

**Entwicklung eines Spill-Monitors zur
Untersuchung der SIS-Strahlstruktur im
Rahmen des HADES-Experiments**

II. Physikalisches Institut

Justus-Liebig-Universität

Gießen

September 1998

Diplomarbeit

von

Marc-André Pleier

aus Lautertal, OT. Dirlammen / Hessen

Für meine Mutter

und meinen Vater.

Zusammenfassung

Im Rahmen dieser Arbeit wurde ein „Spillmonitor“ zur elektronischen Überwachung und Analyse der zeitlichen Strahlstruktur des **Schwerionensynchrotrons SIS** der **Gesellschaft für Schwerionenforschung GSI** entwickelt und getestet.

Dieses Gerät wird im Rahmen des **HADES (High Acceptance Di Electron Spectrometer)**-Detektorsystems eingesetzt werden. Mit den vom Spillmonitor aufgezeichneten Daten über die Strahlstruktur steht eine weitere Überwachungsmöglichkeit zur Verbesserung der Strahlqualität „online“ zur Verfügung. Weiterhin ermöglichen die Daten des Spillmonitors eine zusätzliche Ermittlung der Totzeit, die zur korrekten Auswertung der mit **HADES** gewonnenen Meßergebnisse (z.B. für die Bestimmung absoluter Wirkungsquerschnitte) erforderlich ist.

Die Umsetzung dieses Prototyp-Konzeptes erfolgte konform zum bestehenden Hardware-Konzept für den **HADES-Trigger**. Die erforderlichen logischen Funktionen wurden in verschiedenen programmierbaren Logikbausteinen realisiert. Damit ist die Kompatibilität zu den übrigen entwickelten Platinen des **HADES-Trigger**systems bezüglich Initialisierung etc. sichergestellt. Zudem wurde so ein hohes Maß an Flexibilität für eventuelle spätere Erweiterungen des Spillmonitor-Konzeptes erreicht.

Inhaltsverzeichnis

1	EINFÜHRUNG	9
2	DAS HADES-SPEKTROMETER	11
2.1	PHYSIKALISCHE MOTIVATION	11
2.2	DIE EINZELNEN DETEKTORKOMPONENTEN IM ÜBERBLICK	14
2.2.1	<i>Der Cherenkov-Detektor</i>	15
2.2.2	<i>Die Mini Drift Kammern</i>	15
2.2.3	<i>META</i>	16
2.2.4	<i>Das HADES Triggersystem</i>	16
2.2.5	<i>Der Start-Detektor</i>	17
3	DAS SCHWERIONENSYNCHROTRON SIS AN DER GSI	22
3.1	DIE BESCHLEUNIGERANLAGEN DER GSI.....	22
3.2	DIE STRAHLSTRUKTUR DES SIS	27
4	AUFGABENSTELLUNG UND KONZEPT FÜR DEN SPILLMONITOR	29
5	UMSETZUNG DES KONZEPTEES IM DESIGN DER PROTOTYPEN	31
5.1	DIE AUFSTECK-PLATINE	31
5.1.1	<i>Aufbau und Funktionsweise</i>	31
5.1.2	<i>ECL-Logik und deren Designgrundregeln</i>	35
5.1.3	<i>Impedanzbestimmung der Leiterbahnen der Aufsteck-Platine</i>	36
5.2	DIE HAUPT-PLATINE	38
5.2.1	<i>Aufbau der Haupt-Platine</i>	38
5.2.2	<i>Programmierbare Logikbausteine</i>	46
5.2.3	<i>Der digitale Signalprozessor (DSP)</i>	47
5.2.4	<i>Der VME-Bus</i>	48
6	INBETRIEBNAHME UND ERSTE TESTS	49
6.1	INBETRIEBNAHME DER HAUPTPLATINE	49
6.2	INBETRIEBNAHME DER AUFSTECKPLATINE	54
7	AUSBLICK	58
8	LITERATURVERZEICHNIS	59

9	ANHANG.....	61
9.1	ABEL-QUELLTEXTE.....	61
9.1.1	<i>VME-Interface</i>	61
9.1.2	<i>Switching-Matrix (Lattice 1 : Daten)</i>	63
9.1.3	<i>Switching-Matrix (Lattice 2 : Adressen)</i>	65
9.1.4	<i>State-Machine</i>	67
9.2	VERWENDETE C-PROGRAMME	70
9.2.1	<i>Ansteuerung des VME-Busses unter OS9</i>	70
9.2.2	<i>Speicher-Testprogramm 1</i>	72
9.2.3	<i>Speicher-Testprogramm 2</i>	72
9.3	SCHALTPLÄNE DES SPILLMONITOR-PROTOTYPEN	74
9.3.1	<i>Schaltpläne der Aufsteckplatine</i>	74
9.3.2	<i>Schaltpläne der Hauptplatine</i>	81
9.4	SIGNALLAGEN-LAYOUT DER PLATINEN DES SPILLMONITORS	91
9.4.1	<i>Aufsteckplatine</i>	91
9.4.2	<i>Hauptplatine</i>	93

1 Einführung

Bei dem in dieser Arbeit vorgestellten „Spillmonitor“ handelt es sich um ein Modul, welches die zeitliche Strahlstruktur des Schwerionensynchrotrons **SIS** an der Gesellschaft für Schwerionenforschung **GSI** im Rahmen des HADES-Experiments elektronisch überwachen und analysieren soll. Das **High Acceptance Di Electron Spectrometer HADES** [1] ist ein Spektrometer für Dileptonenpaare, die in Schwerionenkollisionen erzeugt werden. Die Bestimmung der invarianten Masse dieser Dileptonenpaare soll u.a. neue Erkenntnisse über die Eigenschaften von Vektormesonen in dichter Kernmaterie liefern.

Der zur Durchführung dieser Experimente benötigte Strahl von hochenergetischen Ionen mit Intensitäten von bis zu 10^8 Teilchen pro Sekunde wird vom SIS der GSI zur Verfügung gestellt. Da diese Strahlteilchen während der Extraktionsphasen des SIS jedoch zeitlich nicht homogen verteilt sind und immer wieder Intensitätsmaxima (sogenannte Spikes) auftreten, ist es aus den folgenden Gründen nötig, die Struktur dieses Ionenstrahls zu überwachen:

- Zum einen wird mit steigender Strahlintensität die Wahrscheinlichkeit für zwei Ereignisse, die vom Detektor nicht mehr voneinander getrennt werden können und somit verfälschte Ergebnisse liefern („pile-up“), immer größer.
- Zum anderen ist es nötig, die Strahlstruktur des für das Experiment verwendeten Ionenstrahls zu kennen, um die Totzeit des Detektorsystems bestimmen zu können. Das Datenaufnahmesystem von HADES ist in der Lage, alle $10\ \mu\text{s}$ ein Ereignis zu verarbeiten. Tritt jedoch innerhalb von $10\ \mu\text{s}$ nach einem Ereignis ein weiteres auf, so geht dieses Ereignis für die Datenaufnahme verloren, auch wenn die Detektoren es noch bei einem zeitlichen Abstand $> 1\ \mu\text{s}$ vom vorigen Ereignis trennen konnten. Zur Bestimmung von absoluten Wirkungsquerschnitten muß diese Totzeit des Systems bekannt sein, um die gemessenen Reaktionsraten richtig auszuwerten. Mit dem Spillmonitor ist eine zusätzliche Möglichkeit zur Bestimmung der Totzeit neben der Totzeit-Ermittlung durch die zentrale Trigger-Einheit gegeben.

Zur Berechnung der Totzeit wäre es auch ausreichend, die Strahlstruktur des Ionenstrahls aufzuzeichnen und später bei der Analyse der Daten mit einzurechnen. Effektiver ist es allerdings, die Struktur des Strahls schon während des Experiments zu überwachen und

„online“ zu analysieren. Damit ist es möglich, die gemessene Strahlqualität den Operateuren des SIS über Transport Control Protocol / Internet Protocol (TCP/IP) zu übermitteln und gegebenenfalls korrigieren zu lassen. Ziel ist es, einen möglichst konstanten Teilchenstrom in den extrahierten Spills zu erreichen, was gleichbedeutend mit minimalen zeitlichen Schwankungen der Raten und damit niedriger pile-up-Wahrscheinlichkeit bei maximaler Durchschnittsrate für eine gute Ereignisausbeute mit minimaler Totzeit ist.

Um die konkreten Randbedingungen und Anforderungen, die in das Konzept des Spillmonitors eingeflossen sind, verstehen zu können, wird in den folgenden zwei Kapiteln das Umfeld, in dem der Spillmonitor eingesetzt werden soll, näher vorgestellt:

Zunächst wird das HADES-Experiment mit seiner physikalischen Zielsetzung und den einzelnen Detektorkomponenten beschrieben. Besonders ausführlich wurde hierbei auf den Startdetektor (s. 2.2.5) eingegangen, der die Zählpulse, die der Spillmonitor verwendet, erzeugen wird.

Die zu erwartende Strahlstruktur basiert auf den grundlegenden Beschleuniger-Konzepten und deren konkreter Umsetzung an den Beschleunigeranlagen der GSI und wird im 3. Kapitel vorgestellt.

Die aus diesem Umfeld resultierenden Anforderungen bilden die Grundlage des im folgenden Kapitel dargestellten Konzeptes und dessen Realisierung in den einzelnen Funktionsgruppen des Spillmonitors.

Das 5. Kapitel zeigt dann die detaillierte Umsetzung der einzelnen Funktionsgruppen im Design des Prototypen mittels elektronischer Bauelemente, deren grundlegende Eigenschaften jeweils kurz erläutert werden.

In den nachstehenden beiden Kapiteln werden dann schließlich die Inbetriebnahme des Spillmonitor-Prototypen und erste Tests mit Bewertung und Ausblick beschrieben.

Die wichtigsten Schaltpläne und grundlegende Programme des Spillmonitors können im Anhang nachgeschlagen werden.

2 Das HADES-Spektrometer

2.1 Physikalische Motivation

Das **H**igh **A**cceptance **D**i **E**lectron Spectrometer **HADES** [1], das gegenwärtig am Schwerionensynchrotron SIS der GSI in Darmstadt aufgebaut wird, dient vor allem der Untersuchung von Veränderungen der Eigenschaften von Kernmaterie unter extremen Drücken und Temperaturen, wie sie in Schwerionen-Kollisionen erreicht werden. Mittels der unter diesen im Labor erzeugten Bedingungen gewonnenen Daten ist es möglich, weiteren Aufschluß über die Zustandsgleichung der Kernmaterie und Informationen über etwaige Veränderungen von Massen und Lebensdauern („In Medium Modifikationen“) zu erhalten. Das breitgefächerte physikalische Programm des HADES-Detektorsystems umfaßt weiterhin die Untersuchung von elektromagnetischen Formfaktoren von Baryonen und Mesonen, die mit Unterstützung eines Photonendetektors wie z.B. dem **T**wo **A**rm **P**hoton **S**pectrometer **TAPS** durchgeführt werden soll.

Die Untersuchung von Schwerionenreaktionen mittels leptonischer Sonden ist vor allem deshalb so aufschlußreich, weil die Leptonen, die in einer solchen Schwerionenreaktion entstehen, nicht der starken Wechselwirkung unterliegen und somit eine weitestgehend unverfälschte Information über die Kollision nach außen tragen. Als Informationsquellen über die Eigenschaften von Materie unter den extremen Bedingungen einer Schwerionenkollision, bei der Kerndichten von bis zu $3\rho_0$ ($\rho_0=0,17 \text{ fm}^{-3}$) und Temperaturen um 70-100 MeV für eine Dauer von etwa $10\text{fm}/c$ ($\approx 3,3 \cdot 10^{-23} \text{ s}$) erreicht werden (nach BUU-Modellrechnungen [2]), bieten sich besonders die dileptonischen Zerfallskanäle der leichten Vektormesonen ρ, ω, ϕ an:

Meson	Lebensdauer•c [fm]	Masse [MeV/c ²]	Breite [MeV/c ²]	bevorzugter Zerfall	Verzweigungs- verhältnis in e ⁺ e ⁻
ρ	1,3	770	151	$\pi\pi$	$4,5 \cdot 10^{-5}$
ω	23,4	782	8,41	$\pi^+\pi^-\pi^0$	$7,1 \cdot 10^{-5}$
ϕ	44,4	1019	4,43	K^+K^-	$3,0 \cdot 10^{-4}$

Tabelle 1 : grundlegende Eigenschaften der leichten Vektormesonen, siehe [3]

Aufgrund seiner extrem kurzen Lebensdauer von 1,3 fm/c ist das ρ -Meson besonders geeignet, Informationen über die kurzlebige heiße Phase der Kernmaterie zu liefern, da es fast ausschließlich eben dort zerfällt. Die ω - und ϕ -Mesonen hingegen werden aufgrund ihrer relativ zur heißen Phase großen Lebensdauer bevorzugt nicht mehr in dieser Region zerfallen.

Über die Rekonstruktion der invarianten Masse und Breite dieser Vektormesonen aus dem entsprechenden Dileptonenpaar ist es somit möglich, Rückschlüsse auf das Verhalten dieser Hadronen in Kernmaterie unter diesen extremen Bedingungen zu ziehen.

Mittels der so gewonnenen Daten wird es möglich sein, die vielen verschiedenen theoretischen Modelle, mit denen man versucht, die Kernmaterie zu beschreiben, auf ihre Aussagekraft hin zu überprüfen.

So wird zum Beispiel bei Berechnungen mit Baryonen und Mesonen als effektive Freiheitsgrade nach Hermann et al.[4] eine Verbreiterung des ρ -Mesons bei unveränderter Lage des Maximums der Massenverteilung mit zunehmender Materiedichte erwartet. Im Gegensatz hierzu geht man bei Berechnungen basierend auf QCD-Summenregeln davon aus, daß eine Verringerung der Masse des ρ -Mesons eintritt (Asakawa et al.[5]): Da in der QCD die Masse der Konstituenten-Quarks aufgrund der Wechselwirkung mit dem chiralen Kondensat dynamisch erzeugt wird, bei zunehmender Baryondichte aber eine Abnahme dieses Kondensates bis hin zum chiralen Limit $\langle \bar{q}q \rangle = 0$ erwartet wird, sollte die (partielle) Wiederherstellung der chiralen Symmetrie mit zunehmender Dichte eine Verschiebung der Mesonenmasse zu kleineren Massen hin bewirken.

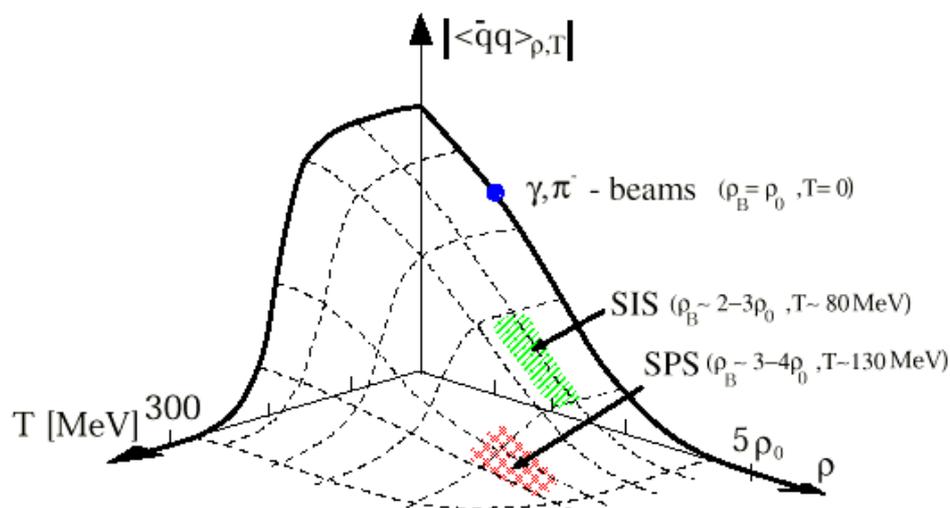


Abbildung 1: Erwartungswert des chiralen Kondensates in Abhängigkeit von Temperatur und Baryondichte basierend auf Berechnungen mit dem Nambu Jona-Lasinio-Modell [6]. Die dem SIS bzw. SPS zugänglichen Bereiche sind hervorgehoben.

Die Vielzahl der theoretischen Beschreibungen der Eigenschaften von Kernmaterie muß nun am Verhalten der Natur überprüft werden, wozu das HADES-Spektrometer einen wichtigen Beitrag liefern wird.

Die erwarteten Eigenschaften des HADES-Detektorsystems wurden mit umfangreichen Monte-Carlo Simulationen untersucht [1],[7]. Abbildung 2 zeigt deren Ergebnis für das Dileptonen-Massenspektrum bei zentralen Au-Au-Stößen bei 1 AGeV:

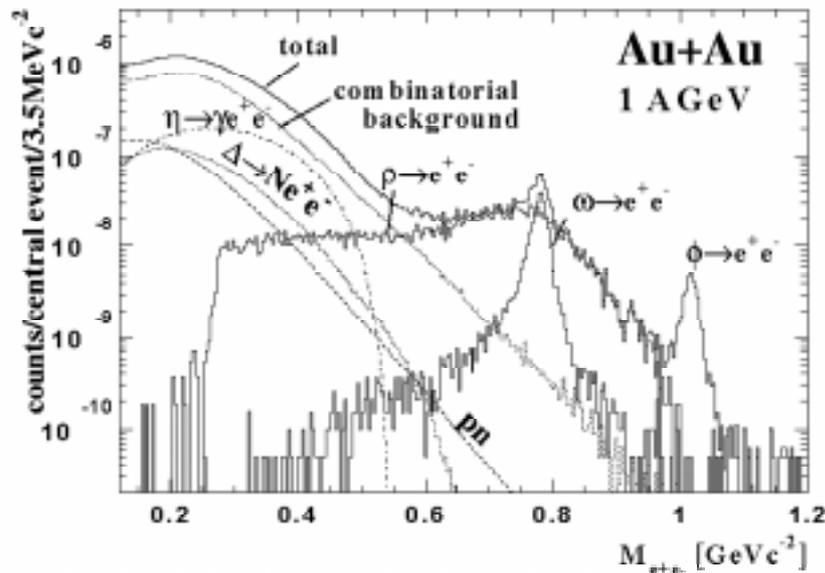


Abbildung 2 : Erwartetes Dileptonenspektrum für eine zentrale 1 AGeV Au+Au-Kollision [1]. Das Spektrum wird unterhalb von 500 MeV/c² hauptsächlich von den η- und Δ-Dalitz-Zerfällen sowie der pn-Bremsstrahlung bestimmt, wohingegen über 500 MeV/c² die Dileptonenzerfälle der leichten Vektormesonen ρ, ω und φ überwiegen.

Oberhalb von 500 MeV/c² dominieren die Dileptonenzerfälle der leichten Vektormesonen ρ, ω und φ, wohingegen unterhalb von 500 MeV/c² die Dalitz-Zerfälle von η und Δ sowie Nukleon-Nukleon-Bremsstrahlung und π⁺π⁻-Annihilation die wesentlichen Beiträge zum Spektrum liefern. Eine Vermessung des Beitrags der Bremsstrahlung könnte wiederum erstmals experimentellen Zugang zu zeitartigen elektromagnetischen Formfaktoren von Nukleonen ermöglichen.

Um aussagekräftige Messungen erzielen zu können, wurde das HADES-Spektrometer für eine hohe Untergrund-Unterdrückung und Massenauflösung konzipiert. Bei einer Massenauflösung von unter 1% (0,8% in der ρ, ω-Region => Δm < 10 MeV/c²) wird erwartet, ρ- und ω-Meson eindeutig voneinander trennen und auch evtl. auftretende Massenverschiebungen oder Verbreiterungen nachweisen zu können. Mit diesen verbesserten Eigenschaften und einer Akzeptanz von ca. 40% wird HADES auch als Experiment der zweiten Generation bezeichnet (zum Vergleich: das Di-Lepton-Spektrometer DLS [8] am BEVALAC Berkeley, USA, 1983-1993 besaß nur eine Massenauflösung von 12 % und eine Akzeptanz von ca. 1%).

Weitere Experimente, die das hohe Auflösungsvermögen von HADES ausnutzen, sind bereits in der Planung. So ist z.B. vorgesehen, den Zerfall von ω-Mesonen zu messen, welche in

pion-induzierten Kernreaktionen erzeugt werden. Hierbei werden Impuls und Energie des Pionenstrahls so gewählt, daß eine rückstoßfreie Produktion der ω – Mesonen ermöglicht wird. Die rückstoßfreie Produktion ist erforderlich, um zu gewährleisten, daß das ω -Meson trotz seiner im Vergleich zum ρ -Meson sehr großen Lebensdauer im Kern zerfällt und so Aufschluß über In-Medium-Modifikationen bei normaler Kerndichte gibt.

2.2 Die einzelnen Detektorkomponenten im Überblick

Um die im vorigen Abschnitt erläuterten Anforderungen an das Detektorsystem zu erfüllen, besteht HADES aus verschiedenen Komponenten, die alle rotationssymmetrisch um die Strahlachse angeordnet sind. Jede Detektorkomponente ist dabei in sechs identische Segmente unterteilt, die jeweils 60° abdecken. Unter Berücksichtigung der relativistischen Kinematik wird für typische Ereignisse eine effektive geometrische Akzeptanz von nahezu 2π erreicht (voller Azimutalwinkel und Polarwinkel zwischen 18 und 85 Grad).

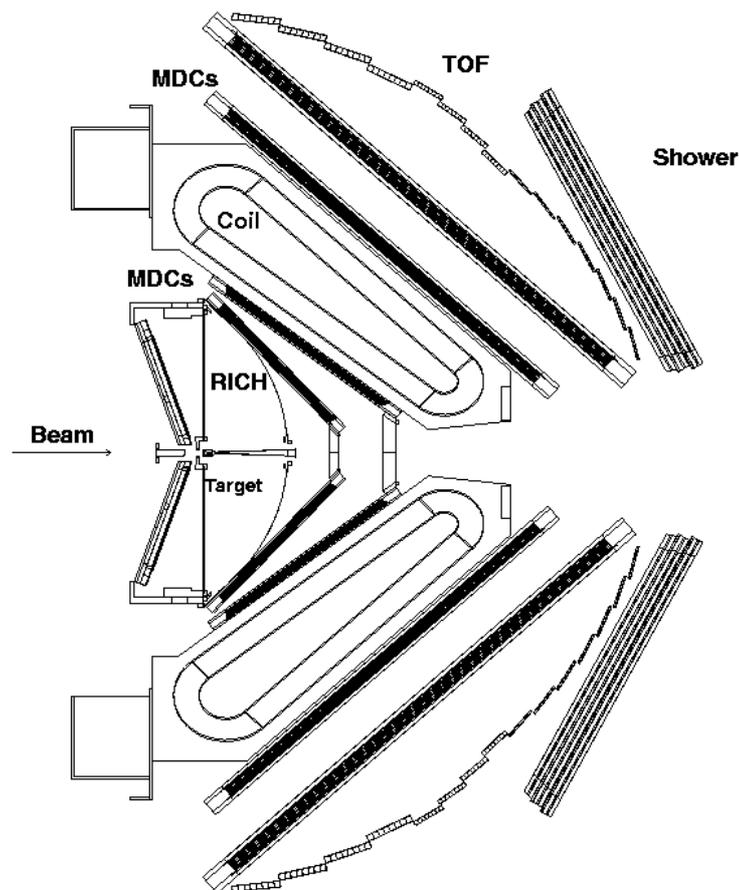


Abbildung 3: Schnitt durch das HADES-Spektrometer entlang der Strahlachse (**Beam**). In Strahlrichtung sieht man zunächst den ringabbildenden Cherenkov-Detektor (**RICH**), je 2 Mini-Drift-Chambers (**MDC**) vor und hinter den supraleitenden toroidalen Magnetspulen (**COIL**) und schließlich die Flugzeitwand (**TOF**) mit Schauerdetektor (**Shower**).

Bei der Messung mit HADES erfolgt die Teilchenidentifikation durch die Detektorkomponenten RICH und META; über die Vermessung der Teilchenspuren im Feld des supraleitenden toroidalen Magneten („Tracking“) mittels der MDCs kann schließlich der Impuls der Teilchen bestimmt werden. Diese verschiedenen Detektorkomponenten sollen nun kurz näher vorgestellt werden:

2.2.1 Der Cherenkov-Detektor

Der **Ring Imaging Cherenkov-Detektor RICH** dient der Elektronenidentifizierung. Um bei den für zentrale Kollisionen erwarteten Teilchenmultiplizitäten von bis zu 200 geladenen Teilchen die Dileptonenpaare herausfiltern zu können, muß der RICH hinreichend hadronenblind sein. Mit dem Radiatorgas Perfluorbutan (C_4F_{10}) ist dies gewährleistet, da es mit einer Schwelle von $\gamma_{thr} \approx 18,3$ für das Einsetzen von Cherenkov-Licht bei SIS-Energien kaum Hadronen geben wird, die diese Schwelle überschreiten.

Neben dem Gasradiator setzt sich der RICH aus einem sphärischen VUV-Spiegel und einem UV-Photonendetektor zusammen. Die auf einem Kegel um das geladene Teilchen emittierten Cherenkov-Photonen werden durch den sphärischen Spiegel unter Rückwärtswinkeln als Ringe auf den Photonendetektor reflektiert, der durch ein CaF_2 -Fenster vom Radiatorgas getrennt ist und aus einer Vieldrahtproportionalkammer und einer segmentierten CsI Photokathode besteht.

Da sich somit der Photonendetektor strahlaufwärts vom Target befindet, wird Untergrund im Photonendetektor durch stark ionisierende geladene Teilchen, die aus der Reaktionszone kommen, minimiert.

2.2.2 Die Mini Drift Kammern

Mit den je zwei Minidriftkammern vor und hinter dem Magnetfeld der supraleitenden toroidalen Magnetspulen können Teilchenspur und somit Impuls des Teilchens gemessen werden. Diese Impulsinformation dient dann der Teilchenidentifikation und zur Bestimmung der invarianten Masse. Wie bereits erwähnt, liegt die angestrebte Impulsauflösung bei $\Delta p/p \approx 1\%$, was mit der daraus resultierenden Massenauflösung von $\Delta m \leq 10 \text{ MeV}/c^2$ eine Trennung der Vektormesonen ω, ρ voneinander ermöglicht.

2.2.3 META

Die Kombination von Flugzeitwand und Schauerdetektor wird als **Multiplicity Electron Trigger Array META** bezeichnet. Er ermöglicht eine zweite Elektronenidentifikation über Flugzeitmessung bei großen Polarwinkeln bzw. über Flugzeitmessung und Schaueridentifizierung von Elektronen bei kleinen Polarwinkeln (bei Vorwärtswinkeln unter 45° können die Elektronen über Flugzeitdifferenzen nicht mehr völlig von Protonen und Pionen getrennt werden). Die Flugzeitwand **TOF** besteht aus 1056 Szintillatoren, jeweils mit einer Länge bzw. Breite zwischen 40 und 237 cm bzw. 1 und 3 cm. Der Schauer-Detektor ist in "Sandwich"-Form aus drei Drahtkammern mit Padauslese und zwei Schichten von Bleikonvertern aufgebaut. Die Dicke der Bleikonverter ist auf eine hohe Wahrscheinlichkeit für die Ausbildung elektromagnetischer Schauer im Bereich der gegebenen Elektronenenergien bei gleichzeitig minimaler hadronischer Schauerbildung hin optimiert. Die Drahtkammern arbeiten im sogenannten „Self Quenching Streamer Mode“, was gewährleistet, daß die in den Kammern erzeugte Ladung fast vollständig unabhängig vom Energieverlust der die Kammer durchdringenden Teilchen und damit proportional zur Anzahl der Teilchen ist.

Über Multiplizitätsermittlung der geladenen Teilchen liefert META zudem einen schnellen ersten Trigger für zentrale Ereignisse. Die gesamten Daten aus Schauer- und TOF-Detektor werden anschließend auch für die zweite Triggerentscheidung verwendet.

2.2.4 Das HADES Triggersystem

Um die beschriebenen physikalischen Ziele erreichen zu können, muß das HADES-Spektrometer verschiedene Anforderungen erfüllen:

Zum einen muß das Detektorsystem einen möglichst großen Raumwinkel abdecken, um die entstehenden Dileptonen mit maximaler Akzeptanz zu registrieren, denn aufgrund der kleinen Produktionsquerschnitte und der schwachen Verzweigungsverhältnisse der leichten Vektormesonen in Dileptonen werden nur wenige relevante Dileptonenpaare erzeugt.

Zum anderen muß das HADES Spektrometer in der Lage sein, hohe Strahlintensitäten zu verarbeiten, um eine ausreichende Statistik der zu untersuchenden Zerfälle zu erhalten.

Das HADES-Detektorsystem wird daher vom SIS in den Extraktionsphasen mit Strahlraten von bis zu 10^8 pps (particles per second) versorgt werden. Bei einem 1%-wechselwirkenden Target ist folglich mit 10^6 Stößen pro Sekunde zu rechnen, von denen 1/10 als zentral erwartet werden. Durch die niedrige Produktionswahrscheinlichkeit der Vektormesonen und die

Unterdrückung des dileptonischen Zerfallskanals um etwa den Faktor α^2 ($\alpha = 1/137$) werden etwa drei Zerfälle von Vektormesonen in Dileptonen innerhalb von 10 Sekunden erwartet.

Um bei derartig hohen Strahlintensitäten die wenigen relevanten Ereignisse herauszufinden, ist ein sehr effizientes Triggersystem nötig, um die Datenraten auf ein verarbeitbares Maß (angestrebt sind 100 Ereignisse pro Sekunde) zu reduzieren.

Das Triggersystem von HADES ist in drei Stufen aufgebaut:

1. Die erste Stufe (1st level trigger) dient dem Herausfiltern der zentralen Ereignisse über deren Multiplizität (in der Flugzeitwand). Hier wird eine Reduktion von 10^6 auf 10^5 Ereignisse pro Sekunde in der „Spill On“-Phase erwartet.
2. Die zweite Triggerentscheidung wird aus den Daten von RICH und META gewonnen. Zunächst werden einzelne Leptonen-Kandidaten über die Informationen aus RICH und META identifiziert und anschließend in der „Matching Unit“ hieraus die für HADES relevanten Paare herausgefiltert. Hierbei soll die Datenrate auf 10^3 reduziert werden.
3. Als letzte Triggerstufe ist der 3rd level Trigger vorgesehen, dessen Algorithmus die MDC Daten zur Eliminierung von Konversionspaar-Ereignissen benutzen wird. Er soll eine Reduktion der Datenrate auf 10^2 Ereignisse pro Sekunde ermöglichen.

Um den erforderlichen Datendurchsatz bis zur Reduktion der Ereignisraten auf 100 Ereignisse pro Sekunde zu gewährleisten, müssen alle Ereignisse vorher in einer Pipelinearchitektur gespeichert werden, bis die entsprechende Triggerentscheidung gefallen ist und die gespeicherten Ereignisse entweder verworfen werden oder in die nächste Entscheidungsstufe aufrücken. Die erforderliche schnelle Verarbeitung der Daten erfolgt durch massiv parallelisierte Prozesse.

2.2.5 Der Start-Detektor

Der Startdetektor weist im Idealfall jedes ihn passierende Strahlteilchen nach und liefert sowohl die Startsignale für die „Time of Flight Messung“ (TOF) im META als auch die Zählpulse für den Spillmonitor, der die zeitliche Struktur des Ionenstrahls überwachen und analysieren wird.

Der Start-Detektor für HADES muß nicht nur mit hohen Strahlintensitäten, sondern auch mit den unterschiedlichsten Projektilen arbeiten können (wie z.B. Pionen, Protonen, Schwerionen von Kohlenstoff bis Uran) sowie unabhängig vom jeweiligen Reaktionstyp reagieren, d.h. ein „Minimum Bias“-Detektor sein.

Prinzipiell gibt es zwei Möglichkeiten, einen Startzähler zu konstruieren:

1. Über einen *Reaktionstrigger*, der die Fragmente der Reaktion nachweist und somit bei einem 1% wechselwirkenden Target nur noch mit 10^6 statt mit 10^8 Signalen pro Sekunde arbeiten müßte. Allerdings wäre es erforderlich, einen derartigen Detektor im Bereich des Targets zu plazieren, was den Nachweis der Reaktionsprodukte um so mehr erschwert, je mehr Raumwinkel dieser Detektor abdeckt. Minimiert man wiederum den abgedeckten Raumwinkel und wählt nur einen bestimmten Bereich aus, so ist es wahrscheinlicher, nur selektiv auf bestimmte Ereignistypen zu reagieren.
2. Mit einem „*In-Beam*“ *Start-Detektor*, der dann allerdings der hohen Strahlenbelastung gewachsen sein müßte und nur geringen Untergrund produzieren dürfte, um das eigentliche Experiment nicht zu behindern. Diese Detektorart bevorzugt dafür keinerlei Reaktionstyp und muß nicht im Targetbereich aufgestellt werden.

Für das HADES-Projekt hat man sich für letztere Möglichkeit entschieden, insbesondere weil man bei der GSI bereits Erfahrungen mit polykristallinen **C**hemical **V**apour **D**eposition (**CVD**-) Diamanten gemacht hat - einem neuen, für die vorgesehenen Experimente mit schweren Ionen ideal geeigneten Material.

Die Untersuchung der physikalischen Eigenschaften von polykristallinen CVD-Diamanten, deren Anwendung und Entwicklung entsprechender Elektronik bei der GSI geschieht in Zusammenarbeit mit der Kollaboration RD42 [9], deren Mitglied die GSI seit 1996 ist.

Physikalische Eigenschaften (bei 300 K)	Diamant	Silizium
Protonenzahl	6	14
Massendichte [$\text{g}\cdot\text{cm}^{-3}$]	3.5	2.33
Gitterkonstante [\AA]	3.57	5.43
Bandlücke [eV]	5.45	1.12
Intrinsische Trägerdichte [cm^{-3}]	$< 10^3$	1.5×10^{10}
Energie zur Erzeugung eines e^-h^+ Paares [eV]	13	3.6
Energie, um ein Atom aus dem Gitterverband zu entfernen [eV]	80	28
Wärmeleitfähigkeit [$\text{W}\cdot\text{cm}^{-1}\cdot\text{K}^{-1}$]	20	1.27
Thermischer Expansionskoeffizient [K^{-1}]	0.8×10^{-6}	2.6×10^{-6}
Spezifischer Widerstand [$\Omega\cdot\text{cm}$]	$> 10^{11}$	2.3×10^5
Maximale Feldstärke [V/cm]	10^7	3×10^5
Elektronen-Mobilität [cm^2/Vs]	2200	1500
Loch-Mobilität [cm^2/Vs]	1600	600
Saturationsgeschwindigkeit [km/s]	220	82
Dielektrizitätskonstante	5.7	11.9

Tabelle 2: Grundlegende Eigenschaften von Diamant und Silizium [10]

Tabelle 2 zeigt die wichtigsten physikalischen Eigenschaften von CVD-Diamanten auf und stellt ihnen die entsprechenden Eigenschaften von Silizium zum Vergleich gegenüber.

Durch die extrem hohe Wärmeleitfähigkeit und starke Bindung der einzelnen Atome im Gitterverband (80 eV) ist der CVD-Diamant ein sehr strahlungsresistentes Detektormaterial, wie es bereits in mehreren experimentellen Tests nachgewiesen wurde.

Die große Bandlücke von 5,45 eV bewirkt eine sehr geringe intrinsische Trägerdichte und erlaubt es, CVD-Diamanten als sehr rauscharme Detektoren zu benutzen, ohne daß Kühlung oder ein pn-Übergang (wie z.B. bei Silizium-Zählern) nötig wären. Mit Anbringen elektrischer Kontakte am Diamanten ist dieser direkt als Teilchendetektor nutzbar.

Der hohe spezifische Widerstand des Materials gestattet es, Felder bis zu $6\text{V}/\mu\text{m}$ anzulegen. In Verbindung mit den hohen Elektron- und Loch-Beweglichkeiten und der niedrigen Kapazität von Diamant-Detektoren führt dies zu extrem schmalen Pulsen:

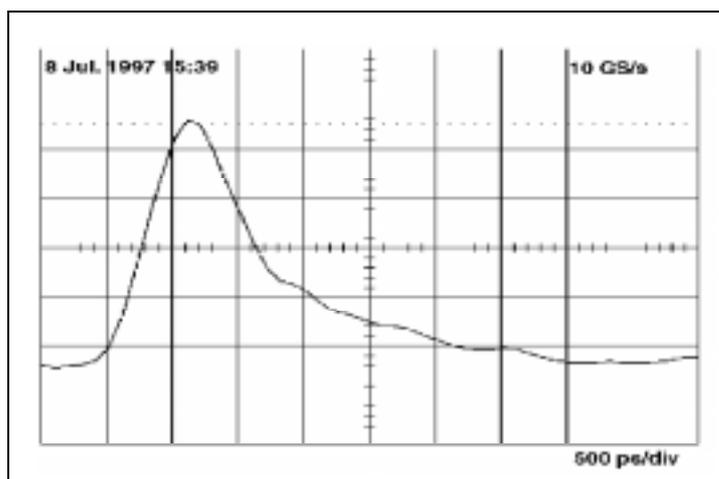


Abbildung 4 : Signal eines Diamantdetektors (aufgezeichnet mit einem 10GS/s „single-shot“ digitalen Speicheroszilloskop), verursacht durch ein einzelnes ^{12}C -Ion mit 200 AMeV [11]. Die Anstiegszeit beträgt ungefähr 300 ps, die $(1/e)$ Abklingzeit 1 ns. Die FWHM-Breite ist 800 ps.

Diese Pulse müssen erst noch aufgearbeitet (d.h. ggf. verstärkt und diskriminiert) werden, bevor sie im Spillmonitor, der differentielle ECL¹-Signale akzeptiert, verwendet werden können. Die benötigte elektronische Zwischenstufe zur Umwandlung wird in der Diplomarbeit von Carsten Lichtblau („Entwicklung schneller Elektronik für das HADES Flugzeitsystem“, Universität Gießen) realisiert werden. Um eine problemlose Übertragung der ECL-Signale via „twisted pair“-Kabel zu erreichen (bei Frequenzen über 300 MHz ist dies nicht mehr trivial, auch wenn sich die beiden Module direkt nebeneinander im VME²-Crate

¹ ECL = Emitter Coupled Logic, s. Kapitel 5.1.2

² VME = Versa Module Eurocard, s. Kapitel 5.2.4

befinden), wird in dieser Zwischenstufe die Signalfrequenz eventuell um einen Faktor 2-4 heruntergeteilt werden.

Die Anstiegszeiten der erhaltenen Pulse ($< 300 \text{ ps}$) sind nur über die Bandbreite der verwendeten Elektronik limitiert; die Abklingzeit wird von Mobilität und Lebensdauer der Ladungsträger und der RC-Konstanten bestimmt, die sich aus der 50Ω Impedanz der verwendeten Elektronik und der Kapazität C des Detektors ergibt. Somit sind Pulsbreite und damit maximale Zählrate über die Kapazität des verwendeten Diamanten bestimmt, die sich optimieren läßt, indem man den Diamanten in Streifen bzw. Pixel unterteilt.

Durch Vergleichsmessungen mit Referenz-Zählern konnte mittlerweile verifiziert werden, daß Diamantzähler bis zu einer Frequenz von einigen 10^8 Teilchen pro Sekunde diese effizient nachweisen können [10].

Da ein Diamantdetektor außerdem sehr schnell und strahlungsresistent ist und Reaktionen des Diamanten mit dem Strahl klein sind im Vergleich zu den vorgesehenen schweren Targets mit Dicken von mehreren $100 \mu\text{m}$, eignet er sich sehr gut als Startdetektor für HADES. Eventuell produzierte Untergrund-Ereignisse, die von den schweren Projektilen erzeugt würden, wären so stark nach vorne fokussiert, daß sie den RICH kaum treffen würden und folglich die Eventcharakterisierung auch nicht stören könnten. Auch wird die Energie von Schwerionen mit Energien $> 50 \text{ AMeV}$ nur um ca. 1-2% beim Durchgang durch die $30\text{-}500 \mu\text{m}$ dicken Diamantdetektoren reduziert.

Aufgrund von Simulationen [12] sind für HADES zwei identische achteckige Diamant-Detektoren vorgesehen, die jeweils 75cm strahlaufwärts bzw. -abwärts des Targets positioniert werden sollen, um deren Detektorfläche mit dem Ionenstrahl optimal „auszuleuchten“.

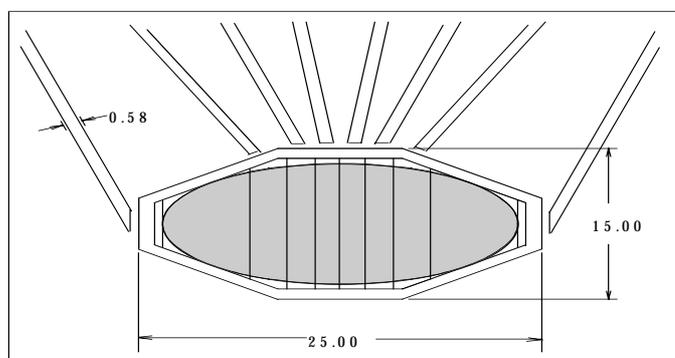


Abbildung 5 : Design des Diamant-Startdetektors für HADES. Der Blick zeigt in Strahlrichtung auf die segmentierte Detektoroberfläche. Außerdem sind die von jedem Segment fortführenden Signalleitungen zu sehen.

Die Unterteilung in acht unterschiedlich breite Streifen (5,4 mm-1,5 mm) wurde auf eine identische Zählrate in den einzelnen Streifen hin optimiert. Durch die achteckige Formgebung konnte die Kapazität der äußersten Streifen um 40% reduziert werden.

Der Detektor hinter dem Target fungiert als Veto-Detektor, der bei Teilchen, die das Target ohne Reaktion passiert haben, ein Veto-Signal erzeugt, so daß die Frequenz der Startsignale für den Level 1-Trigger auf unter 10^7 pro Sekunde fällt. Wie die Simulationen zeigen, reicht eine Koinzidenz mit drei gegenüberliegenden Vetostreifen bereits aus, um eine Veto-Effizienz von 96,5% zu erreichen [12].

Wenn sich auch bisher eine schlechte Pulshöhenauflösung bei polykristallinen Diamantdetektoren gezeigt hat, wird dieser neue Detektortyp sich sicherlich aufgrund seiner herausragenden Eigenschaften in einem großen Einsatzbereich in Schwerionen-Anwendungen im Energiebereich von 50 AMeV bis 2 AGeV etablieren.

Besonders dort, wo hohe Strahlintensitäten in ihrer räumlichen und zeitlichen Struktur überwacht werden müssen, wird der Diamant als rauscharmer, strahlungsresistenter und hoch zeitauflösender Detektor verwendet werden können. Aufgrund der Pulshöheverteilung für schwere Ionen sind deren Signale sehr gut vom Rausch-Untergrund abzutrennen. Daher decken Diamant-Detektoren mit ihrem Meßbereich Strahlintensitäten von einzelnen Ionen bis hin zu 10^8 oder bei geeigneter Segmentierung auch 10^9 Teilchen pro Sekunde ab.

3 Das Schwerionensynchrotron SIS an der GSI

3.1 Die Beschleunigeranlagen der GSI

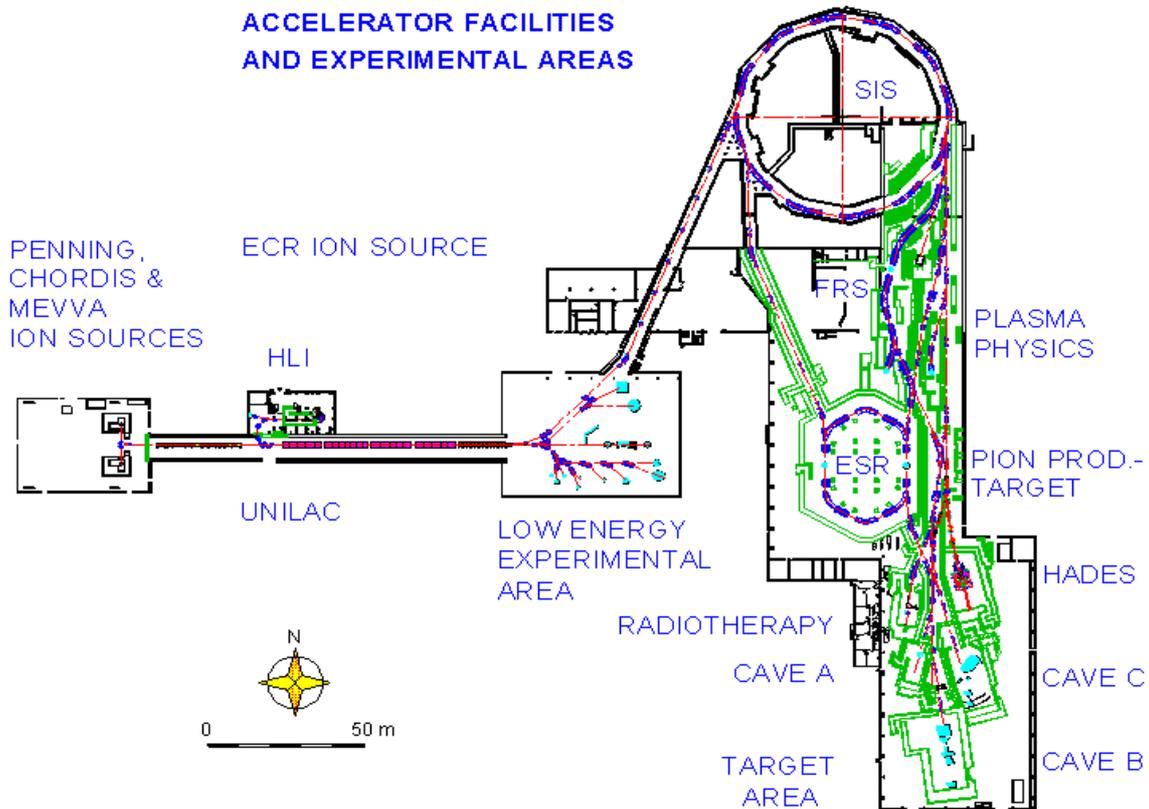


Abbildung 6 : Die Beschleunigeranlagen und Experimentierplätze der GSI. [13]

Die hochenergetischen Ionenstrahlen für das HADES-Experiment werden wie folgt erzeugt:
Zunächst werden die gewünschten Ionensorten aus verschiedenen Ionenquellen (Penning, CHORDIS und MEVVA-Ionenquellen) nach Selektion bezüglich Ladungszustand und Isotop über die Injektoren Nord und Süd in den ersten Teil des **Universal-Linear-Accelerators UNILAC** eingeschossen. Hierbei handelt es sich um einen „Wideröe-Aufbau“ (nach Rolf Wideröe, einem Schweizer Physiker).

Dieser mit 27,12 MHz arbeitende Hochfrequenzbeschleuniger besteht aus 4 verkupferten Stahl tanks von insgesamt 30 Metern Länge, die 130 abwechselnd mit der Hochfrequenz und Masse verbundene Beschleunigungselektroden („Driftröhren“) enthalten. Die aus der Ionenquelle eintretenden Ionenpakete werden zunächst von der negativ geladenen Elektrode angezogen. Sobald die positiven Ionen in der Driftröhre sind, wechselt diese ihre Polarität, was auf die in der Röhre befindlichen Ionen aufgrund des Faraday-Effekts keine Auswirkung

hat. Verlassen die Ionen nun diese Driftröhre, werden sie von der nun positiv geladenen Driftröhre weg auf die nächste, geerdete Driftröhre hin beschleunigt.

Sowie die Ionen sich in dieser Driftröhre befinden, wechselt die Polarität des Hochfrequenzfeldes erneut, und die austretenden Ionen (bzw. das am Anfang des Beschleunigers neu eintretende Ionenpaket) werden vom negativen Potential der folgenden Driftröhre angezogen, usw. .

Damit eine feste Hochfrequenz zur Beschleunigung aller Ionenpakete benutzt werden kann, müssen die Driftröhren immer länger werden, so daß sich die immer schnelleren Teilchen eine halbe Periode in ihnen aufhalten.

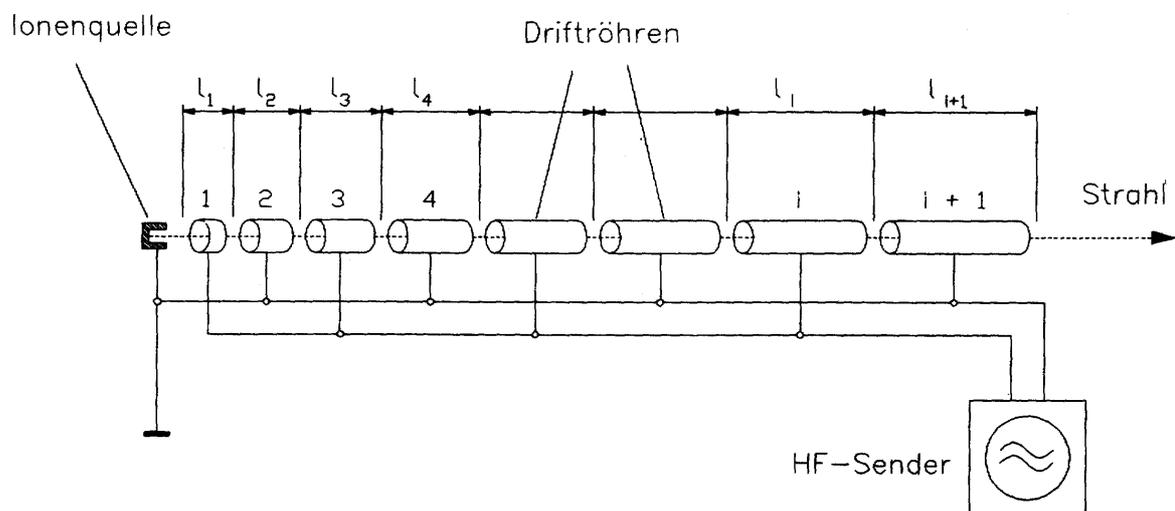


Abbildung 7 : Hochfrequenzbeschleuniger nach Wideröe (entnommen aus [14])

Um Phasenstabilität zu erreichen, verwendet man (eine sinusförmige Hochfrequenz $U(t) = U_0 \sin(\omega t)$ angenommen) als Beschleunigungsphase der Ionen nicht $\varphi = \pi/2$ und damit den maximalen Spannungswert, sondern einen Wert $\varphi < \pi/2$. Dadurch werden Teilchen, die schneller als die Ionen mit Sollgeschwindigkeit sind und vor diesen die Driftröhre verlassen, durch ihre Beschleunigungsphase $\varphi' < \varphi$ auch eine geringere Beschleunigungsspannung als die Sollspannung erfahren und somit wieder etwas zurückfallen. Umgekehrt werden zu langsame Ionen durch diese Phasenfokussierung stärker beschleunigt, so daß alle Teilchen Schwingungen um ihre Sollphase φ ausführen („*Synchrotronschwingungen*“).

Mit der Wideröe-Struktur der GSI werden die verschiedenen Ionensorten je nach Ladungszustand auf bis zu 5,49% der Lichtgeschwindigkeit beschleunigt, was einer Energie von 1,4 MeV pro Nukleon entspricht. Nach der Wideröe-Struktur passieren die Ionen einen

Gasstrahl, in dem sie weitere Elektronen abstreifen und somit einen höheren Ladungszustand erreichen, was eine kürzere restliche Beschleunigungsstrecke ermöglicht.

Nach Selektion der gewünschten Ionenladung treten nun die Ionen aus dem Widerö-Beschleuniger oder aus dem Hochladungsinjektor (HLI), der von einer EZR-Quelle gespeist wird, in die sogenannte Alvarez-Struktur (benannt nach dem amerikanischen Physiker und Nobelpreisträger (1968) Louis Alvarez) ein.

Auch hierbei handelt es sich um einen Hochfrequenzbeschleuniger, der mit 108,48 MHz (dem Vierfachen des Widerö) arbeitet. Die über 150 Driftröhren sind in vier Tanks von je 13m Länge untergebracht. Die Ionen werden hier auf maximal 11,6 MeV (abhängig von Ladung und Masse der verwendeten Ionen) pro Nukleon beschleunigt.

Als dritte und letzte Stufe des UNILAC schließen sich 15 Einzelresonatoren an, die mit unterschiedlichen Frequenzen betrieben werden können. Mittels dieser Resonatoren ist es nun möglich, die Energie des Ionenstrahls zwischen 2 und 18 MeV pro Nukleon zu variieren. Das Umschalten zwischen verschiedenen Endenergien innerhalb von 15 ms ermöglicht es, SIS und die UNILAC-Experimentierhalle mit verschiedenen Energien zu versorgen. Über den Hochladungsinjektor ist es ebenfalls möglich, in 15 ms die Ionenart zu wechseln, so daß UNILAC- und SIS-Experimente mit völlig unterschiedlichen Strahlen arbeiten können.

Die für das HADES-Experiment benötigten hochenergetischen Ionen werden nun vom UNILAC in das Schwerionensynchrotron SIS injiziert.

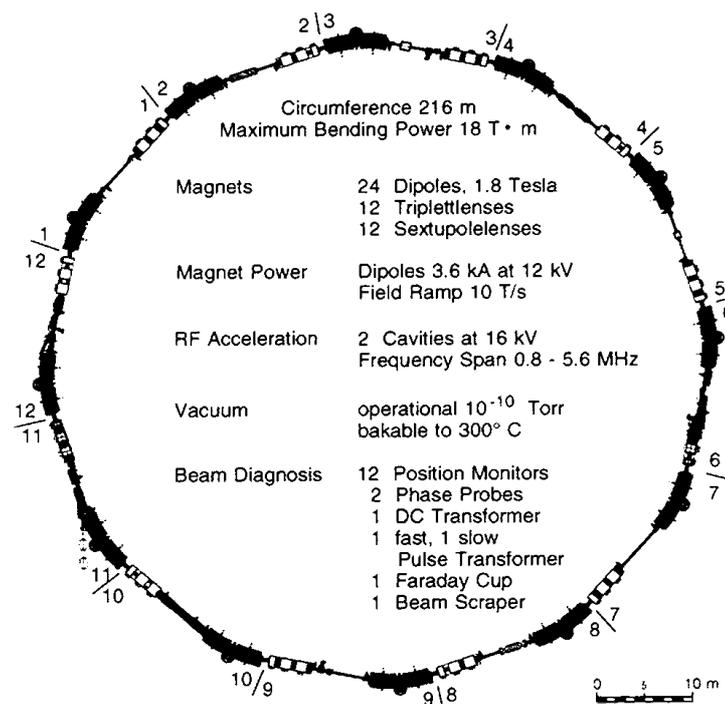


Abbildung 8 : Das SchwerIonenSynchrotron der GSI mit seinen wichtigsten Kenndaten [15].

Um eine möglichst große und effiziente Beschleunigung zu erreichen, werden die Ionen vor der Injektion noch durch eine Kohlenstoff-Folie geschickt, wo sie weiter ionisiert werden (z.B. Uran bis 72+).

Da bei einem Synchrotron der Bahnradius der Teilchen vorgegeben ist, muß mit zunehmender Teilchenenergie das strahlführende Magnetfeld erhöht werden, um die Teilchen auf ihrer Sollbahn halten zu können. Die Frequenz des beschleunigenden elektrischen Feldes und das magnetische Feld werden hier also synchron erhöht, was diesem Beschleunigertyp seinen Namen gab. Da es aufgrund der verbleibenden Koerzitivkräfte in den Dipolmagneten und dem störenden Einfluß des Erdmagnetfeldes nicht möglich ist, Teilchen minimaler Energie von $B=0$ an mit vertretbaren Strahlverlusten zu beschleunigen, benötigen Synchrotrons immer einen Vorbeschleuniger, der die zu injizierenden Strahlteilchen auf eine gewisse Mindestenergie bringt. Im Falle des SIS ist dies der schon beschriebene UNILAC.

Das SIS der GSI hat einen Umfang von 216 Metern. Entlang der Sollbahn der Teilchen sind 24 Dipolmagnete zu Strahlführung und 36 magnetische Linsen unterschiedlicher Multipolordnung zur Fokussierung angebracht. Die Ionen werden durch zwei diametral im Ring liegende Hochfrequenzstrukturen, in denen die Teilchen jeweils eine Spannung von 15kV „durchfallen“, beschleunigt. Die Beschleunigungsfrequenz ist dabei zwischen 800 kHz und 5,6 MHz regulierbar. Abhängig von der Schwere der beschleunigten Ionen, ihrem Ionisationsgrad und dem daraus resultierenden Verhältnis q/m können die Ionen auf Energien zwischen 80-100 AMeV und 1 AGeV (für Uran) bzw. zwischen 80-100 AMeV und 2 AGeV (für Neon) mit einer relativen Energieunschärfe von 10^{-3} beschleunigt werden. Leichte Ionen lassen sich bei UNILAC-Energien (vor Einschub ins SIS) komplett stripfen, so daß sie - auch aufgrund des für leichtere Kerne ausgeglichenen Verhältnisses zwischen Protonen und Neutronen - mit einem Verhältnis $q/m = 1/2$ bis auf 2 AGeV beschleunigt werden können, was für Ionen mit $q/m < 1/2$ nicht mehr der Fall ist.

Gerade zur Beschleunigung schwerer, noch nicht vollständig ionisierter Kerne ist ein extrem hohes Vakuum von 10^{-8} Pascal nötig (zum Vergleich: Im UNILAC werden nur 10^{-5} Pascal benötigt), damit diese Ionen nicht bei Stößen mit Restgas-Molekülen ihren Ladungszustand ändern und damit verlorengehen.

Nachdem die Ionen auf ihre gewünschte Energie beschleunigt wurden, können sie auf zwei verschiedene Arten extrahiert werden:

- Im „*fast extraction mode*“ werden die Ionen innerhalb von einer Mikrosekunde aus dem SIS extrahiert. Dieser Extraktionsmodus wird für einige spezielle Experimente, hauptsächlich aber zum Füllen des Experimentier-Speicherrings **ESR** genutzt. Im ESR können Ionen im Energiebereich 3-560 AMeV für Uran und 3-830 AMeV für Neon bis zu mehreren Stunden gespeichert werden. Über Elektronenkühlung wird in Bruchteilen von Sekunden hier eine relative Genauigkeit in der Energie von 10^{-4} erreicht. Der so erhaltene Ionenstrahl kann wiederum auf die beiden Arten extrahiert werden, wie sie gerade beschrieben werden. Es ist auch möglich, die aus dem SIS kommenden Ionen vor Einschuss in den ESR nochmals zu strippen, dann im ESR zu kühlen und sie schließlich wieder dem SIS zur weiteren Beschleunigung zuzuführen.
- Die zweite Möglichkeit der Extraktion ist der „*slow extraction mode*“. Hierbei wird der Ionenstrahl mittels spezieller Magnete über eine dünne Metallschiene, das sogenannte Septum, „abgeschält“, wodurch die Extraktionsdauer bis zu 9,8 Sekunden betragen kann. Dieser Prozeß ist technisch äußerst anspruchsvoll, da der Ionenstrahl mit wenigen Millimetern Durchmesser das Septum mit einer Dicke um 100 μm während der Extraktion viele tausend Male passieren muß, ohne zu große Verluste zu erleiden. Die meisten Experimente - auch HADES - nutzen diesen Extraktionsmodus, um relativ zur Dauer eines vollständigen Maschinenzyklus einen möglichst lange andauernden, gleich verteilten Teilchenstrahl zu erhalten und somit einen möglichst guten „duty cycle“ zu erreichen.

Je nach Extraktionsart, Strahlrigidität und Strahlenergie beträgt die Dauer eines SIS-Maschinenzyklus 1-14s.

Nach der Slow Extraction vom SIS wird der Ionenstrahl nun entweder direkt dem HADES-Experiment zugeführt, oder es werden die bei der Kollision mit dem Produktionstarget für Pionen entstandenen Pionen zum Experimentieren weiterverwendet.

Einen tieferehenden Einblick in die Physik der Teilchenbeschleuniger findet man z.B. in [14].

3.2 Die Strahlstruktur des SIS

Die Struktur eines solchen Ionenstrahls, wie er bei HADES verwendet werden wird, ist äußerst komplex, wie schon der Prozeß zur Erzeugung dieser Strahlen illustriert. Die folgende Abbildung zeigt eine typische Strahlstruktur des SIS anhand eines Kohlenstoff-Ionenstrahls bei 200 AMeV, der einen Diamantdetektor passiert.

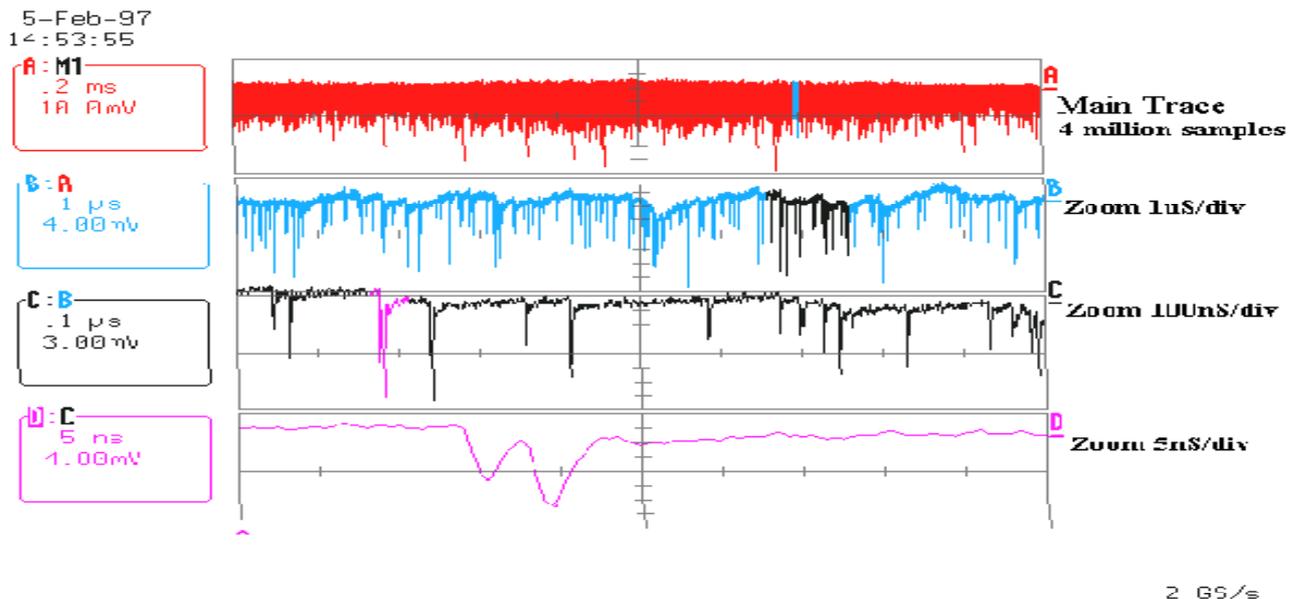


Abbildung 9 : Signalstruktur eines Diamantdetektors, der einem Kohlenstoff-Ionenstrahl von 200 MeV pro Nukleon und einer Intensität von 10^8 Teilchen pro Sekunde ausgesetzt wurde. Die Signale wurden mit einem schnellen digitalen Speicher-Oszilloskop aufgezeichnet. Die Spuren B,C und D sind vergrößerte Ausschnitte der Hauptspur A. In Spur D sind die Pulse einzelner Teilchen erkennbar [16].

Um die mit HADES gewonnenen Daten korrekt analysieren zu können, ist es notwendig, den Ionenstrahl bezüglich seiner zeitlichen Struktur und Intensität ständig zu überwachen, zu analysieren und aufzuzeichnen, damit diese Informationen bei der späteren Datenauswertung zur Verfügung stehen. So ist beispielsweise ein Warnsignal bzw. eine entsprechende Signatur der Daten notwendig, falls die Strahlintensität während eines Peaks 10^8 pps übersteigt, da dann die Totzeit des Detektorsystems zunimmt und die Gefahr von „pile-ups“ (zweier sich überlagernder Ereignisse) besteht, die nicht mehr voneinander getrennt werden können.

Diese Aufgaben soll der Spillmonitor übernehmen.

Wie Abbildung 9 zeigt, kommen die Strahlteilchen zeitlich sehr unregelmäßig verteilt: Immer wieder treten Intensitäts-Spitzen („Spikes“ bzw. „Peaks“) auf, deren innere Strukturen wiederum ungleichmäßige Fluktuationen darstellen, da sie weder durch Gaußsches Rauschen noch mit einer Poisson-Verteilung beschreibbar sind.

Diese Fluktuationen entstehen wahrscheinlich durch die sich überlagernden Einflüsse von Reststörströmen der Magnetnetzgeräte, deren Aufzeichnung zur Untersuchung auf Korrelationen hin vom Aufwand her nur sehr schwer durchführbar ist. Eine FFT („Fast Fourier Transformation“)-Analyse von SIS-Spills liefert in der Regel ein Frequenzspektrum, das von 50 Hz und deren Vielfachen (150, 300, 1200 Hz) dominiert wird.[17]

Als Verhältnisse zwischen durchschnittlicher Strahlintensität und deren Spitzenwerte wurden bis zu 1:20 am SIS gemessen (in der Regel aber eher 1:10).

Basierend auf diesen Erkenntnissen erhält man als Abschätzung der maximalen Zählrate in einem Streifen des Startdetektors und damit der maximalen Signalrate für einen Kanal des Spill-Monitors:

Bis zu 10^8 pps (= 100 MHz) als mittlere Strahlintensität für HADES-Experimente, verteilt auf 8 Streifen, ergeben 12,5 MHz Zählrate pro Streifen im Durchschnitt. Als Spitzenbelastung ergeben sich somit $20 \times 12,5 \text{ MHz} = 250 \text{ MHz}$ pro Streifen. Um auch noch dem Phänomen der sogenannten „blips“ (das Auftreten von im Phasenraum korrelierten Teilchen, die folglich sehr wahrscheinlich nur ein Startdetektor-Segment durchqueren werden) Rechnung zu tragen, muß der Spillmonitor **Spitzenraten von bis zu 400 MHz** verarbeiten können. Dies war eine der wichtigsten Randbedingungen, die in die Entwicklung eines Konzeptes für den Spillmonitor einfloß, das nun im folgenden Kapitel dargestellt werden soll.

4 Aufgabenstellung und Konzept für den Spillmonitor

Basierend auf den bereits dargestellten Anforderungen an den Spillmonitor bezüglich maximaler Zählrate, Überwachungs- und Analysefunktionen, die sich aus den physikalischen Zielsetzungen des HADES-Spektrometers, den Eigenschaften des Startzählers und der Strahlstruktur des SIS ergaben, wurden die Aufgaben des Spillmonitors und deren Umsetzung am 22.1.1998 spezifiziert und genehmigt [18].

Der Spillmonitor soll danach folgende grundlegenden Ziele erfüllen:

1. Hochauflösungs-Überwachung (Auflösung : 10 μ s) der Struktur des Spills, um mögliche Feinstrukturen aufzuzeigen.
2. Die gröbere Gesamtstruktur des Spills (Auflösung: 1 ms) steht dem Kontrollraum zusätzlich zur Verfügung, um den Strahl zu überwachen.
3. Die gesammelten Daten auf dem Board sind via VME Bus erreichbar und können über TCP/IP ausgelesen werden.
4. Grundlegende Analysefunktionen (z.B. zweite Differenzen,...) stehen „onboard“ zur Verfügung, um den Datenverkehr über das Netz zu reduzieren.
5. Ein grobes räumliches Strahlprofil kann über die Zählrate in den je 8 verschiedenen Kanälen des Start- und Vetodetektors ermittelt werden.

Als Ergänzung zu 1. sei erwähnt, daß eine höhere maximale Zeitauflösung von z.B. 1 μ s aus statistischen Gründen nicht sinnvoll ist: *Im Mittel* werden in einem Intervall von 1 μ s bei voller Strahlintensität von 10^8 pps nur 13 Zählpulse pro Kanal auftreten. Dies bedeutet einen statistischen Fehler von 28 % im Vergleich zu einem statistischen Fehler von 9% bei einer maximalen Zeitauflösung von 10 μ s. Außerdem entsprechen 10 μ s dem Zeitintervall, in welchem HADES ein Ereignis verarbeiten kann.

Um die erforderliche hohe Zeitauflösung zu erreichen, müssen Zählerbausteine verwendet werden, deren maximale Zählrate mindestens 400 MHz beträgt (s. vorherigen Abschnitt). Damit ist gewährleistet, daß sich der Zähler auch für diese Intensitätsspitzen nicht an zwei zu schnell aufeinanderfolgenden Pulsen „verschluckt“ und somit zuverlässig die Überschreitung einer einstellbaren Höchstschwelle, bei der ein Alarmsignal ausgelöst wird, bestimmt werden kann. In diesem Frequenzbereich ist der Einsatz herkömmlicher TTL³-Logik nicht mehr möglich. Statt dessen wurde ein ECL-basiertes Zählerkonzept entwickelt, das im folgenden

Kapitel detailliert vorgestellt wird. Zunächst soll hier jedoch einmal das Gesamtkonzept mit den grundlegenden funktionellen Gruppen erläutert werden.

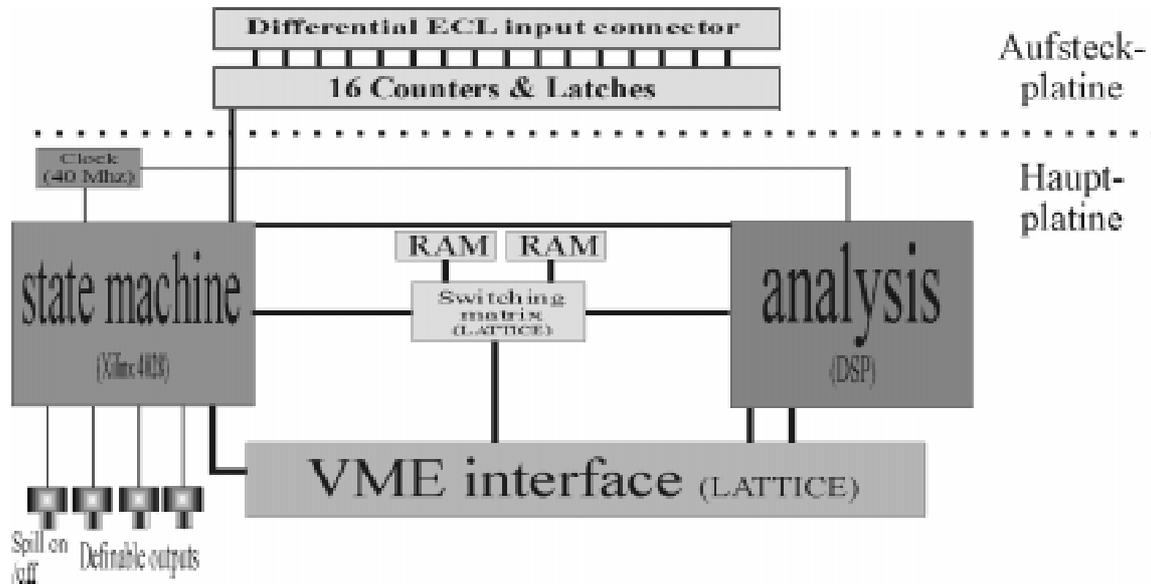


Abbildung 10 : Grundlegender Aufbau des Spillmonitor-Prototypen. Zu sehen sind die wichtigsten Funktionsgruppen und deren Verteilung auf Haupt- und Aufsteckplatine.

Wie man in obiger Abbildung erkennt, werden die in differentielles ECL umgewandelten Signale des Start- und Vetodetektors 16 Zählerbausteinen zugeführt, deren maximale Zählrate über den geforderten 400 MHz liegt. Die Ergebnisse der Zähler werden alle 10 μ s in 16 „Latches“ zwischengespeichert, bis die „State-Machine“, die den Meßprozeß steuert, die Daten zur weiteren Verarbeitung ausliest. Die State-Machine wird (neben weiteren benötigten logischen Funktionen wie z.B. Zählern für den Überlauf der ECL-Zählerbausteine) in einem programmierbaren Logikbaustein realisiert und startet bzw. stoppt den Meßvorgang, je nachdem ob das vom Beschleuniger gelieferte „Spill on/off“ Signal einen ankommenden Ionenstrahl signalisiert oder nicht. Beim Auslesen der Zählerwerte kann die State-Machine über frei definierbare Ausgänge ein Alarmsignal auslösen, falls die gemessene Strahlintensität eine einstellbare Schwelle überschreitet und auch bereits einfache Analysefunktionen übernehmen, bevor sie die Daten in den Speicher des Boards schreibt.

Dieser Speicher ist in zwei Bänke unterteilt, deren Zugang eine Umschalt („Switching“)-Matrix regelt: Schreibt die State-Machine in die erste Speicherbank, so sind die Daten der zweiten Bank, die während des vorherigen Spills aufgenommen wurden, zur Auslese via VME bzw. zur Analyse im vorgesehenen digitalen Signalprozessor (DSP) freigegeben und umgekehrt.

³ TTL = Transistor-Transistor-Logik, s. Kapitel 5.1.2

5 Umsetzung des Konzeptes im Design der Prototypen

Wie bereits erwähnt war durch die hohen Anforderungen bezüglich der maximalen Zählrate die Verwendung von ECL-Bausteinen erforderlich. Um die von diesen Zählern erzeugten logischen Signale in den vorgesehenen restlichen TTL-basierten Bausteinen verwenden zu können, waren zusätzliche Übersetzerkomponenten nötig, so daß nicht mehr alle Bausteine auf der vorgesehenen 6U-VME-Karte (23,36 cm x 16 cm) Platz gefunden hätten.

Deshalb wurde für die ECL-Komponenten mit ihren Übersetzerbausteinen eine separate Platine entwickelt, die auf die Hauptplatine aufgesteckt werden kann. Mit dieser Maßnahme wurde außerdem der hochfrequente ECL-Teil von den TTL-Bausteinen getrennt, so daß sie einander nicht stören und das Layout der Hauptplatine dadurch erheblich vereinfacht.

Im folgenden soll nun ausführlich die Umsetzung des bereits dargestellten Konzeptes im Design dieser beiden Prototyp-Platinen erläutert werden, wobei jeweils anschließend die verwendete Hardware mit ihren grundlegenden Eigenschaften nochmals näher erläutert wird.

5.1 Die Aufsteck-Platine

5.1.1 Aufbau und Funktionsweise

Die vom Spillmonitor zu verarbeitenden, in differentielles ECL umgewandelten 16 Signal-Kanäle von Start- und Vetodetektor werden über ein 34 poliges twisted-pair Kabel (ein Kabelpaar bleibt unbenutzt) und eine entsprechende Steckverbindung den Zählerbausteinen der Aufsteckkarte zugeführt.

Es werden hierbei 8-Bit-breite, synchrone ECL-Zähler MC100E016 der Firma Motorola mit einer maximalen Zählrate von mindestens 700 MHz verwendet. Da diese Komponenten jeweils mit nur einem Signal- („clock“-) Eingang ausgestattet sind, werden die entsprechenden invertierten Signale sofort hinter dem Eingangsstecker mit der den verwendeten Leiterbahnen entsprechenden Impedanz (s. 5.1.3) terminiert.

Die Terminierung der benutzten Signale erfolgt direkt vor den jeweiligen Zählerbausteinen. Über die Signale MR („Master Reset“), CE* („Count Enable“) und PE* („Parallel Enable“) dieser Bausteine ist es möglich, den jeweiligen Zähler anzuhalten, zurückzusetzen und erneut zu starten⁴. Bei einem Überlauf des Zählers generiert dieser ein TC*- („Terminal Count“-)

⁴ Der * hinter einem Signal steht für „low-aktive“ Signale.

Signal, das sich allerdings nur noch mit einer Frequenz von $400 \text{ MHz} / 2^8 = 1,6 \text{ MHz}$ ändert und deshalb mit TTL-Logik weiterverarbeitet werden kann. Zuvor müssen diese Überlauf-Signale allerdings noch in TTL-Pegel umgewandelt werden, was in 2 jeweils 9-Bit-breiten ECL-zu-TTL Umwandlern MC100H6001 der Firma Motorola geschieht.

Bei einer maximalen Zählrate von 400 MHz und Meßintervallen von $10 \mu\text{s}$ sind insgesamt 12-Bit-breite Zähler nötig, um die maximalen Zahlenwerte noch aufnehmen zu können ($2^{12} = 4096 > 4000 = 400 \text{ MHz} \cdot 10 \mu\text{s}$). Die 8-Bit Zählerkomponenten werden deshalb mit 4-Bit-breiten Zählern, die die TC^* -Signale zählen, ergänzt. Diese Zähler können aufgrund ihrer geringen Breite und niedrigen Zählrate in der programmierbaren Logik der State-Machine auf der Hauptplatine implementiert werden.

Die 8-Bit-breiten Ausgänge der ECL-Zähler werden jeweils nach Terminierung dem entsprechenden 9-Bit ECL-zu-TTL Übersetzer mit Latch 100393 der Firma Fairchild zugeführt. Diese Bausteine übernehmen die an ihren Eingängen anliegenden ECL-Signale mit dem Steuersignal LE („Latch Enable“), wandeln diese in TTL um und stellen sie an den Ausgängen bereit. Über das Steuersignal OETTL^* können diese Ausgänge „deaktiviert“ (d.h. hochohmig geschaltet) werden.

Ein Meßzyklus von $10 \mu\text{s}$ wird von der State-Machine nun folgendermaßen gesteuert:

- Zunächst werden nach Ablauf der $10 \mu\text{s}$ alle Zählerbausteine angehalten, um zu verhindern, daß die Übersetzerbausteine mit Latch gerade dann ihre Daten aufnehmen, wenn im entsprechenden Zählerbaustein gerade ein bzw. mehrere Flip-Flops umschalten, da dies zu unvorhersehbaren Ergebnissen führen würde (Dauer: 1 Taktzyklus).
- Nach mindestens 10 ns (Umschaltzeit der Zähler-Flip-Flops plus Signallaufzeit) erfolgt die Übernahme der Zählerwerte in die Latchbausteine (bzw. in 4-Bit-breite in der State-Machine implementierte Latches). (Dauer: 2 Taktzyklen: Start und Stop der Daten-Übernahme).
- Nun werden sowohl die ECL- als auch die implementierten Zähler auf 0 zurückgesetzt, und die Messung wird erneut gestartet (Dauer: 1 Taktzyklus).
- Die State-Machine kann jetzt während der folgenden $10 \mu\text{s}$ die Ergebnisse in den Latches sequentiell über einen 8-Bit-breiten Datenbus auslesen, indem sie nacheinander die Ausgänge der einzelnen Bausteine über das OETTL^* -Signal aktiviert. (Um zu verhindern, daß die verschiedenen Latches gleichzeitig mit aktiven Ausgängen gegeneinander treiben, wurden die OETTL^* -Signale über $10 \text{ k}\Omega$ -Pullup-Widerstände auf High gelegt.)

Zusammen mit den 4-Bit breiten Ergebnissen in den implementierten Latches erhält man so die gewünschten 12-Bit Ergebnisse, die dann weiter verarbeitet werden können.

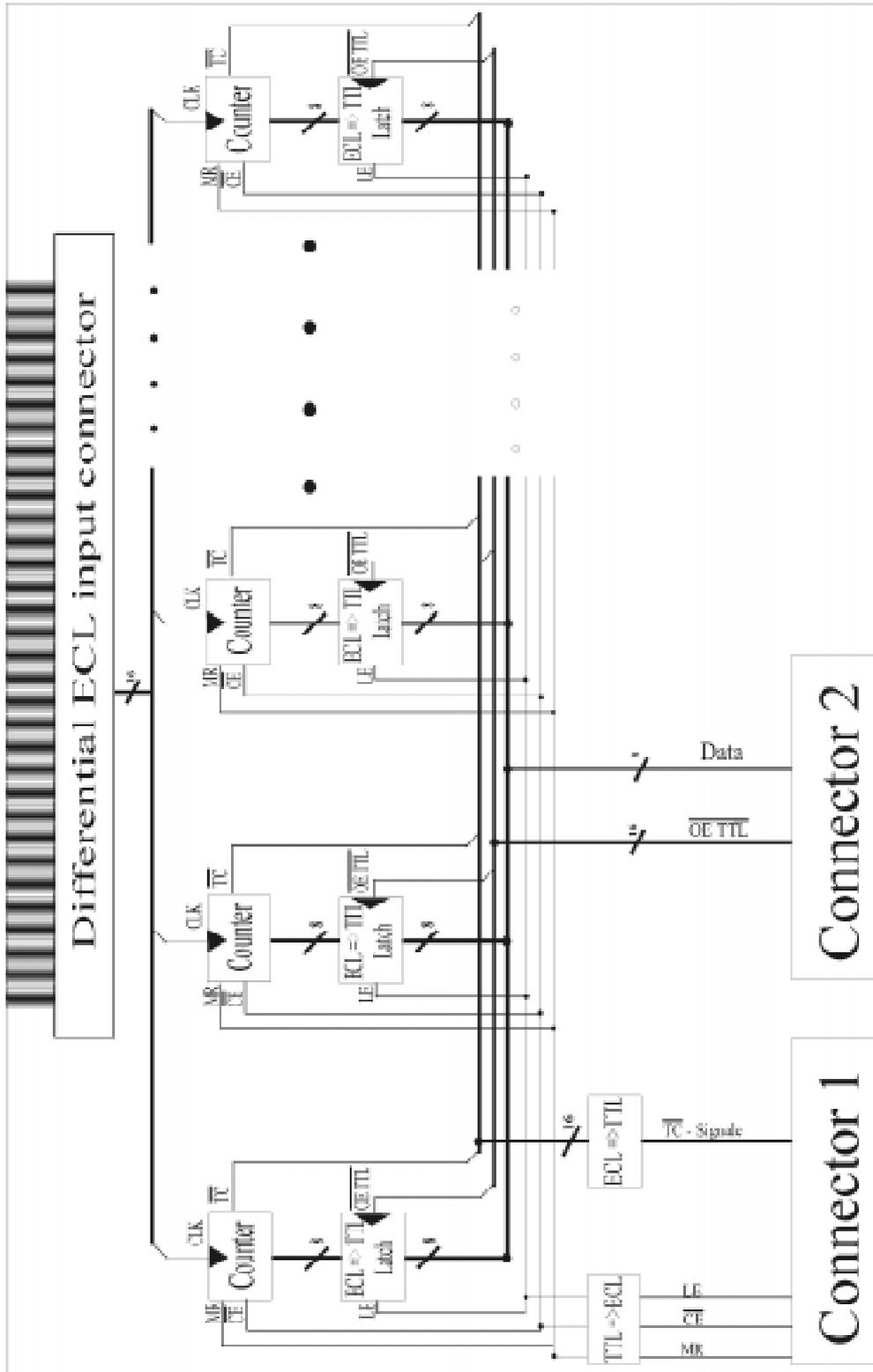


Abbildung 11 : Aufbau der Aufsteckplatine mit den verschiedenen Steuer- und Datenleitungen. Die Zuordnung der Signale zu den Verbindungssteckern ist rein symbolisch. Die genaue Pinbelegung dieser Stecker entnehme man dem Anhang.

Werden die einzelnen Schritte mit den benötigten Wartezeiten jeweils mit Clock-Zyklen realisiert (s.o.), ergeben sich bei Verwendung einer 40 MHz-Clock und maximal vier benötigten Zyklen höchstens 100 ns Totzeit, was 1% der Dauer des Meßzyklus entspricht und somit vernachlässigt werden kann, weil bei den Messungen lediglich die integrale Intensität wichtig ist.

Da die von der State-Machine generierten Signale TTL-Pegel sind, die benötigten Steuersignale MR, CE*, PE* und LE jedoch ECL-Pegel sein müssen, werden diese Signale auf der Aufsteckplatine noch mittels eines 9-Bit-breiten TTL-zu-ECL Konverters MC100H600 (Motorola) umgewandelt, bevor sie den entsprechenden Komponenten zugeführt werden.

Als zusätzliche Funktion sind diese vier Steuerleitungen zweimal vorhanden und steuern jeweils die ersten bzw. letzten acht Kanäle der Aufsteckplatine. Verwendet man nur acht Signal-Kanäle und führt sie sowohl der ersten als auch der zweiten Gruppe von je acht Zählern zu, ist es möglich, für diese acht Kanäle eine quasi unterbrechungsfreie Messung durchzuführen: Beim Auslesen der ersten Zählergruppe wird die zweite gestartet und umgekehrt.

Als Verbindung der Aufsteckplatine mit der Hauptplatine sind somit folgende Signale nötig:

- Zweimal die Steuerleitungen für Zähler und Latches (MR, CE*, PE* und LE) : 8
- TC*-Signale der einzelnen Zählerbausteine : 16
- Datenbus für Zählerergebnisse: 8
- Aktivierungssignale für die einzelnen Latches (OETTL*) 16.

48 Signale

Hinzu kommen noch die vier benötigten Versorgungsspannungen (ECL-Versorgung : -5,2V, ECL-Terminierung : -2V, TTL-Versorgung: +5V , Ground-Plane : 0V). Hierbei ist nun der große Stromverbrauch der verwendeten Komponenten zu berücksichtigen: Addiert man den maximalen Stromverbrauch der ECL-Versorgung, erhält man 4,2 A. Um bei diesen hohen Strömen eine gleichmäßige Versorgung sicherzustellen, wurden pro Spannungsversorgung 12 Versorgungs-Pins vorgesehen (je Pin sind 1,5 A zugelassen).

Insgesamt ergeben sich somit 96 Verbindungen zwischen Haupt- und Aufsteckplatine, die über zwei 48 polige Steckverbinder hergestellt werden.

In den nachstehenden zwei Abschnitten sollen nun die elementaren Besonderheiten der verwendeten ECL-Logik, die in das Platinen-Design einfließen, näher geschildert werden.

5.1.2 ECL-Logik und deren Designgrundregeln

Moderne integrierte digitale Schaltungen lassen sich in zwei Hauptgruppen einteilen: Zum einen in Metal Oxide Silicon MOS-Technologie und in die Bipolarlogik. Die Bipolarschaltungen wiederum lassen sich in zwei Untergruppen unterteilen:

1. Übersteuerungstechnik (z.B. Transistor-Transistor-Logik **TTL**) :

Hier wird der eingeschaltete Transistor im Übersteuerungsbereich betrieben, wodurch sich eine gewisse Ladungsspeicherung ergibt, die sich negativ auf die Schaltzeit auswirkt.

2. Stromschalttechnik (z.B. Emitter Coupled Logic **ECL**):

Der eingeschaltete Transistor wird hier im aktiven Bereich betrieben, so daß keine Ladungsspeicherung auftritt, was zwar zu extrem kurzen Schaltzeiten ($< 1\text{ns}$), aber auch zu hoher Verlustleistung führt.

Ab einem gewissen Frequenzbereich ist die Verwendung obiger Stromschalttechnik unumgänglich. So ist die für den Spillmonitor geforderte maximale Zählrate von 400 MHz nur mit ECL-Komponenten erreichbar. Da ECL-Komponenten im Vergleich zu anderen Technologien eher empfindlich gegenüber Spannungsschwankungen und anfällig für Reflektions- und Übersprechprobleme aufgrund der steilen Signalfanken sind, müssen beim Platinen-Design mit diesen Komponenten auch angesichts des verwendeten Frequenzbereichs gewisse Grundregeln eingehalten werden:

- Die Komponenten benötigen eine sehr saubere, störungsfreie Spannungsversorgung. Dies ist vor allem durch die Verwendung von durchgehenden Kupferlagen mit dem gewünschten Potential („Powerplanes“) sicherzustellen. Besagtes gilt insbesondere auch für das „Erd“-Potential.
- Um zu verhindern, daß sich kurzzeitige Stromspitzen auf die Powerplanes und damit auf die anderen Komponenten übertragen, muß für jeden Stromversorgungs-Anschluß eines ECL-Bausteins ein Entkopplungskondensator vorgesehen werden, der diese Spitzen glättet.
- Zur Verhinderung von Signalreflexionen müssen alle ECL-Signalleitungen mit der richtigen charakteristischen Impedanz der verwendeten Signalleitungen terminiert werden: der Aufbau der Platine muß in sich „impedanzrichtig“ sein.
- Es muß **SMD** (Surface Mounted Devices)-Technologie verwendet werden, um die Länge der Anschluß-Drähte vom elektronischen Baustein zur Platine (sogen. Pins) zu minimieren. Bei Verwendung von Komponenten mit langen Pins wirken diese bei hohen Frequenzen wie Sender und Empfänger, so daß sie sich gegenseitig stören.

- Um Übersprechen zwischen den Signalbahnen zu minimieren, ist der ideale Abstand zwischen den Leiterbahnen gleich dem doppelten Abstand zur Erdungs-Lage („Ground-Plane“) ([19], S.5-21).

Ein ausführlicherer Einblick in Designgrundregeln von ECL-Technologie findet sich z.B. in [19].

5.1.3 Impedanzbestimmung der Leiterbahnen der Aufsteck-Platine

Gemäß den oben beschriebenen Designgrundregeln wurde die Aufsteckplatine als 6-Lagen-Platine designed: die oberste und unterste Lage sind hier Signallagen; die 4 mittleren Lagen (jeweils durch eine Dielektrikums-Schicht voneinander getrennt) wurden als Powerplanes für die vier benötigten Versorgungsspannungen (ECL-Versorgung : -5,2V, ECL-Terminierung : -2V, TTL-Versorgung: +5V , Ground-Plane : 0V) verwendet.

Bei diesem Aufbau sind die Signalbahnen als „Microstrip Lines“ realisiert, d.h. die Signalbahnen sind im Gegensatz zur „Stripline“ nur zu einer Seite hin durch ein Dielektrikum von der Ground-Plane getrennt:



Abbildung 12 : Aufbau einer Microstrip Line

Die charakteristische Impedanz Z_0 einer solchen Microstrip Line läßt sich nun nach folgender Formel berechnen ([19], S. 5-19):

$$Z_0 = \frac{87}{\sqrt{\epsilon_r + 1,41}} \ln \left[\frac{5,98 \cdot h}{0,8 \cdot w + t} \right]$$

mit

ϵ_r = Dielektrizitätskonstante des Substrates

w = Breite der Signalbahn

t = Dicke der Signalbahn

h = Dicke des Dielektrikums.

Obige Formel ist im folgenden Bereich auf 5% genau: $0,1 < w/h < 3,0$ und $1 < \epsilon_r < 15$.

Unter Verwendung der Parameter t , h , ϵ_r für den Multilayer Bautyp 6M16FR4I36K35V5, der von der Firma ILFA Hannover zur Herstellung der Platine benutzt wurde, erhält man Z_0 für verschiedene Bahnbreiten w :

ϵ_r :	4,8	idealer Abstand der
t :	0,017 mm	Leiterbahnen [mil]:
h :	0,16 mm	12,60

w [mil]	Z_0 [Ω]	Z_0 [Ω]*5%	w/h
6	67,369	3,368	0,953
7	62,603	3,130	1,111
8	58,411	2,921	1,270
9	54,668	2,733	1,429
10	51,288	2,564	1,588
11	48,206	2,410	1,746
12	45,375	2,269	1,905

Tabelle 3 : Impedanz der Leiterbahnen mit Fehlerangaben als Funktion der Bahnbreite

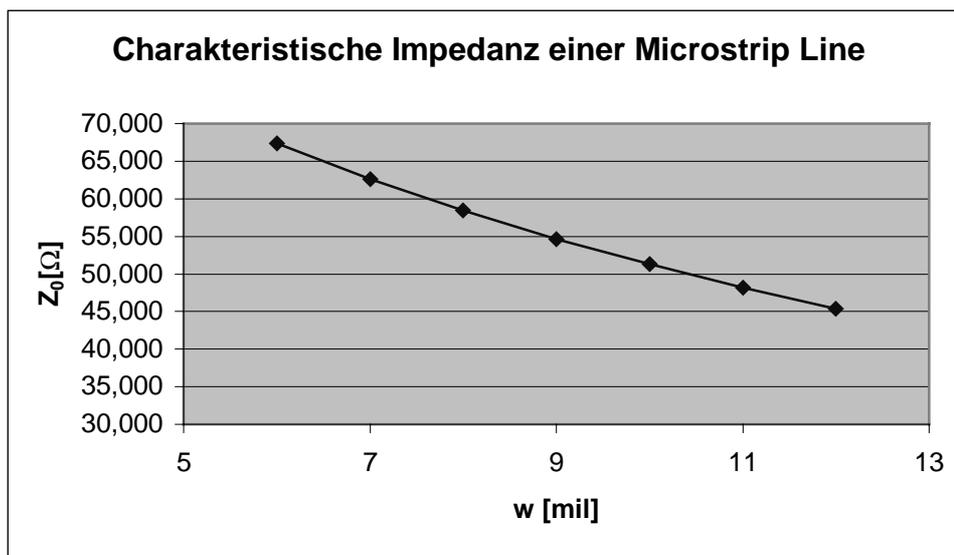


Abbildung 13: Charakteristische Impedanz einer Microstrip Line als Funktion der Bahnbreite

Wie man aus obiger Abbildung erkennt, sind bei einer Leiterbahnbreite von 9 mil ($= 9 \cdot 10^{-3}$ inch $= 9 \cdot 10^{-3} \cdot 25,4$ mm $= 0,23$ mm), die für diese Platine gewählt wurde, idealerweise Widerstände von $54,7 \Omega$ zu verwenden. Da Widerstände in einer solch feinen Abstufung nicht handelsüblich sind, wurden 56Ω -Widerstände zur Terminierung der ECL-Signale benutzt.

Adreß- und weiteren Kontrolleitungen für die Speicher-Ansteuerung, und über die Switching-Matrix in den SRAM-Speicher der Hauptplatine geschrieben. Um neben dieser 10 µs feinen Strahlstruktur, die für jeden Strahl innerhalb eines einstellbaren Zeitfensters („*Feinstruktur-Fenster*“) von 10 ms Länge aufgezeichnet wird, auch noch die grobe Strahlstruktur mit 1 ms Zeitauflösung, die für den *gesamten* Spill aufgenommen wird, zu erhalten, werden intern im FPGA für jeden Kanal die Summen über je 100 aufeinanderfolgende Messungen sowie die Summe über alle Kanäle gebildet und ebenfalls im Speicher abgelegt.

Über den Vergleich der einzelnen Zählraten in den verschiedenen Kanälen mit einer einstellbaren Höchstschwelle kann die State-Machine ein Alarmsignal auslösen und über eine der beiden Ausgangs-Lemo-Buchsen z.B. an die Beschleuniger-Kontrolle weitergeben. Auch erste einfache Analysen wie z.B. Bildung der zweiten Differenzen in den verschiedenen Kanälen könnten bereits in der State-Machine erfolgen.

Um später eine eindeutige Zuordnung zwischen aufgenommenen Daten im Spillmonitor und HADES-Daten vornehmen zu können, werden die Spills vom Beginn des Runs an gezählt und diese Information bei den Rohdaten an einer definierten Stelle abgelegt. Zusätzlich kann bei jedem Auslesen der Rohdaten über den VME-Bus ein Zeitstempel mit abgespeichert werden.

Als einfache Abschätzung des Speicherbedarfs der Grundfunktionen erhält man :

- Für das „*Feinstruktur-Fenster*“ mit 10 µs Auflösung, dessen zeitliche Position und Breite über Statusregister konfigurierbar sein werden, sind 16 Bit · 16 Kanäle alle 10 µs · 1000 bei einer Breite von 10 ms \triangleq 32000 Bytes = 31,25 KB nötig. (Der Einfachheit halber werden die 12-Bit Ergebnisse der Zähler als 16-Bit Worte im Speicher abgelegt.)
- Die grobe Strahlstruktur mit 1 ms Auflösung läßt sich mit 16-Bit-breiten Kanälen nicht mehr erfassen, da bei der Summenbildung größere Zahlen auftreten können: 100 · 4096 (=12 Bit) = 409600 > 2¹⁶ = 65536. Deshalb werden 24-Bit-breite Kanäle benutzt. 24 Bit · 17 Kanäle (16 + 1 für die Summe über alle Kanäle) · 10000 (bei einer maximalen Spilldauer von rund 10 s im Slow Extraction Modus des SIS) = 510000 Bytes = 498,05 KB.

Folglich werden allein für die Speicherung der Rohdaten **530 KB Speicher** benötigt. Um auch bei eventuellen zukünftigen Modifikationen (z.B. einer Verbreiterung des Feinstruktur-Fensters) genügend Speicher zur Verfügung zu haben, können die beiden Speicherbänke mit je 4MB SRAM bestückt werden. Bei den hierfür verwendeten SRAM Modulen handelt es sich um 256K tiefe, 16-Bit-breite (= ½ MB Speicherkapazität) CMOS SRAM Bausteine

KM6164002B der Firma Samsung, mit denen jede Speicherbank mit bis zu 8 Modulen bestückt werden kann.

Wenn die State-Machine nun während einer Messung z.B. in die erste Speicher-Bank schreibt, sind die Daten in der zweiten Bank für den internen Bus, der State-Machine, DSP und VME-Interface miteinander verbindet, zugänglich. Dieser Speicherbereich, der i.d.R. die Rohdaten des vorangegangenen Spills enthält, kann dann via VME ausgelesen werden oder steht dem DSP zur Analyse und Speicherung der Analyse-Ergebnisse zur Verfügung. Mit Beginn des nächsten Spills schaltet die State-Machine ihren Zugriff und den des internen Busses auf die jeweils andere Speicher-Bank um, usw..

Das Umschalten zwischen diesen zwei Speicher-Bänken erfolgt über eine sogenannte Switching-Matrix, die in zwei 2128er Complex Programmable Logic Devices (CPLDs) der Firma LATTICE implementiert wurde :

- Im ersten CPLD werden die jeweils 16 Datenleitungen der beiden Speicher-Bänke mit den 16 Datenleitungen des Xilinx-Datenbusses bzw. des internen Busses verbunden sowie aus den jeweils drei Chip-Select-Bits der letzteren beiden Busse die acht Chip-Select-Signale für die je acht Speicher-Bausteine in einer Bank erzeugt.
- Der zweite CPLD schaltet entsprechend dem ersten die 18 Adreßleitungen der Speicher-Bänke und äußeren Busse aufeinander und ordnet die jeweiligen Steuersignale für den Speicher (UB^{*}, LB^{*} = „upper“ und „lower“ Byte Kontrolle, Write Enable WE^{*} und Output Enable OE^{*}) zu. Für eventuelle spätere Modifikationen bzw. Designerweiterungen ist dieser CPLD noch über jeweils vier frei definierbare Steuerleitungen mit State-Machine und VME-Interface verbunden. Um das Umschalten zwischen den Speicher-Bänken zu synchronisieren und evtl. auch andere Steuersignale weitergeben zu können, ist ein interner 10-Bit-breiter Kontrollbus zwischen beiden CPLDs vorhanden.

Bei aufwendigeren Analysen des DSPs oder länger andauernden VME-Zugriffen auf die Rohdaten kann das Umschalten zwischen den Speicher-Bänken durch die State-Machine blockiert werden, um die entsprechenden Daten nicht zu verlieren. Ein gleichzeitiger Zugriff von VME-Interface und DSP auf die Daten und ein daraus resultierender Buskonflikt kann nicht auftreten, da bei einem VME-Zugriff der DSP immer deaktiviert wird (über das „Suspend Bus Tri State“ SBTS-Signal des DSP), so daß ein VME-Zugriff auf das Board immer höchste Priorität besitzt.

Das VME-Interface wurde ebenfalls in einem 2128er CPLD der Firma LATTICE implementiert. Es dient der Herstellung der Kommunikation der Platine mit dem VME-Bus,

der Steuerung aller wichtigen Funktionen im Testbetrieb und der Konfiguration der Bausteine, die bei einem Spannungsverlust ihr Programm verlieren (DSP und FPGA).

Zunächst einmal müssen die auf dem VME-Bus anliegenden Adressen decodiert werden, um festzustellen, ob ein Zugriff auf das Board erfolgt. Im Testbetrieb wird hierzu nur ein Adreßraum von 24 Bit verwendet; im späteren Betrieb zusammen mit anderen Modulen in einem VME-Crate kann der volle VME-Adreßraum von 32 Bit genutzt werden, um Adreßkonflikte zu vermeiden. Für den Testbetrieb war es ausreichend, nur die obersten 4 Bit des 24-Bit Adreßraumes zu decodieren. Es wurde dabei folgende Belegung gewählt:

Adresse (24 Bit, hexadezimal)	Angesprochene Funktionsgruppe
100000	VME-Interface
200000	1. LATTICE der Switching-Matrix
400000	State-Machine (Xilinx)
500000	Speicherbank A
600000	Speicherbank B

Tabelle 4 : Adreßsegmente für die verschiedenen Funktionsgruppen.

Bei den ersten drei Adressen können jeweils Status- und Kontrollregister im entsprechenden Baustein angesprochen werden. Bei den Speicheradressen handelt es sich jeweils um die Basisadresse; aufsteigend von der jeweiligen Basis an können die einzelnen Speicherbausteine mit 18-Bit Adreßraum und zusätzlich 3 Bit zur Baustein-Auswahl in jeder Bank angesprochen werden.

Während das VME-Interface direkt mit den Signalleitungen des VME-Busses verbunden ist, sind die Daten- und Adreßleitungen des internen Busses über Treiberbausteine (für die Daten: bidirektionale Treiberbausteine 74ALS645; für die Adressen: Latches 74F574) vom VME-Bus getrennt:

Bei VME-Zugriffen, die nicht den Spillmonitor bzw. nur sein VME-Interface betreffen, sind diese Treiberbausteine hochohmig geschaltet, so daß diese beiden Bussysteme voneinander getrennt arbeiten können und der DSP auf die Rohdaten im Speicher zugreifen kann.

Erst bei einem VME-Zugriff auf eine am internen Bus beteiligte Funktionsgruppe (SRAM, Xilinx, DSP) schaltet das VME-Interface die Treiberbausteine auf den internen Bus durch und nimmt die entsprechende Arbitrierung vor, so daß nur die angesprochene Funktionsgruppe auf die nun anliegenden Signale reagiert. Im Testbetrieb wird auch das Umschalten zwischen den einzelnen Speicherbänken komplett vom VME-Interface gesteuert. Die Zustände der

einzelnen Funktionsgruppen können an den 12 LEDs, die mit VME-Interface, State-Machine und Switching-Matrix verbunden sind, abgelesen werden.

Die Konfiguration der einzelnen programmierbaren Bausteine der Platine funktioniert wie folgt:

- Der DSP kann über das VME-Interface gebootet werden, da alle hierfür nötigen Konfigurationspins mit dem Interface verbunden sind und über ein spezielles Statusregister angesprochen werden können. Da die Inbetriebnahme des DSP im Rahmen dieser Diplomarbeit nicht geplant ist, soll hier nicht näher auf diesen Vorgang eingegangen werden. Nähere Informationen hierzu finden sich in [20]. Um den späteren Einsatz des DSP möglichst offen zu gestalten, sind viele Anschlüsse auf der Platine vorhanden, die im ersten Testbetrieb noch unbenutzt sind. So ist z.B. eine Kommunikation zwischen State-Machine und DSP über dessen vier Flag-Leitungen und den seriellen Bus des DSP möglich. Für weitere externe Kommunikation sind drei Linkports mit Stiftleisten und einer von diesen zusätzlich noch mit dem VME-Interface verbunden.
- Die CPLDs können über ein spezielles Kabel, das einen 8-poligen Stecker auf der Hauptplatine mit der parallelen Schnittstelle eines PCs verbindet, über einen PC konfiguriert werden. Bei diesem Ladevorgang werden die Konfigurationsdaten seriell durch die in einer Kette angeordneten LATTICE-Bausteine geschickt:

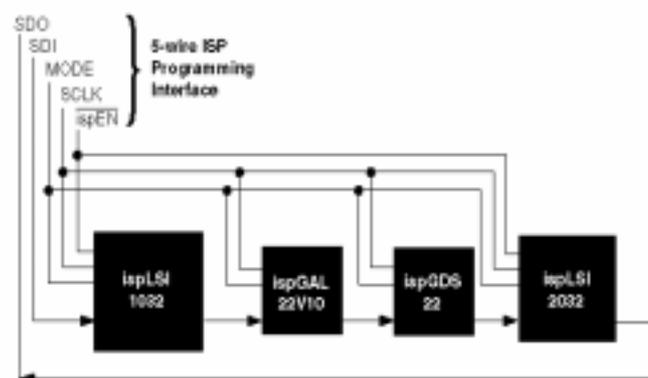


Abbildung 15 : ISP (In System Programming) Interface von LATTICE CPLDs [21]. Die genaue Pinbelegung des 8-poligen Steckers entnehme man dem Anhang.

- Der FPGA bietet verschiedene Konfigurationsmöglichkeiten [22]. Für den Spillmonitor wurde der dem CPLD-Konfigurationsverfahren sehr ähnliche Slave-Serial-Mode gewählt. Auch hier erhält der Xilinx seine Konfigurationsdaten über ein serielles Bitmuster. Dieses Bitmuster wurde im Testbetrieb über die serielle Schnittstelle eines PCs und ein „XChecker-Kabel“ [23], das mit einem 8-poligen Stecker auf der Hauptplatine verbunden

wurde, in den Xilinx hineingeladen. Im späteren Betrieb kann die Konfiguration des FPGAs, der sein Programm bei Spannungsverlust verliert, über ein Kontrollregister im VME-Interface erfolgen.

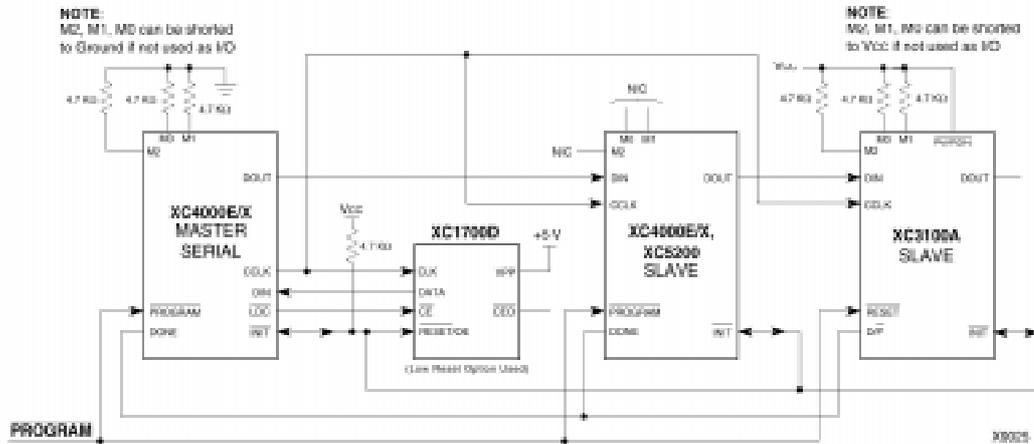


Abbildung 16 : Anschluß eines Xilinx FPGAs für den Slave-Serial Konfigurationsmodus. Die Rolle des die Konfiguration steuernden Masters kann über XChecker-Kabel vom PC oder vom VME-Interface übernommen werden. Die Pinbelegung der Anschlußstiftleiste entnehme man dem Anhang.

Die Umsetzung dieser Hauptplatine erfolgte in einer 6U-VME-Karte, bestehend aus sechs Lagen (vier Signallagen und zwei Stromversorgungslagen).

Da in den VME-Spezifikationen keine ECL-Spannungsversorgung vorgesehen ist, wurde die Hauptplatine zusätzlich zwischen den beiden 96-poligen VME-Steckern mit einem weiteren 30-poligen Stecker, der sogenannten „CERN-Erweiterung“, ausgestattet. Diese CERN-Erweiterung sieht die benötigten ECL-Versorgungsspannungen (ECL-Versorgung : -5,2V, ECL-Terminierung : -2V) vor. Um eine ausreichende Stromversorgung der Aufsteckplatine sicherzustellen, wurde die Verbindung von der CERN-Erweiterung zu den Steckern der Aufsteckplatine über Kupfer-Polygone auf der Ober- und Unterseite der Hauptplatine hergestellt.

Steckt man die beiden Platinen aufeinander, so nimmt der Spillmonitor die Breite von zwei Einschüben (sogen. „Slots“) in einem VME-Crate ein, das natürlich diese CERN-Erweiterung besitzen muß.

Die folgenden Abbildungen zeigen Fotos der Prototypen, wie sie bei der Inbetriebnahme, die im folgenden Kapitel dargestellt wird, verwendet wurden.

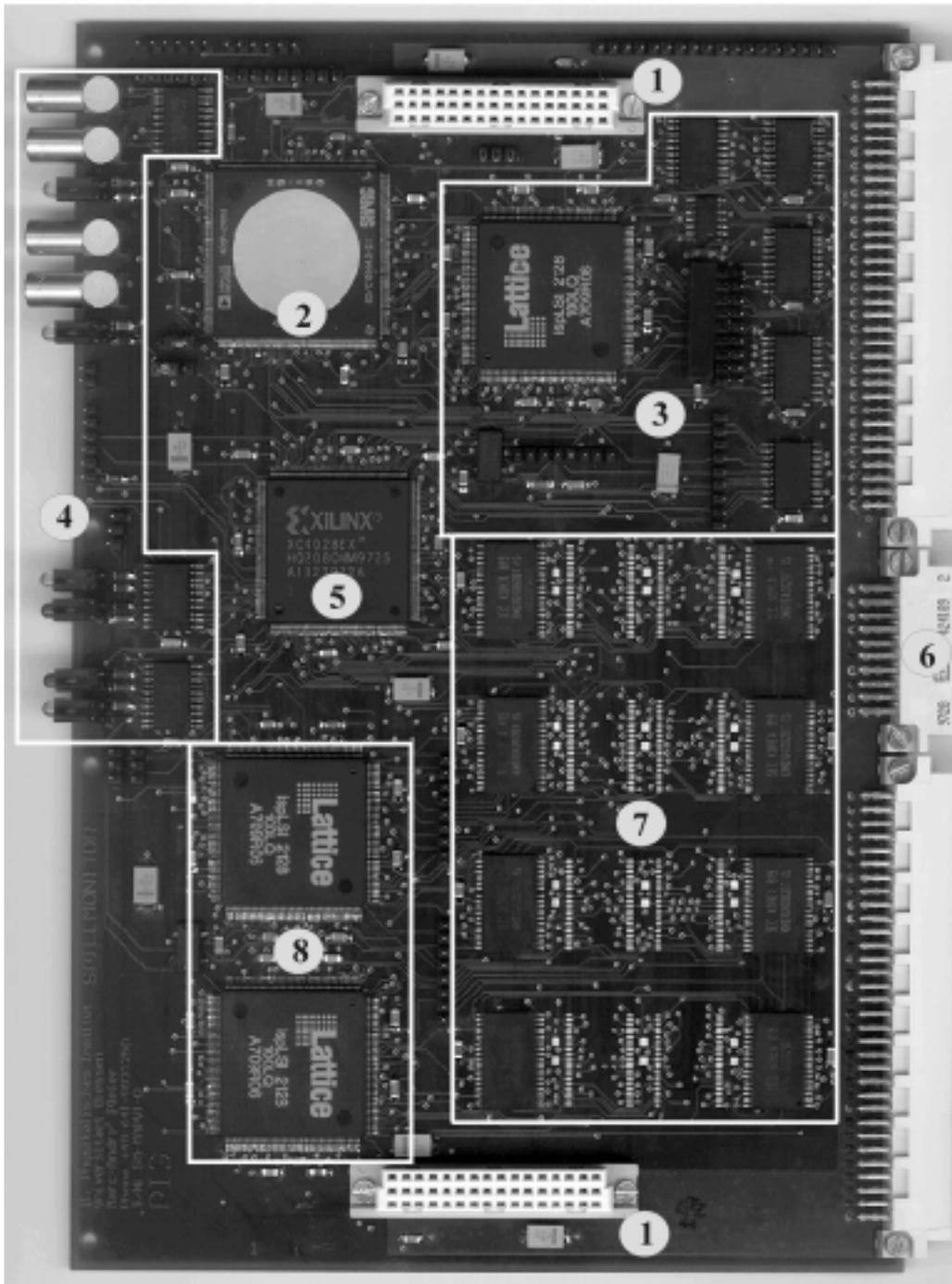


Abbildung 17 : Prototyp der Hauptplatine des Spillmonitors:

- 1) Steckverbinder für die Aufsteckplatine,
- 2) DSP,
- 3) VME-Interface mit Treiberbausteinen,
- 4) LEMO-Buchsen und LEDs mit Treiberbausteinen,
- 5) State-Machine,
- 6) Steckverbinder für CERN-Erweiterung,
- 7) SRAM-Bänke (nur 4 von möglichen 8 MB sind bestückt),
- 8) Switching-Matrix.

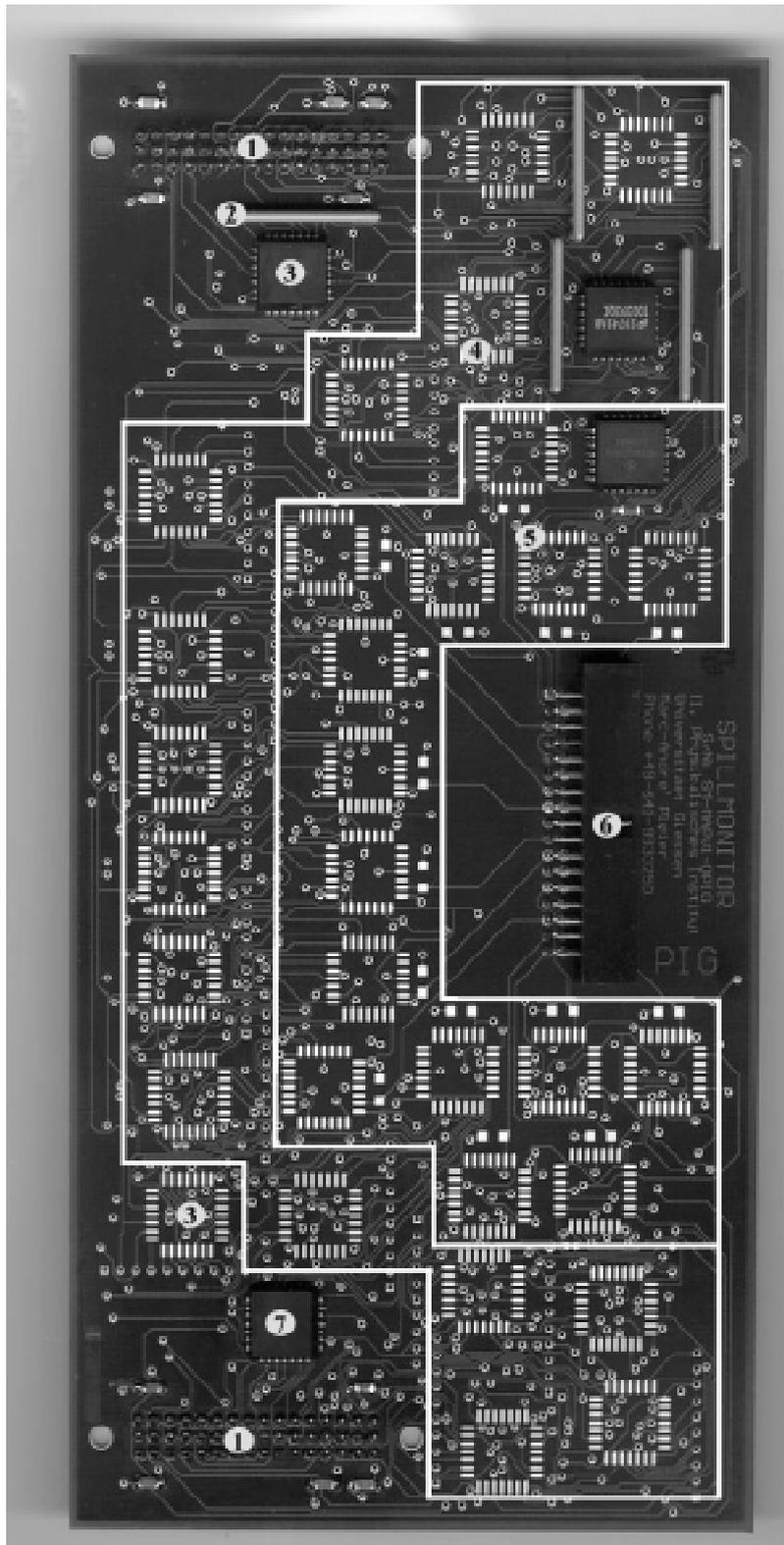


Abbildung 18 : Prototyp der Aufsteckplatine, wie sie beim ersten Test verwendet wurde:

- 1) Steckverbinder für die Hauptplatine (auf der Unterseite der Aufsteckplatine),
- 2) Terminierungs-Widerstands-Arrays,
- 3) ECL-TTL-Übersetzerbausteine (nur 1 bestückt),
- 4) ECL-TTL-Übersetzer mit Latch (nur 1 bestückt),
- 5) ECL-Zählerbausteine (nur 1 bestückt),
- 6) Steckverbinder für Eingangssignale,
- 7) TTL-ECL-Übersetzerbaustein.

In den anschließenden 3 Abschnitten sollen nun die einzelnen verwendeten programmierbaren Bausteine und das VME-Bussystem kurz näher erläutert werden.

5.2.2 Programmierbare Logikbausteine

Die Realisierung komplexer logischer Funktionen, wie sie für den Spillmonitor erforderlich sind, ist durch den Aufbau mittels diskreter Standard Logik-Gatter nicht mehr sinnvoll, da diese meist eine eingeschränkte Funktionalität mit sich bringen und die eigentliche logische Vernetzung erst auf der Platine stattfindet, was eine niedrige Fehlertoleranz mit sich bringt.

Abhilfe schaffen hier programmierbare Logikbausteine („Programmable Logic Devices“ **PLDs**), die sämtliche Grundfunktionen der Boole'schen Algebra zulassen und über die Verkettung der internen logischen Gatter, Flip-Flops und Speicherzellen komplexe logische Designs ermöglichen. Herausragende Vorteile dieser PLDs sind die im Vergleich zum Aufbau mit Standard-Gattern niedrigere Anzahl von verwendeten Bausteinen und damit niedrigerem Platz- und Leistungsbedarf, hohe Flexibilität und damit kürzere Entwicklungszeiten und schnelle Änderungsmöglichkeiten des Designs.

Die Programmierung solcher PLDs kann mittels Hardware-Beschreibungssprachen („Hardware Description Language“ **HDL**, z.B. VHDL oder ABEL) oder über schematische, graphische Beschreibung mit Makrobibliotheken erfolgen. Dieser „Design-Entry“ wird anschließend durch einen Compiler in die bausteinspezifischen Konfigurationsdaten mit optimierten logischen Gleichungen umgewandelt, die dann in den Baustein hineingeladen werden können. Weiterhin ermöglicht heute beinahe jede Software zur PLD-Programmierung die Simulation der zu implementierenden Logik im Hinblick auf Funktionalität und Timing-Verhalten.

Für den Spillmonitor wurden zwei verschiedene Familien von PLDs verwendet: Ein FPGA („Field Programmable Gate Array“) der Firma Xilinx und CPLDs („Complex Programmable Logic Devices“) der Firma LATTICE. Auf die grundlegenden Unterschiede von CPLDs und FPGAs im inneren Aufbau soll hier nicht näher eingegangen werden; detailliertere Informationen zu diesem Thema finden sich u.a. in [24],[25] und [26].

Die wichtigsten, anwendungsrelevanten Unterschiede sind darin zu sehen, daß CPLD-Bausteine auf EECMOS Basis aufgebaut sind, wohingegen FPGAs auf SRAM-Strukturen basieren, was dazu führt, daß CPLDs ihr Programm im Gegensatz zu den FPGAs auch bei Spannungsverlust nicht verlieren. Weiterhin verfügen CPLDs über feste Signallaufzeiten (Pin-to-Pin Timing) , was bei FPGAs aufgrund des komplexeren Aufbaus nicht der Fall ist, aber

wiederum die Realisierung vielschichtigerer Logik erlaubt. Gemeinsam ist beiden PLD-Familien die vielfache Möglichkeit zur Rekonfiguration.

5.2.3 Der digitale Signalprozessor (DSP)

Um das zeitliche Verhalten der Strahlstruktur des SIS-Beschleunigers auch mit aufwendigeren Analysemethoden wie z.B. einer **Fast Fourier Transformation (FFT)** online untersuchen zu können, wurde der Spillmonitor mit einem digitalen Signalprozessor ADSP-21062 SHARC (**S**uper **H**arvard **A**rchitecture **C**omputer) der Firma Analog Devices ausgerüstet.

Einige der wichtigsten Features dieses „Single-Chip-Computers“ sind:

- 32-Bit Fließkomma-Berechnungen: Multiplizierer, ALU und Shifter
- 40 MIPS
- bis zu 120 MFLOPS
- 2 Mbit Dual-Ported SRAM
- 10 DMA-Kanäle

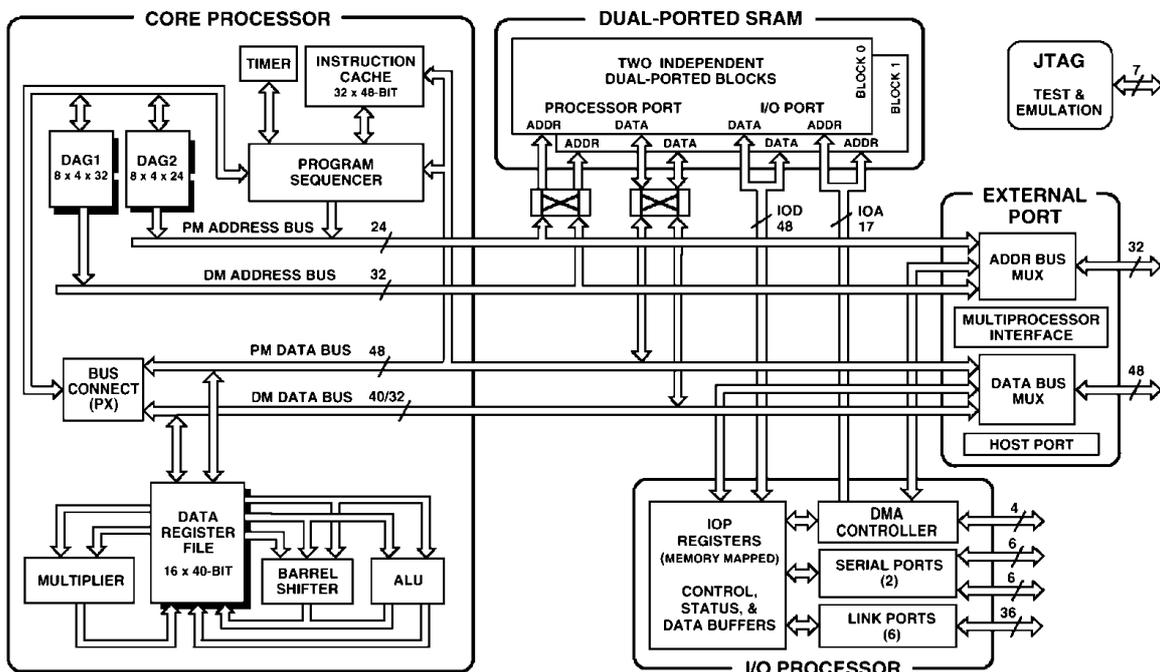


Abbildung 19 : Grundlegender Aufbau des ADSP-21062 von Analog Devices

Dieser leistungsstarke Prozessor wird über das VME-Interface initialisiert werden und ermöglicht auch komplexere Analyse-Algorithmen, als es alleine mit PLDs möglich wäre.

Von besonderem Vorteil ist hierbei, daß der DSP im Gegensatz zu den PLDs in einer Hochsprache wie C programmiert werden kann und es auch schon fertige Software-Module zur Daten-Analyse (z.B. FFT) gibt.

5.2.4 Der VME-Bus

Der VME („Versa Module Eurocard“-)Bus dient der Verbindung von 32-Bit-Multiprozessor-Systemen. Neben der Festlegung eines Busprotokolls, d.h. der Spezifikation des Datenaustausches (Arbitrierungs-, Transfer- und Exzeptionszyklen), sind im VME-Standard auch alle wichtigen elektrischen Anschlußbedingungen (z.B. Pegelwerte, Strombelastung der Ausgänge usw.) und mechanischen Randbedingungen (Abmessungen der Platinen, Art der Steckverbindungen,...) definiert.

Der VME-Bus arbeitet dabei nach dem Master-Slave-Prinzip: Eigenständige Module (Master) steuern die Datenübertragung von und zu den unselbständigen Slave-Modulen. Die Datenübertragung erfolgt hierbei asynchron, so daß in jedem Zyklus die Transferrate vom langsamsten beteiligten Modul bestimmt wird. Weiterhin erlaubt der VME-Bus die Ausführung von Multilevel-Interrupts mit voller Interruptbestätigung durch das auslösende Modul. Auch der Spillmonitor verfügt über die Möglichkeit, Interrupts auszulösen und zu beantworten; im Testbetrieb wurde diese Möglichkeit allerdings nicht genutzt.

Eine ausführliche Beschreibung der verschiedenen Datentransfer-Modi findet sich z.B. in [27]. Der für den Spillmonitor verwendete Read/Write-Modus mit dem zugrundeliegenden Signal-Handshake wird im folgenden Kapitel anhand von Aufzeichnungen eines Logik-Analysators konkret erläutert werden.

6 Inbetriebnahme und erste Tests

6.1 Inbetriebnahme der Hauptplatine

Nachdem die Platinen des Spillmonitor-Prototypen von der Firma ILFA in Hannover gefertigt worden waren, wurden sie im institutseigenen SMD-Labor bestückt und dann auf Kurzschlüsse und kalte Lötstellen hin (z.B. optisch mittels eines 3D-Mikroskops der Firma Vision) untersucht. Nachdem diese Kontrolle abgeschlossen war, konnte die eigentliche Inbetriebnahme des Prototypen beginnen. Aufgrund der elementaren Steuer- und Kontrollfunktionen wurde hierbei zunächst mit der Hauptplatine angefangen. Deren modularer Aufbau ermöglichte es, die einzelnen Funktionsgruppen getrennt voneinander bzw. aufeinander aufbauend auf ihre Funktionstüchtigkeit hin zu überprüfen. Es wurden hierbei ein VME-Crate mit CERN-Erweiterung (s. 5.2.1) und eine ELTEC E7-CPU unter dem Betriebssystem OS9 verwendet. Die Überprüfung der Funktionsweise der einzelnen Bauteile erfolgte über die boardeigenen Kontrollmechanismen (LEDs), ein digitales Multimeter, einen 32-Kanal-Logikanalysator der Firma HP und ein 500 MHz-Oszilloskop ebenfalls von HP.



Abbildung 20 : Testumgebung für die Hauptplatine des Spillmonitors mit VME-Crate, Oszilloskop und Logikanalysator.

Zu Beginn wurde kontrolliert, ob alle CPLDs funktionsbereit waren. Dies war mit der Download-Software für LATTICE CPLDs sehr gut möglich, da diese Software bei ihrem Start zunächst über das Download-Kabel einen Handshake mit den auf der Platine vorhandenen CPLDs durchführt und dann Anzahl und Art der vorhandenen Komponenten zurückmeldet.

Nachdem dieser Test erfolgreich abgeschlossen war, konnte mit der Implementierung des VME-Interfaces begonnen werden. Hierzu wurde zunächst ein C-Programm geschrieben, das es ermöglicht, auf verschiedene (einstellbare) Adressen Schreib- und Lesezugriffe durchzuführen (der Quellcode befindet sich im Anhang). Daraufhin wurde im entsprechenden CPLD in ABEL ein erster einfacher Handshake der Hauptplatine mit dem VME-Bus implementiert, der selektiv auf die für das Board relevanten Adressen mit einem durch einen Delay-Baustein um 100 ns verzögerten DTACK*-Signal reagiert und so das Auslösen eines Bus-Fehlers beim Ansprechen der richtigen Adresse verhindert. Hierauf aufbauend konnten nun ein bidirektionales, 8-Bit-breites VME-Kontroll- und Statusregister und die verschiedenen Steuersignale für State-Machine und Switching-Matrix bzw. SRAM-Speicher eingerichtet werden.

Auch die zu implementierende Logik aller anderen PLDs der Hauptplatine wurde hierbei in der Programmiersprache ABEL beschrieben. Die wichtigsten Quellcodes hierzu befinden sich im Anhang. Nach Implementierung der grundlegenden Funktionen des VME-Interfaces wurden die beiden CPLDs der Switching-Matrix programmiert, um überprüfen zu können, ob eine Kommunikation mit den Speicherbänken der Hauptplatine möglich ist. Folgende Aufzeichnungen des verwendeten Logikanalysators stellen erfolgreiche VME-Schreib- und Lesezugriffe auf den Speicher dar.

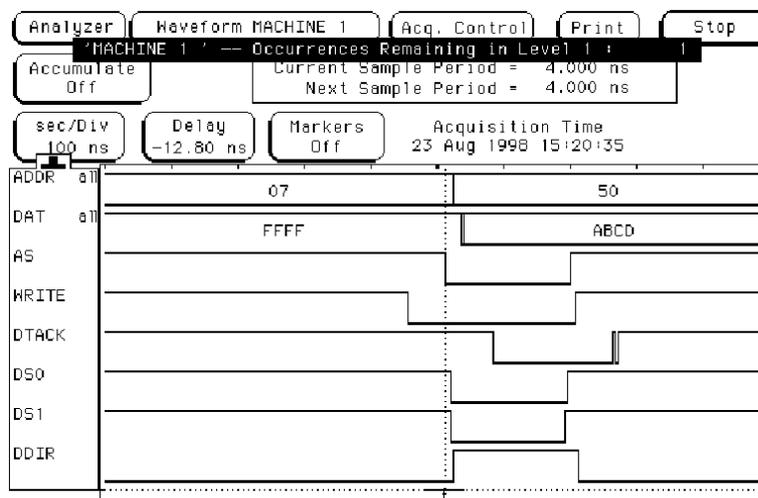


Abbildung 21: VME-Schreibzugriff auf die erste Speicherbank (Basisadresse 500000 hex).

Bei dem VME-Schreibzugriff auf die gewünschte Adresse signalisiert zunächst das WRITE*-Signal einen bevorstehenden Schreibzugriff. Das folgende AS*-Signal teilt dem VME-Interface mit, daß eine gültige Adresse auf dem Bus anliegt, die das Interface mit den für die Hauptplatine relevanten Adressen vergleicht. Ist der Adreßraum der Platine angesprochen, so reagiert das Interface mit einem DTACK*-Signal, das den Erhalt der Adreßinformation bestätigt. Tritt eine solche Bestätigung nicht auf, wird ein Bus-Fehler ausgelöst und der Zugriff abgebrochen. DS0* und DS1* signalisieren je das Anliegen eines gültigen Datenbytes. Die Daten und Adressen, die vom Logik-Analysator aufgezeichnet wurden, wurden hinter den Treiberbausteinen der Hauptplatine abgegriffen, was eine gewisse Verzögerung gegenüber obigen Steuersignalen bewirkt, aber gleichzeitig die korrekte Funktionsweise von VME-Interface und Treiberbausteinen demonstriert. Das DDIR-Signal des Interfaces steuert die Richtung, in die die bidirektionalen Datentreiber geschaltet sind, im Falle des Schreibzuges also in die Hauptplatine hinein.

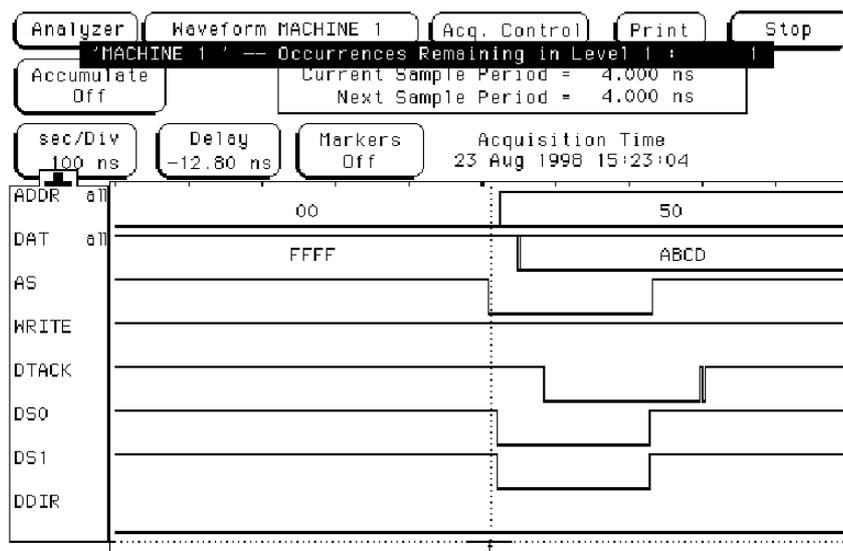


Abbildung 22 : Lesezugriff des VME-Busses auf die erste Speicherbank der Hauptplatine.

Bei einem VME-Lesezugriff bleibt das WRITE*-Signal High, während AS* wiederum das Anliegen einer gültigen Adresse kennzeichnet. Erneut reagiert das VME-Interface mit dem DTACK*-Signal. Wenn DS0* und DS1* aktiv werden, müssen die gültigen Daten auf dem Bus anliegen, wie oben zu sehen ist. Das DDIR-Signal signalisiert diesmal den Treiberbausteinen, auf den VME-Bus zu treiben.

Wie die Aufzeichnungen des Logikanalysators belegen, konnten erfolgreich Schreib- und Lesezugriffe auf den Speicher durchgeführt werden, was die korrekte Funktionsweise von VME-Interface, Treiberbausteinen, Switching- Matrix und SRAM bestätigt. Um sicher sein zu

können, daß der Speicher auch im kompletten Adreßbereich und bei schnell direkt aufeinander folgenden Zugriffen noch korrekt arbeitet, wurden verschiedene Testprogramme in der Programmiersprache C programmiert, die einen Block von Daten schreiben, zurücklesen und mit dem geschriebenen Datenblock vergleichen bzw. über den gesamten Adreßraum eines Chips Daten schreiben, gleich zurücklesen und mit dem Geschriebenen vergleichen. Die Quellcodes dieser Testprogramme befinden sich im Anhang. Alle vorgenommenen Tests des SRAM-Speichers verliefen hierbei erfolgreich.

Als nächster Schritt wurde die State-Machine in Betrieb genommen. Zunächst wurde mit einem einfachen Testprogramm zur Ansteuerung der mit der State-Machine verbundenen vier LEDs verifiziert, daß sich der FPGA programmieren läßt. Daraufhin wurde mit der Programmierung der sequentiellen Logik der State-Machine begonnen. Bei den hier erforderlichen komplexeren logischen Abläufen empfiehlt es sich, eine Simulation des Zusammenspiels von State-Machine und Aufsteckplatine durchzuführen. Eine derartige Simulation erfolgte mit dem Logik-Simulator der Xilinx-Foundation-Software, die zur Programmierung des FPGAs verwendet wurde. Folgende Abbildung zeigt das Ergebnis dieser Simulation.

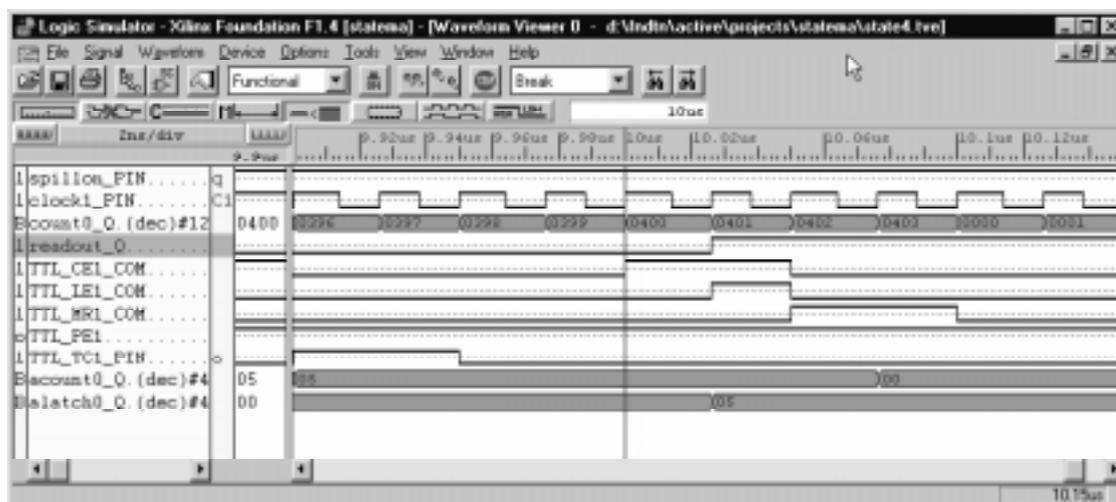


Abbildung 23 : Simulation der Ansteuerung der Aufsteckplatine durch die Hauptplatine. Zu sehen ist das Ende eines Meßzyklus: Nachdem 10 µs bei vorliegendem „Spill On“-Signal vergangen sind (= 400 clock-Zyklen der 40 MHz-clock), werden die Zähler zunächst über das CE*-Signal angehalten, die Zählerergebnisse mittels LE-Signal in die ECL-TTL-Übersetzer mit Latch übernommen und die Zählerbausteine anschließend über das MR-Signal zurückgesetzt. Gleichzeitig werden die in der State-Machine implementierten 4-Bit-breiten Zähler („count“) für den Überlauf der Zählerbausteine (TTL_TC*-Signale) in entsprechende 4-Bit-breite Latches („alatch“) ausgelesen und dann zurückgesetzt. Abschließend wird der interne Zähler der clock-Zyklen auf 0 zurückgesetzt, und der Meßzyklus kann von neuem beginnen. Ein „readout“-Flag signalisiert, daß nun Daten zur Auslese zur Verfügung stehen.

Nach der erfolgreichen Simulation der zu implementierenden Logik wurde für erste Tests die Ansteuerung eines Zähler-Kanals in der State-Machine implementiert. Es wurde weiterhin ein 16-Bit-breites Register im Xilinx eingerichtet, über das die 12-Bit-breiten Zählerergebnisse des zu testenden Kanals via VME-Bus ausgelesen werden können. Das „Spill On“-Signal, das später von der Beschleuniger-Kontrolle über die LEMO-Buchsen der State-Machine zugeführt werden kann, wurde für diese Tests vom VME-Bus aus gesteuert: Ein Zugriff auf die Adresse 700000h setzt das „Spill On“-Signal, ein Zugriff auf jede andere gültige Board-Adresse löscht es. Die korrekte Implementierung des Steuerzyklus‘ der State-Machine wurde wiederum mit dem Logikanalysator überprüft:

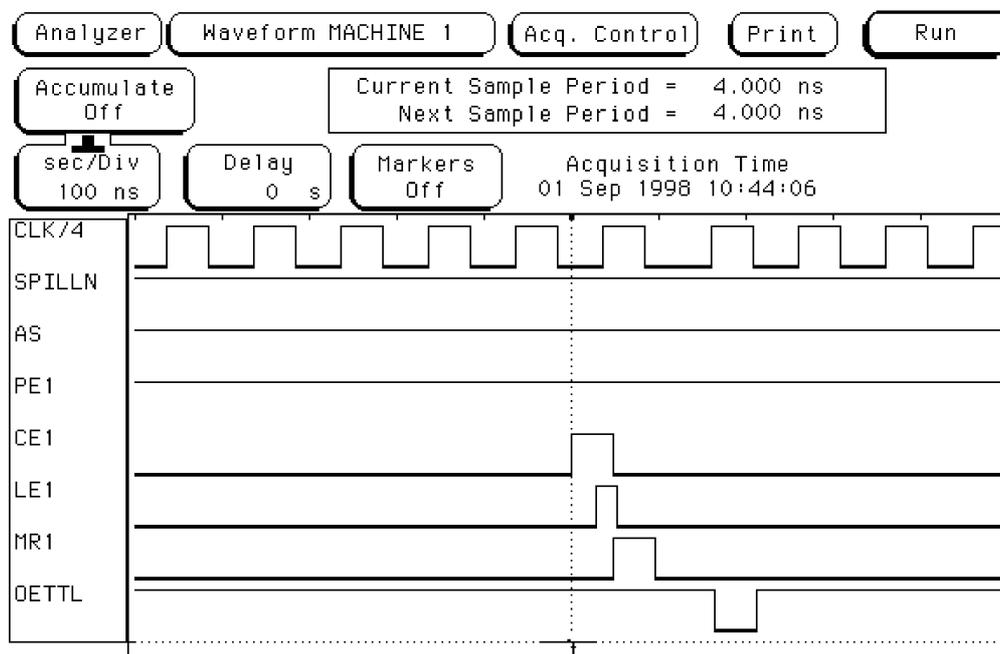


Abbildung 24 : Auslesezyklus analog der Simulation aus Abbildung 26. Zusätzlich sind mit CLK/4 die durch 4 geteilte Clock-Frequenz (Bit 1 des internen Zählers der clock-Zyklen) und das OETTL*-Signal zu sehen, das die Auslese des Ergebnisses aus dem angesprochenen Übersetzerbaustein mit Latch bewirkt.

Um auch während der Phasen, in der kein Strahl vorliegt, die Aufsteckplatine in einem definierten Zustand zu halten, wird bei Wegfallen des Spill On-Signals der Meßzyklus sofort gestoppt und die Steuersignale von Zähler und Übersetzerbausteinen wie folgt belegt:

- OETTL* = 1 (Deaktivieren der Latch-Ausgänge)
- LE = 0 (Sperren der Latch-Eingänge)
- PE* = 1
- CE* = 0
- MR = 1 (Zurücksetzen der Zähler; Ausgänge des Zählers werden auf 0 und das TC*-Signal wird auf 1 gesetzt.)

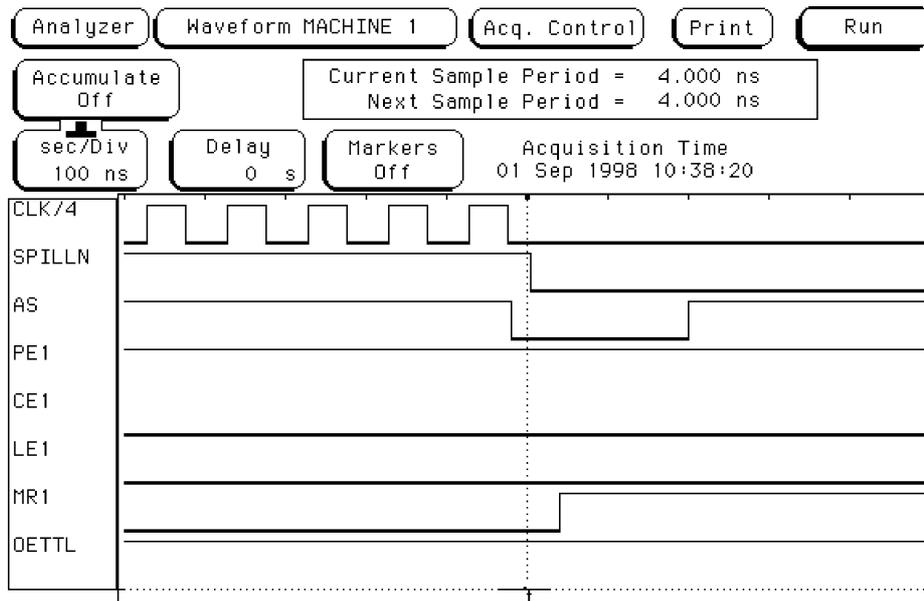


Abbildung 25 : Abbruch der Messung bei Wegfallen des „Spill On“ Signals.

Nachdem die Hauptplatine mit ihren grundlegendsten Funktionen erfolgreich getestet wurde, konnte nun die Inbetriebnahme der Aufsteckplatine erfolgen, die im nächsten Abschnitt dargestellt werden soll.

6.2 Inbetriebnahme der Aufsteckplatine

Da für die ersten Tests des Spillmonitor-Prototypen nur die Ansteuerung eines Zählerkanals vorgesehen war, wurde die Aufsteckplatine zunächst auch nur mit den für diesen Betrieb nötigen Komponenten bestückt (siehe Abbildung 18). Die für die Funktionsüberprüfung des Zählers erforderlichen Pulse wurden wie folgt generiert:

Mit einem Puls-Generator der Firma HP, der Signale mit bis zu 300 MHz Frequenz erzeugen kann, wurden zunächst NIM-Pulse (High-Pegel: 0 V, Low-Pegel: -1 V) der gewünschten Frequenz hergestellt, die dann in einem ECL-NIM-ECL Konverter der Firma Le Croy in differentielles ECL umgewandelt und via „twisted pair“ Kabel (wie auch im später vorgesehenen Betrieb) der Aufsteckplatine zugeführt wurden. Form und Frequenz der NIM-Pulse wurden dabei mit einem 500 MHz-Oszilloskop der Firma HP überwacht.



Abbildung 26: Testumgebung für die Inbetriebnahme der Aufsteckplatine : Oszilloskop, Pulsgenerator und Logik-Analysator befinden sich im oberen Teil des Bildes, links sind das NIM-Crate mit Konverter und das „twisted pair“ Kabel zu sehen.

Als erster Schritt wurde die korrekte Funktionsweise des Zählers bei einem 10 MHz-Eingangssignal (Anstiegs- und Abfallzeit: 3 ns , Breite: 50 ns, Periode 100 ns) über den Logikanalysator verifiziert: Zunächst wurde überprüft, ob der Zähler bei Aktivierung ohne Anhalten fehlerfrei von hexadezimal 00 bis ff durchläuft, um dann von vorne zu beginnen. Nachdem dies gezeigt war, wurde die im vorigen Abschnitt beschriebene Ansteuerung des Zählers durch die State-Machine kontrolliert:

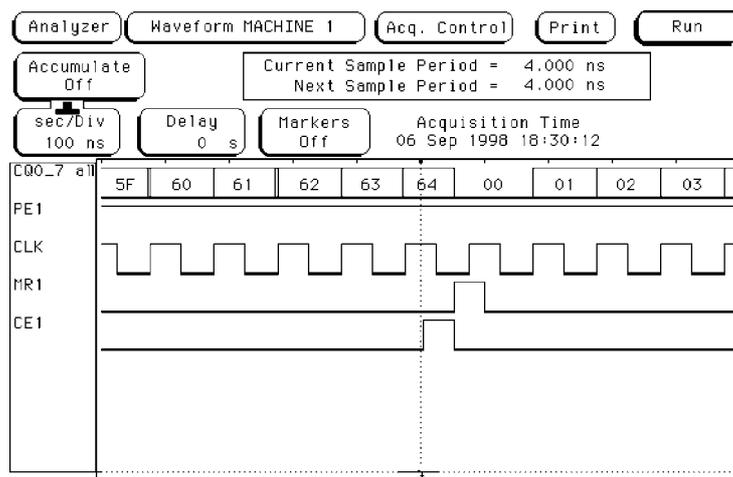


Abbildung 27: Ansteuerung des Zählerbausteins durch die State-Machine: Nach 10 µs wird der mit 10 MHz arbeitende Zähler angehalten und zurückgesetzt, so daß die Messung erneut beginnen kann. Wie man sieht, hat der Zähler Frequenz und Meßdauer entsprechend bis 100 (= 64 hexadezimal) gezählt. Abweichungen hiervon um 1 traten durch die fehlende Synchronisation von State-Machine und Eingangspulsen auf.

Bevor diese Zählerergebnisse auch über VME ausgelesen werden konnten, mußte noch ein Timing-Problem zwischen ECL-Zähler und Übersetzerbaustein mit Latch gelöst werden:

Wie man in Abbildung 24 erkennt, gibt es einen kurzen zeitlichen Überlapp (im ns-Bereich) zwischen aktivem MR-Signal des Zählerbausteins und fallender Flanke des LE-Signals (bei der die Daten im Latch „eingefroren“ werden). Diese kurze Zeitspanne reichte für das Zurücksetzen des Zählers und Signalpropagation zum Latch hin aus, so daß das Latch bei dieser Konfiguration immer nur den Wert Null übernahm. Das Problem konnte behoben werden, indem die State-Machine LE- und CE^* -Signal „gleichzeitig“ (innerhalb weniger ns) aktiviert und das LE-Signal vor dem CE^* -Signal (und ansteigendem MR-Signal) wieder deaktiviert. In Abbildung 24 würde das LE-Signal somit um einen internen Clock-Zyklus vorverlegt. Mit dieser neuen zeitlichen Abfolge der Steuersignale war es nun möglich, die Zählerergebnisse über VME auszulesen.

Darauf aufbauend konnte getestet werden, ob der Zählerbaustein auch im gewünschten Frequenzbereich von einigen 100 MHz fehlerfrei arbeitet. Zunächst wurde die Frequenz auf 100 MHz (Signalbreite: 5 ns, Anstiegs- und Abfallzeit: 1 ns, Periode: 10 ns) erhöht. Bei dieser Frequenz wurden über VME immer wieder die Zahlenwerte 3E8h (=1000) und 3E7h (=999) ausgelesen, was die einwandfreie Funktion des Zählers auch bei dieser Frequenz verifiziert. Insbesondere bestätigte dies den fehlerfreien Überlauf des 8-Bit-breiten Zählers und die korrekte Verarbeitung dieses Überlauf-Signals in der State-Machine.

Abschließend wurde der Zähler bis an die Leistungsgrenze des Pulsgenerators bei 250 MHz (am Oszilloskop wurden um 247 MHz schwankende Frequenzen gemessen) mit 2 ns Pulsbreite, 670 ps Anstiegs- und Abfallzeit und einer Periodizität von 4 ns betrieben.

Höhere Frequenzen konnten nicht stabil generiert werden, wie die Kontrollmessungen mit dem Oszilloskop zeigten.

Innerhalb dieses testbaren Bereiches arbeitet der Spillmonitor einwandfrei, wie die folgenden Abbildungen belegen:

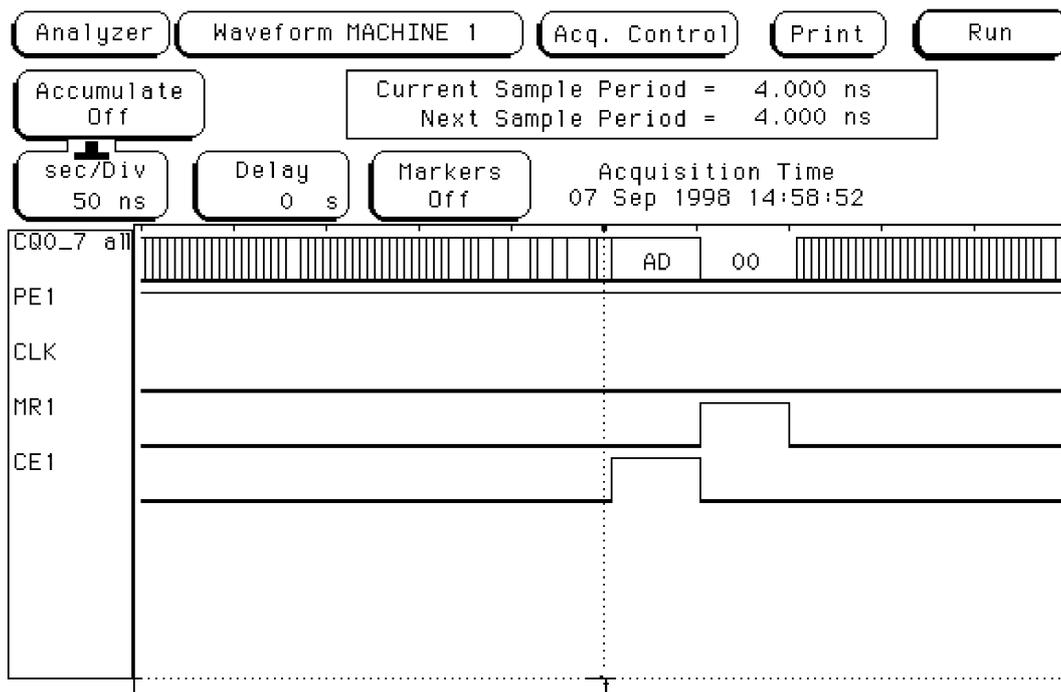


Abbildung 28: Meßzyklus des ECL-Zählers bei ca. 247 MHz. Das Clock-Signal konnte vom Logik-Analysator nicht mehr verarbeitet werden und wird deshalb nicht angezeigt.

```

Telnet - 134.176.17.128
Verbinden Bearbeiten Terminal ?
1
Debug-Programm fuer den Spillmonitor
Worauf wollen sie zugreifen?
[1] VME-Basis (Statusregister)
[2] Switching - Lattice 1 (Statusregister)
[3] Switching - Lattice 2 (Statusregister)
[4] Xilinx / Statemachine (Statusregister)
[5] Memory - Bank 1
[6] Memory - Bank 2
[7] user defined address
[0] LET ME OUT!!!
7
Sie haben 7 eingegeben
Auf welche Adresse soll geschrieben werden?
ff700000
Die eingegebene Adresse lautet: ff700000.
Wollen Sie auf diese Adresse [1] lesen oder [2]schreiben ?1
Diese Adresse enthaelt den Wert 9ad
1- nochmal      2- Abbruch

```

Abbildung 29 : Auslesen des Zählerwerts via VME bei einer Meßdauer von 10µs und ca. 247 MHz. Der dezimale Wert des ausgelesenen Ergebnisses entspricht 2477.

7 Ausblick

Der entwickelte Prototyp eines Spillmonitors erfüllt die Anforderungen, die im Konzept an ihn gestellt wurden. Mit den im institutseigenen Elektronik-Labor zur Verfügung stehenden Mitteln konnte insbesondere gezeigt werden, daß der Spillmonitor in der Lage ist, die erforderlichen Signalfrequenzen von mehreren 100 MHz zu verarbeiten.

Das VME-Interface arbeitet problemlos und erlaubt den Zugriff auf alle Funktionsgruppen über Steuerleitungen, Kontroll- und Statusregister. Durch den DSP sind vielfältige Analysen der gemessenen Strahlstruktur (z.B. FFT) möglich. Auch die Datenaufnahme des Spillmonitors selbst kann in einem großen Rahmen über die Breite und zeitliche Position des „Feinstrukturfensters“ oder die einstellbare Dauer des Meßintervalls frei konfiguriert werden. Über regulierbare Schwellen kann der Spillmonitor online messen, ob die vorliegende Strahlstruktur den Anforderungen des HADES-Experimentes entspricht und dies z.B. den Operateuren des SIS mitteilen. Sämtliche Daten des Spillmonitors können über das VME-Interface ausgelesen werden und stehen z.B. dem Kontrollraum über TCP/IP zur Verfügung. Auch eine Anbindung an das Datenaufnahme-System und die noch in Planung befindliche zentrale Überwachung („Slow Control“) von HADES ist somit möglich.

Besonders hervorzuheben ist hier nochmals die Flexibilität, die durch die z.T. redundante Vernetzung verschiedener programmierbarer Bausteine erreicht wurde. Hierdurch sind auch für spätere Erweiterungen und Veränderungen des Spillmonitors noch Entwicklungsmöglichkeiten vorhanden.

Erste Einsätze des Spillmonitors, z.B. während der nächsten Teststrahlzeiten von HADES Ende 1998, sind bereits geplant. Dabei ist ausdrücklich zu bemerken, daß der Spillmonitor unabhängig vom verwendeten Startdetektortyp eingesetzt werden kann, da er lediglich standardisierte differentielle ECL-Signale als Eingang benötigt. Damit ist der Spillmonitor allgemein für alle geplanten HADES-Experimente einsetzbar, z.B. auch bei den vorgesehenen Experimenten mit dem Pionenstrahl der GSI. Bei diesen Experimenten werden Plastiksintillatoren als Startdetektoren zum Einsatz kommen, und es ist weniger auf Spikes in der Strahlstruktur wie bei den Ionenstrahlen, sondern mehr auf einen kontinuierlichen Strahlstrom zu achten, worauf der Spillmonitor ebenfalls programmiert werden kann.

8 Literaturverzeichnis

1. „Proposal for a **H**igh-**A**cceptance **D**i-**E**lectron Spectrometer“ 1994
2. W.Cassing, V.Metag, U. Mosel, K. Niita : Phys Rep188 (1990) 363
3. C. Caso et al.: „Review of Particle Physics“, The European Physical Journal **C3** (1998) 1
4. M. Hermann et al., Nucl. Phys. A560 (1993) 411
5. Asakawa et al., Phys. Rev. C46 (1992) R1159
6. S. Klimt et al., Phys. Lett. B249 (1990) 386
7. Heike Schön, „HADES Ein Dileptonenspektrometer hoher Akzeptanz“, Dissertation an der Johann-Wolfgang-Goethe-Universität Frankfurt am Main, Institut für Kernphysik, 1995
8. A. Yegneswaran et al. NIM A290 (1990) 61 und darin enthaltene Referenzen
9. CERN-Report : DRDC 94-21/P56
10. E. Berdermann et al, „Diamond Detectors for Heavy Ion Measurements“, XXXVI International Meeting on Nuclear Physics, Bormio, (1998)
11. P. Moritz et al., „Diamond Detectors for Beam Diagnostics in Heavy Ion Accelerators“, DIPAC III, Frascati (1997)
12. „HADES Start Detector - some remarks by W. Koenig, 20.5.97, updated 10.11.97“ on dslep01.gsi.de directory: /nfs/dslep01/dk3a/user/hades/WWW/docs/start/start_W6.doc
13. Entnommen aus den öffentlichen Webseiten der GSI : www.gsi.de/accelerator/main.html
14. K. Wille : „Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen“, Teubner Studienbücher, 2.Auflage 1996 (ISBN 3-519-13087-4)
15. Entnommen aus den öffentlichen Webseiten der GSI : www.gsi.de/accelerator/sis.p14.html
16. P. Moritz et al., „Diamond Detectors with Subnanosecond Time Resolution for Heavy Ion Spill Diagnostics“, BIW 98, Stanford, 1998
17. Private Kommunikation mit Peter Moritz, GSI
18. Marc-André Pleier, „Specifications for a Spill Monitor for HADES“, www.physik.uni-giessen.de/hades/groups/spill/spec/SPILLMONITOR_1.html
19. Motorola: „High Performance ECL Data ECLinPS and ECLinPS Lite“, DL140/D, Rev 4
20. ANALOG DEVICES : „ADSP-2106x SHARC™ User’s Manual“, Second Edition
21. Siehe öffentliche Literatur-Webseiten von LATTICE : www.latticesemi.com

22. Xilinx : „The Programmable Logic Data Book“, 1994
bzw. dessen aktuelle Entsprechung in den Xilinx-Webseiten : www.xilinx.com
23. Xilinx : „The ISP Application Guide and CPLD Data Book“, May 1997
24. Torsten Geist, „Entwicklung einer PC-Einsteckkarte zur Funktionskontrolle von TDC-ASICs“, Diplomarbeit an der FH Gießen-Friedberg FB Elektrotechnik 1, 1997
25. Stefan Plüschke, „Entwicklung eines schnellen Schauererkennungsprozessors als 2. Triggerstufe für TAPS“, Diplomarbeit an der JLU Gießen II. Phys. Institut, 1997
26. Hans-Joachim Blank, „Logikbausteine - Grundlagen, Programmierung und Anwendung“, Markt & Technik Verlag AG, 1992 (ISBN 3-87791-072-6)
27. W. D. Peterson, „The VMEbus Handbook“, VITA Verlag, 1993

9 Anhang

9.1 ABEL-Quelltexte

9.1.1 VME-Interface

```
MODULE vmeneu
DECLARATIONS

"Pin-Zuweisungen :

"VME Adreßleitungen mit Treiber-Kontrollbits
  ACLK,A1,A2,A3,A19      pin 25,26,28,29,30;
  AOE,A20,A21,A22,A23   pin 36,32,33,34,35;
  A24,A25,A26,A27,A28   pin 89,90,92,93,94;
  A29,A30,A31           pin 95,96,105;
  "Es werden hier momentan nur die Bits 20-23 zum Decodieren benutzt.

"VME Datenleitungen mit Treiber-Kontrollbits
  D0,D1,D2,D3,D4       pin 37,38,39,40,41;
  D5,D6,D7             pin 42,43,44;
  DOE1,DOE2,DDIR       pin 47,49,48;

"VME-Adreßmodifier
  AM0,AM1,AM2          pin 57,58,59;
  AM3,AM4,AM5          pin 60,61,62;

"VME-Kontrollsignale
  DS0,DS1              pin 50,52;
  LWORD                pin 53;
  WRITE                pin 54;
  AS                   pin 55;
  BERR                 pin 73;
  DTACK                pin 72;

"Delay-Pins für DTACK
  Del1,Del2,Del3       pin 65,66,67;
  Del4,Del5,DelIn     pin 68,69,70;
  "Hier wird nur DelIn mit Del5 benötigt = Delay um 100 ns

"Interrupt-Signale
  IACK,IACKIN,IACKOUT  pin 78,79,80;
  IRQ1,IRQ6            pin 74,75;
  "Interrupt-Handling ist noch nicht implementiert => feste Werte

"LED-Kontrollpins
  LED6,LED7,LED8,LED9  pin 82,83,84,85;

"Xilinx-Bootpins
  INIT                 pin 17;
  DONE                 pin 16;
  PROGRAM              pin 15;
  DIN                  pin 14;
  DOUT                 pin 13;
  CCLK                 pin 11;
  BUSY                 pin 9;

"Steuersignale für VME-Zugriff auf Xilinx-Register
  XVWRITE              pin 8;
  XVREAD               pin 7;
  XVBSSEL              pin 6;
```

```

        XVDSTRB1,XVDSTRB2      pin 5,4;

"frei definierbare Kontrollpins am Xilinx
    XVDEF0,XVDEF1             pin 3,2;
    XVDEF2,XVDEF3             pin 160,159;

"Steuersignale für VME-Zugriff auf die Switching-Matrix
    SVBSEL                     pin 158;
    OVERRIDE                   pin 157;

"Memory-Steuersignale
    SVWE                       pin 156;
    SVOE                       pin 155;
    SVLB                       pin 154;
    SVUB                       pin 153;

"Chip-Select Bits für das Memory
    SVCS0,SVCS1                pin 152,150;
    SVCS2,SVCS3                pin 149,148;

"frei definierbare Kontrollpins der Switching-Matrix
    SVDEF0,SVDEF1              pin 147,146;
    SVDEF2,SVDEF3              pin 145,144;

"frei definierbare Kontrollpins
    free1,free2                pin 110,112;

"DSP Tri State Kontrolle
    SBTS                       pin 120;

"Adressdefinitionen
    Basis_Adresse               = ^h1;
    "Zugriff auf das Register im VME-LATTICE
    Switch1_Adresse             = ^h2;
    "Zugriff auf den 1. Switch-Lattice (Register)
    Xilinx_Adresse              = ^h4;
    "Zugriff auf den Xilinx (State-Machine)
    Mem1_Adresse                = ^h5;
    "Zugriff auf MemBank A
    Mem2_Adresse                = ^h6;
    "Zugriff auf MemBank B

"Nodes
    Basis_Zugriff               node;
    Switch1_Zugriff             node;
    Xilinx_Zugriff              node;
    Mem1_Zugriff                node;
    Mem2_Zugriff                node;
    Allg_Zugriff                node;
    Int_Zugriff                 node;
    status7..status0           node  istype 'reg_d, buffer';
    switch                      node  istype 'reg_d, buffer';
    spillon                     node  istype 'reg_d, buffer';

"Aliases
    ADR                         = [A23..A20];
    "oberste 4 Adreßbits bei 24 Bit Zugriff
    X,Z                         = .X.,.Z.;
    data                        = [D7..D0];
    status                      = [status7..status0];

EQUATIONS
" Wird das board angesprochen?
    Basis_Zugriff               = (ADR == Basis_Adresse) & !AS;
    Switch1_Zugriff             = (ADR == Switch1_Adresse) & !AS;

```

```

        Xilinx_Zugriff          = ((ADR == Xilinx_Adresse) # (ADR == ^h7)) &
!AS;
        Mem1_Zugriff           = (ADR == Mem1_Adresse) & !AS;
        Mem2_Zugriff           = (ADR == Mem2_Adresse) & !AS;
        Int_Zugriff            = Switch1_Zugriff # Xilinx_Zugriff #
Mem1_Zugriff # Mem2_Zugriff;
        Allg_Zugriff           = Basis_Zugriff # Switch1_Zugriff #
Xilinx_Zugriff # Mem1_Zugriff # Mem2_Zugriff;

"Erzeugung der VME-Signale
        DelIn                   = Allg_Zugriff & (!DS0 # !DS1);
        DTACK                    = Del5;
        BERR                      = 0;
        IRQ1                      = 0;
        IRQ6                      = 0;
        IACKOUT                   = 0;

"Kontrolle der Daten- und Adreßtreiber
        !AOE                      = Int_Zugriff;
        ACLK                      = Int_Zugriff;
        DDIR                      = Int_Zugriff & !WRITE;
        !DOE1                     = Int_Zugriff & (!DS0 # !DS1);
        !DOE2                     = Int_Zugriff & (!DS0 # !DS1);

"Select-Signale für die verschiedenen Funktionsgruppen (z.T. Low-aktiv!)
        !SVBSEL                   = Switch1_Zugriff;
        XVBSEL                   = Xilinx_Zugriff;
        XVWRITE                   = !WRITE;
        XVREAD                    = WRITE;

" DSP aus!!!!
        SBTS                      = 0;

"Memory-Kontrolle
        !SVOE                     = Mem1_Zugriff # Mem2_Zugriff;
        !SVCS1                   = Mem1_Zugriff # Mem2_Zugriff;
        SVWE                      = WRITE;

WHEN (ADR == Mem1_Adresse) THEN switch.d = 1;
        switch.clk                = !AS;

WHEN (ADR == ^h7) THEN spillon.d = 1;
        spillon.clk              = !AS;
        SVDEF0                   = switch;
        XVDEF0                   = spillon;
        free1                    = SVDEF0;
        SVDEF1                   = AS;
        "Zum Umschalten des internen Switch-Registers.

"Bidirektionales Statusregister
        status.d                  = data;
        data                      = status;
        status.clk                = Basis_Zugriff & !WRITE;
        data.oe                   = Basis_Zugriff & WRITE;

END

```

9.1.2 Switching-Matrix (Lattice 1 : Daten)

```
MODULE swlneu
```

DECLARATIONS

```

"VME-Datenleitungen
  VD0,VD1,VD2,VD3,VD4      pin 61,60,59,58,57;
  VD5,VD6,VD7,VD8,VD9     pin 56,55,54,53,52;
  VD10,VD11,VD12,VD13     pin 50,49,48,47;
  VD14,VD15                pin 46,44;

"Xilinx-Datenleitungen
  XD0,XD1,XD2,XD3,XD4     pin 88,87,86,85,84;
  XD5,XD6,XD7,XD8,XD9     pin 83,82,80,79,78;
  XD10,XD11,XD12,XD13     pin 77,76,75,74;
  XD14,XD15                pin 73,72;

"Memory-Datenleitungen
  MAD0,MAD1,MAD2,MAD3      pin 115,114,113,112;
  MAD4,MAD5,MAD6,MAD7      pin 110,109,108,106;
  MAD8,MAD9,MAD10,MAD11    pin 105,96,95,94;
  MAD12,MAD13,MAD14,MAD15  pin 93,92,90,89;

  MBD0,MBD1,MBD2,MBD3      pin 150,149,148,147;
  MBD4,MBD5,MBD6,MBD7      pin 146,145,144,142;
  MBD8,MBD9,MBD10,MBD11    pin 141,140,139,138;
  MBD12,MBD13,MBD14,MBD15  pin 137,136,135,134;

"Chip-Select
  MBACS1                    pin 120;
  MBBCS1                    pin 156;
  SVCS1                     pin 98;

"Interner Bus
  SIB0,SIB1,SIB2,SIB3      pin 37,36,35,34;
  SIB4,SIB5                pin 33,32;

"LED's
  LED10,LED11              pin 16,15;

"Freie Pins
  free1,free2              pin 64,65;

"Kontrolleleitungen
  SVBSEL                    pin 43;
  OVERRIDE                  pin 42;

"Nodes
  status15..status0        node istype 'reg_d, buffer';

"Aliases
  vdat                      = [VD15..VD0];
  madat                     = [MAD15..MAD0];
  mbdat                     = [MBD15..MBD0];
  xdat                      = [XD15..XD0];
  status                    = [status15..status0];

"Interner Bus-Belegung:
"   SIB0                    =   SVDEF0;
"   SIB1                    =   SVOE;
"   SIB2                    =   SVWE;
"   SIB3                    =   XMOE;
"   SIB4                    =   XMWE;

```

EQUATIONS

```

WHEN (SIB0 == 1)
  THEN
  {
    vdat                      =   madat;
  }

```

```

        madat                =      vdat ;

        vdat.oe              =      (!SIB1 & SIB2);
        madat.oe            =      (!SIB1 & !SIB2);
        MBACS1              =      SVCS1;

        xdat                =      mbdatt;
        mbdatt              =      xdat ;

        xdat.oe              =      (!SIB3 & SIB4);
        mbdatt.oe           =      (!SIB3 & !SIB4);
        MBBCS1              =      SIB3;
    }

WHEN (SIB0 == 0)
    THEN
    {
        vdat                =      mbdatt;
        mbdatt              =      vdat ;

        vdat.oe              =      (!SIB1 & SIB2);
        mbdatt.oe           =      (!SIB1 & !SIB2);
        MBBCS1              =      SVCS1;

        xdat                =      madatt;
        madatt              =      xdat ;

        xdat.oe              =      (!SIB3 & SIB4);
        madatt.oe           =      (!SIB3 & !SIB4);
        MBACS1              =      SIB3;
    }

END

```

9.1.3 Switching-Matrix (Lattice 2 : Adressen)

```

MODULE sw2neu
DECLARATIONS
"VME-Adressen
    VA1,VA2,VA3,VA4          pin 68,67,66,65;
    VA5,VA6,VA7,VA8          pin 64,62,61,60;
    VA9,VA10,VA11,VA12      pin 59,58,57,56;
    VA13,VA14,VA15,VA16     pin 55,54,53,52;
    VA17,VA18                pin 50,49;

"Xilinx-Adressen
    XMA1,XMA2,XMA3,XMA4      pin 88,87,86,85;
    XMA5,XMA6,XMA7,XMA8      pin 84,83,82,80;
    XMA9,XMA10,XMA11,XMA12   pin 79,78,77,76;
    XMA13,XMA14,XMA15,XMA16  pin 75,74,73,72;
    XMA17,XMA18              pin 70,69;

"Memory-Adressen
    MAA1,MAA2,MAA3,MAA4      pin 89,90,92,93;
    MAA5,MAA6,MAA7,MAA8      pin 94,95,96,105;
    MAA9,MAA10,MAA11,MAA12   pin 106,108,109,110;
    MAA13,MAA14,MAA15,MAA16  pin 112,113,114,115;
    MAA17,MAA18              pin 116,117;

    MBA1,MBA2,MBA3,MBA4      pin 122,123,124,126;
    MBA5,MBA6,MBA7,MBA8      pin 127,128,129,130;
    MBA9,MBA10,MBA11,MBA12   pin 132,133,134,135;

```

```

MBA13,MBA14,MBA15,MBA16      pin 136,137,138,139;
MBA17,MBA18                  pin 140,141;

"Interner Bus (zum Durchschleifen der Steuersignale)
  SIB0,SIB1,SIB2,SIB3        pin 6,7,8,9;
  SIB4,SIB5                  pin 11,13;

"Frei definierbare Leitungen v. VME (für Switch-Signal)
  SVDEF0,SVDEF1              pin 44,46;

"Frei definierbare Leitungen vom Xilinx
  XSDEF0,XSDEF1              pin 43,42;

"Memory-Kontrollsignale
  MBAWE                      pin 149;
  MBALB                      pin 148;
  MBAUB                      pin 147;
  MBAOE                      pin 146;

  MBBWE                      pin 154;
  MBBLB                      pin 153;
  MBBUB                      pin 152;
  MBBOE                      pin 150;

"VME-Memory-Kontrollsignale
  SVWE                       pin 35;
  SVLB                       pin 33;
  SVUB                       pin 32;
  SVOE                       pin 34;

"Xilinx-Memory-Kontrollsignale
  XMWE                       pin 97;
  XMLB                       pin 102;
  XMUB                       pin 103;
  XMOE                       pin 98;

"Aliases
  vadr                       = [VA18..VA1];
  xadr                       = [XMA18..XMA1];
  maadr                      = [MAA18..MAA1];
  mbadr                      = [MBA18..MBA1];

EQUATIONS
"Da nur 16-Bit Zugriffe stattfinden:
  MBALB                      = 0;
  MBAUB                      = 0;
  MBBLB                      = 0;
  MBBUB                      = 0;

"Durchschleifen der Memory-Steuersignale zum 2. LATTICE
"VME:
  SIB0                      = SVDEF0;
  SIB1                      = SVOE;
  SIB2                      = SVWE;

"Xilinx:
  SIB3                      = XMOE;
  SIB4                      = XMWE;

"Steuerung für internes Register
  SIB5                      = SVDEF1;

"Umschalten jetzt über VME-Bit; wird später v. Xilinx gesteuert werden

```

```

WHEN (SVDEF0 == 1)
  THEN
  {
    maadr          =    vadr;
    mbadr          =    xadr;
    MBAWE          =    SVWE;
    MBAOE          =    SVOE;
    MBBWE          =    XMWE;
    MBBOE          =    XMOE;
  }

WHEN (SVDEF0 == 0)
  THEN
  {
    maadr          =    xadr;
    mbadr          =    vadr;
    MBAWE          =    XMWE;
    MBAOE          =    XMOE;
    MBBWE          =    SVWE;
    MBBOE          =    SVOE;
  }

END

```

9.1.4 State-Machine

module maching

"Grundlegende State-Machine des Spillmonitors mit Ansteuerung eines
 "Zählerkanals von Marc-André Pleier , II. Phys. Inst. JLU GIESSEN

DECLARATIONS

"Pin-Definitionen

```

clock1          pin 204 istype 'buffer';      "= 40 MHz
clock2          pin 4;                        "= 20 MHz

```

```

TTL_TC1        pin 31;
"= Überlauf von Kanal 1

```

```

OETTL1         pin 13 istype 'reg_d';
"= Enable Signal für Kanal 1-Auslese

```

```

TTL_MR1        pin 128 istype 'reg_d';
"= MR der ersten 8 Zähler

```

```

TTL_PE1        pin 124 istype 'reg_d';
TTL_CE1        pin 132 istype 'reg_d';
TTL_LE1        pin 126 istype 'reg_d';
spillon        pin 172;
"=XVDEF0 => Steuerung über VME im Testbetrieb

```

```

CQ0,CQ1,CQ2,CQ3      pin 12,11,10,9;
CQ4,CQ5,CQ6,CQ7      pin 8,7,6,5;
"=Datenbus der ECL-Zähler-Ergebnisse

```

```

VD0,VD1,VD2,VD3      pin 89,91,92,93;
VD4,VD5,VD6,VD7      pin 94,95,96,97;
VD8,VD9,VD10,VD11    pin 98,99,100,109;
VD12,VD13,VD14,VD15  pin 110,111,112,113;
"=Verbindung VME-Xilinx

```

```

" Xilinx-VME Steuersignale
XVBSEL                pin 122;
XVREAD                pin 121;
XVWRITE               pin 120;

free1,free2           pin 199,198;

"Konstantendefinitionen:
dauer                 = 400;
"=Dauer eines Meßzyklus * 25 ns

"Nodes
count11..count0       node istype 'reg_d, buffer';
"Hauptzähler für Aktionsabfolge

acount3..acount0      node istype 'reg_d, buffer';
"4-Bit Ergänzungs-zähler für Kanal 1

wert15..wert0         node istype 'reg_d, buffer';
"Register für gesamten Zählerwert

readout               node istype 'reg_d, buffer';
"Signal zur Datenauslese

ueberlauf             node istype 'reg_d, buffer';

"Aliases
count                 = [count11..count0];
acount                = [acount3..acount0];
CQ                    = [CQ7..CQ0];
VD                    = [VD15..VD0];
VDlow                 = [VD7..VD0];
VDhigh                = [VD11..VD8];
wert                  = [wert15..wert0];
wertlow               = [wert7..wert0];
werthigh              = [wert11..wert8];
wertrest              = [wert15..wert12];

EQUATIONS

"Einrichten der Zähler
TTL_MR1.clk           = clock1;
TTL_PE1.clk           = clock1;
TTL_CE1.clk           = clock1;
TTL_LE1.clk           = clock1;
OETTL1.clk            = clock1;

ueberlauf.clk         = clock1;

count.d               = (count +1) & spillon & !ueberlauf;
count.d               = 0 & (ueberlauf # !spillon);
count.clk              = clock1;

acount.d              = (acount +1) & spillon & !ueberlauf;
acount.d              = 0 & (ueberlauf # !spillon);
acount.clk             = TTL_TC1 & spillon & !ueberlauf;
acount.clk            = clock1 & (ueberlauf # !spillon);

wertrest.d            = 0;
wertrest.clk          = clock1;

werthigh.d            = acount;
werthigh.clk          = TTL_LE1;

```

```

wertlow.d           = CQ;
wertlow.clk        = !OETTL1;

readout.clk        = TTL_LE1 & spillon;
readout.clk        = clock1 & !spillon;

```

"Zustand vor Meßbeginn

```

When (spillon == 0)
  THEN
  {
  TTL_MR1.d         = 1;
  TTL_PE1.d         = 1;
  TTL_CE1.d         = 0;
  TTL_LE1.d         = 0;
  ueberlauf.d       = 0;
  readout.d         = 0;
  }

```

"Messung

```

When ( (spillon == 1) & (count < dauer) )
  THEN
  {
  TTL_MR1.d         = 0;
  TTL_PE1.d         = 1;
  TTL_CE1.d         = 0;
  TTL_LE1.d         = 0;
  ueberlauf.d       = 0;
  readout.d         = 0;
  }

```

```

When ( (spillon == 1) & (count == dauer) )
  THEN
  {
  TTL_MR1.d         = 0;
  TTL_PE1.d         = 1;
  TTL_CE1.d         = 1;
  TTL_LE1.d         = 1;
  ueberlauf.d       = 0;
  readout.d         = 1;
  }

```

```

When ( (spillon == 1) & (count == (dauer + 1)))
  THEN
  {
  TTL_MR1.d         = 0;
  TTL_PE1.d         = 1;
  TTL_CE1.d         = 1;
  TTL_LE1.d         = 0;
  readout.d         = 1;
  ueberlauf.d       = 0;
  }

```

```

When ( (spillon == 1) & (count >= (dauer +2) ) )
  THEN
  {
  TTL_MR1.d         = 1;
  TTL_PE1.d         = 1;
  TTL_CE1.d         = 0;
  TTL_LE1.d         = 0;
  readout.d         = 1;
  ueberlauf.d       = 1;
  }

```

```

    }

"Auslese
When ((readout == 1) & ((count == 1) # (count == 2)))
    THEN
    {
    OETTLL1.d          = 0;
    }
    ELSE
    {
    OETTLL1.d          = 1;
    }

"Register zur Datenauslese via VME
VD                    = wert;
VD.oe                 = XVBSEL & XVREAD;

free1                  = count1;          "Debug
free2                  = spillon;

end machining

```

9.2 Verwendete C-Programme

9.2.1 Ansteuerung des VME-Busses unter OS9

```

# include <stdio.h>
# include <strings.h>
# define basisadr      0xff100000
# define switch1adr   0xff200000
# define switch2adr   0xff300000
# define xilinxadr    0xff400000
# define memadr1      0xff500000
# define memadr2      0xff600000

int nr,c,adresse,daten,ergebnis;
void schreiben(int adresse,int daten);
int lesen(int adresse);

main()
{

do
{

printf("\n Debug-Programm fuer den Spillmonitor");
printf("\n Worauf wollen sie zugreifen?");
printf("\n [1] VME-Basis (Statusregister)");
printf("\n [2] Switching-Lattice 1 (Statusregister)");
printf("\n [3] Switching-Lattice 2 (Statusregister)");
printf("\n [4] Xilinx / State-Machine (Statusregister)");
printf("\n [5] Memory-Bank 1");
printf("\n [6] Memory-Bank 2");
printf("\n [7] user defined address");
printf("\n [0] LET ME OUT!!!\n");
scanf("\n %d",&nr);

```

```

printf("Sie haben %d eingegeben",nr);
switch (nr)
{
case 0: return;
break;
case 1: adresse = basisadr;
break;
case 2: adresse = switchladr;
break;
case 3: adresse = switch2adr;
break;
case 4: adresse = xilinxadr;
break;
case 5: adresse = memadr1;
break;
case 6: adresse = memadr2;
break;
case 7: {printf("\n Auf welche Adresse soll geschrieben werden?\n");
scanf("%x",&adresse);}
break;
default: printf("\n Toller Witz!");return;
break;
}

printf("\n Die eingegebene Adresse lautet: %0x.",adresse);

printf("\n Wollen Sie auf diese Adresse [1] lesen oder [2]schreiben ?");
scanf("\n %d",&nr);
switch (nr)
{
case 1: { ergebnis=lesen(adresse);
printf("\n Diese Adresse enthaelt den Wert %x",ergebnis);}
break;
case 2: { printf("\n Welchen Wert (16 Bit!) moechten Sie
schreiben\n?");
scanf("%x",&daten);
printf("Marc hat %x (0x) bzw. %d
weggeschrieben\n",daten,daten);
schreiben(adresse,daten);}
break;
default: printf("\n Toller Witz!"); return;
}

printf("\n\n 1- nochmal \t 2- Abbruch\n ");
scanf("%d",&c);
}
while (c != 2);
}

void schreiben(int adresse,int daten)
{
*((unsigned short *)adresse)=daten;
return;
}

int lesen(int adresse)
{
daten=*((unsigned short *)adresse);
return daten;
}

```

9.2.2 Speicher-Testprogramm 1

```
# include <stdio.h>
# include <strings.h>

# define memadr1          0xff500000
# define memadr2          0xff600000

int nr,c,adresse,daten,ergebnis;
void schreiben(int adresse,int daten);
int lesen(int adresse);
main()
{
do
{
adresse=memadr1;
printf("\n Testprogramm für das Memory\n");
for (nr=0; nr<=100; nr++)
{
printf("Es wird an %x der Wert %d geschrieben\r",adresse,nr);
schreiben(adresse,nr);
adresse=adresse + 2;
}
adresse=memadr1;
for (nr=0; nr<=100; nr++)
{
printf("\nEs wurde an %x der Wert %d
gelesen.",adresse,lesen(adresse));
if (lesen(adresse) == nr) printf("\n O.K.!");
adresse=adresse + 2;
}

printf("\n\n\n 1- nochmal \t 2- Abbruch\n ");
scanf("%d",&c);
}
while (c != 2);
}

void schreiben(int adresse,int daten)
{
*((unsigned short *)adresse)=daten;
return;
}

int lesen(int adresse)
{
daten=*((unsigned short *)adresse);
return daten;
}
```

9.2.3 Speicher-Testprogramm 2

```
# include <stdio.h>
# include <strings.h>

# define memadr1          0xff500000
# define memadr2          0xff600000

int nr,c,adresse,daten,ergebnis,fehler;
void schreiben(int adresse,int daten);
int lesen(int adresse);
```

```

main()
{

do
{
    adresse=memadr1;
    printf("\n Testprogramm für das Memory\n");
    for (nr=0; nr<128; nr++)
    {
        fehler=0;
        for (c=0; c<1024; c++)
        {

            schreiben(adresse,0xffff-(nr*128+c));
            if (lesen(adresse)!=0xffff-(nr*128+c))
            {
                fehler++;
                printf("\n An Adresse %x ist der Wert %x statt %x
aufgetreten",adresse,lesen(adresse),0xffff-(nr*128+c));
            }
            adresse+=2;
        }

        if (fehler) printf("\n Es sind Fehler aufgetreten");

    }

    printf("\n\n\n 1- nochmal \t 2- Abbruch\n ");
    scanf("%d",&c);
}
while (c != 2);
}

void schreiben(int adresse,int daten)
{
*((unsigned short *)adresse)=daten;
return;
}

int lesen(int adresse)
{
daten=*((unsigned short *)adresse);
return daten;
}

```

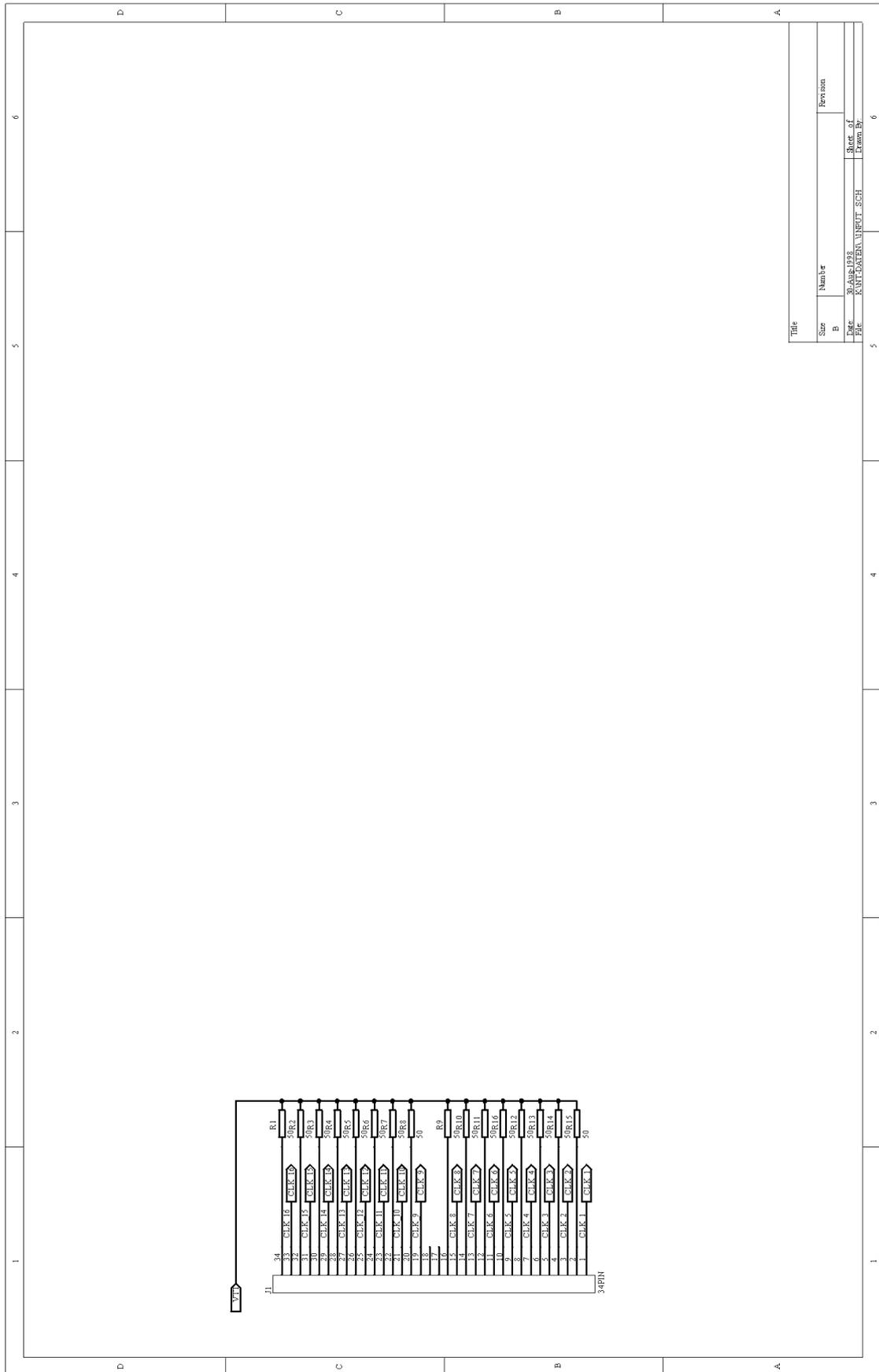
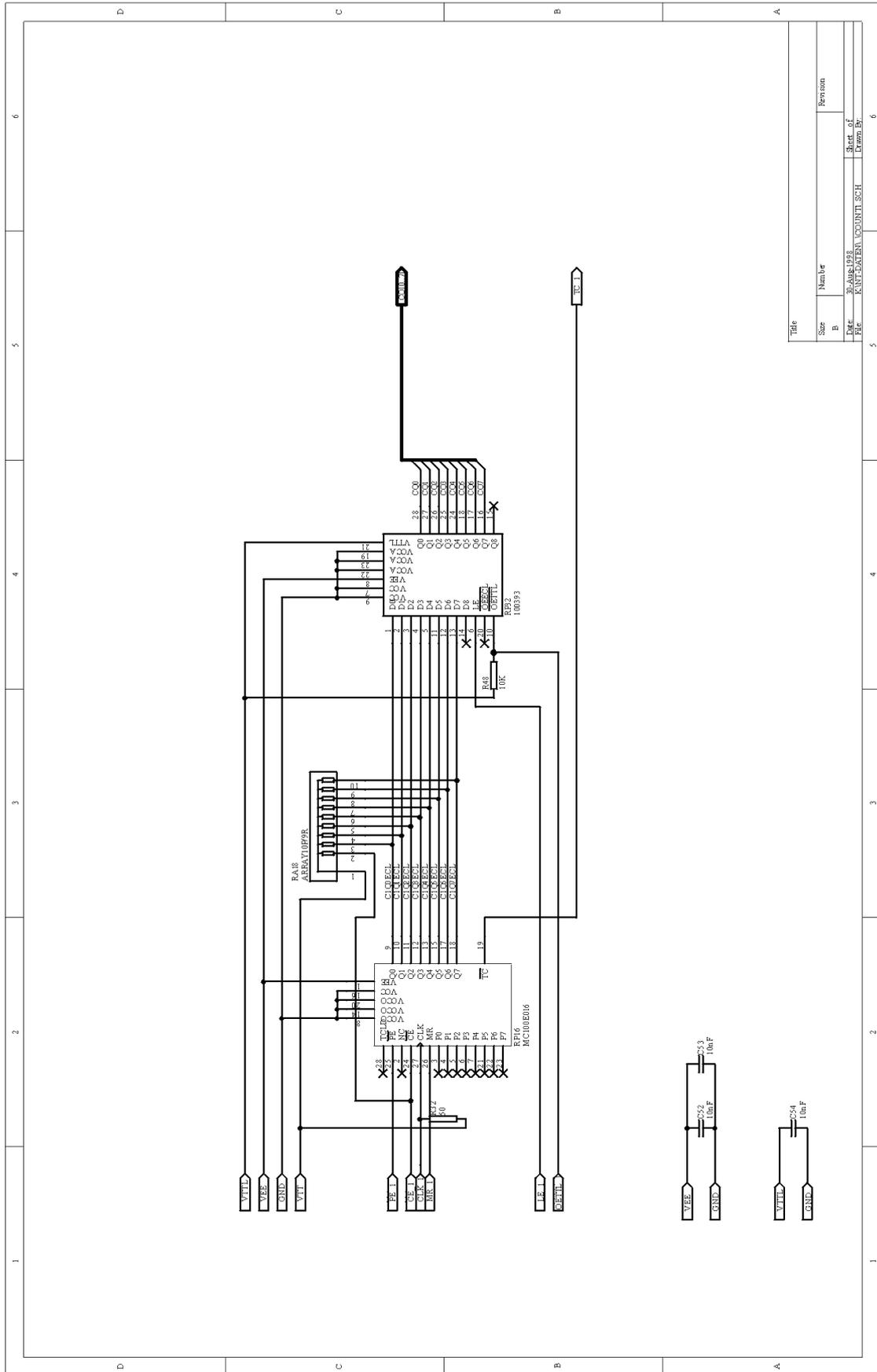



Abbildung 31 : Eingangsstecker für die 16 differentiellen ECL-Kanäle; das invertierte Signal jedes Kanals wird direkt terminiert



Title		Revision	
Sur	Number		
B			
Rev	Rev. No.		
Proj	Proj. No.		
Doc	Doc. No.		

Abbildung 32 : Zählerkanal 1 mit ECL-Zähler, Übersetzerbaustein mit Latch, Terminierungs-Widerständen und Entkopplungskondensatoren

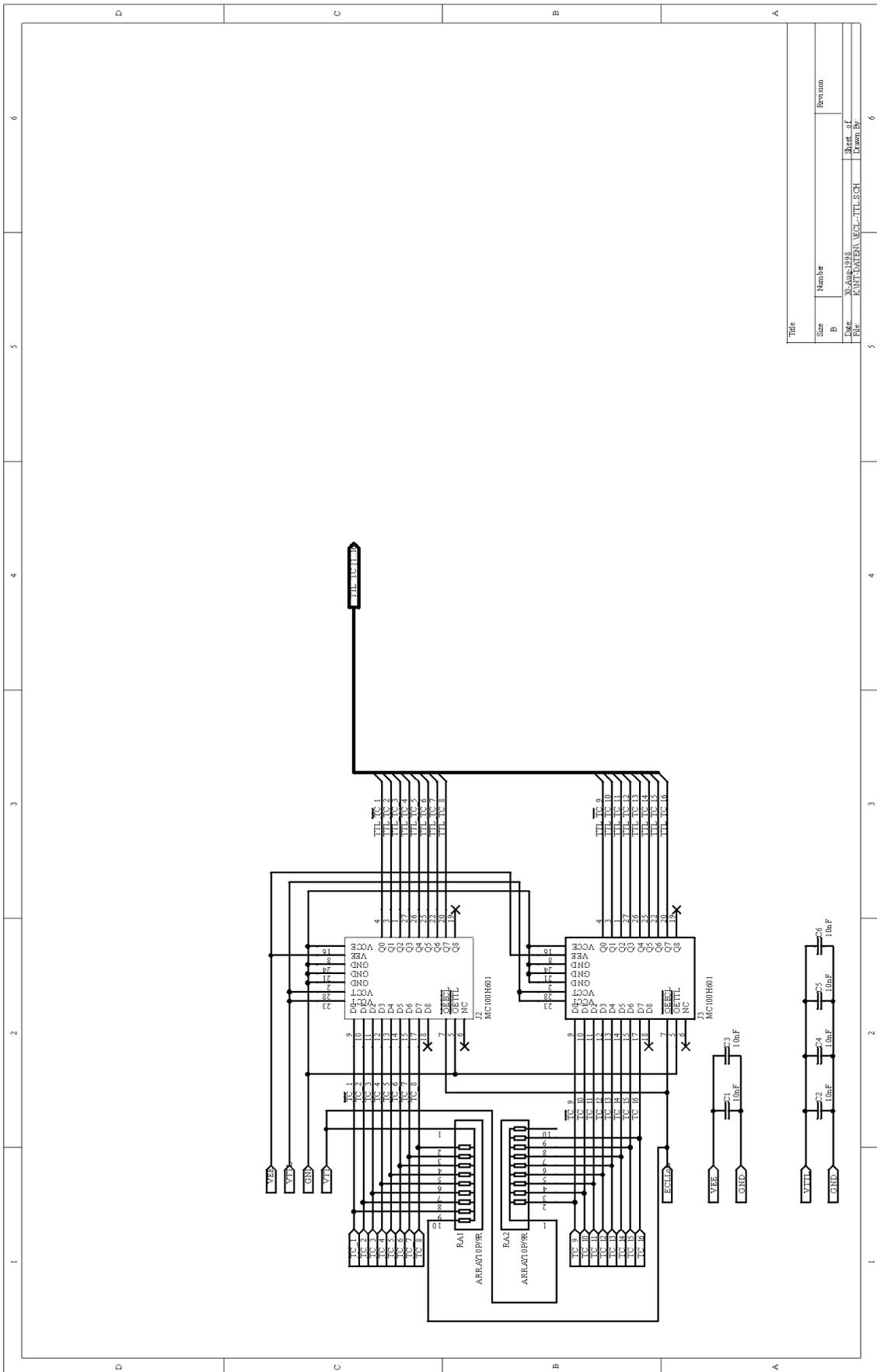


Abbildung 34 : ECL-TTL-Übersetzerbausteine für die Überlaufsignale der ECL-Zählerkomponenten mit Terminierungs-Widerständen und Entkopplungskondensatoren

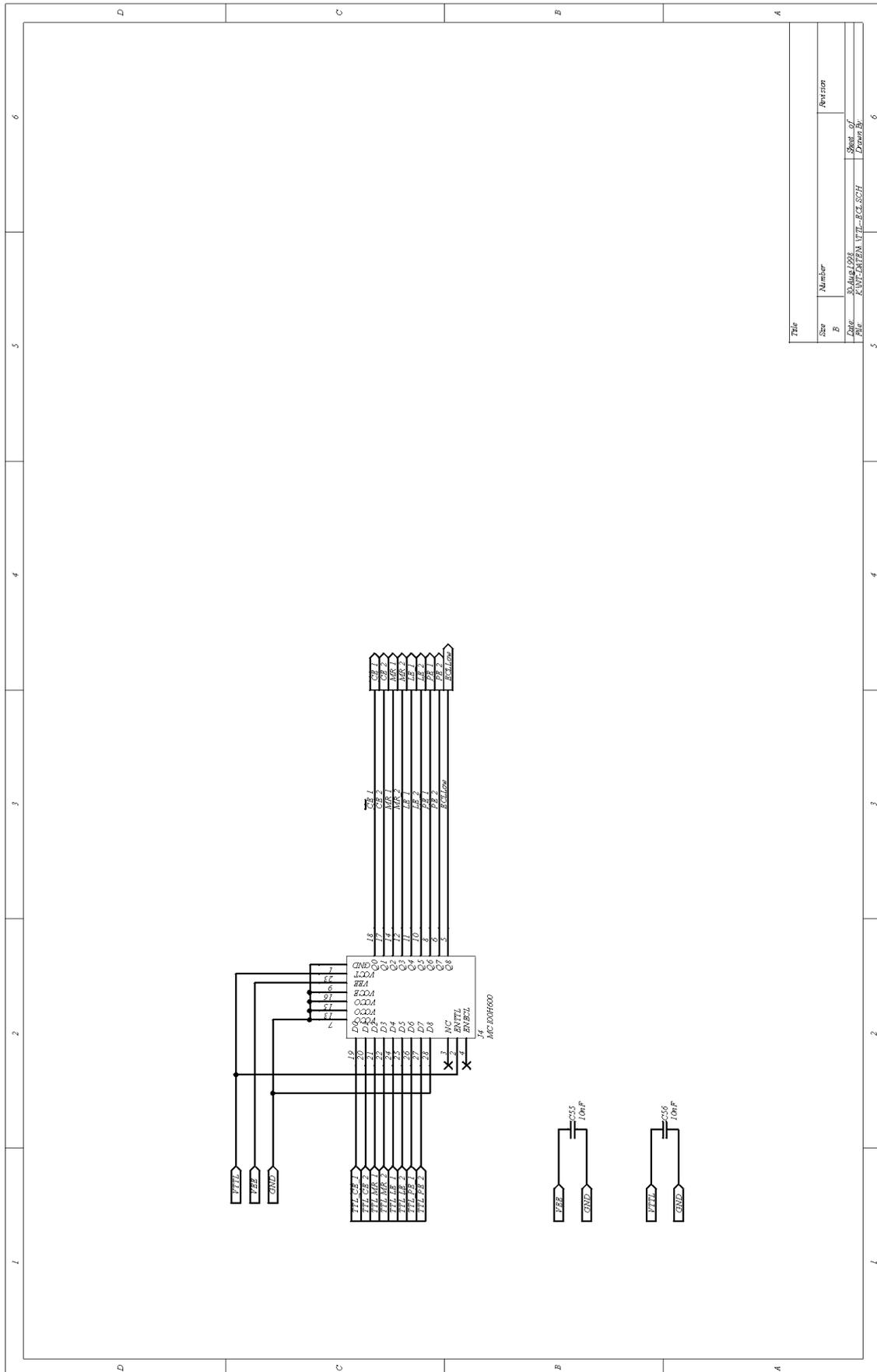


Abbildung 35 : TTL-ECL-Übersetzerbaustein für die verschiedenen ECL-Kontrollsignale mit Entkopplungskondensatoren

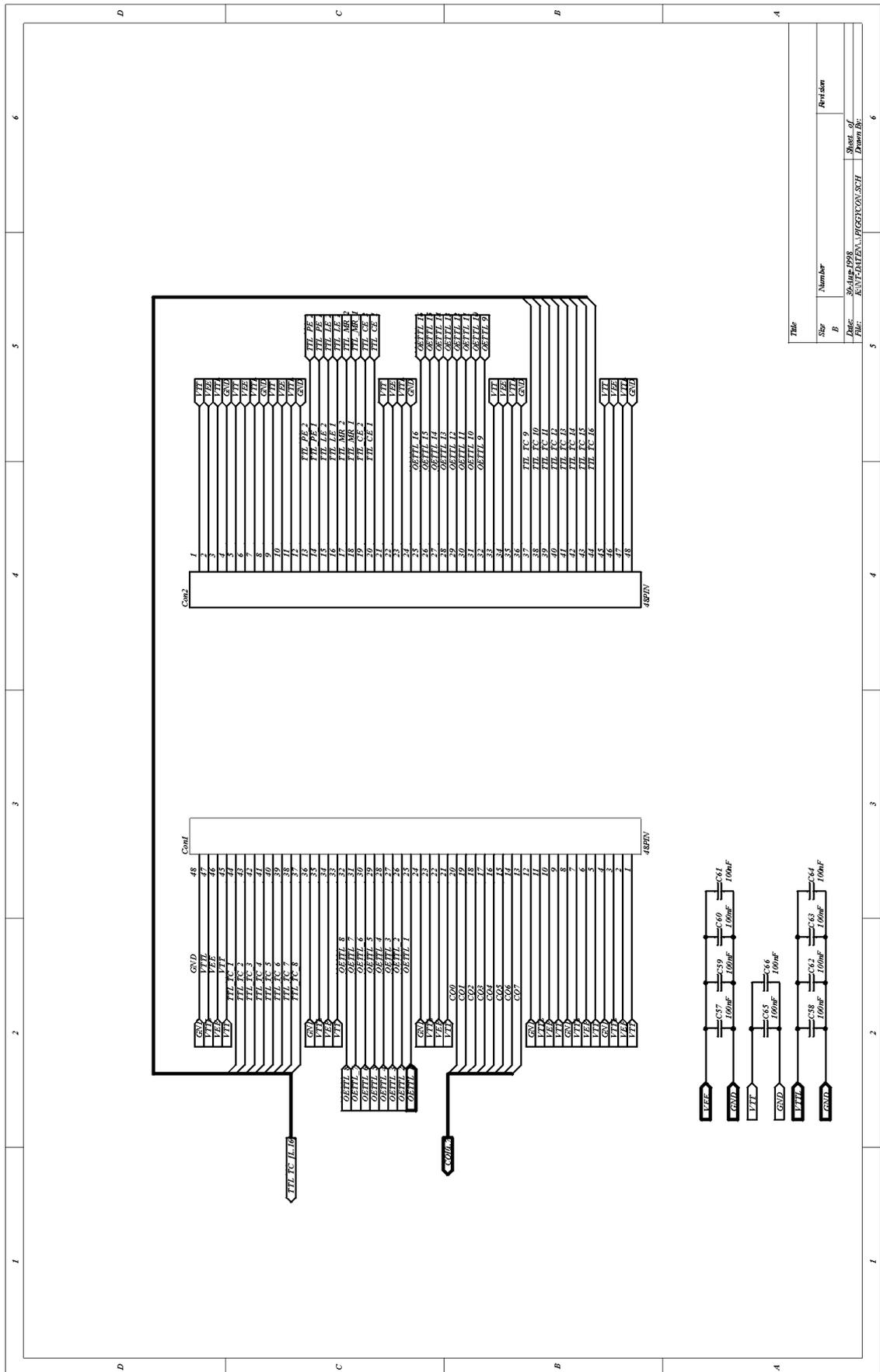


Abbildung 36 : Verbindungsstecker zur Hauptplatine mit Entkopplungskondensatoren für die verschiedenen Versorgungsspannungen

9.3.2 Schaltpläne der Hauptplatine

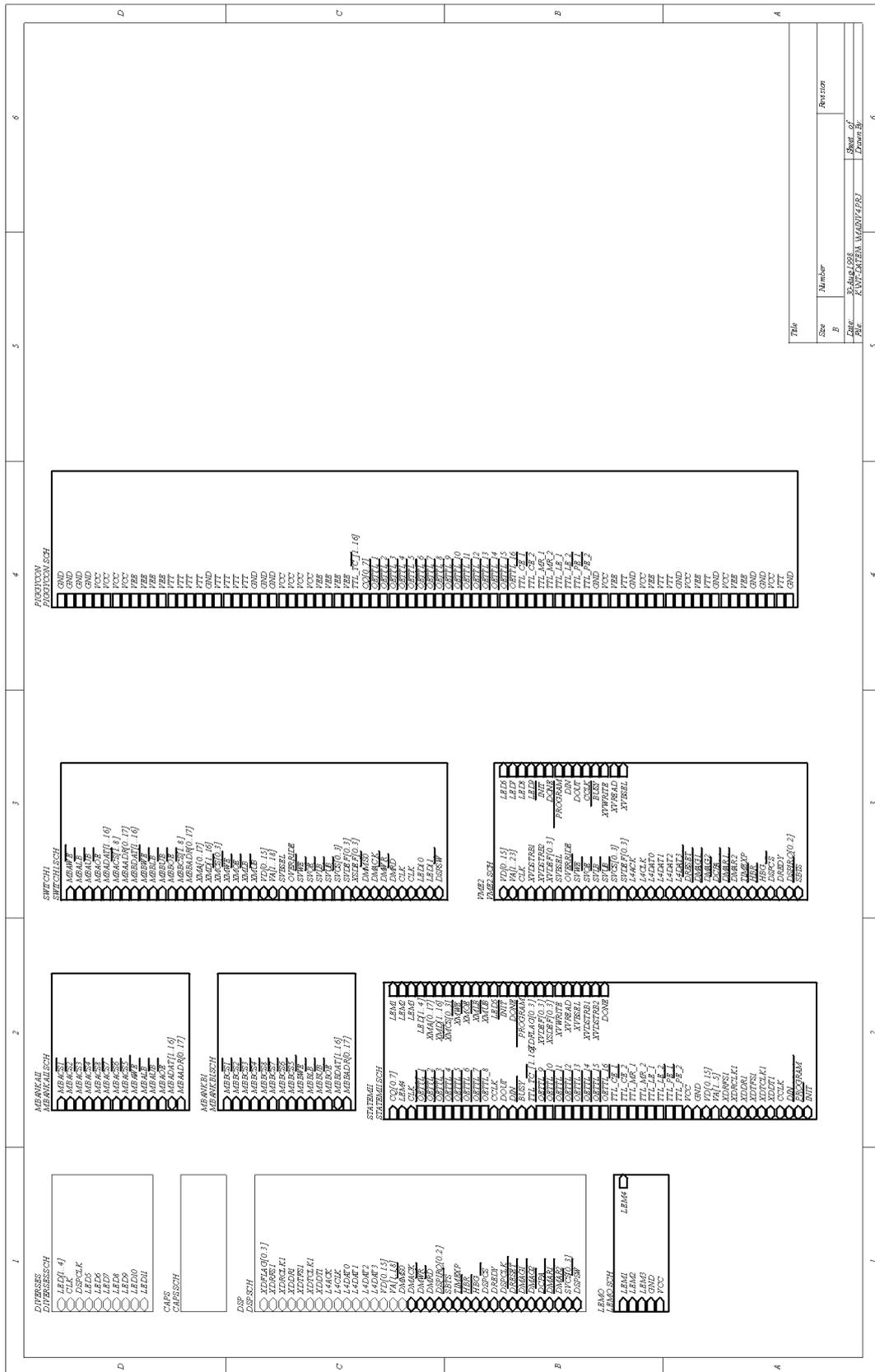


Abbildung 37 : Übersichtsschaltplan der folgenden Komponenten der Hauptplatine

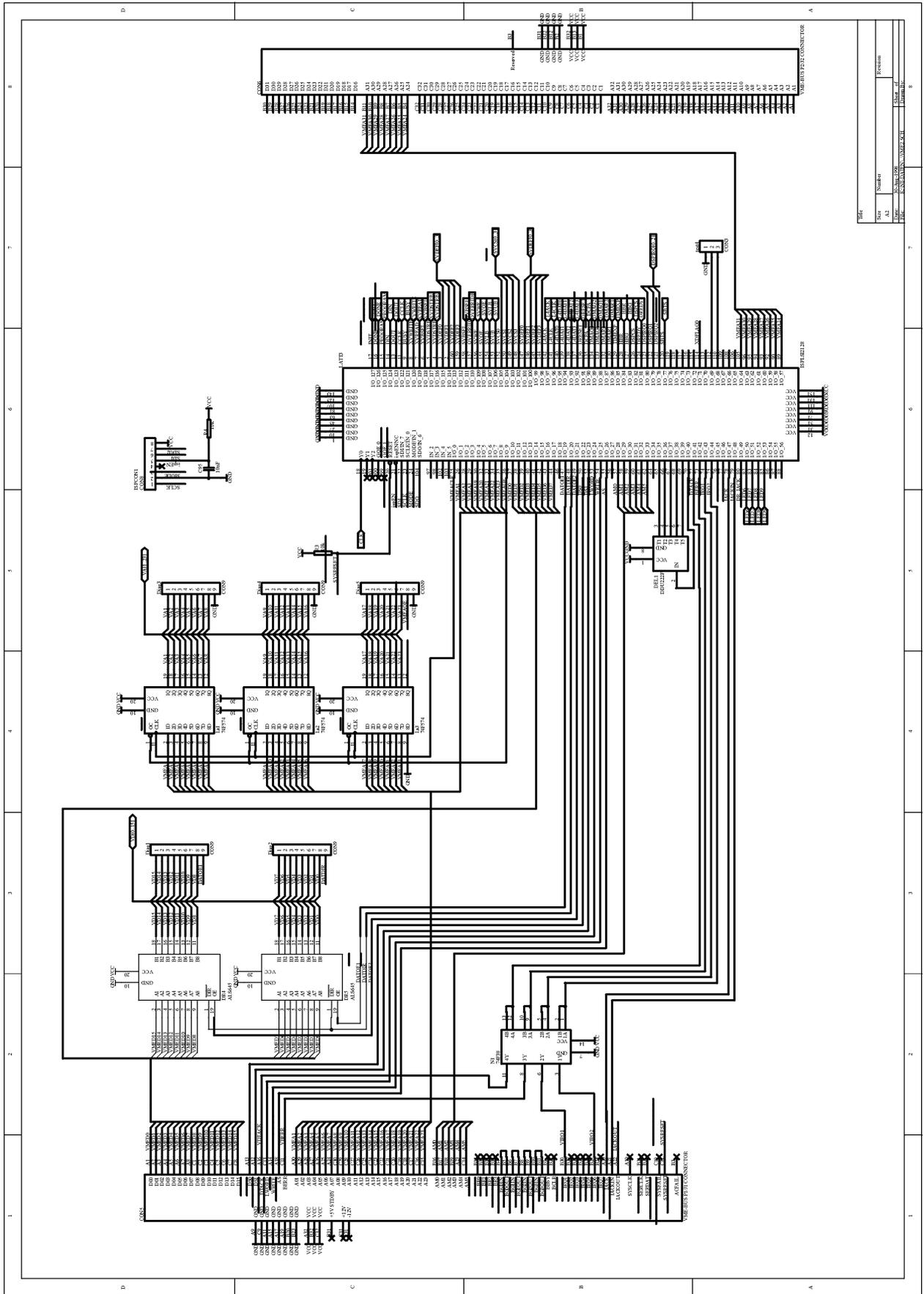


Abbildung 38 : VME-Interface mit VME-Steckverbindern, Daten- und Adreßtreibern, LATTICE CPLD, Delay, „Open-Collector“-Baustein 74F38 und Stiftleisten zum Debuggen und Booten der CPLDs

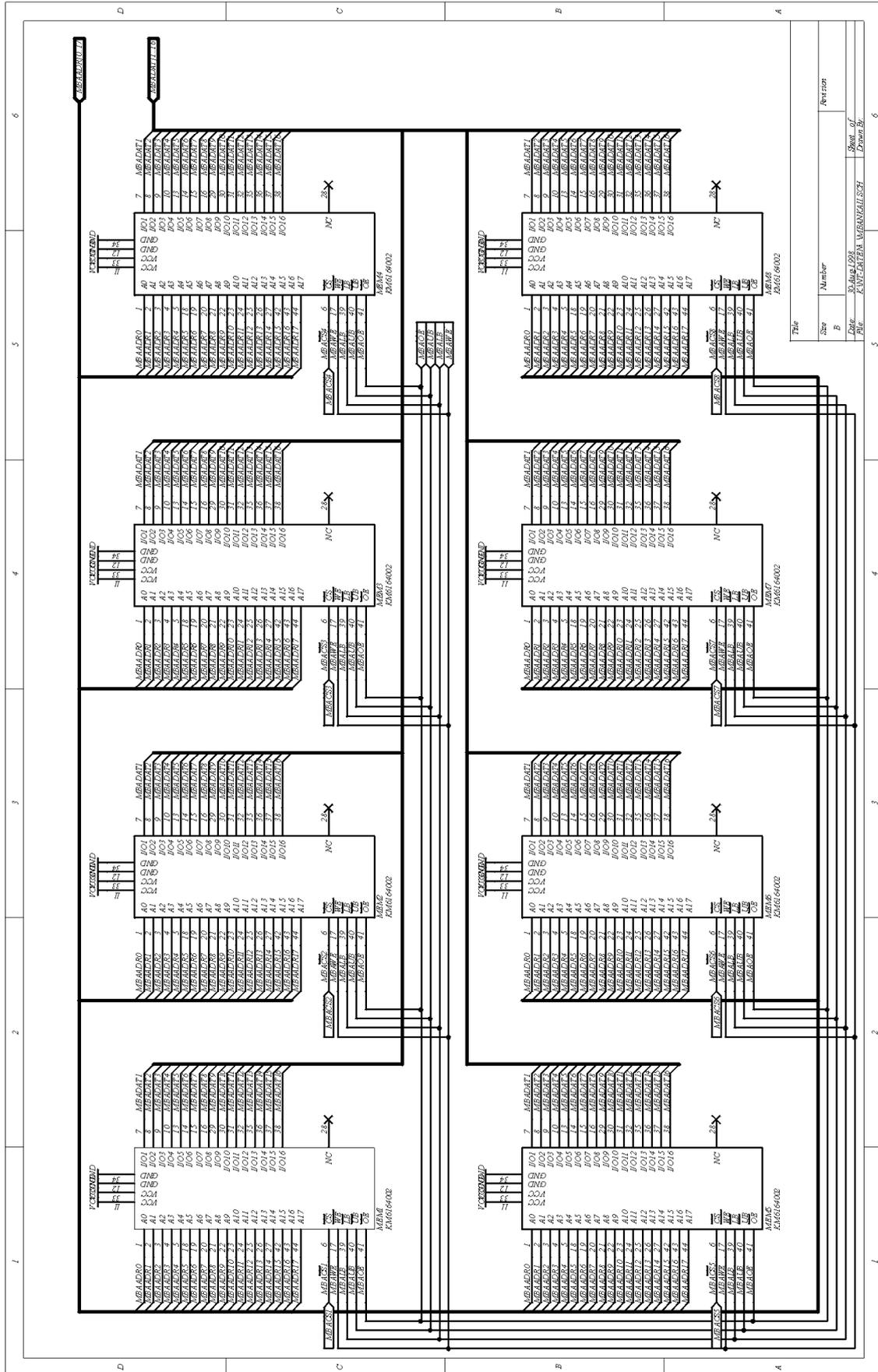


Abbildung 42 : SRAM-Speicher-Bank 1

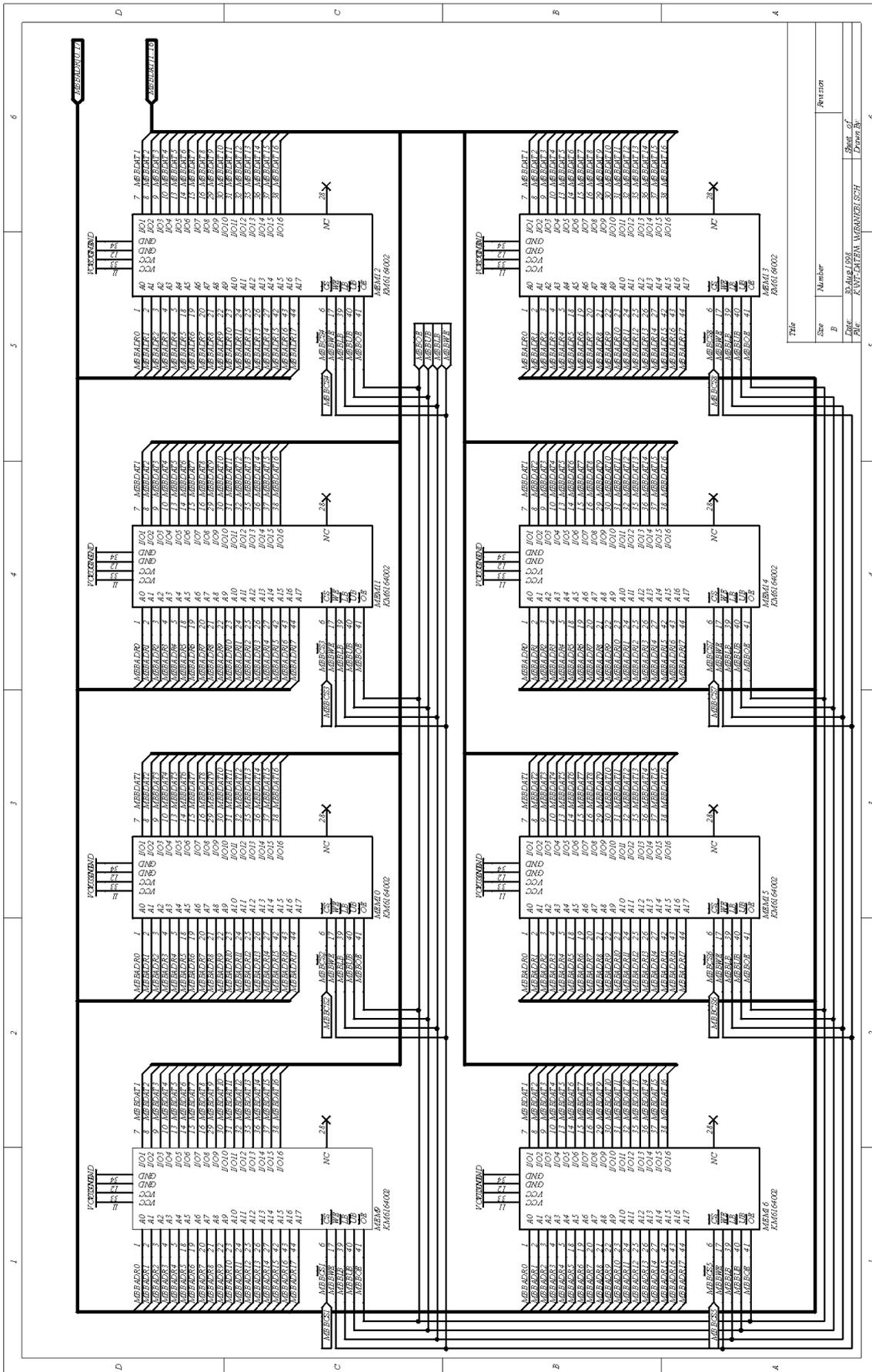
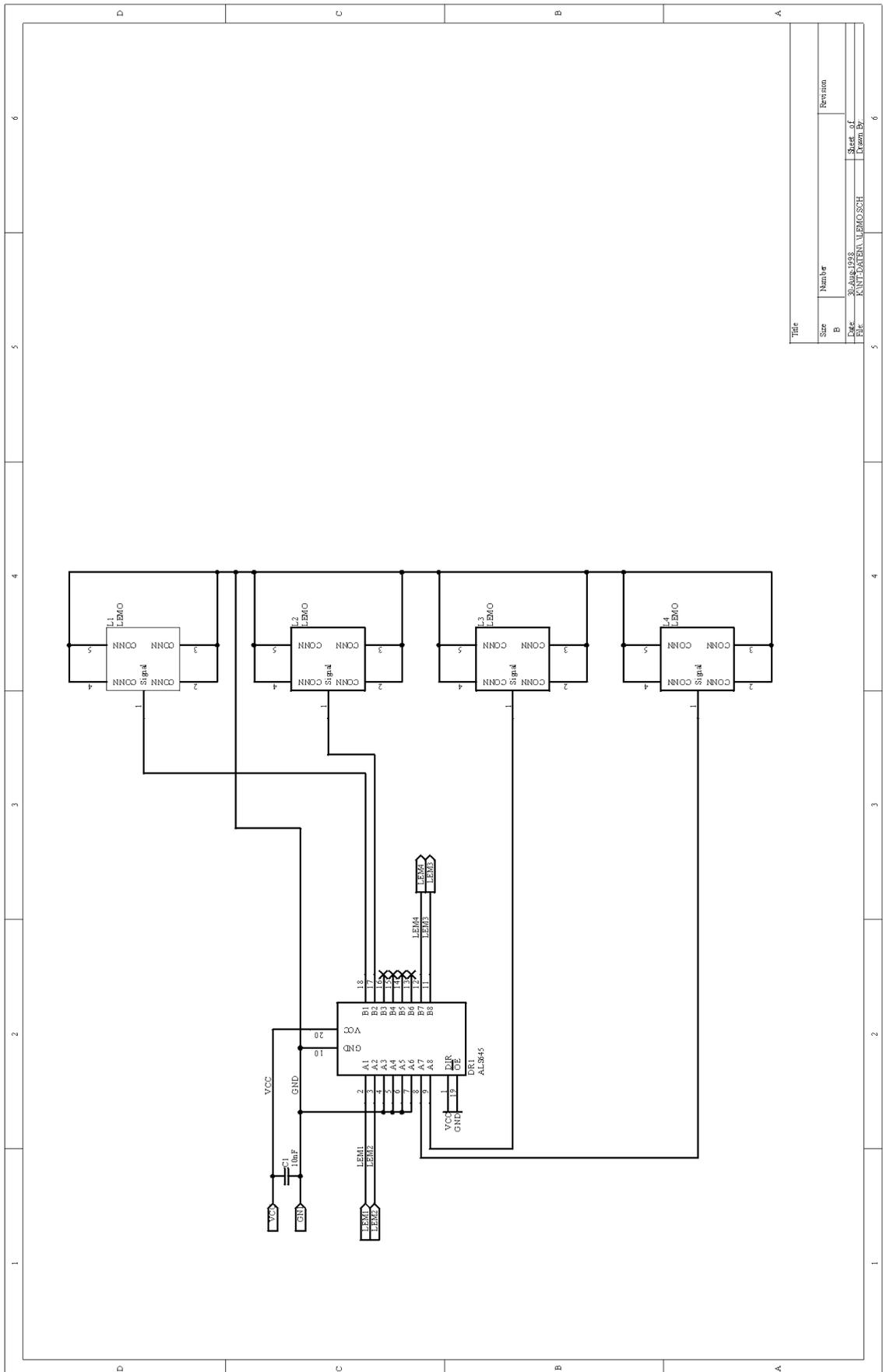
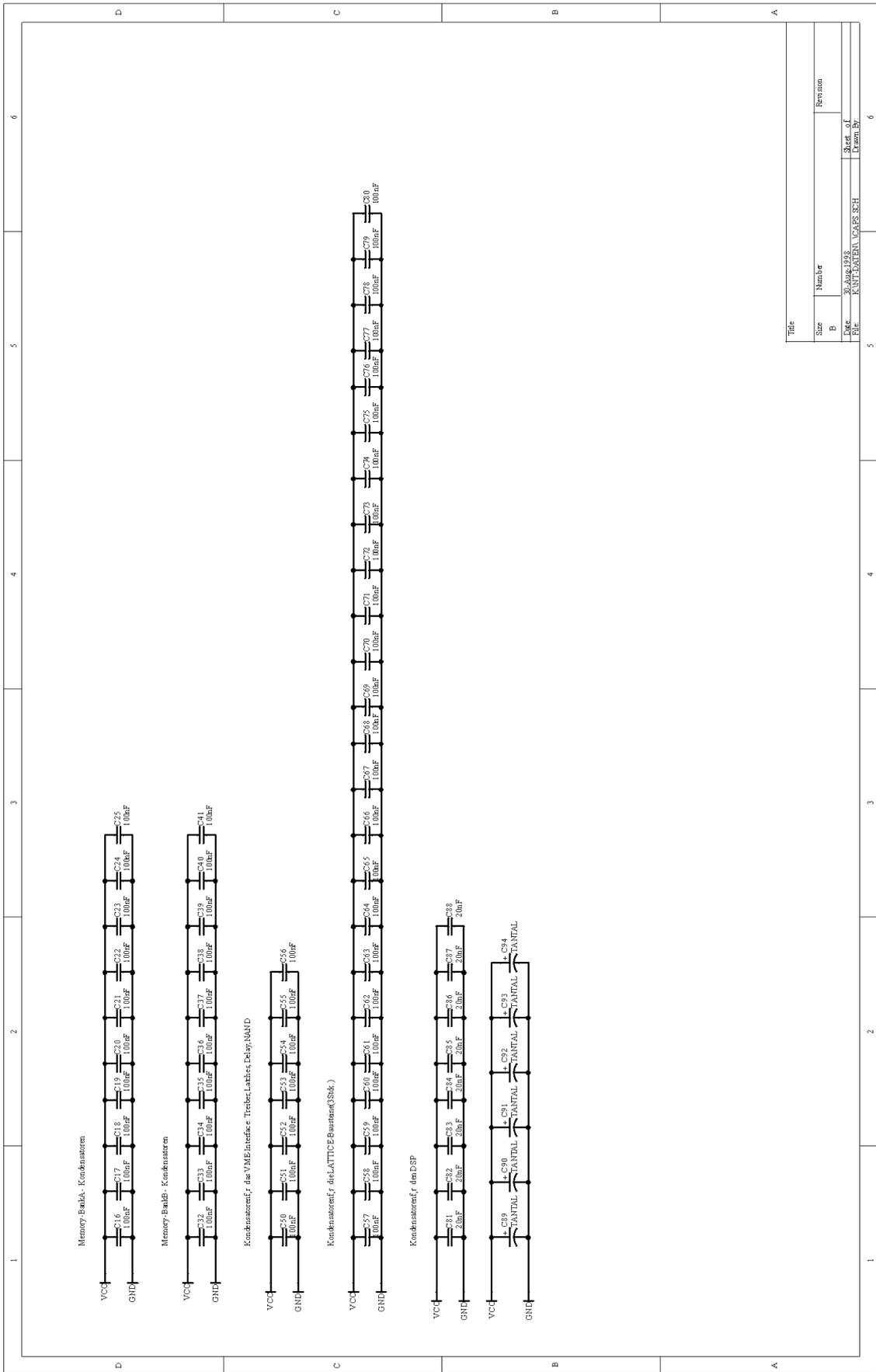


Abbildung 43 : SRAM-Speicher-Bank 2



Title		Revision	
Size	B	Number	
Date	20. Aug. 2018	Start of	
File	KIT/DRIVER/AL5845SCH	Drawn By	

Abbildung 44 : LEMO-Buchsen mit Treiberbaustein



Title	
Size	Number
B	Revision
Date	30. Aug. 1988
File	KONTAKTEHA_VCAPS.SCH
Sheet of	6
Drawn By	

Abbildung 46 : Entkopplungskondensatoren der verschiedenen Funktionsgruppen

9.4.2 Hauptplatine

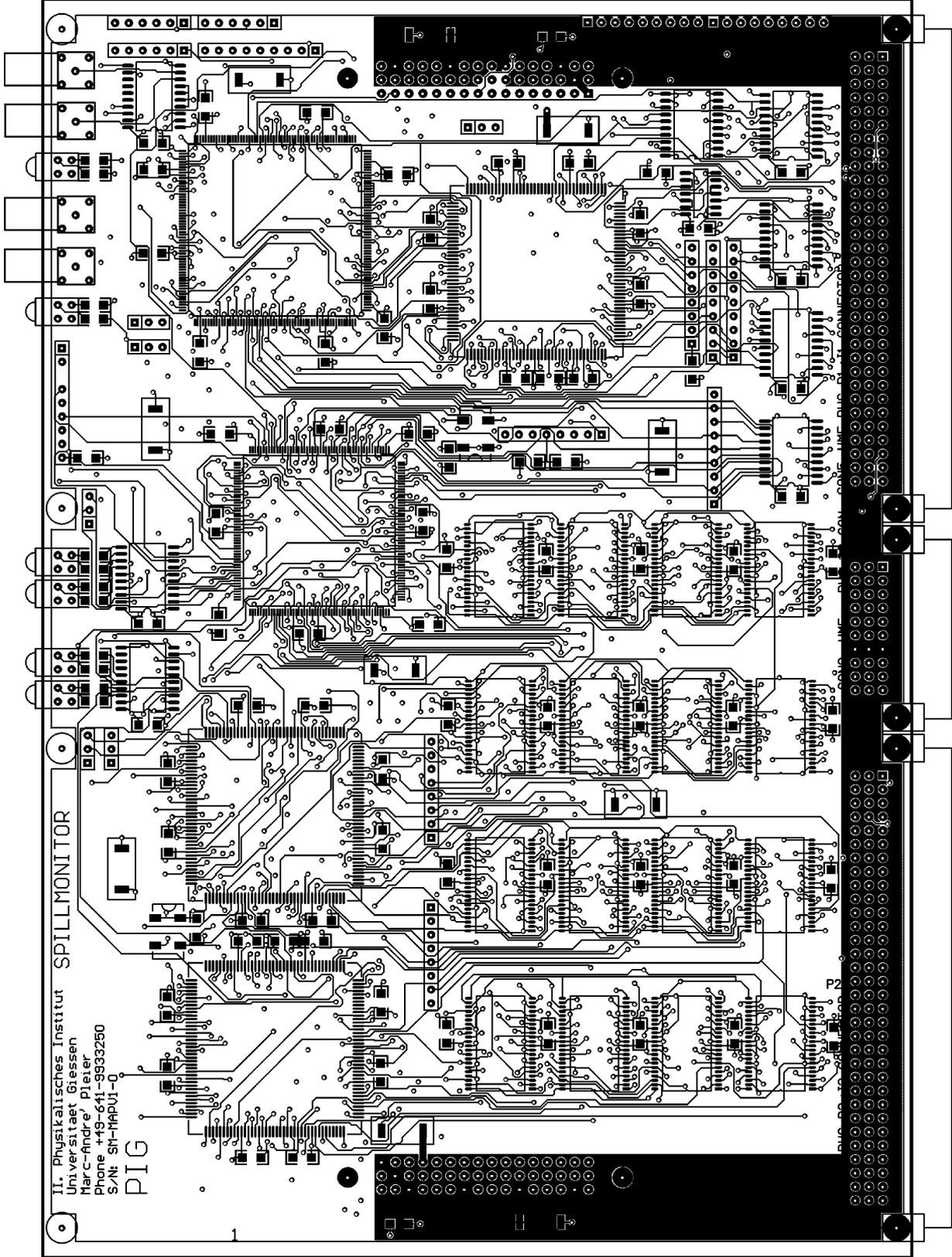


Abbildung 49 : Oberseite der Hauptplatine

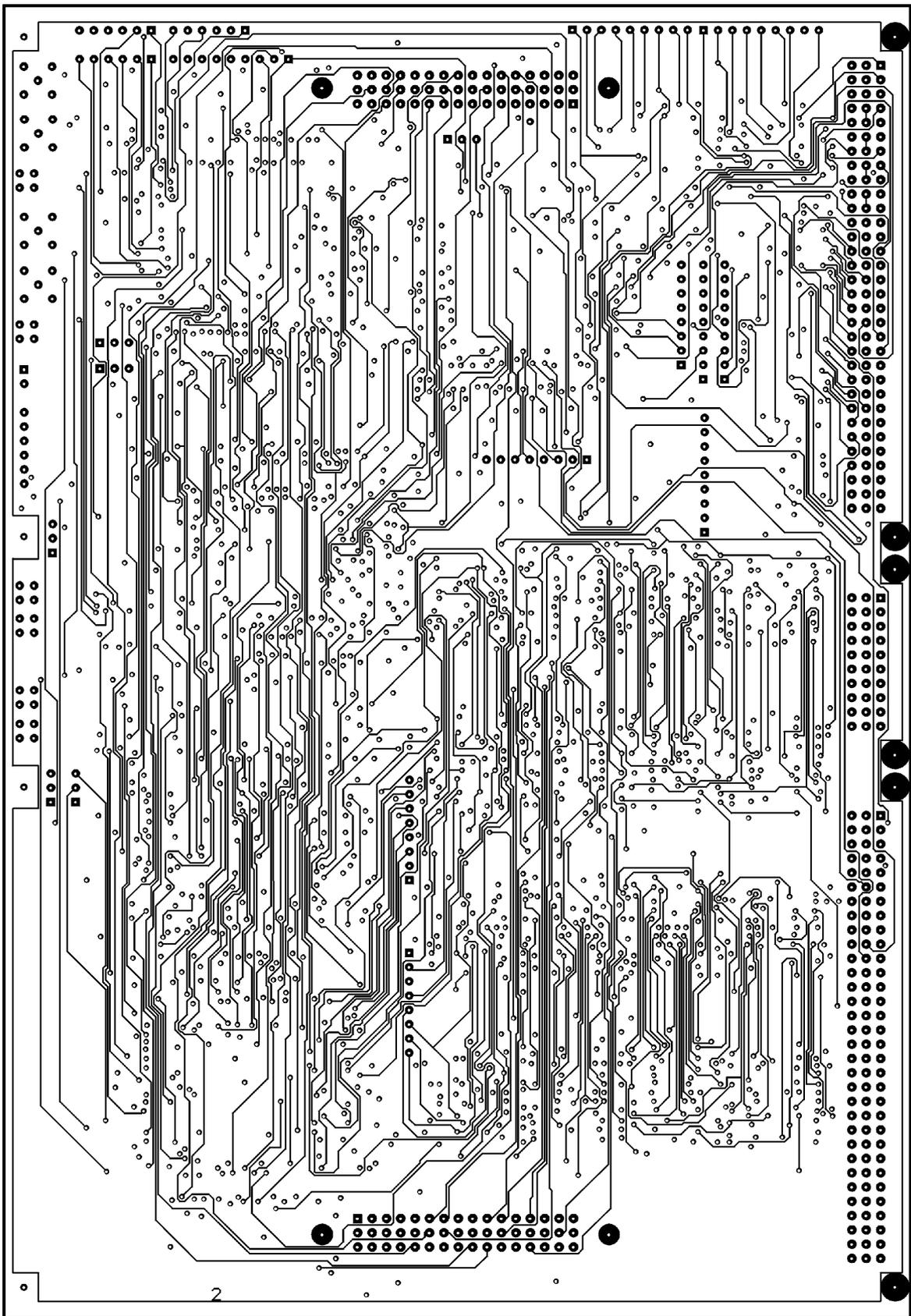


Abbildung 50 : Erste Zwischenlage der Hauptplatine

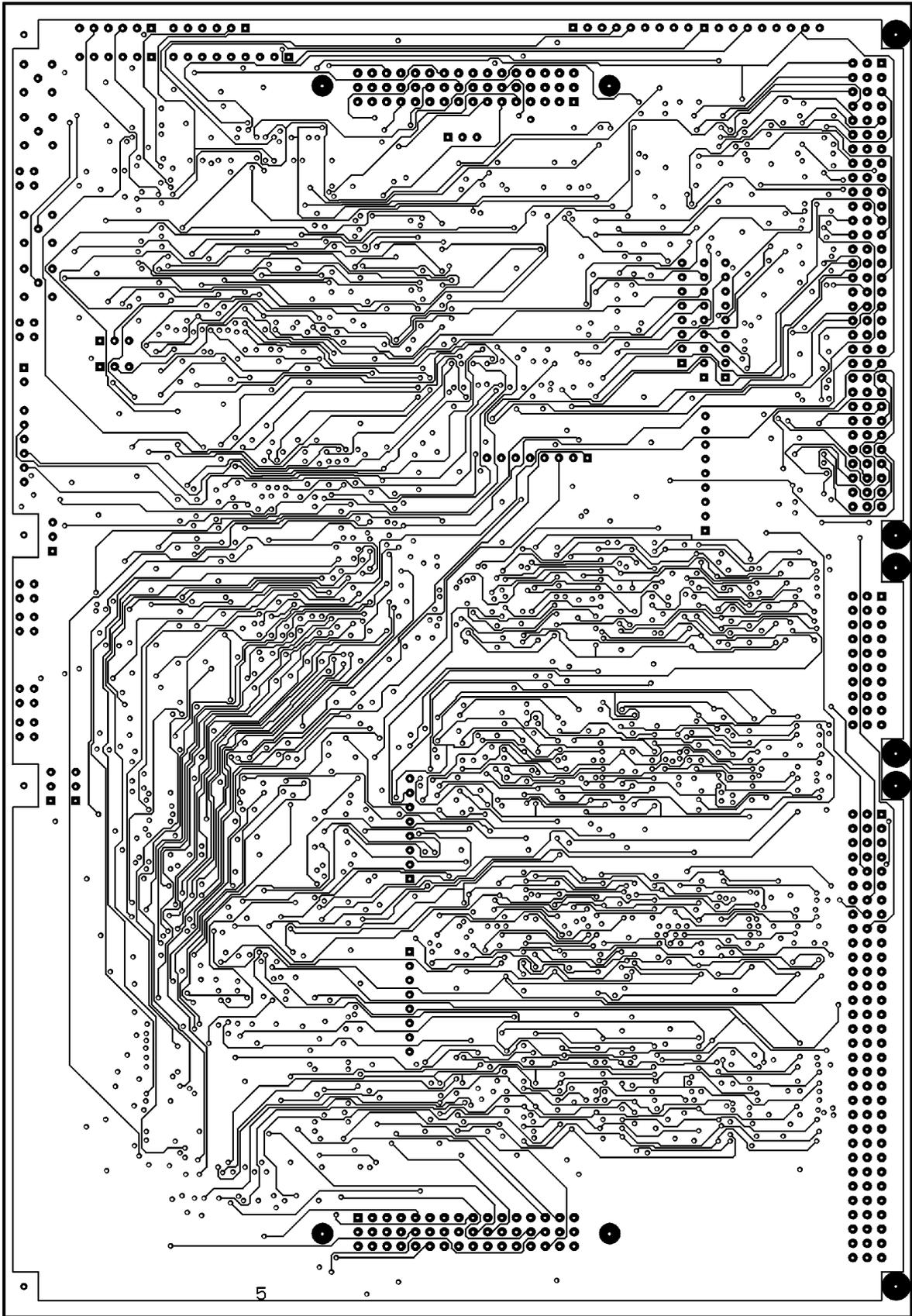


Abbildung 51 : Zweite Zwischenlage der Hauptplatine

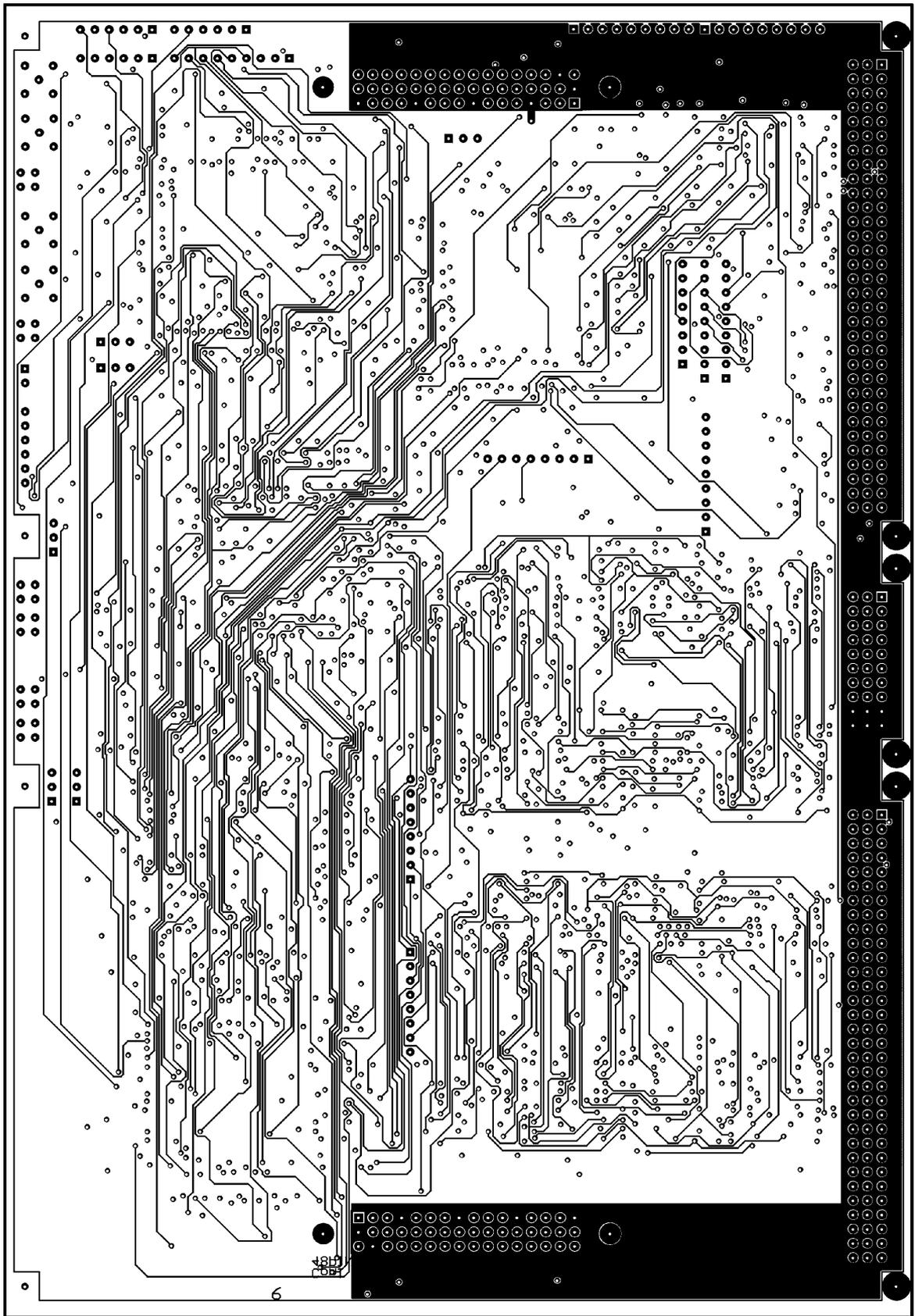


Abbildung 52 : Unterseite der Hauptplatine

Danksagung

Abschließend möchte ich die Gelegenheit nutzen, all denen zu danken, die mit ihrer Unterstützung, Hilfsbereitschaft und ihrem Interesse zum Gelingen dieser Arbeit beigetragen haben.

Mein Dank geht an dieser Stelle zuerst an Herrn Prof. Dr. Wolfgang Kühn, der mir die Durchführung dieser Arbeit in seinem Institut ermöglicht hat, für das Vertrauen in mich beim Ausarbeiten dieser anspruchsvollen und interessanten Arbeit und die Unterstützung zur Bewältigung derselben.

Peter Moritz, Eleni Berdermann, Dr. Wolfgang König und Dr. Burkhard Kolb von der GSI möchte ich für die kompetente fachliche Unterstützung und Hilfestellung danken.

Besonders möchte ich mich auch bei allen Mitgliedern der HADES-Gruppe für das gute Arbeitsklima, die ständige Hilfsbereitschaft sowie die seelische und moralische Unterstützung bedanken:

Markus („Mr. High Speed“), Jörg, Erik und Hans für das Korrekturlesen, die ständige Unterstützung und Aufmunterung und **den** Sessel, René (dem Herrscher über das SMD-Labor), Carsten („Heute Abend Training?“), Michael, Jim für die nahezu unmenschliche Geduld und natürlich meinen beiden Zimmergenossen Ingo und Berengar.

Auch an alle TAPSianer sei hier mit einem „Dankeschön!“ gedacht (Winnie - Badminton???). Natürlich möchte ich auch dem „Rückrat“ unseres Instituts - Anita, Jürgen, Werner und Marianne - und den Mitarbeitern der Elektronik- und Mechanik-Werkstatt danken.

All meinen Freunden und Bekannten danke ich für die Geduld, die sie in letzter Zeit mit einem ziemlich gestressten Marc-André haben mußten.

Ein ganz spezielles Danke geht schließlich noch an meine Eltern, die mich *immer* unterstützt haben (und hier ist keinesfalls nur der finanzielle Aspekt gemeint!) und an meinen ♥-allerliebsten Schatz Verena, der mich auch in der stressigen Abschlußphase noch ertragen und aufgebaut hat.