the spacecraft Data Management System (DMS): telemetry supervisory data are available, the telecommand Sequence-Controlled Service (i.e. the AD Service) can be used, and the data distribution services of the DMS are functional.

However, if and when the DMS fails, these operators may be confronted with a complete lack of nominal telemetry and telecommand functions: there is no telemetry, no data distribution services may be expected on board the spacecraft, and only the telecommand Expedited Service (i.e. the BD Service) can be used. Nevertheless, the spacecraft operators must be provided with the means to reconfigure the DMS, using whatever redundancy is available on board the spacecraft. This is where the CPDU comes into play.

The CPDU, as shown in Figure A-1 of Appendix A, is a simple unit that is **solely** accessible from ground. It may have to operate command pulse lines that also require to be operated through the DMS (for instance: because of a requirement for the delayed (time-tagged) execution of some critical command): there is no contradiction in this, even if it leads to the apparent duplication of command pulse lines driving certain "actuators" (e.g. relays). The CPDU can also be used to "authorise" the DMS to access potentially dangerous "actuators" at certain given times during a mission (e.g. for the delayed execution of the critical command already mentioned above).

The CPDU is identified by the Application Identifier placed in the TC Packet Header. Application Identifiers are mission-specified, and must be specific to each Application Process on board a given spacecraft. Therefore, the Application Identifier of each CPDU shall be programmable.

## 9.2 SPECIFICATION

### 9.2.1 Checking the TC Segment

Notwithstanding cross-coupling of redundant CPDUs, the CPDU receives TC Segments (each Segment containing a complete TC Packet) from one MAP interface that responds to only **one** MAP Identifier. In theory, the MAP Identifier can be any of the possible 64. In practice, the MAP Identifier assigned to that CPDU will be MAP 0.

When CPDUs are cross-coupled, the Identifier of the second MAP interface may be any value other than MAP 0. In practice, MAP 1 will be a likely choice.

When it receives the TC Segment, the CPDU stores it for further processing of the TC Packet it must contain, **unless** it has already received one such TC Packet and **not completed** the execution of the command instructions it contained down to **the last command pulse**: in which case, it shall **ignore any incoming** TC Segment, whether it was transferred in an AD or a BD Transfer Frame. This is important: there is no Packetisation Layer "abort" command foreseen for the CPDU. Once it has accepted a TC Packet, the CPDU cannot release it until all command instructions specified in that packet have been executed.

#### IMPORTANT NOTE

The above must not to be confused with the "Aborted Data Transfer" (ADT) signal provided by the MAP interface and which is specified in Section 11. The ADT signal is used to indicate that a BD frame has arrived, and that this has resulted in TC Segment data being erased (TC Segment data that were resident in the storage device of the Segmentation Layer: the "back-end" buffer). In such a case, any **incompletely** transferred TC Segment data **must** be erased. This also applies to the CPDU.

The CPDU is a simple unit: no Packet Assembly Controller (PAC) is necessary or foreseen. All TC Packets are required to be carried inside a single TC Segment and, therefore, a single TC Transfer Frame (because of the BD Service requirement).

The 6-bit MAP Identifier of the TC Segment Header is ignored (this function is already provided by the MAP interface allocated to the CPDU).

### 9.2.2 Checking the CPDU-specific TC Packet

The TC Packet specified to operate a CPDU shall have a minimum (total) length of 10 octets, and a maximum length of 248 octets. It shall always be made of an **even** number of octets.

The last two octets of the Packet shall be used for error detection: they shall contain a 16-bit CRC identical to that used in the TC Transfer Frame. As for the Transfer Frame, it will be used to detect errors over the complete Packet structure.

When the TC Segment Segmentation Flags have been checked to be **11**, the **actual** number of octets making up the TC Packet shall be verified to be (a) consistent with the Packet Length field and (b) an even number between 10 octets and the maximum value allowed by the particular implementation (see Design Requirements, in Section 9.3).

After this check has been passed, the position of the CRC being known, an error check over the full Packet will take place. If the Packet is found to be error-free ("CLEAN"), the process continues. Otherwise, the complete TC Packet is erased.

After the TC Packet has been found "CLEAN", the following fields are checked:

- Version Number : must be **000**
- Type Bit : must be **1**
- Data Field Header Flag : must be **0**  
(CCSDS term:Secondary Header Flag)
- Application Process Identifier : the 11 bits must be as programmed for the particular CPDU (see also Section 3.4.4)
- Packet Sequence Flags : must be **11**

- Packet Name or Sequence Count : **not verified**, only telemetered back to ground, as specified in Section 10
- Packet Length : already checked by the "CLEAN" verification process

When all the above checks have been passed successfully, the TC Packet is declared "LEGAL" and its Application Data (command instructions) read out and executed as described in the next Subsection.

If any of the checks fails, the Packet is erased. (See also the CPDU Status Report in Subsection 10.3, in which the telemetry reports for these checks are specified.)

### 9.2.3 Processing the Application Data

The Application Data of the CPDU consist of at least one command instruction in the form of one double octet, or several of such double-octet command instructions, up to the maximum allowed by the particular implementation.

Each double octet is formatted as follows:

- First octet : specifies one of 256 Command Pulse outputs.
- Second octet : specifies the duration of the Command Pulse to be issued on the specified output, as follows:
  - Bits 0 to 4 (5 MSBs) are reserved by ESA for future use. They will normally be set to all zeros by the sending end. They will be IGNORED by the CPDU.
  - Bits 5 to 7 (3 LSBs) specify the duration of the pulse as follows:

<b>000</b>	=	1 x D , where D (for "Duration") is a fixed value to be selected by the implementer, and which can be any value between 10 and 15 milliseconds.
<b>001</b>	=	2 x D
<b>010</b>	=	4 x D
<b>011</b>	=	8 x D
<b>100</b>	=	16 x D
<b>101</b>	=	32 x D
<b>110</b>	=	64 x D
<b>111</b>	=	128 x D (i.e. a value between 1.28 and 1.92 seconds)

When there are more than one command instruction in the Packet, each instruction shall be executed one after the other, in the original sequence.

A TC Packet may happen to contain command instructions that correspond to outputs that have not been physically implemented (a possible instance during testing). In such a case, the CPDU is not expected to detect this anomaly, and will behave as if the output existed, i.e. there will be no report of an anomaly by the CPDU (the anomaly is expected to be detected at a higher level, where the lack of response or effect will be reported).

### **9.3 DESIGN REQUIREMENTS**

#### **9.3.1 Application Identifier**

The Application Identifier belongs to the category of "Mission-Programmable Data". For each CPDU, it shall be possible to easily program (or re-program) the 11-bit Application Identifier, as already introduced in Subsection 3.4.4 on "Mission-Specific Data Programming Capability".

#### **9.3.2 Maximum Capability of the CPDU**

The Command Pulse output circuitry, as provided by the "basic" CPDU, shall allow a particular flight unit to be equipped with any number of discrete pulse outputs, up to 256, as required for the mission.

As regards the minimum and maximum lengths of a CPDU-specific TC Packet, they are specified in Subsection 9.2.2 to be 10 and 248 octets respectively. The designer shall select the maximum capacity of his CPDU:

this is the size of the largest TC Packet his CPDU can accept. This shall be selected between the following two capacity values:

- Smallest capacity: packet length of 32 octets (i.e. a capability of 12 command instructions)
- Largest capacity: packet length of 248 octets (i.e. a capability of 120 command instructions)

### 9.3.3 Execution of Command Instructions

A CPDU-specific TC Packet shall always contain a minimum of one command instruction (i.e. 2 octets of command data); otherwise, the packet shall be rejected.

The execution of the command instruction shall always occur some fixed time after the packet has been declared "LEGAL". The exact value of this delay is specific to the particular implementation. However, it shall never exceed:

- $1 \times D$  (i.e. 10 to 15 milliseconds)

When the Packet contains more than one instruction, the delay between a Command Pulse and the next shall also be fixed and specific to the particular implementation. However, the following values are recommended to the designer:

- Minimum delay:  $D/2$  (i.e. 5 to 7.5 milliseconds)
- Maximum delay:  $D$  (i.e. 10 to 15 milliseconds)

### 9.3.4 Data Flow Control

The CPDU shall be equipped with an active Data Terminal Ready (DTR) signal line on its MAP interface, so that it can operate safely in the COP-1 Sequence-Controlled Service mode (i.e. without possible omissions of TC Segments and, therefore, CPDU Packets).

Note that, if the CPDU and the "basic" TC Decoder are implemented in the same VLSI device, all signal lines of the MAP interface selected for that CPDU (e.g. MAP 0) can be permanently connected on-chip.

For details on the DTR signal line of the MAP interface, see Section 11, on Interfaces.

### **9.3.5 Output Waveforms**

This is the subject of Section 11, on Interfaces.

**PAGE INTENTIONALLY LEFT BLANK**



## 10. TELEMETRY REPORTING

### 10.1 GENERAL REQUIREMENTS

Telemetry reporting is essential to the normal operation of the telecommand data communication system.

In this document, for the sake of clarity, the following terms are used with a specific meaning: Status Data and Survey Data. These are discussed in the next two subsections.

#### 10.1.1 Status Data

These are data that provide the status of certain functions and the values of some of their variables. They are essential to the proper operation and control of the spacecraft at **all times**: the last received report is that which matters most from an operational standpoint.

These status data are sampled by the on-board telemetry subsystem (one function of the DMS) at a rate which is defined to suit the operational needs of the particular mission.

The status data reports defined in this document are absolutely required for the proper operation of the spacecraft (that is, in nominal conditions, when telemetry is available). There are two groups of status data reports:

#### **FIRST GROUP:**

The CLCW Status Report, which is required for the automated operation of the data link procedure COP-1. The periodic readout of this report is subject to specific requirements which are developed in Section 10.2.

#### **SECOND GROUP:**

The CPDU Status Report and the Authentication Unit (AU) Status Report (the latter when this option is selected for the mission). Both reports are subject to requirements that are less specific than those of the CLCW Status Report and are developed in Section 10.3 and 10.4.

### 10.1.2 Survey Data

These are data that also provide the status of certain functions and the values of some of their variables, but are not immediately essential to the operation and control of the spacecraft. They are, typically, data destined to be processed off-line to **survey** the behaviour of subsystem functions throughout the lifetime of the spacecraft. They can also be of great value for the diagnosis of anomalies and failures.

Survey data, if collected in their entirety, could be demanding in terms of telemetry sampling rates. However, it appears that, in most cases, it is not necessary to accommodate such data bandwidth requirements, and that good results can be obtained with modest sampling rates.

Only one survey data report is defined in this document: the "Frame Analysis Report" (FAR), which is further discussed and specified in Section 10.5.

The FAR is required for the proper testing and check-out of the TC Decoder.

### 10.1.3 Data Report Storage before Readout

Data reports, whether they are Status Reports or Survey Reports, shall not be disrupted during telemetry readout. In particular, they shall not be affected by the arrival of new report data.

To meet this requirement, the use of two registers is recommended:

- one register to hold the latest data report, ready for readout;
- one register to store new report data when they are generated.

If the telemetry interface sampling rate is slower than the rate at which new data reports are generated, the double register mechanism shall ensure that the latest complete data report is read out.

Survey data reports contain one flag bit that indicates that the report was read out more than once, as follows:

- Flag bit = **0** = NEW SURVEY DATA
- Flag bit = **1** = OLD SURVEY DATA

#### 10.1.4 "Cold Start" Data

In the sections describing the various data reports, several data values are specified as "Cold Start" values. Unless otherwise agreed, these values shall be set as part of the initialisation process, when the power is switched on. Each value has been chosen to reflect the initial state of the relevant process.

### 10.2 CLCW STATUS REPORT

#### 10.2.1 Specification

The CLCW is fully specified in Section 6 of Reference [2]. Furthermore, Reference [3] covers the telemetry communication side of the CLCW specification (the CLCW belongs to the Operational Control Field of the telemetry Transfer Frame).

The TC Decoder is only required to provide the **last** 16 bits of the 32-bit CLCW defined in Reference [2], that is:

- from Bit 16 ("No RF Available" Flag) through Bit 31 (last bit of Report Value field).

The first bit to be read out is Bit 16.

The missing 16 bits (Bit 0 through Bit 15) will be generated by the telemetry subsystem, since all these bits have fixed values that are either already specified in Reference [2] or specific to the mission.

#### 10.2.2 Design Requirements

The 16-bit CLCW Status Report will, typically, be sampled and read out by a telemetry interface provided by the Transfer-Frame-generating element of the DMS, at Master Channel level (rather than at Virtual Channel level). The implementation and configuration of this telemetry element is (as for the TC Decoder element) critical from both a data communication and a reliability point of view.

The current organisation of telemetry hardware developed by ESA foresees that Bit 16 ("No RF Available" Flag) and Bit 17 ("No Bit Lock" Flag) of the telemetry Transfer Frame CLCW will be acquired separately by this hardware. Therefore, in the CLCW Status Report, the state of the same flags is undefined.

The telemetry interface required to read out the CLCW Status Report is specified in Section 11, on Interfaces. This interface is specifically adapted to the serial readout of 16 bits of data and does not present any particular difficulty. Bit 16 of the CLCW Status Report shall be read out first, and Bit 31 read out last.

There shall be one CLCW Status Report for each TC Decoder.

### 10.3 CPDU STATUS REPORT

#### 10.3.1 Specification

The CPDU Status Report consists of 16 bits of status data formatted as follows:

<b>BITS</b>	<b>VALUE</b>	<b>MEANING</b>
0,1	<b>00</b>	"Cold Start"
	<b>01</b>	(Last) TC Packet accepted LEGAL
	<b>10</b>	(Last) TC Packet accepted CLEAN, but erased as NOT LEGAL
	<b>11</b>	(Last) TC Packet erased as NOT CLEAN
2 through 15	All 1	"Cold Start"
	All other values	Packet Sequence Count (or Name) of last LEGAL TC Packet.

The first bit to be read out is Bit 0.

#### 10.3.2 Design Requirements

The 16-bit CPDU Status Report will, typically, be sampled and read out by a telemetry interface provided by a standard data acquisition element of the DMS. In a Packet Telemetry environment, the CPDU Status Report will be placed in a telemetry Source Packet for transfer to ground via one of the mission-specified telemetry Virtual Channels.

The telemetry interface specified to read out the CPDU Status Report is fully specified in Section 11, on Interfaces. It is identical to that used for reading

out the CLCW Status Report. Bit 0 of the CPDU Status Report shall be read out first.

There shall be one CPDU Status Report for each CPDU.

## 10.4 AU STATUS REPORT

### 10.4.1 Specification

The AU Status Report consists of 80 bits (i.e. 10 octets) of status data formatted as follows:

- |                    |   |  |
|--------------------|---|--|
| Bits 0, 1          | : | Shall be permanently set to a value to be defined by the implementer (suggested value : <b>00</b> ).                           |
| Bits 2 through 31  | : | Current value of the contents (LAC Count) of the first (Principal) LAC Register. The LSB of the LAC Count value is in Bit 31.  |
| Bits 32, 33        | : | Shall be permanently set to a value to be defined by the implementer (suggested value: <b>01</b> ).                            |
| Bits 34 through 63 | : | Current value of the contents (LAC Count) of the second (Auxiliary) LAC Register. The LSB of the LAC Count value is in Bit 63. |
| Bit 64             | : | Key in use by AU:<br><div style="margin-left: 100px;"> <b>0</b> = Fixed key,<br/> <b>1</b> = Programmable key </div>           |
| Bits 65 through 71 | : | Reserved for use by ESA (and set to <b>0</b> )   |
| Bits 72 through 79 | : | Current value of the 8 LSBs of the third (Recovery) LAC. The LSB of the LAC is in Bit 79.                                      |

The first bit to be read out is Bit 0.

All "Cold Start" initialisation states of the AU are specified in Section 8.

### 10.4.2 Design Requirements

The 80-bit AU Status Report will, typically, be sampled and read out by a telemetry interface provided by a standard data acquisition element of the

DMS. In a Packet Telemetry environment, the AU Status Report will be placed in a telemetry Source Packet for transfer to ground via one of the mission-specified telemetry Virtual Channels.

The telemetry interface specified to read out the AU Status Report is fully defined in Section 11, on Interfaces. In principle, two 16-bit telemetry interfaces shall be used, each interface being identical to that already used for the CLCW Status Report. One interface is used to read out the first 16 bits of the AU Status Report (Bits 0 through 15). The other telemetry interface is used to read out the last 64 bits of the AU Status Report (Bits 16 through 63), in four successive 16-bit readouts. These four readouts shall necessarily occur after the first 16 bits of the AU Status Report have been acquired by the first interface.

There shall be one AU Status Report for each AU.

## 10.5 FRAME ANALYSIS REPORT (FAR)

### 10.5.1 Specification

The FAR consists of 32 bits of survey data formatted as shown below.

There shall be one report for each CLTU (Frame) event.

When the AU is not used, the Authentication Process Analysis data shall report "No authentication report" (000). When the AU is used, but the MAP Identifier of the TC Segment is not one of the "authenticated" MAPs, the same "No Authentication Report" value (000) shall be reported.

<b>BITS</b>	<b>VALUE</b>	<b>MEANING</b>
0		STATUS OF SURVEY DATA
	0	New survey data (or "Cold Start")
	1	Old survey data
1,2,3		FRAME ANALYSIS (8 STATES)

NOTE: The report of the lowest rank (i.e. of lowest 3-bit value) has precedence in case of conflicting states.

	<b>000</b>	Abandoned CLTU (see Note 1)
	<b>001</b>	Frame declared DIRTY
	<b>010</b>	Frame declared ILLEGAL for one reason
	<b>011</b>	Frame declared ILLEGAL for multiple reasons
	<b>100</b>	Frame (AD) discarded because of LOCKOUT
	<b>101</b>	Frame (AD) discarded because of WAIT
	<b>110</b>	Frame (AD) discarded because of N(S) or V(R)
	<b>111</b>	Frame (AD, BD or BC) "Accepted" by FARM-1
4,5,6		"LEGAL/ILLEGAL" FRAME QUALIFIER (8 STATES)
		NOTE: When a frame is declared ILLEGAL for multiple reasons, only the reason of the first rank (i.e. of lowest 3-bit value) is reported. The fields mentioned are those of the frame header.
	<b>000</b>	No ILLEGAL report (or "Cold Start")
	<b>001</b>	Error in fixed fields (Version & Reserved)
	<b>010</b>	Illegal combination (AC) of Bypass & Control Command flags
	<b>011</b>	Wrong Spacecraft ID (10 bits)
	<b>100</b>	Wrong VC ID (because of Bits 0 to 4 of ID)
	<b>101</b>	Wrong VC ID (because of Bit 5 of ID)
	<b>110</b>	N(S) of BC or BD frame not set to all 0
	<b>111</b>	Wrong BC frame data format (not executable)
7 through 12		COUNT OF ACCEPTED CODEBLOCKS PER CLTU
	xxxxxx	Straight 6-bit binary count of correct or single-error-corrected codeblocks in one CLTU; saturates at maximum value, no roll over. ("Cold Start" value: <b>000000</b> )
13,14,15		COUNT OF SINGLE-ERROR CORRECTIONS PER CLTU
	xxx	Straight 3-bit binary count, saturates at maximum value, no roll over. ("Cold Start" value: <b>000</b> )
16,17		"LEGAL" FRAME QUALIFIER (4 STATES)

	<b>00</b>	AD frame
	<b>01</b>	No report on LEGAL frame (or "Cold Start")
	<b>10</b>	BD frame
	<b>11</b>	BC frame
18,19,20	xxx	SELECTED CHANNEL INPUT (MAXIMUM CAPABILITY: 8 INPUTS)  ("Cold Start" value: <b>111</b> )
21 through 26	xxxxxx	LAST MAP ADDRESSED (64 MAPS)  ("Cold Start" value: shall be defined by the implementer; <b>111111</b> is suggested)
27	<b>0</b>	RESERVED BY ESA (SET TO <b>0</b> )
28,29,30		AUTHENTICATION PROCESS ANALYSIS (8 STATES)
	<b>000</b>	No authentication report (or "Cold Start")
		AUTHORISED SEGMENT QUALIFIER
	<b>001</b>	Authorised data segment
	<b>010</b>	Authorised (and executable) Authentication Control Command
	<b>011</b>	Authorised "dummy segment" received
		REJECTED SEGMENT QUALIFIER
	<b>100</b>	Error in Signature
	<b>101</b>	Error in LAC
	<b>110</b>	Wrong format (not executable) of authorised Authentication Control Command (includes Segment Header)
	<b>111</b>	Wrong length of TC Segment prior to being authenticated (authorised), i.e. length shorter than 10 octets
31	<b>0</b>	RESERVED BY ESA (SET TO <b>0</b> )

---

NOTE (1): "Abandoned CLTU". This state (**000**) shall be used to indicate:

- "Cold Start";



- First TC Codeblock of CLTU was abandoned (erased) because of Event 4 or Event 2;
  - More than 37 TC Codeblocks were found acceptable in the CLTU;
  - Physical Layer input Symbol Clock signal disappeared.
- 

### 10.5.2 Design Requirements

The 32-bit FAR will, typically, be sampled and read out by a telemetry interface provided by a standard data acquisition element of the DMS. In a Packet Telemetry environment, the FAR will be placed in a telemetry Source Packet for transfer to ground via one of the mission-specified telemetry Virtual Channels.

The telemetry interface specified to read out the FAR is fully defined in Section 11, on Interfaces. As for the AU Status Report, two telemetry interfaces are used to read out the 32-bit FAR. The first 16 bits of the FAR (Bits 0 through 15) are first read out through one of the two interfaces. The last 16 bits of the FAR (Bits 16 through 31) are then read out through the second interface.

There shall be one FAR for each TC Decoder.

NOTE: As stated in Section 3.4, the capability to telemeter the FAR shall always be provided by the TC Decoder. However, it is recognised that some particular mission-specific implementation could make it desirable to simplify that part of the TC Decoder design which is devoted to the FAR data collection. This can be achieved by deleting those parts of the report that are unwanted, while not changing the basic FAR format. Such deletions shall require prior agreement.

**PAGE INTENTIONALLY LEFT BLANK**

## 11. INTERFACES

### 11.1 INTRODUCTION

As already stated in Section 1.2, on SCOPE, and in Subsection 3.4.8, on Technology and Electrical Characteristics, this Specification is essentially a **functional design specification**. Consequently, electrical levels, related time values (such as switching times) and technology-specific issues are outside its scope.

The interfaces specified in this section are described in a functional way, with logical states (ACTIVE / INACTIVE) and basic waveforms.

Unless otherwise agreed (e.g. by the statement of work of a particular contract), the choice of the electrical characteristics of the interfaces is left to the designer, who shall optimise the use of the selected technology, and is expected to be familiar with spacecraft data-system interfacing.

### 11.2 PHYSICAL LAYER (SYMBOL STREAM) INTERFACE

#### 11.2.1 General

As specified in Section 5.2, each interface consists of three input lines:

- Symbol Clock signal
- Symbol Stream signal (NRZ-L)
- "Channel Active Indication" signal

#### 11.2.2 Symbol Clock Signal

The Symbol Clock signal shall have, nominally, the form of a square wave, with one period for each symbol. From the functional standpoint of the TC Decoder, this signal shall be considered valid only when the "Channel Active Indication" signal is ACTIVE.

#### 11.2.3 Symbol Stream Signal

The Symbol Stream signal shall be an NRZ-L signal. Its phase relationship with the Symbol Clock signal shall be such that one of the clock signal edges

(e.g. the falling edge) can be used to reliably clock the NRZ-L data in (data level stable at clock-in time).

Definition of the data states for the NRZ-L signal is not required: resolution of data state ambiguity is provided by the Coding Layer functions of the TC Decoder (more accurately: by the Start Sequence detection function).

#### **11.2.4 "Channel Active Indication" Signal**

The "Channel Active Indication" signal serves as an enable signal for as long as there is a Symbol Stream signal to clock in. There are no particular timing requirements for the signal: its change of state must be fast enough to be functional (the input signals are delivered by the radio-frequency and modulation subsystem).

#### **11.2.5 Maximum Symbol Rate**

For the maximum uplink symbol rate, see Section 3.3.

### **11.3 MAP INTERFACE**

#### **11.3.1 General**

The MAP interface provides the command data transfer link between, typically, the "basic" TC Decoder and the DMS. Each MAP is connected, in principle (i.e. when segmentation of TC Packets is effectively used), to a Packet Assembly Controller (PAC) dedicated to the MAP.

The data unit to be transferred is the TC Segment and, therefore, the data transfer is organised on a full TC Segment basis (maximum length: 249 octets). However, the MAP interface features a data-flow control mechanism that allows the serial flow of data to be halted and resumed on an OCTET basis, if required (for example: when the MAP interface is connected to a microprocessor system).

(This data-flow control mechanism is linked to the WAIT State of the FARM-1, which is entered whenever a TC Transfer Frame has been accepted and the previously received TC Segment has not yet been (fully) read out through the MAP interface.)

There are five signal lines:

- Clock Out (CKOUT) signal line (output)

- Data Set Ready (DSR) signal line (output)
- Data Terminal Ready (DTR) signal line (input)
- Segment Data (DATA) signal line (output)
- Aborted Data Transfer (ADT) signal line (output)

Figures 11.1 and 11.2 show two typical sets of signal waveforms:

- The first set illustrates a Segment Data transfer with no data-flow control (DTR line permanently set to ACTIVE level).
- The second set illustrates a Segment Data transfer with data-flow control (DTR line is dynamically activated by the data-receiving process).

Figure 11.3 shows an Aborted Data Transfer caused by the arrival of a BD Transfer Frame and the subsequent erasure of the TC Segment data being transferred.

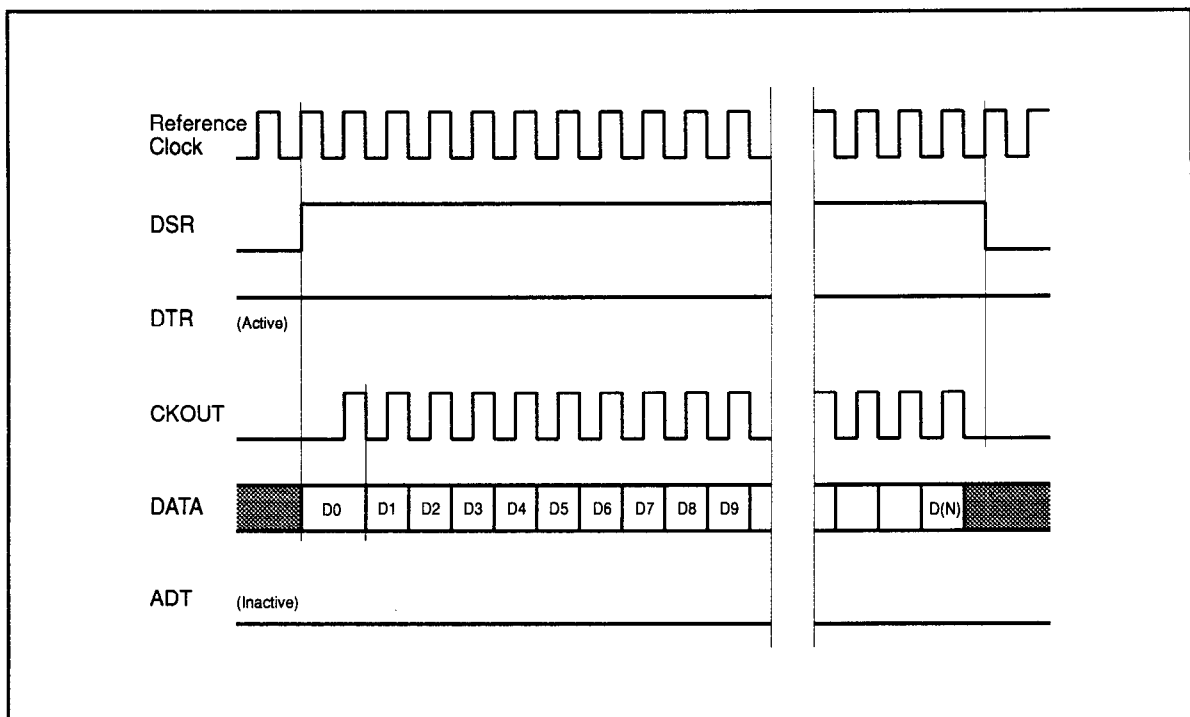


Figure 11.1 MAP INTERFACE WAVEFORMS: EXAMPLE OF DATA TRANSFER WITHOUT DATA-FLOW CONTROL

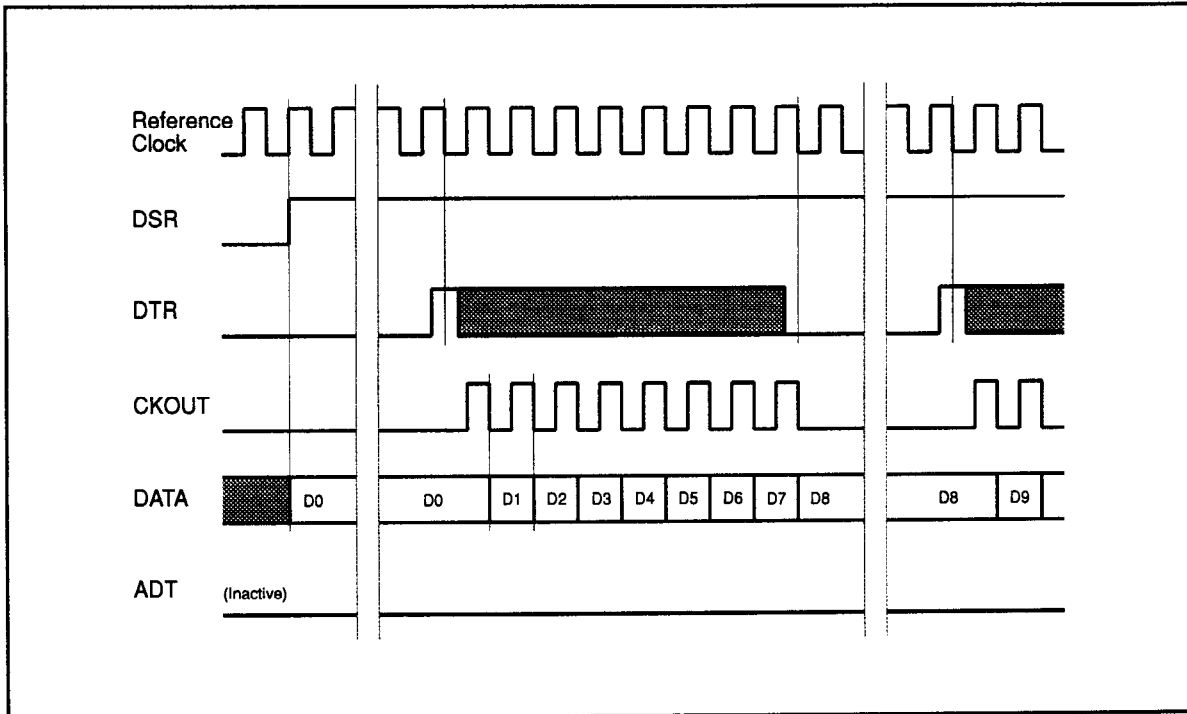


Figure 11.2 MAP INTERFACE WAVEFORMS: EXAMPLE OF DATA TRANSFER WITH DATA-FLOW CONTROL

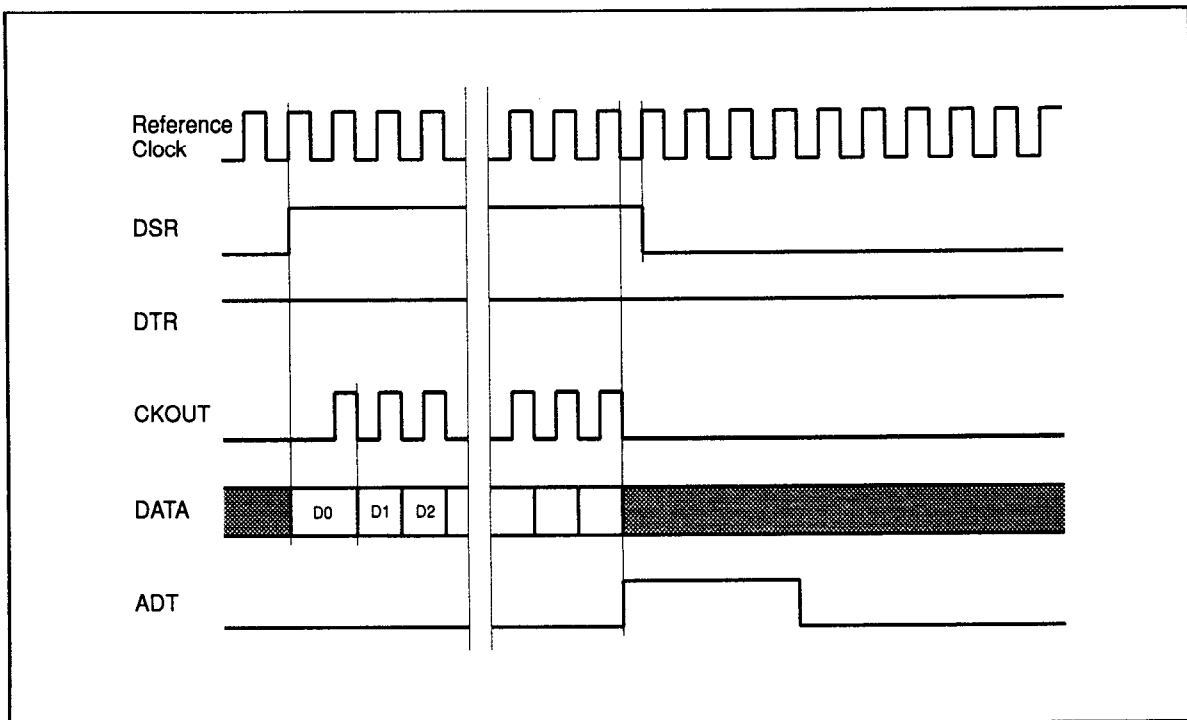


Figure 11.3 MAP INTERFACE WAVEFORMS: EXAMPLE OF AN ABORTED DATA TRANSFER (CAUSED BY BD FRAME ARRIVAL)

### 11.3.2 Clock Out (CKOUT)

The form of the Clock Out (CKOUT) signal is a square wave, with one period for each data bit. The CKOUT signal will start being issued when **both** the Data Set Ready (DSR) and Data Terminal Ready (DTR) are ACTIVE. Once the CKOUT signal has begun, it will last for eight clock periods, even if the DTR signal has become INACTIVE before the eighth CKOUT pulse.

If the DTR signal becomes INACTIVE after the  $n \times 8$ th CKOUT pulse (where  $n = 1, 2, \dots$ , up to 249), but before the  $(n + 1) \times 8$ th CKOUT pulse has occurred, the CKOUT signal starting with that  $(n + 1) \times 8$ th CKOUT pulse shall not be issued.

The implementer is free to systematically separate each "burst" of 8 CKOUT pulses by one (or more) extra clock periods if this can facilitate his interface design.

The exact times of occurrence of the CKOUT signal are discussed in detail in Subsection 11.3.4 on Data Terminal Ready (DTR).

The exact value for the CKOUT signal frequency shall be established by the designer, who may provide the possibility to select one of several frequencies if this capability is found advantageous (see also last paragraph of the next subsection).

A single value of 125 kHz is suggested to be appropriate for a TC Decoder optimised for a maximum bit rate of 4 kilobits/s on the uplink. A higher frequency would be required for TC Decoders operating at higher bit rates.

### 11.3.3 Data Set Ready (DSR)

The front (e.g. rising) edge of the DSR signal indicates that a TC Segment is available in the storage device foreseen to hold the Segment Data (this is the "back-end buffer" of Reference [2]). The DSR signal shall remain ACTIVE during the complete duration of the Segment Data transfer (maximum number of TC Segment octets: 249).

The front edge of the DSR signal shall be generated by one of the edges (e.g. rising) of the MAP reference clock signal (internal to the TC Decoder, and from which the CKOUT signal pulses are directly created).

The end of the Segment Data transfer is indicated by the trailing (e.g. falling) edge of the DSR signal. This trailing edge shall be generated by one of the

edges (e.g. rising) of that reference clock signal pulse occurring immediately after the last CKOUT pulse used to effectively transfer the last bit of Segment Data.

The exact value for the minimum possible delay between one DSR trailing edge and the next DSR front edge shall be established by the designer. The worst case consists of an interface where the DTR line has been permanently set to the ACTIVE state (i.e. the Wait Flag of the CLCW shall never be set "on" by data transfers through this interface). Such an interface shall be able to operate in all conditions of TC Transfer Frame transmission (e.g. a frame containing a one-octet TC Segment, followed by a frame containing a 249-octet TC Segment, and so on, at maximum bit rate) and MAP CKOUT signal frequency.

A value between 16 and 64 microseconds is suggested.

#### **11.3.4 Data Terminal Ready (DTR)**

The DTR signal indicates that the receiving device is ready to clock the Segment Data in.

No specific timing constraints are placed on the front (e.g. rising) edge of the DTR signal: this edge will simply occur after (a) the DSR signal has become ACTIVE and (b) the receiving device (the "Data Terminal") is ready to accept a TC Segment. When a TC Segment is signalled ready for transfer on a particular MAP interface (i.e. when the DSR line signal is ACTIVE), it will not be output as long as the DTR signal remains INACTIVE.

The major effect of the DTR signal is to delay the transfer of the TC Segment. Therefore, when another AD Transfer Frame is received by the TC Decoder, there will be no "back-end" buffer available for that frame; the FARM-1 will go to the WAIT State (S2) and the "Wait" Flag will be set to 1 in the CLCW; the FOP-1 process on ground will wait until the "Wait" Flag is reset to 0, i.e. until the TC Segment has finally be fully read out by the receiving device on board the spacecraft.

After the DTR signal has become ACTIVE, the first data-clocking (e.g. rising) edge of the CKOUT signal shall occur no sooner than one half of a CKOUT period after the front (e.g. rising) edge of the DTR signal, and no later than two CKOUT periods after.

The trailing edge (e.g. falling) edge of the DTR signal shall typically occur only after the entire TC segment has been transferred, that is, after the DSR



signal has become INACTIVE and before it becomes ACTIVE again (see minimum delay specification between DSR signals).

However, the trailing (e.g. falling) edge of the DTR signal may also occur before the DSR signal has become INACTIVE, when the receiving-end device is using the DTR line for flow control (i.e. to indicate that it is (temporarily) not able to store the data being read out). In response to the DTR signal becoming INACTIVE, the TC Decoder will stop issuing the CKOUT signal (as well as the corresponding DATA signal) corresponding to the OCTET BOUNDARY immediately placed after that DTR trailing edge.

When the DTR signal becomes ACTIVE again, the transfer of the next data octet(s) shall be resumed until the full TC Segment has been read out.

### **11.3.5 Segment Data (DATA)**

The DATA signal consists of the data bits, in NRZ-L form, that have been clocked out by the CKOUT signal. DATA are clocked out by one of the edges (e.g. falling) of the CKOUT pulses. No DATA transition shall be delayed by more than an 8th of a clock period, when reference is taken to the data-clocking edge of the CKOUT pulse.

### **11.3.6 Abort Data Transfer (ADT)**

The ADT signal indicates that the TC Decoder has aborted the transfer of a TC Segment:

- that had been placed in its "back-end" buffer, and
- the arrival of which had been signalled to the receiving device connected to the MAP interface by means of the DSR signal.

After the DSR signal has become ACTIVE, there can be instances when a BD Transfer Frame is received and accepted by the TC Decoder, and the DSR-signalled data transfer:

- has already begun, but has not yet been completed – or
- has not yet begun.

In both cases, the TC Segment stored in the "back-end" buffer shall be ERASED, and:

- the ADT signal shall be generated (The exact value for the duration of the ADT signal shall be established by the designer. A value between 8 and 32 microseconds is suggested.);
- the DSR signal shall become INACTIVE, its trailing (e.g. falling) edge being generated by one of the edges (e.g. falling) of the first CKOUT pulse occurring no sooner than one full CKOUT period after the occurrence of the ADT signal, and not later than the end of the same ADT signal.

## 11.4 TELEMETRY INTERFACE

### 11.4.1 General

The telemetry interface described in this section is used to provide the various telemetry data links between the TC Decoder and the telemetry subsystem.

The function of this interface is to transfer, in serial NRZ-L form, a 16-bit data element from the data source to the telemetry subsystem. (In older ESA documents, the same interface is specified under the name: "Serial 16-Bit Digital Channel". This specification is a compatible generalisation of that interface.)

There are three signal lines (as seen from the TC Decoder):

- Sampling (SAMPLING) signal line (input)
- Clock In (CKIN) signal line (input)
- Data (DATA) signal line (output)

Figure 11.4 shows how the waveforms of the three lines are related.

The interfaces required for each telemetry report (as specified in Section 10 on Telemetry Reporting) shall be provided as follows:

#### (a) CLCW Status Report

One complete, separate interface shall be dedicated to the CLCW Status Report of each TC Decoder. Each interface shall be redundant. Redundancy shall be implemented on-chip (rather than outside, where redundant circuits are more difficult to implement).

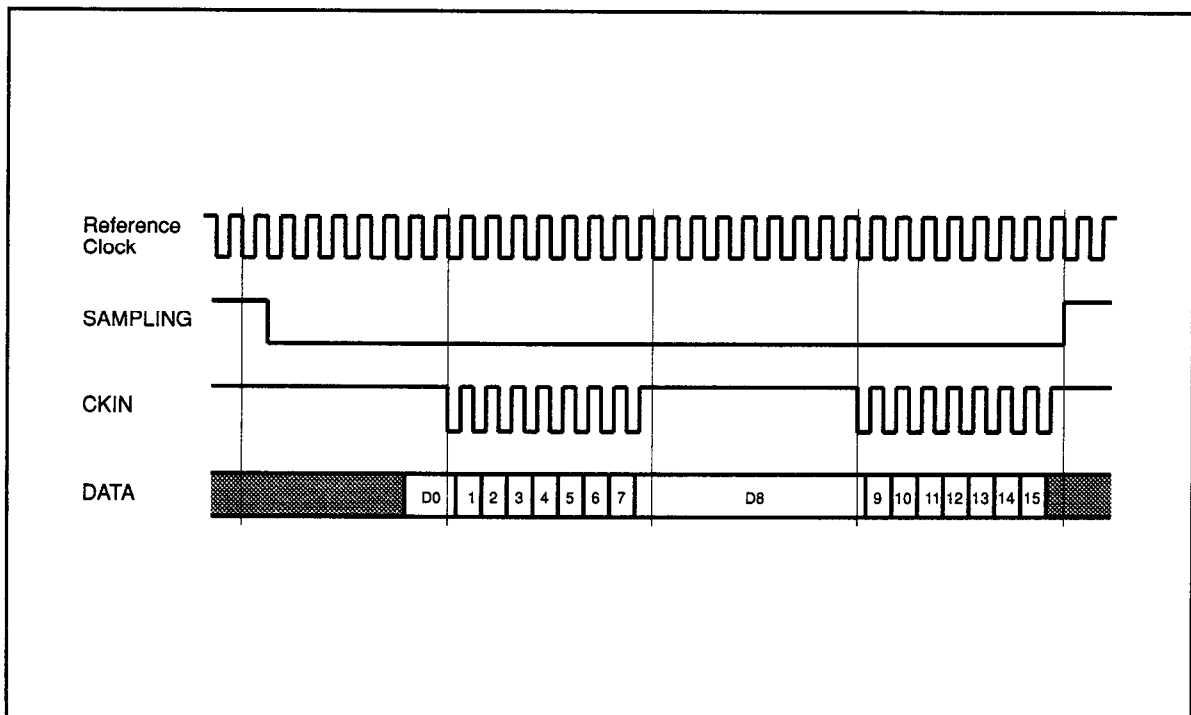


Figure 11.4 TELEMETRY INTERFACE WAVEFORMS

**(b) CPDU Status Report, Frame Analysis Report and AU Status Report**

The implementer is allowed to implement three separate sets of interfaces (one for each report) as follows:

- one complete interface for the CPDU Status Report;
- one complete interface for the FAR, with two SAMPLING lines instead of one (one for the first 16 bits, the other for the last 16 bits), but common DATA and CKIN lines;
- one complete interface for the AU Status Report, with two SAMPLING lines instead of one (one for the first 16 bits, the other for the remaining 64 bits), but common DATA and CKIN lines.

However, the implementer is allowed to provide instead a more economical approach in terms of connecting lines. A suggested solution is the following:

- one complete interface for the three reports (128 bits in total), with two SAMPLING lines (one for the first 16 bits, the other for the remaining 112 bits) and common DATA and CKIN lines. The first 16 bits shall be those of the CPDU Status Report, to be acquired by signalling on the first SAMPLING line. The next 32 bits shall be those of the FAR, to be acquired by signalling on the second SAMPLING line (2 x 16 bits). The last 80 bits shall be those of the AU Status Report, to be acquired by more signalling (5 x 16 bits) on the second SAMPLING line.

When the AU is not used for the mission, it shall be possible to read out only the 48 bits making up the CPDU Status Report and the FAR.

Redundancy of these interfaces shall also be considered, on-chip implementation being recommended

#### 11.4.2 Sampling (SAMPLING)

The SAMPLING signal defines the time during which the transfer of the 16 bits of DATA takes place.

The SAMPLING signal shall last 31 periods of the telemetry subsystem internal clock (given for reference on Figure 11.4).

The front edge (e.g. a falling edge) indicates that the transfer has begun, and that the first bit (D0) of DATA can be read out by the first falling edge of the CKIN signal (see next subsection on CKIN).

The trailing edge (e.g. a falling edge) indicates that the transfer of the entire 16 bits is over. This edge is generated half a clock period after the last bit (D15) of DATA has been read out.

#### 11.4.3 Clock In (CKIN)

The CKIN signal is only issued when the SAMPLING signal is ACTIVE (e.g. when the signal is low, as in Figure 11.4). The CKIN signal is essentially made up of two successive "bursts" of 8 clock pulses, the clock period in each 8-pulse burst being that of the internal reference clock:

- The time between the front edge of the SAMPLING signal and that at which Bit 0 is read out by the first CKIN pulse amounts to 7 reference clock periods.

- The time between the front edge of the SAMPLING signal and that at which Bit 8 is read out by the ninth CKIN pulse (i.e. the first pulse of the second burst) amounts to 23 reference clock periods.

The nominal reference clock signal frequency is 250 kHz, but the TC Decoder should be able to operate at higher frequencies (up to 1 MHz).

#### 11.4.4 Data (DATA)

The DATA signal consists of one element of 16 data bits, in NRZ-L form, each bit having been clocked out by the CKIN signal. DATA are clocked out on the front edge (e.g. a falling edge) of the CKIN pulses. No DATA transition shall be delayed by more than an 8th of a clock period, when reference is taken to the data-clocking edge of the CKIN pulse.

### 11.5 COMMAND PULSE OUTPUTS

#### 11.5.1 Terminology

The name "Command Pulse" is recommended as more accurate – and less misleading – than "On/Off Command".

The name "On/Off Command" originates from the early time of "tone telecommanding", when the transmission of a single tone (usually an unmodulated, low-frequency subcarrier) was used to TOGGLE one single device. Typically, the device was a relay to be switched "ON", and "OFF", and "ON" again, etc.

The key words in this early, simple telecommand system were:

- COMMAND, TOGGLE, ON, OFF.

The concept of toggling electrical devices on board a spacecraft by telecommand was **banned** from the very beginning by NASA and ESA (then called ESRO). Unfortunately, the old – and completely inappropriate – name "On/Off" remained in use, at least at ESA. There is nothing more inaccurate and misleading than to call "On/Off" a signal that is fundamentally linked to a LATCHING concept. This concept, which is one of the basic requirements for the operation of ESA spacecraft, makes it mandatory to provide one Command Pulse output to (for example) switch "ON" one device, and another Command Pulse output to switch "OFF" the same device.

The key words in the ESA telecommand system are:

- COMMAND, LATCHING, PULSE.

### **11.5.2 Specification**

The specification of the actual Command Pulse outputs used to drive the various spacecraft latching devices (e.g. electromechanical relays) is outside the scope of this document.

The "basic" TC Decoder CPDU circuitry shall, however, provide the signals that will allow the necessary drivers to equip the final flight units. These signals are specified in Section 9 as regards their function, durations and delay between consecutive pulses.

The waveform is that of a pulse, the ACTIVE state corresponding to the width of the pulse.

## **APPENDIX A**

### **TELECOMMAND SUBSYSTEM CONFIGURATION ON BOARD AN ESA SPACECRAFT**

Figure A-1 illustrates a typical configuration of an ESA telecommand subsystem on board an ESA spacecraft.

#### **NOTE**

The configuration shown is only an example provided to help the readers of this document in their appreciation of the functional capabilities of the TC Decoder. In particular, the assignment of the various MAP Identifiers is only indicative, and can vary from one spacecraft to another, if so required.

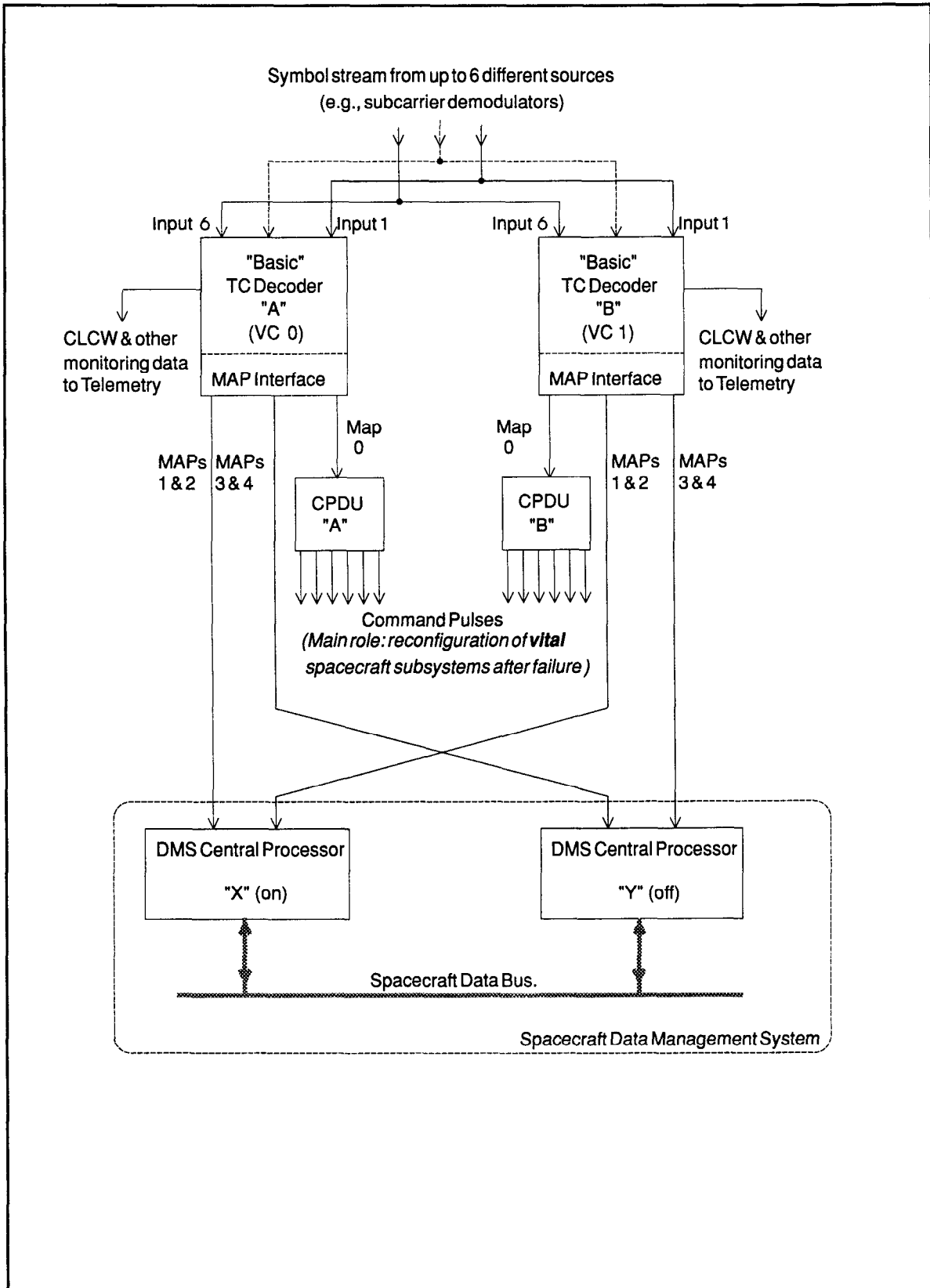


Figure A.1 EXAMPLE OF A TELECOMMAND SUBSYSTEM CONFIGURATION



## **APPENDIX B**

### **VERIFICATION OF COMPLIANCE**

#### **B.1 CONNECTION WITH SENDING-END REFERENCE SYSTEM**

When a TC Decoder is designed, it is expected that the implementer will build a number of simple testing devices which, for the purpose of this Appendix, are grouped under the generic name TEST JIGS.

Test Jigs are required at the early stage of any development. They are easy to operate and economical. They are, however, not sufficient to perform the full functional qualification of a flight unit. The equipment required for this task is, for the purpose of this Appendix, called the (TC DECODER) TEST EQUIPMENT.

The Test Equipment implements the full telecommand sending-end system, from the generation of TC Packets down to the CLTUs and Physical Layer Operational Procedure (PLOP). Also included are the means needed to provide the various telemetry reports (and, chief among them, the telemetry Transfer Frame CLCW).

The implementer is informed that ESA has specifically developed such a system for testing the compliance of TC Decoders. The system was largely implemented as an IBM Personal Computer software (written in C-language) under ESA contract. This software can be made available to European industry.

At the time of issue of this specification, the software is considered as the ESA Telecommand Reference by ESOC for the implementation of its Ground Station systems. It has all the capabilities specified in Reference [2], except for:

- the Suspend/Resume states of the FOP-1 (these states were defined at a very late stage by CCSDS, to satisfy very specific deep-space requirements);
- the Authentication System software.

In view of the complexity of the new ESA (CCSDS) telecommand system, and in order to maintain a high order of compliance, ESA recommends that implementers of TC Decoders use the same Telecommand Reference software for developing their Test Equipment.

Further information can be obtained from:

- at ESTEC, the Head of the On-board Information Management Section (mail code: WDI; current Head is Mr J. SCHMITT) of the On-Board Data Division (mail code: WD) of ESTEC, Noordwijk, The Netherlands.
- at industrial source, "I.B. & M.A. DE LANDE LONG, SOFTWARE & CONSULTANCY", Im Weingarten 13, D-6104 Seeheim, Germany.  
Telephone: + 49 6151 54926  
Telefax : + 49 6151 595701

## **B.2 ENCODED TELECOMMAND SEQUENCES FOR TEST JIGS**

The telecommand sequences defined in this section are provided to the designers of TC Decoders as samples for early testing, so that they may begin to verify the correct operation of their Test Jig tools.

All data are given in hexadecimal notation. Bit 0 of each data field is expressed in the left-most hexadecimal character.

**B.2.1 MISSION-SPECIFIC DATA**

Spacecraft ID: 123  
 Virtual channel ID: 12  
 CPDU Map address: 00  
 CPDU application ID: 456  
 MAP ID Pointer: 00

Authentication Fixed Key:

(Starting from **00**, the value of each octet is that of the preceding one, incremented by 1.)

W0: 00 01 02 03 04 05,  
 W1: 06 07 08 09 0A 0B,  
 W2: 0C 0D 0E 0F 10 11, .. .. .,  
 W42: FC FD FE FF 00 01,  
 W43: 02 03 04 05 06 07, .. .. .,  
 W59: 62 63 64 65 66 67

Hashing feedback coefficients:

C0...C59: A AA AA AA AA AA AA AA

**B.2.2 COLD START VALUES**

CLCW = **2000**  
 RF Available, Bit Lock, Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 0, N(R) = 00

FAR = **00007FE0**  
 New report, Abandoned CLTU, No illegal report,  
 0 Codeblocks, 0 bit corrected, No LEGAL frame,  
 Input 7, Last MAP: 63, No AU Report

AUSR = **3FFFFFFF7FFFFFFF00FF**  
 Primary LAC Count = **3FFFFFFF**,  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **FF**  
 Fixed key

CPDUSR = **3FFF**  
 Cold start

**B.2.3 CLTU 1: BC, UNLOCK**

55 EB 90  
31 23 48 07 00 00 EC 0E  
95 55 55 55 55 55 55 70  
55 55 55 55 55 55 55 55

**CLCW = 0200**  
RF Available, Bit Lock, Not Lockout, Not Wait,  
Not Retransmit, FARM-B Count = 1, N(R) = 00

**FAR = 7010C7E0**  
New report, Frame accepted, No illegal report,  
2 Codeblocks, 0 bit corrected, BC frame,  
Input 0, Last MAP: 63, No AU Report

**AUSR = 3FFFFFFF7FFFFFFF00FF**  
Primary LAC Count = **3FFFFFFF**,  
Auxiliary LAC Count = **3FFFFFFF**  
Recovery LAC Count = **FF**  
Fixed key

**CPDUSR = 3FFF**  
Cold start

**B.2.4 CLTU 2: BC, SET V(R) = FD**

55 EB 90  
31 23 48 09 00 82 00 DC  
FD 22 E3 55 55 55 55 62  
55 55 55 55 55 55 55 55

**CLCW = 04FD**  
RF Available, Bit Lock, Not Lockout, Not Wait,  
Not Retransmit, FARM-B Count = 2, N(R) = FD

**FAR = 7010C7E0**  
New report, Frame accepted, No illegal report,  
2 Codeblocks, 0 bit corrected, BC frame,  
Input 0, Last MAP: 63, No AU Report

**AUSR = 3FFFFFFF7FFFFFFF00FF**  
Primary LAC Count = **3FFFFFFF**,  
Auxiliary LAC Count = **3FFFFFFF**  
Recovery LAC Count = **FF**  
Fixed key

**CPDUSR = 3FFF**  
Cold start

### B.2.5 CLTU 3: BD, MAP 3F, AU COMMAND, RECOVERY LAC, LOAD FIXED KEY IN PROGRAMMABLE MEMORY

```

55 EB 90
21 23 48 11 00 FF 07 EC
BF FF FF FF 11 22 33 6A
44 55 59 1C 55 55 55 E4
55 55 55 55 55 55 55 55

```

CLCW = **06FD**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = FD

FAR = **701887E4**  
 New report, Frame accepted, No illegal report,  
 3 Codeblocks, 0 bit corrected, BD frame,  
 Input 0, Last MAP: 63, Authenticated AU command

AUSR = **3FFFFFFF7FFFFFFF0000**  
 Primary LAC Count = **3FFFFFFF**,  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **00**  
 Fixed key

CPDUSR = **3FFF**  
 Cold start

**B.2.6 CLTU 4: AD, MAP 3F, AU COMMAND, PRIMARY LAC, CHANGE PROGRAMMABLE KEY BLOCK A005**

```

55 EB 90
01 23 48 19 FD FF 0A 26
05 11 22 33 44 55 66 F4
77 3F FF FF FF 94 B2 EE
D0 EF 0D 23 97 55 55 5A
55 55 55 55 55 55 55 55

```

**CLCW = 06FE**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = FE

**FAR = 702007E4**  
 New report, Frame accepted, No illegal report,  
 4 Codeblocks, 0 bit corrected, AD frame,  
 Input 0, Last MAP: 63, Authenticated AU command

**AUSR = 000000007FFFFFFF0000**  
 Primary LAC Count = **00000000**,  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **00**  
 Fixed key

**CPDUSR = 3FFF**  
 Cold start

### B.2.7 CLTU 5: AD, MAP 3F, AU COMMAND, RECOVERY LAC, SELECT PROGRAMMABLE KEY

```

55 EB 90
01 23 48 11 FE FF 06 4E
BF FF FF 00 5E 80 85 38
C8 0B C0 6B 55 55 55 6A
55 55 55 55 55 55 55 55

```

CLCW = **06FF**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = FF

FAR = **701807E4**  
 New report, Frame accepted, No illegal report,  
 3 Codeblocks, 0 bit corrected, AD frame,  
 Input 0, Last MAP: 63, Authenticated AU command

AUSR = **000000007FFFFFFF8001**  
 Primary LAC Count = **00000000**,  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **01**  
 Programmable key

CPDUSR = **3FFF**  
 Cold start



**B.2.8 CLTU 6: AD, MAP 3F, AU COMMAND, PRIMARY LAC, CHANGE PROGRAMMABLE KEY BLOCK B000**

```
55 EB 90
01 23 48 19 FF FF 0B 08
00 88 99 AA BB CC DD 38
EE 00 00 00 00 3C 52 CC
68 7E 94 99 98 55 55 A0
55 55 55 55 55 55 55 55
```

**CLCW = 0600**  
RF Available, Bit Lock, Not Lockout, Not Wait, Not Retransmit,  
FARM-B Count = 3, N(R) = 00

**FAR = 702007E4**  
New report, Frame accepted, No illegal report, 4 Codeblocks,  
0 bit corrected, AD frame, Input 0, Last MAP: 63,  
Authenticated AU command

**AUSR = 000000017FFFFFFF8001**  
Primary LAC Count = **00000001**  
Auxiliary LAC Count = **3FFFFFFF**  
Recovery LAC Count = **01**  
Programmable key

**CPDUSR = 3FFF**  
Cold start

**B.2.9 CLTU 7: AD, MAP 3F, AU COMMAND, PRIMARY LAC,  
CHANGE PROGRAMMABLE KEY BLOCK B103**

```

55 EB 90
01 23 48 19 00 FF 0B 86
67 11 22 33 44 55 66 5C
77 00 00 00 01 12 2A B0
25 F0 CB C0 DA 55 55 CA
55 55 55 55 55 55 55 55

```

**CLCW = 0601**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = 01

**FAR = 702007E4**  
 New report, Frame accepted, No illegal report,  
 4 Codeblocks, 0 bit corrected, AD frame,  
 Input 0, Last MAP: 63, Authenticated AU command

**AUSR = 000000027FFFFFFF8001**  
 Primary LAC Count = **00000002**  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **01**  
 Programmable key

**CPDUSR = 3FFF**  
 Cold start

**B.2.10 CLTU 8: AD, MAP 01**

```

55 EB 90
01 23 48 17 01 C1 12 B0
34 56 78 9A BC DE F0 04
12 34 56 78 9A BC DE F8
F0 4A 3B 55 55 55 55 E0
55 55 55 55 55 55 55 55

```

**CLCW = 0602**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = 02

**FAR = 70200020**  
 New report, Frame accepted, No illegal report,  
 4 Codeblocks, 0 bit corrected, AD frame,  
 Input 0, Last MAP: 1, No AU Report

**AUSR = 000000027FFFFFFF8001**  
 Primary LAC Count = **00000002**  
 Auxiliary LAC Count = **3FFFFFFF**  
 Recovery LAC Count = **01**  
 Programmable key

**CPDUSR = 3FFF**  
 Cold start

**SEGMENT HEADER = C1**

**SEGMENT DATA = 12 34 56 78 9A BC DE F0 12 34 56 78 9A BC DE F0**

**B.2.11 CLTU 9: AD, MAP 00, AUTHENTICATED SEGMENT (WITH CPDU PACKET), AUXILIARY LAC**

```

55 EB 90
01 23 48 1E 02 C0 14 3C
56 F8 9A 00 07 00 00 58
01 F1 02 0F 00 54 7F 9A
FF FF FF F3 D3 1C EA F0
C9 C0 3D 55 55 55 55 74
55 55 55 55 55 55 55 55
55 55 55 55 55 55 55 55

```

**CLCW = 0603**  
 RF Available, Bit Lock, Not Lockout, Not Wait,  
 Not Retransmit, FARM-B Count = 3, N(R) = 03

**FAR = 70280002**  
 New report, Frame accepted, No illegal report,  
 5 Codeblocks, 0 bit corrected, AD frame,  
 Input 0, Last MAP: 0, Authenticated segment

**AUSR = 00000002400000008001**  
 Primary LAC Count = **00000002**,  
 Auxiliary LAC Count = **00000000**  
 Recovery LAC Count = **01**  
 Programmable key

**CPDUSR = 789A**  
 Packet LEGAL, Last Packet Sequence Count = **389A**

**SEGMENT HEADER = C0**

**SEGMENT DATA = 14 56 F8 9A 00 07 00 00 01 F1 02 0F 00 54**

**CPDU COMMANDS =** 10 ms pulse on output 00,  
 20 ms pulse on output 01,  
 1280 ms pulse on output 02

## APPENDIX C

### DIFFERENCES WITH EARLY IMPLEMENTATIONS

#### C.1 INTRODUCTION

The forerunner of this Specification was a development contract document called: "Specification for a Standard ESA/CCSDS Telecommand Decoder", Reference: THB/CS/2077/av. The first issue of this early specification was produced in 1986, at the time of placing two identical contracts:

- one contract with ALENIA SPAZIO (called SELENIA SPAZIO at that time), of Italy;
- one contract with SAAB SPACE, of Sweden.

The clarity of the first specification suffered from the absence of Reference [2] (Issue 1 of the PACKET TELECOMMAND STANDARD was only published at the beginning of 1991). As work progressed on both TC Decoders, as well as on Reference [2], the specification was updated several times, the last issue being produced in May 1988.

A notably difficult task was to ensure that both products would be functionally identical, especially at a time when the elements of an ESA Telecommand Reference system (see Appendix B, on Verification of Compliance) were not yet available.

In spite of all these difficulties, the two designs are remarkably close to each other from a functional standpoint. They also largely comply with this Specification except for a few aspects. Most of these aspects are in fact "upgrades", and are presented in the following sections:

- Section C.2 recapitulates on the "upgrades" introduced in this Specification.
- Sections C.3 and C.4 present and discuss the few minor items of non-compliance between this Specification and the TC Decoders of ALENIA SPAZIO and SAAB SPACE, respectively.

## **C.2 LIST OF UPGRADES**

### **C.2.1 CODING LAYER**

In the early specification, the definition of the "first and last data transfer in the DECODE state", as specified in Subsection 5.1.3 of this document, was not clearly expressed. (This lack of clarity is also repeated in Reference [2] - hence the text of Subsection 5.1.3.) It eventually became apparent that both ALENIA SPAZIO and SAAB SPACE had interpreted the early text in the same different way, as follows:

"When an Event 2 - (E2): CHANNEL DEACTIVATION - occurs, the decoder returns to the INACTIVE state (S1), with the resulting actions:

- the Codeblock is abandoned (erased), and
- no information octets from that Codeblock are transferred to the layer above, and
- all information octets previously transferred to the layer above as "Candidate Frame" are erased, and the CLTU is considered to be ABANDONED."

The consequences of this slightly different mode of operation (referred to as the "Erasure mode" hereafter, for convenience, as opposed to the "Candidate mode" specified in Subsection 5.1.3) are extremely minor, and can be characterised as follows:

- The performance in terms of PFR (Probability of Frame Rejection) and PFU (Probability of Frame with Undetected error) is only affected by E2 events that may occur at the time when the Tail Sequence of the CLTUs are being decoded. The change in performance is insignificant and negligible.
- As far as compliance with ground systems is concerned, this type of behaviour is not significant, and its verification is normally not foreseen (nor easy to perform, as it would require a "channel activation" switch to be commanded "open" at the time when the Tail Sequence is transmitted).

The "Candidate mode" of operation, as specified in Subsection 5.1.3, conforms to the CCSDS Recommendations. It is in line with the layering concept of the CCSDS Telecommand System on the one hand, and simpler to design and implement on the other hand.

### **C.2.2 TRANSFER LAYER**

The early specification defined a fixed, non-symmetrical FARM-1 Sliding Window with only one width value:

- $W = 128$
- $PW = 120$
- $NW = 8$

At that time, further work on Reference [2] resulted in a better understanding of the COP-1 Sliding Windows and their mechanisms. The FARM-1 Sliding Window is now specified to be fixed for the mission, but with any possible value, as required for the mission. Reference [2] defines  $PW$  to be equal to  $NW$ , except for some rare cases (e.g. deep-space operations). Note that the early, fixed and non-symmetrical FARM-1 Sliding Window of value 128 can be safely considered as a symmetrical window of width  $W = 16$ , as long as it is operated with a FOP-1 Sliding Window of maximum value  $K = 8$ .

### **C.2.3 AUTHENTICATION LAYER**

The early specification did not require or define the following features:

- (a) In-flight programmability of the 60 coefficients of the Hashing Function. (It was found that the new feature greatly improves the security of the system).
- (b) Exact organisation of the Knapsack weights (2880 bits in total, making up, at the time, the Programmable Key) in the Programmable Key memory.
- (c) Minimum length of an Authenticated TC Segment (10 octets).
- (d) Authenticated MAP Identifier Pointer (in the early specification, all TC Segments were either authenticated or not).

#### **C.2.4 CPDU**

This Specification completes the functional requirements of the CPDU. In particular:

- (a) The process of the TC Segment is clearly distinct from that of the CPDU Packet.
- (b) All fixed values (Version Number, Type, Data Field Header Flag, Packet Sequence Flags) of the CPDU Packet Header must now be verified (they were previously to be ignored).
- (c) Once a CPDU Packet has been accepted as "LEGAL" by the CPDU, the ensuing process cannot be stopped by the arrival of new TC Segments, and the CPDU must execute the command instructions contained in the packet.

#### **C.2.5 TELEMETRY REPORTING**

The formats of all telemetry reports , as defined in the early specification, are essentially identical to those found in this document. Except for some differences of purely editorial nature, the only significant changes are:

##### **(a) AU Status Report**

- Bits 0, 1, 32 and 33 of the report are now clearly defined to be permanently set to a value to be defined by the implementer, as they are not required for the identification of the LAC Count value in the report.

##### **(b) Frame Analysis Report**

- Bit 0 is now identified as "Cold Start" value.
- The "Cold Start" value for the "Last MAP Addressed" is now to be defined by the implementer (suggested value: **111111**).
- The **111** value of Bits 28, 29 and 30 is now used: it indicates that the format (e.g. length) of a TC Segment was found to be wrong prior to being processed for authentication.



### **C.2.6 INTERFACES**

From a purely functional standpoint, the specification of interfaces has not changed, with the exception of the fail-safe system required to equip the Physical Layer interface. This requirement did not appear in the early specification, but was identified by the designers of SAAB SPACE and ALENIA SPAZIO early during the development activity, and a timer-based circuit implemented on both TC Decoders.

### **C.3 ALENIA SPAZIO**

The first TC Decoder VLSI chips delivered by ALENIA SPAZIO only implemented the "basic" TC Decoder (two VLSI chips) and the AU (one VLSI chip). The CPDU was only provided as an MSI implementation, with a very limited capability.

Since then, ALENIA SPAZIO decided to change the technology of their chip-based TC Decoder (to meet more severe radiation requirements, in particular). At the time of that decision, ALENIA SPAZIO was also participating in the reviews of the first draft issues of this Specification. As a result, their "basic" TC Decoder is now available as one single chip and features most of the upgrades listed in Section C.2.

The ALENIA SPAZIO TC Decoder does not comply with this Specification in the following ways:

#### **(a) Coding Layer**

The "Erasure mode" of operation - described above in C.2.1 - is used, rather than the "Candidate mode". ALENIA SPAZIO requested permission not to change their design in this particular instance, for obvious reasons of cost and quality assurance. This was agreed by ESA, and waived as compliant.

#### **(b) Authentication Layer**

At the time of issue of this document, ALENIA SPAZIO has not yet upgraded their AU VLSI chip with the radiation-hard technology already used for the "basic" TC Decoder. When this is done, all new features are planned to be added (they are missing in the current VLSI chip).

### (c) Telemetry Reporting

The "Last MAP Addressed" report of the FAR is correctly set to **111111** at "Cold Start" time. However, acceptance of a BC frame by the FARM will change that report to **000000**, which it should not. (Note that subsequent acceptance of BC frames will not change the value of the "Last MAP Addressed" report.) Furthermore, when AU Control Commands are accepted by the AU, their MAP ID (**111111**) is not reported (no change of "Last MAP Addressed" value).

The "Abandoned CLTU" report (**000** value for Bits 1,2 and 3) of the FAR indicates more than is specified in this document: it also signals that the "Candidate Frame" data passed to the Transfer Layer has been erased because it only contained 7 octets. The correct report for this event, as specified in this document, is "Frame declared DIRTY" (**001** value).

(This detail on classification was only clarified shortly before this Specification was issued.)

The **111** value for Bits 28, 29 and 30 of the FAR is not implemented. This report value was "Not allocated" in the early specification. (The new definition was added shortly before this Specification was issued.) However, the event itself ("rejection due to wrong format of TC Segment prior to being authenticated") is implemented.

The FAR is only a Survey Data Report (as opposed to a Status Data Report like the CLCW Status Report). The differences reported above are minor and have been waived by ESA.

## C.4 SAAB SPACE

The TC Decoder VLSI chip delivered by SAAB SPACE implements all the specified functions (i.e. "basic" TC Decoder, AU and CPDU) in one single ASIC. It does not feature any of the "upgrades" listed in Section C.2. (At the time of issue of this Specification, SAAB SPACE has made known its intention to perform a re-design of its VLSI chip.)

In addition to the missing "upgrades", the SAAB SPACE TC Decoder was found not to comply with this Specification in the following ways:

**(a) Coding Layer**

Event E10 (Buffer Release Signal) of the FARM-1 is not correctly implemented. The event is not self-standing, but is conditioned by the arrival of a valid frame. Thus, when the Wait Flag of the CLCW is set "on" (to **1**), effective release of the "back-end" buffer is not sufficient to set the flag back to **0**. For this to happen, a TC Transfer Frame must be transmitted and accepted by the FARM. Since the sending-end process is effectively "waiting" (FOP-1 in State 3), no such frame can be transmitted, and the entire system has actually come to a standstill.

NOTE: It is possible to operate such a TC Decoder without modifying the ground system if the Wait Flag is never used, i.e. when there are no active DTR lines on the MAP interfaces.

**(b) Authentication Layer**

The AU rejects authenticated TC Segments when they are shorter than 9 octets (instead of 10). This is a very minor difference.

The procedure for loading the Programmable Key is slightly different: the 5 octets of the "pseudo-signature" are loaded in the opposite order in the Programmable Key memory.

(These particular points of the Authentication Layer specification were not defined in the early specification.)

**(c) Telemetry Reporting**

The **111** value for Bits 28, 29 and 30 of the FAR is not implemented.

The order in which the fields of the AU Status Report have been arranged is not as specified in this document.

**PAGE INTENTIONALLY LEFT BLANK**

## APPENDIX D

### GLOSSARY OF ACRONYMS

This glossary supplements (and sometimes completes) the Glossary of Acronyms provided in Appendix A of Reference [2]. Therefore, only those acronyms not listed in Appendix A of Reference [2] are entered here.

ADT:	Abort Data Transfer
AOS:	Advanced Orbiting Systems
ASIC:	Application-Specific Integrated Circuit
AU:	Authentication Unit
BR:	Buffer (Shift) Register
CKIN:	Clock In
CKOUT:	Clock Out
CPDU:	Command Pulse Distribution Unit
DMS:	Data Management System
DSR:	Data Signal Ready
DTR:	Data Terminal Ready
EOD:	Even-Odd Detector
FAR:	Frame Analysis Report
LAC:	Logical Authentication Channel
LSB:	Least Significant Bit
LFSR:	Linear Feedback Shift Register
MSB:	Most Significant Bit
MSI:	Medium-Scale Integration
NRZ-L:	Non-Return-to-Zero/Level
PAC:	Packet Assembly Controller
PCM:	Pulse Code Modulation
PFR:	Probability of Frame Rejection
PFU:	Probability of Frame with Undetected error
PLR:	Position Location (Shift) Register
SEC:	Single-Error Correction AND Double-Error Detection (SEC is short for SEC & DED)
SR:	Shift Register
TED:	Triple-Error Detection
VLSI:	Very-Large-Scale Integration

**PAGE INTENTIONALLY LEFT BLANK**

**\*\*\* E N D OF DOCUMENT \*\*\***