Die Echtzeit–Datenverarbeitung des IBIS Experiments auf INTEGRAL

Diplomarbeit

vorgelegt von

Martin Stuhlinger

Eberhard-Karls-Universität Tübingen
Institut für Astronomie und Astrophysik
– Abt. Astronomie –

Januar 1999

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abbildung 1: *INTEGRAL*

Einleitung

Das langfristige Wissenschaftsprogramm *Horizont 2000* der ESA wird durch die vier großen Satellitenprojekte Soho/Cluster, XMM, First und Rosetta sowie durch mehrere Projekte mittleren Budgets, die sogenannten M–Projekte (Medium–size–mission), definiert. Als erstes Projekt (M1) dieser Reihe wurde 1989 die Titan–Sonde Huygens in Zusammenarbeit mit dem amerikanischen Projekt Cassini ausgewählt. Zur Auswahl des zweiten Projekts (M2) gingen über 20 Vorschläge ein, von denen 1990 vier zu einer weitergehenden Studie, der Phase A, zugelassen wurden. Nach Vorstellung der Resultate dieser Phase A–Studien Ende April 1993 in Paris und Beratungen der verschiedenen Fachgremien hat das Wissenschaftliche Programmkomitee der ESA Anfang Juni einstimmig das Projekt *International Gamma–Ray Astrophysics Laboratory* (INTEGRAL) als nächstes M2–Projekt ausgewählt. Der Start dieses Satelliten ist für das Jahr 2001 vorgesehen.

In der optischen Astronomie kann man aus der Form von Spektrallinien, gewonnen mittels hochauflösender Spektroskopie, viel über die Emissionsprozesse sowie die physikalischen Zustände am Ort der Emission lernen. Dieses Prinzip sollte auch auf die Röntgen- und Gamma–Astronomie übertragbar sein. Ziel des Satelliten INTEGRAL ist dementsprechend die hochauflösende Linienspektroskopie mit genauer Ortsauflösung von Punktquellen vom harten Röntgenbereich bis in den weichen Gammabereich (15 keV – 10 MeV) des elektromagnetischen Spektrums. Für diese Aufgabe wurden zwei sich ergänzende Instrumente entwickelt, das Spektrometer SPI und der Imager IBIS. Das Spektrometer soll bei mittlerer Ortsauflösung hochaufgelöste Spektren aufnehmen. Der Imager ist mit seiner mittleren spektralen Auflösung für die genaue Ortsauflösung ausgelegt. Begleitend an Bord des Satelliten sind ein Röntgenmonitor, der nahtlos an den unteren Energiebereich der beiden Hauptinstrumente anschließt sowie eine optische Kamera, die gleichzeitig Meßwerte der beobachteten Quelle liefern, um so die Strahlungscharakteristik eines astronomischen Objekts in weiten Energiebereichen zu ermitteln.

INTEGRAL ist damit eine ideale Ergänzung des 2000 zu startenden ESA–Röntgensatelliten XMM (0,1 – 15 keV), der 2001 parallel zu INTEGRAL nach wie vor aktiv sein wird. Außerdem wird er die Messungen des 1991 gestarteten Gammasatelliten CGRO (*Compton Gamma–Ray Observatory* 20 keV – 30 GeV) ergänzen, dessen vier Instrumente gerade im niederenergetischen Gammabereich über keine ausreichend hohe spektrale Auflösung verfügen.

Eines der Forschungsgebiete des Instituts für Astronomie und Astrophysik der Universtät Tübingen ist die UV- und Röntgenastronomie. Da die Erdatmosphäre elektromagnetische Strahlung mit Wellenlängen kleiner als 300 nm absorbiert, müssen die astronomischen Beobachtungsgeräte für UV- und Röntgenlicht mittels Ballonen oder Satelliten über die Atmosphäre gebracht werden. Als einzige universitäre Einrichtung in Deutschland arbeitet das Astronomische Institut Tübingen an der Entwicklung und dem Bau dieser Ballone und Forschungssatelliten wie XMM und eben INTEGRAL mit. Für letzteren wird experiment–spezifische Elektronik sowie Software für das Instrument IBIS bereitgestellt und es wird ein Beitrag zum *INTEGRAL Science Data Cente*r (ISDC) geleistet.

Im Rahmen zweier Diplomarbeiten wurde eine *Instrument Application Software* entwickelt, welche die Steuerung und die Echtzeit–Datenverarbeitung des IBIS–Experiments übernehmen wird. Die Programme wurden mit einem SUN Hostsystem in der Programmiersprache ADA erstellt und in einem 1750A Simulator getestet. Anschließend wurde die Software in EEPROMS gebrannt und im realen System, der *Data Processing Electronics* des Imagers, eingebaut und verifiziert.

Die vorliegende Diplomarbeit befaßt sich ausschließlich mit der Realisierung der Echtzeit-Datenverarbeitung. Beschrieben werden der Weg der Daten durch die Software, das Zusammenspiel der Softwaremodule im Programmablauf, die Datenformate der erzeugten Telemetriepakete und eine Abschätzung der Leistungsfähigkeit. Es werden auch die für die Datenverarbeitung relevanten, via Telekommandos zu verändernden Parameter und Tabellen erläutert. Nicht eingehen werde ich auf die Verfahrensweisen zum Auswechseln dieser Tabellen. Die *Instrument Application Software* entstand in enger Zusammenarbeit mit Eckard Goehler. Er wird in seiner Diplomarbeit hauptsächlich die Abläufe der Instrumentensteuerung behandeln. Die gesamte Funktionalität der *Instrument Application Software* wird im *Software Design Document* [24] zusammengefaßt.

Das „First Light“ hatte der Imager IBIS bereits im November 1998 während der EM-0 Integration bei der Firma LABEN in Mailand/Italien. Photonen einer radioaktiven Quelle wurden vom Detektor ISGRI registriert und die entsprechenden Daten als Telemetriepakete am Ende der Verarbeitungskette ausgegeben.

Kapitel 1

Der Satellit INTEGRAL

Das *International Gamma–Ray Astrophysics Laboratory* (INTEGRAL) ist die nächste mittelgroße Mission innerhalb des *Horizont 2000* Programms der ESA. Der vier mal vier Meter breite und sechs Meter hohe Satellit ist für eine maximale Lebensdauer von fünf Jahren ausgelegt, die nominale Betriebsdauer wurde vorerst auf zwei Jahre festgelegt. Seine wissenschaftliche Nutzlast besteht aus vier Instrumenten. Das Spektrometer SPI und der Imager IBIS ergänzen sich in den beiden Aufgaben der hochauflösenden Spektroskopie und der genauen Ortsauflösung von Objekten im Bereich harter Röntgen- bis weicher Gamma–Strahlung. Der Röntgenmonitor JEM–X ermöglicht die Erweiterung dieses Energiebereichs in Richtung weicherer Röntgenstrahlung. Der Monitor OMC überwacht gleichzeitig das Verhalten der Röntgenquellen im optischen Licht. Tabelle 1.1 stellt die Schlüsselparameter der einzelnen Instrumente einander gegenüber.

	IBIS	SPI	JEM–X	OMC
Energieband:	15 kev – 10 MEV	20 keV – 8 MeV	3 keV – 35 keV	500 nm – 850 nm
Spektrale Auflösung:	7% bei 100 keV 6% bei 1 MeV	0.2% bei 1 MeV	19% bei 6 keV	
Gesichtsfeld:	9°	16°	4.8°	5°
Winkelauflösung:	12 arcmin	1.4°	3 arcmin	17.6 arcsec
Zeitauflösung:	61 μs	0.1 ms	10μs	s (variabel)

Tabelle 1.1: *Schlüsselparameter der Instrumente [4]*

Die Instrumente werden auf ein *Service Module* montiert, welches unter anderem die Positionierung des Satelliten übernimmt. Der rund sechs Tonnen schwere Satellit wird mit einer russischen PROTON–Rakete in eine hochexzentrische Umlaufbahn gebracht, welche es INTEGRAL ermöglicht, sich die meiste Zeit während eines Orbits weit außerhalb des Van Aalens–Gürtels[1] aufzuhalten. Bei der Durchquerung des Strahlungsgürtels müssen die empfindlichen Instrumente ausgeschaltet werden.

[1] Eng mit der Struktur des Erdmagnetfeldes verbundener Strahlungsgürtel. Von der Sonne oder aus dem interplanetaren Raum anfliegende elektrisch geladene Teilchen werden durch das irdische Magnetfeld eingefangen und bewegen sich auf korkenzieherartigen Bahnen entlang der Magnetfeldlinien zwischen den Polen.

	PROTON	ARIANE 5
Bodenstationen	Redu (Belgien)	Redu (Belgien)
	Goldstone (USA)	
Perigäum	10 000 km	7000 km
Apogäum	153 000 km	114 000 km
Inklination	51.6°	65.0°
Umlaufzeit	72 h	48 h
Zeit über 40 000 km	90%	85%
Zeit über 60 000 km	85%	75%

Tabelle 1.2: *Parameter der INTEGRAL–Orbits [4]*

Tabelle 1.2 stellt die Daten des PROTON–Orbits und die Umlaufbahnparameter der Ausweichrakete ARIANE 5 einander gegenüber. Der instrumentelle Aufbau des Satelliten wird in der Abbildung 1.3 auf Seite 20 dargestellt. Die Fotografie von INTEGRAL (Abb. 1, S. 8), entstanden während der strukturellen und thermischen Tests in ESTEC (Juni/Juli 1998) verdeutlicht die Ausmaße dieses Satelliten.

1.1 Wissenschaftliche Objekte und Ziele

Während der Sternentwicklung werden aus leichten Elementen die schwereren synthetisiert. Unter diesen gibt es auch radioaktive Elemente, die bei ihrem Zerfall Gamma–Strahlung emittieren. Die wichtigsten Beispiele werden in Tabelle 1.3 vorgestellt. Solange diese Elemente im Innern des Sterns verbleiben, sind sie unbeobachtbar. Aber die Sterne durchlaufen in ihrem Leben auch Phasen der Instabilität und einige dieser Elemente werden dann in den interstellaren Raum abgegeben. Bewegt sich die Halbwertszeit des Zerfalls in Zeitskalen von Tagen bis Jahren, so bleibt das Emissionsgebiet klein und die Astronomen sehen diese Objekte als Punktquellen. Aber es gibt auch radioaktive Elemente mit Halbwertszeiten in der Ordnung von Millionen Jahren. Diese Kerne haben viel Zeit, sich über große Volumina auszubreiten. Dem Beobachter erscheint diese Strahlung diffus.

1.1.1 Die 1,8 MeV Linie des ^{26}Al

Bei ausreichend hohen Temperaturen bildet sich aus ^{26}Mg durch das Einfangen eines Protons ^{26}Al über die Reaktion ^{26}Mg(p,γ)^{26}Al während des hydrostatischen Wasserstoffbrennens in massiven Sternen oder bei Explosionen wie Novae und Supernovae. Es zerfällt mit einer Halbwertszeit von etwa einer Million Jahren zu 82% durch Positronenemission und zu 18% durch Elektroneneinfang in einen angeregten Kernzustand des ^{26}Mg, welches durch Aussenden der 1.809 MeV Gamma–Linie in den Grundzustand zurückfällt. Die lange Lebensdauer führt zu einer Ansammlung und großräumigen Verteilung in der Galaxis. Frühere Beobachtungen zeigen, daß diese Linie sehr schmal ist, was auf einen Zerfall des Isotops im interstellaren Medium schließen läßt. INTEGRAL wird die vom Instrument COMTEL des Satelliten *Compton Gamma–Ray Observatory* (CGRO) erstellte Himmelskarte der 1.8 MeV Linie mit der 1.4° Winkelauflösung des SPI verfeinern, während IBIS die vom COMTEL entdeckten Hot Spots genauer untersuchen wird. Ein Vergleich der gemessenen Intensitätsverteilung mit der räumlichen Verteilung möglicher galaktischer Quellen sollte es ermöglichen, die dominierende Natur dieser Linienstrahlung zu identifizieren.

Zerfallskette	Halbwertszeit (Jahre)	Emission (MeV)	Entstehung
$^{56}Ni \rightarrow ^{56}Co \rightarrow ^{56}Fe$	0.31	e^+, 0.847, 1.238, 2.599	Supernovae
$^{22}Na \rightarrow ^{22}Ne$	3.8	e^+, 1.275	Novae
$^{44}Ti \rightarrow ^{44}Sc \rightarrow ^{44}Ca$	78	e^+, 0.068, 0.078, 1.156	Supernovae
$^{26}Al \rightarrow ^{26}Mg$	1.1 x 10^6	e^+, 1.809	Wasserstoffbrennen (Protoneneinfang), Novae, Supernovae
$^{60}Fe \rightarrow ^{60}Co \rightarrow ^{60}Ni$	2.2 x 10^6	0.059, 1.173, 1.332	Heliumbrennen (Neutroneneinfang)

Tabelle 1.3: *Beispiele für radioaktive Zerfallsketten [1]*

Denn sollte sich herausstellen, daß die massiven Sterne die Hauptquellen des beobachteten ^{26}Al sind, kann man anhand des Anteils des auch sehr langlebigen Isotops ^{60}Fe im interstellaren Medium, welches ebenfalls in massiven Sternen durch Neutronenanlagerung beim Heliumbrennen entsteht, die gängigen Nukleosynthesemodelle überprüfen [27, 1, 10].

1.1.2 Die 511 keV Linie

Radioaktive Zerfälle können neben Gammastrahlung auch Positronen (β–Zerfall) emittieren. Findet die Emission in Gebieten geringer Materiedichte statt, können die Positronen diesen Regionen entkommen und über weite Strecken fliegen, bis sie bei der Annihilation mit Elektronen zwei 511 keV γ–Quanten erzeugen. Diese Linienstrahlung erscheint dann aufgrund der weiträumigen Verteilung der zugrundeliegenden Positronen diffus. Die Hauptquelle der Positronen scheint die ^{56}Ni- oder ^{44}Ti–Zerfälle bei Supernovae Typ I–Prozessen zu sein. Die Beobachtung dieser Strahlung läßt daher Rückschlüsse auf den Anteil erzeugter Positronen bei diesen Zerfällen und auf die Ereignisrate und galaktische Verteilung der Supernova–Explosionen vom Typ I innerhalb der vergangenen 10^6 Jahre zu. Breite und Form dieser Linie geben Aufschluß über den Zustand des interstellaren Gases, in dem die Positronen vernichtet wurden. Mit der guten Ortsauflösung von INTEGRAL hofft man auch der Frage nach möglichen Punktquellen für die 511 keV Linie, wie sie in Richtung des galaktischen Zentrums vermutet werden, näher zu kommen. Die Vermessung der Energie, Intensität und Linienform läßt wesentliche Rückschlüsse auf die physikalischen Zustände in der Nähe des Objekts zu.

Die 511 keV Photonen können an einer Akkretionsscheibe, die ein akkretierendes Objekt in einer Entfernung von 5 – 50 Schwarzschildradien besitzen kann, gestreut werden. Der Comptoneffekt beschreibt die Streuung von Photonen an Elektronen. Dabei ist die Energie E_2 des gestreuten Photons abhängig vom Streuwinkels σ und der Energie E des Photons vor der Streuung.

$$E_2 = \frac{E}{1 + \frac{E*(1-\cos\sigma)}{m_e c^2}}$$

Wird ein 511 keV Photon um 180° comptongestreut, so besitzt das gestreute Photon eine Energie von 170 keV. Diese 170 keV Linienstrahlung wurde bereits vom *Compton Gamma–Ray Observatory* registriert und soll von INTEGRAL bestätigt und genau vermessen werden [27, 1, 10].

1.1.3 Novae

Bei einem explosiven Wasserstoffbrennen an der Oberfläche eines akkretierenden neonreichen Weißen Zwerges in einem Doppelsternsystem werden mittelschwere Kerne durch Protoneneinfang erzeugt. Das Isotop ^{22}Na zerfällt mit 3.8 Jahren Halbwertszeit in einen kurzlebigen angeregten Zustand des ^{22}Ne, welches durch Emission der 1.275 MeV–Linie in den Grundzustand wechselt. INTEGRAL wird historische Novae als helle Flecken im Lichte dieser Linie entlang der Milchstraße bis in eine Entfernung von einigen Kiloparec identifizieren [27, 1, 10, 20].

1.1.4 Supernovae

Supernovae (SN) werden in zwei Hauptklassen eingeteilt. Die Supernova Typ II ist das Resultat eines Kernkollapses und wird allein von der Gravitationsenergie angetrieben, während die Supernova Typ Ia eine thermonukleare Explosion eines Weißen Zwerges nahe der Chandrasekarmasse in einem Doppelsternsystem darstellt. In beiden Typen werden radioaktive Elemente der Eisengruppe produziert und in den interstellaren Raum abgegeben. Die interessantesten Linien sind die Zerfallsketten von ^{56}Ni zu ^{56}Fe sowie von ^{44}Ti zu ^{44}Sc. Die kurze Lebensdauer des ^{56}Co von nur etwa 79 Tagen läßt die entsprechenden Linien bereits nach wenigen Monaten bis Jahren verschwinden. Zum ersten Mal beobachtete man die Linien des ^{56}Co bei 0.847, 1.238 und 2.599 MeV in der Supernova 1987A und bewies damit die Existenz dieser Prozesse bei Supernovae vom Typ II. Die Strahlungen der Zerfallsreihe von ^{44}Ti werden auch noch nach dem INTEGRAL–Start im Jahre 2001 eine Intensität besitzen, die es ermöglichen wird, diese Linien bei 68 keV und 78 keV zu beobachten. Wegen der Absorption der Strahlung durch massereiche Elemente der explodierenden Sternhülle und wegen der relativ kleinen Menge von ^{56}Ni, die während der SN 1987A erzeugt wurde, sind die Intensitäten von Supernovae Typ II geringer, als sie bei Supernovae des Typs I erwartet werden. INTEGRAL wird historische Supernovae in der Michstraße aufspüren, die bislang durch starke Absorption in anderen Wellenlängenbereichen unentdeckt blieben, denn die bis zu 500 Jahre alten Supernovae zeigen sich im Lichte der ^{56}Ti–Linien als helle Flecken. Auch extragalaktische Supernovae werden wichtige Beobachtungsziele sein [27, 1, 10, 20].

1.1.5 Kompakte Objekte

Neutronensterne und Schwarzlochkandidaten gehören zu den wichtigsten Beobachtungsobjekten von INTEGRAL. Zur Klasse der Neutronensterne gehören die Pulsare wie der Crab- oder Velo–Pulsar, die im Gammabereich studiert werden können. Schon das *Compton Gamma–Ray Observatory* entdeckte weitere Pulsare im Gammabereich. Mit der guten Richtungsauflösung und hohen Empfindlichkeit wird INTEGRAL in der Lage sein, die galaktische Ebene nach weiteren schwachen Pulsaren abzusuchen und in einem weiten Energiebereich von 3 keV bis 10 MeV zu analysieren. Signifikante Unterschiede im spektralen Verlauf und in der zeitlichen Variabilität der Lichtkurven dieser Pulsare zeigen, daß gerade der Energiebereich kleiner 10 MeV geeignet ist, verschiedene Pulsar–Modelle zu testen.

Bei Neutronensternen in Röntgen–Doppelsternsystemen sollen insbesondere die Zyklotronlinien, hervorgerufen durch extrem starke Magnetfelder und das hochenergetische Kontinuum oberhalb von 100 keV untersucht werden. Auch gilt es, die Unterschiede in den Spektren von

Neutronensternen und Schwarzlochkandidaten in Doppelsternsystemen genauer zu charakterisieren. Schwarzlochkandidaten scheinen ein signifikant härteres Spektrum zu besitzen als Neutronensterne, ob aber das Maximum ihrer Leuchtkraft tatsächlich im weichen Gammabereich liegt, soll durch INTEGRAL verifiziert werden. Des weiteren bietet IBIS mit seiner Möglichkeit zu spektraler/zeitlicher Analyse die richtigen Voraussetzungen zur Untersuchung von Quasi–Periodischen Oszillationen (QPO) der inneren Akkretionsscheiben bestimmter Schwarzlochkandidaten [27, 1, 20, 26].

1.1.6 Extragalaktische Systeme

Resultate von CGRO zeigen, daß die Gamma–Spektren der Kerne Aktiver Galaxien (AGN) weit komplexer sind als früher angenommen. Die Spektren von Seyfert–Galaxien brechen häufig oberhalb von 100 keV ab, während sich die der Gamma–Quasare und BL–Lac–Objekte bis weit in den GeV–Bereich erstrecken. Alle haben Potenzgesetz–Spektren ($E^{-\alpha}$) mit Index α zwischen 1.7 und 2.4 und sind zudem zeitlich variabel. INTEGRAL wird mehr als 100 dieser AGN–Objekte untersuchen und mit Breitband- und Linienspektroskopie simultan in einem Energiebereich von 3 keV – 10 MeV Wesentliches zum Verständnis dieser Quellen beitragen [27, 1, 20, 26].

1.2 Der Imager on Board of INTEGRAL Satellite

Der *Imager on Board of INTEGRAL Satellite* (IBIS) besteht aus drei Detektoreinheiten. Der *INTEGRAL Soft Gamma–Ray Imager* (ISGRI) und das *Pixellated Imaging Cäsium Iodide Telescope* (PICsIT) sind als Detektorebenen senkrecht zur Beobachtungsrichtung genau übereinander in 94 mm Abstand[2] ausgerichtet. Um die beiden Detektorebenen herum wurde die dritte Detektoreinheit VETO gebaut, die als aktiver Schild den Hintergrund der beiden Detektoren ISGRI und PICsIT reduzieren soll. Die Winkelauflösung des Instruments wird im Zusammenspiel mit einer kodierten Aperturmaske erreicht, welche sich 3400 mm über den Detektoren befindet. Tabelle 1.4 stellt die wichtigsten Instrumentenwerte von IBIS zusammen.

Energiebereich	20 keV – 10 MeV
Detektorfläche	2621 cm^2 (CdTe), 2994 cm^2 (CsI)
Gesichtsfeld	9° vollkodiert, 19° FWHM
Winkelauflösung	12 arcmin
Punktquellenauflösung (10 σ Quelle)	ISGRI $\cong$ 1.0 arcmin, PICsIT $\cong$ 2.5 arcmin
Spektrale Auflösung	$<$ 8% bei 100 keV, 6% bei 1 MeV
Zeitauflösung	61 μs (Photon by Photon Mode)

Tabelle 1.4: *Parameter von IBIS [6]*

Abbildung 1.1 zeigt die Detektoreneinheit des IBIS–Instruments. Gut zu erkennen sind die Rahmen der beiden Detektorebenen von ISGRI und PICsIT. Dazwischen sieht man die zylindrischen Boxen der Photomultiplier des VETO–Schilds.

[2] Von Oberfläche zu Oberfläche gemessen.

Abbildung 1.1: *Der Imager*

1.2.1 Der INTEGRAL Soft Gamma–Ray Imager

Der *INTEGRAL Soft Gamma–Ray Imager* (ISGRI) besteht aus 16 384 einzelnen und unabhängigen Halbleiterkristallen aus Cadmium–Tellurid (CdTe–Pixel). Je vier Pixel sind mit einem ASIC[3] verbunden, vier ASICs und folglich sechzehn Pixel bilden die kleinste Detektoreinheit, eine Polyzelle. Einhundertundachtundzwanzig (128) Polyzellen ergeben zusammen eine *Modular Detection Units* (MDU), 8 Module den gesamten Detektor. Tabelle 1.5 verdeutlicht noch einmal den Aufbau des ISGRI–Layers.

Pixelgröße (CdTe Kristall)	4 x 4 mm^2	Dicke: 2 mm
Abstand zwischen den Pixeln	0.6 mm	Mitte zu Mitte: 4.6 mm
Kleinste Einheit: Polyzelle	16 Pixel	Anordnung: 4 x 4 Pixel
MDU	128 Polyzellen	Anordnung: 64 x 32 Pixel
Layer	8 MDUs	128 x 128 Pixel
Gesamte sensitive Fläche	2621 cm^2	128 x 128 x 16 mm^2

Tabelle 1.5: *Aufbau des ISGRI–Layers [6]*

Bei der Detektion von Gamma–Strahlung werden durch inneren Photoeffekt Ladungen (Elektron–Loch–Paare) freigesetzt. Während des Auslesevorgangs gehen Ladungen verloren. Bei dieser Art Halbleiter ist es ein signifikanter Effekt, daß sehr viel mehr Löcher als Elektronen

[3]Application Specific Integrated Circuit.

verloren gehen, wodurch die Response des Detektors asymmetrisch wird. Der Ladungsverlust ist abhängig von der Zeitdauer, in der die Ladungen freigesetzt werden, der Risetime. Die Messung der Risetime erlaubt also eine Abschätzung des Ladungsverlusts und eine Korrektur der ausgelesenen Amplitude. Eine Elektronik, die *Charge Loss Correction* (CLC), führt erfolgreich diese Korrektur durch, jedoch nur bis zu Risetime–Zeiten kleiner als zwei Mikrosekunden. Ereignisse mit größeren Risetime–Werten sind für die Gewinnung spektroskopischer Informationen weniger geeignet, da die CLC–Elektronik den Ladungsverlust bei diesen Ereignissen nicht mehr korrigieren kann. Ein Verwerfen dieser Ereignisse bedeutet jedoch einen Verlust von Detektoreffizienz, die wegen der geringeren Ereignisraten gerade oberhalb von 150 KeV – 200 keV wichtig ist. Die Ereignisse mit Risetime–Werten größer als 2 μs werden dehalb ebenfalls an den HEPI weitergegeben.

Die optimale Betriebstemperatur liegt zwischen 0°C und 10°C, die der Detektor im Betrieb allein durch seine Verlustwärme erreicht. In ausgeschaltetem Zustand wird er geheizt, damit die Temperatur annähernd konstant bleibt. Die geringe Dicke der CdTe Pixel beschränkt den Einsatz des Detektors auf den unteren Energiebereich von 15 keV – 200 keV, in Verbindung mit PICsIT erweitert sich der Bereich bis zu 700 keV. Die spektrale Auflösung von ISGRI liegt bei etwa 7% (100keV). Durch die geringe Pixelgröße erreicht ISGRI im Zusammenspiel mit der kodierten Aperturmaske die maximale Winkelauflösung von IBIS. Die Erläuterung hierzu folgt in Kapitel 1.2.4 [6, 9].

1.2.2 Pixellated Imaging Cäsium Iodide Telescope

Das *Pixellated Imaging Cäsium Iodide Telescope* (PICsIT) besteht aus 4096 einzelnen und unabhängigen Szintillationskristallen aus Cäsium–Iodid (CsI). Die kleinste Detektoreinheit bilden 16 Pixel, die an einem ASIC angeschlossen sind. Jedes der acht PICsIT–Module faßt 32 ASICs, also 512 Pixel, zusammen. Tabelle 1.6 verdeutlicht noch einmal den Aufbau des PICsIT–Layers.

Pixelgröße (CsI Kristall)	8.55 x 8.55 mm^2	Dicke: 30 mm
Abstand zwischen den Pixeln	0.65 mm	Mitte zu Mitte: 9.2 mm
Kleinste Einheit	16 Pixel	Anordnung: 4 x 4 Pixel
Modul	512 Pixel	Anordnung: 32 x 16 Pixel
Layer	8 Module	64 x 64 Pixel
Gesamte sensitive Fläche	2994 cm^2	64 x 64 x Pixelgröße

Tabelle 1.6: *Aufbau des PICsIT–Layers [6]*

Ein Gamma–Photon erzeugt innerhalb des CsI–Kristalls Szintillationslicht. Dieses Licht wird mit Photodioden in elektrische Impulse umgewandelt. Die Dicke der CsI–Kristalle von 30 mm wurde auf die Empfindlichkeit bei 511 keV optimiert, wobei eine hohe Detektoreffizienz im MeV–Bereich gewährleistet bleibt. Die optimale Betriebstemperatur liegt ähnlich wie bei ISGRI bei 0°C bis 10°C. Wie ISGRI braucht auch PICsIT nur in ausgeschaltetem Zustand geheizt zu werden, um das Auskühlen zu vermeiden. Wegen der größeren Pixel ist die Winkelauflösung etwas geringer als bei ISGRI [6, 7].

1.2.3 VETO–Schild und Kalibrationsquelle

Mit Ausnahme der Seite, die in Beobachtungsrichtung zeigt, sind die Detektoren ISGRI und PICsIT von einem aktiven Hintergrundschild, VETO genannt, umgeben. Wie auch die Detektorarrays hat VETO einen modularen Aufbau aus Bismuth–Germanium–Szintillatoren. Zwei Module sind an jeder der vier Seiten und acht weitere Module auf der Unterseite der Detektoren angebracht. Jeder Kristall wird von zwei Photomultipliern beobachtet. Aufgrund der hohen Dichte der Kristalle und der hohen Ordnungszahl von Bismut wird mit einer Kristalldicke von 20 mm bereits eine ausreichende Schildwirkung erzielt. VETO hat die Aufgabe, die den Ereignissen zugrundeliegenden elektrisch geladenen Teilchen und Photonen aus dem Datenstrom von ISGRI und PICsIT herauszufiltern, die nicht aus der Beobachtungsrichtung stammen und soll dadurch den Hintergrund dieser Detektoren und somit auch die Datenrate des Instruments verringern. Dieser Hintergrund besteht zum einen aus der diffusen kosmischen, zum anderen aus der im Satelliten selbst erzeugten Strahlung. VETO wird mit seinen Ereignissen Spektren erstellen und ermöglicht dadurch auch das Studium dieser Hintergrundstrahlung [6, 20].

Eingebettet in einen Szintillatorkristall und durch einen zusätzlichen Wolframschild in Richtung Spektrometer und Röntgenmonitor abgeschirmt ist eine Kalibrationsquelle aus ^{22}Na. Zwei Photomultiplier registrieren jedes Kalibrationsereignis. Durch eine Koinzidenzmessung zwischen diesem Kalibrationssystem und den Ereignissen in den Detektoren können diese auch von ISGRI und PICsIT als Kalibrationsereignisse erkannt und markiert werden. Die erzeugten Linien bei 511 keV und 1.275 MeV dienen zur Eichung der beiden Detektoren während des Fluges [6].

1.2.4 Kodierte Aperturmaske

Weiche Röntgenstrahlung läßt sich durch streifenden Einfall mit Hilfe sehr glatter Oberflächen fokusieren. Dadurch ist es möglich, abbildende Instrumente, wie z.B. die Wolter–Teleskope des Satelliten ROSAT, zu bauen. Bei harter Röntgen- und bei Gamma–Strahlung ist dies nicht mehr möglich, da die Wellenlänge der Strahlung die Größenordnung der Atomdurchmesser erreicht. Um dennoch Abbildungen des Himmels zu erhalten, bedient man sich kodierter Aperturmasken. Die Masken sind aus undurchlässigen und durchlässigen Flächen, ähnlich einer Lochmaske, aufgebaut. Jedes Pixel der Detektorarrays hat durch die offenen Flächen der Maske eine etwas andere Projektion des Himmelsausschnitts als seine Nachbarpixel. Abbildung 1.2 versucht, dieses Prinzip schematisch zu verdeutlichen. Mit Hilfe mathematischer Verfahren lassen sich die Meßergebnisse der einzelnen Projektionen zu einer Abbildung des beobachteten Himmelssegments zusammensetzen. Die Winkelauflösung wird durch den Abstand der Maske vom Detektor sowie von der Größe der Maskenelemente bestimmt. Die Distanz zwischen Detektor und Maske unterliegt lediglich bautechnischen Restriktionen des Satelliten. Um eine gute Abbildungsqualität zu gewährleisten, sollten die Maskenelemente größer sein als die Detektorpixel. Letztendlich hängt die Aufösung also von der räumlichen Ausdehnung der Pixel ab. IBIS erreicht eine Winkelauflösung von 12 arcmin. Die 15 mm dicke kodierte Aperturmaske besteht aus 95 x 95 Quadraten mit 11.2 mm Seitenlänge. Etwa die halbe Maskenfläche ist durchsichtig (Transparenz von 60% bei 20 keV), die andere ist abdeckend (Opazität von 70% bei 1.5 MeV) [6, 9].

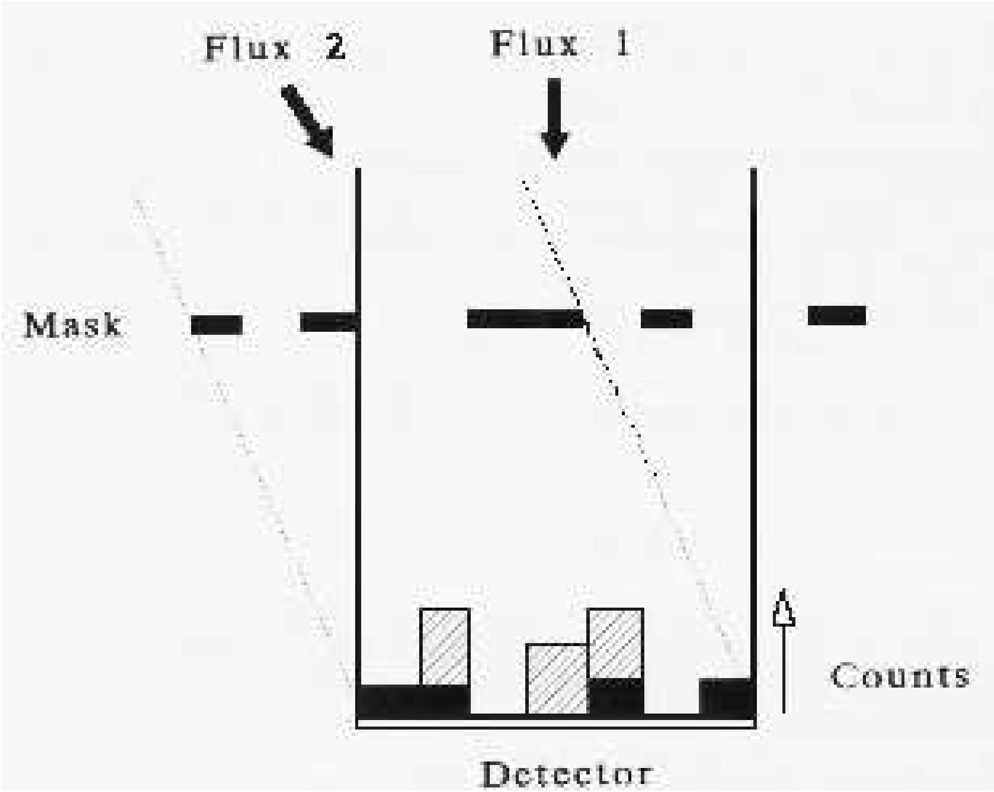

Abbildung 1.2: *Funktionsweise der Aperturmaske*

1.3 Das Spectrometer for INTEGRAL

Das *Spectrometer for INTEGRAL* (SPI) ist ausgelegt für die hochauflösende Linienspektroskopie im Gammabereich von 20 keV bis 8 MeV. Es besteht aus 19 hexagonalen Germanium–Kristallen mit 3.2 cm Seitenlänge und 17 cm Höhe, die zu einem ebenfalls hexagonalen Detektorarray mit 505.5 cm^2 Gesamtfläche angeordnet sind. Das Detektorarray befindet sich in einem Kyrostaten, der die Germaniumkristalle auf die optimale Betriebstemperatur von etwa 85° K abkühlt. Um den intern produzierten Hintergrund so gering wie möglich zu halten, werden alle Bauteile, die den Germaniumdetektor umgeben, aus Material mit niedriger Massenzahl, hauptsächlich aus Beryllium, gebaut. Wie der Imager verwendet auch SPI einen aktiven VETO–Schild zur Reduzierung des Hintergrunds sowie eine kodierte Aperturmaske, 171 cm oberhalb der Detektoroberfläche, zur Abbildung. Die Schlüsselparameter des Instruments wurden bereits in Tabelle 1.1 auf Seite 11 vorgestellt [10].

1.4 Der Joint European X–Ray Monitor

Der *Joint European X–ray Monitor* (JEM-X) für den Energiebereich von 3 – 35 MeV besteht aus zwei identischen Microstrip–Hochdruck–Gaskammern, die mit Xenon (5 bar) gefüllt sind. Jede betrachtet den Himmel durch eine 3.2 m entfernte kodierte Aperturmaske. JEM–X wird gleichzeitig mit den beiden Hauptinstrumenten Beobachtungen durchführen und eine wichtige Rolle bei der Identifikation, Analyse und wissenschaftlichen Interpretation von Gamma–Quellen spielen. Die Schlüsselparameter des Instruments wurden bereits in Tabelle 1.1 auf Seite 11 vorgestellt [5].

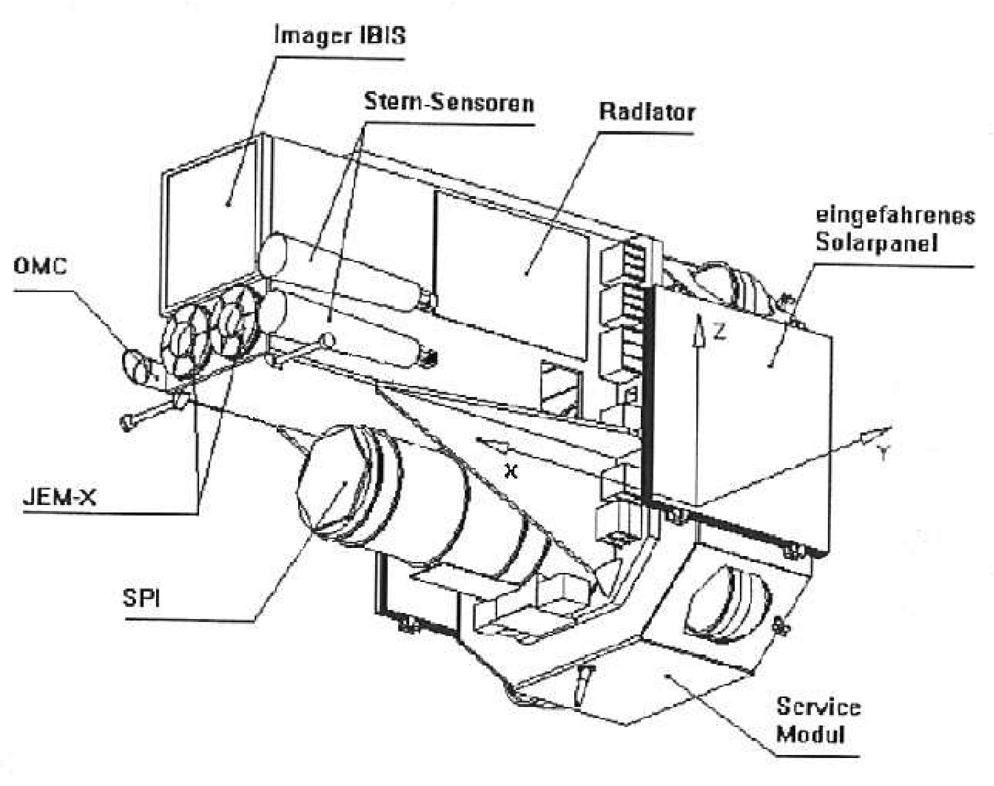

Abbildung 1.3: *Skizze von INTEGRAL*

1.5 Die Optical Monotoring Camera

Ergänzend zu den drei schon beschriebenen Instrumenten wird die *Optical Monotoring Camera* (OMC) das optische Verhalten der Gamma–Quellen aufzeichnen. Sie besteht aus einer CCD in der Fokalebene einer 50 mm Linse. Die CCD (1024 x 2048 Pixel) verwendet einen Teil (1024 x 1024) zur Aufnahme und den anderen Teil zum Verschieben des aufgenommenen Bildes vor dem Auslesen. Die Grenzgröße der Kamera liegt etwa bei 19.5-ter Größenklasse. Die Schlüsselparameter des Instruments wurden bereits in Tabelle 1.1 auf Seite 11 vorgestellt.

Die OMC bietet zum ersten Mal die Möglichkeit, Langzeitbeobachtungen im optischen Band gleichzeitig zu Röntgen- und Gammaobservationen durchzuführen. Solche Multiwellenlängenbeobachtungen sind besonders in der Hochenergie–Astrophysik wichtig, in der sich Veränderlichkeit typischerweise auf schnellen Zeitskalen abspielt [5].

1.6 Dithering

Bei der genauen Beobachtung, Pointing genannt, bleibt der Satellit nicht über die gesammte Beobachtungszeit bewegungslos auf die Quelle ausgerichtet, sondern er fährt ein bestimmtes Pointierungsmuster um die Quelle herum ab, wobei er sich auch um die Achse in Beobachtungsrichtung drehen wird. Diese Art der Pointierung heißt Dithering (engl. Zittern). Es sind zwei Pointierungsmuster vorgesehen:

- Bei punktförmigen Quellen werden außer der zentralen Ausrichtung auf das Objekt die Ecken eines Sechsecks um die Quelle herum abgefahren.
- Bei ausgedehnten Quellen wird ein Quadrat mit 5 x 5 Punkten, in dessen Zentrum sich die Quelle befindet, verwendet.

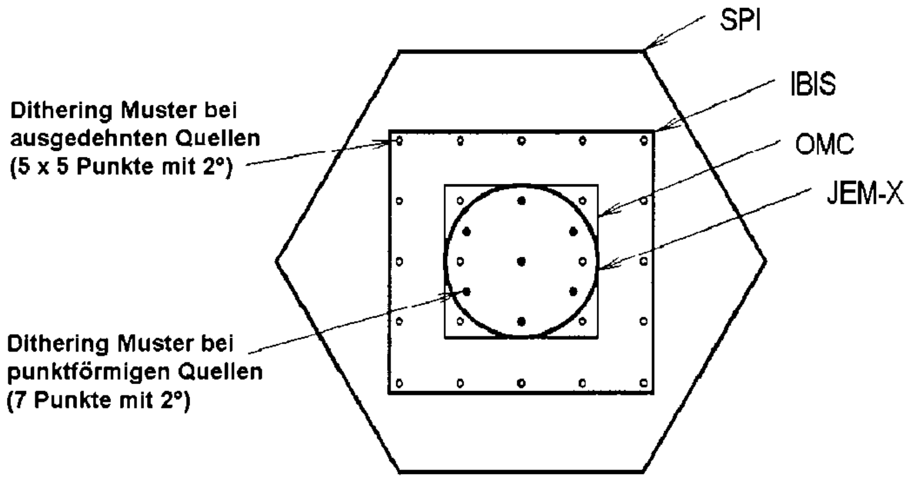

Abbildung 1.4: *Das Dithering–Muster*

Der Abstand zwischen den einzelnen Beobachtungsrichtungen beim Dithering beträgt für beide Modi etwa 2°, die Verweildauer bei jedem der Ditheringpunkte liegt bei 20 Minuten. Die Abbildung 1.4 zeigt die beiden Dithering–Verfahren im Verhältnis zu den Gesichtsfeldern der einzelnen Instrumente. Die ausgefüllten Punkte (Kreise) stellen das Muster bei punktförmigen (ausgedehnten) Quellen dar. Das Dithering dient der Minimierung der Abbildungsfehler des Spektrometers, die anderen Instrumente führen diese Bewegung jedoch ebenfalls aus. IBIS nutzt das Dithering ebenfalls zur Reduzierung der systematischen Fehler. Andererseits verringert sich das vollkodierte Gesichtsfeld, die Beobachtungszeit während der Bewegung (Slew) des Satelliten zu den Ditheringpunkten kann nicht zum Aufnehmen von Histogrammen verwendet werden, und im Gegensatz zu Punktquellen, die auch jederzeit im verkleinerten vollkodierten Gesichtsfeld verbleiben, geht bei ausgedehnten Quellen Intensität verloren. Aufgrund des kleineren Gesichtsfeldes wird JEM–X rund 25% seiner Effizienz bei Punktquellen und nahezu 67% bei ausgedehnten Quellen durch das Dithering einbüßen [4, 21, 26].

Kapitel 2

Vorgaben

2.1 Die Aufgabenstellung

In den Detektoren ISGRI und PICsIT werden γ–Photonen als Ereignisse registriert. Die Ereignisse der beiden Detektoren gelangen auf ihrem Weg im Datenstrom in den *Hardware Event Preprocessor for IBIS* (HEPI). Hier werden die Ereignisse als verschiedene Eventarten markiert und vorverarbeitet. Vom HEPI werden die Events an die *Data Processing Electronics* (DPE) weitergeleitet. Die DPE ist der eigentliche Bordcomputer für die Steuerung und Datenverarbeitung des IBIS–Instruments. In der DPE sind die *Common Service Software* (CSSW) und die *Instrument Application Software* (IASW) untergebracht. Die IASW ist der Teil der Software, der die Steuerung und Datenverarbeitung realisiert. Die CSSW stellt dazu Routinen für die Kommunikation mit der Satellitenumgebung sowie weitere vorgefertigte Programmroutinen bereit. Die von der IASW generierten Telemetriepakete werden über das *On Board Data Handling* (OBDH) System der *Central Data Management Unit* (CDMU) übergeben. Die Daten werden schließlich von der Sendestation des Satelliten zur Erde geschickt und im *INTEGRAL Science Data Cente*r (ISDC) archiviert. Eine detaillierte Beschreibung des Datenwegs wird nochmals in Kapitel 3.3 auf Seite 38 gegeben.

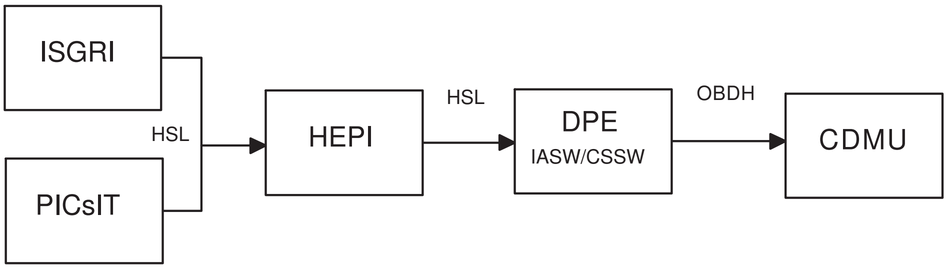

Abbildung 2.1: *Der Datenweg im Satelliten INTEGRAL*

Im Rahmen zweier Diplomarbeiten sollte eine *Instrument Application Software* für die *Data Processing Electronics* entwickelt werden, die sowohl die Steuerung des IBIS–Instruments übernimmt als auch die Echtzeit–Datenverarbeitung der IBIS–Daten durchführt. Die IASW sollte in der Programmiersprache ADA geschrieben und in einem Software–Simulator sowie auf der realen Hardware getestet werden.

2.2 Vorhandene Hardware und Software

Die in den beiden IBIS–Detektoren registrierten Ereignisse müssen aus Gründen der Datenreduktion bereits an Bord des Satelliten vorverarbeitet werden. Wie jedem Instrument steht IBIS dazu ein strahlungsfester Computer, die DPE, und eine *Common Service Software* (CSSW) zur Verfügung. In der DPE wird die Echtzeit–Datenverarbeitung von der *Instrument Application Software* (IASW) durchgeführt. Die IASW wiederum muß, um die schnellstmögliche Arbeitsgeschwindigkeit erreichen zu können, optimal an die vorhandene Hardware angepaßt sein. Diese Anpassungen sind, was das Ansteuern der verschiedenen Interfaces betrifft, bereits in der CSSW realisiert, die auch noch weitere Hilfsprogramme enthält.

2.2.1 Der Hardware Event Preprocessor for IBIS

Die DPE unterliegt aufgrund ihrer strahlungsfesten Eigenschaften starken Einschränkungen in der Rechenleistung. Schon frühzeitig hat man bei der Entwicklung des IBIS Instruments festgestellt, daß die hauptsächlich durch PICsIT anfallenden sehr hohen Datenraten nicht allein durch eine softwaregestützte Datenverarbeitung innerhalb der DPE bewerkstelligt werden können. Ebenso würde die zur Verfügung stehende Telemetrierate um ein Vielfaches überschritten. Es ist daher notwendig, die wissenschaftlichen Daten mittels einer Hardware, die viel höhere Verarbeitungsgeschwindigkeiten als eine Software erlaubt, je nach Forschungsschwerpunkt gezielt auszuwählen und zusammenzufassen. Dieser *Hardware Event Preprocessor for IBIS* (HEPI), der wie die IASW in Tübingen entwickelt wird, befindet sich ebenfalls innerhalb der DPE–Box. Die Kernstücke bilden eine strahlungsfeste integrierte Schaltung, ein sogenannter *Application Specific Integrated Circuit* (ASIC) und ein großer Speicherbereich. Der HEPI verfolgt ein Pipeline–Konzept für die Datenverarbeitung. Da der HEPI die Datenströme der beiden Detektoren nicht nur unterschiedlich verarbeitet, sondern auch synchronisiert, gibt es für jeden Detektor–Datenstrom eine eigene Pipeline, die beide von einem Controller gesteuert werden. In diesen Pipelines sind verschiedene Module vorhanden, die über ein spezielles Register, das Status Register, ein- und ausgeschaltet werden können, wobei nur bestimmte Kombinationen sinnvoll sind. Je nach Zusammenstellung dieser Module werden die vier verschiedenen Betriebsmodi des HEPI (Transparent Mode, Standard Submode, Polarimetry Submode, Photon by Photon Submode) realisiert. Die Modi des HEPI und diejenigen der IASW können unabhängig voneinander gesetzt werden. Ihre Bedeutung und die äquivalenten Modi der IASW werden in Kapitel 3.2 näher beschrieben.

Der HEPI bildet für die wissenschaftlichen Daten die Schnittstelle zwischen den Detektoren ISGRI und PICsIT und der DPE . Er hat vielfältige Aufgaben zu erfüllen:

- Die Detektorereignisse werden vom HEPI bestimmten Eventarten zugeordnet. Dabei wird das Format der Daten von 64 Bit auf 80 bzw. 160 Bit erweitert. Tabelle 3.2 schlüsselt die von HEPI erzeugten Formate der einzelnen Eventarten auf.

- Die Zeitauflösung wird von 238.4 ns innerhalb 2 Sekunden auf 61.035 μs innerhalb von 72.8 Stunden geändert.

- Die gemessenen Amplituden der CsI–Detektoren werden korrigiert.

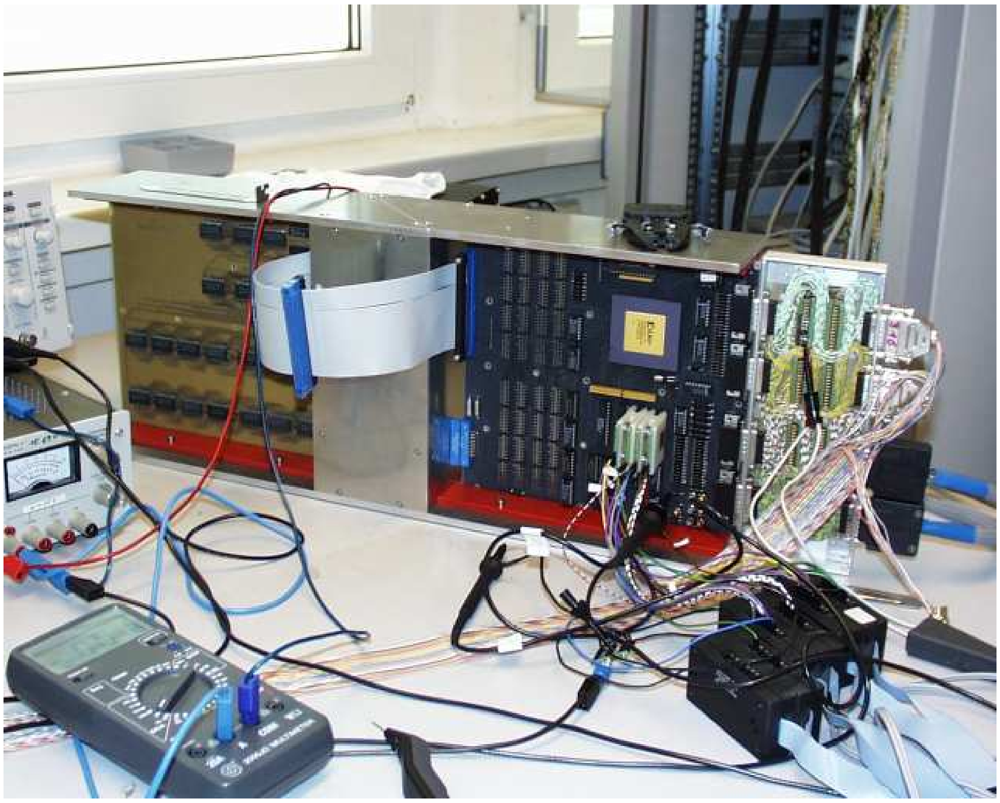

Abbildung 2.2: *Der HEPI*

- Es wird nach Zeitkoinzidenzen zwischen ISGRI und PICSIT–Ereignissen gesucht. Gleichzeitige Ereignisse beider Detektoren können durch ein comptongestreutes Photon erzeugt werden. Deshalb werden beide Ereignisse zu einem Compton–Event zusammengefaßt.

- Die PICSIT–Ereignisse werden in Histogrammen aufintegriert. Eine Erläuterung hierzu findet in den Kapiteln 3.1 und 3.2.5 statt.

Abbildung 2.2 zeigt den HEPI in seinem Testrahmen. Das linke Board enthält die Speicherbausteine, in denen die Histogramme abgelegt sind. Das rechte Board ist der eigentliche Prozessorteil. Der große quadratische Chip ist der nur während der Entwicklungsphase verwendete FPGA[1], der für das Flugmodell durch einen ASIC[2] ersetzt wird.

2.2.2 Die Data Processing Electronics

Die Recheneinheit, welche die Steuerung des Instruments und die Datenverarbeitung auf dem Satelliten übernehmen wird, heißt *Data Processing Electronics* (DPE) und wird jeder Instrumentengruppe von der ESA zur Verfügung gestellt. Herzstück der DPE ist ein MIL-STD-3-1750A Mikroprozessor mit einer Taktfrequenz von 13 MHz. Das Betriebssystem des Prozessors ist ein ASTRES 1750 real time multi–tasking System. Auf dem Prozessorboard befinden sich des wei-

[1] Field Programmable Gate Array.
[2] Application Specific Integrated Circuit.

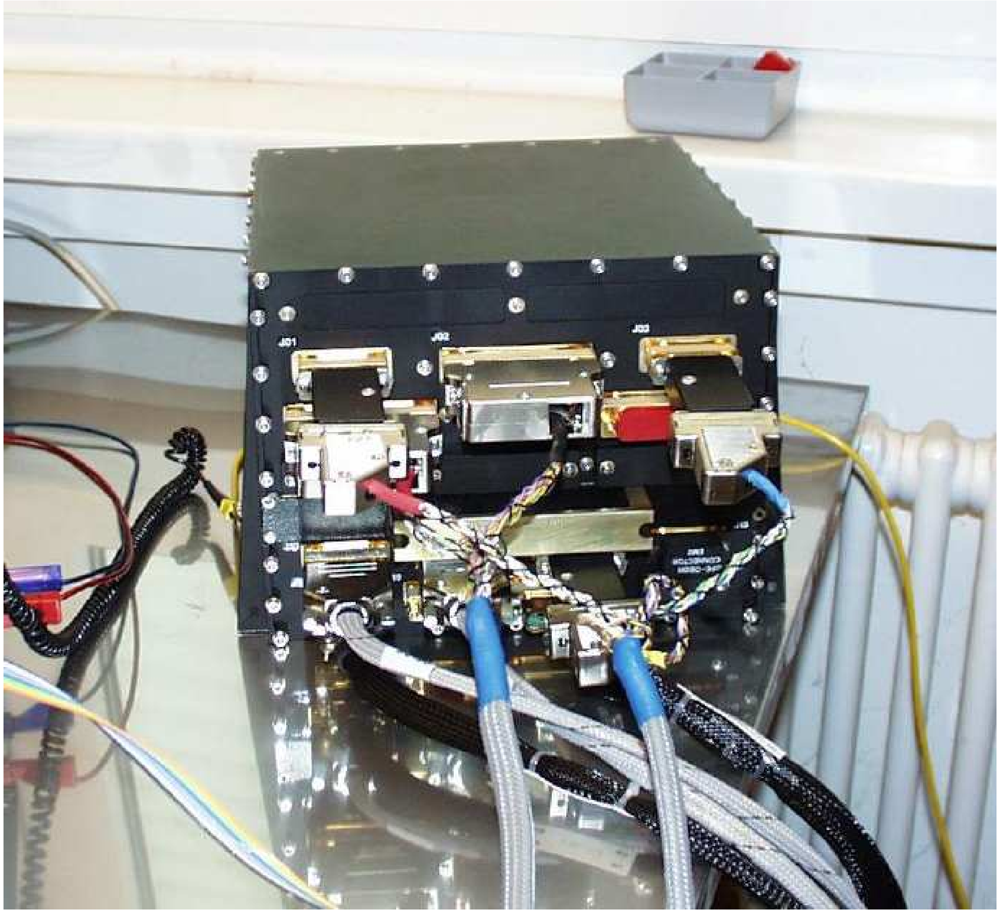

Abbildung 2.3: *Die DPE*

teren der RAM[3] Speicher mit SECDED[4] und die internen CPU Funktionen des Built-In-Self-Test und des Watch-dog. Der Watch-dog ist eine Absicherung bei einem Softwareabsturz. Er muß regelmäßig von der Software aufgerufen werden. Geschieht dies nicht mehr, wird ein Systemabsturz angenommen und die Software muß neu gestartet werden. Der ROM[5]–Speicher mit der *Common Service Software* und der *Instrument Application Software* (maximal 48 kword) ist auf einem separaten Board untergebracht. Die DPE besitzt bidirektionale Interfaceleitungen zu jedem der Detektoren. Alle wissenschaftlichen und Housekeeping–Daten von ISGRI , PICsIT und VETO werden gesammelt, in Telemetriepakete verpackt und dem *On Board Data Handling* (OBDH) System bereitgestellt. Andererseits gibt sie die Telekommandos, das Taktsignal und weitere Synchronisationssignale an die Detektoren weiter. Diese sind:

- Die mit acht multiplizierte OBDH Bus Clock: 4.19304 MHz.
- DPE Real Time Clock (RTC): 8 Hz, ist der bestimmende Zeittakt für die Software. Das multi–tasking System ist auf diese Frequenz ausgerichtet, da der Watch-dog nach jedem RTC aufgerufen werden muß.

[3]Random Access Memory.
[4]Single Error Correction and Double bit Error Correction.
[5]Read Only Memory.

- Broadcast Pulse 1 (BCP 1): 0.125 Hz, ist synchronisiert mit dem acht sekündlichen Polling Zyklus der Satellitentelemetrie.
- Broadcast Pulse 2 (BCP 2): 1 Hz.

Abbildung 2.3 zeigt die DPE im IBIS–Laboratorium der Astronomie in Tübingen. Das oberste, noch leere Fach in der DPE–Box ist das Einschubfach für den HEPI. Das Fach mit dem Griff ist die Platine mit dem ROM–Speicher, auf der sich die CSSW und die IASW befinden. Da für einen Test der Software die EEPROMs des ROM ausgewechselt werden müssen, ist der Griff an dem Schubfach angebracht.

2.2.3 Die Verwaltung des RAM–Speicher

Der Arbeitsspeicher ist word–orientiert, das heißt ein word = 16 Bit werden durch die gleiche Speicheradresse angesprochen. Der DPE stehen 1 Mword RAM zur Verfügung. Um den gesamten Speicherraum eindeutig zu adressieren, benötigt man eine 20 Bit lange Speicheradresse. Der Prozessor besitzt jedoch lediglich einen 16 Bit Adressraum, er kann also nur 64 kword verwalten. Um dennoch den gesamten physikalischen Speicher erreichen zu können, bedient man sich der Methode der *A*ddress–States. Der 16 Bit Adressraum des Prozessors wird nur noch als ein logischer Speicherraum verwendet. Dieser wird unterteilt in 16 Pages zu je 4 kword. Zusätzlich unterscheidet man noch zwischen Anweisungen (Instructions) und Daten (Data). Zusammen ergibt sich ein logischer Adressraum von 64 kword für Anweisungen und 64 kword für Daten. Einen solchen 128 kword großen logischen Adressraum nennt man *A*ddress–State (AS). Mittels 4 Bit im *Processor Status Word* kann man 16 Address–States definieren. Der logische Adressraum ist dadurch doppelt so groß geworden wie der physikalische.

Über ein Abbildungsverfahren (Mapping) kann man jeder Page eines Address–States einen 4 kword Block im physikalischen Speicher zuordnen. Dabei brauchen zwei benachbarte Pages des logischen Adressraums nicht zwingend im physikalischen Speicher nebeneinander zu liegen. Es ist ebenso möglich, mehrere logische Pages auf dieselbe physikalische Page abzubilden. Die Möglichkeit, die einzelnen Pages eines Address–States auf beliebige Bereiche des physikalischen Speicher abzubilden, kann die gesamte Speicherverwaltung sehr schnell unübersichtlich und aufwendig werden lassen. Deshalb wurde bei der Erstellung der IASW darauf geachtet, daß die Anordnung des Speichers weitgehend[6] fortlaufend ist, also die Reihenfolge der Pages im physikalischen Speicher gleich derjenigen innerhalb eines Address–States ist. Es wurden auch alle Daten- und Anweisungs–Pages gleicher logischer Adresses aufeinandergelegt. Wird während des Programmablaufs ein Programmteil benötigt, der nicht im aktuellen Address–State untergebracht ist, so muß mittels einer Transit–Routine die Abbildung des logischen auf den physikalischen Speicher geändert werden. Man spricht in diesem Fall von einem Address–State–Wechsel. Da ein solcher AS–Wechsel relativ viel Zeit (ca. 60 μs) benötigt, sollten bei der Echtzeit–Datenverarbeitung der IASW solche AS–Wechsel vermieden werden [18, 15].

2.2.4 Die Common Service Software

Die *Common Service Software* (CSSW) ist die Schnittstelle zwischen den Hardware–Interfaces und der *Instrument Application Software* (IASW). Sie erledigt die Kommunikation mit dem OBDH (Telekommandos, Telemetrie, On Board Time), enthält Interfaceroutinen für alle Datenleitungen *High–Bit–Rate* (HBR) und *Low–Bit–Rate* (LBR), steuert das Task Scheduling der

[6] Es gibt einige zwingende Page–Mappings, die von ASTRES oder der CSSW gefordert werden.

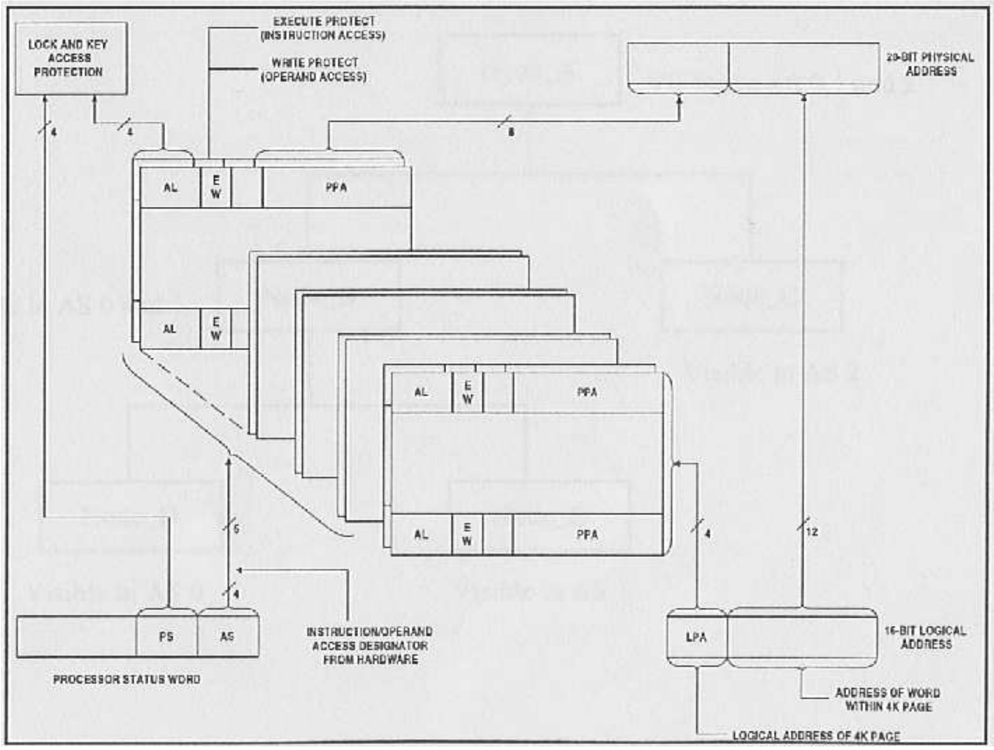

Abbildung 2.4: *Der Speicheraufbau der DPE*

IASW und stellt eine Vielzahl von Unterprogrammen bereit. Die CSSW ist ebenfalls in ADA und teilweise auch in Assembler geschrieben. Details sowie verbindliche Vorgaben sind im CSSW User Requirement Document nachzulesen. Die von der CSSW bereitgestellten Funktionen werden im CSSW Software Interface Document beschrieben. [14]

2.3 Die Programmiersprache ADA

Die Programmiersprache ADA wurde nach der Mathematikerin Ada, Countess of Lovelace, Tochter des englischen Schriftstellers Lord Byron[7] benannt. Sie war Kollegin von Charles Babbage[8], der 1822 den ersten Versuch unternahm, eine programmgesteuerte Rechenanlage zu bauen. Countess Ada wird von Informatik–Historikern als die erste Programmiererin betrachtet. Die Programmiersprache ADA wurde unter Jean Ichbiah bei CII–Honeywell–Bull in Paris entwickelt im Zuge einer weltweiten Initiative des Verteidigungsministeriums[9] der USA im Jahre 1976 zur Entwicklung einer neuen höheren Programmiersprache. Die vorherige Initiative des Department of Defense im Jahre 1962 brachte, nebenbei bemerkt, die Sprache COBOL hervor. Der Hauptgrund für diese Initiative war eine Softwarekrise im DoD. Immer höhere Anteile des Budgets wurden für sogenannte eingebettete Systeme, in denen der Computer ein Teil eines technischen Systems ist, ausgegeben. Dies führte zu einer Vielzahl von Programmiersprachen. Explizite Ziele der Ausschreibung waren insbesondere Wartbarkeit, Zuverlässigkeit, Portabilität,

[7]George Gordon Noel Byron, *1788, †1824; Romantiker; bekanntestes Werk: ≫ Don Juan ≪.
[8]Charles Babbage, *1792, †1871, engl. Mathematiker und Nationalökonom.
[9]Department of Defense, kurz DoD.

Verständlichkeit und Effizienz großer Programmsysteme. ADA verdankt seine Entstehung der Anfang der siebziger Jahre aufkeimenden Idee der Softwaretechnik. Die Konstrukte einer Programmiersprache sollten die Anwendung vieler Methodiken gestatten.

Zu einem zuverlässigen Programm gehört weitgehende Fehlerfreiheit. ADA ist so konzipiert, daß sich jeder Fehler im Programm zum frühestmöglichen Zeitpunkt bemerkbar macht. Fehler, die auf der Zuweisung verschiedener Variablentypen beruhen, werden in ADA durch die *Strenge Typisierung* vermieden: Jedes Datenobjekt muß bei seiner Vereinbarung einem Datentyp zugeordnet werden und Operationen wie Vergleiche können nur auf Werte gleichen Typs angewendet werden. Ein weiteres Konzept zur Fehlervermeidung ist das *Geheimnisprinzip*: An jeder Stelle eines Programms sollte nur das zugänglich sein, was inhaltlich an der Stelle relevant ist.

Der von der ESA vorgegebene Compiler stammt von der amerikanischen Firma TLD und wurde nach dem ANSI–Standard von 1983 validiert. Es gibt zwar schon den objektorientierten ADA '95 Compiler, aber der ADA '83 Compiler erhielt den Vorzug, da seine Fehler aufgrund seines langjährigen Einsatzes auch in der Raumfahrt bereits gut dokumentiert sind und umgangen werden können. Dennoch ist es in enger Zusammenarbeit mit Mike Rennie[10] gelungen, eine fehlerhafte Übersetzung der Hochsprache ADA nach Assemblercode des Prozessors MIL-1750A als undokumentierten Compilerfehler zu identifizieren.

Es gibt verhältnismäßig wenig in ADA, was nicht bereits Bestandteil anderer Programmiersprachen wäre. Der Kern von ADA ist PASCAL sehr ähnlich. Somit liegt die Bedeutung dieser Sprache weniger in ihrer Originalität als darin, viel Wünschenswertes aus verschiedenen Sprachen in einen einheitlichen Rahmen eingepaßt zu haben. Zu ihren großen Stärken gehört sicherlich die maschinennahe Programmierung, denn sie ermöglicht das bitgenaue Ablegen von Variablen im Speicher schon innerhalb der Definition. ADA '83 bietet die Möglichkeit zu einem modularen Programmaufbau. Ein Softwaremodul, bei ADA auch Paket genannt, besteht aus Spezifikation und Body. Definitionen, die für andere Pakete sichtbar und verwendbar sein müssen, werden in der Spezifikation bekanntgegeben, während der Body des Paketes nach dem Geheimnisprinzip unsichtbar bleibt. Wird in einer Spezifikation z.B. ein Unterprogramm definiert, so kann dieses von anderen Paketen aufgerufen werden, ohne nähere Kenntnisse der innerhalb des Unterprogramms verwendeten Variablen oder Algorithmen zu benötigen.

ADA unterstützt auch den Gebrauch von Exception Handlern. Um einen Softwareabsturz aufgrund eines Laufzeitfehlers auszuschließen, kann man jedes Unterprogamm mit einem Exception Handler ausstatten. Eine Exception tritt auf, wenn eine ordnungsgemäße Ausführung des Programms nicht möglich ist, wenn z.B. ein Wertebereich einer Variablen überschritten wurde. In einem Exception Handler kann man diese Fehler auffangen und Befehle eingeben, die ausgeführt werden, wenn diese Exception während der Laufzeit auftritt. Es ist ebenso möglich, sich selbst eine Exception zu definieren und sie durch einen Aufruf auszulösen.

Weitere Eigenschaften der Programmiersprache ADA findetman in der gängigen Fachliteratur. [2, 11, 17]

[10]GMV, Madrid / Spanien; Software Support; April 1998.

Kapitel 3

Datenverarbeitung auf INTEGRAL

3.1 Ziele der On Board Datenverarbeitung

Jedes der vier Instrumente sammelt sowohl wissenschaftliche als auch Housekeeping–Daten, verpackt sie in Telemetriepakete und stellt diese zur Abholung bereit. Die Abholung (Polling) der Telemetriepakete steuert die *Central Data Management Unit* (CDMU). Gemäß einer Tabelle, der Polling Sequence Table, deren Abarbeitung in einem achtsekündlichen Zyklus (Polling Sequence Cycle) erfolgt, werden sequenziell bei den Instrumenten Telemetriepakete angefordert. Die Polling Sequence Table kann via Telekommando verändert und damit den sich möglicherweise ändernden Datenraten der Instrumente angepaßt werden. Während eines Polling Sequence Cycle werden insgesamt 195 Telemetriepakete eingesammelt, von einem einzelnen Instrument können maximal jedoch nur 184 Pakete angefordert werden. Jedes Instrument sollte daher die Verarbeitung von 184 Telemetriepaketen innerhalb acht Sekunden unterstützen können. Von IBIS werden in einem Zyklus nominell zwischen 129 und 136 Telemetriepakete angefordert [4, 6].

Eine kurze Abschätzung zeigt, was die Datenverarbeitung leisten können muß. Ein Telemetriepaket hat eine Größe von 220 word[1] oder 3520 Bits. Nominell werden mindestens 129 Telemetriepakete in acht Sekunden abgeholt, maximal 184. Die Telemetrierate von IBIS liegt also zwischen 56 760 – 80 960 Bit pro Sekunde. Die Datenraten von ISGRI werden auf 350 – 1500 Ereignisse je Sekunde geschätzt, die von PICsIT auf 2000 – 28 000 Ereignisse je Sekunde. Jedes Ereignis wird als 64 Bit großes Datenwort aufgezeichnet. Von ISGRI werden also 22 400 – 96 000 Bit, von PICsIT 128 000 – 1 792 000 Bit zu verarbeiten sein. Das bedeutet aber, daß die Datenmengen im günstigsten Fall um einen Faktor 2, im ungünstigsten Fall jedoch um den Fakor 34 komprimiert werden müssen[2]. Es gibt zwar genügend Algorithmen, die einen Komprimierungsfaktor von über 40 erreichen, doch reicht die Rechenkapazität der DPE für solche Komprimierprogramme bei weitem nicht dazu aus, sie in der Echtzeit–Datenverarbeitung einzusetzen. Außerdem ist der effektive Komprimierungsfaktor in hohem Maße von den Ausgangsdaten abhängig. Sollen keine zusammengehörigen Informationen, z.B. die Datenstruktur eines Ereignisses, auseinandergerissen werden, wäre, um eine optimale Ausnutzung der Telemetrierate zu erreichen, ein hoher Verwaltungsaufwand notwendig.

Es müssen also Möglichkeiten gefunden werden, die Daten in einer Art und Weise zusammenzufassen, ohne daß allzuviel wissenschaftliche Information verloren geht. Eine dieser Möglichkeiten ist das Erstellen von Histogrammen. Hier werden auf Kosten der Zeitinformation

[1] 1 word = 16 Bits.

[2] In dieser Rechnung nicht enthalten sind alle Arten von Housekeeping–Daten, die allerdings, gemessen an den wissenschaftlichen Raten, nur sehr geringen Datenumfang haben.

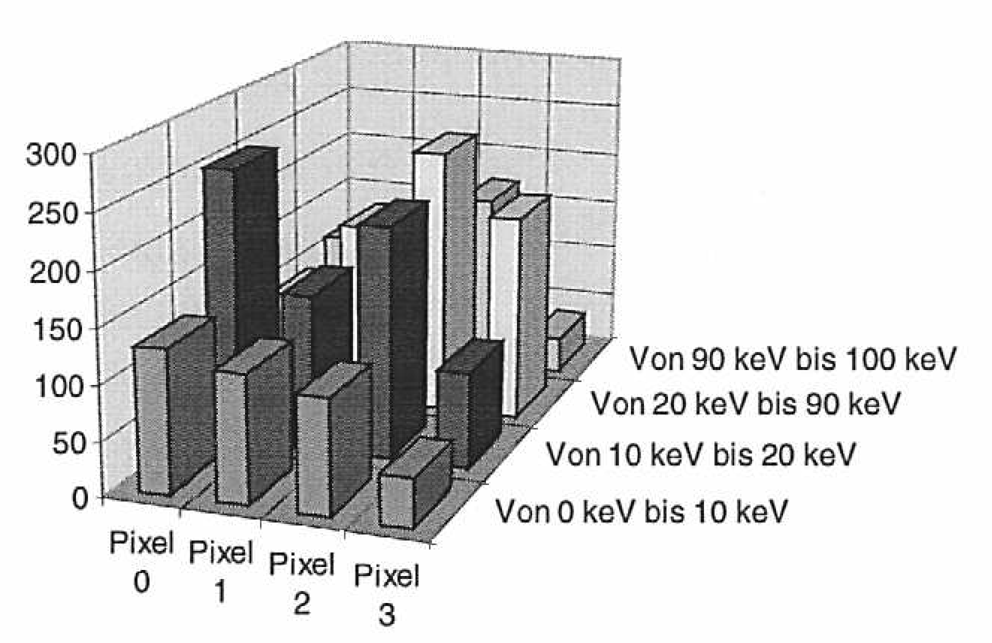

Abbildung 3.1: *Aufbau eines Histogramms*

bei jedem einzelnen Pixel die Ereignisse, nach Energiebereichen unterteilt, gezählt. Die Größe eines Histogramms wird durch die festgelegte Gesamtzahl der Zähler bestimmt. Unabhängig von der Beobachtungszeit und damit der Anzahl der registrierten Ereignisse bleibt die zu verarbeitende Datenmenge, nämlich der zur Erstellung des Histogramms benötigte Speicherplatz, konstant. Die Zeitauflösung verringert sich auf die Integrationszeit des Histogramms, die so gewählt werden sollte, daß ein Überlaufen der Zähler, welches gleichbedeutend mit dem Verlust von Ereignissen ist, unwahrscheinlich bleibt. Diese Methode der Datenreduktion wird für die hohen Ereignisraten des PICsIT–Detektors eingesetzt. Da die DPE weder über genügend Speicher verfügt noch die benötigte Rechenkapazität besitzt, wird das Erstellen der Histogramme von der HEPI–Hardware übernommen. Abbildung 3.1 verdeutlicht noch einmal den Aufbau eines Histogramms.

3.2 Betriebsmodi des IBIS

Das IBIS–Instrument kennt sechs Betriebsmodi, wovon vier durch den *Hardware Event Preprocessor for IBIS* (HEPI) bzw. die *Instrument Application Software* (IASW) realisiert und in Unterkapiteln näher erläutert werden:

- Power Off Mode
- CSSW Configuration Mode
- Stand by Mode
- Patch & Dump Mode
- Diagnostic Mode
- Scientific Mode.

3.2.1 Power Off Mode und Power Save Mode

Der Power Off Mode beschreibt das Instrument im ausgeschalteten Zustand. Die Spannungsversorgungen aller drei an IBIS beteiligten Detektoren und der DPE sind nicht aktiviert. Mittels eines Telekommandos an die CDMU wird die DPE (und damit auch der in der DPE integrierte HEPI) eingeschaltet. Hierbei wird automatisch die CSSW initialisiert und der CSSW Configuration Mode ist erreicht. Jetzt werden auch schon die Telemetriepakete der Essential Housekeeping (H1-Pakete) generiert, welche in diesem Mode allerdings nur die von der CSSW gesammelten Daten enthalten. Die Detektoren sind nach wie vor ausgeschaltet. Die IASW wird zwar in den Arbeitsspeicher (RAM) geladen, ist aber noch nicht gestartet worden. Erst durch ein Telekommando wird die IASW vollständig initialisiert und gestartet. Dieses Start–IASW–Kommando ist im Power Save Mode das einzig gültige Telekommando an die DPE und führt zum Wechsel in den Stand by Mode. Mit einem Telekommando an die CDMU gelangt man vom Stand by Mode wieder in den Power Off Mode, die DPE wird abgeschaltet [19]. Der HEPI befindet sich nach Einschalten der DPE im Transparent Mode. Von ihm stehen noch keine Housekeeping Daten in den H1–Paketen, da diese von der IASW angefordert und an die CSSW weitergegeben werden.

3.2.2 Der Patch & Dump Mode

Das ursprüngliche Design der IASW sah einen eigenen Mode vor, in dem das Auslesen des Speichers (Dump) und das Austauschen von Softwaremodulen (Patch) ermöglicht wird. Da die CDMU bereits in der Lage ist, diese Aufgaben zu erledigen, wird über den Sinn dieses Modes diskutiert. Zum Zeitpunkt der Anfertigung dieser Arbeit war noch keine abschließende Entscheidung über den Verbleib dieses Patch & Dump Mode gefallen. In der IASW ist dieser Mode bislang nicht implementiert.

3.2.3 Der Stand by Mode

Wenn die IASW gestartet wird, befindet sie sich automatisch im Stand by Mode, der HEPI nach wie vor im Transparent Mode. Außer den Daten der CSSW befinden sich nun auch die Statuswerte der IASW und des HEPI in den Essential Housekeeping (H1) Paketen. Via Telekommando können die Subsysteme der Peripherie einzeln an- und ausgeschaltet werden. Eine Tabelle innerhalb der IASW, das *Electrical Configuration Register* (ECR), zeigt der Software an, welche Detektoren ein- bzw. ausgeschaltet sind. Anhand dieser Tabelle fordert die IASW von den Detektoren die Essential Housekeeping Daten an und übergibt sie der CSSW. Sind alle Detektoren eingeschaltet, so sind alle Felder in den H1–Paketen ausgefüllt. Gemäß des ECR werden nun auch die Additional Housekeeping H3 und H4 von den Detektoren angefordert und in Telemetriepakete verpackt. Sollten die Detektoren ausgeschaltet sein, so bleiben die entsprechenden Felder innerhalb der H3- bzw. H4–Pakete leer, werden aber dennoch erstellt und gesendet. Das Einsammeln der Housekeeping Daten findet ebenso im Diagnostic Mode und im Scientific Mode statt. Außer den zyklischen Housekeeping Paketen generiert die IASW bei bestimmten Operationen wie

- Synchronisation der Detektoren,
- Beginn einer Pointing/Slew Sequenz,
- Start/Beendigung einer Histogrammintegration,
- Mode–Wechsel in/aus dem Diagnostic Mode oder Scientific Mode,

ein zusätzliches Housekeeping Paket (H2). In diesem befinden sich alle wichtigen softwareinternen Parameter wie z.B. die aktuellen Sequence Counter aller Telemetriepakete oder Eventcounter. Neben den regelmäßigen Statusinformationen der Detektoren erhält man auf diese Weise auch den Status der IASW zu bestimmten Zeitpunkten.

Im Stand by Mode können auch alle Tabellen, IASW–interne sowie periphere, ausgelesen oder geändert werden. Außerdem kann Patch & Dump von Software der peripheren Systeme durchgeführt werden. Wissenschaftliche Datenverarbeitung findet im Stand by Mode nicht statt. Jedoch müssen im Stand by Mode der Scientific bzw. Diagnostic Mode durch Setzen des HEPI Status Registers mittels eines Telekommandos vorbereitet werden (siehe Kapitel 2.2.1). Vom Stand by Mode kann in jeden anderen Mode (außer Power Off Mode) gewechselt werden. Jeder Mode–Wechsel innerhalb der IASW führt zwingend über den Stand by Mode. Dies gilt auch für Submode–Wechsel innerhalb des Scientific Modes (siehe dazu 3.2.5).

3.2.4 Der Diagnostic Mode

Ziel des Diagnostic Mode ist die Überprüfung der Detektoren sowie des HEPI. Die Daten, die in den Detektoren generiert werden, sollen nicht verarbeitet, sondern unverändert zur Bodenstation gesendet werden, um mögliche Fehlfunktionen in der Detektorelektronik feststellen zu können. Der HEPI muß dafür via Telekommando über sein internes Status Register in seinen Transparent Mode gesetzt werden. Die Daten werden nun durch den HEPI direkt an die DPE weitergeleitet, es findet keine Integration von Histogrammen statt und die Amplitudenkorrektur der CsI–Ereignisse wird nicht mehr durchgeführt. Im Diagnostic Mode bleibt die Zeitauflösung der Detektoren von 238 ns in einem Zeitfenster von zwei Sekunden (23 Bit) erhalten. Es wird nicht nach Koinzidenzen, also comptongestreuten Photonen, gesucht. Lediglich das Datenformat der Detektorereignisse wird umgewandelt. Die Reihenfolge der Informationen wird geändert und die Länge des Datenwortes wird von 64 Bit auf 80 Bit erweitert. Die Detektorereignisse werden als Raw–Events in das Ausgangs–FIFO geschrieben. Alle Kalibrationsereignisse bleiben als solche gekennzeichnet. Bei PICsIT–Ereignissen wird nach wie vor zwischen Einzel- und Mehrfach–Events unterschieden.

In der IASW stehen drei unabhängige ein- und ausschaltbare *Diagnostic Functions* zur Verfügung. Die beiden Diagnostic Functions D1 und D2 sortieren aus dem Datenstrom die Ereignisse des jeweiligen Detektors heraus und reduzieren das Datenformat auf 64 Bit. Die Funktion D1 verarbeitet alle Raw–Events von ISGRI, die Funktion D2 ausschließlich diejenigen von PICsIT. Die Transparenz–Funktion (D3) kopiert die vom HEPI an die DPE übergebenen Daten direkt in Telemetriepakete, ohne irgendwelche Verarbeitungsschritte durchzuführen. Die Länge eines Datenpakets bleibt 80 Bit. Wird diese Funktion (D3) zusätzlich zu einer der beiden anderen D1 oder D2 gewählt, so werden die entsprechenden Ereignisse zweimal zur Erde geschickt, einmal in Transparent Function–Telemetriepaketen und nochmals in den Diagnostic Function–Telemetriepaketen.

Da der Transparent Mode des HEPI und der Diagnostic Mode in der IASW unabhängig voneinander gesetzt werden können, sind auch Kombinationen möglich. Befindet sich die Software im Diagnostic Mode, der HEPI jedoch im Observation Mode (Standard Submode, Polarimetry Submode, Photon-by-Photon Submode), so kann die Ausgabe des HBR-A Interfaces des HEPI mit der Transparenz Funktion der IASW überprüft werden. Diese Möglichkeit wird hauptsächlich in der Entwicklungsphase des HEPI verwendet. Andererseits gestattet der Transparent Mode des HEPI in Kombination mit einem Scientific Mode der IASW die genauere

Zeitauflösung der orginalen Detektordaten (238.4 ns an Stelle von 61.035 μs) zu verwenden. Es muß hierbei jedoch bedacht werden, daß der HEPI die gesamte Datenrate der Detektoren direkt an die DPE weiterleitet, deren Rechenleistung, wie schon in Kapitel 2.2.2 erwähnt, sehr gering ist.

Der einzige gestattete Mode–Wechsel vom Diagnostic Mode ist zurück in den Stand by Mode. Es sind weiterhin alle anderen Telekommandos in diesem Mode zugelassen. So müssen die Detektoren natürlich erreichbar bleiben, damit einzelne Module zur Überprüfung an- und ausgeschaltet werden können, ebenso wie die Diagnostic Functions der IASW .

3.2.5 Der Scientific Mode

Der Scientific Mode ist der eigentliche Beobachtungsmode. Er kann nur aus dem Stand by Mode aktiviert werden. Die einzigen gültigen Telekommandos während des Scientific Modes sind dessen Beendigung durch den Mode–Wechsel zurück in den Stand by Mode sowie die Umstellung zwischen den Pointing- und Slew–Phasen. Die IASW kennt lediglich einen wissenschaftlichen Mode. Die Realisierung der drei wissenschaftlichen Submodi geschieht innerhalb des HEPI. Deshalb muß noch vor dem Wechsel in den Scientific Mode im Stand by Mode der entsprechende Submode im HEPI Status Register eingestellt werden. Dieses Register wird am Beginn des Scientific Mode in den HEPI geladen.

Die wissenschaftlichen Submodi unterscheiden sich lediglich durch die Behandlung der CsI–Ereignisse von PICsIT. Die CdTe–Ereignisse von ISGRI werden in jedem der Submodi als einzelne Events vom HEPI an die DPE übergeben. Die Zeitauflösung der Detektoren von 238.4 ns innerhalb von zwei Sekunden wird auf 61.035 μs innerhalb von 72.8 Stunden geändert und das 64 Bit lange Eventformat der Detektoren auf 80 Bit erweitert. Der HEPI sucht in den Datenströmen der beiden Detektoren nach gleichzeitigen Ereignissen und faßt diese unabhängig vom Submode zu einem 160 Bit großen Compton–Event zusammen.

Ebenfalls vom Submode unabhängig ist die Generierung des Histogramms für die Kalibrationsereignisse des CsI–Detektors. Die Ereignisse eines Pixels werden in 64 Energiekanäle eingeteilt und gezählt. Jeder der 64 Energiekanäle eines Pixels besitzt einen acht Bit tiefen Zähler. Natürlich gilt dies für jedes beliebige PICsIT Pixel. Ein Kalibrationshistogramm besteht also aus 4096 (Pixel) x 64 (Energiekanäle) Zählern von 8 Bit Größe. Die typische Integrationszeit eines Kalibrationshistogramms liegt bei 10 000 Sekunden.

Bei allen Arten von Histogrammen sind sowohl die Integrationszeit als auch die Einteilung der Energiekanäle im HEPI einstellbar.

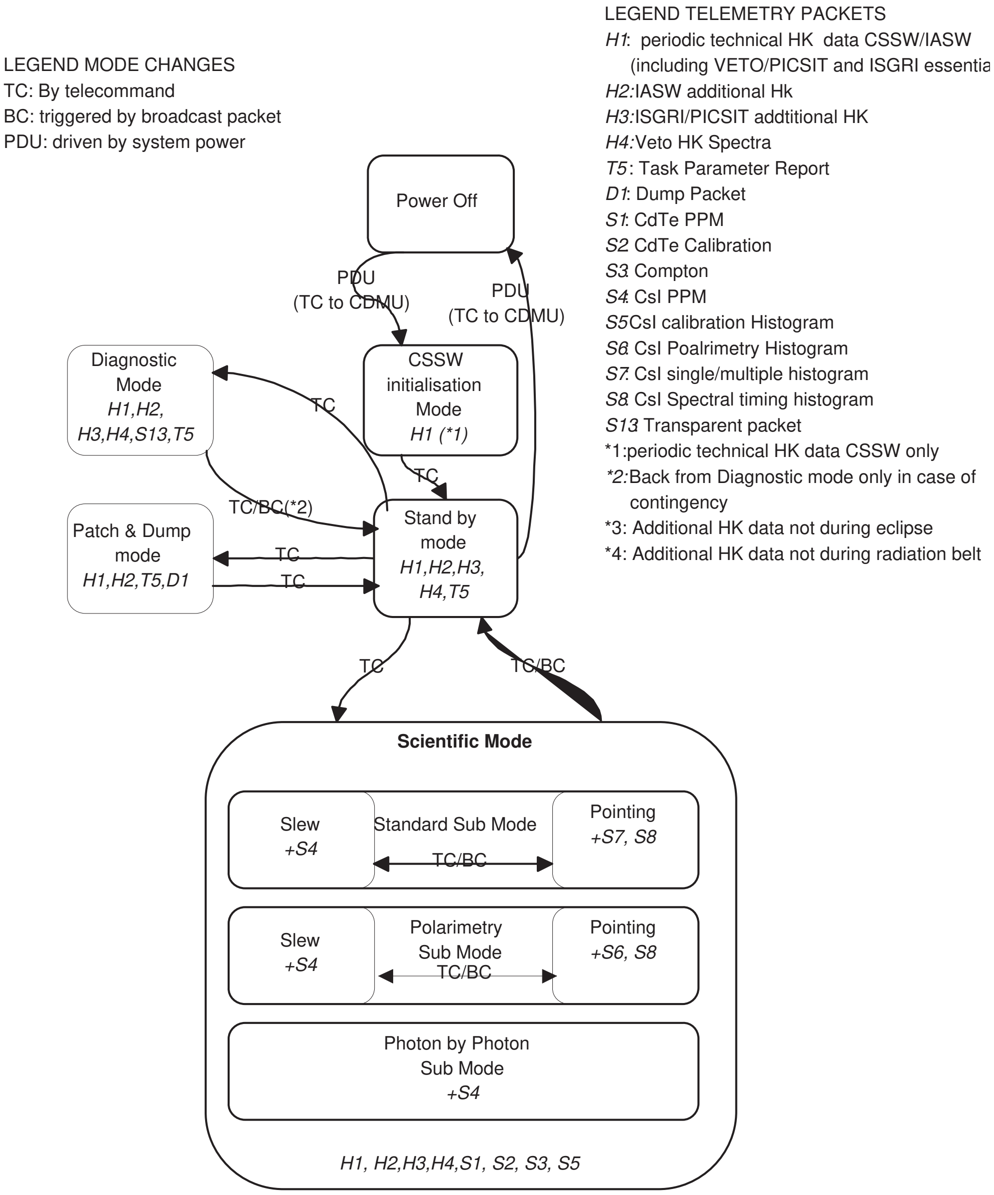

Abbildung 3.2: *Die Betriebsmode von IBIS*

Der Photon by Photon Submode

Im Photon by Photon Submode werden lediglich die Kalibrationshistogramme, jedoch keine wissenschaftlichen Histogramme erstellt. Alle Ereignisse von PICsIT, außer den Kalibrationsevents, werden vom HEPI einzeln an die IASW weitergegeben. Die Zeitauflösung und das Format werden ebenso wie bei den ISGRI–Ereignissen geändert. Es wird nach Einzel- und Mehrfachevents unterschieden.

Der Photon by Photon Submode wird auch beim Slewing des Satelliten, also der Neuausrichtung auf den nächsten Ditheringspunkt (siehe Kapitel 1.6) eingesetzt, da ein Histogramm, das während der Bewegung des Satelliten entstünde, nicht mehr denselben Himmelsausschnitt als Grundlage hätte und die Rekonstruktion der Abbildung durch die kodierte Aperturmaske aufgrund der fehlenden Zeitinformation der Einzelereignisse unmöglich würde. Es könnten keine

Rückschlüsse auf die Beobachtungsrichtungen mehr gewonnen werden.

Der Standard Submode

Es werden zwei Arten von Histogrammen erstellt. Zum einen werden Histogramme mit einer Zeitauflösung zwischen 1 ms und 500 ms erstellt. Sie sollen das spektrale Verhalten einer Quelle zeitaufgelöst darstellen. Mit lediglich acht Energiekanälen ist die Energieauflösung sehr gering. Die gesamte Ortsinformationen werden verworfen. Diese Spectral Timing–Histogramme werden wie die Comptonereignisse der DPE als 160 Bit große Events übergeben.

Die andere Histogrammart, das Spectral Imaging–Histogramm, ist wie das Kalibrationshistogramm aufgebaut. Die Ereignisse von PICsIT werden für jedes Pixel in 256 Energiekanäle unterteilt und gezählt. Diese Histogrammart besteht also aus 4096 x 256 Zählern von wieder 8 Bit. Für Einzelevents und Mehrfachevents wird je ein Histogramm erstellt. Die Integrationszeit entspricht einer Pointingperiode von ungefähr 20 Minuten.

Der Polarimetry Submode

Zusätzlich zu den auch in diesem Submode generierten Spectral Timing–Histogrammen wird eine weitere Histogrammart gebildet, welche die Polarisation der detektierten Strahlung wiederspiegeln soll. Es wird davon ausgegangen, daß Ereignisse, die innerhalb des CsI–Detektors einer Comptonstreuung unterliegen, eine Vorzugsrichtung besitzen, wenn die zugrundeliegende Strahlung eine Polarisation aufweist. Wenn bei Zweifach–Ereignissen zwei direkt benachbarte Pixel beteiligt sind, wird darüber der Winkel bestimmt.

010	**001**	**000**
011	Ursprungs-Pixel	**100**
111	**110**	**101**

Abbildung 3.3: *Definition des Polarisationswinkel*

Wieder gibt es für jedes der 4096 Pixel 64 Energiekanäle. Bei dem Polarimetry–Histogramm ist jeder Energiekanal nochmals in die acht möglichen Streuwinkel unterteilt. Jeder der 4096 x 64 x 8 Zähler hat wieder eine Tiefe von 8 Bit. Das Histogramm zur Bestimmung der Polarität ist damit das größte Histogramm.

3.3 Weg der Daten

Der Entstehungsort des Datenstromes sind die Detektoren. Ein Ereignis wird in den Detektoren mit einer Analog Front End Electronics als elektrische Impulse aufgefangen und durch ASICs zur Weiterverarbeitung bereitgestellt.

Die erste Datenverarbeitung findet bei ISGRI durch die *CdTe Module Control Electronics* (MCE) statt. Diese befindet sich zwischen den *Modular Detection Units* (MDU) und dem *ISGRI FIFO Data Manager* (IFDM). Die MCE übergibt dem IFDM das Ereignis als ein 64 Bit langes serielles Datenwort, welches aus Informationen über Energie, Risetime, Position und einem Zeitfeld besteht. Die Eventzeit wird erst vom IFDM eingesetzt, da er die Daten von den 8 Detektormodulen nach der Zeit sortiert für die Weiterverarbeitung durch die HEPI zwischenspeichert. Treten in einem Detektormodul zwei oder mehr Ereignisse zeitgleich auf, so werden diese von der MCE nicht weitergegeben. Wird ein Ereignis mittels Koinzidenzmessung als ein von der Kalibrationsquelle verursachtes Ereignis erkannt, markiert die MCE dieses Ereignis als Kalibrationsevent durch Setzen eines Flagbits. Auch der IFDM sortiert Ereignisse aus dem Datenstrom heraus. Ereignisse, die zeitgleich in verschiedenen Detektormodulen auftreten, werden nicht zwischengespeichert. Gleiches geschieht mit Ereignissen, die sich in Zeitkoinzidenz mit Ereignissen des aktiven VETO–Schilds befinden, da hierzu keine Gamma–Photonen aus der eingestellten Beobachtungsrichtung gehören.

Bei PICsIT findet die erste Datenverarbeitung durch die *CsI Digital Front End Electronics* (DFEE) statt. Jede der 16 DFEE besitzt ein eigenes lokales FIFO[3] und verarbeitet Daten von je 16 ASICs, von denen jeder wiederum 16 Detektorelementen vorgeschaltet ist. Wird in einem oder mehreren Detektorpixeln ein Ereignis registriert, gibt der entsprechende ASIC ein Data–Ready–Signal zur DFEE. Diese identifiziert das (die) aktive(n) Pixel, liest die Energie aus und markiert das (die) Ereignis(se) mit einer Zeitinformation. Werden mehr als drei aktive Pixel gefunden, werden die Ereignisse verworfen. Die DFEE kontrolliert die Ereignisse auch auf Zeitkoinzidenz mit der Kalibrationsquelle und markiert entsprechende Ereignisse durch Setzen eines Flagbits. Ebenso wird die Zeitkoinzidenz mit VETO–Ereignissen geprüft und entsprechende Ereignisse gelöscht. Gültige Ereignisse werden als 48 Bit langes Datenwort in das lokale FIFO geschrieben. Die 16 Semimodul–FIFOs werden vom *PICsIT FIFO Data Manager* (PFDM) überwacht. Der PFDM besitzt ein Address FIFO und einen Multiplexer Buffer. Wird in ein Semimodul–FIFO ein Ereignis geschrieben, so speichert der PFDM die entsprechende Adresse des Semimoduls in seinem Address FIFO, der dadurch die Ereignisse aller Semimodule in der richtigen Zeitreihenfolge als Adressreferenz enthält. Der PFDM kontrolliert die Zeitgleichheit von Ereignissen von verschiedenen Semimodulen und ändert gegebenfalls das Multiplicity Flag. Wird dieses größer als drei, werden die zugehörigen Ereignisse verworfen. Bei der Datenübergabe an der HEPI ändert der PFDM das Datenformat vom 48 Bit Datenword auf 64 Bit.

Befindet sich mindestens ein Ereignis in einem Ausgangs–FIFO eines Detektors, so zeigt dieser der HEPI durch High–Setzen der FIFO_not_empty–Line an, daß Daten vorhanden sind. Ist der HEPI zur Aufnahme von neuen Daten bereit, generiert er daraufhin einen Sample Pulse, der den Detektor dazu veranlaßt, ein Datenword von 64 Bit über die High Serial Line zu schicken. Sind mehr als ein Datenword in dem Ausgangs–FIFO, so bleibt die FIFO_not_empty–Line auf High und HEPI generiert nach dem Empfang des ersten Datenwortes einen weiteren Sample

[3] First In First Out; kleiner Zwischenspeicher, bei dem die Reihenfolge der Ausgabe äquivalent ist zur Reihenfolge der Eingabe.

Pulse, worauf der Detektor das zweite Datenword sendet. Wurden alle Ereignisse aus dem Detektor–FIFO an die HEPI übermittelt, wird die FIFO_not_empty–Line auf Low gesetzt.

Der HEPI fordert die Daten also von den Detektoren ISGRI und PICsIT an. Werden von den Detektoren mehr Ereignisse registriert als HEPI auslesen kann, so gehen diese Daten auf der Detektorseite verloren, falls die Ausgangs–FIFOs der Detektoren überlaufen. Die Tabelle 3.1 zeigt, in welchem Format die HEPI die 64 Bit Datenworte von den beiden Detektoren empfängt. Im Format sind auch die verschiedenen Flagbits enthalten. Dabei bedeutet:

- cal.flag: 1 = Kalibrationsevent, 0 = kein Kalibrationsereignis;
- MMC: Multiple Module coincidence mit 1 = wahr, 0 = kein MMC;
- Pulse ID: Ordnung des Ereignisses in einem Multiple Event mit 00 = erstes, 01 = zweites, 10 = drittes Event;
- MUX: Multiplicity mit 00 = MUX 1, 01 = MUX 2, 10 = MUX 3.

MSB															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Byte der Zeitinformation (MSB)								2. Byte der Zeitinformation							

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
3. Byte der Zeitinformation (LSB)								Risetime (nur CdTe)							

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
cal.flag					CdTe Amplitude										
cal.flag	MMC	Pulse ID		MUX		CsI Amplitude									

															LSB
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
		CdTe Position													
				CsI Position											

Tabelle 3.1: *Datenformate von den Detektoren zum HEPI [23].*

Der HEPI verarbeitet die eingelesenen Ereignisse gemäß der Einstellung des Status Registers. Vorgesehen sind lediglich vier bereits festgelegte Modi: Der Diagnostic Mode sowie die wissenschaftlichen Modi Standard Pointing Mode, Photon by Photon Mode und Polarimetry Mode. Die genaue Bedeutung dieser Modi wurde bereits in Kapitel 3.2 beschrieben.

Die HEPI ist mit zwei *High–Bit–Rate* Datenleitungen (HBR) mit der DPE verbunden. Jedes HBR besitzt ein acht kword großes Ausgangs–FIFO. Über das HBR-A werden alle Events der wissenschaftlichen Funktionen S1, S2, S3, S4, die S8 Histogramme sowie die Raw–Events des Diagnostic Modes gesendet. Erreicht das FIFO einen einstellbaren Füllstand (Standard ist 1000 word), dann zeigt die HEPI der DPE durch High–Setzen der HBR-A Request–Leitung an, daß Daten verfügbar sind. Die DPE startet die Übertragung durch Low–Setzen der tx–enable–Line, liest 1000 word aus dem FIFO aus und speichert die Daten in einem Buffer innerhalb der IASW. Wie die HEPI bei den Detektoren fordert auch die DPE die Daten bei der HEPI an. Ist die Datenverarbeitung innerhalb der DPE zu langsam, so daß sich das Ausgangs–FIFO der HEPI vollständig füllt, stoppt die HEPI das Anfordern weiterer Daten von den Detektoren. Innerhalb der HEPI gehen keine Daten unbemerkt verloren.

Datentyp	Typfeld	Risetime	Eventzeit	Amplitude	Y-Pos.	Z-Pos.	Amplitude	Y-Pos.	Z-Pos.	Spare	Bit
CdTe single	0-7	8-15	16-47	53-63	65-71	73-79					80
CdTe raw	0-7	8-15	25-47	53-63	65-71	73-79					80
CdTe cal.	0-7	8-15	16-47	53-63	65-71	73-79					80
CsI single	0-7		16-47	54-63	66-71	74-79					80
CsI raw	0-7		25-47	54-63	66-71	74-79					80
CsI multi	0-7		16-47	54-63	66-71	74-79					80
CdTe + CsI single	0-7	8-15	16-47	53-63 CdTe	65-71 CdTe	73-79 CdTe	86-95 CsI	98-103 CsI	106-111 CsI	112-159	160
CdTe + CsI multi	0-7	8-15	16-47	53-63 CdTe	65-71 CdTe	73-79 CdTe	86-95 CsI	98-103 CsI	106-111 CsI	112-159	160
Spect. timing	0-7		16-47	48 - 111 (8 Histogrammzellen zu je 8 Bit)						112-159	160

Tabelle 3.2: *Datentypen und Formate des HBR-A Interface [23].*

Das HBR-B dient zum Auslesen der in der HEPI generierten Histogramme der wissenschaftlichen Funktionen S5, S6 und S7. Übertragen werden die Histogramme in Blöcken von 1 kbyte plus einem vorangestellten Word (16 Bit), welches die Speicheradresse des ersten Datenwords des Blockes enthält. Auch hier steuert die DPE die Integrationszeit und die Auslesegeschwindigkeit der Histogramme und speichert die Daten in einem Buffer innerhalb der IASW zur Weiterverarbeitung.

In der IASW werden die Daten nach Ereignistypen sortiert, komprimiert und in Telemetriepakete verpackt. Eine genauere Beschreibung der Abläufe innerhalb der IASW sowie der genauen Datenformate werden in Kapitel 5 behandelt. Die Telemetriepakete werden durch einen Funktionsaufruf der CSSW übergeben. Diese kann bis zu 50 Telemetriepakete in ihrem Telemetrie–Subbuffer aufnehmen.

Parallel zur Verarbeitung der wissenschaftlichen Daten werden von allen Instrumenten und der HEPI alle acht Sekunden Essential Housekeeping Daten (H1) angefordert. Hierzu ist die DPE mit jedem der Instrumente ISGRI, PICsIT, VETO sowie mit der HEPI über je eine *Low–Bit–Rate* Leitung (LBR) verbunden. Die IASW fordert mit einem Kommado via LBR die Housekeeping

Daten an, die daraufhin ebenfalls via LBR geschickt werden. Diese Daten werden durch Funktionsaufruf an die CSSW weitergeleitet, die ihrerseits eigene Housekeeping Daten sowie aufgetretene On Event Messages hinzufügt. Die Generierung der H1–Telemetriepakete obliegt der CSSW, die IASW sammelt und übergibt lediglich die Daten. Zusätzlich werden Additional Housekeeping Daten von ISGRI, PICsIT (H3) und VETO (H4) gesammelt, in Telemetriepakete verpackt und wie die wissenschaftlichen Daten an die CSSW weitergeleitet. Die Daten von ISGRI und PICsIT werden alle 32 Sekunden, diejenigen von VETO etwa alle 30 Minuten angefordert. Die entsprechenden Formate findet man im User Manual des IBIS–Instruments [19].

Den Telemetriepaketen eines Instruments können unterschiedliche Prioritäten zugeordnet werden. Bei IBIS sind allen Housekeeping Paketen eine hohe, allen wissenschaftlichen oder diagnostischen Daten eine niedrige Priorität zugeordnet. Die CSSW kopiert das Telemetriepaket der höchsten Priorität aus dem Telemetrie–Subbuffer in den *Instrument Communication Buffer* (ICB). Sind mehrere Pakete der gleichen Priorität vorhanden, nimmt die CSSW das zeitälteste. Erst wenn das Paket aus dem ICB weitergeleitet wurde, wird die Kopie im Subbuffer zum Überschreiben freigegeben. Solange sich im Telemetrie–Subbuffer noch Housekeeping Pakete befinden, werden diese vor den wissenschaftlichen Paketen zur Weiterleitung bereitgestellt. Die Essential Housekeeping (H1) genießen eine Sonderstellung. Sie werden aufgrund ihrer Wichtigkeit von der CSSW direkt in den ICB geschrieben. Dabei kann ein anderes, noch nicht weitergeleitetes Telemetriepaket überschrieben werden. Da dieses aber lediglich eine Kopie aus dem Subbuffer war, kann es von dort wieder in den ICB kopiert werden, sobald das H1–Paket abgeholt wurde. Es gehen also keine Telemetriepakete verloren.

Die gesamte Steuerung der Telemetrie unterliegt der Kontrolle der *Central Data Management Unit* (CDMU). Gemäß eines *Polling Sequence Table* (PST) fordert die CDMU von den DPE der verschiedenen Instrumente Telemetriepakete an. Stehen diese bereit, so werden sie über den *On Board Data Handling* Bus zur Sendestation im Service Modul des Satelliten geleitet und zur Erde gefunkt.

Je nach Position des Satelliten werden die Beobachtungsdaten entweder in Goldstone (USA) oder in Redu (Belgien) empfangen und nach Darmstadt zum *Mission Operation Center* (MOC) weitergeleitet. Hier wird durch Auswerten der Housekeepingdaten die Funktionsweise des Satelliten überwacht. Da das MOC für die Kommunikation mit dem Satelliten verantwortlich ist, kann bei eventuellen Fehlfunktionen oder dem Erreichen kritischer Parameter mittels Senden entsprechender Telekommandos an den Satelliten schnell eingegriffen werden. Die gesamten Daten gelangen schließlich zum *INTEGRAL Science Data Cente*r (ISDC) nach Versoix in der Schweiz, wo die Rohdaten archiviert werden. Unter Berücksichtigung der Charakteristika der verschiedenen Instrumente werden hier auch die Rohdaten in physikalische Einheiten umgewandelt und Standardanalysen erstellt. Die Daten sind zunächst der jeweiligen Beobachtungsgruppe vorbehalten, die ein Jahr das alleinige Zugriffsrecht auf ihre Daten besitzt. Nach Ablauf dieses Jahres sind diese Daten im Archiv des ISDC weltweit für alle Interessenten abrufbar. Durch diese Handhabung soll die baldige Auswertung der Daten gefördert werden. [4]

Kapitel 4

Die Instrument Application Software

Wie bereits in Kapitel 2.2.2 erwähnt, ist das Betriebssystem ASTRES der DPE ein multi–tasking–System. Multi–tasking Systeme bieten die Möglichkeit, ein großes Programm gemäß seiner verschiedenen Aufgaben in Programmsequenzen, sogenannte Tasks, aufzuteilen, denen jeweils unterschiedliche Prioritäten zugeteilt werden. Das wesentliche Charakteristikum eines Tasks ist, daß er aus einem Programm besteht, das in einer Endlosschleife untergebracht ist. Hat das Programm seine Aufgabe erfüllt, geht der Task in einen Wartezustand über und ermöglicht anderen Tasks die Aktivität. Während des Programmablaufs werden die Tasks in der Reihenfolge ihrer Prioritäten ausgeführt. Es startet also der Task mit der höchsten Priorität. Ist diese Programmsequenz beendet, startet der Task mit der zweithöchsten Priorität und so weiter. Dabei kann ein Task niedriger Priorität von einem Task höherer Priorität unterbrochen werden. Der Task niedrigerer Priorität setzt beim nächsten Aufruf seinen Ablauf an der Stelle fort, an der er unterbrochen wurde. Das Starten eines Tasks kann ebenso, gekoppelt an einen Taktgeber, zyklisch erfolgen oder durch ein auslösendes Ereignis, wodurch ein Task aufgerufen wird. Gewöhnlich unterwirft man den gesamten multi–tasking Betrieb einem eigenen Zeitzyklus. Hierbei wird nach einem bestimmten Zeitabstand wieder der Task mit der höchsten Priorität gestartet und wiederholt damit regelmäßig die Abarbeitungssequenz der Tasks.

Sowohl die CSSW als auch die IASW sind aus Tasks aufgebaut. Der bestimmende Zeitzyklus des multi–tasking Ablaufs für CSSW und IASW ist der Takt der *Real Time Clock* (RTC) mit ihrer 8 Hz Frequenz. Alle 125 ms wird also der Task mit der höchsten Priorität gestartet, welches ein Task der CSSW ist. Mit Ausnahme des *Background Tasks* wird hier nicht auf die CSSW–Tasks eingegangen. Dieser hat unter anderem die Aufgabe, die auf dem DPE–Board untergebrachte Watch dog Reset Hardware zu deaktivieren. Der Watch dog ist vereinfacht gesagt ein Zähler, der innerhalb jedes RTC–Zyklusses zurückgesetzt werden muß. Geschieht dies nicht, so wird die gesamte Software der DPE, CSSW und IASW, angehalten und muß via Telekommando neu initialisiert werden. Das Verfahren des hardwaregesteuerten Resets dient zum schnellen Erkennen eines Softwareabsturzes. Ist ein Watch dog Reset aufgetreten, wird dies innerhalb der CDMU–Housekeeping angezeigt.

Die Priorität dieses Background Tasks ist geringer als diejenigen aller IASW–Tasks und wird folglich erst dann abgearbeitet, wenn alle IASW–Tasks beendet sind. Die IASW muß also darauf achten, daß sie in jedem 125 ms RTC–Zyklus dem Background Task der CSSW genügend Zeit zur Verfügung stellt.

4.1 Architektur der Instrument Application Software

Die IASW ist aus acht Tasks aufgebaut. Tabelle 4.1 zeigt, welche Prioritäten den einzelnen Tasks zugeordnet sind.

Task ID	Priorität	Name
1	2	Main Task
2	15	Main Control Task
3	12	TC–Handler
4	3	HBR-A Task
5	13	Line Manager Task
6	14	Periodic Housekeeping (HK) Task
7	9	Histogram Integration Task
8	8	Histogram Reading Task

Tabelle 4.1: *Die IASW ADA Tasks*

4.1.1 Der Main Task und Main Control Task

Das eigentliche Hauptprogramm der IASW ist der *Main Task*. In ihm sind die einzelnen Modi realisiert. Er verarbeitet alle Events, die vom HEPI über das HBR-A Interface an die IASW weitergegeben werden und startet die beiden Histogramm–Tasks. Er führt die Operationen durch, die auch länger als einen RTC–Takt dauern können. Da aber in jedem RTC–Zyklus der Background Task zum Zuge kommen muß, darf der Main Task maximal nur eine bestimmte Zeit aktiv sein. Damit der Main Task sich nicht um das Timing zu kümmern muß, wurde der *Main Control Task* eingeführt. Er ist der IASW–Task mit der höchsten Priorität. Seine Aufgabe besteht darin, die IASW rechtzeitig anzuhalten, um das Zurücksetzen des Watch dog Resets durch den Background Task zu garantieren. Der Main Control Task wird periodisch nach jedem RTC–Signal aufgerufen. Er startet einen Timer, welcher den Anteil des 125 ms RTC–Takts, der IASW zur Verfügung steht, abzählt. Dies sind ungefähr drei Viertel eines RTCs. Nach dem Aktivieren des Timers ermöglicht der Main Control Task das Starten des Main Tasks. Dieser hat die tiefste Priorität aller IASW–Tasks. Ist kein anderer Task aktiv, startet der Main Task und läuft, bis der Timer das eingestellte Zeitbudget erreicht hat. Dadurch übernimmt der Main Control Task wieder die Aktivität und unterbindet den Main Task. Da alle weiteren IASW–Tasks eine höhere Priorität haben und somit, falls sie aktiv wurden, vor dem Main Task liefen, verbleibt die restliche Zeit des RTC dem Background Task.

4.1.2 Der Periodic Housekeeping Task

Der zweithöchste Task ist der *Periodic Housekeeping Task*. Er sammelt die Daten für die alle acht Sekunden zu erstellenden Housekeeping Pakete H1 und für die Additional Housekeeping–Pakete H3 und H4. Er ist als periodischer Task an den Acht–Sekunden (oder 64 RTC) Zyklus der H1–Pakete angelehnt. Die ersten 32 RTC befindet er sich lediglich im Wartezustand. Die Daten sammelt er schließlich gleichmäßig verteilt über die zweite Hälfte der acht Sekunden.

4.1.3 Der Line Manager Task

Der *Line Manager Task* ist Teil der Verwaltung von den Interface–Leitungen zu den Instrumenten und zum HEPI. Da verschiedene Tasks Zugriffe auf eine High Speed Line (HSL) oder Low Speed Line (LSL) benötigen, kann es passieren, daß zwei Tasks die gleiche Leitung belegen möchten. Um solche Konflikte zu vermeiden, wurde ein Protokoll eingeführt. Bevor eine Leitung benützt wird, muß angefragt werden, ob die Leitung überhaupt frei ist. Ist sie es, so darf auf sie zugegriffen werden. Falls sie nicht frei ist, muß der Task warten, bis sie freigegeben wird. Hat ein Task Zugriff auf eine Leitung, so muß er sie für andere Tasks blockieren. Diese beiden Grundfunktionen dürfen nicht vom Task Scheduler unterbrochen werden. Wenn ein Task eine Leitung nicht mehr benötigt, muß er die Leitung freigeben und wartende Tasks darüber informieren, daß die Leitung nun wieder verfügbar ist. Der Line Manager Task ist nicht periodisch, sondern wird nur nach der Benutzung einer HSL oder LSL aktiv. Er bekommt von der CSSW ein Hardware–Signal, welches das Ende einer Übertragung anzeigt und gibt ein Software–Signal an den entsprechenden Task weiter.

4.1.4 Der TC–Handler

Auch der *TC–Handler* ist ein aperiodischer Task. Er wird dann aktiv, wenn der CSSW Telekommando Subbuffer nicht leer ist. Ist ein Telekommando angekommen, wird es vom TC-Handler ausgeführt, falls es ein TC an die IASW ist, oder es wird an die Instrumente weitergegeben. Da maximal acht Telekommandos pro Sekunde auftreten können, wird der TC–Handler höchstens einmal je RTC aktiv.

4.1.5 Die Histogramm Tasks

Die beiden Histogramm Tasks sind wiederum periodische Tasks. Da nur im Scientific Mode Histogramme erstellt werden, sind sie auch nur während des Scientific Mode aktiv. Im HEPI ist Speicherplatz für zwei Histogramme jeden Typs. Während in einem Speicherbereich ein Histogramm erstellt wird, kann das vorher erstellte Histogramm aus dem anderen Speicherbereich ausgelesen werden. Dem *Histogram Integration Task* unterliegt die gesamte Verwaltung, die zur Integration eines Histogramms notwendig ist. Er startet zu Beginn des Scientific Mode oder eines Pointings das Erstellen der entsprechenden Histogramme, er kontrolliert, ob die eingestellte Integrationszeit erreicht ist und beendet die Integration vor dem Wechsel in das Slewing oder vor dem Verlassen des Scientific Mode. Er verwaltet auch den HEPI–Speicher und zeigt an, in welchem Speicherbereich gerade ein Histogramm erstellt wird und aus welchem Bereich ein fertig integriertes Histogramm ausgelesen werden kann.

Der *Histogram Reading Task* hingegen steuert das Auslesen der bereits fertiggestellten Histogramme über das HBR-B. Er holt ein–kbyte–weise Daten aus dem Histogrammspeicher und übergibt sie den entsprechenden Scientific Function Routinen, welche die Daten komprimiert in Telemetriepakete umpacken und an die CSSW weiterleiten. Da der Histogram Reading Task eine höhere Priorität als der Main Task besitzt, würde er die Verarbeitung der HBR-A Events unterbrechen, bis alle bereits integrierten Histogramme ausgelesen sind. Damit dies nicht geschieht, ist dieser Vorgang so organisiert, daß das Auslesen gleichmäßig über die gesamte Integrationszeit des aktuell zu erstellenden Histogramms verteilt wird.

Außer den Spectral Timing–Histogrammen werden alle Histogramme über die HBR-B Datenleitung vom HEPI and die DPE übertragen. Das HBR-B Interface wird also nur vom Histogram Reading Task in Anspruch genommen und kann daher als Task–internes Interface realisiert werden.

4.1.6 Der HBR-A Task

Der Task mit der zweitniedrigsten Priorität ist der *HBR-A Task*. Er dient zum Abholen der Events vom HEPI via die HBR-A Datenleitung und wird folglich nur im Diagnostic Mode oder im Scientific Mode benötigt. Der Main Task aktiviert indirekt den HBR-A Task, wenn Events vom HEPI angefordert werden müssen. Zeigt der HEPI an, daß Daten bereitliegen, schreibt der HBR-A Task diese 1000–word–weise in einen Buffer innerhalb der IASW. Sind keine Daten verfügbar, so fragt er regelmäßig beim HEPI nach, ob Daten bereit sind. Durch Setzen eines Flags zeigt er dem Main Task an, daß der Buffer mit neuen Events gefüllt ist. Der Main Task arbeitet dann diesen Buffer ab, indem er ein Event nach dem anderen der entsprechenden Scientific Function Routine übergibt, die das Event schließlich in Telemetriepakete verpackt. Ist der Buffer vollständig abgearbeitet, fragt der Main Task nach neuen Daten und aktiviert dadurch indirekt wieder den HBR-A Task. Dieser fragt nun wieder beim HEPI nach Daten.

4.2 Einige Software Module

In diesem Kapitel sollen die Software Module beschrieben werden, die direkt am Ablauf der Datenverarbeitung beteiligt sind. Es handelt sich um die ADA Packages der wissenschaftlichen und diagnostischen Funktionen und deren Hilfsprogramme, die Module, welche die einzelnen Modi repräsentieren sowie das zentrale Interface der IASW für die erzeugte Telemetrie. Bei der Namensgebung der einzelnen Prozeduren und Funktionen wird immer der Name des Moduls vorangestellt. Durch einen Punkt getrennt folgt dann der eigentliche Name der Prozedur oder Funktion: MODUL_NAME.NAME_DER_ROUTINE. Damit kann man den Routinen, die in verschiedenen Modulen äquivalente Aufgaben erfüllen, gleiche Namen zuordnen und mittels des Modulnamens dennoch jede eindeutig benennen.

4.2.1 Das Modul IASW

Dieses Modul ist das Startmodul der gesamten IASW. Es beinhaltet lediglich die kleine Prozedur IASW.IASW aus fünf Befehlszeilen, welche jedoch die gesamte IASW initialisiert. Sie ruft die Initialisierungsprozeduren der Module DPE CSSW, TASKS, HSL INTERFACE und LSL INTERFACE auf. Erstere initialisiert den CSSW–Anteil der Software, die zweite definiert alle oben beschriebenen IASW–Tasks, die beiden letzten konfigurieren die seriellen Leitungen der DPE zu den Instrumenten und zum HEPI. Bei der fünften und letzten Prozedur dieses Startmoduls spricht der Name des Aufrufs für sich allein und bedarf eigentlich keiner weiteren Erläuterung: DPE_CSSW.START_DPE. Sie startet die durch die vorherigen Prozeduraufrufe initialisierten Tasks und damit den Programmablauf der IASW.

4.2.2 Das Modul TASKS

In diesem Modul werden alle IASW–Tasks initialisiert. Hier sind die Task–IDs, die Prioritäten sowie die Startadressen der Tasks definiert. Die Startadresse ist der Beginn derjenigen Prozedur, mit deren Abarbeitung der Task nach seiner Initialisierung starten soll. Schließlich wird jedem Tasks ein Initialisierungsstatus zugeordnet, der besagt, ob der Task mit der Initialisierung gestartet wird oder ob nicht. Der Main Task sowie die beiden Histogramm–Tasks bekommen den Status „nicht gestartet“, alle anderen den Status „gestartet“. Der Main Task wird vom Main Control Task nur für eine bestimmte Zeitspanne[1] innerhalb des 125 ms RTC–Takts gestartet und danach wieder gestoppt. Histogramme werden lediglich während des Scientific Mode generiert, folglich brauchen die Histogramm–Tasks auch nur für diese Zeit vom Main Task gestartet zu werden.

4.2.3 Das Modul MODE_DISPATCHER

Nach der Initialisierung der IASW befindet sich die Software im Stand by Mode. Wie alle Modi ist auch dieser im Main Task angesiedelt. Allerdings wird man vergeblich nach einem Modul dieses Namens suchen, da der Stand by Mode, im Gegensatz zum Scientific Mode oder Diagnostic Mode, nicht explizit programmiert ist. Der Standby Mode ist IASW–intern ein Warten auf einen Modewechsel. Das Wechseln eines Mode geschieht, indem der gerade aktive Mode gestoppt und der gewünschte Mode gestartet wird. Diese Aufgabe des Startens und Stoppens der Modi–Prozeduren obliegt dem Modul MODE_DISPATCHER. Nach der Initialisierung startet der Main Task durch den Aufruf der Prozedur MODE_DISPATCHER.START. Innerhalb dieser Prozedur wird eine Endlosschleife durchlaufen, welche den Stand by Mode realisiert. In dieser Schleife wird der Wert der Variablen `Running_Mode` abgefragt, welche den aktuell laufenden Mode repräsentiert. Ist als aktiver Mode der Stand by Mode oder der Patch & Dump Mode[2] eingestellt, wird eine leere Anweisung, also nichts, ausgeführt und die Schleife wird ständig wiederholt. Ist der Scientific Mode eingestellt, wird die Startprozedur des Scientific Mode SCIENCE_MODE.START aufgerufen, entsprechend für den Diagnostic Mode die Prozedur DIAGNOSTIC_MODE.START.

Wird via Telekommando ein Modewechsel gefordert, ruft der Task TC–Handler die Funktion MODE_DISPATCHER.MODE_TRANSITION auf. Diese prüft vor der Ausführung, ob der Modewechsel erlaubt ist oder nicht und gibt dies als Antwort dem TC–Handler Task zurück. Der Modewechsel wird realisiert durch eine Neubelegung der Variablen `Running_Mode`. Ein Wechsel vom aktiven Mode zu sich selbst ist zwar unsinnig, aber erlaubt. Es ändert sich nichts in der Software, aber das Telekommando gilt als akzeptiert und ausgeführt. Aus dem Diagnostic Mode oder dem Scientific Mode darf nur in den Stand by Mode gewechselt werden, auch ein Wechsel zwischen den Scientific Submodes muß über den Stand by Mode führen. Dazu muß zuerst der aktive Mode durch Aufruf der entsprechenden Prozedur DIAGNOSTIC_MODE.STOP oder SCIENCE_MODE.STOP beendet werden. Dann wird der Wert der Variable `Running_Mode` auf Stand by Mode gesetzt. Aus dem Stand by Mode kann in jeden anderen Mode gewechselt werden. Die Variable `Running_Mode` wird auf den entsprechenden Wert gesetzt, und beim nächsten Durchlauf der Schleife wird die Startprozedur des Mode aufgerufen. Bei einem unerlaubten Modewechsel gibt die Funktion eine Fehlermeldung zurück, worauf der TC Task das

[1] Zur Zeit der Anfertigung dieser Arbeit waren es 95 ms.

[2] Der Patch & und Dump Mode ist nicht implementiert, da diese Aufgabe bereits von der *Central Data Management Unit* durchgeführt werden kann. Dennoch erlaubt die IASW ein Wechseln in den Patch & Dump Mode via Telekommando, die Software verbleibt jedoch in der Endlosschleife der Prozedur MODE_DISPATCHER.START.

Telekommando als abgelehnt klassifiziert und eine On Event Message erzeugt.

In den Essential Housekeeping Paketen (H1), die alle acht Sekunden zusammengestellt werden, ist ein Feld implementiert, das den aktuellen Mode anzeigt. In diesem Feld steht eine Kopie der Variablen `Running_Mode`. Da diese jedoch lediglich als lokale Variable im Modul MODE_DISPATCHER definiert und somit dem Housekeeping Task wegen des Geheimnisprinzips nicht direkt zugänglich ist, muß sie mittels Aufruf einer Funktion, welche den Wert der Variablen zurückgibt, aus dem Modul ausgelesen werden. Im Modul MODE_DISPATCHER steht dafür die Funktion MODE_DISPATCHER.CURRENT_MODE zur Verfügung.

4.2.4 Das Modul HBR-A INTERFACE

In diesem Modul ist der HBR-A Task mit der Prozedur HBR-A_INTERFACE.INITIALISE realisiert. Die Schnittstelle mit dem HEPI befindet sich jedoch nicht hier, sondern im Modul HSL_INTERFACE, auf welches hier nicht weiter eingegangen werden soll. Im Modul HBR-A_INTERFACE befinden sich zwei Buffer von je 1000 word Größe, welche die vom HEPI ausgelesenen Datenblöcke aufnehmen. Diese sind global definiert, da andere Module auf sie Zugriff haben müssen. Die Aufgabe des HBR-A Task ist lediglich das Füllen dieser beiden Buffer. Zu beachten ist, daß das Auslesen und Füllen der beiden Buffer wechselseitig geschieht. Während der eine Buffer ausgelesen wird, wird der andere gefüllt. Die Variable `Read_Buffer` zeigt an, welcher der Buffer gerade ausgelesen wird. Die Variablen `Buffer_Filled(1/2)` geben an, ob der entsprechende Buffer mit noch nicht verarbeiteten Events gefüllt ist. In einer Endlosschleife wird abgefragt, ob der Buffer, der gerade nicht durch die Variable `Read_Buffer` zur Bearbeitung freigegeben ist, leer bzw ausgelesen ist. Ist das der Fall, so wird die lokale Prozedur HBR-A_INTERFACE.FILL_BUFFER aufgerufen, welche alle Schritte zum Füllen des Buffers enthält. Damit der Task seine Aktivität wieder an den Main Task abgibt, ist ein Wartezustand auf ein Softwaresignal, welches vom Main Task ausgelöst wird, zu Beginn der Endlosschleife eingebaut. Der HBR-A Task startet zwar sofort nach seiner Initialisierung, muß aber als erste Aufgabe auf das Signal vom Main Task warten. Wenn dieses Signal ankommt, durchläuft der Task seine Schleife, bis er wieder an dem Befehl zum Warten auf das Signal angelangt ist. Die Endlosschleife kann also nur einmal am Stück durchlaufen werden. Der Task gibt seine Aktivität wieder an den Main Task ab, bis dieser wieder das Signal sendet. Dieses Signal ist ebenfalls im Modul HBR-A_INTERFACE definiert und wird vom Main Task durch Aufruf der Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE ausgelöst.

Die Prozedur HBR-A_INTERFACE.FILL_BUFFER ist Teil des HBR-A Tasks, sie wird nur von der Prozedur HBR-A_INTERFACE.INITIALISE aufgerufen. Die Prozedur HBR-A_INTERFACE.FILL_BUFFER fragt zunächst einmal beim HEPI an, ob Daten bereitliegen. Wenn ja, meldet die Prozedur durch Aufruf der Prozedur LINE_MANAGER.GET_HSL die Benutzung der HSL an. Ist die Leitung bereits von einem anderen Task blockiert, muß so lange gewartet werden, bis die Leitung wieder freigegeben wird. Ist diese frei, so wird sie für andere Tasks blockiert und die Prozedur HSL_INTERFACE.START beginnt die Datenübertragung vom HEPI via das HBR-A. Nach der Übertragung muß die Variable `Buffer_Filled(1/2)` des entsprechenden Buffers auf „True“ gesetzt werden, um dem Main Task anzuzeigen, daß hier neue Events enthalten sind. Kann der HEPI keine Daten bereitstellen oder war ein Fehler in der Datenübertragung, so bleibt die Variable auf „False“ gesetzt, der Buffer behält den Status leer bzw. ausgelesen. In Kapitel 4.3.1 wird das Zusammenspiel der einzelnen Prozeduren und Module der IASW beschrieben, wenn der HEPI keine Daten zum Auslesen bereitliegen hat.

Der Main Task startet indirekt, wie bereits oben erwähnt, den HRA_A Task durch Aufruf der Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE nach der Abarbeitung eines Buffers. Diese gibt dem Main Task ein „Ja“ zurück, wenn der andere Buffer gefüllt ist und ein „Nein“, wenn keine Daten mehr bereitliegen. Modulintern wechselt sie den Zugriff des Main Tasks von dem ausgelesenen Buffer auf den anderen, falls dieser schon neu gefüllt ist. Daraufhin gibt sie dem HBR-A Task das Signal zum Starten, indem sie die modulinterne Prozedur HBR-A_INTERFACE.FETCH_DATA, welche ausschließlich aus dem Signalbefehl besteht, aufruft. Ist der andere Buffer noch nicht wieder gefüllt, das heißt, beide Buffer sind ausgelesen, dann wird lediglich mit dem Aufruf der Prozedur HBR-A_INTERFACE.FETCH_DATA das Füllen der Buffer veranlaßt, da ein Bufferwechsel ja keinen Sinn ergäbe.

4.2.5 Das Modul TM INTERFACE

Die Prozedur TM INTERFACE.PUT_TM ist die Schnittstelle zwischen allen innerhalb der IASW erzeugten Telemetriepaketen und der CSSW. Alle wissenschaftlichen Daten müssen dieses Interface passieren, genauso wie die Additional Housekeeping Pakete (H2, H3, H4) und die Report Parameter Pakete, welche beim Auslesen bzw. Ändern von Tabellen und Parametern generiert werden. Als erstes fragt die Prozedur mit Hilfe der ebenfalls in diesem Modul definierten Procedur TM INTERFACE.GET_ICB_TM_SUB_BUFFER_STATE die Anzahl der in der CSSW zwischengespeicherten[3] Telemetriepakete ab. Dieser Buffer kann maximal 50 Pakete aufnehmen. Sollte der Buffer bereits vollständig gefüllt sein, kann das Telemetriepaket nicht an die CSSW übergeben werden und geht verloren, wobei es zumindest durch einen Zähler verlorene Pakete registriert und der Verlust des Pakets durch eine On Event Message in den Essential Housekeeping Paketen H1 angezeigt wird. In Kapitel 4.3.2 wird darauf eingegangen, daß bei einer korrekten Einstellung der IASW dieser Fall selbst bei sehr hohen Datenraten nie auftreten kann. Ist Platz im Buffer vorhanden, wird das Paket der CSSW übergeben und der Zähler für gesendete Pakete wird um eins erhöht. Beim Abschicken der Pakete muß noch beachtet werden, ob das *V*acation Bit (VC) in der APID[4] gesetzt ist. Das VC Bit bestimmt, welche Telemetriepakete sofort beim MOC ausgewertet werden. Pakete mit VC-0 werden sofort ausgewertet. Dies sind alle Housekeeping Pakete (H1–H4). Außerdem wird für VC-0 Pakete eine Checksum berechnet und in das letzte word (16 Bit) des Paketes geschrieben. Damit kann kontrolliert werden, ob die Übertragung von der DPE zum MOC fehlerfrei funktioniert hat. Da die Berechnung der Checksum mit 1.53 ms für die Echtzeit–Datenverarbeitung viel Zeitaufwand bedeutet, wird bei den VC-1 Paketen auf deren Berechnung verzichtet.

Die Prozedur TM INTERFACE.PUT_TM weist noch eine Besonderheit auf. Jede Art von Telemetriepaket, gekennzeichnet durch eine eigene APID, hat einen eigenen Paketzähler, den *S*ource Sequence Counter. Normalerweise sind alle Sequence Counter in den Modulen lokal definiert, in denen auch die entsprechenden Telemetriepakete generiert werden. So gibt es in der IASW noch zwei verschiedene Sequence Counter für die Housekeeping Pakete H2 und H3–H4, wobei H3 und H4 einen gemeinsamen Sequence Counter verwenden. Diese Housekeeping Pakete bekamen aber alle drei dieselbe APID, weshalb sie auch nur einen einzigen gemeinsamen Sequence Counter haben sollten. Hinzu kommt die Schwierigkeit, daß die Pakete H3 und H4 von Housekeeping Task, die H2–Pakete jedoch vom Main Task erstellt werden. Um dieses Problem mit wenig Verwaltungsaufwand zu lösen, wurde der gemeinsame Sequence Counter lokal in der Prozedur TM_INTERFACE.PUT_TM definiert, da alle drei Pakete über dieses Interface an die

[3] Siehe Kapitel 3.3.

[4] Application Process Identification, siehe Kapitel 5.

CSSW übergeben werden. Dieser überschreibt die nach wie vor vorhandenen einzelnen Sequence Counter der H2- bzw. H3–H4 Pakete, die er durch Abfrage der gemeinsamen APID erkennt. Falls diese Pakete doch einmal getrennte Sequence Counter haben sollen, kann dies durch Abändern der APIDs, was die IASW ermöglicht, realisiert werden.

Wäre die Datenrate so hoch, daß mehr Telemetriepakete erzeugt würden als abgeschickt werden können, wäre der ICB Telemetrie Subbuffer der CSSW ständig maximal gefüllt und erzeugte Pakete gingen verloren. Damit in diesem Fall keine Pakete verloren gehen, greift das Modul TM_INTERFACE mit der Funktion TM_INTERFACE.MUST_I_PROCESS auch aktiv in die Erzeugung von Telemetriepaketen ein (siehe Kapitel 4.3.2). Dafür wird in einer lokalen Tabelle namens `TM_State` jeder Art von Telemetriepaket ein Status zugeordnet, der besagt, ob von dieser Paketart ein weiteres generiert werden darf oder ob nicht. Entschieden wird das aufgrund des aktuellen Füllstandes des ICB Telemetrie Subbuffers und einer weiteren lokalen Tabelle, der Variablen `TM_Table`, in der jeder Art von Paket zwei Werte zugeordnet sind, einen `High_Buffer_State_Value` (HBSV) und einen `Low_Buffer_State_Value` (LBSV), die beide bestimmte Füllstände des ICB Telemretrie Subbuffers repräsentieren sollen. Wird der aktuelle Füllstand höher als der `High_Buffer_State_Value`, dann wird der `TM_State` der entsprechenden Paketart auf „False“ gesetzt, es dürfen keine weiteren Pakete dieser Art generiert werden[5]. Sinkt der Füllstand wieder unter den `Low_Buffer_State_Value`, wird die Erzeugung der Paketart durch „True“–Setzen des `TM_State` wieder erlaubt. Der Aufrufende der Funktion TM_INTERFACE.MUST_I_PROCESS benötigt keine Kenntnis über den beanspruchten Teil des ICB Buffers, er braucht lediglich die von der Funktion zurückgegebene Antwort „Ja“ oder „Nein“ zu der Frage, ob er verarbeiten darf oder ob nicht.

Auch in diesem Modul gibt es lokale Werte, die mittels Funktionen in Housekeeping Pakete geschrieben werden. So fragt der Housekeeping Task über die Funktion TM_INTERFACE.GET_TM_STATE_FLAGS für die alle acht Sekunden erstellten Essential Housekeeping H1 nach den aktuellen `TM_States` der Telemetriepakete. Die Zähler für die gesendeten und verlorengegangenen Pakete sowie der Füllstand des ICB Telemetrie Subbuffers werden bei der Generierung von H2–Paketen mit der Funktion TM_INTERFACE.GET_HOUSEKEEPING_DATA aus diesem Modul angefordert. Auf diese Funktionen soll aber nicht weiter eingegangen werden. Gleiches gilt auch für die Routinen TM_INTERFACE.READ_TABLE zum Lesen und TM_INTERFACE.WRITE_TABLE zum Auswechseln der in diesem Modul definierten Tabellen.

4.2.6 Das Modul SCIENCE_MODE

Wie der Name bereits erahnen läßt, befindet sich in diesem Modul die Prozedur SCIENCE_MODE.START, welche den Scientific Mode realisiert und vom MODE_DISPATCHER aufgerufen wird. Bevor die eigentliche Mode–Schleife erreicht wird, muß diese Prozedur alle notwendigen Vorbereitungen treffen. Zuerst werden die internen Eventzähler auf Null und alle Scientific Functions auf definierte Anfangszustände zurückgesetzt. Das Verarbeiten aller Events wird durch das Öffnen aller `TM_State`–Flags im Modul TM_INTERFACE gestattet. Dann werden die beiden Einlesebuffer der HBR-A_INTERFACES gelöscht. Die interne Funktion SCIENCE_MODE.INIT_HEPI synchronisiert, nach dem Anmelden der LSL durch die Prozedur LINE_MANAGER.GET_LSL, den HEPI mit den Detektoren ISGRI und PICsIT, wobei die Ausführung der Funktion HEPI.SYNCHRONIZE im schlimmsten Falle bis zu drei Sekunden dau-

[5]Dies gilt auch für die Additional Housekeeping H2, H3 und H4. Diese müssen jedoch jederzeit erzeugt und abgeschickt werden dürfen. Deshalb müssen die Grenzwerte HBSV/LBSV immer korrekt gesetzt sein.

ern kann. Danach stellt sie den gewünschten Scientific Submode durch Setzen des HEPI Status Registers ein. Schließlich gibt sie mit LINE_MANAGER.RELEASE_LSL die LSL wieder frei und beendet sich durch Rückgabe eines logischen Wertes, der anzeigt, ob bei der Kommunikation mit HEPI Fehler aufgetreten sind. Der Wert wird invertiert der modulinternen Variablen `Stopped` zugewiesen, die darüber entscheidet, ob die eigentliche SCIENCE_MODE–Schleife überhaupt durchlaufen wird. Vor dem Erreichen dieser Schleife werden noch die beiden Histogramm–Tasks gestartet und ein Housekeeping H2 erzeugt, um den Status der IASW zum Eintrittszeitpunkt in den Scientific Mode auszugeben. Als Bedingung zum Durchlaufen der Schleife wird der Wert von `Stopped` abgefragt. Solange diese nicht den Wert „True“ enthält, wird die Schleife durchlaufen. In der Schleife wird zuerst geprüft, ob der Satellit zwischen Pointing und Slew gewechselt hat, da sich in diesem Falle die Zusammenstellung der Scientific Function innerhalb des Submodes ändern und dies im HEPI eingestellt werden muß. Ein solcher Wechsel von Pointing nach Slew oder umgekehrt wird durch die Generierung eines H2 Paketes angezeigt. Danach wird mittels der Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE nachgefragt, ob ein gefüllter Buffer zum Auslesen bereitliegt. Falls nein, fragt die IASW die IASW regelmäßig weiter nach Daten an (siehe Kapitel 4.3.1). Ist ein gefüllter Buffer vorhanden, werden die Events einzeln aus dem Buffer herausgenommen und der Eventtyp festgestellt. Wenn die Funktion TM_INTERFACE.MUST_I_PROCESS das Verarbeiten des Events erlaubt, wird es der entsprechenden Scientific Function Packroutine übergeben und anschließend als verarbeitetes Event diesen Typs gezählt. Ist die Verarbeitung nicht gestattet, wird es lediglich als nichtverarbeitetes Event diesen Typs gezählt. Für jeden Eventtyp gibt es einen Zähler für verarbeitete und nicht verarbeitete Events, ebenso für nicht erkannte und daher undefinierte Events. Außerdem werden alle Ereignisse, die durch das Modul SCIENCE_MODE laufen, nochmals mit einem eigenen Zähler registriert, die gesamte Eventzahl kann somit kontrolliert werden. Am Ende der Schleife wird nachgeprüft, ob der HEPI resynchronisiert werden muß. Dies ist der Fall, wenn innerhalb der letzten acht Sekunden die Datenrate ein fest definiertes Minimum (Standard: 1000 Events) unterschreitet oder zu viele undefinierte Events (Standard: ein undefiniertes Event) aufgetreten sind. Die SCIENCE_MODE–Schleife wird erst verlassen, wenn, ausgelöst von einem Telekommando, der TC Task durch Aufruf der Prozedur SCIENCE_MODE.STOP die Schleifenvariable `Stopped` auf „True“ setzt. Die Histogramm–Tasks werden daraufhin gestoppt, die angefangenen Telemetriepakete werden mit SCIENCE_MODE.CLOSE_TMS von den Scientific Function Routinen abgeholt und ein Housekeeping Paket H2 zusammengestellt, welches den Status der IASW beim Verlassen des Scientific Mode ausgibt. Darin befinden sich auch alle Zählerstände aus dem Modul SCIENCE_MODE.

4.2.7 Das Modul DIAGNOSTIC_MODE

Der Diagnostic Mode, als Prozedur DIAGNOSTIC_MODE.START realisiert, ist identisch mit dem Modue SCIENCE_MODE aufgebaut, jedoch fallen viele Operationen weg, die im Scientific Mode notwendig sind. Der Hauptgrund liegt darin, daß während des Scientific Mode nur ein Wechsel von Pointing/Slew oder ein Modewechsel in den Stand by Mode als gültige Telekommandos zugelassen sind. Im Diagnostic Mode sind hingegen alle Telekommandos erlaubt. So werden vor Erreichen der Diagnostic Mode Schleife innerhalb der Prozedur DIAGNOSTIC_MODE.START auch die internen Zähler und die `TM_State`–Flags im Modul TM_INTERFACE zurückgesetzt, die Einlesebuffer des HBR-A_INTERFACE gelöscht und ein Housekeeping Paket H2 erzeugt. Das Synchronisieren des HEPI sowie das Einstellen des HEPI Status Registers wird nicht, wie es im Modul SCIENCE_MODE geschieht, automatisch von der Prozedur erledigt. Falls erforderlich, kann dies mit einem Telekommando nachgeholt werden. Der Aufbau der Mode–Schleife ist identisch zur Schleife im Modul SCIENCE_MODE. Auch im Diagnostic Mode wird die Schleife durchlaufen, bis die modulinterne Schleifenvariable `Stopped` auf „True“ gesetzt wird. Innerhalb der Schleife wird ebenfalls über die Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE nachgefragt, ob Daten bereitliegen. Diese werden auch einzeln aus den Buffern genommen und verarbeitet. Es gibt drei diagnostische Funktionen, die unabhängig voneinander via Telekommando ein- und ausgeschaltet werden können. Es muß also erst einmal abgefragt werden, ob die Funktion eingeschaltet ist. Danach wird bei jeder der Funktionen wieder über Aufruf von TM_INTERFACE.MUST_I_PROCESS nachgefragt, ob die Verarbeitung erlaubt ist. Die *Transparent Function* (D3) kopiert alles aus dem Buffer in Telemetriepakete, ob es sich nun um gültige Events handelt oder um Datensalat. Die diagnostischen Funktionen D1 und D2 verarbeiten nur die gültigen Raw–Events eines Detektors, ISGRI bei D1 und PICsIT bei D2. Auch im Diagnostic Mode gibt es für jede Funktion einen Zähler für verarbeitete und nicht verarbeitete Events. Wie im Modul SCIENCE_MODE gibt es auch einen unabhängigen Zähler für alle Events, die das Modul DIAGNOSTIC_MODE passieren. Nach Aufruf der Prozedur DIAGNOSTIC_MODE.STOP durch den TC Task wird die Schleife beendet. Wie im Modul SCIENCE_MODE werden alle angefangenen Telemetriepakete von den Packroutinen abgeholt und ein Housekeeping Paket H2 generiert, welches alle Zählerstände enthält.

4.2.8 Das Modul ENERGY_CORRECTION

Sowohl für die ISGRI- als auch für die PICsIT–Events wird während des Scientific Mode schon an Bord des Satelliten eine Korrektur der Amplitude durchgeführt. Die Amplituden des CsI–Detektors werden, aufgrung der hohen erwarteten Datenrate von PICsIT, von der viel schnelleren HEPI–Hardware korrigiert. Für die CdTe–Events ist hingegen die IASW zuständig. Es werden jedoch immer die orginalen, nicht korrigierten Amlituden des CdTe–Events in die Telemetriepakete geschrieben. Die korrigierten Werte werden nur für den internen Gebrauch zur Datenselektion verwendet.

Realisiert wird die Korrektur der CdTe–Amplituden von der Funktion ENERGY_CORRECTION.ENERGY_CORR in dieser Form:

Korrigierte Amplitude = Gemessene Amplitude * Gain + Offset.

Die beiden Parameter `Gain` und `Offset` können experimentell ermittelt und als Funktionsgraphen in Abhängigkeit von der Anstiegszeit, der *Risetime*, aufgezeichnet werden. Die IASW nähert diesen Graphen durch fünf gerade Teilstrecken an. Dazu müssen für jeden der Korrekturgraphen von Gain und Offset sechs Stützstellen und deren Funktionswerte in je einer Nachschlagetabelle gegeben sein. Soll eine Amplitude korrigiert werden, prüft die IASW nach, zwischen welchen Stützstellenwerten sich die mit der Amplitude gekoppelten Risetime befindet und ermittelt damit, welche der fünf Ausgleichsgeraden verwendet werden muß. Die Korrekturwerte werden schließlich über die entsprechende Geradengleichung ermittelt:

$$\begin{aligned}
\texttt{Gain} &= \left[\frac{(G(N+1)-G(N))*(\text{Risetime}-R(N))}{R(N+1)-R(N)}+G(N)\right]*\frac{1}{\text{G-Skalierungsfaktor}} \\
\texttt{Offset} &= \left[\frac{(O(N+1)-O(N))*(\text{Risetime}-R(N))}{R(N+1)-R(N)}+O(N)\right]*\frac{1}{\text{O-Skalierungsfaktor}}
\end{aligned}$$

Der Wert der zur Amplitude gehörenden Risetime liegt zwischen den beiden Stützstellen N mit Risetimewert R(N) und N+1 mit Risetimewert R(N+1). Der Gain–Wert an der Stützstelle N beträgt G(N), derjenige an der Stelle N+1 entsprechend den Wert G(N+1), entsprechend für den

Offset. Die Geradengleichung wird schließlich mit dem Kehrwert des G–Skalierungsfaktors multipliziert. Abbildung 4.1 verdeutlicht noch einmal die Approximation an den Graphen. Die beiden Stützstellen zum Begin (Risetime=0) und Ende (Risetime=127) des Risetime Wertebereichs sind fest vorgeschrieben, die anderen vier frei einstellbar. Durch geeignete Wahl der Stützstellen kann ein bestimmter Teil des Graphen genauer approximiert werden, wobei in anderen Bereichen Genauigkeit verloren gehen kann.

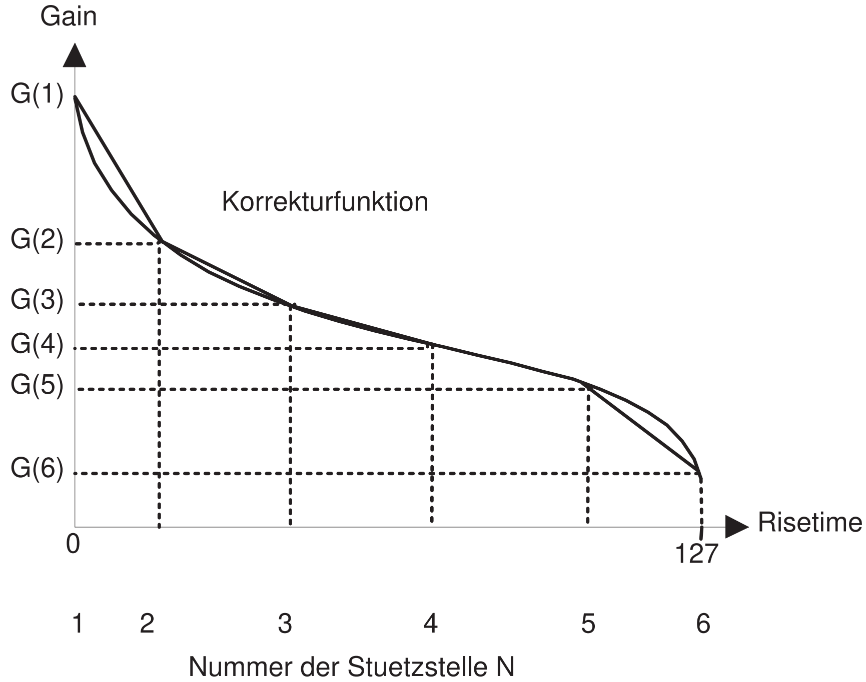

Abbildung 4.1: *Die Amplitudenkorrektur*

Da jedes der 128 x 128 CdTe–Pixel ein eigener Detektor mit individuellen Korrekturgraphen für Gain und Offset ist, werden für jedes Pixel für Gain- und Offsetkorrektur zu je sechs Stützstellenwerte deren Gainwerte bzw. Offsetwerte in einer Nachschlagetabelle abgelegt. Jeder dieser ganzzahligen Werte wird in einem word = 16 Bit gespeichert. Die gesamte Tabelle hat somit einen Speicherbedarf von 128 x 128 x 6 = 196 608 word. Der logische Adressraum eines *Address States*[6] (AS) beträgt maximal 65 536 word. Die Nachschlagetabelle wird daher geteilt und auf fünf AS verteilt gespeichert. Jeder Tabellenteil ist in einem eigenen Modul E_CORR1–5 untergebracht. In diesen Modulen stehen je eine Prozedur zur Verfügung, um die vier benötigten Werte G(N), G(N+1), O(N) und O(N+1) aus den Tabellen auszulesen. Die Tabelle ist zwar in fünf verschiedenen Modulen unterschiedlichen AS untergebracht, wird aber dennoch als eine einzige Tabelle verwaltet. Die implementierten Standardwerte wurden aus der Korrekturfunktion eines vermessenen Pixels bestimmt und für alle Pixel übernommen.

[6]Siehe Kapitel 2.2.3.

Stützstelle	Risetime	Gain–Werte	Offset–Werte
1	0	G(1) = 8944	O(1) = 45938
2	3	G(2) = 9176	O(2) = 39161
3	7	G(3) = 9553	O(3) = 51673
4	18	G(4) = 10836	O(4) = 65533
5	46	G(5) = 20443	O(5) = 44965
6	127	G(6) = 65535	O(6) = 65535

Tabelle 4.2: *Die Standardwerte der Amplitudenkorrektur*

Die Funktion ENERGY_CORRECTION.ENERGY_CORR benötigt als Eingabe die zu korrigierende Amplitude, deren Risetime sowie das Pixel, für welches die Korrektur vorgenommen werden soll. Jedes Pixel wird durch zwei Positionen Y–Pos und Z–Pos in der Detektorebene identifiziert. Mit einer ersten Nachschlagetabelle werden über die Risetime des Events die beiden Stützstellen sowie deren Risetimewerte ermittelt. Danach werden durch Aufruf einer der Prozeduren E_CORR1–5.LOOKUP die zu den Stützstellen gehörigen Funktionswerte von Gain und Offset geholt. Dazu werden die Positionen und die Nummer der unteren Stützstelle übergeben. Die Entscheidung, welche der fünf Prozeduren verwendet werden muß, wird mit der Y–Position getroffen. Die Werte aller Pixel mit Y–Positionen zwischen 0 – 25 liegen im ersten Teil der Tabelle im Modul E_CORR1, diejenigen zwischen 26 – 51 im zweiten Teil in E_CORR2 und so weiter. Nun sind alle notwendigen Werte ermittelt, um die Korrekturberechnung durchführen zu können. Die korrigierte Amplitude wird an den Aufrufenden zurückgegeben.

4.2.9 Die Packroutinen der Scientific und Diagnostic Functions

Die Module der wissenschafltlichen Funktionen S1–S8 sowie der diagnostischen Funktionen DIAGNOSTIC und TRANSPARENT sind, da die Aufgabe ihrer Routinen für alle die gleiche ist, ähnlich aufgebaut. Sie komprimieren die Events bzw. die Histogramme und verpacken diese in einem neuen Format in die entsprechenden Telemetriepakete. In all diesen Modulen findet man die Prozeduren PROCESS_EVENT, CLOSE_TM, RESET sowie eine Funktion zur Ausgabe von lokal definierten Variablen, die in Housekeeping Pakete geschrieben werden wie z.B. die Sequence Counter. Hinzu kommen bei einzelnen Modulen noch interne Prozeduren oder Funktionen.

Exemplarisch für alle wird im folgenden die Packroutine der wissenschaftlichen Funktion Compton Photon by Photon S3.PROCESS_EVENT genauer erläutert. Ein Compton–Event beinhaltet die Informationen von einem CdTe- und einem CsI–Ereignis, welche gleichzeitig registriert und möglicherweise durch ein comptongestreutes Photon ausgelöst wurden. Die beiden Detektoren ISGRI und PICsIT markieren ihre Ereignisse durch je 24 Bit mit einer Zeitauflösung von 238,4 ns. HEPI prüft die Ereignisse auf Zeitkoinzidenz. Die Ereignisse werden als gleichzeitig anerkannt, wenn die Zeitdifferenz kleiner als oder gleich eines vorgegebenen Zeitfensters ist. Mit Hilfe eines 8 Bit großen Registers[7] ist dieses Zeitfenster zwischen 238,4 ns und 61 μs einstellbar. Im Falle einer Koinzidenz faßt der HEPI die beiden Einzelevents zu einem Compton–Event zusammen. Im Compton–Event stehen also vier Positionen, zwei Amplituden, die Risetime des CdTe–Events sowie eine gemeinsame Eventzeit. Zuerst wird das vom SCIENCE_MODE.START

[7] Nr. 17, Register Type 1.

übergebene Event, welches aus zwei Blocks mit je fünf word (2 x 5 x 16 Bit = 160 Bit) besteht, in die Variable `Input_Event` kopiert, welche die Bitfolge des Events gemäß ihrer Bedeutung unterteilt und damit einen genauen Zugriff auf die einzelnen Informationen wie Positionen oder Amplituden erlaubt. Überschreiten die beiden Amplituden zusammenaddiert eine einstellbare Mindestenergie, wird das Compton–Event sofort weiterverarbeitet.

Ist die Gesamtenergie kleiner als der Grenzwert, wird nachgerechnet, ob ein echtes Compton–Ereignis oder nur eine zufällige Koinzidenz zwischen den Detektoren vorliegt (siehe Abbildung 4.3). Als Vorbereitung dazu muß die CdTe–Amplitude von der IASW korrigiert[8] werden, die Korrektur der CsI–Amplitude geschah bereits innerhalb des HEPI[9]. Die nötigen Algorithmen zur Comptonberechnung sind in der internen Funktion C_SELECT zusammengefaßt. Da in diese Berechnungen der geometrische Abstand zwischen den Ereignissen eingeht, müssen die unterschiedlichen Pixelgrößen von ISGRI und PICsIT sowie die Lücken zwischen den einzelnen Modulen eines Detektors mit berücksichtigt werden. Als Entfernungseinheit wird die CdTe–Pixelgröße gewählt. Um die Lücken einzubeziehen, wird die Detektorebene von ISGRI statt als Feld aus 127 x 127 Pixel als ein Array von 130 x 134 (Y–Pos. x Z–Pos.) angesehen, bei PICsIT entsprechend 65 x 67 (Y–Pos. x Z–Pos.) anstelle von 64 x 64 Pixel. Die Umrechnung erfolgt mit den Formeln:

$$
\begin{aligned}
Y_{CdTe} &= \left(\tfrac{Y_{CdTe}}{64} * 2\right) + Y_{CdTe} & \qquad Y_{CsI} &= \left(\frac{Y_{CsI}}{32} + Y_{CsI}\right) * 2 \\
Z_{CdTe} &= \left(\tfrac{Z_{CdTe}}{32} * 2\right) + Z_{CdTe} & \qquad Z_{CsI} &= \left(\frac{Y_{CsI}}{16} + Z_{CsI}\right) * 2.
\end{aligned}
$$

Auch der Abstand h der Detektorebenen zueinander wird durch eine Konstante in Einheiten von CdTe–Pixel beschrieben. Die Amplitudenwerte der Detektoren müßten vor der eigentlichen Berechnung ebenfalls auf eine einheitliche Größe, z.B. Elektronenvolt (eV) gebracht werden. Da jedoch noch keine Eichungen[10] von den Detektorgruppen zum IBIS–Team nach Tübingen gelangten, konnte noch keine Vereinheitlichung der Amplituden implementiert werden. Für das *Engineering Model* (EM) werden die CdTe–Amplitudenwerte, wie sie aus der Korrektur herauskommen, direkt als Kiloelektronenvolt (keV) verwendet. Für die CsI–Amplitude wird bei Einzelereignissen 5 keV, bei Mehrfachevents 10 keV pro Einheit angenommen. Mit den Positionsinformationen und den gemessenen Energien wird nun der Einfallswinkel α des Photons mit Hilfe der selbst hergeleiteten Rekonstruktionsformel

$$
\begin{aligned}
cos\alpha \;=\; & \frac{h}{\sqrt{h^2+\Delta y^2+\Delta z^2}} * \left(1 - \frac{E_{CdTe}}{E_{CsI} * (E_{CdTe}+E_{CsI})} * mc^2\right) \\
& \pm \sqrt{\left(1 - \frac{h^2}{h^2+\Delta y^2+\Delta z^2}\right) * \left(1 - \left[1 - \frac{E_{CdTe}}{E_{CsI} * (E_{CdTe}+E_{CsI})} * mc^2\right]^2\right)}
\end{aligned}
$$

ermittelt. Das Plusminus berücksichtigt die beiden Extremfälle der Ausrichtung des Comptonkegels, einer Steuung zur Beobachtungsachse hin oder von ihr weg. Die Rekonstruktionsformel ist nur für Ereignisse korrekt, deren erste Interaktion in ISGRI stattgefunden hat. Zur Wurzelberechnung wird der Newton–Algorithmus

$$U_{n+1} = \frac{1}{2} * \left(U_n + \frac{Wert}{U_n}\right) \qquad \text{mit Startwert} \qquad U_0 = Wert$$

[8] Siehe Kapitel 4.2.8.

[9] Siehe Kapitel 2.2.1.

[10] Wieviel eV entsprechen $\Delta 1$ Amplitudenwert.

mit der Abbruchbedingung Differenzquadrat $(U_n - U_{n+1}) < 0.01$ verwendet. Nur falls der rekonstruierte Winkel kleiner als 14.5° ist, gilt das Ereignis als ein echtes Compton–Event und wird weiter verarbeitet.

Das Event wird nun in einem anderen Format in ein Telemetriepaket geschrieben. Um die Datenmenge so weit wie möglich zu reduzieren, werden alle Bits, die keine Information beinhalten (Spare Bits), nicht übernommen. Die zweite Komprinierungsmöglichkeit liegt bei der 32 Bit Zeitinformation. Es ist nicht notwendig, für jedes Event die gesamten 32 Bit, die eine Zeitspanne von 72 Stunden erfassen, in das Telemetriepacket zu kopieren, da der Zeitabstand zwischen zwei Ereignissen bei einer erwarteten Datenrate von 280 Compton–Events pro Sekunde im Millisekundenbereich liegt. Vom ersten Event werden deshalb 23 MSB[11] in den *Data Field Header* des Telemetriepakets geschrieben, die restlichen 9 LSB[12] werden in das neue Event-format kopiert. Die gesamte Zeitinformation bleibt erhalten, solange die Zeitdifferenz zweier Ereignisse nicht größer wird als durch 8 Bit ausdrückbar ist, ungefähr 30 Millisekunden. Sollte die Zeitdifferenz größer sein, muß die Rekonstruierbarkeit der Zeitinformation durch Einfügen spezieller Dummy–Events gewährleistet werden (siehe Abbildung 4.6). In das Telemetriepaket werden keine manipulierten Werte geschrieben. Es stehen dort also keine Relativzeiten der Events zueinander, sondern immer die echte Zeitinformation. Auch bei der CdTe–Amplitude wird der nicht korrigierte Wert übernommen, allerdings werden lediglich die 8 MSB der 11 Bit langen Amplitudeninformation ins Telemetriepaket geschrieben. Ebenso verhält es sich mit der CsI–Amplitude, von der ebenfalls nur die 8 MSB von 10 Bit Amplitudeninformation übernommen werden. Ist das Telemetriepaket vollständig mit Events gefüllt, wird es durch Aufruf der Procedure TM_INTERFACE.PUT_TM an die CSSW weitergegeben (siehe Abbildung 4.7). Die folgenden Seiten verdeutlichen schematisch noch einmal den Ablauf der Procedure S3.PROCESS_EVENT.

Die RESET–Prozeduren setzen die internen Variablen der Module auf die Initialisierungswerte zurück. Insbesondere die Variablen, welche die Position innerhalb des aktuell zu erstellenden Telemetriepackets anzeigen, werden auf den Paketanfang gesetzt. Die RESET–Prozeduren sollen gewährleisten, daß sich die Packroutinen bei jedem Start des Scientific- oder Diagnostic Mode im gleichen Ausgangszustand befinden.

Bei Beendigung des Scientific Mode oder des Diagnostic Mode sind in der Regel in allen Modulen teilweise gefüllte Telemetriepackete vorhanden. Die in diesen Paketen enthaltenen Ereignisse gingen bei einem erneuten Start des Mode verloren, da die Positionsvariablen aller Packroutinen zu Beginn des Mode zurückgesetzt und die Events dadurch überschrieben werden. Deswegen werden bei einem Modeende alle angefangenen Telemetriepakete mit Hex(FFFF) definiert aufgefüllt und abgeschickt. Dazu stehen in allen entsprechenden Modulen die Prozeduren CLOSE_TM bereit, die aus der Procedure SCIENCE_MODE.START bzw. DIAGNOSTIC_MODE.START vor deren Beendigung aufgerufen werden.

[11]Most Significant Bit.
[12]Least Significant Bit.

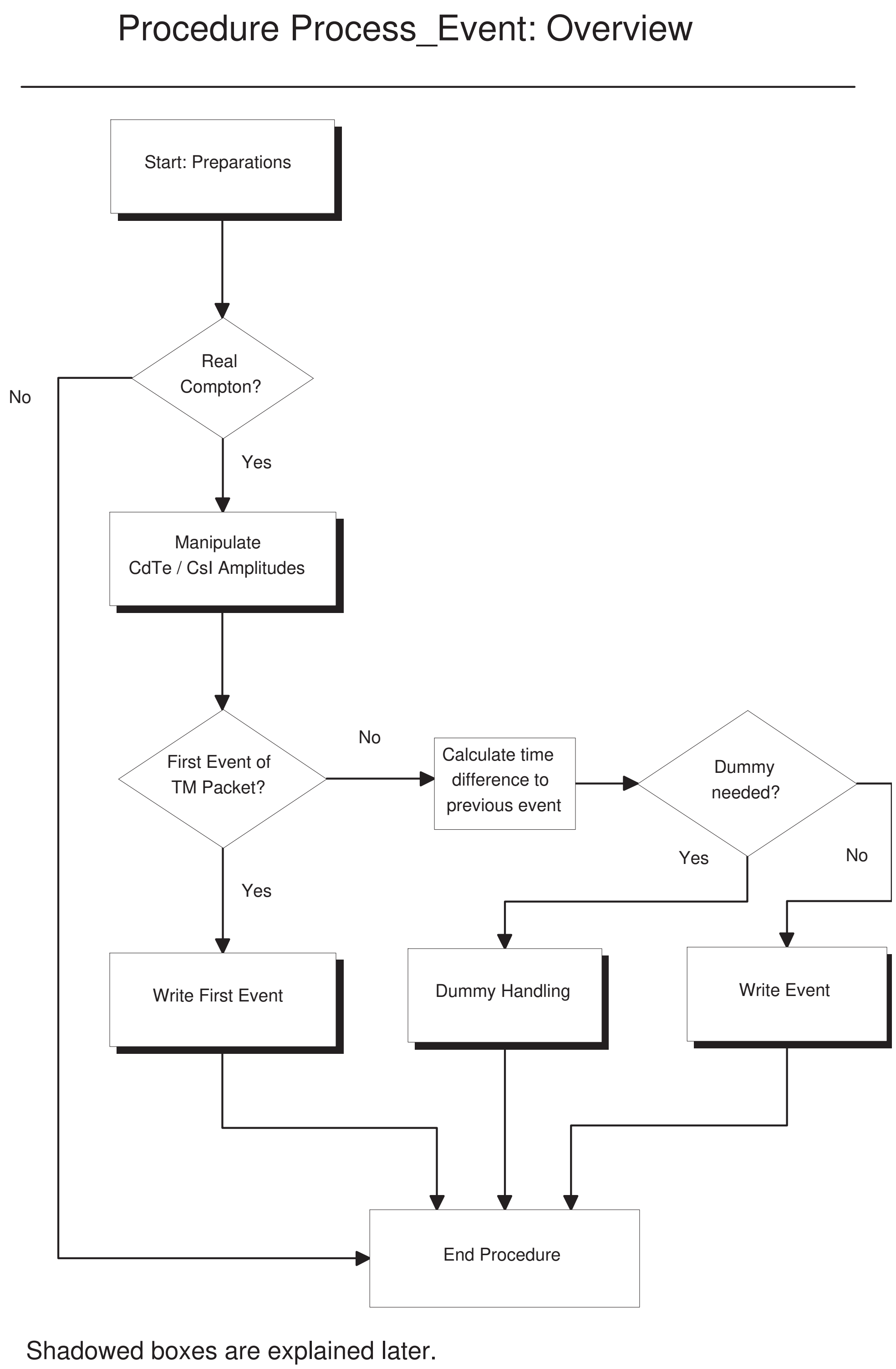

Abbildung 4.2: *Überblick des Gesamtablaufs*

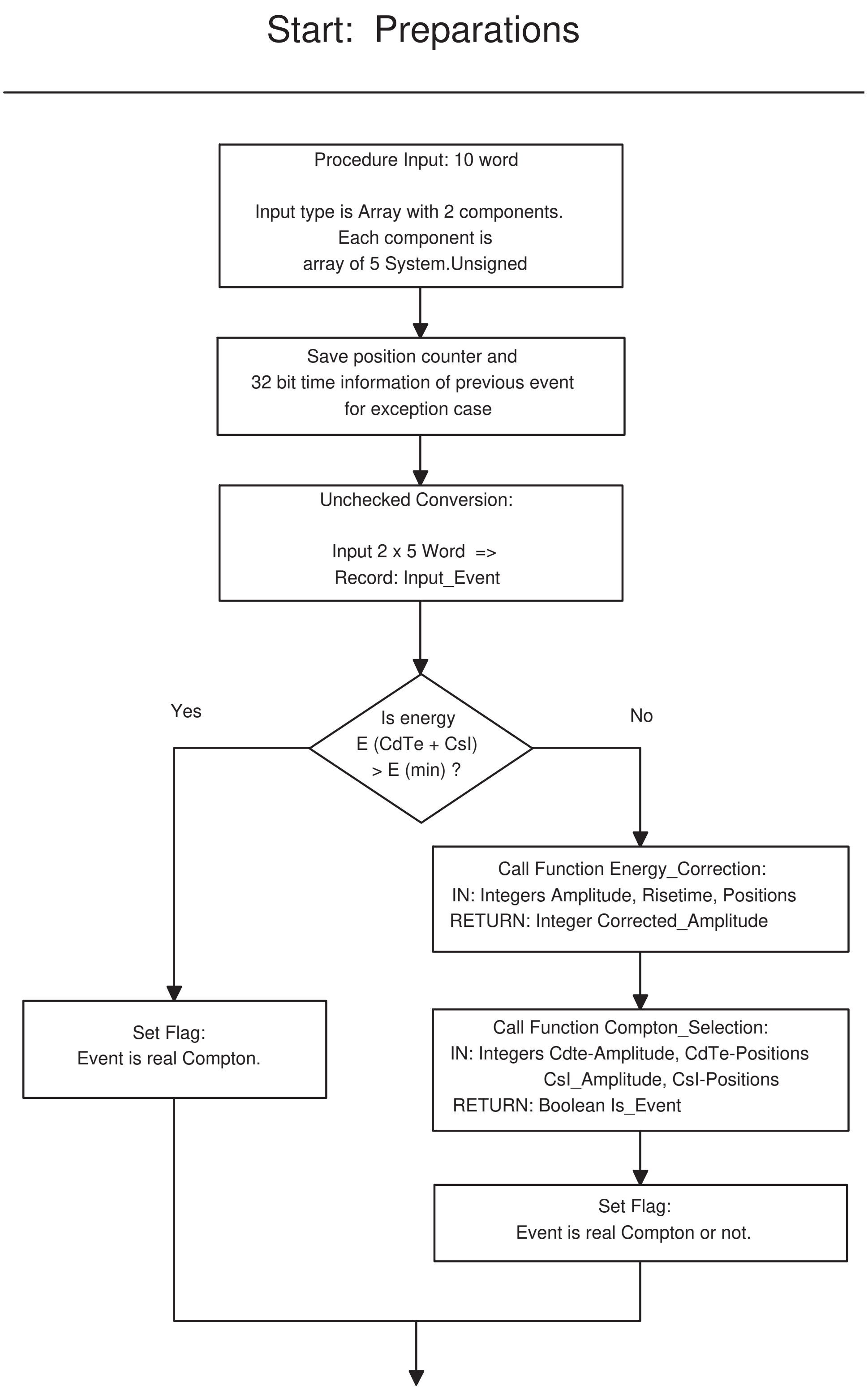

Abbildung 4.3: *Startvorbereitungen*

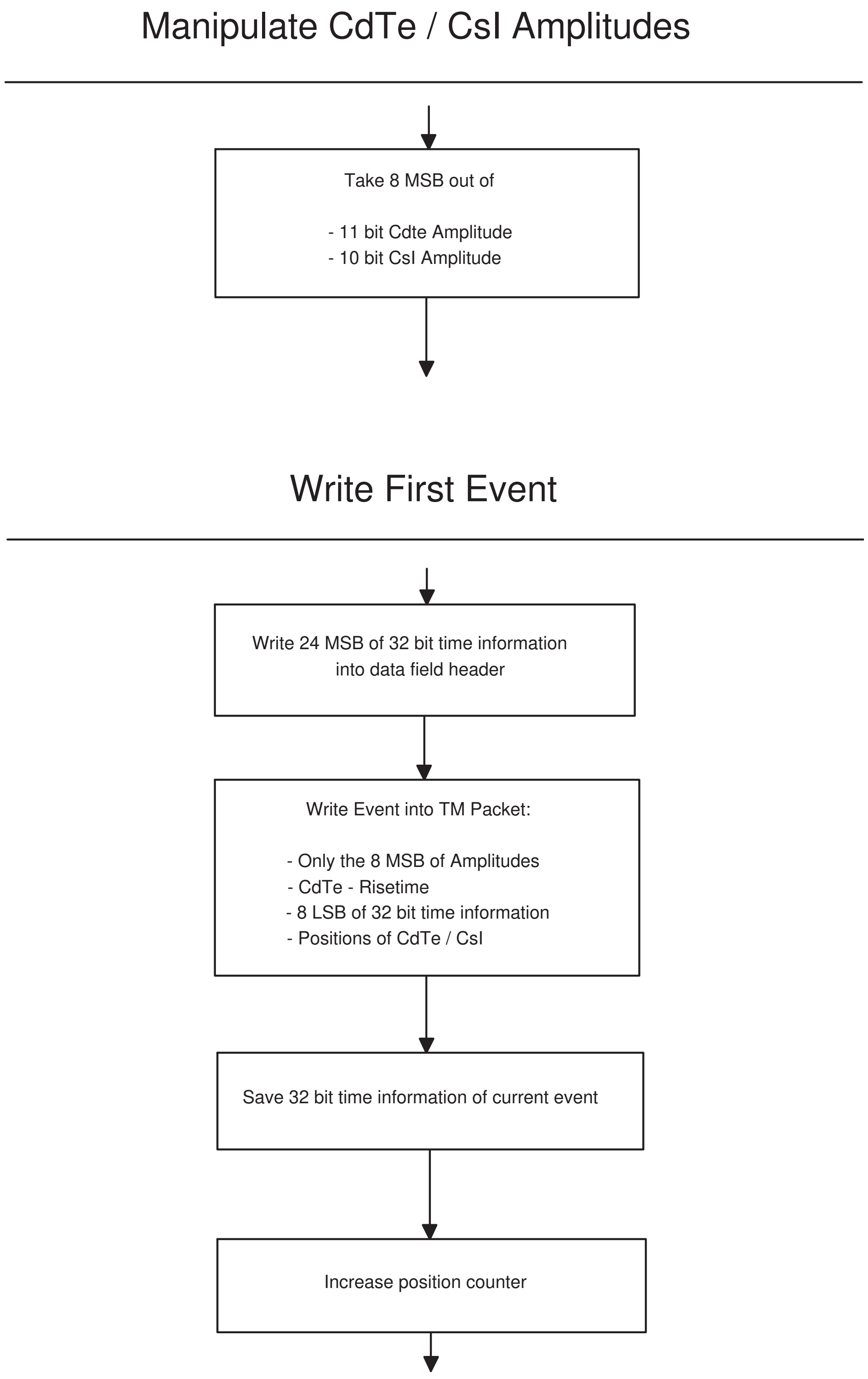

Abbildung 4.4: *Erstes Event des Telemetriepakets*

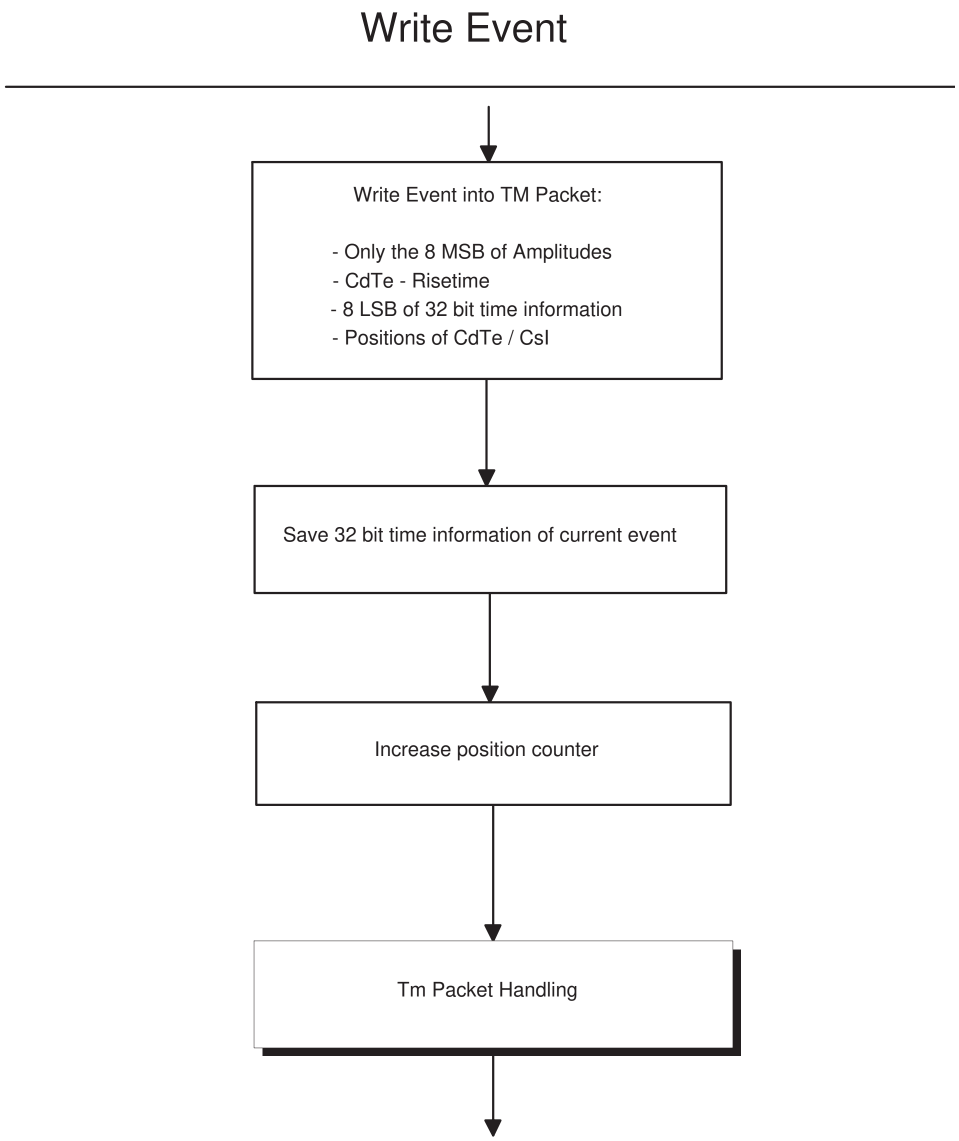

Abbildung 4.5: *Verarbeitung eines Events*

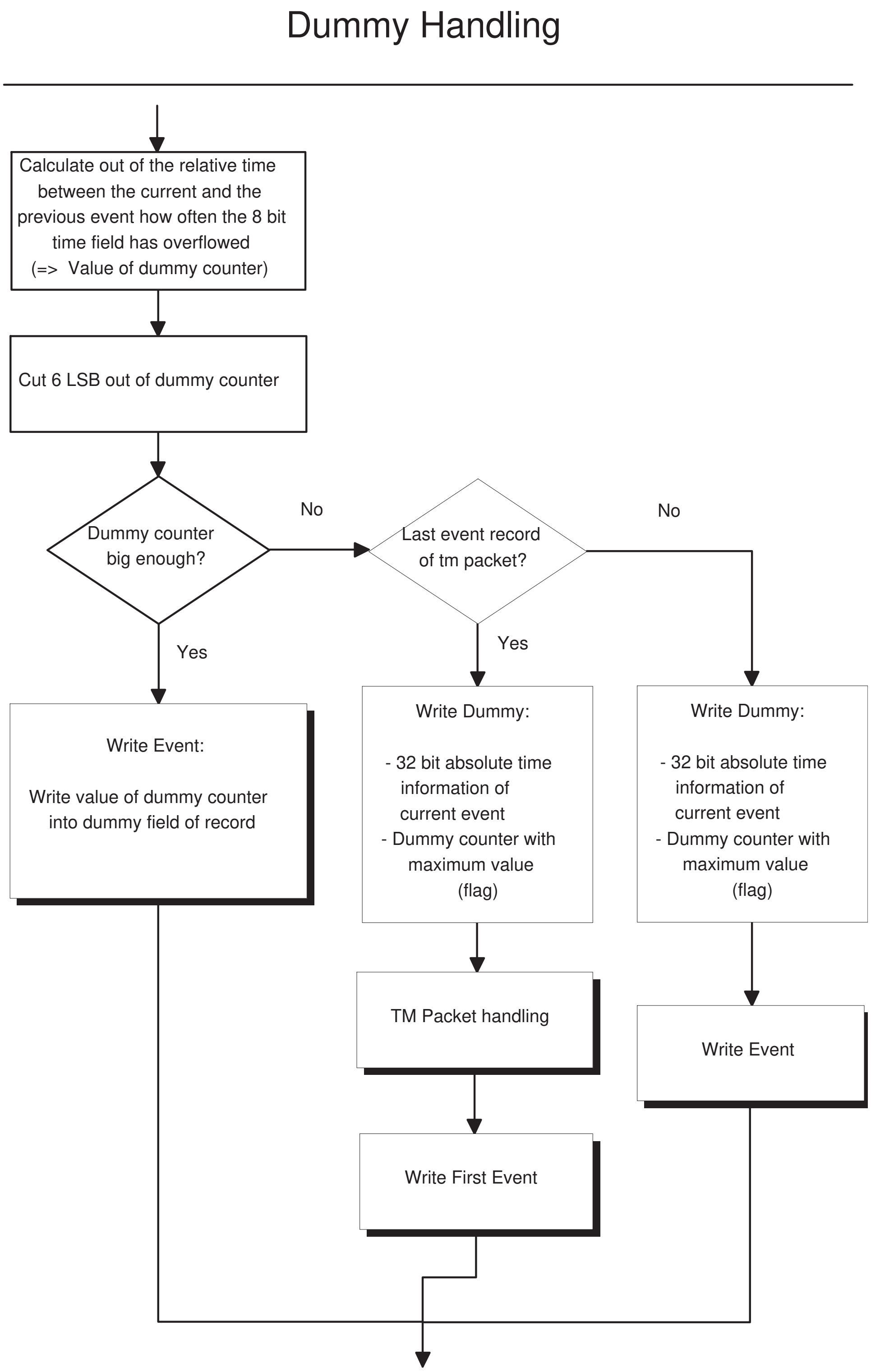

Abbildung 4.6: *Handhabung der Dummy–Events*

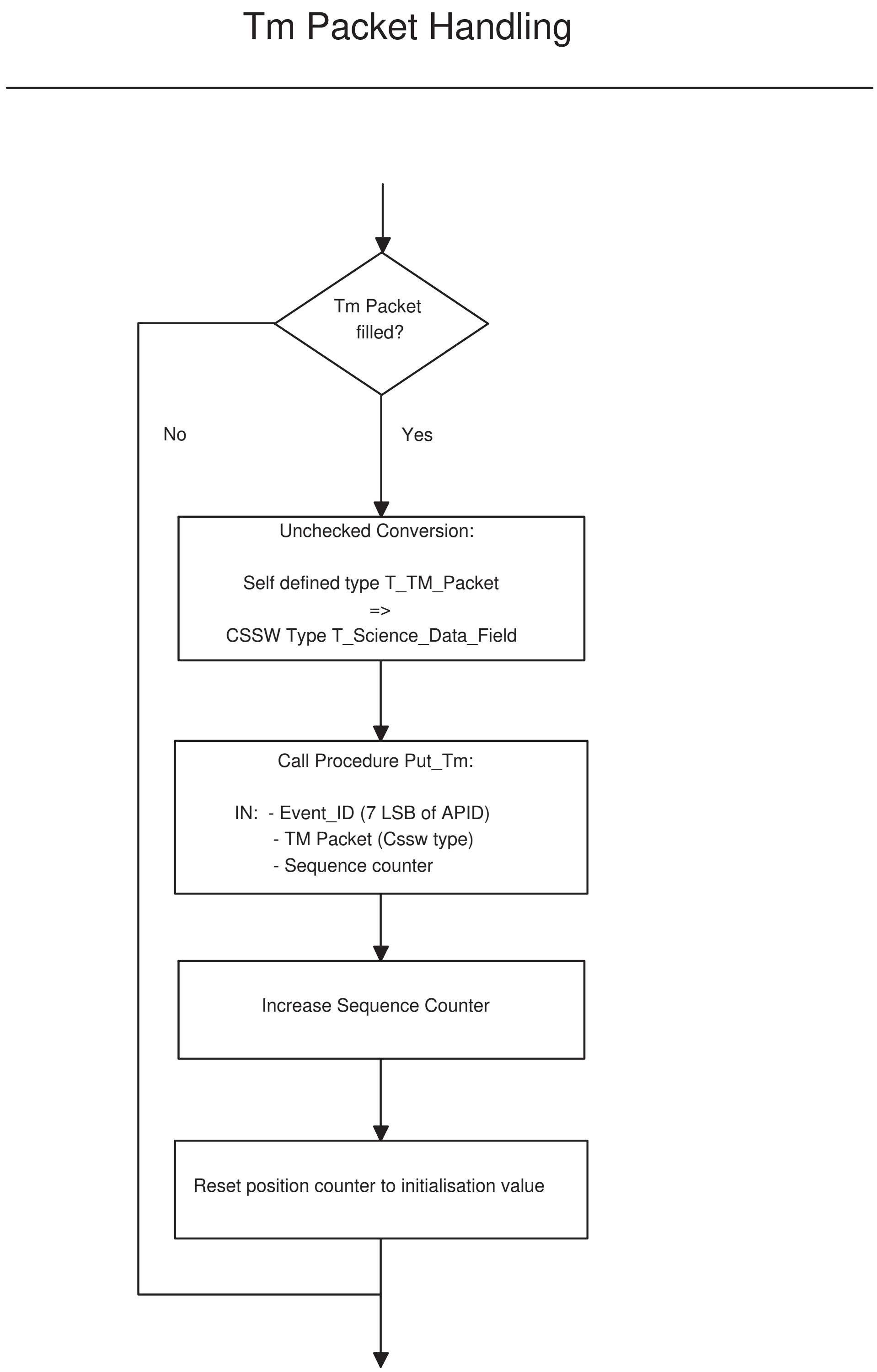

Abbildung 4.7: *Absenden eines Telemetriepakets*

4.2.10 Die restlichen Packroutinen im Überblick

In der Prozedur S1.PROCESS_EVENT der CdTe Photon by Photon Function wird ebenfalls eine Amplitudenkorrektur durchgeführt. Das Event wird weiterverarbeitet, falls die korrigierte Amplitude innerhalb eines einstellbaren Energiebereichs liegt. Auch hier wird, wie bei allen Packroutinen, eine Datenkomprimierung durch die Vermeidung von Spare Bit und eine rekonstruierbare Zeitreihe erreicht. Die 24 MSB der 32 Bit Zeitinformation werden in den Data Field Header kopiert, alle Events haben in ihrem Zeitfeld die 8 LSB stehen. Bei der CdTe Calibration Function S2 enthalten die Events nach der Verarbeitung kein Zeitfeld mehr. Lediglich die 24 MSB der Zeitinformation des ersten Events werden in den Data Field Header des Telemetriepakets geschrieben. Alle Ereignisse werden ohne Vorbedingung abgearbeitet. Auch die Prozedur S4.PROCESS_EVENT der CsI Photon by Photon übernimmt alle Events ohne Vorbedingung, die Handhabung der Zeitinformation ist äquivalent zur CdTe Photon by Photon Funktion. Der HEPI generiert die Histogramme der Spectral Timing Histogram Function S8 in regelmäßigen Zeitabständen gemäß seiner Einstellung. Zwei aufeinanderfolgende Histogramme haben also einen definierten Zeitabstand. Deshalb kann man anhand der Zeit des ersten Histogramms innerhalb des Telemetriepakets allen weiteren die korrekte Zeit zuordnen, solange sichergestellt ist, daß keines der Histogramme verloren ging. Die Prozedur S8.PROCESS_EVENT kontrolliert daher anhand der im HEPI eingestellten Integrationszeit, ob das zu bearbeitende Histogramm den korrekten Zeitabstand zum vorherigen hat. Sollte das nicht der Fall sein, wird das aktuelle Telemetriepaket mit Hex(FFFF) definiert aufgefüllt und abgeschickt. Das Histogramm wird in ein neues Paket geschrieben. Ähnlich arbeiten auch die anderen Histogrammfunktionen S5, S6 und S7. Sie prüfen, ob die HEPI–Speicheradresse der zu komprimierenden Histogrammblöcke wirklich aufeinanderfolgend sind. Ansonsten wird das aktuelle Paket definiert aufgefüllt, abgeschickt und ein neues begonnen, wobei die HEPI–Speicheradresse der ersten 8 Histogrammzelle in den Data Field Header geschrieben wird. Da bei einem ausgelesenen 512 word Histogrammblock nur die Anfangsadresse vom HEPI mitgegeben wird, müssen die Packprozeduren beim Verpacken der Histogrammzellen deren Adressen bestimmen.

4.3 Ereignisbehandlung

In diesem Abschnitt soll anhand von drei Beispielen erläutert werden, welche Möglichkeiten in der IASW vorhanden sind, um bestimmte Situationen zu lösen und wie die beschriebenen Module dabei zusammenwirken. Betrachtet werden zunächst die beiden Extremsituationen, die in der Echtzeit–Datenverarbeitung auftreten können, nämlich eine sehr niedrige bzw. eine sehr hohe Datenrate von den Detektoren. Der dritte Fall behandelt die Möglichkeit, bestimmte Arten von Telemetriepaketen der sofortigen Analyse durch das *Mission Operation Center* in Darmstadt zuzuführen.

4.3.1 Im HEPI liegen keine Daten zum Abholen bereit.

Angenommen, im Scientific Mode wären aus irgendwelchen Gründen keine Daten mehr vom HEPI bereitgestellt. Nach der Abarbeitung eines Buffers innerhalb der Mode–Schleife in der Prozedur SCIENCE_MODE.START wird durch Aufruf der Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE nach einem Buffer mit neuen Events gefragt. Diese prüft, ob der zweite Buffer bereits wieder mit Ereignissen vom HEPI gefüllt ist. Doch dies ist nicht der Fall und die Funktion ruft die Procedure HBR-A_INTERFACE.FETCH_DATA auf, deren einzige Aufgabe darin besteht, ein Softwaresignal an den sich im Wartestatus befindenden HBR-A Task auszulösen. Da der HBR-A Task eine höhere Priorität besitzt als der Main Task, in dem der Scientific Mode arbeitet, kann dieser aktiv werden und die Abarbeitung der Prozedur HBR-A_INTERFACE.INITIALISE fortsetzen, was mit dem Aufruf der Prozedur HBR-A_INTERFACE.FILL_BUFFER geschieht. Diese fragt, bevor sie die HSL zur Datenübertragung anmeldet, mit Hilfe der Funktion HEPI.DATA_AVAILABLE zuerst die HBR-A Request Leitung ab, die anzeigt, ob sich im Ausgangs–FIFO der HEPI genügend Ereignisse zum Auslesen befinden. Falls ja, werden die HBR-A Leitung blockiert und die Daten in einen Buffer geschrieben. Wenn nicht, beendet sich die Procedure HBR-A_INTERFACE.FILL_BUFFER und der HBR-A Task geht wieder in den Wartestatus zurück. Gleichgültig, ob der HBR-A Task Daten holen konnte oder nicht, gibt die Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE ein „False“ an die Prozedur SCIENCE_MODE.START zurück. Diese fragt nun eine logische Variable `HEPI_Resync_Request` ab, ob der HEPI neu synchonisiert werden muß. Die Entscheidung darüber trifft der Housekeeping Task, der mit Hilfe des Event–Zählers bestimmen kann, wieviele Events in den letzten acht Sekunden verarbeitet wurden. Ist die Datenrate geringer als ein via Telekommando definierbarer Wert, stellt der Housekeeping Task die Variable `HEPI_Resync_Request` so um, daß das Modul SCIENCE_MODE die Resynchronisierung des HEPI beginnt. Dieser Resynchronisierungsautomatismus ist der einzige Unterschied zum Ablauf des Diagnostic Mode in dieser Situation, wenn keine Daten erreichbar sind. Er kann vor dem Start des Scientific Mode im Stand by Mode über ein Telekommando ausgeschaltet werden. Die Mode–Schleife ist nun durchlaufen und startet wieder mit dem Aufruf der Funktion HBR-A_INTERFACE.BUFFER_AVAILABLE.

4.3.2 Was geschieht, wenn der CSSW ICB Telemetrie Subbuffer gefüllt ist?

Wenn die Datenrate so hoch ist, daß mehr Telemetriepakete erzeugt werden als abgeschickt werden können, ist der ICB Telemetrie Subbuffer der CSSW ständig maximal gefüllt und erzeugte Pakete gehen verloren. Damit bei so hohen Datenraten keine Pakete verloren gehen, greift das Modul TM_INTERFACE auch aktiv in die Erzeugung von Telemetriepaketen ein. Dafür werden in einer Tabelle namens `TM_Table` jeder Art von Telemetriepaket[13] zwei Werte zugeordnet, einen `High_Buffer_State_Value` und einen `Low_Buffer_State_Value`, die beide bestimmte Füllstande des ICB Telemetrie Subbuffers repräsentieren sollen. Bevor im Diagnostic Mode oder im Scientific Mode ein Event durch eine Packroutine der Diagnostic/Scientific Function verarbeitet wird, fragt der Mode erst einmal nach, ob es denn auch verarbeitet werden soll oder ob es lediglich gezählt wird. Dazu ruft er die Funktion TM_INTERFACE.MUST_I_PROCESS auf. Diese prüft nach, ob der aktuelle Füllstand des ICB Buffers höher ist als der `High_Buffer_State_Value` der entsprechenden Eventsorte, was der Art des Telemetriepakets entspricht. Ist dies der Fall, so gibt die Funktion ein „Nein“ zurück und das Event wird nur gezählt, aber nicht verarbeitet. Damit ist die Erzeugung dieser Art Telemetriepaket gestoppt. Durch eine geschickte Wahl der HBSV und LBSV kann also der Verarbeitung bestimmter Eventarten eine höhere Priorität zugeordnet werden. Da bei gestoppten Eventarten keine Telemetriepakete mehr generiert werden, steht den nicht gestoppten Eventarten eine höhere Telemetrierate zur Verfügung. So kann die Verarbeitung und damit die Erzeugung von Telemetriepaketen von einer Eventart nach der anderen eingestellt werden, bis die Telemetrierate

[13] Jedes Event wird in eine eigene Telemetriepaketsorte verpackt, die durch ihre APIDs unterschieden werden. Außer diesen wissenschaftlichen und diagnostischen Paketen gibt es aber auch noch Pakete, denen keine Events zugeordnet werden können, wie z.B. die Housekeeping Pakete oder die Report Parameter Pakete. Diese müssen natürlich ebenso mit einbezogen werden, deshalb bezieht man sich in der Tabelle auf die Paketart und damit letztendlich auf die APID der Pakete.

für die verbleibenden Sorten ausreicht. Schlimmstenfalls muß die gesamte wissenschaftliche Datenverarbeitung kurz unterbrochen werden. Bei korrekter Einstellung der IASW kann also der ICB Telemetrie Subbuffer der CSSW, wie hoch die Datenrate auch sein mag, nie vollständig gefüllt sein. Da alle Arten der innerhalb der IASW generierten Telemetriepaketen von dieser Regelung betroffen sind, müssen die HBSV- und LBSV–Werte so gesetzt werden, daß kein Housekeeping Paket verloren gehen kann.

4.3.3 Die diagnostische Telemetrie soll auf VC-0 Pakete umgestellt werden

Die Telemetriepakete der diagnostische Funktionen werden wie diejenigen der wissenschaftlichen mit einem in der APID gesetzten VC Bit (VC-1) vom Satelliten zur Erde gesendet. Das MOC in Darmstadt wertet Telemetriepakete, deren VC Bit nicht gesetzt ist, sofort aus. Die VC-1 Pakete werden zum ISDC weitergeleitet. Standardmäßig werden lediglich die Housekeeping Pakete H1, H2, H3 und H4 als VC-0 Pakete direkt vom MOC kontrolliert. Sollte sich jedoch während der INTEGRAL–Mission die Notwendigkeit ergeben, auch die diagnostischen Daten als VC-0 Pakete der sofortigen Analyse zuzuführen, muß die IASW in der Lage sein, die APIDs der Funktionen zu ändern. Im Modul TM_INTERFACE, über welches alle innerhalb der IASW generierten Telemetriepakete an die CSSW übergeben werden, befindet sich mit der bereits in Kapitel 4.2.5 vorgestellten Tabelle `TM_Table` die entscheidende Variable. In dieser wird jeder Art von Telemetriepaket nicht nur ein HBSV und LBSV zugeordnet, sondern auch die sieben LSB der APID. Das oberste dieser sieben Bit ist das VC Bit, danach folgen sechs Bit, die das Identifizieren des Pakettyps ermöglichen. Durch Ändern dieser Tabelle ist außer dem Setzen des VC Bit eines Telemetriepakets auch die Zuordnung einer neuen APID realisierbar.

Kapitel 5

Datenformate und Laufzeiten

Ein Telemetriepaket besteht aus einem Datenvolumen, das sich in einen aus drei word bestehenden Identifikationsteil am Beginn des Pakets, gefolgt von einem großen Datenteil, untergliedern läßt. Das erste word eines Telemetriepakets enthält dessen *Application Process Identification*, kurz APID. Die lediglich die 11 LSB belegende APID enthält alle Informationen, um das Telemetriepaket eindeutig einem bestimmten Prozess eines der Instrumente oder Steuerungssysteme zuzuordnen. Um das Instrument zu erkennen, von welchem das Telemetriepaket generiert wurde, genügen die vier MSB der APID, die *Source ID*. Für die nominelle DPE des IBIS wurde die Source ID 1010 festgelegt, die DPE des Redundanzsystems erhielt die Bitfolge 1011. Im Anschluß an die vier Bits der Source ID befindet sich das *Vacation Bit* (VC). Anhand dieses Bit entscheidet das MOC darüber, ob das Telemetriepaket direkt beim MOC ausgewertet (VC-0) oder unkontrolliert an das ISDC weitergeleitet wird (VC-1). Standardmäßig werden von IBIS nur die Housekeeping Pakete mit dem VC-0 Bit zur Erde übertragen. Sie enthalten alle wichtigen Statusinformationen von Detektoren, HEPI, IASW und CSSW. Ihre sofortige Auswertung ist notwendig, um schnell auf das Erreichen kritischer Parameter oder gar Fehlfunktionen reagieren zu können. Die verbleibenden sechs LSB der APID dienen der eindeutigen Zuordnung der Pakete an einen Telemetrie erzeugenden Prozess innerhalb einer DPE eines Instuments und werden als *Format ID* bezeichnet. IBIS verwendet für die wissenschaftlichen Daten von den sechs Bit der Format ID die vier MSB zur Festlegung eines Pakettyps und die beiden LSB zur weiteren Unterteilung des Typs in Subtypen, wo dies wegen der Eindeutigkeit notwendig ist. Als Typ wird die Numerierung der wissenschaftlichen Funktionen verwendet. Die CdTe Photon by Photon Funktion mit der Bezeichnung S1 generiert folglich Telemetriepakete vom Typ 0001, die CdTe Calibration Funktion S2 Pakete vom Typ 0010 und so weiter. Bei den Funktionen, die Ereignisse des CsI–Detektors verarbeiten, muß zur eindeutigen Kennzeichnung von Einfach- bzw. Mehrfachereignissen ein Subtyp vergeben werden. Da von ISGRI keine Mehrfachevents an die DPE weitergeleitet werden und deswegen der nicht benötigte und auf 00 gesetzte Subtyp Einzelereignisse kennzeichnet, wird auch den Einzelereignissen des PICsIT–Detektors der Subtyp 00 zugeordnet. Die Mehrfachevents erhalten den Subtyp 01. Diese Regel gilt ebenso für die diagnostischen Daten. Als Typ erhalten alle Pakete der drei diagnostischen Funktionen die Bitfolge 1101, was im Hexadezimalsystem den Wert Hex(D) ergibt. Durch den Subtyp, der identisch ist mit der Nummer der entsprechenden Funktion, werden sie eindeutig gekennzeichnet.

Das zweite word des Telemetriepakets ist der *Source Sequence Counter*. Das ist ein fortlaufender Telemetriepaketzähler. Jeder in der IASW generierte Pakettyp besitzt einen eigenen Sequence Counter, der nie von der Software zurückgesetzt wird. Ist der Wertebereich des Zählers erreicht, startet er wieder bei Null. Alle Sequence Counter der wissenschaftlichen und diagnostischen Funktionen werden in den H2–Housekeeping Paketen zur Kontrollmöglichkeit zusammengestellt.

Den Identifikationsteil abschließend steht im dritten word jedes Telemetriepakets die Länge seines Datenteils minus eins in Byte. Alle Telemetriepakete von IBIS haben in diesem Feld die Länge Hex(1B1) oder dezimal 433 stehen [16].

Der bei IBIS 217 word umfassende Datenteil gliedert sich nochmals in einen *Data Field Header* und in den echten Datenblock. Ursprünglich sollten im Data Field Header der Pakettyp und Subtyp sowie ein Ausschnitt aus der Uhr des Satelliten, der *On Board Time*, stehen. Wie im folgenden gezeigt wird, wurde das vorgegebene Format zugunsten der Datenreduktion in den meisten Fällen leicht abgewandelt.

Das Format der eigentlichen Datenbereiche wurde individuell an die wissenschaftlichen Funktionen angepasst. Dadurch ist die Anzahl der in den Telemetriepaketen verpackten Events von Funktion zu Funktion unterschiedlich. Dabei wurde in erster Linie darauf geachtet, daß möglichst wenige Bits ohne Informationsgehalt, sogenannte Spare Bit, in die Telemetriepakete übernommen werden. Wo dies nicht realisiert werden konnte, wurde versucht, den entsprechenden Bits eine Funktionalität zuzuordnen. Zusammengehörige Informationsfelder wurden über word–Grenzen hinweg geschrieben und die word–orientierte Speicheradressierung umgangen. Die Programmiersprache ADA stellt dafür die geeigneten Werkzeuge zur Verfügung und wird sehr gut vom Compiler sowie vom Assembler Code des Prozessors unterstützt, so daß es zeitlich nahezu keinen Unterschied macht, ob eine aufeinanderfolgende Reihe von wenigen Bits in einem einzigen word (= eine Speicheradresse) oder über eine word–Grenze hinweg (= zwei Speicheradressen) gesetzt werden. Im folgenden werden die verschiedenen Formate der wissenschaftlichen und diagnostischen Datenpakete sowie die Laufzeiten der einzelnen Packroutinen tabellarisch aufgeführt. Die Laufzeiten wurden beim Ablaufen der IASW innerhalb eines Software–Simulators ermittelt. Der Simulator zählt die für die Ausführung der Assemblerbefehle benötigten Schritte und berechnet über die Taktfrequenz des Zielprozessors die Laufzeit des Programms. Mit Hilfe eines Debuggers können in der IASW Breakpoints gesetzt und damit die Abarbeitungsgeschwindigkeit von Programmteilen abgestoppt werden.

5.1 Scientific Functions

5.1.1 Die CdTe Photon by Photon–Funktion S1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								–	Risetime (7 Bit)						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB	MSB der Eventzeit (32 Bit)														
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
LSB der Eventzeit (32 Bit)														LSB	
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (5 Bit)					Amplitude (11 Bit)										
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–	Y_Pos (7 Bit)							–	Z_Pos (7 Bit)						

Abbildung 5.1: Das S1–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1348 = Hex(8D44)
Data Field Header (2 word)
43 TM Doppel-Event Records der Länge von 2 x 40 Bit
keine Spare words am Paketende, mit HEX(FFFF) initialisiert.
Zusammen: 217 words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp = HEX(10)								MSB		MSB der Eventzeit (24 Bit)					
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB der Eventzeit des ersten Events (24 Bit)														LSB	

Abbildung 5.2: Das Data Field Header Format der S1–Funktion.

Um eine möglichst hohe Komprimierungsrate zu erhalten, wurde bei den CdTe Photon by Photon Ereignissen das word–orientierte Eventformat durchbrochen. Um dennoch die word–ausgerichtete Adressierung des Speichers zu beachten, wurde ein Format gewählt, das zwei 40 Bit große und damit nicht word–orientierte Eventformate zusammenfaßt zu einem Doppel–Event Record, der wiederum word–ausgerichtet im Speicher abgelegt werden kann.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amplitude1 (11 Bit)											Risetime1 (7 Bit)				
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LSB		Y_Pos1 (7 Bit)							Z_Pos1 (7 Bit)						
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
LSB der Zeit1 (8 Bit)								LSB der Zeit2 (8 Bit)							
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Amplitude2 (11 Bit)											Risetime2 (7 Bit)				
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
LSB		Y_Pos2 (7 Bit)							Z_Pos2 (7Bit)						

Abbildung 5.3: Das TM Event Record Format S1–Funktion.

Das Dummy Handling:
Wenn die Zeitdifferenz zwischen dem vorherigen und dem aktuell zu verarbeitenden Event größer ist als die Zeitspanne, die maximal durch das acht Bit große Zeitfeld des Event Records zu erfassen ist, geht die eindeutige Zeitreihe und damit die Reproduzierbarkeit der Absolutzeiten aller nachfolgenden Events des Telemetriepakets verloren. Deshalb wird ein 40 Bit großes Dummy Event vor dem eigentlichen Ereignis in das Paket geschrieben. Das Dummy Event ist strukturell identisch mit dem Event Record, aber alle Felder mit Ausnahme des Zeitfeldes sind auf Null gesetzt. Im Zeitfeld befindet sich ein Zähler, der angibt, wie oft das acht Bit große Zeitfeld übergelaufen ist. Es entspricht damit indirekt den nächsthöheren Bits der Zeitinformation.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amplitude1 = 2047											Risetime1 = 0				
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LSB		Y_Pos1 = 127							Z_Pos1 = 0						
32	33	34	35	36	37	38	39								
Dummy Counter															

								40	41	42	43	44	45	46	47
								Dummy Counter							
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Amplitude2 = 2047											Risetime2 = 0				
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
LSB		Y_Pos2 = 127							Z_Pos2 = 0						

Abbildung 5.4: Der erste Dummy der S1–Funktion.

Das Zeitfeld des TM Paket Records erlaubt eine Zeitdifferenz zwischen zwei aufeinanderfolgenden CdTe Photon by Photon Ereignissen von etwa 15 Millisekunden. Bei einer erwarteten Mindestensdatenrate von 350 liegt der durchschnittliche Zeitabstand zweier Events bei etwa drei Millisekunden. Das Notwendigkeit zum Einfügen von Dummy–Events ist daher sehr unwahrscheinlich.

Der höchste erlaubte Wert für diesen Zähler ist Hex(FF-1). Der Maximalwert Hex(FF) wird als Flag eingesetzt, daß der Zähler selbst übergelaufen ist. In diesem Fall werden zwei Dummy Events in das Telemetriepaket geschrieben. Der erste Dummy enthält den Zähler mit dem Maximalwert Hex (FF). Der zweite Dummy schreibt die vollständige Zeitinformation von 32 Bit des aktuell zu bearbeitenden Ereignisses in die Record–Partitionen von `Amplitude`, `Risetime`, `Y--Position` und `Z--Position`. Das Zeitfeld des Records ist beim zweiten Dummy auf Null gesetzt.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB		absolute event time (32 Bit)													
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
absolute event time (32 Bit)														LSB	
32	33	34	35	36	37	38	39								
Dummy Counter = 255															

								40	41	42	43	44	45	46	47
								Dummy Counter = 255							
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
MSB		absolute event time (32 Bit)													
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
absolute event time (32 Bit)														LSB	

Abbildung 5.5: Der zweite Dummy der S1–Funktion.

Falls jemals dieser zweite Dummy in einem Telemetriepaket gefunden werden sollte, betrüge der Zeitabstand zweier aufeinanderfolgenden CdTe Ereignisse nahezu vier Sekunden und wäre eher als Anzeichen dafür anzusehen, daß irgendwo in der Transportkette des Datenstroms eine Fehlfunktion vorliegt oder gar Detektormodule von ISGRI ausgefallen sind.

Die Verarbeitung eines CdTe Photon by Photon benötigt etwa 325 μs. Den größten Anteil davon beansprucht mit etwa 234 μs die Amplitudenkorrektur. Allein das Auslesen der Korrekturtabelle dauert 154 μs, da wegen zweier Address State–Wechsel die Abbildung des logischen Adressraums auf den physikalischen Speicher einmal auf den neuen AS und anschließend wieder zurück auf den vorherigen AS geändert werden muß. Das reine Verpacken der Ereignisse in Telemetriepakete, also die Packroutine selbst, benötigt für ein normales Event etwa 91 μs. Müssen Dummy–Events in das Paket geschrieben werden, verlängert sich die Arbeitszeit der Routine bei einem Dummy auf ca. 126 μs, bei zwei Dummies auf 146 μs. Wie oben ausgeführt, braucht bei einer Abschätzung der Leistungsfähigkeit der IASW lediglich der Fall ohne Dummy–Events berücksichtigt zu werden, da nur dieser zeitlich kritisch werden kann.

5.1.2 Die CdTe Calibration–Funktion S2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								–	Risetime (7 Bit)						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB	MSB der Eventzeit (32 Bit)														
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
LSB der Eventzeit (32 Bit)															LSB
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (5 Bit)					Amplitude (11 Bit)										
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–	Y_Pos (7 Bit)							–	Z_Pos (7Bit)						

Abbildung 5.6: Das S2–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1352 = Hex(8D48)
Data Field Header (2 word)
107 TM Event Records der Länge von 32 Bit
1 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Zusammen: 217 words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp = HEX(20)								MSB	MSB der Eventzeit (24 Bit)						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB der Eventzeit des ersten Events (24 Bit)															LSB

Abbildung 5.7: Das Data Field Header Format der S2–Funktion.

Die Empfindlichkeit der CdTe–Pixel kann sich mit der Zeit verändern. Eine Kalibrationsquelle an Bord des Satelliten ermöglicht durch die Aussendung definierter Ereignisse die ständige Eichung sowohl von ISGRI als auch von PICsIT. Dabei ist es nicht notwendig, die genaue Ereigniszeit jedes Kalibrationsevents zu kennen. Die in den Data Field Header kopierte 24 MSB Zeitinformation des ersten der 107 Events des Telemetriepakets ist vollkommen ausreichend. Deshalb wird bei der CdTe Calibration Funktion die Zeitinformation der einzelnen Events nicht mitgenommen.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amplitude (11 Bit)											Risetime (7 Bit)				
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LSB		Y_Pos (7 Bit)							Z_Pos (7 Bit)						

Abbildung 5.8: Das TM Event Record Format der S2–Funktion.

Das Dummy Handling:
Die CdTe Calibration benötigt keine Dummies, da die Zeitinformation der einzelnen Events nicht berücksichtigt wird. Lediglich die 24 MSB der 32 Bit Zeitinformation des ersten der ins Telemetriepaket geschriebenen Ereignisses werden in den Data Field Header kopiert. Die Laufzeit der Packroutine liegt bei etwa 60 μs.

5.1.3 Die Compton Photon by Photon–Funktion S3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								–	CdTe_Risetime (7 Bit)						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB	MSB der Eventzeit (32 Bit)														
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
LSB der Eventzeit (32 Bit)															LSB
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (5 Bit)					CdTe_Amplitude (11 Bit)										
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–	CdTe_Y_Pos (6 Bit)							–	CdTe_Z_Pos (6 Bit)						
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Spare (6 Bit)						CsI_Amplitude (10 Bit)									
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
—		CsI_Y_Pos (6 Bit)						—		CsI_Z_Pos (6 Bit)					
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Spare (16 Bit)															
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Spare (16 Bit)															
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
Spare (16 Bit)															

Abbildung 5.9: Das S3–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1356 = Hex(8D4C) (Single) / 1357 = Hex(8D4D) (Multiple)
Data Field Header (2 word)
53 TM Event Records der Länge von 64 Bit
3 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Zusammen: 217 words

Bei den Compton–Events unterscheidet der HEPI, ob das gemessene Ereignis im PICsIT–Detektor ein Einzelereignis war oder ob das Ereignis von zwei oder mehreren CsI–Pixeln registriert wurde. Damit diese Information beim Zusammensetzten des Compton–Events nicht verloren geht, werden diese ebenfalls als Einzel- und Mehrfachereignisse markiert. Die Einzel- und Mehrfachereignisse werden typenrein in unterschiedliche Telemetriepakete verpackt. Die vom HEPI ausgegebenen Compton–Events haben dasselbe Format, weshalb auch der Aufbau ihrer Telemetriepakete identisch ist. Die beiden Ereignissorten werden vom HEPI durch verschiedene Einträge in den Typfeldern markiert. Die IASW unterscheidet die entsprechenden Telemetriepakete ebenfalls durch das Typ–Subtyp–Feld innerhalb des Data Field Headers. Der Typ ist für beide Ereignisse gleich, der Subtyp verschieden, Hex(30) für Einzel- und Hex(31) für Mehrfachevents. Da Typ und Subtyp in die APID der Telemetriepakete eingehen, können beide Paketarten eindeutig identifiziert werden.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">Typ–Subtyp = Hex(30/31)</td><td>0</td><td colspan="7">MSB MSB der Eventzeit (23 Bit)</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">MSB der Eventzeit des ersten Events (23 Bit) LSB</td></tr>
</table>

Abbildung 5.10: Das Data Field Header Format der S3–Funktion.

Die Compton–Events sind die einzigen Datensätze, bei denen die Amplituden der Ereignisse nicht vollständig in die Telemetriepakete übernommenen werden. Von beiden Amplituden, sowohl von CdTe- als auch von CsI–Events, werden lediglich die 8 MSB der Amplitudeninformation in die Pakete verpackt.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">CsI_Amplitude (8 Bit)</td><td colspan="8">CdTe_Amplitude (8 Bit)</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="7">CdTe_Risetime (7 Bit)</td><td colspan="9">LSB der Zeit (9 Bit)</td></tr>
<tr><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td></tr>
<tr><td colspan="6">CsI_Y_Pos (6Bit)</td><td colspan="6">CsI_Z_Pos (6Bit)</td><td colspan="4">CdTe_Y_Pos</td></tr>
<tr><td>48</td><td>49</td><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td><td>61</td><td>62</td><td>63</td></tr>
<tr><td colspan="3">(7Bit)LSB</td><td colspan="7">CdTe_Z_Pos (7 Bit)</td><td colspan="6">Dummy (6 Bit)</td></tr>
</table>

Abbildung 5.11: Das TM Event Record Format der S3–Funktion.

Das Dummy Handling:
Wenn die Zeitdifferenz zwischen dem vorherigen und dem aktuell zu verarbeitenden Event größer ist als die Zeitspanne, die maximal durch das 9 Bit Zeitfeld des Event Records zu erfassen ist, geht die eindeutige Zeitreihe und damit die Reproduzierbarkeit der Absolutzeiten aller nachfolgenden Events des Telemetriepakets verloren. Der Event Record besitzt selbst bereits einen sechs Bit großen Zähler, der angibt, wie oft das neun Bit große Zeitfeld übergelaufen ist. Er entspricht damit indirekt den nächsthöheren Bits der Zeitinformation. Der höchste erlaubte Wert für diesen Zähler ist Hex(3F-1). Der Maximalwert Hex(3F) wird als Flag eingesetzt, daß der Zähler selbst übergelaufen ist. In diesem Fall wird vor dem eigentlichen Ereignis ein Dummy Event in das Telemetriepaket geschrieben, das die vollständige Zeitinformation von 32 Bit des aktuell zu bearbeitenden Ereignisses in den ersten 32 Bit seines Records enthält. Der record–interne Zähler des Dummies ist auf den Maximalwert von Hex(3F) gesetzt. Der Zähler des hinter dem Dummy

ins Telemetriepaket eingetragenen Ereignisses ist Null.

Das Zeitfeld des TM Paket Records erlaubt eine Zeitdifferenz zwischen zwei aufeinanderfolgenden Compton Photon by Photon Ereignissen von etwa 30 Millisekunden. Da der TM Paket Record der Compton–Events bereits einen Überlaufzähler enthält, ist die Notwendigkeit zum Einfügen eines Dummy–Events erst bei einem Zeitabstand von ca. 2 Sekunden gegeben und daher sehr unwahrscheinlich.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB							Eventzeit (32 Bit)								
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
					Eventzeit (32 Bit)									LSB	
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Spare = 0															
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare = 0										Flag = Hex(3F)					

Abbildung 5.12: Der Dummy der S3–Funktion.

Die Verarbeitungsdauer eines Compton Photon by Photon Event setzt sich aus drei Anteilen zusammen. Wie bei der S1–Funktion beansprucht die Amplitudenkorrektur des CdTe–Ereignisses etwa 234 μs. Die Berechnung, ob der Einfallswinkel des Ursprungsphotons mit den geometrischen Bedingungen des IBIS–Tubus übereinstimmt, benötigt in etwa weitere 330 μs. Das reine Verpacken der Ereignisse in Telemetriepakete dauert für ein normales Event etwa 99 μs. Muß das Dummy–Feld mit ausgefüllt werden, verlängert sich die Arbeitszeit auf ca. 115 μs. Ist ein zusätzliches Dummy–Event notwendig, steigt die Laufzeit auf 141 μs.

5.1.4 Die CsI Photon by Photon–Funktion S4

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB						MSB der Eventzeit (32 Bit)									
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
				LSB der Eventzeit (32 Bit)										LSB	
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (6 Bit)						Amplitude (10 Bit)									
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–		Y_Pos (6 Bit)						–		Z_Pos (6 Bit)					

Abbildung 5.13: Das S4–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1360 = Hex(8D50) (Single) / 1361 = Hex(8D51) (Multiple)
Data Field Header (2 word)
107 TM Event Records der Länge von 32 Bit
1 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Zusammen: 217 words

Die Einzel- und Mehrfachereignisse des PICsIT–Detektors, vom HEPI durch unterschiedliche Typfelder gekennzeichnet, werden von der IASW in verschiedene Telemetriepakete geschrieben. Wie bei den Compton–Events sind die Formate für beide Eventtypen die selben. Der Typ–Subtyp im Data Field Header der Telemetriepakete für die Einzelereignisse ist Hex(40), derjenige der Mehrfachevents Hex(41).

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">Typ–Subtyp Hex(40/41)</td><td colspan="8">MSB MSB der Eventzeit</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">MSB der Eventzeit des ersten Events (24 Bit) LSB</td></tr>
</table>

Abbildung 5.14: Das Data Field Header Format der S4–Funktion.

Die CsI Photon by Photon Funktion hat keinerlei Vorbedingungen zu prüfen oder, wie die entsprechende CdTe–Funktion, eine Amplitudenkorrektur durchzuführen. All diese Aufgaben werden der IASW vom HEPI abgenommen. Im Eventformat befinden sich zwei Bit ohne wissenschaftliche Information. Sie werden dazu verwendet, die Dummy–Events eindeutig zu kennzeichnen.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="10">Amplitude (10 Bit)</td><td colspan="6">Y_Pos (6 Bit)</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="2">D_F</td><td colspan="6">Z_Pos (6 Bit)</td><td colspan="8">LSB der Zeit (8 Bit)</td></tr>
</table>

Abbildung 5.15: Das TM Event Record Format der S4–Funktion.

Das Dummy Handling:
Wenn die Zeitdifferenz zwischen dem vorherigen und dem aktuell zu verarbeitenden Event größer ist als die Zeitspanne, die maximal durch das acht Bit große Zeitfeld des Event Records zu erfassen ist, geht die eindeutige Zeitreihe und damit die Reproduzierbarkeit der Absolutzeiten aller nachfolgenden Events des Telemetriepakets verloren. Deshalb wird ein 32 Bit großes Dummy Event vor dem eigentlichen Ereignis in das Paket geschrieben. Das Dummy Event ist strukturell identisch mit dem Event Record, aber alle Felder mit Ausnahme des Dummy–Flag–Feldes sind auf Null gesetzt. Das Dummy–Flag–Feld, welches bei normalen Events immer auf Null gesetzt ist, zeigt den Dummy durch den Wert 3 an. Im Zeitfeld befindet sich ein Zähler, der angibt, wie oft das 8 Bit Zeitfeld übergelaufen ist. Es entspricht damit indirekt den nächst höheren Bits der Zeitinformation.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="10">Amplitude = 0</td><td colspan="6">Y_Pos = 0</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="2">D_F = 3</td><td colspan="6">Z_Pos = 0</td><td colspan="8">Dummy_Counter</td></tr>
</table>

Abbildung 5.16: Der erste Dummy der S4–Funktion.

Der höchste erlaubte Wert für diesen Zähler ist Hex(FF-1). Der Maximalwert Hex(FF) wird als Flag eingesetzt, daß der Zähler selbst übergelaufen ist. In diesem Fall werden zwei Dummy Events in das Telemetriepaket geschrieben. Der erste Dummy enthält den Zähler mit dem Maximalwert Hex (FF). Der zweite Dummy schreibt die vollständige Zeitinformation von 32 Bit des aktuell zu bearbeitenden Ereignisses in das Telemetriepaket. Das aktuelle Ereignis steht hinter den beiden Dummies.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB								Eventzeit (32 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
							Eventzeit (32 Bit)							LSB	

Abbildung 5.17: Der zweite Dummy der S4–Funktion.

Für die Zeitdifferenz zwischen zwei aufeinanderfolgenden CsI–Events gilt dasselbe wie für die CdTe Photon by Photon Ereignisse. Die Datenrate des PICsIT wird meistens deutlich höher sein als diejenige von ISGRI. Die Abarbeitung eines normalen CsI Photon by Photon Ereignisses beansprucht ca. 76 μs. Die Gesamtzeit beim Einschieben eines Dummy–Events vor dem eigentlichen Ereignis benötigt ca. 103 μs, bei der Festsetzung einer neuen Zeitbasis ca. 119 μs.

5.1.5 Die Spectral Timing Histogram–Funktion S8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB								MSB der Eventzeit (32 Bit)							
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
						LSB der Eventzeit (32 Bit)									LSB
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Histogrammzelle1 (8 Bit)								Histogrammzelle2 (8 Bit)							
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Histogrammzelle3 (8 Bit)								Histogrammzelle4 (8 Bit)							
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Histogrammzelle5 (8 Bit)								Histogrammzelle6 (8 Bit)							
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
Histogrammzelle7 (8 Bit)								Histogrammzelle8 (8 Bit)							
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Spare (16 Bit)															
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Spare (16 Bit)															
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
Spare (16 Bit)															

Abbildung 5.18: Das S8–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1376 = Hex(8D60).
Data Field Header (2 word).
860 Histogramm–Datenzellen der Länge von je 4 Bit.
Keine Spare words am Paketende.
Zusammen: 217 words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB						Time des ersten Events(32 Bit)									
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
				Time des ersten Events (32 Bit)										LSB	

Abbildung 5.19: Das Data Field Header Format der S8–Funktion.

Die Komprimierung des Histogramms:
Das vom HEPI ausgegebene Histogramm besteht aus acht Zellen zu je acht Bit. Die kürzeste einstellbare Integrationszeit der Spectral Timing Histogramme ist eine Millisekunde. Bei einer geschätzten maximalen Datenrate von 28 000 Ereignissen [22] pro Sekunde sind im Histogramm also etwa 28 Events zusammengefaßt. Der durchschnittliche Zählerstand, eine statistische Verteilung vorausgesetzt, liegt dann bei etwa vier Ereignissen in einer Histogrammzelle. Es ist demnach ausreichend, anstelle einer acht Bit großen eine auf vier Bit reduzierte Histogrammzelle zu verwenden. Diese reduzierten Zellen mit dem maximalen Wertebereich von 0 – 15 sollten selbst mit erheblichen statistischen Schwankungen zurechtkommen und damit eine Datenreduktion von 50 Prozent erreichen. Die IASW schreibt nur vollständige Histogramme in ein Telemetriepaket, die Zellen eines Histogramms werden nie auf zwei Telemetriepakete verteilt. Durch Abzählen der Zellen in einem Telemetriepaket können die zu einem Histogramm gehörenden Zellen sicher identifiziert werden. Sollte der verbleibende Platz in einem nahezu gefüllten Paket nicht mehr ausreichen, um ein komplettes Histogramm hinzuzufügen, wird das Paket definiert mit Zellen aufgefüllt, die Hex(F) enthalten.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Zelle1				Zelle2				Zelle3				Zelle4			
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Zelle5				Zelle6				Zelle7				Zelle8			

Abbildung 5.20: Das Format komprimierbarer Histogramm–Zellen.

Nicht komprimierbare Histogrammzellen:
Sollte eine der acht Bit großen Histogrammzellen nicht auf vier Bit reduzierbar sein, so müssen alle acht Bit der Ursprungszelle in das Telemetriepaket übernommen werden. Dies muß vorher jedoch durch ein Flag angezeigt werden, da ansonsten die Eindeutigkeit verloren geht. Als Flag nimmt man eine vier Bit große Zelle, die den Maximalwert Hex(F) enthält. Eine komprimierte Zelle darf deswegen nur einen Höchstwert von Hex(E) enthalten.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Zelle1				Flag Hex(F)				Zelle2							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Zelle3				Zelle4				Flag Hex(F)				MSB		Zelle5	
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Zelle5		LSB		Zelle6				Zelle7				Zelle8			

Abbildung 5.21: Das Format nichtkomprimierbarer Histogramm–Zellen.

Die regelmäßige Zeitreihe der Histogramme ist unterbrochen:
Entspricht der Zeitabstand zwischen zwei aufeinanderfolgenden Histogrammen nicht der im HEPI

eingestellten Generierungsfrequenz, wird hinter einem weiteren Flag die dem Histogramm mitgegebene 32 Bit Zeitinformation vollständig ins Telemetriepaket kopiert, damit die Zeitabfolge der Histogramme rekonstruierbar bleibt. Diese Zeit stellt für die nachfolgenden Histogramme eine neue Zeitbasis dar, so wie für die vorherigen Histogramme die Zeitinformation des ersten Histogramms im Data Field Header die Zeitbasis war. Da eine 4 Bit Zelle mit Wert Hex(F) bereits als Flag bei nichtkomprimierbaren Zellen eingesetzt wird, benötigt man eine Zellenkombination, die als Flag eindeutig ist. Realisiert wird dieses zweite Flag durch eine Sequenz von drei 4 Bit Zellen mit Hex(F00). Die Zelle mit Hex(F) zeigt an, daß die folgende Histogrammzelle nicht komprimiert ist und aus acht Bit besteht. Eine nichtkomprimierte Zelle darf aber nie den Wert Null haben, da sie sonst komprimierbar gewesen wäre. Die Sequenz Hex(F00) ist also eindeutig und als Flag einsetzbar.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Zelle6				Zelle7				Zelle8				Flag Hex(F)			
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Flag Hex(00)								MSB		Eventzeit (32 Bit)					
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Eventzeit (32 Bit)															
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Eventzeit (32 Bit)						LSB		Zelle1				Zelle2			
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Zelle3				Zelle4				Zelle5				Zelle6			

Abbildung 5.22: Histogrammfolge nicht regelmäßig.

Die Laufzeit der Packroutine für die Spectral Timing Funktion ist, ebenso wie die Anzahl der erzeugten Telemetriepakete, abhängig von der Anzahl komprimierbarer Zellen innerhalb des Histogramms. Wie oben bereits ausgeführt, ist es relativ unwahrscheinlich, daß sich eine nicht komprimierbare Zelle im Histogramm befindet. Die Routine benötigt für das Komprimieren und Verpacken eines vollständig komprimierbaren Spectral Timing Histogramms ca. 343 μs. Sollte dagegen keine Zelle komprimierbar sein, verlängert sich die Laufzeit auf ca. 687 μs. In diesem Fall würde der Platzbedarf des Histogramms aufgrund der eingeschobenen Flags um den Faktor 1.5 vergrößert.

5.1.6 Die Histogrammfunktionen S5, S6 und S7

Die Histogramme werden von der DPE blockweise über das HBR-B vom HEPI–Speicher ausgelesen. Die Blocklänge beträgt 1026 Byte (1 Byte = 8 Bit). In den ersten beiden Bytes steht die Startadresse des sich direkt anschließenden Histogrammblocks. Die 16 Bit lange Adresse ist viel zu klein, um die genaue HEPI–Speicheradresse des Histogrammblocks aufzunehmen. Doch das ist auch gar nicht notwendig. In einem Block werden 1024 Histogrammzellen aus je 8 Bit zusammen an die DPE übermittelt. Da der HEPI–Speicher byteweise aufgebaut ist, jedes Byte und damit jede Histogrammzelle eine eigene Adresse besitzt, umfaßt ein gesendeter Histogrammblock die Zellen von einer bestimmten Adresse bis zur Adresse + 1024. Ein Histogrammblock ist also immer noch eindeutig adressiert, wenn man die Startadresse der ersten Zelle in diesem Block durch 1024 teilt. Für einen im Binärsystem arbeitenden Prozessor bedeutet eine Division durch 1024 eine Verkleinerung der Adresslänge um 10 Bit, was auf das Auslesen eines gesamten Histogramms umgerechnet eine Reduzierung des Datenstroms auf dem HBR-B um bis zu 1280 Byte bedeutet.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Startadresse (1 word)		Data 1	Data 2	Data 3	Data 4
Byte 7	Byte 8	...	Byte 1024	Byte 1025	Byte 1026
Data 5	Data 6	...	Data 1022	Data 1023	Data 1024

Abbildung 5.23: Das HBR-B Histogramm Ausgabeformat des HEPI.

Wie in Kapitel 3.1 gesehen, ist die einzige Aufgabe der Histogrammfunktionen das Komprimieren und Verpacken eines festgelegten Speicherbereichs im HEPI, in dem die Histogramme erstellt werden. Die Histogramme sind aus Zellen einer Größe von 8 Bit aufgebaut. Welche wissenschaftliche Bedeutung (Energiebereich, Polarisationswinkel usw.) innerhalb des Histogramms eine Zelle besitzt, ist für die Packroutinen irrelevant. Es wurden nur deshalb drei eigenständige Module bzw. Prozeduren programmiert, weil die Verwaltung der Histogramme und der Telemetriepakete sehr viel einfacher zu realisieren ist, als wenn nur eine Packroutine alle Histogramme bearbeitet. Die drei wissenschaftlichen Funktionen *C*sI Calibration, *S*pectral Imaging und *C*sI Polarimetry sind daher völlig identisch aufgebaut. Sie unterscheiden sich lediglich durch den Typ im Data Field Header und damit in der APID ihrer Telemetriepakete.

	Funktion	Typ–Subtyp	APID
S5	CsI Calibration	Hex(50)	1364 = Hex(8D54)
S6	CsI Polarimetry	Hex(60)	1368 = Hex(8D58)
S7	Spectral Imaging	Hex(70)	1372 = Hex(8D5C)
		Hex(71)	1373 = Hex(8D5D)

Tabelle 5.1: Die Histogrammfunktionen

Die Histogrammfunktionen verwenden denselben Komprimierungsalgorithmus, der schon bei den Spectral Timing Histogrammen erläutert wurde. Deswegen wird hier auf eine Wiederholung verzichtet. Aufgrund der hohen Integrationszeiten ist es, im Gegensatz zu den Spectral Timing Histogrammen, bei sehr hohen Datenraten keineswegs mehr sehr unwahrscheinlich, daß sich einige nicht komprimierbare Zellen im Histogramm befinden. Außerdem sollen diese Histogramme mit Hilfe der kodierten Aperturmaske Himmelsausschnitte abbilden und daher erfährt jedes Pixel während eines Pointings eine etwas unterschiedliche Bestrahlung durch die Maske.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp (8 Bit)								Histogramm ID (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
HEPI Startadresse der ersten Histogrammzelle															
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
innerhalb dieses Telemetriepakets (32 Bit)															

Abbildung 5.24: Das Data Field Header Format der Histogrammfunktionen.

Das TM Paket Format: Data Field Header (3 word).
856 Histogramm–Datenzellen der Länge von je 4 Bit.
Keine Spare words am Paketende.
Zusammen: 217 words

Die Laufzeiten der Histogrammfunktionen können nicht mit Hilfe des Software Simulators ermittelt werden. Sie werden lediglich anhand der Zeiten der Spectral Timing Histogramme abgeschätzt.

5.2 Diagnostic Functions

Die diagnostischen Funktionen brauchen keinerlei Bedingungen zu prüfen oder Vorbereitungen durchzuführen. Ihre einzige Aufgabe ist das Verpacken der Rohdaten in Telemetriepakete. Im Gegensatz zum wissenschaftlichen Mode ist der Ablauf des Diagnostic Mode zeitlich nicht kritisch. Das bedeutet nicht, daß die Eventrate so gering ist, daß die DPE mühelos mit dem Umpacken der Daten zu Rande kommt. Es sollen lediglich einzelne Detektormodule auf ihre Funktion getestet werden. Ob dabei alle Ereignisse übermittelt werden oder ob Events verloren gehen, ist nur zweitrangig.

Die Laufzeiten der diagnostischen Funktionen wurden nicht ermittelt. Bei einem Versuchsaufbau der DPE, bei dem dem auf Transparenz Mode eingestellten HEPI ein Detektorsimulator vorgeschaltet wurde, der beliebig viele Daten liefert, wurde die maximale Telemetrierate um den Faktor drei überboten, d.h. es hätten dreimal mehr Ereignisse verarbeitet als in Telemetriepaketen verschickt werden können.

5.2.1 Die CdTe Diagnostic Funktion D1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								–	Risetime (7 Bit)						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Spare (8 Bit)								MSB Eventzeit (24 Bit)							
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Eventzeit (24 Bit) LSB															
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (5 Bit)					Amplitude (11 Bit)										
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–	Y_Pos (7 Bit)							–	Z_Pos (7 Bit)						

Abbildung 5.25: Das D1–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1397 = Hex(8D75)
Data Field Header (2 word)
53 TM Event Records der Länge von 64 Bit
2 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Das letzte word ist einem möglichen CRC vorbehalten.
Zusammen: 217 words.

Im Data Field Header aller diagnostischen Funktionen steht der Zeitwert der On Board time, an dem das erste Event in das Paket geschrieben wurde. Die Einheit der *Packet Creation Time* ist eine Sekunde.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp = HEX(D1)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB Packet Creation Time (16 Bit) LSB															

Abbildung 5.26: Das Data Field Header Format der D1–Funktion.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								MSB			Eventzeit (24 Bit)				
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
				Eventzeit (24 Bit)											LSB
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Amplitude (11 Bit)											Risetime (7 Bit)				
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
LSB		Y_Pos (7 Bit)							Z_Pos (7 Bit)						

Abbildung 5.27: Das TM Event Record Format der D1–Funktion.

Dummy Handling:
Da bei allen diagnostischen Daten die Eventzeit so, wie sie vom HEPI ausgegeben wird, in die Telemetriepakete geschrieben wird, ist ein Dummy Handling überflüssig.

5.2.2 Die CsI Diagnostic Funktion D2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Spare (8 Bit)								MSB			Time (24 Bit)				
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
					Time (24 Bit)										LSB
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (6 Bit)						Amplitude (10 Bit)									
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
–		Y_Pos (6 Bit)						–		Z_Pos (6 Bit)					

Abbildung 5.28: Das D2–Ausgabeformat des HEPI.

Das TM Paket Format: APID = 1398 = Hex(8D76)
Data Field Header (2 word)
53 TM Event Records der Länge von 64 Bit
2 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Das letzte word ist einem möglichen CRC vorbehalten.
Zusammen: 217 words.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp = HEX(D2)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB					Packet Creation Time (16 Bit)										LSB

Abbildung 5.29: Das Data Field Header Format der D2–Funktion.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Event–Typ (8 Bit)								MSB			Time (24 Bit)				
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Time (24 Bit)															LSB
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Spare (6 Bit)						Amplitude (10 Bit)									
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Spare (4 Bit)				Y_Pos (6 Bit)						Z_Pos (6 Bit)					

Abbildung 5.30: Das TM Event Record Format der D2–Funktion.

Dummy Handling:
Da bei allen diagnostischen Daten die Eventzeit so, wie sie vom HEPI ausgegeben wird, in die Telemetriepakete geschrieben wird, ist ein Dummy Handling überflüssig.

5.2.3 Die Transparenz–Funktion D3

Die Transparenz–Funktion wurde eigentlich nur zum Testen des HEPI–Boards entwickelt. Treten in den über die HBR-A Leitung übertragenen Daten Verschiebungen im Format des auszulesenden Datenblocks auf, liegen die Typfelder der einzelnen Events nicht mehr an den definierten Stellen innerhalb des Buffers und die IASW kann die Eventarten nicht identifizieren. Deshalb wurde diese Packroutine entwickelt, die den vom HEPI ausgelesenen Datenblock unabhängig von dessen Inhalt in Telemetriepakete kopiert. Lediglich das 80 Bit Format der Rohdaten wird beachtet. Da die Modi von IASW und HEPI unabhängig voneinander gesetzt werden können, werden mit dieser Funktion auch die HEPI–Ausgabeformate der wissenschaftlichen Events kontrolliert. Um die ebenfalls vom HEPI gelieferten Spectral Timing Histogramme bzw. Compton–Events, die beide eine Länge von 160 Bit haben, nicht auseinanderzureißen, werden beide Formate von der Routine unterstützt. Kann die IASW den Typ des Events nicht erkennen, werden 80 Bit in das Telemetriepaket kopiert.

Das TM Paket Format: APID = 1399 = Hex(8D77)
Data Field Header (2 word)
205 oder 210 words Daten
9 oder 4 Spare word am Paketende, mit HEX(FFFF) initialisiert.
Das letzte word ist einem möglichen CRC vorbehalten.
Zusammen: 217 words.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Typ–Subtyp = HEX(D3)								Spare (8 Bit)							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSB			Packet Creation Time (16 Bit)												LSB

Abbildung 5.31: Das Data Field Header Format der D3–Funktion.

Das Format der Daten in den Telemetriepaketen ist eine genaue Kopie der Ausgabe von HEPI. Eine Stuktur besteht nur hinsichtlich des 80 Bit bzw. 160 Bit Formats der von HEPI ausgegebenen Events.

5.3 Die Leistungsfähigkeit der IASW

Um eine Aussage darüber zu treffen, inwieweit die IASW die Datenverarbeitung des IBIS–Instruments in Echtzeit bewerkstelligen kann, müssen viele interne und externe Parameter berücksichtigt werden. In den vorangegangenen Abschnitten wurden die Laufzeiten der wissenschaftlichen Funktionen und die komprimierten Eventformate ihrer Telemetriepakete vorgestellt. Um eine realistische Abschätzung der Leistungsfähigkeit durchführen zu können, müssen die Abarbeitungszeit der Schleife im Modul SCIENCE_MODE genauso berücksichtigt werden wie die Aktivitätszeiten der Tasks, die vor dem Main Task ablaufen, wie der Housekeeping Task und die Histogramm Tasks sowie der Tasks, die den Main Task unterbrechen, wie der HBR-A Task. Daß der IASW von einem 125 ms RTC Takt nur eine bestimmte Zeitspanne zur Verfügung steht, muß auch in die Abschätzung einfließen. Die Datenrate des Housekeeping Tasks mit einem H1–Paket alle acht Sekunden und einem H3–Paket alle 32 Sekunden ist zwar nicht hoch, geht aber dennoch mit in die Berechnung ein. Die zu bestimmten Ereignissen generierten H2–Pakete können nur sehr schwer mit einbezogen werden.

Der bestimmende Hauptparameter ist natürlich die Datenrate der Detektoren. Die effektive Datenrate ergibt sich jedoch erst, wenn man die getroffenen Selektionen mit einbezieht. So werden von ISGRI nur Einzelereignisse verarbeitet, die in einem in der IASW eingestellten Energiebereich liegen. Bei den CsI–Events wird diese Energieselektion bereits durch den HEPI vorgenommen. Hinzu kommen die Integrationszeiten der Histogramme. Je kürzer die Integrationszeit der Spectral Timing Histogramme eingestellt ist, desto mehr wird die Rechenkapazität der DPE belastet und umso stärker steigt die Anzahl der generierten Telemetriepakete an. Genauso verhält es sich bei den großen im HEPI zusammengestellten Histogrammen, bei denen ein festgelegter Speicherbereich ausgelesen werden muß. Da die Abarbeitung eines Histogramms über den gesamten Integrationszeitraum des nachfolgenden Histogramms gleichmäßig verteilt wird, beansprucht der Histogram Reading Task um so mehr Zeit eines RTC Zykluses, je kürzer die Integrationszeit gewählt wird.

		Solar Minimum ESA–Szenario	Solar Maximum ESA–Szenario	Solar Maximum Instrumentengruppe
S1	CdTe Photon by Photon	980	630	325
S2	CdTe Calibration	20	20	20
S3	Compton Photon by Photon	240	120	20
S4	CsI Photon by Photon	1000	1000	635
S8	Spectral Timing	1024	1024	1024

Tabelle 5.2: *Datenraten der verschiedenen Szenarien*

Zu einem ganz erheblichen Teil wird die Datenrate der Detektoren vom Hintergrund bestimmt sein. Die Stärke des Hintergrunds schwankt mit der Sonnenaktivität. Erreicht die Sonne ein Aktivitätsminimum, ist der Hintergrund in den Detektoren am größten, bei einem Maximum der Sonnenaktivität hingegen am geringsten. Dementsprechend wurden zwei Szenarien von der ESA entwickelt, welche die Ereignisrate von ISGRI und PICsIT abschätzen. Tabelle 5.2 zeigt die HBR-A Datenraten der einzelnen wissenschaftlichen Funktionen für beide Szenarien. Einziger Unterschied sind die deutlich niedrigeren Datenraten der CdTe- sowie der Compton Photon by Photon–Funktion. Von der IBIS Instrumentengruppe wurde ebenfalls ein Solar Maximum

Szenario erstellt. Ihre Datenraten sind erheblich tiefer angelegt als die Werte des entsprechenden ESA–Szenarios.

Mit Hilfe einer Excel–Tabelle können die von der IASW benötigte CPU–Zeit und die resultierende Telemetrierate direkt aus den Werten für Laufzeiten, Formatgrößen und Ereignisraten gewonnen werden. Angezeigt wird auch, zu welchen prozentualen Anteilen sich diese Resultate auf ISGRI- bzw. PICsIT–Ereignisse zurückführen lassen. Die Excel–Tabelle erlaubt es auch, durch das Verändern eines einzigen Parameters, z.B. einer Eventrate, direkte Rückschlüsse auf CPU–Zeit und Telemetrierate zu erhalten. Durch das Experimentieren an einzelnen Parametern kann ermittelt werden, bei welchen Laufzeiten dringender Optimierungsbedarf besteht und bei welchen Telemetriepaketen die Art der Datenkomprimierung noch einmal überdacht werden muß. Es genügt dabei, für diese Abschätzungen nur die beiden wissenschaftlichen Submodi Standard Poining und Photon by Photon zu betrachten. Die Submodi unterscheiden sich, wie in Kapitel 3.2.5 gesehen, lediglich durch die zu erstellenden Histogramme. Das Histogramm mit der größten Anzahl von Zellen ist das Spectral Imaging Histogramm, das während des Pointing im Standard Submode integriert und ausgelesen wird. Dazu werden in diesem Submode auch die Spectral Timing Histogramme erzeugt. Der Standard Submode benötigt damit von allen wissenschaftlichen Submodes im Pointing die meiste CPU–Zeit. Der Photon by Photon Mode ist zum einen ein eigenständiger Submode, zum anderen benutzen die anderen Submodes die identischen Einstellungen in den Slewing–Phasen des Satelliten. Durch die Betrachtung dieser beiden Submodi wird der gesamte Scientific Mode abgedeckt.

In Tabelle 5.3 sind die wichtigsten Leistungsmerkmale der IASW für die beiden ESA–Szenarien Solar Minimum/Solar Maximum und die wissenschaftlichen Submodi Standard Pointing/Photon by Photon des Scientific Modes zusammengestellt. In der ersten Zeile steht die Laufzeit in Millisekunden, welche die IASW zur Bewältigung aller Aufgaben innerhalb eines RTC–Takts von 125 ms braucht. Die zweite Zeile gibt dasselbe als Prozentwert (100% = 125 ms) an. Die folgenden Zeilen enthalten die jeweiligen Laufzeitanteile, die für die Eventverarbeitung eines Detektors benötigt werden. In gleicher Art und Weise werden die Werte für die Telemetrierate angegeben. Zuerst wird die absolut benötigte Datenrate als Telemetriepakete (TM) pro RTC–Takt angegeben. Mit der Standardeinstellung der *Polling Sequence Table* von 129 TMs innerhalb von acht Sekunden ergibt sich ein Standardwert von etwa 2 TMs/RTC. Es folgen derselbe Wert als prozentuale Angabe (100% = 129 TMs/RTC) und die Anteile, zu denen die Events des jeweiligen Detektors zur Datenrate beitragen.

Die Berechnungen der benötigten CPU–Zeit stützen sich auf die im vorherigen Abschnitt vorgestellten Laufzeiten und Formate. Die zugrundeliegenden Eventraten der einzelnen Funktionen für die beiden ESA–Szenarien können der Tabelle 5.2 entnommen werden. Es wird dabei angenommen, daß lediglich 75% der CdTe Photon by Photon Events nach der Amplitudenkorrektur in die Telemetriepakete übernommen werden. Für jedes Compton–Ereignis muß sowohl die Amplitudenkorrektur als auch der Selektionsalgorithmus angewandt werden, wobei lediglich 25% die Bedingungen zur Weiterverarbeiung erfüllen. Bei allen anderen Eventarten findet keine Selektion innerhalb der IASW statt. Die Integrationszeit der Spectral Imaging Histogramme sei auf 20 Minuten, die der CsI Kalibrationshistogramme auf 60 Minuten eingestellt. Von allen Housekeeping Paketen werden nur die alle acht Sekunden generierten Essential Housekeeping berücksichtigt. Desweiteren sind in der Excel–Tabelle viele weitere Parameter enthalten, die hier nicht weiter beschrieben werden (z.B. Transferzeiten, Zeitabschätzungen für Monitoring Prozesse u.a.). Zusätzlich wird zur Berechnung der CPU–Zeit noch ein konstanter Unsicherheitsfaktor von 10 Millisekunden pro 125 ms RTC Zyclus addiert.

Solar Minimum		
Scientific Submode	Standard Pointing	Photon by Photon
benötigte CPU Zeit in [ms/RTC]	169	114
CPU Zeit [%] der IASW (95 ms/ RTC)	178	120
CPU Anteil [%] durch ISGRI	70	70
CPU Anteil [%] durch PICsIT	98	40
Datenrate in [TMs/RTC]	2,72	2,44
Datenrate [%] der IASW (129TMs/8s)	135	121
TM Anteil [%] durch ISGRI	58	58
TM Anteil [%] durch PICsIT	77	63

Solar Maximum		
Scientific Submode	Standard Pointing	Photon by Photon
benötigte CPU Zeit in [ms/RTC]	136	81
CPU Load [%] der IASW (95 ms/ RTC)	143	86
CPU Anteil [%] durch ISGRI	43	43
CPU Anteil [%] durch PICsIT	90	32
Datenrate in [TMs/RTC]	2,27	1,99
Datenrate [%] der IASW (129TMs/8s)	113	99
TM Anteil [%] durch ISGRI	37	37
TM Anteil [%] durch PICsIT	75	61

Tabelle 5.3: *Werte für das Solar–Min und Solar–Max Szenario der ESA*

Entscheidend für die Bewertung der IASW ist ihre Leistungsfähigkeit bei hohen Ereignisraten. Können diese vollständig verarbeitet werden, ist sie den Anforderungen einer Flugversion ein Stück näher gekommen. Mit einer CPU–Auslastung von 178% während des Standard Pointings im Solar Minimum Szenario ist sie allerdings weit davon entfernt, der Aufgabe der Echtzeit–Datenverarbeitung aller auftretenden Ereignisse gerecht zu werden. Die Abarbeitung der Spectral Timing Histogramme benötigt bereits 63% der vorhanden Rechenkapazität. Mit 54% braucht die Verarbeitung der CdTe Photon by Photon Events ebenfalls mehr als die Hälfte der möglichen CPU–Zeit. Weitere 32% beansprucht die Compton Funktion, die großen Histogramme 19%. In 11% der zur Verfügung stehenden Rechenzeit erledigt die IASW alle noch nicht aufgezählten Aufgaben.

Mit ihrem CPU–Load von 178% würde die IASW mehr Telemetriepakete erzeugen, als durch die vorgesehene Einstellung des Polling Sequence Tables der CDMU von IBIS abgeholt werden. Allein 60% der bereitstehenden Telemetrierate werden durch die Spectral Timing Histogramme beansprucht. Die CdTe Photon by Photon Funktion fordert mit 55% nahezu genauso viel ein, obwohl ein Viertel dieser Events nicht in die Pakete übernommen wurde[1]. Die beiden großen Histogramme beanspruchen 17% der Telemetrierate, die Compton–Ereignisse 8%[2]. Mit dem Bedarf der anderen Funktionen und der Housekeeping wird die zur Verfügung stehende Telemetrierate um 35% überschritten.

Etwas besser sieht es für den Photon by Photon Submode aus. Sowohl die Rechenzeit als auch die Telemetrierate werden dennoch um etwa 20% überschritten. Der Bedarf der auch im Standard

[1] Das Verpacken aller Ereignisse würde zu den 55% noch eimal weitere 17% hinzufügen. Der CPU–Load wäre dabei nur um ein Prozent angestiegen.

[2] Auch hier hätte das Mitnehmen aller Ereignisse den CPU–Load nur um ein Prozent, die Telemetrierate jedoch um 21% erhöht.

Submode aktiven Funktionen ist natürlich auch in diesem Submode derselbe. Es läßt sich hier jedoch leicht die Notwendigkeit des HEPI erkennen. Obwohl die Verarbeitung der CsI–Events nur 23% der Rechenzeit verbraucht, ist der Anteil an der Telemetrie mit 59% der Gesamtrate sehr hoch. Da im Photon by Photon Mode die CsI–Kalibration als einzige Histogrammart aktiv ist, kann ihr Bedarf an Rechenzeit und Telemetrierate mit je einem Prozent bestimmt werden. Mit dem Ergebnis aus dem Standard Submode ergibt sich für das Spectral Imaging Histogramm ein CPU–Bedarf von 18% und ein Anteil an der Telemetrierate von 16%.

Kapitel 6

Zusammenfassung

Die Entwicklung und Realisierung einer *Instrument Application Software* für das wissenschaftliche Instrument *Imager on Board of INTEGRAL Satellite* war und ist für mich eine große Herausforderung und ich bin nun, da ich meine Diplomarbeit nahezu beendet habe, froh, daß ich mich für diese Arbeit entschieden habe, obwohl mir jegliche Vorkenntnisse über Computerhardware oder Softwaredesign fehlten.

Zu Beginn galt es, mir erst einmal die Syntax und strukturellen Möglichkeiten der Programmiersprache ADA anzueignen und die Entwicklungsumgebung mit Software–Simulator und Debugger kennenzulernen. Parallel dazu machte ich mich durch das Studium zahlreicher Dokumente mit dem INTEGRAL–Projekt und den Aufgaben der IASW vertraut.

Die ersten Arbeiten an der eigenlichen Software begannen mit dem Austesten von Variablentypen beim Umkopieren von selbst erstellten Datenblöcken unter Berücksichtigung der benötigten Laufzeiten. So entstanden schon recht früh einige kleine Programme, die sich langsam zu den heute in der IASW implementierten Packfunktionen entwickelten. Parallel entstanden die ersten Tasks und das jetzige Design der Software.

Eine gründliche Einweisung in Aufbau und Anwendung der CSSW und des Simulators erhielt ich während insgesamt drei Besuchen bei der Firma GMV in Tres Cantos/Spanien. Mike Rennie ging mit mir zusammen die Quellcodes der damaligen Routinen durch, erklärte mir verschiedene compilerinterne Abläufe, und mit seiner Hilfe konnte ich den bislang undokumentierten Compilerfehler, der die Initialisierung der großen Wertetabelle der Energiekorrektur bislang nicht zuließ, umgehen. Die meisten meiner Kenntnisse über die Speicherverwaltung mit Address States und der richtigen Konfiguration des Linkerfiles konnte ich aus der Zusammenarbeit mit ihm erwerben.

Die *Instrument Application Software* wurde in einem Software-Simulator entwickelt und lief in Tübingen, eingebaut in einer DPE, innerhalb einer Entwicklungsumgebung mit einem Detektorsimulator bereits mehrere hundert Stunden lang erfolgreich. Die ersten offiziellen Funktionalitätstests wurden ab Oktober 1998 in Tübingen durch Mitarbeiter der Firma LABEN aus Mailand/Italien durchgeführt. Diese Tests dienten der Softwareverifikation zur EM-0 Integration des Imagers IBIS im Novebmer 1998 bei LABEN. Die IASW stellte dabei ihre Funktionalität auch innerhalb der realen Satellitenumgebung unter Beweis, als zum ersten Mal Photonen aus einer radioaktiven Quelle von einem CdTe-Pixel des ISGRI – Detektors registriert und anschließend das resultierende Telemetriepaket auf einem Bildschirm ausgegeben wurde.

Sicherlich erfüllt sie noch nicht die Anforderungen, die an eine Flugversion der IASW gestellt werden müssen. Die einzelnen Laufzeiten der Routinen sind noch nicht kurz genug, um gerade bei hohen Datenraten mit der vorhandenen Rechenkapazität der DPE alle Ereignisse verarbeiten zu können. Daß dennoch die zur Verfügung stehende Telemetrierate weit überschritten wird, obwohl bis auf die CdTe Photon by Photon Ereignisse alle Eventarten die angestrebte Komprimierungsrate von 50% bereits erfüllen, wirft die Frage auf, ob die Sendeleistung des Satelliten ausreichend dimensioniert ist.

Die IASW ist eine Software, die sich mit ihren zahlreichen Einstellungsmöglichkeiten den gegebenen Erfordernissen anpassen kann. Zusammen mit dem *Hardware Event Preprocessor for IBIS* können aus dem Datenstrom die Ereignisse aus einem bestimmten Energiebereich herausselektiert werden. Das Verfahren, nicht nur die wissenschaftlichen Funktionen einzeln beim Erreichen bestimmter Telemetrieauslastungen ausschalten zu können, läßt indirekt die Vergabe von Prioritäten an alle innerhalb der IASW generierten Arten von Telemetriepaketen zu. Das Ausschalten von Paketarten verteilt die verfügbare Telemetrierate auf die noch eingeschalteten Funktionen und gibt diesen die Gelegenheit, ihre entsprechenden Datensätze lückenlos zur Erde weiterzugeben. Die Software steuert also selbständig den Datenstrom von IBIS an die CDMU gemäß der in einer Tabelle einstellbaren Prioritäten der Telemetriepaketarten.

Kapitel 7

Ausblick

Die *Instrument Application Software* wird bereits nach einem Jahr Entwicklungszeit ihrer Aufgabe, die Steuerung und die Echtzeit–Datenverarbeitung des IBIS–Instruments durchzuführen, in vollem Umfang gerecht. Zwar ist sie aufgrund der geringen Rechenkapazität der DPE nicht in der Lage, bei hohen Ereignisraten alle von den Detektoren registrierten Ereignisse zu verarbeiten, aber sie kann deutlich mehr Daten bereitstellen, als IBIS an Telemetrierate zugeteilt wurde. Der „Flaschenhals“ der Echtzeit–Datenverarbeitung ist damit nicht, wie anfänglich befürchtet, die DPE, sondern die Telemetrie des Satelliten. Das bedeutet jedoch nicht, daß damit die Entwicklung der IASW beendet ist. In diesem Kapitel soll beschrieben werden, an welchen Stellen in der Software nach weiteren Optimierungsmöglichkeiten gesucht wird.

Der Watch–dog Reset, bis jetzt im Background Task der CSSW untergebracht, zwingt die IASW, nach einer definierten Zeitspanne ihre Aktivität an den Background Task abzugeben. Das aktuelle Design erlaubt der IASW, 95 ms von 125 ms eines RTC Takts zu arbeiten. Nach 95 ms stoppt der Main Control Task die Aktivität der IASW und läßt dem Background Task die verbleibenden 25 ms des RTC. Der Watch-dog Reset, der im wesentlichen aus dem Zurücksetzen eines Hardwarezählers besteht, benötigt jedoch im Normalfall weit weniger Zeit. Die 25 ms dienen also der Sicherheit, schließlich werden CSSW und IASW bei Nichtzurücksetzen des Watch–dogs angehalten und Beobachtungszeit geht verloren. Wird der Watch-dog Reset von einem Task durchgeführt, der eine höhere Priorität besitzt als alle IASW Tasks, ist er nur über die Zeit aktiv, die er wirklich benötigt, direkt nach einem RTC–Signal. Die freigewordene Zeitspanne kann dann an die IASW abgetreten werden, deren Aktivitätszeit dann bis zum nächsten RTC dauern kann. Die IASW bekommt also einen höheren Anteil eines RTC–Zyklusses. Gleichzeitig sinkt der Verwaltungsaufwand innerhalb der IASW, da die erlaubte Arbeitsdauer nicht mehr abgestoppt zu werden braucht. Dieses Verlegen der Aufgaben aus dem Background Task in Tasks, die nach einem RTC–Signal vor der IASW starten, wird gerade von Mike Rennie von der Firma GMV umgesetzt und wird in einer zukünftigen Version der CSSW implementiert sein. Für die IASW bedeutet diese Änderung, daß der Main Control Task überflüssig geworden ist und wegfallen kann. Inwieweit das gesamte Design der IASW betroffen ist, muß noch abgeklärt werden.

Bei dem Data Field Header Format der Histogrammpakete ist ebenfalls eine Änderung vorgesehen. Da er bislang noch keinerlei Zeitinformation über das entsprechende Histogramm enthält, soll er von drei word auf fünf word erweitert werden. Der Data Field Header wird dann zusätzlich zu der Startadresse der ersten Histogrammzelle, deren Länge von 32 Bit auf 24 Bit reduziert werden kann, die Integrationszeit (16 Bit) und die Startzeit des Histogramms (24 Bit) enthalten. Der Histogrammtyp und die Histogramm–ID (je acht Bit) bleiben im Format unverändert.

Die Histogrammfunktionen verwenden einen Komprimierungsalgorithmus, der auf dem Halbieren der acht Bit großen Histogrammzellen auf vier Bit basiert. In jeder Zelle sollten daher nicht mehr als vierzehn Events registriert sein, da nur dann die Zellengröße von vier Bit ausreicht. Für die Spectral Timing–Histogramme werden die Zellen nahezu immer komprimierbar sein, da man die Integrationszeit so kurz wie sinnvoll möglich einstellen wird. Lediglich bei den großen Histogrammarten kann es eventuell Schwierigkeiten geben. Bei langen Integrationszeiten (1200 s) wird der kritische Füllstand der Zellen für eine statistische Energieverteilung bei etwa 13 000 Ereignissen pro Sekunde erreicht. Diese Eventrate liegt zwar deutlich über der erwarteten Durchschnittsrate, aber auch noch unter dem worst case Szenario. Wenn die Anzahl der nichtkomprimierbaren Zellen keine Einzelfälle mehr bleiben, kann das implementierte Komprimierungsverfahren die Histogramme auf das Eineinhalbfache vergrößern. Eventuell sollte deshalb das Komprimierverfahren in Fällen sehr hoher Ereignisraten abstellbar sein. Ob die Verteilung der Ereignisse bei einer Abbildung durch die kodierte Aperturmaske näherungsweise als statistisch betrachtet werden kann, sollen Computersimulationen mit angenommenen Quellenverteilungen zeigen.

Die IASW unterstützt die Möglichkeit, IBIS wie ein Compton–Teleskop zu verwenden. Da für Photonen mit Energien im MeV–Bereich der Tubus des Satelliten transparent wird, beträgt das Gesichtsfeld des Imagers als Compton–Teleskop nahezu 2π. Damit alle MeV–Compton–Ereignisse verarbeitet werden, ist in der Compton Photon by Photon–Funktion bereits eine via Telekommando einstellbare Energieschwelle implementiert, die nur Ereignisse geringerer Energie einer Selektionsfunktion zuführt. Die selbst hergeleitete Compton–Rekonstruktionsformel soll einem anderen Algorithmus weichen, welcher mit berücksichtigt, ob die erste Interaktion in ISGRI oder in PICsIT stattgefunden hat. Dieser Algorithmus kann also auch rückgestreute Photonen rekonstruieren. In ihm sind vier verschiedene Wurzelberechnungen enthalten, die in der Berechnung zeitaufwendig sind. Anstelle des implementierten relativ langsamen Wurzelalgorithmus von Newton kann die Wurzel auch, etwas vereinfacht gesagt, über das Halbieren der Potenz einer Gleitkommazahl berechnet werden. Ein entsprechender in Assembler Code geschriebener Algorithmus wurde in der TLD–Umgebung des Entwicklungssystems gefunden. Er verkürzt die Berechnungszeit einer Wurzel um den Faktor fünf.

In der IASW ist des weiteren auch noch Spielraum für Laufzeitoptimierungen enthalten. Um einen Softwareabsturz aufgrund eines Laufzeitfehlers auszuschließen, wurde nahezu jedes Unterprogramm mit einem Exception Handler ausgestattet. Eine Exception tritt auf, wenn eine ordnungsgemäße Ausführung des Programms nicht möglich ist, wenn z.B. ein Wertebereich einer Variablen überschritten wurde. Stellt man aber sicher, daß im Ablauf eines Unterprogramms keine Exception auftreten kann, ist das Anbringen eines Exception Handlers überflüssig. Bei jedem Aufruf und Verlassen eines Unterprogramms beansprucht der Exception Handler ca. 15-20 μs Rechenzeit, die eingespart werden können, wenn das Auftreten von Laufzeitfehlern ausgeschlossen ist.

Der gesamte Quellcode der IASW wird ab Mitte Januar 1999 von Mike Rennie auf Laufzeitoptimierung kontrolliert. Die Abarbeitungszeit eines CdTe Photon by Photon Events mit Amplitudenkorrektur konnte durch Entfernen der Exception Handler und durch Vermeiden von internen Bereichsprüfungen bei Zuweisungen, die bei der Übersetzung des ADA Codes in Assembler vom Compiler vorgenommen werden, um 40% reduziert werden. Es ist anzunehmen, daß ein ähnliches Ergebnis auch bei allen anderen Packroutinen zu erzielen ist. Die IASW wird durch all diese Optimierungen sicherlich noch leistungsstärker werden. Damit bleibt jedoch die Telemetrierate mehr als bisher der „Flaschenhals“ der Echtzeit–Datenverarbeitung.

Literaturverzeichnis

[1] S. Bergeson-Willis et al. INTEGRAL - Report on the Phase A Study. Technical report, ESA, April 1993.

[2] Cii Honeywell Bull and Honeywell Systems. *ADA Reference Manual*. United States Department of Defense, Juli 1980.

[3] F. Ciceri. IBIS On-Board Data Management User Requirements Document. Technical Report 2, LABEN, Januar 1998. IN-IB-LAB-UR-0001.

[4] ESA. *INTEGRAL Experiment Interface Document Part A for Instruments*, November 1998.

[5] N. A. Gehrels. International Gamma-Ray Astrophysics Laboratory: a future ESA mission for gamma-ray astronomy. In Ramsey and Parnell [13], pages 210–216.

[6] IBIS Team. *INTEGRAL Experiment Interface Document Part B for Instruments*, September 1998. IN.IB.IAS.EID.0022/9/98.

[7] C. Labanti et al. PICSIT: the high-energy detector plane of the IBIS instrument on board INTEGRAL. In Ramsey and Parnell [13], pages 269–279.

[8] Philippe Laurent. *The On Board Compton selection*. Istituto di Astrofisica Spaziale, September 1998. IN-IB-Sap-RP-045.

[9] F. Lebrun et al. ISGRI: a CdTe array imager for INTEGRAL. In Ramsey and Parnell [13], pages 258–268.

[10] G. G. Lichti, V. Schönfelder, et al. The spectrometer SPI of the INTEGRAL mission. In Ramsey and Parnell [13], pages 217–233.

[11] Manfred Nagel. *ADA*. Vieweg, 3 edition, 1991.

[12] Brian D. Ramsey and Thomas A. Parnell, editors. *Gamma-Ray and Cosmic-Ray Detectors, Techniques, and Missions*, volume 2806. The International Society for Optical Engineering, SPIE, August 1996.

[13] Brian D. Ramsey and Thomas A. Parnell, editors. *Gamma-Ray and Cosmic-Ray Detectors, Techniques, and Missions*, volume 2806. The International Society for Optical Engineering, SPIE, August 1996.

[14] Mike Rennie. *Software Interface Control Document*. GMV, Tres Cantos, Spanien, 3.7 edition, November 1998. INT-IC-GMV-0001.

[15] Bruno Reuillon, Christian Puig, and Anne Ly. *Astres 1750A User's Manual*. Matra Marconi Space, 1 edition, Januar 1997. AST-MA-UM-220.

[16] L. Sarlo and M. Sella. INTEGRAL Packet Structure Definition. Technical report, Alenia, Oktober 1996.

[17] Diana Schmidt. *Erfolgreich Programmieren mit ADA*. Springer - Verlag, 1994.

[18] Aeronautic Systems Division. Military Standard Sexteen-Bit Instruction Set Architecture. Technical report, United States Department of Defense, Juli 1980.

[19] IBIS Team. *User Manual for the IBIS Instrument*. Istituto di Astrofisica Spaziale, 2.5 edition, November 1998. IN.IB.IAS.UM.0017/98.

[20] P. Ubertini et al. IMAGER on board INTEGRAL. In Ramsey and Parnell [13], pages 246–257.

[21] Pietro Ubertini. Scientific Performance Report. Technical report, IBIS Team, August 1996. IN.IB.IAS.RP.0010/96.

[22] Reiner Volkmer. HEPI Design Report. Technical report, IAAT, April 1998. IN-IM-TUB-DES-001.

[23] Reiner Volkmer. HEPI Interface Description. Technical report, IAAT, April 1998. IN-IM-TUB-TN/EL-018.

[24] Reiner Volkmer, Eckart Göhler, Patrick Risse, and Martin Stuhlinger. *Software Design Document*. IAAT, 5.0 edition, Januar 1998. IN-IM-TUB-SDD-001.

[25] Nikolai von Krusenstiern. Entwicklung und Realisierung des Hardware Event Processors für den Imager IBIS an Bord des Satelliten INTEGRAL. Master's thesis, Universität Tübingen, Institut für Astronomie und Astrophysik, 1998.

[26] N. J. Westergaard et al. Science with JEM-X. Technical report, JEM-X Team, 1996.

[27] Christoph Winkler. INTEGRAL. *Sterne und Weltraum*, 2:114–121, 1994.