Fakultät für Physik und Astronomie
Ruprecht-Karls-Universität Heidelberg

Diplomarbeit
im Studiengang Physik

vorgelegt von
Daniel Baumeister
aus Aschaffenburg

März 1999

Entwicklung und Charakterisierung eines ASICs zur Kathodenauslese von MWPCs für das H1-Experiment bei HERA

Die Diplomarbeit wurde von Daniel Baumeister ausgeführt am
Physikalischen Institut
unter der Betreuung von
Herrn Prof. Dr. Ulrich Straumann

Zusammenfassung

In dieser Arbeit wird die Entwicklung und Charakterisierung des CMOS-ASICs *CIPix* vorgestellt. Der *CIPix* wird als Auslesechip für die innere Vieldrahtproportionalkammer des H1-Detektors im Rahmen des CIP-Upgrade-Projektes im Jahr 2000 eingesetzt.
Der Chip integriert 64 Kanäle mit ladungsempfindlichen, rauscharmen Vorverstärkern und Pulsformern. Die verstärkten Signale werden durch 64 Komparatoren mit einstellbarer Polarität und Schwellenspannung digitalisiert und durch einen vierfachen Multiplexer auf 16 Digitalausgänge geführt. Die Spannungen und Ströme zur Regelung der analogen Komponenten sowie verschiedene andere Parameter sind über ein serielles I^2C-Bus Protokoll programmier- und rücklesbar. Es werden die Architektur des *CIPix* und einzelne Funktionselemente beschrieben, sowie durchgeführte Simulationen und Meßergebnisse präsentiert.

Abstract

This thesis describes the development and characterization of the CMOS-ASIC *CIPix*. The *CIPix* will be used as a readout chip for the central inner proportional chamber (CIP) of the H1-detector for the CIP upgrade project in the year 2000.
The chip integrates 64 channels with low noise charge sensitive preamplifiers and shapers. The input signals are digitized by 64 comparators with configurable polarity and threshold level and multiplexed by 4 onto 16 digital outputs. The bias settings and various other parameters are programmable and monitorable via an I^2C-bus serial protocol. The architecture of the *CIPix* and different functional elements are described as well as the simulation and test results are presented.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Am H1-Experiment am Elektron-Proton Speicherring HERA des Deutschen Elektronen Synchrotrons (DESY) in Hamburg wird ein breites Gebiet der Forschung im Bereich der Hochenergiephysik abgedeckt. Um neue kinematische Regionen zu erschließen, sind eine Reihe von Veränderungen am Speicherring und am H1-Detektor geplant, unter anderem eine Erhöhung der Luminosität um einen Faktor 5, sowie eine Verbesserung des H1-Detektors. Das H1-Upgrade-2000 Projekt umfaßt einen Umbau der inneren Multi Wire Proportional Chamber (MWPC) und den Einsatz einer integrierten Ausleseelektronik für diese.
Diese Diplomarbeit beschäftigt sich mit der Entwicklung und Charakterisierung des Auslesechips *CIPix* für die innere Proportionalkammer. Der Schwerpunkt liegt auf der Beschreibung der Entwicklung der digitalen Komponenten des Chips. Während der Arbeit wurden alle wesentlichen Schritte einer Chip-Entwicklung durchlaufen: Von der Erstellung einer Spezifikation der künftigen Eigenschaften, über die Schaltplan-Generierung und Layout-Erzeugung bis hin zum Test des produzierten Chips.
Nach einer Beschreibung des künftigen Einsatzgebietes des Chips in Kapitel 2, wird in Kapitel 3 die Architektur des CIPix1.0 vorgestellt.
Kapitel 4 zeigt Entwickungskonzepte eines digitalen Designs auf und erläutert den Entwicklungsprozeß einer digitalen Schaltung an einem Beispiel.
In Kapitel 5 werden die digitalen Komponenten des Chips ausführlich erklärt. Es werden Aufbau und Funktion beschrieben, sowie Simulationsergebnisse gezeigt.
Kapitel 6 stellt den Messaufbau und die durchgeführten Messungen vor, und präsentiert die Ergebnisse.
Während der Messungen an dieser ersten Version des Chips ergaben sich viele Verbesserungsansätze und es wurden Fehler aufgedeckt. In Kapitel 7 werden diese beschrieben.
Im Anhang A und B findet sich eine Beschreibung der Anschlüsse des *CIPix*, sowie ein Schaltplan für die externe Beschaltung mit Kapazitäten und Widerständen.

Kapitel 2

Das H1-Experiment

Der in der vorliegenden Arbeit beschriebene integrierte Schaltkreis dient zur Auslese der zentralen inneren Proportionalkammer (central inner proportional chamber, CIP), die Teil des H1-Detektors am Speicherring HERA[1] der Großforschungsanlage DESY[2] in Hamburg ist. Im folgenden sollen nach einer kurzen Beschreibung des Beschleunigersystems die wichtigsten Komponenten des H1-Detektors sowie die Funktionsweise des Triggersystems beschrieben werden. Im weiteren wird auf das H1-Upgrade-2000-Projekt eingegangen, wobei der Schwerpunkt auf der Beschreibung der Neuerungen der CIP liegt.

2.1 Der Speicherring HERA

Im Speicherring HERA, dem weltweit einzigen Beschleunigersystem dieser Art, werden Elektronen[3] und Protonen bei einer Schwerpunktsenergie von $\sqrt{s_{ep}} \approx 300GeV$ zur Kollision gebracht. Die Abb. 2.1 zeigt einen Überblick über HERA. Der Speicherring verläuft unterirdisch und hat einen Umfang von 6,3 km.

In einem System aus Linearbeschleuniger und Vorbeschleuniger (PETRA) werden Positronen und Protonen auf Energien von 12 bzw. 40 GeV gebracht. Anschließend werden diese in den Speicherring (HERA) eingekoppelt und auf die Endenergien von 27,5 GeV für die Positronen bzw. 820 GeV für die Protonen beschleunigt. Die beiden Teilchenstrahlen bestehen aus bis zu 210 einzelnen Teilchenpaketen, sogenannten „bunches“. Jedes dieser Pakete besteht aus 10^{10} bis 10^{11} Teilchen, so daß die Ströme beider Strahlen typisch I_{e^+}=20 mA und I_p=60 mA betragen. Der zeitliche Abstand zwischen zwei Paketen und damit die Zeit mit der diese aufeinandertreffen („bunchcrossing“) beträgt 96 ns. Das entspricht einer Frequenz von 10,4 MHz, der sogenannten „HERA-Clock“. Diese gibt die Datenrate der von HERA durchgeführten Experimente vor und definiert den externen Referenztakt.

Die beiden Strahlen werden getrennt in zwei Hochvakuum-Strahlröhren geführt. An zwei von vier Wechselwirkungszonen werden Positronen und Protonen zur Kollision gebracht. In der Nord- und Südhalle sind die Experimente H1 und ZEUS aufgebaut, die seit Sommer 1992 Daten nehmen. Die physikalischen Fragestellungen, die in diesen beiden Experimenten untersucht werden sollen, sind im wesentlichen:

---

[1] Hadron-Elektron-Ring Anlage

[2] Deutsches Elektronen-Synchrotron

[3] Seit Mitte 1994 werden aus technischen Gründen ausschließlich Positronen verwendet. Mit diesen lassen sich höhere Strahlströme und damit höhere Luminositäten erreichen.

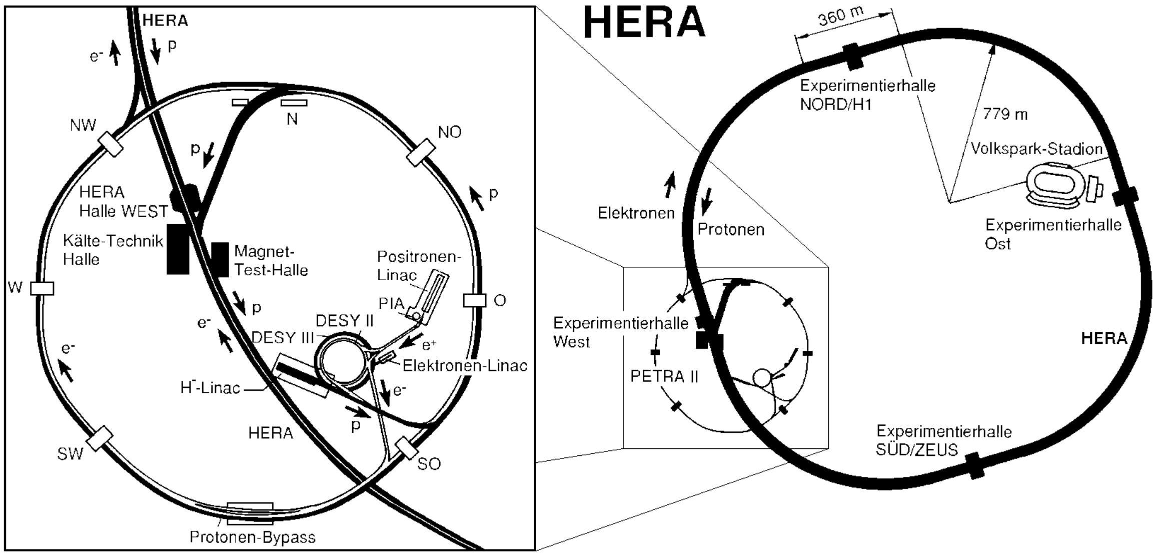

Abbildung 2.1: Der Speicherring HERA am DESY in Hamburg [H1].

- Bestimmung der Protonenstrukturfunktion $F_2(x,Q^2)$. x ist dabei im Quark-Parton-Modell der Impulsbruchteil des Partons am Gesamtimpuls des Protons. Q^2 ist das negative Quadrat des Viererimpulsübertrags.
- Untersuchung der hadronischen Struktur des Photons. Ein Photon kann in ein virtuelles Quark-Antiquark-Paar fluktuieren, das mit anderen Hadronen stark wechselwirken kann.
- Präzisionstests der Theorien zur starken und elektroschwachen Wechselwirkung.
- Suche nach Physik jenseits des Standardmodells (z. B. SUSY[4], GUTs[5]).

Die Abb. 2.2 zeigt im linken Diagramm die von HERA produzierte integrierte Luminosität[6] $\int \mathcal{L}dt$ für die Jahre 1992-1998 und Januar 1999. Im rechten Diagramm ist die für den gleichen Zeitraum von H1 akkumulierte Luminosität dargestellt. Wegen Ausfällen von Detektorkomponenten, Start- und Stop-Phasen und einer endlichen Triggereffizienz werden nur ungefähr zwei Drittel der produzierten Luminosität vom Detektor H1 für die Datennahme genutzt.

An den anderen zwei Wechselwirkungszonen wird jeweils nur einer der beiden Teilchenstrahlen verwendet. HERMES (Halle Ost) untersucht die Streuung polarisierter Elektronen an polarisierten Atomkernen. HERA-B (Halle West), das sich noch im Aufbau befindet, soll die CP-Verletzung im System der neutralen B-Mesonen nachweisen. Dazu werden Protonen

[4]Super Symmetry
[5]Grand Unified Theories
[6]Die Lumnosität definiert die Intensität von Collider-Maschinen. Die Anzahl von Ereignissen einer bestimmten Klasse ist gegeben durch: $N_{class} = A\sigma_{class} \int \mathcal{L}dt$, wobei A die Akzeptanz, σ der Wirkungsquerschnitt und $\int \mathcal{L}dt$ die integrierte Luminosität ist.

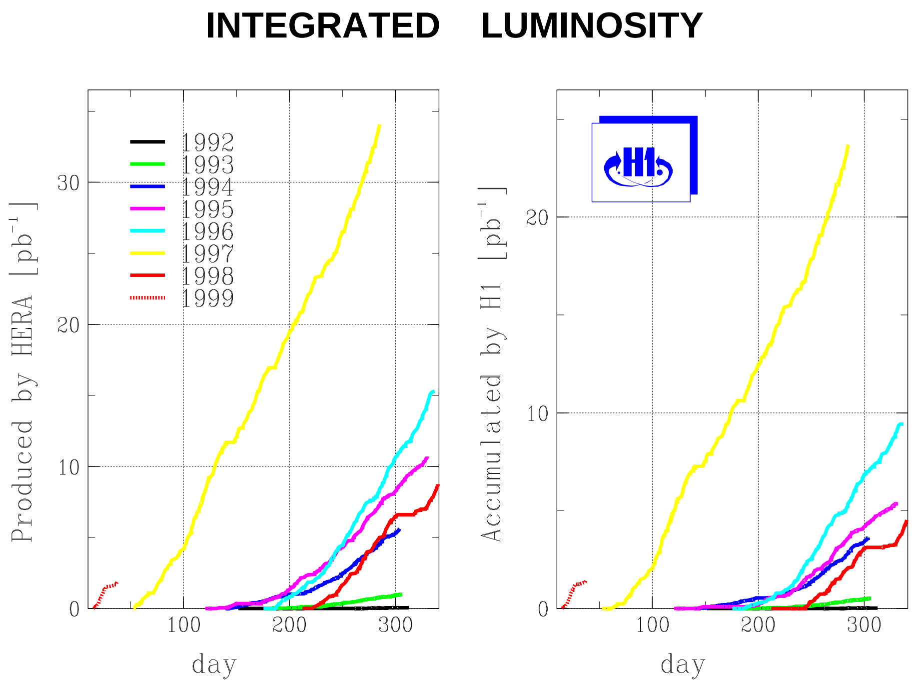

Abbildung 2.2: Von HERA produzierte integrierte Luminosität (links) und von H1 genutzte (rechts). Für das Jahr 1999 sind nur die Daten für Januar eingetragen [H1].

aus dem Halo des Protonenstrahls an Drahttargets gestreut. Eine erste Datennahme soll 1999 erfolgen.

2.2 Der H1-Detektor

Der H1-Detektor (Abb. 2.3) besteht aus mehreren Komponenten, die die Wechselwirkungszone annähernd vollständig umschließen (4π-Detektor). Aufgrund der unterschiedlichen Energie der Protonen und Positronen bewegt sich das Schwerpunktsystem gegenüber dem Laborsystem. Deshalb ist der Detektor stark asymmetrisch gebaut.

Das H1-Koordinatensystem wird durch den HERA-Ring definiert. Der nominelle Wechselwirkungspunkt (Vertex) legt den Ursprung des Koordinatensystems fest (Abb. 2.4). Die z-Achse zeigt in Richtung des einlaufenden Protonenstrahls, die x-Achse zeigt auf den Mittelpunkt des Beschleunigerrings und die y-Achse nach oben. Auf diese Weise ergibt sich ein rechtshändiges Koordinatensystem. Der Azimutwinkel ϕ wird von der x-Achse im Uhrzeigersinn und der Polarwinkel θ von der z-Achse gegen die Protonenflugrichtung gemessen. Den Bereich negativer z-Koordinaten bzw. großer Polarwinkel bezeichnet man als Rückwärtsbereich.

Im folgenden Abschnitt werden die wichtigsten Komponenten des Detektors beschrieben. Eine detaillierte Beschreibung des H1-Detektors findet sich in [H1 97b]. Von der Wechselwirkungszone aus gesehen ist der H1-Detektor wie folgt aufgebaut[7]:

[7]Die angegebenen Ziffern beziehen sich auf Abb. 2.3

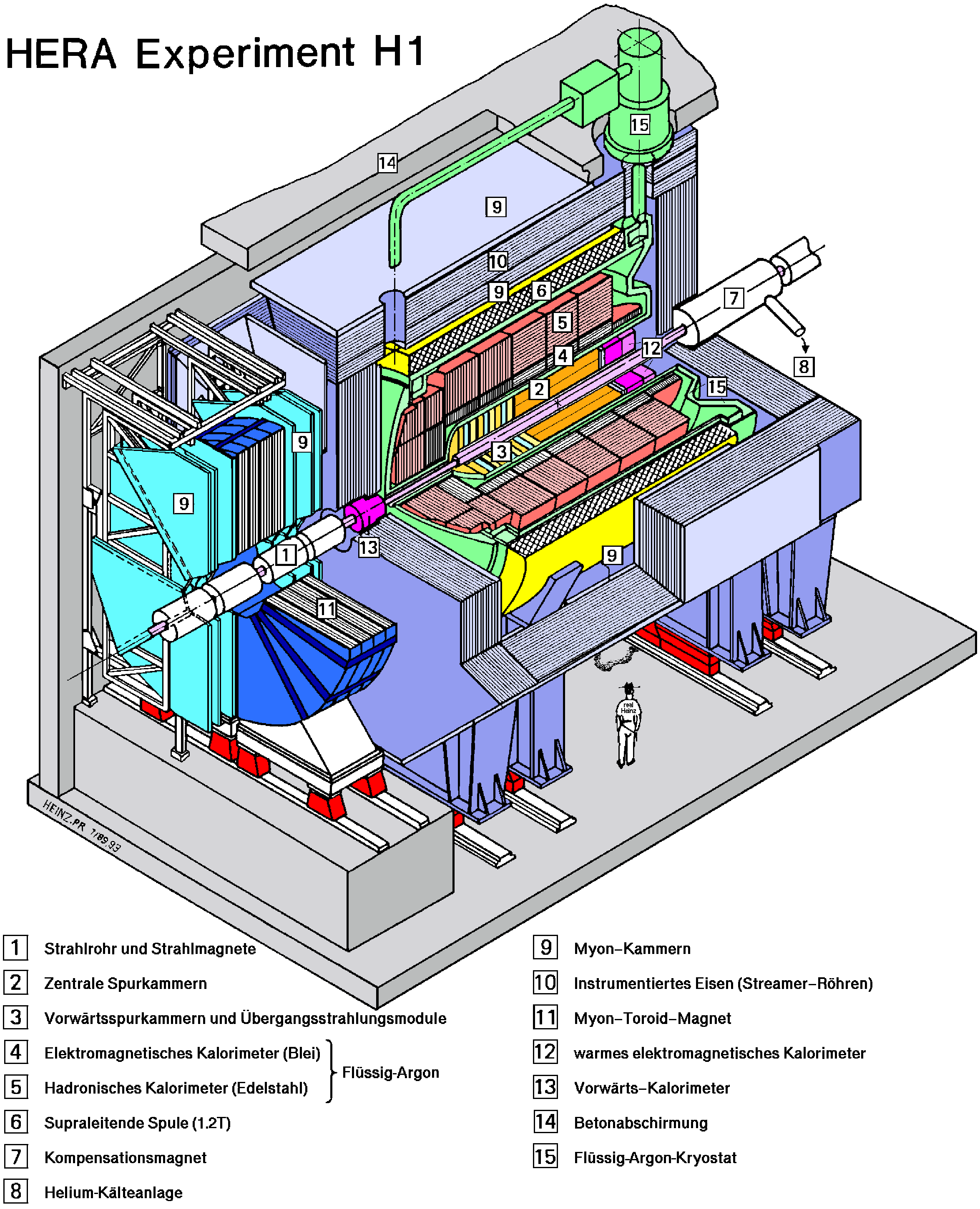

Abbildung 2.3: Schematische Ansicht des H1-Detektors [H1]. An der Stelle des warmen elektromagnetischen Kalorimeters $\boxed{12}$ befindet sich seit 1995 das Blei-Faser-Kalorimeter (Spacal).

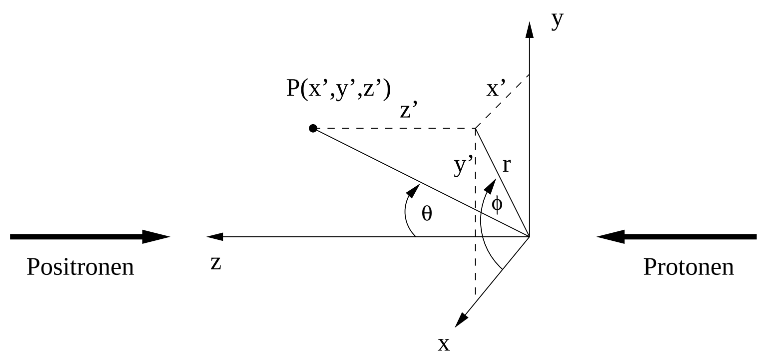

Abbildung 2.4: Das H1-Koordinatensystem

- Silizium-Vertexdetektor
  Der Kollisionspunkt, sowie die Zerfallsorte von Teilchen aus der ep-Reaktion werden durch einen zweilagigen Silizium-Vertexdetektor (central silicon tracker, CST) bestimmt. Dessen intrinsische Ortsauflösung liegt bei 10 μm. Der CST ist eine Erweiterung des ursprünglichen H1-Konzeptes und erst seit 1997 voll einsetzbar.

- Zentrales Spurkammersystem $\boxed{2}$
  An das Strahlrohr und den CST schließt sich ein komplexes System von sechs, konzentrischen Spurkammern an (vergrößerte Seitenansicht in Abb. 2.5, radiale Ansicht in Abb. 2.6). Ihre Aufgabe ist es, die Spur von geladenen Teilchen und ihren Winkel θ gegenüber der Strahlachse zu bestimmen. Von inneren zu äußeren Radien ist das innere Trackingsystem wie folgt aufgebaut:

  - eine doppellagige Vieldrahtproportionalkammer (central inner proportional chamber, CIP)
  - eine z-Driftkammer (central inner z-chamber, CIZ) zur Bestimmung der z-Koordinate von Spuren, mit einer Auflösung von 200μm.
  - zwei Jet-Driftkammern CJC1 und CJC2 (central jet chambers, CJC) zur Rekonstruktion geladener Teilchenspuren in einem Winkelbereich von $30° < \theta < 150°$. Die Winkelauflösung beträgt $\sigma_\theta \approx 1$ mrad bei einer Impulsauflösung von $\frac{\sigma_p}{p^2} \approx 3 \cdot 10^{-3} GeV^{-1}$.
  - eine äußere z-Driftkammer (central outer z-chamber, COZ) und eine weitere doppellagige Vieldrahtproportionalkammer central outer proportional chamber, COP), die sich zwischen den beiden CJCs befinden.

- Vorwärtsspurkammersystem $\boxed{3}$
  Das Vorwärtsspurkammersystem wird aus drei identischen Supermodulen gebildet. Es besteht aus einer planaren und einer radialen Driftkammer sowie einem Übergangsstrahlungsdetektor und einer Vieldrahtproportionalkammer (forward proportional chamber, FPC). Durch dieses Kammersystem wird die Winkelakzeptanz für geladene Teilchen bis θ=5° ausgedehnt.

- Flüssig-Argon-Kalorimeter $\boxed{4}$ und $\boxed{5}$
  Das Flüssig-Argon-Kalorimeter (LAr-Kalorimeter) besteht aus einer elektromagneti-

schen $\boxed{4}$ und einer hadronischen $\boxed{5}$ Komponente. Es erlaubt eine Energiemessung in einem Winkelbereich von $3° < \theta < 155°$. Der elektromagnetische Teil besteht aus Bleiabsorberplatten und ist, abhängig von θ, $20-30$ Strahlungslängen dick. Der hadronische Teil benutzt als Absorber Edelstahlplatten. Zwischen den Platten befindet sich flüssiges Argon als aktives Material, was die Unterbringung des Kalorimeters in einem Kryostaten bedingt.

Das Kalorimeter ist nicht kompensierend, das heißt, daß die Antwort auf Elektronen und Hadronen gleicher Energie unterschiedlich ausfällt. Durch die Lage der Schauer lassen sich beide Teilchensorten unterscheiden. Elektronen deponieren ihre Energie in den ersten Lagen des Kalorimeters, Hadronen dagegen passieren die ersten Schichten und deponieren ihre Energie in den hinteren Kalorimeterlagen. Die Energieauflösung liegt bei $\sigma(E)/E = 12\%/\sqrt{E/GeV} \oplus 1\%$ für Elektronen und $\sigma(E)/E = 50\%/\sqrt{E/GeV} \oplus 2\%$ für Hadronen. Die absolute Energieskala hat eine Unsicherheit von 3% für Elektronen und 4% für Hadronen.

- Vorwärtskalorimeter $\boxed{13}$
  Im Vorwärtsbereich wird der H1-Detektor durch ein Kupfer-Silizium Kalorimeter ergänzt. Dieser sogenannte Plug-Detektor erschließt den Akzeptanzbereich von $0{,}7° < \theta < 3,2°$.

- Supraleitende Spule $\boxed{6}$
  Die supraleitende Spule erzeugt im Bereich der Spurkammern ein nahezu homogenes, axiales (parallel zur Strahlachse ausgerichtetes) Magnetfeld von 1,16 T. Dieses ermöglicht die Impulsbestimmung geladener Teilchen in den Driftkammern durch Messen der Spurkrümmung. Die Rückführung des magnetischen Flusses geschieht durch ein Eisenjoch, in dem zusätzlich Streamerkammern für den Nachweis von Myonen eingebettet sind ($\boxed{9}$, $\boxed{10}$). Damit das Magnetfeld der supraleitenden Spule nicht die Teilchenbahn im Strahlrohr beeinflußt, ist am Ende des Detektors um das Strahlrohr ein supraleitender Kompensationsmagnet angebracht $\boxed{7}$.

- Spaghetti-Kalorimeter $\boxed{12}$
  Im Rückwärtsbereich des Detektors steht seit 1995 ein Blei-Faser-Kalorimeter (Spaghetti calorimeter, SpaCal). Es besteht aus in Bleiplatten eingebetteten, szintillierenden Fasern. Zusammen mit einer rückwärtigen Driftkammer (backward drift chamber, BDC) können damit Elektronen und Hadronen bei großen Streuwinkeln von $\theta = 152° - 177°$ nachgewiesen werden.

- Luminositätssystem
  Die von HERA gelieferte Luminosität wird von H1 über den Bethe-Heitler Prozeß ep$\rightarrow$epγ (Bremsstrahlung) gemessen, dessen Wirkungsquerschnitt theoretisch sehr gut beschrieben ist. Die Luminosität wird über die Beziehung

$$\mathcal{L} = \frac{R_{tot} - (I_{tot}/I_0)R_0}{\sigma_{vis}}$$

bestimmt, wobei R_{tot} die gesamte, beobachtete Rate der Bremsstrahlungsereignisse ist, R_0 die Rate Elektron-Pilotbunches[8], I_{tot} und I_0 die entsprechenden Ströme des Elektronenstrahls. σ_{vis} ist der mit der Akzeptanzfunktion der Detektoren und der Trigger-Effizienz gefaltete Wirkungsquerschnitt. Zwei kleine Kalorimeter, 105 m und 33 m

[8] Elektronenpakete ohne ein Protonenpaket als Kollisionspartner.

von der Wechselwirkungszone installiert, weisen diese Photonen (Photonentagger) und Elektronen (Elektronentagger) nach.

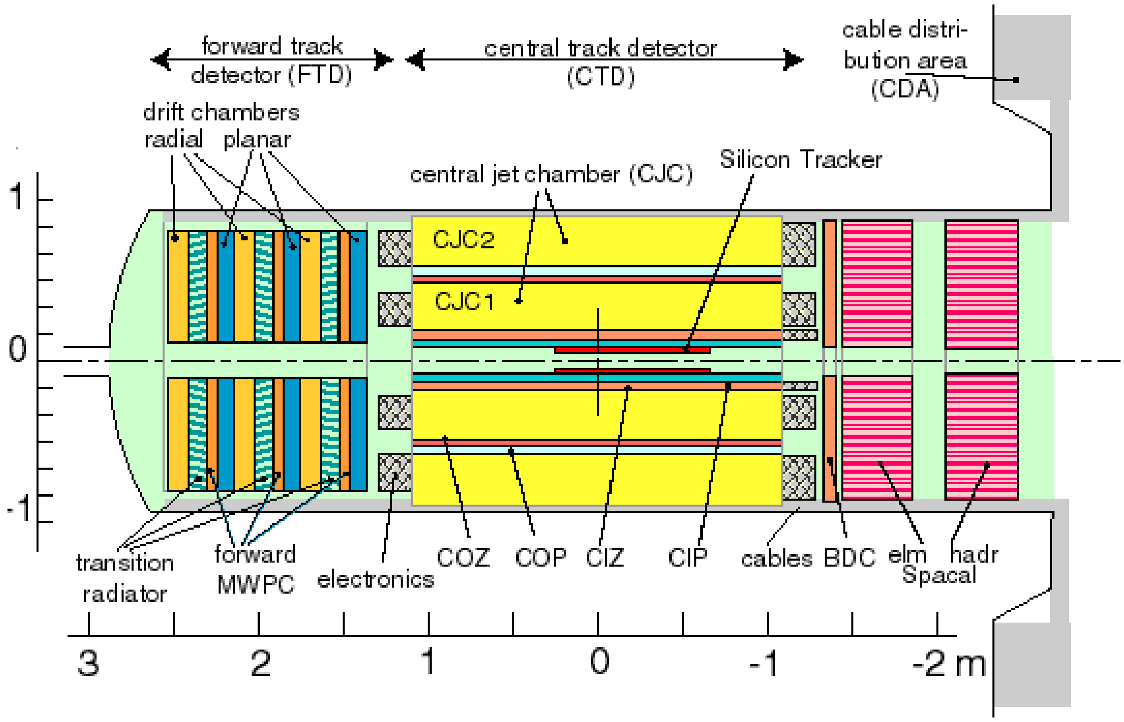

Abbildung 2.5: Seitliche Ansicht des inneren Spurkammersystems [H1].

2.3 Das H1-Triggersystem

Der Trigger[9] hat die Aufgabe, die zahlenmäßig dominierenden Untergrundereignisse von physikalisch interessanten ep-Streuereignissen zu trennen. Zu den Untergrundquellen zählen Synchrotron-Strahlung der Positronen, Proton-Gas-Wechselwirkungen in der Beampipe[10] und Streuprotonen, die durch Zusammenstöße mit dem Strahlrohr und anderem Material um den Detektor Teilchenschauer erzeugen. Myonen aus dem Strahlhalo und kosmische Myonen spielen ebenfalls eine Rolle.

Die wesentlichen Aufgaben des Triggers sind es, die Datenrate auf ein für die Speicherung und Nachverarbeitung mögliches Maß zu reduzieren und die Totzeit des Detektors zu minimieren. Da die Auslese des Detektors Zeit in Anspruch nimmt und währenddessen die Auslesekette keine neue Daten annimmt, soll der Detektor im Idealfall nur für physikalisch interessante Ereignisse ausgelesen werden. Das Triggersystem besteht bei H1 aus drei unabhängigen Stufen (L1, L2, und L4), die synchron zur Datennahme („online“) arbeiten. In

[9] Auslöser
[10] Strahlröhre

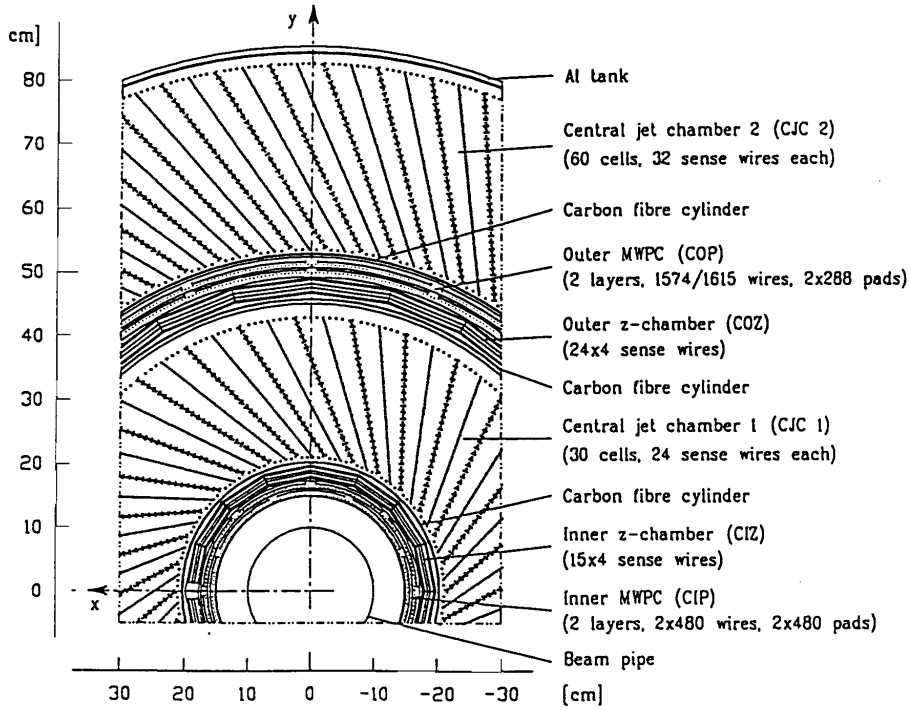

Abbildung 2.6: Radiale Ansicht des inneren Spurkammersystems [H1].

Abb. 2.7 ist die Hierarchie des H1 Triggersystems schematisch dargestellt. Es sind die Ein- und Ausgaberaten sowie die Entscheidungszeiten jeder Stufe gezeigt.

Die erste Triggerstufe (L1) wird von schnellen, in Hardware realisierten *Triggerelementen* gebildet. Ein Triggerelement nutzt unmittelbar nach einer Wechselwirkung verfügbare Informationen aus verschiedenen Detektorkomponenten für erste Aussagen über ein Ereignis. Diese können z. B. eine bestimmte Spurtopologie in einer Driftkammer oder das Überschreiten einer voreingestellten Schwelle für Energie-Deposition in einem Kalorimeter repräsentieren. Die Triggerelemente werden logisch miteinander zu 128 *Subtriggern* (Triggerelement-Koinzidenzen) verknüpft (siehe Abb. 2.3).

Diese Triggerkoinzidenzen werden elektronisch über sogenannte RAM[11]-Lookup-Tables gebildet. Die logischen Signale der Triggerelemente eines Subsystems werden dabei auf den Adressbus eines schnellen RAM-Bausteins gegeben, dessen 8-Bit-Speicherzellen die frei programmierbare Kombination logischer Ausdrücke der Triggerelemente ermöglichen[12]. Jeder Ausgang eines RAM-Bausteins kann mit Ausgängen anderer RAM-Bausteine durch UND-Gatter logisch verknüpft werden. Zusammen bilden diese eine von insgesamt 128 möglichen Subtrigger-Bedingungen. Eine erfüllte Subtrigger-Bedingung führt jedoch nicht automatisch zu der Entscheidung, daß ein Ereignis weiter behandelt werden soll („L1 keep“). Man bedient sich eines Skalierungsverfahrens (prescaling), mit welchem nur jeweils eines aus einer voreingestellten Anzahl von Ereignissen (prescale factor), die die Subtrigger-Bedingung ausgelöst haben, tatsächlich ausgelesen wird.

[11]Random Access Memory

[12]In jeder Triggerkoinzidenz muß ein sogenanntes t_0-Triggerelement vorhanden sein, das das zu dem anliegenden Ereignis gehörende Bunchcrossing markiert.

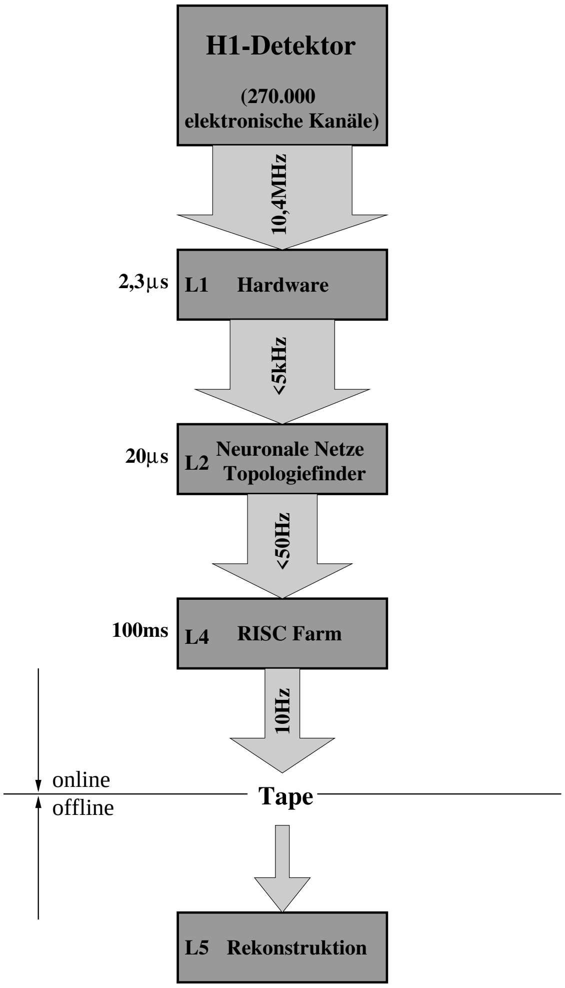

Abbildung 2.7: Schema der Hierarchie des H1-Triggersystems. Es sind die Ein- und Ausgaberaten, sowie die Entscheidungszeiten jeder Stufe gezeigt [Nic96].

Von den verschiedenen Triggerelementen des L1-Triggers soll an dieser Stelle der z-Vertex-Trigger näher beschrieben werden. Das Prinzip der z-Vertex-Bestimmung wird in Abb. 2.9 veranschaulicht. Aus den aktivierten Pads der CIP, COP und der fMWPC werden alle möglichen Spuren rekonstruiert und in Richtung der Strahlachse extrapoliert. Alle Spuren, die ihren Schnittpunkt mit der Strahlachse im Bereich $z = 43,9$ cm und $z = -43,9$ cm haben, werden in ein Histogramm eingetragen. Ein signifikanter Peak in diesem Histogramm bestimmt die z-Position des Vertex. Der z-Vertex Trigger liefert mehrere Triggerelemente. Das schwächste Element (*T0*) wird bei einem nicht-leeren Histogramm generiert.

Je nach Strahlbedingung (Strahlströme und Untergrund) wird ein Run[13] einer Runphase zugeordnet; diese reichen von 1 („hohe Luminosiät, hoher Untergrund“) bis 4 („niedrige Lunimosität, niedriger Untergrund“)[14]. Abhängig von der Runphase werden die Subtrigger

[13]Periode der Datennahme.

[14]Eine Ausnahme bilden *Cosmic Runs* und *Test Runs*, die u. a. zur Kalibrierung von Detektorkomponenten dienen

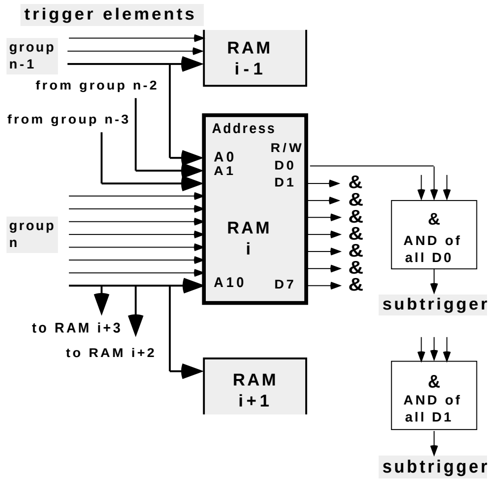

Abbildung 2.8: Generierung der *Subtrigger* durch RAM-Lookup Tables. Triggerelemente eines Subsystems werden zusammen mit den t_0-Triggerelementen auf die Adressleitungen eines RAM-Bausteins gelegt. Die mittels konventioneller UND-Gatter erzeugte Koinzidenz der RAM-Ausgänge aller Subsysteme erzeugt die *Subtrigger*-Bedingung [Sef92].

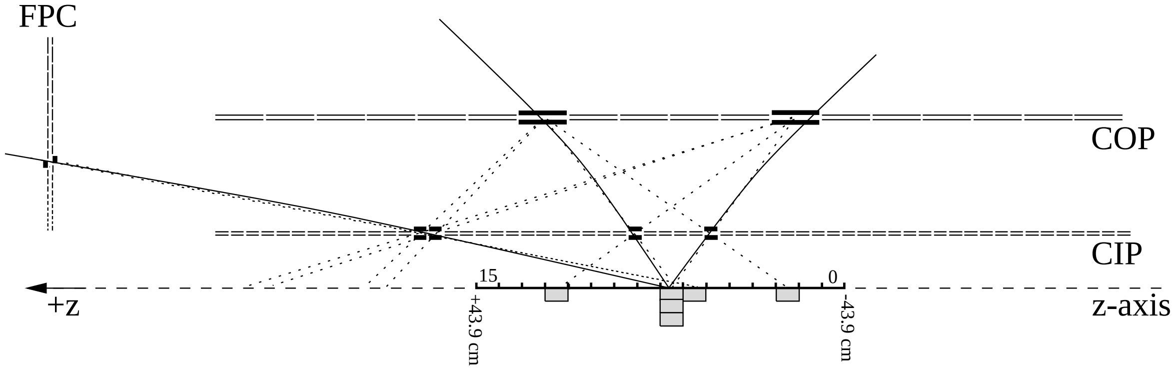

Abbildung 2.9: Funktionsweise des z-Vertex-Triggers. Koinzidenzen von jeweils vier Proportionalkammer-Pads von CIP, COP und FPC definieren einen Spurkandidaten [HB96].

festgelegt, die ein Ereignis triggern können.

Die Triggerstufe L1 arbeitet totzeitfrei bei einer Frequenz von 10,4 MHz. Die Entscheidung ist um 2,5 μs verzögert, da die volle Triggerinformation über ein Ereignis erst nach dieser Zeit vorliegt. Diese Zeitspanne wird bestimmt durch die längste Driftzeit in den Driftkammern ($\approx 1\mu$s) und der Integrationszeit der Vorverstärker des LAr-Kalorimeters ($\approx 1,5\mu$s). Die Signale der Detektorkomponenten müssen also bis zum Eintreffen einer L1-Triggerentscheidung elektronisch zwischengespeichert werden. Dies erfolgt mittels zyklisch überschriebener Ringspeicher, sogenannter „Pipelines“.

Eine positive Entscheidung der Triggerstufe L1 („L1 keep“) stoppt die Pipeline und führt eine Entscheidung der nachfolgenden Triggerstufe L2 herbei. Diese benutzt topologische Informationen und neuronale Netzwerke zur Entscheidungsfindung. Mit dem Stoppen der Pipeline beginnt die Totzeit des Detektors. Eine Entscheidung von L2 liegt im Mittel nach 20μs vor. Erst nach einer positiven Entscheidung („L2 keep“) wird die zeitaufwendige Auslese der Driftkammern und die analoge Auslese des Kalorimeters initiiert. Andernfalls wird der Ringspeicher zurückgesetzt.

Die folgende Triggerstufe L4 verifiziert die L2-Entscheidung durch eine schnelle, reduzierte Ereignisrekonstruktion. L4 besteht aus einer massiv parallelen Rechnerfarm. Die Stufe arbeitet asynchron zu den vorangegangenen Stufen und trägt daher nicht weiter zur Totzeit bei, solange eine bestimmte „L2 keep“-Rate nicht überschritten wird. Auf L4 wird zum ersten mal die gesamte zu einem Ereignis gehörende Detektorinformation zusammengeführt. Sämtliche L1-Subtrigger werden anhand der vollen Information überprüft und gegebenenfalls verworfen. Zu Kontrollzwecken werden 1% der von L4 verworfenen Ereignisse aufgezeichnet.

Die von L4 akzeptierten Ereignisse, typischerweise 10 pro Sekunde, werden der Ereignisrekonstruktion (L5) zugeführt. Die Stufe L5 arbeitet unabhängig von der Datennahme („offline“). Hier werden die Ereignisse nach Physikklassen gegliedert.

2.4 Das H1-Upgrade-2000-Projekt

Für das Jahr 2000 ist geplant, ein Upgrade bei HERA durchzuführen, um die Luminosität $\mathcal{L}$ um einen Faktor 5 gegenüber den Design-Parametern zu erhöhen. Das würde eine Vergrößerung auf $\mathcal{L} = 1,5 \cdot 10^{31}\mathrm{cm}^2\mathrm{s}^{-1}$ bedeuten. Die Hauptmotivation für dieses Projekt ist das Bestreben, die zur Zeit bestehenden Experimente sensibler auf Physik außerhalb des Standardmodels zu machen und die Sensitivität auf seltene Ereignisse mit großem Q^2 oder großem Transversalimpuls zu vergrößern. Die Erhöhung der Luminosität wird durch einen höheren Protonenstrom und eine bessere Fokussierung der beiden Strahlen erreicht. Eine ausführliche Beschreibung des jetzigen Stands, der Motivation und der Maßnahmen für das Upgrade befindet sich in [H1 97a].

Die bessere Fokussierung der Strahlen bedingt es, daß die zur Strahltrennung bzw. -zusammenführung notwendigen Quadropolmagnete näher an den Kreuzungspunkt herangebracht werden. Für den H1-Detektor bedeutet das, daß sich die Magnete dann im Detektor selbst befinden. Ein Magnet ragt in das Vorwärtsspurkammer-System (FTD), der andere in das elektromagnetische und hadronische Kalorimeter (SpaCal) im Rückwärtsbereich. Eine weitere Konsequenz der neuen Quadropolmagnete ist, daß sich die e^+- und p-Strahlen unter einem größeren Winkel im Wechselwirkungspunkt schneiden. Um zukünftig nahe genug an das Strahlrohr heranzukommen, dennoch aber nicht die seitliche Strahltrennung zu behindern, wird die Beampipe durch ein ovales Rohr ersetzt. Weiterhin muß die Abschirmung der Synchrotronstrahlung, das Argon-Kühlsystem sowie das Luminositäts-System komplett überarbeitet werden. Diese Veränderungen ziehen Modifikationen einiger bestehender Detektorkomponenten nach sich. Zusätzlich werden aber auch primär nicht betroffene Komponenten im Rahmen der Upgrade-Maßnahmen verbessert.

Nachfolgend sollen die geplanten Veränderungen für den Bereich des Trackingsystems beschrieben werden:

- Central silicon tracker (CST), backward silicon tracker (BST)
  Der CST muß der neuen ovalen Strahlröhre angepaßt werden, wobei die vorhandenen Detektormodule wiederverwendet werden können und nur ihre Anordnung verändert wird. Die verbesserte z-Information der CIP wird in späteren Analysen der Daten dazu dienen, die dreifach Ambiguität[15] des CST aufzulösen.

  Der BST wird mit neuen ϕ-Detektoren ausgerüstet und ebenfalls an die veränderte Geometrie der Strahlröhre angepaßt.

[15]= Mehrdeutigkeit aufgrund der Signalauslese des CST: Der CST ist in z-Richtung dreifach segmentiert. Signale aus diesen drei Bereichen werden über eine Leitung ausgelesen, wodurch eine eindeutige Zuordnung eines Signals zu einem bestimmten z-Segment nicht möglich ist.

- Central jet chamber (CJC)
  Die beiden Jetkammern (CJC1 und CJC2) erhalten eine schnellere Ausleseelektronik, die es gestattet, Informationen über die invariante Masse bereits nach 20 μs für den L2-Trigger zur Verfügung zu stellen. Bisher konnte die invariante Masse erst von der Triggerstufe L4 berechnet werden.

- Central outer z-chamber (COZ), central inner z-chamber (CIZ)
  Die COZ wird unverändert beibehalten. Aufgrund von Alterungsprozessen der CIZ, bedingt durch den geringen Abstand zum Strahl, wird diese Kammer ausgebaut. Ihre Aufgaben werden vom CST übernommen.

- Central inner proportional chamber (CIP)
  Die Anzahl der Kammerlagen der CIP wird von 2 auf 5 Lagen erhöht und die Zahl der Pads in z-Richtung von 60 auf 120 verdoppelt (siehe Abschnitt 2.5 für eine ausführlichere Beschreibung). Durch diese Veränderungen erreicht man vor allem eine Verbesserung der z-Vertex-Information. Die Akzeptanz für sehr flache Spuren in Vorwärts- und Rückwärtsrichtung wird stark verbessert und macht den Ausbau einer Komponente des FTD möglich (siehe unten). Protonstrahl-Untergrundereignisse, die einen Vertex außerhalb des H1-Detektors haben, werden besser erkannt. Außerdem wird man sensitiver auf Ereignisse mit kleinem Q^2, bei denen das Elektron unter einem sehr flachen Winkel rückwärts gestreut wird. Die neue CIP soll in der Lage sein, die dreifache Ambiguität des CST aufzulösen, was zuvor von der CIZ wahrgenommen wurde.

- Forward track detector (FTD)
  Die MWPC, eine der Komponenten der drei Supermodule des FTD, werden nicht mehr benötigt. An ihre Stelle wird eine planare Driftkammer eingebaut, die zur Feststellung der (r,z)-Komponente eines durchfliegenden Teilchens dient. Die zuvor von den MWPC gelieferte Triggerinformation bei kleinen θ wird durch die neue CIP bereitgestellt.

2.5 Die alte und neue CIP

Die alte CIP

Die CIP ist eine dünne (0,14% Strahlungslänge) zwei Meter lange, zylindrische Vieldrahtproportionalkammer mit Kathodenauslese. Die bisher im H1-Detektor eingebaute Kammer besteht aus zwei übereinanderliegenden Lagen mit einer 8-fach Segmentierung in ϕ. Die beiden Lagen sind gegeneinander um $22,5°$ (entspricht einem halben Segment) gedreht, wodurch effektiv eine 16-fach Segmentierung erreicht wird. Entlang der Zylinderachse befinden sich 60 Pads mit einer Länge von 3,5 cm und einer Breite von etwa 13 cm. Die Kammer ist in einer Art „Sandwich-Bauweise“ aus dünnen Metallschichten und Rohazell aufgebaut.

Die beiden Lagen der CIP sind identisch aufgebaut. Die radialsymmetrische Konstruktion soll im folgenden kurz beschrieben werden: Eine Kammerlage besteht aus einem 6 mm dicken Gasvolumen, in dessen Mitte vergoldete Wolframdrähte als Anoden gespannt sind. Der Gasraum wird von innen durch eine Aluminiumfolie als Kathode und von außen von einer hochohmigen Kohleschicht begrenzt. An die Kohleschicht schließt sich eine 20 μm dicke Kaptonfolie an, auf der die Kathodenpads aufgebracht sind. Die Abmessungen der Pads betragen 3,5 cm in z-Richtung und etwa 13 cm in ϕ. Die Pads werden durch die darüberliegende 2mm dicke Rohazellschicht mit den Auslesedrähten aus Aluminium kontaktiert. Den Abschluß der

Kammer bilden eine 20 μm dicke Kaptonfolie, eine 0,5 μm dicke Aluminium-Abschirmung, eine 1 mm dicke Rohazellschicht und schließlich eine Aluminiumschicht. Letztere dient wiederum als Aluminium-Kathode der nächsten Kammerlage. Eine genaue Beschreibung der CIP findet sich in [KM92] und [MK98].

Die Hauptaufgabe der CIP ist die Lieferung der z-Vertexinformation für den L1-Trigger (zusammen mit der COP oder der ersten fMWPC[16]). Abb. 2.10 zeigt einen Ausschnitt der CIP und der COP, sowie die rekonstruierte Spur zwischen den aktivierten Pads beider Kammern. Die bisher verwendete CIP bietet einen erprobten und stabilen Betrieb mit Triggerinformationen für fast alle Ereignisklassen.

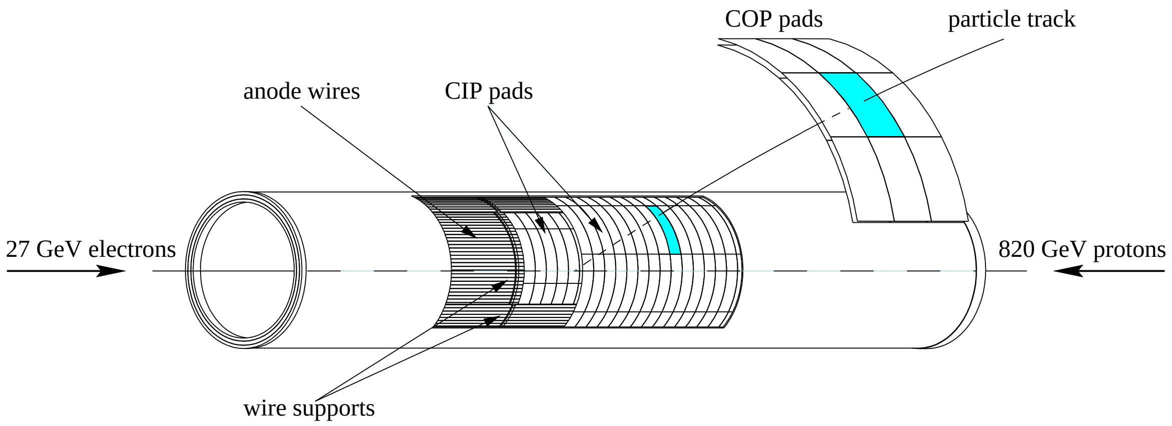

Abbildung 2.10: Rekonstruktion einer Spur durch aktivierte Pads in der CIP und COP [HB96].

Die neue CIP

Geometrie

Bei der neuen Kammer wird die Anzahl der Lagen von 2 auf 5 erhöht und die Zahl der Pads in z-Richtung von 60 auf 120 verdoppelt. Die Lagen sind einheitlich 16-fach in ϕ segmentiert.

Der prinzipielle Aufbau der einzelnen Lagen gegenüber der alten Kammerversion bleibt unverändert: Der wesentliche Unterschied besteht in der Anpassung der Größe der Kupferpads an die erhöhte Padanzahl. Außerdem werden die Auslesedrähte der alten Kammer in der neuen Version durch Microstriplines ersetzt. Diese verlaufen durch eine Rohazellschicht getrennt auf einer Kaptonfolie über den Kupferpads. In Abb. 2.11 wird die Schichtfolge der neuen Kammerlagen gezeigt.

Die auf den Kupferpads induzierten elektrischen Signale werden über die Microstrips an die $-$z-Seite der Kammer geführt. Hier sitzt, neben anderer Elektronik, der Auslese-ASIC[17] *CIPix* (siehe Abb. 2.11), der die Nachverarbeitung der Kammersignale übernimmt. Dazu zählen im wesentlichen: Pulsverstärkung, Pulsformung, Digitalisierung und Multiplexing. Im Rahmen dieser Diplomarbeit wurde dieser ASIC in Zusammenarbeit mit Sven Löchner [SL98] entwickelt und getestet.

[16]=forward MWPC.
[17]ASIC = Application Specific Integrated Circuit.

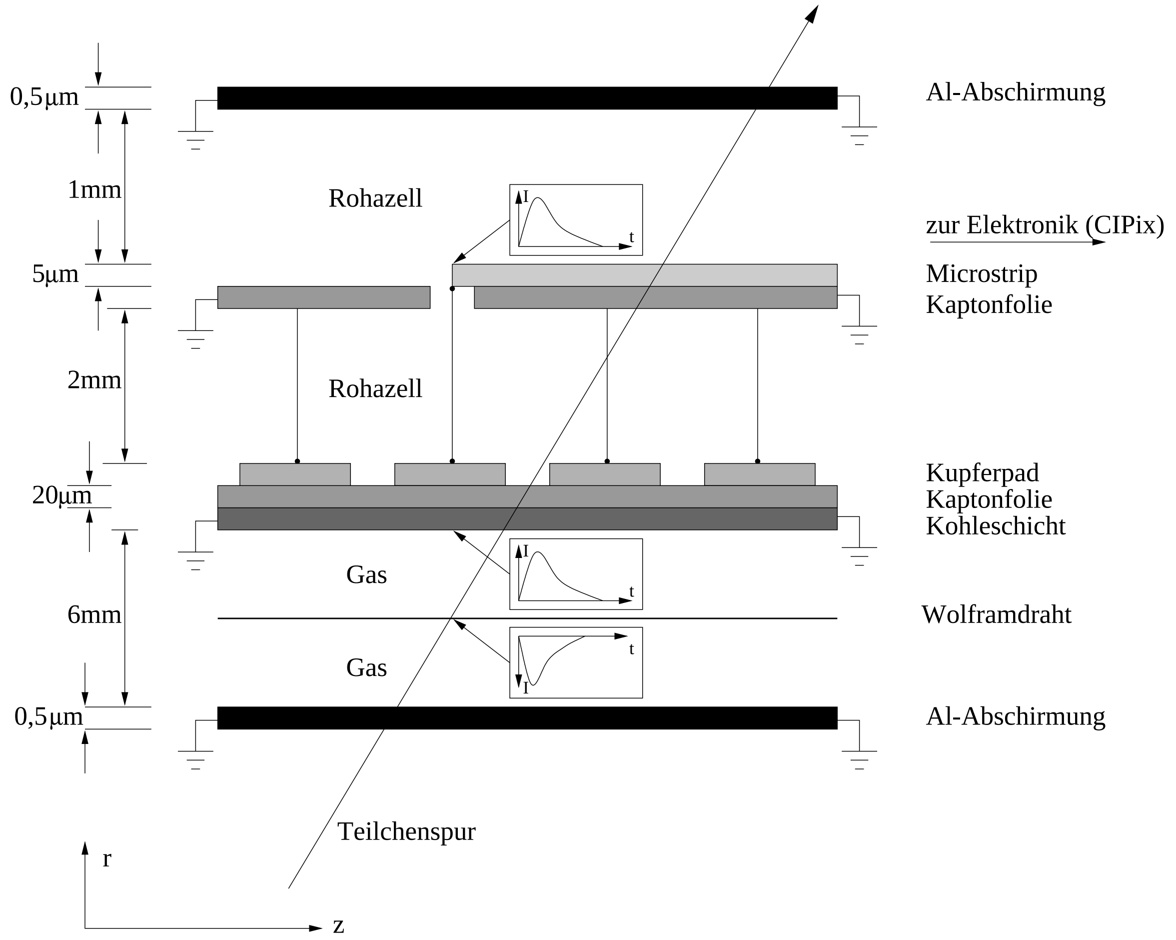

Abbildung 2.11: Schematischer Aufbau einer Kammerlage der neuen CIP [MK98].

z-Vertex Trigger

Die neue Kammer liefert eine verbesserte z-Vertex Information. Protonenstrahl-induzierte Untergrundereignisse können besser unterdrückt werden und die Akzeptanz für sehr flache Spuren in Vorwärts- und Rückwärtsrichtung wird stark erhöht. Damit verbunden ist eine größere Sensitivität für Ereignisse mit kleinem Q^2, bei denen das Elektron unter einem sehr flachen Winkel rückwärts gestreut wird.

Studien zur Vertex-Rekonstruktion und zur Unterdrückung von Untergrundereignissen durch die neue 5-lagige CIP wurden von Katharina Müller an der Universität Heidelberg durchgeführt und sollen im folgenden kurz vorgestellt werden [H198].

Zunächst wurde versucht, die Padverteilung der *alten* CIP durch Simulation der Ladungsverteilung in der Kammer zu reproduzieren. Man stellt fest, daß ein Großteil der aktiven Pads mit Spuren im H1-Trackingsystem korreliert. In Abb. 2.12 (a) ist die Anzahl der aktiven Pads gegen die z-Distanz zum Track für Daten[18] und die Simulation aufgetragen. Abb. 2.12 (b)

[18]Für diese Untersuchungen wurden Daten aus einem speziellen Run im Jahr 1997 verwendet, bei dem nur Protonen in der HERA Maschine waren. Die Ereignisse wurden für eine Spur aus der Wechselwirkungsregion mit der schwächsten Bedingung des alten z-Vertex-Triggers (zvtx-T0) getriggert.

zeigt die Anzahl der Pads innerhalb von 5 cm um die Spur.

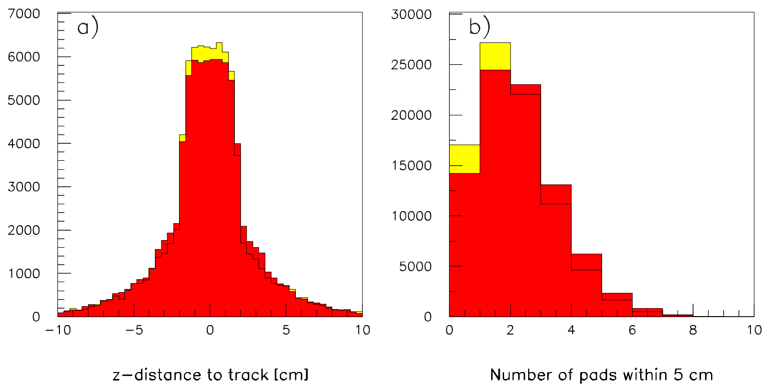

Abbildung 2.12: z-Abstand aktivierter Pads zur Spur (a) und Anzahl der Pads innerhalb von 5 cm (b) für Daten (hell) und Simulation (dunkel) [H198].

Man erkennt jedoch, daß ein Teil der aktiven Pads nicht mit gemessenen Spuren korreliert. Dies gilt vor allem für den Vorwärts- und Rückwärtsbereich. Die zusätzlichen Pads werden durch geladene Teilchen aktiviert, die infolge von Ineffizienz oder Akzeptanzverlust von den Spurkammern nicht detektiert oder rekonstruiert werden. Für Untergrunddaten verstärkt sich dieser Effekt noch, da viele sehr flache Spuren vorhanden sind. Die Anzahl dieser zusätzlichen Spuren hängt von der Topologie des Ereignisses, insbesondere von der Gesamtenergie und der Gesamtanzahl der Treffer in den Spurkammern ab (Abb. 2.13).

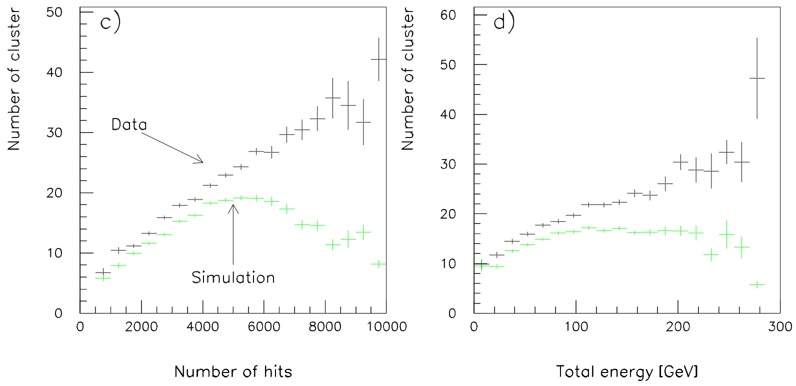

Abbildung 2.13: Zahl der Cluster in einer Lage der CIP gegen die Gesamtzahl an Hits im Spurkammersystem (c) und die Gesamtenergie (d) für Daten und Simulation [H198].

Für die Simulationen der *neuen* CIP wurde diese Abhängigkeit, sowie die z-Verteilung der aktivierten Pads berücksichtigt. Die 5-lagige CIP wurde für verschiedene Geometrien hinsichtlich der Unterdrückung von Untergrundereignissen simuliert.

Die Rekonstruktion der z-Vertex Position der Wechselwirkung liefert ein starkes Kriterium für die Seperation von Untergrundereignissen. Typische protonen-induzierte Untergrundereignisse haben eine Vertex-Position außerhalb des H1-Detektors bei $z = -150 \ldots -200\ cm$, wohingegen die Vertex-Position von Physik-Ereignissen bei $z = 0 \pm 30\ cm$ liegt.

Die Idee eines neuen z-Vertex Triggers besteht nun darin, einen *Spurfindungs-Algorithmus* zu verwenden. Anders als bei der Histogrammbildung und Spitzenwertfindung des alten Triggers werden hierbei die Anzahl der von *upstream* kommenden Spuren (z < -70 cm) mit der Anzahl der aus der Vertexregion kommenden Spuren (z=$\pm$ 50 cm) verglichen (Abb. 2.14). Mit einem einfachen Schnitt kann zwischen den beiden Ereignisklassen unterschieden werden.

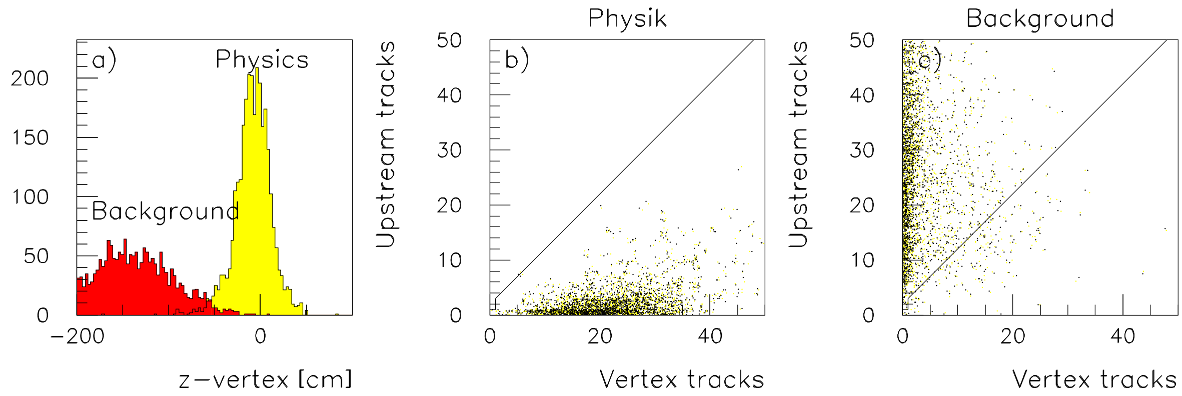

Abbildung 2.14: Verteilung von *upstream*-Spuren (z $<$ -70cm) gegen Vertex-Spuren (z=$\pm$ 50 cm) für Physik- (b) und Untergrundereignisse (b) [H198].

Der Verlust für Physikereignisse beträgt weniger als 1%, Untergrundereignisse können mit einer Effizienz von mehr als 95% verworfen werden. Die Auflösung für die z-Vertex Position beträgt etwa 6 cm. Mit dem neuen Triggersystem kann der Protonenstrahl-Untergrund für das schwächste z-Vertex Triggerelement (zvtx-T0) um einen Faktor 20 reduziert werden. Mit Daten der neuen CIP und des CST alleine erreicht man in der *Offline*-Rekonstruktion eine typische Vertex-Auflösung von 0,46 cm.

Kapitel 3

Der CIP-Auslese-ASIC

Im nachfolgenden Kapitel soll die Architektur des CIPix1.0 beschrieben werden. Nach einer kurzen Spezifizierung des künftigen Einsatzgebietes des Chips werden die einzelnen Komponenten dem Signalfluß folgend dargestellt. Die digitalen Baugruppen werden, mit Ausnahme des Multiplexers, separat in Kapitel 5 behandelt.

Betrachtet man die alte Version der CIP-Kammer, so kommt diese bei 960 auszulesenden Pads noch mit *konventioneller*, d. h. nicht integrierter Ausleseelektronik aus. Die neue Kammer liefert jedoch einen Faktor 10 mehr Signale (9600). Für diese Größenordnung ist die Entwicklung eines *integrierten* Verstärkerschaltkreises sinnvoll. Der im Rahmen des CIP-Upgrade Projektes für die Entwicklung des Auslesechips vorgesehene Zeitraum (ungefähr ein Jahr, [Prop98]) schloß eine völlige Neuentwicklung eines ASICs aus. Stattdessen sollte versucht werden, soweit wie möglich vorhandene Komponenten ähnlicher Systeme zu verwenden. Für das Experiment HERA-B am DESY wird im *ASIC-Labor*[1] Heidelberg ebenfalls ein Auslesechip entwickelt, der schon mehrere Iterationen durchlaufen hat und sich in der Massenproduktion befindet. Der Helix128-2.2 ist ein analoger Auslesechip für den Vertexdetektor und die MSGCs[2] des inneren Spurkammersystems des HERA-B Detektors. Der Helix128-2.2 integriert 128 ladungsempfindliche rauscharme Vorverstärker und Pulsformer, sowie einen analogen Zwischenspeicher aus Kapazitäten. Dem Vorverstärker und Pulsformer ist ein Komparator nachgeschaltet, der beim Überschreiten einer einstellbaren Schwelle ein Triggersignal liefert. Dieses wird bei HERA-B für den *First Level Trigger* (FLT) verwendet. Die verstärkten und geformten analogen Signale werden über einen Multiplexer zeitlich hintereinander ausgegeben [ES97].

Sven Löchner untersuchte im Rahmen seiner Diplomarbeit [SL98], inwieweit Komponenten des Helix128 für den CIPix zu verwenden sind und modifizierte diese.

3.1 Überblick

Der CIPix wurde in einem 0,8μm CMOS-Prozeß entworfen und im Oktober 1998 von der Firma *Austria Micro Systeme* (AMS) produziert. Der Chip hat eine Fläche von 4040 μm$\times$6615 μm und integriert die folgenden Schaltkreise (siehe Abb. 3.1):

[1]Das ASIC-Labor Heidelberg wurde 1994 gegründet und ist eine Gemeinschaftseinrichtung des Physikalischen Instituts, des Max-Planck-Instituts für Kernphysik (MPIK), des Instituts für Hochenergiephysik (IHEP) und des Instituts für Angewandte Physik.

[2]Micro Strip Gaseous Chambers

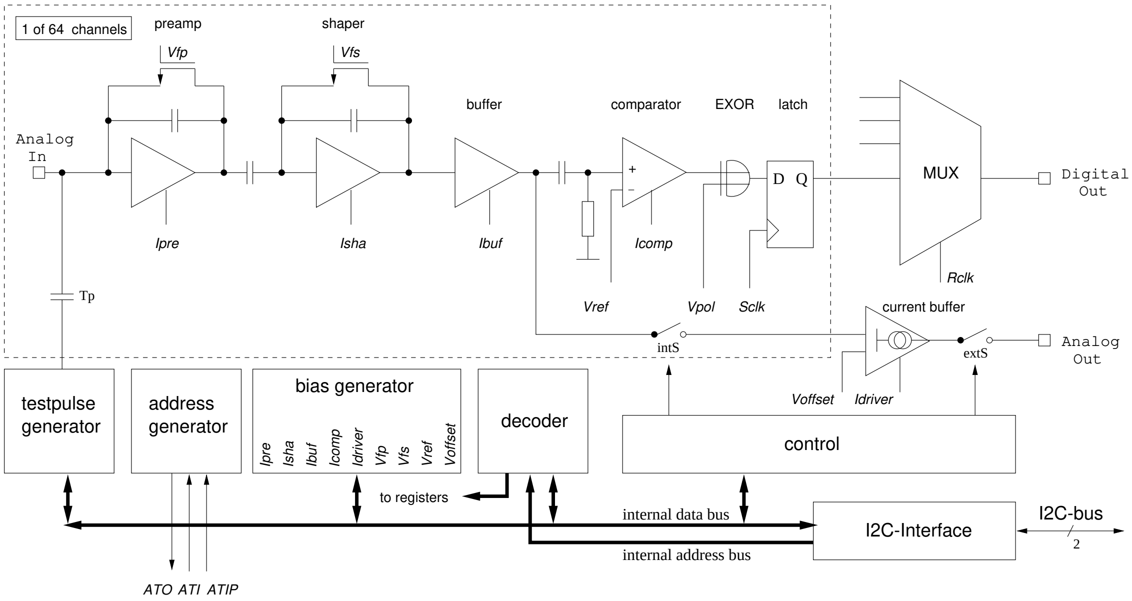

Abbildung 3.1: Blockdiagramm des CIPix1.0.

- 64 Kanäle bestehend aus
  - einem ladungsempfindlichen rauscharmen Vorverstärker mit einem Verstärkungsfaktor von etwa $40\text{mV}/10^5$ Elektronen.
  - einem CR-RC Pulsformer (Shaper), der einen semigausschen Puls mit einer Peaktime von 50...70ns[3] formt.
  - einer Treiberstufe (Buffer).
  - einem Komparator mit einstellbarer Polarität.
- Multiplexer: Der Multiplexer arbeitet mit der vierfachen Abtastrate, d. h. 40 MHz. Dieser Takt (*Rclk*) wird, wie die Abtastrate *Sclk*, extern zugeführt.
- Analoger Ausgangstreiber: Dieser führt einen der 64 analogen Kanäle als Stromausgang nach außen. Die Auskopplung des analogen Signals erfolgt nach dem Frontend und ist über die Programmierschnittstelle steuerbar.
- I^2C-Interface: Der CIPix benutzt das I^2C-Protokoll [I^2C98] zur Programmierung und zum Rücklesen der Register.
- Adressgenerator: Die Chip-Adresse wird in einem selbstprogrammierenden Verfahren vergeben. Dadurch wird vermieden, jeden Chip in unterschiedlicher Weise zu bonden[4], um so eine Kennung zu erreichen.
- Testpulsgenerator: Für Testzwecke lassen sich Pulse intern über Kapazitäten in die einzelnen Kanäle einkoppeln. Diese werden vom Testpulsgenerator erzeugt, der selbst über die Programmierschnittstelle gesteuert wird.

[3] Der Scheitelwert hängt von der Lastkapazität am Eingang ab (s. u.).
[4] Kontaktieren der Anschlußflächen eines Chips.

- Bias-Generatoren: Der CIPix enthält 5 Strom- und 4 Spannungsquellen. Die Ströme und Spannungen werden durch 8-Bit DACs[5] generiert und dienen zur Regelung der analogen Verstärkerkomponenten.

Abb. 3.2 zeigt das Layout des Chips. Um die einzelnen Komponenten besser identifizieren zu können, wurden diese entsprechend markiert und beschriftet.

3.2 Die äußeren Randbedingungen

Beim Betrieb des CIPix an der MWPC[6] sind zwei Aspekte zu berücksichtigen. Zum einen wird der Eingang des Vorverstärkers des CIPix durch die Signalleitung kapazitiv belastet, zum anderen stellt der CIPix durch seine Eingangsimpedanz einen Terminierungs-Widerstand für die Microstrip-Leitung dar.

Der Eingangsschaltkreis des CIPix ist ein *ladungsempfindlicher Vorverstärker*. Diese haben die Eigenschaft, daß ihr Rauschen proportional zur Lastkapazität am Eingang ist. Für den im CIPix verwendeten Vorverstärker wurde ein Rauschen von $380e^- + 38e^-/\mathrm{pF}$ gemessen [UT98]. Pulsformmessungen am Frontend des Helix2.0 [Pad98] zeigen, daß sich zum einen der Zeitpunkt der maximalen Signalhöhe (Peaktime) mit steigender Lastkapzität nach hinten verschiebt und zum anderen die Signalamplitude mit steigenden Lastkapazitäten abnimmt (Abb. 3.3). Dieses Verhalten hat mehrere Konsequenzen: Das Signal-zu-Rausch-Verhältnis (SNR) verschlechtert sich und die Triggerschwellen des Komparators müssen, im Verhältnis zum Rauschen, sehr klein eingestellt werden. Das wiederum bewirkt, daß der Trigger mehrere Buchcrossings anspricht. Vor diesem Hintergrund wurde die maximal zulässige Eingangskapazität auf 50pF festgelegt.

Michael Kollak entwickelte in seiner Diplomarbeit verschiedene Modelle einer Stripline-Auslese für die künftige CIP-Kammer [MK98]. In einem ersten Modell betrug die Lastkapazität für ein maximal weit vom CIPix entferntes Pad 200pF. Ein auf eine Kapazitätsminimierung optimiertes Modell liefert eine Lastkapazität für ein solches Pad von etwa 32pF.

Um Reflexionen am Ende einer Signalleitung zu vermeiden, muß diese mit ihrem Wellenwiderstand terminiert werden. Der CIPix schließt die Microstripline jedoch nicht mit dem richtigen Widerstand ab. Am CIPix-Eingang reflektierte Signale kommen dort nach einer Laufzeit von 22ns wieder an. Untersuchungen von Sven Löchner und Michael Kollak haben gezeigt, daß diese Fehlanpassung keine Konsequenzen für das Frontend hat.

3.3 Das Frontend: Preamplifier, Shaper, Buffer

Das im CIPix verwendete Frontend wurde für den Helix128 entwickelt und hinsichtlich dessen Anforderungen an Verstärkung, Signalform, Rauschen und Platzbedarf optimiert [WFB98]. Das Frontend gliedert sich in drei Stufen (Abb. 3.4). *Preamplifier* und *Shaper* haben prinzipiell den gleichen Aufbau, sie unterscheiden sich nur in der Dimensionierung der Transistoren.

Der *Preamplifier* selbst besteht aus einer Verstärker- und einer Rückkoppelstufe. Der Schaltplan ist in Abb. 3.5 dargestellt: Es wurde eine *gefaltete Kaskodenschaltung* verwendet. Der Transistor *M3* ist im Signalpfad in Reihe zu den parallel geschalteten Eingangstransistoren *M0* und *M1* geschaltet. Der Transistor *M2* dient zur Einstellung des Arbeitspunktes der

[5] DAC=Digital to Analog Converter
[6] Multi Wire Proportional Chamber

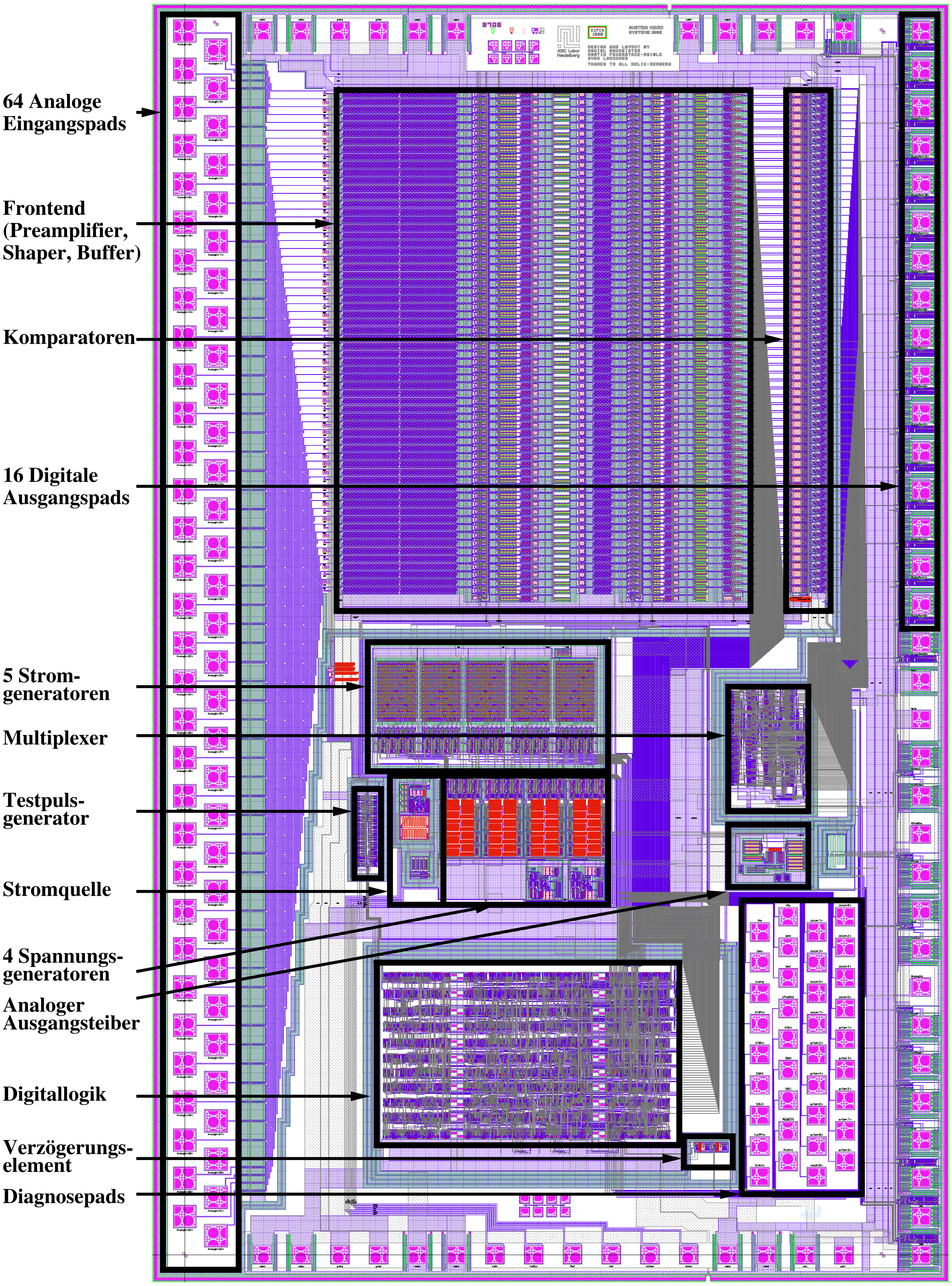

Abbildung 3.2: Layout-Ansicht des CIPix1.0.

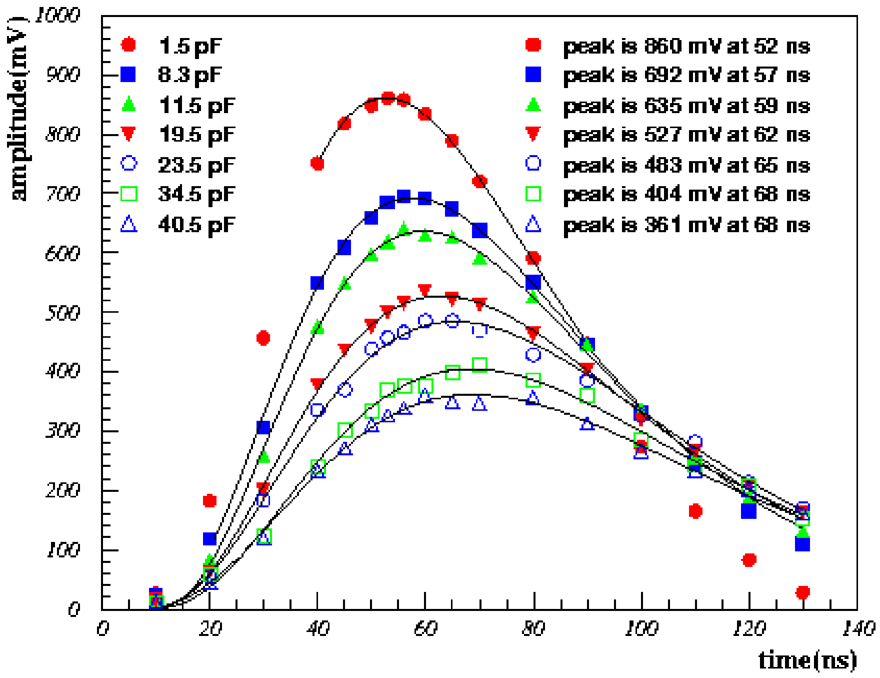

Abbildung 3.3: Pulsshape-Scan am Helix2.0 mit verschiedenen Lastkapazitäten. Es wurde ein Signal von –10MIP (1MIP=24.000e^-) auf Kanal 29 eingekoppelt [Pad98]. Die Zeit bis zum Pulsmaximum vergrößert sich von 52ns bei 1,5pF auf 68ns bei 40,5pF. Der Verstärkungsfaktor reduziert sich um einen Faktor 2 von 1,5pF auf 23,5pF.

Kaskodenschaltung. Durch die Spannung *pre_b1* am Gate von *M2* wird dessen Widerstand variiert. *M4* hat die Funktion einer Konstantstromquelle, deren Wert über die Spannung *pre_bias* am Gate eingestellt wird (entspricht *Ipre* in Abb. 3.4). Die Transistoren *M6* und *M7* bewirken einen Spannungsoffset am Ausgang. Der Rückkoppeltransistor *M5* hat die Aufgabe, die Kapazität *C1* zu entladen. Dadurch wird verhindert, daß sich die Rückkoppelkapazität durch mehrere aufeinanderfolgende Pulse auflädt und so der Verstärker in den Sättigungsbereich gerät. Die Spannung *Vfp* am Gate von *M5* regelt dessen Widerstand und bestimmt so die Zeitkonstante der Entladung.

Der auf das analoge Eingangspad eingekoppelte Ladungspuls wird auf das Gate der beiden Eingangstransistoren *M0* und *M1* geführt und durch die Rückkoppelkapazität *C1* invertierend integriert. Die Kapazität zwischen *Preamplifier* und *Shaper* bildet mit dem Widerstand des Rückkoppeltransistors des *Shapers* einen Hochpaß. Dieser differenziert das Signal, bevor es erneut durch den *Shaper* invertierend integriert wird. Die sich anschließende Treiberstufe (*Buffer*) ist als einfacher Source-Folger realisiert und erniedrigt den Innerwiderstand auf circa 1 kΩ.

Die Verstärkung des Vorverstärkers berechnet sich alleine aus der Rückkoppelkapazität nach $U = \frac{Q_{in}}{C}$. Für 10^5 eingekoppelte Elektronen am Eingang ergibt sich für *C1*=342 fF eine Ausgangsspannung von U=47 mV.

Im folgenden soll der Einfluß der einzelnen Regelspannungen und -ströme des Frontends auf die Signalform erläutert werden:

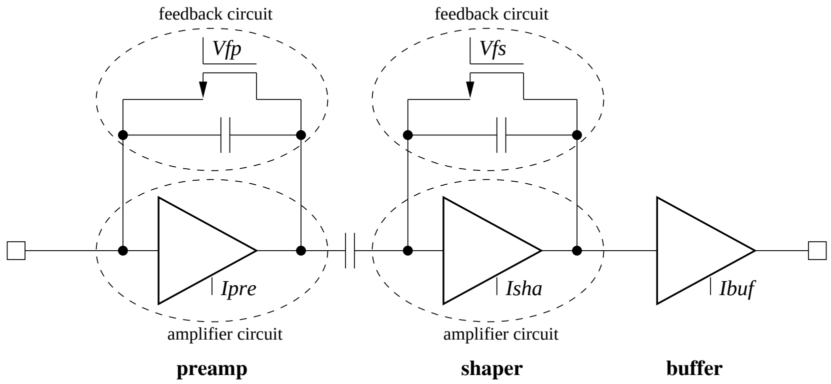

Abbildung 3.4: Blockschaltbild der analogen Eingangsstufe. Deutlich zu erkennen ist der dreiteilige Aufbau.

- Vorverstärker-Biasstrom *Ipre*
  Ipre bestimmt die Anstiegsgeschwindigkeit der Spannung am Ausgang des Vorverstärkers. Ein größerer Strom bewirkt eine längere Anstiegszeit und damit verbunden eine größere Amplitude des Ausgangssignals, da der Vorverstärker durch einen Bandpaß beschrieben werden kann. Ein grösserer Wert von *Ipre* reduziert zudem den Unterschwinger.

- Vorverstärker-Biasspannung *Vfp*
  Vfp steuert die Größe des Rückkoppelwiderstandes, der für eine Entladung der Rückkoppelkapazität sorgt und somit verhindert, daß der Verstärker in Sättigung gerät. Ist die Zeitkonstante dieses RC-Gliedes groß gegen die Integrationszeit des Pulsformers befindet sich der Verstärker im linearen Bereich und *Vfp* hat keinen Einfluß auf die Pulsform.

- Pulsformer-Biasstrom *Isha*
  Isha hat den gleichen Einfluß auf den Pulsformer wie *Ipre* auf den Vorverstärker.

- Pulsformer-Biasspannung *Vfs*
  Vfs steuert die Größe des Rückkoppelwiderstand des Pulsformers wodurch die Integrationszeit des Shapers bestimmt wird. Ein grösserer Wert von *Vfs* erniedrigt den Widerstand und der Puls fällt schneller auf die Basislinie ab. Dadurch wird auch die Amplitude am Ausgang erniedrigt, da nicht alle Ladungsträger am Eingang aufintegriert werden können.

- Treiber-Biasstrom *Ibuf*
  Ibuf hat keinen Einfluß auf die Pulsform.

Typische Werte für die Regelspannungen und -ströme mit denen der CIPix bei den Messungen programmiert wurde, sind in Tabelle 6.2 in Kapitel 6 aufgeführt.

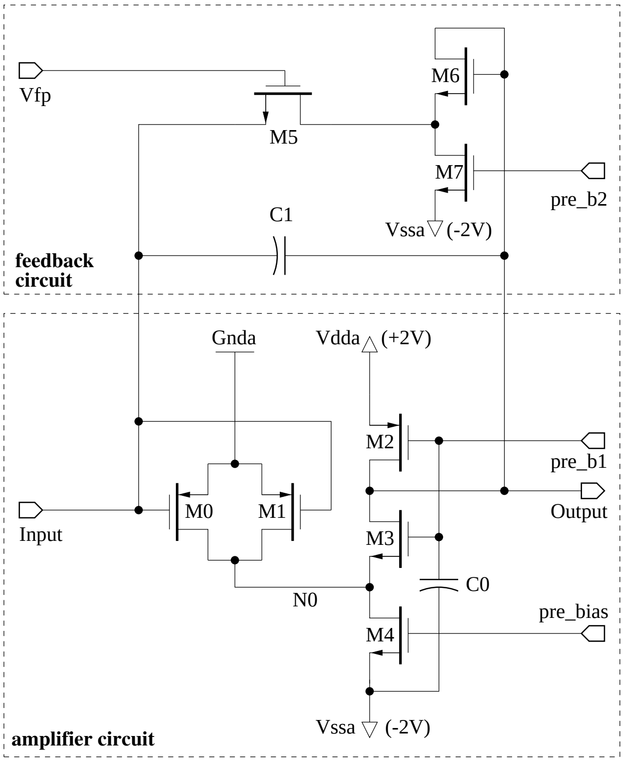

Abbildung 3.5: Schaltplan des Vorverstärkers [WFB98].

3.4 Der Komparator

Der Komparator wurde von B. Glass [BG97] für den Helix128 entwickelt und von S. Löchner [SL98] für den CIPix modifiziert. Er hat die Aufgabe, aus dem verstärkten und geformten analogen Eingangssignal beim Überschreiten einer einstellbaren Referenzspannung ein digitales Signal zu erzeugen. Die Komparatorschaltung setzt sich aus einer AC-Kopplung, dem Komparator selbst, einem EXOR-Gatter und einem D-Flip-Flop zusammen (Abb. 3.6).

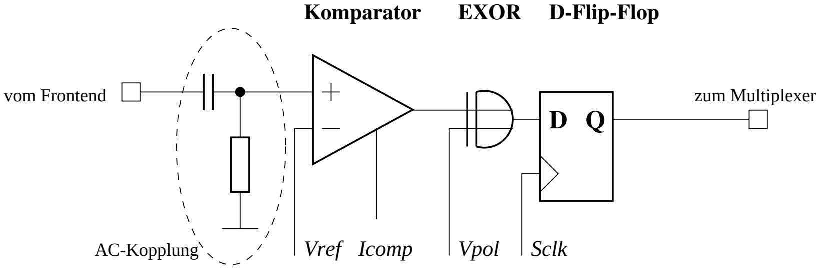

Abbildung 3.6: Schematischer Aufbau der Komparatorstufe.

AC-Kopplung

Eine AC-Kopplung dient dazu, die Gleichspannungskomponente eines Signales zu unterdrücken. Der CIPix nutzt die AC-Kopplung aus, um Offsetvariationen zwischen verschiedenen Kanälen zu beseitigen, mit dem Ziel, bei allen Kanälen die gleiche Referenzspannung zu verwenden.

Eine gebräuchliche Art der Realisierung einer solchen Kopplung ist ein Hochpaß. Ein CR-Glied mit hinreichend tiefer Grenzfrequenz (d. h. großer Zeitkonstante τ=RC) entkoppelt niedrige Frequenzen und unterdrückt so den Gleichspannungsanteil des Signals. Die angestrebte Zeitkonstante liegt in der Größenordnung von einigen Mikrosekunden.

Bei integrierten Schaltkreisen können Kapazitäten in der Größenordnung von 100 fF bis 10 pF hergestellt werden. Kleinere Kapazitäten werden vergleichbar mit parasitären Effekten, größere Werte haben einen zu großen Platzbedarf. Der benötigte Widerstand liegt in der Größenordnung von 1 MΩ. Ohmsche Widerstände werden durch Bahnen aus einem möglichst hochohmigen Material realisiert. Bei dem verwendeten Prozeß besitzt die Polysiliziumschicht „Poly-2“ den größten Flächenwiderstand mit $R_{\square}^{poly2} = 27\frac{\Omega}{\square}$[7]. Ein Poly2-Widerstand hätte dann bei einer minimalen Breite von 1,6 μm eine Länge von $l = \frac{R}{R_{\square}^{poly2}} = 37mm$. Bei Chipabmessungen von einigen Millimetern ist das völlig ausgeschlossen. Stattdessen benutzt man zwei Transistoren mit großer Kanallänge L (Abb. 3.7), um einen großen Widerstand (R$\geq$ 2 MΩ) zu realisieren.

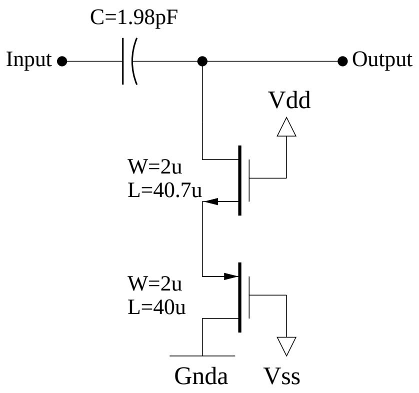

Abbildung 3.7: Schaltplan der AC-Kopplung des Komparators. Die beiden Transistoren bilden einen Widerstand in der Größenordnung 2 MΩ [WFB98].

Komparator

Der Komparator ist aus einer Differenzstufe und einem Drain-Verstärker aufgebaut (Abb. 3.8). Die Transistoren *M4* und *M8* arbeiten als Konstantstromquellen. *M4* bewirkt, daß die Summe der Ströme durch die beiden Zweige (*M0*, *M1*) und (*M2*, *M3*) konstant bleibt.

Im Ruhezustand sind die beiden Eingangsspannungen *Vin* und *Vref* gleich groß und der Referenzstrom I_{D_4} teilt sich aus Symmetriegründen auf die beiden Zweige auf.

Der „selbststeuernde“ Transistor *M5* (Gate und Drain sind miteinander verbunden) verhält sich im Sättigungsbereich wegen $R_{DS} = \frac{U_{DS}}{I_D}$ und $U_{GS} \sim I_D$ wie ein ohmscher Widerstand.

Die Spannungen $V1 = V_{dda} - V_{DS_5} = V2 = V_{dda} - V_{DS_6}$ bleiben solange konstant, wie die Ströme durch beide Zweige sich nicht ändern. Auch bei einer Änderung der Eingangsspannungen um den gleichen Wert (Gleichtaktunterdrückung) bleiben die Potentiale an diesen Knoten

[7] Der Flächenwiderstand ist als Widerstand pro Anzahl der Flächenquadrate einer Leiterbahn definiert. Ist W die Breite und L die Länge der Bahn, so gilt: $R_{\square} = \frac{R}{\square} = R\frac{W}{L}$.

gleich. Erst bei einer relativen Änderung der Eingangsspannungen ändert sich die Stromaufteilung in den Zweigen. Ist Vref > Vin, nimmt der Strom im rechten Zweig zu, im linken ab. Die Summe bleibt jedoch konstant gleich I_{D_4}. Damit sinkt $I_{D_5} \sim V1 = V_{GS_5} = V_{GS_6}$. Für *M6* gilt $R_{DS_6} \sim \frac{V_{DS_5}}{I_{D_6}}$. Wenn V_{GS_6} sinkt, vergrößert sich R_{DS_6} und $V2 = V_{dda} - V_{DS_6}$ sinkt.

Die Spannung *V2* regelt den Widerstand des Transistors *M7*, der als Drain-Verstärker arbeitet. Der Arbeitspunkt wird über den Widerstand des Transistors *M8* eingestellt. Sinkt die Spannung *V2* verringert sich der Widerstand von *M7*, was zu einer höheren Ausgangsspannung *Vout* führt.

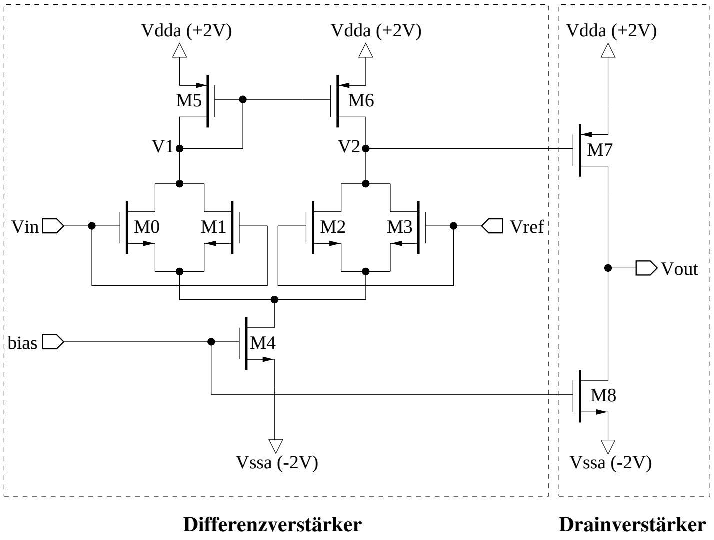

Abbildung 3.8: Schaltplan des differentiellen Komparators [BG97].

Nachfolgend sollen die Spannungs- und Strom-Parameter des Komparators erläutert werden:

- Die Referenzspannung *Vref* bestimmt die Schaltschwelle des Komparators. Der Wert ist über die I^{2}C-Schnittstelle im Bereich von -200 mV bis $+200$ mV mit einer Schrittweite von 1,5625 mV programmierbar. Die von den Bias-Generatoren erzeugte Spannung zwischen -2 V und $+2$ V wird durch einen Spannungsteiler um den Faktor 10 reduziert.

- Der Komparator-Biasstrom *Icomp* bestimmt den Referenzstrom für die beiden Konstantstromquellen. Ein grösserer Strom erhöht die Verstärkung und damit die Geschwindigkeit des Komparators.

Typische Werte für *Vref* und *Icomp* werden in Tabelle 6.2 in Kapitel 6 angegeben.

EXOR

Ein EXOR[8] kann als „optionaler" Inverter aufgefaßt werden: Liegt eine 0 am Steuereingang ist der Ausgang *gleich* dem Wert des Signaleingangs. Liegt eine 1 am Steuereingang nimmt der Ausgang den *invertierten* Wert des Signaleingangs an.

[8]Exclusive OR: $a \oplus b = \overline{a}b + a\overline{b}$

Im CIPix wird diese Eigenschaft eines EXORs dazu benutzt, Eingangssignale beider Polaritäten behandeln zu können. Durch die externe Spannung *Vpol* kann die invertierende oder nicht-invertierende Funktion des Gatters selektiert werden. Dabei entspricht ein Spannungswert von 3,3 V einer logischen 1 und ein Pegel von 0 V logisch 0.

D-Flip-Flop

Das D-Flip-Flop dient zur Synchronisation des EXOR-Ausgangs mit der steigenden Flanke der Sampling-Clock *Sclk*. Für den Trigger ist es wichtig, daß ein Triggersignal zum Zeitpunkt der steigenden Flanke von *Sclk* anliegt. Ist das Komparatorsignal nicht zeitgleich mit der *Sclk*-Flanke, wird es bis zur nächsten aktiven Flanke im Fip-Flop zwischengespeichert.

3.5 Der Multiplexer

Das Prinzip eines Multiplexers [SL98] besteht darin, mehrere Eingangssignale in einer zeitlichen Abfolge auf einen Ausgang zu führen (Abb. 3.9). Dadurch reduziert man die Anzahl der Signalleitungen.

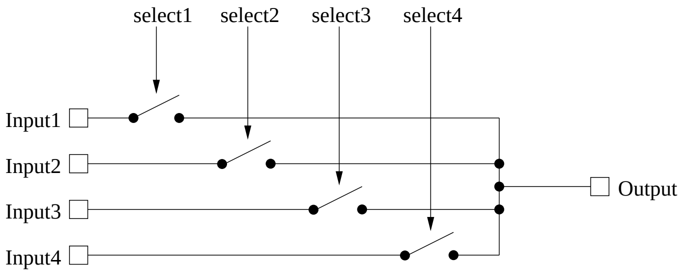

Abbildung 3.9: Funktionsprinzip eines Multiplexers.

Im CIPix werden vier Komparatorausgänge periodisch auf einen Digitalausgang des Chips geführt. Die Periodendauer des Auslesetaktes *Rclk* beträgt 25ns. Wichtig ist hierbei, daß der Takt *Rclk* phasengleich mit *Sclk* ist. Um einen ordnungsgemäßen Betrieb des Multiplexers zu gewährleisten, muß eine gewisse Anfangsbedingung erfüllt sein: Nach einem Reset-Signal (*notReset* geht auf 1) muß eine positive Flanke von *Sclk* und *Rclk* folgen (Abb. 3.10).

Der Multiplexer des CIPix ist zweistufig aufgebaut. In einer ersten Stufe werden aus dem Auslesetakt die Signale generiert, die einen der vier Eingänge selektieren. Abb. 3.10 stellt die zeiliche Abfolge der Selektionsleitungen *select[3:0]* dar. Es ist jeweils nur eines der vier Signale für die Dauer einer Periode aktiv. Im zweiten Teil der Schaltung erfolgt die eigentliche Aneinanderreihung der Eingangsignale (Abb. 3.11). Der Multiplexer-Ausgang wird zudem durch ein Flip-Flop mit der negativen Flanke des Taktes *Rclk* synchronisiert.

Die Zuordnung einer Kanalnummer zu einem Digitalausgang und einer *Rclk*-Periode kann der Tabelle 3.1 entnommen werden.

Zusätzlich wird im Multiplexer ein weiteres Ausgangssignal des CIPix generiert. Das Signal *EmptyDataSet* soll nachfolgenden Stufen in der Auslesekette anzeigen, ob überhaupt ein Komparator ein Triggersignal erzeugt hat. Dazu wird eine ODER-Verknüpfnung aller 64 Komparator-Ausgänge gebildet. Wurde mindestens ein Triggersignal auf einem Kanal erzeugt, ist *EmptyDataSet* 1, ansonsten 0.

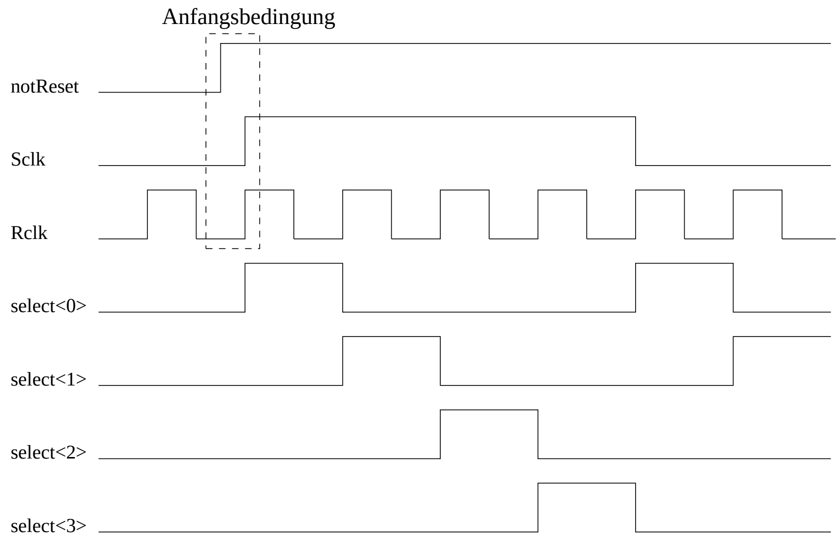

Abbildung 3.10: Multiplexer-Startbedingung und zeitliche Abfolge der Selektionsleitungen [SL98].

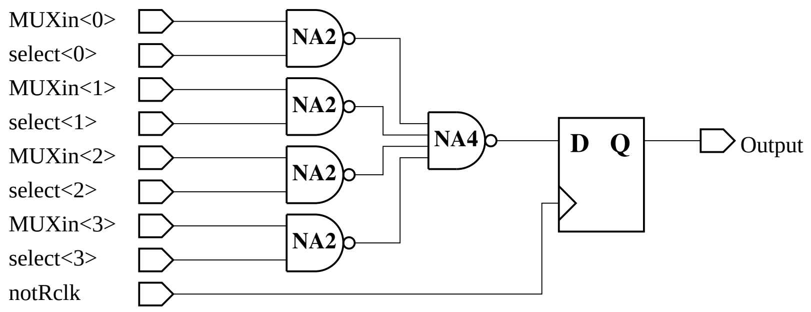

Abbildung 3.11: Schaltplan der Multiplexerstufe [SL98].

3.6 Der Analog-Ausgang

Der CIPix soll primär Triggerinformationen liefern. Deshalb werden die analogen Eingangssignale digitalisiert. Um die Arbeitsweise des Chips und Vorgänge in der Kammer besser überwachen zu können, wurde jedoch die Möglichkeit vorgesehen, einen der 64 analogen Eingangskanäle nach dem Frontend auszukoppeln und über einen Stromtreiber [WFB98] nach außen zu führen. Die Auswahl eines Kanals erfolgt durch Programmierung des Registers *AnalogOut* über die I^2C-Schnittstelle. Entsprechend dem Registerwert wird einer der 64 Schalter, die als n-MOS Transistoren realisiert sind, geschlossen. Es ist vorgesehen die analogen Ausgänge mehrerer Chips über ein Kabel auszulesen. Ein zusätzlicher Schalter regelt deshalb den konfliktfreien Zugriff auf diesen analogen Bus. Der Schalter ist durch ein sogenanntes *Transmission-Gate* realisiert und wird vom MSB des Registers *AnalogOut* gesteuert.

Digital-Ausgang	*Rclk*-Periode			
	1	2	3	4
0	0	16	32	48
1	1	17	33	49
2	2	18	34	50
3	3	19	35	51
4	4	20	36	52
5	5	21	37	53
6	6	22	38	54
7	7	23	39	55
8	8	24	40	56
9	9	25	41	57
10	10	26	42	58
11	11	27	43	59
12	12	28	44	60
13	13	29	45	61
14	14	30	46	62
15	15	31	47	63

Tabelle 3.1: Zuordnung der analogen Kanäle (0–63) zu den Digitalausgängen (0–15) und *Rclk*-Perioden.

3.7 Der Testpuls-Generator

Mit Testpulsen hat man ein Mittel zur Hand, unabhängig von den analogen Eingangssignalen, die Auslesekette des CIPix zu testen. Dazu wird über eine Kapazität auf dem Chip ein deltaförmiger Ladungsimpuls in den Vorverstärker eingekoppelt. Die Einkoppelstufe ist so konstruiert, daß auf vier benachbarte Kanäle jeweils unterschiedliche Ladungen gegeben werden. Bei einem Spannungshub von 4 V werden Ladungen von $Q = +10^5 e^-$, $Q = +5 \cdot 10^4 e^-$, $Q = -10^5 e^-$ und $Q = -5 \cdot 10^4 e^-$ eingekoppelt. Abb. 3.12 verdeutlicht das Schema, welches sich periodisch wiederholt.

Es wurden mehrere Modi der Testpulseinkopplung realisiert. Zum einen kann zwischen einer internen Generierung eines Pulses und der Einkopplung von außen über das Pad *fcsTP* gewählt werden. Zum anderen kann ein Testpuls auf alle 64 oder auf vier ausgewählte Kanäle (0, 21, 42, 63) gegeben werden. Die Einkopplung auf nur vier Kanäle hat den Vorteil, daß in Abhängigkeit der eingestellten Komparator-Schwelle, maximal ein weiterer Kanal neben dem gerade betrachteten schaltet. Somit kann ein Übersprechen von anderen Kanälen vermieden werden.

Die oben genannten Testpulsmodi lassen sich durch Programmieren des Registers *TpReg* über die I^2C-Schnittstelle selektieren. Die internen Testpulse werden von einer Steuerlogik abgeleitet, die in Kapitel 5 ausführlicher erklärt wird.

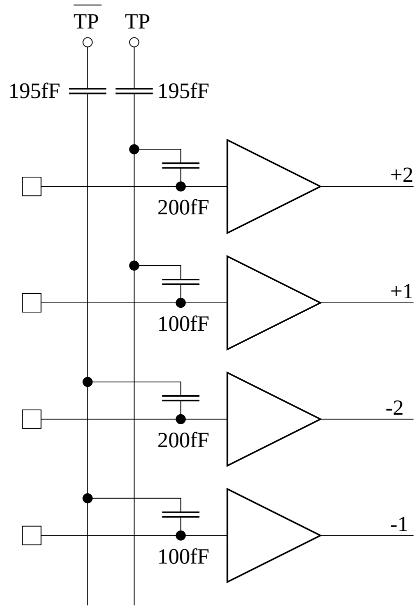

Abbildung 3.12: Schema der Testpulseinkopplung.

3.8 Die Bias-Generatoren

Die Biasgeneratoren [UT99] haben die Aufgabe, die für die Regelung des Frontends, des Komparators und des analogen Ausgangstreibers notwendigen Spannungen und Ströme zu liefern. Der CIPix besitzt 5 Strom- und 4 Spannungsquellen. Die Ströme und Spannungen werden von 8-Bit DACs generiert, die über die I^2C-Schnittstelle programmiert werden.

Die Biasstrom-Quellen

Die Stromquellen arbeiten mit einer Auflösung von 2,5 μA/LSB. Ein Registerwert von 0 entspricht einem Ausgangsstrom von 0 A. Um eine korrekte Arbeitsweise zu gewährleisten benötigen die Stromquellen einen Referenzstrom von 100 μA, der in das Pad *IrefIn* fließen muß. Dieser Referenzstrom kann entweder extern zugeführt werden oder intern durch eine Referenzquelle generiert werden. Wird die interne Stromquelle benutzt, muß ein 20 kΩ Widerstand vom Pad *Rref* gegen *Vssa* (-2 V) gelegt werden und das Ausgangspad *IrefOut* direkt mit dem Eingangspad *IrefIn* verbunden werden (siehe Anhang B).

Die Biasspannungs-Quellen

Die Spannungsquellen arbeiten mit einer Auflösung von 15 mV/LSB im Spannungsbereich von -2 V bis $+2$ V. Ein Registerwert von 0 entspricht einer Ausgangsspannung von -2 V.

3.9 Die Pads

Der CIPix arbeitet intern mit Spannungen von -2 V (*Vss*) bis $+2$ V (*Vdd*). Bei den Spezifikationen zum Chip wurde die Philosophie verfolgt, nach außen hin für digitale Signale nur CMOS-Spannungspegel von 0 bis 3,3 V zu verwenden. Das macht eine Anpassung beider Pegelbereiche durch sogenannte *Levelshifter* notwendig. Insgesamt besitzt der CIPix vier verschiedene Typen von digitalen Anschlusspads, für die M. Feuerstack-Raible jeweils einen Levelshifter entwickelte [MFR98b]:

- Eingangs-Levelshifter für die Pads *ATI*, *ATIP*, *notReset*, *fcsTP*, *Pause* und *Vpol*.
- Ausgangs-Levelshifter für die Pads *ATO*, *DigitalOut[15:0]* und *EmptyDataSet*.
- Bidirektionale Levelshifter für die Pads *SCL* und *SDA*.
- LVDS[9]-Eingangs-Levelshifter für die Pads *Sclk*, $\overline{Sclk}$, *Rclk* und $\overline{Rclk}$.

Eingangs-Levelshifter

Ein Schaltplan des Eingangs-Levelshifter ist in Abb. 3.13 dargestellt. Man erkennt deutlich den zweiteiligen Aufbau. Die erste Stufe ist ein Differenzverstärker, in dem die Eingangsspannung mit dem Referenzwert *vref* verglichen wird. Das Ausgangssignal der Differenzstufe wird auf eine Inverterkette geführt, die die Pegel auf ±2 V anpaßt. Die Regelspannungen *vbias* und *vref* haben Werte von 2,5 V und 1,65 V und werden intern durch einen Spannungsteiler aus *Vcc* generiert.

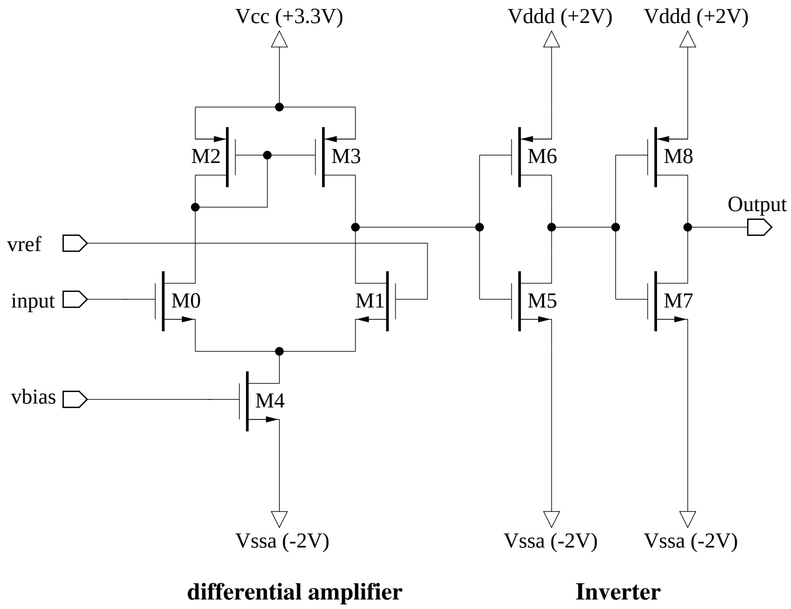

Abbildung 3.13: Schaltplan des Eingangs-Levelsifters [MFR98b].

[9]LVDS = Low Voltage Differential Signaling. Bei dieser Methode der Datenübertragung nutzt man die Amplitudendifferenz (Swing) zweier differentieller Signale als Information (Differentielle Signale besitzen entgegengesetzte Amplituden). Der Vorteil dieser Methode liegt in einem reduzierten Leistungsverbrauch und einer Unempfindlichkeit gegenüber Rauschen, da Störungen auf beiden Signalen in gleicher Weise induziert werden (common-mode noise).

Ausgangs-Levelshifter

Der Ausgangs-Levelshifter besteht aus einer Kette von vier Invertern (Abb. 3.14). Der erste Inverter arbeitet zwischen ±2 V, alle weiteren zwischen 0 und 3,3 V. Die Dimensionierung der Transistoren wird von Stufe zu Stufe größer.

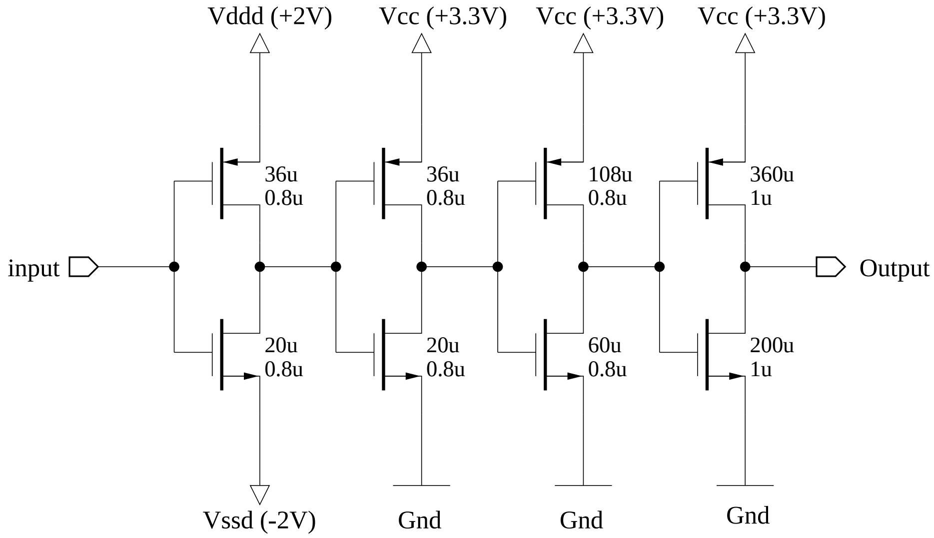

Abbildung 3.14: Schaltplan des Ausgangs-Levelshifter. Die Zahlen an den Transistoren geben Breite (W) und Länge (L) an. Die Dimensionierung wird von Inverter zu Inverter größer [MFR98b].

Bidirektionale Levelshifter

Die bidirektionalen Levelshifter werden für die beiden I^{2}C-Leitungen *SCL* und *SDA* eingesetzt. Hierfür ist es notwendig, eine Tristate-Funktion[10] zu implementieren. Das bedeutet, daß durch ein Steuersignal der Ausgang in einen hochohmigen Zustand gebracht werden kann. Der Levelshifter ist eine Kombination der zuvor beschriebenen Eingangs- und Ausgangs-Levelshifter. Zusätzlich kann jedoch der Ausgang durch ein Steuersignal abgeschaltet werden (Abb.3.15).

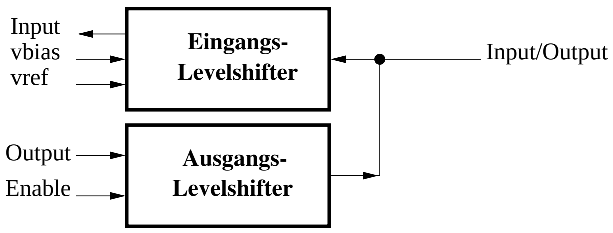

Abbildung 3.15: Schematischer Aufbau der bidirektionalen Levelshifter [MFR98b].

[10]Die Motivation hierfür wird in Kapitel 5 gegeben.

LVDS-Eingangs-Levelshifter

Der LVDS-Eingangs-Levelshifter ist baugleich zu dem oben beschriebenen Eingangs-Levelshifter. Anstelle der Regelspannung *vref* wird jedoch das invertierte Eingangssignal auf den Differenzeingang geführt.

3.10 Die Spannungsversorgung

Die analogen und digitalen Spannungsnetze sind auf dem CIPix streng voneinander getrennt. Insgesamt existieren sieben verschiedene Spannungen, die in Tabelle 3.2 zusammengestellt sind.

Symbol	Elektrischer Pegel	Beschreibung
Vdda	+2V	Positive analoge Versorgungsspannung
Vssa	-2V	Negative analoge Versorgungsspannung
Gnda	0V	Analoge Masse
Vddd	+2V	Positive digitale Versorgungsspannung
Vssd	-2V	Negative digitale Versorgungsspannung
Gnd	0V	Digitale Masse
Vcc	3.3V	Digitale I/O Versorgungsspannung

Tabelle 3.2: Versorgungsspannungen des CIPix.

Der analoge Massenanschluß *Gnda* soll mit der Detektormasse verbunden werden, um so eine gemeinsame Potentialreferenz von Chip und Detektor zu gewährleisten. Durch die Trennung der Spannungen auf dem Chip will man ein Übersprechen von digitalen Schaltstörungen auf analoge Komponenten, vor allem auf den Vorverstärker, verhindern. Außerhalb des Chips können die analogen und digitalen Versorgungsspannungen gemeinsam zugeführt werden. Allerdings sind die einzelnen Leitungen durch Kapazitäten gegen Störungen abzublocken. Im Anhang B ist ein Plan für die externe Beschaltung des Chips mit Blockkapazitäten gezeigt.

Kapitel 4

Digital-Design

Im folgenden Kapitel sollen nach einer kurzen Einführung in die CMOS-Technologie Prinzipien und Methoden der Entwicklung eines digitalen Designs dargestellt werden. Anschließend wird die Schaltungsentwicklung mit Hilfe der Hardwarebeschreibungssprache Verilog und der CAD[1]-Software *Cadence* [Cad] beschrieben.

4.1 CMOS-Technologie

Das wesentliche Bauelement der MOS[2]-Technologie stellt der Feldeffekttransistor (MOSFET[3]) dar. FETs besitzen eine kompakte Bauweise, gute Linearität und zeichnen sich im Vergleich zu Bipolartransistoren durch eine geringere Leistungsaufnahme aus.

Man unterscheidet weiter zwischen n- und p-Kanal MOSFET (Abb. 4.1). Ein n-Kanal MOSFET (auch n-MOSFET) wird in einem p-dotierten Substrat (B, wie Bulk) aus zwei stark n-dotierten Bereichen Drain (D) und Source (S) gebildet. Beide Gebiete sind räumlich voneinander getrennt. Das Prinzip des FETs besteht nun darin, den Leitwert eines leitenden Kanals zwischen Source und Drain durch eine Steuerelektrode, das Gate (G), zu regeln. Das Gate ist durch eine leitfähige Schicht aus Metall oder Polysilizium realisiert und vom Substrat durch eine isolierende Oxidschicht getrennt. Beim p-Kanal MOSFET (p-MOSFET) sind die Dotierungsarten gerade umgekehrt. Zwei stark p-dotierte Bereiche sind in einem n-Substrat integriert.

Beim CMOS-Prozeß (Complementary MOS) werden beide MOSFET-Arten auf *einem* Substrat untergebracht. Dafür ist es notwendig, einen der beiden Transistortypen in einer dem Substrat entgegengesetzt dotierten Wanne (well) zu integrieren. Je nach der Dotierungsart der Wanne spricht man von n-well CMOS oder p-well CMOS. Abb. 4.1 zeigt einen n- und p-MOSFET in einer p-well CMOS-Technologie. Die Schaltungssymbole von n- und p-MOSFET sind in Abb. 4.1 dargestellt.

Durch die Steuerspannung U_{GS} läßt sich nun der Widerstand zwischen Drain und Source regeln. Ist $V_{GS} = V_{BS} = 0$, fließt bei $V_{DS} \neq 0$ lediglich der Sättigungsstrom, da es sich auf der Strecke Source-Drain um zwei pn-Übergänge handelt, von denen einer in Sperrichtung geschaltet ist. Legt man eine positive Spannung an das Gate eines n-MOSFETs, so sammeln sich negative Ladungsträger unter der Oxidschicht und es kommt, ab einer bestimmten

[1] Computer Aided Design
[2] Metal-Oxid-Semiconductor
[3] Metal Oxid Semiconductor Field Effect Transistor

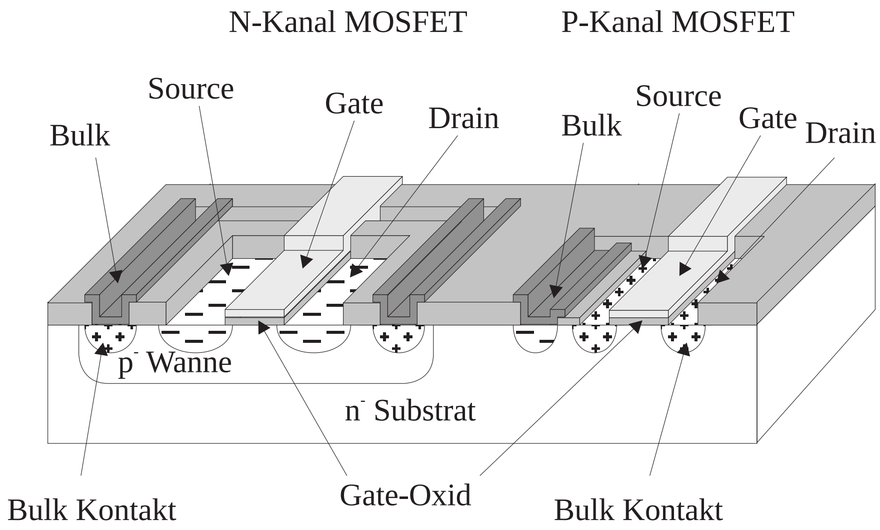

Abbildung 4.1: p-MOSFET und n-MOSFET in einer p-well CMOS-Technologie [MFR98a].

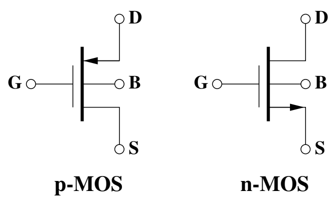

Abbildung 4.2: Schaltungssymbole eines p-MOS- (links) und n-MOS-Transistors (rechts). Ist der Bulk-Kontakt nicht eingezeichnet, wird er bei n-MOSFETs mit dem Source-Kontak, bei p-MOSFETs mit dem Drain-Kontakt verbunden.

Schwellenspannung V_{TH}, zu einer Ladungsträgerinversion. Im p-dotierten Substrat bildet sich unter dem Gateoxid ein Ladungsträgerkanal mit freien Elektronen aus. Bei anliegender Spannung U_{DS} zwischen Drain und Source kann jetzt ein Strom fließen.

Prinzipiell verhalten sich n- und p-Kanal MOSFETs identisch. Aufgrund der unterschiedlichen Mobilität ($\mu_p^{Si} = 450\text{cm}^2/\text{Vs}$, $\mu_e^{Si} = 1500\text{cm}^2/\text{Vs} \approx 3\mu_p^{Si}$) und Polarität der Ladungsträger sind die Übertragungskennlinien (Strom-Spannungskennlinien) jedoch voneinander verschieden. Die Schwellenspannungen beider Transistortypen besitzen gerade entgegengesetzte Vorzeichen. In Abbildung 4.3 (links) ist eine Strom-Spannungs-Kennlinienschar eines MOSFETs gezeigt. Aufgetragen ist der Drainstrom als Funktion der Drain-Source-Spannung für verschiedene Werte der Gate-Source-Spannung. Man unterscheidet zwei Bereiche: den Ohmschen Bereich und den Sättigungsbereich. Im Ohmschen Bereich verhält sich der Transistor wie ein nichtlinearer ohmscher Widerstand. Im Sättigungsbereich sind alle freien Ladungsträger am Stromtransport beteiligt und der MOSFET verhält sich fast wie eine ideale Stromquelle. Bei hinreichend großen Drain-Source-Spannungen kommt es zu einem Durchbruch zwischen Drain und Substrat und der lineare Kennlinienverlauf endet. Die Abhängigkeit des Drainstromes von der Gatespannung, die sogenannte Transkonduktanzkurve, ist in Abb. 4.3 (rechts) gezeigt. Eine genauere Betrachtung zeigt, daß auch hier zwei Bereiche existieren: ein Bereich mit quadratischer Übertragungskennlinie und der sogenannte Subthreshold-Bereich, der eine exponentielle Übertragungscharakteristik besitzt.

Neben dem Transistor als aktivem Element, können in der MOS-Technologie als pas-

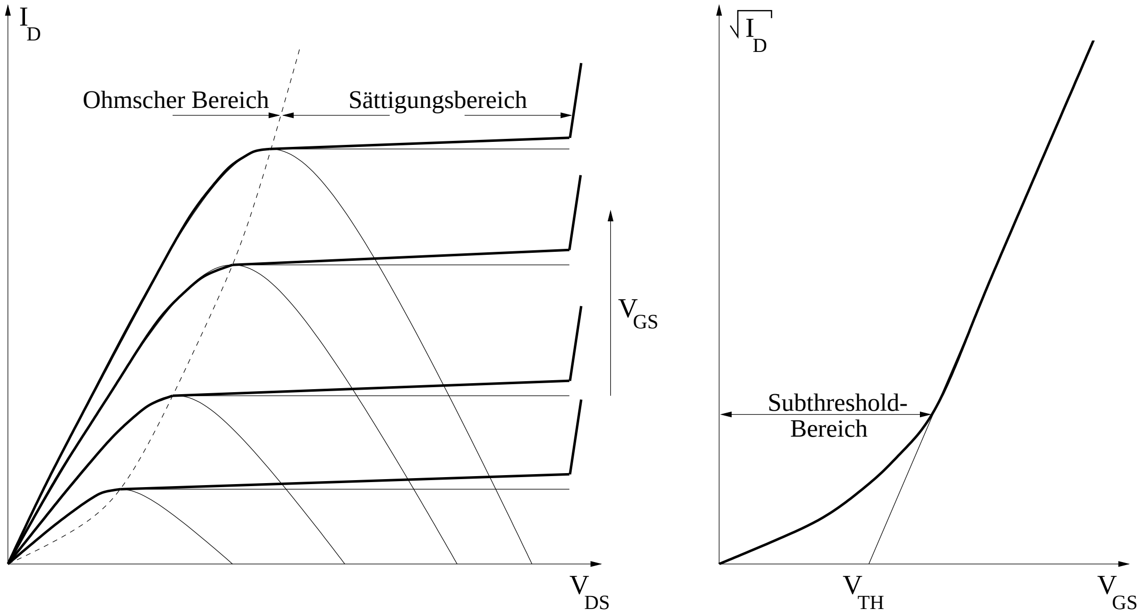

Abbildung 4.3: Übertragungskennlinie (links) und Transkonduktanzkurve (rechts) eines MOSFETs [MFR98a].

sive Bauelemente Widerstände und Kondensatoren hergestellt werden. Induktivitäten lassen sich jedoch nicht realisieren. Die einzelnen Elemente werden durch eine bestimmte Abfolge verschiedener Prozesschritte realisiert. In dem 0,8 μm CMOS-Prozess[4], in dem der CIPix1.0 gefertigt wurde, stehen neben verschiedenen Diffusionszonen im wesentlichen zwei Metallagen und zwei Polysiliziumlagen zur Verfügung.

Niederohmige Widerstände werden durch Polysiliziumbahnen gebildet. Da diese einen großen Flächenbedarf haben, werden hochohmige Widerstände entweder durch sogenannte n-well-Widerstände realisiert oder in einer Transistorschaltung implementiert. Kapazitäten werden durch zwei übereinanderliegende Polysiliziumschichten gebildet, die durch eine Oxidschicht als Dielektrikum voneinander getrennt sind. Die spezifische Kapazität zweier überlappender Schichten liegt bei etwa $1\frac{fF}{\mu m^2}$. Damit können Kapazitäten in der Größenordnung von 100fF bis 10pF hergestellt werden. Kleinere Kapazitäten werden vergleichbar mit parasitären Effekten, größere Werte haben einen zu großen Platzbedarf. Parasitäre Kapazitäten kommen aufgrund der Leiterbahnführung zwischen einzelnen Materialschichten zustande.

4.2 Design-Methoden

Eine digitale Schaltung kann von einer mikroskopischen Ebene aus, auf der Basis von Transistoren und Gattern, oder von einem makroskopischen Standpunkt aus aufgebaut werden. Die makroskopische Sicht wird auch Top-Down-Design genannt und ist die bevorzugte Methode in modernen Entwicklungen. Sie soll daher hier kurz vorgestellt werden.

Nach D. Winkler [DW80] ist einer der ersten Schritte eines Top-Down-Designs die Unterteilung des Entwurfs in einen *Steuer-Algorithmus* (Controller) und eine *Architektur*, die von

[4]Die Zahl gibt die minimale Strukturbreite an.

diesem Algorithmus kontrolliert wird (siehe Abb. 4.4). Der Controller sendet eine sequentielle Folge von Kommandos an das zu steuernde Gerät. Im Gegenzug sendet das Gerät Statusinformationen an die Steuer-Einheit, die diese als Entscheidungsvariablen in ihrem Algorithmus benutzt.

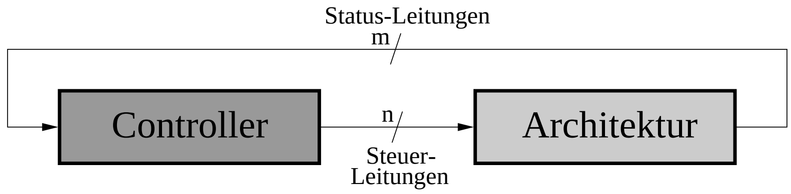

Abbildung 4.4: Schematische Struktur eines digitalen Top-Down Designs [DW80].

Der entscheidende Vorteil einer solchen Herangehensweise ist es, daß man sich von Hardware-Eigenschaften so lange wie möglich löst und nur abstrakte Bausteine wie Speicher, Register und arithmetische Einheiten verwendet.

Zustandsmaschine

Das zentrale Element in einem digitalen Top-Down-Design ist die Steuereinheit. Diese wird in Form einer *Zustandsmaschine* implementiert. Das Konzept der Zustandsmaschine soll im folgenden erläutert werden: Eine *Zustands-Maschine* (state machine) ist ein synchroner Schaltkreis, der sich zu jedem Zeitpunkt in einem definierten Zustand befindet. Ein Zustand ist dabei durch eine bestimmte Konfiguration von Eingangssignalen (Statussignalen) festgelegt. Synchron zu einem externen Takt erfolgt ein Wechsel der Zustände, wobei die Abfolge eindeutig als Funktion der Eingangssignale festgelegt ist. Jedem Zustand entspricht eine bestimmte Konfiguration von Ausgangssignalen (Steuersignalen).

Bei der Realisierung einer State-Machine in Hardware unterscheidet man zwei Typen: die MOORE-Machine und die MEALY-Machine. Beiden gemeinsam sind eine *Next-State-Logik*, die den Übergang in einen neuen Zustand bestimmt und ein *Zustandsregister*, in dem der aktuelle Zustand gespeichert wird. Die MEALY-Machine besitzt im Gegensatz zur MOORE-Machine noch eine *Ausgangs-Dekoder-Logik* (siehe Abb. 4.5). Fehlt diese Ausgangslogik, werden die Ausgänge des Zustandsregisters direkt nach außen geführt. Der Vorteil einer MOORE-Machine liegt darin, daß die Ausgangssignale aus den Flip-Flops garantiert frei von undefinierten Impulsen sind. Durch die Verwendung einer Ausgangs-Dekoder-Logik bei der MEALY-Machine wird das Design dagegen kompakter.

In Abbildung 4.6 ist die Umsetzung einer zeitlichen Abfolge von Zuständen in Abbhängigkeit eines Eingangssignals in ein *Zustandsdiagramm* exemplarisch dargestellt.

4.3 Der Design-Prozeß

Der Design-Prozeß läuft im wesentlichen in drei Stufen ab, wobei den ersten beiden Stufen jeweils Simulationsschritte nachfolgen (siehe Abb. 4.7). Am Anfang steht die Beschreibung des Schaltungsverhaltens mit Hilfe einer Hardware Beschreibungssprache (HDL[5]). Zwei weit

[5]Hardware Description Language

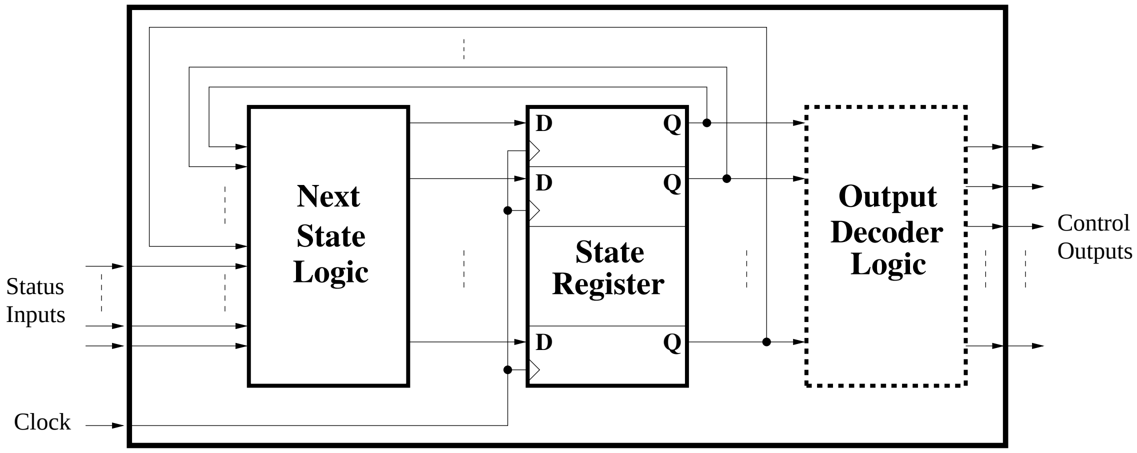

Abbildung 4.5: Schematische Struktur einer Zustandsmaschine [Syn97]. Die Ausgangs-Dekoder-Logik ist nur bei einer Mealy-Machine vorhanden. Bei der Moore-Machine fehlt sie, und die Flip-Flop-Ausgänge des Zustandsregisters werden direkt nach außen geführt.

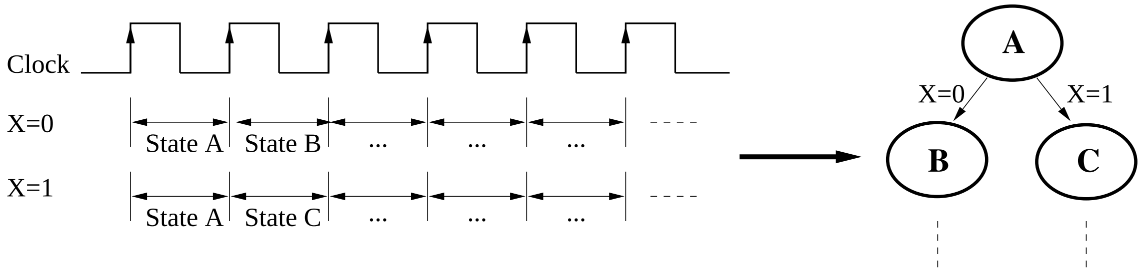

Abbildung 4.6: Umsetzung einer zeitlichen Abfolge von Zuständen in ein Zustandsdiagramm (Die Pfeile am Taktsignal kennzeichnen die aktive Flanke).

verbreitete Sprachen sind *Verilog* und *VHDL*[6]. Die Spezifizierung der Schaltung erfolgt auf einer völlig abstrakten Ebene. Es werden keine Aussagen über die Realisierung des Designs mit konkreten Schaltungselementen gemacht. Erst im nachfolgenden Schritt, der *Synthese*, wird die Schaltungsbeschreibung in einen Schaltplan umgewandelt. Dies geschieht weitgehend automatisch. Die verwendeten Schaltungselemente werden aus einer für den Zielprozeß spezifischen Bibliothek (Standardzellenbibliothek) entnommen. Die dritte Stufe ist die Generierung eines Layouts (*Place and Route*[7]). Hier wird der Schaltplan auf konkrete Prozesstrukturen abgebildet.

Die Simulationsschritte dienen dazu, das Schaltungsverhalten zu überprüfen. Dabei ist vor allem die Simulation nach erfolgter Synthese wichtig. Hier können erhebliche Veränderungen im Verhalten gegenüber der abstrakten Schaltungsbeschreibung auftreten. Vor allem bei asynchron arbeitenden Schaltungen ist man auf Gatterzeiten, wie Setup- und Holdzeiten oder auf undefinierte Impulse sensibel.

[6] Very High-Speed Integrated Circuit HDL
[7] Plazieren und Verbinden der Logik

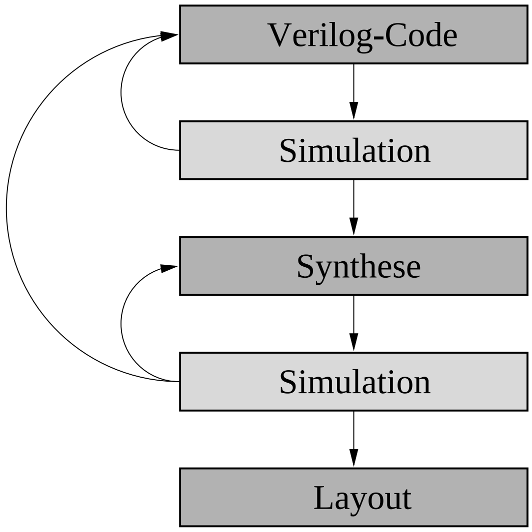

Abbildung 4.7: Typischer Top-Down ASIC-Design Prozeß

4.3.1 Schaltungsbeschreibung mit Verilog

Für die Digitalentwicklungen des CIPix wurde die Sprache *Verilog* verwendet. Zunächst stellt sich die Frage, warum man Hardware-Beschreibungssprachen benutzt. Der Grund für den Einsatz einer solchen Sprache liegt in der Komplexität heutiger Digital-Entwicklungen. Kleine Schaltungen lassen sich durch das Zeichnen eines Schaltplanes aus Elementen wie Transistoren, logischen Gattern und Flip-Flops realisieren. Durch moderne Technologien, die eine sehr hohe Dichte von Bauelementen ermöglichen, sind sehr große und komplexe Schaltungen zugänglich. Diese machen den Einsatz alternativer Methoden des Hardwareentwurfs notwendig. Für große digitale Systeme zeigt ein Schaltplan nur ein Netz von Verbindungen und nicht die *Funktionalität* des Designs.

Bei einer Hardwarebeschreibungssprache beschreibt man eine Schaltung nicht mehr als die Summe ihrer Bauteile, sondern man spezifiziert ihr Verhalten in einer abstrakten Sprache.

Verilog HDL

Die Sprache *Verilog* [TM96] [Gol95] dient sowohl zur *Beschreibung* als auch zur *Simulation* digitaler Logik. Zu den grundlegenden Modellierungskonzepten zählen Parallelität, Ereignissteuerung, Hierarchie und Zeitkontrolle. Man unterscheidet zwischen zwei grundlegenden Beschreibungsmodellen, die das Design jeweils auf unterschiedlichen Ebenen betrachten.

- Strukturmodell: Hier beschreibt man das Design als aus diskreten Elementen aufgebaut. Man verwendet entweder logische Gatter (gate-level model) oder Schalter (switch-level model) für die Beschreibung.

- Verhaltensmodell: Hier wird das *Verhalten* der Schaltung mit Hochsprachen-Elementen beschrieben.

Im folgenden soll nur das Verhaltensmodell betrachtet werden. Verilog lehnt sich mit seinen Sprachelementen stark an eine herkömmliche prozedurale, sequentielle[8] Programmiersprache an. Der wesentliche Unterschied besteht jedoch darin, daß nicht Variablen manipuliert, sondern Signale geschaltet werden. In realen Schaltungen arbeiten alle Komponenten parallel und besitzen Eigenschaften wie Laufzeiten, Verzögerungen et cetera. Diese Parallelität der Verarbeitung muß auch in der Beschreibungssprache nachgebildet werden. Da die Reihenfolge der abzuarbeitenden Befehle nicht von außen vorgegeben wird, muß die Ausführung einer Anweisung *ereignisgesteuert* erfolgen. Ein Ereignis ist dabei die Änderung eines bestimmten Signals. In der Simulation übernimmt ein Ereignis-Scheduler die Aufgabe des Programmzählers.

Verilog ist der Programmiersprache C sehr ähnlich. Es existieren alle typischen Sprachelemente wie Variablen-Deklarationen, Zuweisungen, arithmetische und logische Operationen, Bedingungs-Abfragen, Schleifen-Konstrukte, Unterprogramme, Ein- und Ausgabe-Routinen. Zusätzlich sind Konstrukte vorhanden, die die parallele und ereignisgesteuerte Ausführung von Anweisungen ermöglichen, sowie Sprachelemente zur Definition von Signalverzögerungen.

Im Hinblick auf die spätere Umsetzung der Beschreibung in einen Schaltplan (Synthese) unterliegt man gewissen Einschränkungen hinsichtlich der Art und Weise wie man einzelne Konstrukte verwendet [Syn97]. Das Synthesewerkzeug erkennt einige Elemente (beispielsweise ein asynchrones Reset[9]) nur dann richtig, wenn sie in einem festgelegten Stil beschrieben wurden. Durch diese Regeln wird der Entwickler nicht auf eine Untermenge von Verilog beschränkt, der Synthetisierer legt lediglich die Art und Weise der Verwendung fest. Die Aufgabe der Schaltungsbeschreibung ist also nicht mit einem syntaktisch richtigen Code gelöst. Vielmehr kommt es darauf an, daß das Syntheseprogramm die beschriebenen Elemente erkennt und in der beabsichtigten Weise in eine Schaltung umsetzt.

Ein Beispiel-Modul

Im folgenden soll ein Beispiel einer Verilog-Beschreibung vorgestellt werden, das einen 4-Bit Up-Down-Zähler mit asynchronem Reset beschreibt (Abb. 4.8). Einige der oben genannten Sprachelemente sollen hier konkret aufgezeigt werden:

Der Code beginnt mit der Deklaration eines *Moduls* mit dem Namen `Cntr4udar` und den Ein- bzw. Ausgängen `notReset`, `UpDown`, `Clk`, `CntEN` und `CntrOut`. Ein Modul ist das Basiselement in Verilog. Es repräsentiert kleine oder große Hardware-Komponenten wie AND-Gatter, Zähler, eine CPU oder ganze Rechnernetze. Ein Modul kann aus verschiedenen oder gleichen Untermodulen hierarchisch aufgebaut sein. Dadurch lassen sich sehr komplexe Systeme aufbauen. Daran schließt sich der erste `always`-Block an, der das asynchrone Reset beschreibt. Bei jeder Änderung des `notReset`-Signals wird geprüft, ob eine logische 1 oder 0 anliegt und der entsprechende Zweig der `if-else`-Anweisung abgearbeitet. Die Art und Weise, wie dieser Block beschrieben ist, wird im *Synergy HDL, Synthesizer and Optimizer Style Guide* [Syn97] festgelegt. Nur in diesem Stil erkennt der Synthesizer, daß ein asynchrones Reset beschrieben wird. Bei einer Verilog-Beschreibung ohne nachfolgende Synthese kann dieser Block auch anders beschrieben werden. Der zweite `always`-Block beschreibt die In- bzw. Dekrementierung des Zählerstandes `CntrOut` mit jeder steigenden Flanke des externen

[8] Bei sequentiellen Programmiersprachen wird die Reihenfolge der Ausführung der Befehle eindeutig durch einen Programmzähler (program counter) festgelegt. Es existiert nur *ein* Kontrollfluß

[9] Ein *asynchrones* Reset wird unabhängig vom Zustand des Taktsignals zum Zeitpunkt der Aktivierung ausgeführt. Im Gegensatz dazu bewirkt ein *synchrones* Reset erst mit der nächsten aktiven Flanke des Taktes eine Reaktion des Systems.

```
module Cntr4udar(CntrOut, notReset, Clk, CntEN, UpDown);
//
// 4-bit counter with switchable count direction and asynchronous reset
//

  output [3:0] CntrOut;   //Counter Output
  input        notReset;  //active-low reset
  input        UpDown;    //1: enforces counting up, 0: enforces counting down
  input        Clk;       //external clock signal
  input        CntEN;     //Count Enable: 1 enables count action, 0 disables it

  reg    [3:0] CntrOut;

  always @(notReset)
    if(!notReset)
      begin
        assign CntrOut = 0;
        disable cntloop;
      end
    else
      deassign CntrOut;

  always @(posedge Clk)
    begin: cntloop
      if(!notReset)
        disable cntloop;
      else
        if(CntEN)
          if(UpDown)
            CntrOut = CntrOut + 1;
          else
            CntrOut = CntrOut - 1;
    end

endmodule
```

Abbildung 4.8: Verilog-Code eines 4-bit Up-Down Zählers mit asynchronem Reset.

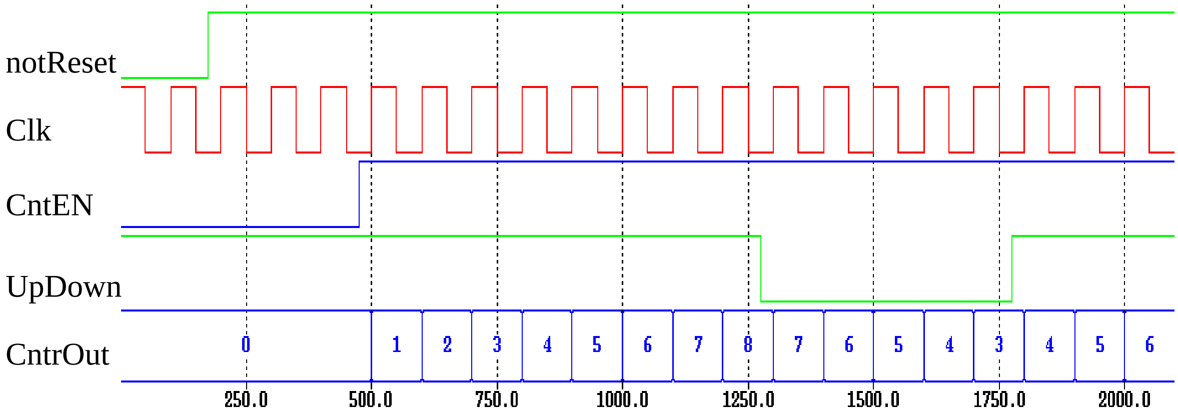

Abbildung 4.9: Simulationsergebnis des 4-bit Zählers. Die Zahlen geben die Simulationszeit in Nanosekunden an.

Taktes in Abhängigkeit der Steuersignale `CntEN` und `UpDown`. Die beiden `always`-Blöcke unterscheiden sich grundlegend voneinander. Der erste beschreibt eine *asynchrone* Logik, die bei jeder Änderung von notReset (`@notReset`) arbeitet, der zweite eine *synchrone* Logik, die mit der steigenden Flanke des Taktes (`@posedge Clk`) aktiviert wird.

Mit dem Simulator und Verilog-Interpreter *Verilog-XL* läßt sich dieses Modul simulieren. Die Simulation verläuft ereignisgesteuert und es gehen noch keine Informationen über das Zeitverhalten ein. Auf dieser Basis kann die Funktion und logische Korrektheit des Designs überprüft werden. Aussagen über das Zeitverhalten, wie zum Beispiel Gatter-Laufzeiten und damit verbundene maximale Taktraten sind erst nach einer Hardware-Implementation möglich. In Abbildung 4.9 ist das Timing-Diagramm für den Test des Moduls `Cntr4udar` dargestellt.

4.3.2 Synthese

Unter Synthese versteht man die Generierung eines Schaltplanes aus der abstrakten Beschreibung einer Schaltung. Im ASIC-Labor stehen hierfür die zwei Programme *Synergy* und *Synopsis* zur Verfügung. Für die Digitalentwicklungen am CIPix wurde *Synergy* verwendet, welches zu *Cadence* gehört. Die für den Synthese-Schritt verwendeten Schaltelemente (Standardzellen), werden aus einer Bibliothek entnommen, die spezifisch für den Zielprozess des Designs ist. Die Schaltplangenerierung erfolgt weitgehend automatisch. Um jedoch zu zufriedenstellenden Ergebnissen zu gelangen, müssen der Synthese-Software einige Vorgaben gemacht werden. Abb. 4.10 zeigt das Ergebnis der Synthese. Man erkennt vier Flip-Flops (3 D-Flip-Flops vom Typ `DFAQ` und ein J-K-Flip-Flop vom Typ `JKA`), die den Zählerstand speichern und den *sequentiellen* Teil der Logik darstellen, sowie Elemente *kombinatorischer* Logik, die die Steuersignale umsetzen.

4.3.3 Layout-Generierung

In den Standardzellenbibliotheken zu einem Zielprozeß sind die Layout-Ansichten der einzelnen Gatter bereits enthalten. Bei der Layout-Generierung werden diese Elemente plaziert und untereinander verbunden (place and route). Dies geschieht wieder weitgehend automatisch.

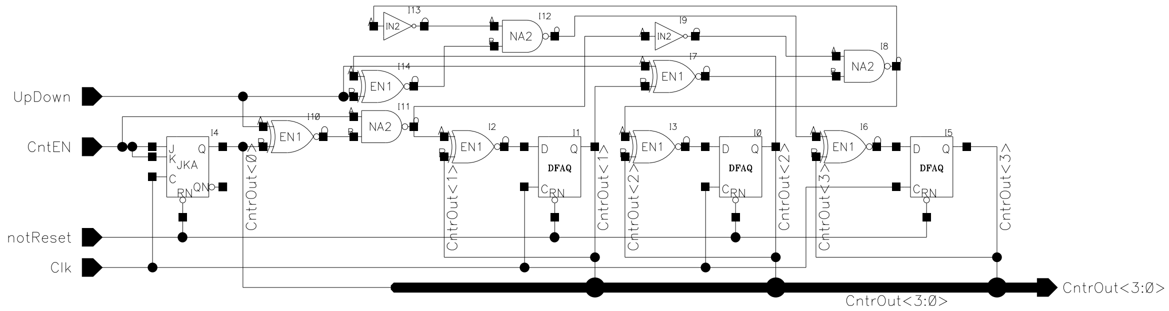

Abbildung 4.10: Synthese-Ergebnis der Verilog-Beschreibung des Zählers.

Der Entwickler kann u. a. das Verhältnis der Seitenmaße des Logikblocks, die Konfiguration der Anschlüsse sowie die Art und Weise der Spannungszuführung festlegen.

Das Layout hat weitere Auswirkungen auf das Zeitverhalten einer Schaltung. Durch die Leiterbahnführung entstehen parasitäre Lastkapazitäten. Diese führen dazu, daß sich die Anstiegs- und Abfallzeiten der Signale verlangsamen. Besonders bei einer Taktleitung, die einige tausend Gatter treibt, spielt dieser Effekt eine Rolle. Um die kapazitive Last der Clockleitung zu reduzieren, baut man das Leitungsnetz in einer baumartigen Struktur (clock tree) auf und versieht die einzelnen Äste mit Treiberstufen (Abb. 4.11). Der Clock-Tree sollte symmetrisch und die Treiber bis in die gleiche Tiefe implementiert und wenig belastet sein.

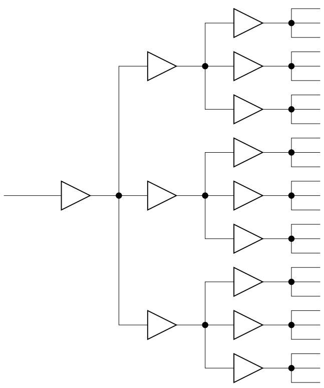

Abbildung 4.11: Schema eines Clock-Trees [AMS]. Der Aufbau ist völlig symmetrisch. Die Treiberstufen sind bis in die gleiche Tiefe implementiert und wenig belastet.

In Abb. 4.12 ist das Ergebnis der Layout-Generierung des 4-Bit Zählers dargestellt.

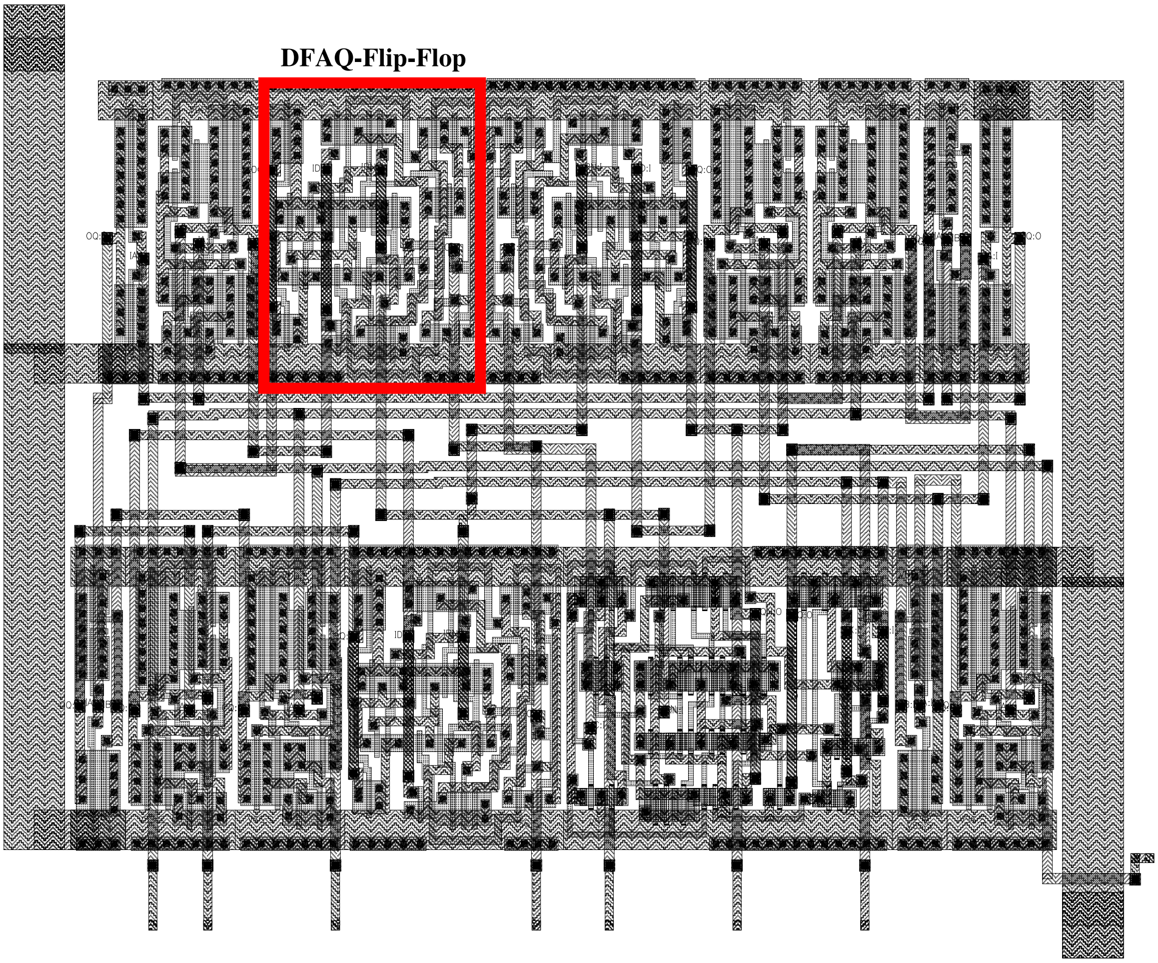

Abbildung 4.12: Ergebnis der Layout-Generierung des 4-bit Zählers. Der Rahmen kennzeichnet den Bereich eines Flip-Flops vom Typ *DFAQ*.

Kapitel 5

Der Digitalteil des CIPix

Die Entwicklung der digitalen Komponenten des CIPix war die wesentliche Aufgabe dieser Diplomarbeit. Deshalb sollen im folgenden Kapitel diese Teile des Chips vertiefend dargestellt werden. Neben der Beschreibung des Aufbaus werden Simulationsergebnisse präsentiert und bei der Entwicklung aufgetretene Probleme genannt.

5.1 Das I^{2}C-Interface

Das neu entwickelte I^{2}C-Interface des CIPix dient als Schnittstelle zwischen zwei Datenformaten. Die seriell auf dem externen I^{2}C-Bus [I^{2}C98] übertragenen Daten werden durch die Schnittstelle auf interne, parallele Daten- und Adressbusse umgesetzt. Umgekehrt werden Daten aus dem CIPix in das serielle I^{2}C-Format konvertiert. Im folgenden Abschnitt wird zunächst die Struktur und das Protokoll des I^{2}C-Busses vorgestellt bevor auf den Aufbau und die Funktionsweise des eigentlichen Interfaces eingegangen wird.

5.1.1 Der I^{2}C-Bus

Der I^{2}C (Inter-IC)-Bus wurde in den achtziger Jahren von *Philips* im Bereich der Unterhaltungselektronik entwickelt, um periphäre ICs (Intergrated circuits) mit Microcontrollern zu verbinden. Dabei stand die Einfachheit und Kostengünstigkeit der Busarchitektur im Vordergrund. Der I^{2}C-Bus erlaubt es, eine Vielzahl von periphären Geräten, wie Speicherbausteine, I/O[1]-Ports oder LCD[2]-Treiber an einen oder mehrere Microcontroller anzuschließen. Ursprünglich fand der I^{2}C-Bus Anwendung in Systemen wie Fernsehern, Waschmaschinen und Mobiltelefonen. Heutezutage hat sich der Bus zu einem Standard entwickelt und wird in nahezu jeder Heimelektronik sowie in vielen kommerziellen und industriellen Systemen verwendet.

Das Bus-Konzept

Der I^{2}C-Bus ist ein synchrones, serielles System mit zwei bidirektionalen[3] Leitungen, einer Datenleitung (SDA: Serial DAta) und einer Clockleitung (SCL: Serial CLock). Informationen werden synchron zum Takt SCL übertragen. Jeder Busteilnehmer wird durch eine eindeutige

[1]Input/Output
[2]Liquid cristal display
[3]Daten können in beiden Richtungen übertragen werden.

Adresse angesprochen. Ein Busteilnehmer kann entweder als Sender (engl.: transmitter) oder Empfänger (engl.: receiver) arbeiten. Zudem kann er als Master oder Slave angesehen werden (siehe Tabelle 5.1). Ein Master ist der Busteilnehmer, der den Datentransfer auf dem Bus initialisiert und beendet und den Takt generiert, zu dem die Daten synchron übertragen werden. Gleichzeitig wird jeder andere adressierte Busteilnehmer als Slave angesehen.

Terminus	Beschreibung
Transmitter	Sendet Daten an den Bus
Receiver	Empfängt Daten vom Bus
Master	Teilnehmer, der den Transfer initiiert, den Takt generiert und den Transfer beendet
Slave	Das vom Master adressierte Gerät
Multi-Master	Mehr als ein Master kann zu einem Zeitpunkt versuchen den Bus zu kontrollieren, ohne einen Buskonflikt hervorzurufen.
Arbitrierung	Prozedur, die sicherstellt, daß bei einem Zugriff mehrerer Master auf den Bus, nur *einer* den Zugang erhält.
Synchronisation	Prozedur, die die Taktsignale zweier oder mehrerer Geräte synchronisiert.

Tabelle 5.1: Definition der I^2C-Terminologie

Der I^2C-Bus ist ein Multi-Master Bus. Das bedeutet, daß mehr als ein Busteilnehmer in der Lage sein kann, den Bus zu kontrollieren. Um bei gleichzeitigem Zugriff mehrerer Master auf den Bus einen eindeutigen Zugriff zu erhalten, wurde eine *Arbitrierungsprozedur* entwickelt. Diese beruht auf einer wired-AND Verbindung aller I^2C-Interface mit dem Bus. Um diese wired-AND Funktion zu erfüllen, besitzt die Ausgangsstufe jedes I^2C-Interfaces einen open-drain oder open-collector Ausgang. Ein open-drain Ausgang besteht lediglich aus einem unipolaren Transitor, bei dem der Source-Anschluß auf Masse liegt. Solche Ausgänge kann man ohne weiteres parallel schalten und mit einem gemeinsamen Widerstand versehen (Abb. 5.1).

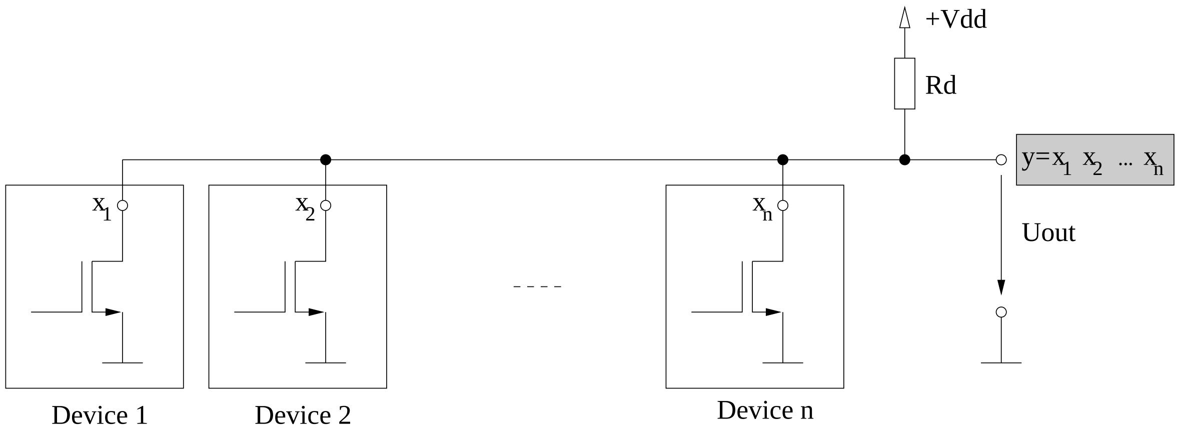

Abbildung 5.1: Schema einer wired-AND Schaltung.

Das Ausgangspotential geht nur dann in den High-Zustand, wenn alle Ausgänge im High-Zustand sind. In positiver Logik ergibt sich demnach eine UND-Verknüpfung. Abb. 5.2 zeigt schematisch die Verschaltung der Ausgangsstufen mit den beiden bidirektionalen Busleitungen SDA und SCL. Diese sind über einen Pullup-Widerstand mit einer positiven Spannung verbunden. Ist der Bus frei, sind beide Leitungen auf High, das heißt auf logisch Eins. Wenn nun zwei oder mehr Master versuchen, auf den Bus zuzugreifen, verliert derjenige die Arbitrierung, der zuerst eine logische Eins produziert, während der andere eine logische Null erzeugt. Das Clock-Signal während des Arbitrierungsvorgangs ist eine synchronisierte Kombination der Clock-Signale der einzelnen Master, die auf der wired-AND Verbindung der SCL-Leitungen beruht.

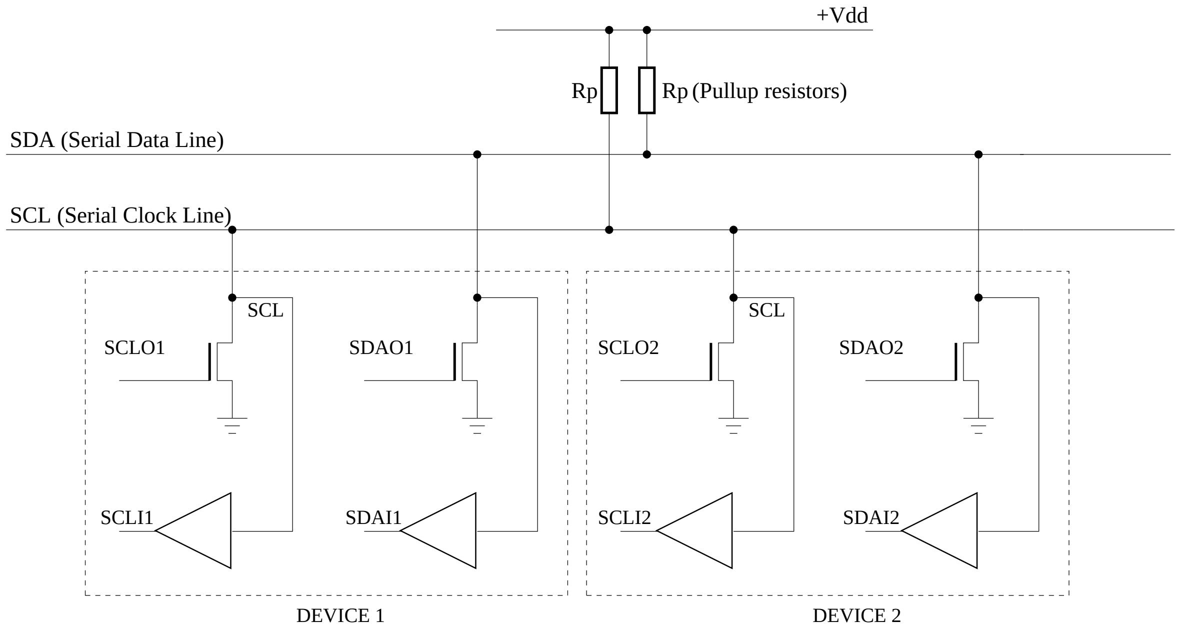

Abbildung 5.2: Blockschaltbild der I^{2}C-Ausgangsstufen [I^{2}C98].

Der Bus kann in zwei Modi betrieben werden:

- *Standard*-Modus mit Transferraten bis 100kbit/s.
- *Fast*-Modus mit Transferraten bis 400kbit/s: Dieser Modus stellt eine Erweiterung des ursprünglichen Standard-Modus dar. Infolge der höheren Überragungsrate müssen unter anderem die Ein- und Ausgangsstufen, sowie die Pullup-Widerstände modifiziert werden ([I^{2}C98, Seite 16]).

Die Zahl der an einen Bus angeschlossenen Teilnehmer wird theoretisch durch die verfügbaren Busadressen begrenzt. In der Praxis limitiert jedoch die Beschränkung der Buskapazität auf 400pF die Anzahl von Geräten. Ein reales System wird typischerweise aus einigen zehn Teilnehmern pro Bus bestehen. An der zukünftigen Kammer ist es geplant, für jede Zylinderlage einen eigenen I^{2}C-Bus zu implementieren. Für 1920 (=120·16) Kanäle pro Lage würden dann 30 (=1920/64) CIPix-Chips an den Bus angeschlossen.

Der Datentransfer

Beim Datentransfer wird für jedes übermittelte Datenbit ein Clockpuls generiert. Daten sind nur gültig, wenn sie während der High-Phase der Clock stabil sind. Die Zustände der Datenleitung SDA können sich nur ändern, wenn das Clocksignal auf SCL Low ist (siehe Abb. 5.3). Über diese Validitätsbedingung für die Daten lassen sich nun eindeutige Start- und Stopbedingung für den Transfer definieren (Abb. 5.4):

- *Start*-Bedingung: Während der High-Phase von SCL erfolgt ein High-Low Übergang auf der Datenleitung SDA.
- *Stop*-Bedingung: Während der High-Phase von SCL erfolgt ein Low-High Übergang auf der Datenleitung SDA.

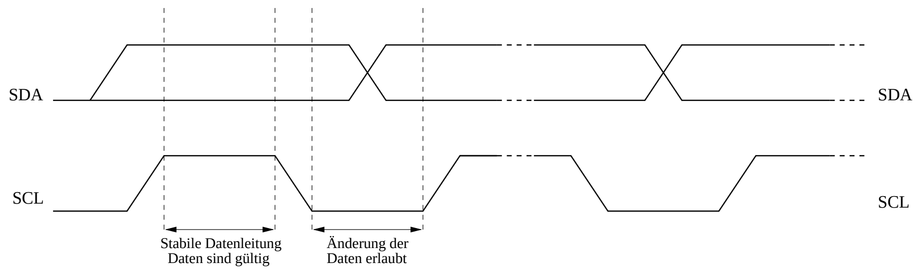

Abbildung 5.3: Schema des bitweisen Transfers auf dem I^2C-Bus [I^2C98].

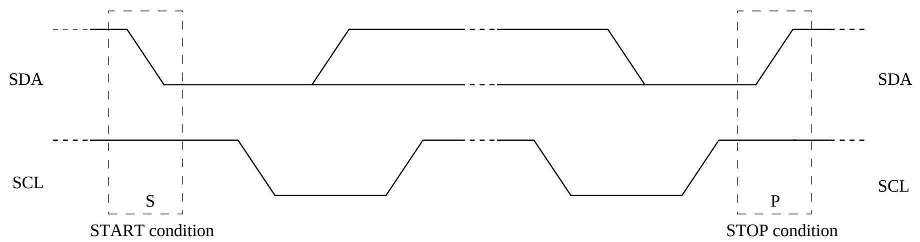

Abbildung 5.4: Start- und Stop-Bedingungen auf dem I^2C-Bus [I^2C98].

Start- und *Stop*-Bedingungen werden immer vom Master generiert. Nach einer *Start*-Bedingung wird der Bus als belegt angesehen, nach einer gewissen Zeit ($4,7\mu s$ im Standard-Modus) nach der *Stop*-Bedingung als frei. Daten werden mit dem most significant bit (MSB) zuerst übermittelt. Jedes Datenbyte, das auf die SDA-Leitung gelegt wird, muß 8 Bits lang sein und von einem Acknowledge[4]-Bit gefolgt werden. Der Datentransfer mit Acknowledge ist obligatorisch (siehe Abb. 5.5): Der für das Acknowledge erforderliche Clockpuls wird vom Master generiert. Während dieses Clockpulses setzt der Sender (transmitter) die SDA-Leitung auf High. Um nun ein *Acknowledge* zu erzeugen setzt der Empfänger (receiver) die Datenleitung auf Low, so daß sie während der High-Phase des Acknowledge-Clockpulses stabil auf Low bleibt. Um ein *notAcknowledge* zu erzeugen, setzt der Empfänger die Datenleitung

[4]=(Empfangs)Bestätigung

auf High. Ist der Master gleichzeitig Empfänger (master-receiver) signalisiert er das Ende der Datenübertragung, indem er nach dem letzten Datenbyte, das vom Slave gesandt wurde, ein *notAcknowledge* generiert. Der Slave-Sender muß dann die Datenleitung auf High setzten, um es dem Master zu ermöglichen eine *Stop*- oder *repeated Start*[5]-Bedingung zu generieren.

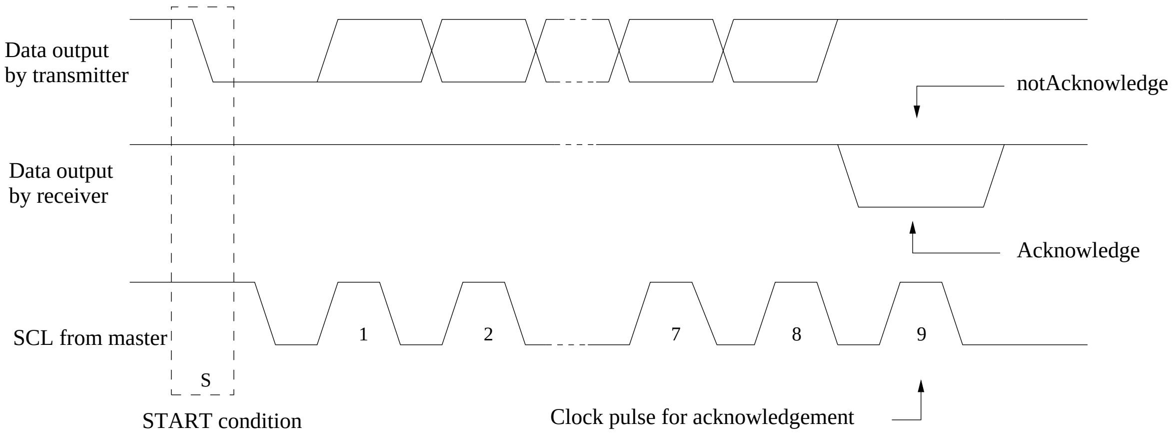

Abbildung 5.5: Acknowledge-Signale auf dem Bus [I²C98].

Adressierung

Das I²C-Protokoll definiert verschiedene Arten der Adressierung. Der Adressierungsmodus wird durch die ersten 7 Bits des ersten Bytes nach einer *Start*-Bedingung bestimmt. In Tabelle 5.2 sind die verschiedenen Modi mit ihrer zugehörigen Bitkennung zusammengestellt.

Bitfolge	Richtungsbit $R/\overline{W}$	Beschreibung
0000 000	0	General Call: alle Busteilnehmer werden simultan adressiert
0000 xxx	x	reserviert für andere oder zukünftige Zwecke
1111 1xx	x	reserviert für andere oder zukünftige Zwecke
1111 0xx	x	10-bit Adressierung: xx steht für die beiden MSBs der 10-bit Adresse. Die restlichen 8 Bit werden im nachfolgenden Byte übertragen.
andere	x	7-bit Adressierung

Tabelle 5.2: Definition des ersten Bytes nach einer *Start*-Bedingung [I²C98].

Das LSB (least significant bit) des ersten Bytes ist das Richtungsbit ($R/\overline{W}$), das die Richtung der zu transferrierenden Daten festlegt. Eine 0 leitet eine Schreib (Write)-, eine 1 eine Leseoperation (Read) ein. In Abb. 5.6 ist ein kompletter Datentransfer mit einer 7-Bit Adressierung gezeigt. Der Transfer wird immer durch eine *Start*-Bedingung (S) initialisert und mit einer *Stop*-Bedingung (P) beendet. Möchte ein Master nach einem Transfer mit einem

[5]Eine *repeated Start*-Bedingung folgt direkt auf eine *Start*-Bedingung, ohne daß zuvor eine *Stop*-Bedingung generiert wurde.

Slave weiter auf den Bus zugreifen, kann er eine *repeated Start*-Bedingung (Sr) generieren und einen anderen Slave adressieren, ohne zuvor eine *Stop*-Bedingung erzeugt zu haben.

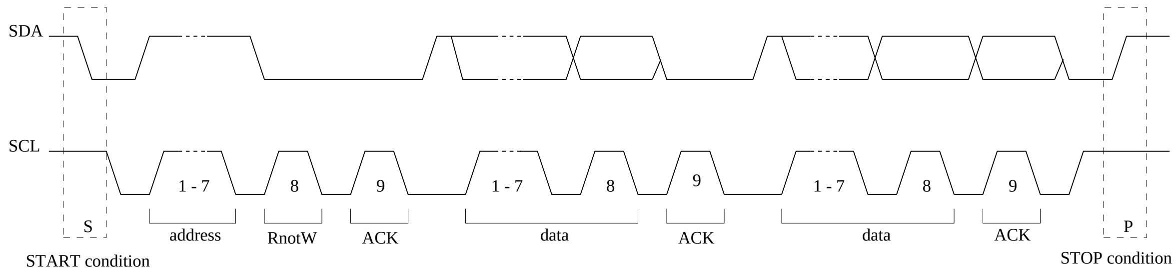

Abbildung 5.6: Schema eines kompletten Datentransfers [I^2C98].

Im folgenden sollen nun, zunächst für den Fall der 7-Bit Adressierung, mögliche Übertragungssequenzen aufgezeigt werden:

- Ein Master-Transmitter adressiert einen Slave-Receiver, wobei die Übertragungsrichtung während des Transfers nicht geändert wird (Abb. 5.7 (a)).
- Ein Master-Receiver liest einen Slave-Transmitter aus (Abb. 5.7 (b)).
- Kombiniertes Format: Während der Sequenz ändert sich die Übertragungsrichtung und damit auch die Terminologie der Teilnehmer (Abb. 5.7 (c)).

Wie aus der Tabelle 5.2 ersichtlich ist, sind zwei Gruppen von je 8 Adressen (0000xxx und 1111xxx) reserviert und stehen im 7-Bit Adressraum nicht zur Verfügung. Dieser umfaßt also nur $2^7 - 2 \cdot 8 = 112$ Adressen. Durch den *10-Bit Adressierungsmodus* wird nun der verfügbare Adressraum um 1024 ($= 2^{10}$) Adressen vergrößert. Die 10-Bit Adressierung beeinflußt die 7-Bit Adressierung jedoch in keiner Weise. Geräte mit 7-Bit und 10-Bit Adressen können am gleichen Bus betrieben werden. Die 10-Bit Slave-Adresse wird durch die ersten beiden Bytes nach einer *Start*- oder *repeated Start*-Bedingung gebildet (siehe Abb. 5.8):

Die ersten 5 Bits des ersten Bytes sind die 10-Bit Kennung (11110). Darauf folgen die beiden ersten MSBs (xx) der Adresse und das Richtungsbit R/$\overline{\mathrm{W}}$. Ist das R/$\overline{\mathrm{W}}$-Bit 0, folgen im nächsten Byte die restlichen 8 Bits der Adresse (xxxx xxxx). Ist das R/$\overline{\mathrm{W}}$-Bit 1, enthält das nächste Byte Daten, die vom Slave an den Master übermittelt werden. Es sind nun verschiedene Kombinationen von Schreib-/Lese-Formaten möglich. Die beiden grundlegenden Übertragungssequenzen sollen im folgenden erläutert werden:

- Ein Master-Transmitter übermittelt Daten an einen Slave-Receiver. Die Transferrichtung ändert sich hierbei nicht. (Abb. 5.9 (a)):
  Wird nach einer *Start*-Bedingung die 10-Bit Kennung (11110) gesandt, vergleicht jeder Slave die Bits 2 und 1 (xx) mit seiner eigenen Adresse und testet, ob das Richtungsbit R/$\overline{W}$ (Bit 0) 0 ist. Bei einer Übereinstimmung wird ein *Acknowledge* (A1) generiert. Es ist möglich, daß dies mehr als ein Slave tut. All diejenigen, die eine Übereinstimmung gefunden haben vergleichen das zweite Byte der gesendeten Adressen (xxxx xxxx) mit ihrer eigenen. Jetzt wird nur noch ein Slave eine Übereinstimmung finden, ein *Acknowledge* (A2) generieren und die nachfolgenden Daten empfangen. Der Slave bleibt adressiert bis er eine *Stop*- oder *repeated Start*-Bedingung mit anderer Adresse empfängt.

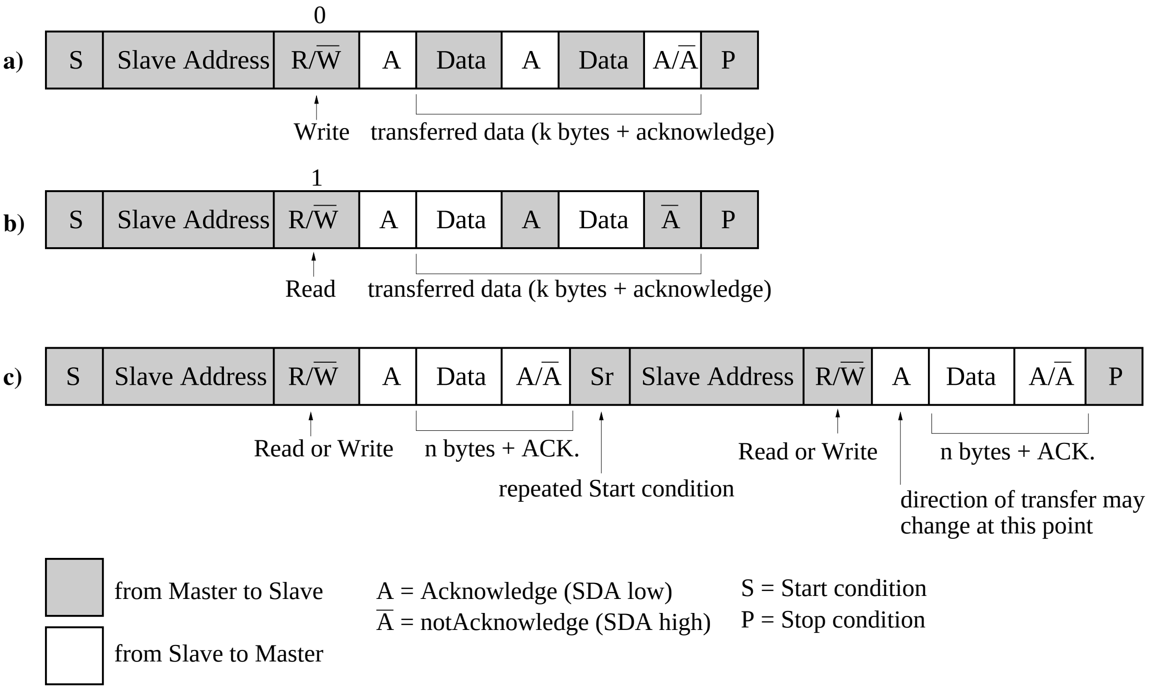

Abbildung 5.7: 7-bit Übertragungssequenzen: a) Schreibsequenz b) Lesesequenz c) Kombinierte Sequenz mit einer wiederholten *Start*-Bedingung [I^{2}C98].

S	1	1	1	1	0	X	X	R/$\overline{\text{W}}$	A	X	X	X	X	X	X	X	X	A

Abbildung 5.8: Format der 10-bit Adresse. Grau hinterlegt sind die Bits der Adresse [I^{2}C98].

- Ein Master-Receiver liest einen Slave-Transmitter aus. Die Richtung der transferrierten Daten wird hierbei geändert (Abb. 5.9 b)):
  Bis einschließlich dem zweiten Acknowledge (A2) ist die Prozedur mit der zuvor beschriebenen Schreibsequenz identisch. An das Acknowledge A2 schließt sich eine *repeated Start*-Bedingung an (Sr). Der Slave der eine vollkommene Übereinstimmung seiner Adresse fand, erinnert sich nun daran, daß er zuvor adressiert wurde. Nach der erneuten 10-Bit Kennung überprüft er, ob die Bits 2 und 1 (xx) mit denen nach der *Start*-Bedingung übereinstimmen und testet, ob das Richtungsbit R/$\overline{W}$ (Bit 0) 1 ist. Ist das der Fall, betrachtet sich der Slave als adressierten Sender und generiert das Acknowledge A3. Der Slave-Sender bleibt adressiert bis er eine *Stop*-Bedingung oder eine weitere *repeated Start*-Bedingung mit anderer Adresse empfängt.

Bei den bisher vorgestellten Adressierungsmodi wird immer nur ein bestimmter Slave adressiert und nimmt am Datentransfer teil. Das I^{2}C-Protokoll definiert auch einen Modus, den sogenannten *General Call*, bei dem alle mit dem Bus verbundenen Teilnehmer simultan angesprochen werden. In diesem Modus ist natürlich nur ein Schreibzugriff des Masters möglich. Eine Leseoperation von mehreren Slaves würde zu einem Buskonflikt führen. Der *General Call* wird durch die Bitfolge 0000 0000 nach der *Start*-Bedingung eingeleitet und

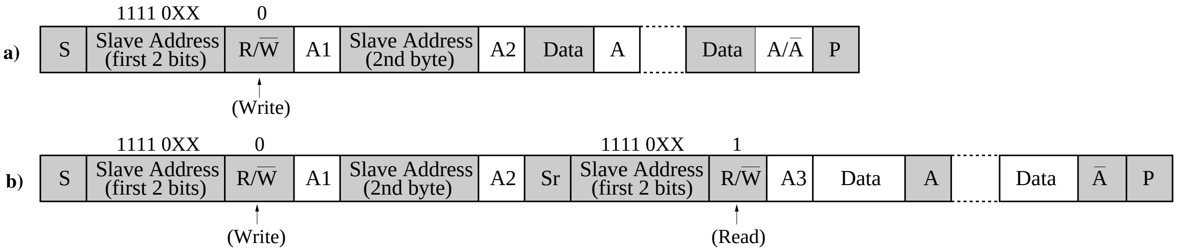

Abbildung 5.9: 10-bit Übertragungssequenzen: a) Schreibsequenz b) Lesesequenz [I^{2}C98].

umfaßt selbst mehrere Untermodi, die im nachfolgenden Byte spezifiziert werden. Das Format des *General Call* ist in Abb. 5.10 dargestellt. Hier soll nur einer der Untermodi erklärt werden. Die anderen sind in [I^{2}C98] nachzulesen: Ist das LSB des nachfolgenden Bytes 1, handelt es sich um einen sogenannten *Hardware-General Call*. Abb. 5.11 zeigt die entsprechende Übertragungssequenz. Die verbleibenden sieben Bits des zweiten Bytes enthalten die Adresse des Masters.

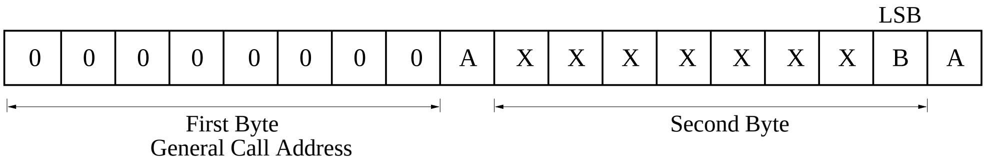

Abbildung 5.10: Format einer *General Call* Übertragung [I^{2}C98].

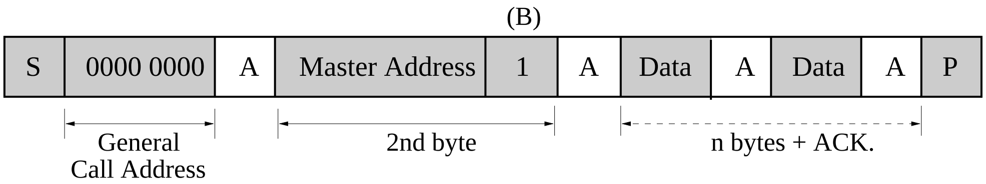

Abbildung 5.11: Übertragungssequenz eines *Hardware-General Call* [I^{2}C98].

Transfer-Unterbrechung

Das I^{2}C-Protokoll sieht die Möglichkeit vor, daß ein adressierter Slave während eines Transfers diesen unterbricht. Hierzu zieht der Slave die Clockleitung SCL herunter (setzt sie auf Low) und zwingt so den Master in einen Wartezustand. Beispielsweise kann der Empfänger in dieser Zeit einen internen Interrupt[6] verarbeiten. Der Datentransfer wird fortgesetzt, wenn der Receiver die Clockleitung wieder losläßt (auf High setzt). Das Herunterziehen der Clockleitung darf jedoch nur nach einem vollständig übermittelten Datenbyte erfolgen. In Abbildung 5.12 ist ein solcher Transfer dargestellt.

[6] Unterbrechung einer Prozessorroutine durch Hard- oder Softwarezugriff.

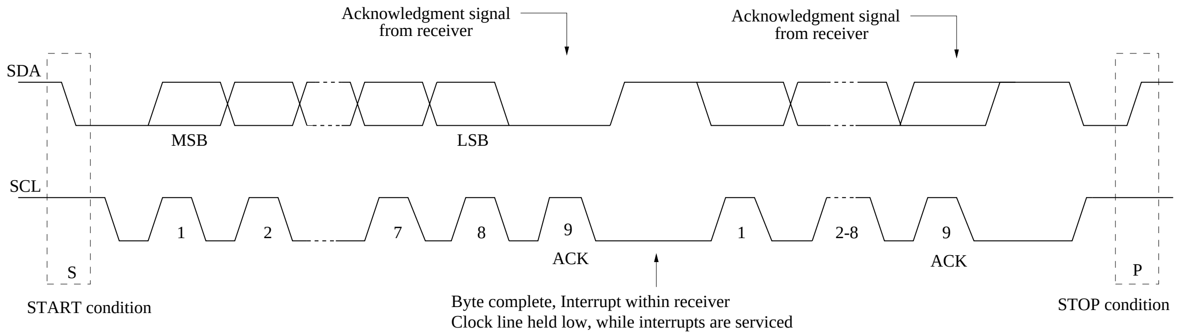

Abbildung 5.12: Erzwingung eines Wartezustands auf dem Bus [I²C98].

5.1.2 Das Interface

Die für den CIPix entwickelte Schnittstelle ist ein reines *Slave-I²C-Interface*. Das bedeutet, daß sie nur in der Lage ist, auf einen extern initialisierten Transfer zu reagieren und eine Schreib- oder Leseoperation auf dem Bus auszuführen. Eine Clock-Generierung verbunden mit einer Synchronisationsprozedur auf dem Bus, sowie eine Arbitrierung mit anderen zugreifenden Mastern wird nicht unterstützt. Das Interface ist für einen Betrieb im *Standard-Modus* mit maximalen Transferraten von 100 kBit/s ausgelegt.

Abb. 5.13 zeigt das Toplevel-Symbol des I²C-Interfaces. In Tabelle 5.3 sind die 10 Anschlüsse zusammengestellt und erklärt[7]. Die bidirektionalen I²C-Busleitungen SDA und SCL werden intern in jeweils zwei unidirektionale Signale aufgespalten ((SDAI, SDAO) und (SCLI, SCLO))[8]. Den Ein- und Ausgängen der Datenleitung SDA sind zusätzlich Verzögerungselemente vor- bzw. nachgeschaltet, die gewährleisten, daß Setup- und Holdzeiten auf dem Bus eingehalten werden (s. u.). Die Ausgangsleitungen SCLO und SDAO werden auf Tristate-Gatter geführt, um einen konfliktfreien Zugriff auf den Bus sicherzustellen (s. u.).

Das Interface ist in 6 Unterblöcke aufgeteilt (Abb. 5.14), wobei jedem Block ein eigenes Modul im Verilog-Code entspricht. Im folgenden werden die einzelnen Unterblöcke beschrieben. In Tabelle 5.4 werden alle *internen* Signale aufgelistet und erklärt.

Die Schieberegister

Die I²C-Schnittstelle besitzt zwei unterschiedliche Typen von Schieberegistern, die die Umwandlung zwischen seriellen und parallelen Daten übernehmen: ein Seriell-Parallel-Schieberegister (SIPO: Serial In Parallel Out) zur Umsetzung der Adressen und ein Seriell-Parallel-/Parallel Seriell-Schieberegister (SIPO/PISO: Serial In Parallel Out/ Parallel In Serial Out) zur Konvertierung der Daten.

Das Adress-Schieberegister (Modul *AddrShiftReg*) ist 8 Bit breit und erlaubt die Umwandlung der Daten in einer Richtung. Durch ein Reset-Signal (`notReset` geht auf 0) wird der Registerinhalt auf 0 gesetzt. Liegt das Enable-Signal `ASE` an, werden die an SDA anliegenden Daten mit der *steigenden* Flanke der Clock SCL in das Register geschoben. Nach acht Zyklen steht das 8-Bit breite Adresswort an den parallelen Ausgängen zur Verfügung.

[7]Ein Signal negativer Logik trägt das Präfix `not`. Signale ohne diesen Vorsatz gehorchen positiver Logik.
[8]I=Input, O=Output

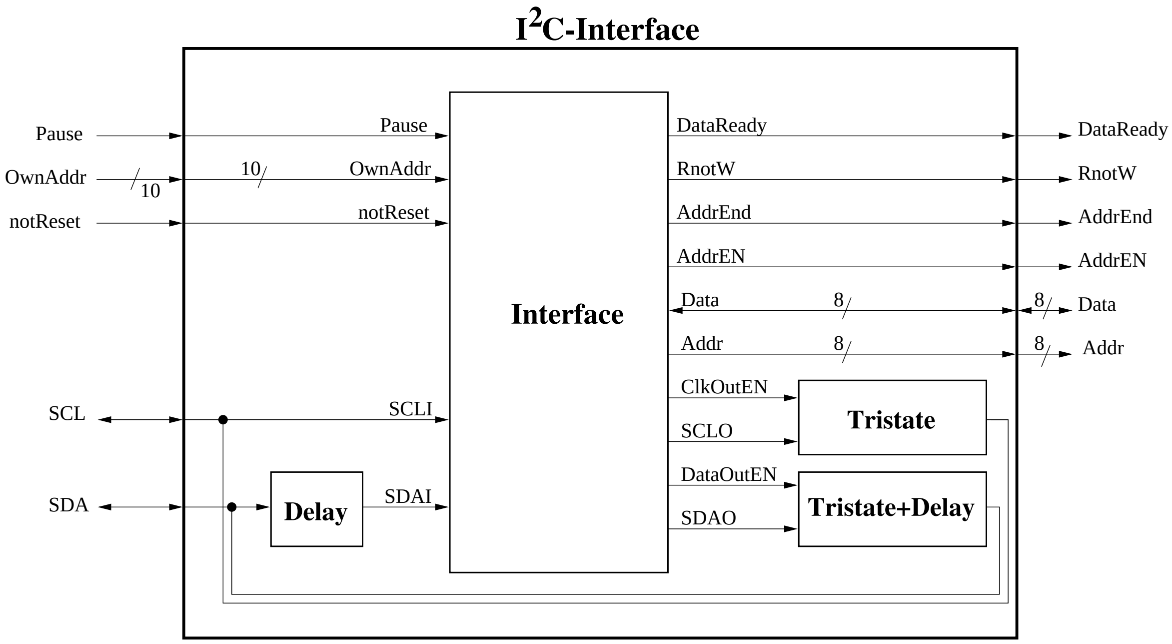

Abbildung 5.13: Blockdiagramm der obersten Ebene des I^{2}C-Interfaces.

Signal	Typ	Beschreibung
`SDA`	Ein-/Ausgang	Serial DAta: bidirektionale, serielle I^{2}C-Datenleitung
`SCL`	Ein-/Ausgang	Serial CLock: bidirektionale, serielle I^{2}C-Clockleitung
`notReset`	Eingang	Reset-Signal
`OwnAddr`	Eingang	Own Address: Eigene Adresse des Interfaces
`Pause`	Eingang	Pause=1 erzwingt einen Wartezustand auf dem Bus
`Addr`	Ausgang	8-Bit breiter interner Adressbus
`Data`	Ein-/Ausgang	8-Bit breiter interner Datenbus
`DataReady`	Ausgang	DataReady=1 zeigt ein komplettes 8-Bit Datenwort an
`AddrEnd`	Ausgang	Address End: AddrEnd=1 markiert eine vollständig übermittelte Adresse.
`AckEN`	Ausgang	Acknowledge Enable: AckEN=1 generiert ein *Acknowledge*, AckEN=0 ein *notAcknowledge*
`RnotW`	Ausgang	ReadnotWrite: Richtungsbit für den Transfer (1: Leseoperation, 0: Schreiboperation)

Tabelle 5.3: Externe Signale der I^{2}C-Schnittstelle.

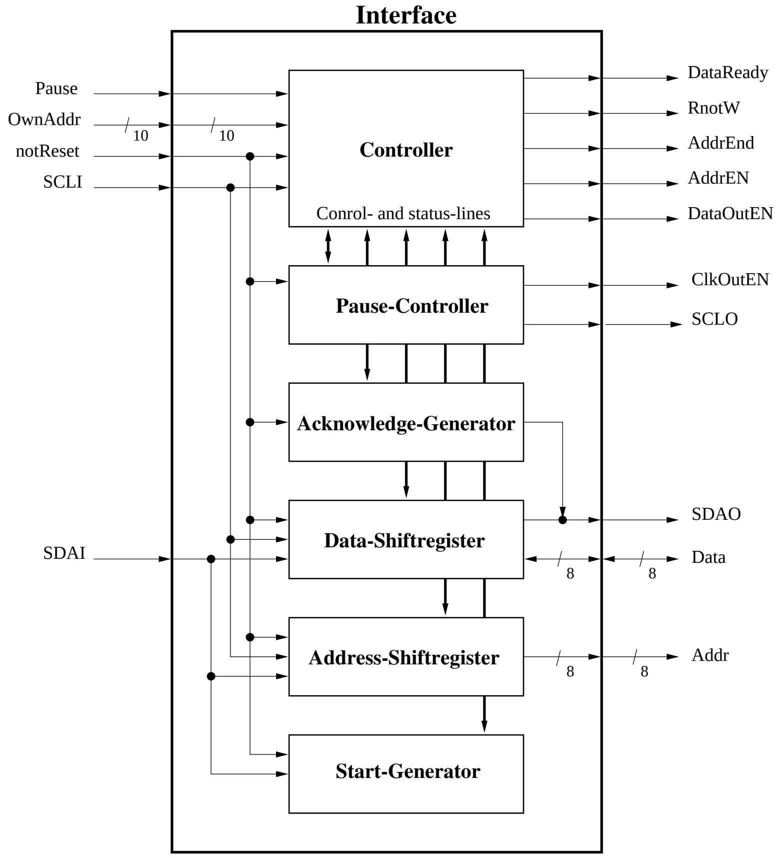

Abbildung 5.14: Blockdiagramm des Interfaces.

Das Daten-Schieberegister (Modul *DataShiftReg*) ist ebenfalls 8 Bit breit, unterstützt aber, im Gegensatz zu dem zuvor beschriebenen Adress-Schieberegister, neben der Seriell-Parallel-Wandlung auch den umgekehrten Vorgang der Parallel-Seriell-Wandlung. Dazu müssen parallel anliegende Daten simultan in das Register geladen werden. Dies ist auf zwei unterschiedliche Arten möglich: Bei einem *asynchronen* Ladevorgang werden die parallelen Daten an die Preset-Eingänge der Register-Flip-Flops gelegt. Bei einem *synchronen* Ladevorgang wird jedem D-Eingang eines Flip-Flops ein Multiplexer vorgeschaltet, der zwischen den parallel und seriell anliegenden Daten selektiert. Das Daten-Schieberegister des CIPix verwendet einen synchronen Ladevorgang der parallel anliegenden Daten. Bei Simulationen der asynchronen Variante nach der Synthese trat das Problem auf, daß Setup-Zeiten der D-Flip-Flops aus den Standardbibliotheken nicht eingehalten wurden. Typische Zeiten liegen für den Typ *DFAQ* bei t_{SU}=800 ps und für den Typ *DF8Q* bei t_{SU}=700 ps. Daraufhin wurde die synchrone Umsetzung gewählt.

Der Registerinhalt wird durch ein asynchrones Reset-Signal auf 0 initialisiert. Mit jeder *fallenden* Flanke der Clock `SCL` wird, falls das Enable-Signal `DSE` anliegt, ein Schiebevorgang ausgeführt. Die Richtung der Schiebeoperation wird durch den Wert des Richtungsbit $\mathrm{R}/\overline{\mathrm{W}}$ bestimmt. Ein paralleler Ladevorgang der Register wird synchron zum Takt ausgeführt, wenn der Adressvergleich zwischen eigener und gesendeter Adresse positiv war und $\mathrm{R}/\overline{\mathrm{W}}$ den Wert 1 hat.

Signal	Beschreibung
`AddrACK`	Address Acknowledge: `AddrACK`=1 führt zur Erzeugung eines Acknowledge-Pulses nach positivem Adressvergleich.
`ASE`	Address Shift Enable: `ASE`=1 ermöglicht einen Schiebevorgang des Adress-Schieberegisters.
`ClkOutEN`	Clock Output ENable: Steuersignal des SCL-Tristate-Buffers. `ClockOutEN`=1 legt `SCLO` auf die bidirektionale Busleitung `SCL`.
`DataOutEN`	Data Output ENable: Steuersignal des SDA-Tristate-Buffers. `DataOutEN`=1 legt `SDAO` auf die bidirektionale Busleitung `SDA`.
`DSE`	Data Shift Enable: `DSE`=1 ermöglicht einen Schiebevorgang des Daten-Schieberegisters.
`PauseEN`	Pause ENable: `PauseEN`=1 ermöglicht die Generierung eines Wartezustands auf dem Bus.
`Start`	`Start`=1 zeigt eine Start-Bedingung auf dem i2c-Bus an.
`StartACK`	Start ACKnowledge: `StartACK`=1 setzt `Start` wieder auf 0 zurück (siehe Handshake-Prozedur).

Tabelle 5.4: Interne Signale der I^2C-Schnittstelle.

Die Start- und Acknowledge-Generatoren

Im Modul *StartGen* wird eine *Start*-Bedingung auf dem I^2C-Bus detektiert und ein entsprechendes Signal (`Start`) generiert. Durch ein Reset-Signal wird `Start` auf 0 initialisiert.

Ein entsprechendes Modul, das eine *Stop*-Bedingung auf dem Bus detektiert, existert nicht. Das hat folgenden Grund: Nach einer *Stop*-Bedingung wird die SCL-Leitung dauerhaft auf logisch 1 gesetzt. Die State-machine *Ctrl* ist aber bei nicht existierendem Takt nicht funktionsfähig. Ein Statussignal, das eine *Stop*-Bedingung auf dem Bus anzeigt ist dann sinnlos, weil dadurch keine neuen Anweisungen ausgeführt werden können. Um nun aber das Interface auf eine erneute *Start*-Bedingung auf dem Bus sensibel zu machen, muß das `Start`-Signal auf logisch 0 zurückgesetzt werden. Dies geschieht mit einer *Handshake-Prozedur* (Abb. 5.15): Mit der ersten fallenden Taktflanke wird `StartACK` auf logisch 1 gesetzt. Das bewirkt, daß `Start` wieder den Anfangswert 0 annimmt. Mit der nächsten fallenden SCL-Flanke wird `StartACK` auf 0 zurückgesetzt.

Das Modul *Acknowledge* generiert die für einen korrekten Transfer notwendigen Acknowledge-Bits. Dabei wird zwischen einem Adress-Acknowledge und einem Daten-Acknowledge unterschieden: Nach einem positiven Adressvergleich (gesendete und eigene Adresse stimmen überein) wird ein *Acknowledge* generiert, ansonsten ein *notAcknowledge*. Bei einem Daten-Acknowledge ist der Wert des Richtungsbits $R/\overline{W}$ entscheident. Während einer Schreibsequenz eines Masters ($R/\overline{W} = 0$) werden nur *Acknowledge*-Pulse generiert. Bei einem Lesezyklus eines Masters ($R/\overline{W} = 1$) ist der Acknowledge-Puls nach dem letzten übertragenen Datenwort ein *notAcknowledge*.

Die Pause-Kontrolleinheit

Ein Slave kann einen Wartezustand auf dem I^2C-Bus erzwingen, um beispielsweise in dieser Zeit eine interne Interrupt-Anforderung zu verarbeiten. Um diese Situation herbeizuführen,

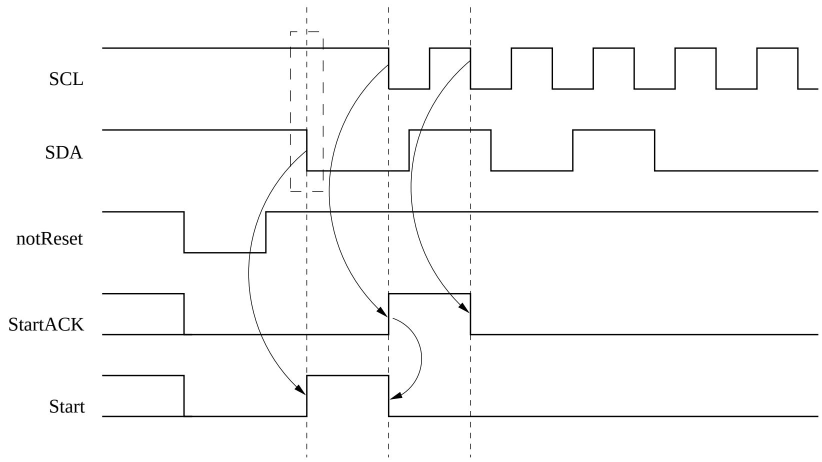

Abbildung 5.15: Handshake-Prozedur nach einer *Start*-Bedingung.

wurde das externe Signal `Pause` eingeführt. Ist `Pause`=1 soll zum nächstmöglichen Zeitpunkt die Clockleitung auf 0 gesetzt werden. In den Spezifikationen zum I²C-Bus findet sich die Einschränkung, daß dies nur nach einem vollständig übermittelten Datenbyte erfolgen kann. Um dies sicherzustellen, erzeugt der Controller, wenn er sich im richtigen Zustand befindet, ein weiteres Signal (`PauseEN`). `Pause` und `PauseEN` werden im Modul *PauseCtrl* durch eine Konjunktion verknüpft. Ist diese UND-Verknüpfung 1 wird die SCL-Leitung auf 0 gesetzt. Geht das externe `Pause`-Signal wieder auf 0, liefert die UND-Verknüpfung ebenfalls den Wert 0 und der Zugriff des Moduls *PauseCtrl* auf die Clock-Leitung SCL wird beendet.

Die zentrale Steuereinheit

Die Kontrolleinheit wird im Verilog-Modul *Ctrl* beschrieben, und ist das zentrale Steuerelement des Interface. Hier laufen alle Statusinformationen zusammen und von hier aus werden alle nötigen Steuersignale für die anderen Module generiert. Der Controller ist als explizite Zustandsmaschine mit 7 Zuständen realisiert. Ein Wechsel der Zustände findet mit der *negativen* Flanke der Clock `SCL` statt. In Abbildung 5.16 ist das zugehörige Zustandsdiagramm gezeigt.

Durch ein asynchrones Reset-Signal (`notReset` geht auf 0) wird die State-Machine initialisiert und nimmt den Zustand *IDLE* ein. Wird eine `Start`-Bedingung auf dem Bus detektiert, geht das *Start*-Signal auf 1 und die Zustandsmaschine wechselt bei der nächsten negativen Flanke in den Zustand *RADDR*. Nach acht Takzyklen erfolgt der Übergang in den Zustand *SAACK*. Von hier aus erfolgen die wesentlichen Verzweigungen: War der Adressvergleich negativ (`AckEN`=0) wird die Zustandsmaschine in den *IDLE*-Zustand zurückversetzt. Wurde die 10-Bit-Kennung empfangen (`AddrEND`=0) ist der nächste Zustand *RADDR*. Hier wird das zweite Adresswort der 10-Bit Adresse empfangen. Nach acht Taktzyklen wechselt die State-Machine erneut in den Zustand *SAACK*. Ab hier ist der weitere Ablauf identisch mit den 7-bit Modi. In Abhängigkeit des Richtungsbits $R/\overline{W}$ ist der nächste Zustand *SDATA* ($R/\overline{W}=1$) bzw. *RDATA* ($R/\overline{W}=0$)[9]. An diese Zustände schließen sich die Acknowledge Zustände an, in denen eine *Acknowledge* generiert (*SDACK*) bzw. empfangen (*RVACK*) wird. Anschlie-

[9]Die Namen der Zustände sind aus Sicht des Interfaces vergeben.

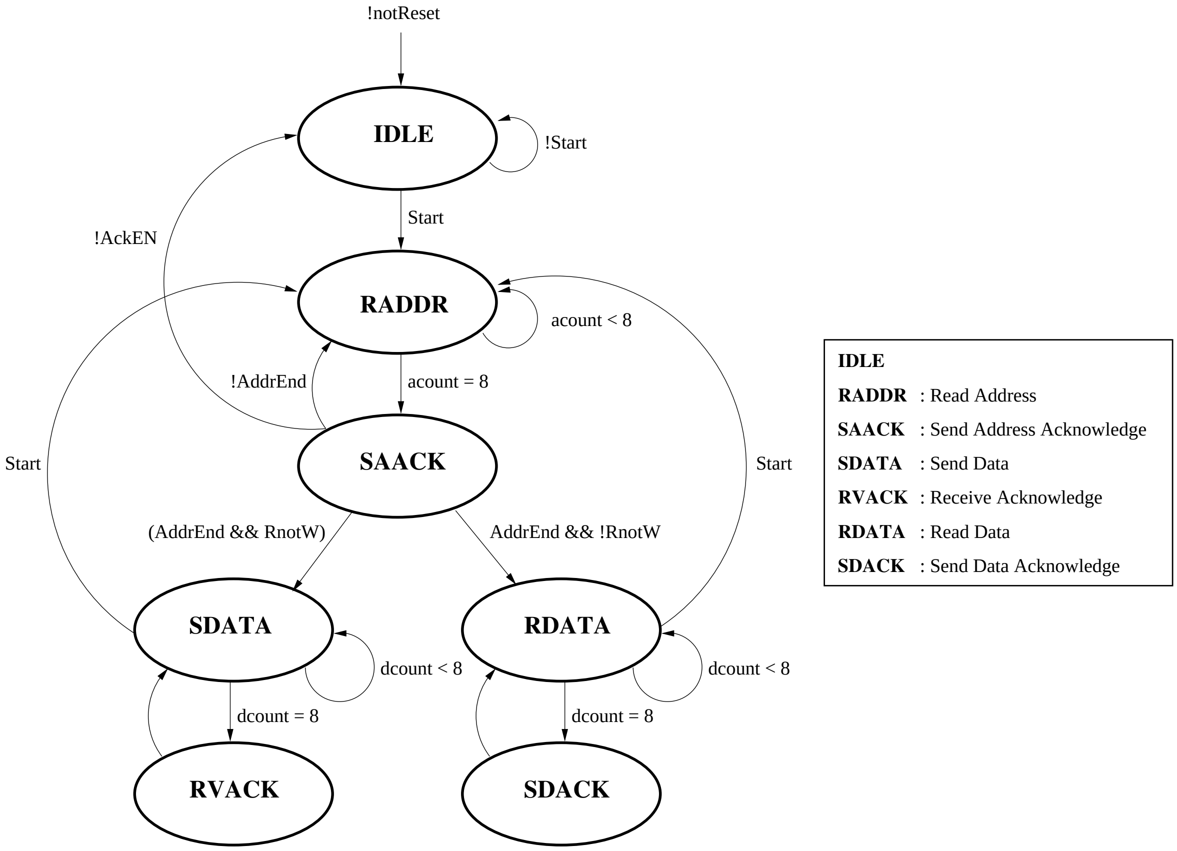

Abbildung 5.16: Zustandsdiagramm des Moduls *Ctrl* (Die Namen der Zustände sind aus Sicht des Interfaces vergeben).

ßend erfolgt der Übergang zurück in die Zustände *SDATA* und *RDATA*. Hier verbeibt die Zustandsmaschine bis eine neue *Start*-Bedingung auf dem Bus detektiert wurde. Ist `Start` logisch 1, wird in den Zustand *RADDR* gewechselt.

Das Verzögerungselement

Die Spezifikationen zum I^2C-Bus schreiben eine interne Hold-Zeit für das SDA-Signal von mindestens 300ns vor ([I^2C98, Seite 20]), um den undefinierten Bereich der fallenden Flanke von `SCL` zu überbrücken. Aus diesem Grund wurde ein Verzögerungselement in das Interface integriert.

Bei der Verzögerung eines digitalen Pulses kann man die Zeitkonstante eines RC-Gliedes ausnutzen. Das Prinzip ist in Abb. 5.17 veranschaulicht. Der SCHMITT-Trigger ist ein Komparator mit zwei Schwellen, die durch ein positives Feedback bewirkt werden.

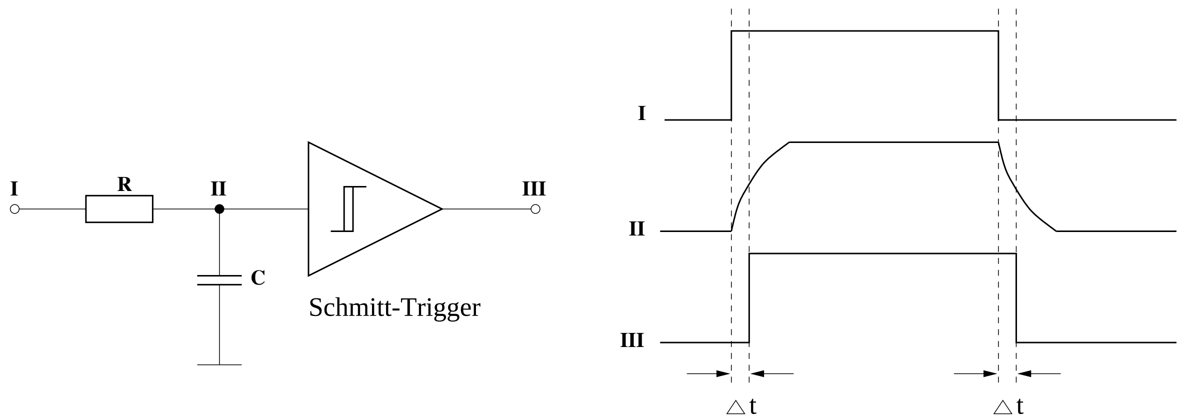

Abbildung 5.17: Prinzip der Verzögerung eines digitalen Pulses.

Die zu realisierende Verzögerungszeit für das I^2C-Protokoll liegt bei τ = RC = 300ns. Bei einem CMOS-Prozeß ist man für Kapazitäten auf Werte zwischen 100 fF und 10 pF beschränkt. Der benötigte Widerstand liegt damit zwischen 3 MΩ und 30 kΩ, was die Realisierung durch einen einfachen ohmschen Widerstand ausschließt.

Die verwendete Kapazität hat einen Wert von 1pF. Damit ergibt sich ein Widerstand von 300kΩ. Dieser wurde durch die Transistorschaltung in Abb. 5.18 realisiert. Die Transistoren *M0* und *M1* dienen als Stromquellen. Der Strom wird über die Spannung am Gate gesteuert. Die benötigten Biasspannungen werden über eine Spannungsteilerkette erzeugt. Abb. 5.19 zeigt das Simulationsergebnis des Verzögerungselements.

Die Tristate-Buffer

Ein Tristate-Gatter ist ein Logik-Element mit *drei* Ausgangszuständen. Neben den beiden gewöhnlichen Zuständen logisch 1 und 0 existiert noch ein offener Zustand, in dem der Ausgang hochohmig ist (die gebräuchliche Notation hierfür ist Z). Der dritte Zustand wird über eine Steuerleitung selektiert. Im CIPix sind die Tristate-Gatter in den Schaltungen für die Pads integriert.

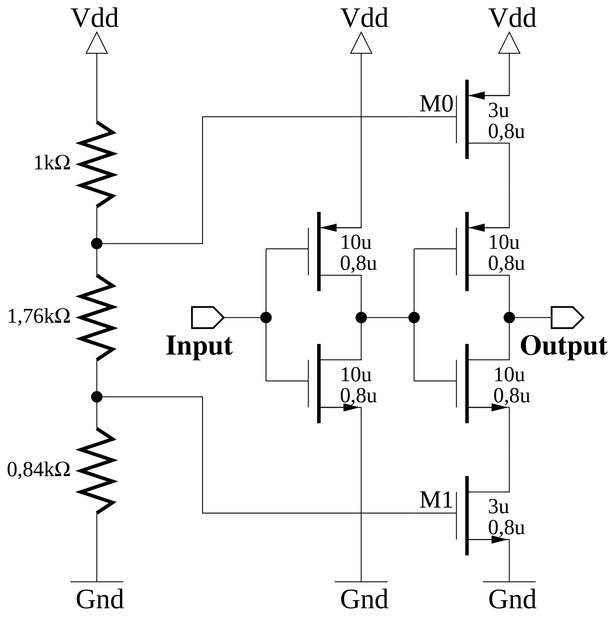

Abbildung 5.18: Realisierung des Verzögerungswiderstandes durch Transistoren.

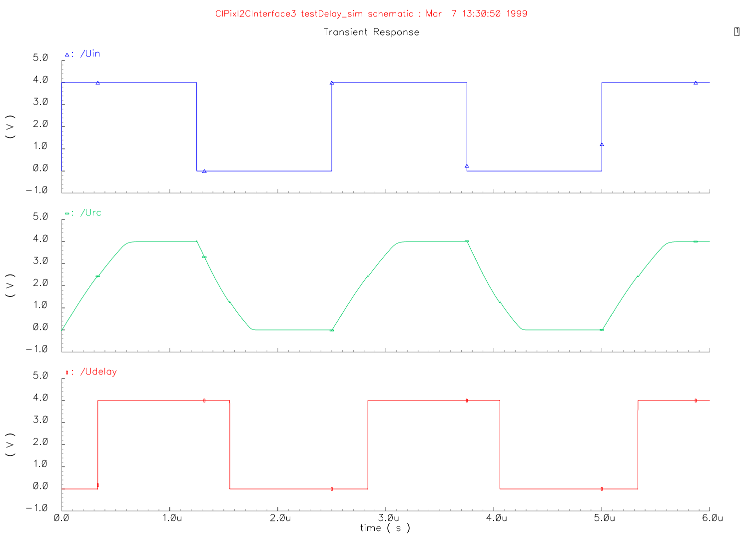

Abbildung 5.19: Simulationsergebnis des Verzögerungselements.

5.2 Der Adress-Generator

Die Aufgabe des Adressgenerators ist die Vergabe einer Chip-Adresse in einem selbstprogrammierenden Verfahren. Dazu werden die Chips in einer Kette verschaltet, wobei jeder Chip in der Adressierungskette einen Adressgenerator besitzt. Das Prinzip der ID-Vergabe besteht darin, die positiven Flanken eines externen Taktes zu zählen. Der Zähler eines Chips stoppt, wenn er ein entsprechendes Signal (Token) eines vorhergehenden Chips in der Adressierungskette erhalten hat. Der gestoppte Adressgenerator sendet ein Token an den nachfolgenden Chip. Der Zählerstand nach einem erfolgten Stop bestimmt die Adresse. Die Ein- und Ausgangssignale werden in Tabelle 5.5 erklärt.

Symbol	Typ	Beschreibung
`notReset`	Eingang	power-on Reset
`Sclk`	Eingang	Sampling clock
`ATO`	Ausgang	Token-Ausgang von Chip n
`ATI`	Eingang	Token-Eingang vom vorhergehenden Chip (n-1)
`ATIP`	Eingang	Token-Eingang vom Chip (n-2)
`AOD`	Ausgang	Zählerstand nach erfolgtem Stop. Ist kein Stop erfolgt, hat `AOD` den initialen Wert 0.

Tabelle 5.5: Ein- und Ausgangssignale des Adressgenerators.

Ein Verfahren, bei dem ein Glied in der Kette auf ein Signal eines vorhergehenden Teilnehmers wartet, ist sehr empfindlich auf den Ausfall eines Teilnehmers oder die Unterbrechung einer Verbindung. Um nun zu gewährleisten, daß die Adressvergabe redundant gegenüber dem Ausfall *eines* Chips in der Adressierungskette ist, besitzt jeder Adressgenerator zwei Tokeneingänge (`ATI` und `ATIP`): `ATI` ist dabei der Tokenausgang des vorhergehenden Chips, `ATIP` der des Chips vor diesem. Jeder der beiden Token kann den Zähler stoppen. Dabei wird `ATIP` intern um eine Clock-Periode verzögert. Das stellt sicher, daß `ATIP` den Zähler zum richtigen Zeitpunkt stoppt. Ohne die Verzögerung wäre die vergebene Adresse um 1 zu klein, und gleich der des vorhergehenden Chips. Fallen jedoch *zwei*, in der Adressierungskette *benachbarte* Chips aus, erhält der nachfolgende Chip weder ein `ATIP`- noch ein `ATI`-Signal und die Adressierung bricht ab. Das heißt, die Zähler der nachfolgenden Chips laufen in den Anschlag und der Adressausgang `AOD` behält seinen initialen Wert 0 bei. In diesem Fall können die Chips weiterhin über den *General Call*-Modus programmiert werden, in dem alle Chips simultan angesprochen und beschrieben werden. Eine Auslese ist nur für die Chips mit einer Nummer ungleich null möglich. In Abbildung 5.20 ist die Verschaltung der Chips in der Adressierungskette gezeigt.

5.3 Die Decoder-Logik

Ein Decoder ist ein Schaltkreis, der mittels einer bestimmten Vorschrift (Code) den diskreten Eingangsgrößen bestimmte digitale Ausgangsgrößen zuordnet. Ein n-bit Code am Eingang setzt genau eine von 2^n Ausgangsleitungen auf logisch eins. Für den CIPix wurden zwei unterschiedliche Decoder implementiert. Ein Decoder regelt den Schreibzugriff (Modul *DecoderWRITE*), der andere den Lesezugriff (Modul *DecoderREAD*). Beide sind als Zustands-

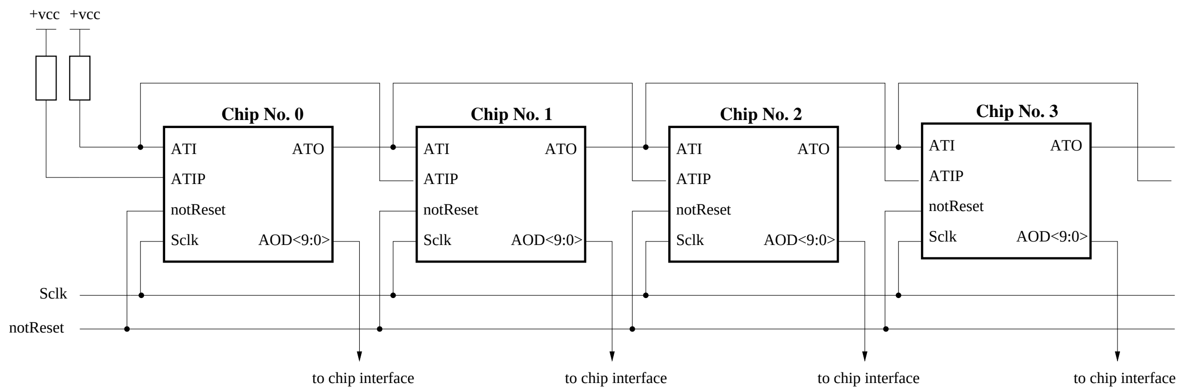

Abbildung 5.20: Schematische Darstellung der Adressierungskette.

maschinen realisert, arbeiten aber mit unterschiedliche Flanken der Clock.

Für den CIPix wurden zwei unterschiedliche Schreibsequenzen und eine Lesesequenz spezifiziert [CIP98a] [CIP98b]:

- Im *Single Write Mode* wird der Chip über eine 7-Bit Adresse angesprochen. Für jedes zu beschreibende Register wird die Adresse und das Datenwort übersandt (Abb. 5.21).
- Im *Consecutive Write Mode* wird der Chip über eine 7-Bit Adresse identifiziert. Nach der Adressierung des *CMA*[10]-Registers werden die Datenworte aller 11 Register in Folge übermittelt (Abb. 5.22).
- Im *Read Mode* wird der Chip über eine 10-Bit Adresse angesprochen. Die obersten vier Bits der gesendeten Adresse sind die Chip-ID, die unteren vier Bits die Register-ID.

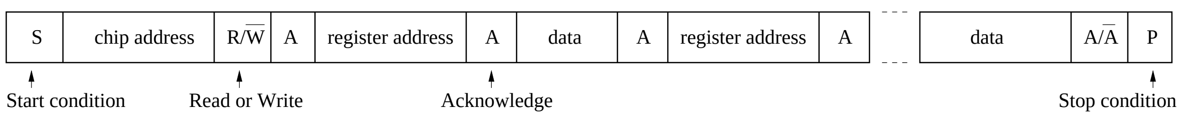

Abbildung 5.21: Übertragungssequenz im *Single Write Mode*.

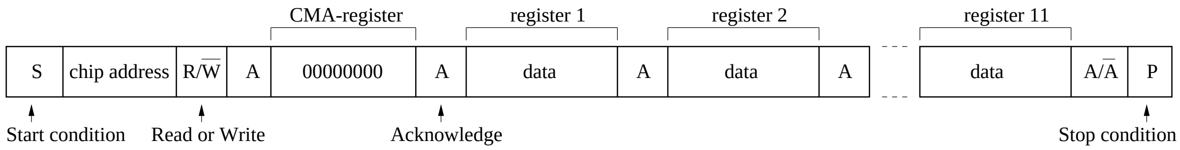

Abbildung 5.22: Übertragungssequenz im *Consecutive Write Mode*.

Abb. 5.23 zeigt die Formate der gesendeten 7-Bit und 10-Bit Chip-Adressen. Die Größe des Chip-Adressraums wird durch die 10-Bit Adresse und die Anzahl der Register bestimmt.

[10] Consecutive Mode Addressing

Für 11 Register werden 4 Bits benötigt. Damit verbleiben 6 Bits für die Chip-Adresse, wodurch die Anzahl der CIPix-Chips an einem I^2C-Bus auf 64 limitiert ist. Eine Auslese des Chips über eine 7-Bit Adresse ist nicht möglich, weil nach dem Adresswort direkt das gelesene Datenwort folgt. Es müssen also Chip- und Register-Adresse gemeinsam in einem Wort übertragen werden.

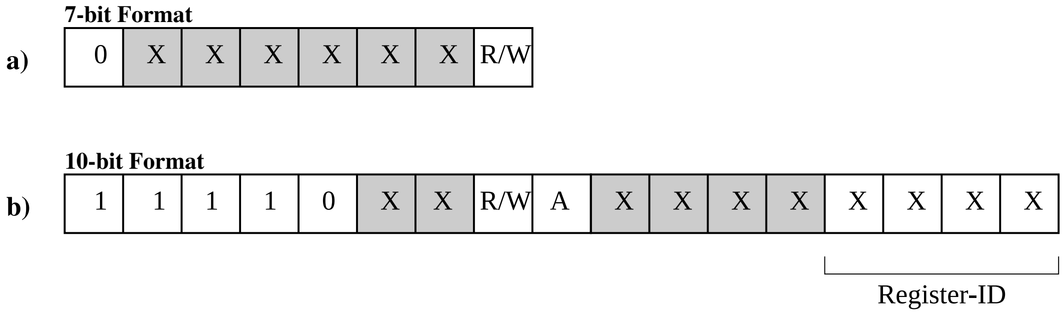

Abbildung 5.23: Format der gesendeten 7-Bit (a) und 10-Bit (b) Adressen. Grau hinterlegt sind die Bits der Chip-ID.

Die Aufgabe der Dekoder ist es, zu gewährleisten, daß entsprechend diesen Sequenzen die vom Interface empfangenen Daten an das adressierte Register geleitet werden, bzw. der Inhalt des adressierten Registers ausgelesen wird. Das Prinzip ist in Abb. 5.24 verdeutlicht. Die `WriteDAC`-Leitungen werden auf die Clock-Eingänge der Register-Flip-Flops geführt, die `ReadDAC`-Leitungen dienen als Steuereingänge für die Tristate-Gatter und regeln somit den Zugriff der Register auf den internen Datenbus. Die Decoderclock `decCLK` ist eine logische Verknüpfung von Ausgangssignalen des I^2C-Interfaces. Dadurch wird sichergestellt, daß der Decoder zum richtigen Zeitpunkt auf ein Register zugreift, nämlich nach der vollständig empfangenen Adresse oder nach einem kompletten Datenbyte.

$$\mathtt{decCLK} = (\mathtt{AddrEnd} \wedge \mathtt{AckEN}) \vee \mathtt{DataReady}$$

Der *Schreib*-Decoder muß die beiden Schreibmodi *Single* und *Consecutive* umsetzten können. Er ist als explizite Zustandsmaschine mit vier Zuständen realisiert. Ein Zustandswechsel findet mit der negativen Flanke der Clock `decCLK` statt. Das zugehörige Zustandsdiagramm ist in Abb. 5.25 gezeigt.

Durch ein Reset nimmt die State-machine den Zustand SELECT ein und die `WriteDAC`-Leitungen werden auf 0 initialisiert. Ist das Richtungsbit $R/\overline{W} = 0$, erfolgt mit der nächsten negativen Flanke der Clock der Wechsel in den Zustand WMSEL (=Write Mode SELect). Wird die Kennung des Consecutive Modes empfangen (`Data` = 0x00), erfolgt mit der nächsten Clock-Flanke der Übergang in den Zustand CWM (= Consecutive Write Mode). Andernfalls wird, entsprechend dem Inhalt von `Data[3:0]` ein Zwischenspeicher beschrieben. Mit der nächsten negativen Flanke der Clock (entspricht dem nächsten kompletten Datenbyte) wird der Zustand SWM (=Single Write Mode) eingenommen und eine der 11 `WriteDAC`-Leitungen aktiviert (auf 1 gesetzt). Damit wird das adressierte Register beschrieben. Im Zustand CWM wird mit jeder Flanke die Nummer der aktivierten (auf logisch 1 gesetzten) `WriteDAC`-Leitung inkrementiert, bis nacheinander alle 11 Register beschrieben sind.

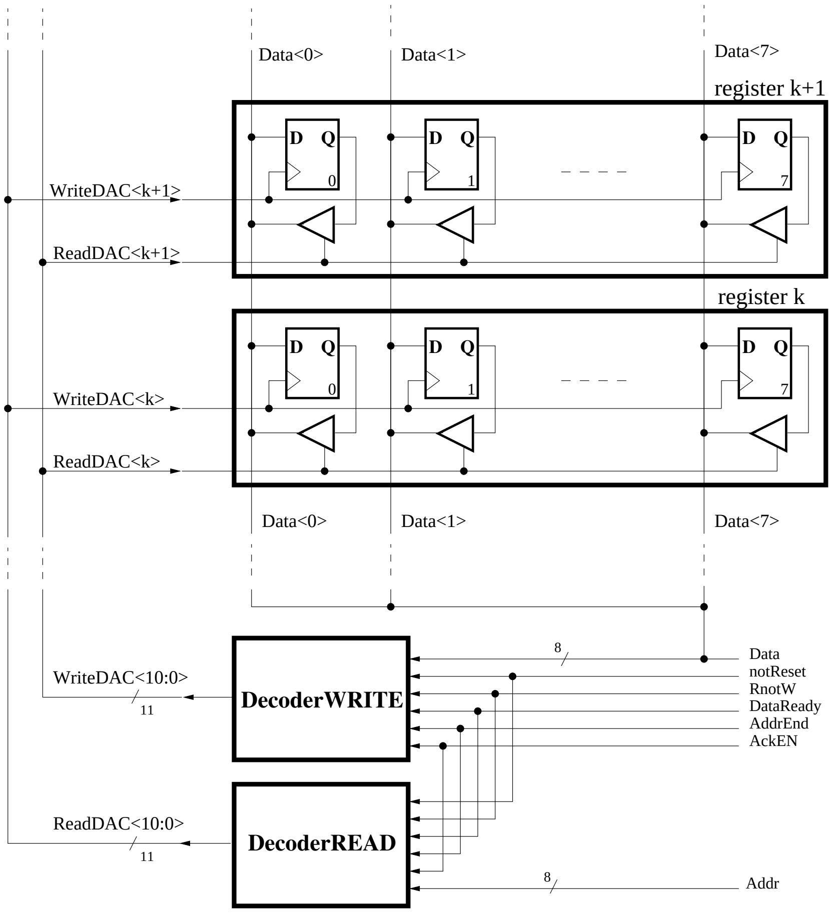

Abbildung 5.24: Schematische Verschaltung der Dekoder und DAC-Register.

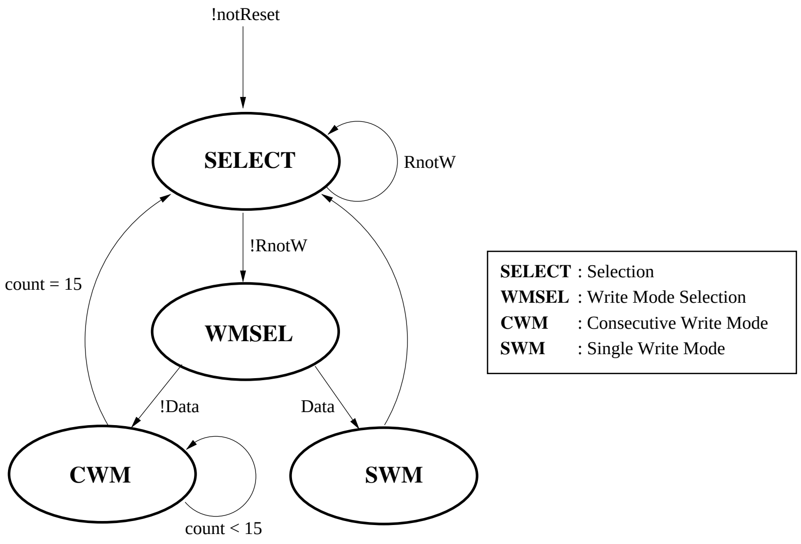

Abbildung 5.25: Zustandsdiagramm des Schreib-Dekoders.

Der *Lese*-Decoder wird ebenfalls durch ein Reset initialisiert. Mit jeder positiven Flanke der Clock wird der Wert von `Addr[3:0]` gelesen und dem Inhalt entsprechend eine der 11 `ReadDAC`-Leitungen auf 1 gesetzt.

5.4 Die Testpuls-Steuerlogik

Die Testpuls-Steuerlogik ist als positiv-flankengetriggerte Zustandsmaschine im Modul *TestpulseLogic* realisiert. Als Takt wird die 10,4 MHz Sampling-Clock *Sclk* benutzt. Das Modul besitzt drei Steuerausgänge die in Tabelle 5.6 erklärt werden. In Abhängigkeit des Inhaltes des Registers `TpReg` werden die Ausgangssignale gesetzt. Die Belegung des Registers ist in Abb. 5.26 gezeigt.

Symbol	Beschreibung
MSel	Mode Selection: 0: Interner Testpuls, 1:Externer Testpuls
CSel	Channel Selection 0: Single Channel (0,21,42,63), 1: All Channels
IntPulse	Interner Testpuls

Tabelle 5.6: Ein- und Ausgänge der Testpuls-Logik.

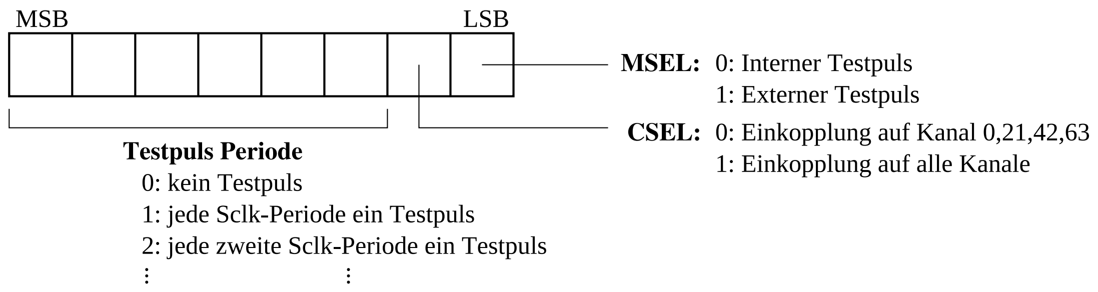

Abbildung 5.26: Belegung des Registers *TpReg*.

5.5 Die Steuerlogik für die Analogschalter

Die Steuerlogik für die Analogschalter ist als implizite Zustandsmaschine im Modul *AnalogControl* realisiert und beinhaltet im wesentlichen einen 1-aus-64-Dekoder. Durch ein Reset-Signal werden die Selektionsleitungen `ChipON` und `SEL[63:0]` auf 0 initialisiert. Das Signal `ChipON` steuert den Zugriff des Chips auf den analogen Bus (siehe Abb. 3.1). `SEL[63:0]` kontrolliert die 64 Schalter, die zur Auskopplung des analogen Signals nach der Verstärkerstufe dienen. Mit jeder positiven Flanke der Sampling Clock *Sclk* wird, entsprechend dem Inhalt des Registers *AnanlogOut*, `ChipON` auf 0 oder 1 gesetzt und eine der 64 Ausgangsleitungen `SEL[63:0]` aktiviert. In Abb. 5.27 ist die Registerbelegung von *AnanlogOut* gezeigt.

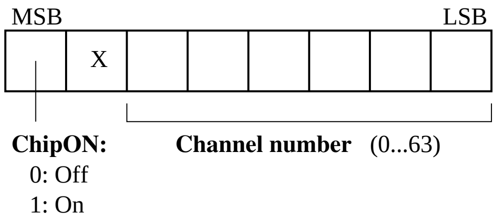

Abbildung 5.27: Belegung des Registers *AnalogOut*.

5.6 Simulationen

Die Simulationen zu den einzelnen Verilog-Modulen wurden mit dem Simulator *Verilog-XL* [VXL97] durchgeführt. Das zu simulierende Modul wird hierzu in ein anderes Verilog-Modul eingebunden (instanziiert), in dem die Teststimuli generiert werden.

Adressgenerator

Abb. 5.28 zeigt die Ergebnisse der Simulation des Adressgenerators. Der Ausschnitt zeigt vier Chips aus der Adressierungskette. Es sind die drei Situationen dargestellt, die auftreten können: Ein Chip erhält beide Tokeneingänge, nur einen Tokeneingang oder keinen Tokeneingang. In den ersten beiden Fällen erfolgt eine Adressvergabe, im letzten Fall schlägt diese fehl. Der entsprechende Chip und alle nachfolgenden behalten den initialen Adresswert 0 bei.

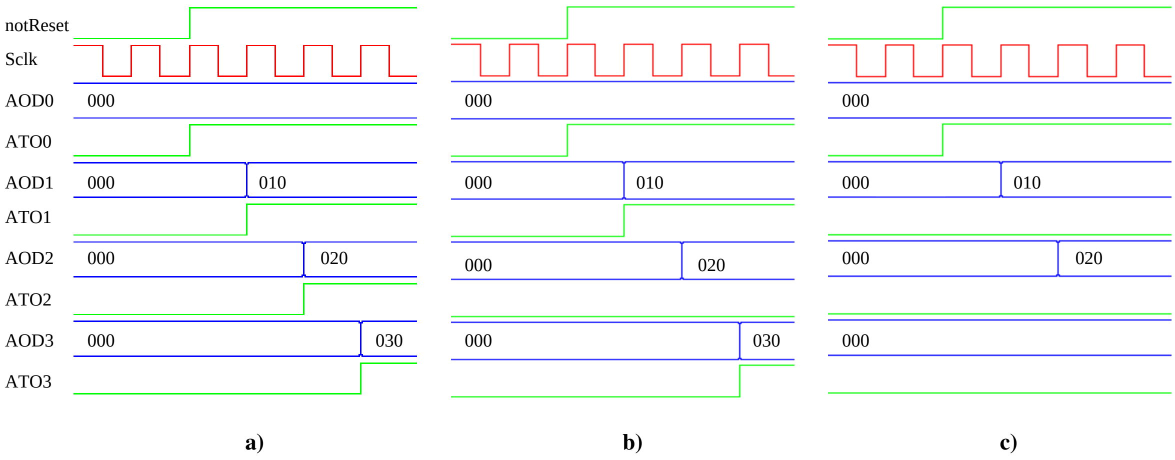

Abbildung 5.28: Simulationsergebnis des Adressgenerators. a) Chip 3 erhält beide Signale *ATI* und *ATIP*. b) Chip 3 erhält nur *ATIP*. c) Chip 3 erhält keinen Tokeneingang. Es erfolgt keine Adresszuweisung und keine Generierung eines Tokenausgangs. *AOD3* behält den initialen Wert 0 bei.

I²C-Interface

Nachfolgend sind die Simulationsergebnisse für einen *Single Write Mode* (Abb. 5.29), einen *Consecutive Write Mode* (Abb. 5.30), eine Auslesesequenz (Abb. 5.31) und einen Wartezustand auf dem Bus (Abb. 5.32) dargestellt. Am Fuß der jeweiligen Graphik ist die Simulationszeit in Mikrosekunden angegeben. Die auftretenden Zahlenangaben sind in hexadizimalem Format. Ein X steht für einen unbestimmten Wert.

- *Single Write Mode* (Abb. 5.29): Es wird das Interface mit der Adresse `0x290` angesprochen und der Wert `0x64` in Register `0x03` geschrieben.
- *Consecutive Write Mode* (Abb. 5.30): Es wird das Interface mit der Adresse `0x290` angesprochen. Nachfolgend werden alle Register beschrieben. Der dargestellte Ausschnitt zeigt jedoch nur die ersten fünf Registerinhalte.
- Auslesesequenz (Abb. 5.31): Vom Interface mit der Adresse `0x290` wird Register `0x09` ausgelesen. Der Zeitbereich, in dem der Registerinhalt auf der SDA-Leitung anliegt, ist markiert.
- Wartezustand auf dem Bus (Abb. 5.32): Das externe Signal *Pause* erzwingt einen Wartezustand. Der Bereich, in dem die Taktleitung *SCL* auf logisch 0 gezogen wird, ist markiert.

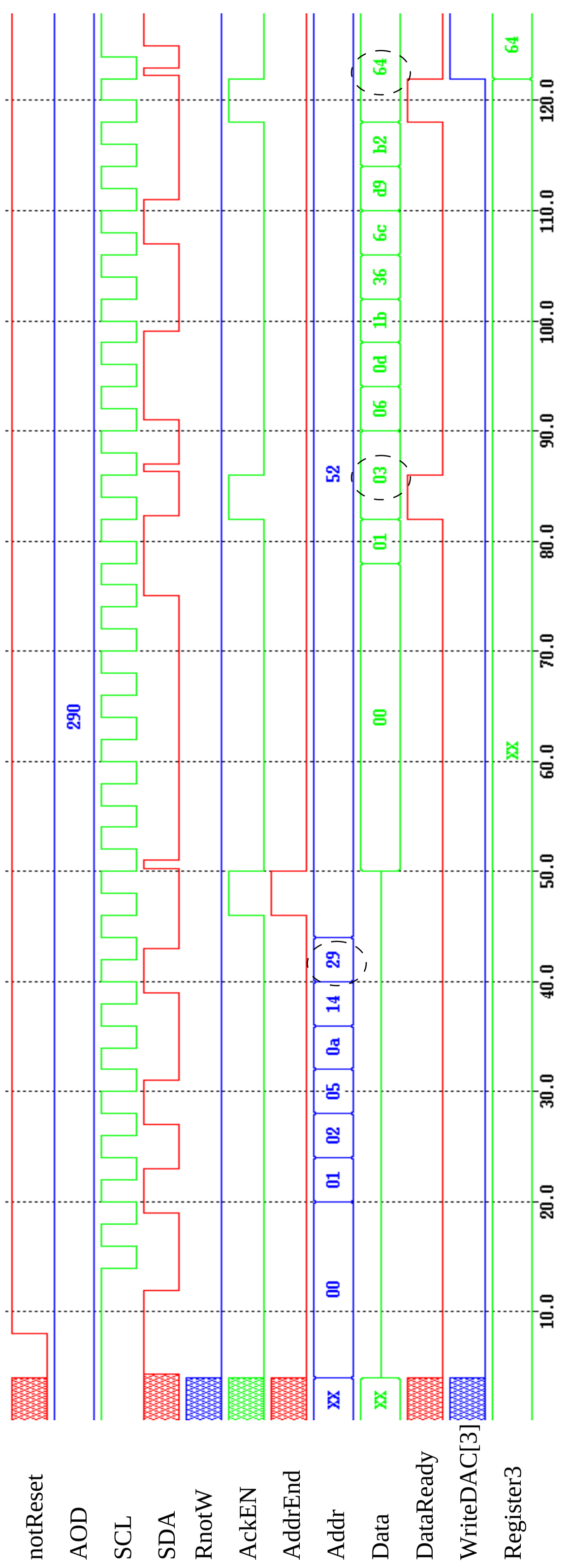

Abbildung 5.29: Simulation des *Single Write Mode*.

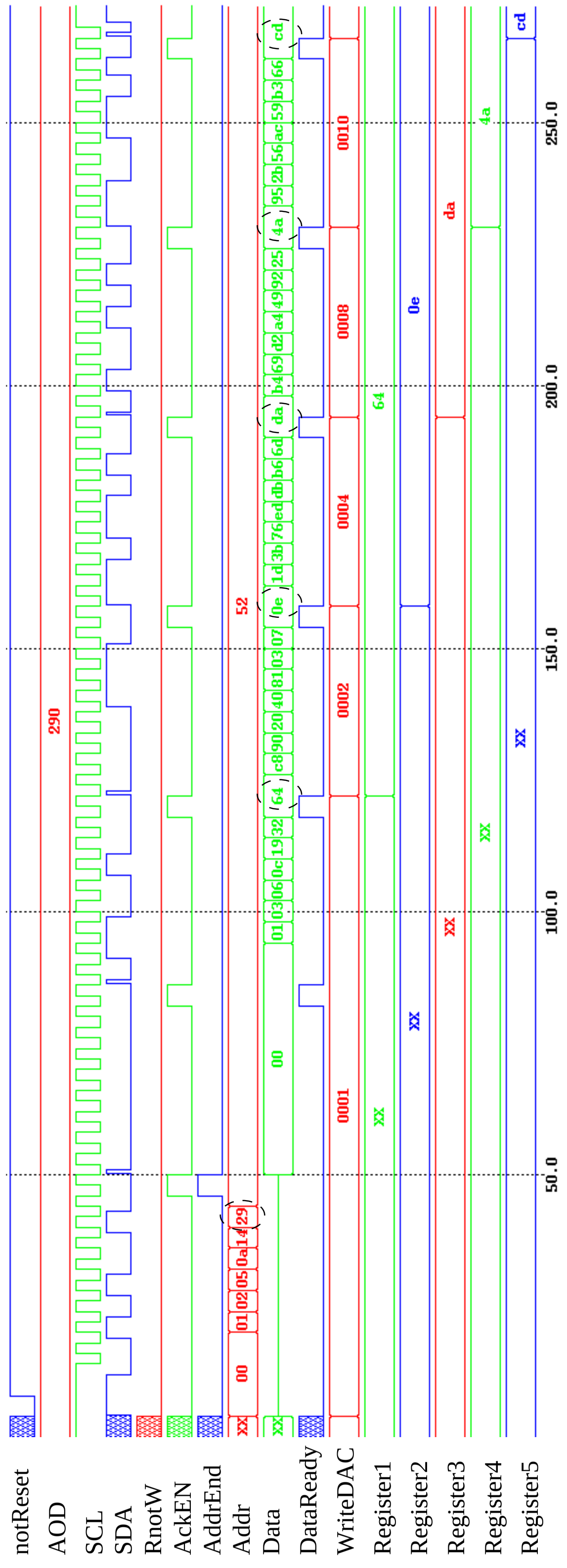

Abbildung 5.30: Simulation des *Consecutive Write Mode*. Der Ausschnitt zeigt die ersten 5 Register.

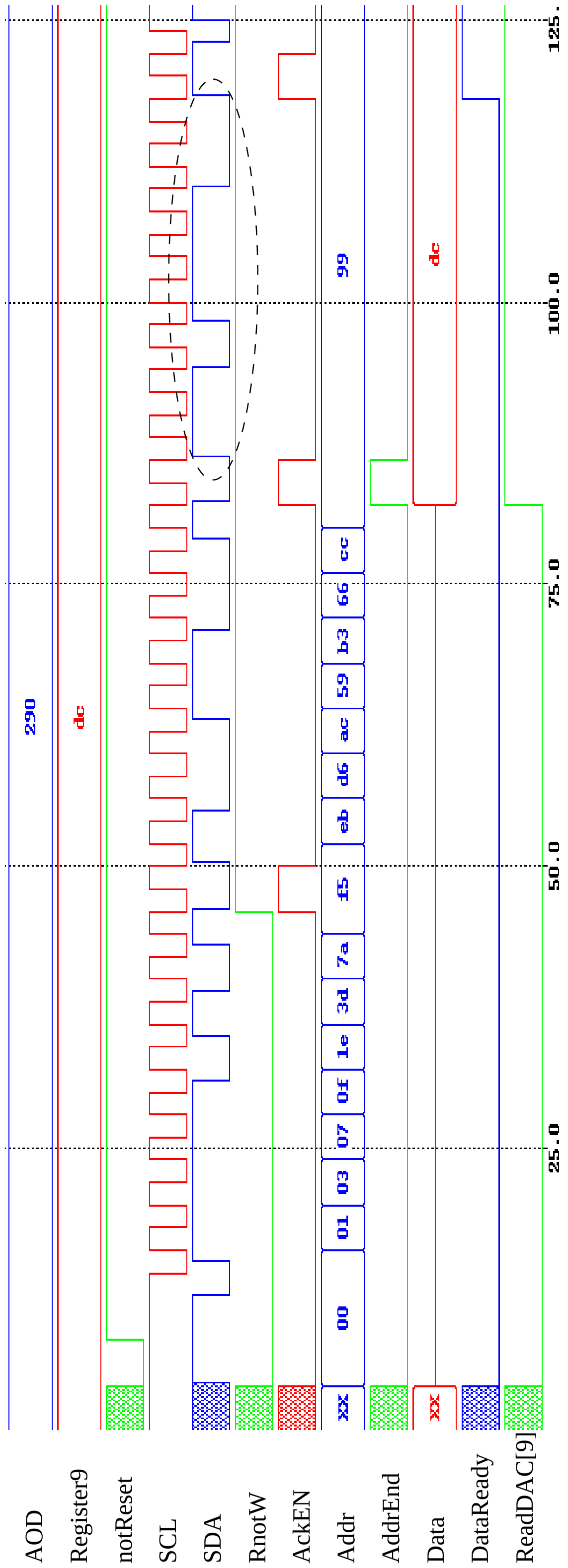

Abbildung 5.31: Simulation der implementierten Auslesesequenz.

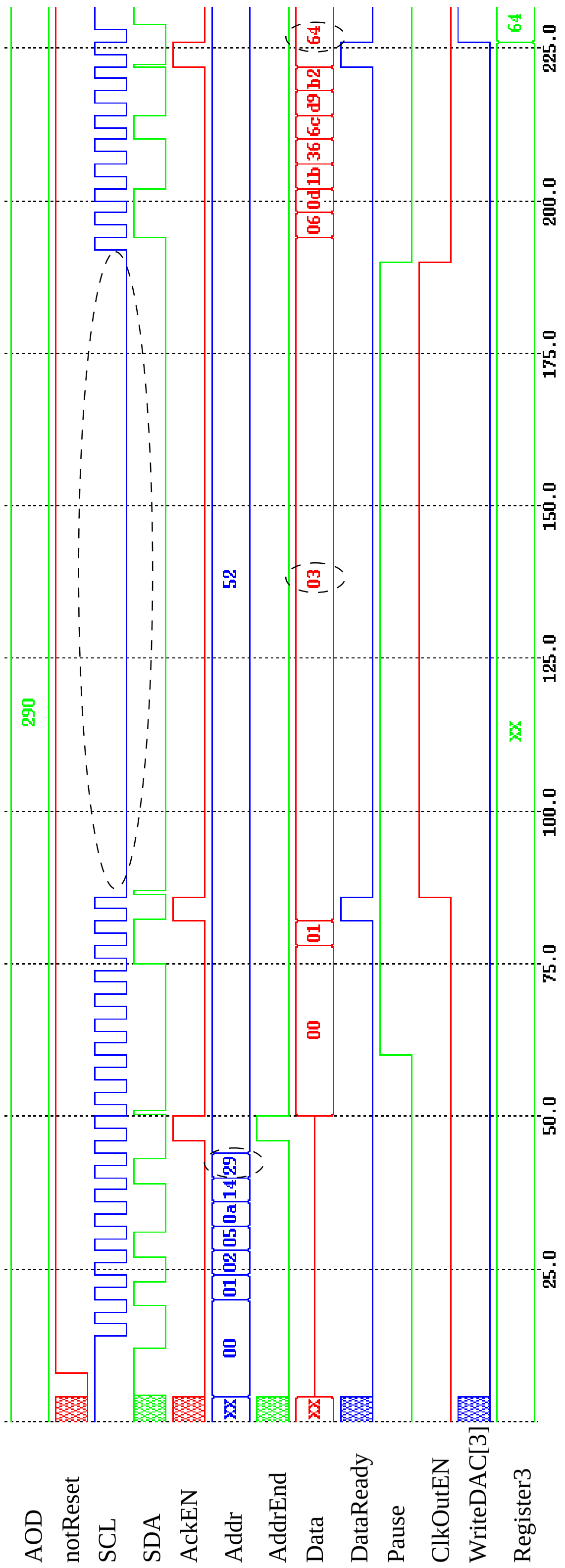

Abbildung 5.32: Simulation eines Wartezustands auf dem I^{2}C-Bus.

Kapitel 6

Messungen am CIPix

6.1 Messaufbau

Für die Messungen am CIPix mußte dieser erst in eine geeignete Testumgebung gebracht werden. Der Chip wurde in ein 84-poliges Gehäuse (CLCC84) geklebt und die Kontaktierungsflächen (Pads) mit den Gehäuse-Pins durch Bonddrähte[1] verbunden. Der Chip muß mit einigen passiven Bauelementen, wie Blockkapazitäten und Widerständen extern beschaltet werden. Ein entsprechender Schaltplan ist in Anhang B dargestellt. Das Chip-Gehäuse, die Blockkapazitäten und Widerstände, sowie die Einkoppelschaltung für den Testpuls (Abb. 6.3) und verschiedene Steckverbindungen wurden auf einer Loch-Platine untergebracht. Abbildung 6.1 zeigt den schematischen Meßaufbau.

Der CIPix benötigt drei verschiedene Versorgungsspannungen. Diese werden vom *Triple Power-Supply NGPT35* von *Rohde&Schwarz* bereitgestellt. Die digitalen Steuerleitungen des Chips, die Clocks sowie der Testpuls werden vom *Data Generator DG2020A* von *Tektronix* erzeugt.

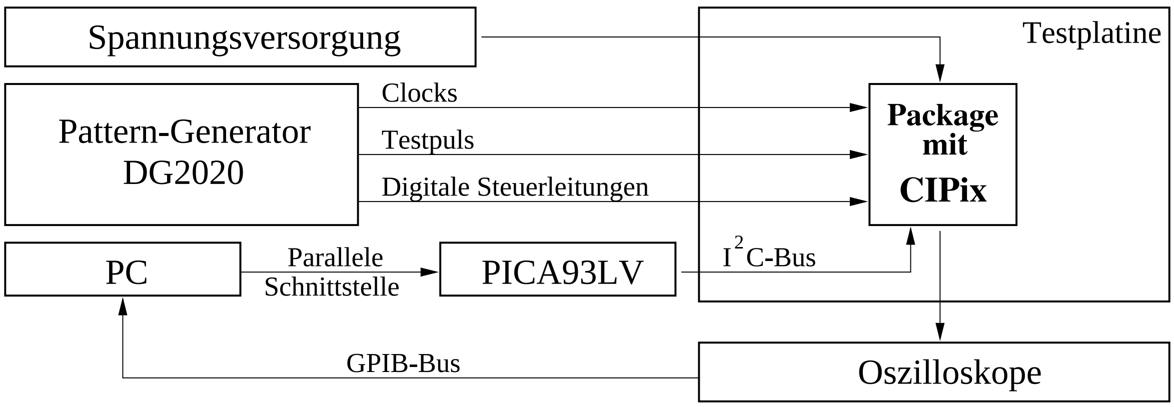

Abbildung 6.1: Schematischer Überblick über den Meßaufbau.

[1]Ein Bonddraht ist ein etwa 25 μm dicker Aluminium- oder Golddraht, der zwischen zwei Kontakten durch Ultraschallschweißen befestigt wird.

Programmierung

Zur Programmierung der Chipregister wurde ein kommerzieller Baustein eingesetzt, der ein Standard-I^2C-Protokoll erzeugt. Bei funktionierender Programmierung wird damit sichergestellt, daß die Schnittstelle des CIPix den Spezifikationen des I^2C-Bus genügt. Der eingesetzte Adapter *PICA93LV* [Cal98a] [Cal98b] der Firma *Calibre* wird über die parallele Schnittstelle eines PCs angesteuert. Er arbeitet intern mit dem I^2C-Controller *PCF8584* von *Philips* [Phil97]. Die Ausgangspegel der beiden I^2C-Leitungen sind 3,3 V CMOS. Zum Lieferumfang des Adapters gehören DLL[2]-Bibliotheken für *Windows95* in der Programmiersprache C. Diese enthalten die grundlegenden Funktionen zur Umsetzung eines Transfers auf dem I^2C-Bus. Mit Hilfe dieser C-Funktionen wurde ein Programm entwickelt, das die verschiedenen Programmiermodi des CIPix aufsetzt. Dadurch ist eine einfache und schnelle Programmierung des Chips möglich. In Tabelle 6.1 ist die Registerbelegung des CIPix1.0 zusammengestellt. Tabelle 6.2 zeigt die Nominalwerte der analogen Regelspannungen und -ströme, mit denen der Chip programmiert wurde.

Adresse (HEX)	Name	Beschreibung
0	*CMA*	Consecutive Mode Addressing
1	*Ipre*	Vorverstärker-Regelstrom
2	*Isha*	Shaper-Regelstrom
3	*Ibuf*	Buffer-Regelstrom
4	*Icomp*	Komparator-Regelstrom
5	*Idriver*	Ausgangstreiber-Regelstrom
6	*Vfp*	steuert den Rückkoppelwiderstand des Vorverstärkers
7	*Vfs*	steuert den Rückkoppelwiderstand des Shapers
8	*Vref*	steuert die Schaltschwelle des Komparators
9	*Voffset*	Regelspannung des Ausgangstreibers
A	*AnalogOut*	selektiert den analogen Ausgangskanal
B	*TpReg*	steuert die Testpulslogik

Tabelle 6.1: Belegung der 11 internen Register des CIPix1.0.

6.2 Messungen am Interface

Adressierung

Zunächst wurde die Funktionsweise des Adressgenerators überprüft. Die beiden Tokeneingänge *ATI* und *ATIP* wurden vom Patterngenerator erzeugt. Die erfolgreiche Zuweisung einer Adresse wird durch eine logische 1 auf der *ATO*-Leitung angezeigt. Abb. 6.2 zeigt die entsprechenden Sequenzen. Im zweiten Fall wird die Situation eines Chip-Ausfalls simuliert. Das *ATI*-Signal des vorhergehenden Chips wird nicht generiert. *ATIP* führt jedoch eine korrekte ID-Vergabe herbei. Damit konnte gezeigt werden, daß das Adressierungsschema redundant gegenüber dem Ausfall eines Chips ist.

[2]Dynamic Link Library

Name	Nominalwert und dez. Registerwert
Ipre	200μA=80
Isha	100μA=40
Ibuf	100μA=40
Icomp	50μA=20
Idriver	90μA=36
Vfp	0,2V=141
Vfs	1,5V=224
Voffset	-0,5V=96

Tabelle 6.2: Nominalwerte der analogen Regelspannungen und -ströme.

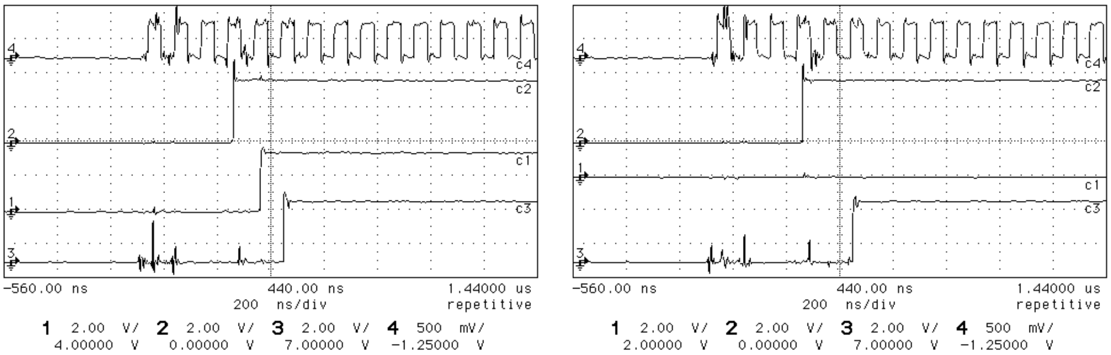

Abbildung 6.2: CIPix-Adressvergabe. Im rechten Bild wird der Ausfall eines Chips in der Adressierungskette simuliert. Kanal 1 zeigt *Sclk*, Kanal 2 *ATIP*, Kanal3 *ATI* und Kanal 4 *ATO*.

Programmierung

Eine erfolgreiche Programmierung der Register des Chips setzt die Funktionsfähigkeit von I^2C-Interface und Dekoder voraus.

Eine korrekte Arbeitsweise des Interfaces ist durch die Erzeugung eines Acknowledge-Bits nach jedem Byte zu erkennen. Alternativ dazu kann eine Überprüfung mittels Software erfolgen. Der zur Ansteuerung des CIPix verwendete I^2C-Adapter besitzt ein internes Register, in dem der Transferstatus gespeichert wird.

Wartezustand

Für den Test des Wartezustands wurde ein periodisches Signal während der Programmiersequenz auf den Anschluß *Pause* gegeben. Es zeigte sich, daß die Taktleitung *SCL* auf logisch 0 gelegt wurde, allerdings nur für die Dauer einer *SCL*-Periode. Außerdem scheint das Verhalten von der Art der Programmierung (*General Call*, *7-Bit Adressierung*) abzuhängen, was noch genauer zu untersuchen ist. Da jedoch der Wartezustand eine reine Eigenschaft des Interface ist und keine Auswirkungen auf den Betrieb des CIPix hat, tritt dieses Verhalten in den Hintergrund.

6.3 Frontend-Messungen

Für die Messungen am Frontend des *CIPix* wurde ein Testpuls auf das analoge Eingangspad 21 eingekoppelt. Abb. 6.3 zeigt die entsprechende Schaltung.

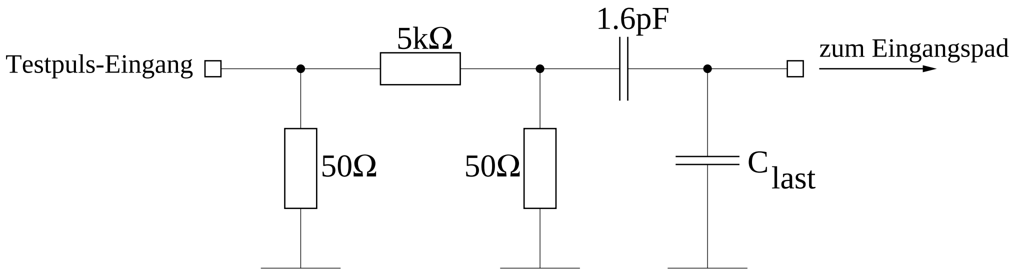

Abbildung 6.3: Schematische Schaltung für die Einkopplung eines Signals auf ein analoges Eingangspad.

Es wurde das Verhalten des analogen Ausgangssignals bei verschiedenen Lastkapazitäten C_{last}, sowie bei Einkopplung über ein Mikro-Koaxialkabel[3] untersucht (Abb. 6.4, Abb. 6.5, Abb. 6.6). Das Koaxialkabel hat eine Länge von 2,2 m bei einer Leitungskapazität von 29 pF/m [US99]. Bei den Aufnahmen wurden die Taktsignale *Sclk* und *Rclk* abgeschaltet, um ein externes Übersprechen auf das analoge Ausgangssignal zu vermeiden. Es ist sehr wahrscheinlich, daß dieses Übersprechen auf der Strecke zwischen Gehäusepin und Bondpad stattfindet. Hier laufen die Bonddrähte der Clocks und des analogen Ausgangs über eine Distanz von einigen Millimetern parallel.

Die Amplitude des Ausgangssignals nimmt von etwa 200 mA bei $C_{last} = 21$ pF auf etwa 100 mA bei $C_{last} = 100$ pF ab. Für die Einkopplung über das Koaxialkabel ergibt sich eine Lastkapazität von etwa 65 pF. Die entsprechende Bildschirmaufnahme (Abb. 6.4 rechts) korrespondiert mit der Aufnahme bei einer Lastkapazität von 66 pF (Abb. 6.6 links).

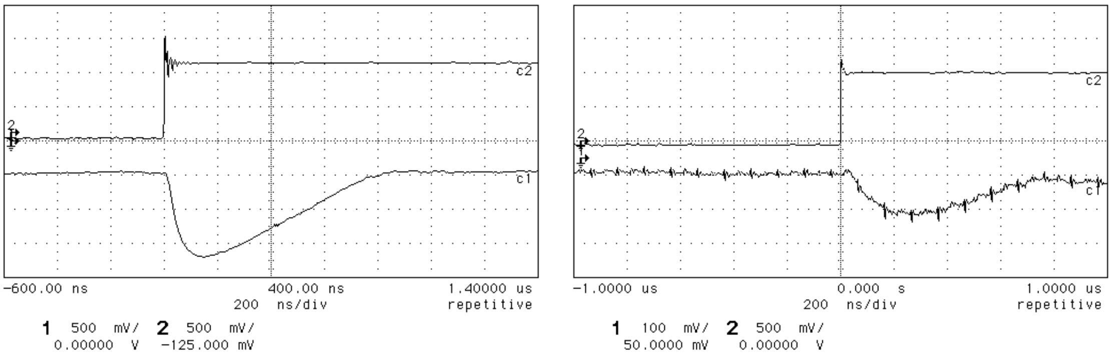

Abbildung 6.4: Testpulseingang (oben) und analoges Ausgangssignal (unten). Im linken Bild ist $C_{last} = 0$. Im rechten Bild wurde über das Mikrokoaxialkabel eingekoppelt.

[3]Bei der Konstruktion der neuen CIP wird neben der Auslese über Microstriplines auch eine Auslese über Mikrokoaxialkabel erwogen.

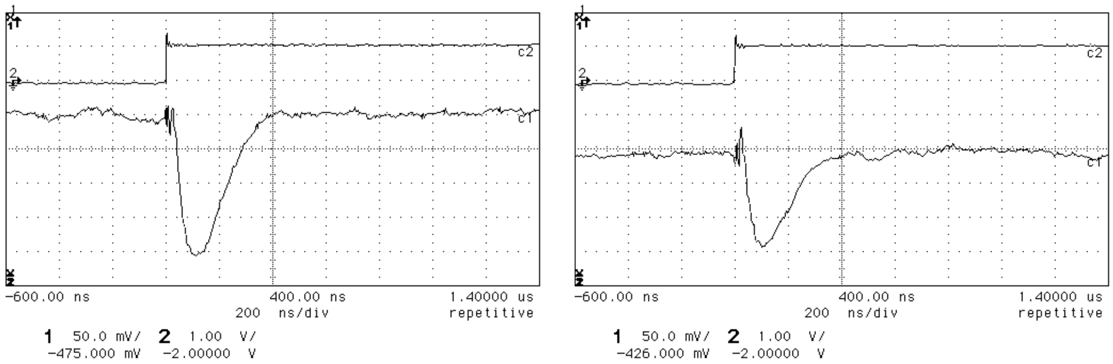

Abbildung 6.5: Testpulseingang (oben) und analoges Ausgangssignal (unten) für Lastkapazitäten von 21 pF (links) und 45 pF (rechts).

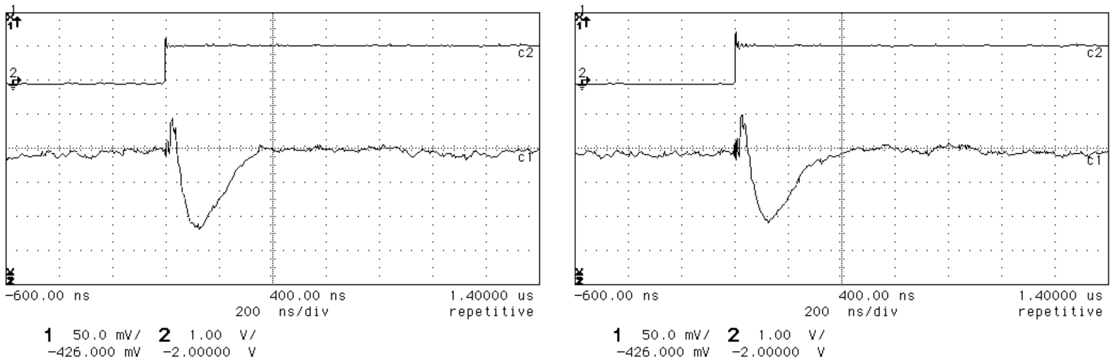

Abbildung 6.6: Testpulseingang (oben) und analoges Ausgangssignal (unten) für Lastkapazitäten von 66 pF (links) und 100 pF (rechts).

Kapitel 7

Ausblick

Bei Simulationen, die aus Zeitgründen erst nach der Submission durchgeführt wurden, sowie bei Messungen am Chip, ergaben sich eine Reihe von Verbesserungsansätzen und Korrekturen. Diese sollen im nachfolgenden Kapitel beschrieben werden.

7.1 Korrekturen am Adressgenerator

Die erste vom Adressgenerator vergebene Adresse ist die 0. Diese ist aber im Adressraum des I^{2}C-Protokolls für den Modus *General Call* reserviert. Der Adressgenerator wurde daher insoweit modifiziert, daß die erste vergebene Adresse 1 ist.

7.2 Korrekturen am I^{2}C-Interface

Bei den ersten Messungen am CIPix zeigte sich sehr schnell, daß der Auslesemodus nicht korrekt funkionierte. Zwar konnten beschreibene Register rückgelesen werden, allerdings nicht über ein Standard-I^{2}C-Protokoll. Bei der Entwicklung des Interfaces wurde der 10-Bit Lesemodus nicht korrekt implementiert. In Abb. 7.1 sind der fehlerhafte und der spezifizierte Modus gegenübergestellt. Abb. 7.2 zeigt die Ergebnisse einer Messung am Chiptester HP82000. Die Markierungen kennzeichnen die Bereiche, in denen die auszulesenden Daten auf der *SDA*-Leitung anliegen. Eine Auslese erfolgt jedoch nur in der unteren Sequenz, die nicht dem I^{2}C-Protokoll entspricht.

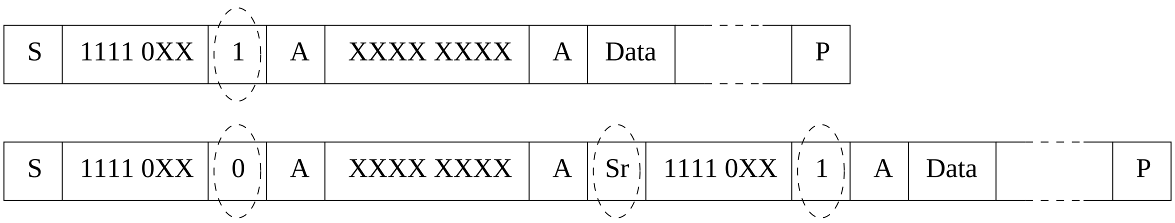

Abbildung 7.1: Gegenüberstellung der implementierten (oben) und der spezifizierten (unten) I^{2}C-Auslesesequenz bei einer 10-Bit Adressierung.

Eine nachträgliche genauere Betrachtung des *General Call* machte auch hier eine Modifikation notwendig. Der *General Call* umfaßt mehrere Untermodi, die im zweiten Byte nach

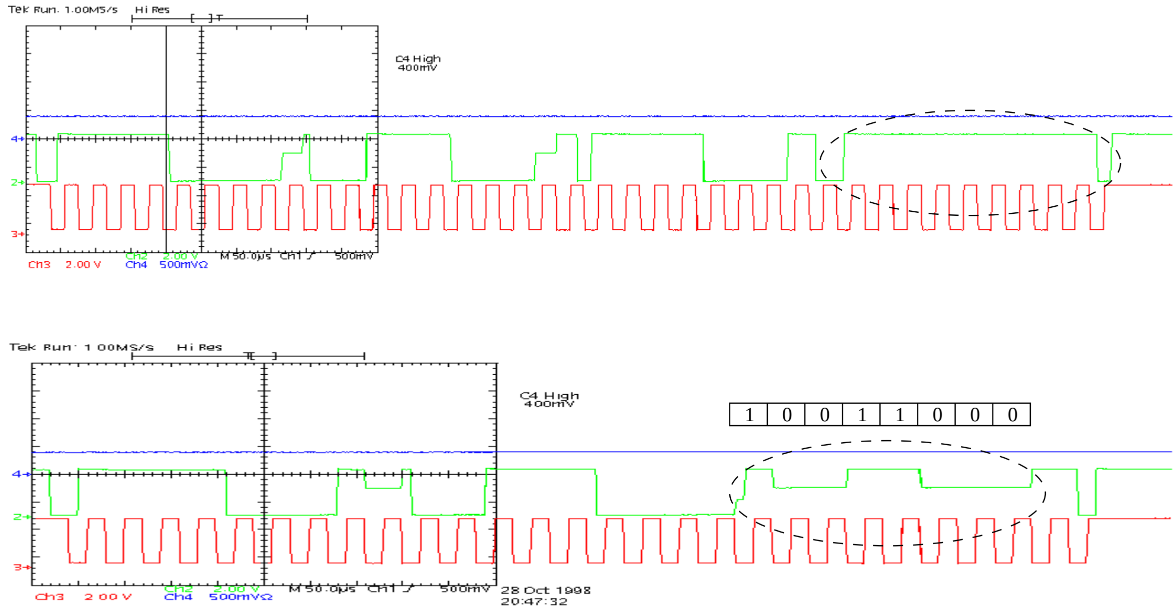

Abbildung 7.2: Fehlerhafte I^{2}C-Auslesesequenz (oben) und funktionsfähige Sequenz (unten) (aufgenommen mit dem *HP82000 Chiptester*) [SL98].

einer *Start*-Bedingung spezifiziert werden. Hier können also noch keine Daten oder Register-Adressen übertragen werden. Von den *General Call*-Protokollen kommt für den CIPix nur der sogenannte *Hardware General Call* in Frage.

Die notwendigen Modifikationen haben nur Auswirkungen auf die Steuereinheit *Ctrl* des Interfaces und die Dekoderlogik. Die Erweiterung der Zustandsmaschine ist im Zustandsdiagramm in Abb. 7.3 dargestellt. Es wurde ein neuer Zustand GCALL, sowie neue Verzweigungsvariablen (`Read10EN`, `gcall`, `hardgc`) eingeführt.

7.3 Korrekturen am Dekoder

Die Spezifikationen zum *Single Write Mode* sehen es vor, daß mehrere Register nacheinander über ihre Adresse angesprochen und beschrieben werden. Wird mehr als ein Register programmiert, arbeitet die Dekoderlogik jedoch fehlerhaft. Beim Schreib-Dekoder erfolgt im Zustand *SWM* eine unkorrekte Zuweisung in den nächsten Zustand.

In der durchgeführten Modifikation wird nun der Wechsel aus dem Zustand SWM in Abhängigkeit des Wertes des Bits `Data[4]` ($\equiv$ `lastframe`) des Adresswortes vollzogen. Beim letzten adressierten Register wird `Data[4]` auf 1 gesetzt. Dadurch erfolgt der Übergang in den Ausgangszustand SELECT. Ansonsten wird in den Zustand WMSEL gewechselt. Abb. 7.4 zeigt die modifizierte Version des Zustandsdiagramms.

7.4 Simulationen

Nachfolgend sollen Simulationen der korrigierten I^{2}C-Sequenzen präsentiert werden. Die Zahlen am Fuß der Graphiken geben die Simulationszeit in Mikrosekunden an. Die auftretenden

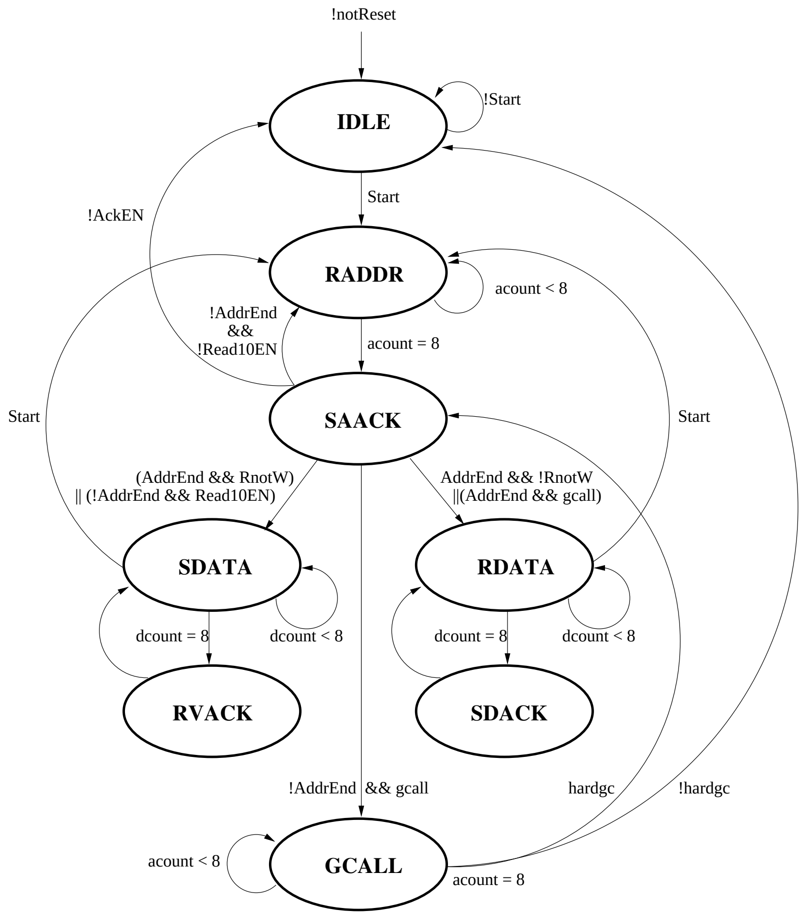

Abbildung 7.3: Erweitertes Zustandsdiagramm der Zustandsmaschine *Ctrl*.

Zahlenangaben sind in hexadezimalem Format. Ein X steht für einen unbestimmten Wert. Abb. 7.5 zeigt das Ergebnis einer Simulation der Auslesesequenz. Der Zeitbereich, in dem der auszulesenden Registerinhalt auf der SDA-Leitung anliegt, ist markiert. Abb. 7.6 zeigt ein Simulationsergebnis des *General Call*-Modus. Die *General Call*-Adresse und die Adresse des Hardware-Masters sind hervorgehoben. Die Sequenz setzt einen *Single Write Mode* um, bei dem zwei Register über ihre Adresse angesprochen und beschrieben werden.
Diese Sequenzen entsprechen den spezifizierten Protokollen des I^2C-Standards, sowie den spezifizierten Modi des CIPix.

7.5 Korrekturen an der Testpulssteuerlogik

Gibt man einen externen Testpuls auf das Pad *fcsTP*, so zeigt sich, daß die Einkopplung nur bei laufendem Takt *Sclk* funktioniert. Vielfach möchte man aber gerade den Takt abschalten, um so ein Übersprechen der Clock auf andere Leitungen zu untersuchen oder zu verhindern.

Die Testpulssteuerlogik arbeitet synchron zum Takt *Sclk*. Das Signal *MSEL*, das zwischen interner und externer Testpulseinkopplung selektiert, wird bei der aktuellen Chipversion innerhalb der Logik vom Register *TpReg* abgeleitet, wobei *MSEL* der Wert des LSBs von *TpReg*

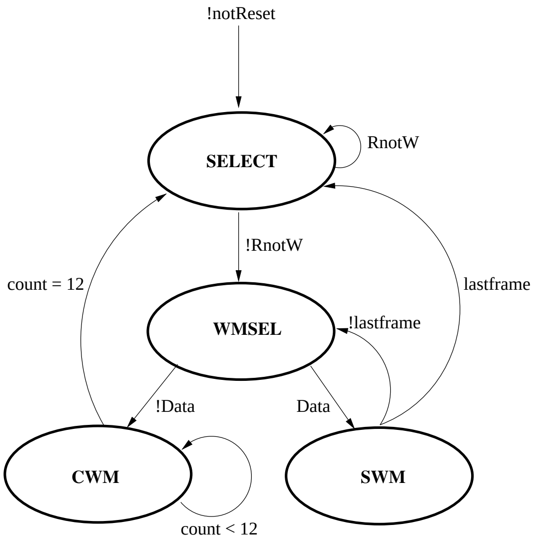

Abbildung 7.4: Erweitertes Zustandsdiagramm der Zustandsmaschine *DecoderWRITE*.

zugewiesen wird. Diese Implementierung hat den Nachteil, daß das Signal nur bei laufendem Takt gültig ist und einen definierten Wert besitzt.

In der modifizierten Version der Testpulslogik werden die Signale *MSEL* und *CSEL* außerhalb des Logikblockes direkt vom Register *TpReg* abgeleitet und sind somit unabhängig vom Takt *Sclk*.

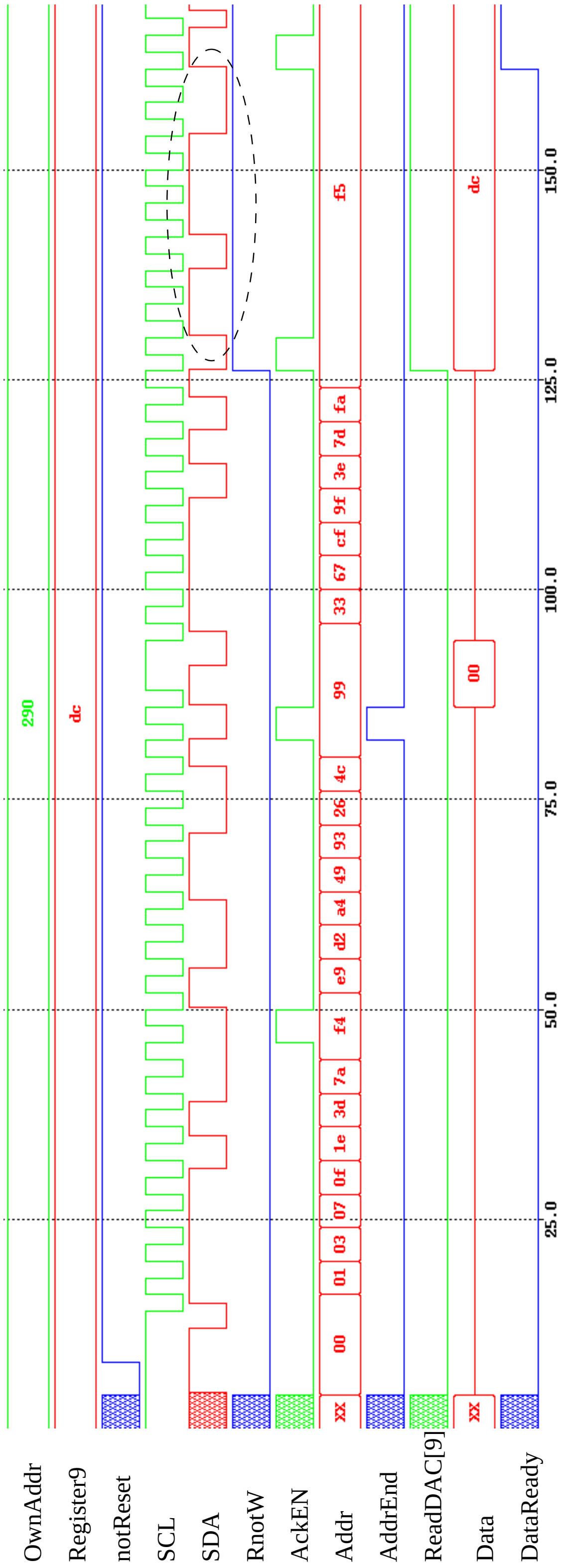

Abbildung 7.5: Simulation der korrigierten I²C-Auslesesequenz bei einer 10-Bit Adressierung.

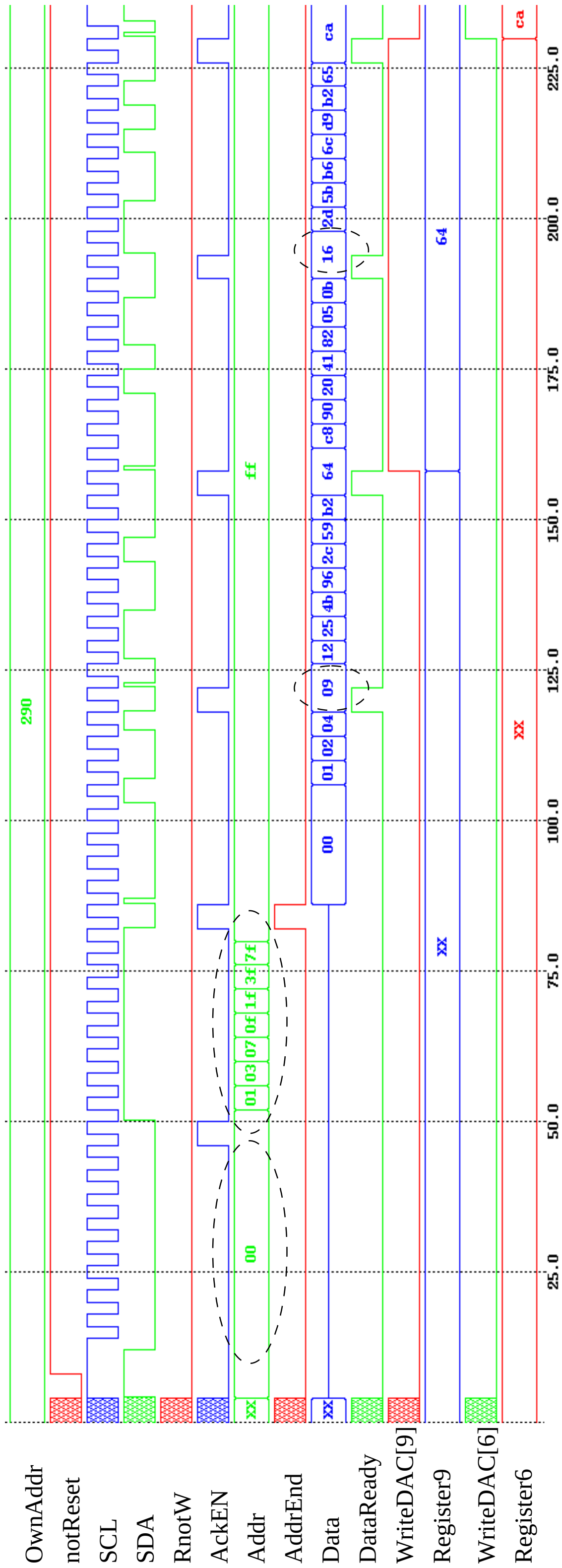

Abbildung 7.6: Simulation des korrigierten *General Call*-Modus.

7.6 Korrekturen an den LVDS-Receivern

Nach der Submission des Chips wurden Simulationen an den LVDS-Receivern durchgeführt, auf die aus Zeitgründen zuvor verzichtet werden mußte. Die Spezifikationen zum LVDS-Standard [lvds97] legen den Bereich der Offset-Spannung auf 1,125...1,375 V und den Bereich des Spannungs-Swings auf 247...454 mV fest. Abb. 7.7 stellt die Simulationsergebnisse graphisch dar. Die Offset-Spannung wurde bei einem konstanten Swing von 350 mV entlang einer Rampe von 0 bis 3,3 V verfahren. Rot dargestellt ist die Ausgangsspannung des Receivers. Man erkennt, daß dieser nur in einem Offset-Bereich von 0 bis ≈ 850mV schaltet. Das entspricht *nicht* den LVDS-Spezifikationen.
Die Schaltschwellen der Receiver sind also noch an die Nominalwerte anzupassen.

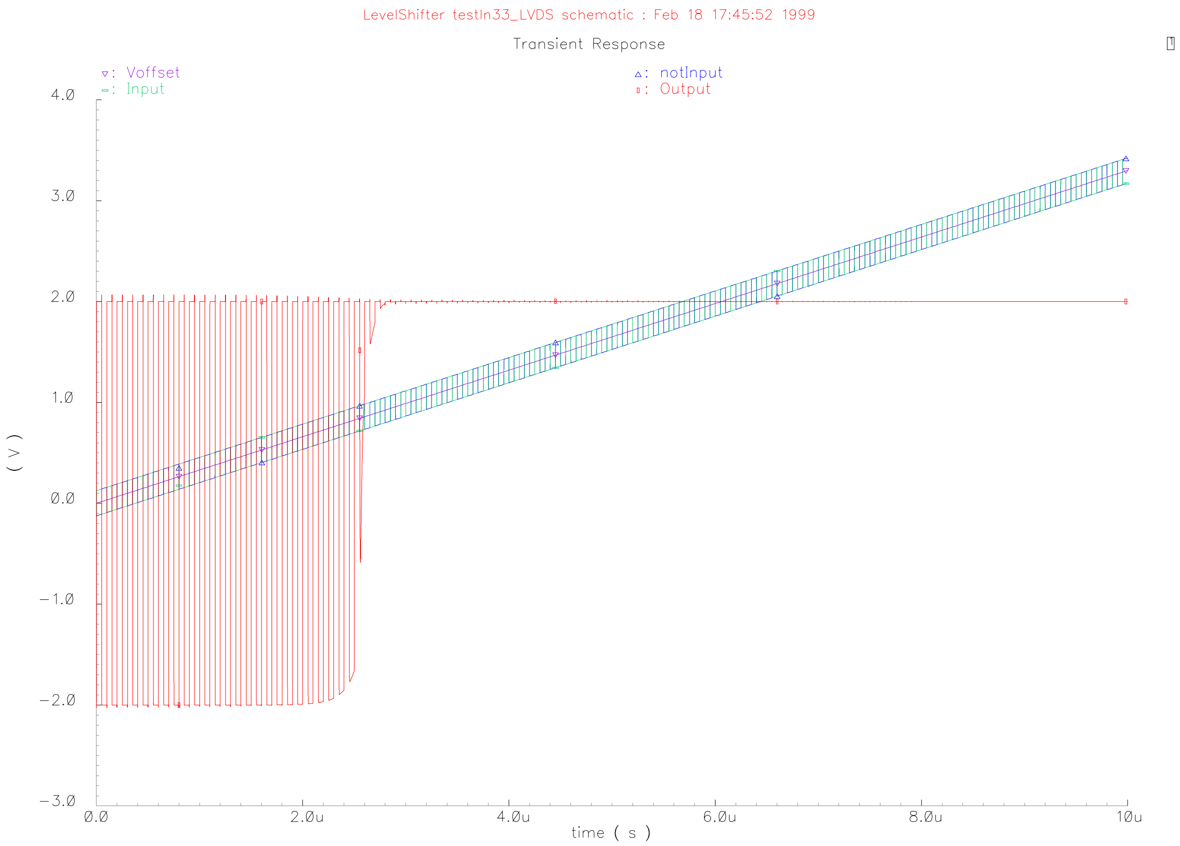

Abbildung 7.7: Simulationsergebnis der LVDS-Receiver. Die Offset-Spannung wird entlang einer Rampe von 0 bis 3,3V verfahren. Der Spannungsswing zwischen den differentiellen Signalen beträgt 350mV. Man erkennt deutlich, daß ab einer Offset-Spannung von etwa 850mV der Ausgang der Stufe nicht mehr schaltet.

Kapitel 8

Zusammenfassung

Ziel dieser Arbeit war die Entwicklung und Charakterisierung des neuen Auslesechips *CIPix* für die zentrale innere Proportionalkammer CIP des H1-Detektors. Der Schwerpunkt lag auf der Entwicklung der digitalen Komponenten. Zu diesen zählen ein serielles Interface, ein Dekoder, eine Logik zur Vergabe einer Adresse sowie zwei Steuerblöcke für den internen Testpulsgenerator und die Auskopplung eines analogen Signals.
Das serielle Interface ist über eine Standard-I^2C-Schnittstelle programmierbar. Die Register des Chips können rückgelesen werden, allerdings weicht die Auslesesequenz vom spezifizierten I^2C-Protokoll ab. Die notwendigen Korrekturen wurden durchgeführt und durch Simulationen die Funktionsfähigkeit gezeigt.
Die anderen Komponenten arbeiten, bis auf kleine Abweichungen von den spezifizierten Eigenschaften, einwandfrei.
Die wesentliche Funktionalität dieser ersten Version des Chips konnte in den Messungen nachgewiesen werden. Ein noch zu klärendes Problem bleibt die fehlerhafte Programmierung des Chips bei laufendem Takt *Sclk*. Wird das Clocksignal angelegt, verläuft die Programmierung in einer nicht reproduzierbaren Weise fehlerhaft. Ein externes Übersprechen kann ausgeschlossen werden.
Im jetzigen Meßaufbau ist man in der Anzahl der kontaktierbaren Chippads durch das verwendete Gehäuse limitiert. Für Präzissionstests sollte ein verbesserter Messaufbau entwickelt werden, bei dem alle Chipanschlüssse auf einem PCB kontaktierbar sind und der ein Übersprechen minimiert.
Eine neue Version des Chips wird die beschriebenen Korrekturen, sowie aus den weiteren Messungen gewonnene Erkenntnisse umsetzen. Außerdem wird ein Temperatursensor als neue Komponente integriert.

Anhang A

Pad-Beschreibung des CIPix1.0

An alle Pads[1] wurde eine Referenznummer vergeben. Die Nummerierung der periphären Pads erfolgt entgegen dem Uhrzeigersinn und beginnt in der oberen linken Ecke (mit den analogen Eingangspads links). Abb. A.1 zeigt die geometrische Lage der Pads und das Nummerierungsschema der Testpads im Innenbereich des Chips. In den nachfolgenden Tabellen werden die zu den Pads gehörenden Signale noch einmal zusammengestellt und erklärt. Wenn nicht anders vermerkt, sind die digitalen Ein- und Ausgangspegel CMOS-Pegel zwischen 0 und 3,3 V.

A.1 Pads vorne

Die analogen Eingangspads befinden sich an der Vorderseite des Chips. Sie sind zweifach gestaffelt (in zwei Reihen angeordnet) mit einem Pitch-Abstand von 100μm.

Ref.no.	Pin name	Type	Description
1	`AnalogIn<0>`	input	input of channel 0
2	`AnalogIn<1>`	input	input of channel 1
3	`AnalogIn<2>`	input	input of channel 2
⋮	⋮	⋮	⋮
64	`AnalogIn<63>`	input	input of channel 63

Tabelle A.1: Pads an der Vorderseite des CIPix1.0. Das erste Pad in er Tabelle entspricht dem obersten Pad auf der Vorderseite des Chips.

[1] Kontaktierungsfläche

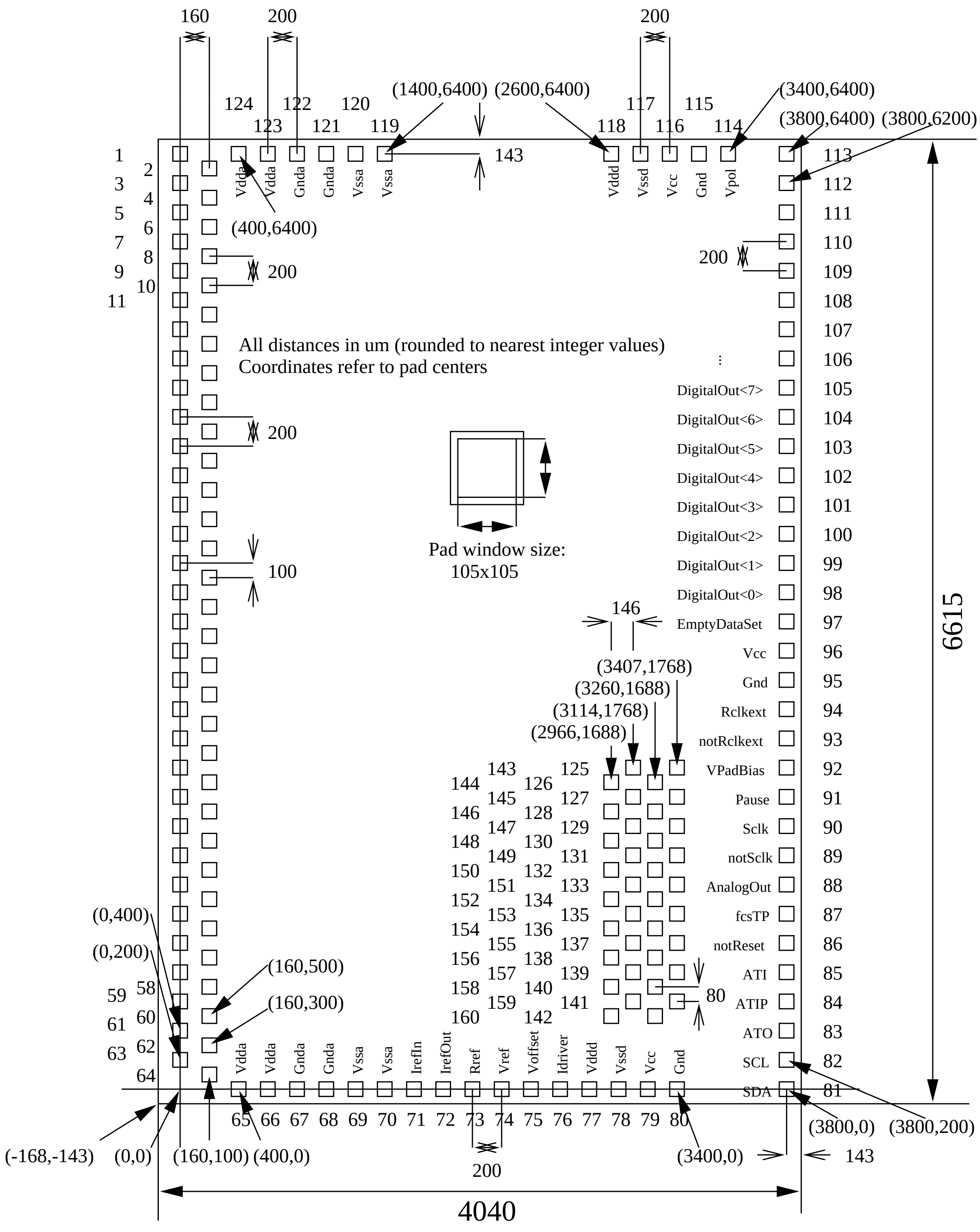

Abbildung A.1: Schematische Zeichnung der Padanordnung des CIPix1.0.

A.2 Pads unten

Die Pads an der Chip-Unterseite sind in einem Pitch-Abstand von 200 μm positioniert.

Ref.no.	Pin name	Type	Description
65	`Vdda`	supply	positiv analog supply voltage (+2V)
66	`Vdda`	supply	positiv analog supply voltage (+2V)
67	`Gnda`	supply	analog ground (0V)
68	`Gnda`	supply	analog ground (0V)
69	`Vssa`	supply	negative analog supply voltage (-2V)
70	`Vssa`	supply	negative analog supply voltage (-2V)
71	`IrefIn`	input	reference current input for internal current DAC: may either be connected to an external reference current source or to the `IRefOut` pin, if internal reference current source is to be used.
72	`IrefOut`	output	output of internal reference current source
73	`Rref`	output	to be connected to external resistor (20kΩ to -2V) if internal reference current source is used.
74	`Vref`	blocking output	to be connected to external blocking capacitor (100nF to gnd)
75	`Voffset`	blocking output	to be connected to external blocking capacitor (100nF to gnd)
76	`Idriver`	blocking output	to be connected to external blocking capacitor (100nF to gnd)
77	`Vddd`	supply	positiv digital supply voltage (+2V)
78	`Vssd`	supply	negative digital supply voltage (-2V)
79	`Vcc`	supply	digital I/O power supply (3.3V)
80	`Gnd`	supply	digital ground (0V)

Tabelle A.2: Pads an der Unterseite des CIPix1.0

A.3 Pads hinten

Die Pads an der Rückseite des Chips haben einen Pitch-Abstand von 200 μm.

Ref.no.	Pin name	Type	Description
81	`SDA`	input/output	I^2C-bus serial data input/output
82	`SCL`	input/output	I^2C-bus serial clock input/output
83	`ATO`	output	token output for addressing procedure
84	`ATIP`	input	token input for addressing procedure
85	`ATI`	input	token input for addressing procedure
86	`notReset`	input	active low power-on reset
87	`fcsTP`	input	digital test pulse input: the rising edge signals moment of charge injection
88	`AnalogOut`	output	serial analog output
89	`notSclk`	LVDS-input	active low sampling clock
90	`Sclk`	LVDS-input	active high sampling clock (i.e. the bunch-crossing clock of 10.4 MHz)
91	`Pause`	test input	I^2C-bus input forcing the bus-master into an wait state
92	`VPadBias`	input/output	external reference voltage for I/O-pads; to be used, if a voltage greater than the internally generated one (2.5V) is needed.
93	`notRclkext`	LVDS-input	inverted readout clock for data multiplexer
94	`Rclkext`	LVDS-input	active high readout clock for data multiplexer (41.6MHz i.e. 4-fold Sclk)
95	`Gnd`	supply	digital ground (0V)
96	`Vcc`	supply	digital I/O power supply (3.3V)
97	`EmptyDataSet`	output	active-low control signal for external multiplexer, indicating if there are no input signals on the 64 input pads. It is generated as an OR of all input channels.
98	`DigitalOut<0>`	output	digital output of channels 0, 16, 32 and 48
99 - 112	⋮	⋮	⋮
113	`DigitalOut<15>`	output	digital output of channels 15, 31, 47 and 63

Tabelle A.3: Pads an der Rückseite des CIPix1.0. Das erste Pad in der Tabelle entspricht dem untersten Pads auf der Rückseite des Chips.

A.4 Pads oben

Die Pads an der Oberseite des Chips haben einen Pitch-Abstand von 200 μm.

Ref.no.	Pin name	Type	Description
114	`Vpol`	input	polarity switch for comparator; 0V: detection of negative signals; 3.3V: detection of positive signals
115	`Gnd`	supply	digital ground (0V)
116	`Vcc`	supply	digital I/O power supply (3.3V)
117	`Vssd`	supply	negative digital supply voltage (-2V)
118	`Vddd`	supply	positiv digital supply voltage (+2V)
119	`Vssa`	supply	negative analog supply voltage (-2V)
120	`Vssa`	supply	negative analog supply voltage (-2V)
121	`Gnda`	supply	analog ground (0V)
122	`Gnda`	supply	analog ground (0V)
123	`Vdda`	supply	positiv analog supply voltage (+2V)
124	`Vdda`	supply	positiv analog supply voltage (+2V)

Tabelle A.4: Pads auf der Oberseite des CIPix1.0

A.5 Testpads innen

Die Pads im Innenbereich des Chips wurden für Test- und Diagnosezwecke implementiert und sollen in zukünftigen Chipversionen nicht mehr vorhanden sein. Sie haben einen Pitch-Abstand von 80 μm, wobei die Padspalten ungefähr 146 μm Abstand voneinander haben.

Ref.no.	Pin name	Type	Description
125	data<0>	test input/output	internal data bus
126	data<1>	test input/output	"
127	data<2>	test input/output	"
128	data<3>	test input/output	"
129	data<4>	test input/output	"
130	data<5>	test input/output	"
131	data<6>	test input/output	"
132	data<7>	test input/output	"
133	write<1>	test input/output	write enable for DAC register
134	write<2>	test input/output	"
135	write<3>	test input/output	"
136	write<4>	test input/output	"
137	write<5>	test input/output	"
138	write<6>	test input/output	"
139	write<7>	test input/output	"
140	write<8>	test input/output	"
141	write<9>	test input/output	"
142	read<8>	test input/output	read enable for DAC register
143	Vfp	test output	preamplifier *Vfp* voltage
144	Vfs	test output	shaper *Vfs* voltage
145	Ipre	test output	preamplifier bias current *Ipre*
146	Isha	test output	shaper bias current *Isha*
147	Ibuf	test output	buffer bias current *Ibuf*
148	Icomp	test output	comparator bias current *Icomp*
149	VPadRef	test input/output	external reference voltage (1.65V) for I/O-pads
150	ATIPint	test input	token input for addressing procedure; connected internally to ATIP
151	ATIint	test input	token input for addressing procedure; connected internally to ATI
152	ATOint	test output	token output for addressing procedure; connected internally to ATO
153	SDAI	test input	I^2C-bus serial data input; connected internally to SDA
154	SDAO	test output	I^2C-bus serial data output; connected internally to SDA
155	SCLI	test input	I^2C-bus serial clock input; connected internally to SCL
156	SCLO	test output	I^2C-bus serial clock output; connected internally to SCL
157	notResetint	test input	active low power-on reset; connected internally to notReset
158	fcsTPint	test input	digital test pulse input; connected internally to fcsTP
159	Rclkint	test input/output	active high readout clock for data multiplexer; connected internally to Rclk
160	Sclkint	test input/output	active high sampling clock; connected internally to Sclk

Tabelle A.5: Beschreibung der inneren Testpads des CIPix. Die Nummerierung beginnt in der oberen rechten Ecke des Blockes (siehe Abb. A.1).

Anhang B

Externe Beschaltung des CIPix1.0

Abb. B zeigt den Schaltplan für die externe Bestückung des CIPix mit Kapazitäten. Diese dienen dazu, Störungen auf den Spannungszuführungen zu unterdrücken. Bei Verwendung der internen Stromquelle (siehe Abschnitt 3.8) muß ein 20 kΩ Widerstand vom Pad *Rref* gegen *Vssa* (-2 V) gelegt werden und das Ausgangspad *IrefOut* direkt mit dem Eingangspad *IrefIn* verbunden werden.
Bei Messungen wurde eine Oszillation der über dem Referenzwiderstand abfallenden Spannung festgestellt. Die parallel zu dem 20 kΩ Widerstand geschaltete Blockkapazität von 10nF reduziert die Frequenz dieser Oszillation um einen Faktor 100.

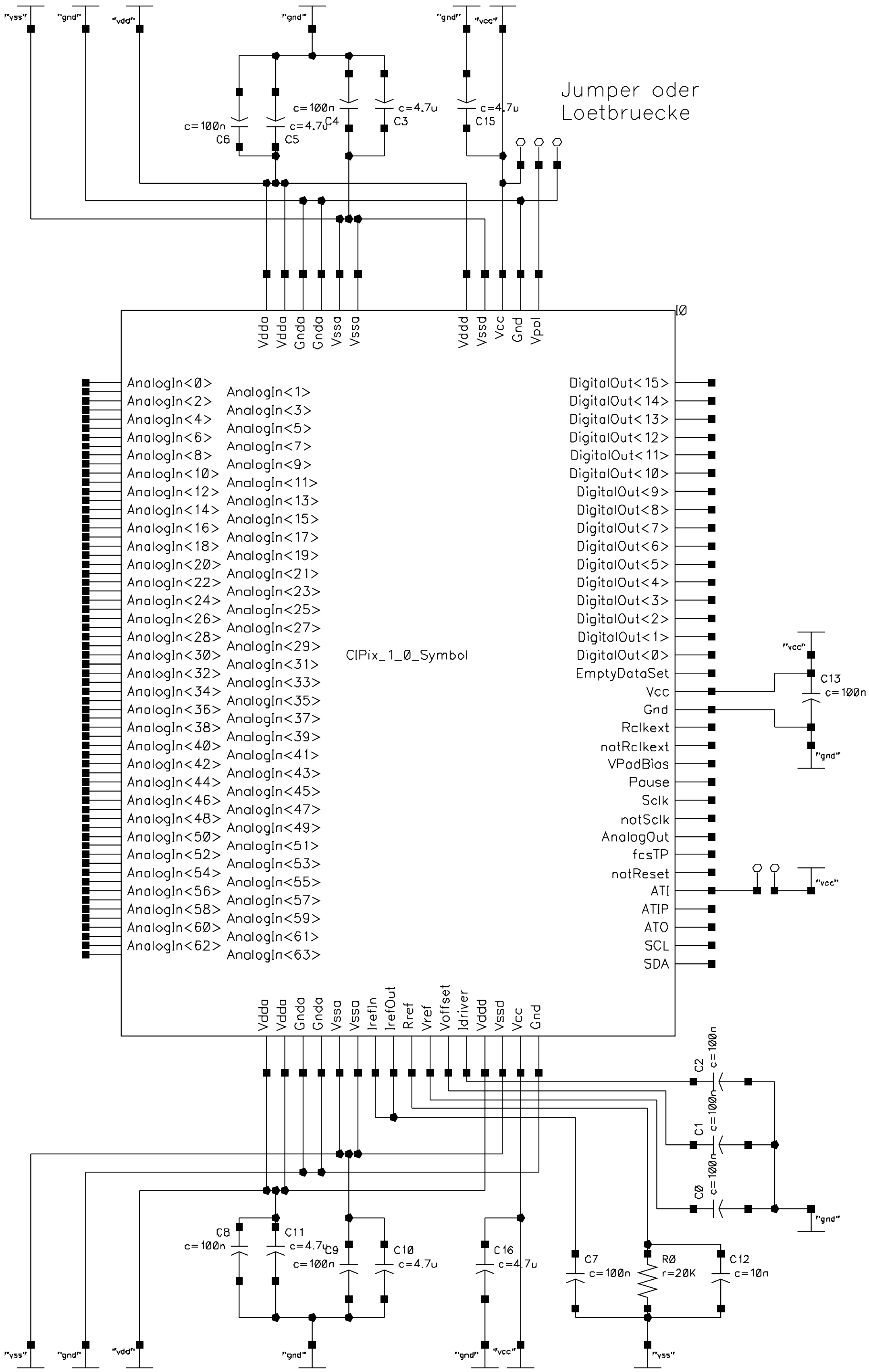

Abbildung B.1: Schaltplan für die externe Beschaltung des CIPix1.0.

Literaturverzeichnis

[AMS] Austria Mikro Systeme, *Digital Design Application Note*

[BG97] B.Glass, *Analoge Auslese- und Triggerelektronik für Mikrostreifen-Gaszähler*, Diplomarbeit, Universität Heidelberg, Physikalisches Institut, ASIC-Labor, 1997

[Cad] Cadence Design Systems, Inc.

[Cal98a] Calibre UK, *Parallel I2C Communications Adapter*, User Manual, 1998

[Cal98b] Calibre UK, *Parallel I2C Communications Adapter*, Driver Manual, 1998

[CIP98a] D. Baumeister, M. Feuerstack-Raible, S. Löchner: *Preliminary Specification of CIPix*, ASIC-Labor, CIP-Upgrade 2000, Version 1.0, April 1998

[CIP98b] D. Baumeister, M. Feuerstack-Raible, S. Löchner: *Specification of CIPix*, ASIC-Labor, CIP-Upgrade 2000, Version 2.0, Juni 1998

[DW80] D. Winkler, *The art of digital design*, Prentice Hall, Inc., 1980

[ES97] E. Sexauer, *Charakterisierung des Helix128 Auslesechips für HERA-B*, Diplomarbeit, Universität Heidelberg, Max-Planck-Institut für Kernphysik Heidelberg, ASIC-Labor, 1997

[Gol95] U. Golze, *VLSI-Entwurf eines RISC-Prozessors*, Vieweg 1995

[H1] H1 WWW-Homepage, http://www-h1.desy.de

[H1 97a] H1 Collaboration: *ep Physics beyond 1999*. H1-10/97-531. Paper submitted to the representatives of the H1 Funding Agencies for their meeting on 11/24/97 at DESY, Hamburg, 1997

[H1 97b] H1 Collaboration: *The H1-Detector at HERA*. Nuclear Instruments and Methods in Physics Research, A386 (1997)

[H198] H1 High Luminosity Upgrade 2000, *Progress Report on CIP and Level 1 Vertex Trigger*, June 1998

[HB96] H. Beck, *Principles and Operation of the z-Vertex Trigger*, H1 Collaboration, H1-05/96-479 (1996)

[I^{2}C98] Philips Semiconductors: *The I^2C-bus and how to use it (including specifications)*, http://www-us2.semiconductors.philips.com/i2c/facts/#specification

[KM92] K.Müller et al.: *Construction and performance of a thin cylindrical multiwire proportional chamber with cathode pad readout for the H1-experiment*, Nuclear Instruments and Methods in Physics Research, 457-466, A312 (1992)

[lvds97] National Semiconductors, *LVDS Owner's Manual*, Design Guide, Spring 1997

[MFR98a] M. Feuerstack-Raible, *Hochintegrierte elektronische Schaltungen - Architekturen und Methoden*, Skript zur Vorlesung, 1998

[MFR98b] M. Feuerstack-Raible, *Entwicklung einer Eingangs- und Ausgangs-Levelshifterstufe*, Schaltplan und Layout, 1998

[MK98] M. Kollak, *Entwicklung einer Stripline-Auslese für Kathodensignale einer langen, zylindrischen Vieldrahtproportionalkammer*, Diplomarbeit, Universität Heidelberg, Physikalisches Institut, 1998

[Nic96] T. Nicholls et al., *Concept, Design and Performance of the Second Level Triggers of the H1 Detector*, 1997

[Pad98] *Pulsformmessungen am Helix2.0*, Universität Padua, http://axpze0.pd.it:8080/ZEUS/MVD/shape.html

[Phil97] Philips, *PCF8584 I^2C-Controller*, Product Specification, 1997

[Prop98] U. Straumann et al., *CIP and Level1 Vertex Trigger*, H1 High Luminosity Upgrade 2000, Proposal, 1998

[Sef92] F. Sefkow et al., *Experience with the First Level Trigger of H1*, H1 Internal Note, H1-11/94-407 (1992)

[SL98] S. Löchner, *Charakterisierung und Entwicklung eines CIP-Auslese-ASIC für das H1-Upgrade-Projekt 2000*, Diplomarbeit, Universität Heidelberg, Physikalisches Institut, 1998

[Syn97] Cadence Design Systems, Inc., *Synergy HDL Synthesizer and Optimizer Modeling Style Guide*, Online-Dokumentation, 1997

[TM96] D. Thomas, P. Moorby, *The Verilog Hardware Description Language*, Kluwer Academic Publishers, 3rd Edition, 1996

[US99] U. Straumann, *private Mitteilungen*, 1999

[UT98] U. Trunk, *private Mitteilungen*, 1998

[UT99] U. Trunk, Dissertation in Vorbereitung

[VXL97] Cadence Design Systems, Inc., *Verilog-XL Reference*, Online-Dokumentation, 1997

[WFB98] W. Fallot-Burghardt, *A CMOS Mixed-Signal Readout Chip for the Microstrip Detectors of HERA-B*, Dissertation, Universität Heidelberg, ASIC-Labor, 1998

Danksagung

An dieser Stelle möchte ich allen danken, die zum Gelingen meiner Diplomarbeit beigetragen haben:

Besonders Herrn Prof. Dr. Ulrich Straumann für die interessante Aufgabenstellung und die sehr angenehme Form der Betreuung.
Herrn Prof. Dr. Bock für die Übernahme der Zweitkorrektur.
Herrn Dr. Martin Feuerstack-Raible für die Betreuung im ASIC-Labor.
Herrn Sven Löchner für die intensive Zusammenarbeit, vor allem für seine sehr große Hilfsbereitschaft bei Problemen und Fragen jeglicher Art.
Herrn Harald Deppe für die vielen Gespräche, die mir über die ein oder andere Durststrecke hinweghalfen.
Herrn Edgar Sexauer für die vielen erfrischenden Kommentare und das Korrekturlesen der Arbeit.
Der CIP-Upgrade Gruppe: Dr. Katharina Müller, Michael Kollak und Andreas Schweitzer für die intensive Zusammenarbeit. Besonders danke ich Katharina Müller für das Korrekturlesen der Arbeit und die Bereitstellung von Literatur.
Allen Mitgliedern des ASIC-Labors für die sehr angenehme Arbeitsatmosphäre und die stetige Hilfsbereitschaft. Insbesondere danke ich Michael Keller und Johannes Schemmel, die mir bei Rechnerproblemen weiterhalfen.
Mein ganz spezieller Dank gilt Katja Schmitt für ihre endlose Geduld während der sehr arbeitsintensiven Phasen dieser Arbeit und ihre Unterstützung und Aufbauarbeit.
Meiner Mutter danke ich dafür, daß sie mir mein Studium ermöglichte und mich stets unterstützte.