SPIRE-ALC-DOC-002359

HERSCHEL / PLANCK

Time Adjustment
H-P-1-ASPI-TN-0153

Product Code: 000000

Written by	*Responsibility-Office -Company*	Date	Signature
Rachid YAKOUBI	Data Handling – ASPI	30.11.2001	
Verified by			
Joël MORISSE	Data Handling Group Manager – ASPI	30.11.2001	
Patrice COUZIN	Electrical Architect – ASPI	30.11.01	
Pascal RIDEAU	Engineering Manager – ASPI	30.11.2001	
Approved			
Jean-Jacques JUILLET	Project Manager – ASPI	30.11.2001	

Entité Emettrice : **Alcatel Space - Cannes**
(détentrice de l'original) :

HERSCHEL/PLANCK	DISTRIBUTION RECORD
DOCUMENT NUMBER : H-P-1-ASPI-TN-0153	Issue 1 Rev. 0 Date : 28 November 2001

EXTERNAL DISTRIBUTION		INTERNAL DISTRIBUTION	
ESA	X	HP team	X
ASTRIUM			
ALENIA	X		
CONTRAVES			
TICRA			
TECNOLOGICA			
		Clt Documentation	Orig.

ENREGISTREMENT DES EVOLUTIONS / *CHANGE RECORDS*

ISSUE	DATE	§ : DESCRIPTION DES EVOLUTIONS *§ : CHANGE RECORD*	REDACTEUR *AUTHOR*
1	28 November 2001	First Issue	Rachid YAKOUBI

TABLE OF CONTENTS

1. SCOPE......5

2. APPLICABLE DOCUMENTS......5

3. INTRODUCTION......6

4. DEFINITIONS......8

5. LIST OF CONTRIBUTIONS......9

5.1 OSCILLATORS FEATURES : 9
5.2 SAVE PHASE......9
5.2.1 SW Command......9
5.2.2 HW Command......11
5.3 READ PHASE......12
5.3.1 The user is an ASW application......12
5.3.2 The user is a unit connected on a data bus......13

6. TRANSMISSION TO GROUND......17

7. UPDATE PROCEDURE......18

7.1 TIME CONTROL PHASE......18
7.1.1 Save phase......18
7.1.2 Read phase......18
7.1.3 Comparison phase......18
7.2 LOAD PHASE......19
7.3 LATCH PHASE......19
7.4 PROCESSING......19

8. BUDGETS......21

1. Scope

This note identifies the main contributions that result in an inaccuracy of the on board timing function. Then, for each contribution, a performance specification is intended to be declined in order to comply with the overall function performance.

2. Applicable documents

- "System Requirements Specification" **SCI-PT-RS-05991**
- "Packet Structure - Interface Control Document" **SCI-PT-ICD-7527**
- "Operation Interface Requirements Document" **SCI-PT-RS-07360**
- "Packet Telemetry Standard" **ESA-PSS-04-106 Issue 1 January 1988**

3. Introduction

The On Board Time aims at dating spacecraft events with a certain degree of accuracy. This requirement leads logically to have an autonomous "timekeeper" (the OBT function) that is capable of distributing a date to spacecraft users and being periodically calibrated or updated according to information gathered from more accurate sources (ground control, ...).

It follows that the accuracy problems then breaks down into three categories :

- the performance of the autonomous timekeeper itself (stability,…),
- the distribution of the date to the users and
- the calibration method.

The sources of the errors for the second and third points can come from both hardware and software. Hardware sources are generally deterministic (the delays and jitter depend on the number of gates a signal passes through) and insignificant (e.g. a 74AC00 gate delay is between 5 and 20 ns with jitter well under 1 ns), whereas software sources are generally larger and more difficult to treat as the processing delays can vary.

To study all of these points, we will assume an elementary OBT function, representative of OBT function implemented within the Herschel-Planck CDMU, with only a counter driven by a clock signal from an oscillator. The length of the counter and the time duration to count define the resolution of the time value. This elementary OBT function provides a Central Time Reference for the different on board users.

The counter is a free running counter. That is to say one must not stop it. However, it shall be possible to update it with a specific value and to read its content. For that, the counter is associated with Load and Save registers respectively. A write to/read from the counter is done on Load and Save commands respectively.

The following symbolic representation may be used as a visual support for this note :

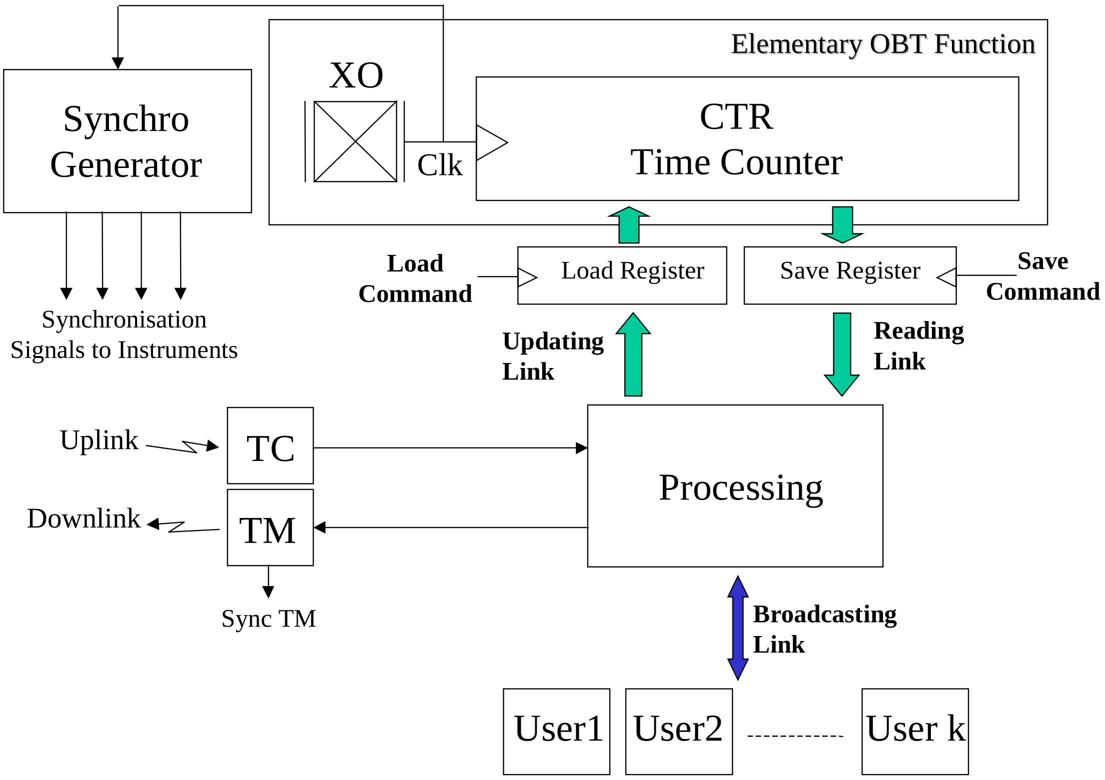

Figure 1 : General overview

This note points out all contributions that can lead to a shift between the local OBT and the time as recovered by the user, and the shift between the OBT and the reference ground time.

Not all these contributions are involved in all processing phases of the on board time (e.g. update, time broadcast…). Moreover, a contribution will be an error in some cases when it will be a delay in an other.

4. Definitions

a) Contribution

A contribution must be understood as a worst case value. That is to say a contribution corresponds to the maximum time duration between the beginning of an action and its end. It includes all design causes such as (propagation delay, processing, temperature sensitivity, jitter…).

b) Error/Delay

A contribution that results in a drift between the instant of an event and the effective taking into account of this event will be named an **error** otherwise it will be named a **delay**. Moreover, an error could be from any deterministic causes (e.g. propagation delay, processing time of a Low level software service,…) or from any random causes (e.g. jitter). It must be noted that a deterministic contribution could be taken into account by the application software or by ground to provide a time sample with better accuracy.

c) Jitter

Jitter is the difference, in time, between the expected temporal position of a pulse (or edge) and its real position.

A typical value for jitter contributions on a serial data bus is less than 1 µs.

5. List of contributions

5.1 OSCILLATORS FEATURES :

A first contribution concerns the oscillator performances in term of period (or frequency) **stability** generally expressed in ppm unit over a short, medium and long time period. A typical stability value for an oscillator with no specific constraint is 10^{-6} ppm at the best. That is to say, if one considers the drift over one second it will result in an error of 1 µs. In fact, the drift of the oscillator can be regularly compensated by an update procedure, from ground, of the CTR.

NB : A confusion is generally done between the stability and the **accuracy**. The accuracy is the difference between the knowledge of the local On Board Time and a reference time (e.g. ground time) when the term stability defines the period variation over an observation window (short term, medium term and long term). The stability can be poor but the accuracy very good if the non stability of the local OBT is well modelled, and the other error contributors are minimum.

5.2 SAVE PHASE

This *Save phase* is dedicated to the generation of a Save command that consists in loading the current content of the counter in a save register. The save command may be generated either on software request (it will be called SW Command) or directly from a hardware event (it will be called HW Command).

5.2.1 SW Command

A first solution is to use as save command a signal generated on software request through a link (it will be called SW latch). This solution then provides an error on the dating due to the delay between the date of the event occurrence and the read operation of the counter on SW latch.

This solution leads to the time errors shown in the following diagram :

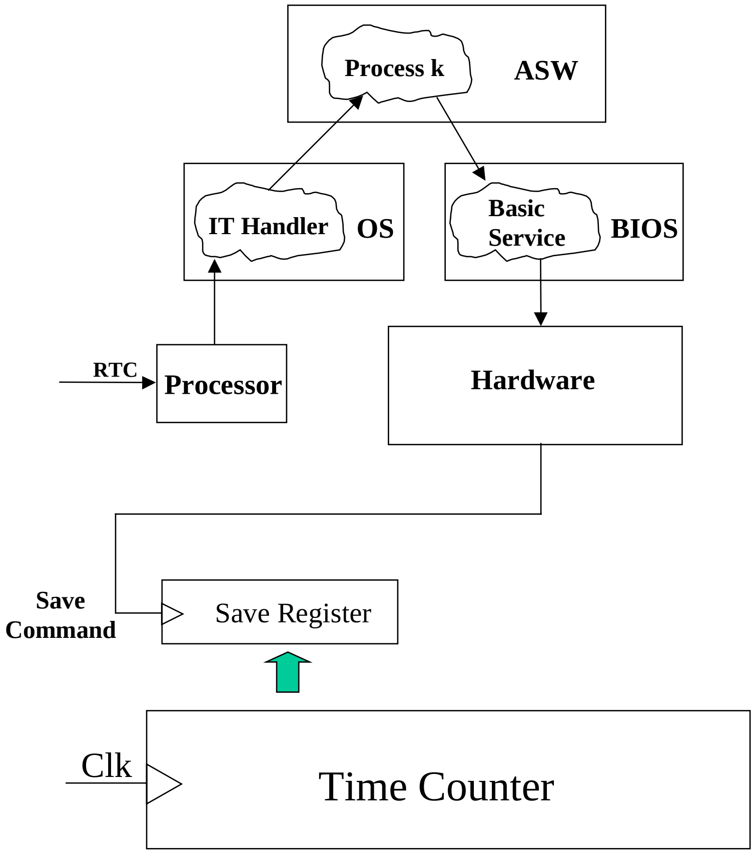

Figure 2 : *Save phase* on SW Command

The figure above shows the progress from a real time pulse (or RTC for Real Time Clock) request to the generation of the corresponding software command. This progress highlights several contributions which are :

- A **Processor_Save contribution** that corresponds to the time between the reception of a hardware signal and the transmission of the request to an interruption handler.
- An **OS_Save contribution** that is the processing duration in the IT Handler, which is located in the OS layer. Following this processing the request is then transmitted to the higher software layer that is the Application SoftWare or ASW.
- An **ASW_Save contribution** that is the error due to the processing in this layer. Then, the process concerned has to engage a *save phase* by activating the corresponding basic service in the lower software layer which is called BIOS for Basic Input Output Software.
- A **BIOS_Save contribution** that is the error due to the processing of the basic service involved in this layer. Afterwards, a signal (i.e. the SW Command) is generated to latch the save register. As the transmission of the command signal is controlled by the basic service then the delay due to this transmission is included in the BIOS contribution.

5.2.2 HW Command

A second solution consists in a signal directly from the concerned event (it will be called HW latch). For example, if the event to date is the RTC, one can use it as save command. Then, the event is sent directly to the save register and to the process of the ASW as previously at the same time. This second solution could be represented by the following diagram :

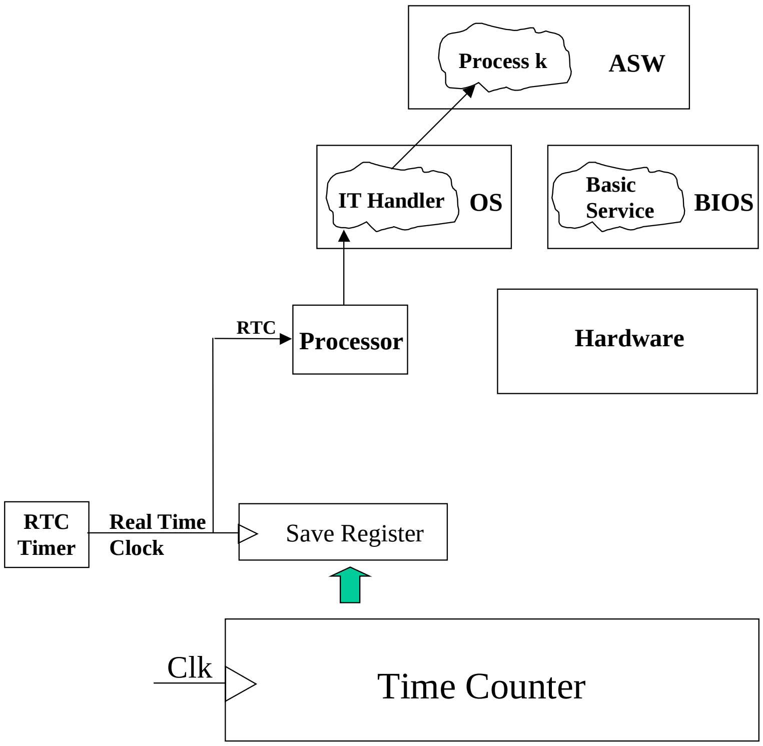

Figure 3 : *Save Phase* on HW Command

Once the date is available in the save register, it could be sent or broadcast to the users during a *read phase*. A user could be a SW service of the ASW, an equipment on board or even a ground station.

The single contribution to take into account is the propagation time between the generator of the HW Command and the latch input of the save register. This duration is generally low enough to be neglected. Then, one can assume that there is no contribution during the *save phase* when the save command is a HW Command. That is to say the date value saved into the save register corresponds almost exactly to the date when the HW event has occurred.

5.3 READ PHASE

The contributions that lead to an error during the *read phase* depend on the user synchronisation method. Indeed, a user could be synchronised on a HW event (RTC signal, PPS or other) or through a SW command (the "sync" mode code in a MIL-STD-1553 data bus solution). Moreover, it depends if the user is a SW application or an equipment on a data bus or even the ground station.

Firstly, we are going to identify all contributions involved in the progress of the date. In a second phase, we will identify, according to the synchronisation method, among all of these contributions which are errors and which are delays.

One assumes that the *read phase* is engaged after the safeguard of the counter content in the save register.

5.3.1 The user is an ASW application

The following figure will be used as a general visual support for this case.

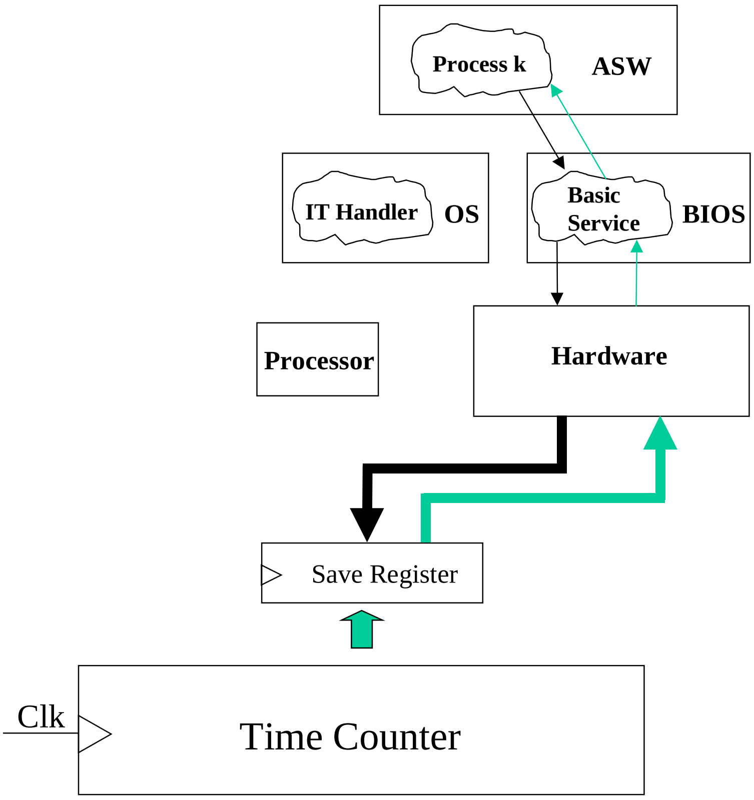

Figure 4 : *Read Phase* with SW application as user

During this phase, the process k has to run firstly the concerned basic service from the BIOS layer. This leads to a **BIOS_Read contribution**. As the transmission of the read command signal is controlled by a dedicated basic service (e.g. Get_Date), the delay due to this transmission is assumed to be included in this BIOS contribution. An **ASW_Read contribution** has to be taken into account too. Indeed, the process k comprises a part to manage the *save phase* and also a part to manage the *read phase*.

However, and depending on the design, the ASW contribution and BIOS contribution of the *save phase* and those of the *read phase* can be joined in a unique ASW contribution and BIOS contribution.

5.3.2 The user is a unit connected on a data bus

The first step of this *read phase* is the same as in the case where the user is an ASW application. For this case, we assume that the user is connected to the MIL-STD-1553 data bus.

Then, the date information has to be transmitted to this user through a data bus as follows.

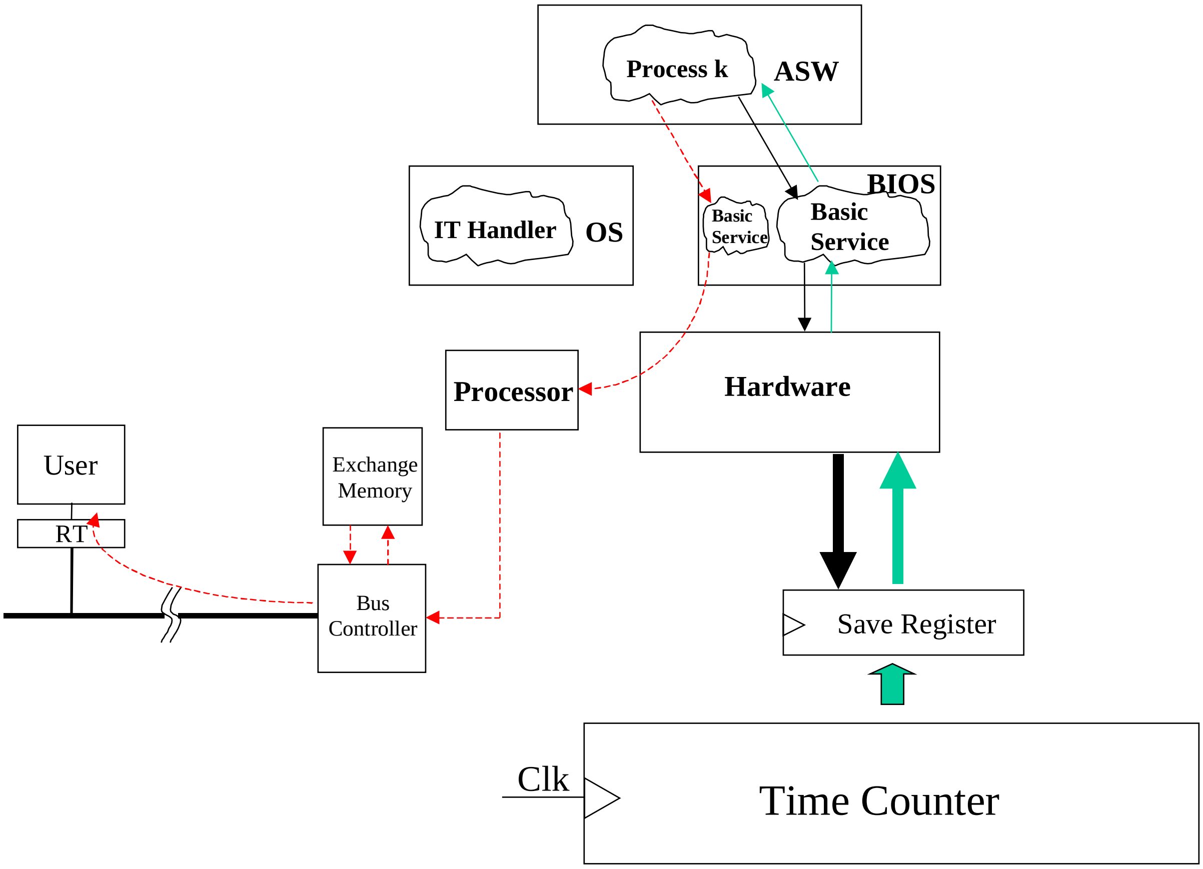

Figure 5 : *Read Phase* with Equipment as user

The first step, after the previous *save phase*, is identical with the case where the user is a SW application. That is to say, this *read phase* includes all the contributions involved in the *read phase* where the user is a SW application.

After this, the process k has to run the basic service in the BIOS layer that has in charge to transmit data to the bus controller (BC) of data bus on which the user is connected and especially in the exchange memory associated with the BC. This action is then composed of an **ASW_ToBC contribution** and a **BIOS_ToBC contribution**.

This data is afterwards managed to be transmitted on the data bus and also it may be formatted according to a packet protocol during a **Buffer contribution**. Moreover, an incoming data is not immediately transmitted on the data bus and is associated with a timing contribution. This depends essentially on the protocol and on the management of the exchange memory. Finally, when data is ready to be sent through the data bus to the user, the bus controller engage the concerned action. The duration of the transmission on the data bus is defined by a **Transmission contribution**.

However, it seems necessary to distinguish two cases. Indeed, the user may be synchronised either with a hardware signal, the date being sent through the data bus, (we will call it HW synchronisation) or by using a received date information as both synchronisation signal and date (we will call it SW synchronisation).

In these two cases the progress of the date includes the same contributions, the only difference is the means used to supply the synchronisation signal. Then, the contribution will be identical.

To identify the overall contributions that take part in this case, one has to add the contribution highlighted in the first step and those of the second step.

5.3.2.1 HW synchronisation

The following representation shows the transmission from a processing unit to the user location when the user and the save register are driven by the same signal.

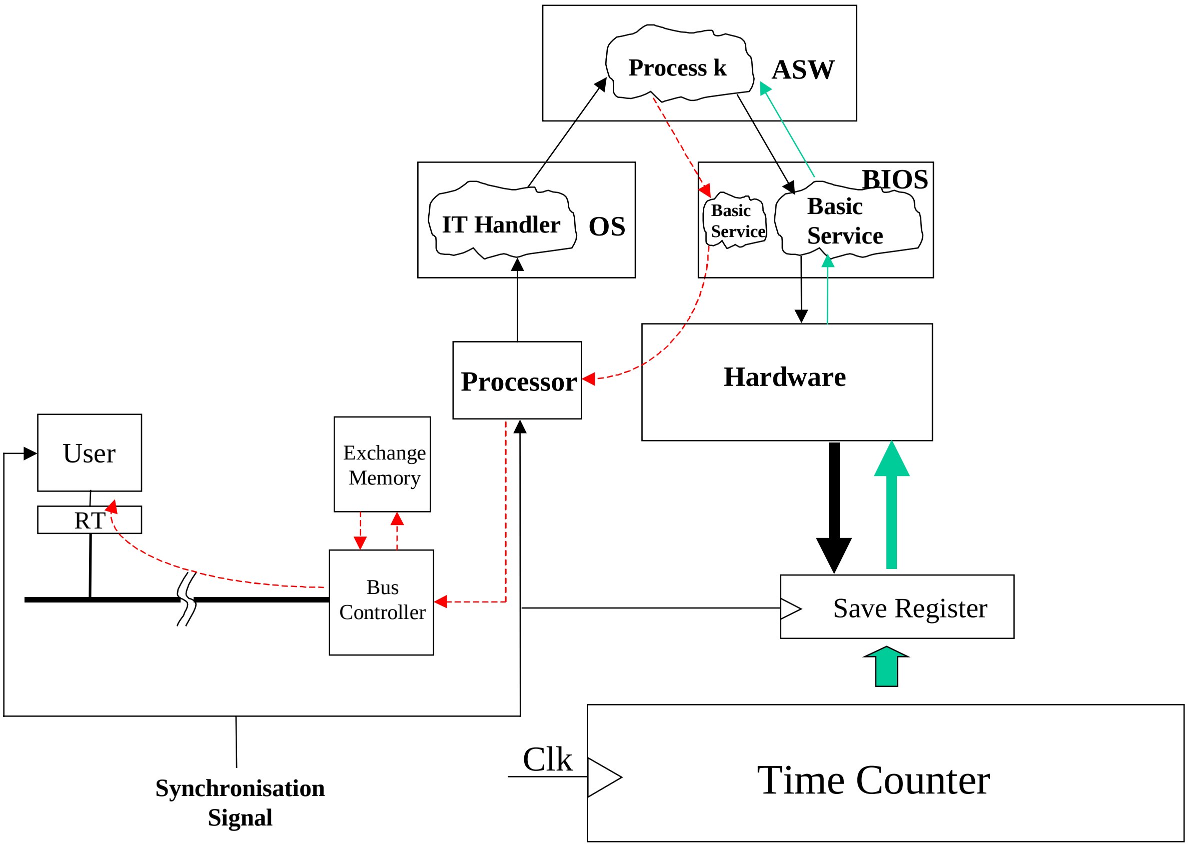

Figure 6 : HW Synchronisation

Then, all contributions identified during the *read phase* are not error but delay since the user has been immediately informed of the occurrence of the event to date.

5.3.2.2 SW synchronisation

The following representation shows the transmission from a processing unit to the user location when the user is synchronised directly by the date signal it received.

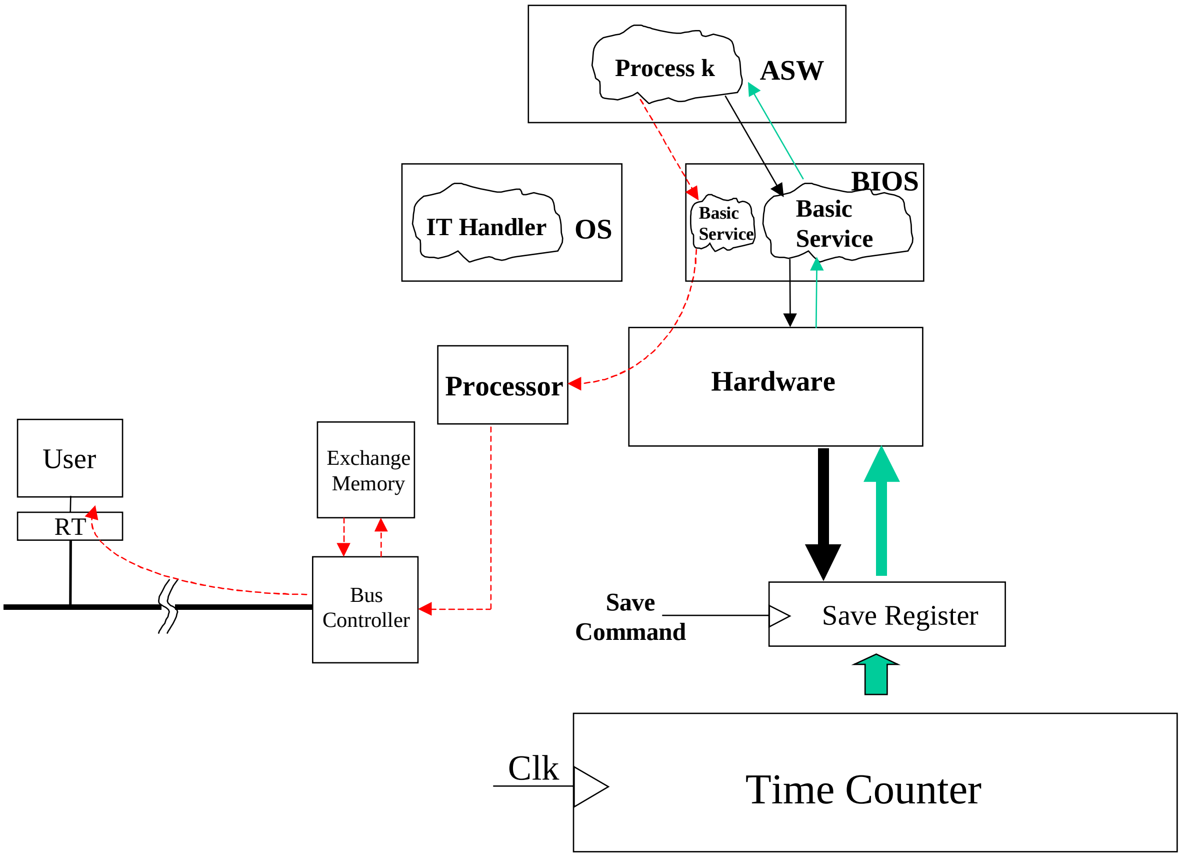

Figure 7 : SW Synchronisation

6. Transmission to ground

Regularly, the on board date has to be compared with the reference ground time during a *correlation phase*. The associated transmission procedure is described especially in the telemetry packet standard document (Ref. ESA PSS-04-106 Issue 1 January 1988).

This document describes how and when the on board date has to be inserted in a telemetry transfer frame. Indeed, each transfer frame from the virtual channel number 0 (VCA0) includes a counter value which is automatically incremented at each new transfer frame. This counter is a modulo 255 counter by default. One can change this modulo from 2^0 to 2^8.

Then, a save command must be generated each time a transfer frame from VCA0 with a counter value equal to 0 is sent, more precisely at each first leading edge of this transfer frame. Afterwards, the save command generated leads to the save of an on board time value in the save register as previously described (see § *Save phase*).

The content of the save register has to be transmitted to the ground before the frame counter of VCA0 has reached the end of the counting cycle (255 by default). The contributions during this phase are already described in the previously described *save phase and read phase*. One can add a Processing contribution due to the lapse of time between the occurrence instant of the first leading edge and the generation of the save command. Generally, the save command is generated by a hardware device (see HW Command).

When this first leading edge is received by the ground, it is used as a synchronisation signal (i.e. latch of a local counter) that leads to the generation of a ground time sample. As the first leading edge is received after a *propagation phase* then the ground time sample includes the contributions involved in this *propagation phase*. The contributions involved in this phase are :

- a **coding contribution** due to the presence of a coding stage (e.g. convolutional),
- a **TT&C contribution** due to the RF interface,
- a **Space contribution** due to the propagation time through the space,
- a **Position contribution** due to the uncertainty in the knowledge of the spacecraft position (i.e. ranging uncertainty).

For the accuracy of the ground time stamping an allocation of 50 µs is proposed (**Ground contribution**).

This phase could be associated with an update procedure from ground in order to correct the on board time in case of a time discrepancy with the ground station.

7. Update procedure

The ground station determines the value to send on board to update it. For this, it sends a kind of time-tagged telecommand frame that includes the new on board date to set. The time-tagged telecommand frame is referenced to the on board date. Then, at a pre-determined date (i.e. an alarm date) the update value is loaded in the on board free running counter through the load register.

This update procedure makes use of several phases that are a *time control phase*, a *load phase* and a *latch phase*. For the following, we assume that the time-tagged telecommand containing the alarm date value is already received and saved.

7.1 TIME CONTROL PHASE

The aim of this phase is to determine if the on board time counter has reached the target date. When the target date is reached, the *load phase* could be engaged.

The *Time control phase* is in fact a *save phase* followed by a *read phase* and a *comparison phase*. Generally, these *save phase* and *read phase* are cyclic phases. Then, regularly a SW process checks the time.

7.1.1 Save phase

The save command could be a HW or a SW command. In case of a SW Command, the contributions identified previously has to be considered there (i.e. **Processor_Save**, **OS_Save**, **BIOS_Save** and **ASW_Save**). For a HW Command there is no contribution for the *save phase*.

7.1.2 Read phase

The contributions that have to be considered in this phase corresponds to the case where the user is a SW application. These contributions are an **ASW_Read** contribution and a **BIOS_Read** contribution.

7.1.3 Comparison phase

In this phase the date from the save register is compared, at ASW level, to the target date : this is the **Processing_Comp** contribution. If they are equal, the *load* and *latch phases* have to be engaged to set the counter to the update value. As the *Save phase* is a cyclic phase, the time information provided is in fact a succession of discrete values separated by a cycle period longer than the resolution of the counter.

Consequently, it is possible that the alarm date value is not equal to one of these discrete values. Indeed, if for example the serial of time samples provided by the local OBT is 10, 20, 30, 40, 50 etc… when the alarm value is 56. The *comparison phase* will in fact result in an identification, in the OBT samples, of the nearest value (i.e. before or after the alarm value).

Then an approximation will have to be done by the software application. This case will result in an **Approximation_Comp** contribution varying from 0 to the cycle period of these discrete values.

In this example, the **Processing_Comp** will result in a time delay of 4 units because the nearest value for 56 is 60 (by taking 50, the time delay will result in a time advance of 6 units).

7.2 LOAD PHASE

As the *load phase* occurs after having detected that the on board time value is equal to the alarm date value, the *load phase* will then result in an error unless the identified contributions have been taken into account during the calculation of the update value.

This *load phase* is associated with an **ASW_Load** contribution and a **BIOS_Load** contribution.

7.3 LATCH PHASE

The Latch command generation principle of the *Latch phase* is the same as for the save command during the *save phase*. It could be a SW Latch or a HW Latch. The SW Latch results in an **ASW_Latch** and **BIOS_Latch** contributions when the HW Latch does not result in any contributions.

Whatever the solution is the latching date must be the same as the alarm date plus one cycle more. Indeed, as the comparison phase is according to the alarm date the latch phase can not occur at the same time. Then, at least one cycle more has to be taken into account in the update value. If the alarm date value is not synchronous with the Latch Command and if the time difference between these two dates is not taken into account in the update value this will result in a **Sync_Latch** contribution. This time difference can be less than an OBT cycle or equal to several OBT cycles.

NB : One has to take care that an update value does not lead to the loss of a time-tagged telecommand. Indeed, if a command is programmed to be executed at a date D1 when the update procedure is foreseen at a date D0 prior to D1, then one has to take care that the update date D2 is also prior to D1. In the contrary, the update date should be done at a date after D1.

7.4 PROCESSING

A second solution is to let the on board time counter run without an update phase and to take into account any oscillator drift effect on ground. This solution provides a continuous Central Time Reference CTR (without any jump because of an update). Thus, an equipment (payload, …) will work with a continuous time without jump.

On the other hand, the counter is driven by clock affected by a drift from an oscillator that is subject to a drift (see § A-1 Oscillator features).
A CTR need from the Herschel Planck project (see requirement SMCD-205 of the SRS) concerns the maximum authorised drift that is at least 1x10^{-6}. If one considers a daily (i.e. 24 H) update procedure then a maximum budget to not exceed is 86.4 ms.
The total error contribution involved in the transmission to ground phase (for update) is about 70 µs for a HW save or 130 µs for a SW save (see Table 3). The major contributor in the maximum budget (86.4 ms) is the drift due to the oscillator (see § A-1 Oscillators features). By taking into account the previous total error contribution in this budget, the budget part for the oscillator drift contribution (over 1 day) is 86.33 ms or 86.27 ms respectively. Then, the oscillator stability is 0.9992 ppm or 0.9985 ppm respectively.
Consequently, the total error contribution is negligible regarding the impact on the maximum budget authorised.

8. Budgets

Any contribution could be expressed as a delay or an error. This depends on the application. The first table concerns the case of a date transmission to an equipment on a data bus. The second table concerns the update procedure.

Consequently, one can select from these tables the solution adapted to the user requirement (e.g. HW save with the lowest error contribution).

Phase	Contribution	Typical Value (µs)	Transmission to an equipment			
			HW Sync**		SW Sync**	
			HW Save	SW Save	HW Save	SW Save
Save Phase	Processor_Save	10	Delay	Error	NA	Error
	OS_Save		Delay	Error	NA	Error
	ASW_Save	(see comment 1)	NA	Error	NA	Error
	BIOS_Save	50	NA	Error	NA	Error
Read Phase	ASW_Read	(see comment 1)	Delay	Delay	Error	Error
	BIOS_Read	50	Delay	Delay	Error	Error
	ASW_ToBC	(see comment 1)	Delay	Delay	Error	Error
	BIOS_ToBC	60 or 3840 (2)	Delay	Delay	Error	Error
	Buffer	8 or 70 (2)	Delay	Delay	Error	Error
	Transmission	20 (3)	Delay	Delay	Error	Error
	User error	10	Error	Error	Error	Error
Total Error			10	70	148 or 3990	208 or 4050
** This indicates the user synchronisation means						
NA means Not Applicable						

Table 1 : **Transmission to an equipment**

Phase		Contribution	Typical Value (µs)	Update			
				SW Save		HW Save	
				HW Latch	SW Latch	HW Latch	SW Latch
Time control phase	Save Phase	Processor_Save	10	Error	Error	NA	NA
		OS_Save		Error	Error	NA	NA
		ASW_Save	(see comment 1)	Error	Error	NA	NA
		BIOS_Save	50	Error	Error	NA	NA
	Read Phase	ASW_Read	(see comment 1)	Delay	Delay	Delay	Delay
		BIOS_Read	50	Delay	Delay	Delay	Delay
	Comparison Phase	Processing_Comp	(see comment 1)	Delay	Delay	Delay	Delay
		Approximation_Comp	(see § C-I-2 Comparison Phase)	Delay	Delay	Delay	Delay
Load Phase		ASW_Load	(see comment 1)	Error	Error	Error	Error
		BIOS_Load	50	Error	Error	Error	Error
Latch Phase		ASW_Latch	(see comment 1)	NA	Error	NA	Error
		BIOS_Latch	50	NA	Error	NA	Error
		Sync_Latch	50 (see § C-III Latch Phase)	Error	Error	Error	Error
Total Error				160	210	100	150

NA means Not Applicable

Table 2 : Update procedure

Phase	Contribution	Typical Value (µs)	Transmission to Ground	
			HW Save	SW Save
Save Phase	Processor_Save	10	NA	Error
	OS_Save		NA	Error
	ASW_Save	(see comment 1)	NA	Error
	BIOS_Save	50	NA	Error
Read Phase	ASW_Read	(see comment 1)	Delay	Delay
	BIOS_Read	50	Delay	Delay
	ASW_ToTM	(see comment 1)	Delay	Delay
	BIOS_ToTM	50	Delay	Delay
Propagation Phase	Coding		Error	Error
	TT&C		Error	Error
	Space	T	Delay	Delay
	Position	10	Error	Error
	Jitters	3 to 9	Error	Error
Ground Datation Phase	Ground	50	Error	Error
Total Error			63 to 69	123 to 129

Tableau 3 : Transmission to ground

Comment 1 : Generally, the ASW contributions involved in OBT aspects and identified above are not independent software process. They are grouped by function (e.g. update, communication with a remote terminal, …). Then, a process from the ASW calls several consecutive BIOS services. That is why a dimensioning of an ASW contribution can be estimated as an overall value normally negligible regarding the BIOS contribution.

Comment 2 : An estimation based on the use of a BU61582 (DDC) component as bus controller leads to affect the efficiency (due especially to the mandatory use of wait states). Then, the typical values for the Buffer contribution are 15 µs for a single command request, that is an elementary 1553 command word with 32 data word, and 140 µs for a frame command request (i.e. packet mode), that is a succession of up to 64 single commands. Concerning the BIOS_ToBC contribution we have also to consider the case of a single command request with 32 data words and this of a frame command request with 64 commands. The results for the DDC bus controller are 120 µs and 7680 µs respectively.

The typical values presented in the table 1 are provided assuming that the bus controller is now associated with a zero wait states component. With this assumption, one can expect a reduction of the typical value by a factor two. Consequently, the typical values given in the table 1 are estimated to 8 µs for a single command request and 70 µs for a frame command request for the Buffer contribution and 60 µs and 3840 µs for the BIOS_ToBC contribution respectively.

Seemingly, only single command request should be used to send time information to a data bus user.

Comment 3 : An estimation of this contribution depends particularly on the period of the clock and on the size of the number of words to transmit. A first approximation could be about 20 µs.

For this estimation, a transmission, over a MIL-STD-1553 serial data bus with a 1 µs clock period, of 1 command word (20 bits) followed by n data words (nx20 bits) for the time has been considered. As the word size of a 1553 data word message is 16 bits and the time format is 7 bytes, the transmission of a time sample over such a data bus requires 4 data words. Consequently, the command word is followed by n=4 data words. The command word is used by the equipment user remote terminal to provide internally a synchronisation pulse.

The transmission contribution comprises in fact only the duration of the command word transmission over this serial data bus. Consequently, the transmission contribution is about 20x1µs being 20 µs. The equipment has to wait 4x20 µs more to receive the time sample.

Furthermore, when an equipment is receiving a message from the bus (e.g. a synchronisation command word), it has to process it. This results in an equipment processing error called **User error** that depends on the equipment design.

Afterwards, a synchronisation pulse can be generated inside the equipment by taking into account the user error. The reception of the time sample follows the reception of the command word and is used by the equipment as the date of the internal pulse event.

Comment 4 : These estimations are given by considering the use of an ERC32 processor at 20 MHz.