Entwicklung und Realisierung des Hardware Event Processors für den Imager IBIS an Bord des Satelliten INTEGRAL

Diplomarbeit

Eberhard-Karls-Universität zu Tübingen

Institut für Astronomie und Astrophysik

Abteilung Astronomie

vorgelegt von

Nikolai von Krusenstiern

1998

Selbstverlegt von: Nikolai von Krusenstiern
Hohentwielgasse 7
72070 Tübingen

Inhaltsangabe

Abbildungsverzeichnis

Tabellenverzeichnis

Gleichungsverzeichnis

1 Einführung und Überblick

1.1 Einleitung

Das *International Gamma Ray Astrophysics Laboratory*, kurz INTEGRAL, ist ein wissenschaftlicher Röntgenstrahlen- und Gammastrahlen-Satellit, der 2001 gestartet werden soll. Als Observatorium zur Untersuchung harter Röntgen- und Gammastrahlung vereint INTEGRAL gute Abbildungseigenschaften (Winkelauflösung: 12') mit einer guten spektroskopischen Auflösung ($E/\Delta E = 500$)[1]. INTEGRAL ist, wie der Satellit *X-Ray Multi Mirror* (XMM), der 1999 gestartet werden soll, eine Mission im Rahmen des ESA Programms *„Horizon 2000"*.

An Bord befinden sich vier Instrumente:

- ein Spektrometer: *Spectrometer on INTEGRAL* (SPI),
- ein hochauflösendes Gammastrahlen-Teleskop mit abbildenden und spektroskopischen Fähigkeiten: *Imager on Board the INTEGRAL Satellite* (IBIS),
- ein Monitor für Röntgenstrahlung: *Joint European Monitor for X-Rays* (JEM-X) und
- ein optischer Monitor: *Optical Monitoring Camera* (OMC).

Jedes dieser Instrumente benötigt einen eigenen Datenverarbeitungsrechner. Da eine individuelle Entwicklung dieser Datenverarbeitungsrechner höhere Kosten verursacht hätte, hat die *European Space Agency* (ESA) eine *Data Processing Electronics* (DPE) entwickelt, die jedem Instrument zur Verfügung gestellt wird. Diese unterliegt aber, aufgrund ihrer strahlungsfesten Eigenschaften, starken Einschränkungen in Bezug auf ihre Rechenleistung.

Prinzipiell verarbeitet jede DPE die anfallenden Daten und steuert das ihr zugeordnete Instrument. Es stellte sich jedoch als Problem heraus, daß die DPE für den *Imager* IBIS

---

1 Siehe European Space Agency (ESA), INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 2.8.

diese Aufgabe nicht vollständig übernehmen kann. Die erwartete, sehr hohe Datenrate (etwa 15.000 Ereignisse/s[2]) von einer der beiden Detektor-Ebenen des *Imager* IBIS überschreitet die Verarbeitungskapazität der DPE. Hinzu kommt, daß die zur Verfügung gestellte Telemetrie-Rate teilweise um den Faktor 40 überschritten wird[3]. Sie ist selbst im Falle eines solaren Maximums nicht ausreichend für die Fähigkeiten des Instruments[4].

Die erwarteten Detektor-Daten-Raten überschreiten also nicht nur die Telemetrie-Rate, sondern auch die zur Verfügung stehende Rechenleistung der datenverarbeitenden Elektronik DPE[5]. Deshalb ist es notwendig, die Detektor-Daten wissenschaftlich zu verarbeiten, das heißt, sie je nach Forschungsschwerpunkt gezielt auszuwählen und zusammenzufassen.

Diese Aufgabe übernimmt eine elektronische Platine, der *Hardware Event Processor for IBIS* (HEPI). Die DPE befindet sich in einem Kasten, der sogenannten DPE-Box. Innerhalb dieses Kastens befindet sich ein freier Einschub für die HEPI-Platine. Die Detektor-Daten werden bis zur DPE-Box geführt und dort zunächst von der HEPI und dann von der DPE weiterverarbeitet. Ein wesentliches Bauteil dieser Platine ist eine integrierte Schaltung. Diese Platine wird im Folgenden HEPI-Platine genannt. Die integrierte Schaltung wird im Folgenden HEPI genannt. Siehe dazu die Abbildung 1 „Ein Foto der HEPI-Platine und der HEPI (FPGA)“ auf Seite 11.

Da es notwendig ist, sowohl während der Entwicklung als auch während des umfangreichen Testprogramms das Design der HEPI flexibel ändern zu können, wird für das Entwicklungsmodell ein *Field Programmable Gate Array* (FPGA) verwendet. Für das Qualifizierungsmodell und das Flugmodell der HEPI ist jedoch ein strahlungsfestes Bauteil notwendig. Hier wird dann ein *Application Specific Integrated Circuit* (ASIC) verwendet.

Die Entwicklung einer solchen integrierten Schaltung ist aufwendig und kostenintensiv. Dennoch überwiegen die Vorteile gegenüber einer diskret aufgebauten Schaltung:

- der geringe Platzbedarf,
- die kurzen Schaltzeiten,
- die schnelle Verarbeitung der Signale,

2 Siehe IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 84 ff.

3 Siehe ebd. Seite 84 ff.

4 Siehe ebd. Seite 49.

5 Siehe Software und Manager Meeting, Minutes of Meeting, 20. Januar 1998, Tübingen, IN-IM-TUB-9801-I, Annex 3.

- der geringere Leistungsbedarf bei gleicher Geschwindigkeit und
- die verbesserte Störsicherheit durch fehlende Leitungsverbindungen.

Der *Hardware Event Processor for IBIS* und seine Entwicklung am Institut für Astronomie und Astrophysik in Tübingen (IAAT) wird im Folgenden dargelegt[6].

Diese Arbeit gliedert sich in vier Abschnitte. Nach einer allgemeinen Einführung in die Astronomie und den Aufbau des Satelliten INTEGRAL werden die Aufgaben der HEPI beschrieben, und es wird erläutert, welche Hilfsmittel bei der Entwicklung verwendet wurden. Danach folgt eine Beschreibung und Darstellung der einzelnen Module der HEPI sowie ein Ausblick. Ein Anhang beschließt diese Arbeit.

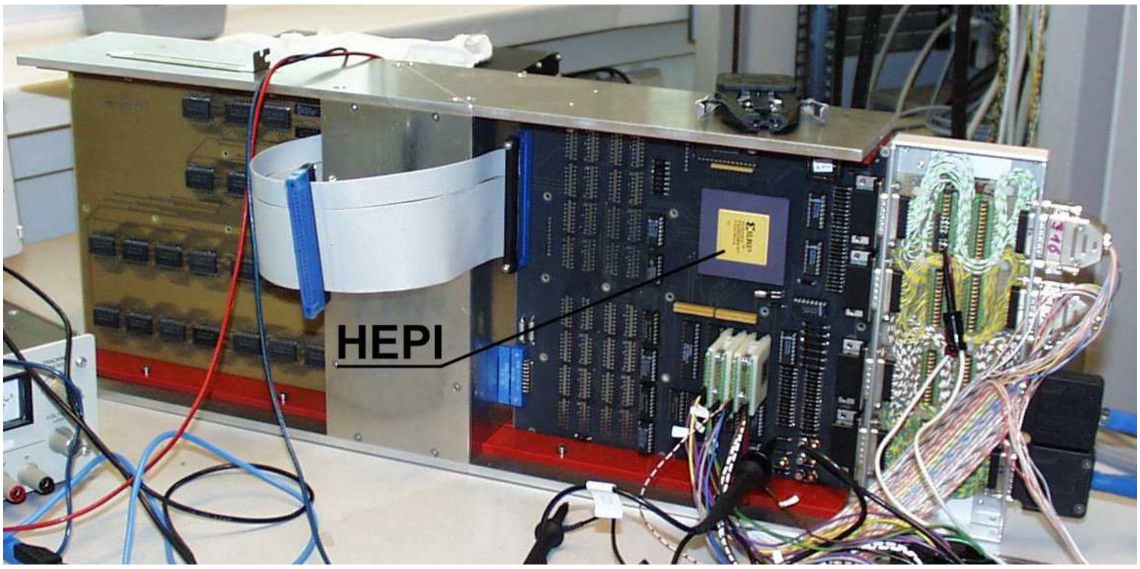

Abbildung 1: Ein Foto der HEPI-Platine und der HEPI (FPGA)[7]

1.2 Astronomie

Der Ursprung der Astronomie lag wahrscheinlich in der Notwendigkeit, die Zeitpunkte für Saat und Ernte genau zu bestimmen. Durch Beobachtung der Sonne und der Sterne wurden auf langen Reisen Himmelsrichtungen und Standorte bestimmt.

Die Erde galt als Mittelpunkt der Welt bis Galileo Galilei (1564-1642)[8] Beweise fand, die das kopernikanische heliozentrische System untermauerten. Mit der Erfindung des

6 Am IAAT und im Folgenden wird „die“ HEPI verwendet, obwohl es „der“ HEPI heißen müßte.

7 Foto von Claus Dreischer, August 1998.

8 Siehe Tipler, Paul A., Physik, Spektrum Akademischer Verlag, Berlin, 1994, Seite 59.

Teleskops bot sich nun auch die Möglichkeit, die kopernikanische Theorie zu überprüfen. 1609 baute GALILEI einen kleinen Refraktor, richtete ihn gegen den Himmel und entdeckte die Venusphasen. Sie wiesen darauf hin, daß dieser Planet um die Sonne kreist. Später entdeckte er auch vier Monde, die um den Jupiter kreisten. Durch Beobachtungen mit unseren Augen im sichtbaren Bereich von etwa 350 nm (violett) bis 750 nm (rot) erschließt sich uns das Universum durch die elektromagnetische Strahlung, die durch die Erdatmosphäre zu uns gelangt.

Erst in der zweiten Hälfte des 20. Jahrhunderts konnte die elektromagnetische Strahlung mit Wellenlängen kleiner als 100 nm erforscht werden. Denn diese wird von der Erdatmosphäre absorbiert, und erst neue astronomische Geräte, wie hochfliegende Ballons[9] oder Satellitenobservatorien, führten zu ihrer astronomischen Erforschung.

Mit solchen Geräten kann man Strahlungen unterschiedlichster Wellenlängen aufzeichnen (zum Beispiel Gammastrahlung, Röntgenstrahlung, Ultraviolettstrahlung und Infrarotstrahlung). Mit Hilfe moderner Technologien lassen sich neue Objekte und Phänomene erforschen: Das sind zum Beispiel Plasmen, die Doppelsterne umgeben, interstellare Bereiche, die Geburtsstätten neuer Sterne darstellen und kalte Staubwolken, die im optischen Bereich unsichtbar sind. Diese, wie auch Kerne von Galaxien, die Schwarze Löcher enthalten können und die Hintergrundstrahlung, die Auskunft über die Frühgeschichte des Universums geben kann, sind zu Forschungsobjekten der modernen Astronomie geworden.

Die Gamma-Astronomie ist ein wichtiges Feld der astronomischen Forschung. Mit Gammastrahlung bezeichnet man in der Astronomie nicht nur, wie in der Physik üblich, die Photonen, die beim radioaktiven Zerfall entstehen, sondern allgemein die Strahlung mit Energien ab 500 keV. Den Gammastrahlung entsteht nicht nur durch radioaktiven Zerfall sondern auch in Wechselwirkungen zwischen Teilchen, Photonen und Magnetfeldern. Deren Erforschung ermöglicht Erkenntnisse über sehr energiereiche Prozesse im ferneren Weltraum, etwa bei Neutronensternen, Schwarzen Löchern und Quasaren. Obwohl sie sehr energiereich ist, wird Gammastrahlung, größtenteils von der Atmosphäre der Erde absorbiert. Daher wurde die Gamma-Astronomie erst möglich, seit Detektoren mit Hilfe von Satelliten außerhalb der Erdatmosphäre positioniert werden können.

Die beiden ersten Satelliten mit Gammastrahlungs-Detektoren waren SAS-2[10] und COS-B[11], die vor mehr als 20 Jahren gestartet wurden. Sie waren mit Funkenkammern ausge-

---

9 Siehe Braccesi et al., 1960 und Cline 1961 nach Trimble, Virginia: „When one photon was a discovery, two was a spectrum, and three was a Rossi Prize", Documents from the Beginnings of Gamma Ray Astronomy, AIP Conference Proceedings 304, pp. 40-44, AIP Press, New York 1993.

10 SAS-2, der zweite *Small Astronomy Satellite* der NASA arbeitete von November 1972 bis Juni 1973 im Energieband oberhalb von 35 MeV.

rüstet, bei denen man aus den Spuren der Elektron-Positron-Paare auf die Einfallsrichtung der Gammaquanten schließen kann. Die Winkelauflösung dieser Instrumente war vergleichsweise gering: mit rund 2° betrug sie ungefähr das Vierfache der scheinbaren Mondgröße.

COS-B fand oberhalb von 50 MeV neben der Gamma-Emission der Milchstraße auch zahlreiche Punktquellen, von denen aber nur wenige mit bekannten Objekten identifiziert werden konnten, so zum Beispiel der Crab-Pulsar[12], der Vela-Pulsar[13] oder der Quasar 3C273[14].

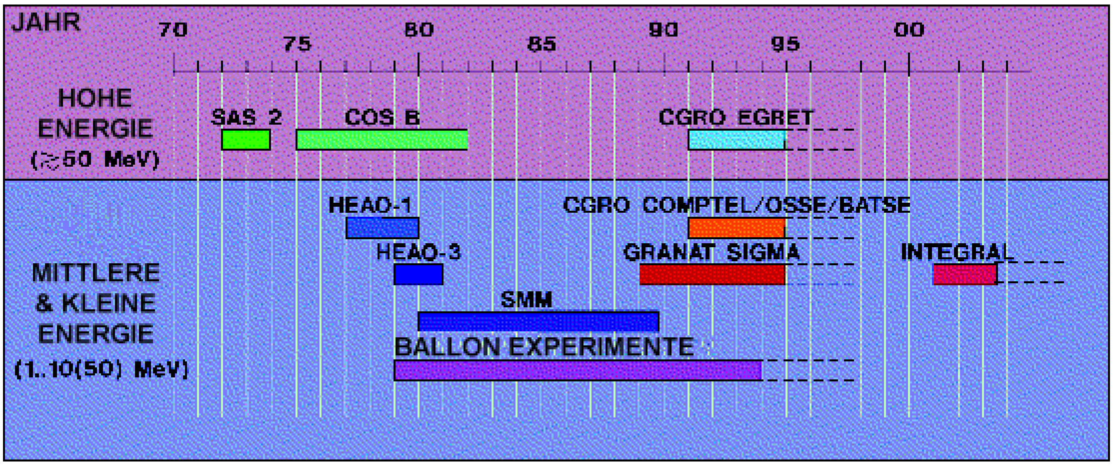

Abbildung 2: Ein Überblick über einige astronomische Gammastrahlen-Experimente15

Viel besser war die Winkelauflösung des am 5. April 1991 gestarteten *Compton Gamma Ray Observatory* (CGRO). Mit 17 Tonnen ist es die schwerste astrophysikalische Nutzlast, die je geflogen wurde. Es wurde vom *Space Shuttle Atlantis* in eine 450 km hohe Umlaufbahn gebracht.

So hat sich die Gamma-Astronomie in den letzten 20 Jahren enorm entwickelt. Da gerade die ausgesandte Gammastrahlung Rückschlüsse auf die energiereichsten Prozesse im Universum zuläßt, trägt dieser Zweig der Astronomie zum Verständnis eben der Vorgänge bei, die sich sonst völlig dem Bereich der Laborphysik entziehen.

---

11 COS-B untersuchte von August 1975 bis April 1982 intensiv die galaktische Ebene im Energieband von 50 MeV bis zu 5 GeV.

12 Kniefen, et al., Nature, Bd. 269, 1974, Seite 494.

13 Thompson, et al., Astrophysical Journal, (Letters), Bd. 200, 1975, Seite L79.

14 Swanenburg, et al., Nature, Bd. 275, 1978, Seite 298.

15 Aus: http://esapub.esrin.esa.it/bulletin/bullet79/clause79.htm, August 1998.

Der Satellit INTEGRAL wird die astronomischen Gammastrahlen-Experimente weiterführen, die in der Abbildung 2 „Ein Überblick über einige astronomische Gammastrahlen-Experimente“ aufgeführt sind.

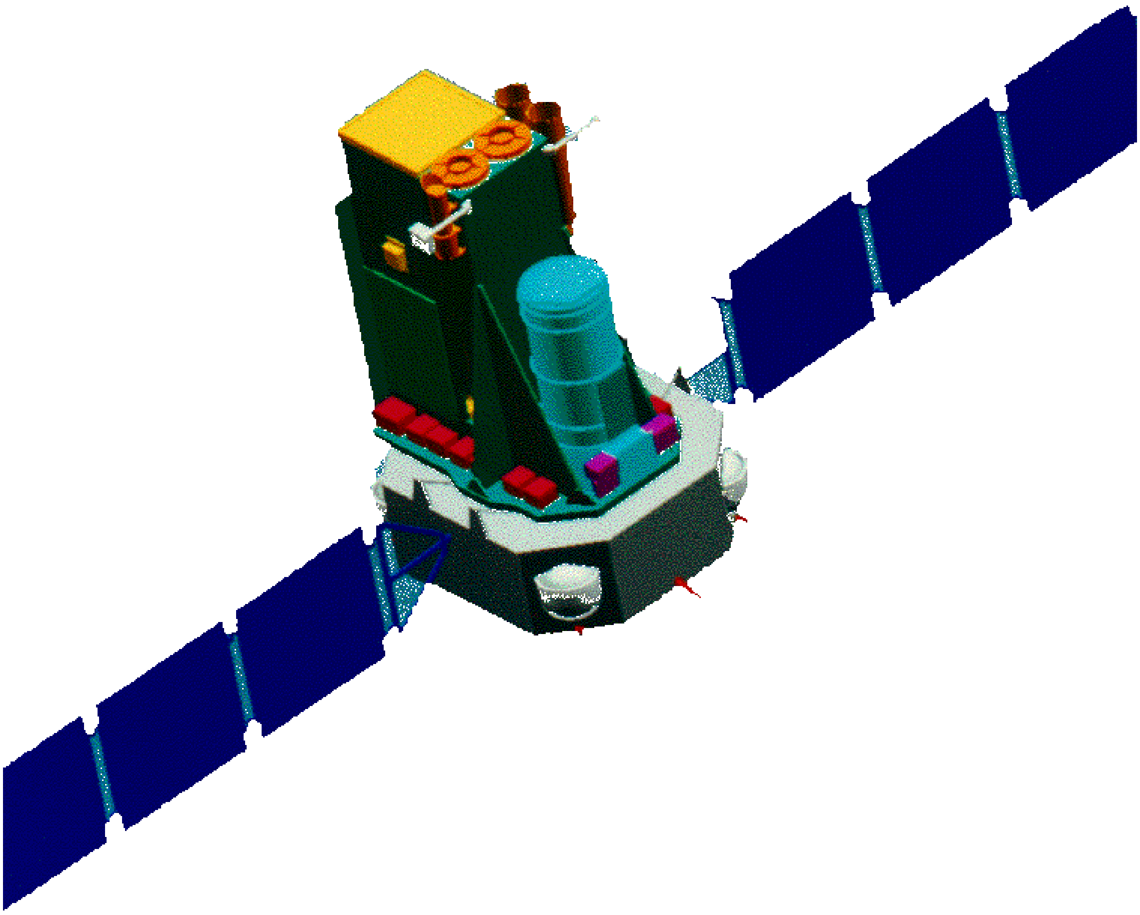

Abbildung 3: Der Satellit INTEGRAL mit ausgefahrenen Solarpanelen[16]

1.3 Satellit INTEGRAL

Der Satellit INTEGRAL (siehe Abbildung 3 „Der Satellit INTEGRAL mit ausgefahrenen Solarpanelen“) ist die nächste mittelgroße Mission der ESA. Sie ist eine Plattform für die Gamma-Astronomie im Bereich bis zu 10 MeV mit hochauflösender Spektroskopie und Abbildungen mit hoher Winkelauflösung. Die wissenschaftliche Nutzlast besteht aus einer integrierten, sich gegenseitig ergänzenden Anordnung, die darauf op-

16 Aus: http://astro.estec.esa.nl/SA-general/Projects/Integral/integral.html, August 1998.

timiert wurde, die wissenschaftlichen Ziele der Mission zu erfüllen. Die Instrumente sind Seite an Seite angeordnet und beobachten gleichzeitig alle Objektklassen.

Die Kombination der Instrumente des Satelliten INTEGRAL ermöglicht es, bei verschiedenen Energien, von etwa 1 eV (OMC) bis 10 MeV (IBIS), über sieben Größenordnungen hinweg, eine Vielzahl sehr energiereicher Objekte im Universum zu beobachten. Dies sind zum Beispiel aktive Galaxien (AGNs), Supernova Explosionen, aktive binäre Systeme, Schwarzloch-Kandidaten, *Transients* und *Gamma Ray Bursts* (GRB). Diese Breitband-Beobachtungen sind besonders bei der Hochenergie-Astrophysik wichtig.

1.3.1 Instrumente

Nach den Ergebnissen der CGRO Mission erschien es notwendig, die Abbildungsmöglichkeiten und im Besonderen die Winkelauflösung und die Quellenauflösung bei einer hohen Empfindlichkeit über ein großes Energieband zu verbessern. Dies ist beim Satelliten INTEGRAL, wie es in Tabelle 1 „Einige Parameter der Instrumente“ zu erkennen ist, gelungen.

Tabelle 1: Einige Parameter der Instrumente[17]

Was	SPI	IBIS	JEM-X	OMC
Gesichtsfeld Fully Coded Field of View (FCFOV)	16°	9°	4,8°	5°
Winkelauflösung	2°	12'	3'	20'
Spektrale Auflösung (E/E)	0,2% @ 1 MeV	7% @ 100 keV 6% @ 1 MeV	19% @ 6 keV	
Energieband	20 keV-8 MeV	15 keV-10 MeV	3 keV-35 keV	550 nm-850 nm

Das Spektrometer SPI dient zur Analyse von Gammastrahlungs-Quellen im Energiebereich von 20 keV bis 8 MeV. Es ist ortsauflösend und besteht aus 19 hexagonal geformten Germanium-Detektoren. Um die Ereignisrate durch den Hintergrund zu minimieren, wird der Detektor aktiv abgeschirmt. Diese aktive Abschirmung reicht vom Detektor bis fast hinauf zur Maske. Die Apertur wird mit dieser Abschirmung auf 30° begrenzt.

Der *Imager* IBIS ist ein Instrument für die abbildende Gamma-Astronomie. Die spektrale Empfindlichkeit reicht von 15 keV bis 10 MeV. Die instrumentelle Winkelauflösung beträgt ungefähr 12'. Je länger jedoch eine Punktquelle beobachtet wird und je

---

17 Siehe ESA, INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 2.8.

heller diese ist, desto besser kann mit Hilfe statischer Methoden ihre Position bestimmt werden. Auf diese Weise kann die Position einer Punktquelle mit einer Genauigkeit von etwa 1' bestimmt werden.

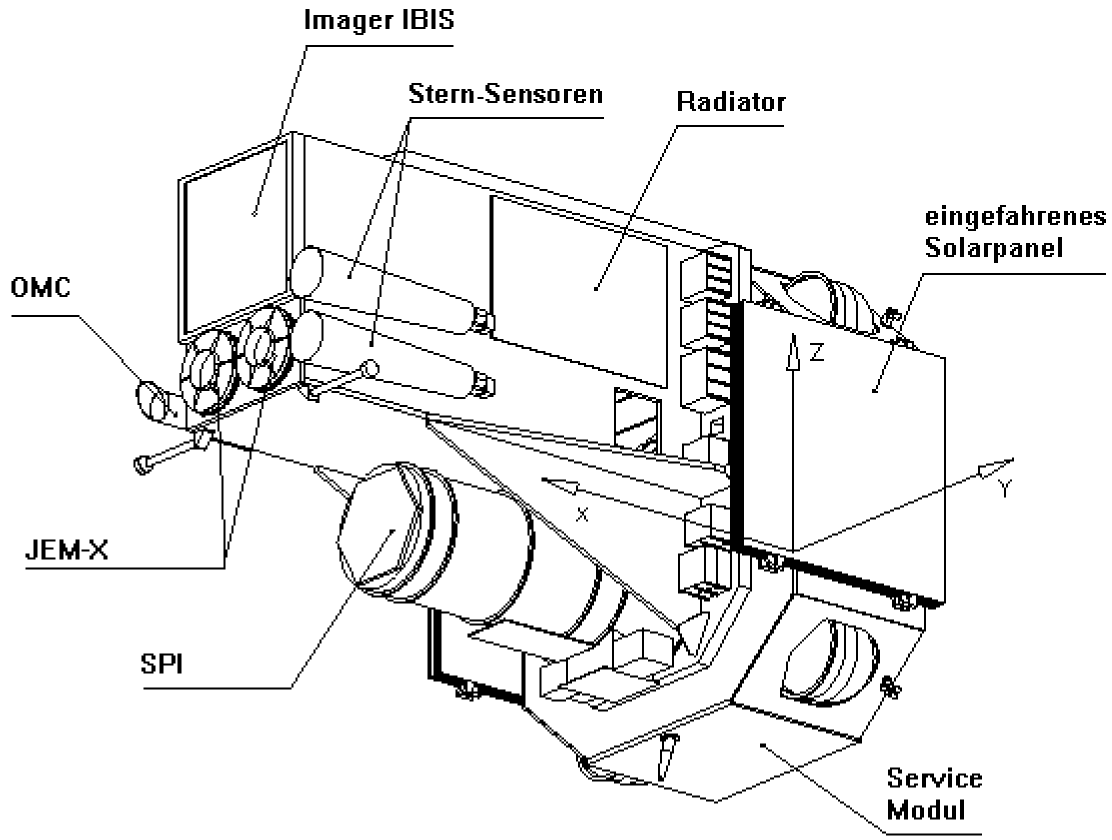

Abbildung 4: Der Satellit INTEGRAL, schematisch[18]

Der Röntgen-Monitor JEM-X detektiert und identifiziert die Gammastrahlungs-Quellen im Röntgenbereich und bietet zusätzliche Analyse- und Beobachtungsmöglichkeiten. Die beiden Detektoren des JEM-X sind ortsauflösende Microstrip-Hochdruck-Gaskammern. Die Gaskammern sind mit Xenon gefüllt. Beide Detektoren beobachten den Himmel durch eine kodierte Aperturmaske und befinden sich etwa 3,2 m oberhalb der Beobachtungsebene.

Der optische Monitor OMC ergänzt, wie der JEM-X, die beiden Hauptinstrumente. Er beobachtet die optische Komponente der beobachteten Röntgen- und Gammastrahlungs-

18 Aus: ESA, INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 2.11.

Quellen. Damit ergibt sich mit dem Satelliten INTEGRAL zum ersten Mal die Möglichkeit, gleichzeitig Photonen-Energien von etwa 1 eV (OMC) bis 10 MeV (IBIS) also über sieben Größenordnungen hinweg zu beobachten.

Darüber hinaus haben bis auf den optischen Monitor alle Instrumente eine gute zeitlich Auflösung. Die räumliche Anordnung der genannten Instrumente und einiger weiterer Details, wie der Radiator, die zwei Stern-Sensoren, das Service Modul und die eingefahrenen Solarpanelen, ist in der folgenden Abbildung zu sehen.

1.3.2 Service Modul

Die oben erwähnte wissenschaftliche Nutzlast ist auf das Service Modul montiert, welches von der ESA bereitgestellt wird. Um Entwicklungszeiten und Kosten zu sparen, wurde ein ähnliches Service Modul wie bei dem schon vorher geplanten Röntgensatelliten XMM gewählt, der 1999 gestartet werden soll. Auch die Entwicklung dieses Satelliten findet unter Mitwirkung des Instituts für Astronomie und Astrophysik in Tübingen statt.

Das Service Modul stabilisiert den Satelliten in allen drei Achsen. Das Koordinatensystem dieser drei Achsen ist rechtshändig orthogonal. Die x-Achse liegt parallel zur optischen Achse, die z-Achse zeigt in Richtung der Sonne, und die y-Achse vervollständigt das Koordinatensystem. Der absolute Pointierungs-Fehler ist in der y- und z-Achse kleiner als 5' und in der x-Achse (Rollen) kleiner als 15'.

Das Leergewicht des Service Moduls beträgt 3.600 kg, das der wissenschaftlichen Nutzlast 2.087 kg. Die wissenschaftliche Nutzlast verbraucht eine Leistung von 721 W. Für die Übertragung der wissenschaftlichen Daten und der Daten für die Überwachung wesentlicher Parameter des Satelliten (*Housekeeping*) zu den Bodenstationen stehen etwa 85,8 kbits/s[19] zur Verfügung.

Der Satellit INTEGRAL, dessen nominale Arbeitsdauer zwei Jahre beträgt und der für eine maximale Arbeitsdauer von fünf Jahren konstruiert wurde, hat mit eingefahrenen Solarpanelen die Dimensionen 4 m x 4 m x 6 m (Länge x Breite x Höhe).

1.3.3 Start und Umlaufbahn

Der Satellit INTEGRAL ist so konstruiert, daß ein Start sowohl mit einer russischen PROTON, als auch mit der europäischen ARIANE 5 möglich ist. Die ARIANE 5 ist dabei als Ausweichsystem vorgesehen ist, falls ein Start mit der PROTON nicht möglich sein sollte.

---

19 Siehe ESA, INTEGRAL System Requirements Document, Issue 2 Revision 1, 25.5.98, Seite 18.

Wie die folgende Tabelle 2 „Die Start- und Umlaufbahn-Parameter“ zeigt, führen die unterschiedlichen Trägersysteme und Startplätze zu unterschiedlichen Parametern für die Umlaufbahnen. Grundsätzlich gilt für jede gewählte Umlaufbahn, daß sie die Zeit außerhalb des Strahlungsgürtels, die Beobachtungsmöglichkeiten und den Kontakt mit den Bodenstationen maximiert und gleichzeitig die durchschnittliche Strahlendosis und den Protonenfluß minimiert.

Im Gegensatz zu einem Start mit der PROTON durchquert der Satellit bei einem Start mit der ARIANE 5 zweimal pro Umlauf den Van Allen-Gürtel[20]. Dabei müssen vor allem die empfindlichen Detektoren und einige weiteren Instrumente deaktiviert werden. Deshalb ist ein Start mit der PROTON vom wissenschaftlichen und technischen Standpunkt aus vorzuziehen. Außerdem erlaubt die erreichbare Umlaufbahn eine bessere Durchmusterung des Himmels[21].

Tabelle 2: Die Start- und Umlaufbahn-Parameter[22]

Was	PROTON	ARIANE 5
Umlaufbahn	HEO *(Highly eccentric orbit)*	HEO *(Highly eccentric orbit)*
Bodenstationen	Redu (Belgien), Goldstone (USA)	Redu (Belgien)
Umlaufzeit	72 h	48 h
Inklination	51,6°	65,0°
Perigäum	10.000 km	7.000 km
Apogäum	153.000 km	114.000 km
Zeit über 40.000 km	~90%	~85%

1.3.4 Pointierung und Dithering

Eine Besonderheit des Satelliten INTEGRAL ist das Pointierungs-Muster. Um den systematischen Abbildungsfehler des Spektrometers SPI zu minimieren, wird bei Punktquellen zunächst auf die Quelle selbst und dann in einem Sechseck um die Quelle herum pointiert.

Dieses „Zittern“ um die Quelle wird als *Dithering* bezeichnet. Bei ausgedehnten oder mehreren Quellen wird ein Pointierungs-Muster mit 25 Punkten eines 5 x 5 Quadrates verwendet. Die Punkte sind, wie in Abbildung 5 „Das Dithering-Muster und das Gesichtsfeld der Instrumente“ zu sehen ist, etwa je 2° voneinander entfernt.

20 Van Allen, James Alfred, amerikanischer Physiker, in: Bibliographisches Institut, Das neue Duden Lexikon, 2. Auflage, Bibliographisches Institut Mannheim, Wien, Zürich 1989.

21 Siehe ESA, INTEGRAL System Requirements Document, Issue 2 Revision 1, 25.5.98, Seite 10 ff.

22 Siehe ebd., Seite 10.

Bei dieser Abbildung sind die Gesichtsfelder der Instrumente überlagert dargestellt. Die ausgefüllten Punkte sind die des *Dithering*-Musters für punktförmige Quellen. Auf jeden dieser Punkte wird während einer Beobachtung nacheinander pointiert. Die Pointierung des Satelliten auf den nächsten Punkt des *Dithering*-Musters wird *Slew* genannt. Die nicht ausgefüllten Punkte sind die des *Dithering*-Musters für ausgedehnte Quellen.

Die anderen Instrumente führen diese Bewegung ebenfalls aus. Dies bedeutet zum Beispiel, daß sich für eine gegebene Beobachtung die Größe des vollständig abgebildeten Gesichtsfeldes des *Imager* IBIS verringert. Die Pointierungs-Muster wurden so gewählt, daß die Hauptquelle innerhalb des Gesichtsfeldes des *Imager* IBIS bleibt, um einen Intensitätsverlust zu vermeiden.

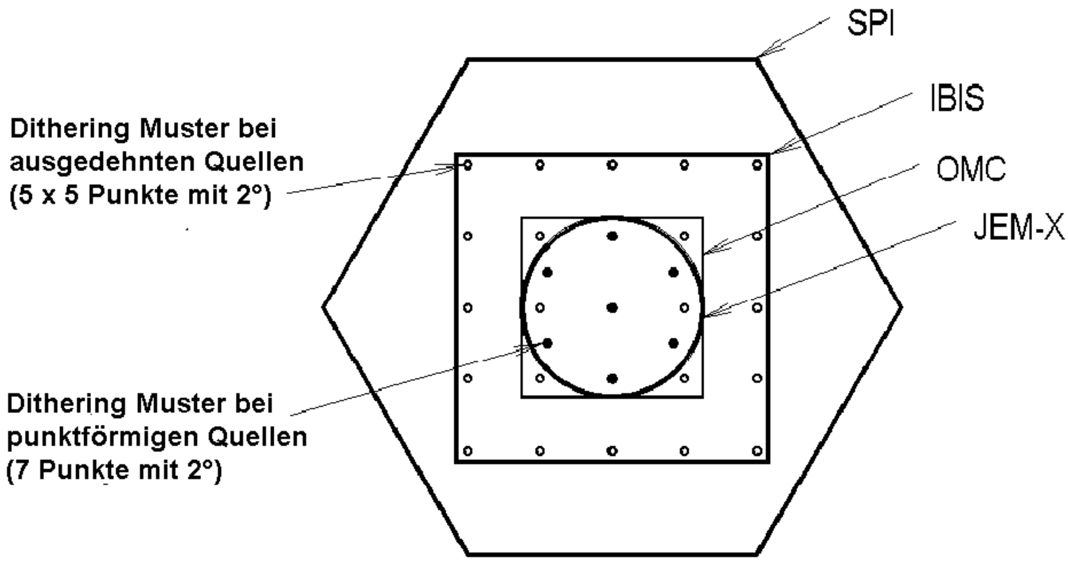

Abbildung 5: Das Dithering-Muster und das Gesichtsfeld der Instrumente[23]

1.3.5 Kodierte Aperturmasken

Alle Instrumente, bis auf den optischen Monitor, verwenden kodierte Aperturmasken (*Coded Masks*). Denn obwohl sich bei mittleren Röntgenenergien mit Hilfe von Spiegeln und optischen Prinzipien durchaus Abbildungen erzeugen lassen, wird es etwa ab 10 keV schwierig, geeignete Spiegel herzustellen.

23 Aus: ESA, INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 3.14.

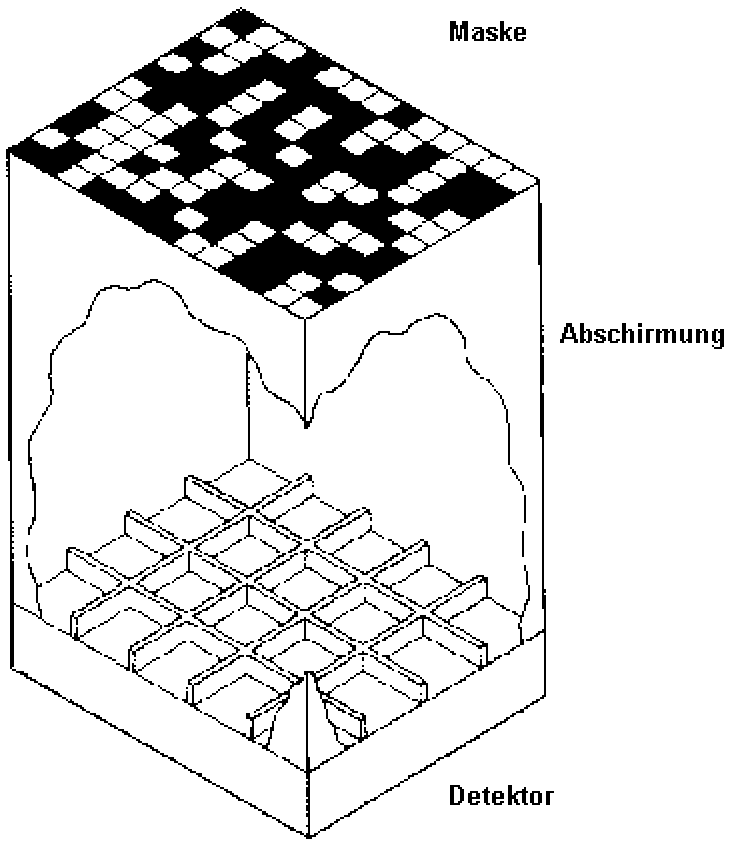

Abbildung 6: Die kodierte Aperturmaske, dreidimensional[24]

Es ist jedoch möglich, Abbildungen nach dem Prinzip der *Camera Obscura*, also der Lochkamera, zu erzeugen. Bei der Gamma-Astronomie ist der vorhandene Fluß dafür jedoch zu gering.

Daher ist es sinnvoller, Masken, wie in Abbildung 6 „Die kodierte Aperturmaske, dreidimensional“, als Öffnung zu verwenden. Diese Masken bestehen im Wesentlichen aus einem bestimmten replizierten Muster von Flächen. Die Flächenstücke dieser Muster sind für das jeweils zu beobachtende Energieband entweder durchlässig oder undurchlässig. Eine Punktquelle erzeugt dann auf dem Detektor eine Projektion des Musters. Aus der gewonnenen Intensitätsverteilung kann mit verschiedenen Rekonstruktions-Verfahren innerhalb des Gesichtsfeldes eindeutig auf die Quellen und deren Stärke geschlossen werden.

Der Detektor sammelt die Projektionen der beobachteten Quellen. Es ist aber zu beachten, wie obige Abbildung 7 „Die kodierte Aperturmaske, zweidimensional“ zeigt, daß jeder Punkt auf dem Detektor Projektionen verschiedener Quellen detektieren kann.

24 Aus: S. Bergeson-Willis et al., Integral, Report on the Phase-A Study, Appendix X.

Nach einer bestimmten Belichtungszeit wird aus den gesammelten Daten ein Bild des Himmels rekonstruiert.

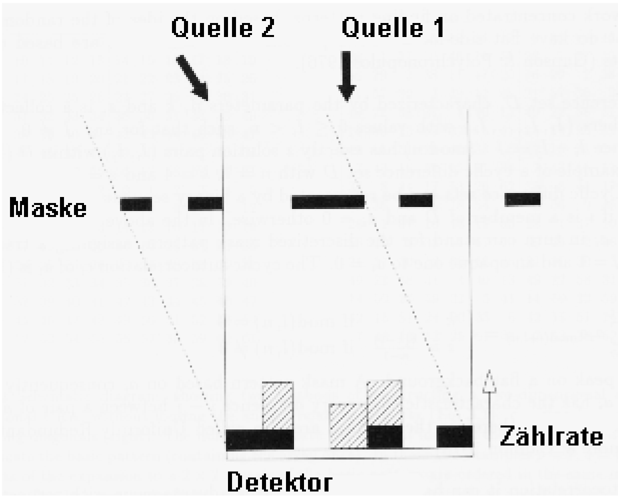

Abbildung 7: Die kodierte Aperturmaske, zweidimensional[25]

1.3.6 Datenverarbeitung

Die Datenverarbeitung, das *On Board Data Handling System* (OBDH), an Bord des Satelliten wird von der *Central Data Management Unit* (CDMU) kontrolliert. Diese sammelt die von den einzelnen Instrumenten bereitgestellten Daten, verteilt die Telekommandos und überwacht den Satelliten. Weiterhin stellt sie allen Instrumenten eine Zeitbasis und ein Signal des *Attitude and Orbit Control System* (AOCS) zur Verfügung.

Unter Telekommandos versteht man in diesem Zusammenhang Anweisungen an die Instrumente. Diese Telekommandos werden im Wesentlichen per Funk an den Satelliten geschickt, manche werden jedoch an Bord generiert.

Die Datenverarbeitungsrechner (DPEs) der einzelnen Instrumente verpacken die jeweiligen wissenschaftlichen Daten und das *Housekeeping* in Telemetrie-Pakete. Die DPEs sind intern nicht redundant, deshalb hat jedes Instrument, außer dem OMC, zwei DPEs. Die Kommunikation findet über einen gemeinsamen Datenbus statt.

Zwei *Remote Terminal Units* (RTUs) sind an den Datenbus angeschlossen. Sie kontrollieren und überwachen bestimmte Parameter, wie zum Beispiel die Temperatur der Instrumente, auch wenn diese abgeschaltet sind. Sie schalten auch die Spannungs-Versorgungen der Instrumente ein oder aus.

---

25 Aus: http://www.sron.ruu.nl/~jeanz/cai/coded_intr.html, August 1998.

1.3.7 Telemetrie

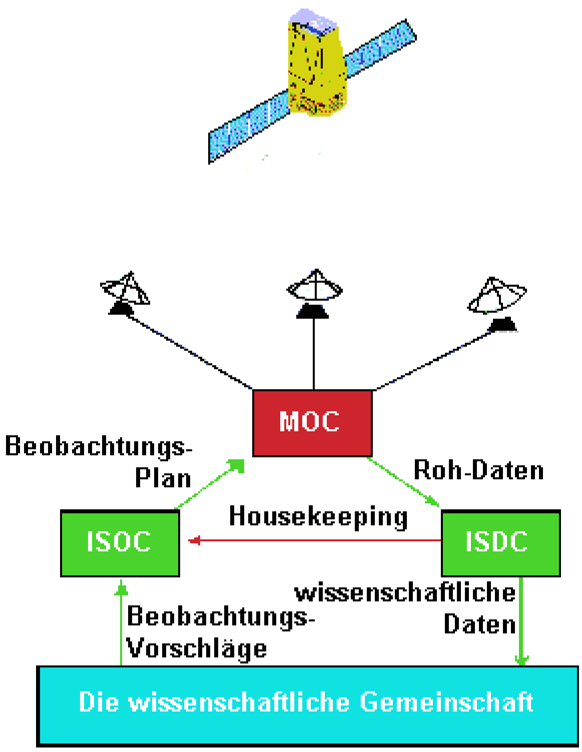

Abbildung 8: Der Datenfluß[26]

Die Beobachtungsdaten werden dann zur Erde übertragen. Dazu werden sie von der CDMU von den Instrumenten abgerufen. Die Daten werden je nach Position des Satelliten INTEGRAL entweder in Goldstone (USA) oder in Redu (Belgien) empfangen und von dort nach Darmstadt zum *Mission Operational Center* (MOC) weitergeleitet. Das MOC überwacht die Funktion des Satelliten und weitere essentielle *Housekeeping*-Daten. Von dort werden auch die Telekommandos an den Satelliten geschickt. Abschließend gelangen sie zum wissenschaftlichen Datenzentrum *INTEGRAL Science Data Center* (ISDC) in Versoix (Schweiz).

Unter Berücksichtigung der Charakteristik des jeweiligen Instruments werden beim ISDC die Rohdaten in physikalische Einheiten umgewandelt und einige Standard Analysen erstellt. Diese Daten werden zunächst der jeweiligen Beobachtungsgruppe zur

26 Aus: ESA, INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 2.1.

Verfügung gestellt. Später werden sie dann in einem weltweit abrufbaren Archiv gespeichert.

Das Integral Science Operations Center (ISOC) in Noordwijk (Holland) hingegen erstellt aus den akzeptierten Beobachtungsvorschlägen einen optimierten Beobachtungsplan. Dabei werden die von zu Zeit zu Zeit notwendigen Operationen des Servicemoduls berücksichtigt. Dieser Beobachtungsplan enthält unter anderem einen Zeitplan für die Reihenfolge der zu beobachtenden Quellen und die dabei zu verwendenden Konfiguration der Instrumente. Siehe dazu die Abbildung 8 „Der Datenfluß“ auf Seite 22.

Nach diesem Überblick über den Satelliten INTEGRAL wird im Folgenden genauer auf den *Imager* IBIS, für den die HEPI entwickelt wurde, eingegangen.

1.4 Imager IBIS

Das Institut für Astronomie und Astrophysik in Tübingen beteiligt sich vielfältig an der Entwicklung des *Imager* IBIS. Neben der Entwicklung der HEPI wird an diesem Institut auch die Programmierung der DPE des *Imager* IBIS vorgenommen. Diese Beteiligung zeigt sich unter anderem in der Vielzahl der Diplomarbeiten, die in diesem Umfeld entstanden sind oder gerade entstehen: im Rahmen dieser Diplomarbeiten wurden zum Beispiel ein Detektor-Simulator und ein HEPI-Simulator[27] entwickelt, um die entwikkelten Komponenten unabhängig voneinander testen zu können. Desweiteren wurde ein Detektor System[28] gebaut, um die HEPI mit realen, nicht simulierten Daten testen zu können.

Wesentlich für den *Imager* IBIS sind seine beiden Detektor-Ebenen: die der Beobachtungsquelle zugewandte Detektor-Ebene heißt *INTEGRAL Soft Gamma Ray Imager* (ISGRI) und die andere, darunterliegende Detektor-Ebene heißt *Pixelated Imaging Cesium Iodide Telescope* (PICSIT). Diese Anordnung ermöglicht die Beobachtung bei einer hohen durchgängigen Empfindlichkeit in einem großen Spektralbereich mit guter spektraler Auflösung. Die jeweils acht Module beider Detektor-Ebenen werden in zwei übereinander liegenden, gitterförmigen Rahmenebenen montiert, wie sie Abbildung 9 „Der Imager IBIS“ aufzeigt. PICSIT und ISGRI werden im Folgenden auch Cäsium-Jodid-Detektor und Cadmium-Tellurid-Detektor genannt.

27 Siehe Decker, Oliver, Entwicklung einer Software Simulation des Hardware Event Pre-Processors des INTEGRAL Satelliten, IAAT 1998.

28 Siehe Rilk, Albrecht J., Detektor System zum Test eines Hardwareprozessors des Satelliten INTEGRAL, IAAT 1996.

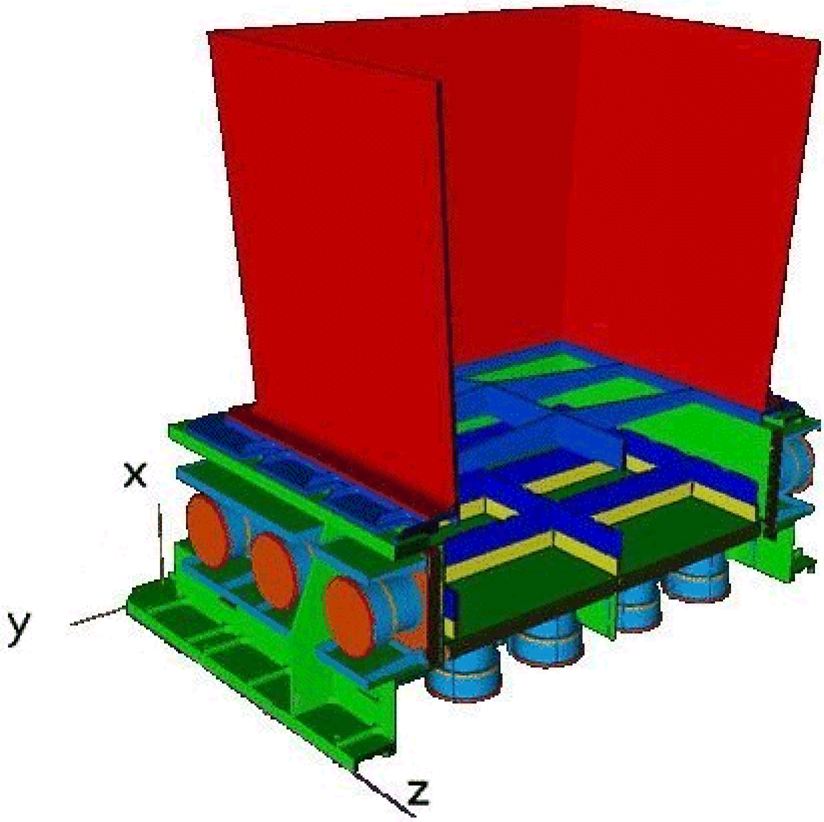

Abbildung 9: Der Imager IBIS[29]

Die Detektoren generieren für jedes detektierte Photon Daten, die digitalisierte Informationen über den Ort, die Zeit und die Amplitude des Detektions-Ereignisses enthalten.

Diese Daten werden über eine Schnittstelle, je eine pro Detektor, von der HEPI angefordert. Dann verarbeitet die HEPI diese Daten und stellt sie der DPE zur Verfügung, die sie ihrerseits verarbeitet und in Telemetrie-Pakete verpackt.

1.4.1 Kodierte Aperturmaske

Auch der *Imager* IBIS hat eine kodierte Aperturmaske, deren Prinzip in Kapitel 1.3.5 „Kodierte Aperturmasken“ auf Seite 19 erläutert wurde. Sie besteht aus Wolfram-Elementen und ist auf eine hohe Winkelauflösung hin optimiert. Die Diffraktion der Gammastrahlen ist bei einer Apertur dieser Größe zu vernachlässigen. Die Auflösung wird deshalb entweder durch die räumliche Auflösung der Detektoren oder der Auflösung (a), die durch die Geometrie gegeben ist, begrenzt. Die durch die Geometrie gege-

29 Aus IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 53.

bene Auflösung ist definiert durch das Verhältnis der Fläche (f) eines Maskenelementes zu der Gesamtfläche (F) und dem Abstand zu den Detektoren (d). So gilt:

$$a = \frac{f}{F * d}$$

Gleichung 1: Auflösung kodierter Aperturmasken

1.4.2 Cadmium-Tellurid-Detektor

Um die Gammastrahlung über einen breiten Energiebereich mit einer hohen Empfindlichkeit beobachten zu können, werden zwei Detektor-Ebenen verwendet. Die den Beobachtungsobjekten zugewandte Detektor-Ebene besteht aus 16.384 Cadmium-Tellurid-Halbleiter-Zählern (CdTe) vom Typ II-VI. Im Folgenden werden diese Zähler auch kurz Pixel genannt. Diese sind für den niederen Energiebereich sehr gut geeignet. Jeder dieser Halbleiter-Zähler hat eine Fläche von 4 x 4 mm^2 und eine Stärke von 2 mm. Bei 150 keV Photonen Energie beträgt die Effizienz nur noch 50%. Der Arbeitsbereich reicht von 15 keV bis zu 700 keV. Je 16 dieser Halbleiter-Zähler sind zu Polyzellen und je 128 von diesen Polyzellen sind wiederum zu Modulen zusammengefaßt.

Der Detektor besteht aus acht dieser Module und bietet so eine gute räumliche Auflösung. Die spektrale Auflösung liegt bei etwa 5 bis 6%. Die Polyzellen werden von je einem ASIC kontrolliert. Die gesamte empfindliche Oberfläche ist 2.621 cm^2 groß. Die optimale Arbeitstemperatur liegt zwischen 0 und 10° Celsius. Während der Beobachtungen werden beide Detektor-Ebenen auf einer Temperatur von etwa 10° Celsius gehalten[30]. Werden im Cadmium-Tellurid-Detektor mehrere Ereignisse gleichzeitig registriert, überträgt er diese nicht an die HEPI.

30 Siehe IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 66.

1.4.3 Cäsium-Jodid-Detektor

Die Photonen, die den Cadmium-Tellurid-Detektor passiert haben, können vom Cäsium-Jodid-Detektor (CsI) nachgewiesen werden. Dieser hat einen Arbeitsbereich von 100 keV bis 10 MeV bei einer spektralen Auflösung von etwa 6 bis 7%[31]. Er besteht aus 4.096 Cäsium-Jodid-Halbleiter-Zählern vom Typ I-VII in einem 64 x 64 Quadrat. Jeder dieser Zähler hat eine Kantenlänge von 8,65 mm und eine Stärke von 30 mm. Je 16 dieser Halbleiter-Zähler sind zu Polyzellen und je 32 von diesen Polyzellen sind wiederum zu Modulen zusammengefaßt. Wie der Cadmium-Tellurid-Detektor besteht dieser Detektor aus acht Modulen. Er bietet eine gute räumliche Auflösung, die aufgrund seiner geringeren Anzahl an Halbleiter-Zählern etwas schlechter ist, als die des Cadmium-Tellurid-Detektors. Auch diese Polyzellen werden durch je einen ASIC kontrolliert. Die gesamte empfindliche Oberfläche umfaßt 3.065 cm^2.

Die beiden Detektoren sind sich vom Aufbau her sehr ähnlich, wobei der Cäsium-Jodid-Detektor natürlich aufgrund seiner geringeren Anzahl an Halbleiter-Zählern auch eine geringere Ortsauflösung hat.

Bei beiden Detektoren werden zur Minimierung von Hintergrundereignissen auch all jene Ereignisse verworfen, also nicht an die HEPI übergeben, die gleichzeitig mit einem Signal des aktiven Hintergrundschildes registriert werden. Der aktive Hintergrundschild wird im anschließenden Kapitel 1.4.4 „Aktiver Hintergrundschild“ beschrieben.

Um die Arbeit der HEPI zu vereinfachen, hat jeder der beiden Detektoren eine Einrichtung, die die registrierten Ereignisse nach ihrer Ankunftszeit sortiert. Diese werden dann der HEPI in einem Puffer zur Verfügung gestellt.

1.4.4 Aktiver Hintergrundschild

Die Detektoren sind von einem Wismut-Germanium-Schild umgeben, welches die Seiten der Detektoren sowie deren Unterseite umfaßt. Der Schild ist aus einzelnen Modulen aufgebaut. Diese sind in der Abbildung 9 „Der Imager IBIS“ auf Seite 24 als zylindrische Bauteile an den Seiten und unten zu erkennen.

Die Wismut-Germanium-Kristalle werden von Photo-Multipliern beobachtet. Wird in einem Kristall ein Ereignis registriert, erzeugt der Photo-Multiplier ein analoges Signal, welches als Vetosignal zu beiden Detektoren geleitet wird. Der aktive Hintergrundschild verringert die Anzahl der Ereignisse, die durch die kosmische, diffuse Hintergrund-Gammastrahlung und die an Bord erzeugte Gammastrahlung hervorgerufen werden,

31 Siehe IBIS Team, Scientific Performance Report, Issue 3 Revision 1, 30.6.98, Seite 2.

soweit, daß die Anzahl der durch andere Hintergrundstrahlung erzeugten Ereignisse überwiegt. Auf Grund der hohen Dichte und Atomzahl der Wismut-Germanium-Kristalle ist für den Schild eine Stärke von etwa 20 mm ausreichend.

1.4.5 Kalibrations-System

Der *Imager* IBIS besitzt ein Kalibrations-System mit einer Natrium-22-Quelle, die in einem 60 mm breiten Wismut-Germanium-Szintillator untergebracht ist. An diesem Szintillator befinden sich zwei Photo-Multiplier. Ein zusätzlicher Schild aus Wolfram ist in Richtung des Spektrometers und des Röntgenmonitors angebracht, um eine ausreichende Abschirmung zu gewährleisten. So wird kein zusätzlicher Hintergrund erzeugt. Im Wismut-Germanium-Schild befindet sich eine konische Apertur, um die Emission der Gammastrahlung zur Detektor-Ebene zu ermöglichen.

Die erzeugte Gammastrahlung hat eine Energie von 511 keV und 1.275 keV. Jedesmal wenn diese vom Photo-Multiplier im Szintillator registriert wird, schickt die Elektronik des Kalibrations-Systems ein Signal zu den Detektoren. Jedes Ereignis, das gleichzeitig mit diesem Signal in den Detektoren registriert wird, stammt normalerweise von einem Photon aus der Kalibrations-Quelle und wird als solches gekennzeichnet. Mit Hilfe dieses Kalibrations-Systems ist es möglich, den *Imager* IBIS während des Fluges zu eichen[32].

1.4.6 Passiver Kollimator

Zusätzlich zur Abschirmung des aktiven Hintergrundschildes ist eine weitere Abschirmung notwendig. Der Kollimator ist ein passiver Schild. Er verkleinert den sichtbaren Himmelsabschnitt, den die kodierte Maske komplett abbildet. Er besteht im Wesentlichen aus drei Teilen:

1. einer Art Trichter, dem sogenannten *Hopper*. Er wird aus vier Wänden gebildet, die an den Detektoren befestigt sind. In diese Wände ist eine Wolfram-Folie eingebettet. Siehe auch die Abbildung 9 „Der Imager IBIS“ auf Seite 24;
2. einer Konstruktion, der sogenannten *Tube*. Diese umgibt den *Hopper* und wird aus vier Wänden gebildet, die zum Nutzlastmodul gehören. In diese Wände ist eine Bleifolie eingebettet oder festgenietet;
3. einer 1,2 mm starken Wolfram-Folie. Diese schließt die Spalten zwischen den Kanten der kodierten Maske und den *Tube*-Wänden.

---

32 Siehe IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 3.

1.4.7 Datenverarbeitung

Die DPE stellt die zentrale Verarbeitungselektronik des *Imager IBIS* dar. Sie verfügt über einen 1.750 Prozessor, 2 MByte Speicher und Schnittstellen unter anderem zu dem OBDH, der HEPI und den Detektoren. Sie übernimmt die komplexere Datenverarbeitung bei niedriger Datenrate vor allem von Ereignissen des Cadmium-Tellurid-Detektors.

Die DPE empfängt die wissenschaftlichen und *Housekeeping*-Daten aus den Detektoren, dem aktiven Hintergrundschild und dem Kalibrations-System. Sie verpackt diese Daten in Telemetrie-Pakete und stellt sie dem OBDH zur Verfügung. Von der DPE werden die Telekommandos, das Taktsignal und die Synchronisations-Signale verteilt. Sie kontrolliert die verschiedenen Bauteile des Instruments.

Nach diesem Überblick über den Satelliten INTEGRAL und den *Imager* IBIS wird in Kapitel zwei erläutert, welche Aufgaben die HEPI als Teil der Datenverarbeitung des *Imager* IBIS hat und mit welchen Mitteln die HEPI konstruiert wurde.

2 Aufgabe und Realisierung

Es gibt zwei Hauptgründe für die Notwendigkeit eines speziell entwickelten Mikroprozessors für den *Imager* IBIS: aufgrund der sehr hohen Datenraten des Cäsium-Jodid-Detektors von etwa 15.000 Ereignissen/s[33] würde

1. die theoretische Grenze der Verarbeitungsgeschwindigkeit der DPE erreicht und
2. die zur Verfügung gestellte Telemetrie-Rate um den Faktor 40 überschritten werden[34].

Die Detektor-Daten müssen also wissenschaftlich aufbereitet werden. Je nach Forschungsschwerpunkt und Status des Satelliten kann die HEPI verschiedene Aufgaben wahrnehmen und dabei die Beobachtungsdaten gezielt auswählen und zusammenfassen.

Zu den weiteren Aufgaben der HEPI gehören unter anderem das Synchronisieren der Detektoren und das Aufbereiten der Datenstrukturen für die DPE. So untersucht die HEPI die Daten aus beiden Detektoren auf Zeitkoinzidenzen und erstellt Histogramme aus Daten des Cäsium-Jodid-Detektors. Für die Erstellung der Histogramme wird mehr Speicher benötigt, als die DPE zur Verfügung hat; auch das spricht für die spezielle Entwicklung der HEPI. Eine Erklärung, was in diesem Zusammenhang unter Histogrammen zu verstehen ist, findet sich im Kapitel 2.1.7 „Histogramme“ auf Seite 38.

2.1 Realisierung

2.1.1 Entwicklungskonzept

Bei der Entwicklung und Realisierung eines Satelliten können sich die Aufgaben und die Rahmenbedingungen dynamisch ändern. Für die Lösung der gestellten Aufgaben war es notwendig, ein geeignetes Entwicklungskonzept zu entwerfen. Auf der elektroni-

33 Siehe IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 84 ff.

34 Siehe die Tabelle 3 „Die Datenraten“ auf Seite 30.

schen Seite wurde durch die Verwendung einer integrierten, veränderbaren Schaltung (FPGA) eine sehr hohe Flexibilität gewährleistet. Eine integrierte Schaltung ist sehr viel leichter anzupassen als eine diskret aufgebaute Schaltung. Nach dem erfolgreichen Abschluß der notwendigen Tests wird jedoch ein strahlungsfester ASIC verwendet werden, dessen innere Konfiguration nicht mehr verändert werden kann. Um auch bei der Entwicklung möglichst flexibel zu sein, war es notwendig, ein neues Entwicklungskonzept zu entwerfen und dafür eigene Werkzeuge zu erstellen.

Tabelle 3: Die Datenraten[35]

Was	Ereignisse/s	Bit/Ereignis	Bit/s
Datenrate der Detektoren zur HEPI			
CdTe-Detektor	1.220	64	78.080
CsI-Detektor	2.000 bis 25.000	64	128.000 bis 1.600.000
Summe der Detektoren zur HEPI			206.080 bis 1.678.080
Maximale theoretische Datenrate beider Detektoren zur HEPI	2 x 65.536	64	2 x 4.194.304
Datenrate der Schnittstelle A zur DPE[36]			
Cadmium Telluride Photon by Photon	980	80	78.400
Compton	240	160	38.400
Cesium Iodide Photon by Photon	1.024	160	163.840
Summe Schnittstelle A			280.640
Datenrate der Schnittstelle B zur DPE			
Übertragungszeit der Histogramme: 1.300 s			14.577
Summe der HEPI zur DPE			295.217
Maximale theoretische Datenrate HEPI zur DPE			5.000.000
Verarbeitungsgeschwindigkeit der DPE[37]			
CdTe-Detektor	1.000	80	80.000
CsI-Detektor	2.000	80	160.000
Maximale Telemetrie-Rate[38] für Imager IBIS			85.800

35 Siehe Volkmer, Reiner, HEPI Design Report, Issue 6.0, Juli 1998 IAAT, Seite 8 ff.

36 Siehe die Tabelle 4 „Die Module“ auf Seite 31.

37 Siehe Software und Manager Meeting, Minutes of Meeting, 20. Januar 1998, Tübingen, IN-IM-TUB-9801-I, Annex 3.

38 Siehe ESA, INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997, Seite 3.40.

2.1.2 Modularisierung

Bei der Entwicklung der HEPI erwies es sich als wichtig, eine Modularisierung der verschiedenen Einzelaufgaben vorzunehmen und die verschiedenen Datenpfade zu definieren. Das heißt, die Aufgaben der HEPI wurden analysiert und in voneinander unabhängige Blöcke verteilt. Damit der Datenfluß konsistent beschrieben werden kann, sind auch Bauteile, die über Schnittstellen direkt mit der HEPI verbunden sind, wie zum Beispiel die Detektoren und die DPE, als externe Module definiert. Um die notwendige Flexibilität bei der Konstruktion zu erreichen, wurde ein objektorientiertes Vorgehen beim Design der einzelnen Module und ein *Pipeline*-Konzept für die Datenverarbeitung verwendet. Eine Auflistung aller Module der HEPI befindet sich in Tabelle 4 „Die Module".

Tabelle 4: Die Module

Modulname	Modul
CdTe_HBR	Cadmium Telluride High Bit Rate Interface
CsI_HBR	Cesium Iodide High Bit Rate Interface
HBR_A	High Bit Rate Interface A
HBR_B	High Bit Rate Interface B
LBR	Low Bit Rate Interface
CSY	Clock & Synchronization
STAT	Status
CTR	Controller
LUT	LUT Memory Controller
MEM	Memory Controller
AC	Amplitude Correction
MP	Multiple Event Reconstruction and Polarimetry
TC	Time Coincidence and Time Stamp
ES	Energy Selection
SPT	Spectral Timing
HIST	Histogram

In diesem Zusammenhang ist eine *Pipeline* mit einem Fließband vergleichbar. Wie bei einem Fließband verarbeitet jedes Modul innerhalb eines *Pipeline*-Taktes ein Detektor-Ereignis und die verarbeiteten Daten werden dann an das nächste Modul übergeben. Jedes Modul ist eine sogenannte Stufe der *Pipeline*. Da die Daten aus den Detektoren aus jeweils 64 Bit zusammengesetzt sind und seriell übertragen werden, dauert die Übertragung eines Detektor-Ereignisses 64 Takte. Deshalb war es sinnvoll, den *Pipeline*-Takt auf 64 *Clock*-Takte festzulegen. Da die HEPI die Datenströme der beiden Detektoren nicht nur unterschiedlich verarbeitet, sondern auch synchronisiert, gibt es für jeden Detektor-Daten-Strom eine eigene *Pipeline*, die beide vom *Controller* gesteuert werden.

Desweiteren wurden jedem Modul durch definierte Datenformate identische Schnittstellen zugeordnet. Die Kommunikation zwischen den einzelnen Modulen ließ sich so auf Signale zur Statusänderung der zentralen Steuereinheit und den Ereignis-Daten beschränken.

Durch dieses Verfahren sind die einzelnen Module weitgehend unabhängig voneinander, und es ist während der gesamten Entwicklungsphase möglich, die einzelnen Module zu verändern, ohne daß dies die anderen Module beeinflußt. Bei Bedarf können auch ganze Module hinzugefügt oder entfernt werden. In der Tabelle 4 „Die Module“ sind alle Module aufgeführt. Die Module und die möglichen Datenströme sind in der Abbildung 13 „Die Module und der Datenfluß“ auf der Seite 42 im Überblick dargestellt.

2.1.3 Datenbank

Nach der Modularisierung der HEPI war es notwendig, ein besonderes Entwicklungskonzept zu entwerfen, um flexibel auf Entwicklungs- und Designänderungen reagieren zu können. Es ist in der Abbildung 11 „Der Designfluß des Entwicklungskonzeptes“ auf Seite 34 dargestellt.

Wichtiges Bestandteil dieses Entwicklungskonzeptes ist eine Datenbank[39], weshalb es Datenbank gestütztes *Top Down Design* genannt wird. Es galt eine Datenbank zu entwickeln, in der sowohl die einzelnen Module und deren Schnittstellen als auch die Datenpfade und Netzlisten zur Bearbeitung bereitgehalten werden. Zudem müssen größere Änderungen mit vertretbarem Aufwand und fehlerfrei möglich sein. Der Kern der Datenbank besteht aus drei Tabellen: der Tabelle der Module, der Netze und der Signale. Diese sind in der Abbildung 10 „Die drei zentralen Tabellen der HEPI-Datenbank“ dargestellt.

Das *Top Level* entspricht der Abbildung 13: „Die Module und der Datenfluß“ auf Seite 42. Es stellt die zentrale Beschreibung der HEPI innerhalb des Entwicklungssystems dar, auf das im Kapitel 2.1.6 „Schematics“ auf Seite 36 eingegangen wird. Alle Module werden über dieses *Top Level* in die Gesamtentwicklung eingebunden. Aufgrund seiner Größe von DIN A0 ist es im Anhang im Kapitel 4.5 „Technische Zeichnungen“ ab Seite 81 abgebildet.

Die vollständige Beschreibung des *Top Levels* in einer Datenbank vorliegen zu haben bringt eine Reihe von Vorteilen mit sich. Zum einen ist es möglich, durch entsprechende

39 Siehe auch Dunn, Melissa, Creating Access Applications, Que Corporation, Indianapolis 1994, Seite 66 ff.

Programmierung innerhalb der Datenbank die Daten auf Konsistenz und auf Einhaltung gegebener Regeln zu prüfen. Dadurch kann zum Beispiel ein Netz nicht zweimal erstellt werden. Ein Netz kann auch nicht eingegeben werden, wenn es nicht mit mindestens zwei Modulen verbunden wird. Zum anderen wird überprüft, ob zwischen den Modulen Netze erlaubt sind. Denn das objektorientierte Modulkonzept erlaubt nur Verbindungen zwischen zwei aufeinanderfolgenden Modulen, sogenannten Stufen einer Pipeline oder mit der zentralen Steuerung.

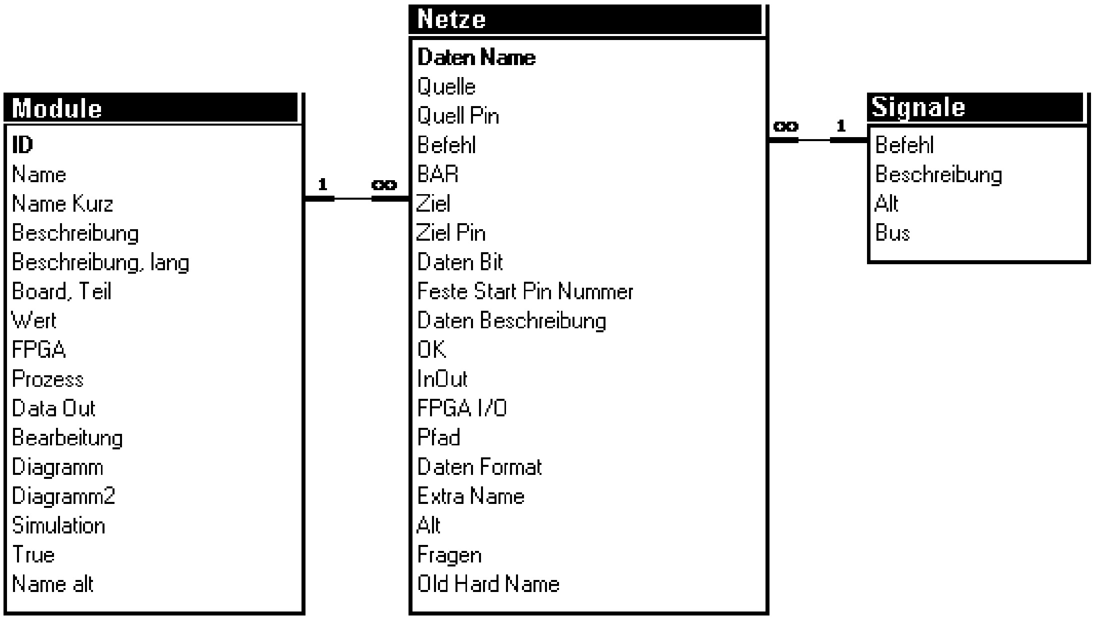

Abbildung 10: Die drei zentralen Tabellen der HEPI-Datenbank

Bei Bedarf wird abschließend aus dieser konsistenten und vollständigen Beschreibung des *Top Levels* der HEPI mit Hilfe eines zu diesem Zweck programmierten Bericht-Generators eine Datei für das verwendete *Computer Aided Design* (CAD) System generiert. Außerdem erstellt die Datenbank eine jeweils aktuelle Dokumentation über alle Module mit allen Netzen und zusätzlich die nach allen Designänderungen notwendigen Testmuster für die Funktionstests in der Simulation.

Alle 299 Ein- und Ausgabe-*Pins* des FPGAs wurden den entsprechenden Netzen, Stromversorgungen usw. zuzuordnen. Das Ergebnis dieser Zuordnung ist ein sogenannter *Constrain File*. Diese Datei ist für die Entwicklung der HEPI-Platine und für die Programmierung des FPGAs erforderlich.

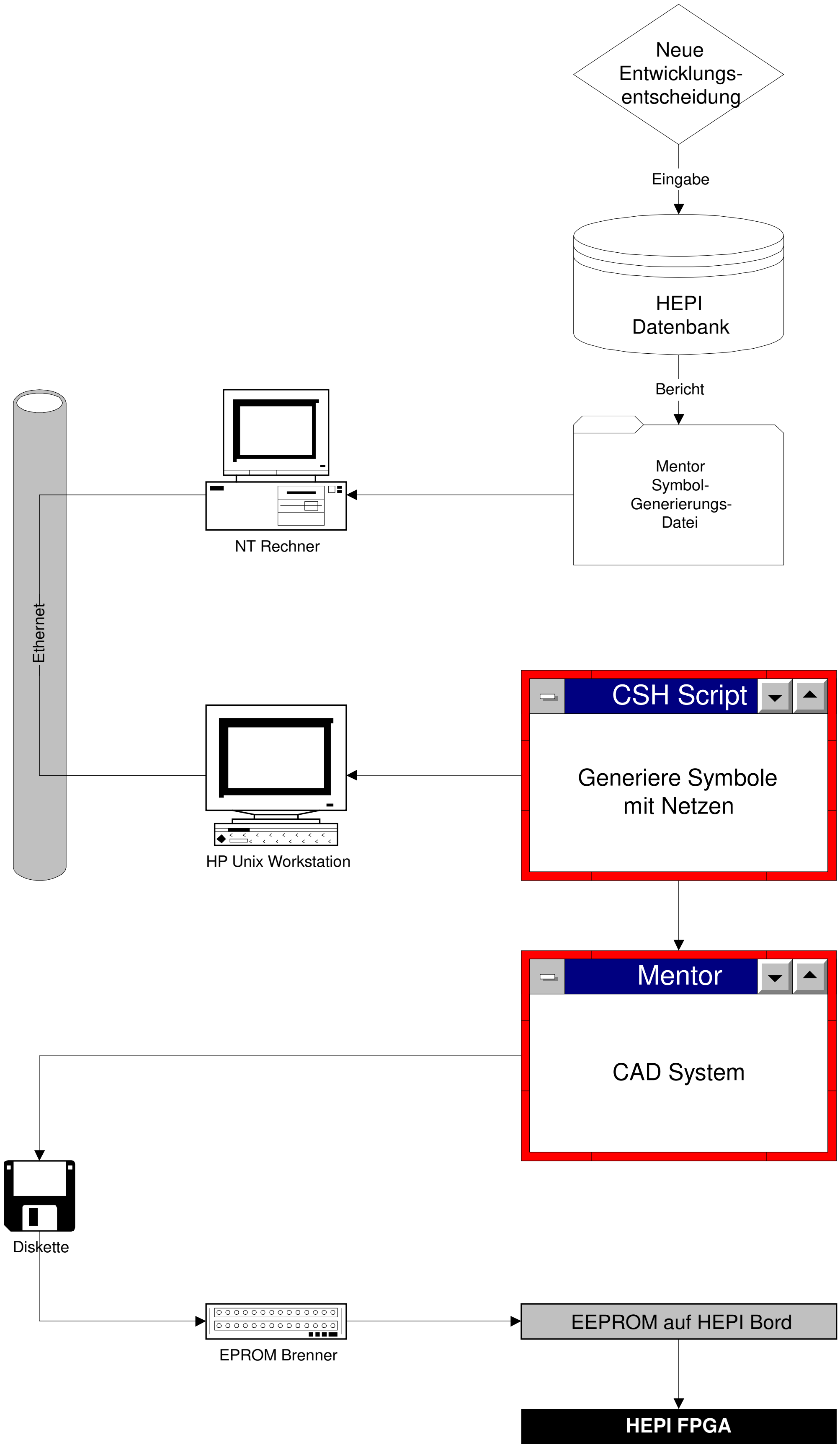

Abbildung 11: Der Designfluß des Entwicklungskonzeptes

Da sich Änderungen während der Entwicklung auf diese Datei auswirken, war es notwendig, die Erstellung dieser Datei zu automatisieren. Auch diese Aufgabe wurde innerhalb der Datenbank mit Hilfe der Programmiersprache *Visual Basic for Applications* realisiert. Eine Moduls dieses Programmes ist als Quelltext im Anhang in Kapitel 4.4 „Quelltexte und Ausgaben“ auf Seite 78 abgebildet.

Mit einer Erweiterung dieses Programmes wurde ein Adapter für einen leistungsfähigeren FPGA in einem HQ240 Gehäuse mit 240 *Pins* entwickelt.

2.1.4 Namens-Schema

Da jede Verbindung der Module mit einem eineindeutigen Netznamen bezeichnet werden mußte, und aufgrund der Vielzahl der Netze, wurde ein systematisches Namens-Schema etabliert.

Dieses Namens-Schema ist wie folgt aufgebaut: jedes Modul hat einen Kurznamen, so heißt zum Beispiel das Modul für die Amplituden-Korrektur AC, für *Amplitude Correction*, und das Modul zur Feststellung von Zeitkoinzidenzen TC, für *Time Coincidence*.

Jedes Netz hat einen Kurznamen, welcher den jeweiligen Zweck ausdrückt. Zum Beispiel steht VD (*Valid*) für ein Netz, das ein Gültigkeitssignal überträgt.

Diese beiden Elemente werden wie folgt zu gültigen Netznamen mit Unterstrichen verbunden: Zunächst wird das Quell-Modul bezeichnet, von dem das Signal ausgeht, dann der Name des Signals und dann der Name des Ziel-Moduls. In diesem Beispiel also *AC_VD_TC*.

Geht ein Signal an mehrere Module, kann das Ziel nicht mehr eindeutig festgelegt werden. In diesem Fall wird es weggelassen. In unserem Beispiel wäre der Name des Netzes dann *AC_VD*. Falls zum Beispiel acht Datenbits übertragen werden sollen, gilt zusätzlich folgende Nomenklatur: *AC_DAT_TC(7:0).*

Normalerweise ist es in der Digitaltechnik üblich, den Zustand eines Signal als *active high* zu bezeichnen. Das bedeutet zum Beispiel für das oben genannte Signal VD für *valid* (gültig), daß die Information VD dann übertragen wird, wenn dieses Signal unter Spannung steht.

Weicht ein Signal von dieser Definition ab, ist es also *active low*, dann wird an den Signalnamen ein BAR angehängt. In diesem Beispiel hieße das Signal dann: *AC_VD_TC_BAR.*

Alle Module und alle Netze sowie die Anzahl der zu übertragenen Bits sind in der Datenbank in verknüpften Tabellen enthalten. Diese Daten repräsentieren eine vollständige

Beschreibung des sogenannten *Top Levels* der HEPI mit allen Schnittstellen nach außen. Alle gültigen Modulnamen sind in der Tabelle 4 „Die Module“ auf Seite 31, alle gültigen Signalnamen sind in der Tabelle 13 „Die Signalnamen“ auf Seite 73 im Anhang aufgelistet.

2.1.5 Symbolgenerator

Die von der Datenbank erzeugten Dateien mit den Informationen über die einzelnen Module werden über das lokale Netzwerk auf die *HP UNIX Workstation* übertragen, auf der sich das CAD-System *Mentor* befindet. *Mentor* ist eine herstellerunabhängige und vollständige Entwicklungsumgebung, die sehr weit verbreitet ist.

Alle großen Hersteller stellen Bauteil-Bibliotheken für *Mentor* zur Verfügung. Durch ihre zum Teil offenen Standards ist es möglich, diese Entwicklungsumgebung zu erweitern und anzupassen. Der Symbolgenerator ist eine solche Erweiterung.

Nach Aufruf eines Skriptes, welches verschiedene weitere Skripte, Filter und Programme aufruft, wird eine dieser Dateien zunächst in die jeweiligen Symboldateien aufgeteilt. Diese Symboldateien können verwendet werden, um mit Hilfe eines *Mentor*-Programmes die Symbole der Module zu erzeugen. Um aber die elektrischen Verbindungen, also die Netze zwischen den neuen Modulen zu erzeugen, wäre es notwendig, jedesmal einige hundert Netze mit Hilfe der graphischen Werkzeuge des *Mentor*-Systems von Hand zu verbinden. Dieser Vorgang ist extrem fehleranfällig.

Unter *Mentor* gibt es die Möglichkeit, eine elektrische Verbindung zu beschreiben, indem zwei graphisch nicht verbundene Netze einen identischen Namen tragen. Deshalb war es notwendig, einen Symbolgenerator zu programmieren, der nicht nur die Symbole generiert, sondern auch die entsprechenden Netze erzeugt und richtig benennt.

In der Datenbank sind 510 Netznamen verzeichnet[40]. Durch den Einsatz der Datenbank als Entwicklungswerkzeug ist die Handhabung dieser komplexen Entwicklung innerhalb des gegebenen Zeitrahmens möglich.

Ein *C-Shell Script* des Symbolgenerators ist in Kapitel 4.4 „Quelltexte und Ausgaben“ auf Seite 78 im Anhang abgebildet.

2.1.6 Schematics

Nachdem die Modularisierung und die Schnittstellen bis zur elektrischen Ebene vorbereitet sind, werden die Module entwickelt, damit sie die ihnen übertragenen digitalen

40 Dieser Wert ist aus der Datenbank, Stand: 21. September 1998.

Funktionen ausführen können. Dies erfolgt mit dem *Designarchitect* des CAD Systems *Mentor*, mit dessen Hilfe sogenannten *Schematics* gezeichnt werden können. Dabei werden zum Beispiel aus einer Symbolbibliothek logische Gatter ausgewählt, plaziert und dann graphisch verdrahtet. Die Bauteil-Bibliotheken werden von den Herstellern der verschiedenen FPGAs und ASICs geliefert. Diese Art der Eingabe einer Schaltung entspricht im Prinzip der Zeichnung eines Schaltplans auf einem Blatt Papier. Diesen Schaltplan nennt man *Schematic*.

Mentor ist ein Entwicklungssystem, das zum Entwurf und zur Simulation von analogen und digitalen Schaltungen dient. Im Folgenden werden einige weitere seiner Werkzeuge beschrieben.

Der *Designmanager* ist das Programm zum Aufruf und zur Verwaltung der Werkzeuge und der Entwürfe. Neben dem Designarchitekten steht zusätzlich noch ein *Very High Speed Integrated Circuit Hardware Description Language* (VHDL) *Editor* zur Verfügung. VHDL ist eine *Hardware*-Beschreibungs-Sprache, die zur Entwicklung von Schaltungen verwendet wird. Sie zählt zu den objektorientierten Sprachen.

Quicksim ist ein Logiksimulator, mit dem die Entwicklungen getestet werden. Mit ihm lassen sich funktionelle und zeitliche Simulationen durchführen. Bei der funktionellen Simulation wird nur das logische Verhalten der Gatter und der einzelnen Bausteine simuliert. Bei der zeitlichen Simulation wird zusätzlich noch die Signallaufzeit berücksichtigt. Dazu müssen allerdings Informationen über diese Laufzeiten zur Verfügung stehen. Diese erhält man aus der Plazierung der Schaltung auf dem ASIC oder dem FPGA. Zum Test der Schaltung lassen sich Testmuster, die zum Teil automatisch von der Datenbank generiert werden, auf die Eingänge der Schaltung legen und die Signale an den Ausgängen verfolgen. Dabei werden nicht nur die Ein- und Ausgänge, sondern auch interne Signale angezeigt.

Das Programm *Xmake* plaziert die Entwicklung auf dem FPGA oder dem ASIC. Dabei verteilt es die einzelnen Komponenten des Schaltplans auf Logikblöcke, den sogenannten *Logic Cell Array*s (LCA), und verbindet diese. Bei einem FPGA ist das Ergebnis eine Datei, mit der ein *EEPROM* gebrannt wird. Auf der HEPI-Platine befindet sich ein Sockel, in den dieser *EEPROM* eingesetzt wird. Beim Einschalten liest der FPGA seine aktuelle Konfiguration in wenigen Sekunden ein und ist dann betriebsbereit. Bei einem ASIC wird eine entsprechende Datei verwendet, um den ASIC herzustellen. Außerdem werden diese Dateien für die zeitliche Simulation benötigt, da daraus die Laufzeitinformationen der plazierten Schaltung gewonnen werden können.

2.1.7 Histogramme

Das Erstellen von Histogrammen aus den Ereignissen des Cäsium-Jodid-Detektors ist eine wesentliche Aufgabe der HEPI. Zum Beispiel wird die Zählrate der Einfach-Ereignisse innerhalb von 256 Energieintervallen für jedes Pixel des Cäsium-Jodid-Detektors bestimmt. Im Kapitel 3.4.4 „Zusammenfassen von Ereignissen: HIST“ auf Seite 65 werden alle Histogramme erläutert, welche in der HEPI erstellt werden können. Als Beispiel für ein Histogramm für nur vier Pixel dient folgende Abbildung 12 „Ein Histogramm mit vier Amplituden-Bereichen für vier Pixel“.

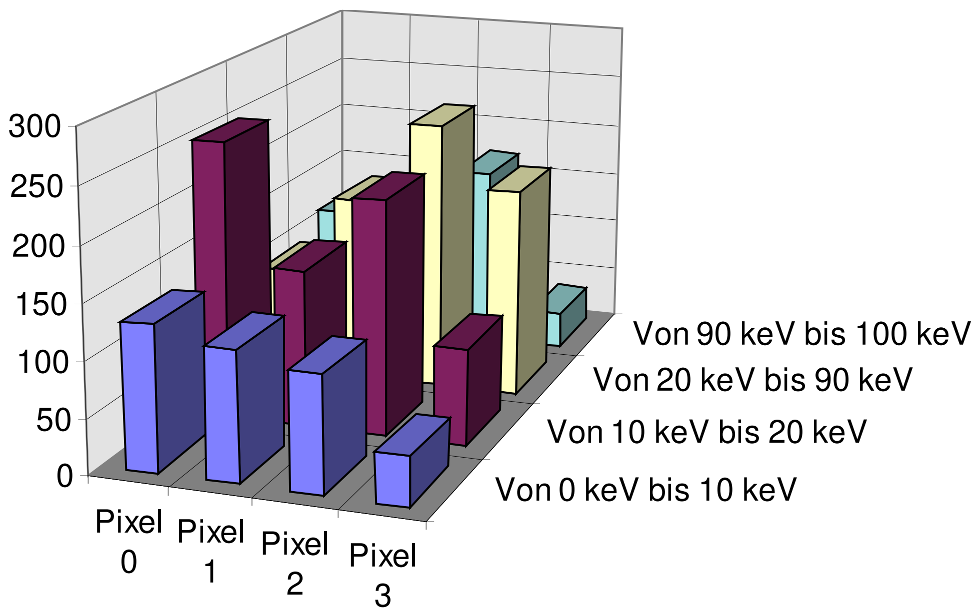

Abbildung 12: Ein Histogramm mit vier Amplituden-Bereichen für vier Pixel

Wie in Kapitel 1.3.4 „Pointierung und Dithering“ auf Seite 18 bereits erwähnt, führt der Satellit bei Beobachtungen zwischen den Pointierungen Lageveränderungen aus. Da sich dabei die Abbildung der Quellen auf der Detektor-Ebene verschiebt, ist die Erstellung der Histogramme während einer solchen *Slew* genannten Lageveränderung nicht sinnvoll. Der Satellit INTEGRAL hat deshalb zwei besondere *Slew* Modi, die unter anderem im anschließenden Kapitel näher erläutert werden.

2.2 Modi

Der *Imager* IBIS hat verschiedene Modi, um seine Aufgaben wahrzunehmen. Für die HEPI sind nur die folgenden Modi relevant. Diese werden durch Aktivierung und Deaktivierung der verschiedenen Module und damit der Datenpfade realisiert.

- *Standard Pointing*
- *Polarimetry Pointing*
- *Photon by Photon.*

Sowohl beim Standard-Modus als auch beim Polarimetrie-Modus werden in der HEPI Histogramme angelegt. Wie schon im Kapitel 1.3.4 „Pointierung und Dithering“ auf Seite 18 erläutert, wird INTEGRAL bei Beobachtungen ein Muster um das Beobachtungsobjekt herum Punkt für Punkt anvisieren.

Deshalb gibt es noch zusätzlich die *Slew* Modi:

- *Standard Slew*
- *Polarimetry Slew.*

Tabelle 5: Die Zuordnung der Module zu den Modi

Beschreibung	Modulname	Diagnostic	Standard Pointing	Polarimetry Pointing	Photon by Photon & Standard Slew & Polarimetry Slew
Cadmium Telluride Detector High Bit Rate Interface	CdTe_HBR	aktiviert	aktiviert	aktiviert	aktiviert
Cesium Iodide Detector High Bit Rate Interface	CsI_HBR	aktiviert	aktiviert	aktiviert	aktiviert
Look Up Table Controller	LUT		aktiviert	aktiviert	aktiviert
Amplitude Correction	AC		aktiviert	aktiviert	aktiviert
Multiple Event Reconstruction and Polarimetry	MP		aktiviert	aktiviert	aktiviert
Time Coincidence and Time Stamp	TC		aktiviert	aktiviert	aktiviert
Energy Selection	ES				aktiviert
Spectral Timing	SPT		aktiviert	aktiviert	
Histogram Module	HIST Calibration		aktiviert	aktiviert	aktiviert
Histogram Module	HIST Single & Multiple		aktiviert		
Histogram Module	HIST Polarimetry			aktiviert	
Memory Controller	MEM		aktiviert	aktiviert	aktiviert
High Bit Rate Interface A	HBR_A	aktiviert	aktiviert	aktiviert	aktiviert
High Bit Rate Interface B	HBR_B	aktiviert	aktiviert	aktiviert	aktiviert

In der Tabelle 5 „Die Zuordnung der Module zu den Modi“ ist aufgeführt, welche Module in welchem Modus aktiviert sind.

Nach eingehender Analyse sind die von der HEPI auszuführenden Aufgaben bei den Modi *Photon by Photon, Standard Slew* und *Polarimetry Slew* identisch.

Nach dem Einschalten der Stromversorgung geht die HEPI automatisch in den sogenannten Diagnostik-Modus. In diesem Modus kann die HEPI von der DPE Telekommandos entgegennehmen und dann in einen anderen Modus wechseln. Der Diagnostik-Modus ist auch der Testmodus. In Ausnahmefällen ist es möglich, diesen Modus auch für Beobachtungen zu verwenden.

Erst auf besondere Anforderung der DPE werden die einzelnen Module aktiviert. Je nach Kombination der aktiven Module befindet sich die HEPI in verschiedenen Modi. Da die Software der DPE verändert werden kann, ist es prinzipiell möglich, durch eine andere Kombination von aktiven Modulen andere Modi zu realisieren oder das Verhalten der vorhandenen Modi zu ändern.

2.2.1 Diagnostik

Zunächst zu dem Diagnostik-Modus. In diesem Modus ist die HEPI transparent. Nur das 64 Bit breite Detektor-Datenformat wird auf das 80 Bit breite DPE-Datenformat abgebildet. Zu beachten ist, daß der digitale Zeitwert der Ereignisse im Detektor-Zeitformat bleibt, also eine Zeitauflösung von 238,4 ns innerhalb von 2 s hat. Nur in diesem Modus erhält die DPE Daten mit dieser Zeitauflösung. Sie muß diese dann gesondert behandeln. Die an diesem Modus beteiligten Module und Datenpfade sind in der Abbildung 52 „Der Diagnostik Modus“ auf Seite 102 zu sehen.

2.2.2 Standard Pointing

Standard Pointing ist der normale wissenschaftliche Modus, der bei langen pointierten Beobachtungen eingesetzt wird. Alle Ereignisse des Cadmium-Tellurid-Detektors und alle von der HEPI erkannten Compton-Ereignisse werden an die DPE weitergereicht. In diesem Modus werden die Zeitwerte der Ereignisse in das HEPI-Zeitformat abgebildet, also in eine Zeitauflösung von 61,04 µs innerhalb von 72,8 h.

Zur Datenreduktion werden sowohl die Einzel-Ereignisse als auch die Mehrfach-Ereignisse aus den 4.096 Halbleiter-Zählern des Cäsium-Jodid-Detektors in jeweils 265 verschiedenen Energiekanälen gesammelt. Die Grenzen der Energiekanäle können innerhalb der Parameter-Grenzen beliebig gewählt werden. Auch die Ereignisse der von der Natrium-22-Quelle des Kalibrations-Systems erzeugten Gammastrahlung mit Energien von 511 keV und 1.275 keV werden in Histogrammen gesammelt. Für beide Energien stehen je 32 Energiekanäle zur Verfügung.

Die Histogramme, die in der vorhergehenden Beobachtung erstellt wurden, werden gleichzeitig von der DPE ausgelesen. Dies ist möglich, da die HEPI zwei Speicherbereiche hat und zwischen diesen auf Befehl der DPE wechselt. Zusätzlich werden, je nach Einstellung, alle 1 bis 500 ms aus den Detektor-Daten des Cäsium-Jodid-Detektors spektroskopische Histogramme mit acht verschiedenen Energiekanälen erstellt. Auch diese Energiekanäle können innerhalb der Parameter-Grenzen beliebig gewählt werden. Siehe dazu die Abbildung 53 „Der Standard Pointing und Standard Polarimetry Modus“ auf Seite 103.

2.2.3 Standard Polarimetry

Der Modus *Standard Polarimetry* ist dem *Standard Pointing* Modus sehr ähnlich. Der einzige Unterschied besteht in der Art der erstellten Histogramme. Statt der Histogramme für Einzel-Ereignisse und Mehrfach-Ereignisse wird ein Histogramm für Polarimetrie-Ereignisse angelegt. Hierbei sind für jeden der 4.096 Halbleiter-Zähler des Cäsium-Jodid-Detektors und für jede der acht möglichen Richtungen 64 Energiekanäle vorhanden. Siehe dazu die Abbildung 53 „Der Standard Pointing und Standard Polarimetry Modus“ auf Seite 103.

2.2.4 Photon by Photon, Pointing Slew und Polarimetry Slew

Die drei Modi des *Imager* IBIS *Photon by Photon, Pointing Slew* und *Polarimetry Slew* sind, soweit es die HEPI betrifft, identisch. Die Ereignisse beider Detektoren und alle von der HEPI erkannten Compton-Ereignisse werden an die DPE weitergereicht. Die Ereignisse aus dem Cäsium-Jodid-Detektor durchlaufen einen Amplituden Bandpaß, der *Energy Selection* genannt wird, um die zu erwartende hohe Datenrate an die Telemetrie-Rate anzupassen. Wie bei den anderen beiden wissenschaftlichen Modi wird die Zeitauflösung geändert, und es werden Histogramme der Kalibrations-Ereignisse erzeugt. Siehe dazu die Abbildung 54 „Der Photon by Photon, Pointing Slew und Polarimetry Slew Modus“ auf Seite 104.

Die Aufgaben der HEPI werden durch die beschriebenen Modi, in der bestimmte Module aktiviert oder deaktiviert sind, realisiert. Die Module, wie sie in Tabelle 4 „Die Module“ aufgeführt sind, werden im nächsten Kapitel im Einzelnen dargestellt.

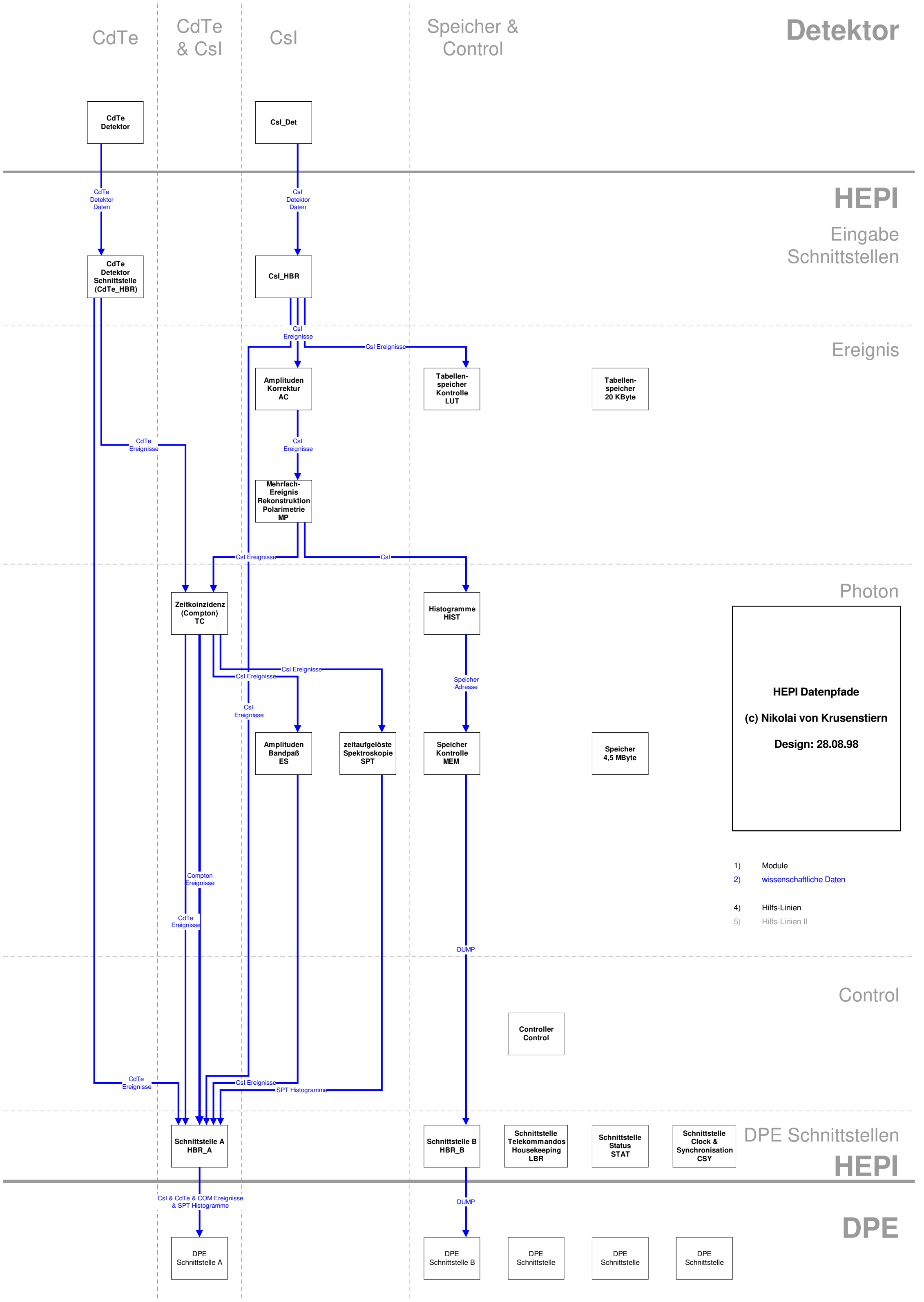

Abbildung 13: Die Module und der Datenfluß

3 HEPI Module

In diesem Kapitel wird auf die einzelnen Module der HEPI eingegangen. Diese technischen Beschreibungen beruhen auf dem Stand der Entwicklung der HEPI vom August 1998. Zu diesem Zeitpunkt haben erfolgreiche Tests bereits die Funktionalität der HEPI-Platine und einiger Module der HEPI gezeigt. Es sind aber noch eine Reihe weiterer Tests notwendig, um sicher zu gehen, daß die HEPI die ihr gestellten Aufgaben erfüllt.

Die Module der HEPI teilen sich funktionell in drei Gruppen auf:

1. die Schnittstellen zur Kommunikation mit der Peripherie der HEPI,
2. die Module zur Steuerung und zur Verwaltung der verschiedenen Speicher und
3. die Module zur Verarbeitung der Detektor-Daten.

Alle Module der HEPI wurden zunächst in der Datenbank definiert. Mit Hilfe der im Kapitel 2.1.3 „Datenbank“ auf Seite 32 und im Kapitel 2.1.5 „Symbolgenerator“ auf Seite 36 beschriebenen Werkzeuge werden sie auf das sogenannte *Top Level* der HEPI unter *Mentor* abgebildet.

Die technischen Zeichnungen des *Top Levels* und der Module, das heißt die Symbole, die *Schematics* und die Simulationen, finden sich aufgrund ihres Formates nicht fortlaufend im Text, sondern ab Seite 81 im Anhang im Kapitel 4.5 „Technische Zeichnungen“. Zusammen mit der Übersicht der Abbildung 13 „Die Module und der Datenfluß“ auf Seite 31 erlauben diese zusammen mit den nun folgenden Beschreibungen der Module einen Einblick in die Funktion und das Zusammenspiel der Module der HEPI.

Zunächst werden einige gemeinsame Elemente der Module beschrieben.

3.1 Schnittstellen der Module

Wie schon im Kapitel 2.1.2 „Modularisierung“ auf Seite 31 erwähnt wurde, gibt es vereinheitlichte Schnittstellen der Module innerhalb der HEPI. Die Vereinheitlichung hat viele Gründe. Gemeinsame und vereinheitlichte Schnittstellen der Module haben den

Vorteil, daß deren Konstruktion nicht nur fehlerfreier und schneller, sondern auch übersichtlicher erfolgen kann.

3.1.1 Schnittstellen für Ereignis-Daten

Schnittstellen für Ereignisse des Cadmium-Tellurid-Detektors

Die Module, die die Ereignis-Daten des Cadmium-Tellurid-Detektors verarbeiten, verwenden folgende Eingabe- und Ausgabe-Schnittstellen:

Die Eingabe-Schnittstelle verteilt das 80 Bit breite Datenformat auf die einzelnen Parameter-Gruppen, wie zum Beispiel die Amplitude, die Koordinaten und die Zeit des Ereignisses. Das Datenformat für Ereignisse des Cadmium-Tellurid-Detektors ist in Tabelle 15 „Das Datenformat der DPE für CsI- und CdTe Ereignisse“ auf Seite 75 definiert.

Um die Gültigkeit der über diese interne Schnittstelle übertragenen Daten zu signalisieren, wird das parallel zu den Daten verlaufende Gültigkeitssignal (VD) gesetzt. Dies gilt analog auch für die Daten des Cäsium-Jodid-Detektors, für die Daten der Compton-Ereignisse und die Daten des Moduls für zeitaufgelöste Spektroskopie. Für das Symbol siehe Abbildung 15 „Die Eingabe-Schnittstelle für Cadmium-Tellurid-Ereignisse“ auf Seite 81.

Die Ausgabe-Schnittstelle sammelt die verschiedenen Parameter-Gruppen und führt sie wieder in ein gemeinsames Datenformat zusammen. Das Symbol ist in der Abbildung 16 „Die Ausgabe-Schnittstelle für Cadmium-Tellurid-Ereignisse“ auf Seite 81 beschrieben.

Schnittstellen für Ereignisse des Cäsium-Jodid-Detektors

Analog zu den Schnittstellen für Ereignisse des Cadmium-Tellurid-Detektors sind die Ein- und Ausgabe-Schnittstellen für Cäsium-Jodid-Ereignisse definiert. Das Datenformat für diese ist in Tabelle 15 „Das Datenformat der DPE für CsI- und CdTe Ereignisse“ auf Seite 75 aufgelistet. Die Symbole werden in der Abbildung 17 „Die Eingabe-Schnittstelle für Cäsium-Jodid-Ereignisse“ auf Seite 82 und der Abbildung 18 „Die Ausgabe-Schnittstelle für Cäsium-Jodid-Ereignisse“ auf Seite 82 beschrieben.

Schnittstellen für Compton-Ereignisse

Analog zu beiden eben genannten Schnittstellen ist auch die Ausgabe-Schnittstelle für Compton-Ereignisse definiert. Die Definition dieses Datenformates ist in Tabelle 16

„Das Datenformat der DPE für Compton-Ereignisse und Spectral Timing“ auf Seite 76 aufgelistet. Das Symbol ist in der Abbildung 19 „Die Ausgabe-Schnittstelle für Compton-Ereignisse“ auf Seite 83 beschrieben.

3.1.2 Housekeeping-Schnittstelle

Viele Module werden in ihrer Funktion durch Parameter beeinflußt. Diese Parameter, die alle 8 Bit breit sind, werden in sogenannten Registern gespeichert und mittels dieser standardisierten Schnittstelle gesetzt, gelöscht und ausgelesen.

Darüber hinaus haben einige Module eingebaute Zähler. Auch diese sind Teil des sogenannten *Housekeepings* und können über diese Schnittstelle ausgelesen und auf Null gesetzt werden.

Die *Housekeeping*-Schnittstelle wurde nicht im Rahmen dieser Diplomarbeit erstellt. Sie wurde von der Firma DD&T, Reutlingen entwickelt. Das Symbol wird in der Abbildung 20 „Die Ein- und Ausgabe-Schnittstelle für Housekeeping-Daten“ auf Seite 83 beschrieben.

3.1.3 Gemeinsame Signale für Module

Neben den standardisierten Schnittstellen sind weitere gemeinsame Signale für die Module notwendig. Bei der Diskussion der einzelnen Module werden sie nur dann noch einmal erwähnt, falls dies für das Verständnis notwendig ist:

- die *Clock* (CLK[41]) wird allen Modulen zur Verfügung gestellt. Sie hat eine Frequenz von 4.194.304 Hz. Ein Takt entspricht somit etwa 238,4 ns;
- der Pipeline-Takt (RNE[42]) der beiden *Pipelines*, wie sie in Kapitel 2.1.2 „Modularisierung“ auf Seite 31 besprochen wurden;
- das Signal zum Löschen (CLR[43]) aller internen Register[44]. Danach befinden sich die Module in einem definierten Grundzustand;
- das Signal zur Initialisierung (INIT[45]) für Module, die Speicher verwalten;
- das Signal zum Löschen (CLR_CNT) interner Zähler;
- das Signal zur Aktivierung (EN) des Moduls;

41 Name: CLK.

42 Namen: CTR_RNE, CTR_RNE_CSI und CTR_RNE_CDTE.

43 Name: CTR_CLR.

44 Register sind Speicher für Parameter der HEPI.

45 Name: CTR_INIT.

- drei Signale, die nur Testzwecken dienen. Es ist möglich, den Zustand der HEPI „einzufrieren“ und dann den Inhalt aller dafür vorbereiteten Register auszulesen. In diesen Registern sind die oben beschriebenen Parameter gespeichert. Dazu sind alle Register in Serie miteinander verbunden. Diese Verbindung wird *Scan Path* (SP) genannt. Aus diesem Grund ist jedem Modul sowohl ein spezielles Eingabe- als auch Ausgabesignal zugeordnet. Dieser Test wird auf der Modulebene durch das Signal *Scan Enable* (SE[46]) aktiviert.

3.2 Schnittstellen der HEPI

Diese erste Gruppe der Module der HEPI, die Schnittstellen, ermöglicht den Austausch von Steuersignalen, Daten und Telekommandos zwischen der HEPI, den Detektoren und der DPE.

Die Schnittstellen benötigen zum Teil einige externe Bauteile, wie zum Beispiele sogenannte Treiber, welche die benötigte Leistung zur Verfügung stellen, oder Optokopler, welche Leitungen galvanisch trennen. Diese sind Bestandteil der HEPI-Platine. Die Signale, die durch *Pins* des FPGAs nach außen geführt werden sind als sogenannte *IO-Buffer*, die in der technischen Zeichnung des *Top Levels* im Kapitel 4.5 „Technische Zeichnungen“ ab Seite 81 auf der linken Seite abgebildet sind.

Die externen Schnittstellen der HEPI sind:

1. das *Cadmium Telluride Detector High Bit Rate Interface* (CdTe_HBR) zur Eingabe der Ereignisse aus dem Cadmium-Tellurid-Detektor,
2. das *Cesium Iodide Detector High Bit Rate Interface* (CsI_HBR) zur Eingabe der Ereignisse aus dem Cäsium-Jodid-Detektor,
3. das *High Bit Rate Interface A* (HBR_A) zur Ausgabe einzelner Ereignisse,
4. das *High Bit Rate Interface B* (HBR_B) zur Ausgabe der Histogramme,
5. das *Low Bit Rate Interface* (LBR) zur Ein- und Ausgabe der Telekommandos und der *Housekeeping*-Daten,
6. das *Clock and Synchronization Interface* (CSY) zur Eingabe der *Clock* und des Synchronisations-Signals und
7. das *Status Interface* (STAT) zur Ausgabe des Status der HEPI.

Im Folgenden werden eine Reihe von Signalnamen verwendet. Eine Liste mit Erklärungen zu den Signalnamen befindet sich im Anhang in Tabelle 13 „Die Signalnamen“ auf

46 Name: STAT_SE.

Seite 73. Die Listen mit Erklärungen zu den Signalen der Datenformate befinden sich in den Tabellen auf den Seiten 73 bis 76. Da die verwendeten Signalnamen für das Verständnis der Funktionen der Module nicht wesentlich sind, sind diese jeweils als Fußnote aufgeführt.

3.2.1 Cadmium-Tellurid-Detektor: CdTe_HBR

Diese serielle Eingabe-Schnittstelle[47] kontrolliert die Übertragung der Ereignis-Daten des Cadmium-Tellurid-Detektors und stellt sie für die weitere Verarbeitung in der HEPI zur Verfügung. Außerdem versorgt sie den Detektor mit einem Synchronisations-Signal.

Im Einzelnen bildet diese Schnittstelle das 64 Bit breite Detektor-Datenformat[48] auf das 80 Bit breite DPE-Datenformat[49] ab und kehrt die von 0 bis 63 gehende Numerierung der einzelnen Bits um, da die von der ESA verwendete Numerierung in der Digitaltechnik unüblich ist. Somit wird ein Umdenken bei der Entwicklung vermieden, welches zu Fehlern führen könnte. Das ESA Datenformat wird am Ausgang der HEPI wieder hergestellt. Dieses Modul kontrolliert die Kommunikation mit dem Detektor durch eine *Clock* (CLK[50]) und einem Signal (RNE[51]), das den Ausgabe-FIFO[52] des Detektors veranlaßt, die Daten des nächsten Detektor-Ereignisses zu übertragen.

Beim Beginn einer neuen Beobachtung wird an den Detektor ein Signal (CLR[53]) geschickt, damit dieser den Inhalt seines Ausgabe-FIFOs löscht. Nur wenn der Ausgabe-FIFO des Detektors nicht leer ist, schickt dieser ein entsprechendes Signal (NE[54]), und das Modul kann gegebenenfalls die Übertragung initiieren.

Nachdem ein gültiges Detektor-Ereignis erfolgreich empfangen worden ist, werden die Signale *Type_Format, Type_Dat* und *Type_Raw* des 80 Bit breiten Datenformats (DAT[55]) auf Null gesetzt. Im Diagnostik-Modus wird das Signal *Type_Raw* jedoch auf eins gesetzt. Die Bedeutung dieser Signale ist in der Tabelle 6 „Die Definition einiger Signale der Datenformate“ erläutert.

---

47 Die serielle Schnittstelle folgt dem Standard RS 422.

48 Siehe die Tabelle 14 „Das Datenformat der Detektoren“ auf Seite 73.

49 Siehe die Tabelle 15 „Das Datenformat der DPE“ auf Seite 75.

50 Name: CDTE_HBR_CLK_CDTE_DET.

51 Name: CDTE_HBR_RNE_CDTE_DET.

52 First In, First Out (FIFO), ein Pufferspeicher, der die zuerst eingegeben Daten wieder zuerst ausgibt.

53 Name: CDTE_HBR_CLR_CDTE_DET_BAR.

54 Name: CDTE_DET_NE_CDTE_HBR.

55 Name: CDTE_DET_DAT.

Alle gültigen Detektor-Ereignisse werden gezählt. Der Wert dieses Zählers wird als Teil der *Housekeeping*-Daten zur Erde übertragen. Ein Überlauf dieses Zählers wird signalisiert (*CNT_OF*).

Tabelle 6: Die Definition einiger Signale der Datenformate

Signal	Null	Eins
Type_Cal	Ereignis nicht aus der Kalibrations-Quelle	Ereignis aus der Kalibrations-Quelle
Type_Dat	Ereignis des Cadmium-Tellurid-Detektors	Ereignis des Cäsium-Jodid-Detektors
Type_Format	Datenformat mit 80 Bit	Datenformat mit 160 Bit
Type_MMC	-	Mehrfachereignis aus mehr als einem Detektor-Modul
Type_Mux	Einfachereignis	Mehrfachereignis
Type_Pol	Ereignis ohne Polarimetrie Daten	Ereignis mit Polarimetrie Daten
Type_Raw_BAR	Ereignis des Diagnostik Modus	Ereignis der wissenschaftlichen Modi der HEPI
Type_SPT	Datenformat normal	Datenformat für Spectral Timing

Beide Schnittstellen übernehmen noch eine weitere wichtige Funktion, die über die der Datenübertragung hinausgeht. Aufgrund der nicht zu vernachlässigenden Laufzeiten der Daten von und zu den Detektoren senden die Detektor-Schnittstellen ein Synchronisations-Signal (SYNC[56]), dessen individuelle Verzögerung für jeden Detektor im Bereich zwischen 1 und 64 *Clock*-Takten eingestellt werden kann. Der Wert dieser individuellen Verzögerung wird in einem Register gespeichert, das über die in Kapitel 3.1.2 „Housekeeping-Schnittstelle“ auf Seite 45 beschriebene Schnittstelle gesetzt, gelöscht und ausgelesen werden kann.

Das verzögerte Signal wird alle zwei Sekunden gesendet und in der HEPI aus dem Synchronisations-Signal (SYNC[57]) der DPE abgeleitet, welches von dieser jede Sekunde gesendet wird. Die Detektoren setzen dann ihre interne laufende Zeit wieder auf Null. Das Symbol ist in der Abbildung 21 „Das Symbol der Schnittstelle zum Cadmium-Tellurid-Detektor“ auf Seite 84 beschrieben.

3.2.2 Cäsium-Jodid-Detektor: CsI_HBR

Die Schnittstelle zum Cäsium-Jodid-Detektor, die im Wesentlichen mit der Schnittstelle zum Cadmium-Tellurid-Detektor identisch ist, unterscheidet sich nur darin, daß das *Type_Dat* Signal auf eins gesetzt wird.

56 Name: CDTE_HBR_SYNC_CDTE_DET.

57 Name: CTR_SYNC.

Darüber hinaus findet in diesem Modul eine weitere Abbildung statt: Im Cäsium-Jodid-Detektor können Mehrfach-Ereignisse registriert werden. Dies sind Ereignisse, die innerhalb eines kleinen Zeitfensters[58] von etwa 1µs in einem der 8 Module des Detektors registriert werden. Es wird dann davon ausgegangen, daß diese Ereignisse einem Photon zuzuordnen sind. Die Daten der einzelnen Ereignisse erhalten eine Auszeichnung, die sogenannte *Multiplicity*, welche angibt, aus wie vielen Ereignissen sie sich zusammensetzen. Siehe dazu die Tabelle 7 „Die Definition der Multiplicity Bits“.

Tabelle 7: Die Definition der Multiplicity Bits

Multiplicity DET	Multiplicity HEPI	Definition Ereignis
00	01	Einfach- Ereignis
01	10	Zweifach-Ereignis
10	11	Dreifach-Ereignis
11	00	Nicht Gültig

Die Elektronik des Detektors stellt sicher, daß diese Ereignisse hintereinander an die HEPI übertragen werden. Das von der ESA gewählte Format wird, wie in der Tabelle 7 „Die Definition der Multiplicity Bits“ aufgelistet, auf das DPE-Format abgebildet.

Wie im Kapitel 3.2.1 „Cadmium-Tellurid-Detektor: CdTe_HBR“ auf Seite 47 beschrieben, schickt auch diese Schnittstelle ein Synchronisations-Signal (SYNC[59]) zum Cäsium-Jodid-Detektor. Das Symbol ist in der Abbildung 22 „Das Symbol der Schnittstelle zum Cäsium-Jodid-Detektor“ auf Seite 84 beschrieben.

In der HEPI gibt es ein Modul zur Rekonstruktion dieser oben genannten Mehrfach-Ereignisse. Dieses Modul wird in Kapitel 3.4.2 „Rekonstruktion von Mehrfach-Ereignissen und Polarimetrie: MP“ auf Seite 60 näher erläutert.

Die genannten Schnittstellen sind den Detektoren des *Imager* IBIS zugeordnet. Im nächsten Kapitel sollen die Schnittstellen zur DPE behandelt werden.

3.2.3 Data Processing Electronics: *HBR_A*

Das *High Bit Rate Interface A* als Schnittstelle zur DPE hat die Aufgabe, alle nicht in Histogrammen zusammengefaßten Ereignisse (DAT) an die DPE zu übertragen. Die Datenübertragung wird von der DPE gesteuert.

---

58 Siehe IBIS Team, INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998, Seite 10.

59 Name: CSI_HBR_SYNC_CSI_DET.

Hierzu überträgt das Modul alle ankommenden gültigen Daten, außer jenen, die direkt von den beiden Detektor-Schnittstellen kommen. Nur im Diagnostik-Modus ignoriert (EN_DIA[60]) es die ankommenden Daten und überträgt ausschließlich die Daten, die direkt von den beiden Detektor-Schnittstellen kommen und gültig sind. In diesem Modus haben die Ereignis-Daten eine Zeitauflösung von 238,4 ns innerhalb von 2 s.

Über die beschriebenen Modi hinaus ist es möglich, per Telekommando die beiden *Pipelines* (EN[61]) unabhängig voneinander ein- oder auszuschalten.

Um das Übertragungsschema zur DPE zu vereinfachen, werden die 80 Bit breiten Ereignis-Daten immer nur in Zweier-Paaren übertragen. Das Datenformat der Compton-Ereignisse und das Datenformat[62] der zeitaufgelösten Spektroskopie ist 160 Bit breit. Somit sind alle übertragenen Datenblöcke 160 Bit breit. Außerdem wird hier die ESA konforme Numerierung des Datenformates wiederhergestellt.

Im Anschluß an die Ausgabe durch die Schnittstelle A befindet sich ein Zwischenspeicher (FIFO), der etwa 1.600 Ereignisse aufnehmen kann. Dieser Zwischenspeicher kann auch gelöscht (CLR[63]) werden. Ab einem einstellbaren Füllstand dieses Zwischenspeichers, der als Register über die *Housekeeping*-Schnittstelle gesetzt wird, wird die DPE, die diese Schnittstelle[64] kontrolliert, über ein Signal (HF[65]) aufgefordert, die anliegenden Daten auszulesen. Dazu stellt die DPE für die Übertragung eine *Clock* (CLK[66]) zur Verfügung und initiiert (TX_EN[67]) die parallele Datenübertragung (DAT[68]). Um die 160 Bit breiten vorbereiteten „Pakete“ zu übertragen, wird immer ein Vielfaches von 10 dieser 16 Bit Daten übertragen. Ist der FIFO leer (FE[69]), wird die Datenübertragung beendet. Das Symbol ist in der Abbildung 23 „Das Symbol der Schnittstelle A zur DPE“ auf Seite 84 beschrieben. Die weiteren Signale gehören zu der elektrischen Schnittstelle nach der Norm[70] RS 422 und werden daher hier nicht näher erläutert.

---

60 Name: CTR_EN_DIA.

61 Namen: CTR_EN_CDTE_PPM_HBR_A und CTR_EN_CSI_PPM_HBR_A.

62 Siehe die Tabelle 16 „Das Datenformat der DPE für Compton-Ereignisse und Spectral Timing“ auf Seite 76.

63 Name: CTR_CLR_FIFO_HBR_A.

64 Die parallele Schnittstelle folgt dem Standard RS 422.

65 Name: HBR_A_HF_CTR.

66 Name: B_HBR_A_CLK_HBR_A.

67 Name: B_HBR_A_TX_EN_HBR_A.

68 Name: HBR_A_DAT_B_HBR_A(15:0).

69 Name: HBR_A_FE_CTR.

70 Siehe Volkmer, Reiner, HEPI Interface Description, 4.1, Juni 1998 IAAT, Seite 19.

3.2.4 Data Processing Electronics: *HBR_B*

Auch das *High Bit Rate Interface B* ist eine parallele Schnittstelle[71], die von der DPE kontrolliert wird. Sie dient zur Übertragung der Histogramme, die im 4,5 MByte Speicher der HEPI gesammelt werden.

Die DPE liest die erstellten Histogramme wie folgt aus: ein Telekommando der DPE initiiert den Auslesevorgang. Dabei wird ein Speicherabschnitt durch Angabe der Anfangsadresse und seiner Länge definiert. Dieser Speicherabschnitt wird dann über die Schnittstelle A an die DPE übertragen.

Auf diese Weise werden die Histogramme ausgegeben, die im Speicher der HEPI erstellt wurden. Diese enthalten nur Daten (DAT[72]) aus dem Cäsium-Jodid-Detektor. Die Funktion der Schnittstelle ist analog zu der des *High Bit Interfaces A*. Dies schließt auch den Zwischenspeicher (FIFO) ein. Das Symbol ist in der Abbildung 24 „Das Symbol der Schnittstelle B zur DPE“ auf Seite 85 beschrieben. Auch hier gehören die weiteren Signale zu der elektrischen Schnittstelle nach der Norm[73] RS 422 und werden daher nicht näher erläutert.

3.2.5 Data Processing Electronics: LBR

Über die *Low Bit Rate* Schnittstelle[74] erhält die HEPI die Telekommandos und gibt die *Housekeeping*-Daten aus. Auch diese Schnittstelle wird von der DPE kontrolliert.

Die DPE kommuniziert über einen gemeinsamen *Bus* mit den verschiedenen Baugruppen des *Imager* IBIS. Das *Low Bit Rate Interface* ist auch an diesen *Bus* angeschlossen. Unter dem Begriff *Bus* ist hier ein Bündel von elektrischen Verbindungsleitungen zu verstehen, an die über Schnittstellen mehrere Geräte parallel angeschlossen sind. Jede der daran angeschlossenen Schnittstellen wird bei Bedarf durch eine bestimmte 6 Bit Adresse[75] selektiert.

---

71 Die parallele Schnittstelle folgt dem Standard RS 422.

72 Name: MEM_DAT_HBR_A(15:0).

73 Siehe Volkmer, Reiner, HEPI Interface Description, 4.1, Juni 1998 IAAT, Seite 19.

74 Die bidirektionale serielle Schnittstelle folgt dem Standard für asynchrone Datenübertragungen mit *Start, Even Parity* und *Stop* Bit.

75 Für die Funktion sind aber nur vier (ADD 3 bis ADD 6) der sechs Leitungen notwendig.

Die Schnittstelle verfügt über drei Leitungen:

- der *Clock* (CLK[76]) für die Übertragung,
- der Datenleitung für das Empfangen von Daten (DTF[77]) von der DPE und
- der Leitung für das Senden von Daten (DTR[78]) an die DPE.

Es wird ein asynchrones Protokoll verwendet. Die maximale Datenrate ist 16 kBaud, die effektive Übertragungsrate ist aufgrund des verwendeten Protokolls 9,8 kBaud.

Die empfangenen Daten (DAT[79]), wie zum Beispiel Telekommandos, werden an den *Controller* weitergeleitet. Auch die *Housekeeping*-Daten (DAT[80]) werden auf Anforderung der DPE über diese Schnittstelle an die DPE übergeben. Bestimmte Parameter[81] der HEPI, welche die Funktionen der einzelnen Module steuern, wie zum Beispiel Schwellwerte, werden alle acht Sekunden angefordert. Alle anderen *Housekeeping*-Daten werden nur auf besondere Anforderung der DPE übertragen. Das Symbol ist in der Abbildung 25 „Das Symbol der Schnittstelle LBR zur DPE“ auf Seite 85 beschrieben. Die weiteren Signale gehören zu dem verwendeten Übertragungsprotokoll[82] und werden hier nicht näher erläutert.

3.2.6 Data Processing Electronics: CSY

Es folgen zwei sehr einfach aufgebaute Schnittstellen, die direkt Signale zwischen der DPE und dem *Controller* übertragen.

Die eine ist die Schnittstelle, über welche die HEPI von der DPE die *Clock* (CLK[83]) mit 4.194.304 Hz ein Signal zur Initialisierung (RES[84]) und jede Sekunde ein Synchronisations-Signal (SYNC[85]) erhält. Diese werden zum Teil (SYNC[86] und RES[87]) nur an den *Controller* weitergeleitet. Die *Clock* (CLK[88]) wird an alle Module weitergeleitet.

---

76 Name: B_LBR_CLK_LBR.

77 Name: B_LBR_DTF_LBR.

78 Name: LBR_DTR_B_LBR.

79 Name: LBR_DAT_CTR(7:0).

80 Name: CTR_DAT_ LBR(7:0).

81 Siehe Volkmer, Reiner, HEPI Interface Description, 4.1, Juni 1998 IAAT, Seite 10.

82 Siehe ebd., Seite 9.

83 Name: B_CSY_CLK_CSY.

84 Name: B_CSY_RES_CSY.

85 Name: B_CSY_SYNC_CSY.

86 Name: CSY_SYNC_CTR.

87 Name: CSY_RES_CTR.

88 Name: CLK.

Dies sind die zugehörigen Abbildungen:

- Abbildung 26 „Das Symbol der Schnittstelle CSY zur DPE“ auf Seite 85 und
- Abbildung 27 „Das Schematic der Schnittstelle CSY zur DPE“ auf Seite 86.

3.2.7 Data Processing Electronics: STAT

Die andere Schnittstelle, die direkt Signale zwischen der DPE und dem *Controller* überträgt, ist die zur Ausgabe des Status der HEPI. Diese Schnittstelle ist speziell für die Testreihen der HEPI konzipiert. Jede Sekunde liegt hier ein Signal (BA[89]), ein sogenannter Herzschlag, an, der zeigt, daß die HEPI noch aktiv ist.

Dies sind die zugehörigen Abbildungen:

- Abbildung 28 „Das Symbol der Schnittstelle STAT zur DPE“ auf Seite 86 und
- Abbildung 29 „Das Schematic der Schnittstelle STAT zur DPE“ auf Seite 87.

3.3 Steuerung und Speicherverwaltung

Die Module der zweite Gruppe steuern die HEPI und verwalten eine Vielzahl von Parametern und Schwellwerten. Es sind die Module zur Steuerung und Speicherverwaltung. Der in den Modulen verwendete Speicher ist nicht innerhalb der HEPI realisiert, sondern ist mit externen Bauteilen auf der HEPI-Platine realisiert.

Die Kontrollmodule der HEPI sind:

1. der *Controller* (CTR) zur Steuerung der HEPI,
2. der *Look Up Table Controller* (LUT) zur Verwaltung einiger Tabellen und
3. der *Memory Controller* (MEM) zur Verwaltung des Speichers für die Erstellung der Histogramme.

3.3.1 Controller: CTR

Die zentrale Steuerung der HEPI erfolgt über den *Controller*. In diesem Modul werden die Telekommandos der DPE interpretiert und der gesamte Ablauf gesteuert und überwacht.

89 Name: STAT_BA_B_STAT.

Der *Controller* der HEPI ist eine sogenannte *State Machine*. An den Eingängen des *Controllers* liegen verschiedene Signale an. Jeder sinnvollen Kombination dieser Signale wird ein sogenannter Zustand zugeordnet. Darüber hinaus ist es noch von Belang, aus welchem Zustand der *Controller* in den neuen Zustand gelangt ist. Aus diesen Informationen über eine Zustandsänderung ergibt sich eine bestimmte Reaktion, das heißt, ein bestimmtes Muster an Ausgangssignalen. Durch diesen sehr vereinfacht dargestellten Mechanismus wird die HEPI gesteuert.

Der *Controller* führt in erster Linie durch Aktivieren oder Deaktiveren der Module sowie durch Setzen der Register, die den Modulen zugeordnet sind, die von der DPE kommenden Telekommandos aus. Er steuert also die gesamte HEPI und führt die Modi-Wechsel durch. Weiterhin liefert der *Controller* die *Housekeeping*-Daten an die DPE und kontrolliert die Aktivität der beiden *Pipelines*.

Da der *Controller* mit allen Modulen in Verbindung steht, werden die den Modulen zugeordneten Signale in den einzelnen Unterkapiteln zu den Modulen besprochen. Das Symbol ist in der Abbildung 30 „Das Symbol des Controllers“ auf Seite 87 beschrieben.

3.3.2 Tabellenspeicher: LUT

Der Tabellenspeicher steuert die Verwaltung eines speziellen Speichers, in dem 20.000 Parameter gespeichert werden. Dies sind unter anderem die Grenzen der Energiekanäle für die Erstellung der Histogramme.

Die Parameter werden über die Schnittstelle LBR von der DPE in diesen Speicher geladen, sie können über die Schnittstelle LBR aber auch ausgelesen werden. Der Datenaustausch findet über den *Controller* mittels der standardisierten *Housekeeping*-Schnittstelle statt.

Der Kontrolle des zugeordneten Speichers findet durch folgende Signale statt:

- die Adresse (ADR[90]) des Speichers,
- die übertragenen Daten (DAT[91]),
- das Lesesignal (RW[92]) und
- die Aktivierung (CS[93]) eines Speicherzugriffs.

90 Name: LUT_ADR_B_LUT(14:0).

91 Name: LUT_DAT_B_LUT(7:0).

92 Name: LUT_RW_B_LUT.

93 Name: LUT_CS_B_LUT_BAR.

Die Parameter sind, wie in Tabelle 8 „Die Organisation der Tabellen in der Look Up Table“ ersichtlich, organisiert.

Für die individuelle Korrektur der einzelnen Amplituden Werte jedes Pixel des Cäsium-Jodid-Detektors erhält der Tabellenspeicher die Ereignis-Daten (DAT[94]) direkt von der Schnittstelle zu dem Cäsium-Jodid-Detektor. Er kann so im nächsten Pipeline-Takt dem Modul zur Amplituden-Korrektur die entsprechenden Parameter (DAT[95]) zur Verfügung stellen. Dieser Vorgang geschieht in zwei Schritten (WR[96]), wobei jeweils 8 Bit übertragen werden.

Tabelle 8: Die Organisation der Tabellen in der Look Up Table[97]

Tabelle für	Größe
Rate Meter	8 kbyte
Amplituden-Korrektur Byte 1	8 kbyte
Amplituden-Korrektur Byte 2	8 kbyte
Histogramm: Einzel-Ereignisse	1 kbyte
Histogramm: Mehrfach-Ereignisse	1 kbyte
Histogramm: Kalibrations-Ereignisse	1 kbyte
Histogramm: Polarimetrie-Ereignisse	1 kbyte

Außerdem führt der Tabellenspeicher für jedes der 4.096 Pixel des Cäsium-Jodid-Detektors einen Zähler: das sogenannte *Rate Meter*. Jener Teil der Funktionalität des Moduls kann individuell aktiviert (EN_RM[98]) werden. Falls eines der Pixel außergewöhnlich oft Ereignisse registriert, kann es von der Detektor-Elektronik deaktiviert werden. Der erste Hinweis darauf ist die allgemein hohe Datenrate und ein im Modul generiertes Signal (CNT_OF[99]). Über die *Housekeeping*-Daten des Detektors läßt sich die hohe Datenrate einem der acht Module des Detektors zuordnen. Per Telekommando werden dann die entsprechenden Zähler des *Rate Meters* ausgelesen und als *Housekeeping*-Daten zur weiteren Analyse zur Erde geschickt.

Zudem sind die Parameter für die Grenzen der Energiekanäle der verschiedenen Histogramme für Kalibrations-, Polarimetrie- Einfach- und Mehrfach-Ereignisse in diesem Tabellenspeicher hinterlegt.

---

94 Name: CSI_HBR_DAT(79:0).

95 Name: LUT_DAT(7:0).

96 Namen: LUT_WR_B1_AC und LUT_WR_B2_AC.

97 Aus: Volkmer, Reiner, HEPI Design Report, Issue 6.0, Juli 1998 IAAT, Seite 20.

98 Name: CTR_EN_RM_LUT.

99 Name: LUT_CNT_OF_CTR.

Um die Amplitude eines Ereignisses einem Energiekanal zuzuordnen, erhält es von dem Modul, das die Erstellung der Histogramme steuert eine Adresse (ADR[100]). Der Energiekanal entspricht einer Adresse (DAT[101]) im Speicher der HEPI, die über dasselbe Signal ausgegeben wird wie die beiden Byte für die Amplituden-Korrektur. Hier lautet das entsprechende Steuersignal WR[102].

All diese Aufgaben erledigt dieses Modul innerhalb eines Pipeline-Taktes. Das Symbol ist in der Abbildung 31 „Das Symbol der Look Up Table“ auf Seite 88 beschrieben.

3.3.3 Speicherverwaltung: MEM

Tabelle 9: Die Organisation der Histogramme im Speicher[103]

Modus	Polarimetrie	Sonst
0,00 MByte	Polarimetrie-Ereignisse A	Einfach-Ereignisse A
0,25 MByte		
0,50 MByte		
0,75 MByte		
1,00 MByte		Mehrfach-Ereignisse A
1,25 MByte		
1,50 MByte		
1,75 MByte		
2,00 MByte	Polarimetrie-Ereignisse B	Einfach-Ereignisse B
2,25 MByte		
2,50 MByte		
2,75 MByte		
3,00 MByte		Mehrfach-Ereignisse B
3,25 MByte		
3,50 MByte		
3,75 MByte		
4,00 MByte	Kalibrations-Ereignisse A	Kalibrations-Ereignisse A
4,25 MByte		
4,50 MByte	Kalibrations-Ereignisse B	Kalibrations-Ereignisse B
4,75 MByte		

100 Name: HIST_ADR_LUT(11:0)

101 Name: HIST_DAT(7:0).

102 Name: LUT_WR_HIST.

103 Siehe Volkmer, Reiner, HEPI Design Report, Issue 6.0, Juli 1998 IAAT, Seite 22.

Die Speicherverwaltung verwaltet den Histogramm-Speicher der HEPI. Die Erstellung der Histogramme wird jedoch von dem Modul ausgeführt, welches in Kapitel 3.4.4 „Zusammenfassen von Ereignissen: HIST“ auf Seite 65 beschrieben wird.

Die 4,5 MByte Speicher, die dieses Modul verwaltet, sind in zwei Bereiche und diese wiederum in zwei Bänke geteilt. Die zwei Bereiche sind zum einen für die Kalibrations-Ereignisse und zum anderen für die anderen Ereignisse vorgesehen. Die jeweils zwei Speicherbänke sind notwendig, um gleichzeitig neue Histogramme zu erstellen und die Histogramme der vorherigen Beobachtungsperiode zur Erde zu schicken.

In welche der beiden Speicherbänke die Histogramme geschrieben werden und aus welcher die Histogramme ausgelesen werden, liegt alleine in der Verantwortung der DPE. Die DPE weist die HEPI über ein Telekommando an, welche Speicherbank für das Speichern der Histogramme verwendet werden soll. Siehe dazu die Tabelle 9 „Die Organisation der Histogramme im Speicher“ auf der Seite 56.

Um einen bestimmten Zähler eines Histogramms um eins zu erhöhen, wird von dem Modul, das die Histogramme erstellt, die entsprechende Adresse (ADR[104]) des Speichers angegeben. Mit einem Signal (WR[105]) wird die Speicherverwaltung aufgefordert, den Wert, der an dieser Adresse gespeichert ist, auszulesen, um eins zu erhöhen und wieder abzuspeichern. Erreicht dieser Wert den maximalen Wert von 255, wird dieser Überlauf an den *Controller* (OF[106]) signalisiert.

Die DPE liest die erstellten Histogramme wie folgt aus: Ein Telekommando der DPE initiiert den Auslesevorgang. Dabei wird ein Speicherabschnitt durch Angabe der Anfangsadresse und seiner Länge definiert. Dieser Speicherabschnitt wird dann über die Schnittstelle A an die DPE übertragen (DAT[107]). Während eines Pipeline-Taktes ist es möglich, sowohl in den Speicher zu schreiben als auch mehrere Werte aus dem Speicher auszulesen. So ist gewährleistet, daß während der Erstellung neuer Histogramme die der vorherigen Beobachtungsperiode auch ausgelesen werden können.

---

104 Name: HIST_ADR_MEM(22:0).

105 Name: HIST_WR_MEM.

106 Name: MEM_OF_CTR.

107 Name: MEM_DAT_HBR_B(15:0).

Die Kontrolle des zugeordneten Speichers findet über folgende Signale statt:

- die Adresse (ADR[108]) des Speichers,
- die übertragenen Daten (DAT[109]),
- das Schreibsignal (WR[110]) und
- die Aktivierung (DS[111]) eines Speicherzugriffs über die verschiedenen *Decoder*.

Das Symbol ist in der Abbildung 32 „Das Symbol des Memory“ auf Seite 88 beschrieben.

3.4 Daten

Die bisher beschriebenen Module stellen die notwendige Infrastruktur innerhalb der HEPI zur Verfügung. Diese wird von den folgenden Modulen benötigt, um die wissenschaftliche Verarbeitung der einzelnen detektierten Ereignisse je nach Modus vorzunehmen. Im Folgenden wird die dritte und letzte Gruppe der Module der HEPI erläutert. Dies sind die Module:

1. *Amplitude Correction* (AC) zur Korrektur der Amplituden-Werte für jedes individuelle Pixel des Cäsium-Jodid-Detektors,
2. *Multiple Event Reconstruction and Polarimetry* (MP) zur Rekonstruktion von Mehrfach-Ereignissen und deren Polarisation,
3. *Time Coincidence and Time Stamp* (TC) zur Erkennung von Comton-Ereignissen und Transformation des Zeitwertes der Ereignisse,
4. *Energy Selection* (ES) zur Auswahl bestimmter Ereignisse durch einen Amplituden-Bandpaß,
5. *Spectral Timing* (SPT) zur zeitlich hoch aufgelösten spektralen Analyse mit acht Energiekanälen und
6. *Histogram Module* (HIST) zur Erstellung spektraler und anderer Histogramme.

---

108 Name: MEM_ADR_B_MEM(19:0).

109 Name: MEM_DAT_B_MEM(15:0).

110 Name: MEM_WR_B_MEM.

111 Name: MEM_DS_B_MEM(4:0).

3.4.1 Lineare Amplituden-Korrektur: AC

Die lineare Amplituden-Korrektur korrigiert die Amplituden-Werte der Ereignisse aus dem Cäsium-Jodid-Detektor.

Wie in der Tabelle 14 „Das Datenformat der Detektoren“ auf Seite 73 zu sehen ist, wird die Amplitude eines Ereignisse des Cäsium-Jodid-Detektors mit 10 Bit an die HEPI übergeben. Das entspricht Amplituden-Werten von 0 keV bis 5 MeV. Da nicht jeder der 4.096 Cäsium-Jodid-Halbleiter-Zähler die gleichen Verstärkungseigenschaften hat, werden die gemessenen Amplituden für jedes Pixel des Detektors individuell korrigiert.

Die Amplitude (a) des Ereignisses (DAT[112]) wird dabei mit einem 10 Bit Verstärkungswert (*Gain*) zwischen Null und Zwei multipliziert, der eine Schrittweite von etwa zwei Tausendstel hat. Anschließend wird auf die berechnete neue Amplitude ein 6 Bit Wert (*Offset*) zwischen 0 keV und 512 keV addiert, der eine Schrittweite von etwa 8 keV hat. Die beiden Werte *Gain* und *Offset* sind zusammen 16 Bit breit. Sie werden nacheinander (WR[113]) von dem Modul zur Verfügung (DAT[114]) gestellt, das die Tabellen verwaltet. Die neue Amplitude (A) wird also wie folgt berechnet:

$$A = a * Gain + Offset$$

Gleichung 2: Amplituden-Korrektur

Falls die lineare Korrektur der Amplitude einen Wert größer als 5 MeV ergibt, wird der Wert auf 5 MeV gesetzt und ein Überlauf-Signal (OF[115]) generiert, welches in die *Housekeeping*-Daten eingeht.

Im *Schematic* ist der Datenfluß von links nach rechts gut zu erkennen. Die 2 mal 8 Bit aus dem *Look Up Table* Modul werden auf die 10 Bit *Gain* und 6 Bit *Offset* verteilt. Der Amplituden-Wert des Ereignisses wird zunächst durch den Multiplizierer und dann von dem Addierer manipuliert und danach ausgegeben. Eine interne Logik faßt die verschiedenen Überläufe zu einem Signal zusammen.

In der Simulation ist unter anderem zu erkennen, daß die lineare Amplituden-Korrektur für Amplituden-Werte[116] von 0 bis 9 , einem *Gain*[117] von 1 (Hexadezimal: 200) und einem *Offset*[118] von 1 jeweils das richtige Ergebnis[119] liefert.

112 Name: CSI_HBR_DAT(79:9)

113 Namen: LUT_WR_B1_AC und LUT_WR_B2_AC.

114 Name: LUT_DAT(7:0).

115 Name: AC_OF_CTR.

Dies sind die zugehörigen Abbildungen:

- Abbildung 33 „Das Symbol der Amplituden-Korrektur“ auf Seite 88,
- Abbildung 50 „Das Schematic der Amplituden-Korrektur“ auf Seite 99 und
- Abbildung 34 „Die Simulation der Amplituden-Korrektur“ auf Seite 89.

3.4.2 Rekonstruktion von Mehrfach-Ereignissen und Polarimetrie: MP

Das Modul zur Rekonstruktion von Mehrfach-Ereignissen und Polarimetrie rekonstruiert Mehrfach-Ereignisse und stellt, wenn möglich, die Polarisationsrichtung fest. Einfach-Ereignisse werden jedoch unverändert durchgelassen.

Detektiert der Cäsium-Jodid-Detektor in einem seiner acht Module innerhalb von 1 µs mehr als ein Ereignis, wird davon ausgegangen, daß ein Photon seine Energie in mehr als einem Halbleiter-Zähler deponiert hat. Diese Mehrfach-Ereignisse werden gezählt, und jedes dieser Ereignisse wird dann mit der Gesamtanzahl, der sogenannte *Multiplicity*[120], als Parameter seiner Daten (DAT[121]) ausgezeichnet. Die Elektronik des Detektors stellt sicher, daß jedes dieser Ereignisse mit der selben Zeitinformation versehen wird.

2	1	0
3	Ursprungs-Pixel	4
7	6	5

010	001	000
011	Ursprungs-Pixel	100
111	110	101

Abbildung 14: Die Definition der Polarimetrie-Ereignisse (Dezimal und Binär)

116 Name: DAT_CSI_AMP(9:0).

117 Name: MUL(9:0).

118 Name: ADD(5:0).

119 Name: SUM(9:0).

120 Siehe Tabelle 7: „Die Definition der Multiplicity Bits“ auf Seite 49 im Kapitel 3.2.2 „Cäsium-Jodid-Detektor: CsI_HBR“.

121 Name: AC_DAT_MP(79:0).

Nur 48% aller Photonen generieren Einfach-Ereignisse im Cäsium-Jodid-Detektor. Alle anderen Photonen generieren Zweifach-Ereignisse (34%) oder sogar Mehrfach-Ereignisse (18%)[122].

Zweifach- und Dreifach-Ereignisse werden im Modul wieder zu „Photonen“ rekonstruiert, also ihrem ursprünglichen Ereignis zugeordnet. Weiterhin wird dessen Gesamtenergie berechnet. Wenn bei Zweifach-Ereignissen zwei direkt benachbarte Pixel beteiligt sind, wird darüber hinaus der Winkel festgestellt, das *Type_Pol* Signal gesetzt und der Winkel wie in der Abbildung 14 „Die Definition der Polarimetrie-Ereignisse (Dezimal und Binär)“ digital kodiert.

Tabelle 10: Die Rekonstruktion von Zweifach-Ereignissen

Energie für Zweifach-Ereignisse	Ursprungspixel
Summe beider Energien **kleiner** als der Schwellwert 3	Pixel mit **kleinster** Energie
Summe beider Energien **größer** als der Schwellwert 3	Pixel mit **größter** Energie

Der bei der Rekonstruktion verwendete vereinfachte Algorithmus wählt ein Ursprungspixel aus, welches mit einer bestimmten Wahrscheinlichkeit[123] dem wirklichen Ursprungspixel entspricht. Siehe dazu Tabelle 10 „Die Rekonstruktion von Zweifach-Ereignissen“ und Tabelle 11 „Die Rekonstruktion von Dreifach-Ereignissen“.

Tabelle 11: Die Rekonstruktion von Dreifach-Ereignissen

Energie für Dreifach-Ereignisse	Ursprungspixel
Summe aller Energien **kleiner** als der Schwellwert 1	Pixel mit **kleinster** Energie
Summe aller Energien **zwischen** den Schwellwerten 1 & 2	Pixel mit **mittlerer** Energie
Summe aller Energien **größer** als der Schwellwert 2	Pixel mit **größter** Energie

Die Schwellwerte 0 bis 3[124] werden über die *Housekeeping*-Schnittstelle gesetzt, gelöscht und ausgelesen.

Bei der Rekonstruktion von Mehrfach-Ereignissen kann durch die Addition der einzelnen Energien eine Energie von mehr als 10 MeV herauskommen. Diese Ereignisse wer-

122 Siehe Lier, Frank, Hardware Ereignis Prozessor für den Imager auf dem INTEGRAL Satelliten, IAAT November 1996, Seite 63 ff.

123 Siehe Lier, Frank, Hardware Ereignis Prozessor für den Imager auf dem INTEGRAL Satelliten, IAAT November 1996, Seite 53 ff.

124 Interne Namen: TH, TH1, TH2 und TH3

den gezählt und dann verworfen. Ein Überlauf (CNT_OF[125]) dieses Zählers, der wie alle Register und Zähler über die standardisierte *Housekeeping*-Schnittstelle gelöscht und ausgelesen werden kann, wird dem *Controller* signalisiert. Die Energie der Dreifach-Ereignisse wird so kodiert, daß eine Energie von 0 bis 10 MeV (statt 5 MeV) mit einer Schrittweite von etwa 10 keV (statt 5 KeV) dargestellt werden kann.

Das Symbol des Moduls zur Rekonstruktion von Mehrfach-Ereignissen und Polarimetrie ist in der Abbildung 35 „Das Symbol der Ereignisrekonstruktion" auf Seite 89 beschrieben.

3.4.3 Zeitkoinzidenz: TC

Um sogenannte Zeitkoinzidenzen zwischen Ereignissen aus den beiden Detektor-Ebenen festzustellen und das Detektor-Zeitformat auf des DPE-Zeitformat abzubilden, wird ein weiteres Modul benötigt: das Zeitkoinzidenz-Modul.

Wenn innerhalb eines kleinen und einstellbaren Zeitfensters, das als Register über die *Housekeeping*-Schnittstelle gesetzt wird, in beiden Detektoren Ereignisse auftreten, hat mit einer bestimmten Wahrscheinlichkeit ein Photon durch Compton-Wechselwirkung seine Energie in beiden Detektor-Ebenen deponiert. Im Zeitkoinzidenz-Modul wird festgestellt, für welche Ereignisse (DAT[126]) dies zutrifft, und es wird gegebenenfalls ein Compton-Ereignis generiert, wofür dann die beiden Einzel-Ereignisse verworfen werden.

Das Verhalten der *Pipelines* der HEPI hängt empfindlich von dem Zeitkoinzidenz-Modul ab. Jedem der beiden Detektoren ist in der HEPI eine eigene *Pipeline* zugeordnet. In der Regel ist nur eine der beiden *Pipelines* aktiv. Ist eine *Pipeline* aktiv, dann bearbeiten die Module, die Stufen dieser Pipeline sind, je *Pipeline*-Takt ein Ereignis. Aufgrund der höheren Ereignisrate des Cäsium-Jodid-Detektors wird dies in den meisten Fällen die *Pipeline* für Cäsium-Jodid-Ereignisse sein.

Befindet sich die HEPI in einem der wissenschaftlichen Modi, wird im Zeitkoinzidenz-Modul festgestellt, welche Pipeline im nächsten *Pipeline*-Takt aktiv sein soll. Dem *Controller*, der beide *Pipelines* steuert, wird signalisiert (RQ[127]), die *Pipeline* mit dem jeweils „älteren" Ereignis (DAT[128]) im nächsten *Pipeline*-Takt zu aktivieren.

---

125 Name: MP_CNT_OF_CTR.

126 Namen: MP_DAT(79:0) und CDTE_HBR_DAT(79:0).

127 Namen: TC_RQ_CDTE_CTR und TC_RQ_CSI_CTR.

128 Namen: TC_CDTE_DAT(79:0) und TC_CSI_DAT(79:0).

Falls die Ereignisse beider *Pipelines* „gleich alt“ sind, wird ein sogenanntes Compton-Ereignis (DAT_COM[129]) generiert und dem *Controller* wird signalisiert, im nächsten *Pipeline*-Takt beide *Pipelines* zu aktivieren. Zwei Ereignisse sind „gleich alt“, wenn die Differenz ihrer Zeitwerte kleiner einer Schwelle ist. Dieses Verfahren führt zur einer zeitlichen Synchronisation der beiden *Pipelines*. Falls ein Photon mit beiden, hintereinander liegenden Detektor-Ebenen in Wechselwirkung tritt, wird es möglich, die dreidimensionale Flugbahn zu rekonstruieren.

Unter Umständen ist einer der beiden Detektoren nicht aktiv, so das die Aktivität (EN[130]) der Detektoren dem Zeitkoinzidenz-Modul mitgeteilt wird. So wird verhindert, daß für die Synchronisation nicht vergeblich Daten von einem inaktiven Detektor angefordert werden. Die Detektoren versehen jedes Ereignis mit einer Zeit, die innerhalb von 2 s auf 238,4 ns genau ist. Das Modul TC bildet diese Zeitinformation für jedes Ereignis auf eine Beobachtungszeit ab, die innerhalb von 72 h auf 61,04 µs genau ist. Siehe dazu die folgende Tabelle 12 „Die Rekonstruktion von Dreifach-Ereignissen“.

Tabelle 12: Einige Zeitauflösungen und die Clock[131]

Modus[132]	*Clock* Frequenz der HEPI	238,4 ns	4.194.304 Hz
Diagnostik	Zeitauflösung der Ereignisse der Detektoren	238,4 ns	innerhalb von 2 s
Sonst	Zeitauflösung der Ereignisse der HEPI	61,04 µs	innerhalb von 72,8 h

Jedesmal, wenn die Detektoren von den Detektor-Schnittstellen ein Synchronisations-Signal erhalten, stellen sie ihre innere Zeit wieder auf Null zurück. Die Zeitwerte werden also kleiner. Dies wird Nulldurchgang genannt. Die Nulldurchgänge werden gezählt, damit die Zeitwerte auf das DPE-Zeitformat abgebildet werden können. Für jede *Pipeline* gibt es einen Zähler. Ein Überlauf (CNT_OF[133]) dieser Zähler wird signalisiert.

Im Diagnostik-Modus ist das Zeitkoinzidenz-Modul jedoch nicht aktiv. Es findet also weder eine Synchronisation der *Pipelines* noch eine Abbildung der Zeitauflösung statt.

In dem *Schematic* ist der Datenfluß im Zeitkoinzidenz-Modul von links nach rechts zu sehen. Die Abbildung der Zeit wird durch eine Zusammenfassung der Bits aus den Zählern der Nulldurchgänge und einiger Bits der Detektorzeit generiert.

129 Name: TC_DAT_COM_HBR_A(159:0)

130 Namen: CTR_EN_CDTE und CTR_EN_CSI.

131 Siehe Volkmer, Reiner, HEPI Design Report, Issue 6.0, Juli 1998 IAAT, Seite 8.

132 Siehe Kapitel 2.2 „Modi“ auf Seite 39.

133 Namen: TC_CNT_OF_CDTE_CTR und TC_CNT_OF_CSI.

Die Simulation verdeutlicht, daß Compton-Ereignisse als solche erkannt werden. Die Schwelle[134] ist dabei auf 2 * 238,4 ns = 476,8 ns eingestellt. Der Zeitwert des Cäsium-Jodid-Ereignisses[135] ist 915,933 ms (Hexadezimal: F02). Der Zeitwert des Cadmium-Tellurid-Ereignisses[136] läuft von 914,979 ms bis 916,886 ms (Hexadezimal: von EFE bis F06). Die Ereignisse sind gültig[137] und kommen nicht aus der Kalibrations-Quelle. Es werden jeweils die richtigen Signale[138] generiert.

Dies sind die zugehörigen Abbildungen:

- Abbildung 36 „Das Symbol der Zeitkoinzidenzen“ auf Seite 89,
- Abbildung 51 „Das Schematic der Zeitkoinzidenz“ auf Seite 101 und
- Abbildung 37 „Die Simulation der Zeitkoinzidenzen“ auf Seite 90.

Zeitkoinzidenzen finden: Sub-Modul FIND_TC

Wenn im Symbol *FIND_TC* ein Compton-Ereignis (RQ_COM) gefunden wird, wird dies dem *Controller* des Zeitkoinzidenz-Moduls signalisiert. Dazu werden die Zeitwerte aus beiden *Pipelines* miteinander verglichen, und einige logische Betrachtungen werden angestellt. So werden zum Beispiel keine Compton-Ereignisse gefunden, wenn ein Kalibrations-Ereignis beteiligt ist. Außerdem werden hier die Signale (RQ_CDTE und RQ_CSI) erzeugt, die dem *Controller* des Zeitkoinzidenz-Moduls signalisieren, welche *Pipeline* prinzipiell weiter laufen sollte.

Die Compton-Ereignisse werden also von diesem Sub-Modul erkannt. Dies ist auch in der Simulation zu sehen, die anlog zu der Simulation des gesamten Moduls erstellt wurde. Einige Netznamen sind jedoch unterschiedlich. Die Schwelle[139] ist dabei auf 2 * 238,4 ns = 476,8 ns eingestellt. Der Zeitwert des Cäsium-Jodid-Ereignisses[140] ist 915,933 ms (Hexadezimal: F02). Der Zeitwert des Cadmium-Tellurid-Ereignisses[141] läuft von 914,979 ms bis 916,886 ms (Hexadezimal: von EFE bis F06). Die Ereignisse

134 Name: DELTA_TI(7:9).

135 Name: CSI_TI(31:0).

136 Name: CDTE_TI(31:0).

137 Namen: CSI_VD und CDTE_VD.

138 Namen: RQ_COM, RQ_CSI, RQ_CDTE, RNE_CSI, RNE_CDTE, VD_COM, VD_CSI und VD_CDTE.

139 Name: DELTA_TI(7:9).

140 Name: CSI_TI(31:0).

141 Name: CDTE_TI(31:0).

sind gültig[142] und kommen nicht aus der Kalibrations-Quelle[143]. Es werden jeweils die richtigen Signale[144] generiert.

Dies sind die zugehörigen Abbildungen:

- Abbildung 38 „Das Symbol des Sub-Moduls Find_TC“ auf Seite 90,
- Abbildung 39 „Das Schematic des Sub-Moduls Find_TC“ auf Seite 91 und
- Abbildung 40 „Die Simulation des Sub-Moduls FIND_TC“ auf Seite 91.

Controller: Sub-Modul TC_CTR

Das Symbol TC_CTR ist der *Controller* dieses Moduls. Hier werden je nach Zustand der anliegenden Signale die entsprechenden Steuersignale generiert. Zum Beispiel wird bei einem gültigen Compton-Ereignis das Gültigkeitssignal (VD[145]) für das Compton-Ereignis gesetzt.

In der Simulation ist zu erkennen, wie aus den Informationen[146] des Sub-Moduls FIND_TC die notwendigen Signale[147] für die Steuerung des Zeitkoinzidenz-Moduls generiert werden. Die anliegenden Daten aus beiden Detektoren sind gültig[148].

Dies sind die zugehörigen Abbildungen:

- Abbildung 41 „Das Symbol des Sub-Moduls TC_CTR“ auf Seite 92,
- Abbildung 42 „Das Schematic des Sub-Moduls TC_CTR“ auf Seite 92 und
- Abbildung 43 „Die Simulation des Sub-Moduls TC_CTR“ auf Seite 93.

3.4.4 Zusammenfassen von Ereignissen: HIST

Das Modul für das Zusammenfassen von Ereignissen erstellt die verschiedenen Histogramme für Kalibrations-, Polarimetrie-, Einfach- und Mehrfach-Ereignisse.

---

142 Namen: CSI_VD_IN und CDTE_VD_IN.

143 Namen: CSI_CAL und CDTE_CAL.

144 Namen: RQ_COM, RQ_CSI und RQ_CDTE.

145 Name: TC_COM_VD.

146 Namen: CSI_VD_IN und CDTE_VD_IN.

147 Namen: RQ_COM, RQ_CSI und RQ_CDTE.

148 Namen: RNE_CSI, RNE_CDTE, VD_COM, VD_CSI und VD_CDTE.

Welche Histogramme erstellt werden sollen, wird durch die DPE gesteuert. Diese veranlaßt per Telekommando den *Controller*, die entsprechenden Signale (EN[149]) zu setzen. Aus den gültigen Daten (DAT[150]) wird die Amplitude und der Typ des Ereignisses verwendet, um gegebenenfalls daraus eine Adresse (ADR[151]) zu bilden. Als Antwort darauf generiert das Modul, welches den Tabellenspeicher kontrolliert, eine Adresse (DAT[152]) für den entsprechenden Zähler des Histogramms und übergibt (WR[153]) diesen.

Je nach dem, welcher Speicherbereich (MEMA[154]) gerade aktiviert ist und welchem Typ das Ereignis angehört, wird daraus eine Adresse (ADR[155]) gebildet. Diese wird an die Speicherverwaltung übergeben (WR[156]), welches den Speicher verwaltet. Wie schon in Kapitel 3.3.3 „Speicherverwaltung: MEM“ auf Seite 56 beschrieben, addiert diese eins auf den dort abgelegten Wert und speichert ihn wieder.

Das Symbol ist in der Abbildung 44 „Das Symbol des Histogramm-Moduls“ auf Seite 93 beschrieben.

3.4.5 Amplituden-Bandpaß: ES

Der Amplituden Bandpaß beschränkt die erwartet hohe Datenrate des Cäsium-Jodid-Detektors. Es läßt nur Ereignisse mit jenen Amplituden-Werten durch, die innerhalb zweier Grenzen liegen (Bandpaß). Die Grenzen sind zwei Register, die über die *Housekeeping*-Schnittstelle gesetzt, gelöscht und ausgelesen werden. Ereignisse außerhalb dieser Grenzen werden verworfen.

Wie das *Schematic* erkennen läßt, ist das Gültigkeitssignal (VD[157]) nur dann aktiv, wenn die Eingangsdaten gültig waren, und die beiden Komperatoren bestätigen, daß die Amplitude des Ereignisses innerhalb der gewählten Grenzen liegt. Der Amplituden Bandpaß ist in die Schnittstelle A mit aufgenommen worden.

In der Simulation ist unter anderem zu erkennen, daß Ereignisse, die außerhalb der Grenzen liegen, nicht gültig sind. Die untere Grenze[158] ist etwa 14,65 keV (Hexadezi-

149 Namen: CTR_EN_SM_HIST, CTR_EN_POL_HIST und CTR_EN_CAL_HIST.

150 Name: MP_DAT(79:0).

151 Name: HIST_ADR_LUT(11:0).

152 Name: LUT_DAT(7:0).

153 Name: LUT_WR_HIST.

154 Namen: CTR_MEMA_HIST und CTR_MEMA_CAL_HIST.

155 Name: HIST_ADR_MEM(22:0).

156 Name: HIST_WR_MEM.

157 Name: ES_VD_HBR_A.

158 Name: CTR_TH_L_ES(7:0).

mal: 03) und die obere Grenze[159] ist etwa 1.171,88 keV (Hexadezimal: F0). Diese Werte werden in die entsprechenden Register[160] geladen[161] und dann werden drei gültige[162] Ereignisse[163] nacheinander untersucht, wobei nur eines[164] den Bandpaß passiert[165].

Dies sind die zugehörigen Abbildungen:

- Abbildung 45 „Das Symbol des Amplituden-Bandpasses“ auf Seite 93,
- Abbildung 46 „Das Schematic des Amplituden-Bandpasses“ auf Seite 94 und
- Abbildung 47 „Die Simulation des Amplituden-Bandpasses“ auf Seite 95.

3.4.6 Zeitlich aufgelöste Spektroskopie: SPT

Dieses Modul für zeitlich aufgelöste Spektroskopie erlaubt eine zeitlich hoch aufgelöste Spektroskopie der Ereignisse des Cäsium-Jodid-Detektors.

Durch neun Parameter, die über die *Housekeeping*-Schnittstelle gesetzt werden, werden acht Energiebereiche definiert, über die dieses Modul eine bestimmbare Zeit von 0,976 ms bis zu 500 ms integriert. Diese 8 Zähler werden nach Ablauf der Zeit in ein 180 Bit breites Datenformat (DAT[166]) übertragen und an die Schnittstelle A übergeben. Falls einer dieser Zähler überläuft (CNT_OF[167]), wird dies dem *Controller* signalisiert.

Falls im Zeitkoinzidenz-Modul der Zähler für die Nulldurchgänge der Zeitwerte des Cäsium-Jodid-Ereignisse überläuft (CNT_OF[168]), wird dies dem Modul für zeitlich aufgelöste Spektroskopie mitgeteilt. So wird verhindert, daß dessen Steuerung nicht vergeblich auf den Ablauf der Integrationsperiode wartet.

Das Symbol ist in der Abbildung 48 „Das Symbol der zeitaufgelösten Spektroskopie“ auf Seite 95 beschrieben.

159 Name: CTR_TH_H_ES(7:0).

160 Name: ES_TH_L_ES(7:0) und ES_TH_H_ES(7:0).

161 Name: CTR_LD_TH_L_ES und CTR_LD_TH_L_ES.

162 Name: TC_CSI_VD.

163 Name: TC_CSI(79:0).

164 Name: ES_DAT_HBR_A(79:0).

165 Name: ES_VD_HBR_A.

166 Name: SPT_DAT_HBR_A(159:0).

167 Name: SPT_CNT_OF_CTR.

168 Name: TC_CNT_OF_CSI.

3.4.7 Erkennen von Gamma Ray Bursts: BD

Das Modul zum Erkennen von *Gamma Ray Bursts* erkennt einen sprunghaften Anstieg der Ereignisraten in einem oder in beiden Detektor-Ebenen.

Anhand sprunghafter Anstiege der Ereignisrate können sogenannte *Gamma Ray Bursts* erkannt werden. Bis zur Jahreswende 1997 / 1998 war ein Modul zum Erkennen von *Gamma Ray Bursts*, der *Burst Detector* (BD), ein zyklischer Speicher für die Speicherung der Ereignisse vor dem *Burst,* der *Pre Burst Recorder* (PBR), sowie ein zusätzlicher *Burst-Mode*, der den gesamten Speicher überschrieben hätte, vorgesehen. Diese Funktionalität wurde vom IBIS-Konsortium gestrichen. Zu diesem Zeitpunkt war dieses Modul, der *Burst Detector*, kurz vor seiner Fertigstellung.

3.5 Resümee

Ein Projekt wie die Planung, Fertigstellung und Nutzung eines Satelliten ist eine große Herausforderung für alle beteiligten Personen und Institutionen. Erfahrung und Wissen aus nahezu allen Bereichen der Technik und Wissenschaften sind dazu notwendig. Nur die internationale Zusammenarbeit ermöglicht die Realisierung dieser kostenintensiven Projekte.

Der Satellit INTEGRAL ist mit seinen vier Instrumenten und seiner Höhe von etwa 3 Stockwerken nicht nur imposant, sondern auch interessant. Allein die Konzepte zur Redundanz sind ein spannender Teilaspekt. Die HEPI erscheint dabei wie ein kleines Zahnrädchen einer großen Uhr.

Die Auseinandersetzung mit den verschiedenen Aufgaben der HEPI haben mich in sehr unterschiedliche und interessante Gebiete geführt. Sei es die aktuelle, spannende Entwicklung in der Erforschung der *Gamma Ray Bursts* oder die Theorie relationaler Datenbanken.

Bei der Entwicklung und Realisierung der HEPI hat sich gezeigt, daß es richtig war, die sehr komplexe Aufgabenstellung mit einer mindestens genauso komplexen Vielfalt an Werkzeugen und Programmen zu bewältigen. Es war jedoch zu Anfang nicht festzustellen, ob der erhebliche Aufwand für die Entwicklung der Datenbank und der Programme für die Abbildung des Inhalts der Datenbank auf das *Top Level* gerechtfertigt ist.

Im Rückblick wäre die termingerechte Fertigstellung der Entwicklung der HEPI jedoch anders nicht möglich gewesen, denn andernfalls hätten die Modifikationen zu Ende der Projektzeit keinesfalls realisiert werden können.

Wie in Abbildung 1 „Ein Foto der HEPI-Platine und der HEPI (FPGA)“ auf Seite 11 zu sehen ist, wird die HEPI zur Zeit intensiv getestet. Für diese Tests werden unter anderem ein sogenannter *Spacecraft Simulator*[169], ein *DPE Simulator*[170], eine DPE und der

169 Siehe Satellite Services B.V., Integral SIS, ICD, Issue 1, Revision 1, April 1996, Seite 12 ff.

170 Siehe ebd., Seite 84 ff.

am Institut für Astronomie und Astrophysik Tübingen entwickelte Detektor-Simulator verwendet.

Im Gespräch mit Fachleuten, die Erfahrung auf dem Gebiet der Entwicklung solcher FPGAs und ASICs haben, hat dieses Verfahren Staunen erregt und Neugier geweckt. Eine Recherche im *Internet* hat ergeben, daß es meines Wissens zwar Entwicklungsansätze für die Datenbank gestützte Entwicklung von FPGAs und ASICs gibt, aber kein entsprechendes Produkt auf dem Markt ist. Um dieses Verfahren für andere Entwicklungen zu verwenden, wären allerdings einige Ergänzungen notwendig.

Um den Termin einzuhalten, konnte ich leider eine wichtige Verbesserung der Entwicklung nicht realisieren: Da die Module immer nur bestimmte Teile der Daten manipulieren, wäre es möglich gewesen, statt der 80 Bit breiten lediglich einen 3 Bit-Zeiger, der auf einen entsprechenden Ereignisspeicher zeigt, durch die *Pipeline* zu schieben. Dann hätte es vollkommen gereicht, nur den entsprechenden Teil der Daten über einen bidirektionalen Bus zu übertragen. Ein solches Konzept würde weniger Ressourcen verbrauchen. Darüber hinaus wäre es interessant, die gesamte Entwicklung unter VHDL neu zu erstellen und beide Entwicklungen miteinander zu vergleichen.

Durch die nun folgenden Tests der HEPI und der sukzessiven Integration des *Imager* IBIS werden eventuell einige kleinere Änderungen der HEPI notwendig. Ich hoffe, daß die von mir erstellten Werkzeuge und weiteren ausführlichen Dokumentationen dabei weiterhin gute Dienste leisten.

4 Anhang

4.1 Verzeichnis der Literatur

Autor / Autorin	Titel
Bibliographisches Institut	Das neue Duden Lexikon, 2. Auflage, Bibliographisches Institut Mannheim, Wien, Zürich 1989
Decker, Oliver	Entwicklung einer Software Simulation des Hardware Event Pre-Processors des INTEGRAL Satelliten, IAAT 1998
Dunn, Melissa	Creating Access Applications, Que Corporation, Indianapolis 1994
ESA	INTEGRAL Experiment Interface Document Part A for Instruments, Issue 1 Revision 5, Oktober 1997
ESA	INTEGRAL System Requirements Document, Issue 2 Revision 1, 25.5.98
IBIS Team	INTEGRAL Experiment Interface Document Part B for Instruments, Issue 4 Revision 1, 30.7.1998
IBIS Team	Scientific Performance Report, Issue 3 Revision 1, 30.6.98
Kniefen, et al.	Nature, Bd. 269, 1974
Lier, Frank	Hardware Ereignis Prozessor für den Imager auf dem INTEGRAL Satelliten, IAAT November 1996
Rilk, Albrecht J.	Detektor System zum Test eines Hardwareprozessors des Satelliten INTEGRAL, IAAT 1996
S. Bergeson-Willis et al.	Integral, Report on the Phase-A Study
Satellite Services B.V.	Integral SIS, ICD, Issue 1, Revision 1, April 1996
Software und Manager Meeting	Minutes of Meeting, 20. Januar 1998, Tübingen, IN-IM-TUB-9801-I
Swanenburg, et al.	Nature, Bd. 275, 1978
Thompson, et al.	Astrophysical Journal, (Letters), Bd. 200, 1975
Tipler, Paul A.	Physik, Spektrum Akademischer Verlag, Berlin, 1994
Trimble, Virginia	„When one photon was a discovery, two was a spectrum, and three was a Rossi Prize", Documents from the Beginnings of Gamma Ray Astronomy, AIP Conference Proceedings 304, pp. 40-44, AIP Press, New York 1993
Tua, P.F.	User Manual for the IBIS Instrument, Issue 1, Juni 1998
Volkmer, Reiner	HEPI Design Report, Issue 6.0, Juli 1998 IAAT
Volkmer, Reiner	HEPI Interface Description, 4.1, Juni 1998 IAAT

4.2 Verzeichnis der Abkürzungen

Abkürzung	Beschreibung	erste Erwähnung auf Seite
AOCS	*Attitude and Orbit Control System*	21
ASIC	*Application Specific Integrated Circuit*	10
CAD	*Computer Aided Design*	33
CDMU	*Central Data Management Unit*	21
CdTe	Cadmium-Tellurid	25
CGRO	*Compton Gamma Ray Observatory*	13
CsI	Cäsium-Jodid	26
DPE	*Data Processing Electronics*	9
ESA	*European Space Agency*	9
FCFOV	Fully Coded Field of View	15
FIFO	First In, First Out	47
FPGA	*Field Programmable Gate Array*	10
GRB	*Gamma Ray Bursts*	15
HEO	*Highly eccentric orbit*	18
HEPI	*Hardware Event Processor for IBIS*	10
IAAT	Institut für Astronomie und Astrophysik in Tübingen	11
IBIS	*Imager on Board the INTEGRAL Satellite*	9
INTEGRAL	*International Gamma Ray Astrophysics Laboratory*	9
ISDC	*INTEGRAL Science Data Center*	22
ISGRI	zugewandte Detektor-Ebene heißt *INTEGRAL Soft Gamma Ray Imager*	23
ISOC	Das Integral Science Operations Center	23
JEM-X	*Joint European Monitor for X-Rays*	9
LCA	*Logic Cell Array*	37
MOC	*Mission Operational Center*	22
OBDH	*On Board Data Handling System*	21
OMC	*Optical Monitoring Camera*	9
PICSIT	*Pixelated Imaging Cesium Iodide Telescope*	23
RTU	*Remote Terminal Unit*	21
SPI	*Spectrometer on INTEGRAL*	9
VHDL	*Very High Speed Integrated Circuit Hardware Description Language*	37
XMM	*X-Ray Multi Mirror*	9

4.3 Signalnamen und Datenformate

Tabelle 13: Die Signalnamen

Befehl	Erklärung
ADR	Address
BA	Burst Alert
CI	Clock Inhibit
CLK	Clock
CLR	Clear
CLR_CNT	Clear Counter
CLR_FIFO	Clear FIFO
CNT_OF	Counter Over Flow
CNT_OF_CdTe	Counter Overflow Cadmium Telluride
CNT_OF_CsI	Counter Overflow Cesium Iodide
CS	Chip Select
DAT	Data
Dat_CdTe	Data Cadmium Telluride
Dat_Com	Data Compton
Dat_CsI	Data Cesium Iodide
DS	Decoder Select
DTF	Data Forward
DTR	Data Return
EN	Enable
EN_Cal	Enable Calibration Histogram
EN_CdTe	Enable CdTe_HBR
EN_CdTe_PPM	Enable Cadmium Telluride Photon by Photon Mode
EN_CsI	Enable CsI_HBR
EN_CsI_PPM	Enable Cesium Iodide PPM
EN_Dia	Enable Cesium Iodide Diagnostic
EN_Pol	Enable Polarimetry (MP)
EN_RM	Enable Ratemeter
EN_SM	Single and Multiplicity Events
FE	FIFO Empty
FF	FIFO Full
HF	FIFO Half Full
HK	Housekeeping
HK_Ack	Housekeeping Acknowledge
HK_CLR	Housekeeping Clear
HK_IO	Housekeeping Input and Output Data
HK_Mod	House Keeping Module
HK_RD	House Keeping Read
HK_VD	House Keeping Valid
HK_WR	House Keeping Write
INIT	Initialize
MEMA	Memory Area
MEMA_Cal	Memory Area Calibration
NE	FIFO Not Empty
OF	Over Flow
PARR	Parity Error
RD	Read
RES	Reset
RNE	Read Next Event
RNE_CdTe	Read Next Event Cadmium Telluride
RNE_CsI	Read Next Event Cesium Iodide
RQ	Request for Next Data
RQ_CdTe	Request Cadmium Telluride Data
RQ_CsI	Request Cesium Iodide Data
RW	Read
RXRDY	Receive Ready
SE	Scan Enable
SH_LD	Shift Load
SI	Scan In
SP	Scan Path
SYNC	Synchronize
TO	Transceiver Overrun Error
TX_EN	Transmitter enable
TXRDY	Transceiver Ready
VD	Valid Data
VD_CdTe	Valid Data Cadmium Telluride
VD_Com	Valid Data Compton
VD_CsI	Valid Data Cesium Iodide
WR	Write
WR_B1	Write Byte 1: Multiplikator(9:8) Summand(5:0)
WR_B2	Write Byte 2: Multiplikator (7:0)

Tabelle 14: Das Datenformat der Detektoren

ESA MSB	HEPI MSB	Dat_CsI_Det	Dat_CdTe_Det	ESA	HEPI	Dat_CsI_Det	Dat_CdTe_Det
0	63	Spare	Spare	32	31	Type_Cal	Type_Cal
1	62	Dat_Ti	Dat_Ti	33	30	Type_MMC	Spare
2	61	Dat_Ti	Dat_Ti	34	29	Type_Pulse_ID	Spare
3	60	Dat_Ti	Dat_Ti	35	28	Type_Pulse_ID	Spare
4	59	Dat_Ti	Dat_Ti	36	27	Type_Mux	Spare
5	58	Dat_Ti	Dat_Ti	37	26	Type_Mux	Dat_CdTe_Amp
6	57	Dat_Ti	Dat_Ti	38	25	Dat_CsI_Amp	Dat_CdTe_Amp
7	56	Dat_Ti	Dat_Ti	39	24	Dat_CsI_Amp	Dat_CdTe_Amp
8	55	Dat_Ti	Dat_Ti	40	23	Dat_CsI_Amp	Dat_CdTe_Amp
9	54	Dat_Ti	Dat_Ti	41	22	Dat_CsI_Amp	Dat_CdTe_Amp
10	53	Dat_Ti	Dat_Ti	42	21	Dat_CsI_Amp	Dat_CdTe_Amp
11	52	Dat_Ti	Dat_Ti	43	20	Dat_CsI_Amp	Dat_CdTe_Amp
12	51	Dat_Ti	Dat_Ti	44	19	Dat_CsI_Amp	Dat_CdTe_Amp
13	50	Dat_Ti	Dat_Ti	45	18	Dat_CsI_Amp	Dat_CdTe_Amp
14	49	Dat_Ti	Dat_Ti	46	17	Dat_CsI_Amp	Dat_CdTe_Amp
15	48	Dat_Ti	Dat_Ti	47	16	Dat_CsI_Amp	Dat_CdTe_Amp
16	47	Dat_Ti	Dat_Ti	48	15	Spare	Spare
17	46	Dat_Ti	Dat_Ti	49	14	Spare	Spare
18	45	Dat_Ti	Dat_Ti	50	13	Spare	Dat_CdTe_Y_Pos
19	44	Dat_Ti	Dat_Ti	51	12	Spare	Dat_CdTe_Y_Pos
20	43	Dat_Ti	Dat_Ti	52	11	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
21	42	Dat_Ti	Dat_Ti	53	10	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
22	41	Dat_Ti	Dat_Ti	54	9	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
23	40	Dat_Ti	Dat_Ti	55	8	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
24	39	Spare	Dat_Rise_Ti	56	7	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
25	38	Spare	Dat_Rise_Ti	57	6	Dat_CsI_Y_Pos	Dat_CdTe_Z_Pos
26	37	Spare	Dat_Rise_Ti	58	5	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
27	36	Spare	Dat_Rise_Ti	59	4	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
28	35	Spare	Dat_Rise_Ti	60	3	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
29	34	Spare	Dat_Rise_Ti	61	2	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
30	33	Spare	Dat_Rise_Ti	62	1	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
31	32	Spare	Dat_Rise_Ti	63	0	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos

Tabelle 15: Das Datenformat der DPE für CsI- und CdTe Ereignisse

ESA	HEPI	Dat_CsI	Dat_CdTe	ESA	HEPI	Dat_CsI	Dat_CdTe
MSB	MSB						
0	79	Type_Format	Type_Format	40	39	Dat_Ti	Dat_Ti
1	78	Type_Det	Type_Det	41	38	Dat_Ti	Dat_Ti
2	77	Type_Raw_BAR	Type_Raw_BAR	42	37	Dat_Ti	Dat_Ti
3	76	Type_Cal	Type_Cal	43	36	Dat_Ti	Dat_Ti
4	75	Type_Mux	Type_Spare	44	35	Dat_Ti	Dat_Ti
5	74	Type_Mux	Type_Spare	45	34	Dat_Ti	Dat_Ti
6	73	Type_SPT	Type_SPT	46	33	Dat_Ti	Dat_Ti
7	72	Type_Pol	Type_Pol	47	32	Dat_Ti	Dat_Ti
8	71	Dat_Pol	Dat_Rise_Ti	48	31	Dat_Spare	Dat_Spare
9	70	Dat_Pol	Dat_Rise_Ti	49	30	Dat_Spare	Dat_Spare
10	69	Dat_Pol	Dat_Rise_Ti	50	29	Dat_Spare	Dat_Spare
11	68	Dat_Spare	Dat_Rise_Ti	51	28	Dat_Spare	Dat_Spare
12	67	Dat_Spare	Dat_Rise_Ti	52	27	Dat_Spare	Dat_Spare
13	66	Dat_Spare	Dat_Rise_Ti	53	26	Dat_Spare	Dat_CdTe_Amp
14	65	Dat_Spare	Dat_Rise_Ti	54	25	Dat_CsI_Amp	Dat_CdTe_Amp
15	64	Dat_Spare	Dat_Rise_Ti	55	24	Dat_CsI_Amp	Dat_CdTe_Amp
16	63	Dat_Ti	Dat_Ti	56	23	Dat_CsI_Amp	Dat_CdTe_Amp
17	62	Dat_Ti	Dat_Ti	57	22	Dat_CsI_Amp	Dat_CdTe_Amp
18	61	Dat_Ti	Dat_Ti	58	21	Dat_CsI_Amp	Dat_CdTe_Amp
19	60	Dat_Ti	Dat_Ti	59	20	Dat_CsI_Amp	Dat_CdTe_Amp
20	59	Dat_Ti	Dat_Ti	60	19	Dat_CsI_Amp	Dat_CdTe_Amp
21	58	Dat_Ti	Dat_Ti	61	18	Dat_CsI_Amp	Dat_CdTe_Amp
22	57	Dat_Ti	Dat_Ti	62	17	Dat_CsI_Amp	Dat_CdTe_Amp
23	56	Dat_Ti	Dat_Ti	63	16	Dat_CsI_Amp	Dat_CdTe_Amp
24	55	Dat_Ti	Dat_Ti	64	15	Dat_Spare	Dat_Spare
25	54	Dat_Ti	Dat_Ti	65	14	Dat_Spare	Dat_CdTe_Y_Pos
26	53	Dat_Ti	Dat_Ti	66	13	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
27	52	Dat_Ti	Dat_Ti	67	12	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
28	51	Dat_Ti	Dat_Ti	68	11	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
29	50	Dat_Ti	Dat_Ti	69	10	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
30	49	Dat_Ti	Dat_Ti	70	9	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
31	48	Dat_Ti	Dat_Ti	71	8	Dat_CsI_Y_Pos	Dat_CdTe_Y_Pos
32	47	Dat_Ti	Dat_Ti	72	7	Dat_Spare	Dat_Spare
33	46	Dat_Ti	Dat_Ti	73	6	Dat_Spare	Dat_CdTe_Z_Pos
34	45	Dat_Ti	Dat_Ti	74	5	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
35	44	Dat_Ti	Dat_Ti	75	4	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
36	43	Dat_Ti	Dat_Ti	76	3	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
37	42	Dat_Ti	Dat_Ti	77	2	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
38	41	Dat_Ti	Dat_Ti	78	1	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos
39	40	Dat_Ti	Dat_Ti	79	0	Dat_CsI_Z_Pos	Dat_CdTe_Z_Pos

Tabelle 16: Das Datenformat der DPE für Compton-Ereignisse und Spectral Timing

ESA	HEPI	Dat_Com	Dat_SPT	ESA	HEPI	Dat_Com	Dat_SPT
MSB	MSB			MSB	MSB		
0	159	Type_Format	Type_Format	80	79	Dat_Spare	Dat_SPT
1	158	Type_Det	Type_Det	81	78	Dat_Spare	Dat_SPT
2	157	Type_Raw_BAR	Type_Raw_BAR	82	77	Dat_Spare	Dat_SPT
3	156	Type_Cal	Type_Cal	83	76	Dat_Spare	Dat_SPT
4	155	Type_Mux	Type_Mux	84	75	Dat_Spare	Dat_SPT
5	154	Type_Mux	Type_Mux	85	74	Dat_Spare	Dat_SPT
6	153	Type_SPT	Type_SPT	86	73	Dat_CsI_Amp	Dat_SPT
7	152	Type_Pol	Type_Pol	87	72	Dat_CsI_Amp	Dat_SPT
8	151	Dat_Rise_Ti	Dat_Pol	88	71	Dat_CsI_Amp	Dat_SPT
9	150	Dat_Rise_Ti	Dat_Pol	89	70	Dat_CsI_Amp	Dat_SPT
10	149	Dat_Rise_Ti	Dat_Pol	90	69	Dat_CsI_Amp	Dat_SPT
11	148	Dat_Rise_Ti	Dat_Spare	91	68	Dat_CsI_Amp	Dat_SPT
12	147	Dat_Rise_Ti	Dat_Spare	92	67	Dat_CsI_Amp	Dat_SPT
13	146	Dat_Rise_Ti	Dat_Spare	93	66	Dat_CsI_Amp	Dat_SPT
14	145	Dat_Rise_Ti	Dat_Spare	94	65	Dat_CsI_Amp	Dat_SPT
15	144	Dat_Rise_Ti	Dat_Spare	95	64	Dat_CsI_Amp	Dat_SPT
16	143	Dat_Ti	Dat_Ti	96	63	Dat_Spare	Dat_SPT
17	142	Dat_Ti	Dat_Ti	97	62	Dat_Spare	Dat_SPT
18	141	Dat_Ti	Dat_Ti	98	61	Dat_CsI_Y_Pos	Dat_SPT
19	140	Dat_Ti	Dat_Ti	99	60	Dat_CsI_Y_Pos	Dat_SPT
20	139	Dat_Ti	Dat_Ti	100	59	Dat_CsI_Y_Pos	Dat_SPT
21	138	Dat_Ti	Dat_Ti	101	58	Dat_CsI_Y_Pos	Dat_SPT
22	137	Dat_Ti	Dat_Ti	102	57	Dat_CsI_Y_Pos	Dat_SPT
23	136	Dat_Ti	Dat_Ti	103	56	Dat_CsI_Y_Pos	Dat_SPT
24	135	Dat_Ti	Dat_Ti	104	55	Dat_Spare	Dat_SPT
25	134	Dat_Ti	Dat_Ti	105	54	Dat_Spare	Dat_SPT
26	133	Dat_Ti	Dat_Ti	106	53	Dat_CsI_Z_Pos	Dat_SPT
27	132	Dat_Ti	Dat_Ti	107	52	Dat_CsI_Z_Pos	Dat_SPT
28	131	Dat_Ti	Dat_Ti	108	51	Dat_CsI_Z_Pos	Dat_SPT
29	130	Dat_Ti	Dat_Ti	109	50	Dat_CsI_Z_Pos	Dat_SPT
30	129	Dat_Ti	Dat_Ti	110	49	Dat_CsI_Z_Pos	Dat_SPT
31	128	Dat_Ti	Dat_Ti	111	48	Dat_CsI_Z_Pos	Dat_SPT
32	127	Dat_Ti	Dat_Ti	112	47	Dat_Long_Spare	Dat_Long_Spare
33	126	Dat_Ti	Dat_Ti	113	46	Dat_Long_Spare	Dat_Long_Spare
34	125	Dat_Ti	Dat_Ti	114	45	Dat_Long_Spare	Dat_Long_Spare
35	124	Dat_Ti	Dat_Ti	115	44	Dat_Long_Spare	Dat_Long_Spare
36	123	Dat_Ti	Dat_Ti	116	43	Dat_Long_Spare	Dat_Long_Spare
37	122	Dat_Ti	Dat_Ti	117	42	Dat_Long_Spare	Dat_Long_Spare
38	121	Dat_Ti	Dat_Ti	118	41	Dat_Long_Spare	Dat_Long_Spare

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

4.4 Quelltexte und Ausgaben

Quelltext des Constrain File Generators

```
Sub Make_Constrain()
       ' Create a Constrain File according to the Routing given by the Database
       ' © 1998 Nikolai von Krusenstiern
    Max = DCount("[Daten Name]", "FPGA_IO_Constrains")
    Min = 0
    Set dbs = CurrentDb
              ' Zeiger auf aktuelle Datenbank öffnen
       Set rst1 = dbs.OpenRecordset("FPGA", dbOpenDynaset)
              ' Zeiger auf Tabelle FPGA zum ändern öffnen
       Set rst2 = dbs.OpenRecordset("FPGA_IO_Constrains", dbOpenDynaset)
              ' Zeiger auf Tabelle FPGA_IO_Constrains zum ändern öffnen
       strCriteria1 = "[Prozess] is not null AND [Pin Name] is Null"
              ' Kriterien festlegen.
       rst1.FindFirst strCriteria1
              ' Zum ersten Datensatz gehen, der einen Wert in "Prozess", und keinen "Pin Namen" hat
       Do Until rst1.NoMatch
                     ' Andere passende Datensätze suchen.
              strProzess = rst1.Fields("Prozess")
                     ' gefunden Prozessnamen auslesen
              intPin = rst1.Fields("Pin")
                     ' gefunden "Pin" Nummer auslesen
              strCriteria2 = "([Feste Start Pin Nummer]=Null) And ([Prozess]='" & strProzess & "' )"
                     ' daraus Kriterien festlegen
              rst2.FindFirst strCriteria2
                     ' Zum Datensatz, der selben Wert "Prozess" und keine "Feste Start Pin Nummer"
              If rst2.NoMatch Then
                     rst1.FindNext strCriteria1
                            ' Zum NÄCHSTEN Datensatz, der Wert in "Prozess" hat, keinen "Pin Namen"
              Else
                            ' Andere passende Datensätze suchen.
                     If Min = 0 Then
                            DoCmd.Echo True, "55% fertig"
                     End If
                     strPinName = rst2.Fields("Pin Name")
                            ' gefunden "Pin Namen" auslesen
' Debug.Print strProzess, intPin, strPinName
                     rst1.Edit
                            ' Bearbeiten aktivieren.
' Debug.Print ""
' Debug.Print "1 Vorher:   " & rst1.Fields("Pin Name")
                     rst1![Pin Name] = strPinName
                            ' "Pin Name" in "FPGA" eintragen.
                     rst1.Update
                            ' Änderungen speichern
' Debug.Print "2 Nachher: " & rst1.Fields("Pin Name")
                     rst2.Edit
                            ' Bearbeiten aktivieren.
' Debug.Print "3 Vorher:   "; rst2.Fields("Feste Start Pin Nummer")
                     rst2![Feste Start Pin Nummer] = intPin
                            ' "Feste Start Pin Nummer" in "FPGA_IO_Constrains" eintragen.
                     rst2.Update
                            ' Änderungen abspeichern.
' Debug.Print "4 Nachher: "; rst2.Fields("Feste Start Pin Nummer")
              rst1.FindFirst strCriteria1
                     ' Zum "Neuen" Datensatz gehen, der Wert in "Prozess" hat, und keinen "Pin Namen"
              End If
              If (Min Mod 100 = 0) Then
                     Num = DCount("[Feste Start Pin Nummer]", "FPGA_IO_Constrains")
                     Prozent = CInt(40 * Num / Max + 50)
                     DoCmd.Echo True, Prozent & "% fertig"
              End If
              Min = Min + 1
       Loop
              ' Ende der Schleife.
rst1.Close
rst2.Close
' Debug.Print Date; Time; "Fertig"
              ' Für das Debugging einen "Fertig" Meldung ausgeben
              ' Ende der SUB Routine
End Sub
```

Quelltext des Symbolgenerators

```
#!/bin/csh -f
   # Create a Symbol according to Symbol File created by the Database
   # © 1998 Nikolai von Krusenstiern
set symbol_name  = $1
set FPGA  = ~/FPGA
set tmp_file  = $FPGA/tmp/$symbol_name
set in_path  = $FPGA/pins
set out_path  = $FPGA/COMP_LIB
set std_out  = $FPGA/io/$symbol_name.std_out
set std_err  = $FPGA/io/$symbol_name.std_err
touch $in_path/$symbol_name.input
touch $in_path/$symbol_name.output
touch $in_path/$symbol_name.bidir
set nu_of_i  = `wc $in_path/$symbol_name.input    | awk '{print $1}' `
set nu_of_o  = `wc $in_path/$symbol_name.output | awk '{print $1}' `
set nu_of_io  = `wc $in_path/$symbol_name.bidir    | awk '{print $1}' `
set p_space  = 2
set y_counter = 1
set text_height  = '1.60'
set text_height_small  = '0.5'
set text_height_big = '5'
set xh    = 30
set yh    = `expr $nu_of_io + $nu_of_o`
set yih      = $nu_of_i
set yoh      = `expr $nu_of_io + $nu_of_o`
if ( $nu_of_i > $yh ) then
   set yh = $nu_of_i
endif
set x0    = 0
set y0    = 0
set x1    = `expr   $p_space \* $xh  + $p_space   + $x0 `
set y1    = `expr   $p_space \* $yh  + $p_space   + $y0 `
set x0_left  = `expr $x0 - 1  `
set x0_right  = `expr $x0 + 1  `
set x0_prop  = `expr $x0 - 2  `
set x1_left  = `expr $x1 - 1  `
set x1_right  = `expr $x1 + 1  `
set x1_prop  = `expr $x1 + 2  `
set y0_down  = `expr $y0 - $text_height_big - 1    `
echo '$set_active_window("session");' > $tmp_file
echo '$$open_symbol("'$out_path'/'$symbol_name'", "", @editable, "");' >> $tmp_file
echo '$select_all(@comment, @pin, @property, @symbolbody, @text);' >> $tmp_file
echo '$delete();' >> $tmp_file
echo '$add_rectangle([['$x0', '$y0'], ['$x1', '$y1']]);' >> $tmp_file
echo '$set_text_height('$text_height_big');' >>$tmp_file
echo '$add_text("'$symbol_name'", ['$x0', '$y0_down', "Symbol#1.0"]);' >> $tmp_file
echo '$setup_property_text("stroke", '$text_height_big', @left, @top, 0, @on, @visible, @fixed,
   @instance);' >> $tmp_file
echo '$change_property_value("INST", "'$symbol_name'", @string);' >> $tmp_file
foreach pin_name (`cat $in_path/$symbol_name.input`)
      set y_pin = `expr $y_counter \* $p_space`
      set y_pintype = `expr $y_pin - 1`
   echo '$add_polyline([[ '$x0_left' , '$y_pin' , "Symbol#1.0"], [ '$x0' , '$y_pin',
      "Symbol#1.0"]]);' >> $tmp_file
   echo '$unselect_all(@comment, @pin, @property, @symbolbody, @text);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height', @left, @bottom, 0, @on, @visible, @fixed,
      @instance);' >> $tmp_file
   echo '$add_pin("'$pin_name'", [ '$x0_left', '$y_pin' , "Symbol#1.0"], [ '$x0_right' , '$y_pin',
      "Symbol#1.0"]);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height_small', @left, @bottom, 0, @on, @visible,
      @fixed, @instance);' >> $tmp_file
   echo '$add_property("PINTYPE", "IN", [ '$x0_left', '$y_pintype' , "Symbol#1.0"], @string,
      @visible, @instance, @fixed, @graphic);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height', @right, @bottom, 0, @on, @visible, @fixed,
      @instance);' >> $tmp_file
   echo '$add_property("NET", "'$pin_name'", [ '$x0_prop', '$y_pin' , "Symbol#1.0"], void, @visible,
      void, @fixed, @graphic);' >> $tmp_file
   set y_counter = `expr $y_counter + 1 `
end
set y_counter = 1
foreach pin_name (`cat $in_path/$symbol_name.output`)
   set y_pin = `expr $y_counter \* $p_space`
   set y_pintype = `expr $y_pin - 1`
   echo '$add_polyline([['$x1_right', '$y_pin' , "Symbol#1.0"], ['$x1', '$y_pin', "Symbol#1.0"]]);'
      >> $tmp_file
   echo '$unselect_all(@comment, @pin, @property, @symbolbody, @text);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height', @right, @bottom, 0, @on, @visible, @fixed,
      @instance);' >> $tmp_file
   echo '$add_pin("'$pin_name'", ['$x1_right', '$y_pin' , "Symbol#1.0"], ['$x1_left', '$y_pin',
      "Symbol#1.0"]);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height_small' , @right, @bottom, 0, @on, @visible,
      @fixed, @instance);' >> $tmp_file
   echo '$add_property("PINTYPE", "OUT", ['$x1_right', '$y_pintype' , "Symbol#1.0"], @string,
      @visible, @instance, @fixed, @graphic);' >> $tmp_file
   echo '$setup_property_text("stroke", '$text_height', @left, @bottom, 0, @on, @visible, @fixed,
      @instance);' >> $tmp_file
   echo '$add_property("NET", "'$pin_name'", ['$x1_prop', '$y_pin' , "Symbol#1.0"], void, @visible,
      void, @fixed, @graphic);' >> $tmp_file
```

```
    set y_counter = `expr $y_counter + 1 `
end
foreach pin_name (`cat $in_path/$symbol_name.bidir`)
    set y_pin = `expr $y_counter \* $p_space`
    set y_pintype = `expr $y_pin - 1`
    echo '$add_polyline([['$x1_right', '$y_pin' , "Symbol#1.0"], ['$x1', '$y_pin', "Symbol#1.0"]]);'
        >> $tmp_file
    echo '$unselect_all(@comment, @pin, @property, @symbolbody, @text);' >> $tmp_file
    echo '$setup_property_text("stroke", '$text_height', @right, @bottom, 0, @on, @visible, @fixed,
        @instance);' >> $tmp_file
    echo '$add_pin("'$pin_name'", ['$x1_right', '$y_pin' , "Symbol#1.0"], ['$x1_left', '$y_pin',
        "Symbol#1.0"]);' >> $tmp_file
    echo '$setup_property_text("stroke", '$text_height_small' , @right, @bottom, 0, @on, @visible,
        @fixed, @instance);' >> $tmp_file
    echo '$add_property("PINTYPE", "IXO", ['$x1_right', '$y_pintype' , "Symbol#1.0"], @string,
        @visible, @instance, @fixed, @graphic);' >> $tmp_file
    echo '$setup_property_text("stroke", '$text_height', @left, @bottom, 0, @on, @visible, @fixed,
        @instance);' >> $tmp_file
    echo '$add_property("NET", "'$pin_name'", ['$x1_prop', '$y_pin' , "Symbol#1.0"], void, @visible,
        void, @fixed, @graphic);' >> $tmp_file
    set y_counter = `expr $y_counter + 1 `
end
echo '$check(void, @nokeep, @window, @transcript, @nofile, "da_check_file", void, void, void, void,
    void, void);' >> $tmp_file
echo '$set_active_window("Check#1");' >> $tmp_file
echo '$$close_window(@discard);' >> $tmp_file
echo '$set_active_window("Symbol#1.0");' >> $tmp_file
echo '$save_symbol(@update, @force, @defaultsym, @defaultifc, "'$symbol_name'", void);' >> $tmp_file
echo '$$close_window(@discard);' >> $tmp_file
echo '$set_active_window("session");' >> $tmp_file
echo '$$close_session(void);' >> $tmp_file
( da -nodisplay < $tmp_file > $std_out ) >& $std_err
rm $tmp_file
```

Beispiel eines Constrain Files

```
# This is a Constrain File created by the Database
# © 1998 Nikolai von Krusenstiern
# VCC PLACE INSTANCE_PAD: P240 ;
# I/O, GCK8 (A15) PLACE INSTANCE_PAD: P239 ;
# I/O (A14) PLACE INSTANCE_PAD: P238 ;
PLACE INSTANCE MEM_ADR_B_MEM(0)_PAD: P237 ;
PLACE INSTANCE MEM_ADR_B_MEM(1)_PAD: P236 ;
PLACE INSTANCE MEM_ADR_B_MEM(2)_PAD: P235 ;
PLACE INSTANCE MEM_ADR_B_MEM(3)_PAD: P234 ;
# I/O (A13) PLACE INSTANCE_PAD: P233 ;
# I/O (A12) PLACE INSTANCE_PAD: P232 ;
PLACE INSTANCE MEM_ADR_B_MEM(4)_PAD: P231 ;
...
```

Beispiel einer Datei für den Symbolgenerator

```
// This is a Symbol File created by the Database
// © 1998 Nikolai von Krusenstiern
// FPGA
pins { // pin specifications.
"LUT_DAT_B_LUT", bidir , width 8 , side 1 , position_on_side -1, no_bubble, no_edgesense;
"MEM_DAT_B_MEM", bidir , width 8 , side 1 , position_on_side -1, no_bubble, no_edgesense;
"B_CSY_CLK_CSY", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_CSY_INIT_CSY_BAR", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_CSY_SYNC_CSY", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_HBR_A_CLK_HBR_A", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_HBR_A_FE_HBR_A_BAR", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_HBR_A_FF_HBR_A_BAR", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_HBR_A_HF_HBR_A_BAR", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
"B_HBR_A_TX_EN_HBR_A_BAR", input , width 1 , side 3 , position_on_side -1, no_bubble, no_edgesense;
...
```

4.5 Technische Zeichnungen

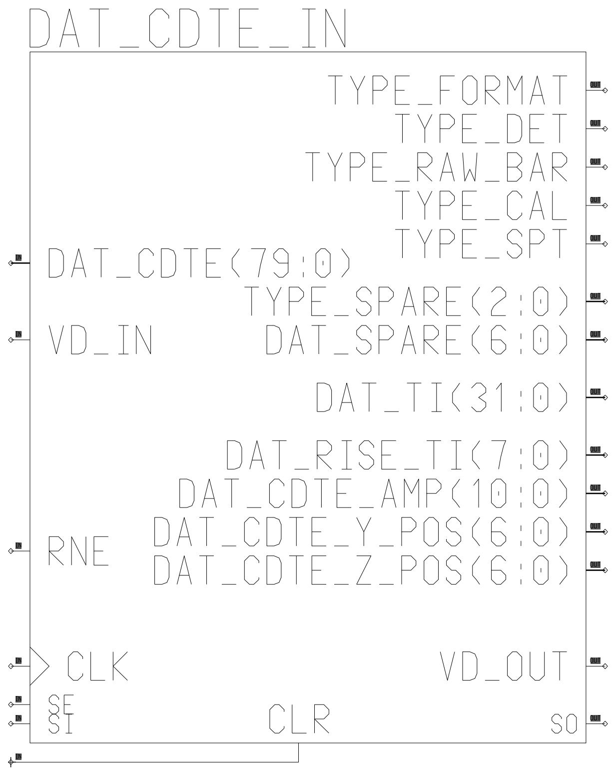

Abbildung 15: Die Eingabe-Schnittstelle für Cadmium-Tellurid-Ereignisse

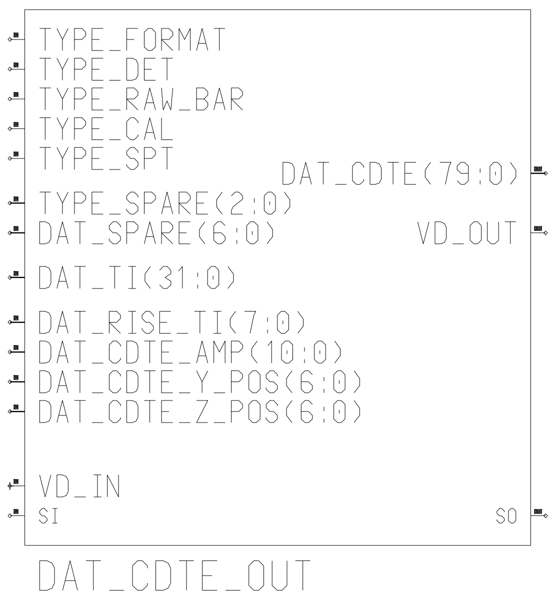

Abbildung 16: Die Ausgabe-Schnittstelle für Cadmium-Tellurid-Ereignisse

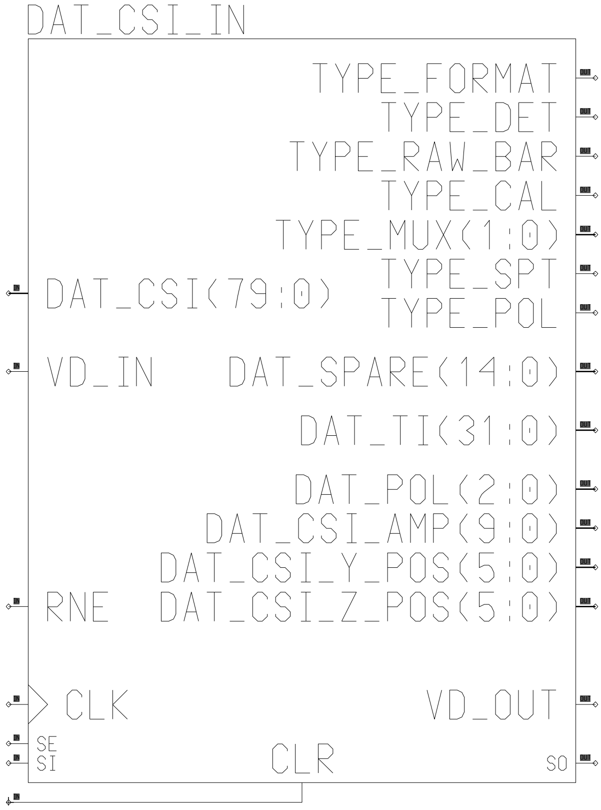

Abbildung 17: Die Eingabe-Schnittstelle für Cäsium-Jodid-Ereignisse

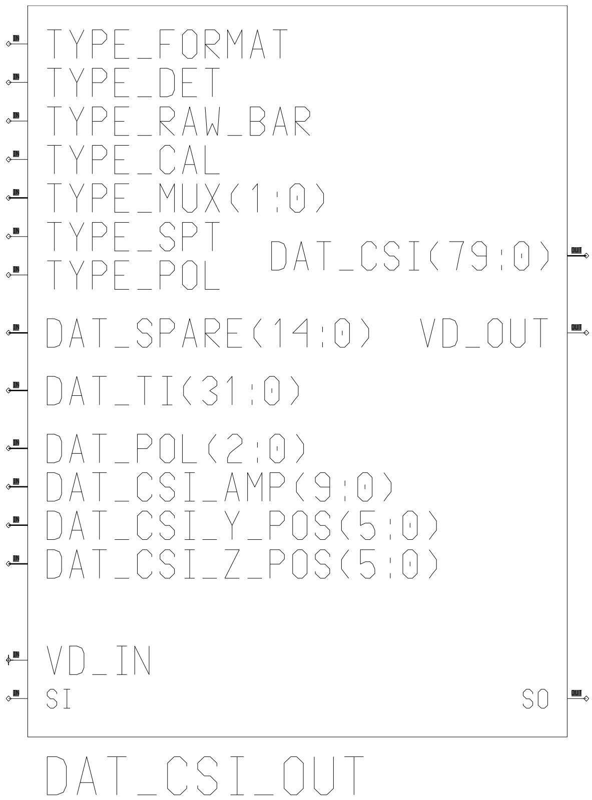

Abbildung 18: Die Ausgabe-Schnittstelle für Cäsium-Jodid-Ereignisse

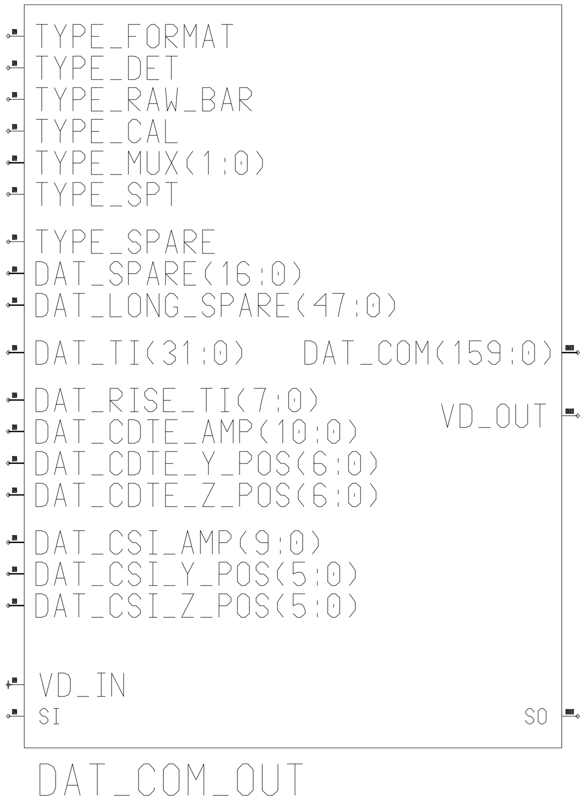

Abbildung 19: Die Ausgabe-Schnittstelle für Compton-Ereignisse

Abbildung 20: Die Ein- und Ausgabe-Schnittstelle für Housekeeping-Daten

Abbildung 21: Das Symbol der Schnittstelle zum Cadmium-Tellurid-Detektor

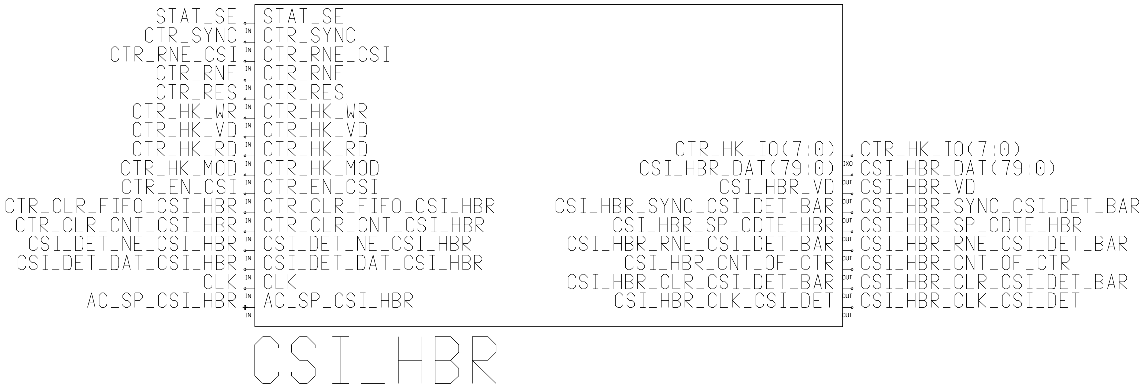

Abbildung 22: Das Symbol der Schnittstelle zum Cäsium-Jodid-Detektor

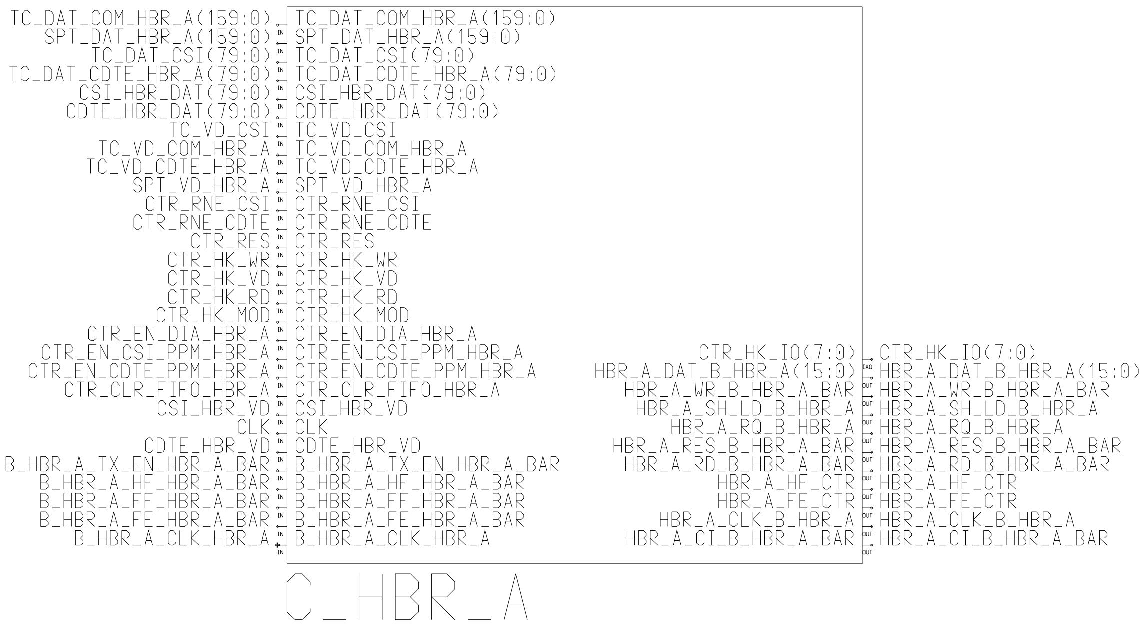

Abbildung 23: Das Symbol der Schnittstelle A zur DPE

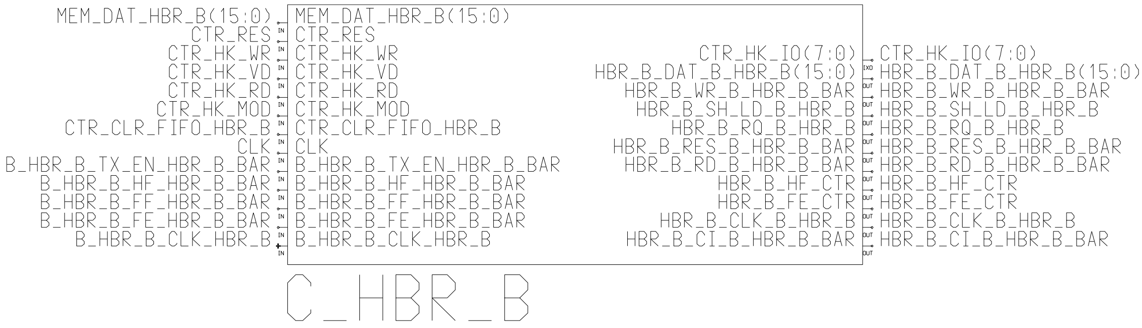

Abbildung 24: Das Symbol der Schnittstelle B zur DPE

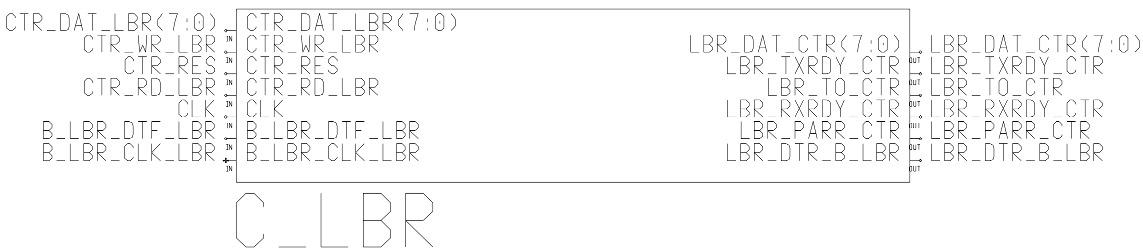

Abbildung 25: Das Symbol der Schnittstelle LBR zur DPE

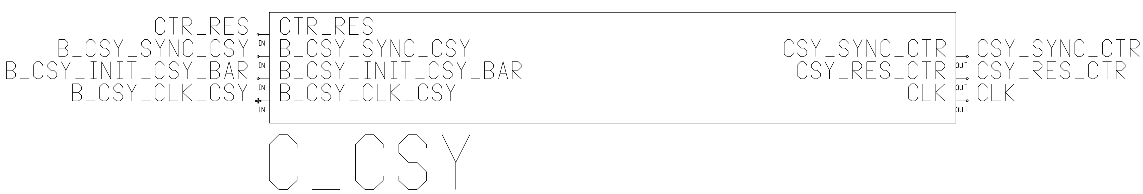

Abbildung 26: Das Symbol der Schnittstelle CSY zur DPE

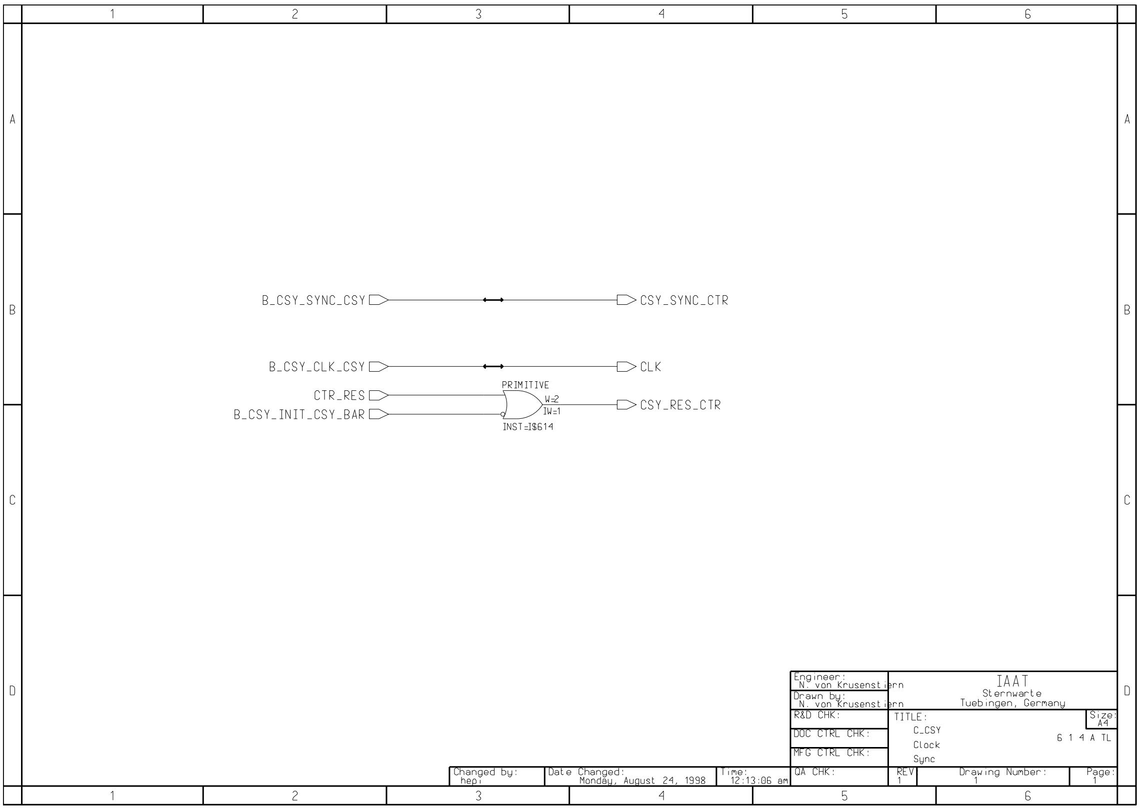

Abbildung 27: Das Schematic der Schnittstelle CSY zur DPE

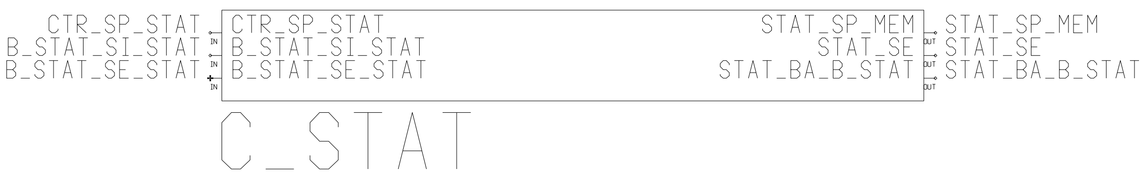

Abbildung 28: Das Symbol der Schnittstelle STAT zur DPE

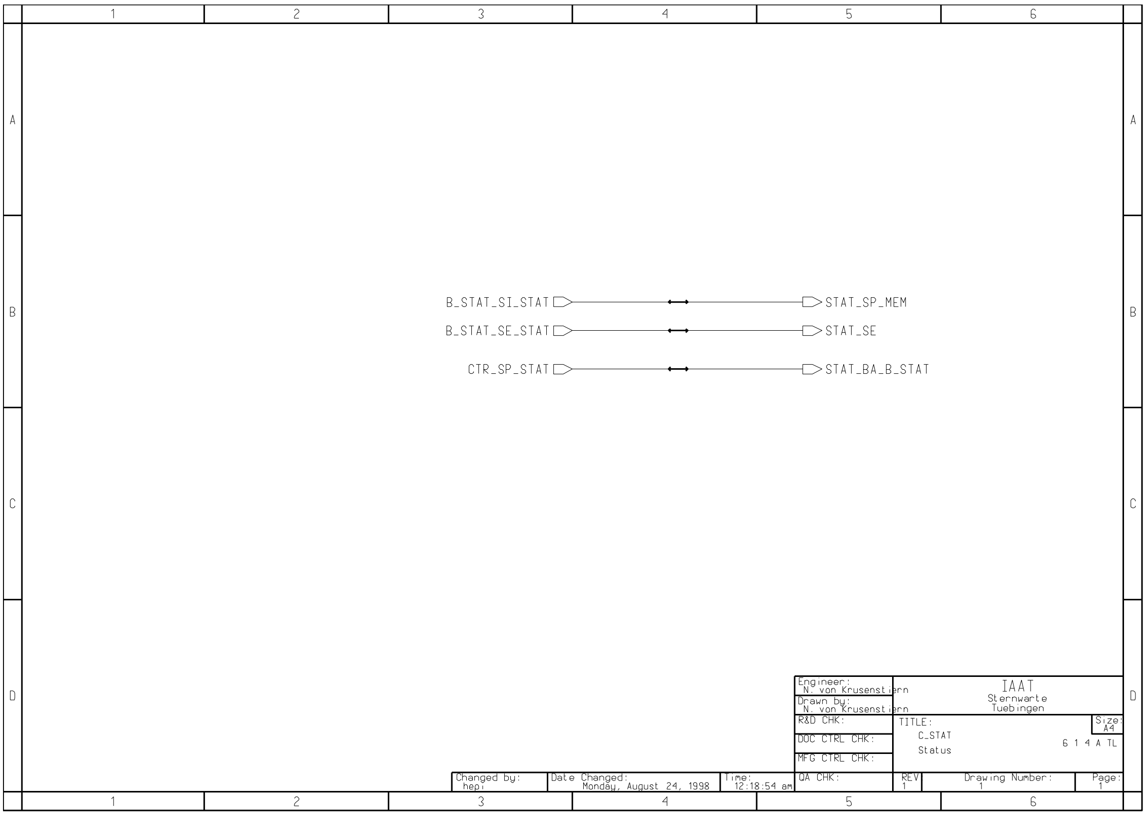

Abbildung 29: Das Schematic der Schnittstelle STAT zur DPE

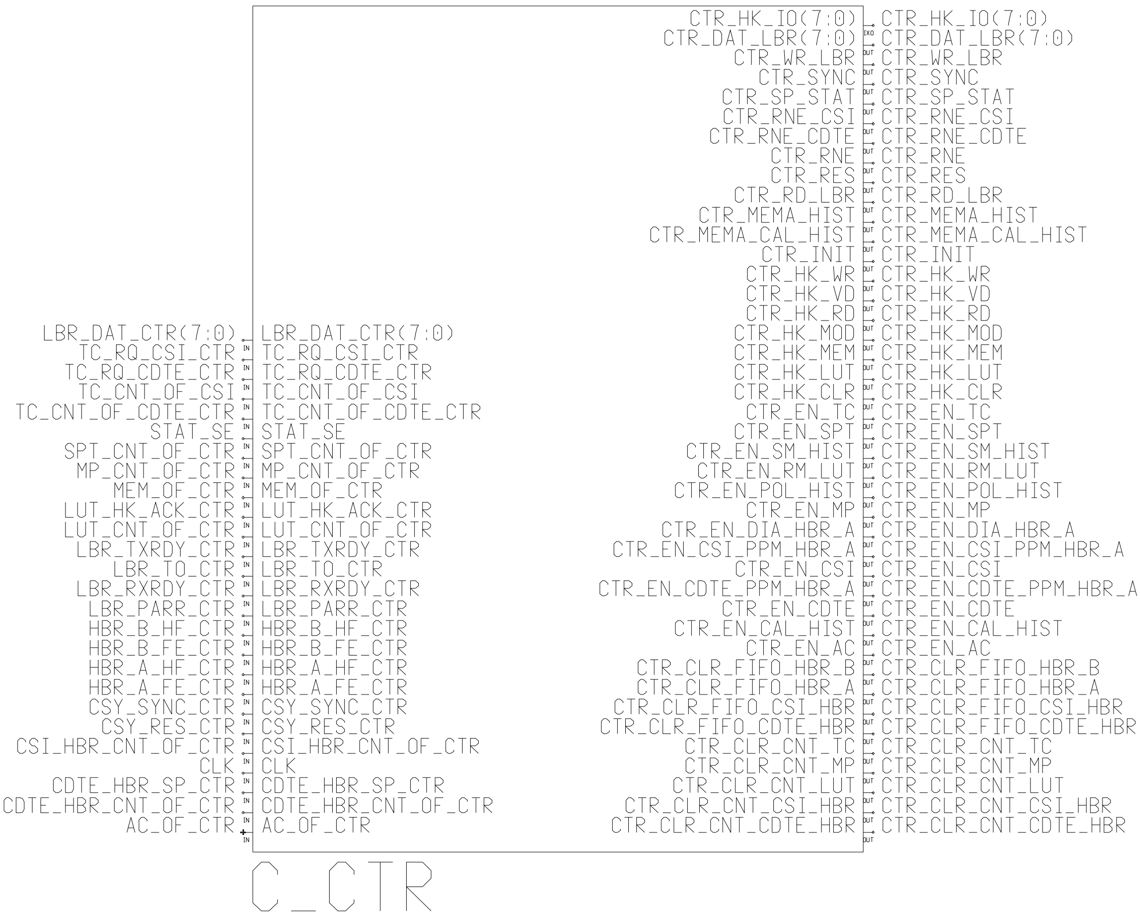

Abbildung 30: Das Symbol des Controllers

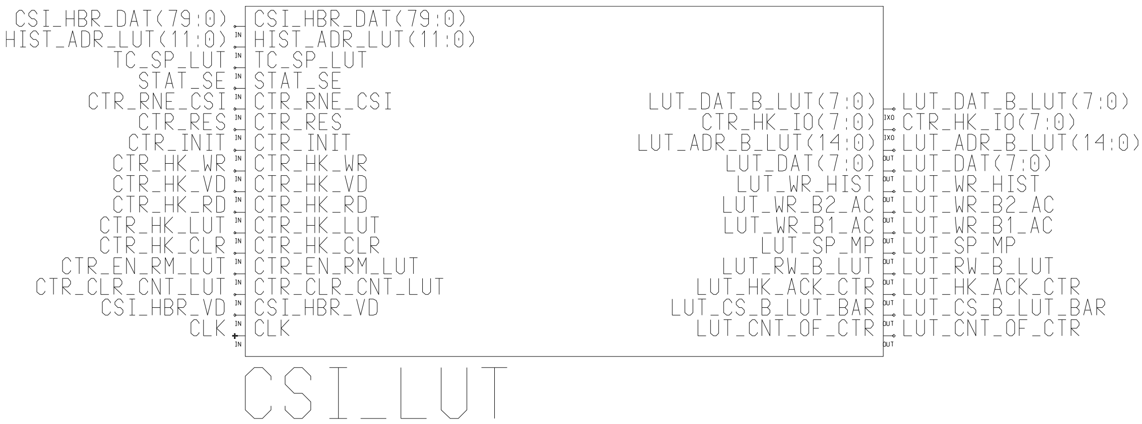

Abbildung 31: Das Symbol der Look Up Table

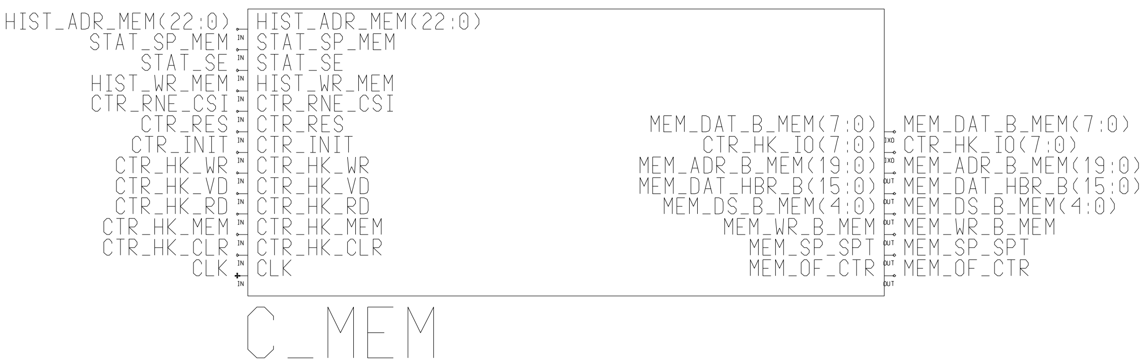

Abbildung 32: Das Symbol des Memory Controllers

Abbildung 33: Das Symbol der Amplituden-Korrektur

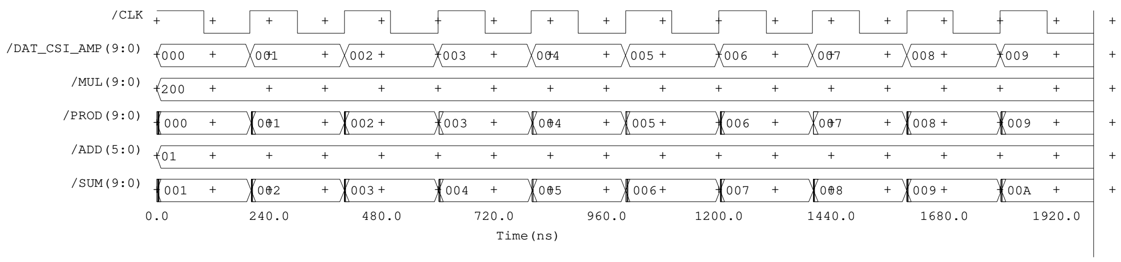

Abbildung 34: Die Simulation der Amplituden-Korrektur

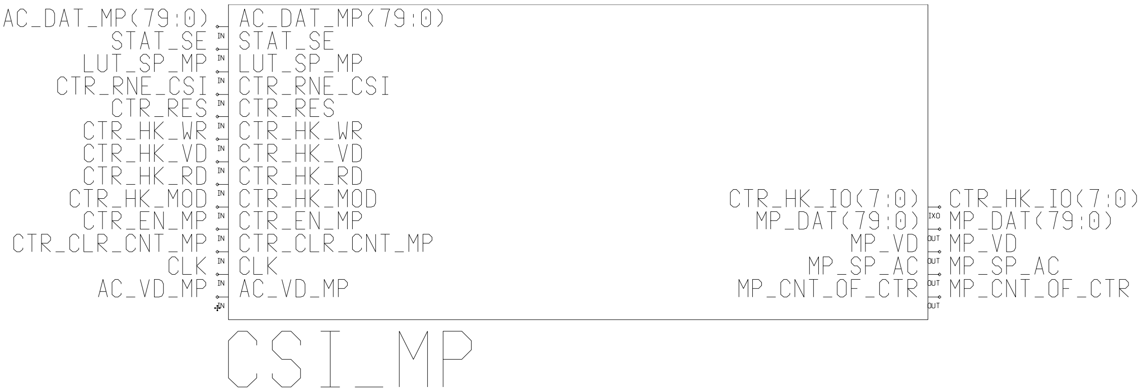

Abbildung 35: Das Symbol der Ereignisrekonstruktion

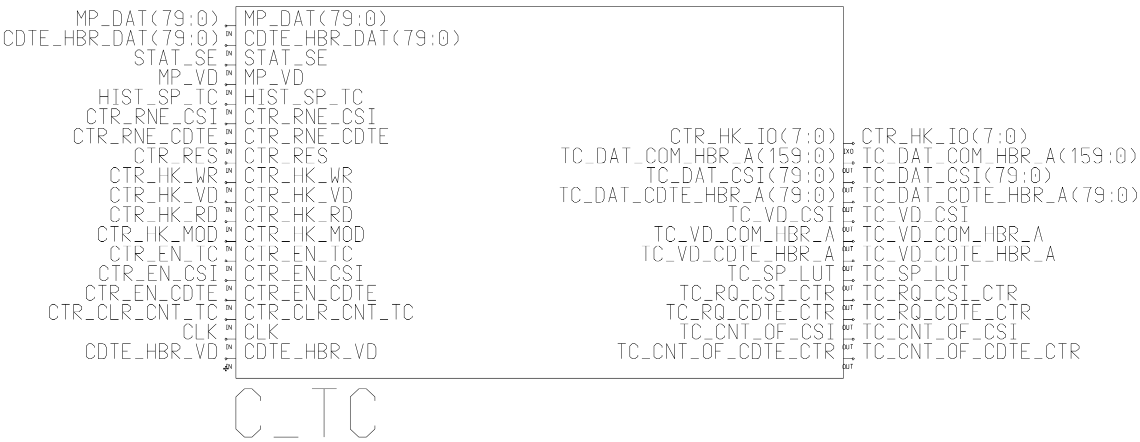

Abbildung 36: Das Symbol der Zeitkoinzidenzen

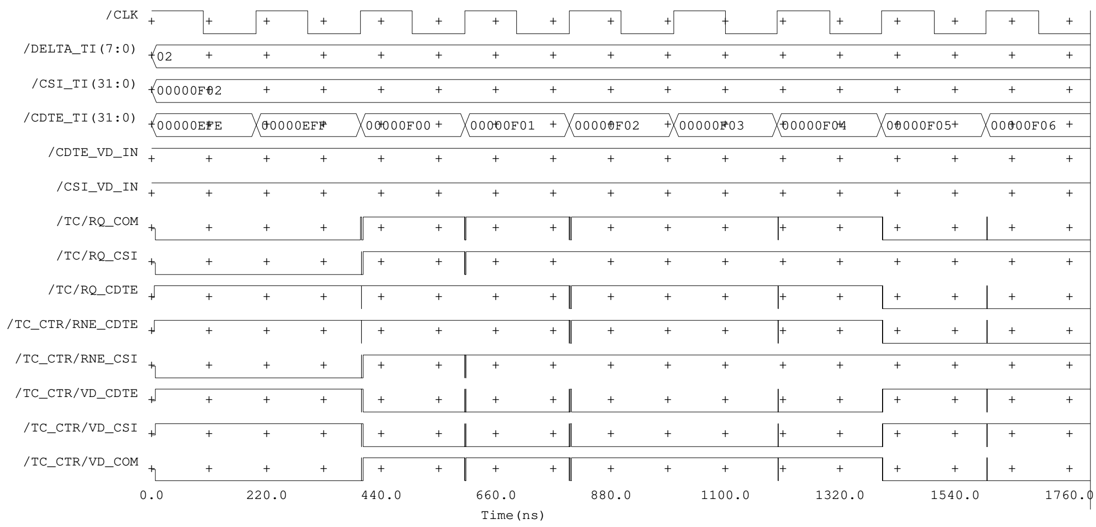

Abbildung 37: Die Simulation der Zeitkoinzidenzen

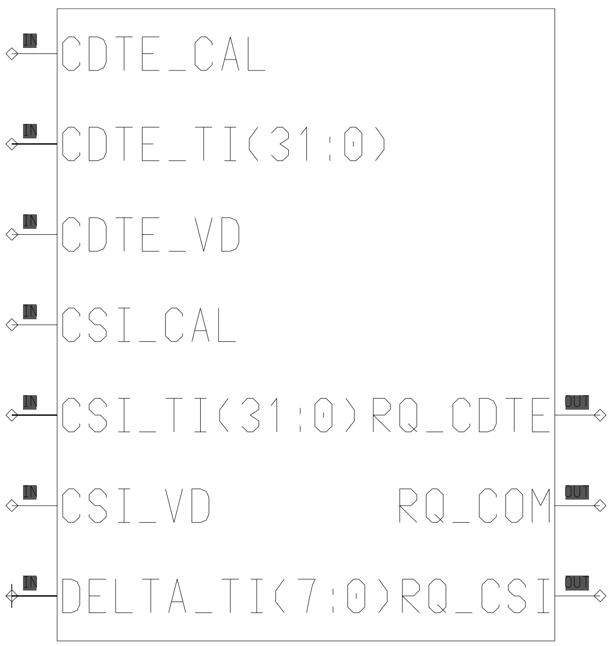

Abbildung 38: Das Symbol des Sub-Moduls Find_TC

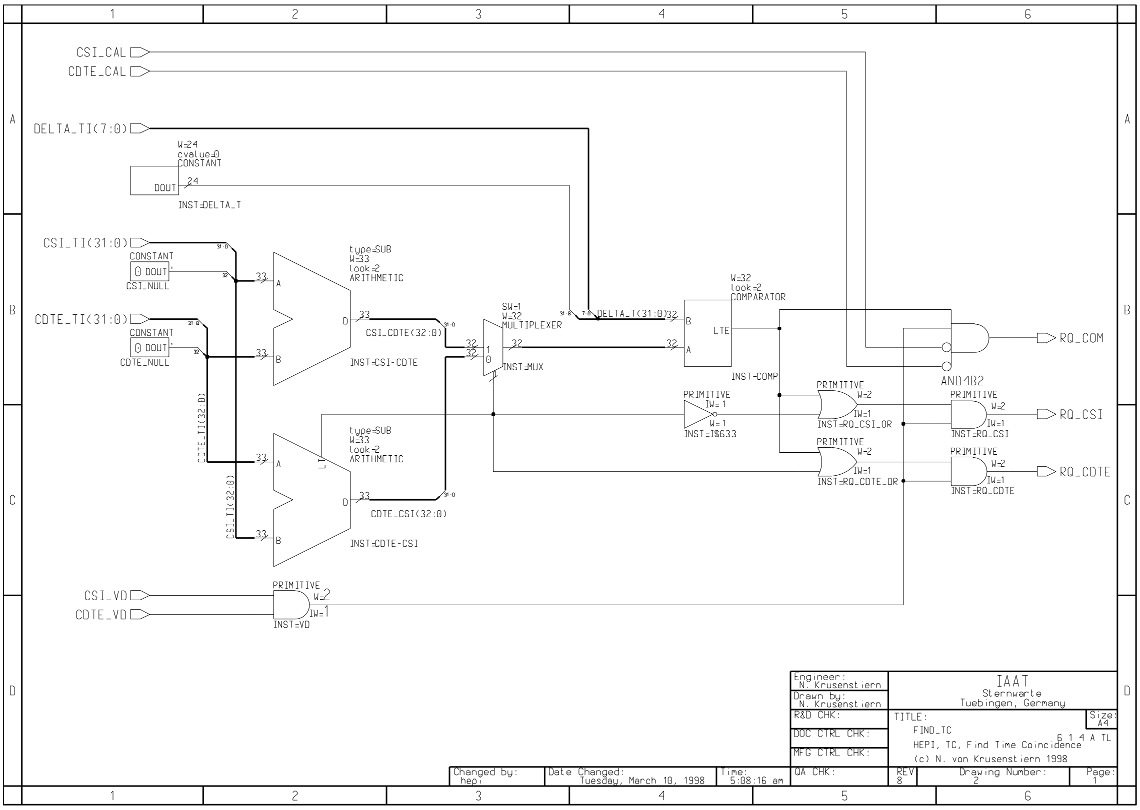

Abbildung 39: Das Schematic des Sub-Moduls Find_TC

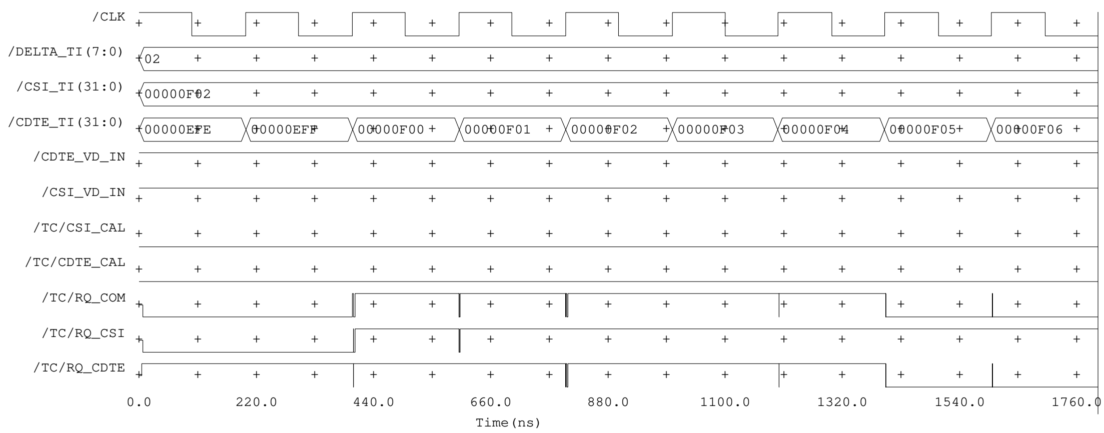

Abbildung 40: Die Simulation des Sub-Moduls FIND_TC

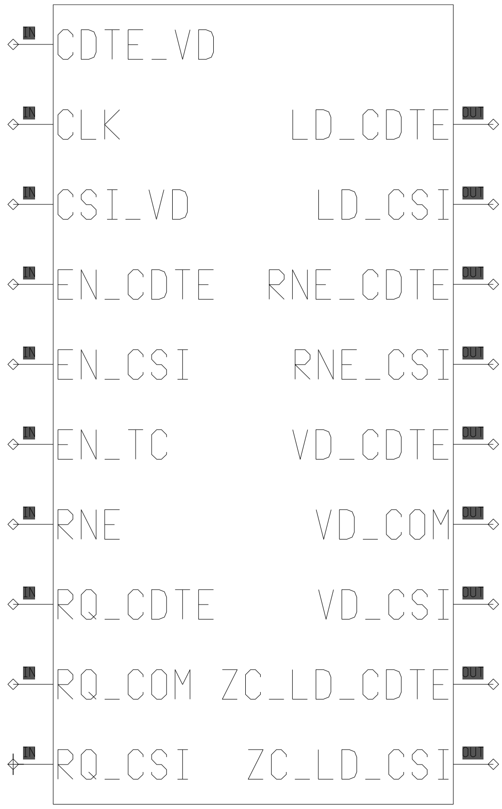

Abbildung 41: Das Symbol des Sub-Moduls TC_CTR

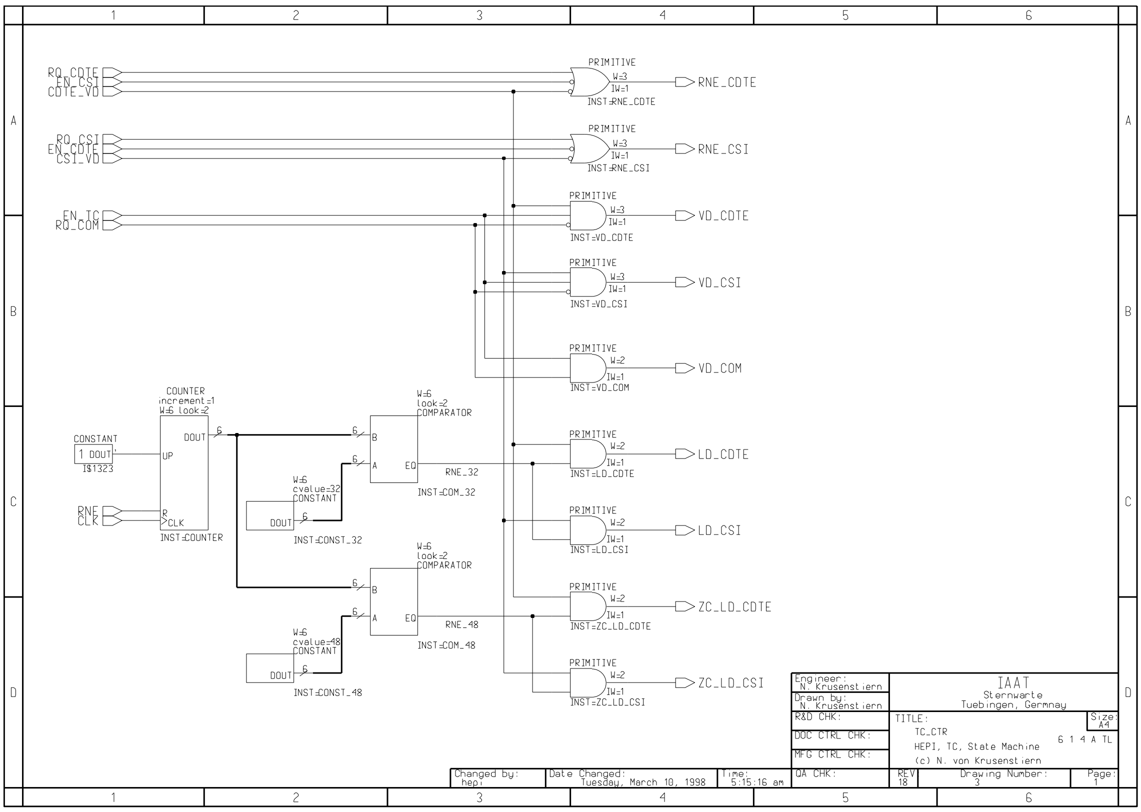

Abbildung 42: Das Schematic des Sub-Moduls TC_CTR

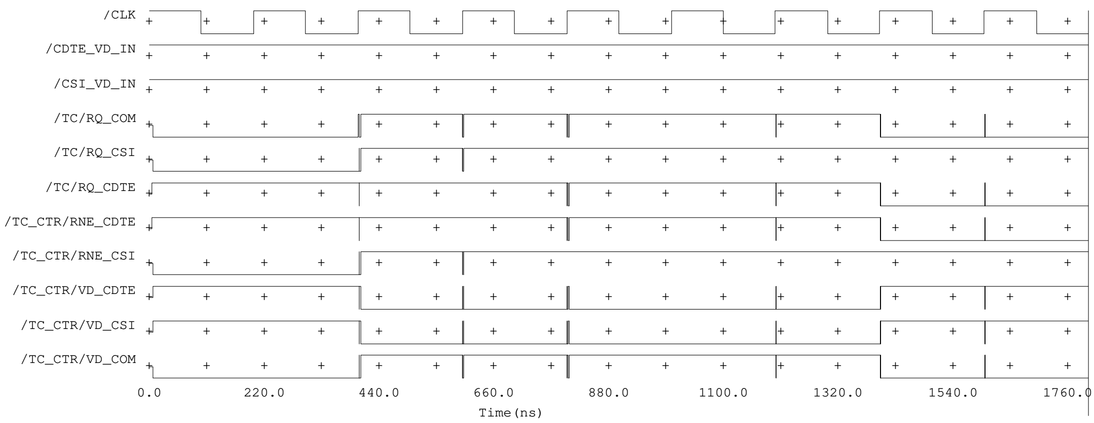

Abbildung 43: Die Simulation des Sub-Moduls TC_CTR

Abbildung 44: Das Symbol des Histogramm-Moduls

Abbildung 45: Das Symbol des Amplituden-Bandpasses

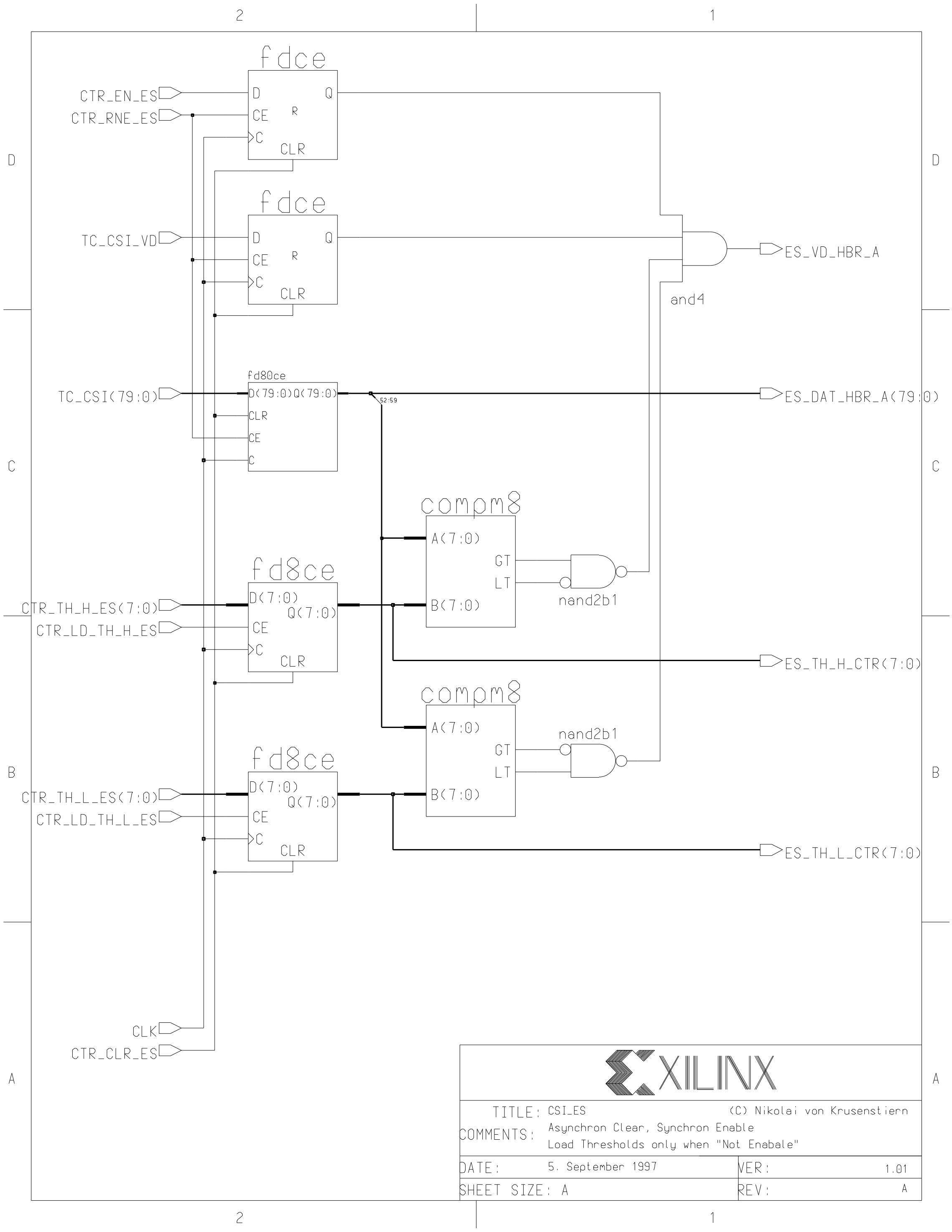

Abbildung 46: Das Schematic des Amplituden-Bandpasses

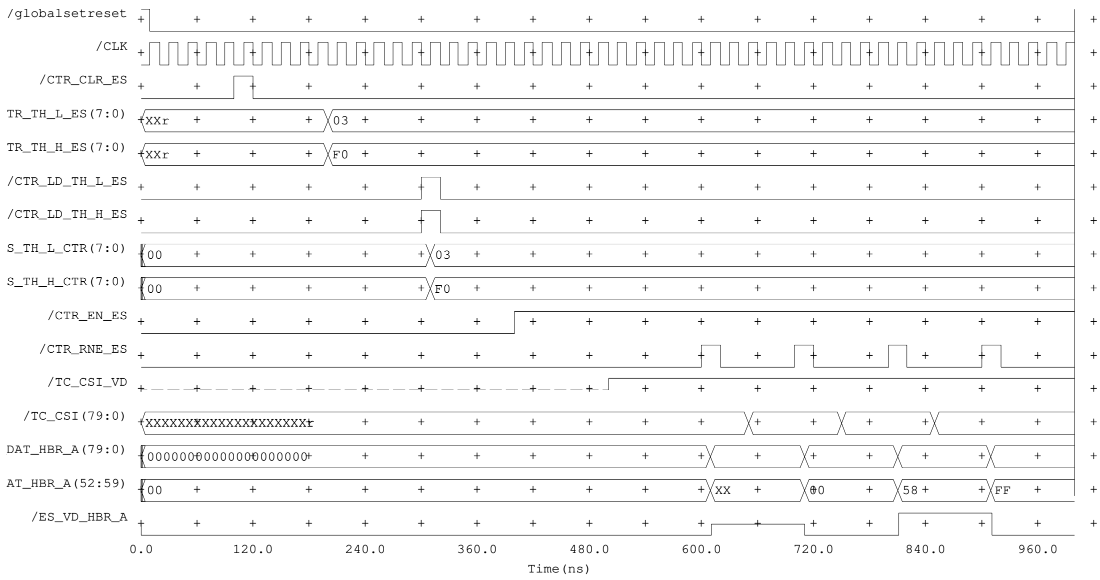

Abbildung 47: Die Simulation des Amplituden-Bandpasses

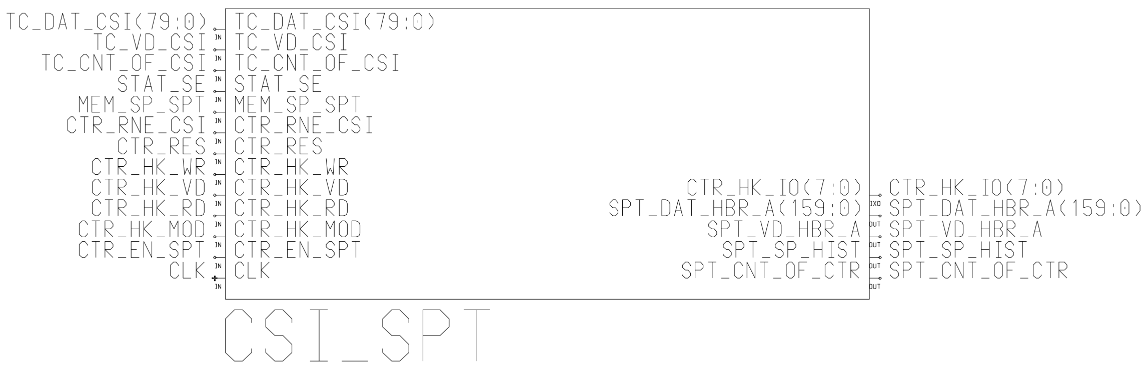

Abbildung 48: Das Symbol der zeitaufgelösten Spektroskopie

Abbildung 49: Das Schematic des Top Level

Hier Top Level einlegen.

Abbildung 50: Das Schematic der Amplituden-Korrektur

Hier AC einlegen.

Abbildung 51: Das Schematic der Zeitkoinzidenz

Hier TC einlegen.

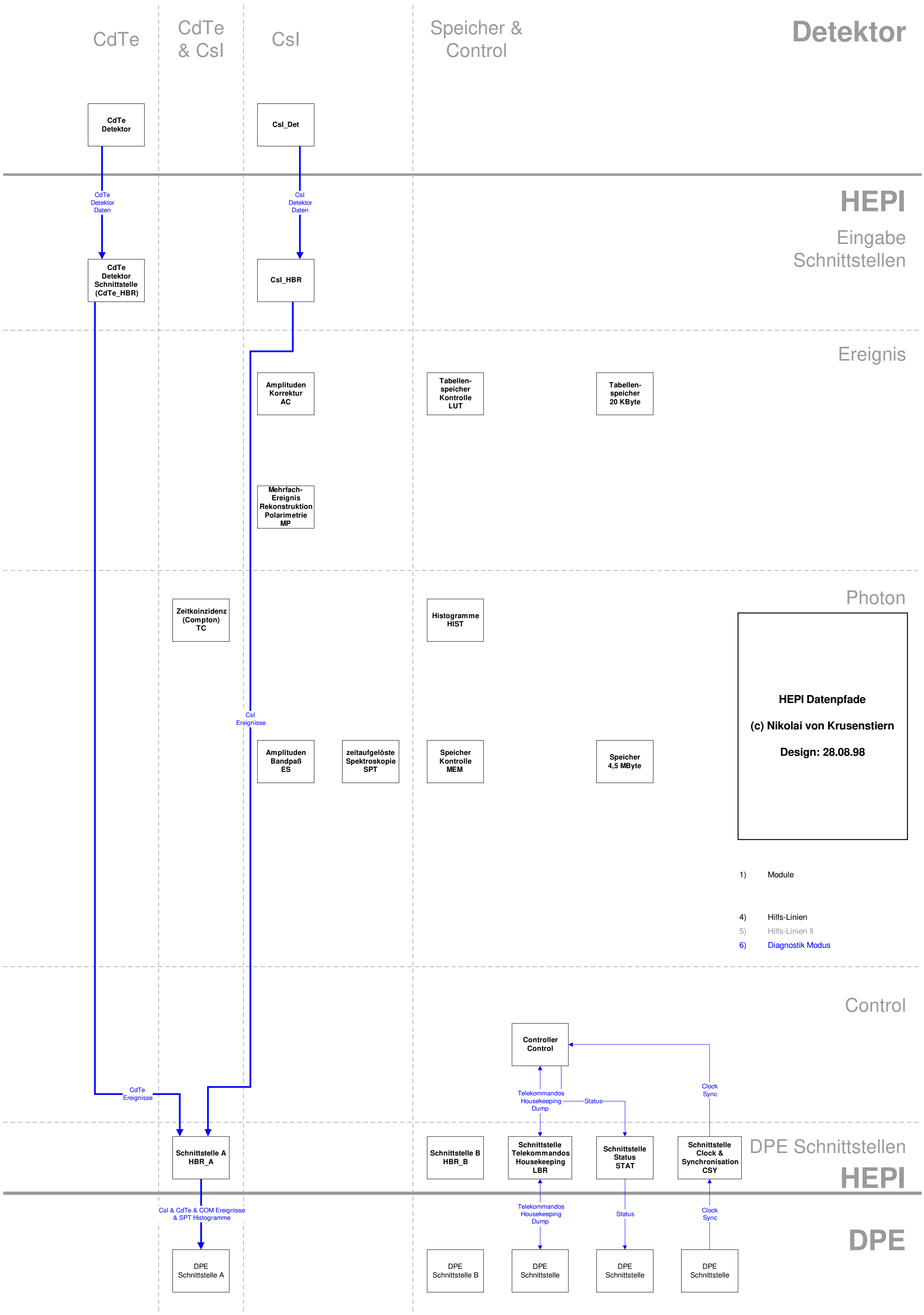

Abbildung 52: Der Diagnostik Modus

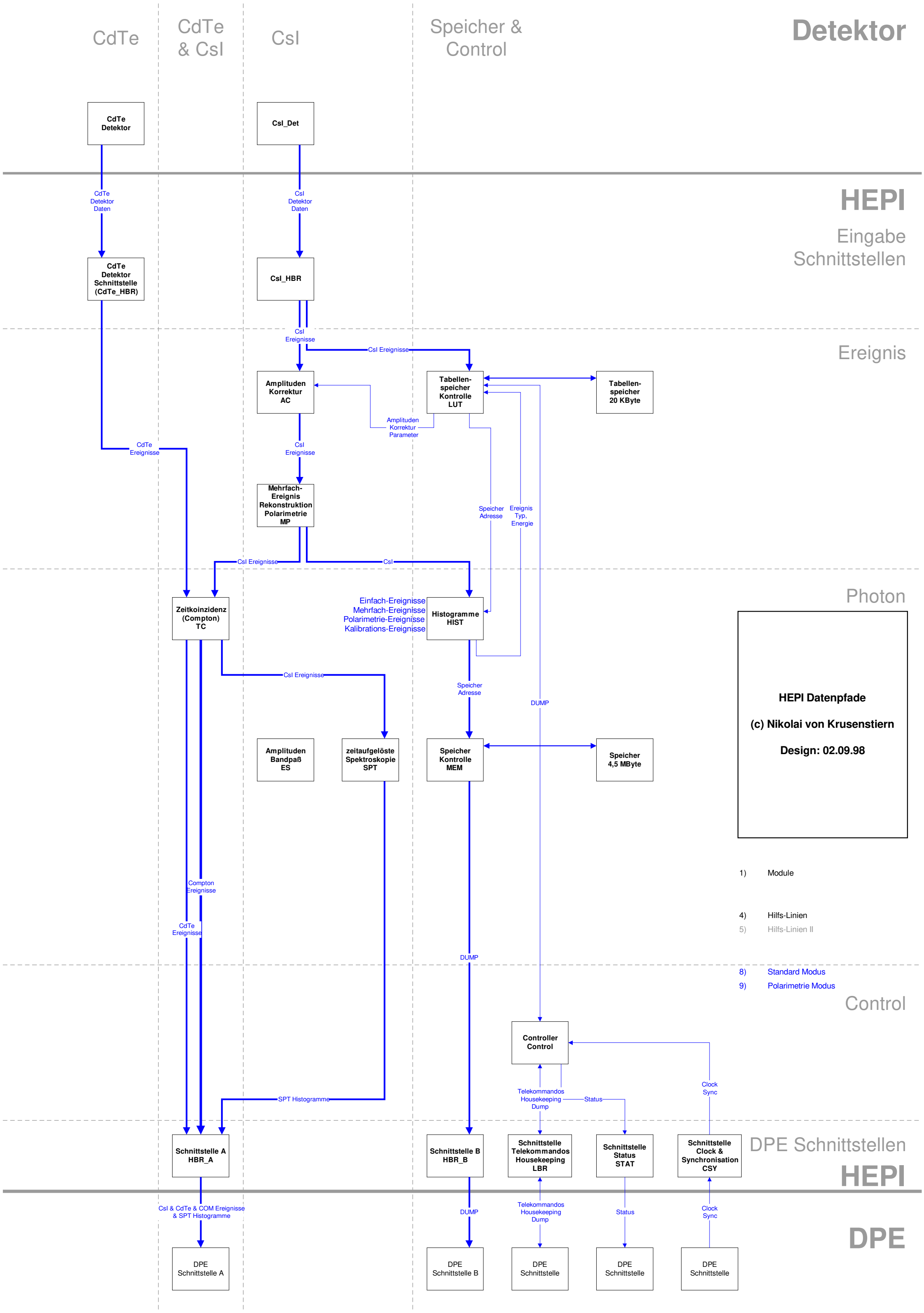

Abbildung 53: Der Standard Pointing und Standard Polarimetry Modus

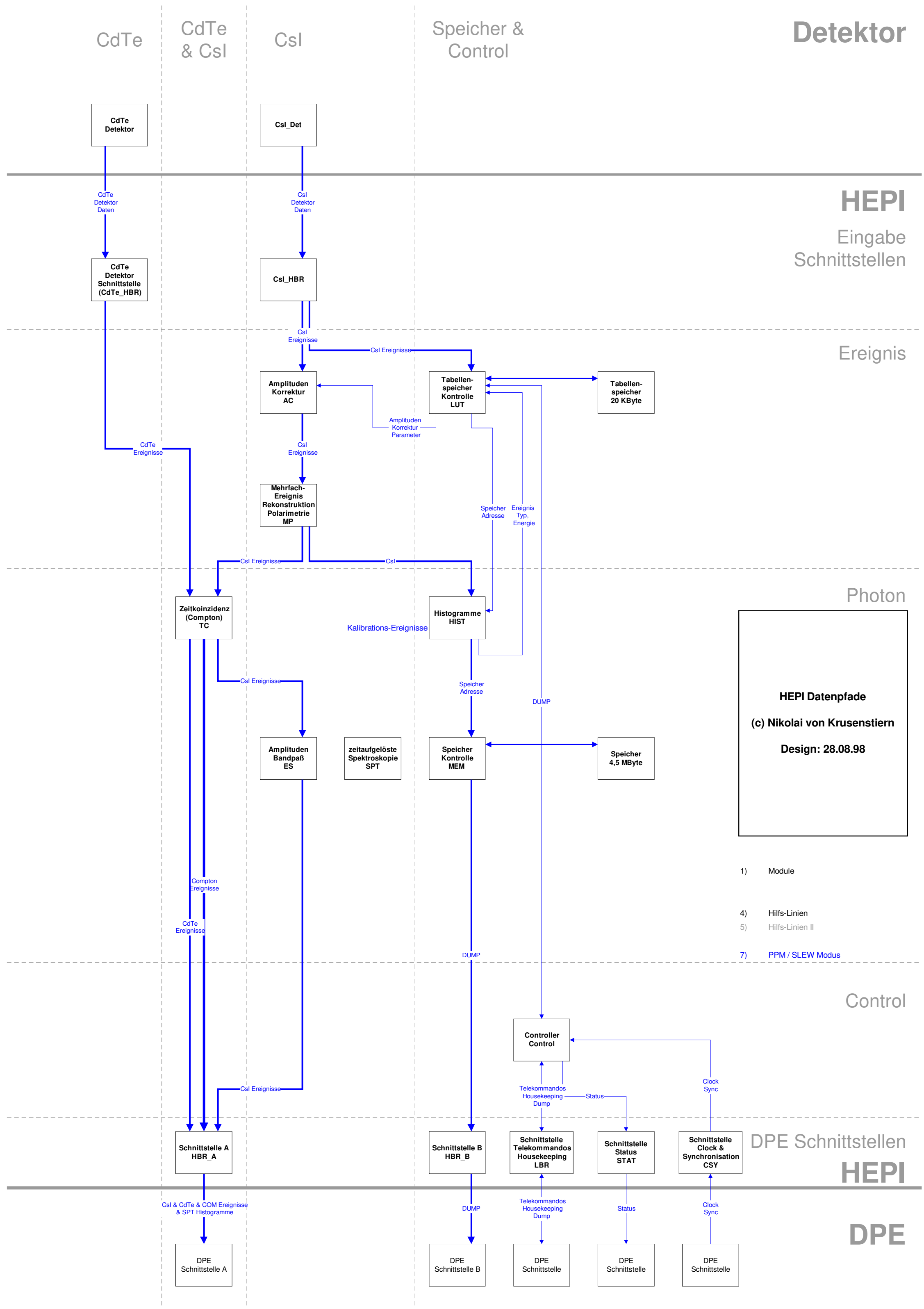

Abbildung 54: Der Photon by Photon, Pointing Slew und Polarimetry Slew Modus

4.6 Danksagung

Schon als kleiner Junge habe ich davon geträumt, daß ich einmal Astronaut werde. Nun, dieser Traum ist mit dieser Diplomarbeit ein Stückchen wahr geworden, denn die HEPI wird 2001 an Bord des Satelliten INTEGRAL in eine Umlaufbahn gebracht. Und wenn es mir möglich ist, werde ich beim Start zugegen sein.

Ich bin tief dankbar dafür, daß ich diesen, meinen Weg bis hier gehen konnte. Allen voran danke ich meinen Eltern, die mir nicht nur mein Studium ermöglicht haben. Auch diese Arbeit ist nicht ohne rat- und tatkräftigen Unterstützung entstanden, für die ich mich bei C. Keck, L. Roth, H. Baronesse von Schilling und R. Werner bedanken möchte.

Danken möchte ich auch meinen Kollegen und Kolleginnen am Institut, die immer eine offene Tür und ein offenes Ohr für mich hatten. „Pars pro toto“ seien hier. M. Gölz, A. Fromm, E. Kendziorra, H. Lenhart, A. Monice, B. Pflüger, F. Rother und J. Wilms genannt.

Ich betrachte es als besonderes Glück und einen Vorteil, daß diese Diplomarbeit Teil eines internationalen und ehrgeizigen Weltraumprojekts ist. Ich danke der Firma DD&T in Reutlingen und meinen Kollegen des IBIS Teams am IAAT für unsere erfolgreiche Zusammenarbeit. Die gelegentlichen Begegnungen mit Kollegen des INTEGRAL Teams aus ganz Europa werde ich als besondere *Highlights* in Erinnerung behalten. Ganz besonders danke ich Herrn Dr. R. Volkmer für seine Unterstützung.

Während meines Studiums habe ich mir immer eine Diplomarbeit gewünscht, die nicht nur interessant ist, sondern auch noch Spaß macht. Ich habe aber weit mehr erhalten. Ich danke deshalb Herrn Professor R. Staubert für die Vergabe dieses Themas und für seine Betreuung und Unterstützung.

Danke.