



Fortgeschrittenen-Praktikum

Versuch Digitalelektronik für Röntgen- und Gammadetektoren

Christoph Tenzer

Kepler Center for Astro and Particle Physics
Institut für Astronomie und Astrophysik
Abteilung Hochenergieastrophysik
Sand 1
72076 Tübingen

Stand: 19. April 2018

<http://www.uni-tuebingen.de/de/4203>

Inhaltsverzeichnis

1	Einführung	1
1.1	Astronomische Quellen in der Röntgen- und Gammaastronomie	2
1.2	Anforderungen an astronomische Detektoren und Elektronik im Röntgen- und Gammabereich	4
2	Detektion von Röntgen-/Gammastrahlung oberhalb von 10 keV	6
2.1	Szintillationsdetektoren	6
2.1.1	Allgemeine Prinzipien	6
2.1.2	Anorganische Szintillatoren	7
2.2	Phoswichdetektoren	10
2.3	Andere Detektoren für Strahlung oberhalb 10 keV	12
2.4	Abbildende Methoden oberhalb von 10 keV	13
3	Modernes Elektronikdesign für die Astronomie	15
3.1	Bausteine	15
3.1.1	IC und ASIC	15
3.1.2	FPGA und CPLD	15
3.1.3	Digitale Signalprozessoren	17
3.2	Hardware-Designwerkzeuge	17
4	Hardwaresynthese mit VHDL	18
4.1	VHDL	18
4.2	Konzept der Sprache	18
4.3	Architectures	19
4.4	VHDL-Syntax	21
4.4.1	Kommentare und Bezeichner	21
4.4.2	Datentypen, Deklarationen und Instanziierungen	21
4.4.3	Datentypen-Konversion	23
4.4.4	Zuweisungen	23
4.4.5	Prozesse	24
4.4.6	Zustandsautomaten	26
4.4.7	Unterprogramme, Packages und Libraries	27
4.5	Components	28
5	Versuchsdurchführung	29
5.1	Inbetriebnahme des Detektors	29
5.2	VHDL Programmierung	30
5.3	Vermessung der Detektoreigenschaften	32
6	Hinweise zur Anfertigung des Protokolls	33
A	Beschreibung der Kommandozeilen Programme	34
B	Pinout	35

Der Versuch “Digitalelektronik für Röntgen-/Gammadetektoren” soll in die moderne Elektronikentwicklung im Gebiet der Röntgen- und Gammaastronomie einführen.

In diesem Versuch soll mit Hilfe der Hardwarebeschreibungssprache VHDL eine Elektronik zum Betrieb und zur Auslese eines Phoswich-Detektors entworfen werden. Hierbei werden die typischen Schritte einer Hardware-Synthese durchlaufen: Skizzierung der notwendigen Komponenten und Module, Programmierung und Einbinden der einzelnen Steuer- bzw. Analyseprozesse, Simulation des Entwurfs am Computer, Syntheseprozess für einen XILINX Spartan-3 FPGA und Inbetriebnahme der Hardware als Schnittstelle zwischen Detektor und Datenerfassungssystem (Computer). Anschließend wird der Detektor kalibriert und seine Energieauflösung bestimmt. Zum Abschluss wird der Detektor mit einem Kollimator zum Gamma-Teleskop ausgebaut und die damit erreichbare Winkelauflösung vermessen.

In diesen einführenden Teil der Anleitung haben wir kurze Aufgaben eingestreut, die Ihnen zum Verständnis der Anleitung helfen sollen. Die Aufgaben 1-7 sind Teil des Praktikums und von Ihnen zum ersten Praktikumstag des Versuches vorbereitet/gelöst mitzubringen. Wenn Sie mit der Lösung der Aufgaben Probleme haben, dann bringen Sie bitte zumindest Ihre Lösungsversuche mit, wir werden die Aufgaben dann gemeinsam durchsprechen.

Viel Spaß beim Experimentieren wünschen:

Jörg Bayer, Henry Gebhardt, Thomas Schanz, Christoph Tenzer

1 Einführung

Die Röntgenastronomie (*engl.: X-ray astronomy*) befasst sich mit einem Teil des elektromagnetischen Spektrums bei Photonenenergien von 0.1 keV bis etwa 100 keV, Gammaastronomie mit Energien oberhalb davon bis zum GeV-Bereich. Dieser Bereich der Astronomie ist im Vergleich zur beobachtenden Astronomie bei niedrigeren Energien durch andere Arten von Teleskopen, Detektoren und durch eine etwas andere Art der Datenauswertung gekennzeichnet.

Strahlung in diesem Energiebereich wird in der Erdatmosphäre nahezu vollständig absorbiert, so dass Instrumente, mit denen Beobachtungen durchgeführt werden sollen, in sehr große Höhe gebracht werden müssen (Abbildung 1).

Diese Notwendigkeit lässt verstehen, warum erst 1948 mit einem Raketenexperiment in New Mexico Röntgenstrahlung der Sonne nachgewiesen werden konnte. Bis zur Entdeckung der Quelle Sco X-1 im Jahre 1962 galt die Detektion weiterer kosmischer Objekte im Röntgenlicht aufgrund der geringen gemessenen Röntgenleuchtkraft der Sonne daraufhin als höchst unwahrscheinlich. Mit der Entdeckung dieser Quelle, deren Emission im Röntgenbereich diejenige im optischen Bereich um den Faktor 10^4 übersteigt, wurde ein Fenster zu einem neuen Forschungsgebiet aufgetan, welches das Grundverständnis der Astrophysik über den Aufbau von Sternen und die physikalischen Abläufe in ihrem Inneren revolutionierte.

Sehr hochenergetische Gammastrahlung ($E > 1$ GeV) lässt sich heute wiederum indirekt über die Ausbildung von ausgedehnten Teilchenkaskaden (so genannte Luftschauer) in der Atmosphäre mit bodengestützten Teleskopen nachweisen.

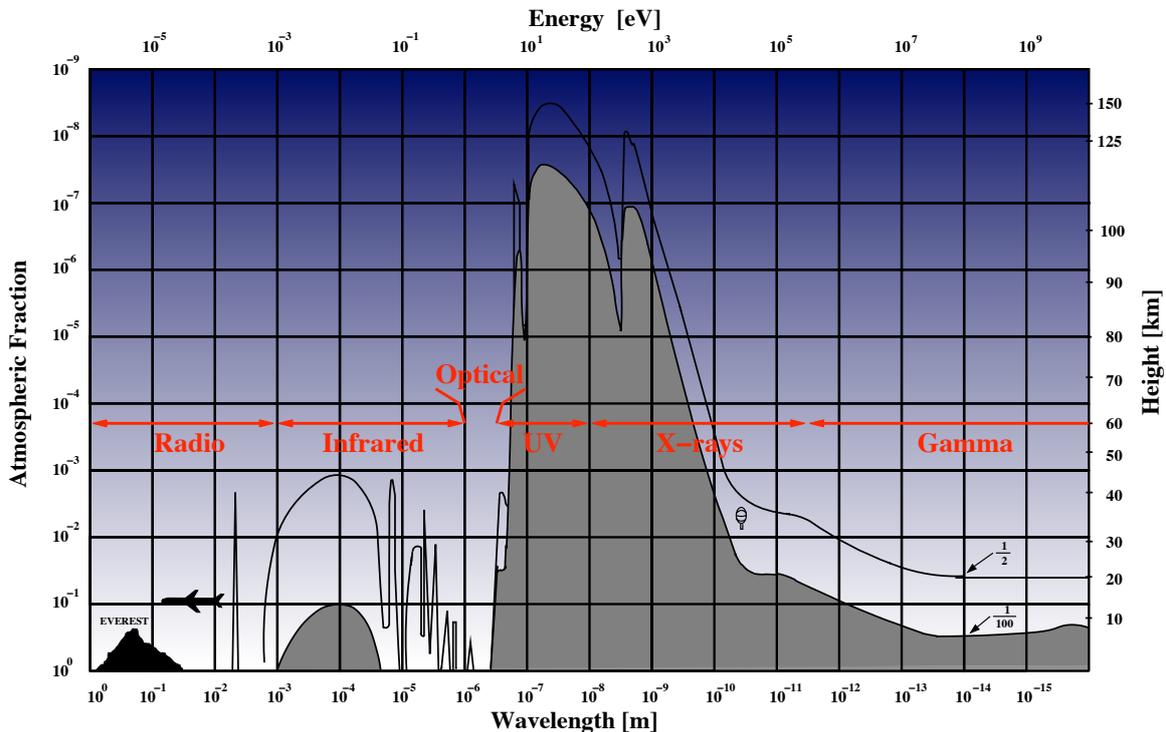


Abbildung 1: Dämpfung elektromagnetischer Strahlung verschiedener Wellenlängen in der Erdatmosphäre (Reproduktion einer Grafik aus Giacconi et al. (1968)).

1.1 Astronomische Quellen in der Röntgen- und Gammaastronomie

Welche physikalischen Prozesse gibt es überhaupt, die Photonenenergien oberhalb von 0.1 keV erzeugen? Nach dem Wien'schen Verschiebungsgesetz sendet ein Körper der Temperatur T das Maximum seiner Strahlung bei einer Photonenenergie von $E_{\max} = 2.8 kT$ aus.

Mit $k \approx 8.6 \times 10^{-8}$ keV/K folgt:

$$E_{\max} = 2.4 \times 10^{-7} \frac{\text{keV}}{\text{K}} \times T \quad (1)$$

Photonen mit Energien im Bereich von 50 bis 100 keV werden also von Körpern mit Temperaturen im Bereich von etwa 100 Millionen Grad ausgesandt! Die Hochenergie-Astrophysik beschäftigt sich also mit der Erklärung von Beobachtungen von Material unter sehr extremen Bedingungen, wie sie normalerweise nicht im Labor hergestellt werden können.

Außer der thermischen Strahlung, gibt es auch noch nichtthermische Strahlungsprozesse, bei denen ebenfalls Photonen im keV und MeV Bereich ausgesandt werden. Einige typische Beispiele für thermische und nichtthermische astrophysikalische Prozesse in der Gammaastronomie sind:

- Bremsstrahlung
- Radioaktiver Zerfall
- Synchrotron Strahlung
- Inverser Comptoneffekt
- Massenakkretion

Typische Objekte, die man am Himmel im Röntgen- und Gamma-Bereich sieht, sind:

- Sterne (Hauptsächlich deren heiße Koronen)
- Röntgendoppelsterne mit kompaktem Objekt
- Supernovaüberreste
- Gamma-ray Bursts
- Galaxienhaufen
- Aktive Galaxien (AGN)

Die genannten Prozesse treten dort zum Teil in Kombination auf.

Aufgabe 1: Schlagen Sie die in den Aufzählungen genannten (Ihnen noch unbekanntem) Begriffe im Internet nach und machen Sie sich damit ausreichend vertraut, so dass Sie die Frage *Was ist ...* zu Beginn des Praktikums zuverlässig beantworten können!

Material, das in einem Doppelsternsystem von einem normalen Stern auf einen kompakten Begleiter, d.h. einen Neutronenstern oder ein Schwarzes Loch, fällt, setzt bei der "Akkretion" einen großen Teil der potenziellen Energie des Materials in Strahlung im Röntgen- und Gammabereich um. Aus dem beobachteten Photonenspektrum können Rückschlüsse über die dabei eine Rolle spielenden physikalischen Mechanismen gezogen werden. Akkretion ist bei weitem der effizienteste Prozess im Universum zur Umwandlung in thermische Energie pro umgesetzter Masse.

Akkretierende Schwarze Löcher zeigen häufig Jets - kollimierte Materieausströmungen, bei denen das ausströmende Gas Geschwindigkeiten im Bereich der Lichtgeschwindigkeit haben kann. Solche Phänomene zeigen sich gleichermaßen bei stellaren Schwarzen Löchern (also Objekten mit Massen im Bereich von Sternmassen) wie auch bei supermassiven Schwarzen Löchern in den Zentren Aktiver Galaxien, die Massen von 10^9 Sonnenmassen erreichen können.

Bei Neutronensternen mit starken Magnetfeldern wird ein großer Teil des akkretierten Materials entlang der Magnetfeldlinien auf die Magnetfeldpole des Neutronensterns fallen. Die dort herrschenden Magnetfelder sind so stark, daß die Bewegung von Elektronen senkrecht zu Magnetfeldlinien gequantelt sind (Landau-Effekt). Dadurch wird die Absorption hochenergetischer Strahlung durch Elektronen, die Übergänge von einem solchen Landau-Niveau auf das nächste machen, möglich. Dies macht sich als Absorptionslinien im beobachteten Röntgenspektrum bemerkbar. Für den Übergang aus dem Grundzustand liegt die Linienenergie bei $E \simeq 12 \text{ keV} \cdot B_{12}$, wobei B_{12} die Magnetfeldstärke in Einheiten von 10^{12} Gauß angibt¹. Die Beobachtung solcher Linien ist die einzige bislang bekannte direkte Methode zur Messung der Polfeldstärke von Neutronensternen.

Auch bei supermassiven schwarzen Löchern in den Zentren Aktiver Galaxien, die Massen von 10^9 Sonnenmassen erreichen können, wird die Akkretion von Material für die große Leuchtkraft dieser Objekte verantwortlich gemacht. Bei aktiven Galaxien wird von Akkretionsraten von 1–3 Sonnenmassen pro Jahr ausgegangen. Mit Ausnahme von Supernovaexplosionen und Gamma-Ray Bursts sind diese Objekte die leuchtkräftigsten im ganzen Universum. Ferner werden in Aktiven Galaxien häufig Jets beobachtet: kollimierte Materieausströmungen, bei denen das ausströmende Gas Geschwindigkeiten im Bereich der Lichtgeschwindigkeit haben kann. Diese Jets strahlen im Radiobereich Synchrotron-Strahlung ab, sind aber auch im Röntgen- und Gamma-Bereich sehr hell. Die Ursache für die hohe Leuchtkraft ist hier, dass die Synchrotron-Strahlung aussendenden Elektronen diese Strahlung über den inversen Compton-Effekt ebenfalls zu sehr hohen Energien streuen (Synchrotron-Selbst-Compton Strahlung).

Aufgabe 2: (Auch im Versuch "Röntgen CCD") Berechnen Sie die maximale Leuchtkraft, die ein sphärisch symmetrischer Körper durch Akkretion haben kann. Diese wird dann erreicht, wenn der Strahlungsdruck auf die akkretierte Materie größer ist als die Gravitationskraft, mit der die akkretierte Materie angezogen wird. Zur Berechnung dieser so genannten Eddington-Leuchtkraft nehmen Sie an, daß die akkretierte Materie nur aus Wasserstoff besteht (warum ist das eine gute Annahme?) und auf ein Objekt der Masse M fällt. Die Gravitationskraft auf jedes akkretierte Proton ist

$$F_{\text{grav}} = \frac{GMm_p}{r^2} \quad (2)$$

¹Wie Sie im Laufe dieses Praktikums vielleicht schon gemerkt haben sind Astronomen konservativ und es werden immer noch die Einheiten des cgs-Systems (cm, g, Sekunde . . .) und nicht die Ihnen wahrscheinlich besser vertrauten Einheiten des SI benutzt. Aller Voraussicht nach wird das auch in Zukunft so bleiben. Machen Sie sich daher bitte mit solchen Einheiten vertraut!

(G: Gravitationskonstante, M: Masse des akkretierenden Objekts, m_p : Ruhemasse des Protons, r: Abstand des Protons zum akkretierenden Objekt). Dieser entgegengesetzt ist die Kraft, die durch den Strahlungsdruck ausgeübt wird. Diese wird hauptsächlich auf die akkretierten Elektronen vermittelt und ist gegeben durch (cgs- System!)

$$F_{\text{Strahlung}} = \frac{\sigma_T S}{c} \quad (3)$$

wobei $\sigma_T = 8\pi e^4 / (3m_e^2 c^4) = 6.652 \times 10^{-25} \text{ cm}^2$: Thomson-Wirkungsquerschnitt

$$\text{mit } S = \frac{L}{4\pi r^2} \quad (4)$$

(S: Energiefluss, L: Leuchtkraft).

Warum können Sie hier die beiden Kräfte gleichsetzen, obwohl F_{grav} auf Protonen wirkt und $F_{\text{Strahlung}}$ auf Elektronen? Geben Sie eine Formel für die Eddington-Leuchtkraft an und berechnen Sie diese für ein Objekt mit $M = 1M_\odot$. Der erhaltene Wert liegt in der typischen Größenordnung der Leuchtkräfte der stellaren Objekte, die in der Röntgenastronomie beobachtet werden. Vergleichen Sie diese Leuchtkraft mit der der Sonne ($M_\odot = 2 \times 10^{33} \text{ g}$, $L_\odot = 3.8 \times 10^{33} \text{ erg s}^{-1}$).

1.2 Anforderungen an astronomische Detektoren und Elektronik im Röntgen- und Gammabereich

Wie Aufgabe 2 gezeigt hat, können die uns hier interessierenden Quellen sehr große Leuchtkräfte haben. Dennoch sind große Anstrengungen notwendig, Informationen über solche Quellen zu bekommen. Warum das so ist, soll die folgende Aufgabe verdeutlichen:

Aufgabe 3: Eine röntgenastronomische Quelle hat nach Aufgabe 2 eine typische Leuchtkraft von $10^{38} \text{ erg s}^{-1}$. Gehen Sie der Einfachheit halber davon aus, dass ein Zehntel dieser Leuchtkraft im Energiebereich von 10 bis 100 keV ausgesandt wird und dass die mittlere Photonenenergie in diesem Bereich 20 keV beträgt. Wieviele Photonen pro Sekunde und Quadratzentimeter können Sie in diesem Energiebereich erwarten? Gehen Sie von einer Quellentfernung von 4 kpc aus und nehmen Sie an, dass die Leuchtkraft gleichmäßig in alle Raumrichtungen ausgestrahlt wird.

Das von uns zu messende Signal ist also alles andere als stark. Außer diesen geringen Zählraten ist eine weitere große Hürde, dass Röntgen- und Gammastrahlung von der Erdatmosphäre sehr gut absorbiert wird. Um daher Röntgen- und Gammaastronomie betreiben zu können, muss man sich, je nach Energie der zu beobachtenden Photonen, oberhalb eines großen Teils der Erdatmosphäre befinden. Das können Instrumente auf Forschungsballons (Flughöhe 40 km), Forschungsraketen (Flughöhen bis 100 km) oder Satelliten (oberhalb 300 km) sein. Aufgrund dieser technischen Schwierigkeiten sind die Röntgen- und Gammaastronomie sehr junge Forschungsgebiete. Die ersten Ballon- und Raketenexperimente fanden durch Herbert Friedman vom Navy Research Laboratory in den 1950er Jahren statt und waren der solaren Astrophysik gewidmet. Anfang der 1960er Jahre wurden dann, zuerst in den USA, später auch in anderen Ländern wie Deutschland, verstärkt Forschungsprogramme an extrasolaren Objekten durchgeführt.



Abbildung 2: Zwei Experimente im Röntgenbereich, die 1977 mit starker Beteiligung des IAAT durchgeführt wurden. Links: Start des High Energy X-ray Experiment (HEXE) in Palestine, Texas; Rechts: Skylark-Rakete auf einer mobilen Startrampe in Huelva, Spanien, zur Beobachtung einer Mondokkultation des Crab-Nebels (Fotos: R. Staubert).

In der damaligen Bundesrepublik wurde die ballonbasierte Röntgen- und Gammaastronomie durch ein Schwerpunktprogramm der Deutschen Forschungsgemeinschaft in den 1970er Jahren unter maßgeblicher Beteiligung des Tübinger Astronomischen Instituts und des Max-Planck-Instituts für extraterrestrische Physik in Garching bei München gefördert (Abbildung 2). Seit den 1980er Jahren ist die Röntgenastronomie fast vollständig von Satelliten dominiert und Ballons werden nur noch zur Erprobung neuer Technologien und zur schnellen Beobachtung besonderer Ereignisse (wie z.B. Supernovae) mit besonders geeigneten Instrumenten eingesetzt.

Das geringe Signal der zu beobachtenden astronomischen Quellen und die Tatsache, dass die Beobachtungen im Weltraum in Gegenwart eines hohen Detektorhintergrundes erfolgen müssen, erzeugen die folgenden Anforderungen, die sich auch in den wichtigsten Kenngrößen eines Detektors wieder spiegeln. Die Detektoren müssen...

- die Photonen mit einer hohen Wahrscheinlichkeit nachweisen (*Quanteneffizienz*),
- eine große *Sammelfläche* bieten, dennoch leicht genug sein,
- eine gute *Energieauflösung* haben,
- eine gute Unterdrückung des teilweise starken Hintergrundes aufweisen (*Empfindlichkeit*),
- gemeinsam mit dem abbildenden Teil des Teleskopes eine hohe *räumliche Auflösung* erreichen.

Diese Eigenschaften werden nur von wenigen Detektorarten hinreichend gut erfüllt. Als besonders gutes "Arbeitspferd" haben sich im Energiebereich von > 10 keV bis zu einigen 100 keV die Szintillationsdetektoren erwiesen, die daher in diesem Versuch eine herausragende Rolle spielen werden. Detektoren, die hauptsächlich bei Energien < 10 keV eingesetzt werden, werden im Versuch "Röntgen-CCD" behandelt.

2 Detektion von Röntgen-/Gammastrahlung oberhalb von 10 keV

Mit den in der Röntgen- und Gammaastronomie heute üblichen Teleskopen kann für jedes im Detektor nachgewiesene Photon 1) der Auftreffzeitpunkt, 2) die Richtung, aus der das Photon eintraf, und 3) die im Detektor deponierte Energie gemessen werden. Spezielle Detektoren erlauben auch die Messung einer weiteren Eigenschaft: 4) der Polarisationssebene. Unser heutiges Bild des Röntgen- und Gammahimmels beruht auf der Messung dieser Eigenschaften zahlloser Photonen.

2.1 Szintillationsdetektoren

Szintillatoren kommen als organische und anorganische Szintillatoren vor. Organische Szintillatoren zeichnen sich durch eine sehr schnelle Antwort auf ein Signal aus, haben aber eine geringere Lichtausbeute als anorganische Szintillatoren und eine sehr schlechte “stopping power” für die hier zu messenden einfallenden Photonen (proportional zur Kernladungszahl Z des Materials). Daher werden in der Gamma-Astronomie normalerweise anorganische Szintillatoren verwendet.

2.1.1 Allgemeine Prinzipien

Der Nachweis ionisierender Strahlung mithilfe von Szintillationsdetektoren ist eines der ältesten Verfahren². Dementsprechend groß ist die zur Verfügung stehende Literatur. Wir empfehlen insbesondere Kapitel 8 von Knoll (1999), dem englischsprachigen Standardwerk über Detektoren. Die sehr schönen Folien einer Vorlesung von Christian Joram über Teilchendetektoren für Sommerstudenten am CERN können ebenfalls empfohlen werden (Joram, 2001).

Unter Szintillation versteht man die Abgabe von Licht durch einen Körper, der durch Strahlung angeregt wurde. In unserem Fall wird diese Strahlung Gamma-Strahlung sein, Szintillatoren werden aber auch zur Detektion von Elektronen, Protonen oder Neutronen eingesetzt. Der Überbegriff der “Szintillation” fasst viele physikalische Mechanismen zusammen, wichtig im Folgenden sind dabei insbesondere

Fluoreszenz: die Abgabe sichtbarer Strahlung direkt nach der Anregung des Materials.

Phosphoreszenz: die Emission längerwelliger Strahlung über einen Zeitraum, der länger ist als der typische Zeitraum bei dem das Fluoreszenzlicht ausgesandt wird.

Verzögerte Fluoreszenz die durch den gleichen physikalischen Mechanismus wie die Fluoreszenz ausgelöst wird, aber ebenfalls mit einer langsameren Zeitkonstanten; sie kommt in organischen Szintillatoren vor.

Ein gutes Szintillatormaterial sollte einen möglichst großen Teil der absorbierten Energie in Form von Fluoreszenzlicht abstrahlen, damit die nachzuweisenden Photonen möglichst schnell detektiert werden können und es möglich ist, auch noch kurz hintereinander auftreffende Photonen zeitlich voneinander zu trennen (kleine *Totzeit*). Wir wollen also durch das auftreffende Photon einen möglichst scharfen Impuls optischen Lichts erzeugen.

²Laut Grupen (1993) waren die ersten Szintillationsdetektoren Zinksulfid-Schirme, auf denen geladene Teilchen Szintillationsblitze auslösten, die dann mit dem Auge registriert wurden. Dieses Verfahren ist relativ empfindlich: im grünen Spektralbereich sind mit dem Auge noch geringste Photonenzahlen von wenigen Photonen/Sekunde nachweisbar. Mit starkem Kaffee oder Strychnin lässt sich die Empfindlichkeit übrigens noch steigern (Grupen, 1993).

In einem Szintillationsdetektor wird der schwache Impuls des sichtbaren Fluoreszenzlichts mittels eines Photomultipliers verstärkt und in ein elektrisches Signal verwandelt (Abbildung 3). Dieses Signal wird dann mit einer nachgeschalteten Elektronik weiterverarbeitet.

Im Folgenden betrachten wir zunächst die oben dargestellten Szintillationsmechanismen und die im Versuch verwendeten Materialien genauer. Danach beschreiben wir den Mechanismus der Signalverstärkung im Photomultiplier und die Nachverarbeitung des Signals.

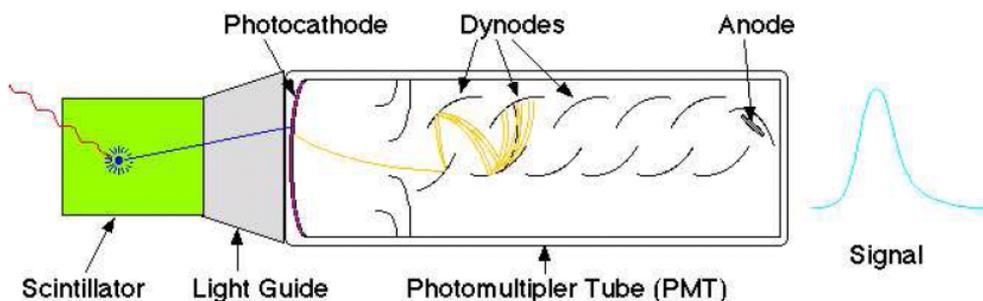


Abbildung 3: Prinzipieller Aufbau eines Szintillationsdetektors. (Abbildung: M. Dahlbohm).

2.1.2 Anorganische Szintillatoren

Die wichtigsten anorganischen Szintillatoren sind die Halogenidkristalle NaI(Tl) und CsI(Na), also Natriumjodid oder Caesiumjodid, dem als Aktivator Thallium oder Natrium beigemischt ist. Andere Szintillatormaterialien, die Ihnen im Bereich der Physik begegnen könnten, sind CsI(Tl), das BGO (Wismut-Germanat), chemische Formel $\text{Bi}_4\text{Ge}_3\text{O}_{12}$, das oben schon angesprochene Zinksulfid, $\text{ZnS}(\text{Ag})$, und Bariumfluorid (BaF_2). Eine vollständige Liste findet sich bei Knoll (1999, S. 235).

Zum Verständnis des Szintillationsmechanismus in anorganischen Szintillatoren ist das *Bändermodell*³ dieser Kristalle geeignet. Anorganische Szintillatoren sind Isolatoren oder Halbleiter, das heißt, dass das Valenzband und das Leitungsband voneinander getrennt sind. Wird ein Photon von einem Isolator absorbiert, dann kann seine Energie ausreichen, Elektronen aus dem Valenzband ins Leitungsband zu befördern, so dass Elektron-Loch Paare entstehen (Abbildung 4).

In einem reinen Kristall würden die Elektronen und die Löcher unabhängig durch den Kristall diffundieren und irgendwann wieder rekombinieren. Bei dieser Rekombination entsteht Licht einer Energie, die der Bandlücke des Kristalls entspricht. Dieser eben geschilderte Mechanismus ist vergleichsweise ineffizient. Zum einen vergeht eine lange Zeit, bis die Rekombinationen stattfinden, zum anderen ist der Kristall für das Rekombinationslicht nicht durchsichtig – die Energie des entstandenen Photons reicht ja aus, ein Elektron vom Valenz- ins Leitungsband zu befördern!

In Szintillatoren erzeugt man daher Verunreinigungen durch die Beimischung so genannter Aktivatoren. Diese werden so gewählt, dass das Kristallgitter durch die Aktivatoren gestört wird und Energiezustände innerhalb der Bandlücke entstehen. Die beim Auftreffen eines Gammaquants auf

³Sollte Ihnen das Bändermodell tatsächlich noch kein Begriff sein, sollten Sie das unbedingt ändern, indem Sie zumindest die Grundlagen rasch nachlesen.

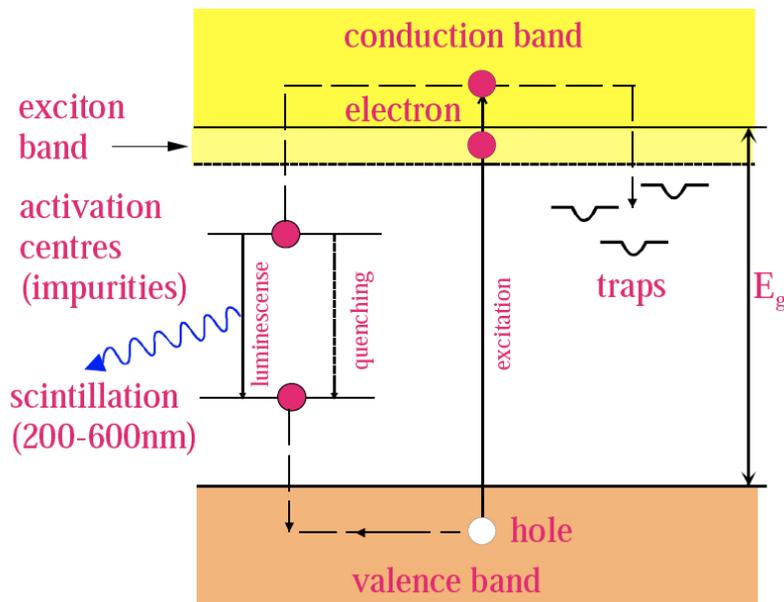


Abbildung 4: Die Entstehung von Szintillationslicht in einem anorganischen Szintillator (Joram, 2001).

den Szintillator entstehenden (positiven) Löcher ionisieren die Aktivatorionen, weil die Ionisationsenergie des Aktivators kleiner ist als die eines typischen Kristallions. Die bei der Absorption des Photons ebenfalls entstandenen Elektronen können dann durch diese Lumineszenz- oder Rekombinationszentren rekombinieren. Bei geschickter Wahl des Aktivormaterials liegt das entstehende Szintillationslicht im sichtbaren Bereich und kann gut mit dem Photomultiplier weiterverarbeitet werden. Ferner ist die Energie des Szintillationslichts kleiner als die Bandlücke, d.h., der Kristall ist für das Szintillationslicht durchsichtig.

Die typische Zeitdauer (Halbwertszeit) für die Rekombination liegt je nach Material zwischen 50 und 1000 ns. Da die Driftzeit der Elektronen durch den Kristall vernachlässigbar ist, bestimmt diese Halbwertszeit das entstehende Lichtsignal: Dieses steigt quasi instantan an und wird dann mit der oben genannten Halbwertszeit exponentiell schwächer. Bei manchen anorganischen Szintillatoren ist der entstehende Impuls komplizierter und kann z.B. durch die Summe mehrerer exponentiell abfallender Pulse dargestellt werden.

Dieses einfache Verhalten kann durch einige Mechanismen gestört werden. Am wichtigsten ist das Phosphoreszenzlicht. Hier wird das Elektron zwar vom Aktivator eingefangen, allerdings in einen angeregten Zustand, von dem der Übergang in den Grundzustand quantenmechanisch verboten ist. Um einen solchen Zustand abzuregen muss das Elektron beispielsweise durch thermische Stöße noch einmal in einen höherliegenden Zustand, von dem ein Übergang in den Grundzustand möglich ist, angeregt werden.

Dies dauert lange im Vergleich zur typischen Länge des Szintillationspulses, so dass wir von Phosphoreszenz reden. Ferner ist es möglich, dass Elektron und Loch nicht unabhängig voneinander durch den Kristall diffundieren, sondern als Elektron-Loch-Paar, als ein so genanntes *Exziton*. Auch Exzitonen werden durch Streuung an Aktivatoren abgeregt. Die dabei relevante Zeitskala liegt in der Größenordnung der Zeitskalen für die Abregung von Einzelelektronen bzw. -löchern. Schlussendlich können die angeregten Elektronen auch noch durch nichtradiative Übergänge wieder ins Valenzband zurückkehren, z.B. indem sie Ihre Energie durch Stöße ans Kristallgitter abgeben ("Quenching").

Auf Grund der komplizierten Bänderstruktur des Szintillatorkristalls entsteht bei der Abregung der Aktivatoren ein breitbandiges Szintillationsspektrum, das für die meisten Materialien sein Maximum im sichtbaren Licht hat (**Abbildung 5**).

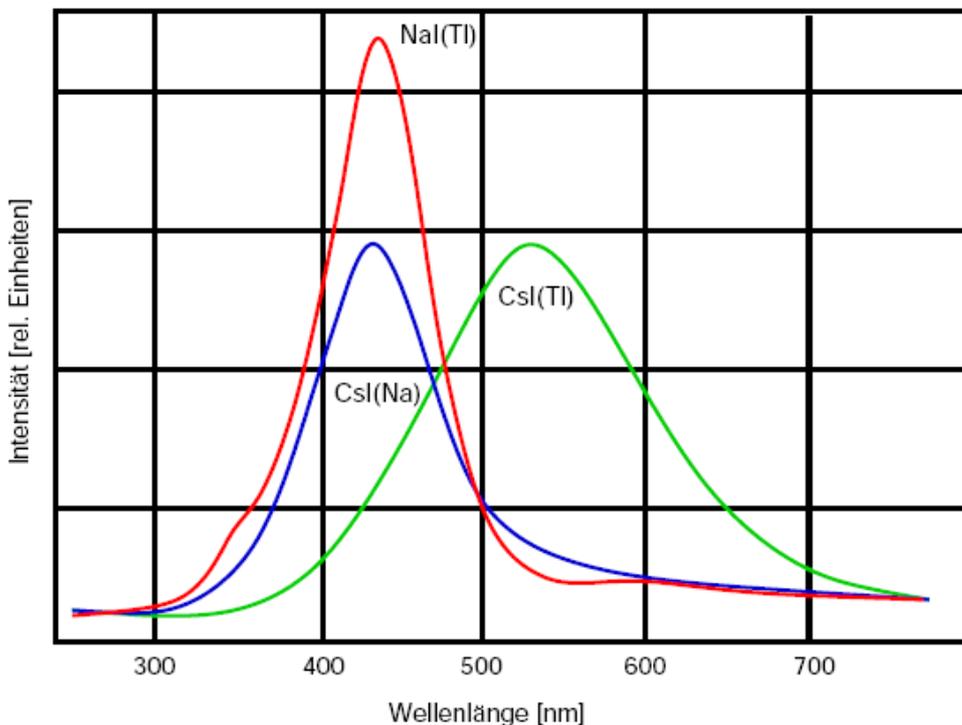


Abbildung 5: Spektren typischer anorganischer Szintillatoren (nach Bicron Technical Note “NaI”).

Die gerade geschilderten Mechanismen sind recht kompliziert, dennoch sind Szintillationsdetektoren sehr effiziente Detektoren: Wie die folgende Aufgabe zeigt, entsteht in NaI(Tl) pro entstandenem Elektron-Loch Paar ein optisches Photon.

Aufgabe 4: Als Faustregel kann angenommen werden, dass die Energie zur Erzeugung eines Elektron-Loch Paares im Schnitt gleich dem Dreifachen der Bandlücke eines Materials ist. Für NaI(Tl) ist die Bandlücke ungefähr 8 eV, d.h. zur Erzeugung eines Elektron-Loch Paares werden 25 eV benötigt. Messungen zeigen, dass die Lichtausbeute von NaI(Tl) 12 % beträgt, d.h. 12 % der Gesamtenergie der eingestrahlteten Photonen wird in Licht konvertiert. Gehen Sie der Einfachheit halber davon aus, dass all dieses Licht beim Maximum des Spektrums emittiert wird (**Abbildung 5**).

Berechnen Sie für ein 50 keV Photon die Zahl der im Kristall entstandenen Elektron-Loch-Paare und vergleichen Sie diese mit der Zahl der erzeugten optischen Photonen. Stimmt die obige Aussage über die Lichtausbeute, dass in NaI(Tl) pro entstandenem Elektron-Loch-Paar ein optisches Photon entsteht?

In anorganischen Szintillatoren ist die Lichtausbeute in guter Näherung proportional zur Zahl der entstandenen Elektron-Loch Paare. Damit ist die Gesamtzahl der gemessenen Photonen proportional zur Energie des Gammaquants. Dies bedeutet, dass Szintillationszähler zur Energiemessung benutzt werden können. Da (unter Vernachlässigung des Quenchings) die Zahl der Szintillationsphotonen linear mit der Gammaenergie ansteigt, spricht man davon, dass Szintillationszähler “lineare Detektoren” sind.

Da es sich durch eine exzellente Lichtausbeute auszeichnet, ist NaI(Tl) der Standarddetektor in der Gamma-Spektroskopie. Der Molanteil der Thallium-Beimischung beträgt 10^{-3} , der Brechungsindex von NaI(Tl) ist 1.85. NaI(Tl) kann gut auch in größeren Volumina (Kristalldurchmesser im Meterbereich) hergestellt werden.

Aufgrund der hohen Kernladungszahlen von NaI absorbiert NaI(Tl) effizient die auftreffende Gammastrahlung – schon in einem kleineren Volumen kann also die Gammastrahlung gut nachgewiesen werden, was für Satellitenexperimente von Vorteil ist. Ein Nachteil in der Handhabung ist, daß NaI(Tl) stark hygroskopisch ist und daher luftdicht eingepackt werden muß.

2.2 Phoswichdetektoren

Wie wir in Abschnitt 2.1.2 gesehen hatten, sind NaI(Tl) und CsI(Na) in ihren prinzipiellen Eigenschaften, wie der spektralen Form des Szintillationsspektrums und dem Brechungsindex, sehr ähnlich. Wichtigstes Unterscheidungsmerkmal dieser Szintillatoren ist die unterschiedliche Dauer des Szintillationspulses. In astrophysikalischen Anwendungen müssen sehr schwache Signale nachgewiesen werden in einer Umgebung, in der sich sehr viele hochenergetische Teilchen befinden. Es müssen also Strategien entwickelt werden, wie der Strahlungsuntergrund möglichst weit reduziert werden kann. In einem Phoswichdetektor macht man sich zur Untergrundreduktion die eben angesprochenen unterschiedlichen Dauern der Szintillationspulse von NaI(Tl) und CsI(Na) zu nutze, um die Herkunft des gemessenen Szintillationsimpulses zu identifizieren.

Phoswichdetektoren (von “Phosphor Sandwich”) bestehen aus einem dünnen NaI(Tl) Kristall, der optisch an einen dicken CsI(Na) Kristall gekoppelt ist. Der NaI(Tl) Kristall ist der eigentliche Detektor, während der CsI(Na) Kristall für die Untergrundreduktion eingesetzt wird: Teilchen, die “von unten oder von der Seite” in den Detektor eintreten, liefern ein Signal im CsI(Na). Eine dem Photomultiplier nachgeschaltete Elektronik analysiert die Pulsform und kann so diese Ereignisse eliminieren (Pulsformdiskriminierung oder auch Rise Time Discrimination). Eine solche Elektronik werden Sie im ersten Teil dieses Versuchs entwickeln.

Abbildung 6 zeigt schematisch, wie ein tatsächlich geflogener Phoswichdetektor aussieht, Abbildung 7 zeigt das Gehäuse eines Arrays aus den im Versuch verwendeten Detektoren. Bei dem in Abbildung 6 gezeigten High Energy X-ray Timing Experiment (HEXTE) auf dem Rossi X-ray Timing Explorer (RXTE) wird der vom Instrument einsehbare Himmelsausschnitt noch mittels eines Kollimators auf knapp ein Quadratgrad (vier Vollmonde) eingeschränkt. Ferner ist vor dem Detektor noch eine ^{241}Am Quelle angebracht⁴. Das kann zur Überwachung der Energieeichung des Phoswichdetektors verwendet werden.

Auch im Versuch werden ^{241}Am Strahlungsquellen verwendet. Dieses Isotop zerfällt unter Aussendung eines α -Teilchens mit einer Halbwertszeit von 432.7 Jahren in ^{239}Np . Gleichzeitig mit dem α -Teilchen wird ein γ -Photon bei 59.5 keV emittiert.

Außerhalb der Astronomie werden Phoswichdetektoren zum Beispiel in der medizinischen Strahlendiagnostik verwendet. Aus medizinischen Gründen ist auch hier die Messung möglichst geringer

⁴Dieses Americiumisotop ist nicht nur in der Wissenschaft von Interesse: Knapp 0.1 mg ^{241}Am sind auch in haushaltsüblichen Rauchdetektoren enthalten.

Mengen an radioaktiver Strahlung erforderlich.

Ein Photomultiplier (englisch: photomultiplier tube, PMT) ist dem Phoswichdetektor nachgeschaltet, um das Fluoreszenzlicht zu detektieren. Er besteht aus einer Photokathode, aus der bei Bestrahlung mit dem Szintillationslicht Elektronen austreten und mehreren Dynoden, zwischen denen jeweils eine Spannung angelegt ist (**Abbildung 3**). Die von der Photokathode primär erzeugten Elektronen (Photoelektronen, pe) werden in der PMT auf die erste Dynode fokussiert und durch ein angelegtes elektrisches Feld beschleunigt. Beim Auftreffen auf die Dynode werden dort weitere Elektronen herausgeschlagen, die dann auf die nächste Dynode beschleunigt werden, usw. Da die Beschleunigungsspannung hoch gewählt wird, ist die Zahl dieser sekundären Elektronen um ein wesentliches größer als die Zahl der Primärelektronen, d.h. das Anfangssignal wird stark verstärkt. Bezeichnen wir den Verstärkungsfaktor einer Dynode mit Q und hat der Photomultiplier n Dynoden, dann ist die Zahl der an der letzten Dynode abgegriffenen Elektronen im Mittel gegeben durch

$$N_{\text{signal}} = N_{pe} Q^n \quad (5)$$

Typische Dynoden-Verstärkungsfaktoren liegen bei $Q = 6 \dots 8$, insgesamt werden Verstärkungsfaktoren von 10^6 bis 10^8 erreicht.

Wie wir gesehen hatten, ist die Zahl der emittierten Szintillator-Photonen proportional zur Energie des Gammaquants:

$$N_{ph} \propto E \quad (6)$$

Diese primären Photonen werden von dem Photomultiplier mit einer gewissen (Quanten-)Effizienz in Photoelektronen (pe , N_{pe}) konvertiert, und diese Photoelektronen werden in der Detektorelektronik 'gezählt'. Gezählt heißt, dass pro Photoelektron eine bestimmte (mittlere) Zahl an Ladungsträgern an der Anode erzeugt und gemessen wird. Das heißt, es gilt insbesondere auch

$$N_{pe} \propto N_{ph} \propto E \quad (7)$$

Aufgrund der Poisson-Statistik ist die Messung einer Zählrate mit einer gewissen Unsicherheit behaftet. Für den Fehler relevant ist die Messung mit der kleinsten Ereigniszahl, also die Zahl der Photoelektronen. Bezeichnet man mit ΔN die Halbwertsbreite (Full Width at Half Maximum, FWHM) des gemessenen Signals, dann ist

$$\Delta N_{pe} = 2.36 \sqrt{N_{pe}} \propto \Delta E \quad (8)$$

Der Vorfaktor 2.36 erklärt sich durch die Umrechnung der Poisson-Varianz ($= N^{1/2}$) in die FWHM. Wegen Gleichung 7 und mit ΔE als FWHM der Energiemessung ist

$$\Delta E \propto \Delta N_{pe}. \quad (9)$$

Wäre der Fehler aus der Photoelektronenzählstatistik der einzige Fehler der Energiemessung, ergäbe sich

$$\frac{\Delta E}{E} = \frac{\Delta N_{pe}}{N_{pe}} \quad \text{oder allgemeiner} \quad \frac{\Delta E}{E} \geq \frac{\Delta N_{pe}}{N_{pe}} \quad (10)$$

Damit ist die relative Energieauflösung eines Szintillationsdetektors gegeben durch

$$\frac{\Delta E}{E} \geq \frac{\Delta N_{pe}}{N_{pe}} = 2.36 \frac{\sqrt{N_{pe}}}{N_{pe}} = \frac{2.36}{\sqrt{N_{pe}}} \propto \frac{1}{\sqrt{E}} \quad (11)$$

Beachten Sie, dass die Energieauflösung der Detektoren also energieabhängig, d.h. für eine spezielle Energie anzugeben ist! Die hier beschriebene Statistik dominiert normalerweise die Energieauflösung. Allerdings können weitere systematische Effekte, wie die von der Vorgeschichte abhängige Energieverstärkung im Photomultiplier, Variationen in der Spannung, "Quenching", Inhomogenitäten im Kristall, Abweichungen von der Linearität des Detektors, usw. die Energieauflösung verschlechtern.

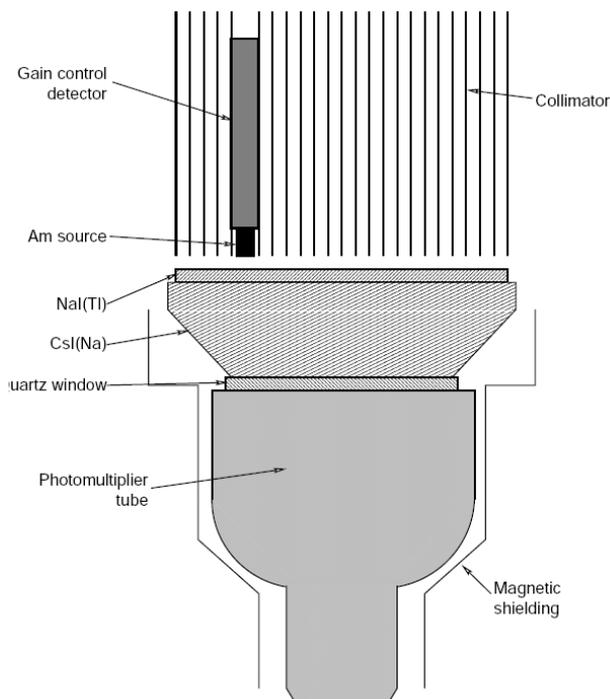


Abbildung 6: Schematischer Aufbau des HEXTE-Instruments auf dem amerikanischen Rossi X-ray Timing Explorer - der klassische Aufbau eines astronomischen Phoswichdetektors.

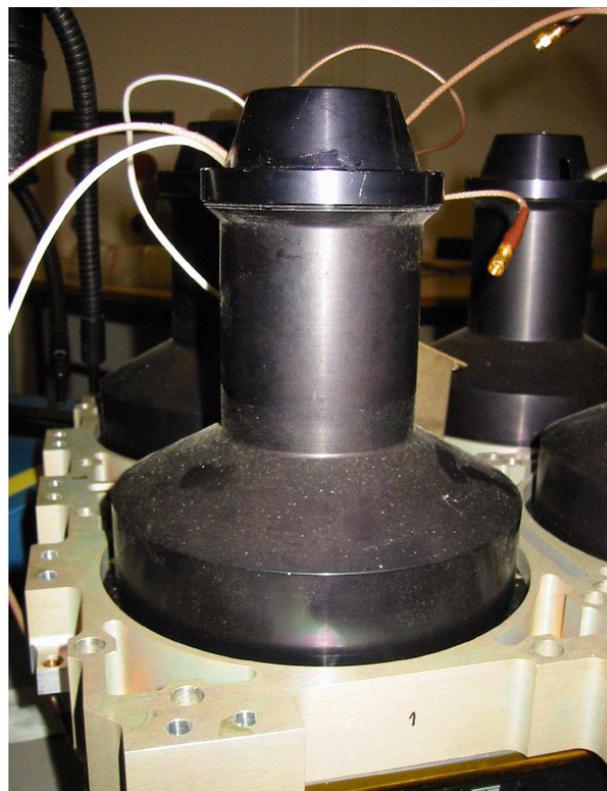


Abbildung 7: Gehäuse eines der im Versuch verwendeten Phoswich-Detektoren.

2.3 Andere Detektoren für Strahlung oberhalb 10 keV

Neben Szintillationsdetektoren sind in den letzten Jahren auch vermehrt Halbleiterdetektoren für den Nachweis von Röntgen- und Gammaquanten zum Einsatz gekommen. Halbleiterdetektoren haben eine wesentlich kleinere Bandlücke, so dass bei gleicher Energie des einfallenden Photons eine deutlich größere Zahl an Elektron-Loch Paaren erzeugt wird. Dies führt mit der obigen Argumentation zu einer verbesserten Energieauflösung. Ein Nachteil ist jedoch, dass die Detektoren häufig gekühlt werden müssen, um das thermische Rauschen zu unterdrücken.

Für Photonenergien oberhalb von 10 keV sind insbesondere Halbleitermaterialien mit hoher Kernladungszahl (Z) geeignet ($\text{Absorption} \propto Z^5$), um effizient hochenergetische Photonen schon mit relativ dünnen Detektoren nachzuweisen. Hier haben sich insbesondere neben Silizium ($Z = 14$) Germanium (Ge , $Z = 32$) und Kadmium-Tellurid (CdTe , $Z = 48 - 52$) bewährt. Da CdTe eine relativ große Bandlücke von 1.47 eV hat (zum Vergleich, Germanium hat eine Bandlücke von 0.74 eV), lässt es sich auch bei Raumtemperaturen einsetzen.

2.4 Abbildende Methoden oberhalb von 10 keV

Wie Sie im Versuch “Röntgen-CCD” bereits gelernt haben oder noch erfahren werden, kann Röntgenstrahlung bei Energien ≤ 15 keV leicht durch streifenden Einfall auf Spiegel fokussiert werden. Daher kann in diesem Energiebereich noch mit fokussierender Optik und positionsaufgelösten Detektoren eine Abbildung erreicht werden. Oberhalb von etwa 15 keV ist dies nicht mehr so einfach. Bislang werden für den Energiebereich von ~ 10 keV bis hin zu einigen MeV andere bildgebende Verfahren benutzt. Unter “Abbildung” verstehen wir dabei alle Verfahren, mit denen erreicht wird, dass Strahlung nur aus einer bestimmten Himmelsregion detektiert wird. Diese kann durchaus größer als der Vollmond sein. Dennoch ist eine solche “Abbildung” wichtig, weil sie es erlaubt, das Quellsignal vom (isotropen) Hintergrund zu trennen und es erlaubt, eine räumliche Verteilung der Strahlungsintensität darzustellen. Da der Teilchenhintergrund normalerweise sehr stark ist, ist dies unbedingt nötig, um Informationen über die beobachteten Quellen (Quellspektrum, zeitliche Variabilität usw.) gewinnen zu können.

Die dafür verwendeten Verfahren lassen sich grob in drei Klassen einteilen:

1. Das Gesichtsfeld der Detektoren kann mit Hilfe so genannter **statischer Kollimatoren** eingeschränkt werden, d.h. nur Röntgenstrahlung aus einer kleinen Himmelsregion kann auf den Detektor fallen, Strahlung aus anderen Regionen wird im Material des Kollimators absorbiert. Dadurch wird der Strahlungshintergrund im Vergleich zur Quellstrahlung stark reduziert. Der einfachste Kollimator besteht aus einem Röntgenstrahlung stark absorbierenden Material, das als Röhre um den Detektor herum angebracht wird. Die Abbildungseigenschaft eines solchen Kollimators hat Dreiecksform: Quellen, die direkt auf der optischen Achse liegen, werden mit vollem Fluss detektiert, je weiter die Quelle von der Achse entfernt ist, desto weniger Photonen werden von ihr detektiert. Der Winkel, unter dem eine Quelle mit ihrem halben Fluss detektiert wird (paralleler Lichteinfall), ist gegeben durch

$$\tan \theta = \frac{d}{2h} \quad (12)$$

wobei d die Breite des Detektors und h die Höhe des Kollimators ist.

Aufgabe 5: Für helle Röntgenquellen sollte ein kollimiertes Instrument ein Gesichtsfeld ($= 2\theta$) von kleiner 2° haben. Nehmen Sie einen quadratischen Detektor mit 4000 cm^2 Sammelfläche an. Wie lang muß der Kollimator sein?

Um mit einem solchen einfachen Kollimator das Gesichtsfeld auf ein vernünftiges Maß einzuschränken, wären also sehr lange Kollimatoren notwendig, ein solcher einfacher Kollimator ist also unpraktisch. Da nach Gleichung 12 das Gesichtsfeld proportional zu d/h ist, ist es besser, viele kleine Kollimatoren vor dem Detektor anzubringen. Ein solches Schema ist in [Abbildung 8](#) auf der linken Seite wiedergegeben. Hier bedecken viele kleine Kollimatoren den Detektor, die das gleiche Seitenverhältnis haben, wie ein “langer” Kollimator.

Das Teleskop hat daher das gleiche Gesichtsfeld, ist aber wesentlich kompakter. Daher ist ein solches Design besser geeignet, z.B. auf einem Satelliten implementiert zu werden, weil dort die Instrumente möglichst platzsparend und leicht sein müssen. In der Praxis werden zweidimensionale Kollimatoren entweder honigwabenförmig aufgebaut, weil dies eine große Stabilität liefert, oder sie werden am Stück aus Mikrokanalplatten gefertigt. In beiden Fällen muss natürlich darauf geachtet werden, dass die Wanddicke der einzelnen Kollimatorzellen möglichst klein

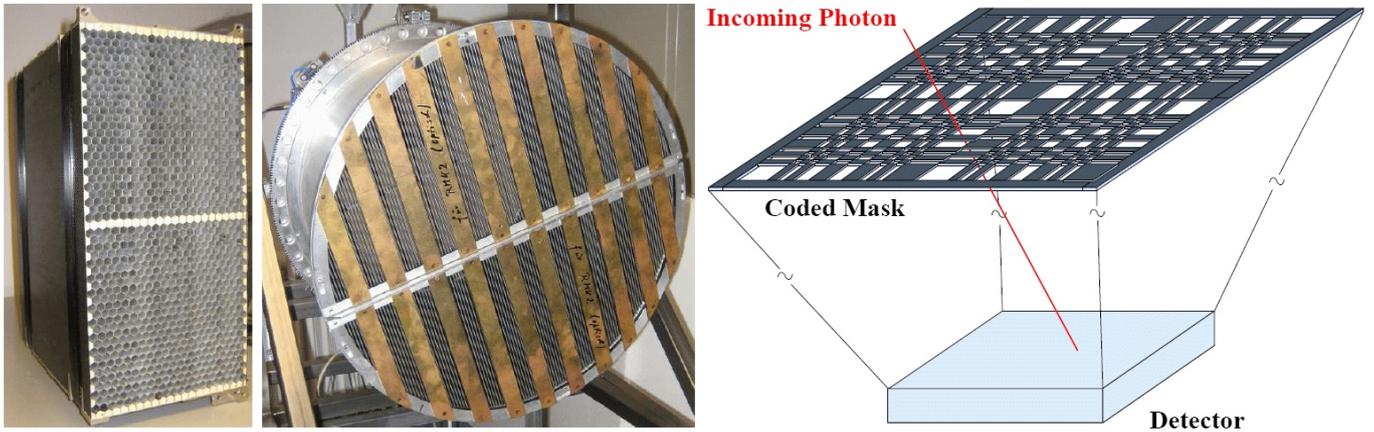


Abbildung 8: Verschiedene Kollimatoren, die in der Röntgen- und Gamma-Astronomie zum Einsatz kommen. Links: Statischer Honigwabekollimator; Mitte: Rotations-Modulations-Kollimator; Rechts: Prinzip der Abbildung mit einer kodierten Maske.

ist, um den Anteil der nichtempfindlichen Detektorfläche klein zu halten. Ein Beispiel für ein Instrument, bei dem Kollimatoren verwendet werden, ist das schon vorher erwähnte HEXTE-Experiment auf dem Rossi X-ray Timing Explorer. Mit Kollimatoren kann das Gesichtsfeld zwar eingeschränkt werden, aber es ist nicht möglich, Quellen voneinander zu trennen, die innerhalb des Gesichtsfeldes des Kollimators liegen.

2. Eine wirkliche Abbildung ist durch zeitliche Aperturmodulation möglich. Eine zeitliche Modulation des Quellsignals kann dadurch erreicht werden, daß ein Plattenkollimator über die Quelle bewegt wird. Eine bessere Lösung sind so genannte Modulationskollimatoren, bei denen die Abbildung ebenfalls durch eine Scanbewegung erreicht wird, oder aber die Rotations-Modulationskollimatoren (**Abbildung 8**, mitte), bei denen das Quellsignal zeitlich moduliert wird. Letztere wurden vor allem bei Ballonexperimenten in den 80er Jahren verwendet. Hierbei wird die Intensität von Quellen, die sich an unterschiedlichen Stellen im Gesichtsfeld befinden, je nach Position auf eindeutige Weise periodisch moduliert: Wenn man die zwei Gitter gegeneinander rotiert, werden die Quellen - aus Sicht des Kollimators - mit unterschiedlicher Frequenz und Phase von den Stäben verdeckt. Die Intensität des Detektorsignals ist dann eine Überlagerung der modulierten Intensitäten aller Quellen im Gesichtsfeld. Die Position jeder Quelle kann später am Computer aus der Periode und der Phasenverschiebung der Lichtkurve bestimmt werden.
3. Alternativ zur zeitlichen Modulation des Signals ist auch die räumliche Aperturmodulation möglich. Die Abbildung mit kodierten Masken (**Abbildung 8**, rechts) beruht auf dem Prinzip des Schattenwurfs von Quellen: unter Verwendung einer Maske, die aus absorbierendem Material aufgebaut ist, wird in der Abbildungsebene mit einem orts aufgelösten Detektor der Schatten aller im Gesichtsfeld liegender Quellen aufgezeichnet. Ist mehr als eine Quelle sichtbar, so muss mit Hilfe von aufwändigen Algorithmen die Position und Stärke der Quellen zurückgerechnet werden, die für die Modulation des Signals in der Bildebene verantwortlich sind. Diese Methode bietet gegenüber zeitlich modulierten Kollimatoren den Vorteil, dass auch zeitlich variable Quellen verlässlich rekonstruiert werden können. Solch eine Maske aus strahlungsundurchlässigem Wolfram kommt zum Beispiel auf dem INTEGRAL-Satelliten zum Einsatz.

3 Modernes Elektronikdesign für die Astronomie

Die Entwicklungsrichtung beim Design von Elektronik-Komponenten für wissenschaftliche und industrielle Anwendungen tendiert seit Jahren klar zu immer stärker miniaturisierten Bausteinen. Durch die bessere Reproduzierbarkeit bei der Herstellung, die höhere Zuverlässigkeit und die Kompatibilität zu IT-Systemen hat die Digitaltechnik die analoge Elektronik bei Steuerungs- und Meßanwendungen längst abgelöst. Daher beschäftigt sich Elektronikdesign heute fast ausschließlich mit dem Entwurf digitaler Schaltungen.

Ein weiterer Vorteil digitaler Elektronik-Hardware besteht darin, dass sich komplexe Schaltungsfunktionen mit Hilfe mathematischer Formeln beschreiben und aus einfachen, kostengünstig herstellbaren Elementen aufbauen lassen. Einige dieser Grundbausteine digitaler Schaltungen sollen in diesem Abschnitt kurz vorgestellt werden.

3.1 Bausteine

3.1.1 IC und ASIC

Ein IC (INTEGRATED CIRCUIT) ist ein nur wenige Millimeter großer Chip-Baustein, der, je nach Funktion, aus bis zu Millionen von Halbleiterelementen - zumeist Transistoren - aber auch aus passiven Komponenten wie Kondensatoren aufgebaut ist. ICs zeichnen sich durch ihre geringen Abmessungen, ihre Zuverlässigkeit, kurze Schaltzeiten und Eignung zur Massenproduktion aus. Welche Funktion ein bestimmter IC später ausführt, wird bereits beim Entwurf festgelegt und kann nicht mehr verändert werden. Somit entstanden im Laufe der Zeit große Sortimente unterschiedlicher ICs für alle nur denkbaren Anforderungen - von einfachen Gattern (digitale Grundsaltungen wie AND, OR, NAND, Inverter oder FlipFlops) bis zu den heute in Handys und Computern verwendeten Mikroprozessoren. Die bekannteste und am weitesten verbreitete Baureihe ist die 74-TTL⁵-Familie, in der viele digitale Grundsaltungen enthalten sind, die vom Entwickler zu komplexen Anwendungen auf PCBs⁶ verdrahtet werden können. Aber auch analoge und so genannte 'mixed signal'-Schaltungen (analog *und* digital) sind als IC erhältlich.

Als ASIC (APPLICATION-SPECIFIC INTEGRATED CIRCUIT) bezeichnet man einen IC, der - im Gegensatz zu den sonst beliebig einsetzbaren ICs - gezielt für eine bestimmte Anwendung entworfen und hergestellt wurde. Für kleinere Anwendungen oder die Herstellung geringer Stückzahlen wurden ASICs bereits von den immer mehr Platz bietenden und schneller werdenden FPGAs (s.u.) abgelöst.

3.1.2 FPGA und CPLD

Ein FIELD-PROGRAMMABLE GATE ARRAY (FPGA) ist eine generische Form des ASIC, bei der die Festlegung der Funktion erst nach dem Herstellungsprozess vom Anwender vorgenommen wird. Je nach Technologie (Antifuse⁷, EPROM⁸, FLASH⁹) ist er ein bis mehrere tausend mal (re-)programmierbar. FPGAs sind ihren ASIC-Gegenständen derselben Funktionalität in Bezug auf Schaltzeiten und Leistungsaufnahme unterlegen, haben jedoch den Vorteil kürzerer Entwicklungszeiten und geringerer Herstellungskosten.

⁵Transistor-Transistor-Logik: Schaltungstechnik für logische Schaltungen.

⁶Printed Circuit Board (dt.: Platine, Leiterplatte).

⁷Irreversible Programmierung nach dem umgekehrten Prinzip einer Sicherung.

⁸Eraseable Programmable Read-Only Memory - mit UV-Licht löscht- und anschließend wiederbeschreibbare Speicherbausteine.

⁹Elektronisch löscht- und beschreibbare Speichertechnologie.

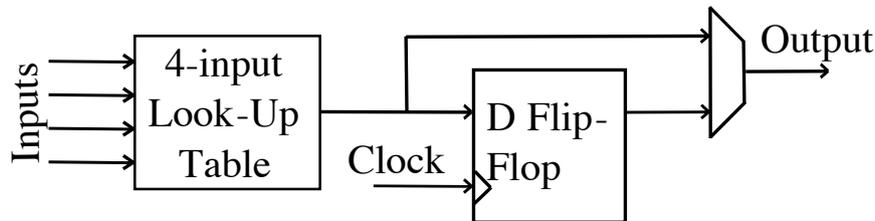


Abbildung 9: Beispiel eines Logikblocks mit LUT und Flip-Flop

Die Grundstruktur des FPGAs ist ein Feld (engl.: array) aus Basisblöcken (engl.: standard-cells) mit jeweils einer einfachen programmierbaren Lookup-Tabelle (LUT) und einem 1-Bit-Register (Flip-flop), siehe [Abbildung 9](#). Die LUTs können, je nach Anzahl der verfügbaren Eingänge, jede beliebige n-stellige Binärfunktion realisieren. Die Programmierung der gewünschten Funktion erfolgt durch die Hinterlegung der definierenden Wahrheitstabelle in den SRAM¹⁰-Zellen der LUT, die Funktionsberechnung durch das Auslesen der durch die Eingänge bestimmten Speicheradresse. Lange Zeit waren LUT-Strukturen mit 4 binären Eingängen üblich. Neuere FPGAs gehen zur Verringerung des Aufwandes an LUT-zu-LUT-Verbindungen zur Realisierung von Funktionen mit mehr Eingängen auf LUTs mit bis zu 6 Eingängen über. Neben den LUTs ist auf dem FPGA auch die Verschaltung der Komponenten in großen Freiheitsgraden konfigurierbar. Multiplexer-Strukturen in den Basisblöcken ermöglichen häufig sehr schnelle lokale Signalpfade, zur Einbindung oder Umgehung des Flipflops, zur Rückkopplung von dessen Ausgang, zur Verbindung von Nachbarblöcken und ähnlichem. Für die ferneren Verbindungen liegt zwischen den Basisblöcken ein Gitter aus immensen Busstrukturen, an das Ein- und Ausgänge angeschlossen werden können. Weitere programmierbare Schaltkomponenten in den Kreuzungspunkten des Gitters erlauben die Signalverteilung über den gesamten Chip.

Um das Verhalten eines FPGAs zu definieren, werden die im nächsten Abschnitt näher erläuterten Hardware-Designwerkzeuge eingesetzt. Damit lässt sich eine Netzliste der Verbindungen zwischen den Basisblöcken erstellen und diese dann in den FPGA übertragen. Der Begriff Programmierung ist in diesem Kontext also anders zu verstehen als bei der Erstellung von Software für einen Prozessor: In einem FPGA werden Schaltungsstrukturen mittels Hardwarebeschreibungssprachen oder in Form von Schaltplänen erstellt und nachfolgend diese Daten zwecks *Konfiguration* in den Baustein übertragen. Im FPGA werden dadurch bestimmte Schalterstellungen aktiviert bzw. deaktiviert, was dann eine konkret implementierte digitale Schaltung ergibt.

In einen modernen FPGA lassen sich auch so komplexe Strukturen, wie z.B. Prozessor-Architekturen laden, so dass sich FPGAs hervorragend für Hardware-Entwicklung und -Tests eignen. Die Idee, die Programmierung während der Laufzeit zu verändern und an die anstehende Aufgabe anzupassen, um so eine optimale Performance bei zeitkritischen Anforderungen, wie z.B. digitaler Video- und Audio-Signalverarbeitung zu erreichen, hat FPGAs in letzter Zeit in diesem Bereich weitere Einsatzgebiete erschlossen.

Im Gegensatz zu dem aus standard-cells aufgebauten FPGA besteht ein CPLD (COMPLEX PROGRAMMABLE LOGIC DEVICE) aus so genannten MAKROZELLEN, von denen jede für sich fähig ist,

¹⁰Static Random-Access Memory - elektronischer Speichertyp. Sein Inhalt ist flüchtig, das heißt, die gespeicherte Information geht bei Abschaltung der Betriebsspannung verloren.

eine boolesche Operation¹¹ auf ihren binären Inputwerten auszuführen und das Resultat am Output-Port auszugeben, bzw. bis zum nächsten CLK-Cycle in einem Flipflop zu speichern. Allerdings ist es aufgrund ihrer internen Struktur oft nicht möglich, beliebige Makrozellen miteinander zu verknüpfen, so dass häufig Hardware-Designs sich nicht in ein gegebenes CPLD einpassen lassen, obwohl viele Makrozellen noch nicht belegt sind. Daher haben CPLDs in letzter Zeit etwas an Bedeutung verloren.

3.1.3 Digitale Signalprozessoren

Ein digitaler Signalprozessor (DSP) ist ein auf Signalverarbeitung in Echtzeit spezialisierter Mikroprozessor mit einem für diese Aufgabe optimierten Befehlssatz. DSPs sind seit 1979 als Einzelchip erhältlich und haben seither stark an Bedeutung gewonnen. Ihre enorme Leistungsfähigkeit bei der Signalverarbeitung hat in letzter Zeit auch die Prozessorentwicklung der CPUs für handelsübliche PCs beeinflusst, z.B. die MMX-Erweiterung, wo Bild- und Videoverarbeitung immer mehr in den Vordergrund rücken. Auch zur Verarbeitung von Signalen in der Röntgenastronomie werden DSPs vorgesehen. Da DSPs im Grunde auch nur Mikroprozessoren sind, können Programme dafür in Programmiersprachen wie C++ oder in Maschinensprache geschrieben werden.

3.2 Hardware-Designwerkzeuge

Mit der Evolution der Hardware-Elemente wurden auch die für deren Entwicklung verwendeten Design-Werkzeuge an die neuen technischen Herausforderungen und Möglichkeiten angepasst. Elektronische Schaltungen, die früher noch am Reißbrett entworfen wurden, werden heute mit CAD-Software am Computer geplant. In Entwicklungsumgebungen eingebettete Design-Editoren und Hardwarebeschreibungssprachen bieten dabei die Möglichkeit, Schaltungen zu entflechten, Platinenlayouts anzufertigen und die fertigen Entwürfe per Simulation auf Funktionalität und Korrektheit zu überprüfen, bevor sie als Hardware realisiert werden.

Moderne Design-Editoren (engl.: *schematics editor*) stellen umfangreiche Bibliotheken mit grafischen Darstellungen von Elektronik-Bausteinen bereit, die per Mausklick auf eine Zeichenfläche gezogen und dort zu komplexen Schaltungsdesigns arrangiert werden können. Diese Bibliotheken, die sowohl gängige Standard-Elemente als auch High-End-Komponenten enthalten, lassen sich über das Internet ständig erweitern und aktualisieren. Die fertigen Schaltungen können anschließend analysiert und ihr Verhalten mit Testsignalen simuliert werden. Einige Editoren bieten nützliche Funktionen, z.B. das Design zu einem Platinenlayout zu verarbeiten, oder daraus (wenn möglich) Netzlisten zur Programmierung von FPGAs oder CPLDs zu erstellen.

Eine Alternative zum grafischen Design stellen Hardware-Beschreibungssprachen (HDLs¹²) dar, mit denen ganze Schaltungen oder auch nur einzelne Hardware-Bausteine in Form eines Modells beschrieben werden. Ein Vorteil gegenüber schematics besteht darin, dass nicht nur bestehende Bausteine miteinander zu neuen Anwendungen verbunden, sondern auch die Funktion eigener Bausteine durch logische, evtl. taktgesteuerte Verknüpfung von Ein- und Ausgangssignalen in einem Programm definiert werden können. Ähnlich dem Vorgehen bei Programmiersprachen werden dabei in einem gewöhnlichen Texteditor mit speziell dafür entwickelter Syntax Komponenten aus Bibliotheken instanziiert, Funktionen aufgerufen und Variablen Werte zugewiesen. Das fertige Design wird mit einem Hardware-Compiler zur Netzliste synthetisiert. Auch für HDLs existieren Entwicklungsumgebungen, die die Analyse und Simulation des Designs vor der Hardware-Implementierung erlauben.

¹¹AND, OR, NAND, etc.

¹²HDL - Hardware Description Language

4 Hardwaresynthese mit VHDL

In diesem Versuch soll mit Hilfe der Hardwarebeschreibungssprache VHDL, die im nächsten Absatz näher beschrieben wird, Elektronik zum Betrieb und zur Auslese eines Phoswich-Detektors entworfen werden. Zunächst möchten wir Ihnen eine kurze Einführung in die Hardwarebeschreibungssprache VHDL geben, die Ihnen die Ähnlichkeiten und Unterschiede zu Ihnen vielleicht bekannten Programmiersprachen verdeutlichen soll.

4.1 VHDL

Die Abkürzung VHDL steht für VHSIC¹³ Hardware Description Language. VHDL enthält viele Elemente der gängigen, prozedural arbeitenden Programmiersprachen, erweitert um die Parallelität der PROZESSE und weiterer typischer Konstrukte für den Hardware-Schaltungsentwurf. Auch die Syntax ist durchaus vergleichbar mit der höherer Programmiersprachen, so dass Entwicklern mit fundierten Programmierkenntnissen der Neueinstieg in modernes Hardware-Design mit VHDL leichter fällt, als Entwicklern mit klassischem Elektronik Know-How. VHDL wurde seit 1983 von Intermetrics, IBM und Texas Instruments entwickelt und 1987 vom IEEE¹⁴ standardisiert.

4.2 Konzept der Sprache

VHDL bietet die Möglichkeit zur vollständigen Beschreibung einer digitalen elektronischen Schaltung. Sowohl deren *Struktur* (Aufbau aus einzelnen Komponenten und deren Verdrahtung miteinander), als auch die *Funktion* (Beschreibung mit Hilfe von logischen Operatoren) lassen sich in einem Entwurf abbilden. Ein VHDL-Design ist von den verwendeten Hard- und Software-Komponenten unabhängig; es kann problemlos zwischen Entwicklern mit verschiedenen Systemplattformen und Ziel-Hardwares ausgetauscht werden.

Jedes VHDL-Design ist in einen ENTITY-Teil und mindestens eine ARCHITECTURE unterteilt. Sind mehrere Architectures für eine Schaltung im Entwurf vorhanden, wird noch eine CONFIGURATION -Anweisung benötigt, die die Auswahl zwischen den Architectures erlaubt. Dieser Grundaufbau eines VHDL-Designs ist in Abbildung 10 schematisch gezeigt.

Im ENTITY-Teil wird die Schnittstelle des Entwurfes nach außen hin beschrieben. Dazu werden alle Ein- und Ausgänge der zukünftigen Hardware mit ihren jeweiligen Signaltypen spezifiziert, unabhängig davon, ob es sich um das Design eines FPGA, eines IC oder einer ganzen Platine handelt.

Mit der CONFIGURATION kann der Entwickler aus mehreren Architectures auswählen. Ist nur eine Architecture vorhanden, so wird der Configuration-Teil optional (dies ist bei den meisten Anwendungen im Alltag der Fall - Configuration wird deshalb hier nur der Vollständigkeit halber erwähnt).

Die ARCHITECTURE enthält die eigentliche Hardware-Beschreibung des VHDL-Designs. Es werden im wesentlichen zwei Beschreibungsformen unterschieden:

- funktionelle Beschreibung
- strukturelle Beschreibung

¹³Very High Speed Integrated Circuit.

¹⁴IEEE - Institute of Electrical and Electronics Engineers.

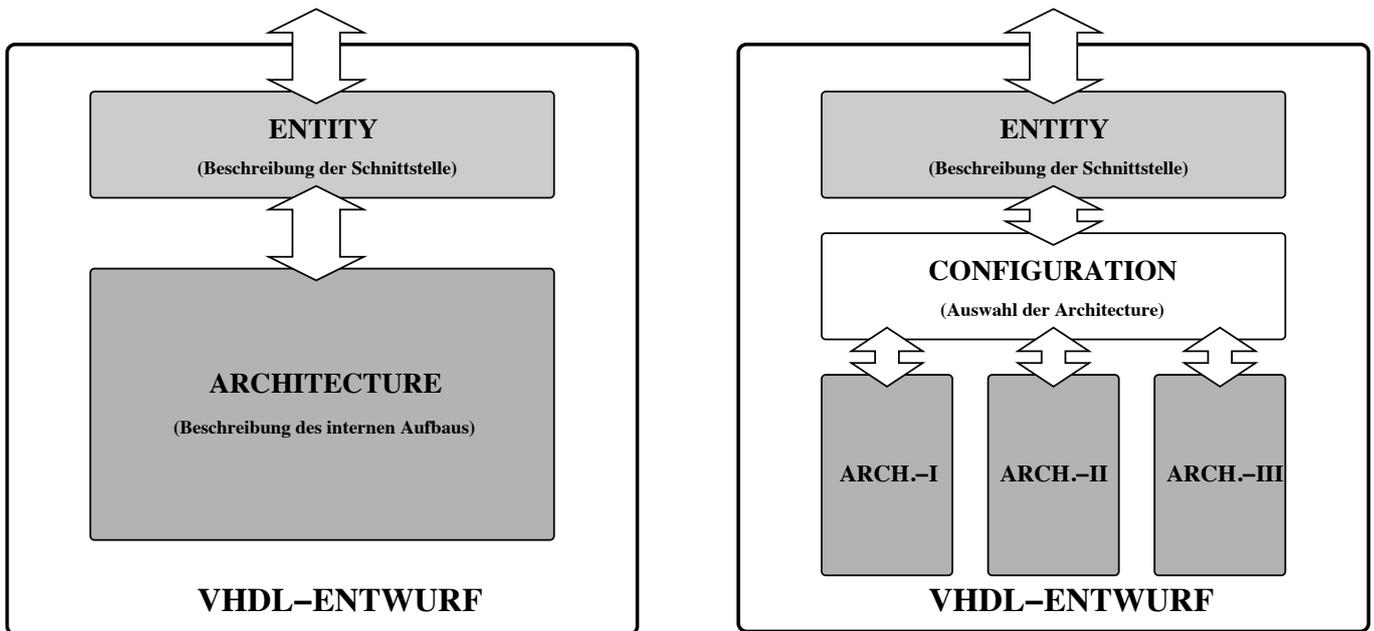


Abbildung 10: Aufbau eines VHDL-Designs mit nur einer ARCHITECTURE (links) und mit CONFIGURATION für mehrere Architectures (rechts). Auch bei mehreren Architectures in einem Entwurf verwenden alle dieselbe ENTITY.

Elektronische Schaltungen bestehen i.a. aus Bauteilen, die auf einer Leiterplatte montiert sind. Die *funktionelle* Beschreibung dient der Modellierung dieser Bauteile. Dabei entspricht die ENTITY der Schnittstelle des Bauteils (den Ein- und Ausgangs-Pins des ICs), die ARCHITECTURE beschreibt stattdessen die Funktion (deshalb funktionelle Beschreibung) des Bauteiles. Mit einer *strukturellen* Beschreibung ist es hingegen möglich, eine ganze Leiterplatte zu modellieren. Die ENTITY entspricht hier dann dem Platinenstecker und die ARCHITECTURE beschreibt dann die Verdrahtung der Bauteile.

4.3 Architectures

Die funktionelle Modellierung beschreibt das Verhalten (engl.: behaviour) von Bauteilen. Abbildung 11 zeigt eine solche Beschreibung. Die ENTITY beschreibt die Schnittstelle des Bauteiles, die ARCHITECTURE die Funktion.

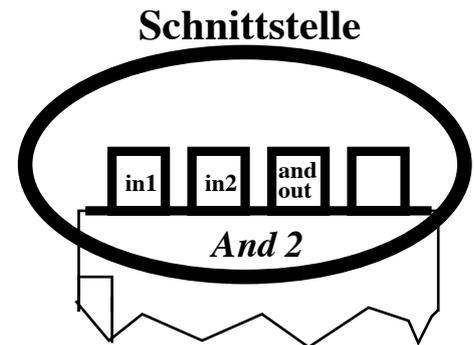
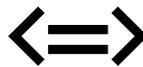
Die funktionelle Modellierung ermöglicht, die Aufgaben der Schaltung in separate funktionelle Gruppen zu gliedern. Diese können genau wie elektronische Bauteile in getrennten Prozessen oder in verschiedenen Teams entwickelt und getestet werden, bevor sie zum Schluß zu einer Beschreibung zusammengefügt werden. Durch dieses Baukastenprinzip können komplexe Schaltungen leicht aus einfachen Gruppen zusammengefügt werden.

Die wichtigste Struktur für die funktionelle Beschreibung ist der PROZESS. Prozesse arbeiten wie Computerprogramme sequenziell und werden von Signalen gestartet. Eine andere sehr wichtige Struktur sind Ablaufsteuerungen, so genannte FSMs¹⁵. Auf beide wird später noch näher eingegangen.

¹⁵Finite State Machine (dt.: Endlicher Zustandsautomat).

Schnittstelle (entity)

```
ENTITY And2 IS
  PORT ( in1      : IN bit;
         in2      : IN bit;
         and_out  : OUT bit
        );
END And2;
```



Architektur (architecture)

```
ARCHITECTURE Verhalten OF And2 IS
BEGIN
  PROCESS (CLK,RESET)
  ....
END Verhalten ;
```

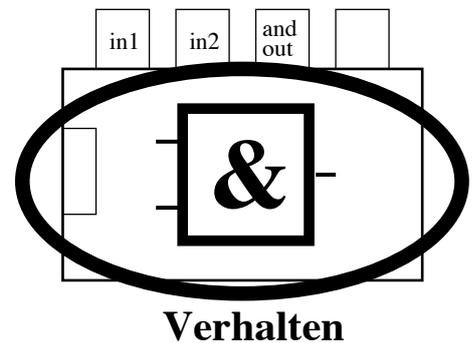
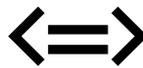


Abbildung 11: Verhaltens- Modellierung von VHDL in der Architecture: Diese Modellierung erzeugt die Funktion von Bauteilen. Mehrere Bauteile können dann mittels ihrer Entities zu komplexen Systemen miteinander verbunden werden.

4.4 VHDL-Syntax

Der folgende Abschnitt soll einen Überblick über die typischen Konstrukte geben, die in einer VHDL-Beschreibung verwendet werden.

4.4.1 Kommentare und Bezeichner

Kommentare im Programmcode werden durch zwei einfache Bindestriche/Minuszeichen eingeleitet:

```
-- Kommentar
```

Bei der Wahl von Bezeichnern für Variablen, Typen, Prozessnamen oder Funktionen gelten die üblichen Einschränkungen für Programmiersprachen. Groß- und Kleinschreibung spielt in VHDL keine Rolle und kann daher beliebig eingesetzt werden.

4.4.2 Datentypen, Deklarationen und Instanziierungen

VHDL ist 'streng getypt' - jede Variable und jedes Signal muss zunächst zu Beginn der Architecture mit einem entsprechenden Typ deklariert werden, der nicht vom Compiler dynamisch/automatisch an die Verwendung angepasst wird. Die im täglichen Gebrauch am häufigsten verwendeten skalaren Typen in VHDL sind *integer*, *boolean*, *std_logic* und so genannte *enumeration types*. Auf ihre Definition und Verwendung wird im Folgenden eingegangen:

Der Typ Integer ist in VHDL vordefiniert und umfasst alle Zahlen von $-2\,147\,483\,647$ bis $+2\,147\,483\,647$. Instanziierungen, also die Deklaration von Konstanten, Variablen und Signalen), dieses Typs sehen z.B. wie folgt aus:

```
constant number_of_bytes: integer := 4;  
constant number_of_bits: integer := 8 * number_of_bytes;  
variable a := 0;  
signal counter: integer range 0 to 255;
```

Einer der vordefinierten *enumeration types* ist *boolean*. Die folgende Zeile Code stellt die tatsächliche (bereits implementierte) Typdefinition dar:

```
type boolean is (false , true);
```

Eigene Typen können also sehr einfach selbst mittels Aufzählung der möglichen Zustände definiert werden. Eine Verwendung (Instanziierung) des Typs sieht dann z.B. wiederum wie folgt aus. Dabei steht rechts von := der Startwert (Instanzierungswert) des Signals, bzw. der Variable:

```
signal switch_open: boolean:= false ;
```

Da VHDL entworfen wurde, um digitale Hardware zu beschreiben, ist ein zentraler Typ notwendig, der digitale Werte repräsentiert. Um die elektrischen Eigenschaften unserer Signale zu berücksichtigen, hat die IEEE einen Typ *std_logic* in der package *std_logic_1164* standardisiert und bereits vordefiniert, der in fast allen Entwürfen verwendet wird:

```

type std_logic is ('U', — Uninitialized
                    'X', — Forcing Unknown
                    '0', — Forcing zero
                    '1', — Forcing one
                    'Z', — High Impedance
                    'W', — Weak Unknown
                    'L', — Weak zero
                    'H', — Weak one
                    '-'); — Don't care
signal b, c, d: std_logic := '0';

```

Mit diesem Typ werden Sie im Versuch am meisten arbeiten. Dabei verwenden Sie nur die Zustände '1' und '0', die entsprechend den Wert 'AN' und 'AUS' repräsentieren.

Um mehrdimensionale Typen zu erhalten, lassen sich skalare Typen in Arrays zusammenfassen:

```

type point is array (2 downto 0) of integer range 0 to 255;
constant origin : point := (0, 0, 0);
type matrix is array (2 downto 0, 2 downto 0) of integer range 0 to 255;

```

Die oben erwähnte Package std_logic_1164 hält aber bereits einen unbeschränkten, eindimensionalen Array-Typ ('Vektor') namens std_logic_vector aus mehreren standard-logic Werten bereit, der wie folgt instanziiert werden kann:

```

signal my_vec : std_logic_vector(7 downto 0) := "11100011";

```

Ein Vektor ist eine geordnete Menge von einzelnen std_logic Signalen, repräsentiert aber noch keine Binärzahl, auch wenn er bereits so aussieht. Ein Typ, der hierzu verwendet werden kann, ist *unsigned*. Er ist sehr ähnlich zu benutzen wie *std_logic_vector*, mit dem Unterschied, dass es vordefinierte Rechen- und Vergleichsoperatoren gibt, sodass das Resultat immer als nicht-negative Binärzahl interpretiert werden kann. Er ist im *numeric_std*-Package definiert, die als allererste Library in der zweiten Zeile Ihres Codes geladen werden sollte, wenn Sie ihn verwenden wollen:

```

use ieee.numeric_std.all;

```

Die wichtigsten Operatoren sind:

- + Addition
- - Subtraktion
- * Multiplikation
- < kleiner als
- > größer als
- >= größer gleich
- = gleich
- /= ungleich

4.4.3 Datentypen-Konversion

VHDL ist, wie oben erwähnt, eine sehr strikt typisierte Sprache. Das heißt in diesem Fall, dass man Signale unterschiedlicher Typen erst explizit in den richtigen Typ umwandeln muss ('type-casting'), bevor man diese einander zuweisen oder miteinander vergleichen kann. Die folgenden Beispiele sollen die wichtigsten Konvertierungen einführen.

- *unsigned* nach *std_logic_vector*:

```
my_vec <= std_logic_vector(my_unsign);
```

- *std_logic_vector* nach *unsigned*:

```
my_unsign <= unsigned(my_vec);
```

- *integer* nach *unsigned*:

```
my_unsign <= to_unsigned(23, 8);
```

Man beachte, dass in diesem Fall die Länge des unsigned Vektors mit 8 angegeben werden muss.

- *unsigned* nach *integer*:

```
my_integer <= to_integer(my_unsigned);
```

Die beiden Funktionen *to_unsigned()* und *to_integer()* sind in der Package *numeric_std* definiert.

4.4.4 Zuweisungen

Signalen können Werte wie folgt zugewiesen werden, z.B.

```
c <= '1';  
d <= b AND (NOT c);  
b <= Schalter1;  
my_vec <= "00000000";  
my_vec2 <= (1 => '1', 3 => '1', others => '0');  
my_vec 3<= "010" & '1' & "0010";  
my_unsigned <= x"1234" + x"0001";
```

Dabei wurde im zweiten Beispiel den Bits 1 und 3 von *my_vec* eine logische '1' zugewiesen, und allen anderen Bits eine '0'. Die logischen Operatoren **and**, **or**, **nand**, **nor**, **xor**, **xnor** and **not** akzeptieren Operanden, die vom Typ *boolean* oder *std_logic* sind, und resultieren auch in einem Ergebnis jeweiligen Typs.

Die abgebildeten Zuweisungen werden vom Compiler außerhalb von Prozessen (s.u.) in direkte Leitungsverbindungen übersetzt. Der Code wird nicht sequentiell ausgeführt, sondern sorgt für permanente Verknüpfungen.

Aufgabe 6: Überlegen Sie sich (rein Formelmäßig, also ohne den VHDL-Syntax einzuhalten), wenn Sie wollen mit einer Wahrheitstabelle, wie Sie mit logischen Operatoren eine Abstimmanlage realisieren könnten. Ziel ist es, eine Lampe dann einzuschalten, wenn mindestens die Hälfte von vier Eingangsschaltern auf 'an' steht.

4.4.5 Prozesse

Prozesse sind zentraler Bestandteil der meisten funktionellen Beschreibungen. Sie reagieren auf Signale und arbeiten eine Liste von Kommandos sequenziell ab. Dabei modifizieren sie andere Signale, die als Ausgabe des Prozesses dienen. Eine VHDL-Beschreibung kann beliebig viele von ihnen enthalten. Prozesse können gleichzeitig starten, wenn sie durch das gleiche Signal ausgelöst werden. Damit wird die parallele Struktur der Elektronik abgebildet.

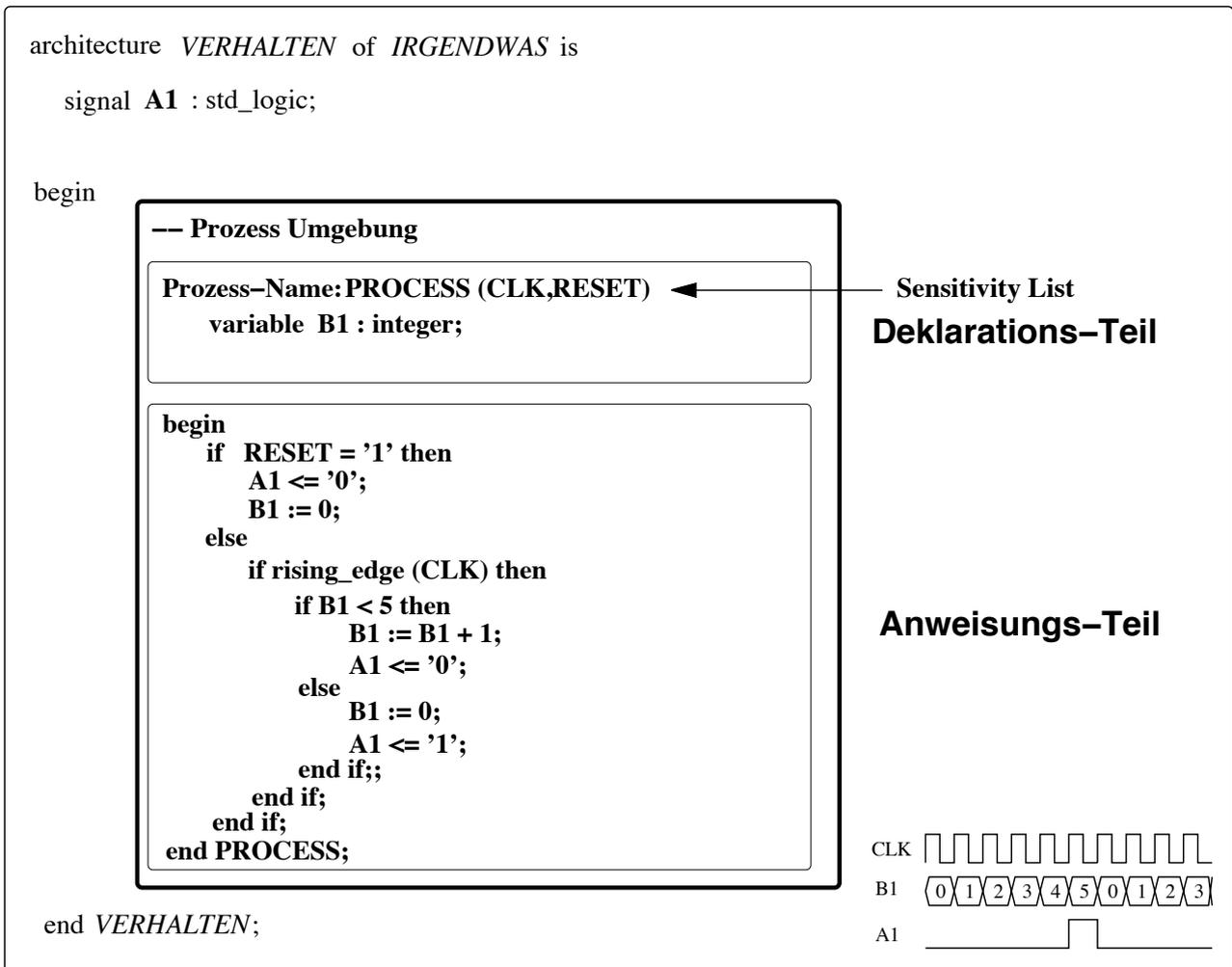


Abbildung 12: Das Beispiel zeigt den Aufbau der Prozess-Umgebung. Auch Prozesse bestehen aus einem Deklarations und einem Anweisungsteil. Die Befehle des Anweisungsteils werden sequenziell abgearbeitet. Der Prozess wird getriggert durch Signale in der Sensitivity-List des Prozess-Kopfes, oder durch eine Wait-Anweisung im Anweisungsteil.

Die wichtigsten Merkmale von Prozessen sind:

- Prozesse arbeiten sequenziell, genau wie Computerprogramme.
- Prozesse werden durch Signale in der SENSITIVITY LIST im Prozess-Kopf gestartet.
- Prozesse sind parallele Strukturen, d.h. beliebig viele Prozesse können gleichzeitig starten.

- Prozesse manipulieren Signale (Prozessausgabe).
- Prozesse können lokale Variablen deklarieren (im Deklarations-Teil).
- Signale des Prozesses müssen im Deklarationsteil der Architecture deklariert werden.
- Variablen des Prozesses ändern sich im Moment der Zuweisung.
- Signale des Prozesses ändern sich erst nach Beendigung des Prozesses (Delta-Zyklus).

Abbildung 12 zeigt den Grundaufbau eines Prozesses. Im Deklarations-Teil steht der Prozess-Kopf mit dem Prozess-Namen und der Sensitivity-List. Die Sensitivity-List ist der Inhalt der runden Klammer hinter dem Wort PROCESS. Sie enthält diejenigen Signale, die den Prozess starten (triggern). Starten bedeutet, daß die Befehle des Anweisungsteiles sequenziell abgearbeitet werden. Jede Änderung eines Signals der Sensitivity-List bewirkt dabei den Start des Anweisungsteils. Im Beispiel von Abbildung 12 bedeutet das, daß mit jeder Zustandsänderung von 'CLK' oder 'RESET' der Prozess erneut gestartet wird, also zweimal pro 'CLK'-Zyklus. Das 'RESET'-Signal könnte in diesem Fall also auch außerhalb eines 'CLK'-Zyklus (asynchron mit 'CLK') den Prozess starten. Würde 'RESET' hingegen aus der Sensitivity-List entfernt, so wird der Reset mit der 'CLK'-Signalflanke synchron ausgeführt. Der Zustand des Signals 'RESET' wird dann nur bei jeder 'CLK'-Flanke abgefragt.

Innerhalb von Prozessen gibt es im wesentlichen zwei besonders häufig gebrauchte Konditionale – die *if-then-else*-Anweisung und das *case*-Konstrukt. Ein Beispiel soll die Verwendung von IF erläutern. Dabei sind die Signale wie oben definiert.

```

if unsigned(my_vec) < my_unsigned(15 downto 8) then
    my_vec <= std_logic_vector(my_unsigned(15 downto 8));
elsif b = '1' and my_vec(7) = '0' then
    Lampe <= '1';
else
    Lampe <= '0';
end if;

```

Manchmal ist es notwendig, mehrere fast gleiche Zuweisungen zu wiederholen. Dazu können *for*-Schleifen wie im folgenden Beispiel benutzt werden. Man beachte, dass die *for*-Schleife in diesem Fall nicht sequentiell durchlaufen wird, sondern der Compiler einfach acht Verknüpfungen erstellt! Um wirklich Dinge nacheinander zu erledigen, benötigen Sie eine Ablaufsteuerung (s.u.).

Die beiden Signale *PHY_DATAOUT* und *crc16_buf* sind Vektoren wie *my_vec*.

```

for i in 0 to 7 loop
    PHY_DATAOUT(i) <= not crc16_buf(7-i);
end loop;

```

Bei Variablen wird im Gegensatz zu Signalen statt des '<=' ein ':=' bei der Zuweisung verwendet. Der wesentliche Unterschied zu Signalen liegt darin, dass Variablen sich sofort ändern. Signale werden erst am Ende des Prozesses bzw. beim Start des Prozesses auf der Clockedge zugewiesen. Außerdem werden Signale im Deklarationsteil der Architecture deklariert, Variablen im Deklarationsteil des Prozesses. Variablen behalten ihren Wert zudem nicht bis zum nächsten Start des Prozesses, während Signale, die im Prozess verwendet werden, in Registern gespeichert werden.

Aufgabe 7: Angenommen, die folgenden Zuweisungen betreffen Signale und stünden in einem Prozess, der gerade zum ersten Mal gestartet wurde. Zuvor wurden alle Signale mit Einsen initialisiert. Welchen Wert haben die Signale nach einem Clockzyklus? Was ändert sich, wenn es keine Signale, sondern Variablen wären?

```
b <= '0';
my_vec <= "010" & b & "0010";
my_unsigned <= x"1234" + x"0001";
my_unsigned(15 downto 8) <= unsigned(my_vec);
```

Die Anzahl der Operationen in einem Prozess hat einen kritischen Einfluss auf die maximal erreichbare Arbeitsfrequenz eines Entwurfes. Das Synthese-Programm legt diese so fest, dass alle Befehle eines CLK-synchronen Prozesses auch innerhalb eines CLK-Zyklus abgearbeitet werden können. Aus diesem Grund ist es ratsam, komplexe Operationen auf mehrere Prozesse zu verteilen und diese z.B. durch eine Ablaufsteuerung oder ZUSTANDSAUTOMATEN zu koordinieren.

4.4.6 Zustandsautomaten

Eine weitere wichtige Struktur in der funktionellen Modellierung sind so genannte Zustandsautomaten oder FSMs. Eine FSM ist eine Logikschaltung, die gesteuert durch pulsartige oder häufig auch periodische externe Signale eine Abfolge von Zuständen durchläuft. Ihre Theorie bildet die Basis jedweder Form von Automation. Somit stellen sie auch ein zentrales Konzept bei der Entwicklung von digitalen Anwendungen mit VHDL dar. In VHDL werden FSMs durch Prozesse mit einer case-Struktur repräsentiert.

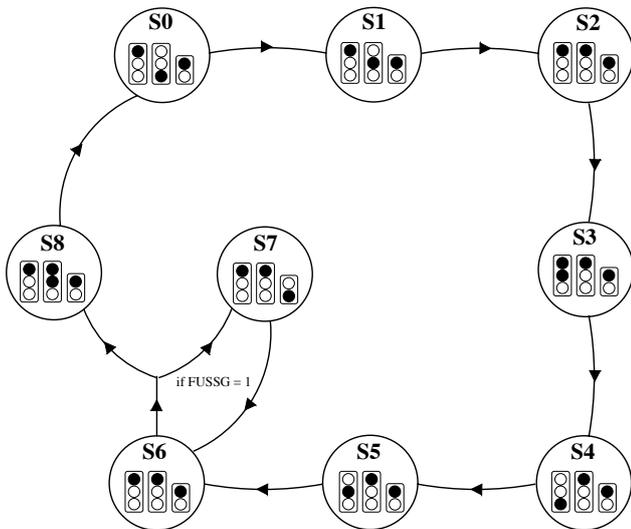


Abbildung 13: Abfolge der Zustände S0 bis S8 der Ampelschaltung, die jeweils nach mehreren Sekunden ineinander übergehen, wobei die Phasen, in denen eine Ampel 'grün' zeigt, um einen beliebigen Faktor 'k' verlängerbar sind. Nach Zustand S6 verzweigt das Übergangsschaltnetz einmalig zu S7 und wieder zurück zu S6, wenn der Fußgängertaster gedrückt wurde.

Bestandteile einer FSM sind ein Signal, in dem der aktuelle Zustand festgehalten wird, ein ÜBERGANGSSCHALTNETZ, das den Wechsel zwischen diesen Zuständen regelt und ein AUSGANGSSCHALTNETZ, das in Abhängigkeit vom aktuellen Zustand Signalen Werte zuweist. Typen, die z.B. Zustände in einer FSM beschreiben, lassen sich auf folgende Weise als *enumeration type* definieren und instanzieren:

```
type alu_function is (disable , pass , add , subtract , multiply , divide);
signal my_alu : alu_function;
```

Case-Anweisungen werden (wie hier) gerne zur Implementierung von FSMs (s.u.) genutzt. Beim case-Konstrukt müssen immer alle Werte angegeben werden, die das referenzierte Signal annehmen kann. Dabei hilft das Keyword *others*, das als letztes aufgelistet wird:

```

case my_alu is
  when disable =>    -- mache nichts
  when add =>
    my_unsigned <= my_unsigned + my_unsigned2;
  when subtract =>
    my_unsigned <= my_unsigned - my_unsigned2;
  when others =>
    my_alu <= pass;
end case;

```

Als einfachstes Beispiel für eine FSM-Anwendung wird in der Elektronik gerne eine Ampelschaltung angeführt. Dabei sollen die vier Ampeln einer Verkehrskreuzung abwechselnd je einer Straße die Durchfahrt freigeben und für die andere Straße 'rot' zeigen (s. Abbildungen 13 und 14).

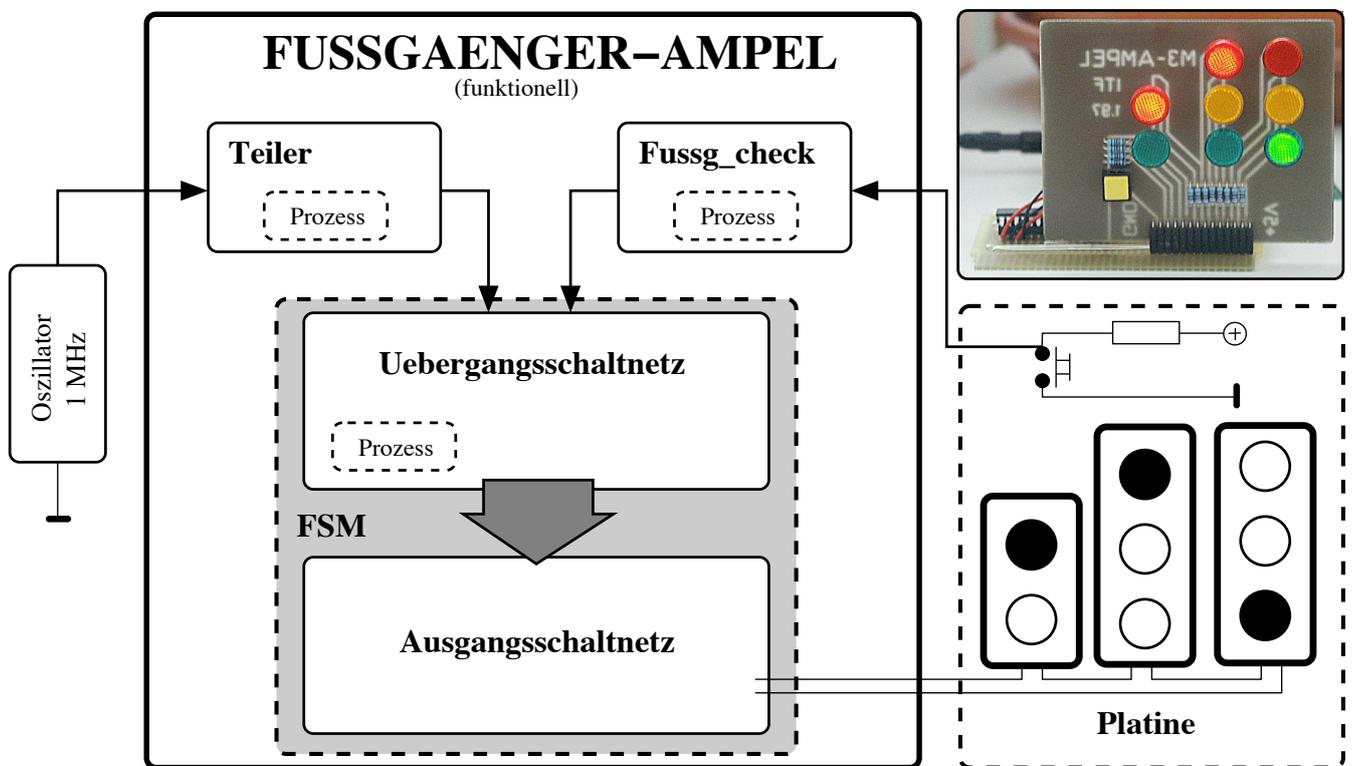


Abbildung 14: Blockschaltbild einer Ampelschaltung mit Fußgängerüberweg - realisiert als FSM.

4.4.7 Unterprogramme, Packages und Libraries

Der Vollständigkeit halber sollen hier noch *Functions*, *Procedures*, *Packages* und *Libraries* erwähnt werden, die dazu dienen, den Code übersichtlicher und modularer zu gestalten. Meistens wird das allerdings bereits durch Components und Prozesse erreicht. Functions können ähnlich wie mathematische Funktionen mehrere Argumente überreicht werden und sie liefern genau einen Rückgabewert. Procedures sind Functions sehr ähnlich, stehen aber für ein vollständiges VHDL Statement und liefern keinen Rückgabewert.

4.5 Components

Als 'Component' bezeichnet man einen vollständigen VHDL-Entwurf aus einer vorhergehenden Entwicklungsphase oder einer vorgefertigten Bibliothek. Components können auch ihrerseits strukturelle Beschreibungen sein und selbst wieder Components enthalten. Auf diese Weise entsteht bei umfangreichen Projekten eine Hierarchie von Entwürfen, deren Abstraktionsgrad nach oben hin zunimmt (s. Abbildung 15). Dabei ist die unterste Ebene immer funktionell modelliert, die oberste oft rein struktural. In den Zwischenebenen ist es möglich, funktionelles und struktureles Design (auch innerhalb eines Entwurfes) zu vermischen.

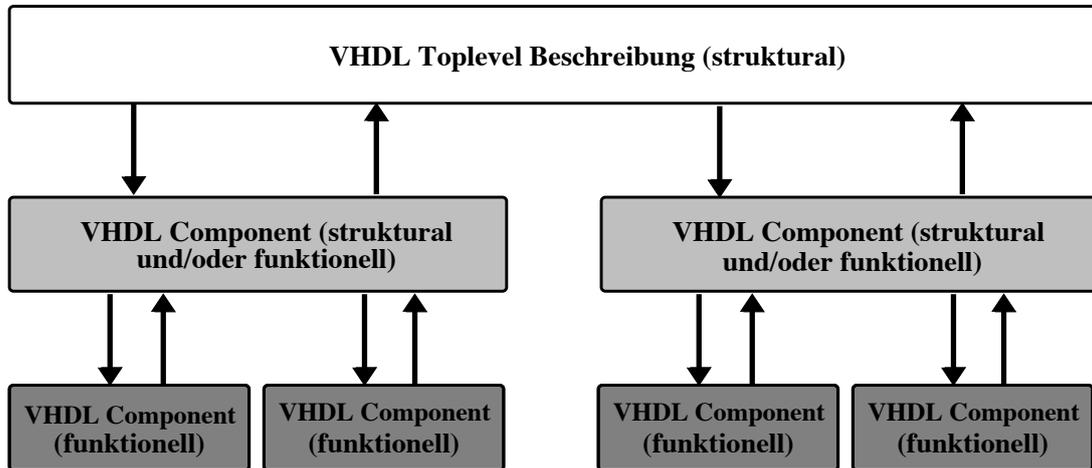


Abbildung 15: Hierarchie eines VHDL-Projektes.

Components werden im Deklarationsteil der Architecture mit ihrer vollständigen Entity deklariert und dann im Anweisungsteil instanziiert. Mit einer 'PORT MAP' werden die Signale der Components untereinander und auch mit der Entity des Entwurfes verknüpft.

Soll eine Component mehrfach instanziiert werden, wie z.B. eine Reihe FlipFlops für einen Zähler, so lässt sich diese Arbeit mit der GENERATE-Anweisung automatisieren. Innerhalb einer Schleife wird dabei die gewünschte Anzahl Components generiert, wobei sich über die Port Maps Ein- und Ausgänge entsprechend miteinander verknüpfen lassen. In vielen Fällen ist jedoch eine funktionelle Beschreibung einfacher und platzsparender als die zahlreiche Instantiierung von Components.

Der Einsatz beider Modellierungstechniken (funktionell und struktural) ermöglicht die Aufteilung der Aufgaben einer Schaltung in separate funktionelle Gruppen. Diese können in getrennten Arbeitsprozessen oder in verschiedenen Teams entwickelt und getestet werden. Es entsteht eine Art Baukastenprinzip, mit dem komplexe Schaltungen aus einfacheren Modulen zusammengefügt werden können.

5 Versuchsdurchführung

Dieser Versuch soll einen Überblick über die zur Entwicklung einer modernen Betriebs- und Auslese-elektronik für einen Röntgen-/Gammadetektor notwendigen Schritte vermitteln.

Ausgehend vom analogen Detektorsignal eines Photomultipliers wird dessen Output zunächst mittels eines ADCs (Analog-Digital-Converter) digitalisiert. Die Ansteuerung dieses Bausteines sowie die digitale Weiterverarbeitung der Signale werden Sie in einem FPGA (Field Programmable Gate Array) mit Hilfe der Hardwarebeschreibungssprache VHDL realisieren. Diese Aufgabe umfasst den Entwurf eines “Sequenzers” zum Betrieb des ADCs, die zur Unterdrückung von Hintergrundereignissen notwendige Analyse der Signalform (Bestimmung der Anstiegszeit und des Maximalwertes des Detektorsignals) und die anschließende Formatierung und Übertragung der Daten als “Event-Packets” zu einem Datenerfassungssystem. Hierbei lernen Sie die typischen Elemente einer “Hardware-Synthese” kennen: Skizzierung der notwendigen Komponenten und Module, Programmierung der einzelnen Steuer- bzw. Analyseprozesse (ISE Webpack), Simulation des Entwurfs am Computer mit ModelSim oder iSim, Syntheseprozess für einen XILINX Spartan-3 FPGA und Inbetriebnahme der Hardware als Schnittstelle zwischen Detektor und Datenerfassungssystem (s. Abb. 16 und 17).

Die Vermessung und Kontrolle der analogen Detektorsignale sowie die Überprüfung der Eigenschaften der entwickelten digitalen Elektronik wird dabei mit einem “Mixed-Signal” (analog+digital) Oszilloskop durchgeführt. Abschließend erfolgt die Vermessung des Strahlungshintergrundes und der Eigenschaften verschiedener radioaktiver Quellen mit dem entstandenen Versuchsaufbau. Die auf diese Weise gewonnenen Daten werden mit Hilfe einer Software analysiert; Eigenschaften des Detektors (Energie/Zeitauflösung) und der verwendeten radioaktiven Quellen werden von Ihnen ermittelt.

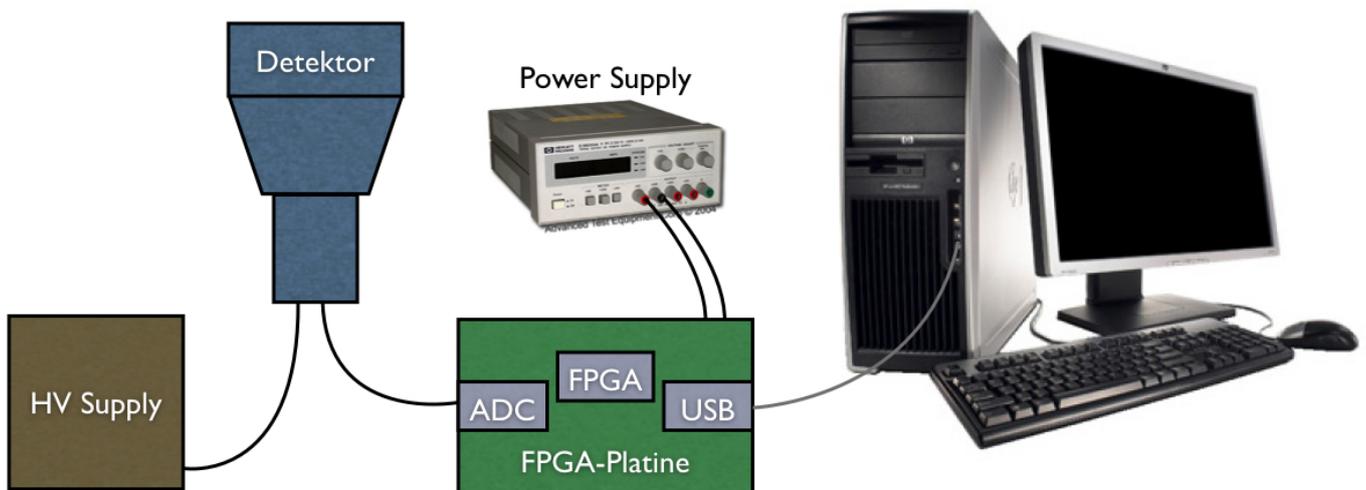


Abbildung 16: Die einzelnen Komponenten des Versuchsaufbaus

5.1 Inbetriebnahme des Detektors

Nehmen Sie den Detektor in Betrieb (Anschließen an Hochspannungsquelle) und schauen Sie sich das Ausgangssignal am Oszilloskop an.

Aufgabe 8: Messen Sie die Anstiegszeiten sowie die maximale Amplitude im NaI und CsI bei Bestrahlung des Detektors mit Am241.

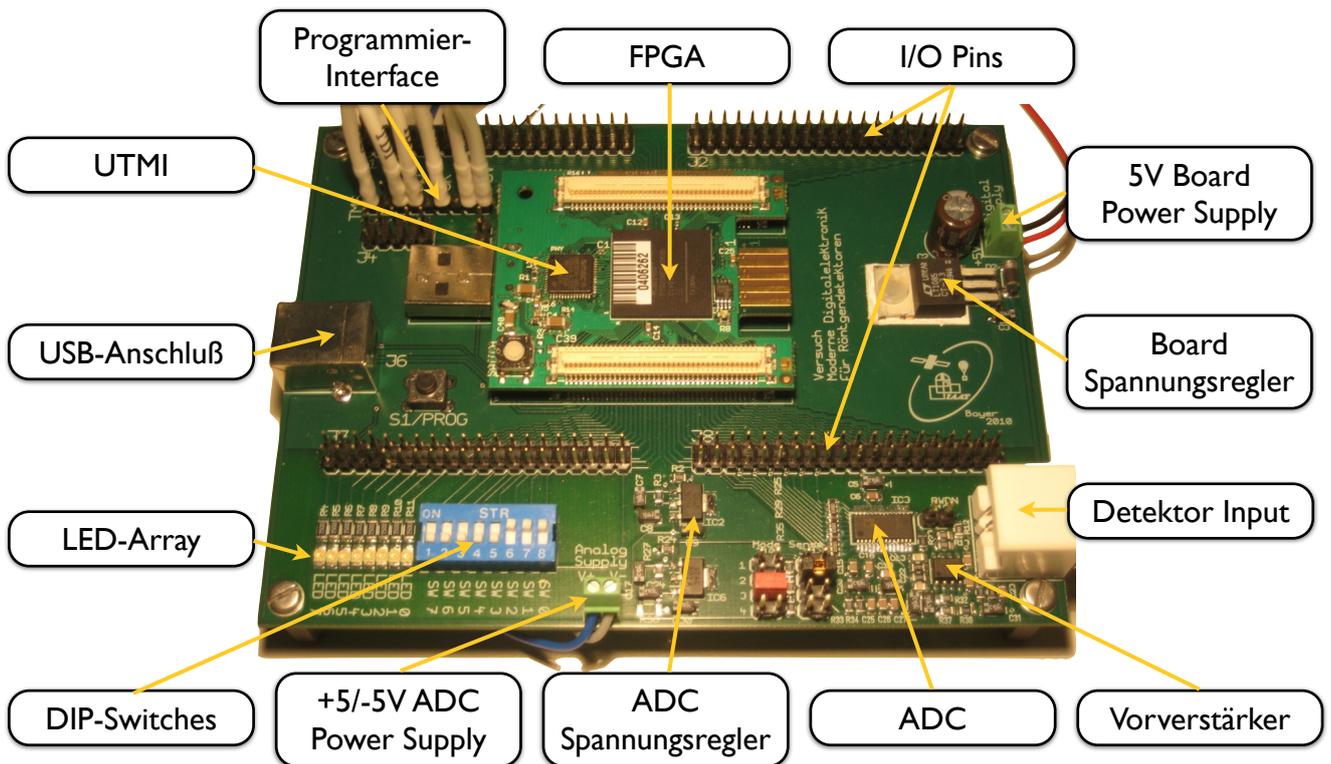


Abbildung 17: Die abgebildete Platine ist das Herzstück des Versuchs. Sie trägt auch das FPGA-Board und den ADC und bildet das Interface zwischen Detektor und Datenerfassungssystem.

Aufgabe 9: Bestimmen Sie als nächste Aufgabe das Gesichtsfeld des Kollimators zunächst rein auf Grund seiner Abmessungen.

5.2 VHDL Programmierung

Im VHDL-Teil des Versuches werden Sie einige Entwürfe schreiben und testen. Die ersten sollen Ihnen die Gelegenheit geben, einfache VHDL-Konstrukte erstmals zu benutzen und deren Funktion besser zu verstehen. Im zentralen Entwurf werden Sie die Komponente “usb_serial” einbinden um damit die ADC-Daten direkt an den Rechner zu schicken und anzusehen. Der letzte Teil stellt das eigentliche Ziel dar: Es ist Ihre Aufgabe, einen VHDL-Prozess zu schreiben, der die ADC-Werte bereits im FPGA verarbeitet, so dass nur noch die Amplitude und die Anstiegszeit eines γ -Ereignisses über den USB an den Rechner übertragen werden müssen. Anschließend werden Sie mit dem so entstandenen Datenerfassungssystem die Eigenschaften des Phoswichdetektors analysieren.

Aufgabe 10: Schreiben Sie in Ihrer ersten Zeile Code die Anweisung, dass die Leuchtdiode LED0 dauerhaft an sein soll. Fügen Sie in die Zeile darüber einen Kommentar ein, z.B. - - Aufgabe 8, Lampe dauerhaft an

Aufgabe 11: Schreiben Sie in Ihrer nächsten Zeile Code die Anweisung, dass die Leuchtdiode LED1 den Zustand von SWITCH0 annehmen soll (d.h.: ist der Schalter auf '1', soll auch die Lampe an sein; ist der Schalter auf '0', soll die Lampe aus sein). Versuchen Sie dabei, so einfach wie möglich zu bleiben - können Sie eventuell auf das IF-Statement verzichten?

Aufgabe 12: Ergänzen Sie ihren Entwurf so, dass Sie den Zustand von LED2 abhängig von den Schaltern SW1 bis SW4 machen. Es bietet sich an, Ihren Entwurf aus Aufgabe 6 umzusetzen. Können Sie die Aufgabe mit einer einzigen Zuweisung lösen?

Fügen Sie Ihrem Entwurf nun ein externes Clocksignal hinzu, um auch zeitgesteuertes Verhalten zu realisieren: Der FPGA im Praktikumsversuch hat Zugang zu einer konfigurierbaren Taktleitung, die die Taktfrequenz 60 MHz zur Verfügung stellt, wenn Sie die entsprechenden Schritte befolgen:

1) Erweitern Sie Ihre Entity um das folgende Signal:

```
CLK:                                in STD_LOGIC;
```

2) Fügen Sie den entsprechenden Pin für das CLK-Signal im UCF constraint-file ein:

```
NET "CLK"                            LOC = D9;
```

Jetzt können Sie das Clocksignal in der nächsten Aufgabe verwenden!

Aufgabe 13: Schalten Sie LED3 mit einer Frequenz von 1 Hz an und aus. Schreiben Sie dazu in VHDL ihren ersten Prozess 'Frequenzteiler', den Sie am besten mit einem Zähler realisieren, der ansteigende Flanken des CLK-Signals zählt und bei einem gewissen Wert wieder auf Null zurückgesetzt wird. Schalten Sie die LED dann in Abhängigkeit des Zählerstandes ein oder aus.

Aufgabe 14: Lesen Sie nun die vom ADC digitalisierten Werte ein. Der ADC muss (genau wie die LED) mit einem *std_logic*-Signal getaktet werden, damit er die Spannungen bei jeder ansteigenden Flanke digitalisiert. Beachten Sie, dass Sie den ADC mit 30 MHz takten, das ist die schnellste Frequenz, mit der der ADC arbeiten kann. Kopieren Sie hierzu einfach den Prozess der Aufgabe davor und passen Sie die Werte sowie die Signalnamen an. Geben Sie zum Test die unteren Bits der eingelesenen Daten auf den noch freien LEDs aus.

Aufgabe 15: Zur Übertragung an den Rechner können Sie eine vorgefertigte Komponente aus dem Internet benutzen. Das wird in der Praxis sehr häufig gemacht. Sie finden diese im Verzeichnis "templates". Fügen Sie diese per "add source"-Knopf dem Projekt als Kopie hinzu. Zudem müssen Sie sich überlegen, wie Sie die 10 Bit eines ADC-Wertes über die USB-Schnittstelle, die immer nur 8 Bit zur Zeit zulässt, verschicken. Wenn Sie sich dabei an folgende Vorgabe halten, dann können Sie die bereits geschriebenen Entpackungsprogramme benutzen: Diese erwarten, dass die obersten zwei Bit markieren, welches Byte gerade übertragen wird. "00" steht für das low-Byte und "0100" für das high-Byte.

Aufgabe 16: Schreiben Sie eine Komponente, die die ADC-Werte kontinuierlich überprüft, und im Falle eines γ -Ereignisses im Detektor die Amplitude und Anstiegszeit bestimmt und an den Rechner schickt. Genau wie die Amplitude soll die Anstiegszeit als 10 Bit (oder 11 Bit) unsigned übertragen werden. Das Programm auf Rechnerseite erwartet nun immer 4 Bytes, wobei die ersten beiden Bytes “00” und “0100” die maximale Amplitude sein sollen, die beiden hinteren ebenfalls, mit “10” und “1100” als den oberen Bits die Anstiegszeit. Um einen Schwellenwertkriterium für ein Ereignis zu bestimmen (Triggerlevel), benutzen Sie Ihre Ergebnisse aus Aufgabe 14.

5.3 Vermessung der Detektoreigenschaften

Nachdem die Datenerfassung und -übertragung zum PC nun funktioniert, sollen von Ihnen noch einige Messungen zu verschiedenen Detektoreigenschaften durchgeführt werden. Zunächst sollen Sie die Datenerfassung am PC so optimieren, dass lediglich die Impulse aus dem NaI Szintillator registriert werden und die CsI-Pulse möglichst unterdrückt werden. Diese Selektion geschieht über die Festlegung eines gültigen Zeitfensters für die Anstiegszeiten.

Aufgabe 17: Erstellen Sie ein Histogramm der Anstiegszeiten bei Bestrahlung des Detektors mit der Quelle von vorne (NaI) und von oben (CsI). Auf diese Weise können Sie die Signale aus dem NaI identifizieren. Bestimmen Sie die Lage der Maxima in den Plots und rechnen Sie diese in Zeiten um. Vergleichen Sie die Ergebnisse mit denen aus Aufgabe 8. Legen Sie nun eine untere und obere Schranke für das Zeitfenster im NaI fest.

Im nächsten Schritt soll eine Kalibration der Energieskala vorgenommen werden. Hierzu ist es notwendig, eine Umrechnung der ADC-Kanäle in Photonenenergien vorzunehmen. Anschließend soll die relative Energieauflösung $\Delta E/E$ des Detektors festgestellt werden.

Aufgabe 18: Erstellen Sie ein Histogramm der maximalen Amplituden bei Bestrahlung des Detektors mit der Quelle von vorne im Zentrum aus wenigen Zentimetern Entfernung. Bestimmen Sie über die zwei im Spektrum vorhandenen Peaks die Umrechnung auf die Energieskala und geben Sie die relative Energieauflösung an. Führen Sie dieselben Schritte für Bestrahlung des Detektors von vorne in einem Eck aus. Wie wird sich hierbei die Energieauflösung ändern? Weshalb?

Aufgabe 19: Bestimmen Sie nun noch die Hintergrund-Zählrate und nehmen Sie ein Spektrum des Hintergrundes im Labor auf.

Vervollständigen Sie nun den Aufbau des Gammateleskopes durch einen Röhrenkollimator.

Aufgabe 20: Um welchen Faktor wird der Hintergrund durch den Röhrenkollimator verringert?

Aufgabe 21: Indem Sie die Zählrate des Detektors als Funktion des Drehtischwinkels messen, haben Sie eine weitere Möglichkeit, das Gesichtsfeld des Kollimators zu überprüfen. Richten Sie dazu den Kollimator zunächst genau auf die Quelle aus und drehen Sie ihn dann 4° nach links. Messen Sie von dort aus ausgehend bei jedem halben Grad die Zahl der Ereignisse in 30 Sekunden, bis Sie bei 4° rechts angekommen sind, und werten Sie die Kurve beim Anfertigen des Protokolles aus. Vergleichen Sie das Ergebnis mit dem der geometrischen Vermessung des Kollimators in Aufgabe 9.

6 Hinweise zur Anfertigung des Protokolls

In diesem letzten Teil geben wir noch ein paar Tipps zur Anfertigung der Protokolle. Auch wenn einige davon eigentlich offensichtlich und selbstverständlich erscheinen mögen - wir bitten, diese trotzdem beim Schreiben nochmals anzusehen und vor allem die Punkte im letzten Abschnitt zu beherzigen.

Das Protokoll sollte mit einer Einleitung beginnen, die die Inhalte und Ziele des Versuchs kurz zusammenfasst. Danach sollte man einen Theorieteil einfügen, in dem der astrophysikalische Zusammenhang und die Grundlagen des Detektorprinzips angerissen werden. Hier empfiehlt es sich, an den Aufgaben 1 - 7 entlangzugehen und diese gleich zu beantworten. Anschließend dann einen Teil 'Versuchsdurchführung' hinzufügen, der die im Versuch durchgeführten Schritte sequenziell beschreibt - jeweils mit wenigen Sätzen, was in diesem Teil weshalb zu tun war, wie die Messung/der Schritt durchgeführt wurde, was dabei herauskam und was man daraus schließen und nun damit anfangen kann. Am Schluss wäre ein Abschnitt 'Fazit' oder 'Zusammenfassung' angebracht, der die wichtigsten Ergebnisse in Bezug auf die anfangs beschriebenen Ziele nochmals festhält.

Vor dem Abgeben des Protokolls sollte unbedingt eine elektronische Rechtschreibkorrektur durchgeführt werden. Eine solche steht heute auf allen Systemen zur Verfügung. Danach sollten ALLE Mitglieder der Praktikumsgruppe das Protokoll lesen und mit der Abgabe einverstanden sein. Das ist ein Vorgang, den man sich als Wissenschaftler zur Gewohnheit machen sollte. Bevor ein Dokument mit Ihrem Namen darauf den Schreibtisch verlässt oder ins Internet gestellt wird, sollten Sie sicherstellen, dass es nicht nur inhaltlich, sondern auch vom äußeren Eindruck her qualitativ in Ordnung ist - erst Recht, wenn Sie nicht selbst der Verfasser sind!

A Beschreibung der Kommandozeilen Programme

Hier sollen die Kommandozeilenprogramme in etwa der Reihenfolge beschrieben werden, in der sie im Versuch benötigt werden. Programme, die mit “./” anfangen befinden sich samt Sourcecode in dem Verzeichnis “/home/student/usb-utmi-cdc-acm/programs”. Mit dem Befehl “make” können alle kompiliert werden. Nicht alle Programme in diesem Verzeichnis werden für den Versuch gebraucht.

Zunächst ein kleiner Überblick. Mit `./rdwr` werden die Daten von der Hardware über den USB empfangen oder auch zur Hardware verschickt. Danach müssen sie mit einem Konvertierungsprogramm entpackt und anschließend mit einem Druckprogramm in eine Form gebracht werden, die die folgenden Tools verwenden können.

- ./rdwr:** Dieses Programm liest die Daten vom USB und schreibt sie unverändert nach *stdout*. Auf *stdin* wartet es auf Daten, die es an das USB Peripheriegerät sendet.
- pv:** Mit *pv* kann die Menge der übertragenen Daten im laufenden Betrieb angezeigt werden.
- ./vra_convert:** Dies nimmt die Daten von `./rdwr` und entpackt sie, wie sie im FPGA verpackt worden sind. Es erwartet immer zwei Byte pro Integer, wobei die ersten beiden Bit jeweils als Markierung dafür verwendet werden, ob es sich um das high-Byte oder das low-Byte handelt.
- ./vra_print:** Druckt die Daten von `./vra_convert` in ASCII aus, einen Integer pro Zeile.
- ./vrdbl_convert:** Ähnlich wie `./vra_convert` entpackt es die Daten vom FPGA. Es erwartet immer vier Bytes – zwei für den ersten Integer, zwei für den zweiten – markiert jeweils durch die höchsten 2 Bit.
- ./vrdbl_print:** Ähnlich `./vra_print` druckt es die Daten von `./vrdbl_convert` in zwei Integer pro Zeile aus.
- ./bin.py:** Bestimmt den Bin in den eine Zeile gehört.
- ./copycol.py:** Kopiert eine Spalte.
- grep:** Der General Regular Expression Parser wird zum Auswählen von Zeilen verwendet.
- cut:** Hiermit lässt sich eine oder mehrere Spalten des Inputstream auswählen.
- ./histogram.py:** Zählt wie oft jede Eingangszeile vorkommt.
- sort -g:** Sortiert die Zeilen des Inputstreams nach einem allgemeinen numerischen Verfahren (general numeric sort).
- ./prepend_time.py:** Fügt einen Zeitstempel am Anfang jeder Zeile ein.
- tee <dateiname>:** Schreibt den Input in die Datei “<dateiname>” und nach *stdout*.
- gnuplot:** Ein leistungsfähiges Plotprogramm.

Die Tools können in einer UNIX Pipe zusammen aufgerufen werden und deren Output in eine Datei umgelenkt werden, z.B.

```
$ ./rdwr | pv | ./vra_convert | ./vra_print > daten.jedemenge
```

Folgendermaßen können die soeben erstellten Daten in eine ASCII Zahl pro Zeile verwandelt werden:

```
$ ./vra_print < daten.jedemenge | pv > daten.nochmehr
```

Man beachte, dass hier der Inhalt der Datei “daten.jedemenge” von der Shell auf den *stdin* des Prozesses `./vra_print` gelegt wird.

B Pinout

Pin Board	Pin FPGA	Funktion
J8_03	K4	ADC_CLK
J8_05	K3	ADC_D0
J8_08	J1	ADC_D1
J8_10	J2	ADC_D2
J8_09	M4	ADC_D3
J8_12	K2	ADC_D4
J8_11	P2	ADC_D5
J8_14	K1	ADC_D6
J8_13	R1	ADC_D7
J8_16	L3	ADC_D8
J8_18	L2	ADC_D9
J8_01	K5	ADC_OR
J7_18	E3	LED_0
J7_16	D1	LED_1
J7_14	D2	LED_2
J7_12	D3	LED_3
J7_08	C1	LED_4
J7_06	C2	LED_5
J7_04	B1	LED_6
J7_02	C3	LED_7
J7_20	B3	S1/PROG
J7_29	F5	SWITCH_0
J7_30	G4	SWITCH_1
J7_27	F4	SWITCH_2
J7_28	F2	SWITCH_3
J7_25	E4	SWITCH_4
J7_26	F3	SWITCH_5
J7_24	E1	SWITCH_6
J7_22	E2	SWITCH_7
-	D9	CLK

Pin FPGA	Funktion	Dir FPGA	Default
B14	DATABUS16_8	OUT	0
B4	RESET	OUT	0
D6	XCVRSELECT	OUT	1
B6	TERMSELECT	OUT	1
T10	OPMODE<0>	OUT	0
C6	OPMODE<1>	OUT	0
B12	TXVALID	OUT	0
A3	LINESTATE<0>	IN	
A5	LINESTATE<1>	IN	
A14	TXREADY	IN	
B13	VALIDH	BIDIR	
A12	RXVALID	IN	
C11	RXACTIVE	IN	
C12	RXERROR	IN	
A13	D0	BIDIR	
B11	D1	BIDIR	
C10	D2	BIDIR	
B10	D3	BIDIR	
A10	D4	BIDIR	
C9	D5	BIDIR	
A9	D6	BIDIR	
B8	D7	BIDIR	
A8	D8	BIDIR	
C8	D9	BIDIR	
C7	D10	BIDIR	
A7	D11	BIDIR	
B7	D12	BIDIR	
B5	D13	BIDIR	
C5	D14	BIDIR	
A4	D15	BIDIR	

Literatur

- Birks, J. B. (1964). *The Theory and Practice of Scintillation Counting*. Pergamon, Oxford.
- Gruppen, C. (1993). *Teilchendetektoren*. BI Wissenschaftsverlag, Mannheim, Leipzig.
- Joram, C. (2001). Particle detectors. Erhältlich unter der URL:
<http://joram.web.cern.ch/Joram/lectures.htm>.
- Knoll, G. F. (1999). *Radiation detection and measurement*. Wiley, New York, 3rd edition.
- Reichard, J., Schwarz, B., *VHDL Synthese*.
- Lehmann, G., *Schaltungsdesign mit VHDL*.