ANNEX 7

DFEE USER'S MANUAL

DFEE

SPI-MU-S-14241-CEA

DFEE User Manual

Document Ref.	Version	Date
SEI-SPI-MM-466-01	**V 1.2**	September 11, 2002

		signature	date
Written by	**Michel MUR** ***Electrical Architect*** ☎ (33)1 69 08 33 86 *michel.mur@cea.fr*		
Checked by	**Stéphane SCHANNE** ***ASIC Designer*** ☎ (33)1 69 08 91 21 *stephane.schanne@cea.fr*		
Authorised by	**Françoise LOUBERE** ***Product Assurance Manager*** ☎ (33)1 69 08 59 77 *francoise.loubere@cea.fr*		
	René DUC ***Project Leader*** ☎ (33)1 69 08 55 03 *rene.duc@cea.fr*		

Commissariat à l'Energie Atomique - Direction des Sciences de la Matière - Département d'Astrophysique, de Physique des Particules, de Physique Nucléaire et d'Instrumentation Associée

Printed 11/09/02

TABLE OF CONTENTS

TABLE OF CONTENTS

FIGURES

TABLES

INTRODUCTION

This document describes the Digital Front-End Electronics (DFEE) subsystem of the SPI (Spectromètre Pour Integral) gamma-ray instrument. Strong emphasis is put on the definition of the in-flight adjustable parameters, and on the detailed description of the various outputs produced by the DFEE.

Applicable documents

Specification documents

- « Spécification technique de besoin du DFEE », SPI-ST-5-1045-CNES, V4.0 ;
- « *Timing definition for DFEE, AFEE, ACS and PSD* », SPI-NT-0-1317-CNES, V3.2 ;
- « *SPI System Datation Budget* » SPI-NT-0-1426-CNES, V1.1.
- « *SPI Science Data Format* », SPI-ST-0-2911-CNES, V2.2

Interface documents

- « *Experiment Interface Document EID-Part B* », SPI-SG-0/SAT-1111-CNES, V4.0 ;
- « *SPI Interfaces Specification* » SPI-SI-0-1324-CNES, V3.1

  Annex 1 : « *General Interface Requirements* »;

  Annex 21.3 : « *DFEE Software Interfaces* ».

Other DFEE documents

- « *L'ASIC DFEEEA51C et son modèle le FPGA DFEEvcNV* », V1.202 [in french].

Related publications

E. Lafond, M. Mur and S. Schanne, « The Digital ASIC for the Digital front-end Electronics of the SPI Astrophysics Gamma-Ray Experiment », *IEEE Transactions on Nuclear Science*, August 1998, Volume 45, Number 4, pp. 1836-1839.

Editorial conventions

Number radix

Throughout this document, binary numbers and fields are expressed in decimal format by default. If they are expressed in hexadecimal format, their value is preceded by the $ sign.

DFEE OVERVIEW

The Digital Front End Electronics subsystem (DFEE) is in charge of the real time acquisition, assembly, time stamping and intermediate storage of the various events produced by the SPI spectrometer. It receives and analyses the logical signals created by the SPI front-end subsystems. The corresponding physics events are classified and formatted into several digital record blocks which are delivered at regular intervals to the DPE flight computer.

Detector input

The DFEE handles the following information :

- Time of occurrence, energy measurement, channel occupancy and saturation conditions of Ge events, as produced by the corresponding 19 independent Analogue Front End Electronics units (AFEE) ;
- Time of occurrence and saturating condition of BGO shield events, as produced by the Anti Coincidence Subsystem (ACS) ;
- Time of occurrence and identification of a qualified subset of Ge events, as produced by the Pulse Shape Discrimination Unit (PSD).

Real time processing

The DFEE first compensates for the differential delays that show up in the various electronic chains. After alignment, the activity of the synchronised signals is then analysed to detect physics events that were observed on one (or several) Ge detector(s). They are identified and isolated, and the corresponding AFEE and PSD information – time, relative time, AFEE energy, PSD Identifier – is recorded. When the ACS system activates the veto signal, an internal veto gate is created, with well-defined delay and width characteristics. Events that occur while the internal veto gate is active are marked as vetoed, and will later be handled separately.

Real time output

When in operation, the DFEE processes the ACS veto signal and creates an internal veto gate for AFEE signal rejection. It also creates and outputs a secondary veto gate, with a different delay, to guide signal rejection inside the PSD.

Temperature and voltage analogue signals are also made available to the DPE for monitoring.

Telemetry output

The DFEE delivers the following information to the DPE, for inclusion in the telemetry packets :

- The list of *single non vetoed AFEE events*, including their time of occurrence and associated energy ;
- The list of *single non vetoed AFEE-PSD events*, for which the PSD unit provided additional information ;
- The list of *multiple non vetoed AFEE events*, for which several Ge detectors were observed active within a predefined signal analysis time window ;
- The list of *event energies*, separately arranged for each Ge detector, and intended to populate individual energy spectra within the DPE ;
- A block of *system monitoring statistics*, giving signal counts, dead time measurements and other system activity values.

Timing

Global system synchronisation is provided by a slow clock signal provided by the DPE (8 Hz frequency). The instants at which science and monitoring packets are delivered to the DPE are derived from this clock.

Fine timing information is provided by a local 20 MHz clock signal, which is also used to synchronise the input signals and schedule the overall internal operation of the DFEE. All internal adjustments and measurements are related to this 20 MHz clock.

Time Frame sequence

The DFEE uses the 8 Hz reference signal to define successive 125 ms interval *time frames*. The overall operation runs in a 2-stage pipeline scheduled at time frame boundaries :

- During stage 1, the real time input information is analysed and processed ;
- During stage 2, the corresponding results are passed to the DPE on a fast dedicated data output link (High Speed Link, or HSL).

Statistics and status are passed at 1 s intervals on the general purpose DPE communication link (Low Speed Link, or LSL), which is also used to prepare, initialise and supervise the DFEE.

Architecture

The DFEE signal processing operations are handled by a single dedicated digital Application Specific Integrated Circuit (ASIC). When it has been configured for action and

started, this circuit synchronises to the 8 Hz reference, runs the cyclic time frame analysis pipeline and creates the corresponding output records in external memory. Monitoring values – signal counts, channel dead time measurements, status and error flags – are created in a set of dedicated internal circuit registers. With a single time frame delay, the circuit retrieves the events found in the previous time frame and serialises them onto the HSL link in the format expected by the DPE.

A programmable micro-system, based on the 8051 micro-controller, supervises the DFEE ASIC. The DFEE micro-system communicates with the DPE flight computer over the bi-directional Low Speed Link (LSL). The flight software runs the standard SPI subsystem control-command sequence, augmented by DFEE-specific variations. When fully configured and started in Operational mode, the micro-system software runs an infinite cyclic loop, according to a recurrent 1-second pattern. At each pass through this loop, the microcontroller extracts the relevant information from the ASIC registers, analyses it and prepares the corresponding Housekeeping packets. These packets are delivered to the DPE, on its demand, in the next following second. When started in Diagnostic mode, the flight software essentially shows the same behaviour, simply making additional diagnostic packets available to the DPE.

A structural block diagram of the DFEE is shown in Fig. 1.

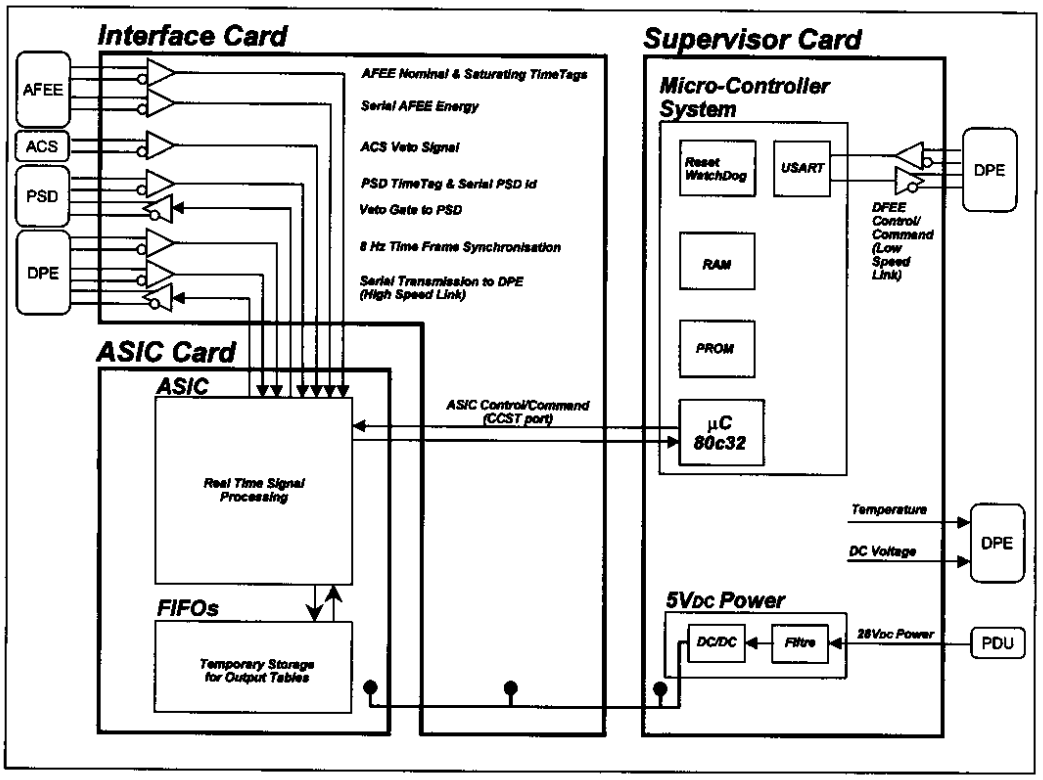

Fig. 1 DFEE structural block diagram

ASIC OPERATION

This section describes the internal operation of the ASIC circuit...

DFEE functional diagram

A functional view of the DFEE, with emphasis on the ASIC internals, is shown in Fig. 2.

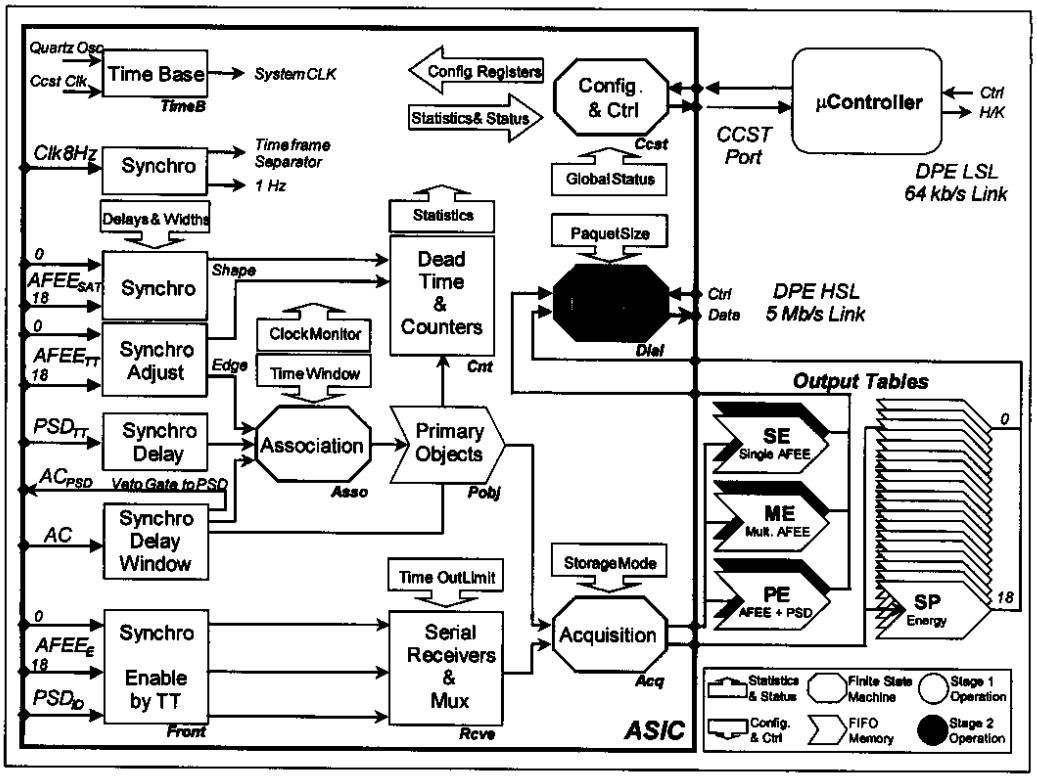

Fig. 2 DFEE functional block diagram

Front-end section

Signals

AFEE signals

For each AFEE subsystem, the DFEE receives the following set of signals :

- Nominal Time Tag pulse ;
- Out of range Time Tag pulse ;
- Energy, supported by a 3-signal serial protocol (envelope, bit clock, bit data).

The AFEE nominal time tag signal indicates both the time of occurrence of the Ge event (leading edge of the pulse) and the channel occupancy time (pulse width). It will be followed by serial transmission of the energy.

The AFEE out of range time tag signal indicates both the time of occurrence of the saturating Ge event (leading edge of the pulse) and the channel recovery time (pulse width). No energy is given.

PSD signals

The PSD system provides :

- Qualified Time Tag pulse ;
- Event Identifier (Id) and early reject flag, supported by a 3-signal serial protocol (envelope, bit clock, bit data).

The PSD time tag signal indicates the time of occurrence of a non-vetoed single Ge event in the primary selection range of the PSD[1]. It will be followed by serial transmission of the event Id and early reject flag.

ACS signal

The ACS system provides a single ACS Veto pulse. The ACS veto signal indicates the time of occurrence of the shield event (leading edge of the pulse) and an indication of saturation (pulse width). Pulse widths larger than an agreed threshold indicate saturation.

Synchronisation

The DPE provides the 8 Hz timing reference. Time frame boundaries are signalled by the falling edge of this clock.

[1] The primary range is wider than the actual PSD energy range. A fraction of these events will have an early reject flag added at Id time, after energy has been calculated more precisely by the PSD.

Signal processing

Synchronisation and alignment

The various *time tag* signals – asynchronous pulses from the AFEE, PSD and ACS units, and 8 Hz time reference signal – are first synchronised by the local 20 MHz clock. On every signal, a dedicated circuit is then placed to filter out any unexpected transition glitch. Next, the differential delays between the different units are compensated by configurable digital delay lines. The resulting aligned digital signals are then passed to the *association* process for correlation.

The ACS input pulse is analysed and processed to build the internal anti-coincidence window used to veto the AFEE time tags. Narrow ACS input pulses indicate normal ACS operation, whereas wide ACS input pulses are an indication of internal saturation in the AC shield. The actual internal veto gate is created with a different width in each case. The threshold used to discriminate normal and saturating conditions, and the pulse width extension used for each case are all three programmable.

This veto gate is also sent to the PSD subsystem, with its own dedicated delay adjustment. The PSD subsystem applies this window to its internal signals before forwarding them as qualified time tags to the DFEE.

Association

The *association* finite state machine continuously observes the aligned signals produced by the input sections, and dynamically aggregates them into the *primary object* table, on the basis of a predefined *association time window*.

Primary objects

A primary object is a chain of (one or more) events, in increasing time of occurrence, such that none of them is distant from the previous one by more than the *association time window*.

The association machine opens a primary object when activity starts on AFEE or PSD, and closes it when activity has ceased for more than the *association time window*. At opening time, it stores the time of occurrence of the object. For each following event, it stores the event pattern value and the interval elapsed from the previous event. At closing time, it has built up sufficient knowledge to identify and label the object for the following processing stage.

Primary objects are labelled according to their type – *Single (SE)*, *Single with PSD (PE)*, and *Multiple (ME)* – and carry additional global properties : The *Veto* flag indicates whether one at least of the object component(s) was observed while the anti-coincidence window was active, and the *TimeFrame* flag indicates that a time frame boundary event occurred during the time span of the object. The object label also indicates the number of data components in the objects.

Time frame boundaries

Isolated time frame boundary transitions create a special type of primary object – *Time Frame Boundary* –, which will be used to schedule the second stage of the time frame operation pipeline, and then disappear.

Object & event timing

At primary object creation, events are time stamped with the current value of a local clock counter incremented at each local clock edge. At each time frame boundary, the local clock counter loops back to value 1 without any clock edge loss.

Later, object time will be stored – albeit with lower precision – as the offset from the beginning of the time frame. Inside a composite object, the relative timing across events generally maintains the full local clock resolution. Due to technical constraints, the exact time interval between events in an object gets fuzzy by 1 clock period when events are at a 1-clock distance of each other. However, this unwanted fuzziness can be recovered – at post processing time – for the 2 first elements in a composite object.

Object queue

The primary objects produced by the real time association are placed in a First In First Out queue ordered in increasing time of arrival. They are made available to the following processing stage – *acquisition* – at the queue head position. The FIFO policy is intended to provide storage time for the primary objects while the corresponding Energy or PSD Id values are awaited for, and also provides de-coupling and de-randomisation between the *association* and *acquisition* activities.

The object label is produced at object closing time, when all corresponding data components have already been stored. The acquisition processing stage first requires the object label, and only then captures the object data. The primary object FIFO provides 2 insertion pointers and the associated logic to present the output information in the required order (object label followed by object data).

Acquisition

The acquisition collects the energy and PSD Id, creates the output tables and schedules the time frame pipeline.

Object loop

When idle, the acquisition finite state machine continuously monitors the head of the primary object FIFO queue. When a label comes in, it starts a chain of actions that depend on the object type and properties, and completes when all corresponding event components have been processed. The machine then returns to idle state to process the next object.

Collection & storage

The acquisition collects the energy or PSD Id attached to each event in the object, formats the information and stores it in the correct destination.

For each event pattern, collection is started until all events have been assigned their corresponding energy or PSD Id, or time out occurs. If time out occurs, the corresponding time frame error status is updated and the pending events are attached special values of energy (E_{to} = $0000) or Id ($Id_{to}$ = $0000) which may later help identify the faulty channels.

Time frame pipeline

Accepted events (SE, ME, and PE objects) are stored in an external double-buffered FIFO memory. At any particular time, one of the buffers is assigned to accept fresh objects from the current time frame, while the other one is delivering the content of the previous time frame to the DPE. When an isolated time frame boundary object occurs, the acquisition toggles the even/odd time frame flag to swap the function of these intermediate output buffers. This operation is also performed when an object carries the embedded time frame boundary flag ; in that case, it is delayed until all components in the object have been processed. As a result, objects that span over a time frame boundary region are not split, and are always stored in the time frame where they were initiated.

Statistics & Status

The DFEE ASIC collects various signal statistics and produces internal status. The corresponding information is accumulated and double-buffered in the circuit. It is made available to the micro-system at time frame boundaries. After retrieval, the flight software post-processes the data and passes the resulting packets to the DPE at 1-second intervals.

Signal counts

The following signal count statistics are constructed in real time and accumulated over 1-second intervals :

- The number of distinct Ge time tags that were received from the AFEEs (19 for nominal range, 19 for saturating range) ;
- The number of distinct nominal range Ge time tags that were received from the AFEEs and declared non-vetoed after internal alignment (19) ;
- The number of PSD time tags that were received.

Signal dead time

For each Ge detector, the DFEE combines the AFEE proper channel dead time and the anti-coincidence gate dead time, to avoid double counting. This is obtained by measuring the duty cycle of an intermediate signal obtained by OR-ing the nominal time tag, the out of range time tag and the anti-coincidence window after signal alignment.

Internal event counts

The following internal event monitors are produced :

- The number of ACS veto gates created with the low threshold width ;
- The number of ACS veto gates created with the high threshold width ;

- The number of distinct resulting ACS veto gates ;
- The number of PE events which were demoted to SE due to an Early Reject Id from the PSD.

Status & error

Status and error conditions maintained by the various concurrent state machines are collected for each time frame and summarised in the corresponding housekeeping records. They are included both in the DFEE standard status block (HK #$00) and in one of the scientific housekeeping blocks (HK #$04). The raw status bits created for each time frame are also available in special housekeeping blocks dedicated to diagnosis.

Clock monitor

The value of the local clock counter is sampled at each time frame boundary and included in the one of the scientific statistics blocks (HK #$0B). From this sequence of numbers, short and long term potential drifts of the local oscillator may be monitored.

HSL COMMUNICATION

*Once in operation, the DFEE sends its results to the DPE on a unidirectional data link channel (*High Speed Link*), under DPE control.*

HSL overview

The HSL link is a block oriented, unidirectional synchronous bit-serial data link. When the system is active, a fixed length packet transfer is initiated by the DPE at each time frame. The DFEE produces a HSL bit-stream with the following properties :

First, the SE, ME and PE blocks corresponding to events taken in the previous time frame are given, in this order, and if space is available in the packet. Due to the double-buffer policy used for SE, ME and PE blocks, events that could not fit due to lack of space are irremediably lost.

Next, the 19 SP blocks are provided in a loop. If all of them do not fit in the remaining space after SE, ME and PE transmission, the loop is aborted, and will restart from where it left in the next time frame packet, for another potential 19-block series. SP events are not lost when abortion occurs. However, the internal DFEE buffers dedicated to SP block storage may overflow in the long run, with no notice.

Caution

The following sections describe the detailed format of the HSL packets provided to the DPE, at the DFEE interface. Their content may later be buffered, combined and reworked by the DPE IASW software when building the actual telemetry packets, and some details may end up differently when data is analysed on ground. Refer to document *« SPI Science Data Format »* for a description of the actual telemetry format.

Block chaining policy

The rationale used to build the serial HSL bitstream packet is shown in Fig. 3 (read from left to right).

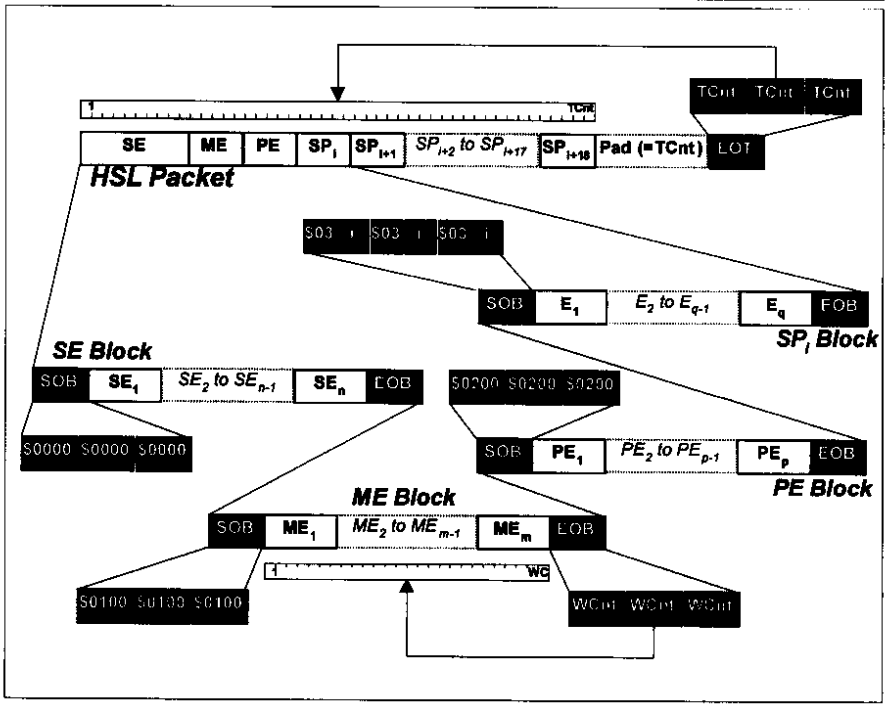

Fig. 3 Block chaining in the HSL packet

The total size of the transmitted HSL packet is fixed, and is defined at DFEE configuration time by the HslXferLen (E7783) parameter. The varying number of useful 16-bit words is *Tcnt*. The value of *Tcnt* is given as the End Of Transfer marker (EOT) at the end of the fixed length packet. In the *Tcnt* length payload, the SE block is provided first, followed by the ME and PE blocks. The sequence of rotating Spectra blocks is then produced, starting from spectrum index #i, for a maximum of 19 SP blocks. The initial spectrum index is set to #0 at the beginning of the Run. The DFEE always tries to produce 19 SP blocks in each packet ; If it does not succeed (due to lack of space), it resumes the SP series from where it left in the immediately following packet. For each block, the SOB (*Start Of Block*) header identifies the source and the EOB (*End Of Block*) trailer indicates the block word count and status. The EOT, SOB and EOB elements are given with triple word redundancy, and may be submitted to majority votation in the IASW sotware.

Generic SOB Format

The SOB headers encapsulate the blocks and identify their source.

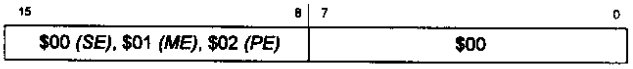

Fig. 4 Format of the SOB header word for SE, ME and PE blocks

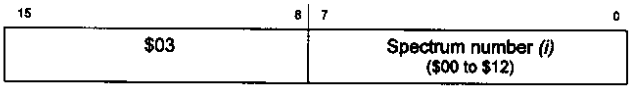

Fig. 5 Format of the SOB header word for SP blocks

Generic EOB Format

The EOB essential function is to give the actual number of words of the block payload. In addition, the *Partial* bit is set if the block could not be installed completely due to lack of space in the HSL packet.

15	14	13 … 0
Partial (1 if truncated)	0	WCnt (0 to 8192)

Fig. 6 EOB trailer word format for SP blocks

For the SE, ME and PE blocks, the EOB trailer also signals with the *ParityErr* bit that data alteration occurred during temporary storage in the corresponding *Output Table* memory.

15	14	13 … 0
Partial (1 if truncated)	ParityErr (1 if error)	WCnt (0 to 8192)

Fig. 7 EOB trailer word format for SE, ME and PE blocks

SE block format

The SE block structure is the following (read from left to right) :

SE SOB	SE body (n = 2*m words, size given by SE EOB.WC field)				SE EOB
(3 identical words)	SE event #1 (2 words)	SE Event #2 (2 words)	…	SE Event #m [last] (2 words)	(3 identical words)

Fig. 8 SE block format

The SE body gives the list of SE events encountered during the current 125 ms interval, ordered in increasing time of occurrence. Each following SE event occupies a fixed space (2 words).

SE block : SOB word

The SE SOB identification word has the following format :

15 … 8	7 … 0
$00	$00

Fig. 9 SE SOB header word format

SE block : EOB word

The SE EOB word gives word count and status information corresponding to the SE events of the current 125 ms interval. The SE EOB word has the following format :

15	14	13 … 0
Partial (1 if truncated)	ParityErr (1 if error)	WCnt (0 to 8192)

Fig. 10 SE EOB trailer word format

SE EOB Word Count

Gives the number of words in the block, excluding itself. Valid range is [0 .. 8192] inclusive. A value of 8192 indicates that the DFEE had to discard SE events during the corresponding 125 ms interval, due to internal storage limitations ; in that case, the block contains the first 4096 SE events detected in this interval. Word Count values higher than 8192 are invalid.

SE EOB Parity flag

Equals "1" when a parity error was detected in the ASIC. This status bit indicates that at least one of the SE events was altered during temporary storage in the DFEE.

SE EOB Partial flag

Equals "1" when the DFEE could not send all the SE data within the current 125 ms cycle, due to lack of space in the HSL packet. In that case, the corresponding ME and PE blocks are empty. If, as is foreseen in the flight configuration, the chosen HSL packet length (HslXferLength, E7783) is greater or equal to 8201, the partial flag will not be set on the SE block because the packet is large enough to contain even the largest SE block.

SE block : Events

A SE event is a fixed format 2-word assembly. The first word gives the AFEE energy, as it was received from the AFEE. It is replaced by $0000 if the AFEE did not produce the energy before the delay defined by DFEE time out parameter TimeOutValMissNrj E7776 (abnormal condition). The second word gives the time of occurrence of the AFEE Time Tag, and the AFEE channel number. The detailed format of an SE event is the following :

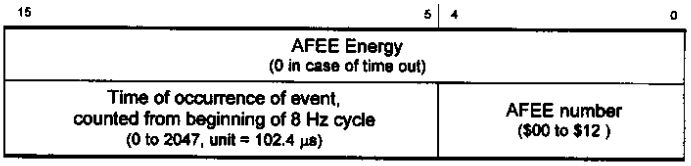

Fig. 11 SE event format

SE AFEE Energy format

When not replaced by $0000 due to time out, the AFEE Energy word shows the following generic format :

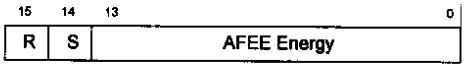

Fig. 12 SE AFEE energy format

R : indicates the energy range where "0" = [20KeV .. 2MeV] and "1" = [2MeV .. 8MeV] ;

S : spare bit equals "1" ;

Energy : 14 bits for energy conversion value.

ME block format

The ME block structure is the following (read from left to right) :

ME SOB	ME body (n words, size given by ME EOB.WC field)				ME EOB
(3 identical words)	ME Event #1 (3 to 63 words)	ME Event #2 (3 to 63 words)	...	ME Event #m [last] (3 to 63 words)	(3 identical words)

Fig. 13 ME block format

The ME body gives the list of Multiple Events encountered during the current 125 ms interval, ordered in increasing time of occurrence. Each following Multiple Event occupies a variable space (from 3 to 63 words), depending on the number of elements it contains, and ends up with a label which gives its size. The ME block body must be analysed by backwards chaining in the reverse (right to left) direction, starting at the end of the list and hopping from a Multiple Event to the previous one until the first one has been reached.

ME block : SOB word

The ME SOB identification word has the following format :

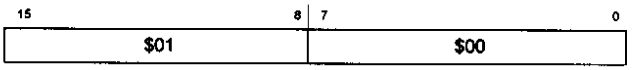

Fig. 14 ME SOB header word format

ME block : EOB word

The ME EOB word gives word count and status information corresponding to the ME events of the current 125 ms interval. The ME EOB word has the following format :

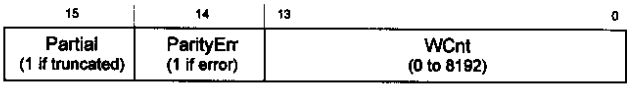

Fig. 15 ME EOB trailer word format

ME EOB Word Count

Gives the number of words in the sub-block, excluding itself. Valid range is [0 .. 8192] inclusive. A value of 8192 indicates that the DFEE had to discard ME events during the corresponding 125 ms interval, due to internal storage limitations ; in that case, the block contains the first ME events detected in this interval, and the last ME event is generally truncated. Word Count values higher than 8192 are invalid.

ME EOB Parity flag

Equals "1" when a parity error was detected in the ASIC. This status bit indicates that at least one of the ME events was altered during temporary storage in the DFEE.

ME EOB Partial flag

Equals "1" when the DFEE could not send all the ME data within the current 125 ms cycle, due to lack of space in the HSL packet. In that case, the corresponding PE sub-block is

empty. If, as is foreseen in the flight configuration, the chosen HSL packet length (HslXferLength, E7783) is greater or equal to 16399, the partial flag will not be set on the ME block because the packet is large enough to contain both the largest SE and ME sub-blocks.

ME block : Events

A Multiple Event gathers the activity of several detectors, for which the corresponding Time Tag sequence formed a cascade. The time distance between any two following elements of the cascade is smaller or equal to the DFEE AssoMultWinSize parameter (E7753). PSD Time Tags may be part of the cascade, at any position, and may create links between AFEE Time Tags which would not be assembled otherwise. Cascades formed with PSD only information (Pure Psd) may be included in the ME sub-block if the DFEE EvtKeepPP parameter (E7780) is set.

A Multiple Event first gives the list of its elements, ordered in increasing time of occurrence, and ends up with a 1-word label which gives the number of elements and the time of occurrence of the first element. Elements are either of the AFEE or PSD type.

The generic format of a Multiple Event is detailed in the following diagram, which shows a m-element ME where both types of elements (AFEE and PSD) are illustrated (read from top to bottom).

15 — 5	4 — 0	
AFEE Energy (0 in case of time out)		*Element # 1* (AFEE)
ΔT (0, 1 or 3)	AFEE number ($00 to $12)	
AFEE Energy (0 in case of time out)		*Element # 2* (AFEE)
ΔT (0 to 31, unit = 50 ns)	AFEE number ($00 to $12)	
...		
PSD Id (0 in case of time out, $1D13 when RouteMode is set to 1 or 3)		*Element # m-1* (PSD)
ΔT (0 to 31, unit = 50 ns)	$13 (Number affected to PSD)	
AFEE Energy (0 in case of time out)		*Element # m* (AFEE)
ΔT (0 to 31, unit = 50 ns)	AFEE number ($00 to $12)	
Time of element #1 counted from beginning of 8 Hz cycle (0 to 2047, unit = 102.4 µs)	Number of elements = *m* (1 to 31)	*ME Label*

Fig. 16 Generic ME event format

ME element

Each element of a Multiple Event occupies 2 words, and can be either of the AFEE or PSD type. The type is defined by the detector number field value in the second word.

ME-AFEE element, 1st word

The first word of an AFEE element gives the Energy word as it was received from the AFEE. It is replaced by $0000 if the AFEE did not produce the energy before the delay defined by DFEE time out parameter TimeOutValMissNrj E7776 (abnormal condition).

ME-PSD element, 1st word

The 1st word of a PSD element shows two variants :

Variant 1 : Normal mode

In the normal mode of operation, the first word of a PSD element gives the PSD Id value, unmodified from what the PSD transmitted to the DFEE after its Time Tag. It is replaced by $0000 if the PSD did not produce the Id before the DFEE time out, as controlled by DFEE parameter TimeOutValMissNrj (E7776). The normal behaviour applies when the DFEE RouteMode parameter is set to its normal value NoRouting (E7848 = 0).

Variant 2 : Timing mode

In special modes devoted to AC gate timing, the PSD elements actually reflect the AC gate timing edges. Time out is not possible, and the PSD Id is forced to value $1D13. This value was chosen to avoid confusion with a normal Psd Id value. The timing behaviour applies when the DFEE RouteMode parameter is set to either RouteAcLeft (E7848 = 2) or RouteAcBoth (E7848 = 3).

ME-AFEE or PSD element, 2d word

The second word of an element gives the detector number ($00 to $12 for AFEE, $13 for PSD) and the fine time interval from the previous element (right-justified ΔT field). When the element is the first element of the Multiple Event, the ΔT value (0,1 or 3) uses a special code that helps resolve timing ambiguities that may arise on the first two elements when they occur at one clock of each other.

AFEE Energy format in MEs

When not replaced by $0000 due to time out, the AFEE Energy word shows the usual format :

Fig. 17 ME element : AFEE energy format

R : indicates the energy range where "0" = [20KeV .. 2MeV] and "1" = [2MeV .. 8MeV] ;

S : spare bit equals "1" ;

Energy : 14 bits for energy conversion value.

PSD Id format in MEs

When not replaced by $0000 due to time out or forced to $1D13 under special timing modes, the PSD Id word shows the following format :

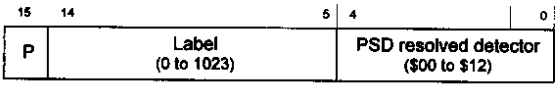

Fig. 18 ME element : PSD ID format

P : Processing flag. Set to "1" when the PSD processed this particular Time Tag, "0" otherwise.

Label : An increasing, modulo-1024 number transmitted by the PSD to the DFEE in the identifier word after each Time Tag.

PSD resolved detector : The number ($00 to $12) of the unique detector which triggered the PSD.

Remark : The $1D13 value provided during timing modes cannot be confused with a normal PSD Id because it sets the detector field to the normally unaffected value $13.

ΔT encoding for ME events

The ΔT field is encoded in a special way that helps analysing the mutual timing between any 2 AFEEs down to the 50 ns unit accuracy.

ΔT from second to last ME element

The ΔT field gives the time elapsed from the previous element in the Multiple Event, expressed in 50 ns units, in the range [0 .. 31].

ΔT of first ME element

A special value is used for the first element, to improve the timing resolution when the two first elements were detected at exactly one clock of each other.

ΔT first = 0 : The 2^d element is either simultaneous (and will then have itself ΔT = 0), or further away at a distance $n \geq 2$ (and will then have ΔT = n), or absent ;

ΔT first = 1 : The 2^d element is at a distance of exactly 1 unit, and the detector with the highest number was observed first ;

ΔT first = 3 : The 2^d element is at a distance of exactly 1 unit, and the detector with the lowest number was observed first.

When codes 1 or 3 are used, the order inside the ME does not necessarily reflect the initial order in time for the first 2 elements. However, the actual order in time may be recovered by analysing the numbers of the involved detectors.

Special ME cases

ME events with PSD element(s)

ME events with one PSD element will occur in the flight configuration. A typical case is a 2-AFEE multiple, when the energy of one of the involved AFEEs is not in the energy range of the PSD.

Analysis of the timing characteristics of both the AFEE and PSD electronics has shown that in the normal SPI configuration, ME events with more than one PSD element cannot

be produced, whatever the DFEE configuration of AssoMultWinSize (E7753) may be. However, ME events with more than one PSD element will be produced by the DFEE when configured for timing.

Pure PSD ME events

Pure PSD Multiple Events only contain PSD elements, and are maintained in the ME block when DFEE parameter EvtKeepPP (E7780) is set. For PP events, the degenerated case of a Multiple Event with one single element is possible.

PP ME events with one single PSD element will occur in the flight configuration. A typical scenario is when the PSD triggers on a detector for which the AFEE is in dead time due to previous activity.

Again, analysis of the timing characteristics of both the AFEE and PSD electronics has shown that in the normal SPI configuration, PP ME events with more than one element cannot be produced, whatever the DFEE configuration of AssoMultWinSize (E7753) may be. However, PP ME events with more than one element will be produced by the DFEE when configured for timing.

PE block format

The PE block structure is the following (read from left to right) :

PE SOB	PE body (n = 3*m words, size given by PE EOB.WC field)				PE EOB
(3 identical words)	PE event #1 (3 words)	PE Event #2 (3 words)	...	PE Event #m [last] (3 words)	(3 identical words)

Fig. 19 PE block format

The PE body gives the list of PE events encountered during the current 125 ms interval, ordered in increasing time of occurrence. Each following PE event occupies a fixed space (3 words).

PE block : SOB word

The SE SOB identification word has the following format :

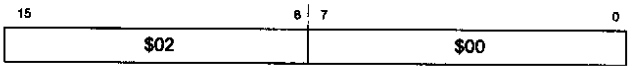

Fig. 20 PE SOB header word format

PE block : EOB word

The ME EOB word gives word count and status information corresponding to the SE events of the current 125 ms interval. The SE EOB word has the following format :

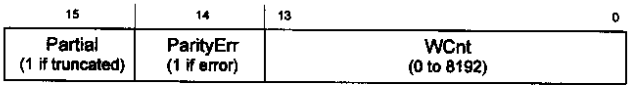

Fig. 21 PE EOB trailer word format

PE EOB Word Count

Gives the number of words in the sub-block, excluding itself. Valid range is [0 .. 8192] inclusive. A value of 8192 indicates that the DFEE had to discard PE events during the corresponding 125 ms interval, due to internal storage limitations ; in that case, the block contains the first 2731 PE events detected in this interval, and the last PE event is truncated. Word Count values higher than 8192 are invalid.

PE EOB Parity flag

Equals "1" when a parity error was detected in the ASIC. This status bit indicates that at least one of the PE events was altered during temporary storage in the DFEE.

PE EOB Partial flag

Equals "1" when the DFEE could not send all the PE data within the current 125 ms cycle, due to lack of space in the HSL packet. In that case, only the first PE events detected in this interval are given, to the extend of the available space. Due to the 3-word span of PE events, the last PE event is generally truncated when Partial is set. The partial flag may be set on the PE sub-block if the chosen HSL packet length (HslXferLength, E7783) is lower than 24597, and thus is not large enough to contain the sum of the largest SE, ME and PE sub-blocks.

PE block : Events

A PE event assembles the information coming from exactly one AFEE channel and from the PSD, when their respective Time Tags lie within a time distance which is smaller or equal to the DFEE AssoMultWinSize parameter (E7753). In the normal mode of operation, the assembly is only effective when the initial PSD Id that was given to the DFEE showed the P processing flag set. The assembly can be forced by setting the EvtForceProcPE (E7785) DFEE configuration parameter.

Generic PE event format

A PSD Event is a fixed format 3-word assembly. The first word gives PSD information and shows several variants, depending on DFEE configuration. The second word gives the AFEE energy. The third word gives the time of occurrence of the first Time Tag in the pair, and the channel number of the AFEE member.

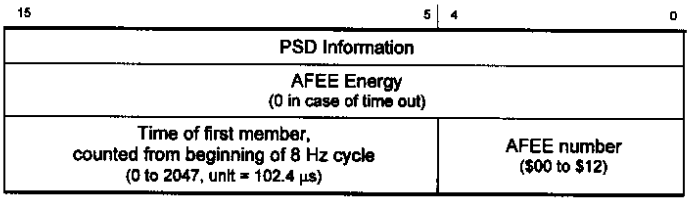

Fig. 22 Generic PE event format

AFEE Energy format in PEs

When not replaced by $0000 due to time out, the AFEE Energy word shows the usual format :

Fig. 23 AFEE energy format in PEs

R : indicates the energy range where "0" = [20KeV .. 2MeV] and "1" = [2MeV .. 8MeV] ;

S : spare bit equals "1" ;

Energy : 14 bits for energy conversion value.

Normal PE format

In the normal configuration (EvtSetTimeFmt, E7784 is false), the PSD information word is an unaltered copy of the PSD Id word if it was correctly received. If time out occurred on PSD Id reception, the PSD information word is replaced by $0000 - This can only occurs if EvtForceProcPE (E7785) is set.

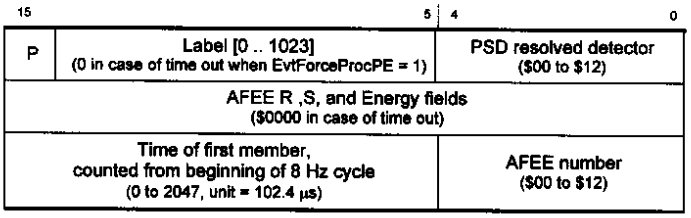

Fig. 24 Normal PE event format

P : Processing flag. Set to "1" when the PSD processed this particular Time Tag, "0" otherwise. PE events with the P flag cleared are not possible in the normal mode, unless EvtForceProcPE (E7785) is set.

Label : An increasing, modulo-1024 number transmitted by the PSD to the DFEE in the identifier word after each Time Tag.

PSD resolved detector : The number ($00 to $12) of the unique detector which triggered the PSD. The DFEE checks if the PSD resolved detector agrees with the number of the accompanying AFEE. If the numbers do not match for at least one PE event in a particular Time Frame, the corresponding warning bit is raised by the DFEE and installed in the relevant diagnostic packet. An accumulated 1-second persistence version of this warning bit appears in the HK #$04 packet.

Alternate PE event formats for timing

When doing detector timing alignment, the DFEE is configured to provide relative timing information for PE events (EvtSetTimeFmtPE, E7784, is set). Depending on the objective of the alignment, the DFEE configuration and the resulting PE format slightly differ.

PSD detector alignment

When doing PSD detector alignment, the DFEE RouteMode parameter is maintained to its default normal configuration value NoRouting (E7848 = 0). AFEE and PSD assembly may be forced by setting EvtForceProcPE (E7785). In that case, the PE event format is the following :

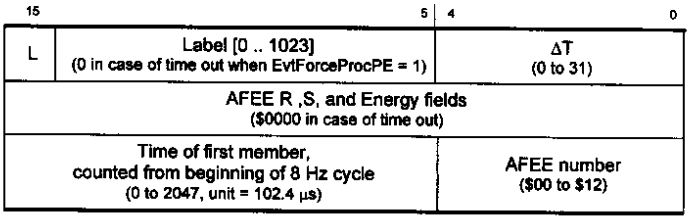

Fig. 25 Alternate PE event format for PSD timing alignment

L: sign of ΔT, equals "0" if the PSD Time Tag is anterior to the AFEE Time Tag ;

Label : An increasing, modulo-1024 number transmitted by the PSD to the DFEE in the identifier word after each Time Tag ;

ΔT : absolute value of the duration elapsed between the PSD Time Tag and the AFEE Time Tag, expressed in 50 ns units, in the range [0 .. 31].

AC detector alignment

When doing AC detector alignment, the DFEE RouteMode parameter is set to RouteAcLeft (E7848 = 2). The PSD element actually reflects the timing of the AC gate left edge. Time out may not occur, and the Label field is forced to value $0E8. AFEE and PSD assembly must be forced by setting EvtForceProcPE (E7785). In that case, the PE event format is the following :

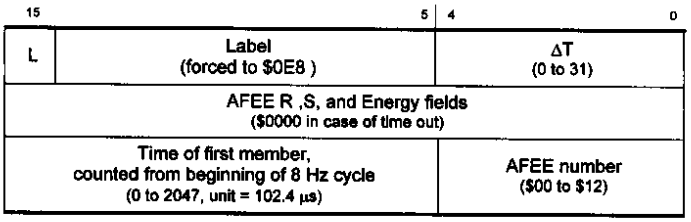

Fig. 26 Alternate PE event format for AC timing alignment

SP block format

The policy of SP storage in the DFEE does not follow the double buffering algorithm that SE, ME and PE blocks utilise. Since the energy values in the SP blocks are intended to populate energy spectrum histograms in the DPE, no event-per-event timing is required for them. The DFEE simply implements 19 independant asynchronous First In First Out buffers, for temporary storage of up to 8192 energy samples. On the HSL side, these buffers are simply extracted in a rotating loop. Due to this policy, it is guaranteed that the initial time order is maintained when delivering data corresponding to a particular SP block. However, it is not guaranteed that data will not be lost if the corresponding internal FIFO buffer gets exhausted, and the DFEE is not able to detect and signal such a situation. The SP block structure is the following (read from left to right) :

SP SOB	SP body (n words, size given by SP EOB.WC field)				SP EOB
(3 identical words)	SP energy #1 (1 word)	SP energy #2 (1 word)	...	SP Energy #n [last] (1 words)	(3 identical words)

Fig. 27 SP block format

The SP body gives the list of received AFEE energies ordered in increasing time of occurrence. Each following SP energy value occupies a single word.

SP block : SOB word

The SP SOB identification word has the following format :

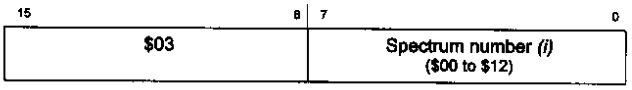

Fig. 28 SP SOB header word format

SP block : EOB word

The ME EOB word gives word count and status information for the SP segment that could be delivered in the current 125 ms interval. The SP EOB word has the following format :

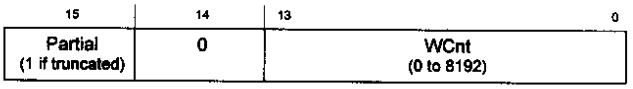

Fig. 29 SP EOB trailer word format

SP EOB Word Count

Gives the number of words in the block, excluding itself. Valid range is [0 .. 8192] inclusive. A value of 8192 indicates that the DFEE had to discard SP energies due to internal storage limitations. Word Count values higher than 8192 are invalid.

SP lack of EOB Parity flag

It was decided that a potential alteration of a SP energy value during temporary storage in the DFEE would have much less harmful effects for SP data than for SE, ME or PE blocks (the main potential source is a Single Event Upset of the electronic memory device induced by space radiation). Following this analysis, no parity protection was included on the SP block storage.

SP EOB Partial flag

Equals "1" when the DFEE could not extract all data from the last provided SP buffer in this HSL transfer, due to lack of space in the HSL packet. Transmission will resume from this very same SP index in the next HSL transfer.

SP block : Body

The SP block body simply gives the list of energy words received from the corresponding AFEE channel. The energy value is replaced by $0000 if the AFEE did not produce the energy before the delay defined by DFEE time out parameter TimeOutValMissNrj E7776 (abnormal condition). When not replaced by $0000 due to time out, the SP AFEE energy word shows the usual generic format.

Fig. 30 SP AFEE energy word format

R : indicates the energy range where "0" = [20KeV .. 2MeV] and "1" = [2MeV .. 8MeV] ;

S : spare bit equals "1" ;

Energy : 14 bits for energy conversion value.

DFEE SOFTWARE

This section describes the micro-system software.

Operation phases

The DFEE is set up, prepared, initialised, supervised and monitored by the DPE. The dedicated bi-directional, synchronous, character oriented link *Low Speed Link* (LSL) is used for these functions. The DFEE software follows the standard SPI subsystem sequence of operations.

Configuration

Entering CONF from STDBY, the DFEE waits until Cfg #$01, Cfg #$02 and Cfg #$03 packets have been received, in any order, before it accepts to move to OPER or DIAG. Later transitions to OPER/DIAG are accepted with no/partial reconfiguration if the system did not step back to STDBY in between. Return configuration packets HK #$01, #$02 and #$03 are available at any time.

Operation

When in OPER, both the ASIC and the software are scheduled by the 8 Hz reference clock, to define a sequence of following *Time Frames*. From the beginning of OPER mode, eight following *Time Frames* are grouped in a *second*. The DFEE software produces HK packets that reflect the activity of a particular *second* (HK #$04 to #$0B), or integrate activity over a longer time span (HK #$00). The DFEE software continuously checks that the ASIC keeps in sync, and that its internal configuration is not altered. Such situations are reflected in the HK #$00 and HK #$04 status words.

Diagnosis

DIAG mode is essentially identical to OPER mode, apart from the standard SPI command accept/reject matrix. Additional HK packets HK #$0C to #$12 are made available to help detailed diagnostic analysis (note that they are also available in OPER, but not requested by the IASW). The diagnostic packets give the detailed internal ASIC status for each time frame, and a copy of all ASIC internal registers for each second.

Maintenance

The DFEE software follows the standard maintenance procedures. LOAD and DUMP packets are used to read and change memory portions, and support the normal code patch sequences.

Software organisation

Once initialised, the software runs several concurrent tasks :

- Task 1 : *RcveLSL* : Command reception, verification and classification, followed by immediate execution or posting for deferred execution ;
- Task 2 : *XmitLSL* : ACK response or requested HK packet transmission ;
- Task 3 : *BuildHK* : HK packet preparation ;
- Task 4 : *ExeCMD* : Deferred command execution.

Tasks *RcveLSL* and *XmitLSL* show strong real time requirements connected to the synchronous nature of the LSL protocol. They are self-scheduled by the character reception and transmission interrupts from the LSL USART circuit. When *RcveLSL* has completed analysis of a command, it is able to classify it for immediate or deferred execution. Following this decision, it may either execute the command immediately or post it on a command stack for deferred execution by *ExeCMD*. *XmitLSL* is then triggered to initiate transmission of the resulting response. The half-duplex nature of the LSL protocol allows for simple chaining of *RcveLSL* and *XmitLSL*.

Task *BuildHK* is bounded to 8 Hz time frames for most HK packets (HK #$01, #$02 and #$03 are not in this category). It is scheduled by the 8 Hz clock interrupt. Status elaboration is also performed in this task.

Task *ExeCMD* is constrained at system level by maximum response time, but may be asynchronously handled through a command stack structure. It is implemented in a continuous background loop, always waiting for a new command on the stack, and then executing it.

Tasks *RcveLSL* and *XmitLSL* are high priority, followed by Task *BuildHK*. Task *ExeCMD* is lower priority (background). Performance analysis and code profiling were used to verify that Task *ExeCMD* still meets constraints in presence of worst case activity for Tasks *RcveLSL*, *XmitLSL* and *BuildHK*. Error detection is used to indicate anomalous command stack overflow.

Self-test

When powered up, the DFEE automatically runs a comprehensive self-test procedure.

The micro-system first checks the integrity of its Random Access Memory (RAM) structure. The program code is then copied from Read Only Memory (ROM) to RAM. The validity of this memory copy operation is verifed by cheksum analysis.

The micro-system then places the ASIC in a special self-test mode, where its internal operation is fully controlled by software. The software may trigger specific test-oriented functions, apply signal inputs and retrieve signal outputs from the ASIC, and advance its low level system clock operations.

Dedicated Built In Self Test structures are first activated to check the healthiness of the signal counter circuits and embedded Primary Object FIFO memory. A typical run is then launched, where, after ASIC configuration, 4 successive time frames are created with SE, ME and PE events, and HSL and statistics results are logged. The software checks the exact validity of the results against the expected value stored in ROM.

When self-test operates, the external inputs (detector signals, 8 Hz clock, HSL clock and envelope, ...) are ignored by the ASIC and internally replaced by software-controlled registers and actions. The normal HSL output signal is left to its inactive level (logical "1"), but the internal HSL bit stream is re-routed internally to the micro-controller ASIC control port data signal.

The self-test operation checks that almost all ASIC functions are healthy, and emulates almost all external interfaces. The only exception concerns the Energy/Id serial transactions from AFEE and PSD, for which only the envelope signal timing is emulated. Inside the envelope signal, the detailed bit-per-bit serial transmission of energy is not emulated, and a pre-defined Energy/Id value (proper to each channel) is produced.

After self-test, program control is transferred from ROM to RAM to speed up execution and to allow for future possible code patch operations.

CONFIGURATION PARAMETERS

This section describes the DFEE configuration parameters, and indicates their position in the configuration packets.

Cfg Parameter classes

The DFEE is configured by a series of parameters that control the behaviour of the micro-system software (attribute S, "*Software*"), the value of detector signal timing adjustments (attribute T, "*Timing*") and the selected variations in the ASIC algorithms (attribute A, "*Algorithm*"). This section describes each parameter independently. The full set of parameters required for operation is grouped into 3 configuration packets (Cfg #$01, Cfg #$02, and Cfg #$03, corresponding to TC #E0101, #E0102 and #E0103). Some hidden parameters are part of the configuration packets, but are not eligible for in-flight operation, and as such are not part of the SPI parameter database. The flight configuration implicitly uses default values for these parameters. They appear as hard-coded fixed values in the configuration packets.

TC #E0101 : Software parameters

DB	Parameter Name	Attr		
E7697	InhibAutotstCODE	S	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE software to execute the Autotest Code section at initialisation. When True, instructs the DFEE software to bypass the Autotest Code section at initialisation.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration]*
			Remarks	
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 1 Cfg #$01 [TC #E0101] : InhibAutotstCODE parameter

DB	Parameter Name	Attr		
E7698	InhibAutotstASIC	S	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE software to execute the Autotest ASIC section at initialisation. When True, instructs the DFEE software to bypass the Autotest ASIC section at initialisation.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration]*
			Remarks	
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 2 Cfg #$01 [TC #E0101] : InhibAutotstASIC parameter

DB	Parameter	Attr		Description
E7699	InhibAutotstRAM	S	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE software to execute the Autotest RAM section at initialisation. When True, instructs the DFEE software to bypass the Autotest RAM section at initialisation.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration]*
			Remarks	
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 3 Cfg #$01 [TC #E0101] : InhibAutotstRAM parameter

DB	Parameter	Attr		Description
E7700	NHslErrThrshold	S	*Type*	*4-bit unsigned integer*
			Function	*Defines the threshold used by the DFEE software to set the AlrtHslErr status bit*
			Purpose	*To detect abnormal conditions where the HSL transfer stays in error for several Time Frames in a particular 1-second interval*
			Remarks	*In very bursty conditions, the HSL transfer may get de-synchronised and fail for some Time Frames, although no error occured in the DFEE ASIC. It may be useful to wait for a certain error threshold before signalling a high severity (Alert) status.*
			Applicable range	*[0..15]*
			Mapping	*This threshold selects the minimal number of erroneous time frames per second required to signal the corresponding Alert. Normal range is [1..8]. Setting it to 0 will force the Alert for test purposes. Setting it to a value of 8 (or more) will disable the Alert even in case of continuous errors.*

Table 4 Cfg #$01 [TC #E0101] : NHslErrThrshold parameter

DB	Parameter	Attr		Description
E7695	EnHslErrAct	S	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE software to take no ASIC auto-reset action, even in presence of HSL stall. When True, instructs the DFEE software to launch the auto-reset ASIC restoring procedure when the number of HSL errors in the current 1-second interval is larger than (or equal to) the threshold defined by NhslErrThrshldAct (E7696).*
			Purpose	*[False :Test and/or integration ; True : nominal mode of operation].*
			Remarks	
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 5 Cfg #$01 [TC #E0101] : EnHslErrAct parameter

ID	Parameter	Acc		Description
E7696	NHslErrThrshldAct	S	*Type*	*4-bit unsigned integer*
			Function	*Defines the threshold used by the DFEE software to decide the auto-reset procedure.*
			Purpose	*To act upon abnormal conditions where the HSL transfer stays in error for several Time Frames in a particular 1-second interval*
			Remarks	*Anomalous conditions may cause the ASIC event processing logic to fail, and cause the HSL to stall. In such cases, the DFEE software can restore the corresponding ASIC section with minimal penalty on overall system operation. This parameter defines the error threshold to reach before the decision to restore is taken. Its value is interpreted in conjunction with the overriding EnHslErrAct (E7695) parameter.*
			Applicable range	*[0..15]*
			Mapping	*This threshold selects the minimal number of erroneous time frames per second required to trigger the action. Normal range is [1..8]. Setting it to 0 will force the decision to auto-reset for test purposes, if EnHslErrAct is set. Setting it to a value of 8 (or more) will impede the auto-reset decision, even if enabled by EnHslErrAct.*

Table 6 Cfg #$01 [TC #E0101] : NHslErrThrshldAct parameter

TC #E0102 : Algorithm variations & HSL settings

DFEE Parameter	Name	Attr.		Description
E7701	RstFrontTFrame	A	*Type*	*1-bit boolean*
			Function	*When False, The Front TFrame sub-unit of the DFEE ASIC, which controls the Run and Time Frame boundaries, is enabled for operation. When True, the Front TFrame unit is kept in the Reset state and does not operate, breaking the 8 Hz Time Frame pipeline.*
			Purpose	*In conjunction with all other Rst* flags, this flag defines the set of DFEE ASIC sub-units that should be left in Reset by the micro-controller software when going to operational mode. In the nominal mode of operation, all sub-units should be activated (Rst* = False). One or more sub-units may be left inactive during test phases, or possibly during flight in degraded mode, if declared defective.*
			Remark	*RstFrontTFrame should not be set in flight.*
			Appl. range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*
E7702	RstFrontVeto	A	*Idem, for the sub-unit in charge of the ACS signal processing (reception, alignment and veto generation). When RstFrontVeto is set, no internal (AFEE) or external (PSD) veto gate is produced, and the corresponding counts are 0. RstFrontVeto may be set if it is desirable to disable the external PSD veto gate for test purposes, but this will also shut off the internal AFEE AC gate. Better ACS control for AFEEs is given by the ACMode parameter.*	
E7702	RstFrontPSD	A	*Idem, for the sub-unit in charge of the PSD input signal processing. When RstFrontPSD is set, the PSD is detached from the system and the corresponding counts return 0. Better PSD control is given by EnAssoPSD.*	
E7704 to E7722	RstFrontAFEE_i	A	*Idem, for each of the 19 identical sub-units in charge of the AFEE$_i$ Time Tag and Energy signal processing [i =0..18]. When RstFront AFEE_i is set, AFEE channel #i is detached from the system and the corresponding counts return 0. Better AFEE$_i$ control is given by the EnAsso AFEE_i parameters.*	
E7723	RstCntVPSD	A	*Idem, for the sub-unit in charge of the ACS veto gate counts and dead time, and of the PSD signal count. Since the counting process is passive and has no effect on event processing, it is advisable to always leave this unit active.*	
E7724 to E7742	RstCntAFEE_i	A	*Idem, for the sub-unit in charge of the AFEE signal activity and dead time counts. Since the counting process is passive and has no effect on actual event processing, it is advisable to always leave this unit active.*	
E7743	RstAssoSm	A	*Idem, for the Finite State Machine in charge of the association of aligned time signals into Primary Objects. No HSL output will be produced by the DFEE if this parameter is set. This parameter should not be set in flight.*	
E7744	RstPobjSm	A	*Idem, for the Finite State Machine in charge of the temporary storage of Primary Objects. No HSL output will be produced by the DFEE if this parameter is set. This parameter should not be set in flight.*	
E7745	RstRcveSm	A	*Idem, for the Finite State Machine in charge of the ordered production of Energy and Id values. No HSL output will be produced by the DFEE if this parameter is set. This parameter should not be set in flight.*	
E7746	RstRcveSerial	A	*Idem, for the Finite State Machines in charge of the reception of the 20 serial word streams corresponding to AFEE energy and PSD Id. Due to forced timeout, all Energy an Id come with value 0 in the HSL output if this parameter is set. There is no particular reason to use this parameter in flight.*	

Table 7 Cfg #$02 [TC #E0102] : Rst* parameters (first part)

DB	Parameter	Attr	Description
E7747	RstAcqSm	A	*Idem, for the Finite State Machine in charge of the construction and storage of the SE, ME and PE and SP output tables. No HSL output will be produced by the DFEE if this parameter is set. This parameter should not be set in flight.*
E7748	RstDialSm	A	*Idem, for the Finite State in charge of the construction and synchronisation of the HSL output bitstream. No HSL output will be produced by the DFEE if this parameter is set. This parameter should not be set in flight.*

Table 8 Cfg #$02 [TC #E0102] : Rst* parameters (continued)

DB	Parameter	Attr	Description	
E7749	EnCntDtAfee	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to ignore the AFEE Nominal Time Tag when evaluating channel dead time ; When True, instructs the DFEE to include its contribution in the channel dead time measurement. Applies to all channels.*
			Purpose	*[False : Test, integration, investigation on dead time contributions; True : Nominal mode of operation].*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 9 Cfg #$02 [TC #E0102] : EnCntDtAfee parameter

DB	Parameter	Attr	Description	
E7750	EnCntDtAfeeSat	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to ignore the AFEE Saturating Time Tag when evaluating channel dead time ; When True, instructs the DFEE to include its contribution in the channel dead time measurement. Applies to all channels.*
			Purpose	*[False : Test, integration, investigation on dead time contributions; True : Nominal mode of operation].*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 10 Cfg #$02 [TC #E0102] : EnCntDtAfeeSat parameter

DB	Parameter	Attr	Description	
E7751	EnCntDtVetoGate	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to ignore the internal AFEE Veto Gate when evaluating channel dead time ; When True, instructs the DFEE to include its contribution in the channel dead time measurement. Applies to all channels.*
			Purpose	*[False : Test, integration, investigation on dead time contributions; True : Nominal mode of operation].*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 11 Cfg #$02 [TC #E0102] : EnCntDtVetoGate parameter

DB	Parameter	Attr	Description	
E7786	ToutScaleDropNrj	T	*Type*	*4-bit enumerated integer*
			Function	*Defines the time interval after which unconsumed AFEE Energies (or PSD Id) are automatically disposed of.*
			Purpose	*To avoid long term propagating errors in the association between Time Tag and Energy/Id, and to help identify faulty channels.*
			Applicable range	*[0..15]*
			Base unit	*ASIC Clk period (50 ns)*
			Mapping	*16 enumerated ToutScale values, used to define 16 interval values according to a $\log_2$ scale. Defines the interval value Tdrop = 2**(ToutScale) ASIC Clk periods. If not attached to a Time Tag on the corresponding channel, Energies/Id are automatically dropped sometime between Tdrop and 3*Tdrop after reception. Applies to all channels. [0 ⇒ 6.4µs, 1 ⇒ 25.6µs, 2 ⇒ 102µs, 3 ⇒ 205µs, 4 ⇒ 410µs,5 ⇒ 819µs, 6 ⇒ 1.64ms, 7 ⇒ 3.28ms, 8 ⇒ 6.55ms, 9 ⇒ 13.1ms, 10 ⇒ 26.2ms, 11 ⇒ 52.4ms, 12 ⇒ 105ms, 13 ⇒ 210ms, 14 ⇒ 839ms, 15 ⇒ ∞]*
			Remark	*ToutScaleDropNrj should be set to map an interval value larger than 64*tmiss [where tmiss is the time interval defined by ToutValMissNrj, E7776], to account for the worst case queuing time in the Primary Object FIFO memory. It should also be as small as possible to reduce the probability of false association cascades. For optimum control, the smallest value that exceeds 64*tmiss should be selected in the increasing enumerated list.*

Table 12 Cfg #$02 [TC #E0102] : ToutScaleDropNrj parameter

DB	Parameter	Attr	Description	
E7753	AssoMultWinSize	T	*Type*	*5-bit unsigned integer*
			Function	*Defines the maximum time interval between two following members in a composite object.*
			Purpose	*Used by the DFEE ASIC to aggregate events after front-end time alignment and to classify them into the 3 basic types : Single Events (SE), Psd Events (PE) and Multiple Events (ME). Reflects the maximum possible time spread between any 2 time tag signals originating from the same physical event. Both AFEE-AFEE pairs and AFEE-PSD pairs use this parameter.*
			Applicable range	*[3..30]*
			Base unit	*ASIC Clk period (50 ns) when LowResMode is False (flight configuration)*
			Mapping	*Linear scale [3 ⇒ 150 ns, .., 30 ⇒ 1500 ns]*

Table 13 Cfg #$02 [TC #E0102] : AssoMultWinSize parameter

OR	Parameter	Acc		Description
E7788	AcMode	A	*Type*	*2-bit enumerated integer*
			Function	*Specifies (alters) the use of the internal AFEE AC gate at the association point. Does not change the internal AC gate itself, and does not disable the PSD output gate.*
			Purpose	*Test and timing.* *The « AcOn » mode is the normal flight operation mode, where the internal AC veto gate is passed unaltered for association. Aligned AFEE Time Tags that stand inside the gate are considered vetoed at the association point.* *The « AcOff » mode –the veto gate is always considered inactive– and the « AcForced » mode –the veto gate is always considered active– are self-explanatory.* *The « AcInverted » mode reverts the gate function : events which stand outside the gate are considered vetoed at the association point. It may be used, in conjunction with RouteMode, to ease system timing verification.*
			Applicable range	*[0..3]*
			Remarks	*AcMode has no influence on the AC gate itself, and only alters the way it is seen by association. All other signal actions related to the AC gate are not affected by AcMode. The ACS gate and dead time counts and the AFEENvetoCnt values refer to the unaltered internal AC veto gate, and are not affected by AcMode. The PSD output AC gate is not affected ; it cannot be forced or shut off by AcMode.*
			Mapping	*0 : AcOff* *1 : AcOn* *2 : AcInverted* *3 : AcForced*

Table 14 Cfg #$02 [TC #E0102] : AcMode parameter

OR	Parameter	Acc		Description
E7787	LowResMode	A	*Type*	*1-bit boolean*
			Function	*When False, the association time window size (AssoMultWinSize, E7753) is expressed in ASIC Clk period units, and the DeltaTime interval between 2 following elements of a ME (or PE) object is also expressed in ASIC clock period units. When True, AssoMultWinSize and DeltaTime are both expressed in double Clk period units ; the association window may then be twice as large, at the cost of a coarser measurement of DeltaTime.*
			Purpose	*Test and timing ; [False : Nominal mode of operation ; True : Test and/or integration, direct measurement of large AC gates when used in conjunction with RouteMode].*
			Applicable range	*[0,1]*
			Remarks	*Ground timing campaigns could be accomplished without using low resolution mode. No use of this parameter is foreseen in flight.*
			Mapping	*[False : 0 ; True : 1].*

Table 15 Cfg #$02 [TC #E0102] : LowResMode parameter

DB	Parameter	Attr	Description	
E7756	AssoEnablePSD	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to ignore its internal PSD Time Tag while building objects. When True, instructs the DFEE to use its internal aligned PSD Time Tag.*
			Purpose	*[False : Test and/or integration, PSD off but still counted ;True : Nominal mode of operation].*
			Remarks	*This control gives the possibility to remove the PSD from the object association process, while still monitoring its activity. If AssoEnablePSD is cleared, PSD will not contribute to the HSL bit stream, but PSD signal monitoring information will still be available in the corresponding LSL HK packet.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 16 Cfg #$02 [TC #E0102] : AssoEnablePSD parameter

DB	Parameter	Attr	Description	
E7757 to E7775	AssoEnableAFEE_i	A	*Type*	*Vector(i = 0 to 18) of 1-bit boolean*
			Function	*When False, instructs the DFEE to ignore its internal AFEE Time Tag for channel #i while building objects. When True, instructs the DFEE to use its internal aligned AFEE Time Tag for channel #i.*
			Purpose	*[False : Test and/or integration, AFEE #i off but still counted ;True : Nominal mode of operation].*
			Remarks	*This control gives the possibility to remove a AFEE channel from the object association process, while still monitoring its activity. If AssoEnableAFEE_i is cleared, AFEE #i will not contribute to the HSL bit stream, but signal monitoring information for this channel will still be available in the corresponding LSL HK packet.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 17 Cfg #$02 [TC #E0102] : AssoEnableAFEE_i parameters

ID	Parameter	Attr		Description
E7776	ToutValMissNrj	T	*Type*	*5-bit unsigned integer*
			Function	*Defines the time interval after which the DFEE, once a Time tag has been observed, ceases to wait for the corresponding Energy/Id serial transaction. If time out is declared, the missing Energy/Id value is replaced by special value $0000.*
			Purpose	*To guarantee non-blocking forward progress in the DFEE event processing, to secure the even/odd Time Frame buffer swap for HSL output tables, and to help identify faulty channels.*
			Remarks	*This parameter is applied identically to all Energy and Id transactions. It should be adjusted to cope with the serial transmission which shows the higher latency (amongst AFEE, Energy and PSD Id transactions).*
			Applicable range	*[0,31]*
			Base unit	*128*ASIC Clk periods (6400 ns)*
			Mapping	*Linear scale in 128 ASIC clk increments. Defines the internal value Tmiss =(26+128*(ToutValMissNrj + 1)) ASIC Clk periods. A Energy/Id Timeout condition is declared if an isolated Time Tag is not followed by a complete serial Energy/Id transaction within Tmiss ASIC Clk periods. The actual waiting time is increased when Time Tag pile-up occurs.*

Table 18 Cfg #$02 [TC #E0102] : ToutValMissNrj parameter

ID	Parameter	Attr		Description
E7777	EvtKeepSE	A	*Type*	*1-bit boolean*
			Function	*When False, prevents the DFEE from writing SE objects into the HSL SE output table ; if the companion flag EvtKeepPE is True, the SE table may still receive the AFEE portion of PE events which are lately reclassified due to the absence of the PSD « Processed » flag. When True, enables the DFEE to write SE objects into the HSL SE output table.*
			Purpose	*[False : Test and/or integration, high rate flight condition ; True : Nominal mode of operation].*
			Remarks	*In case of very high rate situation, this control might be used in conjunction with EvtKeepPE, EvtKeepME and SpeStorageMode to favour spectra transmission of non-vetoed SEs.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 19 Cfg #$02 [TC #E0102] : EvtKeepSE parameter

ID	Parameter	Att		Description
E7778	EvtKeepME	A	*Type*	*1-bit boolean*
			Function	*When False, prevents the DFEE from writing ME objects into the HSL ME output table. When True, enables the DFEE to write ME objects into the HSL ME output table.*
			Purpose	*[False : Test and/or integration, high rate flight condition ; True : Nominal mode of operation].*
			Remarks	*In case of very high rate situation, this control might be used in conjunction with EvtKeepSE, EvtKeepPE and SpeStorageMode to favour spectra transmission of non-vetoed SEs.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 20 Cfg #$02 [TC #E0102] : EvtKeepME parameter

ID	Parameter	Att		Description
E7779	EvtKeepPE	A	*Type*	*1-bit boolean*
			Function	*When False, prevents the DFEE from writing PE objects into the HSL PE output table. When True, enables the DFEE to write PE objects into the HSL PE output table.*
			Purpose	*[False : Test and/or integration, high rate flight condition ; True : Nominal mode of operation].*
			Remarks	*In case of very high rate situation, this control might be used in conjunction with EvtKeepSE, EvtKeepME and SpeStorageMode to favour spectra transmission of non-vetoed SEs.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 21 Cfg #$02 [TC #E0102] : EvtKeepPE parameter

ID	Parameter	Att		Description
E7780	EvtKeepPP	A	*Type*	*1-bit boolean*
			Function	*When False, prevents the DFEE from writing PSD only (Pure Psd) objects into the HSL ME output table. When True, enables the DFEE to write PP objects into the HSL ME output table, whatever the value of the EvtKeepME companion flag.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration, PSD vs AFEE dead time investigation].*
			Remarks	*Single element Pure Psd ME objects are rather common, generally indicating a PE where the AFEE member could not be measured due to dead time incurred by a previous event. Multiple element Pure Psd ME objects are excluded because the PSD repetition rate is too low compared to the AssoMultWinSize signal correlation window.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 22 Cfg #$02 [TC #E0102] : EvtKeepPP parameter

DB	Parameter	Attr	Description	
E7781	EvtForceNVeto	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to write objects into the SE, ME and PE output tables only if they were not vetoed at association. When True, forces the DFEE to write the objects into the HSL SE, ME, and PE output tables with no consideration of their association veto status. In either case, storage depends also on EvtKeepSE, EvtKeepME, EvtKeepPE and EvtKeepPP. Storage into Spectra is independently controlled by SpeStorageMode, and may depend on the association veto status even when EvtForceNVeto is True.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration].*
			Remarks	*The association veto status depends itself on the AcMode parameter value.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 23 Cfg #$02 [TC #E0102] : EvtForceNVeto parameter

DB	Parameter	Attr	Description	
E7782	SpeStorageMode	A	*Type*	*2-bit enumerated integer*
			Function	*Specifies which selection criterion is applied to define the subset of AFEE energies that are written into Spectra tables SP0 to SP18. Spectra can be disabled, filled from vetoed objects only, filled from SE non vetoed objects only, or from any object regardless of its veto status.*
			Purpose	*Scientific/ technical decision ; "SE Non Vetoed Only" mode is intended to address high rate situations where SE, ME and PE are disabled and only spectra are returned to earth. "None" mode might be used for improve DPE processing performance when spectra are not required.*
			Remarks	*If at least one AFEE Time Tag is vetoed in an object, the veto criterion is globally applied to all other energies included in this object.*
			Applicable range	*[0..3]*
			Mapping	*0 : None* *1 : Vetoed Only* *2 : SE Non Vetoed Only* *3 : All*

Table 24 Cfg #$02 [TC #E0102] : SpeStorageMode parameter

DB	Parameter	Attr	Description	
E7783	HslXferLength	T	*Type*	*16-bit unsigned integer*
			Function	*Defines the number of 16-bit words to produce for each HSL transfer.*
			Purpose	*To produce a correctly formatted HSL bit stream towards the DPE. This number must be pre-negotiated and applied to both DPE and DFEE.*
			Remarks	*The bit stream is incorrect if HslXferLength is less than 9.*
			Applicable range	*[9..65535]*
			Base unit	*16-bit word*
			Mapping	*Linear scale*

Table 25 Cfg #$02 [TC #E0102] : HslXferLength parameter

DB	Parameter	Attr		Description
E7784	EvtSetTimeFmtPE	T	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to write objects into the PE output table with the PSD Id unchanged. When True, instructs the DFEE to modify the PSD Id to include the difference in arrival time between the AFEE Time Tag and the PSD Time Tag ; The « Evt Label » subfield remains unchanged, the « Processed » flag is replaced by the « PSD Last » flag, and the « Detector Address » subfield is replaced by the «Delta Time » unsigned 5-bit binary number giving the measured interval length in ASIC clock period units.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration].*
			Remarks	*This parameter is intended for AFEE vs. PSD timing. In conjunction with RouteMode, it also enables ACS vs. AFEE timing measurements.*
			Applicable range	*[0, 1]*
			Mapping	*[False : 0 ; True : 1]*

Table 26 Cfg #$02 [TC #E0102] : EvtSetTimeFmtPE parameter

DB	Parameter	Attr		Description
E7785	EvtForceProcPE	A	*Type*	*1-bit boolean*
			Function	*When False, instructs the DFEE to use the « Processed » flag in the Psd Id to decide whether to write the corresponding PE candidate objects into the PE output table (when « Processed » is True) or to discard the PSD info and re-route the AFEE info to the SE output table (when « Processed » is False). When True, forces the DFEE to write PE candidate objects into the PE output table, whatever the value of the « Processed » flag.*
			Purpose	*[False : nominal mode of operation ; True : Test and/or integration].*
			Remarks	*In conjunction with EvtSetTimeFmtPE and RouteMode, this parameter enables ACS vs. AFEE timing measurements, by guaranteeing that ACS-derived components stick in the PE table.*
			Applicable range	*[0,1]*
			Mapping	*[False : 0 ; True : 1].*

Table 27 Cfg #$02 [TC #E0102] : EvtForceProcPE parameter

TC #E0103 : Detector timing and Veto Gate settings

DES	Parameter	Attr.		Description
E7800 to E7818	DelayAfeeSat_i	T	*Type*	*Vector(i = 0 to 18) of 4-bit unsigned integer elements.*
			Function	*Defines the delay applied to the AFEE saturated Time Tag (channel #i) before contributing to the evaluation of dead time for this channel.*
			Purpose	*To compensate for systematic differential delays, in order to provide a accurate measurement of dead time*
			Remarks	*The optimal value may be obtained by event timing analysis, using RouteMode and AcMode controls.*
			Applicable range (elt)	*[0..15]*
			Base unit	*ASIC Clk period (50 ns)*
			Mapping	*Linear scale. Delay expressed in ASIC Clk units. The internal delay actually applied is 5+DelayAfeeSat_i ASIC Clk periods.*

Table 28 Cfg #$03 [TC #E0103] : DelayAfeeSat_i parameters

DES	Parameter	Attr.		Description
E7820 to E7838	DelayAfeeTT_i	T	*Type*	*Vector(i = 0 to 18) of 4-bit unsigned integer elements.*
			Function	*Defines the delay applied to the AFEE Time Tag (channel #i) to obtain a uniformly aligned set of signals at the association point, where objects are built up.*
			Purpose	*To compensate for systematic differential delays, in order to reach the object association point with aligned Time Tags and to provide a accurate measurement of dead time.*
			Remarks	*The optimal value may be obtained by event timing analysis, using AcMode control.*
			Applicable range (elt)	*[0..15]*
			Base unit	*ASIC Clk period (50 ns)*
			Mapping	*Linear scale. Delay expressed in ASIC Clk units. The internal delay actually applied is 5+DelayAfeeSat_i ASIC Clk periods.*

Table 29 Cfg #$03 [TC #E0103] : DelayAfeeTT_i parameter

E7840	DelayVetoFrst	T	*Type*	*4-bit unsigned integer*
			Function	*Defines the delay applied to the ACS anti-coincidence signal before issuing the corresponding Psd Veto Gate towards the PSD.*
			Purpose	*To adjust the time characteristics of the Veto Gate applied to the PSD.*
			Remarks	*The optimal value may be obtained by event timing analysis.*
			Applicable range	*[0..15]*
			Base unit	*ASIC Clk period (50 ns)*
			Mapping	*Linear scale. Delay expressed in ASIC Clk units. The actual round trip electrical delay from ACS (in) to PsdVetoGate (out) is (3+DelayVetoFrst) ASIC Clk periods+Trcve+Txmit+ErrSync, where Trcve and Txmit are the delays associated with the ACS differential receiver and PsdVetoGate differential transmitter, and ErrSync is the time quantization error induced by the synchronisation of the ACS signal with the DFEE internal clock.*

Table 30 Cfg #$03 [TC #E0103] : DelayVetoFrst parameter

E7841	DelayVetoScnd	T	*Type*	*6-bit unsigned integer*
			Function	*Defines the additional delay applied to the ACS anti-coincidence signal before building the internal veto gate applied to the aligned AFEE Time Tags at the association point.*
			Purpose	*To adjust the time characteristics of the internal Veto Gate applied in common to all AFEE channels.*
			Remarks	*The optimal value may be obtained by event timing analysis, using AcMode control.*
			Applicable range	*[0..63]*
			Base unit	*ASIC Clk period (50 ns)*
			Mapping	*Linear scale. Delay expressed in ASIC Clk units.The internal delay applied from ACS(in) to the AC Veto Gate at association is 3+DelayVetoFrst+ DelayVetoScnd ASIC Clk periods*

Table 31 Cfg #$03 [TC #E0103] : DelayVetoScnd parameter

ID	Parameter	Attr		Description
E7842	XtndThresh	T	Type	8-bit unsigned integer
			Function	Defines the threshold used to discriminate nominal and saturating ACS anti-coincidence signals. If the ACS pulse width is smaller than XtndThresh+2, the ACS veto signal is declared as nominal ; otherwise, it is declared as saturating. Depending on the classification, the applied veto gate is formed by extending the original pulse by a value which is function of either XtndGateBelow or XtndGateAbove.
			Purpose	To produce a larger, adjustable Veto Gate when the ACS encountered a saturating condition.
			Remarks	The optimal value may be obtained by event timing analysis, using AcMode control.
			Applicable range	[0..255]
			Base unit	ASIC Clk period (50 ns)
			Mapping	Linear scale. Threshold expressed in ASIC Clk units. The actually applied threshold is (XtndThresh+2) ASIC Clk periods.

Table 32 Cfg #$03 [TC #E0103] : XtndThresh parameter

ID	Parameter	Attr		Description
E7843	DlyPsd	T	Type	5-bit unsigned integer
			Function	Defines the delay applied to the PSD Time Tag to obtain a uniformly aligned set of signals at the association point, where objects are built up.
			Purpose	To compensate for systematic differential delays, in order to reach the object association point with aligned Time Tags.
			Remarks	The optimal value may be obtained by event timing analysis.
			Applicable range	[0..31]
			Base unit	ASIC Clk period (50 ns)
			Mapping	Linear scale. Delay expressed in ASIC Clk units. The delay actually applied is 5+DlyPsd.

Table 33 Cfg #$03 [TC #E0103] : DlyPsd parameter

ID	Parameter	Attr		Description
E7844	XtndGateAbove	T	Type	8-bit unsigned integer
			Function	Defines the extension applied to the ACS anti-coincidence signal when it is declared as saturating (pulse width above or equal to threshold).
			Purpose	To produce a larger, adjustable Veto Gate when the ACS encountered a saturating condition.
			Remarks	The optimal value reflects the ACS recovery time after saturation.
			Applicable range	[0..255]
			Base unit	4*ASIC Clk periods (200 ns)
			Mapping	Linear scale, 4-Clk steps. The extension is adjusted in 4*ASIC Clk increments. The internal extension actually applied is ((XtndGateAbove*4)-1) ASIC Clk periods.

Table 34 Cfg #$03 [TC #E0103] : XtndGateAbove parameter

DB	Parameter	Act		Description
E7845	XtndGateBelow	T	*Type*	*8-bit unsigned integer*
			Function	*Defines the extension applied to the ACS anti-coincidence signal when it is declared as nominal (pulse width below threshold).*
			Purpose	*To produce an adjustable Veto Gate when the ACS is in nominal condition.*
			Remarks	*The optimal value may be obtained by event timing analysis.*
			Applicable range	*[0..255]*
			Base unit	*ASIC Clk periods (50 ns)*
			Mapping	*Linear scale. Extension expressed in ASIC Clk units. The extension actually applied is (XtndGateBelow-1) ASIC Clk periods.*

Table 35 Cfg #$03 [TC #E0103] : XtndGateBelow parameter

DB	Parameter	Act		Description
E7848	RouteMode	A	*Type*	*2-bit enumerated integer*
			Function	*When « NoRouting » is selected, the AFEE, PSD and AC internal signal paths in the DFEE ASIC are normally connected. When « RouteSat » is selected, the aligned AFEE Saturating Time Tags replace the Nominal Time Tags at the Association location. When « RouteAcLeft » is selected, the actual PSD signal is disconnected from the Association and a fake PSD Time Tag is created at the leading (left) edge of the internal AC Veto Gate. When « RouteAcBoth » is selected, the actual PSD signal is disconnected from the Association and two fake PSD Time Tags are created : one at the leading (left) edge of the internal AC Veto Gate, and the other one at the trailing (right) edge of the internal AC Veto Gate.*
			Purpose	*« NoRouting » : nominal mode of operation ; « RouteSat » : used in integration to align the AFEE Saturating TimeTag signal paths, after ME or PE Delta Time analysis; « RouteAcLeft » and « RouteAcBoth » : used in integration to align the ACS signal path, after ME or PE Delta Time analysis.*
			Remarks	*Note that the AC veto function is not affected by the RouteMode selection. When RouteAcLeft or RouteAcBoth is selected, the fake PSD Time Tag(s) is(are) accompanied by a special constant Id ($1D13).*
			Applicable range	*[0..3]*
			Mapping	*0 : NoRouting* — *1 : RouteSat* *2 : RouteAcLeft* — *3 : RouteAcBoth*

Table 36 Cfg #$03 [TC #E0103] : RouteMode parameter

CONFIGURATION PACKETS

This section describes the detailed bit-level assembly of the configuration packets. Configuration parameters are referred by their name. See section « Configuration parameters » for a a detailed description of each parameter.

Cfg #$01 : Software parameters

The Cfg #$01 configuration block defines the variations on the DFEE microsystem software behaviour.

Byte Num	Bit #7	Bit #6	Bit #5	Bit #4	Bit #3	Bit #2	Bit #1	Bit #0
1	User's Acknowledge							
2	$01							
3	0	InhibAutotst CODE	InhibAutotst ASIC	InhibAutotst RAM	NHslErrThrshold			
4	DiagFreq							
5	0	0	0	EnHslErrAct	NHslErrActThrshold			
6..29	0							
30	CheckSum							

Table 37 Cfg #$01 format (TC #E0101)

Autotest conditions

The various autotest features may be independently inhibited, for the following Restart or Watchdog operations.

HSL error handling

The *AlertHslErr* bit will be set in HK #$00 and HK #$04 when, in any particular second, the number of HSL packets in error is greater or equal to *NHslErrThrshold*. Note that if *NHslErrThrshold* is set to 0, the alert will always be set (a feature implemented to allow for testable code). Any number greater than 8 will disable the Alert (*AlertHslErr* will always be 0, even in case of error).

The *EnHslErrAct* configuration bit globally enables/disables the automatic ASIC reset procedure. This control was included for redundancy, as a protection against possible (SEU induced) alterations of the *NHslErrActThrshold* field.

The *AlertHslErrAct* bit is set in HK #$04 when, in any particular second, the DFEE software takes the action to reset the ASIC. The auto-reset action is taken if *EnHslErrAct* is set and if the number of HSL packets in error is greater or equal to *NHslErrActThrshold*. Note that if *NHslErrActThrshold* is set to 0, the software continuously and endlessly resets

the ASIC (a feature implemented to allow for testable code). No auto-reset action can be taken if *NHslErrActThrshold* is greater than 8 (even if *EnHslErrAct* is set).

See section « Status and Errors » for details on the software behaviour in presence of HSL errors.

Cfg #$02 : Algorithm variations & HSL settings

This block corresponds to DFEE ASIC internal registers AR_1 (CtlRst) and AR_5 (CfgAlgo).

Byte Num	Bit #7	Bit #6	Bit #5	Bit #4	Bit #3	Bit #2	Bit #1	Bit #0
1	"C"							
2	$02							
3	RstFrontTFrame	RstFrontVeto	RstFrontPsd	RstFrontAfee18	RstFrontAfee17	RstFrontAfee16	RstFrontAfee15	RstFrontAfee14
4	RstFrontAfee13	RstFrontAfee12	RstFrontAfee11	RstFrontAfee10	RstFrontAfee9	RstFrontAfee8	RstFrontAfee7	RstFrontAfee6
5	RstFrontAfee5	RstFrontAfee4	RstFrontAfee3	RstFrontAfee2	RstFrontAfee1	RstFrontAfee0	RstCountVPsd	RstCountAfee18
6	RstCountAfee17	RstCountAfee16	RstCountAfee15	RstCountAfee14	RstCountAfee13	RstCountAfee12	RstCountAfee11	RstCountAfee10
7	RstCountAfee9	RstCountAfee8	RstCountAfee7	RstCountAfee6	RstCountAfee5	RstCountAfee4	RstCountAfee3	RstCountAfee2
8	RstCountAfee1	RstCountAfee0	RstAssoSm	RstPobjSm	RstRcveSm	RstRcveSerial	RstAcqSm	RstDialSm
9	EnCntDtAfee	EnCntDtAfeeSat	EnCntDtVetoGate	LowResMode	ToutScaleDropNrj 3 (MSB)	ToutScaleDropNrj 2	ToutScaleDropNrj 1	ToutScaleDropNrj 0 (LSB)
10	AssoMultWinSize 4 (MSB)	AssoMultWinSize 3	AssoMultWinSize 2	AssoMultWinSize 1	AssoMultWinSize 0 (LSB)	AcMode1 (MSB)	AcMode0 (LSB)	AssoEnablePsd
11	AssoEnableAfee 18	AssoEnableAfee 17	AssoEnableAfee 16	AssoEnableAfee 15	AssoEnableAfee 14	AssoEnableAfee 13	AssoEnableAfee 12	AssoEnableAfee 11
12	AssoEnableAfee 10	AssoEnableAfee 9	AssoEnableAfee 8	AssoEnableAfee 7	AssoEnableAfee 6	AssoEnableAfee 5	AssoEnableAfee 4	AssoEnableAfee 3
13	AssoEnableAfee 2	AssoEnableAfee 1	AssoEnableAfee 0	ToutValMissNrj 4 (MSB)	ToutValMissNrj 3	ToutValMissNrj 2	ToutValMissNrj 1	ToutValMissNrj 0 (LSB)
14	EvtKeepSE	EvtKeepME	EvtKeepPE	EvtKeepPP	EvtSetTimeFmtPE	EvtForceNVeto	EvtForceProcPE	SpeStorageMode 1 (MSB)
15	SpeStorageMode 0 (LSB)	1	1	0	HslXferLength 15 (MSB)	HslXferLength 14	HslXferLength 13	HslXferLength 12
16	HslXferLength 11	HslXferLength 10					HslXferLength 5	HslXferLength 4
17	HslXferLength 3	HslXferLength 2	HslXferLength 1	HslXferLength 0 (LSB)	0	0	1	1
18	0	$00						
19	0							
20	CheckSum							

Table 38 Cfg #$02 format (TC #E0102)

Multiple-bit numeric fields (**ToutScaleDropNrj**, **AssoMultWinSize**, **ToutValMissNrj**, **SpaStorageMode**, **HsDXferLength**) are expressed in unsigned integer format. All other fields are single-bit fields (e.g. **RstRcvaSm**) or collections of single-bit fields (e.g. **RstFrontAfee**) ; the name of the corresponding parameter indicates the configuration taken when bit is set to " 1 ".

Refer to section « Configuration Parameters » for a detailed discussion on the usage of each parameter.

Cfg #$03 : Detector timing and Veto Gate settings

This block corresponds to DFEE ASIC internal registers AR_3 (CfgFrontAfee) and AR_4 (CfgFrontVPsd).

Byte No.								
1	"C"							
2	$03							
3	MSB	DelayAfeeTT_18		LSB	MSB	DelayAfeeSat_18		LSB
4	MSB	DelayAfeeTT_17		LSB	MSB	DelayAfeeSat_17		LSB
5..19	...				...			
20	MSB	DelayAfeeTT_1		LSB	MSB	DelayAfeeSat_1		LSB
21	MSB	DelayAfeeTT_0		LSB	MSB	DelayAfeeSat_0		LSB
22	DelayVetoFrst 3 (MSB)	DelayVetoFrst 2	DelayVetoFrst 1	DelayVetoFrst 0 (LSB)	DelayVetoScnd 5 (MSB)	DelayVetoScnd 4	DelayVetoScnd 3	DelayVetoScnd 2
23	DelayVetoScnd 1	DelayVetoScnd 0 (LSB)	XtndThresh 7 (MSB)	XtndThresh 6	XtndThresh 5	XtndThresh 4	XtndThresh 3	XtndThresh 2
24	XtndThresh 1	XtndThresh 0 (LSB)	XtndGateAbove 7 (MSB)	XtndGateAbove 6	XtndGateAbove 5	XtndGateAbove 4	XtndGateAbove 3	XtndGateAbove 2
25	XtndGateAbove 1	XtndGateAbove 0 (LSB)	XtndGateBelow 7 (MSB)	XtndGateBelow 6	XtndGateBelow 5	XtndGateBelow 4	XtndGateBelow 3	XtndGateBelow 2
26	XtndGateBelow 1	XtndGateBelow 0 (LSB)	DelayPsd 4 (MSB)	DelayPsd 3	DelayPsd	DelayPsd	DelayPsd LSB	1
27	1	1	RouteMode 1 (MSB)	RouteMode 0 (MSB)	0			
28	CheckSum							

Table 39 Cfg #$03 format (TC #E0103)

All fields in this block are multiple-bit numeric fields. They are expressed in unsigned integer format.

HOUSEKEEPING PARAMETERS

For each second of operation, the DFEE produces a set of technical and scientific housekeeping parameters. Most of them are directly produced by the DFEE processing ASIC circuit, and are simply forwarded by the micro-system. The periodic (1-second) status information is calculated by the DFEE on-board software and reflected into the HK #$04 Status bytes. The accumulated status information is also calculated by the DFEE on-board software and produced in the HK #$00 Status bytes.

The following tables gives the name, size, function and attributes of these parameters.

HK Parameter description

HK parameters give information concerning Run progression (attribute R), clock time monitoring (attribute T) and, for the most part, signal and dead time counts (attribute C).

Parameter	Attr		
NSecFromStart	R	*Type*	*16-bit unsigned integer*
		Function	*Gives the cardinal number of the corresponding second, starting from the beginning of OPER/DIAG mode.*
		Purpose	*To obtain a consistent numbering of the HK packets in an operational or diagnostic run*
		Remarks	*Created by the micro-system software*
		Applicable range	*[0..65535]*
		Boundary conditions	*The first second after the start of OPER/DIAG mode receives number 1. The number rolls over to 0 if the run lasts more than 65536 seconds.*

Table 40 NSecFromStart parameter

Parameter	Attr		
TimeClkMonTF_i	T	*Type*	*Vector(i = 1 to 8) of 24-bit unsigned integer elements*
		Function	*Gives the number of ASIC Clk periods that were observed during Time Frame #i of the corresponding second (the Time Frame intervals are delimited by the following active edges of the Clk8Hz signal).*
		Purpose	*To monitor the differential drift between the Clk8Hz signal and the ASIC 20 MHz Clock oscillator*
		Base unit	*ASIC clock period (50 ns)*
		Applicable range (elt)	*[0..16777215]*
		Remark	*When adding the TimeClkMonTF_i values over a given period, the exact number of clocks produced over this period is obtained. Short-term and/or long-term drifts can thus be derived.*

Table 41 TimeClkMonTF_i parameters

Parameter	Att		
CntAfeeTT_i	C	*Type*	*Vector(i = 0 to 18) of 16-bit unsigned integer elements*
		Function	*Gives the number of nominal Time Tags seen on AFEE channel #i in the corresponding second*
		Purpose	*To monitor the activity of the corresponding AFEE channel, and allow consistency checks with the SE, ME, PE and SP scientific data.*
		Applicable range (elt)	*[0..65535]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This should not happen under normal circumstances, because the counter range exceeds the maximum rate of the AFEE Time Tag signal.*

Table 42 CntAfeeTT_i parameters

Parameter	Att		
CntAfeeTTSat_i	C	*Type*	*Vector(i = 0 to 18) of 16-bit unsigned integer elements*
		Function	*Gives the number of Saturating Time Tags seen on AFEE channel #i in the corresponding second.*
		Purpose	*To monitor the activity of the corresponding AFEE channel, and allow consistency checks with the corresponding dead time value.*
		Applicable range (elt)	*[0..65535]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This should not happen under normal circumstances, because the counter range exceeds the maximum rate of the AFEE Saturating Time Tag signal.*

Table 43 CntAfeeTTSat_i parameters

Parameter	Att		
CntAfeeNVeto_i	C	*Type*	*Vector(i = 0 to 18) of 16-bit unsigned integer elements.*
		Function	*Gives the number of Non vetoed nominal Time Tags seen on AFEE channel #i in the corresponding second.*
		Purpose	*To monitor the activity of the corresponding AFEE channel, and allow consistency checks with the SE, ME, PE and SP scientific data.*
		Applicable range (elt)	*[0..65535]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This should not happen under normal circumstances, because the counter range exceeds the maximum rate of the AFEE Time Tag signal.*

Table 44 CntAfeeNVeto_i parameters

Parameter	A/D		Description
CntAfeeDtime_i	C	Type	Vector(i = 0 to 18) of 24-bit unsigned integer elements.
		Function	Gives the duty cycle of the signal obtained by OR-ing the nominal Time Tag, the saturating Time Tag and the global internal Veto Gate, after signal alignment.
		Purpose	To monitor the activity of the corresponding AFEE channel, and check consistency with the other count values (the various counts produced for this channel and the ACS related counts).
		Base unit	ASIC Clock period * 2 (100 ns)
		Applicable range (elt)	[0..16777215]
		Mapping	The value is not normalised. A fractional dead time value may be obtained by multiplying it by 2/sum(TimeClkMonTF_i), over the same time period.
		Boundary conditions	The value cannot saturate over a second at the 20 MHz ASIC operating frequency.

Table 45 CntAfeeDtime_i parameters

Parameter	A/D		Description
CntPsdTT	C	Type	16-bit unsigned integer.
		Function	Gives the number of PSD Time Tags seen in the corresponding second
		Purpose	To monitor the activity of the PSD, and allow for consistency checks with the SE, ME and PE scientific data
		Applicable range	[0..65535]
		Boundary conditions	The upper bound value is returned if the counter goes into saturation during the counting interval. This should not happen under normal circumstances, because the counter range exceeds the maximum rate of the qualified PSD Time Tag signal.

Table 46 CntPsdTT parameter

Parameter	A/D		Description
CntPsdDropped	C	Type	16-bit unsigned integer.
		Function	Gives the number of PE events that were demoted to SE over the corresponding second, due to the presence of the "Non Processed" Flag in the PSD Identifier.
		Purpose	To monitor the activity of the PSD, and allow for consistency checks with the SE, ME and PE scientific data
		Applicable range	[0..65535]
		Boundary conditions	The upper bound value is returned if the counter goes into saturation during the counting interval. This should not happen under normal circumstances, because this counter receives only a fraction of the qualified PSD Time Tag signals.

Table 47 CntPsdDropped parameter

Parameter	Attr		Description
CntVetoGateBelow	C	*Type*	*24-bit unsigned integer.*
		Function	*Gives the number of times when the original ACS veto pulse was declared as below threshold (AC non-saturating condition), counted over the corresponding second.*
		Purpose	*To monitor the activity of the ACS*
		Applicable range	*[0..16777215]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This cannot happen under normal conditions because of the ASIC front-end deglitching circuitry, which cannot resolve more than 3 10**6 distinct input ACS pulses per second.*
		Remark	*The value is not a measurement of the actual number of ACS Veto signals below threshold that were received, since internal overlap may occur in the DFEE when building the "below" veto gate.*

Table 48 CntVetoGateBelow parameter

Parameter	Attr		Description
CntVetoGateAbove	C	*Type*	*24-bit unsigned integer.*
		Function	*Gives the number of times when the original ACS veto pulse was declared as below threshold (AC non-saturating condition), counted over the corresponding second.*
		Purpose	*To monitor the activity of the ACS*
		Applicable range	*[0..16777215]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This cannot happen under normal conditions because of the ASIC front-end deglitching circuitry, which cannot resolve more than 3 10**6 distinct input ACS pulses per second.*
		Remark	*The value is not a measurement of the actual number of ACS Veto signals above threshold that were received, since internal overlap may occur in the DFEE when building the " above " veto gate.*

Table 49 CntVetoGateAbove parameter

Parameter	API	Description	
CntVetoGate	C	*Type*	*24-bit unsigned integer.*
		Function	*Gives the number of distinct AC veto gates that were produced over the corresponding second.*
		Purpose	*To monitor the activity of the ACS*
		Applicable range	*[0..16777215]*
		Boundary conditions	*The upper bound value is returned if the counter goes into saturation during the counting interval. This cannot happen under normal conditions because of the ASIC front-end deglitching circuitry, which cannot resolve more than 3 10**6 distinct input ACS pulses per second.*
		Remark	*The AC veto gate is a combination of the "below" and "above" veto gates. The value is not a measurement of the number of ACS Veto pulse signals that were actually received, since internal overlap between following pulses may occur in the DFEE when building the veto gate. It may be lower than the sum of CntVetoGateBelow and CntVetoGateAbove, since overlap may occur between them when doing the combination.*

Table 50 CntVetoGate parameter

Parameter	API	Description	
CntVetoGateDTime	C	*Type*	*24-bit unsigned integer.*
		Function	*Gives the actual dead time created by the combined AC veto gate in the DFEE, over the corresponding second.*
		Purpose	*To monitor the activity of the ACS, and check consistency with the other count values (the various counts produced for the ACS and the AFEE related counts).*
		Applicable range	*[0..16777215]*
		Boundary conditions	*The value may not saturate over a second at the ASIC operating frequency.*
		Remark	*The value is not an normalised measurement of dead time. A fractional dead time value may be obtained by multiplying it by 2/sum(TimeClkMonTF_i), over the same time period.*

Table 51 CntVetoGateDTime parameter

HOUSEKEEPING PACKETS

This section gives the list of the HK acquisition commands, the corresponding block format and the request periodicity.

HK request commands

This section gives the list of the HK block requests supported by the DFEE. Note that HK requests #$0C to #$12, although devoted to diagnostic mode, are not rejected by the DFEE in other situations.

$00	Generic Status	8 s	4
$01	Software parameters (readback)	3840 s	4
$02	Algorithm variations & HSL settings (readback)	3840 s	4
$03	Detector timing & Veto Gate settings (readback)	3840 s	4
$04	Status & Veto/PSD/$AFEE_0$ counts	1 s	4
$05	$AFEE_1$, $AFEE_2$ & $AFEE_3$ counts	1 s	4
$06	$AFEE_4$, $AFEE_5$ & $AFEE_6$ counts	1 s	4
$07	$AFEE_7$, $AFEE_8$ & $AFEE_9$ counts	1 s	4
$08	$AFEE_{10}$, $AFEE_{11}$ & $AFEE_{12}$ counts	1 s	4
$09	$AFEE_{13}$, $AFEE_{14}$ & $AFEE_{15}$ counts	1 s	4
$0A	$AFEE_{16}$, $AFEE_{17}$ & $AFEE_{18}$ counts	1 s	4
$0B	Clock monitor & PSD counts	1 s	4
$0C	TF_1 & TF_2 Time Frame status (DIAG)	2 s (DIAG)	4
$0D	TF_3 & TF_4 Time Frame status (DIAG)	2 s (DIAG)	4
$0E	TF_5 & TF_6 Time Frame status (DIAG)	2 s (DIAG)	4
$0F	TF_7 & TF_8 Time Frame status (DIAG)	2 s (DIAG)	4
$10	AR_3, AR_6, AR_8 & AR_2 ASIC Register status (DIAG)	2 s (DIAG)	4
$11	AR_4, AR_7 & AR_9 ASIC Register status (DIAG)	2 s (DIAG)	4
$12	AR_5, AR_1 & AR_0 ASIC Register status (DIAG)	2 s (DIAG)	4

Table 52 Supported HK block requests

The format of the 4-byte command block used for the corresponding HK requests is given in the following table.

Byte No.	
1	"K" (ASCII character code)
2	HK Request Identifier
3	$00
4	Checksum

Table 53 HK generic request format

HK block list

This following table indicates the the content and length of the HK blocks sent by the DFEE in response to the above described HK requests. The HK #$00 block is described in section « Status & Errors ».

$00	Generic Status	6
$01	Software parameters (readback)	30
$02	Algorithm variations & HSL settings (readback)	20
$03	Detector timing & Veto Gate settings (readback)	28
$04	Status & Veto/PSD/$AFEE_0$ counts	28
$05	$AFEE_1$, $AFEE_2$ & $AFEE_3$ counts	30
$06	$AFEE_4$, $AFEE_5$ & $AFEE_6$ counts	30
$07	$AFEE_7$, $AFEE_8$ & $AFEE_9$ counts	30
$08	$AFEE_{10}$, $AFEE_{11}$ & $AFEE_{12}$ counts	30
$09	$AFEE_{13}$, $AFEE_{14}$ & $AFEE_{15}$ counts	30
$0A	$AFEE_{16}$, $AFEE_{17}$ & $AFEE_{18}$ counts	30
$0B	Clock monitor & PSD counts	34
$0C	TF_1 & TF_2 Time Frame status (DIAG)	36
$0D	TF_3 & TF_4 Time Frame status (DIAG)	36
$0E	TF_5 & TF_6 Time Frame status (DIAG)	36
$0F	TF_7 & TF_8 Time Frame status (DIAG)	36
$10	AR_3, AR_6, AR_8 & AR_2 ASIC Register status (DIAG)	26
$11	AR_4, AR_7 & AR_9 ASIC Register status (DIAG)	26
$12	AR_5, AR_1 & AR_0 ASIC Register status (DIAG)	26

Table 54 HK block list

Cfg readback HK packets

HK #$01 : Software parameters readback

Byte Num.							
1	User Acknowledge						
2	$01						
3	0	InhibAutotst CODE	InhibAutotst ASIC	InhibAutotst RAM	NHslErrThrshold		
4	DiagFreq						
5	0	0	0	EnHslErrAct	NHslErrActThrshold		
6..29	0						
30	CheckSum						

Table 55 HK #$01 format

HK #$02 : Algorithm variations readback

Byte Num	Bit #0	Bit #1	Bit #2	Bit #3	Bit #4	Bit #5	Bit #6	Bit #7
1	User Acknowledge							
2	$02							
3	RstFrontTFrame	RstFrontVeto	RstFrontPsd	RstFrontAfee18	RstFrontAfee17	RstFrontAfee16	RstFrontAfee15	RstFrontAfee14
4	RstFrontAfee13	RstFrontAfee12	RstFrontAfee11	RstFrontAfee10	RstFrontAfee9	RstFrontAfee8	RstFrontAfee7	RstFrontAfee6
5	RstFrontAfee5	RstFrontAfee4	RstFrontAfee3	RstFrontAfee2	RstFrontAfee1	RstFrontAfee0	RstCountVPsd	RstCountAfee18
6	RstCountAfee17	RstCountAfee16	RstCountAfee15	RstCountAfee14	RstCountAfee13	RstCountAfee12	RstCountAfee11	RstCountAfee10
7	RstCountAfee9	RstCountAfee8	RstCountAfee7	RstCountAfee6	RstCountAfee5	RstCountAfee4	RstCountAfee3	RstCountAfee2
8	RstCountAfee1	RstCountAfee0	RstAssoSm	RstPobjSm	RstRcveSm	RstRcveSerial	RstAcqSm	RstDialSm
9	EnCntDtAfee	EnCntDtAfeeSat	EnCntDtVetoGate	LowResMode	ToutScaleDropNrj 3 (MSB)	ToutScaleDropNrj 2	ToutScaleDropNrj 1	ToutScaleDropNrj 0 (LSB)
10	AssoMultWinSize 4 (MSB)	AssoMultWinSize 3	AssoMultWinSize 2	AssoMultWinSize 1	AssoMultWinSize 0 (LSB)	AcMode1 (MSB)	AcMode0 (LSB)	AssoEnablePsd
11	AssoEnableAfee 18	AssoEnableAfee 17	AssoEnableAfee 16	AssoEnableAfee 15	AssoEnableAfee 14	AssoEnableAfee 13	AssoEnableAfee 12	AssoEnableAfee 11
12	AssoEnableAfee 10	AssoEnableAfee 9	AssoEnableAfee 8	AssoEnableAfee 7	AssoEnableAfee 6	AssoEnableAfee 5	AssoEnableAfee 4	AssoEnableAfee 3
13	AssoEnableAfee 2	AssoEnableAfee 1	AssoEnableAfee 0	ToutValMissNrj 4 (MSB)	ToutValMissNrj 3	ToutValMissNrj 2	ToutValMissNrj 1	ToutValMissNrj 0 (LSB)
14	EvtKeepSE	EvtKeepME	EvtKeepPE	EvtKeepPP	EvtSetTimeFmtPE	EvtForceNVeto	EvtForceProcPE	SpeStorageMode 1 (MSB)
15	SpeStorageMode 0 (LSB)	1	1	0	HsIXferLength 15 (MSB)	HsIXferLength 14	HsIXferLength 13	HsIXferLength 12
16	HsIXferLength 11	HsIXferLength 10					HsIXferLength 5	HsIXferLength 4
17	HsIXferLength 3	HsIXferLength 2	HsIXferLength 1	HsIXferLength 0 (LSB)	0	0	1	1
18	0	$00						
19	0							
20	CheckSum							

Table 56 HK #$02 format

HK #$03 : Detector settings readback

Byte No.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	User Acknowledge							
2	$03							
3	MSB	DelayAfeeTT_18			LSB	MSB	DelayAfeeSat_18	LSB
4	MSB	DelayAfeeTT_17			LSB	MSB	DelayAfeeSat_17	LSB
5..19	...					...		
20	MSB	DelayAfeeTT_1			LSB	MSB	DelayAfeeSat_1	LSB
21	MSB	DelayAfeeTT_0			LSB	MSB	DelayAfeeSat_0	LSB
22	DelayVetoFrst 3 (MSB)	DelayVetoFrst 2	DelayVetoFrst 1	DelayVetoFrst 0 (LSB)	DelayVetoScnd 5 (MSB)	DelayVetoScnd 4	DelayVetoScnd 3	DelayVetoScnd 2
23	DelayVetoScnd 1	DelayVetoScnd 0 (LSB)	XtndThresh 7 (MSB)	XtndThresh 6	XtndThresh 5	XtndThresh 4	XtndThresh 3	XtndThresh 2
24	XtndThresh 1	XtndThresh 0 (LSB)	XtndGateAbove 7 (MSB)	XtndGateAbove 6	XtndGateAbove 5	XtndGateAbove 4	XtndGateAbove 3	XtndGateAbove 2
25	XtndGateAbove 1	XtndGateAbove 0 (LSB)	XtndGateBelow 7 (MSB)	XtndGateBelow 6	XtndGateBelow 5	XtndGateBelow 4	XtndGateBelow 3	XtndGateBelow 2
26	XtndGateBelow 1	XtndGateBelow 0 (LSB)	DelayPsd 4 (MSB)	DelayPsd 3	DelayPsd	DelayPsd	DelayPsd LSB	1
27	1	1	RouteMode 1 (MSB)	RouteMode 0 (MSB)	0			
28	CheckSum							

Table 57 HK #$03 format

Scientific HK packets

HK #$04 : Status and Veto/PSD/AFEE$_0$ counts

Byte Number	Parameter	Attribute
1	User Acknowledge	
2	$04	
3	NsecFromStart	*Most Significant Byte*
4		*Least Significant Byte*
5	DfeeStatus1	*See section "Status & Errors"*
6	DfeeStatus2	
7	DfeeStatus3	
8	DfeeStatus4	
9	CntVetoGate	*Most Significant Byte*
10		*medium Significant Byte*
11		*Least Significant Byte*
12	CntVetoGateDTime	*Most Significant Byte*
13		*medium Significant Byte*
14		*Least Significant Byte*
15	CntPsdTT	*Most Significant Byte*
16		*Least Significant Byte*
17	CntPsdDropped	*Most Significant Byte*
18		*Least Significant Byte*
19	CntAfeeTT_0	*Most Significant Byte*
20		*Least Significant Byte*
21	CntAfeeTTsat_0	*Most Significant Byte*
22		*Least Significant Byte*
23	CntAfeeNVeto_0	*Most Significant Byte*
24		*Least Significant Byte*
25	CntAfeeDTime_0	*Most Significant Byte*
26		*medium Significant Byte*
27		*Least Significant Byte*
28	CheckSum	

Table 58 HK #$04 format

See section "Status and Errors" for a detailed description of the DFEE HK #$04 Status Bytes (DfeeStatus1, -2, -3 & -4).

HK #$05 to #$0A : AFEE counts

HK blocks #$05 to #$0A show the same format. Each of them give counting statistics for 3 AFEE channels.

Byte Number	Parameter		Value for AFEE Channel Number					
1	User Acknowledge							
2			$05	$06	$07	$08	$09	$0A
3	CntAfeeTT	*MSByte*	1	4	7	10	13	16
4		*LSByte*						
5	CntAfeeTTsat	*MSByte*						
6		*LSByte*						
7	CntAfeeNVeto	*MSByte*						
8		*LSByte*						
9	CntAfeeDTime	*MSByte*						
10		*mSByte*						
11		*LSByte*						
12	CntAfeeTT	*MSByte*	2	5	8	11	14	17
13		*LSByte*						
14	CntAfeeTTsat	*MSByte*						
15		*LSByte*						
16	CntAfeeNVeto	*MSByte*						
17		*LSByte*						
18	CntAfeeDTime	*MSByte*						
19		*mSByte*						
20		*LSByte*						
21	CntAfeeTT	*MSByte*	3	6	9	12	15	18
22		*LSByte*						
23	CntAfeeTTsat	*MSByte*						
24		*LSByte*						
25	CntAfeeNVeto	*MSByte*						
26		*LSByte*						
27	CntAfeeDTime	*MSByte*						
28		*mSByte*						
29		*LSByte*						
30	CheckSum							

Table 59 HK #$05 to #$0A format

HK #$0B : Clock monitor and PSD counts

Byte Number	Parameter	
1	User Acknowledge	
2	$0B	
3	TimeClkMonTF_1	MSByte
4		mSByte
5		LSByte
6	TimeClkMonTF_2	MSByte
7		mSByte
8		LSByte
9	TimeClkMonTF_3	MSByte
10		mSByte
11		LSByte
12	TimeClkMonTF_4	MSByte
13		mSByte
14		LSByte
15	TimeClkMonTF_5	MSByte
16		mSByte
17		LSByte
18	TimeClkMonTF_6	MSByte
19		mSByte
20		LSByte
21	TimeClkMonTF_7	MSByte
22		mSByte
23		LSByte
24	TimeClkMonTF_8	MSByte
25		mSByte
26		LSByte
27	CntVetoAbove	MSByte
28		mSByte
29		LSByte
30	CntVetoBelow	MSByte
31		mSByte
32		LSByte
33	$00	
34	CheckSum	

Table 60 HK #$0B format

Diagnostic HK blocks

In DIAG mode, additional parameters are retrieved by the DPE software. These parameters give detailed status analysis of the ASIC evolution for 1-second intervals, but are only retrieved every other second by the DPE. In other terms, they provide detailed insight for a 1-second duration, with a factor 2 division sampling.

The following blocks contain ASIC Register information. ASIC Registers are segmented into bytes, and ordered most significant byte first. Inside each byte, bits are ordered most significant bit first. If the number of bits of a ASIC Register is not a multiple of 8, the excess bits are the least significant bits of the last byte, and their content should be ignored.

Refer to document « *L'ASIC DFEEA51C* », CCST section, for a detailed description of the corresponding ASIC Registers.

HK #$0C to #$0F: Detailed status

Byte Number	Parameter		
1	User Acknowledge		
2	$0C		
3	NsecFromStart		*MSByte*
4			*LSByte*
5..17	ASIC Register AR_{30}	Time Frame TF_1	*102 bits in 13 bytes*
18..19	ASIC Register AR_{31}		*10 bits in 2 bytes*
20..32	ASIC Register AR_{30}	Time Frame TF_2	*102 bits in 13 bytes*
33..34	ASIC Register AR_{31}		*10 bits in 2 bytes*
35	$00		
36	CheckSum		

Table 61 HK #$0C format

Byte Number	Parameter		
1	User Acknowledge		
2	$0D		
3	NsecFromStart		*MSByte*
4			*LSByte*
5..17	ASIC Register AR_{30}	Time Frame TF_3	*102 bits in 13 bytes*
18..19	ASIC Register AR_{31}		*10 bits in 2 bytes*
20..32	ASIC Register AR_{30}	Time Frame TF_4	*102 bits in 13 bytes*
33..34	ASIC Register AR_{31}		*10 bits in 2 bytes*
35	$00		
36	CheckSum		

Table 62 HK #$0D format

Byte Number	Parameter		
1	User Acknowledge		
2	$0E		
3	NsecFromStart		MSByte
4			LSByte
5..17	ASIC Register AR_{30}	Time Frame TF_5	102 bits in 13 bytes
18..19	ASIC Register AR_{31}		10 bits in 2 bytes
20..32	ASIC Register AR_{30}	Time Frame TF_6	102 bits in 13 bytes
33..34	ASIC Register AR_{31}		10 bits in 2 bytes
35	$00		
36	CheckSum		

Table 63 HK #$0E format

Byte Number	Parameter		
1	User Acknowledge		
2	$0F		
3	NsecFromStart		MSByte
4			LSByte
5..17	ASIC Register AR_{30}	Time Frame TF_7	102 bits in 13 bytes
18..19	ASIC Register AR_{31}		10 bits in 2 bytes
20..32	ASIC Register AR_{30}	Time Frame TF_8	102 bits in 13 bytes
33..34	ASIC Register AR_{31}		10 bits in 2 bytes
35	$00		
36	CheckSum		

Table 64 HK #$0F format

HK #$10 to #$12 : Detailed status

Byte Number	Parameter	
1	User Acknowledge	
2	$10	
3	NsecFromStart	MSByte
4		LSByte
5..16	ASIC Register AR_{03}	96 bits in 12 bytes
17..19	ASIC Register AR_{06}	23 bits in 3 bytes
20..23	ASIC Register AR_{08}	29 bits in 4 bytes
24	ASIC Register AR_{02}	1 bit in 1 byte
25	$00	
26	CheckSum	

Table 65 HK #$10 format

Byte Number	Parameter	
1	User Acknowledge	
2	$11	
3	NsecFromStart	MSByte
4		LSByte
5..17	ASIC Register AR_{04}	100 bits in 13 bytes
18..20	ASIC Register AR_{07}	22 bits in 3 bytes
21..24	ASIC Register AR_{09}	28 bits in 4 bytes
25	$00	
26	CheckSum	

Table 66 HK #$11 format

Byte Number	Parameter	
1	User Acknowledge	
2	$12	
3	NsecFromStart	MSByte
4		LSByte
5..14	ASIC Register AR_{05}	73 bits in 10 bytes
15..20	ASIC Register AR_{01}	48 bits in 6 bytes
21..24	ASIC Register AR_{00}	32 bits in 4 bytes
25	$00	
26	CheckSum	

Table 67 HK #$12 format

STATUS AND ERRORS

Overview

The DFEE produces status information in both the HK #$00 and HK #$04 blocks.

HK #$00

The HK #$00 block is in the "technology housekeeping" class. As such, it is regularly extracted (at 8 seconds intervals) and can be checked at the MOG level, where decisions should be taken based on the value and severity of the provided status bits. The format of the HK #$00 block is standard for each SPI subassembly, and leaves 3 bytes available for status encoding (bytes #3, #4 and #5).

HK #$04

The HK #$04 block is one of the 8 "scientific housekeeping" blocks which are produced by the DFEE for each second of operation. Four distinct status bytes are available (bytes #5, #6, #7 and #8). The HK #$04 status will not be checked at the MOG level, but can be verified later at the ISDC level. For each second of observation, the HK #04 status gives information on the integrity of the scientific data that were taken in the same second.

Severity levels

This section gives the detailed bit format of the HK #$00 and HK #$04 status bytes. The intended objective was to classify the information according to 3 classes of severity : Note, Warning and Alert.

Severity : Note

An exceptional condition was encountered, but the DFEE is capable of recovering. The DFEE continues working, and the status will disappear as soon as the condition disappears.

Severity : Warning

An abnormal condition was encountered, but the cause is probably external to the DFEE, and the DFEE is probably still working if no Alrt bit is set. If the condition disappears, the DFEE should recover and continue.

Severity : Alert

The DFEE itself does not operate correctly, and requires partial or complete Reset. It is not guaranteed that Reset will solve the problem.

Status update policy

Many of the status bits are given in both HK #$00 and HK #$04 blocks. At first glance, this may appear as an excessive redundancy. However, the update policy is strongly different between them. The bits are not integrated over the same period for HK #$00 and HK #$04, and they should end up into different parameters.

HK #$00 update

The HK #$00 status bits are reset after each HK #$00 block acquisition by the DPE IASW. In other words, they will capture any exceptional condition that will occur in between two following HK #$00 acquisitions. The interval between acquisitions may be 8 seconds in the SPI IASW application, but the DFEE is not aware of this and could integrate the exceptional conditions over any interval larger than 1 second.

HK #$04 update

The HK #$04 status bits follow the same update policy as the other scientific HK blocks. They are reset at each second boundary. They will capture any exceptional condition that will occur during a one second interval. Some additional (HK #$04 specific) bits give additional insight to indicate where the problem occurred during the corresponding second. The HK #$04 status bits are lost if the IASW does not read the HK #$04 block in the following second.

Status classification

The HK #$00 and HK #$04 status bits indicate conditions which are observed inside the DFEE, but may be caused by both external and internal problems. As such, they give information which has a larger scope than the DFEE itself. They can be classified according to the following :

DPE synchro signal analysis

The absence of the 8 Hz clock signal is detected in CFG and OPL/Diag modes. Almost all of the other status bits being updated on this time base, they should be considered as invalid if the 8 Hz clock absence is declared.

The absence of the HSL Clock is detected in CFG and OPL/Diag. Since the DPE does not apply it before the first OPL/Diag Time Frame, this must not be considered as an error before this stage.

Detector activity and problems

High rate detector activity may create Pobj event loss, and even 8 Hz event loss. Temporary ASIC/microcontroller incoherence may result, and empty HSL packets may be produced.

AFEE/PSD problems or signal integrity problems may cause TimeOuts or Energy/Id Drops.

Unexpected transitions on the PSD Time Tag line may awake the known P-PA ASIC bug, which is covered in document « On the erroneous P-PA handling ». This should never occur if the PSD and corresponding signal transmission stay healthy.

ASIC failures

ASIC defects (due to Single Event Upset events or other unexpected conditions) may cause State Machine errors and internal register changes. These are sensed at each Time Frame and create Alerts.

ASIC / micro-controller coherence

The sequence of 8 Hz clock transitions seen by both the ASIC and the micro-controller are continuously counterchecked.

HK #$00

A status bit, if equal to 1, indicates an abnormal condition, classified as Note, Warning or Alert.

The mode can be Stby (b000), Conf (b001), Oper (b010), and Diag (b011), expressed as a 3-bit unsigned binary number. The returned value corresponds to the current mode where the DFEE stood when the packet was sent.

The Autotest bits are remnant (they are not cleared after HK #$00 read). The Asic Autotest is performed at Power On and after each Restart command. The micro-controller RAM is only checked at Power On.

The HK #$00 block format is recalled here for reference. Bytes #1, #2 and #3 folllow the standard SPI definition. Proprietary bytes #4 and #5 are used to post specific DFEE status information.

A HK #$00 read operation clears the bits contained in HK #$00 bytes #4 and #5. Any note, warning or alert occuring during any second thereafter will remain present until the next HK #$00 read operation. Under normal conditions (read every 8 seconds) those two bytes integrate conditions which occured on a 8 seconds basis.

1	User Acknowledge							
2	$00							
3	Mode Stby / Conf / Oper / Diag					AutoTst Nok	AutoTstAsic Nok	ExecTc Ninit
4	Warn RunProgr	Warn HslClkOp	Alrt CoherTst	Alrt CoherCfg	Alrt PobjPrtcl	Alrt SmNRun	Alrt TimeBase	Alrt HslErr
5	Note TimeOut	Note Drop	Note ItemOvf	Note PobjOvf	Note DialPrtl	Note 8HzProgr	Warn 8HzAbsnt	Warn DialPrty
6	CheckSum							

Table 68 HK #$00 format

HK #$00 severity

Status field	Description	
AutoTstNok	The PROM-resident code showed a wrong checksum (InhibAutotstCODE = 0) OR : The periodic or startup RAM verification failed (InhibAutotstRAM = 0) OR The ASIC autotest failed (InhibAutotstASIC = 0)	STDBY, CFG, OPL : Alrt
AutoTstASICNok	The ASIC autotest failed (Inhib Autotst ASIC = 0)	STDBY, CFG, OPL : Alrt
AlrtHslErr	During the previous second the number of erroneous HSL transfers was ≥ NhslErrThreshold	STDBY, CFG : ignore OPL : Alrt
AlrtTimeBase	The ASIC TimeBase is NOT in the correct configuration : HslTstEnabled OR SysTstEnabled OR NOT OscClkEnabled OR ExtClkEnabled OR ScanActive	STDBY, CFG : ignore OPL : Alrt
AlrtSmNRun	At least One of the ASIC State Machines is NOT Running	STDBY, CFG : ignore OPL : Alrt
AlrtPobjPrtcl	The Pobj FIFO encountered a Protocol Error condition on the Write or the Read port	STDBY, CFG : ignore OPL : Alrt
AlrtCoherCfg	The ASIC configuration was found incompatible with the micro-controller Cfg #$02 / Cfg #$03 image (ASIC Registers AR1, -3, -4 ,-5)	STDBY, CFG : ignore OPL : Alrt
AlrtCoherTst	The ASIC test registers were found incompatible with their expected state (ASIC Registers AR6, -7, -8)	STDBY, CFG, OPL : Alrt
WarnHslClkOp	The HSL Clock is not running (NOT HslEnabled)	STDBY, CFG : ignore OPL : Wng
WarnRunProgr	The ASIC did not enter correctly into OPL / DIAG mode : TframeFirst OR RunStarting OR NOT GlobalEnableEnv OR NOT TFrameEnabled	STDBY, CFG : ignore OPL : Wng
WarnDialPrty	At least one of the SE, ME, PE data blocks was altered during temporary FIFO storage	STDBY, CFG : ignore OPL : Wng
Warn8HzAbsnt	The 8 Hz micro-controller interrupt was not entered since the last transition to CFG or OPL/DIAG	CFG, OPL : Alrt **If present, other status bits are invalid**
Note8HzProgr	The ASIC and the microcontroller did not react in phase with the 8 Hz Clock	STDBY, CFG : ignore OPL : Note
NoteDialPrtl	At least one of the SE, ME, PE or SP data blocks was partial on HSL in the previous second	STDBY, CFG : ignore OPL : Note
NotePobjOvf	The POBJ logic had to discard event(s) due to overflow	STDBY, CFG : ignore OPL : Note
NoteItemOvf	A too long ME cascade was detected and truncated, either at Asso or Acq	STDBY, CFG : ignore OPL : Note
NoteDrop	At least one unconsumed Energy / Id was dropped	STDBY, CFG : ignore OPL : Note
NoteTimeOut	At least one TimeTag did not receive the corresponding Energy / Id in time	STDBY, CFG : ignore OPL : Note

Table 69 HK #$00 status severity

The above table indicates the active condition and severity level of the HK #$00 status bits, as a function of the DFEE mode.

Validity of the HK #$00 status bits

Due to pipelining, the status encountered when in CFG mode may still be seen during the first 2 seconds of OPL/DIAG mode. Status bits declared with the Alrt severity should not be checked before the third second of OPL/DIAG mode.

HK #$04 Status

The following shows a detailed view of the 4 Status bytes provided in HK #$04 (DfeeStatus1, -2, -3 and -4). See the « Housekeeping Packets » section for a complete view of this housekeeping block.

A status bit, if equal to 1, indicates an abnormal condition, classified as Alert, Warning or Note. The SpvWound and StsSerHsl fields give the value of an encoded internal DFEE status. Under normal conditions SpvWound = b000 and StsSerHsl = b010 (expressed as 3-bit unsigned binary numbers).

The HK #$04 gives notes, warnings or alerts which occured inside the DFEE during the previous second of operation.

Byte No.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5	WarnHslErr TF8	WarnHslErr TF7	WarnHslErr TF6	WarnHslErr TF5	WarnHslErr TF4	WarnHslErr TF3	WarnHslErr TF2	WarnHslErr TF1
6	Warn RunProgr	Warn HslClkOp	Alrt CoherTst	Alrt CoherCfg	Alrt PobjPrtcl	Alrt SmNRun	Alrt TimeBase	Alrt HslErrAct
7	Warn SpvWound			Warn StsSerHsl			Warn CoherPeAddr	Warn PobjPrtclWr
8	Note TimeOut	Note Drop	Note ItemOvf	Note PobjOvf	Note DialPrtl	Note 8HzProgr	Warn 8HzAbsnt	Warn DialPrty

Table 70 HK #$04 detailed status format

Content of the HK #$04 status explained

All bits in the 4 status bytes of HK #$04 are valid only in an operational mode. The severity level Note, Warning and Alert is given in Table 70, together with the name and position of the HK #$04 status bits.

Those bits which have the same name in HK #$00 and HK #$04 have also the same signification and are given for redundancy purpose. The only difference is the integration time: the HK #$00 bits are accumulating between two successive HK #$00 read operatoins (normally 8 s), and the HK #$04 bits give the status for every second.

The following list gives the signification of those bits which are not present in HK #$00.

Byte #5

A "WarnHslErr" bit is set to 1 if in the corresponding time-frame (TF1 to TF8) of the present second a HSL transaction has been erroneous. In the case of a pathological P-PA event occuring in time frame i (=1..8) of a second (see corresponding section hereafter)

one can expect that the HSL transaction remains blocked. This means that all the bits for time frame i to 8 are set to 1, and all 8 bits remain at 1 in the next seconds. An automatic reset mechanism (see hereafter) exists in order to unblock the ASIC in such a case.

Byte #6

The bit "AlrtHslErrAct" is set to 1, if in the current second the automatic ASIC reset mechanism has been activated (see section hereafter).

Byte #7

The bitfield "WarnSpvWound" expresses and internal DFEE state-machine state and should contain the binary value b000 in case of no warning.

The bitfield 'WarnStsSerHsl' expresses the status of the HSL serializer state-machine and should contain the binary value b010 in case of no warning.

The bit "WarnCoherPeAddr" contains 1 if during the current second, for one or more PE, a mismatch between the AFEE and the PSD detector numbers associated in the PE took place.

The bit "WarnPobjPrtclWr" is set to 1 if during the current second, the POBJ write operation failed with a protocol error. In this case the bit "AlrtPobjPrtcl" is also set to 1 and the "WarnPobjPrtclWr" permits to disentangle if the POBJ had an erroneous read or write operation.

HSL error monitoring

The DFEE micro-controller software is capable of detecting incorrect situations where the HSL ceases to be operational. If instructed to do so by configuration, it is also able to restore the ASIC to a normal state, with minimal impact on overall system behaviour.

DFEE-ASIC status elaboration

The micro-controller software continuously monitors ASIC behaviour, and implements a variable-length survey of the activity over several following Time Frames.

A dynamic status of the ASIC activity is elaborated according to the following principles :

- The status of the operations in the DFEE-ASIC during any Time Frame TF(n) is frozen by the ASIC inside a buffer on the next (8 Hz or 1 Hz) Time Frame boundary.
- During Time Frame TF(n+1), this buffer is read by the DFEE micro-controller and contributes to the formation of the status word of the present second.
- When TF(7) of the present second is reached, the status of all the previous 8 Time Frames have been read, and a status words for the elapsed present second is formed.
- After the Swap of the HK buffers just after the next second boundary, this status word is available for the HK requests concerning the previous second.

Note that for *what the status words are concerned*, the information used to compute the HK response does not exactly cover the corresponding second interval : the 1-second integration time used for status elaboration is actually shifted by one Time Frame towards the past. The elaborated status words are delivered in HK #$00 and HK #$04.

HSL error detection

The HSL error detection mechanism is the following :

- The software-maintained *CHslErr* counter is cleared at every boundary of the 1-second interval.
- If during a Time Frame status read operation, a HSL error is detected, the *CHslErr* count is incremented.
- If during the formation of the accumulated (1-second) status word in TF(7), counter *CHslErr* gets greater than or equal to *NHslErrThreshold* (a Cfg #$01 configuration parameter), then the *AlrtHslErr* bit is activated in HK #$00. In addition, a detailed view of the HSL error sequence during the elapsed second is available in HK #$04. The *AlrtHslErr* bit of HK #$00 is provided in redundancy with the WarnHslErr bits in HK #$04, to allow for reactive action at the MOC level, and possibly data filtering at the data analysis level.

Conditional ASIC reset on HSL error detection

In addition to the above HSL error detection procedure, the DFEE software may be configured to take automatic corrective action when an adjustable number of consecutive HSL errors have been detected.

Two Cfg #$01configuration parameters control this behaviour :

- *EnHslErrAct* : This parameter, when set to 1/0, enables/disables the automatic ASIC reset procedure on HSL error detection.
- *NhslErrActThreshold* :This parameter is a HSL error counter threshold.

If, during TF(7) the automatic reset is enabled (*EnHslErrAct* = 1) and if the *CHslErr* counter gets greater than or equal to *NHslErrActThreshold*, then the automatic ASIC reset is performed, by resetting the *Association, Pobj, Acquisition* State Machines and restoring the previous (de-reset) state (as it was sent in the previous Cfg #$02 configuration command). The reset/de-reset procedure takes 3.4 ms to complete and thus terminates safely, well before any new second boundary.

The reset/de-reset action, when taken, is signalled in HK #$04, by the activation of *AlrtHslErrAct*. This monitoring information is not coded in HK #$00 due to lack of space.

Remarks

- The reset/de-reset action can be globally enabled/disabled by configuration, and programmed to operate only when a certain number of HSL transfers have been found defective during a particular second.
- The occurrence of the reset/de-reset action can be monitored by the *AlrtHslErrAct* HK #$04 status bit.
- The various ASIC signal monitoring counters are not affected by the reset/de-reset action, and may be used to investigate on the cause of the HSL stall.
- The HSL recovers in the Time Frame that follows the reset/dereset action, but scientific event data only reappear in the next following Time Frame.

KNOWN BUGS

Two known bugs are present in the DFEE ASIC. They were considered minor and acceptable because they cannot show up in the SPI operational environment. They are quoted here for reference, for potential users intending to use the DFEE in other contexts (ground and flight timing tests, EGSE DPE emulators, ...).

Pathological P-PA event

A very particular event sequence, the so-called « P-PA event », is not accepted by the DFEE ASIC, which gets stalled if it occurs. Due to the timing characteristics of the PSD, this pathological sequence cannot occur when SPI is operated in the normal mode. However, it may occur if the DFEE ASIC is configured to implement very specific timing test procedures. Work-arounds have been studied and validated to support these timing procedures. A detailed analysis of the problem and the description of the work-arounds are given in document « DFEE erroneous P-PA handling», 2000/07/27, V1.0, ref. DAPNIA/SEI/SPI/STP/00-442 (by S.Schanne).

HSL frequency limit

When in operation, the DFEE ASIC runs from a 20 MHz free-running quartz oscillator clock, and the HSL link runs from a DPE-provided 5 MHz clock. The system operates correctly for this nominal frequency ratio, but will not produce correct HSL output for all values of this ratio. Although it prevented to freely choose the operation frequency of the DFEE ASIC, this imperfection is not an issue in the nominal SPI environment.

DOCUMENT HISTORY

Versions

V1.0 (August 31, 2001)

Draft release for reader feedback.

V1.1 (December 3, 2001)

After feedback from initial readers :

- Official release document number ;
- Clarification of the AcMode parameter, and PSD gate generation conditions ;
- Completion of the ToutScaleDropNrj parameter mapping table, and related comments ;
- Addition of a « Known Bugs » section ;
- Few minor corrections and edits.

V1.2 (July 11 and September 11, 2002)

- Updated the database references for the following parameters (old / new) :

  E7752 / E7786 ToutScaleDropNrj

  E7754 / E7788 AcMode

  E7755 / E7787 LowResMode

  E7846 / E7848 RouteMode

- Precision on HK #$00 read and clear operation.
- In HK #$04 the AlrtHslErr bit is not present as stated, therefore the detailed error sequence is given by the WarnHslErr bits.
- Explanation of the HK #$04 status