Hardware Ereignis Prozessor für den Imager auf dem

INTEGRAL-Satelliten

Diplomarbeit

vorgelegt von

Frank Lier

Institut für Astronomie und Astrophysik
der Universität Tübingen
Abt. Astronomie

November 1996

Inhaltsverzeichnis

Einleitung

Das ESA Satelliten Projekt INTEGRAL (INTErnational Gamma-Ray Astrophysics Laboratory), innerhalb dessen ich meine Diplomarbeit durchgeführt habe, ist eine Medium Sized Mission im Rahmen des ESA Programms „Horizon 2000“. Innerhalb dieses Programms gibt es vier Cornerstone Missions und mehrere kleinere Missionen. Zu den Cornerstone Missions zählt unter anderem XMM (X-Ray Multi Mirror).
Der INTEGRAL-Satellit ist ein Observatorium zur Untersuchung harter Röntgen- und Gammastrahlung, das gleichzeitig gute spektroskopische Auflösung ($E/\Delta E = 500$) und gute Abbildungseigenschaften (Winkelauflösung: 12 arcmin FWHM) besitzt.
Es befinden sich vier Instrumente an Bord. Die beiden Hauptinstrumente sind das Spektrometer SPI (Spectrometer for INTEGRAL) und der Imager IBIS (Imager on Board INTEGRAL Satellite), die den Energiebereich von 15 keV bis 10 MeV abdecken. Außerdem werden noch Himmelsüberwachungen im Röntgenbereich von 3 keV bis 35 keV und im optischen Bereich von 500 nm bis 850 nm durchgeführt. Hierfür werden der Röntgen-Monitor JEM-X (Joint European X-Ray Monitor) und die Optical Monitoring Camera OMC eingesetzt.
Die Röntgen- und Gamma-Astronomie ist noch ein verhältnismäßig junges Gebiet der Astronomie. Der Röntgenbereich reicht dabei etwa von 0.100 - 500 keV. Dies ist jedoch eine relativ willkürliche Festlegung, da keine scharfe Grenze zwischen dem ultravioletten und dem Röntgenbereich des Spektrums gezogen werden kann. Oberhalb von 500 keV spricht man von Gammastrahlen. Ein großes Problem bei dieser Art Astronomie besteht in der Absorption der Röntgen- und Gammastrahlen in der Erdatmosphäre. Die Strahlung gelangt nicht bis zum Erdboden, sondern wird schon lange vorher absorbiert. Die Beobachter sind also auf Raketen und Satelliten oder hochfliegende Ballone angewiesen.
Als nach dem Zweiten Weltkrieg geeignete Raketen zur Verfügung standen, konnte die Röntgenastronomie beginnen. Zuerst wurden mit auf einer Rakete montierten Geigerzählern, Röntgenstrahlen von der Sonnenkorona entdeckt. Im Jahre 1962 schließlich entdeckte man eine helle Röntgenquelle im Sternbild Skorpion und eine schwache Hintergrundstrahlung im Röntgenbereich. Die helle Quelle erhielt den Namen Scorpius X-1. Diesem Raketenflug folgten weitere, und schließlich wurde der erste Röntgensatellit UHURU gestartet. Mit ihm und nachfolgenden Satelliten wurden viele Röntgenquellen entdeckt. Ende der siebziger Jahre wurden drei große Satelliten gestartet. Der zweite dieser HEAO-Serie (High Energy Astrophysics Observatories) war ein Röntgenteleskop (Wolter-Teleskop). Damit konnte gegenüber den bisher verwendeten kollimierten Detektoren, eine deutlich bessere Bildauflösung von etwa zwei Bogensekunden erreicht werden. Dieser zweite Satellit wurde später in Einstein-Observatorium umbenannt. Ein weiterer erfolgreicher Röntgen-Satellit ist der deutsche ROSAT (ROentgen SATellite). Er wurde in Deutschland entwickelt und am 1. Juni 1990 gestartet. Die wissenschaftliche Nutzlast von ROSAT besteht aus zwei Teleskopen, dem unter Federführung des Max-Planck-Instituts für extraterrestrische Physik in Deutschland entwickelten Röntgenteleskop und der englischen Wide Field Camera. Für das Röntgenteleskop stehen zwei verschiedene Detektoren zur Verfügung, der PSPC und der HRI. Der PSPC (Position Sensitive Proportional Counter) ist ein ortsauflösender Proportionalzähler und der HRI (High Resolution Imager) ist ein Microkanalplatten-

Detektor. ROSAT ist noch immer in Betrieb, allerdings kann der PSPC nicht mehr verwendet werden, da kein Gas mehr vorhanden ist.
Für den harten Röntgen- und den Gammabereich lassen sich keine Wolter-Teleskope mehr einsetzen, da die Glanzwinkel für die Reflexion zu klein werden. Um eine Bildgebung zu erreichen, müssen andere Techniken angewendet werden.
Die beiden ersten Gamma-Satelliten SAS-2 und COS-B, die vor mehr als zwanzig Jahren gestartet wurden, waren mit Funkenkammer Detektoren ausgerüstet. Bei Funkenkammern kann man aus den Spuren der Elektron-Positron Paare auf die Einfallsrichtung der Gammaquanten schließen. Die Winkelauflösung dieser Instrumente war vergleichsweise gering, mit rund 2° betrug sie ungefähr das vierfache der scheinbaren Mondgröße. Am 5. April 1991 wurde das Compton Gamma Ray Observatory (CGRO) gestartet. Mit 17 Tonnen ist es die schwerste astrophysikalische Nutzlast, die je geflogen wurde. CGRO hat vier Instrumente an Bord, die im Energiebereich von 15 keV bis 30 GeV beobachten. BATSE (Burst and Transient Spectrometer Experiment) ist ein All Sky Monitor mit einem Energiebereich von 15-600 keV. COMPTEL (Compton Teleskop) kann einen 1 steradian großen Bereich des Himmels im Energiebereich von 0.8-30 MeV abbilden. EGRET (Energetic Gamma-Ray Teleskop Experiment) erfaßt einen Raumbereich von 1/2 steradian für Energien >50 MeV. Das vierte Instrument OSSE (Oriented Scintillation Spectrometer Experiment) schließlich beobachtet den Energiebereich von 0.05-10 MeV.
Ein großes Handikap bei der Beobachtung extraterrestrischer Gammastrahlung besteht in dem sehr ungünstigen Signal zu Rauschverhältnis. Die meisten Photonen, die registriert werden, stammen vom Hintergrund. Quellen heben sich nur geringfügig aus dem Hintergrund hervor und sind damit nicht einfach zu identifizieren.
Durch INTEGRAL soll nun der Energiebereich von 15 keV bis 10 MeV mit Hilfe von drei Instrumenten besser erforscht werden. Alle drei Instrumente werden ortsauflösende Detektoren besitzen, vor denen kodierte Masken als Bildfeldmodulatoren angebracht sind. Mit diesen sogenannten Coded Mask Teleskopen kann man zwar für einzelne Photonen deren Einfallsrichtung nicht bestimmen. Da die Maske aber je nach Position der Quelle unterschiedliche Intensitätsmuster auf dem Detektor erzeugt, kann durch mathematische Methoden die Position der Quellen im Gesichtsfeld bestimmt werden.
An einem dieser drei Instrumente, dem Imager, ist das IAAT (Institut für Astronomie und Astrophysik der Universität Tübingen) durch die Enwicklung von Hardware und Software für die digitale Prozessor Elektronik (DPE), welche die Verbindung zwischen den Detektoren des Imagers und dem Satellitenbus herstellt, beteiligt. Zur Sammlung und Vorverarbeitung der Detektordaten wird am IAAT eine Elektronikkarte für die DPE entwickelt. Meine Diplomarbeit stellt den Anfang dieser Hardware-Entwicklung dar.

KAPITEL 1

1 INTEGRAL

1.1 Wissenschaftliche Zielsetzungen

Mit INTEGRAL sollen Beobachtungen in der Röntgen- und Gamma-Astronomie durchgeführt werden. Zum ersten Mal wird man dabei im Energiebereich von 15 keV bis 10 MeV kosmische Quellen mit guter spektraler Auflösung verbunden mit guter Bildauflösung simultan beobachten können. Die Möglichkeit, Linienstrukturen im Energiespektrum der Quellen genauer zu studieren, wird mit Sicherheit viele neue Erkenntnisse über die Strahlungsmechansimen und die Struktur der Quellen ergeben. Die mit INTEGRAL beobachtbaren Quellen sind sehr vielfältig. Eine genauere Darstellung der Beobachtungsziele würde den Rahmen dieser Arbeit bei weitem sprengen, daher sollen hier nur die wichtigsten Quellen ohne weitere Diskussion der wissenschaftlichen Fragestellungen aufgelistet werden [1].

- Das galaktische Zentrum
- Kompakte Objekte, Neutronensterne und Schwarze Löcher
- Transiente Quellen in der Milchstraße
- interstellares Medium
- Novae und Supernovae
- nahe Galaxien
- Aktive Galaktische Kerne
- Galaxienhaufen
- diffuser kosmischer Hintergrund
- Gamma-Ray Bursts

1.2 Der Satellit

Der Start des Satelliten ist für das Jahr 2001 geplant. Die nominelle Lebensdauer beträgt zwei Jahre, wobei jedoch geplant ist den Betrieb auf fünf Jahre auszudehnen.
Es wird der gleiche in drei Achsen stabilisierte Bus verwendet, der auch bei XMM zum Einsatz kommt. Das Koordinatensystem des Satelliten ist ein rechtshändiges, orthogonales System. Die X-Richtung liegt parallel zur optischen Achse. Die Z-Achse zeigt in Richtung der Sonne und die Y-Achse vervollständigt das Koordinatensystem (siehe Abbildung 1.2). Die absolute Ausrichtgenauigkeit für die Y,Z-Achsen beträgt <=5’ und für die X-Achse <=15’. Der Bus enthält die notwendigen Subsysteme für den Satelliten. Lediglich der Aufbau mit den wissenschaflichen Instrumenten wird modifiziert.

Dieses Vorgehen wurde aus finanziellen Gründen gewählt. Man kann dadurch auf bereits entwickelte Teile zurückgreifen und spart somit Kosten. Die Abmessungen des Satelliten betragen 4x4x6 m^3. Diese Abmessungen beziehen sich auf den Satelliten ohne ausgeklappte Solargeneratoren. Seine Startmasse beläuft sich auf 3600 kg, wobei 2019 kg auf die wissenschaftliche Nutzlast entfallen.
Zur Energieversorgung dienen zwei Solargeneratoren. Sie liefern eine Leistung von 520 W. Für die Telemetrie der wissenschaftlichen Daten stehen 67 kbit/s zur Verfügung.

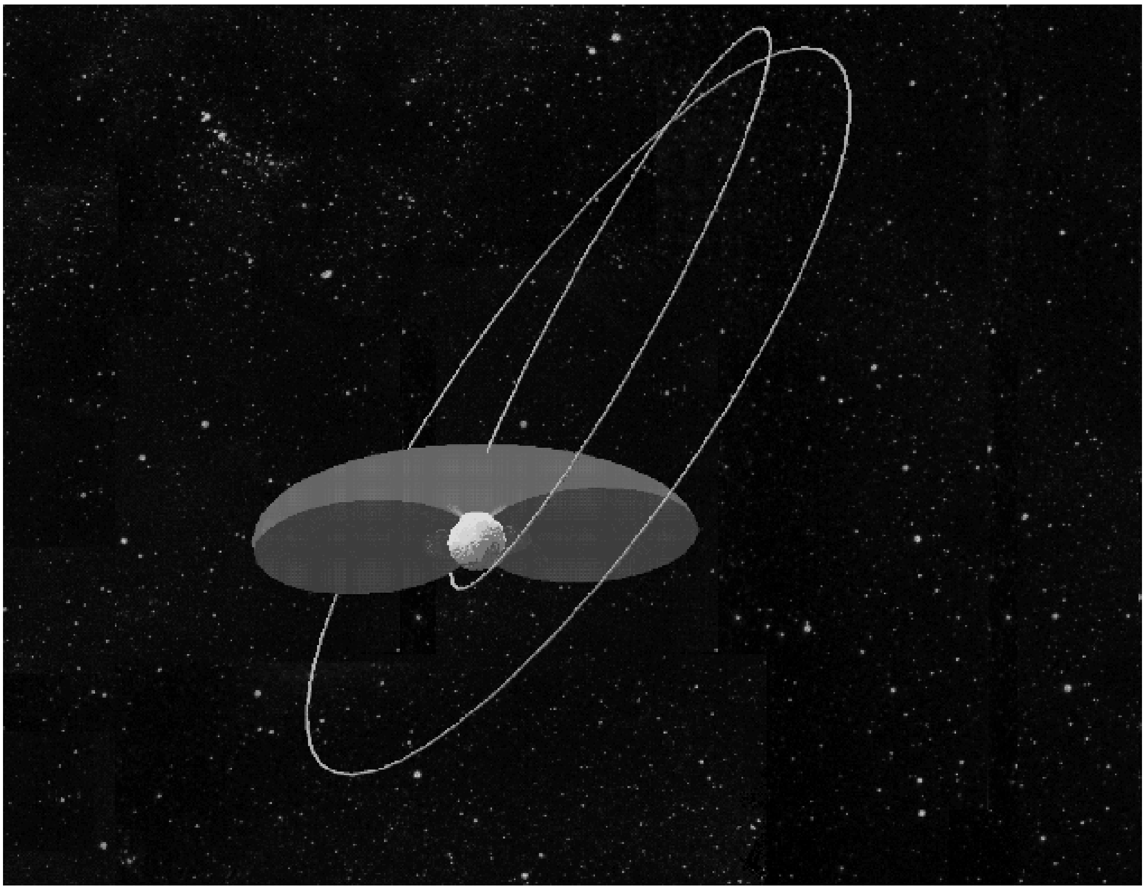


Abbildung 1.1: Die beiden möglichen Orbits von Integral. Die Strahlungsgürtel sind schematisch dargestellt.

INTEGRAL soll in einen hochexzentrischen Orbit gebracht werden. Der bevorzugte Träger ist die russische Proton Rakete. Mit ihr könnte ein 72 h Orbit mit einer Inklination von 51.6° erreicht werden. Das Perigäum beträgt 48000 km und das Apogäum 115000 km. Damit hält sich das Observatorium immer außerhalb von 40000 km Erdentfernung auf. Innerhalb dieses Bereich liegen die VanAllen Gürtel, in denen relativ hohe Strahlendosen zu erwarten sind. Die Hintergrund-Strahlung bleibt in dieser Umlaufbahn lange Zeit konstant und die Stahlenbelastung wäre deutlich geringer als im Orbit, der mit dem Ersatzlauncher Ariane 5 zu erreichen ist. Die Umlaufdauer des Satelliten im Orbit, der mit der Ariane 5 zu erreichen ist, beträgt 48 h mit einer Inklination von 60°. Der erdfernste Punkt liegt bei 117000 km und die größte Erdnähe wird mit 4000 km erreicht. Bei dieser Bahn durchfliegt der Satellit die Strahlungsgürtel, wodurch die Strahlenbelastung deutlich ansteigt und nur strahlungsfeste elektronische Bauteile Verwendung finden können.
Die vorgesehenen Bodenstationen liegen in Villafranca, Goldstone und/oder Canberra.

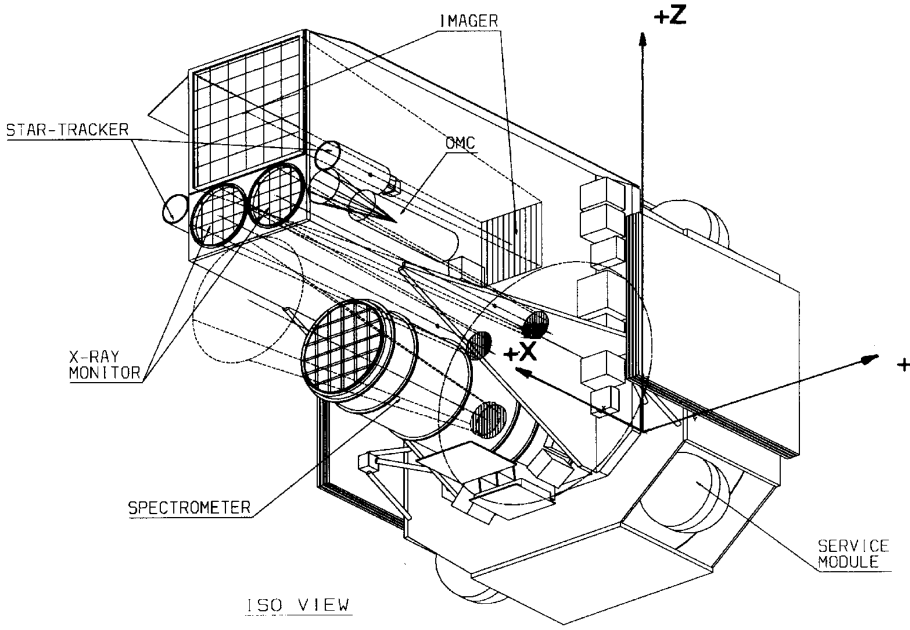


Abbildung 1.2: Der INTEGRAL-Satellit, hier mit zusammengefalteten Solargeneratoren dargestellt

1.3 Die Instrumente

An Bord von INTEGRAL gibt es vier wissenschaftliche Instrumente. Ich möchte zuerst kurz die drei Instrumente vorstellen, an denen das IAAT nicht beteiligt ist. Dies sind das Spektrometer, der Röntgen-Monitor und die Optical Monitoring Camera. Danach gehe ich detaillierter auf den Imager und seine Elektronik ein.

1.3.1 Das Spektrometer SPI

Das Spektrometer ist ein ortsauflösender Germaniumdetektor, vor dem eine kodierte Maske als Bildfeldmodulator angebracht ist (Coded Mask Teleskop). Es besteht aus einem aus einzelnen Pixeln aufgebauten Detektor und einer darüber angebrachten Maske. Diese Maske besteht ebenfalls aus einzelnen Elementen, wobei einige dieser Elemente für die Strahlung durchlässig sind und andere nicht. Die Elemente sind in einem bestimmten Muster angeordnet. Aus dem Schattenwurf, den eine Quelle durch die Maske auf den Detektor wirft, kann die Position der Quelle zurückgerechnet werden. Der Detektor, der in diesem Teleskop zum Einsatz kommt, besteht aus 19 hexagonal geformten Germaniumdetektoren. Sie werden mit einem aktiven Kühlsystem auf eine Betriebstemperatur von 85 K gekühlt. In einem Abstand von 1.7 Metern zur Detektorebene befindet sich die kodierte Maske, die ebenfalls aus hexagonalen Elementen besteht. Um den Hintergrund zu minimieren, befindet sich um den Detektor eine aktive Antikoinzidenz-Abschirmung (Veto Shield). Sie reicht vom Detektor bis fast hinauf zur Maske. Die Apertur wird mit dieser Abschirmung auf 30° begrenzt. Der Energiebereich des Spektrometers reicht von 20 keV bis hin zu 8 MeV. Die spektrale Auflösung beträgt dabei 2 keV (FWHM) bei einer Gamma-Energie von 1 MeV. Das voll kodierte Gesichtsfeld ist 16° groß und die Winkelauflösung beträgt 2°.

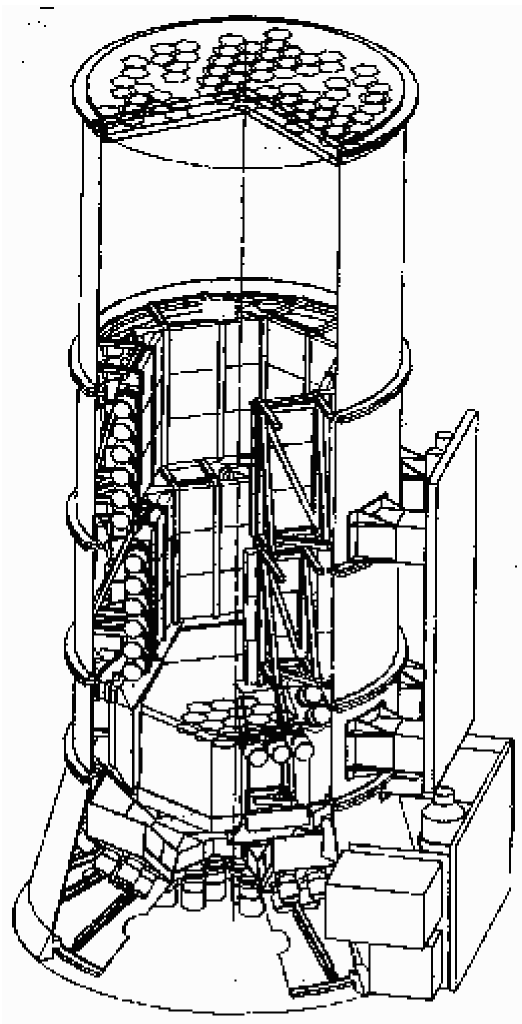


Abbildung 1.3: Das Spektrometer

1.3.1.1 Der Dithering-Modus

Um systematische Fehler bei der Beobachtung mit dem Spektrometer auszugleichen, führt der Satellit ein Dithering-Manöver durch. Dabei wird das Gesichtsfeld regelmäßig um einen bestimmten Winkel (vorgesehen sind etwa 2°) verschoben. Da die einzelnen Instrumente nicht gegeneinander beweglich sind, führen sie diese Bewegung ebenfalls aus. Es sind zwei verschiedene Pointierungs Muster (Pointing Patterns) geplant. Das erste Muster wird für die Beobachtung von Einzelquellen verwendet. Es besteht aus 6 hexagonal angeordneten Pointierungen um die Quelle herum. Das zweite ist ein quadratisches Pattern mit 24 Pointierungen. Es wird für die Beobachtung von mehreren Quellen im Gesichtsfeld des Spektrometers verwendet.
Die Dauer einer Pointierung beträgt 1000 Sekunden. Für die Umorientierung auf eine neue Blickrichtung braucht der Satellit etwa 260 Sekunden. Dieses Dithering-Manöver

kann abgeschaltet werden, um eine lange andauernde Beobachtung mit allen vier Instrumenten zu ermöglichen. Dies ist notwendig, da die anderen Instrumente von diesem Dithering-Manöver gestört werden.

1.3.2 Der Röntgen-Monitor JEM-X

Die Abkürzung JEM-X steht für Joint European X-Ray Monitor. Er ist eines der Monitor-Instrumente von INTEGRAL. Der Röntgen-Monitor spielt eine wichtige Rolle für die Entdeckung und Identifikation einer Gamma-Quelle. Der JEM-X beobachtet gleichzeitig mit den beiden Hauptinstrumenten im Energiebereich von 3-35 keV. Die Ortsauflösung liegt bei 3 Bogenminuten (FWHM). Der JEM-X ist ebenfalls ein Coded Mask Teleskop. Das voll kodierte Gesichtsfeld besitzt eine Größe von 4.8°. Der Detektor besteht aus zwei Einheiten, wobei jede dieser Einheiten aus einem mit Xenon gefülltem ortsauflösendem Proportionalzähler besteht. Das Xenon in den Detektoreinheiten steht unter einem Druck von 5 bar. Für jeden Detektor gibt es eine eigene kodierte Maske, die 3.2 Meter über der Detektorebene liegt.

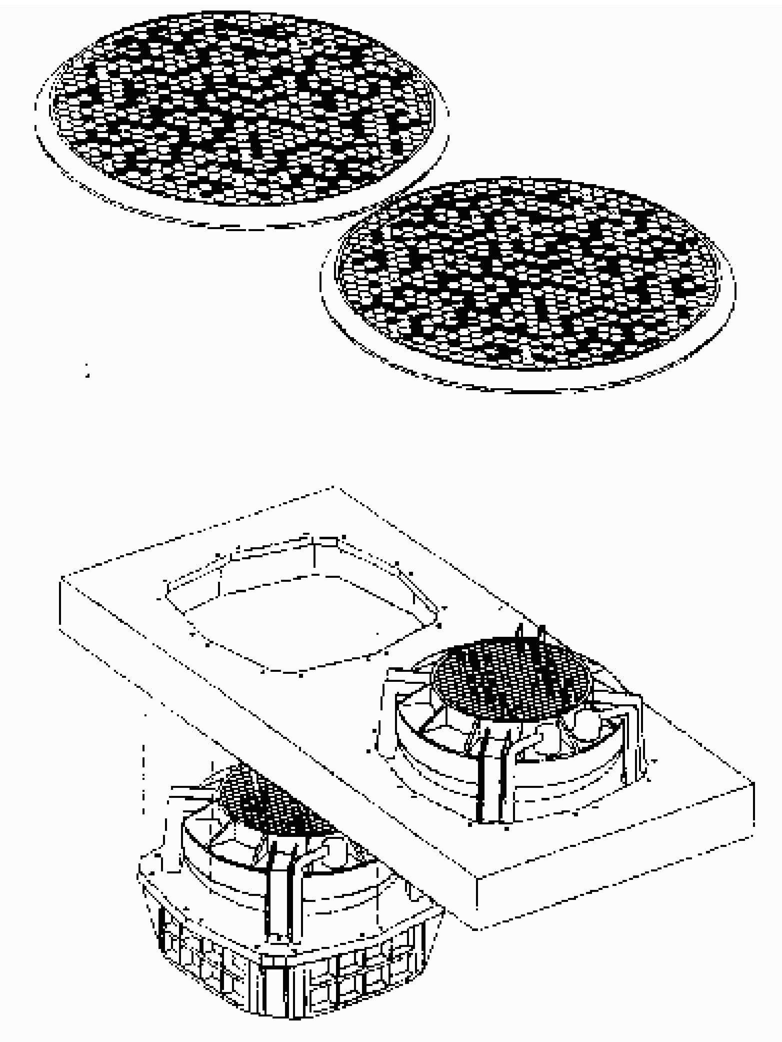


Abbildung 1.4: Der JEM-X

1.3.3 Die Optical Monitoring Camera

Mit Hilfe der OMC (Optical Monitoring Camera) soll die optische Komponente der beobachteten Röntgen- und Gamma-Quellen gemessen werden. Damit ergibt sich mit INTEGRAL zum ersten Mal die Möglichkeit, mit einem Satelliten über sieben Größenordnungen in der Photonenenergie hinweg, gleichzeitig zu beobachten. Die Optical Monitoring Camera besteht aus einer Optik und einem passiv gekühlten CCD, der im Frame Transfer Mode arbeitet. Der CCD hat eine Größe von 2048 x 1024 Pixeln (1024 x 1024 Imaging Area) und befindet sich in der Fokal-Ebene einer Linsenoptik mit 50 mm Öffnung, in die ein Johnson V Filter integriert ist. Der Transmissionsbereich des Filters liegt bei 500-850 nm. Das Gesichtsfeld besitzt eine Größe von 5° x 5°. Die Empfindlichkeit liegt bei 19.5 Magnituden.

1.3.4 Der Imager IBIS

Der Imager stellt die geforderte gute Bildauflösung im Gammabereich zur Verfügung. Auch dieses Instrument ist ein Coded Mask Teleskop. Der ortsauflösende Detektor besteht aus zwei Schichten. Die obere Schicht wird aus 16384 einzelnen Pixeln aufgebaut, wobei jedes dieser Pixel aus dem Halbleiter Cadmium Tellurid (CdTe) besteht. Bei der unteren Schicht werden 4096 Pixel verwendet, die aus dem Szintillationskristall Cäsium Jodid (CsI) aufgebaut sind. Die Detektorfläche umfaßt bei der CdTe-Schicht 2621 cm^2 und bei der CsI-Schicht 3100 cm^2. Durch die Aufteilung in zwei Schichten wird die Empfindlichkeit in den einzelnen Energiebereichen optimiert, und zusätzlich kann der Weg eines Teils der Photonen durch den Detektor nachvollzogen werden. Die kodierte Maske besteht aus Wolfram und befindet sich 3.20 m über der Detektorebene. Die Apertur wird durch ein Wolfram Schild begrenzt. Der Detektor ist von einem aktiven Bismuth Germanat (BGO) Antikoinzidenz-Schild umgeben, der die Rückwand und die Seiten der Detektoren abschirmt. Der Energiebereich des Imagers reicht von 15 keV bis 10 MeV. Die spektrale Auflösung $\Delta E/E$ beträgt 6% bei einer Photonenenergie von 1 MeV. Bei Energien von 100 keV liegt sie unter 7%. Die Winkelauflösung beträgt 12 arcmin FWHM. Das voll kodierte Gesichtsfeld hat eine Größe von 9° x 9° und das teilweise kodierte Gesichtsfeld umfaßt 19° x 19° [2].

In Abbildung 1.6 ist ein Funktionsdiagramm des Imagers dargestellt. Im folgenden möchte ich auf die hier dargestellten Teile der Hardware näher eingehen.

1.4 Der Imager im Detail

1.4.1 Der CdTe-Detektor

Jedes der 16384 CdTe-Pixel hat eine Fläche von 4x4 mm^2 und eine Dicke von 2 mm. Bedingt durch diese geringe Dicke ist dieser Detektor nur für den niederenergetischen Bereich einsetzbar. Bei 150 keV Photonenenergie beträgt die Effizienz nur noch 50%. Der Arbeitsbereich des CdTe-Detektors reicht beim Imager von 15 keV bis zu 200 keV. Die einzelnen Pixel werden im Detektor zu Modulen zusammengefaßt, wobei jedes Modul aus 128 Polyzellen und jede Polyzelle aus 16 Pixeln besteht. Insgesamt gibt es 8 dieser Module, so daß in jedem Modul 2048 Pixel zusammengefaßt sind.

1.4.2 CdTe Analog Front End Electronics

Die analoge Front End Electronics (AFEE) wird aus ASICs (Application Specific Integrated Circuit) aufgebaut. Jeder dieser ASICs bearbeitet vier Kanäle. Für jeden Kanal übernimmt der ASIC die Verstärkung, die Pulsformung und die obere und untere Energieschwellen-Diskriminierung.

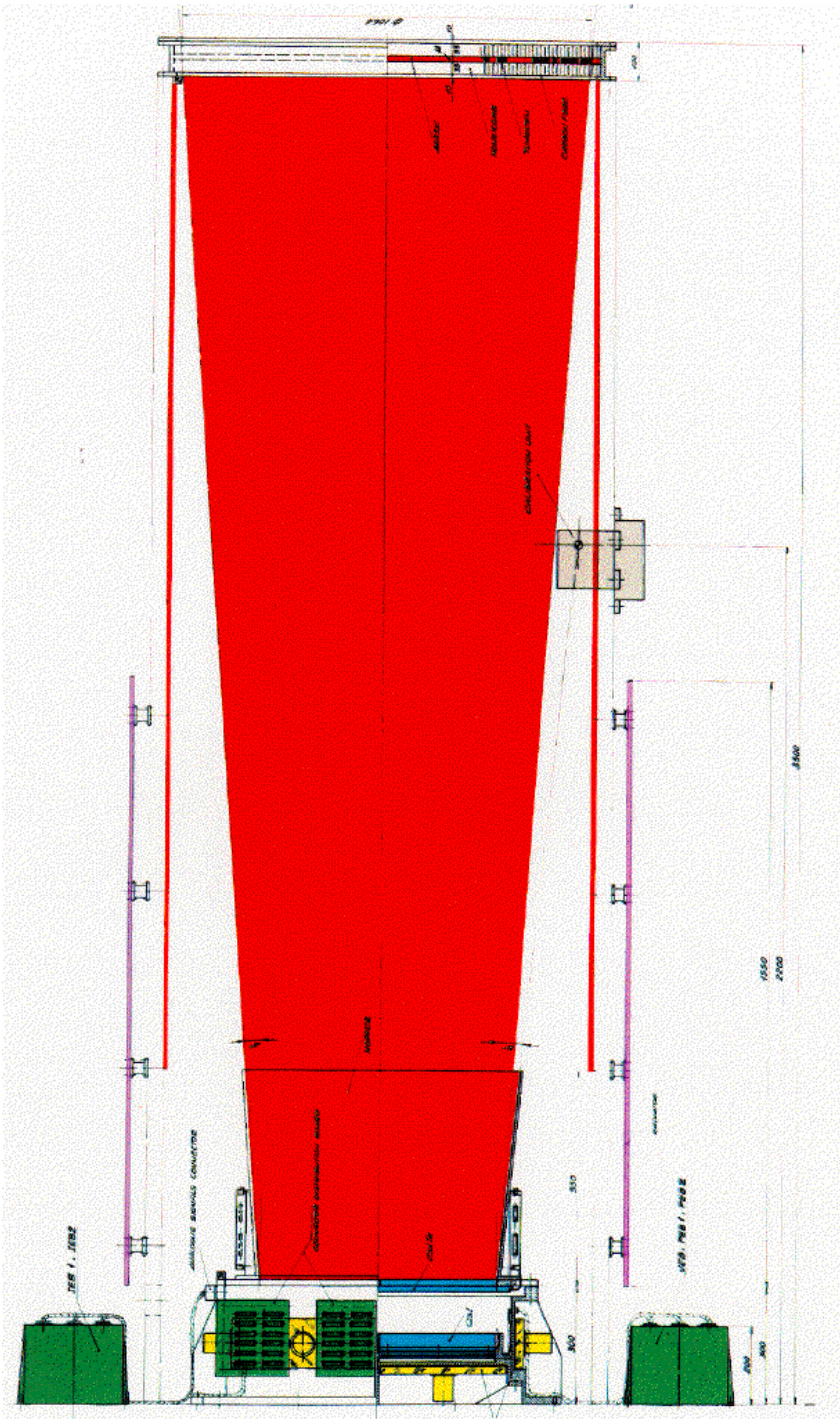


Abbildung 1.5: Der Imager

1.4.3 CdTe Module Control Electronics

Jedes der acht Module des CdTe-Detektors ist mit einer Module Control Electronics (MCE) verbunden. Die MCEs haben verschiedene Aufgaben. In ihnen findet die A/D-Wandlung des Detektorsignals statt. Außerdem werden dort das Zeitwort, die Pixeladresse und das Kalibrations-Flag für die Ereignisse kreiert. Das Kalibrations-Flag wird für alle Ereignisse gesetzt, die durch ein Photon der Kalibrations-Quelle erzeugt wurden. Zusätzlich wird die Pulshöhe und die Anstiegszeit bestimmt. Diese beiden Werte werden für die Energiekorrektur der Ereignisdaten benötigt. Die Daten werden nach der

Zeit geordnet in einem Ausgangs-FIFO gespeichert. Jede der acht MCEs besitzt ein solches Ausgangs-FIFO. Die MCEs beinhalten außerdem die Stromversorgung für jedes Modul.

1.4.4 Der CsI-Detektor

Jeder der 4096 CsI-Szintillationskristalle, aus denen die Detektorschicht aufgebaut ist, wird mit einer PIN-Diode verbunden, die das Szintillationslicht einfängt. Auch diese Detektorschicht ist in acht Module aufgeteilt, wobei jedes dieser Module 512 Kristalle umfaßt.

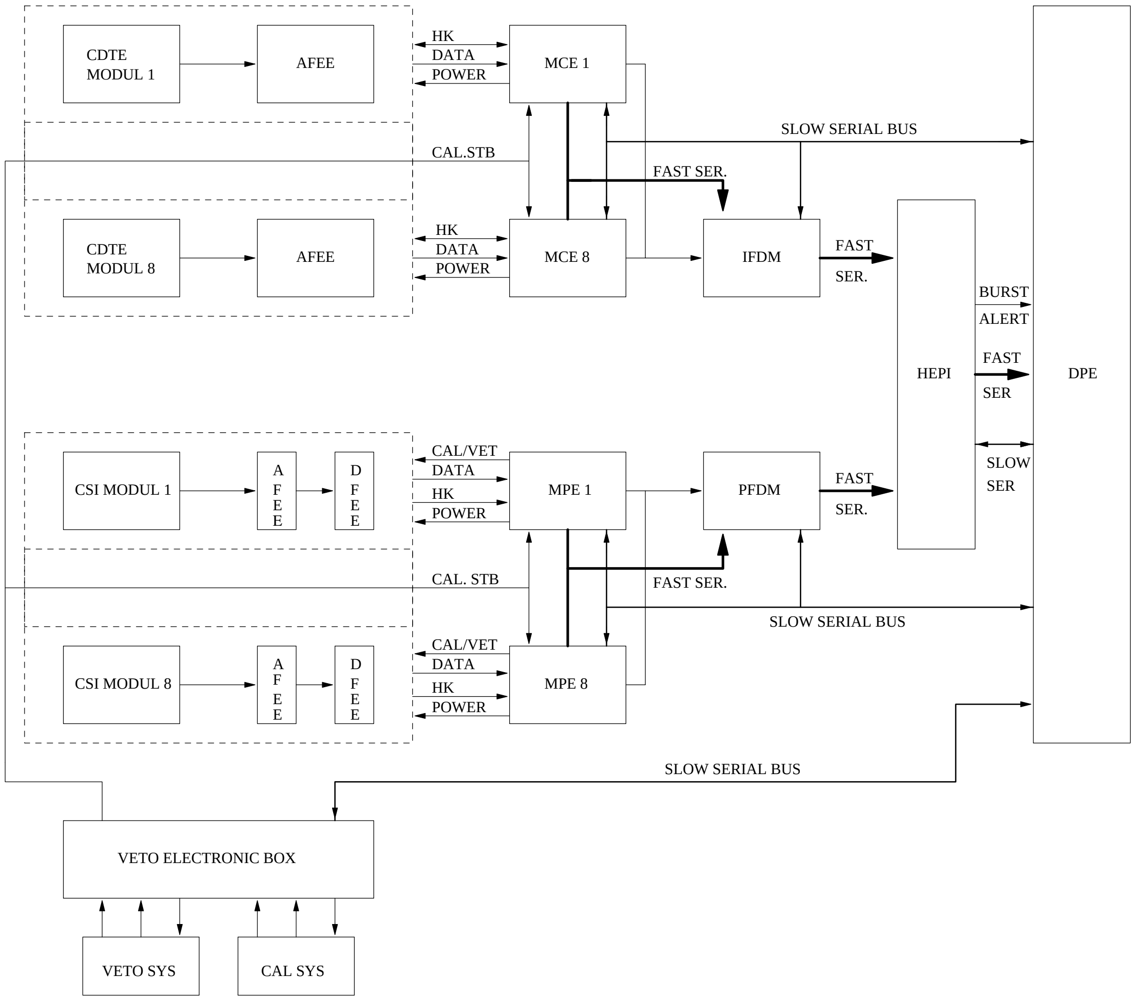


Abbildung 1.6: Komponenten der Imagerelektronik

Die einzelnen CsI-Module besitzen die gleiche Fläche wie die CdTe-Module. Das heißt ein CsI-Pixel entspricht der Fläche von vier CdTe-Pixeln. Die Länge der Kristalle beträgt 30 mm. Der Arbeitsbereich dieser zweiten Detektorschicht beginnt bei etwa 100 keV und endet bei 10 MeV.

1.4.5 CsI Analog Front End Electronics und Digital Front End Electronics

Für jedes Detektormodul steht eine AFEE (Analog Front End Electronics) und zwei DFEE (Digital Front End Electronics) zur Verfügung. Die zweite DFEE dient Redundanzzwecken. Auch die CsI-AFEE besteht aus ASICs, wobei pro ASIC 16 Kanäle verarbeitet werden. Bei dieser Verarbeitung wird das Signal verstärkt, die Pulsformung und die Amplitudendiskriminierung durchgeführt.
Die Digitale Front End Electronics digitalisiert das Analogsignal aus den AFEE und erzeugt außerdem das Zeitwort und die Pixeladresse. Zusätzlich werden in der DFEE die Ereignisse mit dem Kalibrations-Flag und dem Mehrfach-Ereignis-Flag versehen. Mit dem Mehrfach-Ereignis-Flag werden alle Mehrfach-Ereignisse aus dem CsI-Detektor gekennzeichnet. Diese Mehrfach-Ereignisse stammen von Photonen, die durch Compton Streuung gestreut wurden und mehrere Ereignisse im CsI-Detektor ausgelöst haben.
In der DFEE werden alle Ereignisse herausgefiltert, die in Koinzidenz mit dem Veto-Signal aus dem Antikoinzidenz-Schild stehen. Die Daten werden auch hier nach der Zeit geordnet in einem Ausgangs-FIFO gespeichert.
Die Module Powering Electronics (MPE) stellt die Stromversorgung für die AFEE und die DFEE zur Verfügung.

1.4.6 Der FIFO Data Manager

Im Anschluß an die Front End Electronics folgt bei beiden Detektorschichten ein FIFO Data Manager (IFDM für die CdTe-Schicht, PFDM für die CsI-Schicht). Dieser FIFO Data Manager sortiert die Daten aus einer Detektorschicht nach der Zeit. Auf den genaueren Aufbau möchte ich in Kapitel 5 näher eingehen.

1.4.7 Das INTEGRAL Kalibrations-System

Das Kalibrations-System an Bord von INTEGRAL besteht aus einer ^{22}Na Quelle. Sie ist in einem 60 mm dicken BGO-Szintillator untergebracht. An diesen Szintillator sind zwei Photomultiplier angekoppelt. Ein zusätzlicher Schild, bestehend aus Wolfram, ist in Richtung des SPI und des JEM-X angebracht. Der Wolfram- und der BGO-Schild sollten eine ausreichende Abschirmung bieten, so daß kein zusätzlicher Hintergrund im Spektrometer und im Röntgen-Monitor erzeugt wird. Im BGO-Schild befindet sich eine konische Apertur, um die Emission der Gamma-Strahlung zur Detektorebene zu ermöglichen. Die Gamma-Strahlung hat eine Energie von 511 keV und 1275 keV. Wird vom Photomultiplier ein Ereignis im BGO-Szintillator registriert, so wird das Signal zur VEB (Veto Electronics Box) gesendet und dort verarbeitet. Sie gibt ein Signal zum CsI-Detektor weiter. Jedes Ereignis, das gleichzeitig mit diesem Signal im Detektor registriert wird, stammt normalerweise von einem Photon aus der Kalibrations-Quelle und wird im Datenwort mit einem Flag gekennzeichnet.

1.4.8 Der Antikoinzidenz-Schild

Die Detektorschichten sind von einem aktiven BGO-Schild umgeben. Dieser Schild umfaßt die Seiten der Detektoren sowie deren Rückseite. Der BGO-Schild ist aus einzelnen Modulen aufgebaut. Die BGO-Kristalle werden von Photomultipliern beobachtet. Wird in einem BGO-Kristall ein Ereignis registriert, erzeugen die Photomultiplier ein analoges Signal und leiten es zur Veto Electronics Box. Diese gibt ein Vetosignal zum CsI-Detektor mit dessen Hilfe koinzidente Hintergrund-Ereignisse unterdrückt werden können.

1.4.9 Data Processing Electronics

Die DPE (Data Processing Electronics) stellt die zentrale Verarbeitungselektronik des Imagers dar. Sie verfügt über einen 1750 Prozessor, Speicher und Interfaces zu dem OBDH (On Bord Data Handling) Satelliten Bus. Die DPE empfängt die wissenschaftlichen und Housekeeping Daten aus den Detektoren. Von der DPE werden die Telekommandos und das Taktsignal verteilt. In der DPE Box ist noch eine zusätzliche Elektronikkarte eingefügt. Dieses Hardware Event Prozessing Interface (HEPI) wird aufgrund der hohen Ereignisraten benötigt, da die CPU-Leistung der DPE nicht zur Verarbeitung aller Daten ausreicht. Die Entwicklung der HEPI war Gegenstand dieser Arbeit. Auf sie werde ich in Kapitel 6 noch ausführlich eingehen.
Die DPE übernimmt alle komplizierteren Aufgaben, bei denen eine größere Flexibilität benötigt wird. Außerdem generiert sie die Datenpakete für die Telemetrie. Sollte eine Datenkompression nötig sein, wird diese von der DPE vorgenommen.

1.5 Zählraten und Empfindlichkeit

Die Hintergrund-Zählraten und die Empfindlichkeit des Imagers wurden mit Hilfe einer Monte-Carlo-Simulation abgeschätzt [3],[4]. In die Simulation gingen dabei das geometrische Modell des Imagers nach G.Malaguti, ein detailliertes Modell des Spektrometers und ein vereinfachtes Satellitenmodell ein. Als Hintergrund-Quellen wurden der diffuse kosmische Röntgen- und Gamma-Hintergrund und die Protonen der Kosmischen Strahlung angenommen. Die Ereignisse, die von den Protonen der Kosmischen Strahlung ausgelöst werden, lassen sich in zwei Gruppen unterteilen. Die eine Gruppe besteht aus Ereignissen, die direkt durch ein hochenergetisches Photon ausgelöst werden. Die meisten dieser Ereignisse stammen von außerhalb des Detektors erzeugten Photonen. Sie durchdringen den Antikoinzidenz-Schild und geben ihre Energie im Detektor ab, ohne ein Vetosignal zu erzeugen. Die zweite Gruppe stammt aus radioaktiven Zerfällen im Detektor oder im umgebenden Material. Diese radioaktiven Isotope wurden durch die hochenergetischen Protonen erzeugt. Da der Hintergrund mit der Sonnenaktivität variiert, wurde für die Simulationen der maximale Hintergrund, das solare Minimum, angenommen. Die folgenden Abbildungen zeigen die aus der Monte-Carlo Simulation gewonnenen Hintergrundspektren für den Imager, die beiden Detektorschichten und die Empfindlichkeit des Imagers im Vergleich mit früheren Missionen.

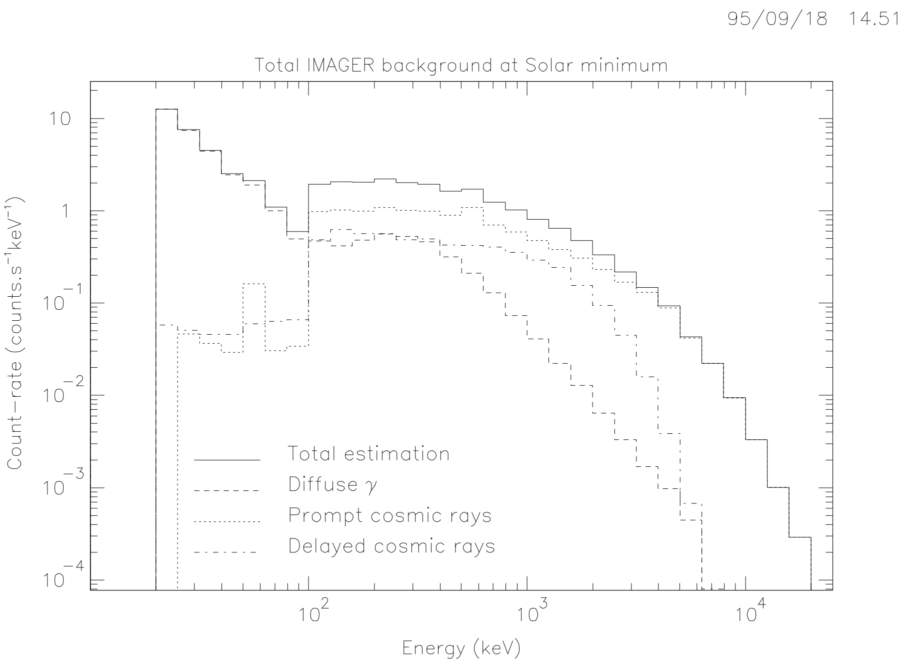


Abbildung 1.7: Der gesamte Hintergrund

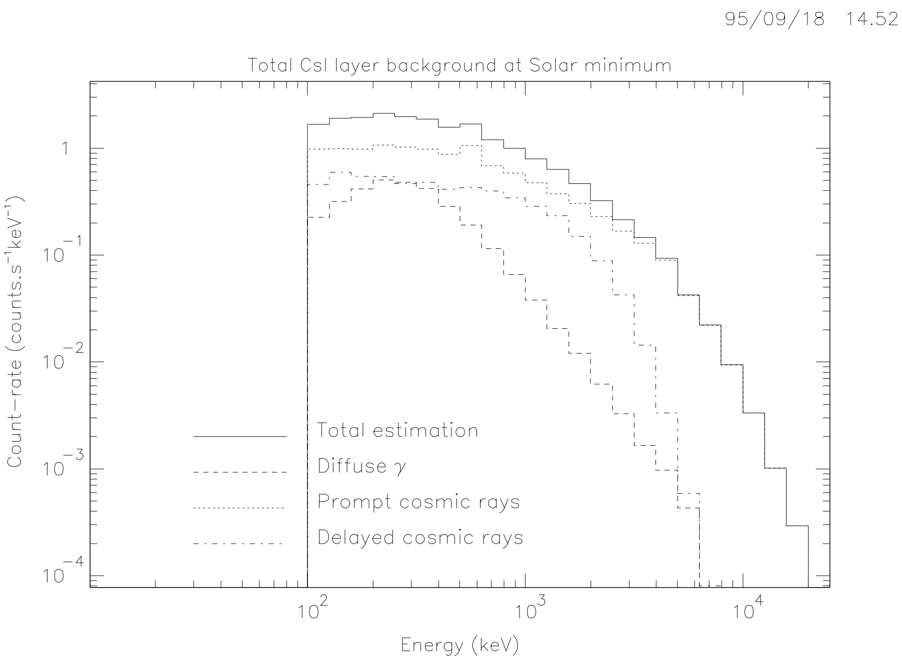


Abbildung 1.8: Der CsI-Hintergrund

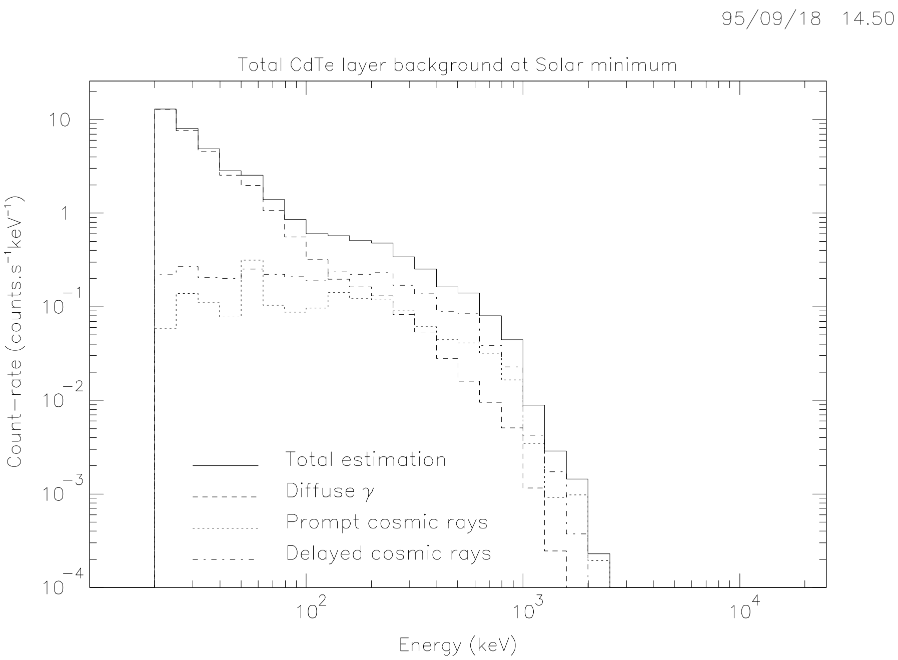


Abbildung 1.9: Der CdTe-Hintergrund

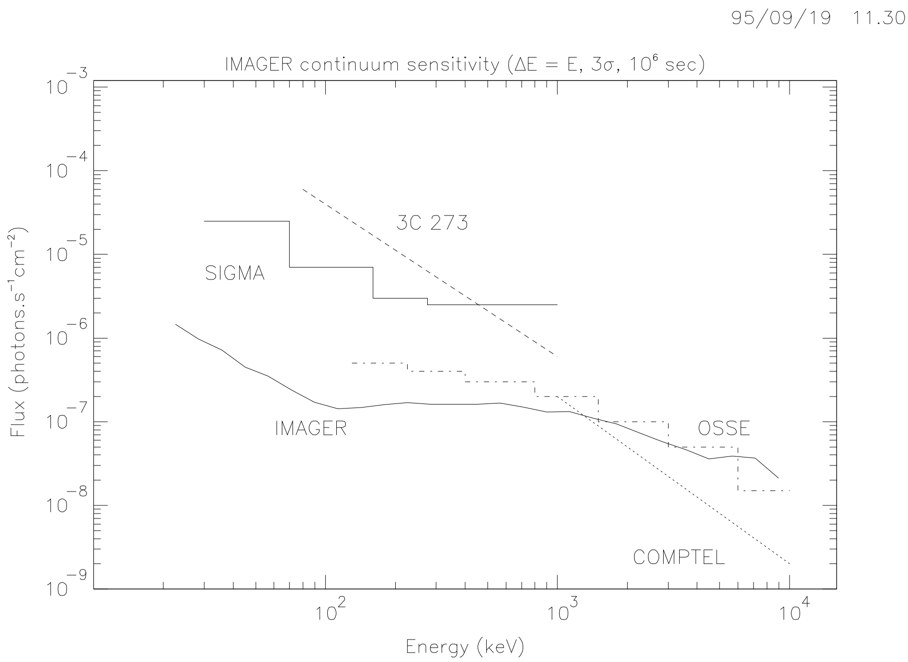


Abbildung 1.10: Kontinuums Empfindlichkeit (ΔE=E, 3σ, 10^6 sec)

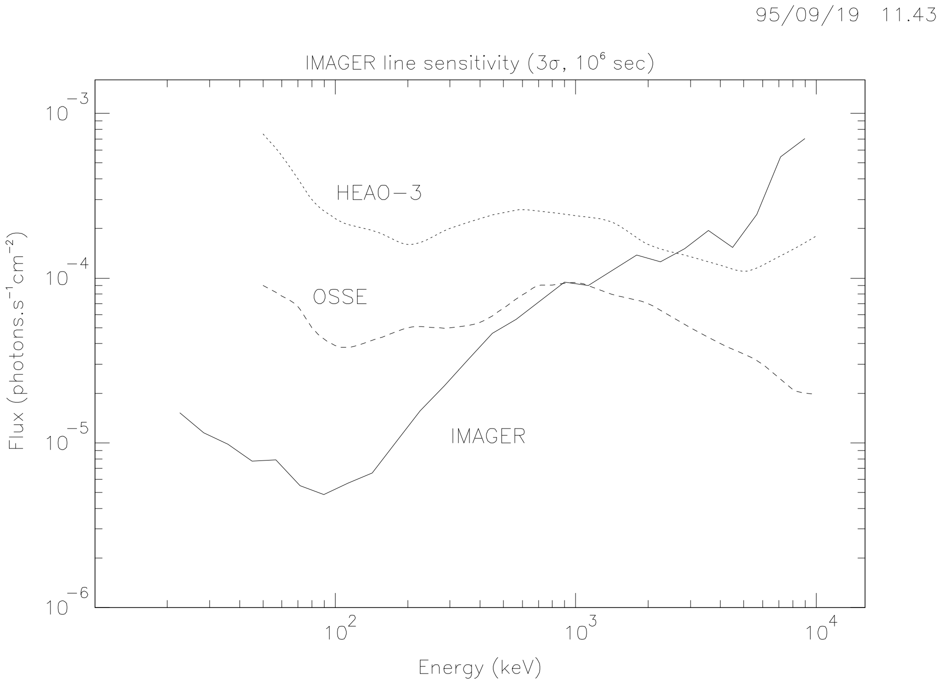


Abbildung 1.11: Linien Empfindlichkeit (3σ, 10^6 sec)

Der Wert für die Zählraten liegen für die CsI-Detektorschicht bei etwa 15000 counts/s und für die CdTe-Detektorschicht bei etwa 1000 counts/s. Ein Teil der Ereignisse des CsI-Detektors rührt dabei von Mehrfach-Ereignissen her. Bei diesen Ereignissen löst ein Photon mehrere Ereignisse im CsI-Detektor aus.
Überträgt man bei diesen Zählraten für jedes Ereignis die vollständige Information, so liegt man deutlich über der maximal zur Verfügung stehenden Telemetrierate. Daher ist es unbedingt nötig die Meßdaten an Bord zu verarbeiten, um eine Reduktion der Datenmenge zu erreichen.
Diese Aufgabe sollte urspünglich die DPE übernehmen. Abschätzungen der benötigten Rechenkapazität haben jedoch eine zu hohe Prozessorbelastung ergeben. Darum hat man sich für ein zusätzliche Elektronikkarte (HEPI) entschlossen, die diese Verarbeitung der Ereignisse übernehmen soll.

KAPITEL 2

2 Die Bauteilfamilie der ASICs

Die digitale Logikschaltung, die auf der HEPI zum Einsatz kommt, wird nicht diskret aufgebaut, sondern wird auf einen speziellen Baustein, einen ASIC (Application Specific Integrated Circuit), aufgebracht.

Die Vorteile solcher ASICs gegenüber diskreten Schaltungen sind vielfältig:

- geringer Platzbedarf
- kurze Schaltzeiten und schnelle Verarbeitung der Signale
- geringerer Leistungsbedarf bei gleicher Geschwindigkeit
- verbesserte Störsicherheit durch fehlende Leitungsverbindungen
- kostengünstige Produktion, da die Leiterplatte nur noch mit wenigen Bauteilen zu bestücken ist
- geringere Kosten bei der Lagerhaltung, da die Vielfalt der benötigten Bauteile zurückgeht

Diesen Vorteilen stehen aber auch Nachteile gegenüber, die den Einsatzbereich beschränken:

- hoher Entwicklungsaufwand
- hohe Entwicklungskosten
- eine spezielle Entwicklungsumgebung wird benötigt, um den Entwurf und die Simulationen durchzuführen

Für unsere Anwendung auf INTEGRAL sind die ersten vier Punkte von entscheidender Bedeutung. Aus diskreten Bauteilen aufgebaut, würde die Karte zuviel Platz beanspruchen. Die Vorteile der Platz- und der Gewichtsersparnis rechtfertigen den höheren Kostenfaktor.
Für die Karte, die im Flugmodell eingesetzt wird, kommt ein Standard-Zellen ASIC zum Einsatz. Dieser ASIC muß strahlungsfest sein, da er höheren Strahlendosen ausgesetzt ist. Im Entwicklungsmodell kommt ein FPGA (Field Programmable Gate-Array) zum Einsatz. Dieses FPGA ist ein programmierbarer Baustein und ist somit ideal zur Schaltungsentwicklung geeignet. In den nächsten Abschnitten möchte ich auf die verschiedenen Arten von ASICs, ihren Aufbau und die Anwendungsmöglichkeiten eingehen. Außerdem möchte ich kurz darstellen, welche Schritte bei der Entwicklung eines solchen ASICs durchlaufen werden.

2.1 Bauformen

Der Trend, Schaltungen immer mehr zu verkleinern und immer komplexere Probleme zu lösen, führt dazu, Digital- und Analogbausteine speziell zu entwerfen. Dies wird durch die Bauteilfamilie der kundenspezifischen Schaltkreise möglich. Diese ASICs beinhalten die Teilfunktionen einer Schaltung oder sogar die ganze Schaltung auf einem Chip.
Die ASICs lassen sich in zwei Gruppen aufteilen. Die eine Gruppe sind die Semi-Custom ASICs und die andere Gruppe stellen die Custom-ASICs dar. Die Semi-Custom-ASICs haben eine vorgefertigte Struktur. Sie werden nachträglich verdrahtet, um die vom Kunden gewünschte Funktion zu erfüllen. Diese Verdrahtung kann durch programmierbare Verknüpfungsfelder oder durch nachträgliches Aufbringen von Metallisierungsebenen geschehen, welche die Kontaktierung der auf dem Chip befindlichen Elemente übernimmt. Daher der Name Semi-Custom-Bauteile, da nur diese Kontaktierung nachträglich zu fertigen ist.
Die Custom-ASICs werden nach dem Wunsch des Kunden entworfen und gefertigt. Dabei kommen keine vorhandenen Fertigungsmasken zum Einsatz, sondern speziell für diesen ASIC entworfene Masken.

In Abbildung 2.1 sind die gebräuchlichsten ASIC in einer Übersicht dargestellt.

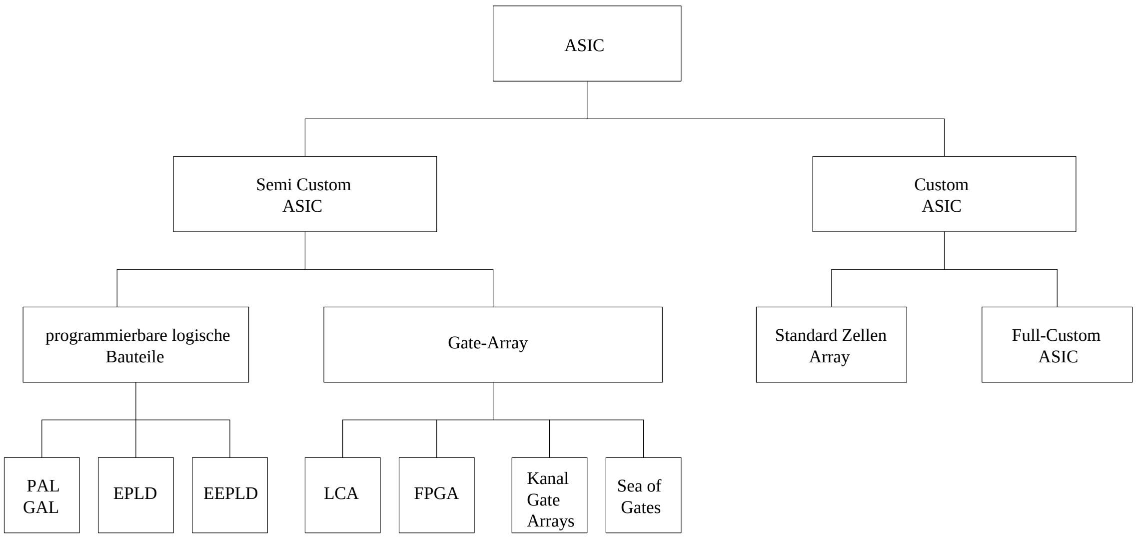


Abbildung 2.1: Bauteile aus der Gruppe der ASICs

2.1.1 PAL

PAL ist die Abkürzung für Programmable Array Logic [5]. Oft wird auch die Abkürzung PLA (Programmable Logic Array) verwendet. In Abbildung 2.2 ist ein PAL dargestellt, das in seinem Aufbau einem PROM (Programmable Read Only Memory) ähnelt. Die Eingangssignale und ihr Inverses werden in einem Array aus UND-Gattern miteinander verknüpft. Diese Verknüpfungen im UND-Array sind programmierbar. Sie bestehen aus Sicherungen, die bei der Programmierung durchgebrannt werden. Diese Programmierung kann mit einem einfachen Programmiergerät erfolgen, wie es auch für PROMs eingesetzt wird. Die von dem UND-Array erzeugten Signale werden über ein festverdrahtetes

ODER-Array zu den Ausgangssignalen verknüpft. Die PAL finden vor allem bei der Realisierung von Schaltungen Verwendung, die kombinatorische Logik enthalten.

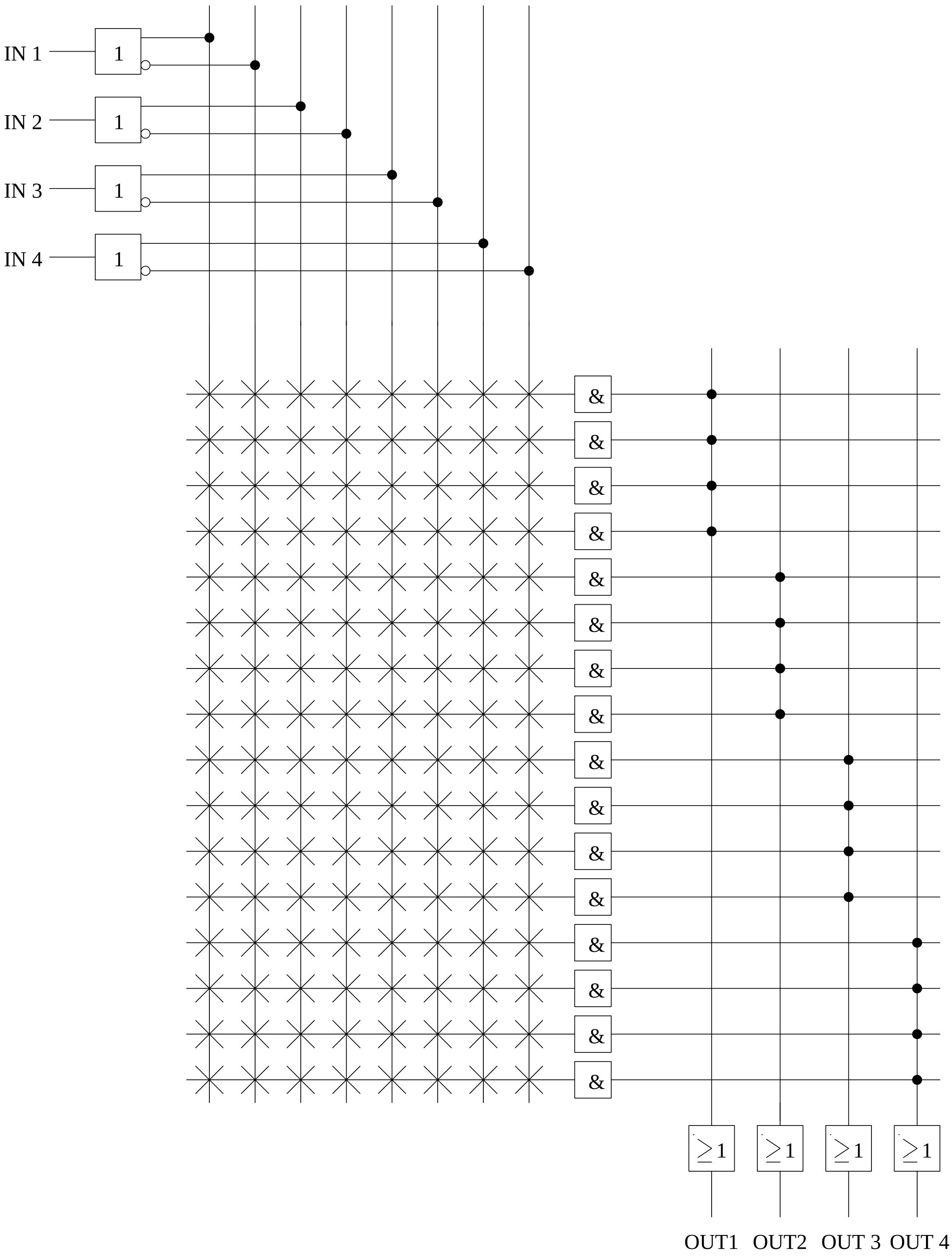


Abbildung 2.2: Prinzip eines PLAs

2.1.2 GAL

Die GAL (Generic Array Logic) sind durch ihre programmierbare ODER-Matrix flexibler als die PAL. Außerdem besitzten sie Ausgangsmodule, die unterschiedliche Konfigurationen zulassen. Damit lassen sich Rückkopplungen der Ausgänge auf die Eingänge und Tri-State Ausgänge verwirklichen.

2.1.3 EPLD und EEPLD

Die Abkürzungen stehen für Erasable Programmable Logic Device und Electrical Erasable Programmable Logic Device. Bei diesen beiden Arten von Bausteinen, die vom Prinzip her wie ein PAL funktionieren, sind die Verknüpfungen in den Arrays löschbar und neu programmierbar. Dieses Löschen funktioniert für die EPLD wie bei einem EPROM mit ultraviolettem Licht und für die EEPLD wie bei einem EEPROM durch eine definierte Löschspannung [6].

2.1.4 Gate-Arrays

Mit den Gate-Arrays steht eine deutlich größere Komplexität der ASICs zur Verfügung. Damit lassen sich Schaltungen mit über 300000 Gattern verwirklichen. Die Verdrahtung dieser Gatter kann nicht mehr durch eine direkte Programmierung erfolgen. Es wird daher in einem zusätzlichen Fertigungsprozeß mit Hilfe einer Maske eine Metallisierungsebene aufgebracht, welche die Verbindung zwischen den Gattern herstellt. Dabei unterscheidet man zwischen zwei Arten von Gate-Arrays. Bei den Kanal-Gate-Arrays verlaufen Verdrahtungskanäle zwischen den Bereichen in denen die Gatter untergebracht sind. In diesen Verdrahtungskanälen werden die Verbindungen der Gatter hergestellt. Bei den „Sea of Gates Gate-Arrays“ gibt es diese Verdrahtungskanäle nicht mehr. Sie verfügen über einen zentralen Bereich, der vollständig mit Gattern gefüllt ist. Die Verdrahtung erfolgt über unbenutzte Transistorzellen hinweg. Diese stehen dann nicht mehr als Schaltungselemente zur Verfügung.

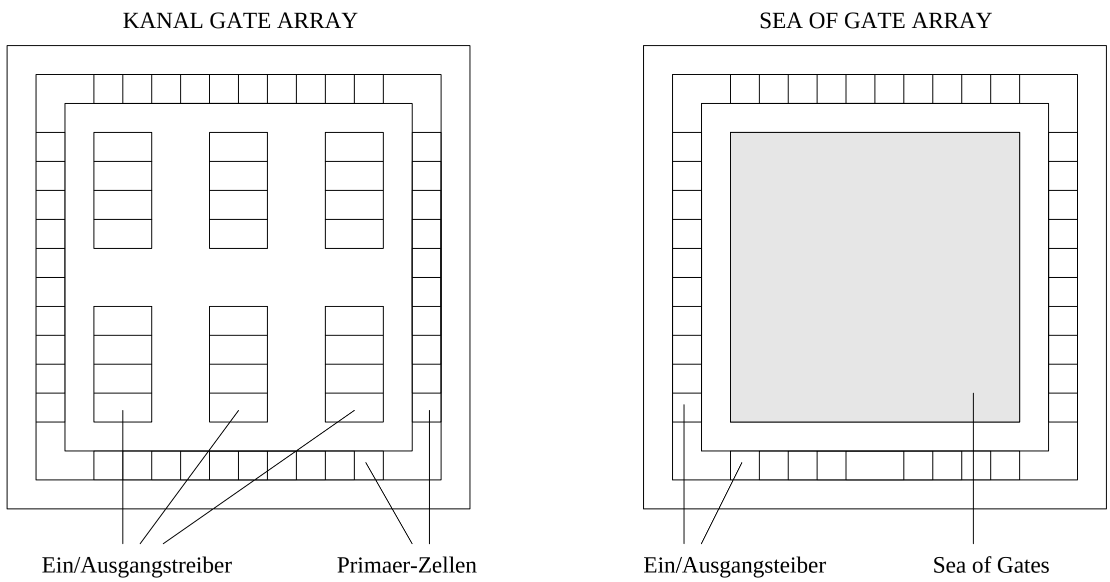


Abbildung 2.3: Gate-Arrays

2.1.5 Programierbare Gate-Arrays

2.1.5.1 Logic Cell Array

Beim LCA (Logic Cell Array) handelt es sich um ein vom Anwender programmierbares Bauteil. Wegen seiner Struktur und Komplexität wird es zu den Gate-Arrays gezählt und nicht zu den PLD (Programmable Logic Device). Das LCA besteht aus vielen gleichartigen Zellen. Diese Zellen enthalten die logischen Funktionen und sind vom Anwender konfigurierbar. Zwischen den Zellen verlaufen Verdrahtungskanäle über die man die einzelnen Zellen verbinden kann. An den Kreuzungspunkten dieser Verbindungsleitungen sitzen programmierbare Zellen, welche die einzelnen Verbindungsleitungen verknüpfen. Da diese Verbindungen immer wieder neu programmierbar sind, kann das LCA neu konfiguriert werden. Es muß aber auch bei jedem Einschalten konfiguriert werden, da die Programmierung ohne Versorgungsspannung erlischt.

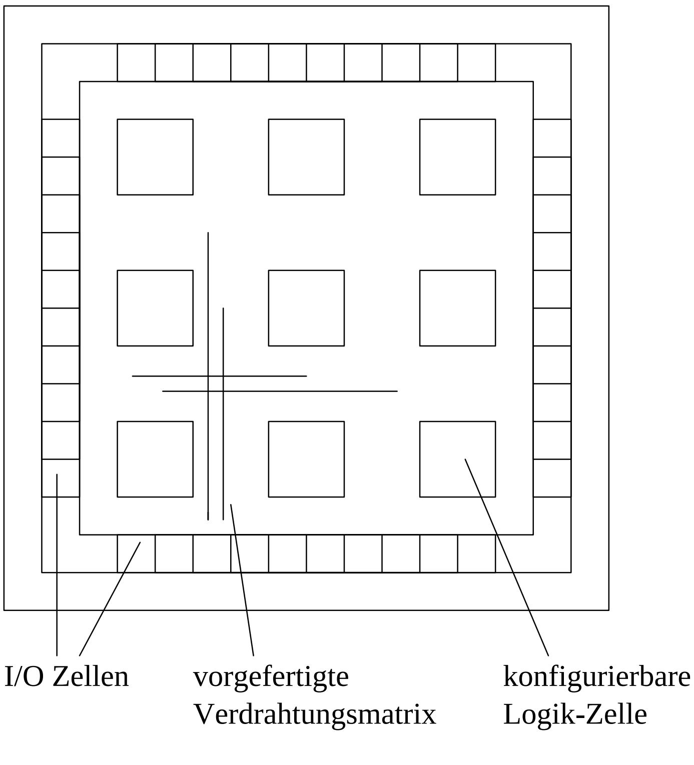


Abbildung 2.4: Architektur eines LCA

2.1.5.2 Field Programmable Gate-Arrays

Die FPGAs (Field Programmable Gate-Arrays) sind in ihrem Aufbau den Gate-Arrays sehr ähnlich. Auch sie haben horizontale und vertikale Verdrahtungen. Die horizontalen Verbindungen verlaufen zwischen den Logik Blöcken. Die vertikalen Verbindungen verlaufen über diese Logikblöcke hinweg. Die Verbindungsleitungen haben verschiedene Längen. Es gibt Leitungen die zwei benachbarte Module verbinden, aber auch solche, die sich über den ganzen Chip erstrecken. Werden mehrere lange Leitungen benötigt, verbindet man kurze Leitungen miteinander. Bei solchen Verbindungen kommt es zu einer Verlängerung der Signallaufzeit. Darauf muß beim Entwurf geachtet werden. Die Logikblöcke sind in Modulreihen angeordnet. Jeder Logikblock beinhaltet Flipflops und kombinatorische Logik, um die booleschen Verknüpfungen zur Verfügung zu stellen. Diese Logikblöcke könne frei konfiguriert werden, um verschiedene Funktionen zu erfüllen. Zudem lassen sich die Verbindungen zwischen den Logikblöcken programmieren. FPGAs gibt es inzwischen mit bis zu 100000 Gattern.

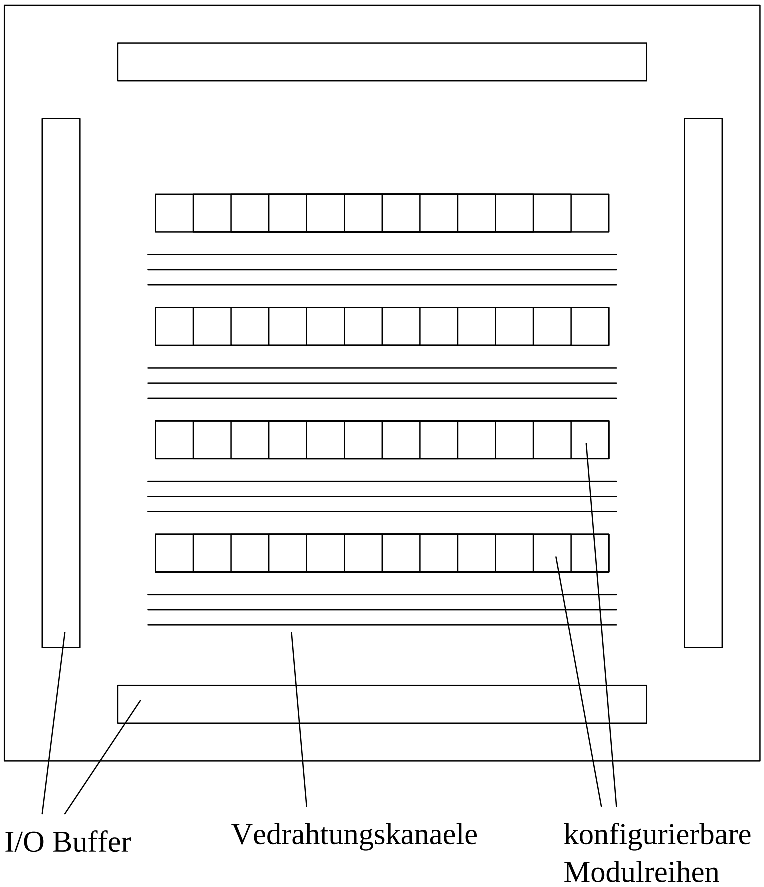


Abbildung 2.5: Architektur eines FPGAs

2.1.6 Standard-Zellen-Array

Die Standard-Zellen ASICs kommen ab einer Fertigungsmenge von 100000 Stück pro Jahr in Betracht, da ihre Fertigung sehr kostspielig ist. Es müssen alle Masken für die Chipfertigung speziell für den Schaltkreis entwickelt werden. Dabei kann jedoch die Chipfläche optimal ausgenutzt werden. Bei den Gate-Arrays ist dies nur theoretisch möglich. Daran sind einerseits Layout-Schwierigkeiten schuld, die daher rühren, daß nicht genug Verdrahtungskanäle zur Verfügung stehen. Andererseits wird die entwickelte Schaltung oft nicht genau die Größe des vorgefertigten Gate-Arrays besitzten, so daß Gatter ungenutzt bleiben. Da aber jede Fläche auf dem Chip mit Kosten verbunden ist, wird es ab einer bestimmten Stückzahl günstiger mit höheren Entwicklungskosten die Chipfläche zu optimieren.
Bei den Standard-Zellen-Arrays greift man auf Makros zurück. Das sind standardisierte Schaltungsteile, die vom Hersteller in einer Bibliothek geliefert werden. Sie lassen sich von der Software für die Schaltungsentwicklung verwenden.

2.1.7 Full-Custom-IC

Bei einem Full-Custom-IC wird ein noch höherer Aufwand bei der Entwicklung betrieben. Wie bei den Standard-Zellen-ICs sind ebenfalls alle Masken zu erzeugen. Man verwendet allerdings keine vorgefertigten Schaltungsteile aus der Bibliothek, sondern baut die Schaltung auf der Siliziumebene auf. Das heißt, die Schaltung wird mit den Transistoren auf dem Chip aufgebaut und auf dieser Ebene optimiert.

2.2 Designablauf

Der Entwurf elektronischer Schaltungen wird heuzutage mit Hilfe von Workstations und CAD/CAE Software (Computer Aided Design/ Computer Aided Engineering) durchgeführt. Im nächsten Kapitel werde ich die Software Werkzeuge (Tools) vorstellen, mit denen ich die Entwicklung durchgeführt habe. Hier will ich den Designablauf beschreiben, der durchgeführt wird, um einen ASIC zu fertigen und ein FPGA mit der entwickelten Logik zu programmieren.
Zuerst muß eine Spezifikation der benötigten Schaltung erstellt werden. Dabei wird das Verhalten der zu entwickelnden Schaltung und ihrer einzelnen Module definiert. Die Spezifikation nimmt eine wichtige Rolle ein, da Versäumnisse in der Spezifikation zu Fehlern beim Zusammenfügen der Module führen. Diese müssen dann überarbeitet werden, was mit Zeit- und Arbeitsaufwand verbunden ist.
Nach der Spezifikation kann das eigentliche Design beginnen. Dazu wird die Schaltung mit Hilfe der Software Tools im Rechner erstellt. Zur Überprüfung wird die Logik mit der Simulationssoftware simuliert. Dabei wird anfangs nur auf die logische Funktionalität geachtet. Treten Fehler auf, wird das Design korrigiert und erneut simuliert.

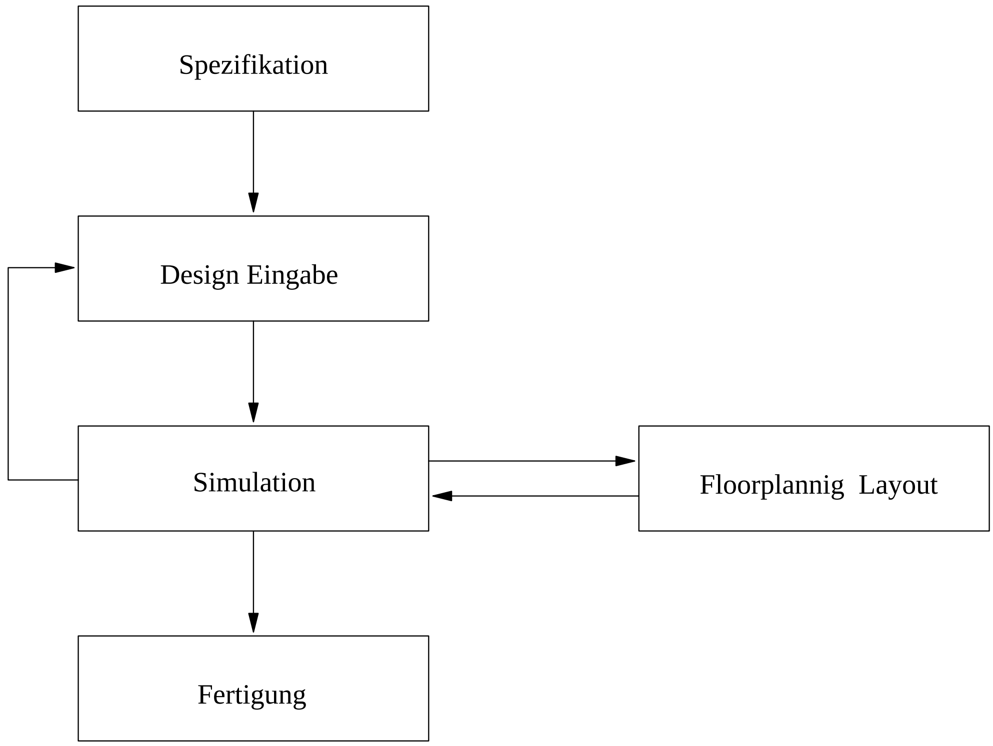


Abbildung 2.6: Der Designflow bei der ASIC Herstellung

Ist diese Überprüfung zur Zufriedenheit abgeschlossen, kann mit der Implementation der Schaltung begonnen werden. Die vom Rechner erzeugte Netzliste wird zum Hersteller gegeben. Diese Netzliste enthält die verwendeten Bauteile und ihre Verbindungen zueinander. Der Hersteller führt ein Floorplanning und ein Layout durch. Dies geschieht alles mit Rechner-Unterstützung. Beim Floorplanning werden die einzelnen Module auf dem ASIC plaziert. Das nachfolgende Layout plaziert die in der Netzliste beschriebenen Gatter am Platz des zugehörigen Moduls. Außerdem werden die Verbindungen zwischen den Gattern erstellt. Die durch das Routing gewonnenen tatsächlichen Signallaufzeiten werden für eine weitere Simulation herangezogen. Dabei wird überprüft, ob das Timing der Ein- und Ausgänge und der Verbindungen zwischen internen Registern noch stimmt. Dieser Vorgang ist meist ein iterativer Prozeß. Ist diese Simulation schließlich zur Zufriedenheit abgeschlossen, kann beim Hersteller mit der Maskenerstellung begonnen werden. Danach wird der Chip gefertigt.

Beim FPGA läuft der Vorgang ähnlich ab. Nachdem die logische Funktionalität überprüft wurde, wird mit Hilfe der Entwurfsoftware die Schaltung auf dem FPGA plaziert. Danach wird eine Timing Simulation durchgeführt. Funktioniert die Schaltung mit den durch die Plazierung auf dem FPGA entstandenen Laufzeiten, kann ein File erzeugt werden mit dem das FPGA programmiert wird.

2.3 Strahlungsfeste ASICs

Aufgrund der im Orbit auftretenden Strahlenbelastung muß der ASIC, der bei INTEGRAL eingesetzt wird, strahlungsfest sein. Diese Strahlungsfestigkeit wird durch ein spezielles Herstellungsverfahren erreicht.

Die ionisierende Strahlung kann in Form von Röntgen- oder Gammastrahlen oder in Form von Teilchenstrahlung auftreten. Diese Strahlung kann im Silizium oder in den isolierenden Materialien des Transistors Elektron-Loch-Paare erzeugen. Sind dies nur wenige, werden sie im Silizium zum Plus- und Minus-Pol hin abgezogen. Die entstehenden Elektronen in den Isolationschichten des Bauteils gelangen gleichfalls zum positiven Pol der Stromversorgung. Die Löcher bewegen sich jedoch langsamer als die Elektonen und neigen dazu, in der Silizium-Isolator-Grenzschicht hängenzubleiben. Durch die dadurch gebildete Ladung, kann sich die Transistor-Schwell-Spannung verändern. Ist die Menge der erzeugten Elektron-Loch-Paare groß genug, wird ein großer Strompuls zu den Polen der Stromversorgung erzeugt. Dieser Strompuls kann ausreichen, den Zustand einer RAM Speicherzelle oder eines Flipflops zu verändern.
Herkömmliche CMOS (Complementary Metal-Oxide-Semiconductor) Bauteile verkraften Strahlungsdosen bis 10 krad (Si). Bipolare Bauteile sind wesentlich strahlungsfester, können aber aufgrund der höheren benötigten Leistung nicht eingesetzt werden. Für INTEGRAL werden Bauteile mit einer Strahlungfestigkeit von 100 krad benötigt. Sollte eine ausreichende Abschirmung verfügbar sein, kann diese Anforderung auf 20 krad gesenkt werden. Da normale Bauteile solche Strahlendosen nicht verkraften, werden spezielle Bauteile benötigt.
Strahlungsfeste Bauteile werden mit einem SOS (Silicon on Saphire) Prozeß erzeugt. Bei diesem Prozeß bekommt jeder Transistor sei eigenes Stück Silizium. Jedes dieser Stücke ist gegenüber den anderen isoliert. Beim Herstellungsprozeß wird alles Silizium von den nichtaktiven Zentren entfernt, so daß dort nur das Saphir Substrat übrigbleibt. Dieser Herstellungprozeß ist aufwendiger zu realisieren und verteuert die Bauteile um ein Vielfaches.

Kapitel 3

3 Mentor Graphics

Mentor ist ein CAE-Entwicklungssystem, das zum Entwurf und zur Simulation von analogen und digitalen Schaltungen dient. Es kann zum Entwurf von ASICs und FPGAs verwendet werden. Mit Hilfe dieses Programms läßt sich die Zeit von der Planung bis zum fertigen Produkt deutlich verkürzen. Die Anzahl der Iterationen bis zur Entwicklung eines funktionierenden Schaltungsentwurfs wird verringert. Die Entwickler sind nicht mehr darauf angewiesen, ihren Entwurf aufzubauen oder fertigen zu lassen, um ihn zu testen. Sie können die Schaltung im Rechner simulieren und Fehler frühzeitig entdecken. Im folgenden sollen die wichtigsten Werkzeuge (Tools) des Mentor-Entwicklungssystems kurz beschrieben werden.

3.1 Der Designmanager

Der Designmanager stellt das zentrale Programm zum Aufruf und zur Verwaltung der einzelnen Werkzeuge und Entwürfe dar. Selbstverständlich können die einzelnen Programme auch von der Kommandozeile aus gestartet werden. Der Designmanager bietet jedoch eine benutzerfreundliche graphische Oberfläche dazu an.
Designs können kopiert und in andere Verzeichnisse verschoben werden. Das Setup der Session läßt sich mit dem Designmanager verändern. Abbildung 3.1 zeigt den Designmanager, mit seinen zwei Fenstern. Im rechten Fenster wird der Inhalt des angewählten Verzeichnisses angezeigt. Das linke Fenster enthält Icons für die Werkzeuge. Ganz rechts sind Buttons für verschiedene Funktionen zu sehen.

3.2 Der Designarchitekt

Der Designarchitekt dient zur Eingabe des Designs. Die Eingabe kann dabei auf verschiedene Arten erfolgen. Zum einen kann das Design als Schaltplan gezeichnet werden. Diesen Schaltplan nennt man Schematic. Das Design wird dabei aus Bauteilen aufgebaut, die in einer Bibliothek zur Verfügung stehen. Die Hersteller der verschiedenen FPGAs und ASICs liefern diese Bibliotheken für die von ihnen hergestellten Bauteile. Sie enthalten Makros von Bauteilen, die von den Herstellern erstellt wurden. In diesen Makros sind Beschreibungen über die Funktionalität der Bauteile enthalten, die vom Simulationsprogramm verwendet werden können. Dabei garantieren die Hersteller, daß die Funktionalität des auf dem Baustein implementierten Teiles, der im Makro beschriebenen entspricht. Diese Bauteile werden auf einem Fenster plaziert. Dort können dann auch die Verbindungsleitungen zwischen den Bauteilen gezogen werden. Diese Eingabe der Schaltung entspricht im Prinzip der Zeichnung eines Schaltplans auf einem Blatt Papier. Dabei sind die Möglichkeiten, diese Zeichnung zu manipulieren, jedoch sehr groß, da der Designarchitekt im Prinzip ein CAD-Programm darstellt. Damit lassen sich die Bauteile plazieren, verschieben, löschen und viele andere Operationen durchführen.

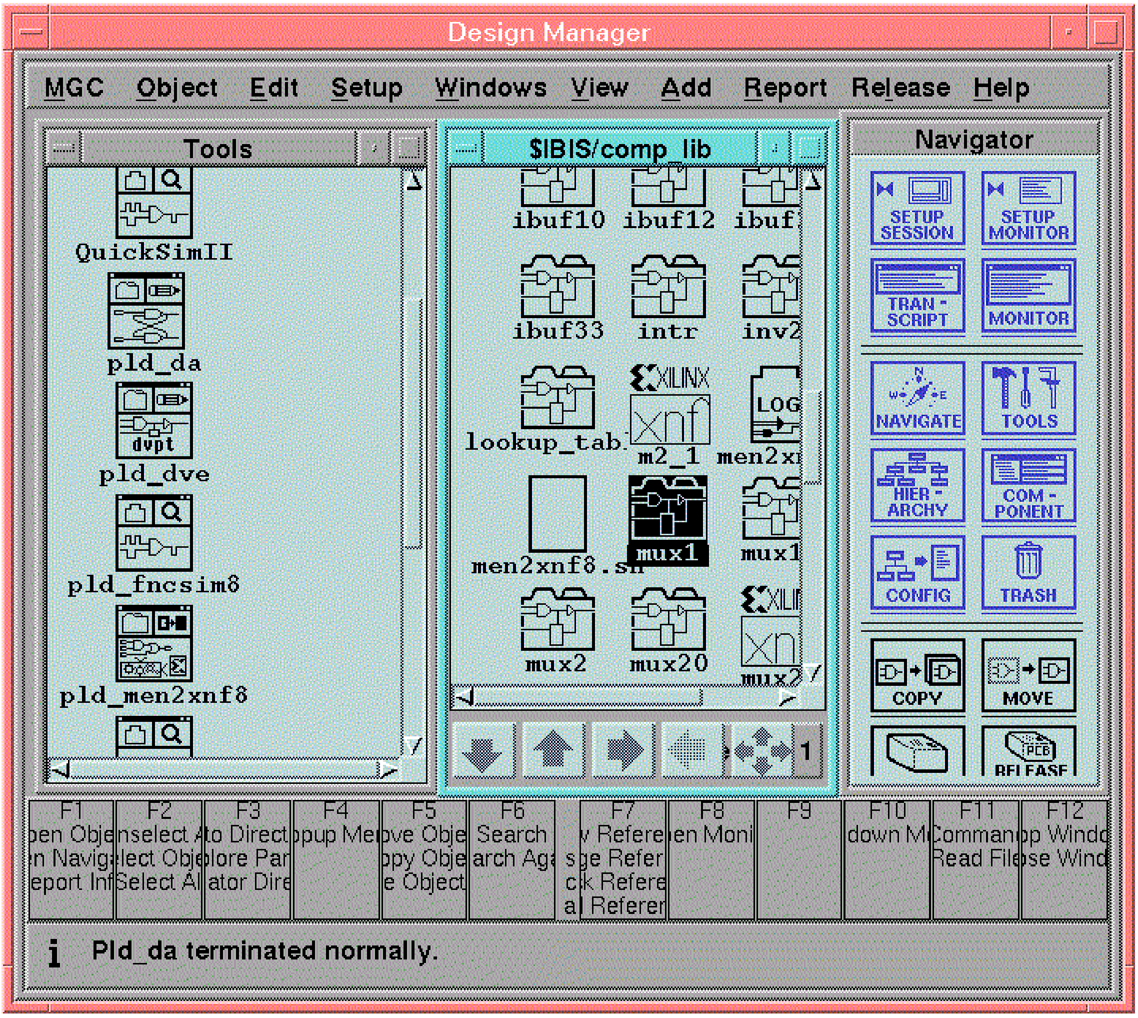


Abbildung 3.1: Bildschirmkopie des Designmanagers

Eine besondere Funktion bei der Bedienung dieses CAD-Programms nehmen die sogenannten Strokes ein. Ein Stroke wird ausgeführt, indem man mit gedrückter mittlerer Maustaste ein einfaches Symbol auf dem Bildschirm zeichnet. Jedes Symbol hat dabei eine andere Funktion. Man kann mit diesen Strokes die Zeichnung vergrößern, verkleinern, Bauteile selektieren und verschieben. Dabei wird kein Pull-Down-Menue oder die Tastatur benötigt. Damit erleichtert sich die Bedienung des Programms erheblich. Voraussetzung ist natürlich die Kenntnis der Strokes, sowie eine gewisse Übung bei ihrer Bedienung. Das gezeichnete Symbol wird vom Programm in eine Zahlenfolge übersetzt. In Abbildung 3.2 ist dies für den ‘U’ Stroke gezeigt. Der ‘U’ Stroke wird dabei in die Zahlenfolge 1478963 übersetzt.

Die Abbildung 3.3 zeigt ein einfaches Beispiel eines Entwurfs. Es stellt einen Multiplexer dar, der abhängig von einem Select-Signal, zwischen zwei Eingängen auswählt. Das linke Fenster zeigt das Schematic mit seinen Bauteilen. Auf dem rechten Fenster ist das Symbol für dieses Schematic dargestellt

Der Designarchitekt überprüft außerdem das Design auf formale Fehler hin. Zu diesen Fehlern zählen unter anderem überlappende Bauteile, unverbundene Pins und falsch definierte Busse.

Außer der Eingabe eines Designs über ein sogenanntes Schematic steht im Designarchitekt noch ein VHDL (Very High Speed Integrated Circuit Hardware

Description Language) Editor zur Verfügung. VHDL ist eine Hardware-Beschreibungssprache, die zur Entwicklung von Schaltungen verwendet werden kann. Sie zählt zu den objektorientierten Sprachen. Dabei erfolgt der Entwurf nicht, indem man einen Schaltplan zeichnet, sondern wie bei einer Programmentwicklung mit anderen Programmiersprachen. Es stehen dabei von anderen Sprachen bekannte Befehle wie Schleifen, If-Abfragen, etc. zur Verfügung. Mit Hilfe dieser Elemente wird das Verhalten der zu entwickelnden Schaltung beschrieben. Zusätzlich gibt es noch die Möglichkeit, die Schaltung über ihre Struktur zu beschreiben. Im Programmcode werden dabei die einzelnen Komponenten der Schaltung und ihre Verbindungen zueinander angegeben. Hält man sich bei der Beschreibung der Schaltung an bestimmte Richtlinien, so kann sie durch Programme in eine für FPGAs oder ASICs verwendbare Form synthetisiert werden. Der VHDL-Editor im Designarchitekt dient zur Eingabe des Codes und zur Compilierung.

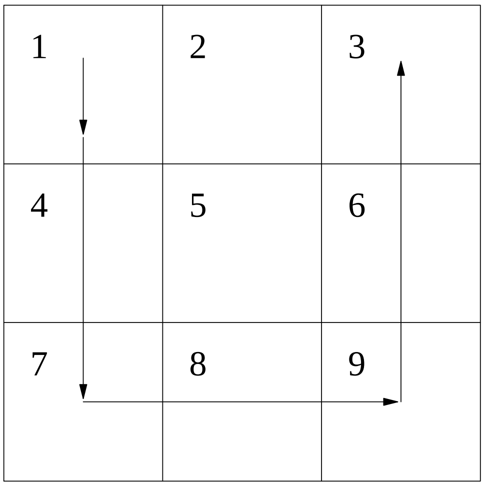


Abbildung 3.2: Der ‘U’ Stroke

3.2.1 Vorgehensweisen bei einem Entwurf

Das Design kann mit Hilfe des Designarchitekten hierarchisch aufgebaut werden. Einzelne Komponenten der Schaltung werden als separater Schaltplan gezeichnet. Von diesem Schaltplan wird ein Symbol erzeugt, welches diese Komponente repräsentiert. Dieses Symbol besitzt die gleichen Ein- und Ausgänge wie der Schaltplan, aus dem es erzeugt wurde. Die Funktionalität des Symbols wird durch den Schaltplan beschrieben. Wird eine Komponente der Schaltung mehrmals gebraucht oder in anderen Designs verwendet, kann das Symbol kopiert werden. Damit läßt sich eine eigene Bauteilbibliothek aufbauen. In jedem Symbol können weitere Symbole enthalten sein, die wiederum aus einer Zeichnung bestehen. Hat man nun alle Komponenten entworfen, so fügt man sie auf einem sogenannten Top-Level zusammen. Damit wird der Schaltplan natürlich auch übersichtlicher, da weniger Teile in diesem Top-Level vorhanden sind. Dies nennt man hierarchisches Design, da nicht nur eine Ebene vorhaben ist, sondern mehrere ineinander geschachtelte Ebenen.

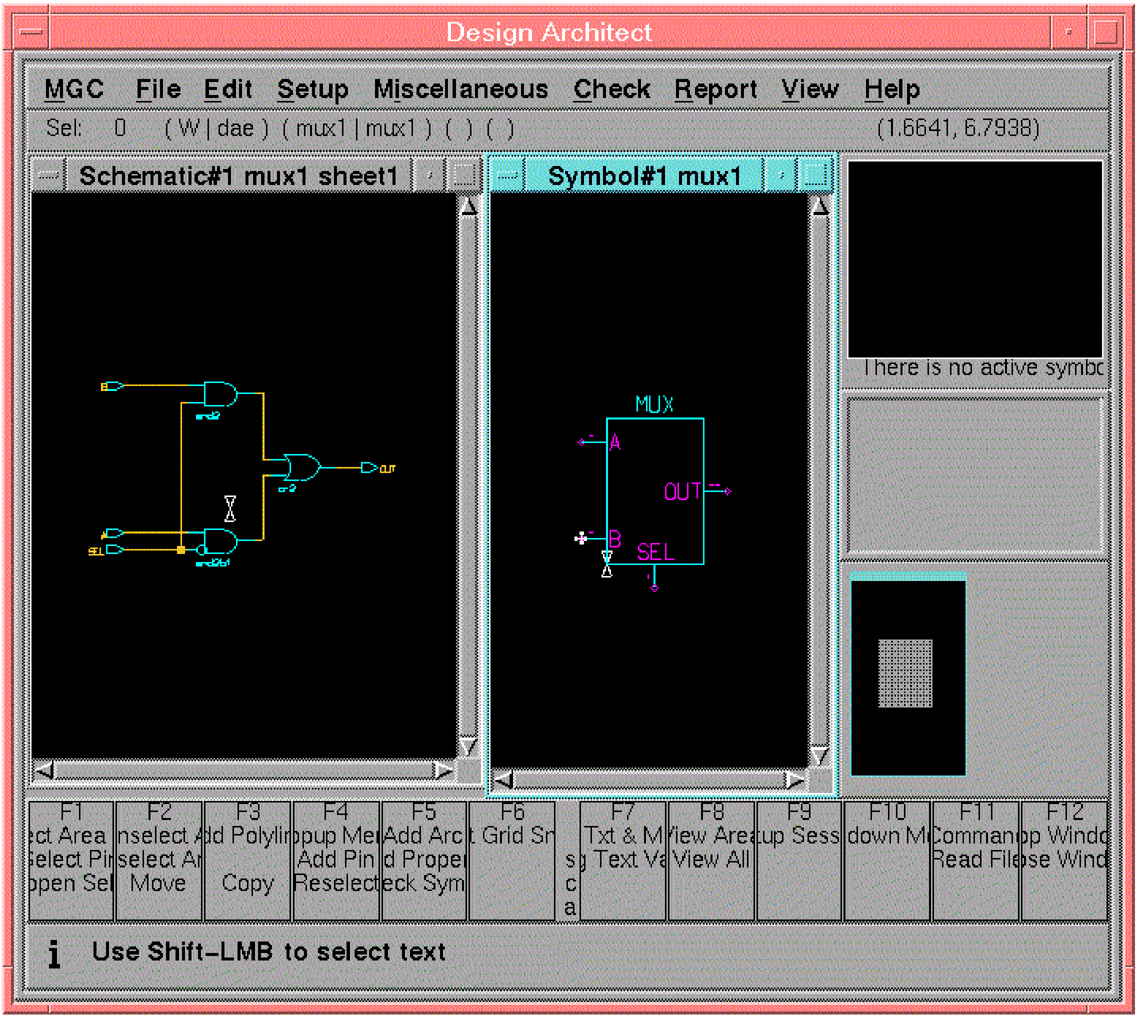


Abbildung 3.3: Bildschirmkopie des Designarchitekten

Beim Design einer Schaltung gibt es grundsätzlich zwei Vorgehensweisen. Die eine nennt sich Top-Down-Design. Dabei geht man von einem Top-Level aus. Dieser Top-Level hat Ein- und Ausgänge und soll eine gewisse Funktionalität besitzen. Diese Funktionalität teilt man in verschiedene Funktionsblöcke auf. Jeden dieser Blöcke versucht man wieder in Funktionsgruppen zu zerlegen, bis man schließlich auf der Ebene der Gatter angelangt ist, und die Schaltung damit vollständig beschrieben wird. Die andere Möglichkeit besteht darin, das Design von der Gatterebene her aufzubauen. Das heißt, der Entwurfsprozeß verläuft in umgekehrter Richtung wie beim Top-Down-Design. Darum nennt sich diese Art auch Bottom-Up-Design.

3.3 Das Simulationsprogramm Quicksim II

Quicksim II ist ein Logik-Simulator, der es ermöglicht, das mit dem Designarchitekt eingegebene Design zu testen. Es lassen sich eine funktionelle Simulation und eine zeitliche Simulation 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 mitberücksichtigt. Dazu müssen allerdings Informationen über diese Laufzeiten zur Verfügung stehen. Diese Timinginformationen gewinnt man aus der Plazierung der Schaltung auf dem ASIC oder FPGA. Zum Test der Schaltung lassen sich Signale auf die Eingänge der Schaltung legen und die Signale an

den Ausgängen verfolgen. Dabei lassen sich nicht nur Ein- und Ausgänge anzeigen, sondern auch interne Signale. Dies erleichtert die Fehlersuche erheblich. Im Simulator lassen sich vier verschiedene Fenster aufrufen. Eines zeigt dabei den Schaltplan an. Es dient zur Orientierung, welche Signale angezeigt werden und läßt sich zur Auswahl der Signale verwenden. In den restliche drei Fenstern werden die ausgewählten Signale angezeigt. Das „Trace-Fenster“ zeigt den Pegelverlauf des Signals dabei graphisch an. Das „List-Fenster“ zeigt die Signale, die Zeit und den Signalpegel an. Diese Informationen sind in tabellarischer Form aufgelistet. Im „Monitor-Fenster“ wird der momentane Zustand der ausgewählten Signale angezeigt.
In der Abbildung 3.4 ist eine Simulation des Multiplexers dargestellt. Links oben ist das Schematic-Window zu sehen, daneben das List-Window und im unteren Teil schließlich das Monitor- und Trace-Window.

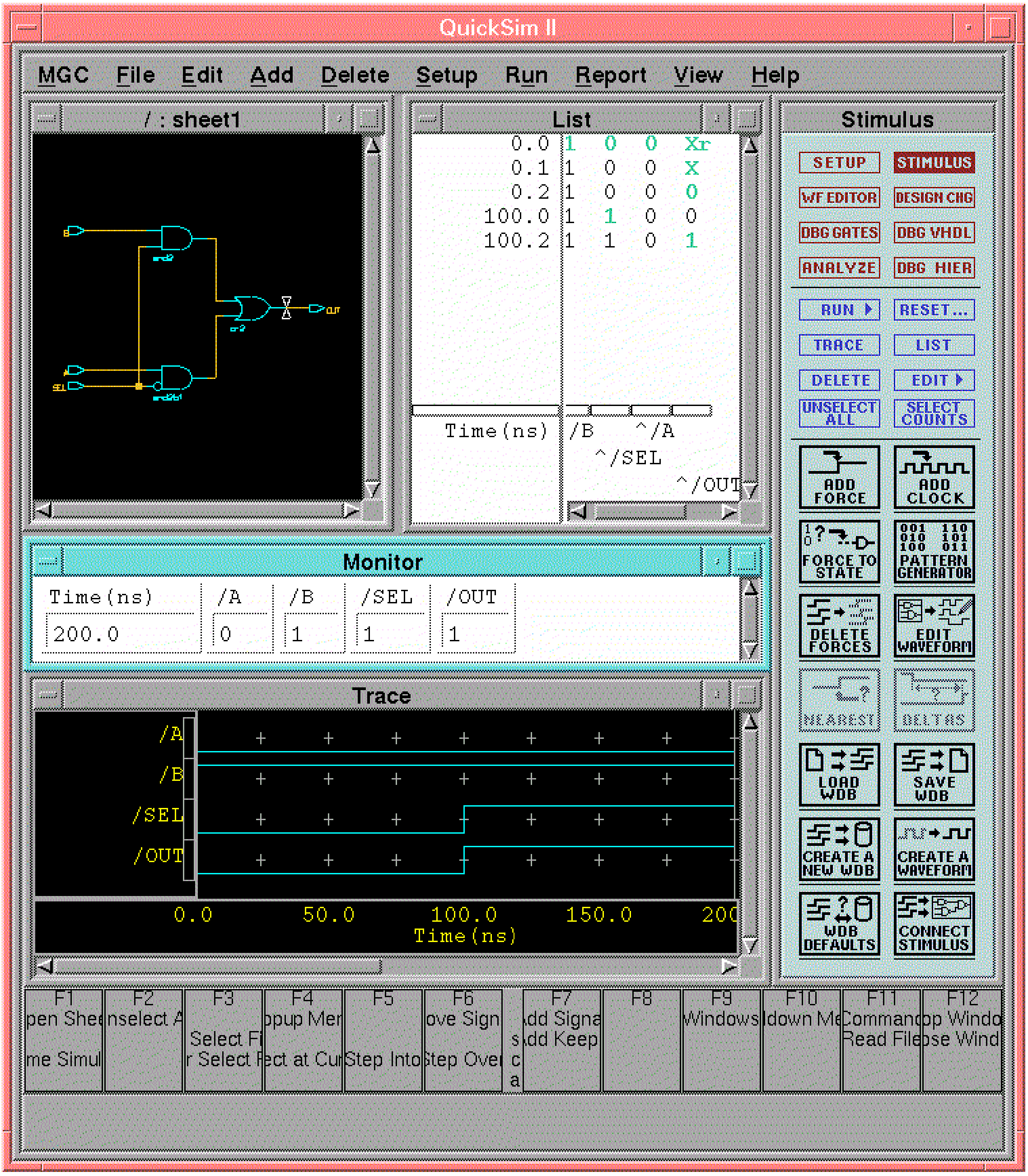


Abbildung 3.4: Bildschirmkopie des Simulationsprogramms Quicksim II

3.4 Die Xilinx-Werkzeuge zur Implementation des Designs

In unserem Institut werden FPGAs der Firma Xilinx verwendet. Auf dem Entwicklungsmodell der HEPI soll ein solcher FPGA zum Einsatz kommen. Um das erstellte Design auf dem FPGA von Xilinx zu implementieren sind verschiedene Schritte nötig. Um diese Schritte durchzuführen, liefert Xilinx verschiedene Werkzeuge, die den Entwickler bei dieser Arbeit unterstützen. Ich werde im folgenden die Werkzeuge vorstellen, die notwendig sind, um das Design simulieren zu können.

3.4.1 Bibliotheken

Eine Grundvoraussetzung, um den Schaltungsentwurf auf FPGAs von Xilinx verwenden zu können, ist die Benutzung der Bauteilbibliotheken des Herstellers. Es gibt zwei dieser Bibliotheken.
Die erste Bibliothek umfaßt die Standard-Bauteile, die Funktionen wie logische Gatter, Komparatoren, Zähler und Flipflops enthalten. Diese Bauteile haben dabei festgelegte Ein- und Ausgänge. Zum Beispiel ist die Breite des Dateneingangs eines Komparators vordefiniert. Auf dem FPGA wird dieses Bauteil aus den grundlegenden Gattern aufgebaut.
Die zweite Bibliothek nennt sich XBLOX. Sie enthält Module, die nicht durch Gatter, sondern durch eine High Level Description Language beschrieben werden. Der Benutzer kommt damit jedoch nicht in Kontakt. Er verwendet diese Bauteile fast gleich wie jene aus der normalen Bibliothek. Der Unterschied besteht darin, daß der Benutzer die Busbreiten der Bauteile frei wählen kann. Er nimmt das Bauteil, setzt es auf seinen Schaltplan und trägt am Bauteil die Busbreite ein. Für die spätere Implementation und Simulation müssen diese XBLOX-Module durch ein Synthese-Tool in eine Gatterbeschreibung umgesetzt werden.

3.4.2 Die funktionelle Simulation

Um einen im Designarchitekt eingegebenen Entwurf simulieren zu können, müssen verschiedene Schritte durchlaufen werden. Diese Schritte lassen sich manuell oder durch ein Script durchführen. Das Script ruft dabei die nötigen Programme automatisch auf.

3.4.2.1 MEN2_XNF

Das Men2XNF Script erzeugt aus dem Design ein Xilinx Netlist File (XNF). Dabei ruft es die Programme PLD_DVE, ENWRITE, EDIF2XNF und MemGen auf.

3.4.2.2 FNCSIM8

Dieses Script bereitet das Design für die funktionelle Simulation vor. Es ruft dazu mehrere Programme auf. Diese Programme sind Gen_Sch8, PLD_DVE_SIM und QUICKSIM-II.

Abbildung 3.5 zeigt die zur Simulation notwendigen Schritte. Der PLD_DVE (Design Viewpoint Editor) erzeugt einen XNF-Viewpoint. In ihm sind Regeln für die Konfiguration eines Designs gespeichert. Danach liest ENWRITE diesen Viewpoint und

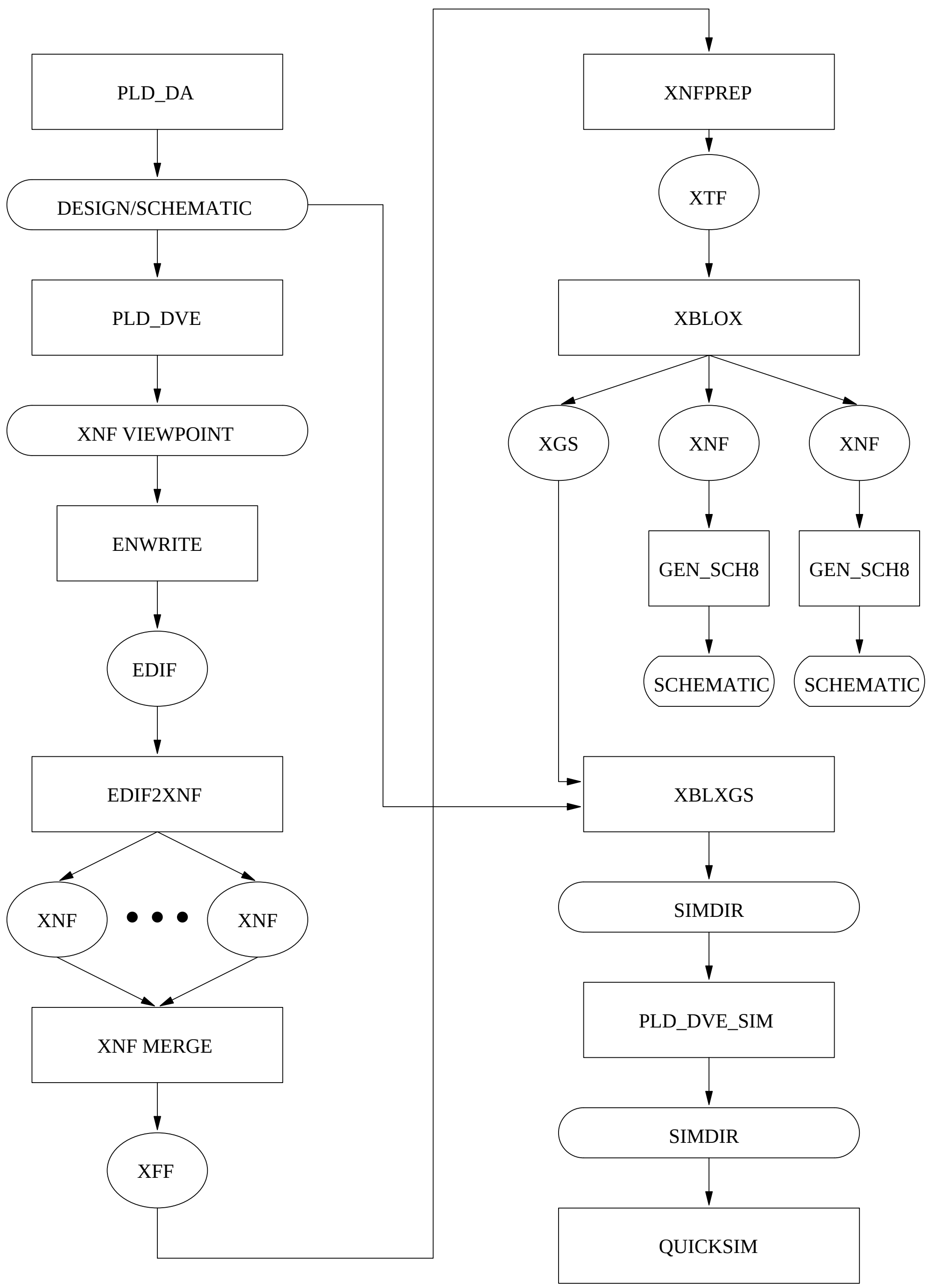


Abbildung 3.5: Der Design Flow bei der Vorbereitung zur Simulation [7]

erzeugt ein EDIF-File. Dieses EDIF-File ist das Netzlistenformat von Mentor. Danach wird das EDIF-File in ein XNF-File umgewandelt. Dies geschieht durch das EDIF2XNF Programm. Für jedes Modul im Schaltplan wird ein XNF-File erzeugt. Anschließend werden die einzelnen XNF-Files durch XNFMerge zusammengefaßt. Dieses File enthält keine Hierarchie mehr. Es besteht nur noch aus einer Ebene. Danach wird mit XNFPrep das File nach Fehlern und unbenutzter Logik durchsucht. XBLOX setzt nun die XBLOX Symbole in Logikgatter um. Die dabei entstehenden XNF-Files werden von Gen_Sch8 in Schaltpläne für die Simulation umgesetzt und im Simdir-Verzeichnis gespeichert. Zusätzlich erzeugt das XBLOX-Programm ein File mit der XGS-Endung. Es beschreibt, wie der Schaltplan für die Simulation gezeichnet werden soll. Das Programm XBLXGS liest dieses File und das ursprüngliche Schematic (Schaltplan) und erzeugt daraus ein neues Top-Level-Schematic. Mit PLD_DVE_SIM wird schließlich ein Simulations-Viewpoint erzeugt und die Simulation kann durchgeführt werden.

3.4.3 Die Implementation des Designs

Xmake plaziert das Design auf dem FPGA. Dabei verteilt es die einzelnen Komponenten des Schaltplans auf die Logikblöcke des FPGAs. Außerdem werden die Verbindungen zwischen den einzelnen Logikblöcken erstellt. Sind alle logischen Funktionen der Schaltung plaziert und die Verbindungen erstellt, wird ein LCA (Logic Cell Array)-File erzeugt. Dieses File beschreibt nun die Plazierung und Verdrahtung auf dem FPGA. Es kann in einen seriellen Datenstrom umgewandelt werden, mit dem das FPGA programmiert werden kann. Außerdem wird es für die zeitliche Simulation gebraucht, da daraus die Timing- Informationen der plazierten Schaltung gewonnen werden können.

3.4.4 Die zeitliche Simulation

3.4.4.1 TIMSIM8

Dieses Script bereitet die zeitliche Simulation vor. Dazu verwendet es das von Xmake erstellte LCA-File, in dem die Routing Information enthalten ist. Um das Design für die zeitliche Simulation vorzubereiten, ruft das Script die Programme LCA2XNF, XNFBA, PLD_DVE_BA, Gen_Sch8 und PLD_DVE_SIM auf.
Abbildung 3.6 zeigt den Ablauf bei der Vorbereitung zur zeitlichen Simulation. Begonnen wird dabei mit dem von Xmake erzeugten LCA-File. Danach fügt XDELAY dem LCA-File die Signallaufzeiten hinzu. Dieses LCA-File wird von LCA2XNF in ein XNF-File zurücktransformiert. Beim ersten Durchlauf dieser Simulationsvorbereitung ist es nicht notwendig, XNFBA zu starten. Dies ist nur dann notwendig, wenn sich Timing-Probleme bei der Simulation ergeben. XNFBA liest dieses XNF-File und das von XNFMerge bei der funktionellen Simulation erzeugte File und kombiniert diese beiden Files. Das XNF-File wird von Gen_sch8 in ein Schematic für die Simulation umgesetzt. Der PLD_DVE_SIM schließlich erzeugt einen neuen Viewpoint mit den Signallaufzeiten. Danach kann die Simulation beginnen.

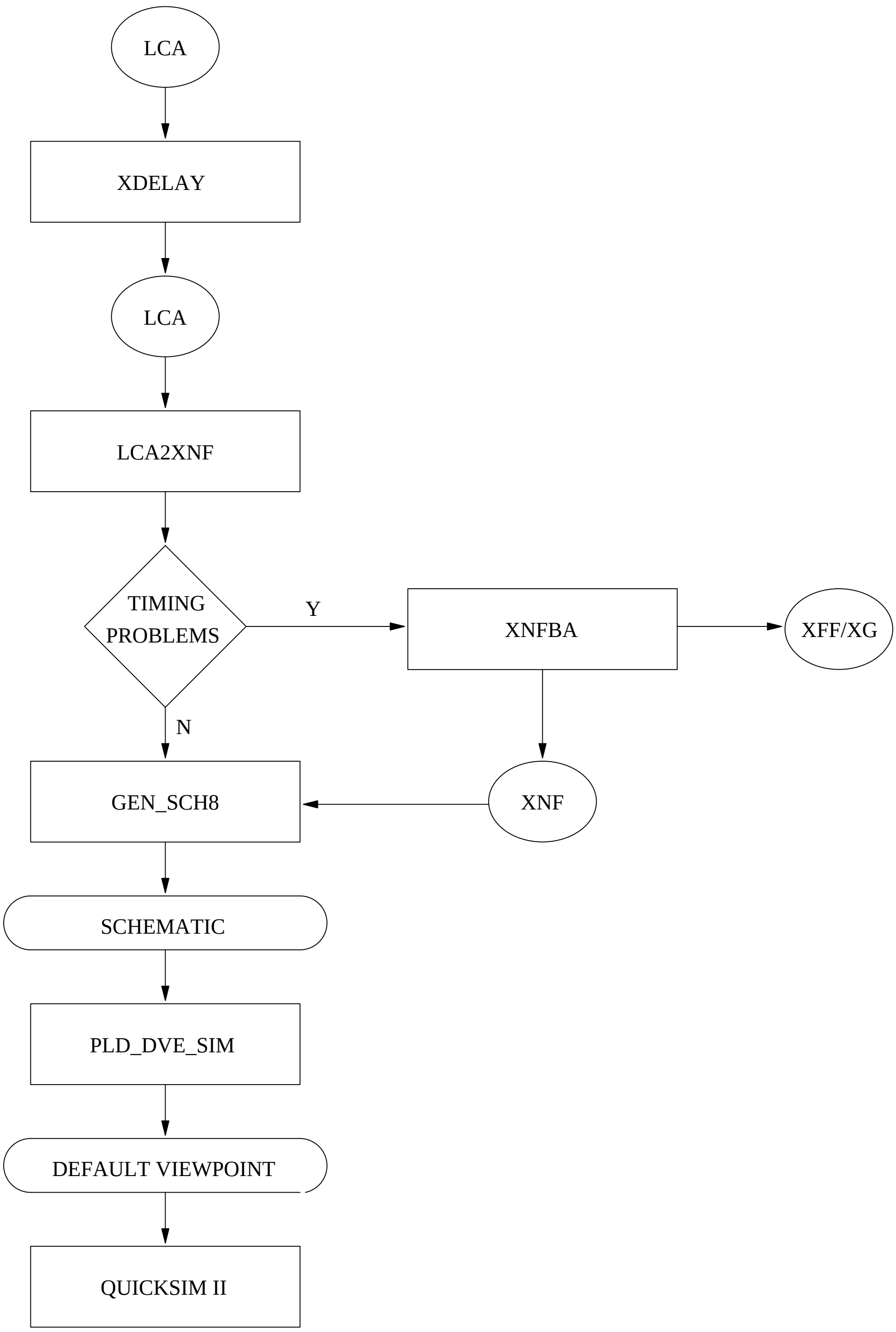


Abbildung 3.6: Der Design Flow bei der Vorbereitung der zeitlichen Simulation [7]

Kapitel 4

4 Telemetrierate und Prozessorbelastung

4.1 Ereignistypen

Die Ereignisdaten lassen sich, nachdem sie von der DPE verarbeitet wurden, in 5 Typen einteilen. Eine Zeitinformation für die einzelnen Ereignisse wird nicht benötigt, da die DPE die Daten zu Paketen variabler Länge zusammenfaßt, in deren Header eine Zeitinformation eingetragen wird. Die Daten werden dabei in der Reihenfolge ihres Eintreffens in diese Pakete geschrieben.

Typ 1: CdTe-Ereignisse im Energiebereich zwischen 15 und 600 keV
2 Bit Identifikation (ID)
14 Bit Pixel Adresse
11 Bit Energie
8 Bit Anstiegszeit

Typ 2: CdTe-Kalibrations-Ereignisse
siehe Type 1

Typ 3: zeitgleiche CdTe und CsI Ereignisse
2 Bit ID
14 Bit Pixel Adresse CdTe
8 Bit Energie CdTe
12 Bit Pixel Adresse CsI
8 Bit Energie CsI

Typ 4: nur CsI-Ereignisse
12 Bit Pixel Adresse
8/9 Bit Energie
1 Mehrfach-Ereignis-Flag

Typ 5: CsI-Kalibrations-Ereignisse
12 Bit Pixel Adresse
5 Bit Energie
1 Bit Line Flag

4.1.1 Benötigte Telemetrierate

Für den Imager steht eine Telemetrierate von 17-57 kBit/s zur Datenübertragung zur Verfügung [8]. Für die maximal erwartete Rate von Ereignissen, ergibt sich eine benötigte Telemetrierate von ungefähr 400 kBit/s. Dadurch ist es nicht möglich, alle Daten, die ein Ereignis beschreiben, zur Erde zu übertragen. Es muß eine Datenreduktion an Bord durchgeführt werden.
Die Tabelle 4.1 schlüsselt die einzelnen Ereignistypen und die erwartete Datenrate noch einmal auf.

Ereignis	Bits/s	Hintergrund-Zählrate (c/s)		Telemetrierate (kBits)	
		Solar max	Solar min	Solar max	Solar min
Typ 1	35	675	975	23.6	34.1
Typ 2	35	20	20	0.7	0.7
Typ 3	44	300	600	13.2	26.4
Typ 4	21/22	7500	15000	158/165	315/330
Typ 5	18	100	100	1.8	1.8
Total	--	8595	16695	197.4/204.4	378/393

Tabelle 4.1 : Hintergrund-Zählraten und anfallende Datenmenge für minimalen Hintergrund (Solares Maximum) und maximalen Hintergrund (Solares Minimum) [9]

4.1.2 Möglichkeiten zur Datenreduktion

Die Möglichkeiten zur Datenreduktion sind hier schlagwortartig aufgelistet und werden in Kapitel 6 genau besprochen.

- Suche nach Zeitkoinzidenzen zwischen Ereignissen aus den beiden Detektorschichten
- Erstellung von Histogrammen für die CsI-Ereignisse
- Rekonstruktion der Mehrfach-CsI-Ereignisse und Bestimmung des Ursprungspixels
- Anwendung eines Algorithmus um Compton Ereignisse bei den Zeitkoinzidenzen zur registrieren und offensichtliche Hintergrund-Ereignisse auszufiltern

Werden die Daten auf diese Weise bearbeitet, so kann die benötigte Telemetrierate in den zur Verfügung stehenden Bereich gesenkt werden.

4.2 Die Prozessorauslastung

Der in der DPE verwendete Prozessor besitzt eine Verarbeitungskapazität von ungefähr 1 Mips (10^6 Instruktionen pro Sekunde)
Außer den Aufgaben zur Datenreduktion hat die DPE noch weitere Aufgaben zu erledigen; dies sind:

- Sammeln der Housekeeping Daten
- Erzeugung von Datenpaketen für die Telemetrie
- Datenkomprimierung, sofern möglich
- Unterstützung verschiedener Betriebsarten
- Verteilung der Kommandos an das Instrument
- Energiekorrektur der CdTe-Ereignisse

- Ratemeter (siehe Kapitel 6) für die Detektoren

Die folgende Auflistung zeigt eine Abschätzung der benötigten Prozessorleistung der DPE für die einzelnen Aufgaben bei der Datenreduktion. Die Summe der benötigten Prozessorleistung liegt im ungünstigen Fall des Solaren Minimums bei ungefähr 3 Mips. Damit überschreitet sie deutlich die zur Verfügung stehende Prozessorkapazität.

	Zählrate	Mips
Rekonstruktion von Mehrfach-Ereignissen		
Einfach-Ereignisse	3600/7200	0.19/0.37
Zweifach-Ereignisse	5200/10400	0.29/0.58
Dreifach-Ereignisse	3900/7800	0.22/0.44
Akkumulation aller CsI-Ereignisse in Histogrammen	7500/15000	0.33/0.66
Energie-Korrektur der CdTe-Ereignisse	975/1575	0.02/0.03
Compton-Algorithmus	300/900	0.44
Zeitkoinzidenz	600	0.31/0.62

Tabelle 4.2: Aufgaben der DPE bei der Datenreduktion und die hierfür benötigte Prozessorleistung für minimalen und maximalen Hintergrund

4.3 Zusätzliche Hardware

Da die DPE nicht alle an sie gestellten Aufgaben erfüllen kann, müssen einige Funktionen von einer hartverdrahteten Logik übernommen werden. Dazu wird eine zusätzliche Karte, die HEPI (Hardware Event Processing Interface), benötigt. Sie wird in den freien Kartenschlitz der DPE eingebaut und von dieser mit +5 V versorgt. Die HEPI übernimmt anstelle der DPE die Suche nach Zeitkoinzidenzen und die Rekonstruktion der Mehrfach-Ereignisse. Außerdem erzeugt sie die Ratemeter für den CsI-Detektor und speichert die CsI-Ereignisse in Histogrammen. Diese Aufgaben lassen sich durch eine relativ einfache Hardware Logik bearbeiten. Aufgaben, die kompliziertere Abläufe und Berechnungen erfordern, werden von der DPE erledigt.
In Abbildung 4.1 sind die Verbindungen dieses Boards zu den Detektoren und der DPE dargestellt. Von der DPE bekommt die HEPI die Versorgungsspannung, den Takt und die Synchronisationsimpulse.
Die in den Detektoren registrierten Ereignisse, werden von der Detektorelektronik digitalisiert und über ein serielles High Bit Rate Interface (HBR I/F) mit 4 MBit/s Übertragungsrate zur HEPI übertragen. Zusätzlich führen von der HEPI noch jeweils zwei Leitungen zu den Detektoren. Über diese Leitungen werden das Clear_Fifo-Signal und das Resetsignal für die internen Zeitzähler an die Detektoren geleitet.
Für die Datenübertragung von der HEPI zur DPE findet ebenfalls ein HBR I/F Verwendung. Zusätzlich gibt es noch ein langsames, serielles Interface (LBR I/F). Es dient zur Übertragung der Steuerkommandos von der DPE zur HEPI. In umgekehrter Richtung werden die Statusinformationen der HEPI übertragen. Außer diesen zwei Interfaces verläuft noch eine Statusleitung zur DPE, die Auskunft über einen

Bursttrigger gibt (siehe 6.2.6). Von der DPE werden die Detektordaten zum Datenbus des Satelliten übertragen und danach zur Erde telemetriert.
Die DPE steht mit den Detektoren und der Veto- und Calibration Box (VEB) ebenfalls über drei langsame Interfaces in Verbindung. Außerdem verteilt die DPE einen 4.194 MHz Takt, der synchron zur „An Bord Zeit“ des Satelliten ist.

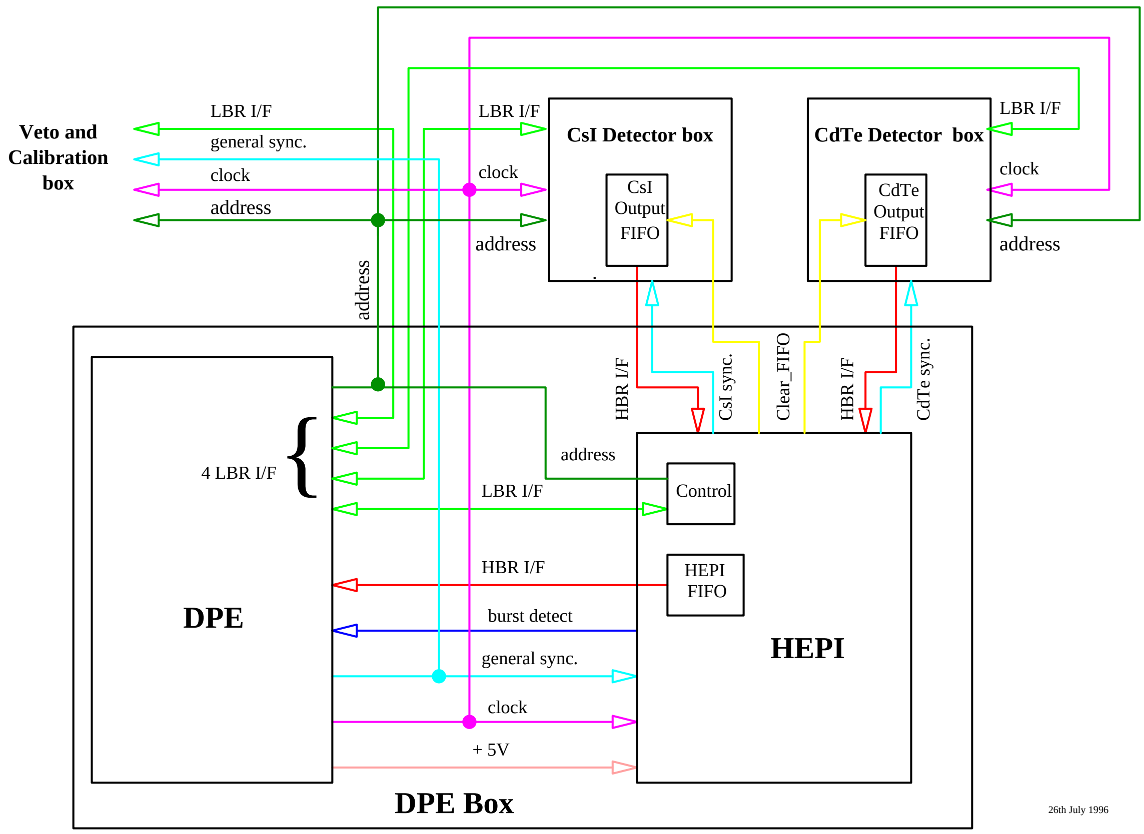


Abbildung 4.1: Interfaces der HEPI mit DPE, Detektoren und VEB [10]

4.3.1 Technische Daten

Die Abmessungen der HEPI betragen 215x220x18 mm^3. Die für die HEPI benötigte Leistung inklusive einer Reserve beträgt für die maximal mögliche Menge an Speicher 5 Watt.
Die Bausteine auf der HEPI teilen sich in drei Gruppen auf. Der Baustein, welcher die Logik enthält, die RAM Bausteine und die RS 422 Treiber für die seriellen Leitungen. Auf der HEPI, die im Satelliten eingesetzt wird, kommt ein ASIC zum Einsatz. Er enthält die Logikschaltung der HEPI. Die Speicherbausteine, die dort zum Einsatz kommen, sind 128 k SRAM’s des Typs 190A325 von der Firma LORAL. Alle Bausteine müssen für das Flugmodell strahlungsfest sein.
Bevor mit der Fertigung des ASICs begonnen werden kann, muß die Funktion der HEPI mit den Detektoren und der DPE überprüft werden. Dazu wird die Logik nicht in einem ASIC sondern in einem FPGA untergebracht. Für das Entwicklungsmodell wird ein FPGA der Firma Xilinx verwendet. Der FPGA ist durch seine Programmierbarkeit ideal

für die Entwicklungsphase geeignet. Die Speicherbausteine für das Entwicklungsmodell sollten vom gleichen Typ sein, es wird jedoch keine Strahlungsfestigkeit verlangt.

Kapitel 5

5 Die Sortierer

Wie in Kapitel 4 dargelegt, ist die Suche nach Zeitkoinzidenzen eine wichtige Aufgabe bei der Datenreduktion. Dabei ist es von Vorteil, wenn die Daten aus jeder Detektorschicht nach der Zeit geordnet übertragen werden. Durch diese zeitliche Sortierung kann der Aufwand bei der Suche nach Zeitkoinzidenzen reduziert werden.
Die Daten sind in den FIFOs der einzelnen Detektormodule zeitlich geordnet. Zwischen den FIFOs besteht keine zeitliche Ordnung, da die einzelnen Module voneinander unabhängig sind. Mit dem Auslesen der Daten aus den FIFOs geht somit die zeitliche Ordnung verloren.
Wird nun nach Zeitkoinzidenzen gesucht, müssen für jedes Ereignis aus der einen Detektorschicht die acht Module der anderen Schicht durchsucht werden. Das ergibt insgesamt 8x8 Kombinationen [11]. Sind jedoch alle Daten einer Detektorschicht zeitlich geordnet, so ist im Durchschnitt nur ein Ereignis aus dem CdTe-Detektor mit fünfzehn Ereignissen aus dem CsI-Detektor zu vergleichen, da das Verhältnis der CdTe-Ereignisse zur Anzahl der CsI-Ereignisse ungefähr 1:15 beträgt. Der Rechenaufwand wird damit deutlich gesenkt.
Für jede Detektorschicht benötigt man eine Einheit, die diese Sortierung nach der Zeit durchführt. Dazu wurden folgende Vorschläge gemacht.

5.1 Der Timesorter

Als dieser Vorschlag gemacht wurde, ging man noch davon aus, daß die DPE alle an Bord notwendigen Verarbeitungsschritte übernehmen kann [12]. Der Timesorter würde für jede Detektorschicht eine Elektronikkarte benötigen, in der die zeitliche Sortierung der Daten durchgeführt wird. Der Funktionsablauf des Timesorters ist in Abbildung 5.1 dargestellt. Die ersten Zeitdaten werden aus den acht FIFOs der Detektormodule ausgelesen. Innerhalb dieser FIFOs sind sie zeitlich geordnet, was eine notwendige Voraussetzung darstellt, ohne die der Timesorter nicht funktionieren würde. Abbildung 5.1 zeigt, wie die Ereignisdaten in den 16 Bit breiten FIFOs abgelegt werden. Dieses Datenformat gilt nur für den Timesorter und wurde für das spätere Konzept abgeändert. An erster Stelle steht das Reset-Flag und das Detektormodul (MDL) aus dem die Daten stammen. Anschließend ist das Zeitwort des Ereignisses aufgeführt. Das bei der Simulation verwendete Datenformat des Zeitwortes hat eine Größe von 16 Bit, wobei Bit 0-11 die Zeitinformation beinhaltet, Bit 12-14 das Detektormodul und Bit 16 das Reset-Flag. Nach einem Zeitzyklus wird immer ein bestimmtes Bitmuster in die FIFOs geschrieben. Bei diesem Bitmuster ist das Reset-Flag auf eins gesetzt und alle anderen Bits haben den Wert null (x8000 in hexadezimaler Darstellung). Es ist in der Abbildung 5.1 als RESET in den FIFOs eingetragen. Ein Zeitzyklus hat eine Länge von 2 s. Nach der Modul- und Zeitinformation folgen die Energieinformation und die Pixeladresse des Ereignisses.

Nach dem Einlesen werden die Daten in der Sortierstufe nach der Zeit geordnet, wobei das Ereignis mit der kleinsten Zeit, das heißt das älteste, bestimmt wird. Diese Zeitdaten werden in ein Datenregister ausgelesen. Danach werden die Energie und Positionsdaten aus dem FIFO, aus dem die Zeitinformation stammt, ausgelesen. Aus den Datenregistern werden die Informationen an eine serielle Schnittstelle geschickt, die sie zur DPE weitertransportiert. Wird an einem FIFO-Ausgang ein RESET detektiert, so werden aus diesem FIFO keine Daten mehr gelesen. Der ganze Vorgang wiederholt sich so lange, bis an allen FIFO-Ausgängen ein RESET Signal anliegt. Danach beginnt man wieder mit dem Einlesen der ersten Daten aus den acht FIFOs. Die von mir entwickelte Schaltung dient dazu, dieses Prinzip der Sortierung zu verifizieren. Ich habe mich darauf beschränkt, die Zeitdaten zu sortieren und zu übertragen. Eine Erweiterung der Funktion auf die Übertragung der Energie und Positionsdaten wäre leicht möglich.
Der Timesorter besteht aus mehreren Komponenten. Seinen Aufbau zeigt Abbildung 5.2. Zum einen gibt es eine I/O-Einheit, deren Aufgabe darin besteht, die Daten aus den FIFOs auszulesen und an die Sortierstufe weiterzugeben. Die I/O-Einheit besteht aus einer parallel-zu-seriellen Schnittstelle und einer seriell-zu-parallelen Schnittstelle. Dies wurde von mir so ausgeführt, um die Simulation zu erleichtern. Die Daten können dadurch im Simulationsprogramm (Quicksim) in paralleler Form eingegeben werden und müssen nicht zuvor in einen seriellen Datenstrom umgewandelt werden. Es wäre natürlich möglich gewesen, die Daten direkt am Eingang der Sortierstufe anzulegen und auf die I/O-Einheit zu verzichten, allerdings wollte ich die komplette Schaltung aufbauen, um den gesamten Datenpfad zu simulieren.

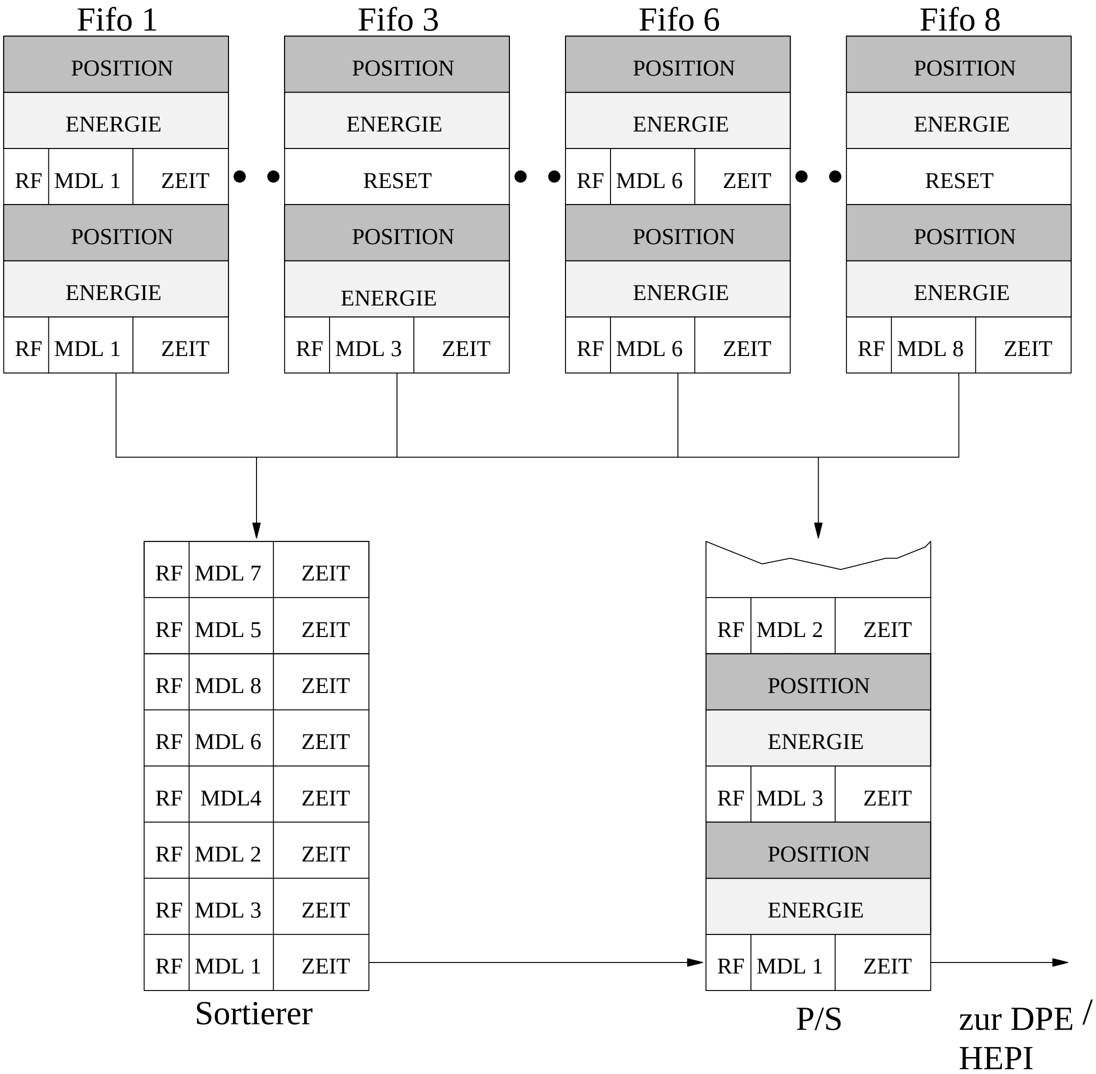


Abbildung 5.1: Der Ablauf im Timesorter

In die I/O-Einheit ist ein Multiplexer integriert, über den die Schnittstelle ausgewählt wird, deren Daten zum Eingang der Sortierstufe gelangen. In der Sortierstufe werden die Daten nach der Zeit sortiert und gespeichert. Aus der Sortierstufe werden die Daten dann in ein Register ausgelesen und können zur DPE weitergegeben werden.

In der Abbildung 5.2 wurde zusätzlich der von mir nicht realisierte Datenpfad dargestellt, der erforderlich ist, um auch die Energie- und Positionsdaten zu übertragen. Dazu ist zwischen der Sortierstufe und der Schnittstelle noch ein Multiplexer eingefügt. Über diesen Multiplexer werden die Zeitdaten aus der Sortierstufe zusammen mit ihren jeweiligen Energie- und Positionsdaten auf den Eingang der Schnittstelle gelegt. Der gesamte Ablauf wird durch die Steuerung, der letzten Komponente des Timesorters, kontrolliert.

Im folgenden möchte ich auf zwei Teile der Schaltung noch genauer eingehen. Zum einen auf die Sortierstufe und zum anderen auf die Steuerung.

5.1.1 Die Sortierstufe

Die Aufgabe der Sortierstufe ist es, das kleinste der acht Zeitworte zu finden. Die Sortierstufe wurde nach einer Schaltung gebaut, die auch im sogenannten Eventanalyser der pn-CCD Kamera für den XMM-Satelliten Verwendung findet. Sie wurde jedoch von mir etwas modifiziert und den Erfordernissen des Timesorters angepaßt. Bei XMM ordnet die Sortierstufe 64 Datenworte nach der Größe in aufsteigender Reihenfolge. Danach wird der 32. Wert ausgelesen, ein Reset durchgeführt und 64 neue Werte eingelesen. Die Sortierstufe des Timesorters muß nur 8 Worte sortieren. Diese werden der Größe nach absteigend sortiert. Außerdem verläuft der Sortiervorgang etwas anders ab.

Abbildung 5.3 zeigt den Sortiervorgang. Ganz links stehen die Daten, die in die Sortierstufe eingelesen werden. Der Pfeil deutet dabei auf den Platz an dem sie einsortiert werden. In der dritten Spalte ist die Reihenfolge nach der Sortierung dargestellt. Die Sortierstufe besteht aus acht Untereinheiten, wobei jede dieser Untereinheiten ein Register enthält, das die Ereignisse speichert. Die Sortierstufe funktioniert nach folgendem Prinzip. Die Daten werden von der I/O-Einheit in die Sortierstufe eingelesen. Danach wird in jeder der acht Untereinheiten geprüft, ob die eingelesenen Daten größer, kleiner oder gleich den Daten in den Untereinheiten sind. Sind die Daten größer, geschieht nichts. Sind sie kleiner oder gleich, werden sie in das Register eingelesen, an dem dies der Fall ist. Die restlichen Daten wandern eine Stelle nach oben. Nachdem man die Daten aus den acht FIFOs eingelesen hat, erhält man in der obersten Untereinheit der Sortierstufe die Daten mit der kleinsten Größe, das heißt das älteste Ereignis. Dieses Ereignis wird ausgelesen und anschließend werden die Daten des nächsten Ereignis eingelesen. Durch die zeitliche Ordnung der Daten in den FIFOs der Detektormodule, ist das neue Datenwort größer, als die Daten in der obersten Untereinheit. Sie werden somit aus der Sortierstufe hinausgeschoben.

Die Abbildung 5.3 a) zeigt, wie der erste Wert in die Sortierstufe eingelesen wird. In Abbildung 5.3 b) sind schon Daten in der Sortierstufe vorhanden, und die neuen Daten werden in die korrekte Untereinheit eingelesen. Abbildung 5.3 c) schließlich zeigt die Sortierstufe, nachdem acht Werte eingelesen wurden. Der oberste Wert wurde schon weitergegeben, steht jedoch noch im Register. Die neuen Daten werden an ihrer Position eingefügt und das älteste Datum wird aus der Sortierstufe geschoben.

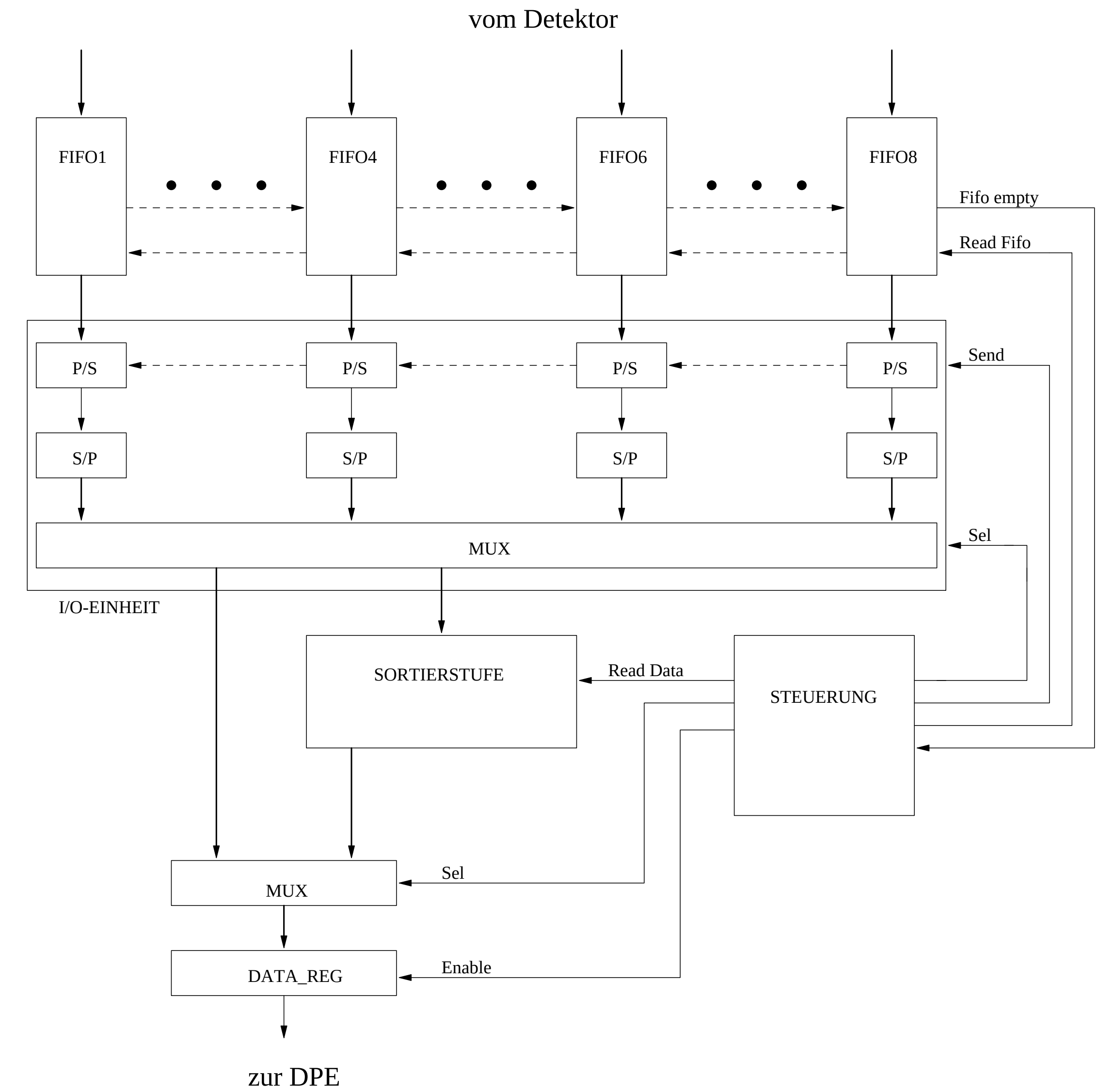


Abbildung 5.2: Blockschaltbild des Timesorters

5.1.2 Die Steuerung

Die Steuerung kontrolliert den gesamten Ablauf im Timesorter. Auf ein Startsignal folgend, wird ein Auslesesignal an das erste FIFO gegeben. Sind in diesem FIFO Daten vorhanden, so wird das ersten im FIFO gespeicherte Ereignis eingelesen.

Ist das FIFO leer, so wird diese Information in einer Untereinheit der Steuerung gespeichert. Gleichzeitig wird der Zeiger, der auf die Register der Sortierstufe zeigt, in dem die Daten mit der kleinsten Zeit gespeichert sind, um eine Position zurückgesetzt. Dies ist notwendig, da nach dem Einlesen der ersten Ereignisdaten aus den acht Modul-FIFOs, nur sieben Positionen mit validen Daten belegt sind. Da die Daten von unten in die Sortierstufe geschoben werden, muß dann die zweite Position, das heißt das Register 2, ausgelesen werden.

Ist das Einlesen aus dem ersten FIFO beendet, so wird ein Signal zum zweiten FIFO geschickt, um die nächsten Daten einzulesen. Auch hier wird der Zeiger wieder um eine Position zurückgesetzt, sofern keine Daten in diesem FIFO vorhanden sind. Nachdem alle acht FIFOs ausgelesen wurden, wird abgefragt, ob in den zuvor leeren FIFOs

inzwischen Daten vorhanden sind. Ist dies der Fall, so werden diese anschließend in die Sortierstufe gelesen, und der Zeiger für jedes eingelesene Ereignis um eine Position nach vorne gesetzt.
Danach werden die Daten mit der kleinsten Zeit aus der Sortierstufe ausgelesen und in ein Datenregister geschrieben. Von dort aus können sie an eine serielle Schnittstelle weitergegeben werden.
Anschließend werden aus dem FIFO, aus dem das kleinste Zeitwort stammt, die nächsten Daten in die Sortierstufe gelesen. Dort werden sie wieder sortiert und danach die ältesten Daten ins Datenregister geschrieben. Sind im FIFO keine Daten vorhanden, wird der Auslesezeiger ein Register zurückgesetzt. Danach wird abgefragt, ob in einem zuvor leeren FIFO inzwischen Daten vorhanden sind. Falls vorhanden, werden sie in die Sortierstufe eingelesen, das kleinste Ereignis bestimmt und ins Datenregister ausgelesen. Sind keine vorhanden, wird das älteste Ereignis aus der Sortierstufe zum Datenregister gelesen.
Wird von der Steuerung ein Resetsignal (x8000) detektiert, so wird aus diesem FIFO nichts mehr ausgelesen, bis von allen FIFOs ein Reset-Signal registriert wurde. Danach kommt das Clear-Signal, welches die Datenregister in der Sortierstufe zurücksetzt, und der gesamte Vorgang beginnt von neuem.

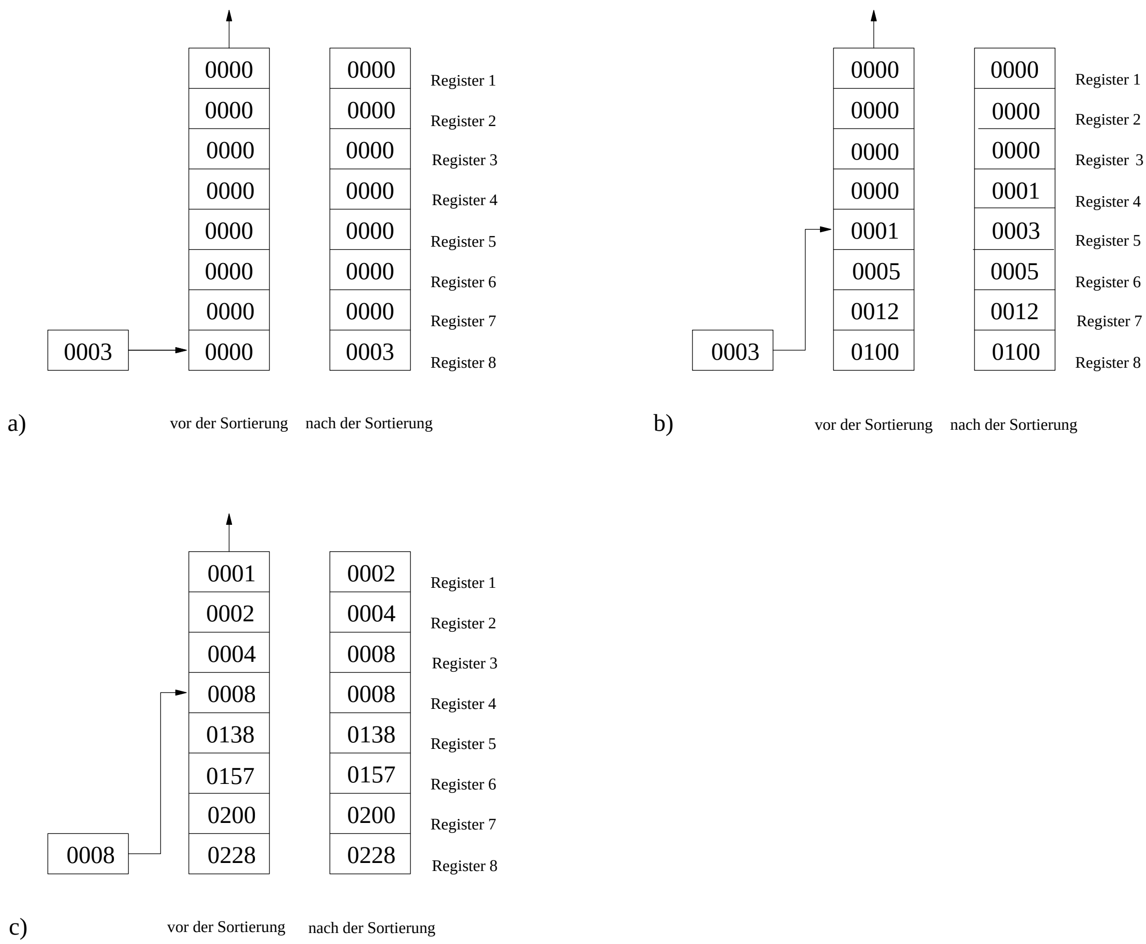


Abbildung 5.3: Der Funktionsablauf in der Sortierstufe

Abbildung 5.4 zeigt ein Blockschaltbild der Steuerung mit ihren vier Komponenten. Die Einheit COMMAND_I/O gibt die Lesesignale an die FIFOs und die Schnittstellen. Das

START-Signal setzt den Sortierablauf in Gang. Mit dem EMPTY-Signal wird der Einheit mitgeteilt, welche FIFOs Daten enthalten. Mit den Signalen SLICE_IN, FIFO_SEL und FIFO_READ wird ausgewählt, welches FIFO einen Lese-Impuls erhält. Der Ausgang READ_FIFO gibt ein Lese-Signal an die FIFOs weiter. WORD_SEND regelt die Übernahme der Daten in die Schnittstelle und setzt die Datenübertragung zur Sortierstufe in Gang. Ist das EMPTY-Flag für ein FIFO gesetzt, so werden das READ_FIFO und das WORD_SEND-Signal unterdrückt und es erfolgt keine Datenübertragung. Mit Hilfe von MUX_SEL wird der Multiplexer geschaltet, der die Ausgänge der Schnittstellen auf den Eingang der Sortierstufe legt. Das COMMAND-Modul ist die zentrale Einheit in der Steuerung. Dieses Modul empfängt Signale aus den übrigen Modulen und verschickt Steuersignale je nach Zustand dieser Komponenten. Es regelt den Ablauf in der Steuerung.

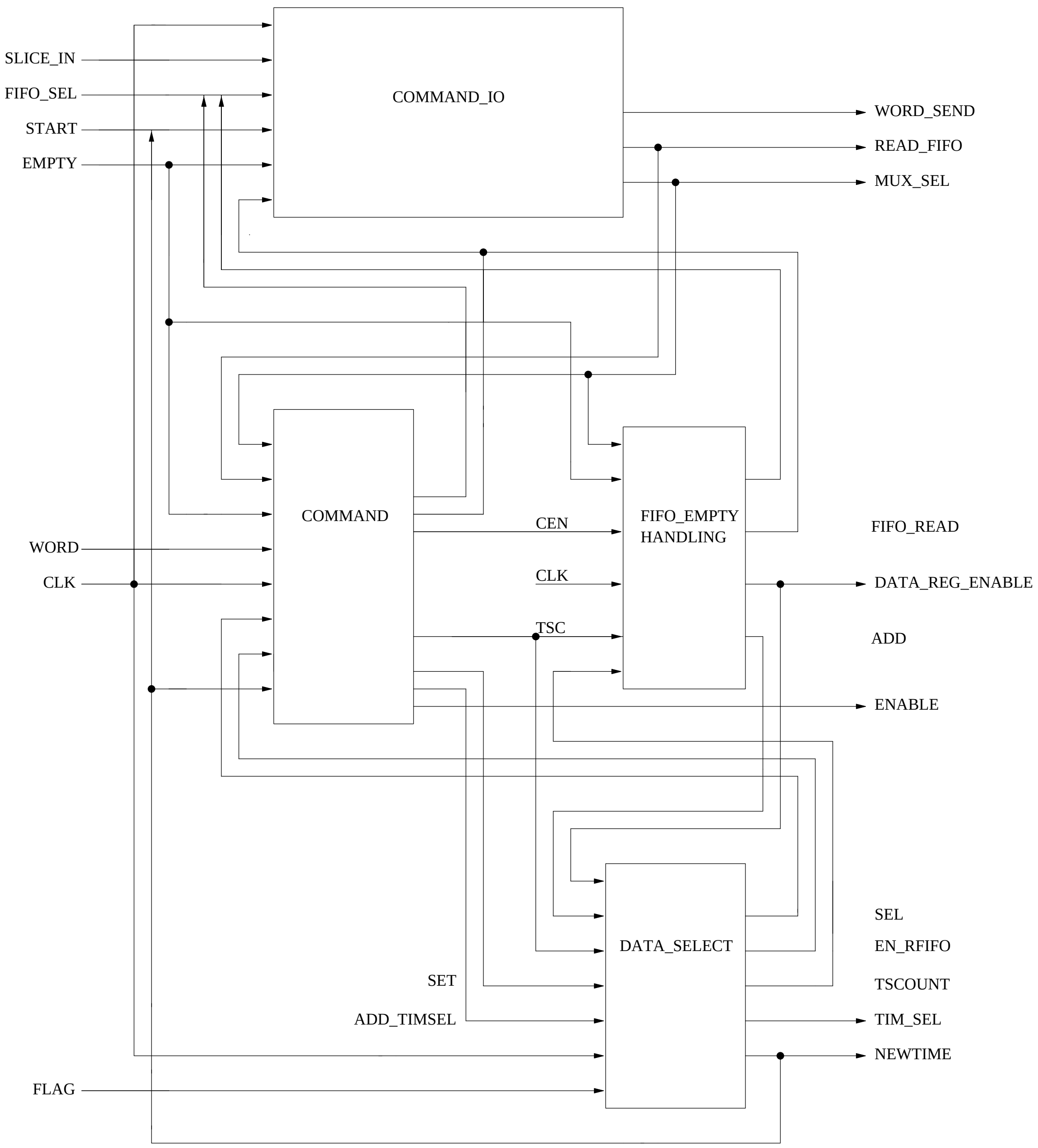


Abbildung 5.4: Blockschaltbild der Steuerung

In der FIFO_EMPTY_HANDLING Einheit wird festgehalten, ob und welches FIFO leer ist. Ist ein FIFO bei einem Lesevorgang ohne Daten, so schickt das COMMAND Modul ein Signal (TSC) an die FIFO_EMPTY_HANDLING Einheit.
Nach jedem Einlesevorgang wird geprüft, ob in vorher leeren FIFOs nun Daten vorhanden sind, und diese werden anschließend eingelesen. Dieser Ablauf wird mit dem CEN-Signal gestartet. Die vierte Einheit der Steuerung ist das Modul DATA_SELECT. Mit seiner Hilfe wird die Position in der Sortierstufe ausgewählt, aus der die Daten gelesen werden. Das Signal TIM_SEL gibt diese Position an die Sortierstufe weiter. Wurde an allen FIFOs ein Reset-Signal registriert, so löst das Modul einen Reset aus, so daß der neue Sortiervorgang beginnen kann.

5.1.3 Simulation des Timesorters

Die folgende Simulation wurde mit dem Simulationsprogramm Quicksim erstellt. In den Trace-Diagrammen sind die Eingangs- und Ausgangssignale, sowie weitere wichtige Signale dargestellt. In Abbildung 5.5 ist ein Teil der gesamten Simulation dargestellt. Die vollständige Simulation findet sich im Anhang.

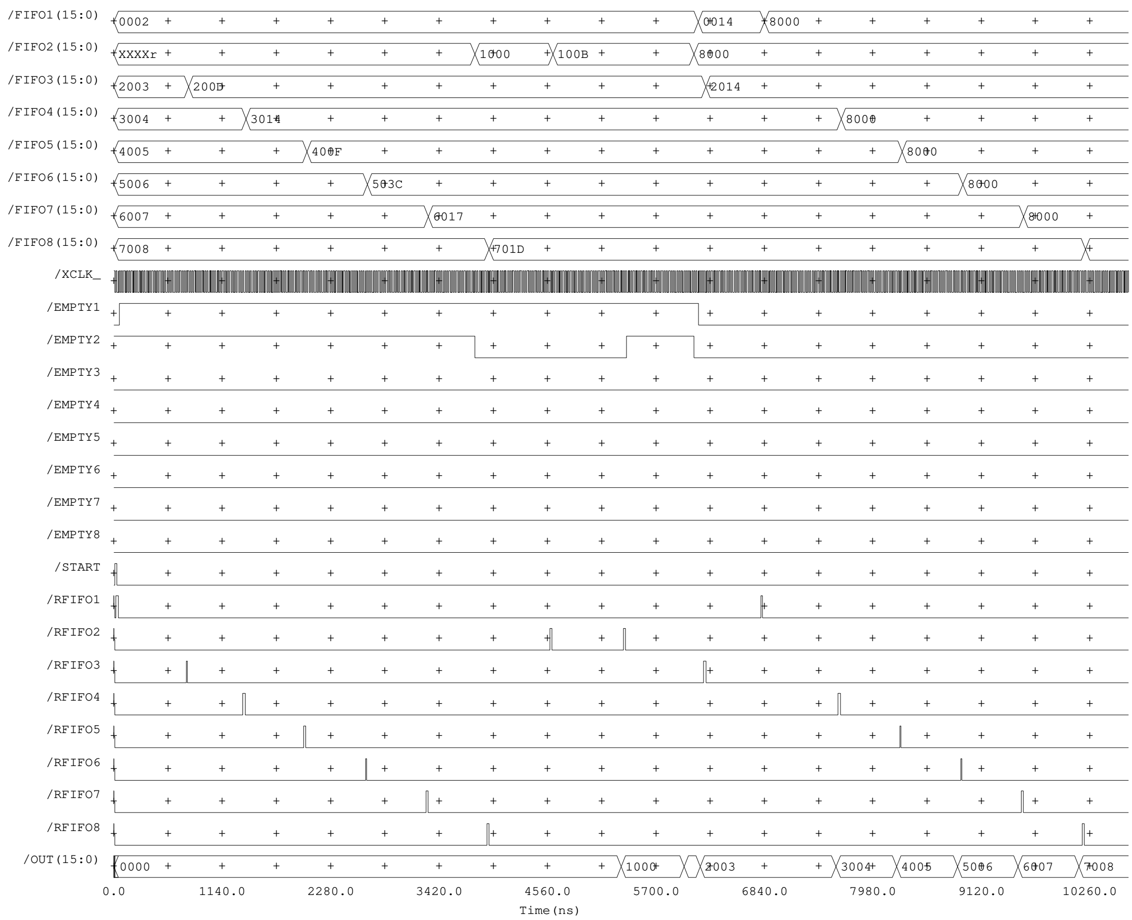


Abbildung 5.5: Simulation des Timesorter

Das in Abbildung 5.5 abgebildete Tracediagramm zeigt die Ausgänge der FIFOs (FIFO1 bis FIFO8), den Takt (XCLK_), die Read-FIFO (RFIFO) und die FIFO-Empty Signale

(EMPTY). Außerdem sind noch das START-Signal und der Datenausgang, der mit OUT(15:0) bezeichnet ist, dargestellt.

Zu Anfang der Simulation stehen an allen FIFOs außer dem FIFO2 Daten zur Verfügung. Daher ist das EMPTY2-Flag gesetzt. Die Daten sind in hexadezimaler Schreibweise angegeben. Nach dem Startsignal bekommen alle FIFOs, außer FIFO2 einen Read-FIFO Impuls, um ihre Daten zu senden. Während die Daten übertragen werden, kommen in FIFO2 Daten an und das EMPTY2-Flag wird zurückgesetzt. Die Steuerung überprüft dieses Flag und liest nun die Daten aus FIFO2 ein. Anschließend wird das Ereignis mit der kleinsten Zeit in das Ausgangsdatenregister geschrieben. In diesem Fall ist es das Ereignis aus FIFO2 mit dem Wert 000. Aus diesem FIFO wird nun das nächste Ereignis in die Sortierstufe gelesen. Das älteste Ereignis (kleinstes Zeitwort) wird wieder in das Ausgangsdatenregister geschrieben. Dieser Vorgang wiederholt sich fortlaufend. Wie in der Simulation zu sehen ist, ergibt sich so die korrekte zeitliche Reihenfolge.

5.2 Der FIFO-Sorter

Der zweite Vorschlag zur Sortierung der Daten ist in Abbildung 5.6 gezeigt. Für jedes Detektormodul gibt es ein FIFO in dem die Daten abgespeichert werden. Außerdem ist noch ein Adress-FIFO vorhanden. Es dient dazu die Adresse des FIFOs aufzunehmen in dem ein Ereignis abgespeichert wird.

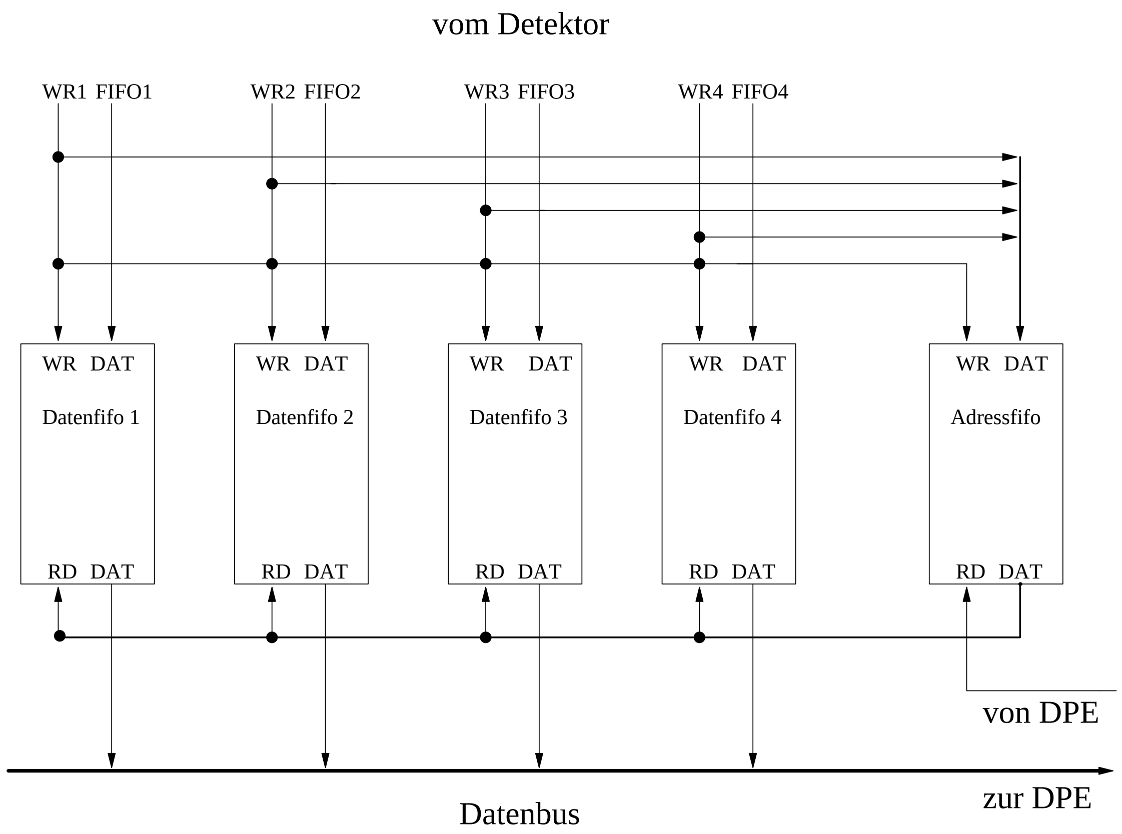


Abbildung 5.6: Blockschaltbild des FIFO-Sorters

Der Vorgang der Sortierung sieht folgendermaßen aus. Wird in einem Detektormodul ein Photon registriert, werden die Daten in das zugehörige FIFO geschrieben. Die Schreib-Leitungen der Daten-FIFOs (in Abbildung 5.6 mit WR bezeichnet) werden zu einem Bus zusammengefaßt. Jeder WR-Leitung ist dabei ein Bit des Datenbus zugeordnet. Bei einem WR-Signal wird nun das zugehörige Bit auf eins gesetzt. Dieser Bus führt zu dem Dateneingang des Adress-FIFOs. Gleichzeitig mit dem WR-Impuls der Daten-FIFOs bekommt das Adress-FIFO einen WR-Impuls, so daß es die Adresse des FIFOs, welches Daten empfangen hat, abspeichert. Dies geschieht für jedes Ereignis. Im Adress-FIFO werden die Adressen der Daten-FIFOs in der Reihenfolge gespeichert, in der sie Daten empfangen haben. Liest man nun das Adress-FIFO aus und gibt danach ein Schreib-Signal auf das FIFO, dessen Adresse man aus dem Adress-FIFO erhielt, so liest man die Daten wieder in der gleichen Reihenfolge aus, in der sie in die FIFOs geschrieben wurden. Die zeitliche Reihenfolge der Ereignisse, die beim Einschreiben in die FIFOs verloren ging, wird so wieder hergestellt.

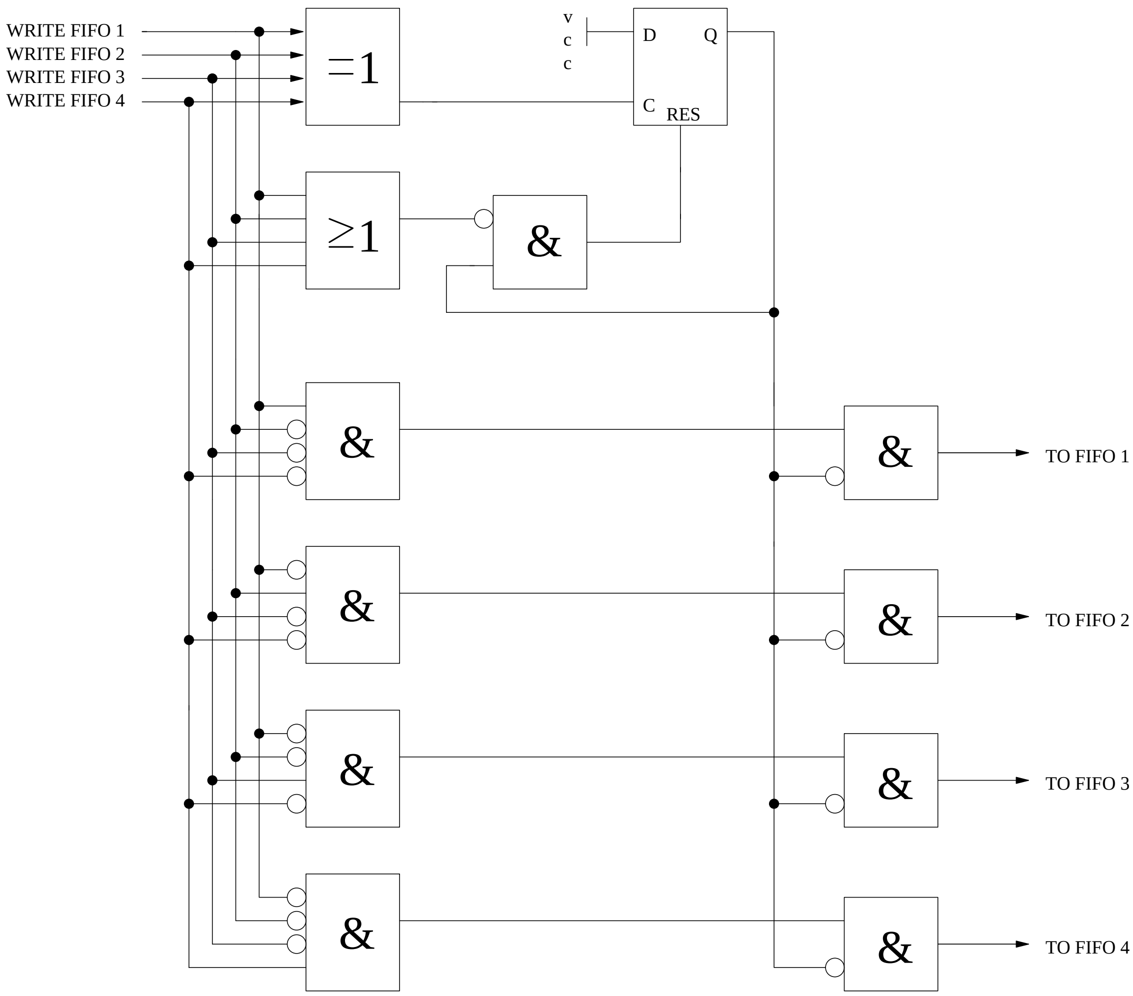


Abbildung 5.7: Blockschaltbild der Sperrlogik

Bei dieser Art der Sortierung ergibt sich allerdings folgendes Problem. Kommen zwei Ereignisse genau zeitgleich an, so werden beide Adressen im Adress-FIFO gespeichert, da beide Bits im Datenbus gesetzt werden. Beim Auslesen würden dann auch beide FIFOs gleichzeitig ausgelesen und ihre Daten auf die Datenleitung gegeben. Dies ist aber nicht erlaubt. Es gibt nun verschiedene Möglichkeiten, dieses Problem zu lösen. Zum einen könnte man die FIFOs beim Auslesen nacheinander abarbeiten. Das heißt, wenn zwei FIFO-Adressen im Adress-FIFO gespeichert sind, würde zuerst das eine, dann das andere FIFO ausgelesen. Die zweite Möglichkeit wäre, schon beim Einlesen zu verhindern, daß zwei FIFO-Adressen gleichzeitig gespeichert werden. Mit einer Logik, die jedem Daten-FIFO eine bestimmte Priorität zuordnet, läßt sich diese Möglichkeit realisieren. Bei zeitgleichem Eintreffen würde zuerst das Ereignis mit der höheren Priorität gespeichert und anschließend das Ereignis mit der niedrigeren Priorität. Die dritte Möglichkeit besteht darin, die Ereignisse, die zeitgleich ankommen, nicht in die FIFOs zu übernehmen. Diese zeitgleich ankommenden Ereignisse sind dann natürlich für die Weiterverarbeitung verloren. Ich habe mich bei meinem Aufbau für die dritte Möglichkeit entschieden.

Die von mir aufgebaute Schaltung besteht aus vier Daten-FIFOs und einem Adress-FIFO. Die Datenleitungen sind mit den Dateneingängen der FIFOs verbunden. Außerdem sind die Schreib-Leitungen zu einem Bus zusammengefaßt. Dieser Bus liegt am Dateneingang des Adress-FIFOs. Er liefert die Adresse des FIFOs, in welches Daten geschrieben wurden. Der Datenausgang des Adress-FIFOs liegt bitweise an den Lese-Eingängen der Daten-FIFOs (siehe Abbildung 5.6). Mit dem Lese-Impuls werden die Adressen aus dem Adress-FIFO ausgelesen und ergeben den Lese-Impuls für die Daten-FIFOs. Die Datenausgänge der Daten-FIFOs werden zu einem Datenbus zusammengefaßt. Dies ist möglich, da die Ausgänge der FIFOs im nicht aktivierten Zustand hochohmig sind. Es muß natürlich sichergestellt sein, daß immer nur ein FIFO gleichzeitig den Bus treibt.

Die verwendeten FIFOs haben einen Schreib-Lese-Zyklus, der im ungünstigsten Fall vier Takte beträgt [13]. Während dieser Zeit müssen die Daten stabil am Eingang anliegen, damit sie korrekt übernommen werden. Es muß gewährleistet sein, daß die Daten mindestens im Abstand von vier Takten ankommen oder daß Ereignisse unterdrückt werden, die während dieser Zeit ankommen. Ein gleichzeitiges Schreiben und Lesen ist jedoch möglich.

Um zu verhindern, daß zeitgleich ankommende Ereignisse eingelesen werden, wurde eine zusätzliche Logik eingebaut. Sie stellt außerdem sicher, daß während des Schreibzyklus eines FIFOs die Schreib-Eingänge der FIFOs gesperrt sind. Diese Sperrung hält solange an, bis alle Schreib-Signale wieder im Low Zustand sind.

Abbildung 5.7 zeigt diese Logik. Die Schreib-Leitungen liegen über UND-Gatter mit vier Eingängen, wobei drei Eingänge invertiert sind, an den Schreib-Eingängen der FIFOs. An den invertierten Eingängen der UND-Gatter liegen jeweils die Schreib-Leitungen der übrigen FIFOs. Das heißt, der Schreib-Impuls wird nur weitergegeben, wenn an keiner der anderen Schreib-Leitungen ein Impuls anliegt. Die Schreib-Leitungen verlaufen zusätzlich über ein vierfach XOR-Gatter zum Takt Eingang eines Flipflops. Liegt an mehr als einer Leitung ein Impuls an, so bekommt das Flipflop kein Takt-Signal. Sein Ausgang bleibt im Low Zustand. Liegt an nur einem Eingang ein Signal, so wird das Flipflop gesetzt und sperrt über die UND-Gatter die Schreib-Eingänge der Daten-FIFOs. Dabei wird die Gatterlaufzeit durch das Flipflop ausgenutzt. Durch die etwas längere Zeit, die das Signal durch das Flipflop benötigt, um am UND-Gatter anzukommen, wird noch ein kurzer Schreib-Impuls zum FIFO durchgelassen. Der Reset-Eingang des Flipflops ist über ein UND-Gatter mit einem invertierten Eingang, mit dem Ausgang des Flipflops und einem ODER-Gatter verbunden. An diesem ODER-Gatter liegen die Schreib-Leitungen der Daten-FIFOs. Damit ist sichergestellt, daß ein Reset nur durchgeführt wird, wenn das Flipflop gesetzt und alle Schreib-Eingänge im Low Zustand sind. Sind die Schreib-Impulse nun so gewählt, daß ihre Länge der Dauer eines Lesezyklus entspricht, so ist die korrekte Datenübernahme sichergestellt.

5.2.1 Simulation des FIFO-Sorters

In der in Abbildung 5.8 dargestellten Simulation sind die vier Dateneingänge der FIFOs zu sehen. Außerdem sind noch die Schreib-Eingänge (WR), der Lese-Eingang (RD) des Adress-FIFOs und der Datenausgang dargestellt. Bei der Simulation wurde kein spezielles Datenformat verwendet. Die eingegebenen Daten sind beliebige Datenworte, die in der Reihenfolge ihres Eintreffens sortiert werden. Am Dateneingang von FIFO1 liegen zum Zeitpunkt 1000 ns Daten an. Diese werden in das FIFO übernommen und die Adresse in das Adress-FIFO geschrieben. Der Schreib-Impuls steht so lange an, bis die Daten ins FIFO übernommen wurden. Während dieser Zeit sind die anderen FIFOs gesperrt. Die Daten, die zum Zeitpunkt 1250 ns am Eingang von FIFO2 anliegen, werden nicht ins FIFO übernommen. Erst nachdem alle Schreib-Eingänge der FIFOs wieder im Low Zustand sind, können neue Daten übernommen werden. Dies geschieht bei 2000 ns am FIFO1. Bei 3000 ns und 4000 ns kommt ein Lese-Impuls und legt die eingelesenen Daten auf den Datenausgang. Wie man sieht, wurden nur die Daten gespeichert, die von FIFO1 kamen. Die Daten an FIFO2 wurden unterdrückt, da sie während des Schreib-Impulses von FIFO1 ankamen. Die zeitliche Reihenfolge der Ereignisse wird korrekt wiedergegeben.

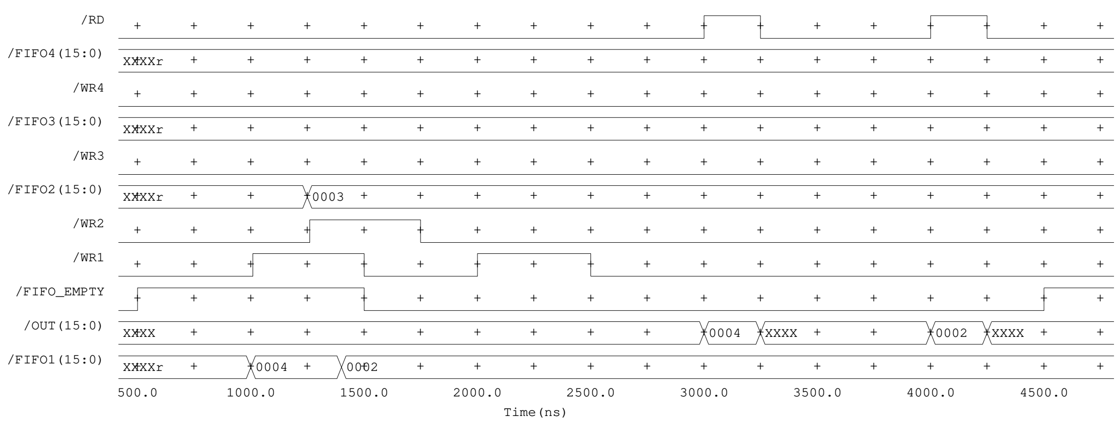


Abbildung 5.8: Simulation des FIFO-Sorters

Unter den Möglichkeiten zur Durchführung der Sortierung hat man sich schließlich für den FIFO-Sorter entschieden. Für ihn sprechen mehrere Punkte. Er ist deutlich einfacher aufgebaut als der Timesorter und benötigt keine zusätzliche Karte, sondern wird in die Detektorelektronik als FIFO Data Manager (siehe 1.4.6) integriert.

Kapitel 6

6 Die HEPI

Wie in Kapitel 4 gezeigt wurde, lassen sich bei der erwarteten Zählrate nicht alle Daten Photon für Photon übertragen. Da außerdem die DPE nicht über genügend Prozessorkapazität verfügt, um die Daten vorzuverarbeiten, wurde eine zusätzliche Karte notwendig. Diese HEPI übernimmt einige Aufgaben, die ursprünglich für die DPE vorgesehen waren.
Die in Kapitel 5 beschriebene zeitliche Sortierung aller Daten aus den Detektorschichten war ein erster Schritt, um die benötigte Verarbeitungskapazität zu reduzieren. Außerdem ist diese zeitliche Ordnung eine notwendige Voraussetzung für das von mir entwickelte Modul zur Suche nach Zeitkoinzidenzen.
Im folgenden Kapitel möchte ich nun ausführlicher auf die Aufgaben der HEPI und die von mir gewählte Realisierung dieser Aufgaben eingehen.

6.1 Die seriellen Schnittstellen

Zuerst möchte ich die Schnittstellen der HEPI kurz vorstellen. Es kommen zwei unterschiedliche Schnittstellen auf der HEPI zum Einsatz. Das ist zum einen die schnelle serielle Schnittstelle (High Bit Rate Interface) mit 4 MBit/s und zum anderen die langsame serielle Schnittstelle (Low Bit Rate Inteface) mit 64 kBit/s. Wie in Abbildung 6.2 dargestellt, dient die schnelle serielle Schnittstelle zur Übertragung der Daten von den Detektorschichten zur HEPI. Über die langsame serielle Schnittstelle werden die Kommandos von der DPE und die Status-Informationen der HEPI übertragen.

6.1.1 Das High Bit Rate Interface

Das High Bit Rate Interface (HBR I/F) ist ein unidirektionales synchrones Interface. Mit ihm werden die 64 Bit breiten Daten von den Detektoren zur HEPI übertragen. Es besteht aus vier Leitungen. Eine Leitung überträgt die Daten, die zweite den Sample-Impuls. Die beiden anderen übertragen das Taktsignal und das Fifo_not_Empty-Signal. Der Takt und der Sample-Impuls verlaufen dabei vom Empfänger zum Sender. Der Daten und Fifo_not_Empty-Impuls kommen vom Sender und verlaufen zum Empfänger.
Die Datenübertragung mittels des High Bit Rate Interface erfolgt in der in Abbildung 6.1 dargestellten Weise. Zur Übertragung eines Datenwortes schickt die HEPI einen Sample-Impuls zum Detektor. Der Sample-Impuls beginnt mit der abfallenden Flanke des Clock-Signals. Ist die Fifo_not_empty-Leitung gesetzt, so wird kein Sample-Impuls erzeugt. Das Fifo_not_empty-Signal ist Low aktiv, das heißt der aktive Zustand ist der Low Zustand. Bei der nächsten ansteigenden Taktflanke schreibt der Fifo Data Manager ein Datenwort in das Schieberegister des Senders. Sollte nun kein Datenwort im Ausgangs-

FIFO des Detektors mehr vorhanden sein, so geht das Fifo_not_empty-Signal in den Low Zustand. Mit der fallenden Flanke des Taktsignals übernimmt der Empfänger das Datenbit in sein Schieberegister. Dieser Vorgang wiederholt sich nun 64 mal. Danach wird wieder ein Sample-Impuls mit der abfallenden Flanke des 64. Taktpulses erzeugt, sofern das Fifo_not_empty Signal nicht aktiviert ist. Mit der nächsten ansteigenden Flanke des Taktsignals wird vom Empfänger das Datenwort ausgelesen. Gleichzeitig schreibt der Sender ein neues Datenwort in das Sender-Schieberegister.

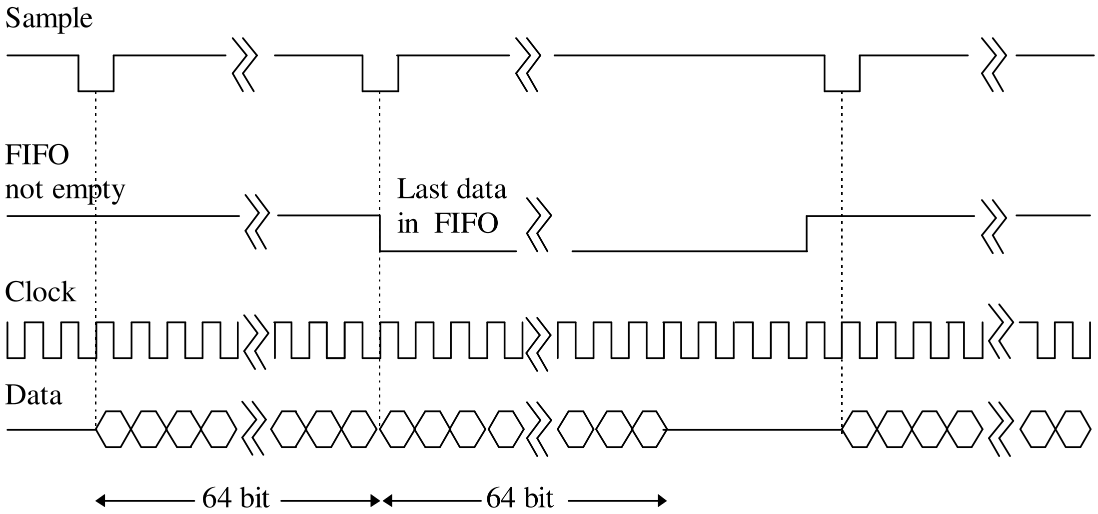


Abbildung 6.1: Der Signalverlauf in der Schnittstelle

6.1.1.1 Das Datenformat des HBR I/F

Von der Detektorseite werden die Ereignisdaten in vier Datenworten zur HEPI geschickt. Jedes Datenwort ist 16 Bit breit. Ein Ereignis wird also durch 64 Bit beschrieben. Die einzelnen Datenworte enthalten dabei folgende Information.

Datenwort 1

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Zeit 1. Byte								Zeit 2. Byte							

Datenwort 2

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Zeit 3. Byte								Rise Time (nur CdTe)							

Datenwort 3

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
CF					CdTe Amplitude										
CF	MP				CsI Amplitude										

Datenwort 4

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
		CdTe Adresse													
				CsI Adresse											

Das ersten Datenwort enthält die oberen 16 Bit der Zeitinformation. Das zweite Datenwort enthält die unteren 8 Bit der Zeit. Bei einer Taktfrequenz von 4.19 MHz beträgt die Größe einer Zeiteinheit 238.4 ns. Im zweiten Datenwort steht außerdem noch die 8 Bit Rise Time Information, die jedoch nur für die CdTe-Ereignisse verfügbar ist. Das dritte Datenwort enthält die Amplitude. Bei CdTe-Ereignissen sind dies 11 Bit und ein Bit für das Kalibrations-Flag. Das Kalibrations-Flag wird gesetzt, wenn ein Ereignis von der Kalibrations-Quelle stammt. CsI-Ereignisse haben nur 10 Bit Amplituden Information. Hier kommen noch zwei Flags hinzu, eines für die Kalibrations-Ereignisse und eines für die Mehrfach-Ereignisse. Im vierten Datenwort wird die Pixel Adresse gespeichert. Für ein CdTe-Ereignis ist die Adresse 14 Bit (16384 Pixel) groß. Die CsI-Adresse ist 12 Bit (4096 Pixel) groß.
Nachdem die Daten über die Schnittstellen zur HEPI übertragen wurden, werden sie dort in zwei Eingangsregistern gespeichert (siehe Abbildung 6.3). Danach durchlaufen die Daten die nachfolgend beschriebenen Module und gelangen dann in ein Ausgangs-FIFO, wovon sie dann zur DPE übertragen werden.

6.1.1.2 Das Low Bit Rate Interface

Das Low Bit Rate (LBR I/F) Interface umfaßt 6 Adressleitungen, eine Taktleitung und zwei Datenleitungen. Die eine Datenleitung überträgt die Daten von der DPE, die andere überträgt Daten zur DPE. Von den Adressleitungen werden nur vier benutzt. Mit diesen Leitungen erfolgt eine Auswahl der einzelnen Module. Jede Leitung verwendet einen RS 422 Treiber zur Übertragung des Signals.

6.2 Die Aufgaben der HEPI

Die HEPI (Hardware Event Processing Interface) hat, wie in Kapitel 4 gezeigt, verschiedene Aufgaben zu erfüllen. Die Abbildung 6.2 zeigt den schematischen Aufbau der HEPI mit ihren einzelnen Modulen, deren Aufgabe ich im folgenden näher erläutern will.

6.2.1 Die Suche nach Zeitkoinzidenzen

Eine Aufgabe ist die Suche nach Zeitkoinzidenzen zwischen den Ereignissen aus den beiden Detektorschichten. Diese Ereignisse werden zusammengefaßt und zur DPE geschickt. Dabei wird eine doppelte Zeitinformation eingespart, was zu einer Datenreduktion führt. Außerdem kann in der DPE auf diese Ereignisse ein Filteralgorithmus angewendet werden, der die Hintergrund-Ereignisse unterdrückt.

6.2.2 Die Rekonstruktion der Mehrfach-Ereignisse

In der HEPI werden die Mehrfach-CsI-Ereignisse rekonstruiert. Die Wahrscheinlichkeit, mit der ein Photon ein Mehrfach-Ereignis erzeugt, ist energieabhängig. Bei einer Photonenenergie von 500 keV liegt sie bei 16 % ein Zweifach-Ereignis zu erzeugen. Ein 2 MeV Photon erzeugt mit 8 % Wahrscheinlichkeit ein Zweifach-Ereignis. In den Tabellen 6.2 und 6.3 (siehe Seite 62) sind die Wahrscheinlichkeiten, ein Mehrfach-Ereignis zu erzeugen für verschiedene Energien aufgeführt.

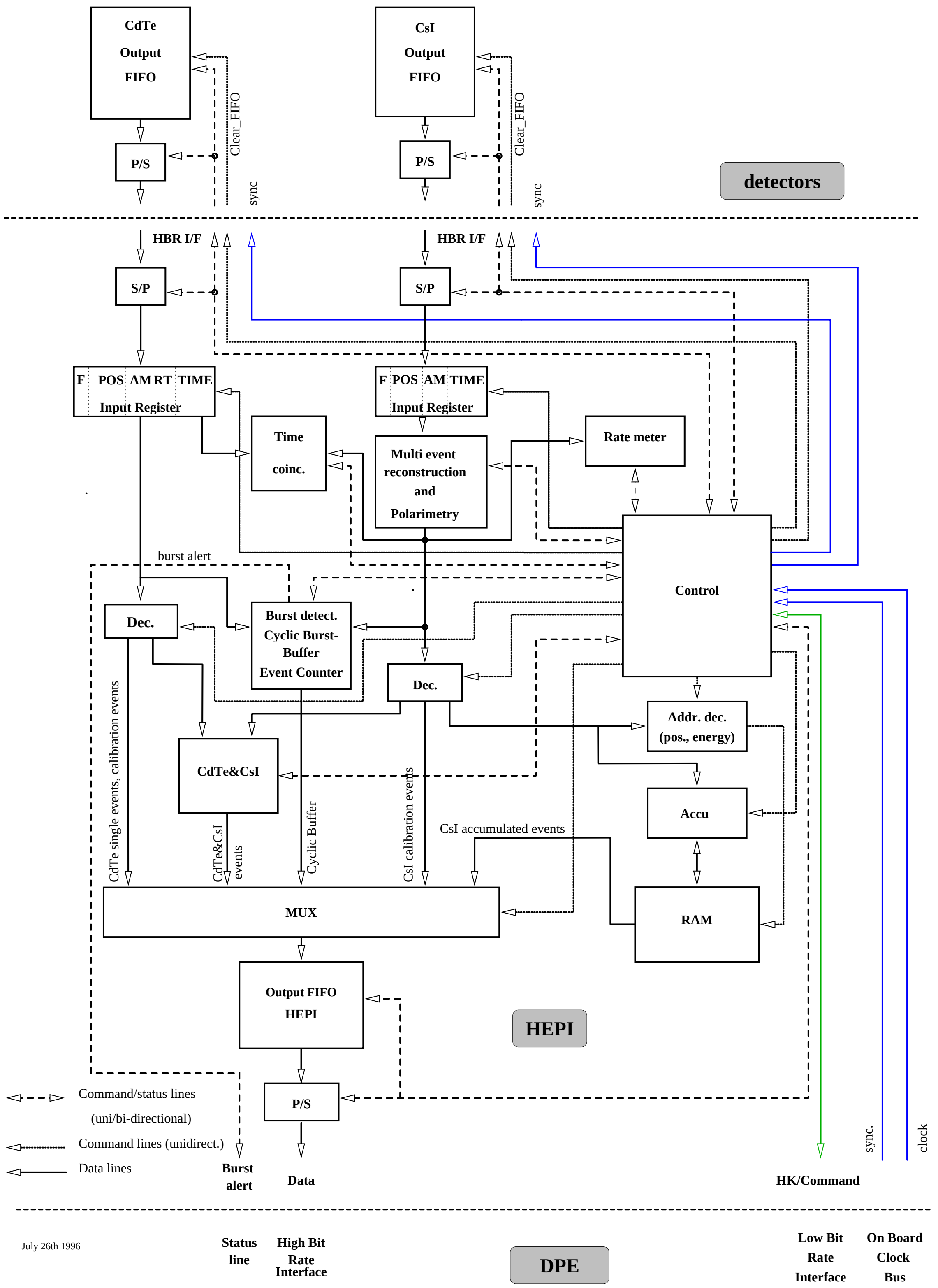


Abbildung 6.2: Schematischer Aufbau der HEPI [10]

Bei der Rekonstruktion werden die Amplituden der Ereignisse aufsummiert, die zu einem Mehrfach-Ereignis gehören. Diese Summation wird durchgeführt, um die ursprüngliche

Energie des Photons zu erhalten. Zusätzlich wird das Ursprungspixel rekonstruiert. Das ist jenes Pixel, in dem die erste Wechselwirkung des Photons mit dem CsI-Kristall stattgefunden hat.

6.2.3 Die Histogramme der CsI-Ereignisse

Da die CsI-Ereignisse aufgrund der hohen Zählraten nicht Photon für Photon übertragen werden können, sollen sie in Histogrammen abgespeichert werden. In den Histogrammen wird auf die Zeitinformation der Einzelereignisse verzichtet und die Energieauflösung wird von 10 auf 8 Bit reduziert. Durch diese Maßnahmen erreicht man bei einer Integrationszeit von 1000 sec eine Datenreduktion um ungefähr den Faktor 40. Es wird jeweils ein Histogramm für die Einfach-CsI-Ereignisse und für die Mehrfach-CsI-Ereignisse angelegt. Des weiteren wird ein Histogramm für die Kalibrations-Ereignisse angelegt. In diesem Histogramm werden alle Ereignisse gespeichert, deren Kalibrations-Flag gesetzt ist. Die Integrationsdauer für ein Histogramm liegt bei 1000 Sekunden. Dies entspricht der Dauer einer Pointierung des Spektrometers.

6.2.4 Die Ratemeter

Die Aufgabe der Ratemeter besteht darin, die Ereignisse, die pro Pixel auftreten, zu zählen. Mit Hilfe dieser Ratemeter können rauschende oder ausgefallene Pixel und Module erkannt werden. Diese Ratemeter werden in Form eines Histogramms angelegt. Dieses Histogramm umfaßt nur Pixeladressen und Anzahl der in einem Pixel registrierten einfach CsI-Ereignisse. In der HEPI werden die Ratemeter für die CsI-Ereignisse erstellt. Für die CdTe-Detektorschicht wird diese Aufgabe von der DPE übernommen.

6.2.5 Der Polarimetrie-Modus

Für die Zweifach-CsI-Ereignisse ist ein Polarimetrie-Modus geplant. Dabei werden die Zweifach-CsI-Ereignisse so verarbeitet, daß man Informationen über die Polarisation der einfallenden Gammastrahlung daraus gewinnen kann. Es wird die Tatsache ausgenutzt, daß die Comptonstreuung von der Polarisation abhängig ist.
Die HEPI bestimmt den Streuwinkel zwischen den Zweifach-Ereignissen. Sind die Streuwinkel der Ereignisse nicht mehr gleichverteilt, kann damit auf eine Polarisation geschlossen werden. Auch diese Polarisations-Ereignisse werden in Histogrammen abgespeichert. Im Polarimetrie-Modus wird der Streuwinkel in 3 Bit kodiert, womit sich acht Streuwinkel darstellen lassen. In Abbildung 6.3 ist dies noch einmal verdeutlicht. Nur die in erster Ordnung gestreuten Photonen werden berücksichtigt. Die in zweiter und höherer Ordnung gestreuten Photonen werden vernachlässigt. In erster Ordnung gibt es acht dem Urspungspixel benachbarte Pixel, so daß die Kodierung des Streuwinkels in 3 Bit ausreicht. Die Energieinformation wird durch die für den Streuwinkel benötigten 3 Bit auf 6 Bit beschränkt (siehe 4.1.2 Ereignistypen).
Der Polarimetrie-Modus wird immer nur alternativ zum Histogramm Modus der CsI-Ereignisse eingesetzt.

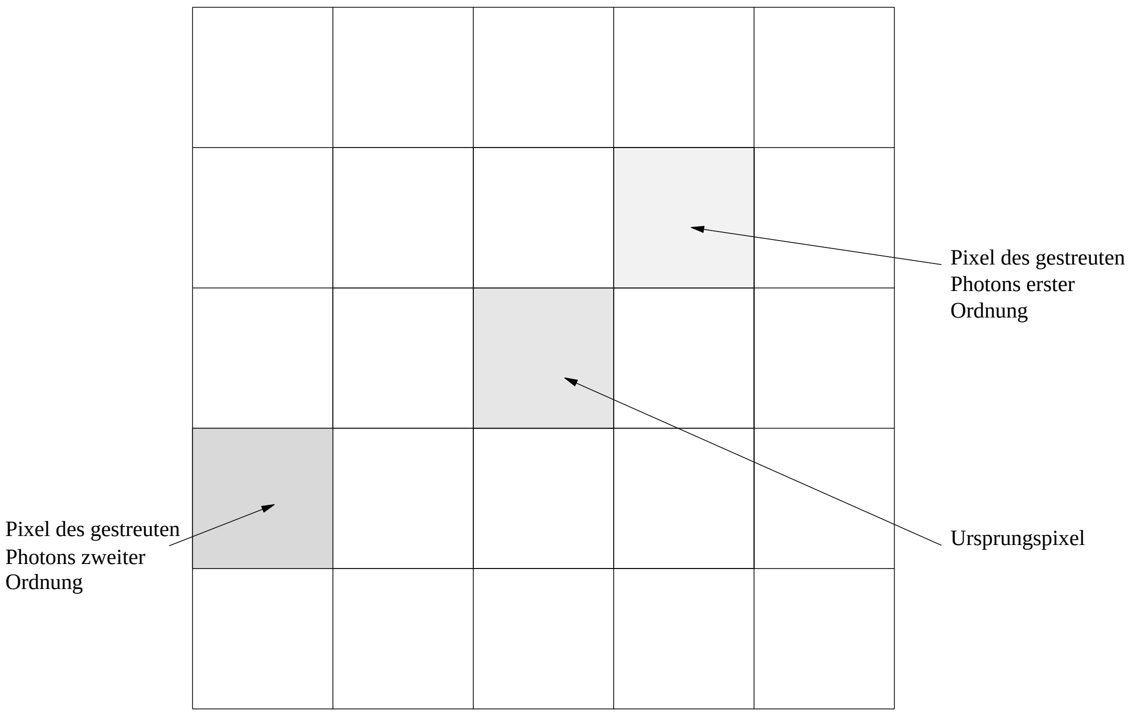


Abbildung 6.3: Streuung erster und zweiter Ordnung. Die Ordnung gibt die Entfernung des gestreuten Photons vom Ursprungspixel an.

6.2.6 Der Burst-Modus

Eine weitere wichtige Aufgabe der HEPI besteht in der Erkennung von Gamma-Ray Bursts (GRB). Die HEPI stellt fest, ob ein Burst vorliegt, meldet dieses Ereignis weiter und wechselt in einen besonderen Datenmodus. Wie dieser Modus aussieht und welche Daten verarbeitet werden, muß noch genau festgelegt werden. Ein Gamma-Ray Burst tritt selten auf, wobei man mit einem Ereignis pro Monat im Gesichtsfeld des Imager rechnet. Um Hardware-Speicher zu sparen, werden die Burst-Ereignisse im Histogrammspeicher der CsI-Ereignisse abgelegt. Dadurch gehen im ungünstigsten Fall Daten von 1000 sec Integrationszeit verloren. Da ein Gamma-Ray Burst aber nur selten auftritt, ist dieser Datenverlust tolerierbar. Um auch Informationen über den Beginn des Bursts vor dem Triggerzeitpunkt zu erhalten, speichert die HEPI während des Histogramm-Modus die einlaufenden Daten in einem sogenannten Pre-Burst-Buffer. Dieser Pre-Burst-Buffer besteht aus einem zyklisch organisierten RAM. Das heißt, der Speicher wird ständig mit den registrierten Ereignisdaten beschrieben. Sobald er vollständig beschrieben wurde, beginnt man wieder an der Anfangadresse des RAMs und überschreibt die zuvor gespeicherten Daten. Auf diese Weise sind immer die Ereignisse einer gewissen Zeitspanne gespeichert. Wird nun ein Burst detektiert, wird die Aufzeichnung in diesem Buffer gestoppt und der Inhalt telemetriert, um den Verlauf des Bursts vor der Burstdetektion zu erhalten.

Die Größe des Pre-Burst-Buffers wird durch das Zeitintervall t_p , das die Zeit zwischen dem Anfang des Burst und dem Trigger Zeitpunkt angibt, und der Zählrate während dieses Zeitintervalls bestimmt. Die Abbildung 6.4 zeigt den Prozentsatz an Bursts aus dem BATSE 1B [14] Katalog mit $t_p<t$.

Da die CdTe-Ereignisdaten direkt übertragen werden, müssen nur die CsI-Ereignisdaten in diesem Buffer gespeichert werden. Bei einem Hintergrund von 15000 counts/s und zwei Byte pro Ereignis ergeben sich 30 kByte an Daten pro Sekunde.

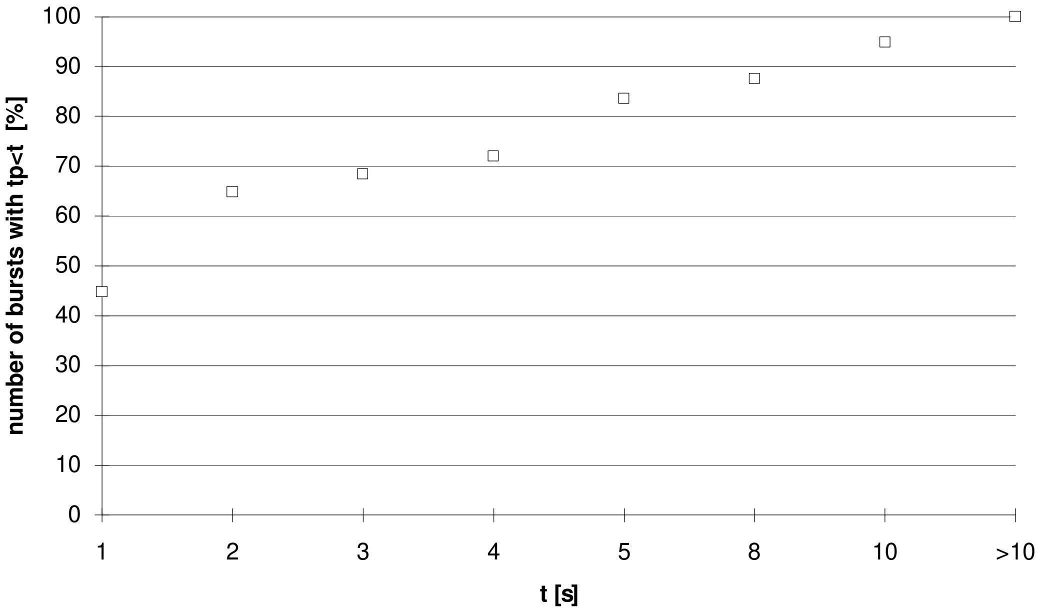


Abbildung 6.4: Prozentsatz der Bursts mit $t_p<t$ (siehe Text) [15]

Die Tabelle 6.1 zeigt die Buffergröße, die damit meßbare Zeitspanne und den Prozentsatz an Gamma-Ray Bursts, der damit erfaßt werden kann. Da die Zählraten bei einem Burst drastisch ansteigen, können nur noch die relevanten Daten gespeichert werden. Sie werden im RAM der HEPI zwischengespeichert und nach dem Burst übertragen. Es wird geplant, einen Pre-Burst-Buffer von 300 kByte Größe zu verwenden.

Buffer Größe	tp bei 30kB/s speicherbar	Prozentsatz der GRBs
32kB	1.1s	46 %
64kB	2.1s	65 %
128kB	4.3s	72 %
200kB	6.7s	84 %
256kB	8.5s	88 %
300kB	10s	94.8 %
400kB	13.3s	95.2 %
512kB	17.1s	95.6 %

Tabelle 6.1: Buffergröße, speicherbare Zeit, Prozentsatz der Gamma-Ray Bursts (GRB), deren Verläufe vor dem Triggerzeitpunkt vollständig mit der jewiligen Speichergröße erfaßt werden können.

6.2.7 Die Synchronisation der Detektorebenen

In der DPE wird synchron zur „An Bord Zeit“ ein Reset-Signal erzeugt. Dieses wird von der HEPI zu den Detektoren weitergeleitet. Um unterschiedliche Signalverarbeitungszeiten in den Detektoren auszugleichen, kann dieses Reset-Signal in der HEPI für jede Detektorschicht individuell verzögert werden. Dadurch werden die Zeitzähler in den Detektoren zueinander synchronisiert.

In den nachfolgenden Kapiteln werde ich die von mir bearbeiteten Module der HEPI vorstellen. Die vollständigen Schaltpläne und Simulationen finden sich im Anhang.

6.3 Der CsI-Akkumulator

Beim Entwurf des Moduls CsI-Akkumulator waren folgende Vorgaben zu erfüllen. Die CsI-Ereignisse sollen in ein Histogramm geschrieben werden. Dieses Histogramm wird in einem RAM abgespeichert. Die Zeitinformation der Einzelereignisse geht hierbei verloren. Außerdem wird die Breite des Energiewortes, um Speicher zu sparen, von ursprünglich 10 Bit auf 8 Bit reduziert. Die Energiekanäle werden also teilweise zusammengefaßt (Binning). Welche Kanäle zusammengefaßt werden, soll nach Möglichkeit frei wählbar sein. Damit kann der Beobachter die beste Aufteilung der Energiekanäle für sein Objekt wählen. Um diese programmierbare Reduktion der Energieinformation auf 256 Energiekanäle zu realisieren, kommen verschiedene Möglichkeiten in Betracht.
Mit einer Dekodierlogik könnte jedem der 1024 Eingangswerte ein Ausgangswert aus 256 Möglichkeiten zugeordnet werden. Der Nachteil dieser Lösung besteht in ihrer Unflexibilität. Die Zuordnung der Energieinformation auf die Energiekanäle würde bei der Entwicklung des ASICs festgelegt und könnte danach nicht mehr verändert werden.
Eine andere Möglichkeit besteht aus einem Array aus Komparatoren. Jeder Komparator vergleicht den Energiewert mit einem Wert, der die Grenze eines Energiekanals bestimmt. Dazu werden 255 Komparatoren benötigt. Ist ein Wert kleiner oder gleich dem Vergleichswert des Komparators und größer als der Vergleichswert des Komparators des nächst kleineren Bereichs, so fällt er in diesen Energiekanal. Die Vergleichswerte der Komparatoren können frei geladen werden. Zu ihrer Speicherung muß jeweils ein Datenregister vorhanden sein. Bei dieser Realisierung ist die Flexibilität der Zuordnung der Energiekanäle gewährleistet. Allerdings wird eine große Anzahl an Gattern benötigt. Abgeschätzt ergeben sich ungefähr 20000 Gatter. Sie würden einen großen Teil des ASICs benötigen. Außerdem würde ein großer FPGA für das Entwicklungsmodell benötigt. Die üblicherweise am Institut verwendeten FPGAs besitzten eine Kapazität von 13000 Gatteräquivalenten.
Die Realisierung für die ich mich entschieden habe, benötigt deutlich weniger Gatter auf dem ASIC und bietet trotzdem die Flexibilität der freien Zuordnung der Energiekanäle. Um die Energieinformation den Energiekanälen zuzuordnen, wird eine sogenannte Look-Up Table (LUT) verwendet. Diese ist in einem RAM gespeichert. Die 10 Bit Energieinformation wird an die Adresseingänge dieses RAMs angelegt. In jeder Adresse des RAMs ist ein 8 Bit Energiekanal gespeichert. Damit wird jeder Eingangsenergie ein reduzierter Energiekanal zugeordnet.
Abbildung 6.5 zeigt das Prinzip des CsI-Akkumulators. Das SELINT-Signal schaltet vom Lademodus für das Look-Up RAM in den Histogrammodus um. Im Lademodus des Look-Up RAMs werden über die DAT- und ADR-Eingänge die Daten des Look-Up

Tables zum RAM übertragen. Mit dem WR-Signal übernimmt das RAM diese Daten. Das INIT-Signal startet die Initialisierung des Histogramm RAMs. Das Modul HRAM INITIALISIERUNG erzeugt dabei die RAM-Adressen und ein Enable-Signal INITENA mit dem die Daten in das RAM übernommen werden. Die Adressen gelangen über einen Multiplexer zum Adresseingang des RAMs. Dieser Multiplexer wird vom INITSEL-Signal gesteuert, das ebenfalls im Modul HRAM INITIALISIERUNG erzeugt wird. Zusätzlich wird mit dem INITSEL-Signal der Multiplexer umgeschaltet, der die Daten zur Initialisierung (ZERO) an den Dateneingang des RAMs legt. Jede Speicherzelle des RAMs wird mit dem Wert Null initialisiert.

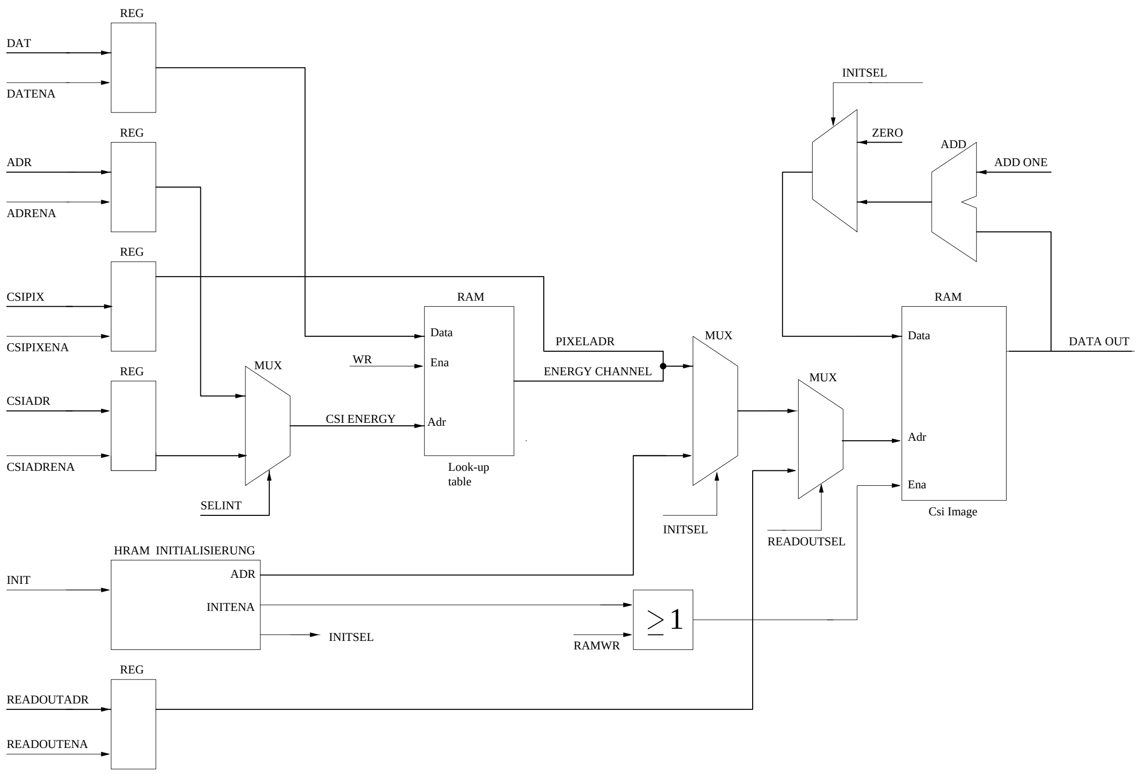


Abbildung 6.5: Schemadarstellung des CsI-Akkumulators (siehe Text)

Im Histogramm-Modus wird die CsI-Energieinformation (CSIADR) auf den Eingang des Look-Up Table RAMs geführt. Am Ausgang des RAMs liegt nun der zugeordnete Energiekanal an. Dieser wird mit den Positionsdaten (CSIPIX) zusammengeführt und adressiert das Histogramm RAM (CsI Image). Über den Addierer (ADD) wird zum Wert der adressierten RAM-Speicherzelle Eins hinzugezählt und der neue Wert mit dem RAMWR-Signal wieder im RAM gespeichert. Somit wird jedes Ereignis gezählt, das in diesen Energiebereich fällt und diese Pixelposition besitzt.

Mit dem READOUTENA-Signal werden die Adressen (READOUTADR) zum Auslesen des Histogramm-RAMs übernommen.

Dieses Modul kann problemlos für die Speicherung der Histogramme der Kalibrations- und Mehrfach-Ereignisse erweitert werden. Das Kalibrations- und Mehrfach-Ereignis-Flag dient dabei zur Auswahl des Adressbereichs in dem die Daten gespeichert werden. und zur Auswahl der Look-Up Tabelle. Bei den Kalibrations-Ereignissen wird eine Auflösung von 32 Energiekanälen pro Linie (511 keV und 1275 keV) der Kalibrations-Quelle benötigt.

Da auch im Polarimetrie-Modus und bei den Ratemetern die Ereignisse in Histogrammen gespeichert werden, läßt sich der CsI-Akkumulator auch für diese Aufgaben verwenden.

Für die Histogramme werden insgesamt 2.262 Mbyte an Speicher auf der HEPI benötigt. Die Einfach- und Mehrfach-CsI-Ereignisse belegen dabei jeweils 1 Mbyte Speicher. Für die Kalibrations-Ereignisse werden 262 kByte gebraucht. Der Speicher, der für die Ratemeter benötigt wird beträgt 8 kByte. Beim Polarimetrie-Modus werden ebenfalls 2 MByte Speicher benötigt. Da dieser Modus immer nur alternativ zu dem CsI-Histogramm-Modus verwendet wird, erhöht sich die insgesamt benötigte Speichermenge nicht.

6.3.1 Simulation

Die eben beschriebene Schaltung wurde in einer vereinfachten Form simuliert. Diese Testschaltung arbeitet mit einem 4 Bit breiten Energiewert und einem einzigen Positionswert, ist aber vom Prinzip her identisch mit der kompletten Schaltung. Als Takt wurde für die Testschaltung eine Periode von 100 ns gewählt. Die Ergebnisse der Simulation sind in Abbildung 6.6 dargestellt und sollen kurz erläutert werden.

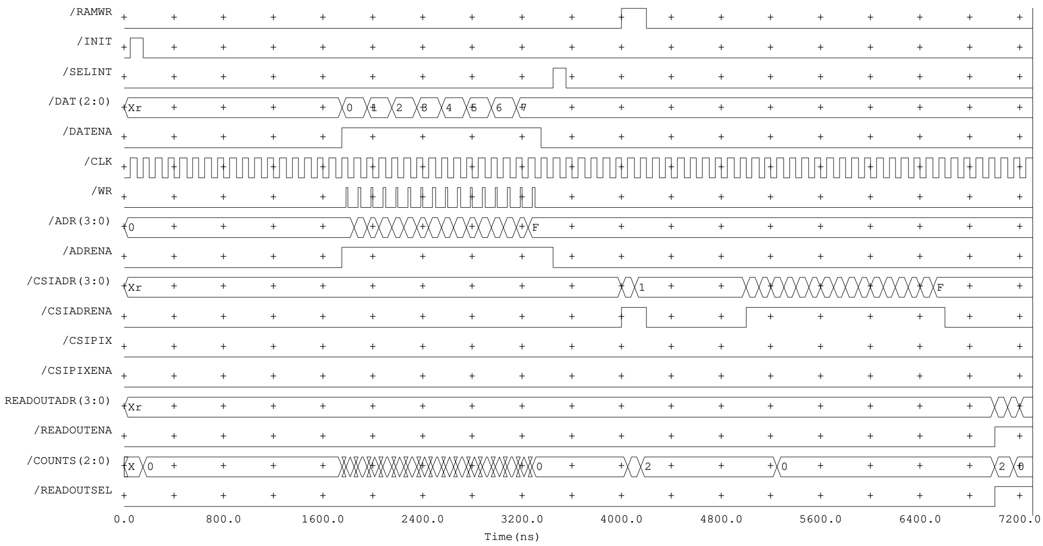


Abbildung 6.6: Simulation des CsI-Akkumulators (siehe Text)

Zuerst wird mit dem INIT-Signal der Initialisierungsprozeß gestartet. Dabei wird jede Adresse im CsI-Image-RAM mit einer Null beschrieben. Danach wird die LUT in das Look-Up RAM geladen. Dazu werden die Adressen über das ADR-Signal und das ADRENA-Signal an das RAM angelegt. Die Daten werden über DAT und DATENA eingelesen. Das WR-Signal liest die Daten in das RAM ein. Das Look-Up Table RAM wurde so initialisiert, daß immer zwei Energiewerte zu einem Energiekanal zusammengefaßt werden.
Anschließend gibt man einen Impuls auf den SELINT-Eingang, woraufhin das Look-Up RAM vom Initialisierungs-Modus zum Histogramm-Modus umgeschaltet wird. Im Histogramm-Modus werden die CsI-Ereignisse aufsummiert. Hierbei liegt der CsI-Energiewert (CSIADR) über das Look-Up RAM am Image RAM an. Die CsI-

Pixeladresse (CSIPIX) liegt direkt an den Adresseingängen des Image RAMs. Mit dem RAMWR-Signal wird der Wert der ausgewählten Adresse jeweils um eins erhöht. Hier in der Simulation werden bei 4000 ns für den Energiewert null und eins je ein Ereignis im Pixel null registriert. Da Energiewert null und eins zusammengefaßt werden, sollte bei Auslesen dieses Energiekanals der Wert zwei am Ausgang erscheinen.
Zum Auslesen werden die gewünschten Adressen über READOUTADR und READOUTENA an die Adresseingänge des Image RAMs angelegt. Mit Hilfe des Signals READOUTSEL wird am Ende vom Histogramm-Modus in den Auslese-Modus umgeschaltet. Am Ende der Simulation (7000 ns) wird das Histogramm RAM ausgelesen. Der Ausgang COUNTS zeigt den Wert zwei, was erwartet wurde.

6.3.2 Das Interface des CsI-Akkumulators zur Steuerung

In diesem Abschnitt möchte ich die vom CsI-Akkumulator benötigten Eingangs- und Ausgangsignale auflisten.

INIT : Initialisierung des Histogramm RAMs. Es muß nach jedem Auslesevorgang des RAMs gesendet werden, um alle Speicherplätze zurückzusetzen.
SELINT : Umschaltung zwischen Histogramm und Look-Up RAM Initialisierungs-Modus
DAT : Eingang für die Daten der Look-Up Table
DATENA : Enable-Signal für die Daten der Look-Up Table
WR : Schreibsignal für die Look-Up Table
ADR : Adresse zur Initialisierung des Look-Up RAMs
ADRENA : Enable-Signal für die Adressdaten zur Initialisierung des Look-Up Table RAMs
CSIADR : Energiewert der CsI-Ereignisse
CSIPIX : Pixeladresse der CsI-Ereignisse
RAMWR : Schreibsignal für das Histogramm RAM bei Initialisierung und Normal-Modus
READOUTADR : Adresse die zur Schnittstelle ausgelesen wird
READOUTENA : Übernahmesignal für die Ausleseadresse
READOUTSEL : Umschaltung zwischen Histogramm- und Auslese-Modus
CLK : Takt
CSIADRENA : Enable-Signal für die Übernahme der CsI-Adresse
CSIPIXENA : Übernahmesignal für die CsI-Pixeladresse
COUNTS : Wert der ausgelesenen Adresse

6.4 Zeitkoinzidenz zwischen CsI- und CdTe-Ereignissen

Im Zeitkoinzidenz-Modul werden die Daten aus den beiden Detektorenschichten auf zeitliche Korrelationen untersucht. Sind Ereignisse in beiden Detektorschichten zeitgleich, stammen sie mit hoher Wahrscheinlichkeit vom selben Photon. Im Zeitkoinzidenz-Modul werden die Zeitinformationen der Detektordaten miteinander verglichen und im Falle einer Übereinstimmung zu einem Datenwort zusammengefaßt. In der DPE wird auf diese Ereignisse ein Filteralgorithmus angewendet, um Hintergrund-Ereignisse unterdrücken zu können.

Könnte man auf der Detektorseite den Zeitpunkt, zu dem ein Ereignis ausgelöst wurde, genau bestimmen, wäre der Vergleich einfach. Dies ist aber aus folgendem Grund nicht möglich. Löst ein Photon im Detektor ein Ereignis aus, so wird es erst ab einer bestimmten Triggerschwelle registriert. Aufgrund von Signallaufzeiten in den ASICs der Frontend-Elektronik und Rauschen kann der Zeitpunkt nicht genau bestimmt werden. Trägt man die Differenz zwischen wahrer Zeit und gemessener Zeit in einem Histogramm auf, erhält man eine gaußförmige Verteilung mit einer Standardabweichung (1σ) von etwa 0.2-0.3 µs [16]. Dies macht es erforderlich, für die Bestimmung der Zeitkoinzidenz einen Zeitspanne heranzuziehen. Ereignisse, die in dieser Spanne liegen, gelten dann als vom selben Photon stammend. Wie groß dieses Zeitfenster gewählt werden muß, hängt von der Ungenauigkeit der Zeitbestimmung für das Ereignis im Detektor ab. Ein guter Wert für die Größe des Zeitfensters liegt bei 3σ der Gauß-Verteilung.

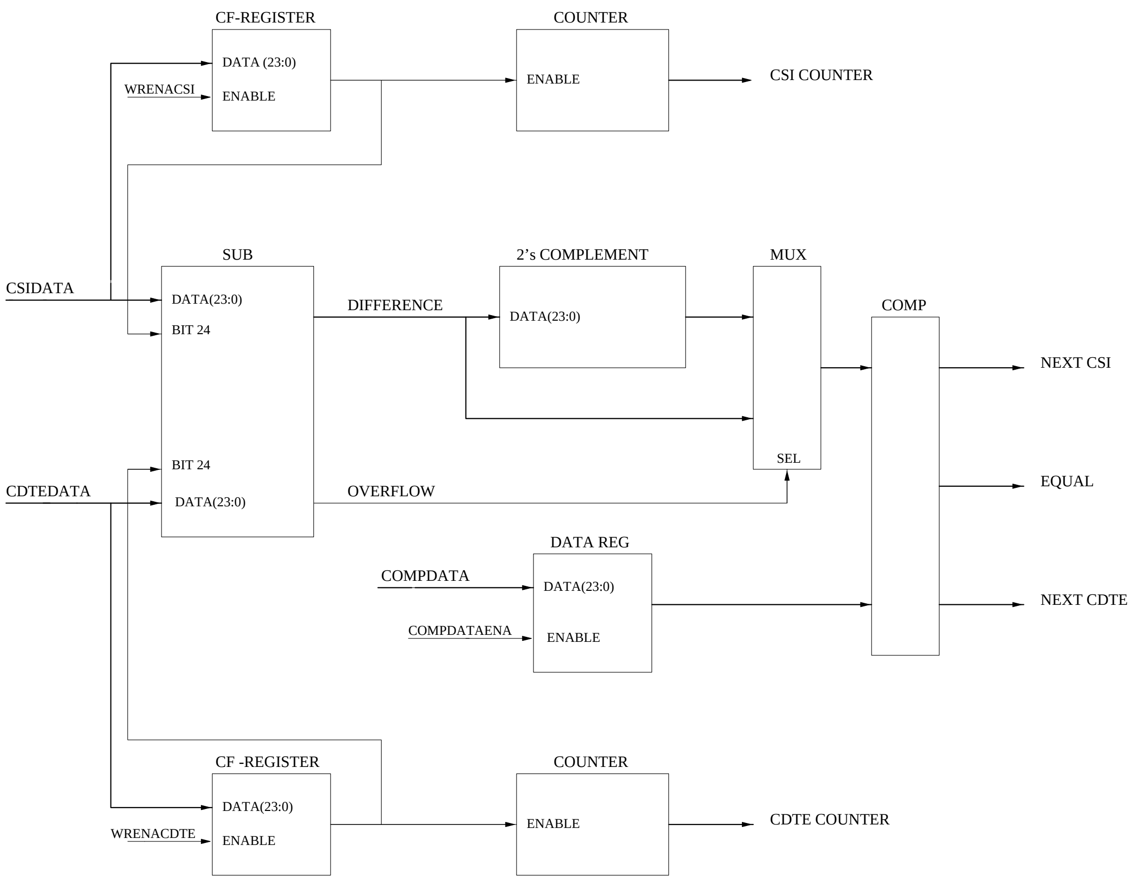


Abbildung 6.7: Blockschaltbild des Zeitkoinzidenz Moduls

In Abbildung 6.7 ist ein Prinzipschaltbild des Zeitkoinzidenz-Moduls dargestellt. Um eine Zeitkoinzidenz zu detektieren, werden die beiden Zeitinformationen zuerst voneinander subtrahiert. Für den Fall, daß die CdTe-Zeitinformation größer ist als die CsI-Zeitinformation, kommt es zu einem Überlauf. In diesem Fall wird das Zweierkomplement der Differenz gebildet um den Betrag des Abstandes der beiden Zeitinformationen zu erhalten. Der Betrag des Abstandes wird in einem Komparator mit dem gewählten Zeitfenster (Comp Data) verglichen. Das heißt, sobald die Differenz kleiner oder gleich dem Zeitfenster ist, werden die Daten als zeitgleich betrachtet und beide Ereignisse ausgelesen. Bei zeitgleichen Ereignissen wird nur eine Zeitinformation weitergegeben. Ist die Differenz größer als das Zeitfenster, so wird für den Fall eines

Überlaufs das nächste CsI-Ereignis eingelesen. Liegt kein Überlauf vor, wird das nächste CdTe-Ereignis eingelesen. Wird keine Zeitkoinzidenz detektiert, wird immer das ältere Ereignis ausgelesen.
Alle 2 s bekommen die Zeitzähler der Detektoren einen Reset-Impuls, mit dem sie zurückgesetzt und synchronisiert werden. Um eine Information darüber zu erhalten, ob es zu einem Nulldurchgang der Zeitinformation bei zwei aufeinanderfolgenden Ereignissen gekommen ist, werden die an den Eingängen anliegenden Ereignisdaten im CF-Register mit den Daten des vorherigen Ereignisses verglichen. Ein Nulldurchgang tritt immer nach Ablauf des zwei Sekunden Zeitzyklus auf. Ist die vorherige Zeit größer, so wird ein Zähler um eins erhöht. Diese Zähler, von denen jeder für eine Detektorebene zuständig ist, können abgefragt werden, um eine Kontrolle der Synchronität des Auslesevorgang aus den Ausgangs-FIFOs der beiden Detektorebenen zu ermöglichen. Die Zähler dürfen höchstens einen Unterschied von eins aufweisen. Ist die Differenz zwischen ihnen größer, so läuft der Auslesevorgang aus den FIFOs nicht mehr synchron zueinander. Zusätzlich wird am Subtrahierer das Bit 24 des jeweiligen Einganges (CdTe oder CsI) auf eins gesetzt. Dieses Bit wird benötigt, damit die Differenz der beiden Zeiten wieder den korrekten Wert annimmt und das Overflow-Flag korrekt gesetzt wird. Sobald ein Nulldurchgang bei den Daten des anderen Detektors auftritt, wird dieses Bit 24 wieder zurückgesetzt. Die Daten aus den Detektoren stammen dann wieder aus der gleichen Zeitperiode. Somit ist gewährleistet, daß die Differenz der beiden Zeitinformationen den korrekten Abstand ergibt.

6.4.1 Simulation

Das Zeitkoinzidenz Modul wurde ebenfalls mit Quicksim simuliert. Die Ergebnisse sind in Abbildung 6.8 dargestellt.
Zuerst wird mit dem COMPDATAENA-Signal der Wert für das Zeitfenster in das Zeitkoinzidenz-Modul geladen. Dieser Wert liegt danach an COMPDATA an. Die beiden Zeitdaten der Detektorereignisse liegen an den Eingängen, CSIDATA und CDTEDATA, an. Da der Wert des CSIDATA-Eingangs kleiner als der Wert am CDTEDATA-Eingang ist, wird das NEXTCSI-Flag gesetzt. Das heißt, die CsI-Daten können zur Weiterverarbeitung gegeben werden, und danach wird das nächste CsI-Ereignis eingelesen. Mit den WRENACSI- und WRENACDTE-Signalen werden die Zeitinformationen in das Zeitkoinzidenz-Modul übernommen, um für den nächsten Vergleich zur Verfügung zu stehen. Beim dritten CsI-Ereignis erfolgt ein Nulldurchgang (~1900 ns). Das heißt der Counter, der diese Ereignisse zählt (CsICounter), wird um eins erhöht. Im Falle eines solchen Nulldurchgang brauchen die Ausgangsflags (NEXTCSI, NEXTCDTE, EQUAL) eineinhalb Takte, bis sie sich stabilisiert haben. Diese Verzögerung muß man bei der Auswertung der Flags beachten, um keine falschen Werte zu bekommen.

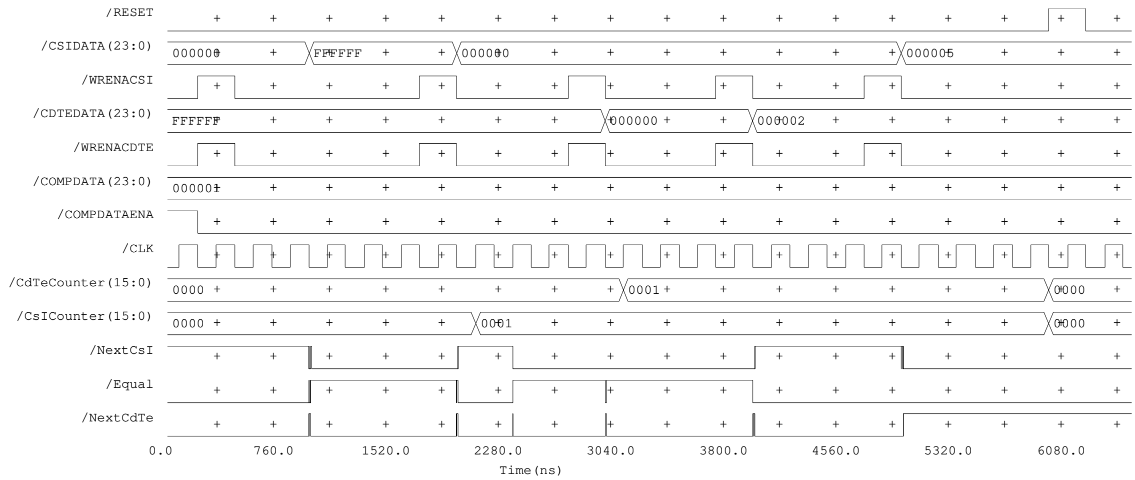


Abbildung 6.8: Simulation des Zeitkoinzidenz Moduls

6.4.2 Das Interface des Zeitkoinzidenz-Moduls zur Steuerung

Folgende Eingangs- und Ausgangssignale werden vom Zeitkoinzidenz-Modul gebraucht. Ihre Funktion wird in Stichworten beschrieben.

CSIDATA : Eingang für das CsI-Zeitwort
CDTEDATA : Eingang für das CdTe-Zeitwort
COMPDATA : Eingang für die Daten, welche die Größe des Zeitbereichs bestimmen. Die Daten werden mit der ansteigenden Taktflanke übernommen.
CLK : Takt
COMPDATAENA : Enable-Eingang für das Datenregister, in dem die Größe des Zeitbereichs gespeichert wird
WRENACSI : Übernahmesignal für die CsI-Zeitdaten in das Vergleichsregister
WRENACDTE : Übernahmesignal für die CdTe-Zeitdaten in das Vergleichsregister
CSICOUNTER : Ausgang des Nulldurchgang-Zählers CsI
CDTECOUNTER : Ausgang des Nulldurchgang-Zählers CdTe
NEXT CSI : Bei gesetztem Flag wird von der Steuerung das nächste CsI-Ereignis eingelesen
NEXT CDTE : Bei gesetztem Flag wird von der Steuerung das nächste CdTe-Ereignis eingelesen
EQUAL : Eine Zeitkoinzidenz wurde detektiert

6.5 Die Rekonstruktion der Mehrfach-Ereignisse

Im Modul Multiple Event Reconstruction (MER) werden die Mehrfach-CsI-Ereignisse aufsummiert und das Ursprungspixel rekonstruiert. Das Ursprungspixel ist das Pixel, in dem die erste Wechselwirkung des einfallenden Photons mit dem CsI-Kristall stattfand.

Das MER-Modul summiert die Amplituden der Ereignisse auf, so daß man die ursprüngliche Energie des Photons wieder erhält. Hierbei tritt jedoch ein Problem auf. Jedes Pixel des CsI-Detektors hat eine unterschiedliche Empfindlichkeit. Die Amplitude ist für jedes Pixel proportional zur Energie des Photons. Aufgrund der unterschiedlichen Empfindlichkeit differieren bei gleicher Photonenenergie die Amplituden für die verschiedenen Pixel. Für den Fall, daß der Fehler bei der Summation der Energie zu groß wird, muß an Bord eine Energiekorrektur durchgeführt werden. Da dies einen zusätzlichen Aufwand bei der Konstruktion der HEPI bedeutet, sollte genau geprüft werden, ob diese Korrektur notwendig ist.
Die Korrektur kann mit Hilfe linearer Regression erfolgen. Auf der HEPI wird dazu eine zusätzliche Look-Up Table benötigt, in der für jedes Pixel Steigung und Offset der Eichgeraden gespeichert sind. Außerdem benötigt man eine ALU (Arithmetic Logic Unit), welche die Amplituden der Ereignisse mit der Steigung multipliziert und den Offset hinzu addiert. Im vorliegenden Modul ist diese Energiekorrektur noch nicht realisiert.
Zusätzlich zur Summation der Amplitude wird mit folgendem Rekonstruktionsverfahren das Ursprungspixel des Mehrfach-Ereignisses bestimmt.

6.5.1 Das Ursprungspixel Rekonstruktionsverfahren

Um ein geeignetes Verfahren zur Rekonstruktion des Ursprungspixels zu finden, wurde eine Reihe von Monte-Carlo Simulationen durchgeführt [17]. Dabei wurden folgende Annahmen gemacht. Der Abstand zwischen den Detektorschichten wurde mit 12 cm angenommen. Auf jedes CsI-Pixel wurden 10000 Photonen geschossen. Ein CsI-Pixel entspricht der Fläche von vier CdTe-Pixeln. Bei den CdTe-Pixeln lag der Energiebereich bei 20 keV - 1000 keV und bei den CsI-Pixeln reichte er von 120 keV - 4500 keV.
Bei den Zweifach-Ereignissen kam man zu den in der Tabelle 6.2 aufgeführten Ergebnissen. In der ersten Spalte steht die ursprüngliche Photonenenergie. In der zweiten Spalte die Effizienz mit der ein CsI-Zweifach-Ereignis erzeugt wird. In Spalte drei und vier stehen die Anteile der Ereignisse, die ein Maximum (E_{max}) bzw. ein Minimum (E_{min}) der Energie im ursprünglichen Pixel abgegeben haben. Die Anteile von E_{min} und E_{max} ergeben zusammen nicht genau 1.00, da bei einigen Wechselwirkungen die Energie pro Pixel unterhalb der unteren Schwellenenergie liegt, so daß diese Ereignisse nicht berücksichtigt werden.

Ursprüngliche Photonenenergie	eff (CsI-Zweifach-Ereignis)	E_{min}	E_{max}
300keV	0.086	0.79	0.14
500keV	0.166	0.53	0.43
700keV	0.157	0.30	0.69
1000keV	0.131	0.16	0.83
2000keV	0.087	0.05	0.93
5000keV	0.043	0.02	0.87

Tabelle 6.2: Die Tabelle zeigt für Photonen unterschiedlicher Energie die Wahrscheinlichkeiten, ein Zweifach-Ereignis auszulösen. Zusätzlich sind der prozentuale Anteile von Photonen aufgeführt, welche die maximale bzw. die minimale Energie im ersten Pixel deponieren.

Rekonstruiert man nun das ursprüngliche Pixel anhand der abgegebenen Energie, so erreicht man bei den Zweifach-Ereignissen einen Anteil richtig rekonstruierter Ursprungspixel von ungefähr 50% bei 500 keV bis zu ungefähr 90% bei 5000 keV. Dieser Algorithmus ist auf der HEPI recht einfach zu realisieren. Es muß nur bestimmt werden, in welchem Pixel das Ereignis die maximale und in welchem es die minimale Energie abgegeben hat. Anschließend wird, je nachdem in welchem Bereich die aufsummierte Energie liegt, entsprechend dem Wert aus Tabelle 6.2, das Ursprungspixel der maximalen oder der minimale Energieabgabe zugeordnet. Der graue Bereich zeigt an, welches Pixel für die jeweiligen Energiebereiche als Ursprungspixel ausgewählt wird.

Urspüngliche Photonenenergie	eff (CsI-Dreifach-Ereignisse)	E_{min}	E_{mid}	E_{max}
300 keV	0.0066	0.32	0.55	0.11
500 keV	0.0308	0.22	0.38	0.37
700 keV	0.0430	0.15	0.22	0.61
1000 keV	0.0487	0.09	0.13	0.78
2000 keV	0.0422	0.02	0.06	0.90
5000 keV	0.0395	0.00	0.01	0.93

Tabelle 6.3: Die Tabelle zeigt für ein Photon unterschiedlicher Energie die Wahrscheinlichkeit ein Dreifach-Ereignis auszulösen. Außerdem sind der prozentuale Anteil von Photonen aufgeführt, welche die maximale, minimale oder die mittlere Energie im ersten Pixel deponieren.

Bei Dreifach-Ereignissen (siehe Tabelle 6.3) läuft der Vorgang analog ab, nur kommt noch der mittlere Energiebereich (E_{mid}) hinzu. Die Hardware-Realisierung ist im nächsten Abschnitt beschrieben.

6.5.2 Funktionsbeschreibung des MER-Moduls

Die Abbildung 6.9 zeigt ein Blockschaltbild des MER-Moduls. Es rekonstruiert den Ort des Eintreffens eines Gammaquants aus den koinzidenten Signalen zweier oder mehrerer Pixel. Im Register DATA REG wird die Zeitinformation der Ereignisse gespeichert. Der Komparator vergleicht diese Daten mit den an den Eingängen des MER-Moduls anliegenden, und überprüft sie auf Gleichheit. Am Flag Eingang liegt das Mehrfach-Ereignis-Flag der Daten an. Es wird immer dann gesetzt, wenn es sich um ein Mehrfach-Ereignis handelt. Der COUNTER hat die Aufgabe zu bestimmen, ob es sich um ein Zweifach- oder Dreifach-Ereignis handelt. Der SORTER sortiert die Daten nach der Energie. Im Modul COMP ENERGY wird schließlich anhand der Gesamtenergie und des Zählerwertes bestimmt, aus welchem Register des Sortierers die Daten für die Position ausgelesen werden. Über die Gesamtenergie wird das Register des Sortierers angewählt, in dem die Daten mit der größten Wahrscheinlichkeit für das Ursprungspixel stehen. An den Eingängen des COMP ENERGY-Moduls liegen die Vergleichsdaten an, mit deren Hilfe der Energiebereich festgelegt wird, in den die Daten fallen.

Der Funktionsablauf im MER-Modul sieht folgendermaßen aus. Zu Beginn wird mit dem INIT-Signal die Verarbeitung des im CsI-Inputregister der HEPI gespeicherten Ereignisses gestartet. Der Ablauf innerhalb des MER-Moduls wird durch das CONTROL-Modul gesteuert. Zuerst wird geprüft, ob das Mehrfach-Ereignis-Flag

(FLAG) gesetzt ist. Ist das nicht der Fall, werden die Daten sofort an die Ausgangsregister geleitet und dort gespeichert. Das READY-Flag fordert das nächste Ereignis an.

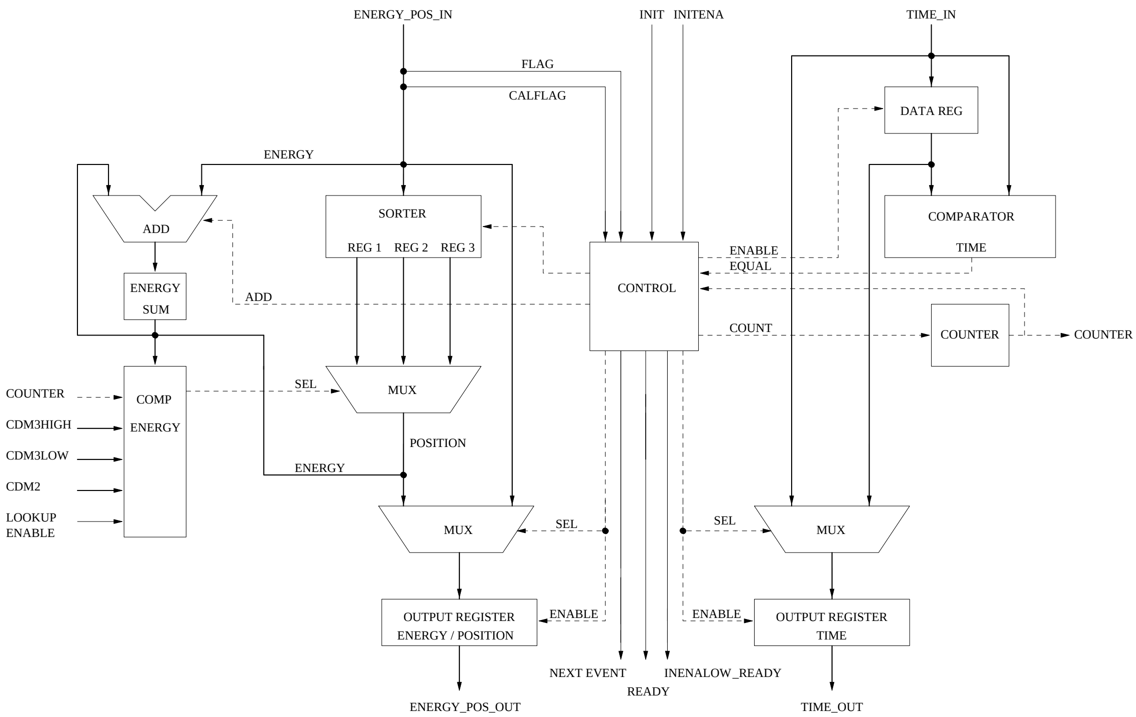


Abbildung 6.9: Blockschaltbild der Multiple Event Reconstruction

Ist dagegen das Mehrfach-Ereignis-Flag gesetzt, so werden die Zeitdaten in ein Register (DATA REG) übernommen. Die Energie und Position werden in den SORTER und den Akkumulator (ADD) gelesen. Gleichzeitig wird der COUNTER auf Eins gesetzt und die Energie aus dem Akkumulator in ein Datenregister (ENERGY SUM) übernommen. Mit dem NEXT_EVENT-Flag wird das nächste Ereignis angefordert. Hierbei sind nun mehrere Fälle zu unterscheiden.
Im ersten Fall ist das nun eingelesene Ereignis nicht zeitgleich mit dem ersten Ereignis. Da nur solche Ereignisse als Mehrfach-Ereignis gelten, deren Zeitinformation identisch sind und deren Mehrfach-Ereignis-Flag gesetzt ist, gilt das Ereignis nicht als Mehrfach-Ereignis. Das gespeicherte Ereignis wird nun in die Ausgangsregister ausgelesen. Das INENALOW_READY-Flag zeigt an, daß ein verarbeitetes Ereignis vorliegt. Gleichzeitig zeigt es an, daß das INENA-Flag auf null gesetzt werden soll, damit die HEPI dieses Ereignis weiterverarbeiten kann. Mit dem INENA-Flag wird die Datenverarbeitung in der MER angehalten. Ist diese Weiterverarbeitung in der HEPI abgeschlossen, setzt die Steuerung der HEPI das INENA-Flag wieder auf eins. Bei den internen Registern wird dann ein Reset durchgeführt und anschließend das neue Ereignis wie das vorherige verarbeitet.
Im zweiten Fall ist das Mehrfach-Ereignis-Flag nicht gesetzt. Auch dann wird das gespeicherte Ereignis in die Ausgangsregister geladen und anschließend das neue Ereignis weiterverarbeitet.
Im dritten Fall sind beide Flags gesetzt. Es liegt ein Mehrfach-Ereignis vor. Dann werden die Daten wiederum im Sortierer gespeichert und im Akku aufsummiert. Der COUNTER wird auf zwei gesetzt. Danach wird das nächste Ereignis eingelesen. Auch hier können

wieder die vorhergehenden Fälle auftreten, die analog zum oberen Abschnitt behandelt werden. Für den Fall, daß beide Flags gesetzt sind, wird der COUNTER auf drei gesetzt und dann das neue Ereignis von der Schnittstelle gelesen. Hierbei spielt es nun keine Rolle, um was für ein Ereignis es sich handelt, da nur Mehrfach-Ereignisse aus höchstens drei Pixeln betrachtet werden und der COUNTER schon auf drei gesetzt ist. Die gespeicherten Daten werden also in jedem Fall ausgelesen. Danach erfolgt ein Clear und anschließend beginnt der Vorgang von neuem.
Einfach-Ereignisse mit gesetztem Kalibrations-Flag werden sofort zu den Ausgangsregistern geleitet. Mehrfach-Kalibrations-Ereignisse werden nicht verarbeitet, sondern verworfen.
Das INIT- und INENA-Signal wurden von mir im Simulationsprogramm erzeugt, da noch keine HEPI- Steuerung zur Verfügung stand.

6.5.3 Simulation

In dieser Simulation ist ein Beispiel für den oben beschriebenen Ablauf dargestellt. Zu Beginn der Simulation wird mit dem CLR-Signal das Modul für die Rekonstruktion der Mehrfach-Ereignisse in den Ausgangszustand versetzt. Danach wird das COMP ENERGY-Modul mit den Werten, die an den CMD3LOW, CMD3HIGH und CMD2 Leitungen anliegen, initialisiert. Am TIME_IN und ENERGY_POS_IN Eingang liegen die gesamten Daten des CsI-Ereignisses an. Bit 0-9 des ENERGY_POS_IN Signals enthalten die Amplitude des Ereignisses. Bit 10-22 enthalten die Position und Bit 22,23 das Mehrfach-Ereignis-Flag und das Kalibrations-Flag.
Mit dem erstem INIT-Signal werden die Ereignisdaten in das MER-Modul übernommen. Es handelt sich bei diesen Daten um ein Einfach-CsI-Ereignis, da weder das Kalibrations-Flag noch das Mehrfach-Ereignis-Flag (FLAG) gesetzt ist. Da es sich um ein Einfach-Ereignis handelt und das INENA Flag gesetzt ist, wird das Ereignis sofort zu den Datenausgängen des MER-Moduls geschickt. Mit dem nächsten INIT-Signal werden die nächsten Daten eingelesen. In diesem Fall ist das Mehrfach-Ereignis-Flag gesetzt. Die Daten werden zunächst im MER-Modul gespeichert. Sobald mit dem nächsten INIT-Signal die Daten des nächsten Ereignisses eingelesen werden, wird geprüft, ob die Zeitinformation dieses Ereignisses gleich der Zeitinformation des gespeicherten Ereignisses ist. Da die beiden Zeitinformationen in diesem Fall nicht übereinstimmen, können die beiden Ereignisse nicht von dem gleichen Mehrfach-Ereignis stammen. Daher wird das gespeicherte Ereignis ausgelesen und das neue Ereignis in das MER-Modul eingelesen. Bei den zwei folgenden INIT-Signalen liegen Daten mit dem gleichen Zeitwort an den Eingängen. Das Mehrfach-Ereignis-Flag ist gesetzt. Diese werden in die MER eingelesen und dort verarbeitet. Beim letzten INIT-Signal wird ein Ereignis mit einem anderen Zeitwort eingelesen. Das aufsummierte Mehrfach-Ereignis wird zu den Ausgängen gegeben. Die Energieinformation entspricht der Summe aus den Energieinformationen der drei Ereignisse. Die Position des Ursprungspixel wurde mit Hilfe des COMP ENERGY-Moduls ausgewählt.

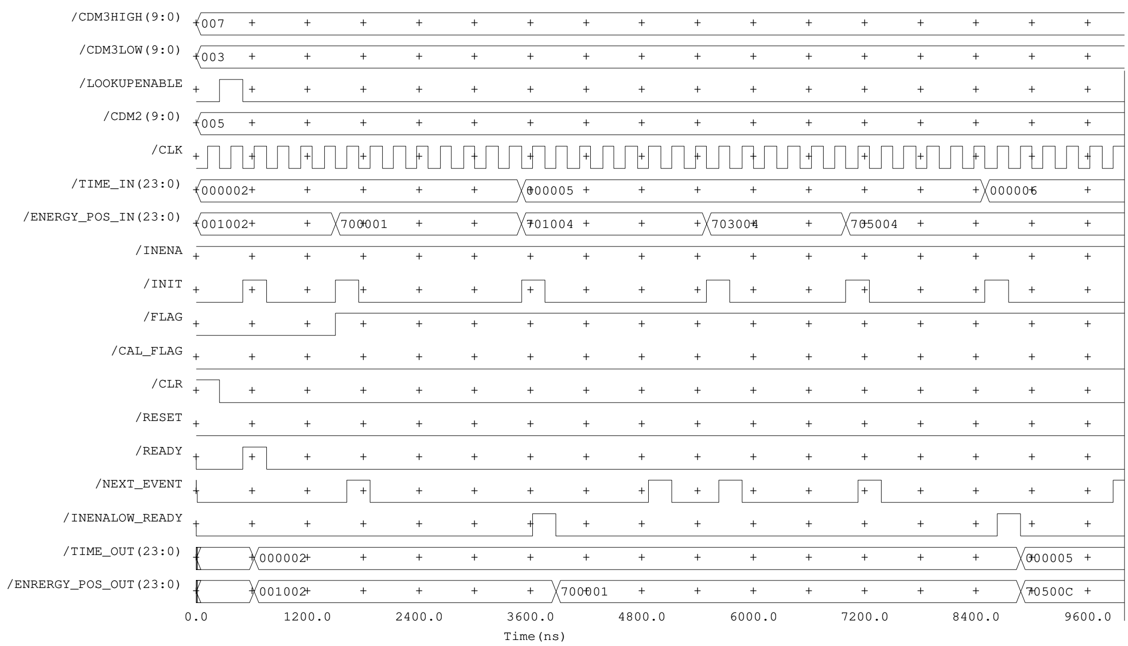


Abbildung 6.10: Simulation der Multiple Event Reconstruction

6.5.4 Das Interface der MER zur Steuerung

Hier sind die Ein- und Ausgangsignale noch einmal aufgelistet.

TIME_IN : Eingang für das Zeitwort des CsI-Ereignisses
CLK : Takt
INENA : Freigabesignal von der HEPI. Das MER-Modul kann die bereits eingelesenen Daten weiterverarbeiten.
INIT : Startsignal zur Verarbeitung der Ereignisse
ENERGY_POS_IN : Eingang für das Energiewort des CsI-Ereignisses
CLR : Clearsignal für die Datenregister und den Sortierer
RESET : Reset-Signal
CDM3LOW : Obere Bereichsgrenze für den Energiewert bei Dreifach-Ereignissen
CDM3HIGH : Untere Bereichsgrenze für den Energiewert bei Dreifach-Ereignissen
CDM2 : Bereichsgrenze zwischen oberem und unterem Energiewert bei Zweifach-Ereignissen
LOOKUPENABLE : Übernahmesignal für die oberen Vergleichswerte
READY : MER hat die Rekonstruktion des Mehrfach-Ereignisses beendet.
NEXT_EVENT : Die nächsten Ereignis Daten können von der Schnittstelle eingelesen werden.
INENALOW_READY : Das INENA Flag wird rückgesetzt, damit die HEPI die vom MER-Modul gelieferten Daten verarbeiten

kann. Die Verarbeitung im MER-Modul wird
solange gestoppt.
TIME_OUT : Zeitwort des rekonstruierten Ereignisses
ENERGY_POS_OUT : Energie und Pixeladresse des rekonstruierten Ereignisses

7 Zusammenfassung und Ausblick

Meine Diplomarbeit stellt den Anfang der Entwicklung eines Hardware Ereignis Prozessors für den Imager an Bord von INTEGRAL dar.
Für die HEPI wurden von mir bisher die Module für die Rekonstruktion der Mehrfach-Ereignisse (MER), die Suche nach Zeitkoinzidenzen und zur Akkumulation der CsI-Histogramme entworfen. Ihre Funktion wurde im Kapitel 6 detailliert beschrieben. Das MER-Modul summiert die Mehrfach-CsI-Ereignisse auf und bestimmt das Ursprungspixel. Dabei werden auch die Kalibrations-Ereignisse richtig verarbeitet. Der Entwurf der Logik für die Suche nach Zeitkoinzidenzen berücksichtigt einen Zeitbereich in dem die Ereignisse als zeitgleich gelten. Außerdem wurde durch den Einbau der Zähler für die Nulldurchgänge die Möglichkeit geschaffen, die Synchronität des Auslesevorgangs der Daten aus den beiden Detektorschichten zu überprüfen. Mit dem CsI-Akkumulator erhält man ein sehr flexibles Modul. Durch die frei wählbare Aufteilung der Energiekanäle kann der Benutzer die optimale Konfiguration für seine Beobachtung wählen.
Bis zur fertigen HEPI sind noch einige Entwicklungsschritte durchzuführen. An erster Stelle steht dabei die endgültige Spezifikation der Funktionen der HEPI. Ohne diese Festlegung kann die Entwicklung nicht sinnvoll fortgeführt werden. Die bisher entworfenen Module decken noch nicht den vollen Funktionsumfang der HEPI ab. Mit dem MER-Modul, dem CsI-Akkumulator und dem Zeitkoinzidenz-Modul steht jedoch schon ein guter Teil der benötigten Funktionen zur Verfügung. Es werden nun noch Module für den Burst-Modus, den Polarimetrie-Modus und die Ratemeter benötigt.
Beim Burst-Modus ist zu spezifizieren, welche Daten gespeichert werden. Vom wissenschaftlichen Standunkt aus gesehen sollen natürlich möglichst alle aufgenommenen Daten zur Weiterverarbeitung verfügbar sein. Die Möglichkeiten der Hardware sind jedoch beschränkt. Es wird nicht möglich sein, alle gewonnenen Daten zu speichern und zu übertragen. Daher muß ein Kompromis gefunden werden, wobei die Verwertbarkeit der Daten gewährleistet sein muß, das heißt mit den gespeicherten Daten müssen die angestrebten wissenschaftlichen Ziele erreichbar sein. Neben der Frage welche Daten gespeichert werden, ist noch die Größe des Pre-Burst-Buffers festzulegen. Außerdem stellt sich natürlich die Frage welche Kriterien herangezogen werden, um einen Bursttrigger auszulösen.
Für den Polarimertie-Modus ist das Verfahren zur Bestimmung des Streuwinkels festzulegen. Dazu wird die Form benötigt, in der die Pixeladresse übertragen wird. Für das Polarimetrie-Modul kann ein Teil des MER-Moduls verwendet werden. Die Summation der einzelnen Amplituden der Zweifach-Ereignisse wird beim Polariemetrie-Modus gleich wie bei der Rekonstruktion der Mehrfach-Ereignisse durchgeführt. Eventuell kann das Polarimetrie-Modul noch in das MER-Modul integriert werden, wie in Abbildung 6.3 zu sehen.
Auch bei den Ratemetern kann auf schon vorhandene Teile zurückgegriffen werden. Für die Summation der Ereignisse pro Pixel kann das Prinzip des CsI-Akkumulators verwendet werden.
Ein weiterer Punkt, der geklärt werden muß, ist die Notwendigkeit einer an Bord durchgeführten Energiekorrektur der Daten. Dies würde eine zusätzliche Look-Up-Table

und eine ALU erfordern. Bisher ist die Auswertung der Mehrfach-Ereignisse bis zu Dreifach-Ereignissen vorgesehen. Zur Zeit wird allerdings geprüft, ob die zusätzliche Übertragung von Dreifach-Ereignissen einen meßbaren Informationsgewinn bedeutet.
Um die Daten von den Detektoren zur HEPI zu übertragen, werden noch die Empfänger der Schnittstellen auf der HEPI gebraucht. Sie sollen in den ASIC integriert werden und stellen somit eine Komponente der Logik dar.
Nachdem alle Module entwickelt und getestet sind, müssen sie zur Gesamtschaltung zusammengefügt werden. Um den Ablauf in der HEPI zu steuern, wird ein Steuerwerk benötigt. Für die Entwicklung dieses Steuerwerks können spezielle Software Werkzeuge zum Einsatz kommen, welche die Entwicklung erleichtern. Hand in Hand mit der Entwicklung der Logik sollte die Karte entwickelt werden, auf der die Komponenten der HEPI aufgebaut werden.
Für die Tests der HEPI wird ein Simulator benötigt, der die HEPI mit Daten versorgt. Die Daten können hierbei vom Rechner erzeugt und über eine Schnittstelle zur HEPI übertragen werden. Die von der HEPI verarbeiteten Daten werden anschließend zum Rechner zurückgeschickt und überprüft. Ein anderer Weg die HEPI zu testen, besteht im Aufbau eines Detektor Arrays. Dieses Array registriert die Photonen einer Quelle und erzeugt damit Daten für die HEPI. Solch ein Array befindet sich am Institut in der Entwicklung [18]. Es soll in der Endversion aus 16 Pixeln bestehen. Nachdem diese Tests durchgeführt sind, wird das ASIC Design erstellt. Dabei werden die Bibliotheken des ASIC Herstellers verwendet. Bei der Umsetzung des FPGA Designs zum ASIC Design werden die in Kapitel 2 beschriebenen Schritte durchlaufen. Sind die Simulationen des ASIC Designs zur Zufriedenheit abgeschlossen, kann mit der Fertigung des ASICs begonnen werden. Nach der Fertigung des ASICs wird die Karte für das Flugmodell auf seine Funktionsfähigkeit getestet. Sind alle Tests zur Zufriedenheit abgeschlossen, kann mit dem Einbau in den Satelliten begonnen werden.

Literaturverzeichnis

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

[2] Scientific Performance Report, Issue 1, 1996

[3] Lei F., 1995, IN-IM-SOT-TN-0020, New Efficiency, Background and Sensitivity Estimates for IBIS

[4] Lei F., 1995, IN-IM-SOT-TN-0024, An Update on Efficiency, Background and Sensitivity Estimates

[5] Horrowitz, Hill, 1989, The Art of Electronics, Cambridge University Press, Cambridge

[6] Tietze, Schenk, 1993, Halbleiter-Schaltungstechnik, Springer, Berlin

[7] Xilinx, Xactstep Mentor Graphics Interface Guide

[8] EID-A, 1996 IBIS Experiment Interface Document - A, Issue 1

[9] EID-B, 1996 IBIS Experiment Interface Document - B, Issue 1

[10] Volkmer R., IN-IM-TUB-TN/EL-008, HEPI Description Electrical Interface

[11] Kendziorra E., Mohan M., Staubert R., 1995, IN-IM-TUB-TN/EL-002, Scientific Data Processing tasks for the INTEGRAL IMAGER

[12] Kendziorra E., Mohan M., Staubert R., 1995, IN-IM-TUB-TN/EL-003, Proposal for a Hardware Event Sorter for the IMAGER (ESI)

[13] Fromm A., Diplomarbeit, Juni 1995, Institut für Astronomie und Astrophysik, Abt. Astronomie, Tübingen

[14] Fishman G.J. et.al. : The First BATSE Gamma-Ray Catalog, Astrophys. J. Supp. Ser. 92 : 229-283, May 1994

[15] R. Volkmer, A. Rilk, 1996, IN-IM-TUB-RP-004, Cyclic Burst Buffer Size

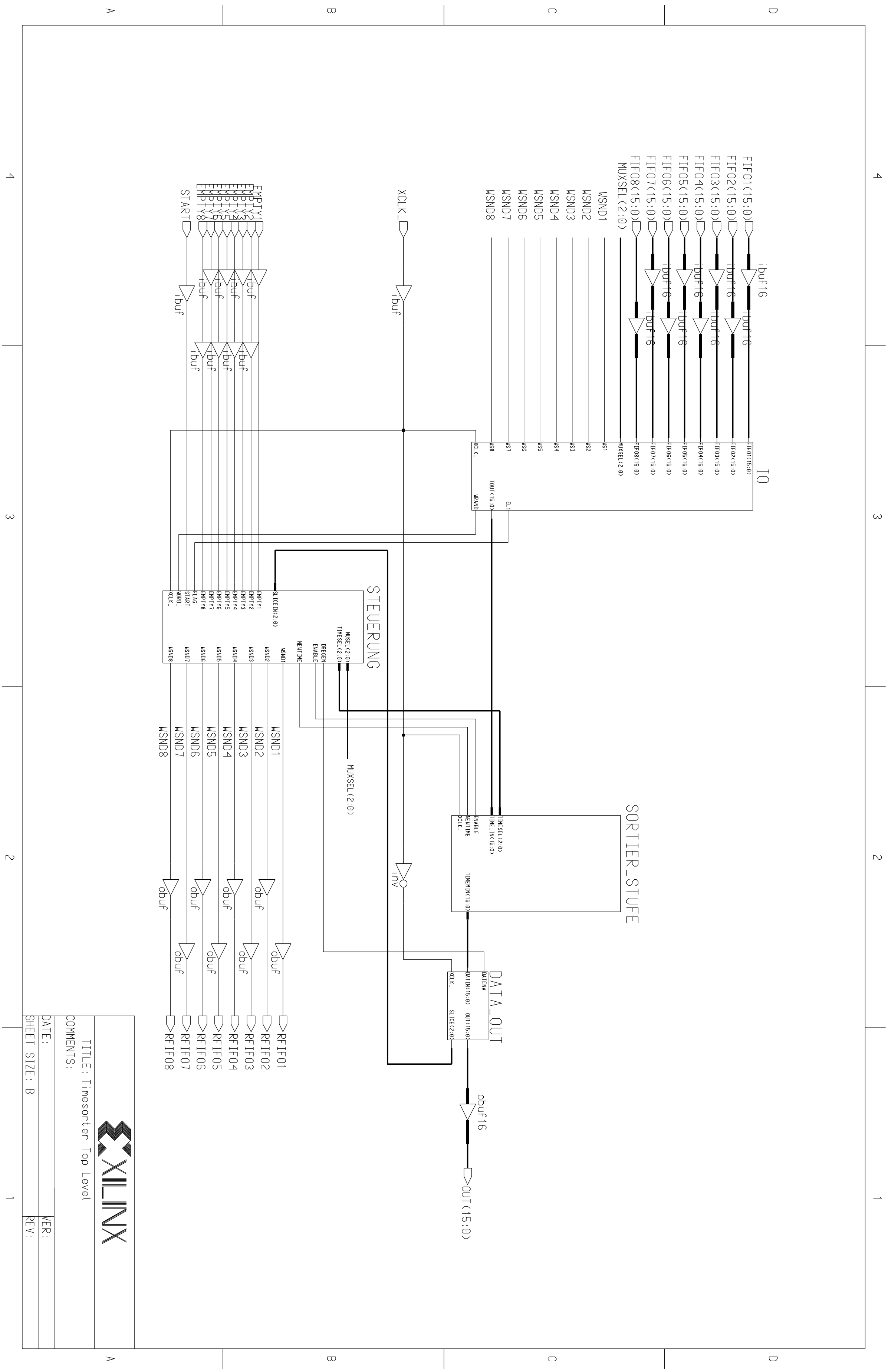
[16] Minutes of IBIS Consortium Meeting, 1996, Rom

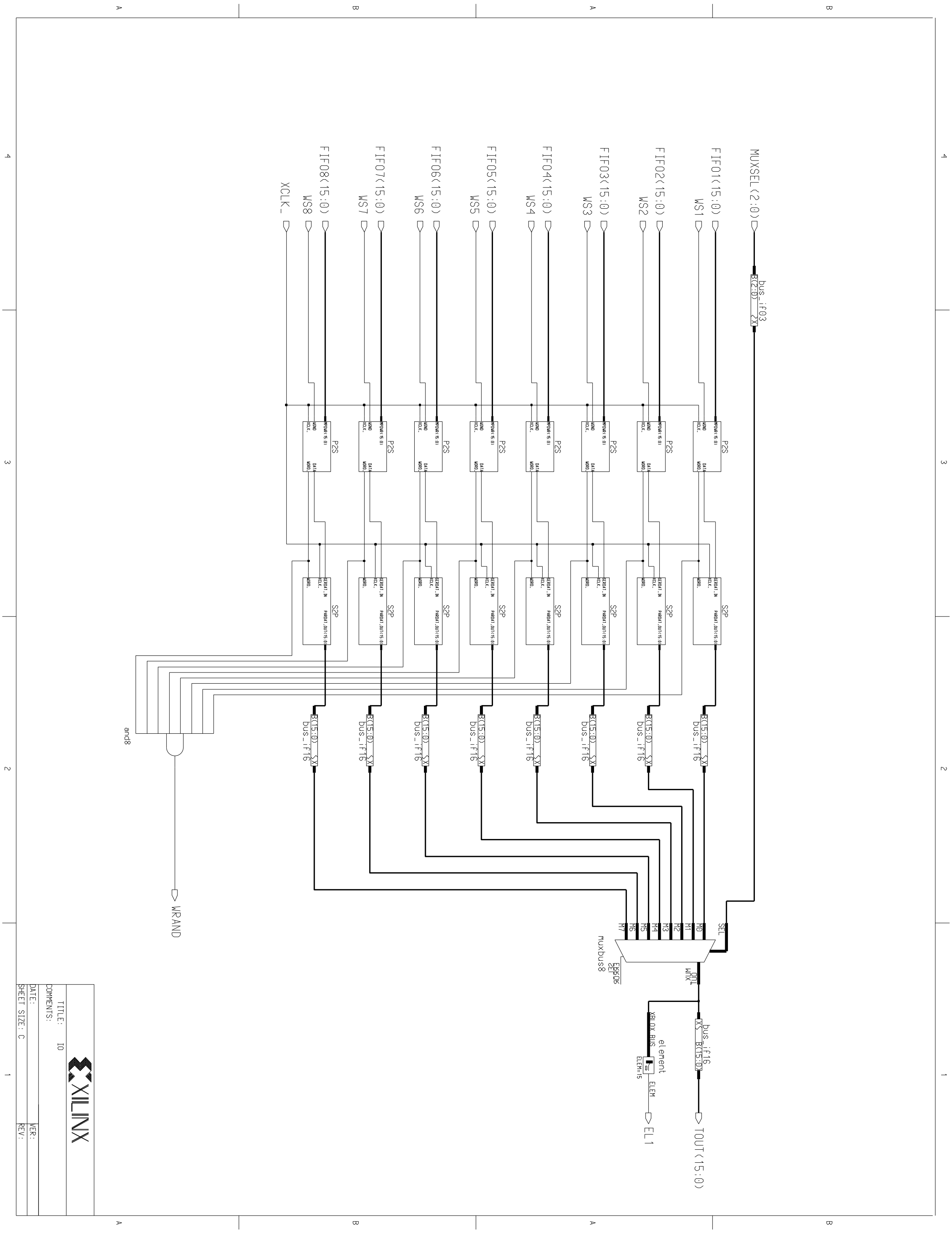
[17] G. Malaguti and G. DI Coco, April 1996, PICsIT multiple event handling, Technical report, TESRE C.N.R., Bologna

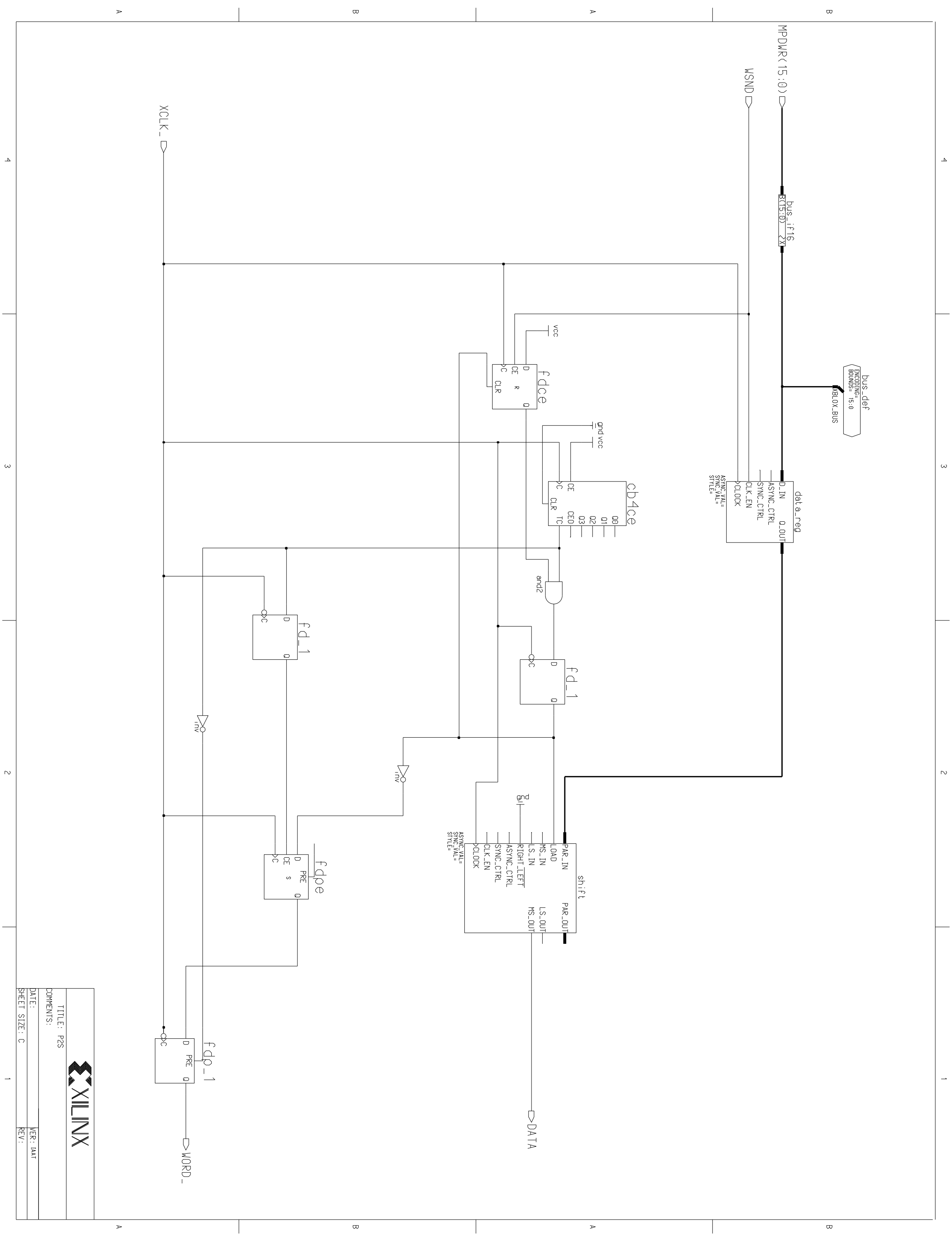
[18] A. Rilk, Diplomarbeit,1996, Institut für Astronomie und Astrophysik, Abt. Astronomie, Tübingen

8 Anhang A

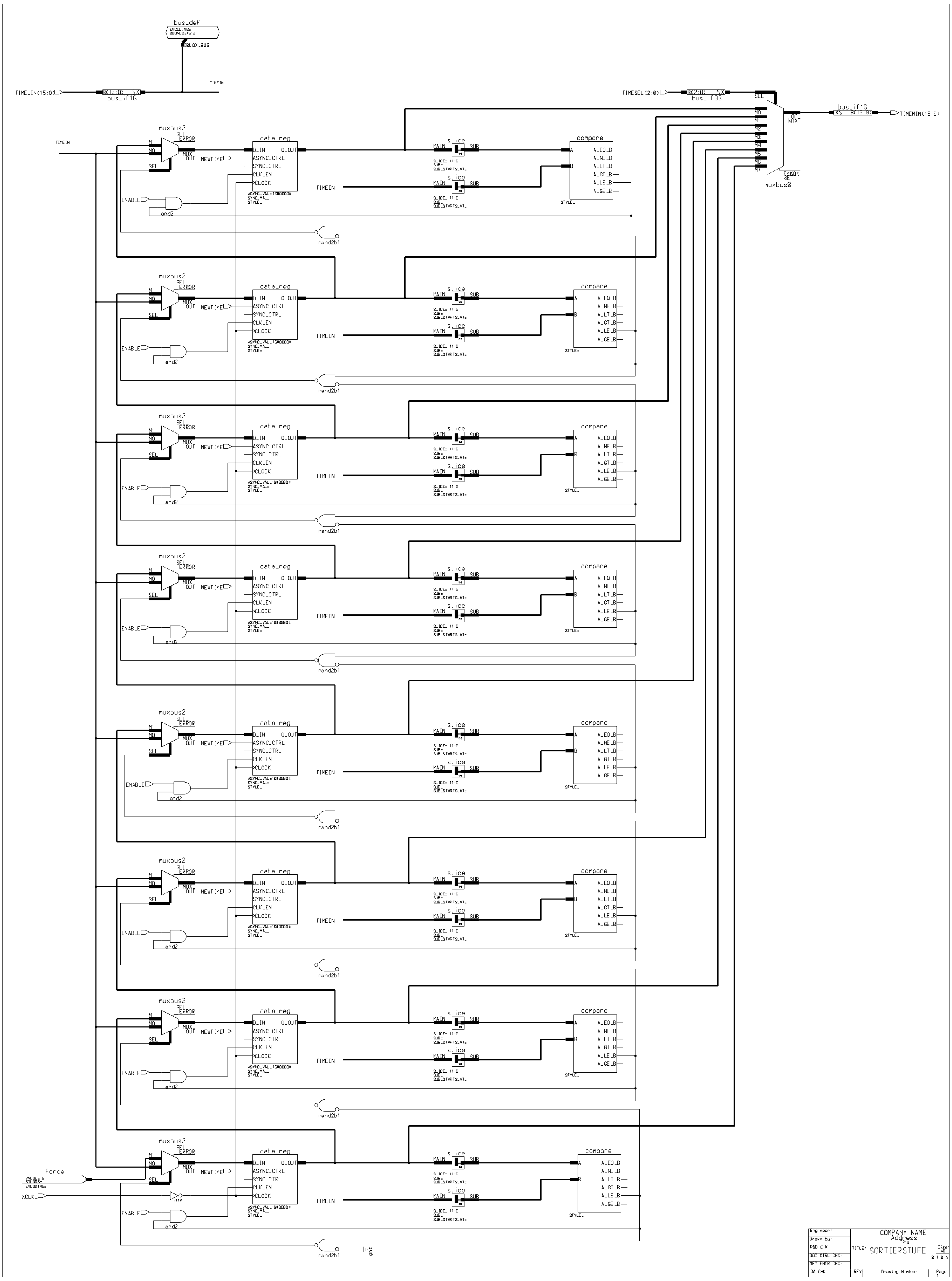
8.1 Schaltplan und Simulation des Timesorters

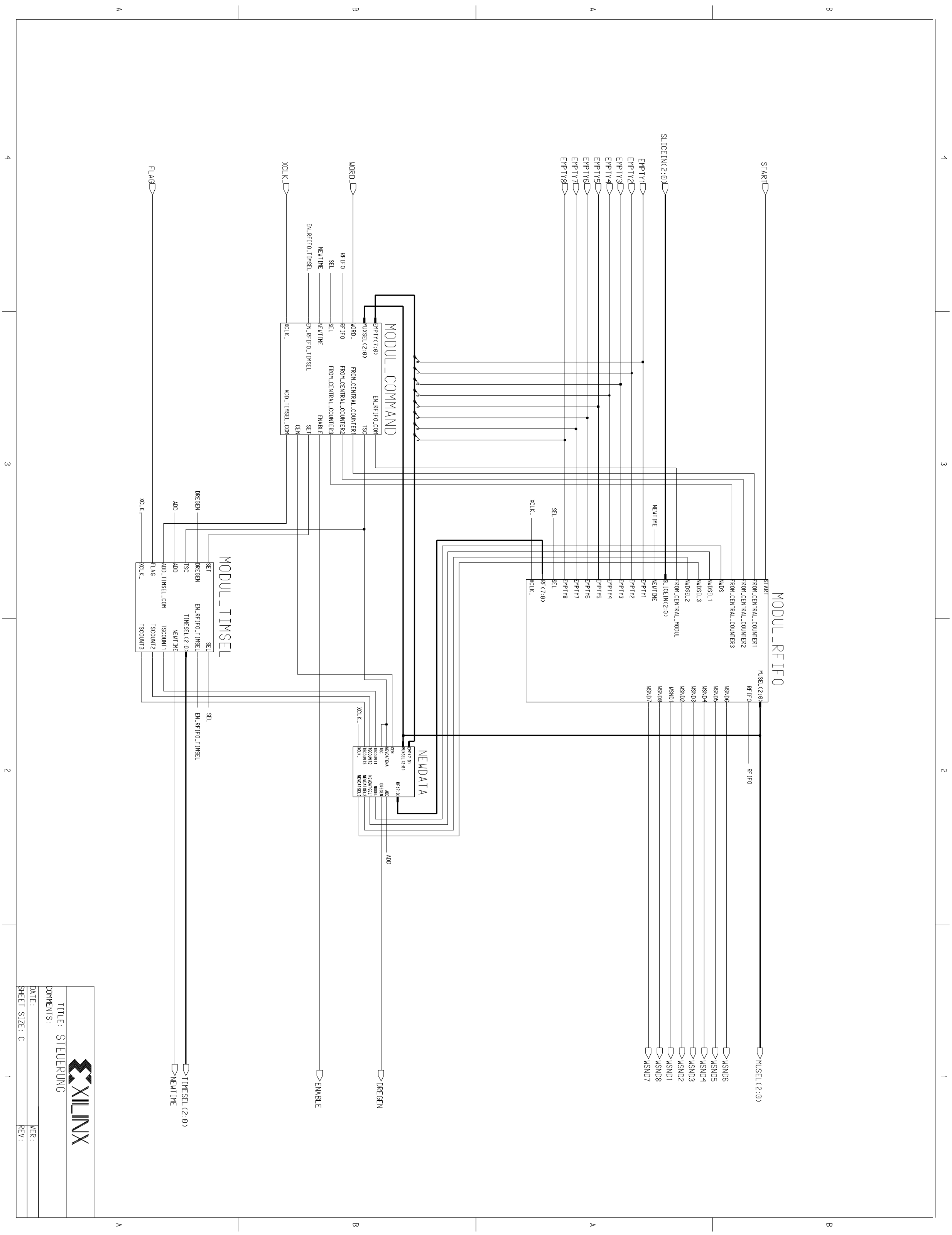

IO
STEUERUNG
SORTIER_STUFE
DATA_OUT
XILINX
TITLE: Timesorter Top Level
COMMENTS:
DATE:
SHEET SIZE: B
VER:
REV:

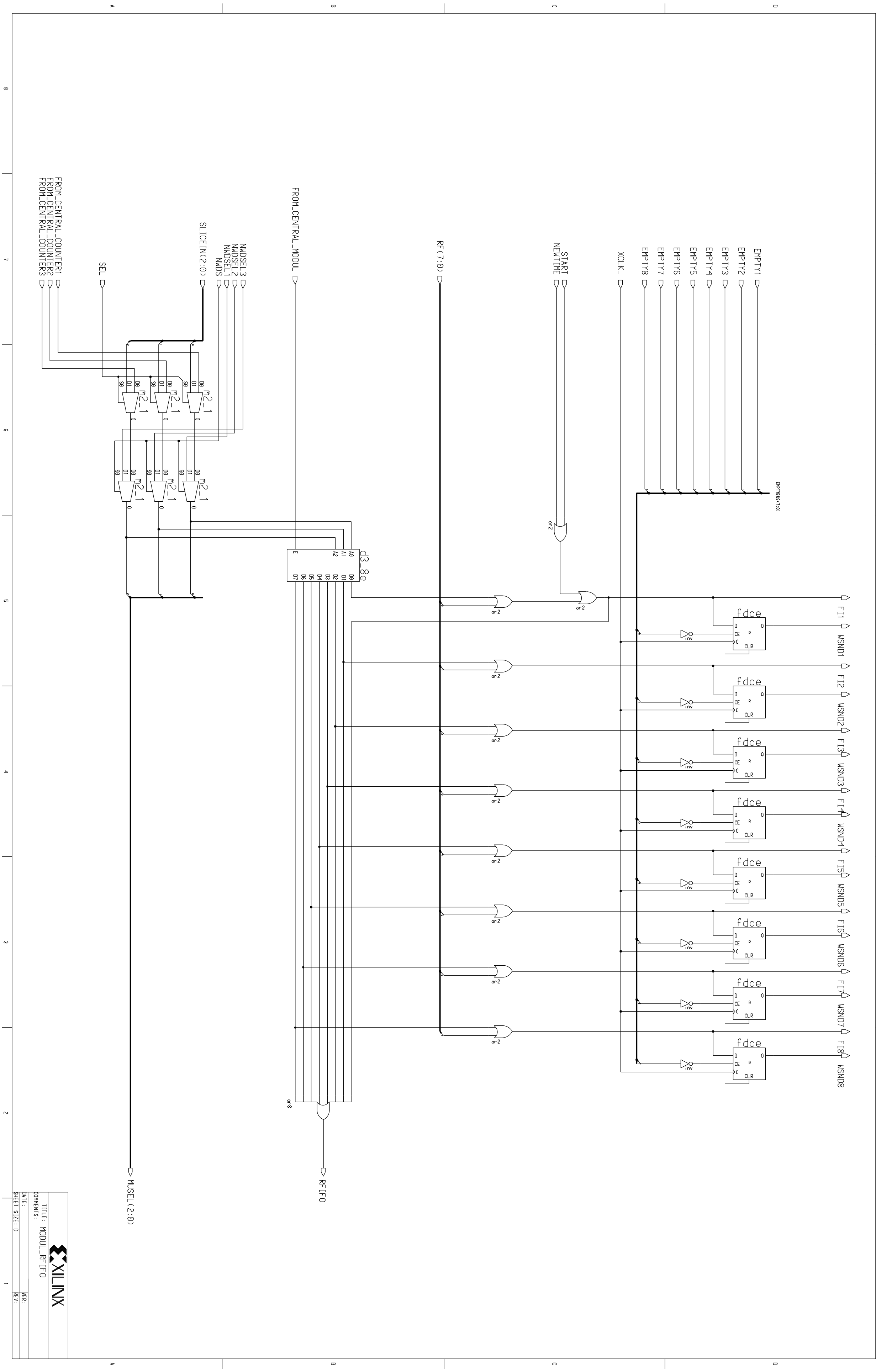

MUXSEL(2:0)
FIFO1(15:0)
WS1
FIFO2(15:0)
WS2
FIFO3(15:0)
WS3
FIFO4(15:0)
WS4
FIFO5(15:0)
WS5
FIFO6(15:0)
WS6
FIFO7(15:0)
WS7
FIFO8(15:0)
WS8
XCLK_
bus_if03
P2S
S2P
bus_if16
and8
WRAND
muxbus8
SEL
M0
M1
M2
M3
M4
M5
M6
M7
element
ELEM
TOUT(15:0)
EL1
XILINX
TITLE: IO
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:

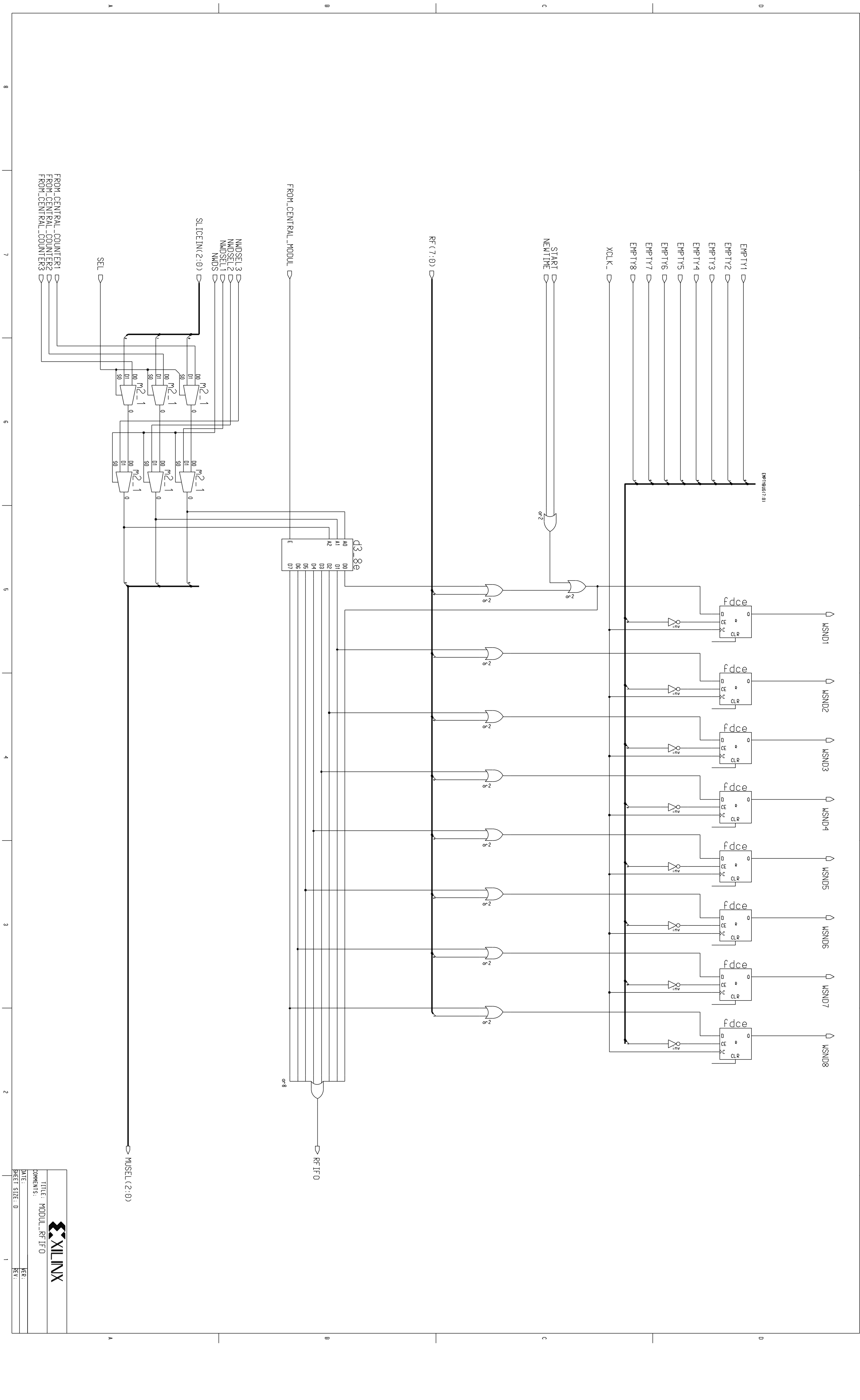

XCLK_
WSND
MPDWR(15:0)
bus_if16
bus_def
data_reg
fdce
vcc
gnd
cb4ce
and2
fd_1
inv
fdpe
shift
fdp_1
WORD_
DATA
TITLE: P2S
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:
XILINX

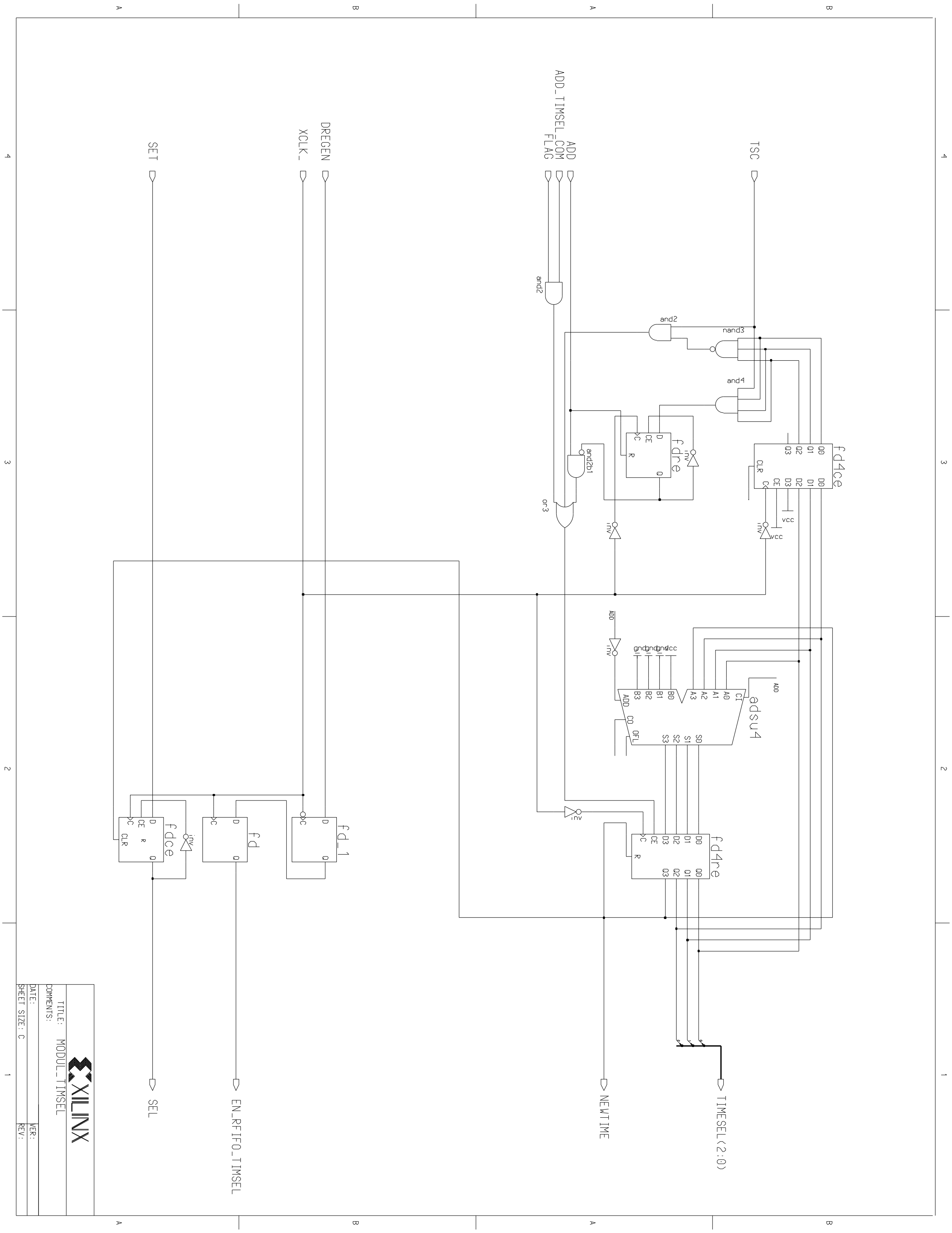
SERDAT_IN
XCLK_
WORD_
inv
inv
shift
PAR_IN
LOAD
MS_IN
LS_IN
RIGHT_LEFT
ASYNC_CTRL
SYNC_CTRL
CLK_EN
CLOCK
ASYNC_VAL=
SYNC_VAL=
STYLE=
PAR_OUT
LS_OUT
MS_OUT
bus_def
ENCODING= UBIN
BOUNDS= 15:0
XBLOX_BUS
data_reg
D_IN
ASYNC_CTRL
SYNC_CTRL
CLK_EN
CLOCK
ASYNC_VAL=
SYNC_VAL=
STYLE=
Q_OUT
bus_if16
XS__B(15:0)
PARDAT_OUT(15:0)
XILINX
TITLE: S2P
COMMENTS:
DATE:
SHEET SIZE: B
VER:
REV:

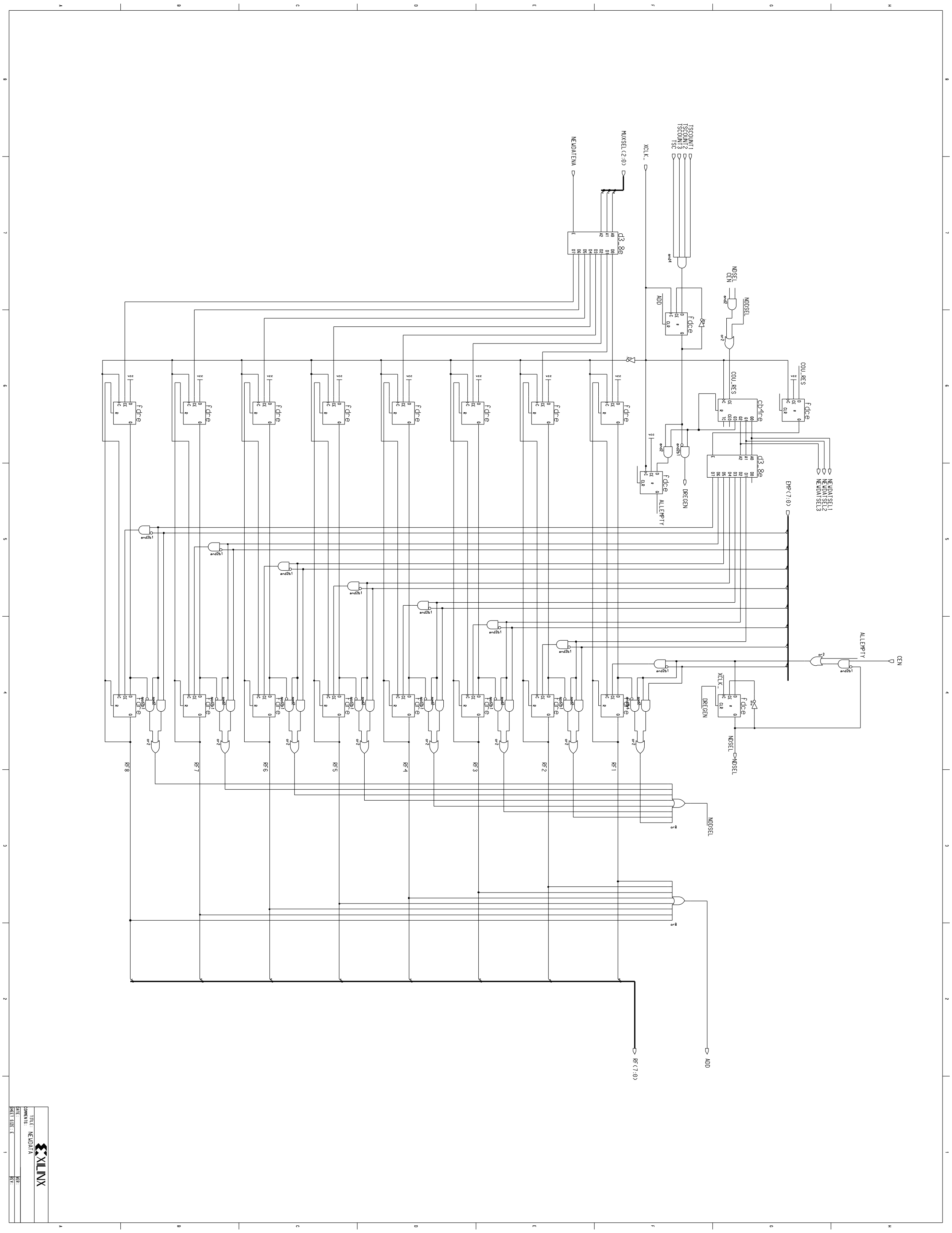

bus_def
TIME_IN(15:0)
bus_if16
TIMESEL(2:0)
bus_if03
TIMEMIN(15:0)
muxbus8
muxbus2
data_reg
slice
compare
nand2b1
and2
force
XCLK
COMPANY NAME
Address
SORTIERSTUFE

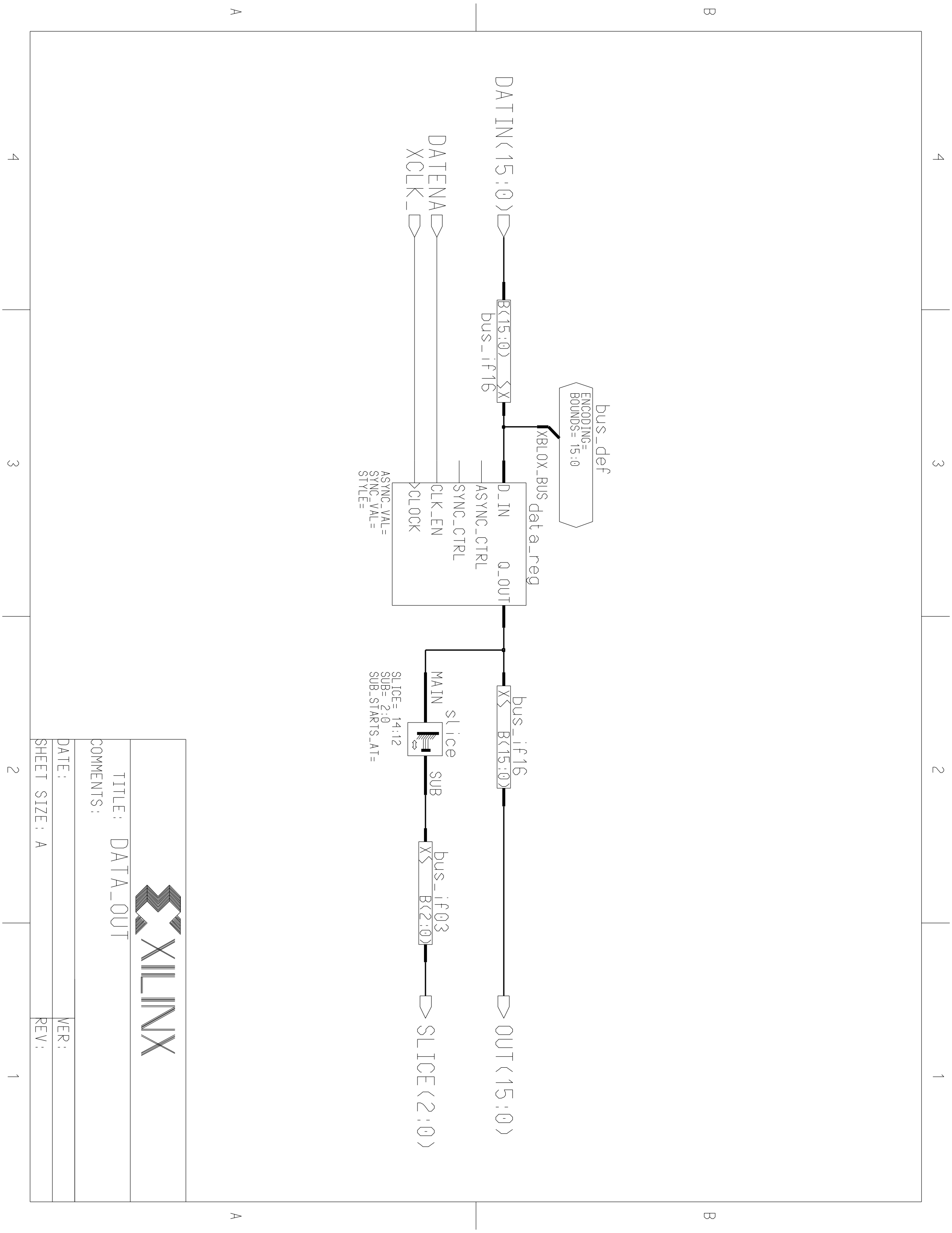

MODUL_COMMAND
MODUL_TIMSEL
MODUL_RFIFO
NEWDATA
XILINX
TITLE: STEUERUNG
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:


FROM_CENTRAL_COUNTER1
FROM_CENTRAL_COUNTER2
FROM_CENTRAL_COUNTER3
SEL
SLICEIN(2:0)
NWDS
NWDSEL1
NWDSEL2
NWDSEL3
FROM_CENTRAL_MODUL
RF(7:0)
START
NEWTIME
XCLK_
EMPTY8
EMPTY7
EMPTY6
EMPTY5
EMPTY4
EMPTY3
EMPTY2
EMPTY1
m2_1
d3_8e
or2
or8
inv
fdce
FI1
WSND1
FI2
WSND2
FI3
WSND3
FI4
WSND4
FI5
WSND5
FI6
WSND6
FI7
WSND7
FI8
WSND8
MUSEL(2:0)
RFIFO
XILINX
TITLE: MODUL_RFIFO
COMMENTS:
DATE:
SHEET SIZE: D
VER:
REV:

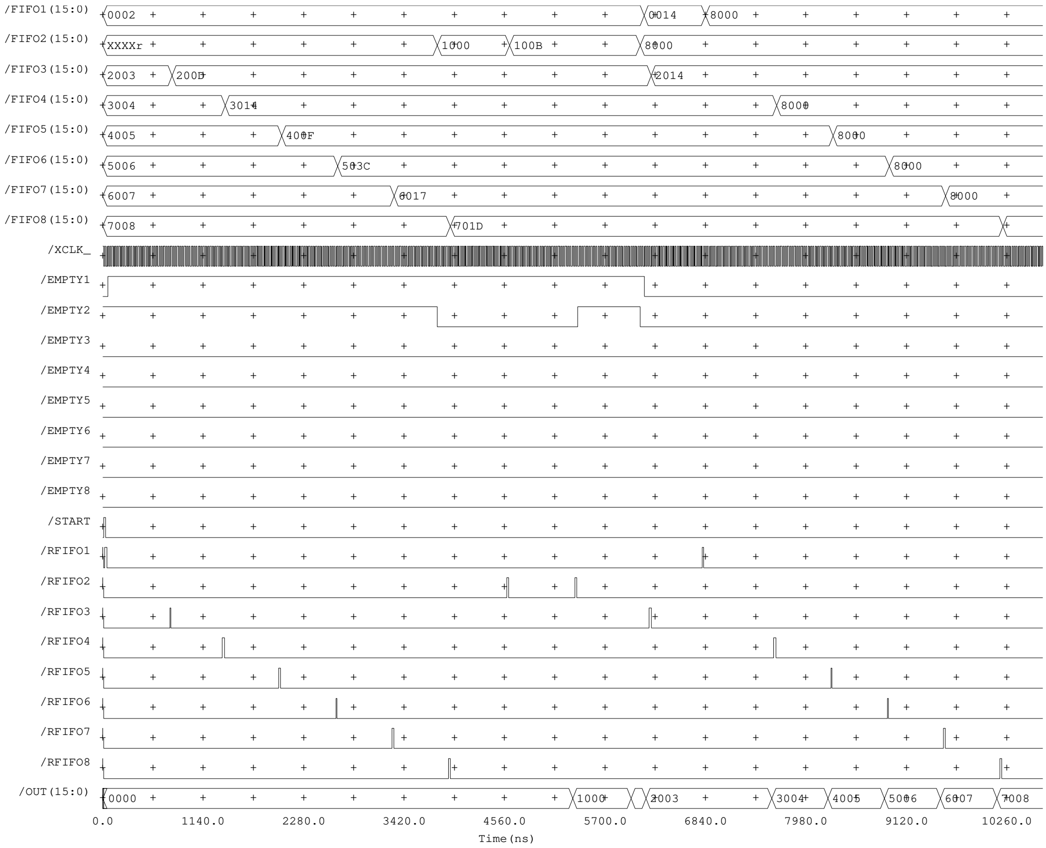

FROM_CENTRAL_COUNTER1
FROM_CENTRAL_COUNTER2
FROM_CENTRAL_COUNTER3
SEL
SLICEIN(2:0)
NWDSEL3
NWDSEL2
NWDSEL1
NWDS
FROM_CENTRAL_MODUL
RF(7:0)
START
NEWTIME
XCLK_
EMPTY8
EMPTY7
EMPTY6
EMPTY5
EMPTY4
EMPTY3
EMPTY2
EMPTY1
m2_1
d3_8e
or2
or8
inv
fdce
WSND1
WSND2
WSND3
WSND4
WSND5
WSND6
WSND7
WSND8
MUSEL(2:0)
RFIFO
XILINX
TITLE: MODUL_RFIFO
COMMENTS:
DATE:
SHEET SIZE: D
VER:
REV:

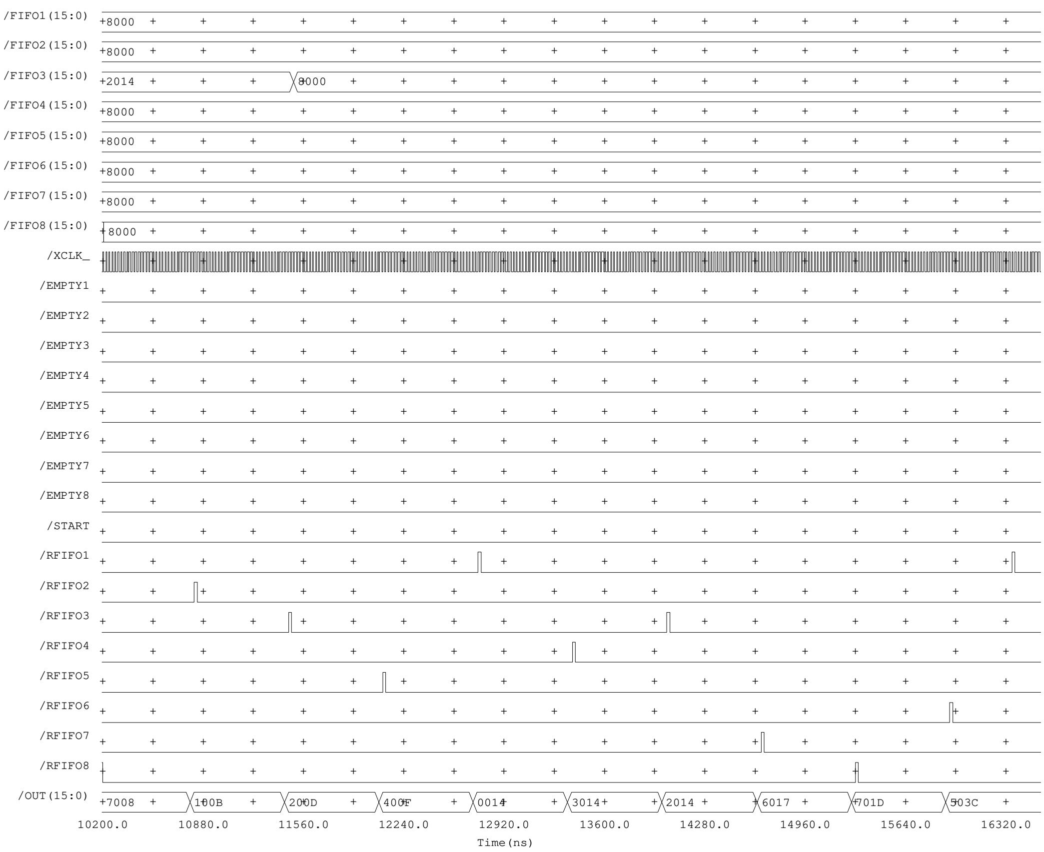

SET
XCLK_
DREGEN
ADD_TIMSEL_COM
FLAG
ADD
TSC
and2
nand3
and4
fdre
and2b1
or3
fd4ce
adsu4
fd4re
fdce
fd
fd_1
SEL
EN_RFIFO_TIMSEL
NEWTIME
TIMESEL(2:0)
XILINX
TITLE: MODUL_TIMSEL
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:


NEWDATENA
MUXSEL<2:0>
XCLK
ADD
NOSEL
CEN
ALLEMPTY
DREGEN
RF<7:0>
XILINX
TITLE: NEWDATA



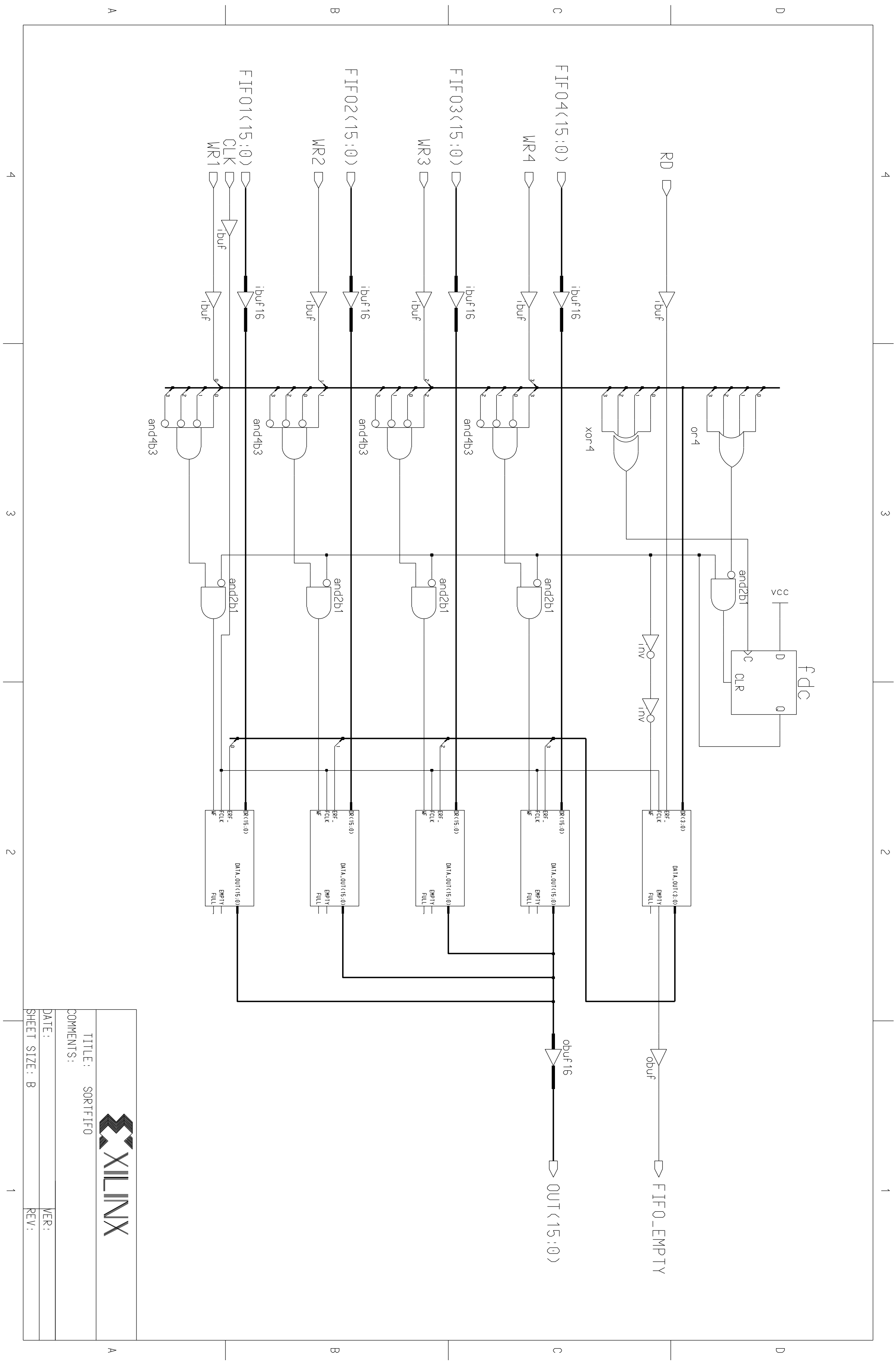
DATIN(15:0)
B(15:0) SX
bus_if16
bus_def
ENCODING=
BOUNDS= 15:0
XBLOX_BUS
data_reg
D_IN
Q_OUT
ASYNC_CTRL
SYNC_CTRL
CLK_EN
CLOCK
DATENA
XCLK_
ASYNC_VAL=
SYNC_VAL=
STYLE=
bus_if16
XS B(15:0)
OUT(15:0)
slice
MAIN
SUB
SLICE= 14:12
SUB= 2:0
SUB_STARTS_AT=
bus_if03
XS B(2:0)
SLICE(2:0)
XILINX
TITLE: DATA_OUT
COMMENTS:
DATE:
VER:
SHEET SIZE: A
REV:

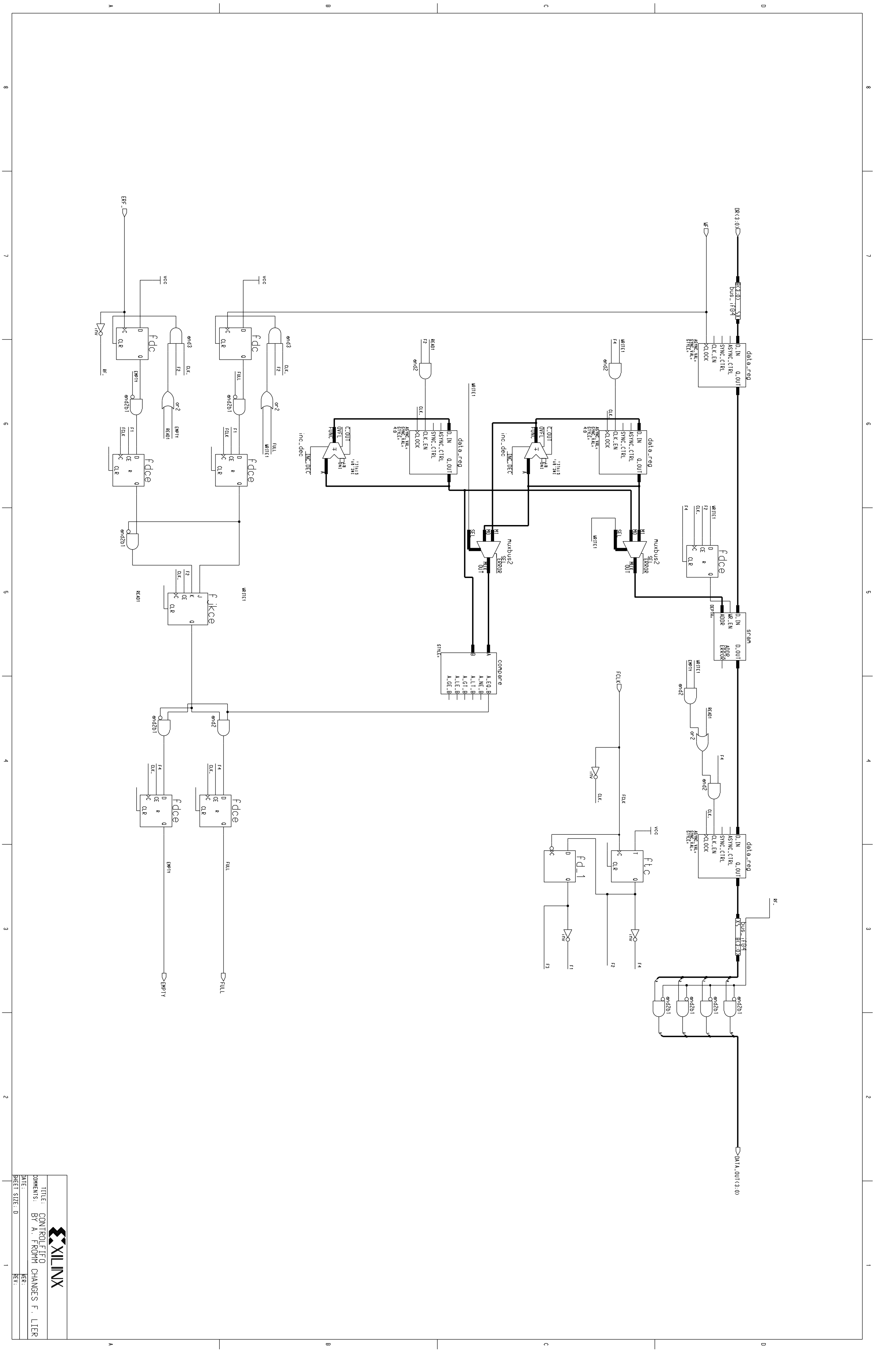


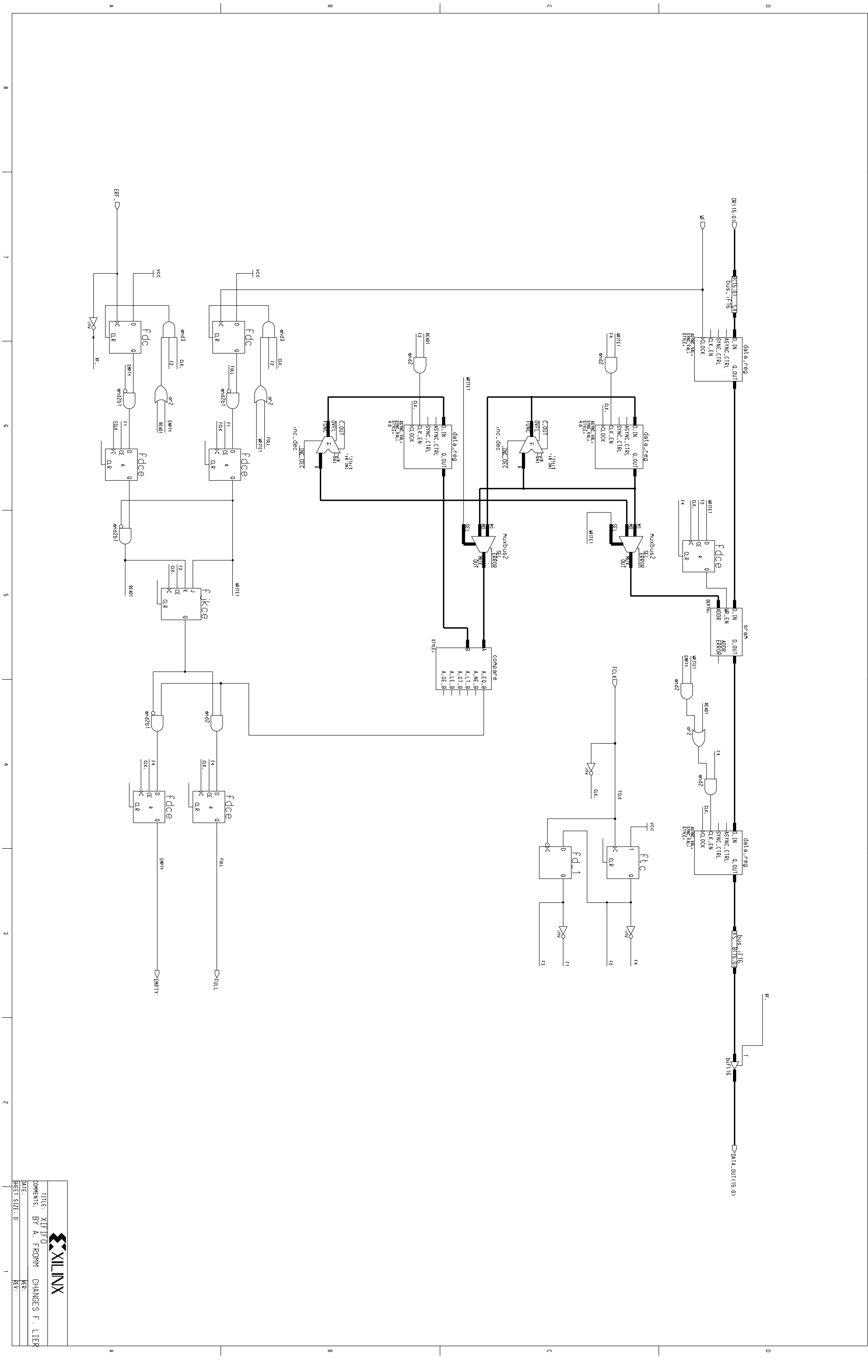


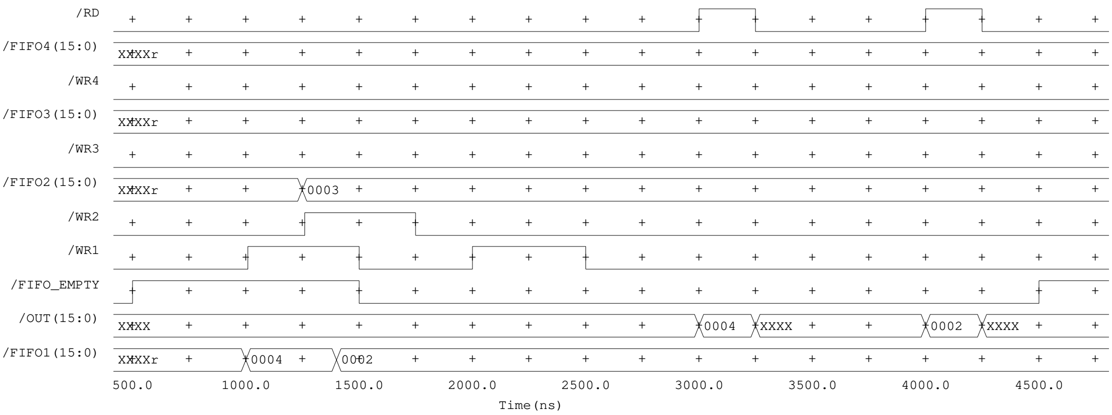
9 Anhang B

9.1 Schaltplan und Simulation des Fifosorters


FIFO1(15:0)
CLK
WR1
FIFO2(15:0)
WR2
FIFO3(15:0)
WR3
FIFO4(15:0)
WR4
RD
ibuf
ibuf16
and4b3
xor4
or4
and2b1
vcc
fdc
inv
obuf16
obuf
OUT(15:0)
FIFO_EMPTY
XILINX
TITLE: SORTFIFO
COMMENTS:
DATE:
SHEET SIZE: B
VER:
REV:

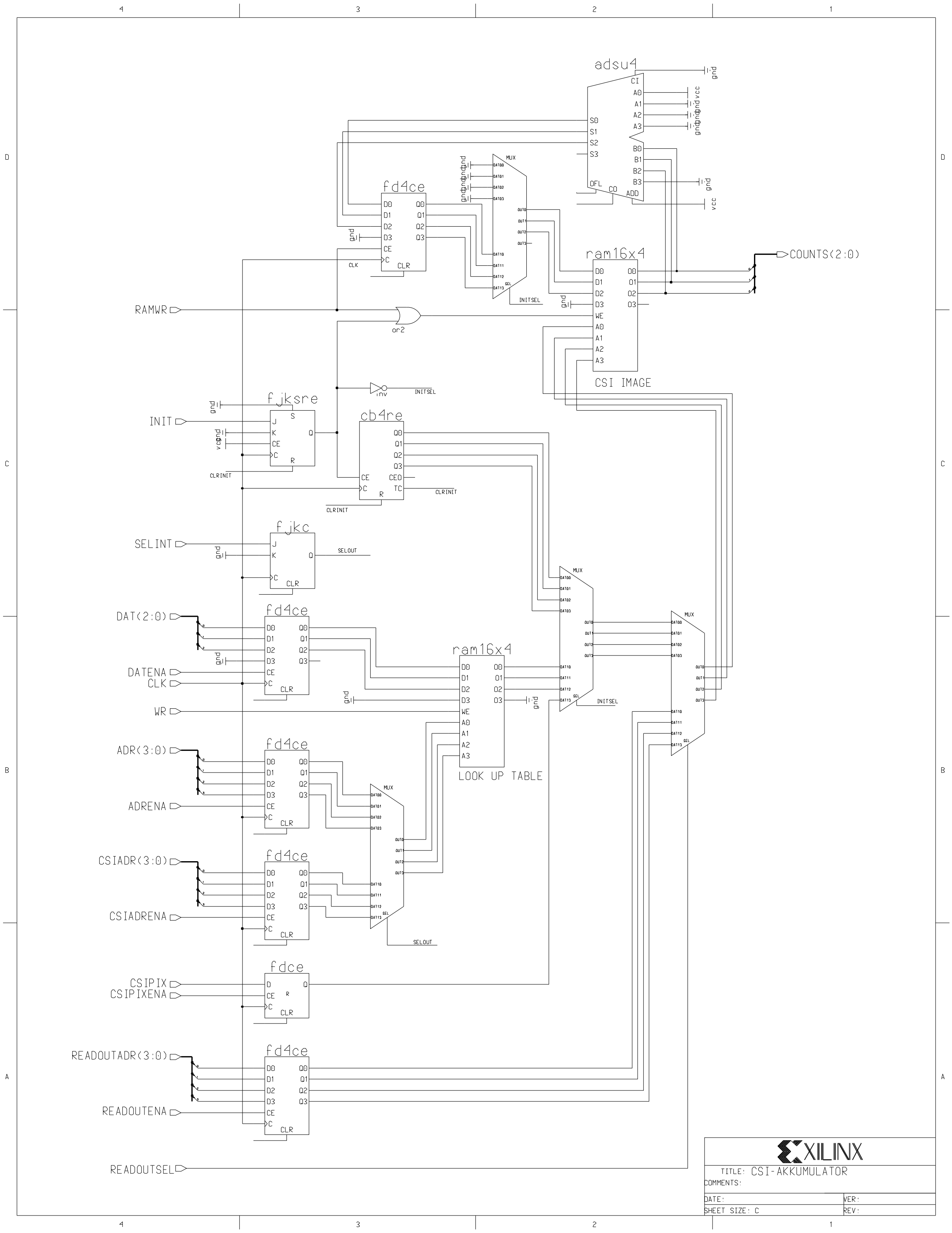

XILINX
TITLE: CONTROLFIFO
COMMENTS: BY A. FROMM CHANGES F. LIER
DATE:
SHEET SIZE: D
VER:
REV:

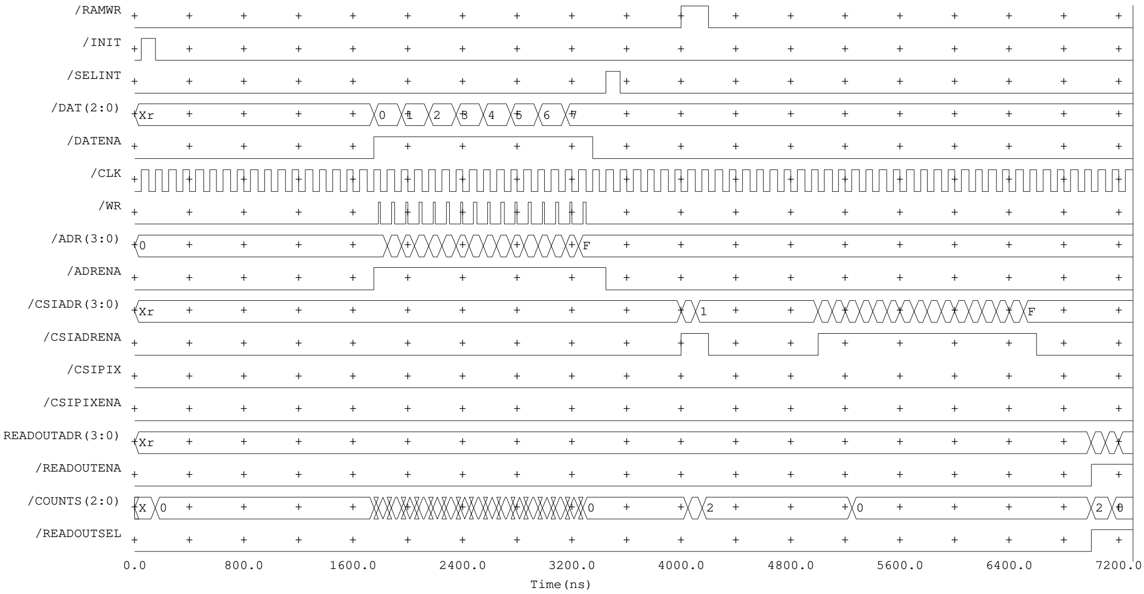

TITLE: XIFIFO
COMMENTS: BY A. FROMM CHANGES F. LIER
DATE:
SHEET SIZE: D
VER:
REV:


/RD
/FIFO4(15:0)
XXXXr
/WR4
/FIFO3(15:0)
XXXXr
/WR3
/FIFO2(15:0)
XXXXr
0003
/WR2
/WR1
/FIFO_EMPTY
/OUT(15:0)
XXXX
0004
XXXX
0002
XXXX
/FIFO1(15:0)
XXXXr
0004
0002
500.0
1000.0
1500.0
2000.0
2500.0
3000.0
3500.0
4000.0
4500.0
Time(ns)

10 Anhang C

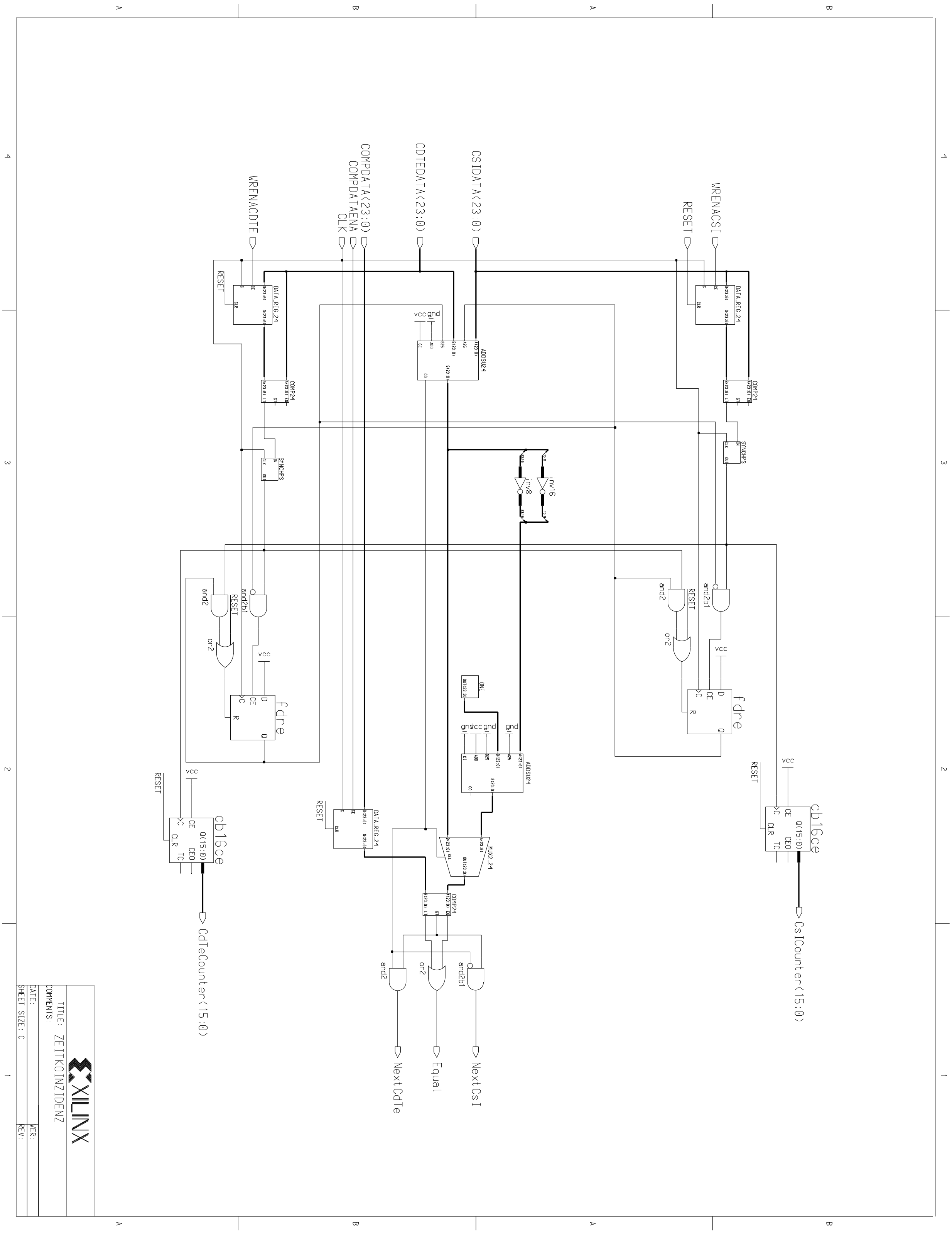
10.1 Schaltplan und Simulation des CsI-Akkumulators

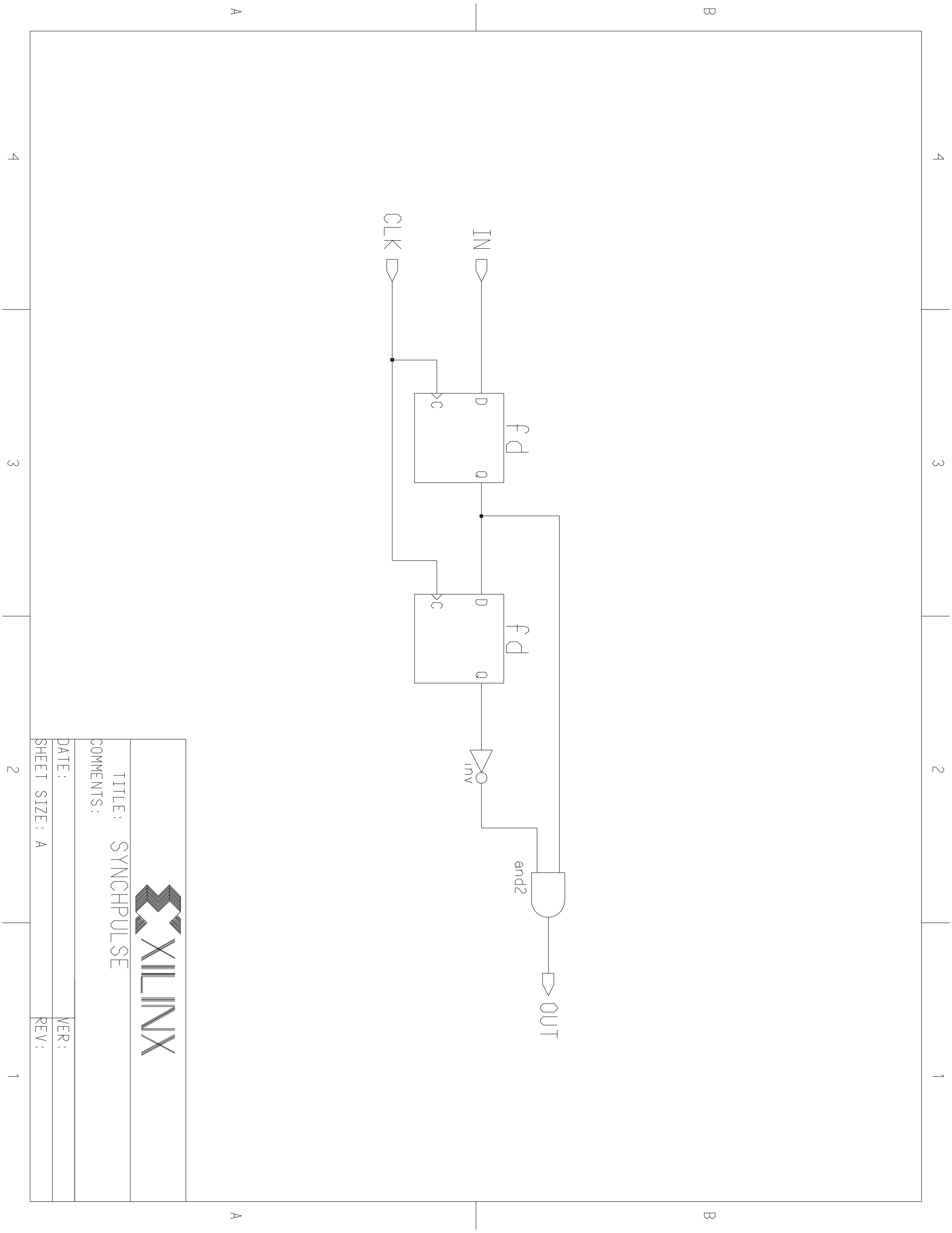

adsu4
ram16x4
CSI IMAGE
fd4ce
fjksre
cb4re
fjkc
fdce
LOOK UP TABLE
COUNTS(2:0)
RAMWR
INIT
SELINT
DAT(2:0)
DATENA
CLK
WR
ADR(3:0)
ADRENA
CSIADR(3:0)
CSIADRENA
CSIPIX
CSIPIXENA
READOUTADR(3:0)
READOUTENA
READOUTSEL
INITSEL
SELOUT
CLRINIT
XILINX
TITLE: CSI-AKKUMULATOR
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:

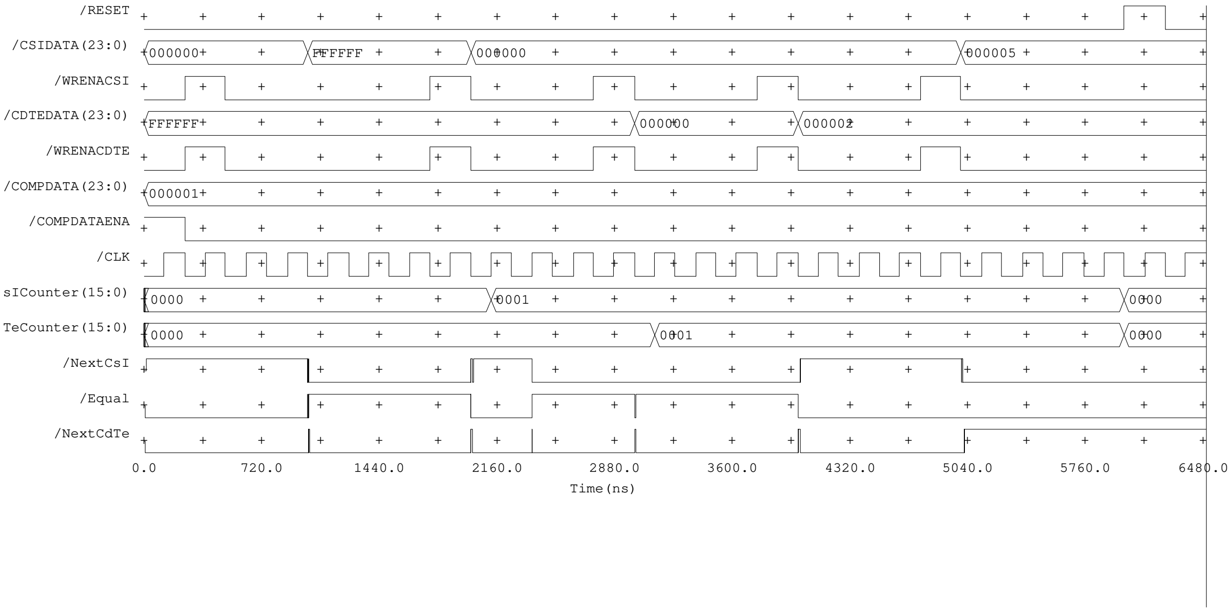

/RAMWR
/INIT
/SELINT
/DAT(2:0)
/DATENA
/CLK
/WR
/ADR(3:0)
/ADRENA
/CSIADR(3:0)
/CSIADRENA
/CSIPIX
/CSIPIXENA
READOUTADR(3:0)
/READOUTENA
/COUNTS(2:0)
/READOUTSEL
0.0
800.0
1600.0
2400.0
3200.0
4000.0
4800.0
5600.0
6400.0
7200.0
Time(ns)

11 Anhang D

11.1 Schaltplan und Simulation des Zeitkoinzidenz Moduls

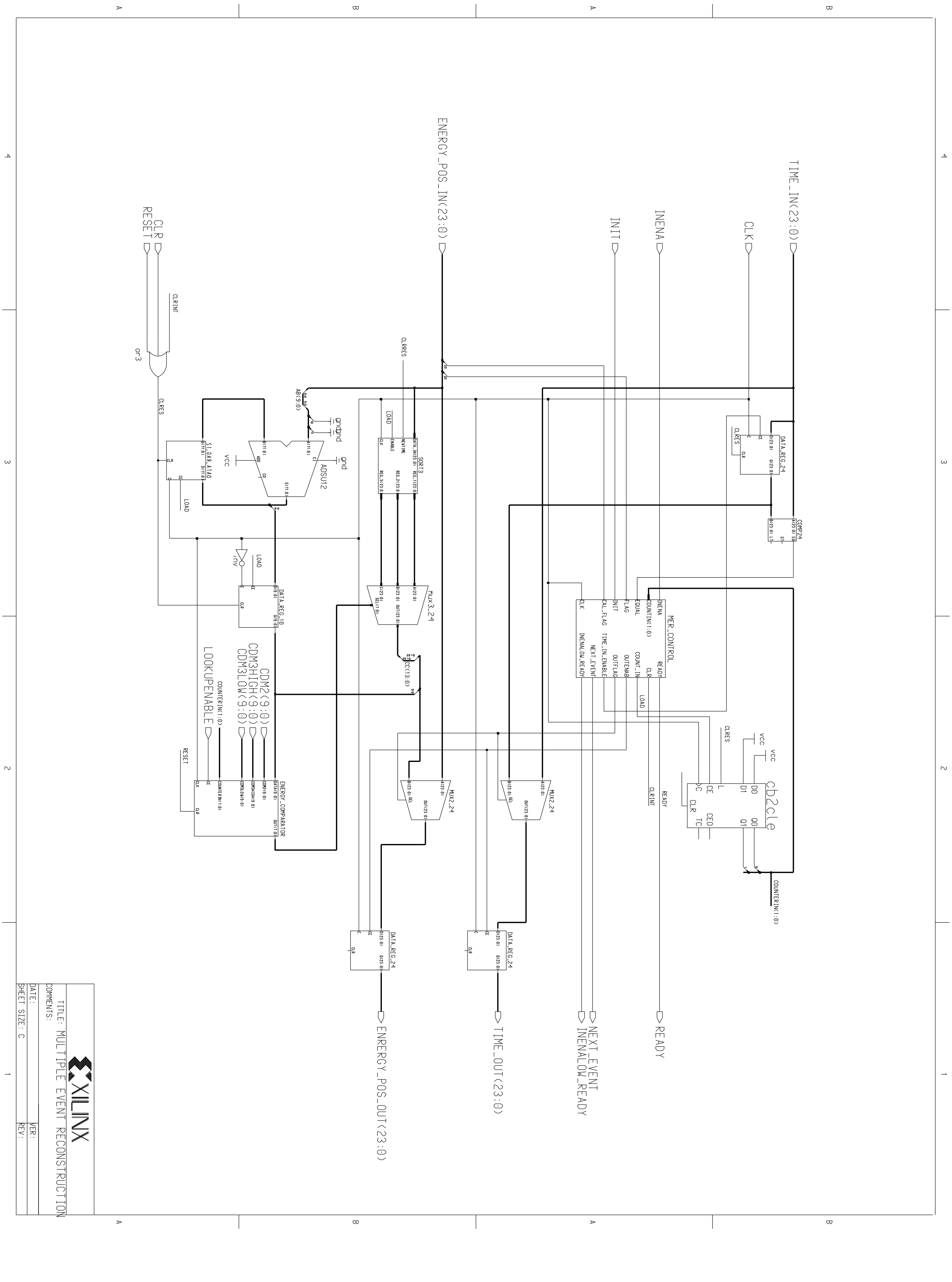

WRENACDTE
COMPDATA(23:0)
COMPDATAENA
CLK
CDTEDATA(23:0)
CSIDATA(23:0)
RESET
WRENACSI
DATA_REG_24
COMP24
ADDSU24
SYNCHPS
inv8
inv16
and2
and2b1
or2
fdre
ONE
MUX2_24
cb16ce
vcc
gnd
CdTeCounter(15:0)
CsICounter(15:0)
NextCdTe
Equal
NextCsI
TITLE: ZEITKOINZIDENZ
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:

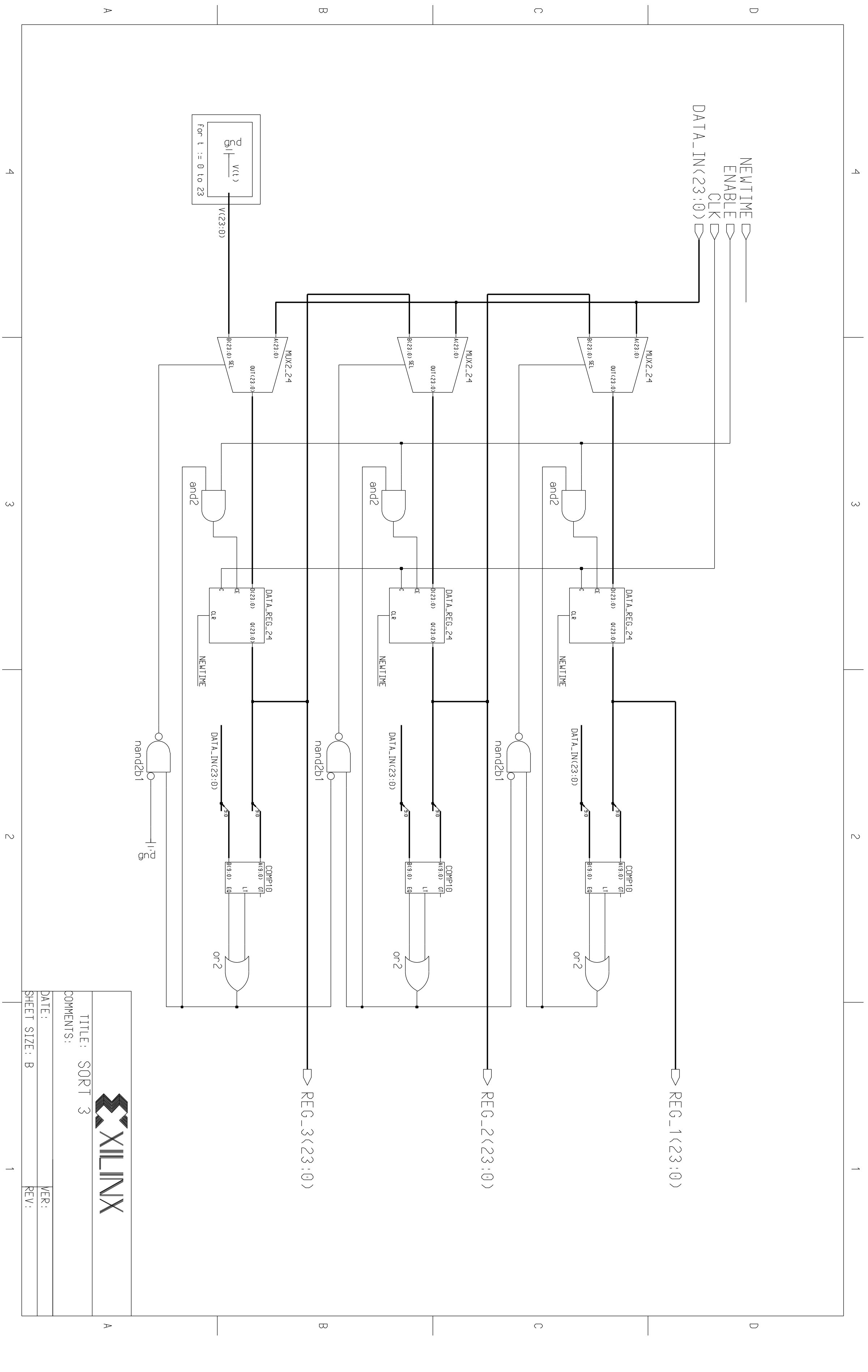

CLK
IN
C
D
fd
Q
C
D
fd
Q
inv
and2
OUT
TITLE: SYNCHPULSE
COMMENTS:
DATE:
SHEET SIZE: A
VER:
REV:
A
B
4
3
2
1

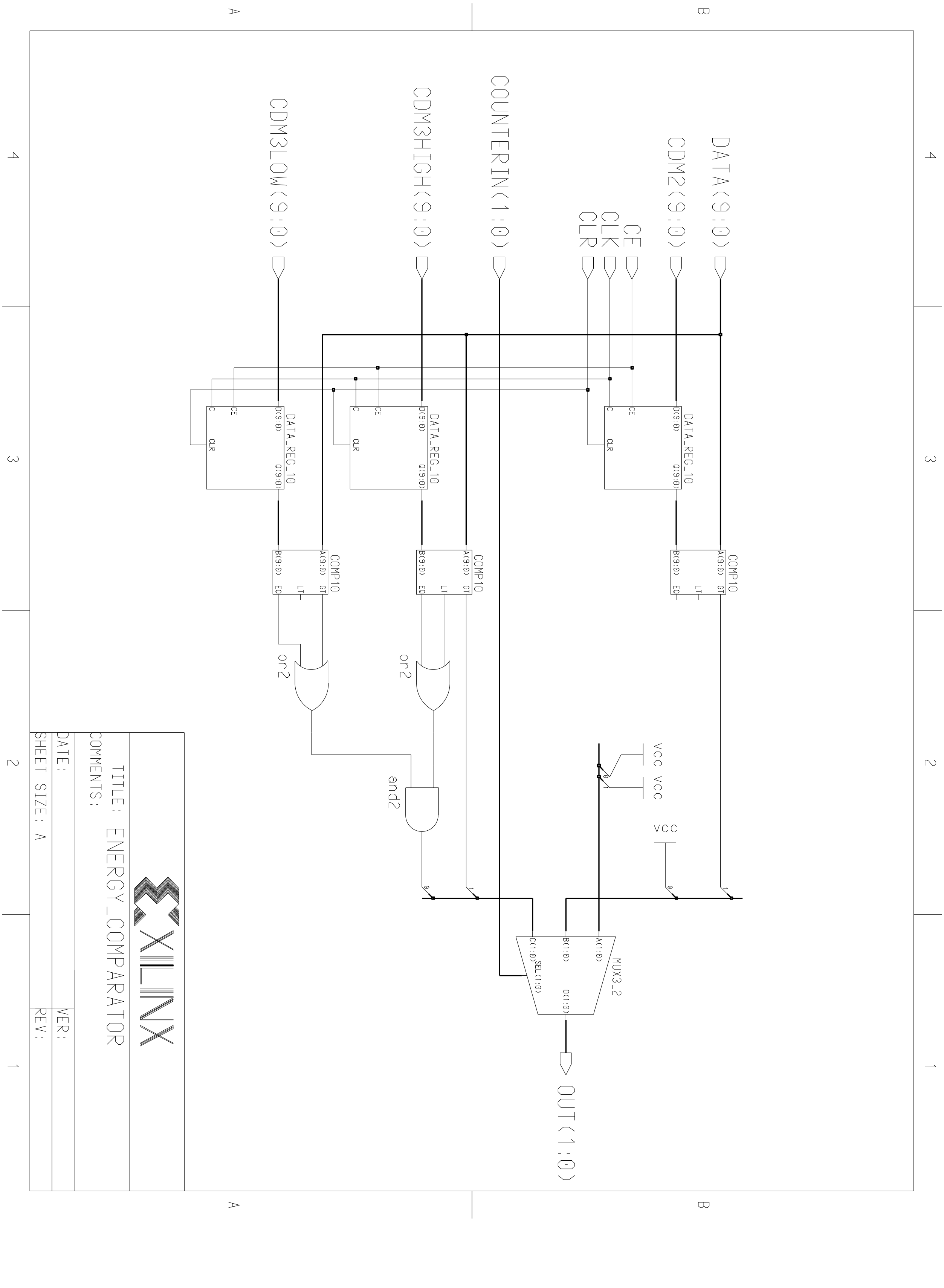

/RESET
/CSIDATA(23:0)
000000
FFFFFF
000000
000005
/WRENACSI
/CDTEDATA(23:0)
FFFFFF
000000
000002
/WRENACDTE
/COMPDATA(23:0)
000001
/COMPDATAENA
/CLK
sICounter(15:0)
0000
0001
0000
TeCounter(15:0)
0000
0001
0000
/NextCsI
/Equal
/NextCdTe
0.0
720.0
1440.0
2160.0
2880.0
3600.0
4320.0
5040.0
5760.0
6480.0
Time(ns)

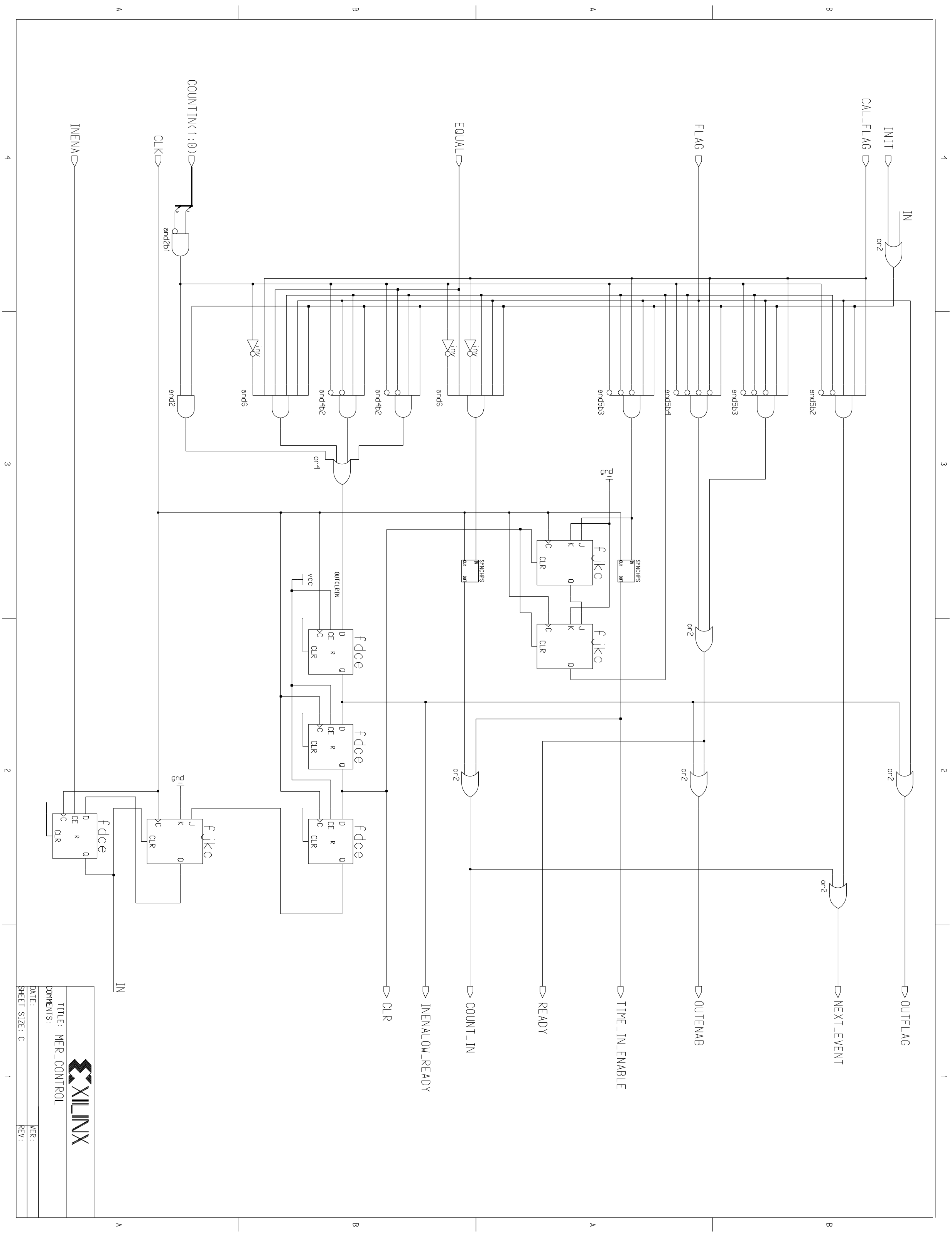
12 Anhang E

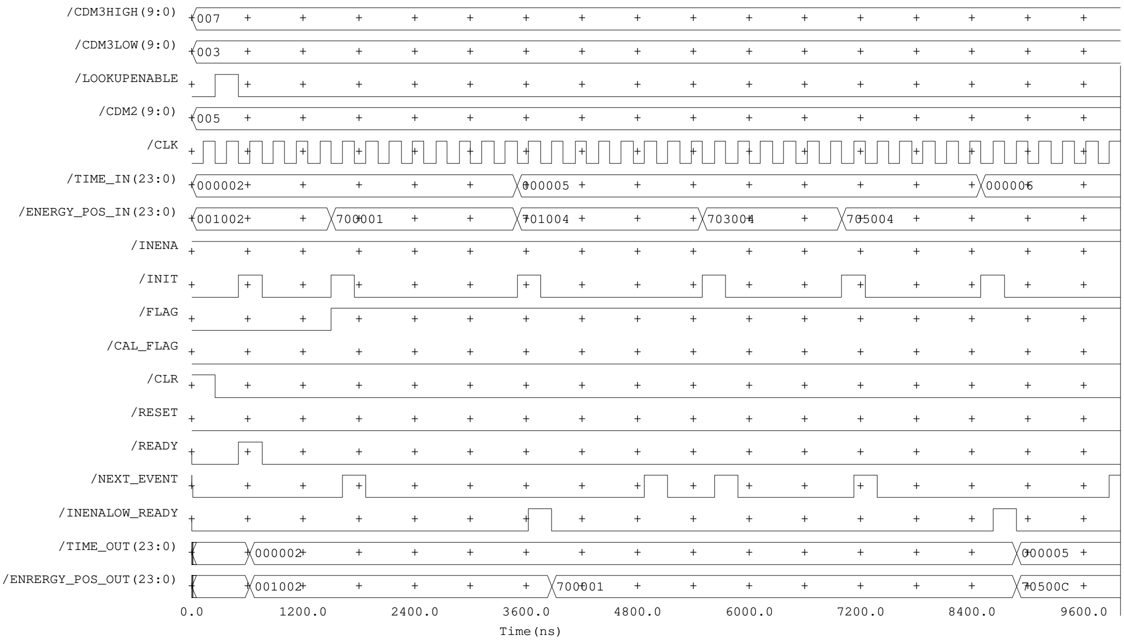
12.1 Schaltplan und Simulation der Multiple Event Reconstruction


ENERGY_POS_IN(23:0)
TIME_IN(23:0)
CLK
INENA
INIT
CLR
RESET
CDM2(9:0)
CDM3HIGH(9:0)
CDM3LOW(9:0)
LOOKUPENABLE
MER_CONTROL
cb2cle
ENERGY_POS_OUT(23:0)
TIME_OUT(23:0)
NEXT_EVENT
INENALOW_READY
READY
XILINX
TITLE: MULTIPLE EVENT RECONSTRUCTION
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:


NEWTIME
ENABLE
CLK
DATA_IN(23:0)
gnd
V(t)
for t := 0 to 23
V(23:0)
MUX2_24
and2
DATA_REG_24
NEWTIME
nand2b1
DATA_IN(23:0)
COMP10
or2
REG_3(23:0)
REG_2(23:0)
REG_1(23:0)
XILINX
TITLE: SORT 3
COMMENTS:
DATE:
SHEET SIZE: B
VER:
REV:


CDM3LOW(9:0)
CDM3HIGH(9:0)
COUNTERIN(1:0)
CLR
CLK
CE
CDM2(9:0)
DATA(9:0)
DATA_REG_10
COMP10
or2
and2
VCC
MUX3_2
OUT(1:0)
TITLE: ENERGY_COMPARATOR
COMMENTS:
DATE:
SHEET SIZE: A
VER:
REV:


INENA
CLK
COUNTIN(1:0)
EQUAL
FLAG
CAL_FLAG
INIT
IN
CLR
INENALOW_READY
COUNT_IN
READY
TIME_IN_ENABLE
OUTENAB
NEXT_EVENT
OUTFLAG
IN
fdce
fjkc
XILINX
TITLE: MER_CONTROL
COMMENTS:
DATE:
SHEET SIZE: C
VER:
REV:


/CDM3HIGH(9:0)
/CDM3LOW(9:0)
/LOOKUPENABLE
/CDM2(9:0)
/CLK
/TIME_IN(23:0)
/ENERGY_POS_IN(23:0)
/INENA
/INIT
/FLAG
/CAL_FLAG
/CLR
/RESET
/READY
/NEXT_EVENT
/INENALOW_READY
/TIME_OUT(23:0)
/ENRERGY_POS_OUT(23:0)
007
003
005
000002
000005
000006
001002
700001
701004
703004
705004
000002
000005
001002
700001
70500C
0.0
1200.0
2400.0
3600.0
4800.0
6000.0
7200.0
8400.0
9600.0
Time(ns)