Scientific Data Processing tasks for the INTEGRAL *IMAGER*

E.Kendziorra, M.Mohan and R.Staubert

During the Consortium Meeting 1/2 8.95 we took the action of evaluating the processing requirements for the Imager SPU/DPE.

Figure 1 shows the data flow from the detectors to the telemetry packetizer of the DPE and indicates in round boxes the data processing tasks necessary to cope with the limited telemetry capacity asigned to the *IMAGER* on INTEGRAL.

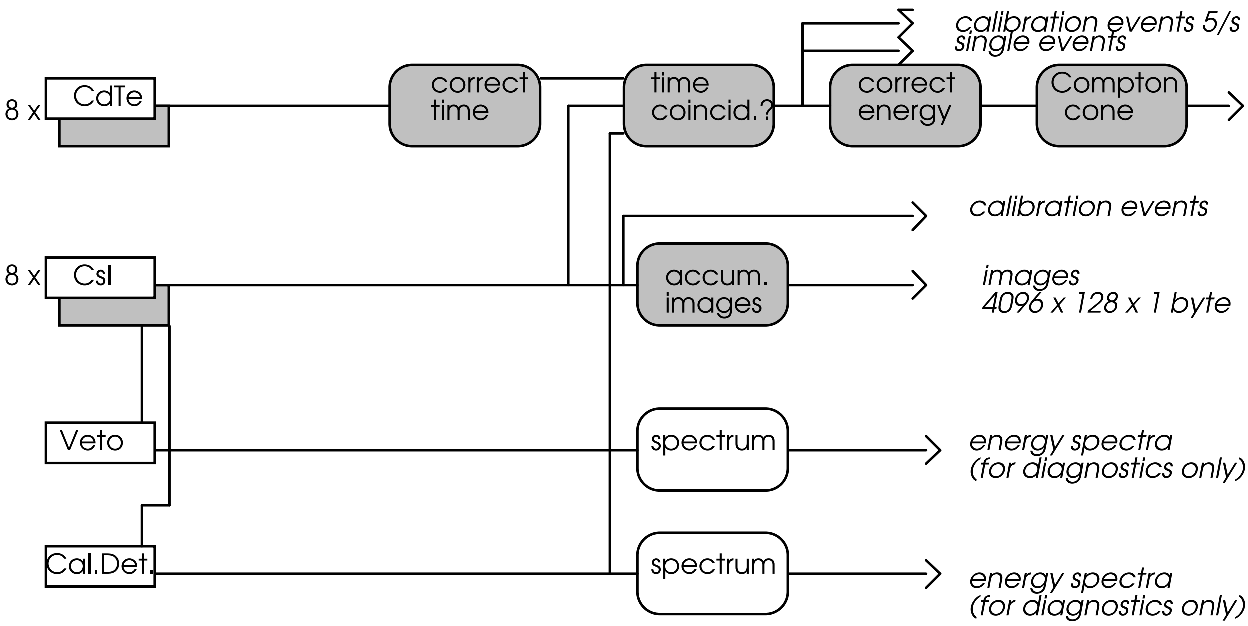


Figure 1. Data flow in DPE, filled round boxes indicate major processing tasks

We have tried to estimate the processing load for the different tasks indicated in Figure 1. Energy spectra from veto detectors and from the calibration detector will only be accumulated for diagnostic purposes. These task do not contribute to the processing load during normal operation. Therefore these two tasks are not further considered.

Searching for time coincidences between CdTe and CsI events is the most demanding task. The main question was, if this can be done by current DPE or if this must be done by a separate unit (SPU) either by a dedicated hardware processor or by S/W on a more powerful processor.
Initially we tried to estimate the processing power needed to perform all the tasks excluding the time coincidences. We have written the algorithms in C language. The code then was compiled by an 68000 cross compiler and the number of assembler instructions counted. A UTMC1750 microprocessor (not 68000) will be used and the high level language used will be ADA (not C) therefore the results can only be taken as a first order estimate.

As shown in Table 1 less than 0.9 MIPS are needed. This is well within the capabilities of the current DPE.
(In his FAX from 7 August 1995 R.Timm had estimated, that up to 1.4 MIPS should be available for scientific instrument needs.)

Table 1:

Description of task	instructions/ event	events/s	computing load in MIPS
correct time of CdTe events	40 [1)]	2400	≤ 0.1
correct energy of CdTe events	40 [1)]	2400	≤ 0.1
compute Compton cone for CsI/CdTe events	80 [1)]	1200	≤ 0.1
accumulate CsI images	34	16800	0.57
miscellanenous	40 [1)]	2400	≤ 0.1
subtotal	-	-	≤ 1.0

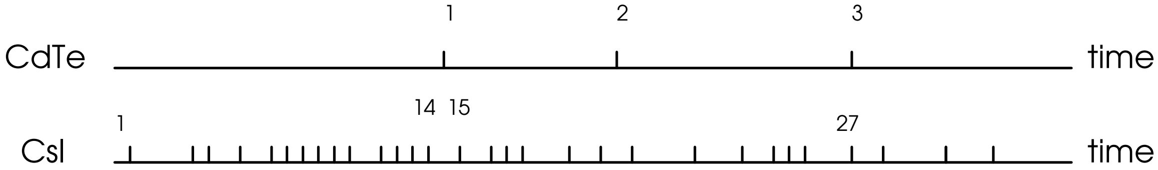
Remarks:

1). Only very preliminary assumption, real algorithm not yet defined. We propose to use look-up tables in order to reduce computing time, if necessary.

Time coincidences between CSI and CdTe

We can make use of the fact that the events from each of the 2 x 8 module control electronics (MCE) are already ordered in time (TBC). Comparing the event times from one CdTe module with those from one CsI module would require comparing ≈300 events with ≈2000 events. If this comparison is done in order of increasing event time, then one CdTe event must be compared with 8 CsI events on average.

Figure 2 illustrates the situation.



The first CdTe event has only be compared in time with the first 15 CsI events. CsI event No. 15 had already arrived after the first CdTe event. One can therefore go to the next CdTe Event and compare the arrival times starting again with CsI event 15.

On average one has to compare 300 CdTe event times with (7 + 1) CsI event times. We have estimated that this would require 300 x 120 instruction = 0.036 MIPS.
But one would have to do this for all 8 x 8 combinations of CdTe and CsI MCEs, requiring 64 x 0.036 = 2.3 MIPs, which is too large to be handled by the DPE.

Table 2 summarizes the computing load for this case, where the events are only sorted in time within each module.

Description of task	computing load in MIPS
subtotal from Table 1	1.0
search event data for time coincidences	2.3
total procesing load	3.7

Table 2: Computing load for DPE

However, if **all** CdTe and **all** CsI events are pre-sorted by time, it would only require 2400 x 120 instructions = 0.29 MIPS.
In Table 3 the estimated computing loads for the case of events which are pre-sorted are summarized. **The total of 1.29 MIPS could be handled by the current DPE**.

Description of task	computing load in MIPS
subtotal from Table 1	1.00
search pre sorted event data for time coincidences	0.29
total procesing load	1.29

Table 3: Computing load for the case of pre-sorted events

Pre-sorting of all CdTe and all CsI events by time

Pre-sorting of CdTe and CsI events could either be done by hardware or by software. Figure 3 shows the resulting modified data flow diagram for the software solution.

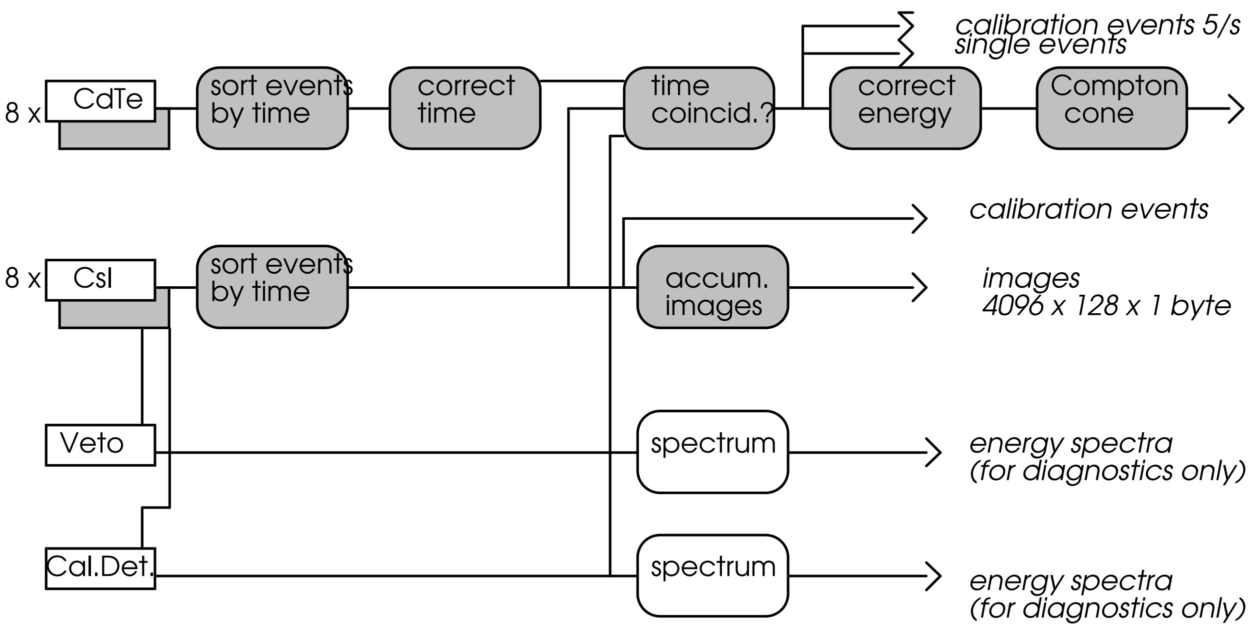


Figure 3. Data flow in DPE including sorting of all events by time

Table 4 summarizes the results of our study. It is obvious, that the DPE can not pre-sort the data from the different MCEs.

Description of task	instructions/ event	events/s	computing load in MIPS
sort all CdTe events by time	107	2400	0.26
sort all CsI events by time	107	16800	1.80
search pre-sorted event data for time coincidences	-	-	0.29
subtotal from Table 1	-	-	1.00
total processing load	-	-	3.35

Table 4

As suggested by Kai Clausen, an alternative solution would be to pre-sort only CdTe events by time and compare these data with the 8 CsI channels. The resulting processing load is summarized in Table 5. The total load is still clearly above the processing power provided by the DPE.

sort all CdTe events by time	107	2400	0.26
search for time coincidences (8 CsI channels)	296	2400	0.71
subtotal from Table 1	-	-	1.00
total processing load	-	-	1.97

Table 5

Figure 4 shows how the computing load varies with count rate. For simplification we have assumed, that the count rate in CsI is seven times the CdTe rate.

Conclusions:

- Sorting of all events by time can not be handled by the DPE
- DPE could handle all other scientific processing tasks, if all CdTe events and all CsI events read by the DPE are already sorted by time.
- We have to do the sorting of events by hardware