

Entwicklung
eines 250-MHz-Zählers
mit totzeitfreier Auslese
für das COMPASS-Experiment

Marc Niebuhr



FAKULTÄT FÜR PHYSIK
ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Entwicklung eines 250-MHz-Zählers mit totzeitfreier Auslese für das COMPASS-Experiment

Diplomarbeit
vorgelegt
von
Marc Niebuhr

FAKULTÄT FÜR PHYSIK
ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG IM BREISGAU

November 2000

Inhaltsverzeichnis

1	Einführung	5
2	Das COMPASS-Experiment und seine Ziele	7
2.1	Die Physik	7
2.2	Der Aufbau des Experiments	10
2.2.1	Der Strahl	10
2.2.2	Das polarisierte Target	12
2.2.3	Der COMPASS-Detektor	13
3	Das Auslese-System	17
3.1	Das Trigger-Kontroll-System	17
3.2	Der Weg der Daten zur Datennahme	19
4	Physikalische und technische Vorgaben	23
4.1	Vorgaben des Detektors	23
4.2	Ein improvisierter Test	27
5	Grundlagen eines Zählers	29
5.1	Elektronische Grundlagen	29
5.1.1	Das Flip-Flop	29
5.1.2	Der Standardzähler	30
5.1.3	Konzept zweier Zähler in einer Kaskade	32
5.1.4	Gray-Code	33
5.1.5	Das Schieberegister	34
5.1.6	Der Johnson-Counter	35
5.1.7	Das FIFO	37
5.2	Programmierbare Logik	38
5.2.1	Der FPGA — Xilinx Virtex-E XCV600E-HQ240	38
6	Das Zähler-Design	41
6.1	Die Zähler-Platine	41
6.2	Das Design für den FPGA	47
6.2.1	Der Johnson-Counter und dessen Auslese	47

6.2.2	Der Dekoder	48
6.2.3	Das programmierbare Schieberegister	49
6.2.4	Der Subtrahierer und der 31-bit-Addierer	51
6.2.5	Die Auslese	52
6.2.6	Zusammenfassung des Zählmechanismus	53
6.2.7	Die Auslese der Zähler-FIFOs	54
6.2.8	Die serielle Programmierung	55
6.2.9	Der Reset-Mechanismus	56
6.3	Die Datenstruktur	57
6.4	Weiter Anwendungsmöglichkeiten für die Zähler-CMC	57
7	Die Testphase	61
7.1	Der Verbindungstest	61
7.2	Das Datenformat	62
7.3	Der Zähltest	62
7.4	Der Bit-Test	65
8	Der Einsatz im COMPASS-Experiment	67
8.1	Das räumliche Strahlprofil	68
8.2	Das räumliche Strahlprofil im zeitlichen Verlauf	68
8.3	Das zeitliche Strahlprofil	73
9	Fazit	75
A	Programmierbare Logik	77
A.1	Entwicklung der programmierbaren Logik	77
A.2	Der FPGA — Xilinx Virtex-E XCV600E-HQ240	78
A.3	Der Designprozess	83
B	Der Schaltplan	89
C	Das Platinen-Design	95
D	Bilder	99
E	Glossar	103
	LITERATURVERZEICHNIS	107

Kapitel 1

Einführung

Im Jahr 2000 wurden beim COMPASS-Experiment am CERN die ersten Daten genommen. Dieses Experiment verfolgt eine Vielzahl physikalischer Ziele. Es beabsichtigt, sowohl die Struktur der Nukleonen zu erforschen, als auch die Spektroskopie von charm-haltigen Hadronen zu betreiben. Ein wichtiges Ziel ist dabei die Messung des Gluonenanteils am Nukleonspin. Das Experiment entstand durch den Zusammenschluss der HMC- und der Cheops-Kollaboration [1, 2] und ist somit indirekt ein Nachfolger des SMC- und des WA89-Experiments [3, 4] am CERN. Bei COMPASS handelt sich um ein Experiment mit stationärem Target¹ und mit wahlweise leptonischem oder hadronischem Strahl.

Mit dem COMPASS-Experiment beginnt ein neuer Abschnitt in der Geschichte der Fixed-Target-Experimente. Ähnlich hohe Ereignisraten wurden bei vergleichbaren Experimenten noch nie erreicht. Diese Raten stellen hohe Anforderungen an alle Detektorkomponenten, insbesondere an die Auslese-Elektronik, die pro Sekunde mehrere Gigabyte an Daten bewältigen muss. Aus diesem Grund wird bei COMPASS ein neues Konzept verfolgt. Der gesamte Detektor wird totzeitfrei ausgelesen. Die Daten werden direkt am Detektor digitalisiert, zwischengespeichert und schon während der Auslese, also bevor sie auf statischen Medien gespeichert werden, sortiert und reduziert. Das Herzstück dieses Verfahrens ist das an der Universität Freiburg entwickelte CATCH-Modul².

Diese Arbeit beschäftigt sich mit der Entwicklung eines Teils der Auslese-Elektronik des Detektors: einem Zähler. Auch für einen Zähler ergeben sich bei diesem Experiment sehr hohe Anforderungen. Es ist beabsichtigt, Profile des hochintensiven Strahls damit aufzunehmen, wobei 200-MHz-Signale von Detektoren gezählt werden müssen. Die Auslese soll totzeitfrei mit jedem

¹engl.: *fixed target*

²COMPASS Accumulate, Transfer & Control Hardware

Trigger erfolgen können, also im Mittel alle $10\ \mu\text{s}$, wobei zwischen einzelnen Triggern im Extremfall nur $50\ \text{ns}$ liegen. Ein Zähler, der diese Anforderungen erfüllt, war auf dem Markt nicht erhältlich. Deshalb wurde im Frühjahr 2000 mit der Entwicklung, basierend auf einem FPGA, begonnen.

Kapitel 2 dieser Arbeit beinhaltet eine kurze Beschreibung der Ziele, Methoden und Detektoren des COMPASS-Experiments. Dabei werden die Teilchenstrahlen und ihre Erzeugung vorgestellt, da der Zähler zur Vermessung des Strahls dient. Daraufhin wird in Kapitel 3 die teilweise an der Universität Freiburg entwickelte Auslese-Elektronik, in die der Zähler integriert ist, detaillierter beschrieben. Nachdem die Anforderungen in allen Einzelheiten aufgezählt worden sind (Kapitel 4), werden in Kapitel 5 ausführlich die Grundlagen von Elektronik-Design und programmierbarer Logik aufgearbeitet, und schließlich folgt in Kapitel 6 eine fundierte Beschreibung des Zähler-Designs.

Doch nicht nur Entwicklung und Produktion waren Gegenstand dieser Arbeit. Ganz am Beginn der Beschäftigung stand eine Abschätzung, ob die ehrgeizigen Ziele überhaupt zu erreichen sind. Das Ergebnis dieser Überlegungen war ein ungewöhnliches Design, das in allen Tests überzeugen konnte.

Schon im Sommer war eine einsatzfähige Version fertig. Diese Version wurde ausführlich getestet (Kapitel 7). Als das einwandfreie Funktionieren gewährleistet war, wurden mit dem Zähler im August und September 2000 schließlich Daten genommen. Als Detektor diente dabei eine Anordnung von szintillierenden Fasern. Neben den für einen Zähler obligatorischen Messungen der Projektionen des Strahlprofils auf die x- und y-Achse konnte ein zeitliches Profil des Strahls aufgezeichnet werden. Hierzu sind gewöhnliche Zähler auf Grund ihrer Totzeit während der Auslese nicht in der Lage. Im abschließenden Kapitel 8 werden die Ergebnisse dieser Messungen vorgestellt.

Kapitel 2

Das COMPASS-Experiment und seine Ziele

Das COMPASS¹-Experiment am europäischen Forschungszentrum CERN ist ein Experiment mit stationärem Target, das der Untersuchung der Struktur der Hadronen dient [5]. Es besteht aus einem myonischen Experimentierprogramm, bei dem polarisierte Myonen auf ein polarisiertes Target gelenkt werden, und einem anschließenden hadronischen Experimentierprogramm, bei dem Pionen, Kaonen oder Protonen auf ein unpolarisiertes Target treffen.

2.1 Die Physik

Das Quark-Parton-Modell der Nukleonen gibt es seit 30 Jahren [6]. Doch bis heute sind einige Fragen zur Struktur der Nukleonen ungeklärt geblieben. So ist z.B. nicht bekannt, wie sich der Spin des Nukleons auf seine Bestandteile verteilt. Die einfachste Annahme wäre, dass allein die Quarks und Anti-Quarks den Spin tragen. Doch diese Annahme konnte bereits vor zehn Jahren vom EMC-Experiment widerlegt werden. Die Messung des Beitrags der Quarks zum Nukleonspin zeigte, dass der Spin des Nukleons zu einem großen Teil von anderen Effekten bestimmt wird:

$$\Delta\Sigma = \Delta u + \Delta d + \Delta s \ll 1$$

(gegenüber der naiven Annahme $\Delta\Sigma = 1$), wobei

$$\Delta q_f = \int_0^1 \{ (q_f(x)^{\downarrow\uparrow} + \bar{q}_f(x)^{\downarrow\uparrow}) - (q_f(x)^{\uparrow\uparrow} + \bar{q}_f(x)^{\uparrow\uparrow}) \} \cdot dx = \int_0^1 \Delta q_f(x) \cdot dx$$

ist, und $q_f(x)^{\uparrow\uparrow}$ bzw. $q_f(x)^{\downarrow\uparrow}$ die Wahrscheinlichkeitsdichteverteilungen dafür sind, dass an einem (Anti-)Quark der Sorte $f = u, d, s$ gestreut wird. Hierbei stehen $\uparrow\uparrow$ und $\downarrow\uparrow$ für parallele bzw. anti-parallele Spin-Einstellungen von

¹Common Muon and Proton Apparatus for Structure and Spectroscopy

Quark und Nukleon (Abbildung 2.1). $x = x_{\text{Bjorken}} = \frac{Q^2}{2M\nu}$ ist die Skalenvariable, die den Anteil des gestreuten Quarks am Viererimpuls des Nukleons in einem schnell bewegten Koordinatensystem² angibt.

Ein möglicher Beitrag zum Spin des Nukleons könnte von den Gluonen kom-

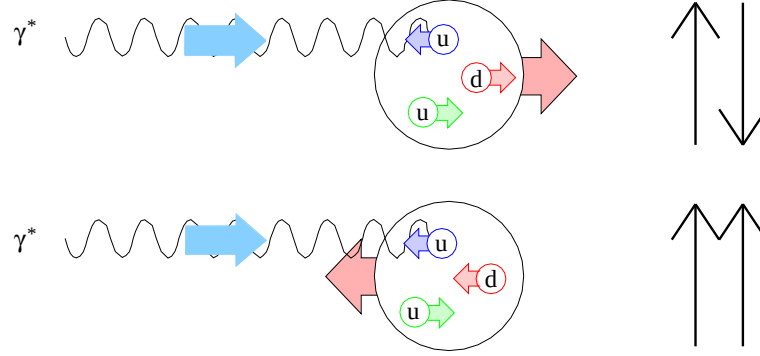


Abbildung 2.1: Streuung an einem stationärem Target mit anti-paralleler und paralleler Spineinstellung von Quark und Nukleon. Wegen der Helizitätserhaltung sind die Spineinstellungen von virtuellem Photon und Nukleon immer anti-parallel.

men.

Die Bestimmung der Gluonenpolarisation $\Delta G/G$ im Nukleon ist eines der Hauptziele des COMPASS-Experiments. Diese Größe wird mit Hilfe des Prozesses der Photon-Gluon-Fusion (Abbildung 2.2) gemessen.

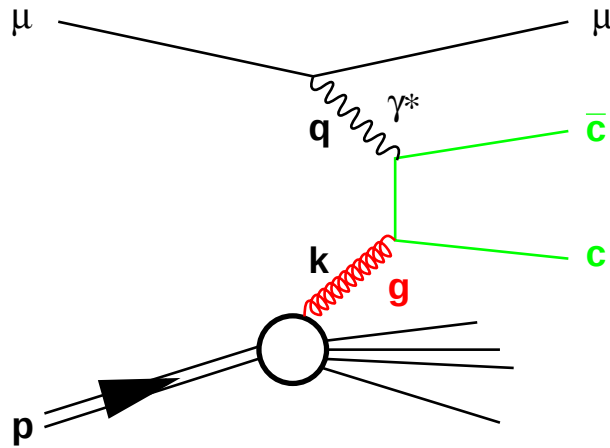


Abbildung 2.2: Die Photon-Gluon-Fusion.

Bei COMPASS werden longitudinal polarisierte Myonen auf ein ebenfalls longitudinal polarisiertes Target geschossen. Durch Subtrahieren der Anzahl

²Infinite Momentum Frame

der durch Streuung mit paralleler bzw. anti-paralleler Spineinstellung *von Strahl und Target* erzeugten $c\bar{c}$ -Paare ($N_{c\bar{c}}^{\uparrow\uparrow}, N_{c\bar{c}}^{\uparrow\downarrow}$) erhält man die gemessene Asymmetrie A^{exp} , die mit der Asymmetrie $A_{\mu p}^{c\bar{c}}$ der Reaktion $\mu + N \rightarrow c\bar{c} + X$ im folgendem Zusammenhang steht:

$$A^{\text{exp}} = \frac{N_{c\bar{c}}^{\uparrow\downarrow} - N_{c\bar{c}}^{\uparrow\uparrow}}{N_{c\bar{c}}^{\uparrow\downarrow} + N_{c\bar{c}}^{\uparrow\uparrow}} = P_S \cdot P_T \cdot f \cdot A_{\mu p}^{c\bar{c}}(y).$$

Dabei ist P_S die Polarisation des Strahls, P_T die des Targets und f der Bruchteil der polarisierbaren Nukleonen im Targetmaterial³. Aus $A_{\mu p}^{c\bar{c}}$ lässt sich die Asymmetrie $A_{\gamma p}^{c\bar{c}}$ berechnen:

$$A_{\mu p}^{c\bar{c}} = D \cdot A_{\gamma p}^{c\bar{c}}.$$

D ist der Depolarisationsfaktor, der die Wahrscheinlichkeit angibt, dass das virtuelle Photon die Polarisation des Myons übernimmt. D hängt vom Verhältnis $y = \frac{\nu}{E}$ der Energien ν und E von virtuellem Photon und Myon ab:

$$D(y) \approx \frac{1 - (1 - y)^2}{1 + (1 - y)^2}.$$

Die Asymmetrie $A_{\gamma p}^{c\bar{c}}$ kann auch durch ein Funktional des bekannten, elementaren Photon-Gluon-Wirkungsquerschnitts $\sigma^{\gamma g \rightarrow c\bar{c}}$ und der Gluon-Polarisation $\Delta G/G$ berechnet werden:

$$A_{\gamma p}^{c\bar{c}} = \frac{\Delta\sigma^{\gamma p \rightarrow c\bar{c}X}}{\sigma^{\gamma p \rightarrow c\bar{c}X}} = \frac{\int_{4m_c^2}^{2MEy} d\hat{s} \Delta\sigma(\hat{s}) \Delta G(\eta, \hat{s})}{\int_{4m_c^2}^{2MEy} d\hat{s} \sigma(\hat{s}) G(\eta, \hat{s})},$$

mit

$$\sigma^{\gamma g \rightarrow c\bar{c}} = \sigma(\hat{s}) + \lambda_\gamma \lambda_g \Delta\sigma(\hat{s}),$$

wobei $\Delta\sigma^{\gamma p \rightarrow c\bar{c}X}$ und $\sigma^{\gamma p \rightarrow c\bar{c}X}$ der helizitätsabhängige und der helizitätsunabhängige Wirkungsquerschnitt der Charm-Quark-Produktion sind. Die Faktoren λ_γ und λ_g sind die Helizitäten des Photons und des Gluons. $\hat{s}$ ist das Quadrat der Schwerpunktsenergie des Photon-Gluon-Systems. M und m_c sind die Massen des Nukleons und des Charm-Quarks. $Ey = \nu$ ist die Energie des Photons.

$$\eta = x_g = \frac{\hat{s}}{2MEy} = \frac{\hat{s}}{2M\nu}$$

ist das Gluon-Äquivalent der Skalenvariable x ($x_{\text{Bjorken}} = \frac{Q^2}{2M\nu}$).

Damit ist es gelungen einen Zusammenhang zwischen A^{exp} und $\Delta G/G$ herzustellen.

Die Bestimmung von $\Delta G/G$ ist der wichtigste Teil des myonischen Programms.

³engl.: dilution factor, $f = 0.176$ für NH_3 , $f = 0.5$ für LiD

Mit dem Myonen-Strahl wird auch die transversale Spin-Strukturfunktion untersucht werden.

Ein Hauptziel des hadronischen Experimentierprogramms ist die Suche nach exotischen, stabilen Gluonen-Zuständen wie f_0 bei 1500 MeV/c, dessen Existenz durch das Crystal-Barrel-Experiment angedeutet wurde [7]. Ein weiteres Ziel ist die Spektroskopie von charm-haltigen Hadronen. Bisher gibt es keine ausführlichen Untersuchungen auf diesem Gebiet. Bei COMPASS können dank der hohen integrierten Luminosität des SPS eine große Anzahl solcher Hadronen hergestellt werden, und dadurch wird eine genaue Untersuchung von Produktion und Zerfall dieser Teilchen möglich. Mit den Ergebnissen wird die Heavy-Quark-Effective-Theorie geprüft.

2.2 Der Aufbau des Experiments

Dieser Abschnitt besteht aus drei Bereichen: der Strahl, das Target und die Detektoren. Detaillierter wird hier die Erzeugung und Führung des Strahls behandelt, da der Zähler, der im Mittelpunkt dieser Arbeit steht, zur Vermessung der räumlichen und zeitlichen Strukturen des Teilchenstrahls eingesetzt wird.

2.2.1 Der Strahl

Beim COMPASS-Experiment gibt es mehrere Strahlen mit unterschiedlichen Teilchensorten. Wie schon erwähnt, besteht das Experiment aus einem myonischen und einem hadronischen Experimentierprogramm. Beim myonischen Programm wird ein ca. 200 GeV Myonen-Strahl mit 2×10^8 Teilchen pro Spill⁴ verwendet. Zur Durchführung des hadronischen Programms können verschiedene Arten von hadronischen Strahlen erzeugt werden: ein Protonenstrahl, ein Pionenstrahl und ein Kaonenstrahl.

Alle Strahlarten kommen ursprünglich aus dem SPS⁵. Dort werden Protonen auf Energien von bis zu 450 GeV beschleunigt. Beide Teile des Experiments profitieren von einer Umstellung des SPS-Zyklus. Während der Ring früher mit einer Periode von 14.4 s, davon ca. 2 s Extraktion (13.9 %), arbeitete, gibt es jetzt eine 18.9 s lange Periode mit ca. 5 s Extraktion (26.5 %). Dadurch konnte die nutzbare Zeit fast verdoppelt werden.

Die erzeugten Protonen werden aus dem SPS in die M2-Beamline (Abbildung 2.3) [8] in Richtung Nordareal gelenkt, wo sie auf das T6-Produktionstarget

⁴Der Beschleuniger erzeugt keinen kontinuierlichen Strahl, sondern Teilchenpakete. Diese Pakete werden als Spill bezeichnet

⁵Super Proton Synchrotron accelerator

treffen. Das T6-Target besteht aus Beryllium. Seine Länge richtet sich nach der Art des gewünschten Sekundärstrahls. Zur Erzeugung eines Myonenstrahls hat das Target eine Gesamtlänge von 500 mm, für einen hadronischen Strahl 40 mm (Tabelle 2.1).

Durch Stöße mit den Beryllium-Kernen erzeugen die SPS-Protonen sekundäre

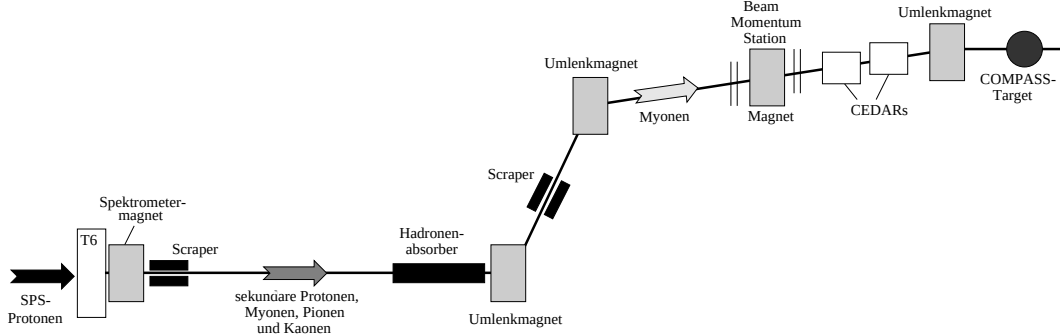


Abbildung 2.3: Die M2-Beamline (ohne Quadrupolmagnete) vom T6-Produktionstarget bis zum COMPASS-Target.

Hadronen. Dies sind überwiegend Protonen, Pionen und Kaonen. Direkt hinter dem Produktionstarget steht ein Spektrometermagnet mit Apertur. Durch diesen Aufbau werden Teilchen mit definiertem Impuls selektiert.

Dadurch wurde ein hadronischer Strahl mit definiertem Impuls erzeugt, der im hadronischen Programm auf das COMPASS-Target gelenkt wird. Für das myonische Programm werden zusätzliche Komponenten benötigt.

Die Myonen entstehen durch Zerfall der Mesonen. Die Erzeugung des Myo-

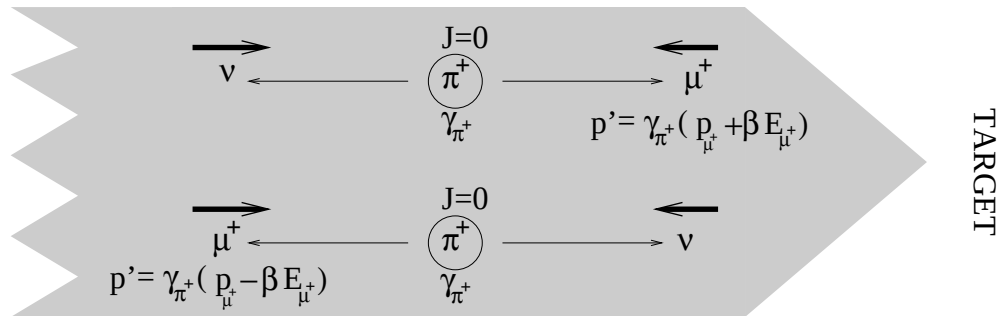


Abbildung 2.4: Zerfall der 208-GeV-Pionen (γ_{π^+}): Myonen mit relativ hohen Impulsen sind überwiegend entgegen der Strahlrichtung polarisiert. E_{μ^+} und p_{μ^+} sind Energie und Impuls des Myons im Ruhesystem des Pions.

nenstrahls ist komplizierter, denn für diesen Teil des Experiments werden *longitudinal polarisierte* Myonen benötigt. Dazu wird ausgenutzt, dass der schwache Zerfall $\pi^+ \rightarrow \mu^+ + \nu_\mu$ maximal paritätsverletzend ist [9]: Das Pion (Spin 0)

zerfällt in ein Neutrino und ein Myon mit jeweils Helizität $-\frac{1}{2}$ (Abbildung 2.4). Bei der Betrachtung der Impulsverteilung der entstandenen Myonen kann festgestellt werden, dass Myonen mit hohen Impulsen entgegen ihrer Bewegungsrichtung polarisiert sind. Durch einen zweiten Spektrometermagneten mit Apertur wird der gewünschte Impulsbereich der Myonen ausgewählt (Tabelle 2.1). Mit diesem Verfahren wird eine mittlere Polarisation $P_B = 0.8$ erreicht.

Um einen reinen Myonenstrahl zu erhalten, müssen die Hadronen herausgefiltert werden. Dies geschieht durch Einbringen eines Absorbers am Ende einer Zerfallsstrecke. Die Myonen können diesen beinahe ungehindert durchdringen, während die stark wechselwirkenden, hadronischen Bestandteile herausgefiltert werden.

Die bisher geschilderten Vorgänge laufen alle auf einer ca. 600 m langen Strecke im unterirdischen Teil der M2-Beamline ab. Da das COMPASS-Experiment aber in einer Halle aufgebaut ist, muss der Strahl nach oben gelenkt werden. Dazu dienen starke Umlenkmagnete, die den Strahl erst der Oberfläche entgegen richten und ihn dann wieder horizontal auf das Target lenken.

Für den Myonenstrahl werden Detektoren benötigt, die die Impulse der einzelnen Myonen bestimmen. Die genauen Impulse werden zur exakten Berechnung der Strahlpolarisation benötigt. Diese Detektoren sind die *Beam Momentum Stations* (BMS). Sie sind um den Umlenkmagneten 60 m vor der Experimentierhalle angeordnet. Mittels jeweils zweier Ebenen Szintillator-Hodoskope wird der Ort der Myonen vor und nach dem Magneten ermittelt. Weitere Detektoren sind für das myonische Programm des Experiments an dieser Stelle nicht nötig. Doch für den hadronischen Strahl fehlt noch ein wichtiger Detektor, denn ein Strahl von sekundären Hadronen, z.B. ein Kaonenstrahl, enthält immer große Anteile von anderen Teilchen (Pionen, Protonen, Myonen). Da alle Teilchensorten durch die Spektrometermagneten einen definierten Impuls haben, können die Teilchen allein durch eine Geschwindigkeitsmessung mit einem Čerenkov-Detektor identifiziert werden. Zu diesem Zweck wird das CEDAR-System benutzt. Dies ist ein Čerenkov-Detektor, der auf bestimmte Čerenkov-Winkel sensibilisiert ist. Er kann also genau auf eine Teilchenart eingestellt werden. Mit zwei dieser Systeme können entsprechend zwei Arten identifiziert werden. Diese Detektoren haben den Vorteil, dass sie sehr schnell sind.

2.2.2 Das polarisierte Target

Das polarisierte Target des COMPASS-Experiments ist dem Target des SMC-Experiments [10] sehr ähnlich, bietet jedoch eine höhere Akzeptanz (180 mrad gegenüber 65 mrad).

Das eigentliche Target-Material (NH_3 , ^6LiD) befindet sich in zwei hinterein-

Tabelle 2.1: Eigenschaften der Teilchenstrahlen bei COMPASS.

	Myonenstrahl	Hadronischer Strahl
SPS-Energie	400-450 GeV	400-450 GeV
Protonen/Spill auf T6	ca. 10^{13}	ca. 10^{12}
T6-Target (Be)	500 mm	40 mm
π^+ -Impuls für μ -Produktion	208 GeV/c	-
mittlerer Strahlimpuls	190 GeV/c	280 GeV/c
Teilchen pro Spill	$2 \cdot 10^8$	10^8
Spot a. d. pol. Target (σ)	$8 \times 8 \text{ mm}^2$	$3 \times 3 \text{ mm}^2$
Polarisation	0.80	0

ander liegenden, 60 cm langen Target-Zellen, in welchen es entgegengesetzt polarisiert wird. Spezielle Magnete ermöglichen sowohl longitudinale, als auch transversale Polarisation bezüglich des Strahls. Da magnetische Flussdichten von bis zu 2.5 T benötigt werden, werden supraleitende Spulen benutzt. Nach erfolgreicher Polarisation wird das Target auf 50 mK heruntergeköhlt. Bei dieser Temperatur wird die Polarisation bei 0.5 T für viele Stunden aufrecht erhalten.

2.2.3 Der COMPASS-Detektor

Mehrere Komponenten werden benötigt, um die physikalischen Vorgänge im Target beim Auftreffen des Strahls innerhalb eines möglichst großen Bereichs mit hoher Genauigkeit aufzuzeichnen. Dazu müssen die Produkte der Myon-Nukleon-Streuvorgänge nachgewiesen werden. Die Teilchen müssen identifiziert und ihre Impulse gemessen werden. Bei COMPASS beträgt der maximale Winkel, unter dem Teilchen nachgewiesen werden, ca. 200 mrad. Um dieses Ziel zu erreichen, besteht der COMPASS-Detektor aus zwei separaten Spektrometerstufen (Abbildung 2.5): eine Stufe für die Teilchen, die das Target unter großen Winkeln zum Strahl verlassen, und eine für die kleinen Winkel. Jede dieser Stufen hat einen Spektrometernagneten (SM1, SM2) und ein System zur Spurrekonstruktion und zur Teilchenidentifizierung.

Zur Spurrekonstruktion wird eine Vielzahl von Detektoren verwendet, deren Granularität den erwarteten Teilchenflüssen angepasst ist. Wenn die Spur vor und hinter dem Magneten rekonstruiert ist, ist die Bestimmung des Impulses und der Ladung möglich. Zur Teilchenidentifizierung wird zusätzlich die Energie oder die Geschwindigkeit des Teilchens benötigt, dann kann die Masse berechnet und damit das Teilchen identifiziert werden. Bei COMPASS werden

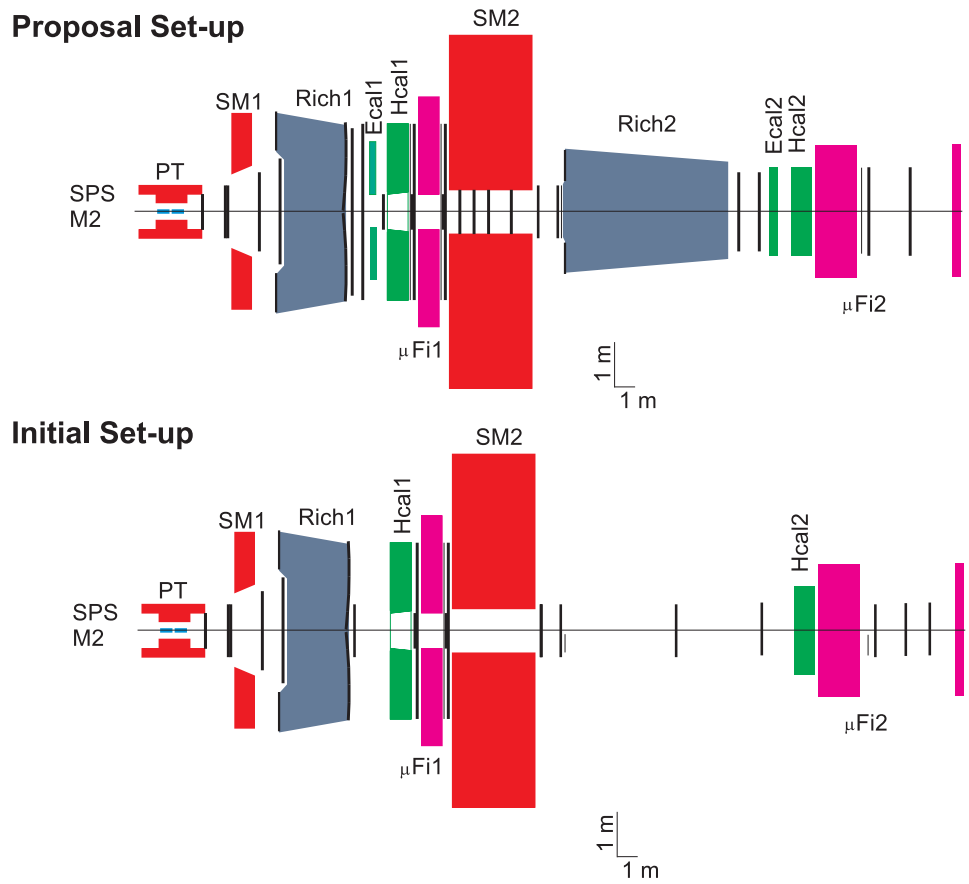


Abbildung 2.5: Die COMPASS-Detektoren — heute und morgen.

Teilchengeschwindigkeiten mit einem RICH⁶ bestimmt. Zur Bestimmung von Energien gibt es elektromagnetische und hadronische Kalorimeter.

Beim Aufbau des Experiments gibt es auch bei den Detektoren Unterschiede zwischen dem myonischen und dem hadronischen Programm des Experiments. Bis zum jetzigen Zeitpunkt ist nur ein vorläufiger Aufbau (Abbildung 2.5) des Detektors vorhanden, mit dem aber die Bestimmung von ΔG möglich ist [11].

⁶Ring Imaging Čerenkov-Counter

Kapitel 3

Das Auslese-System

Da Experimente in der Elementarteilchen-Physik immer auf der Jagd nach hoher Statistik sind, werden auf diesem Gebiet hohe Anforderungen an die Auslese-Elektronik gestellt. Das gilt auch für das COMPASS-Experiment [12]. Resultierend aus den hohen Strahlteilchen-Raten (10^7 bis 2×10^8 Teilchen pro Spill), ergeben sich je nach Target und Triggereinstellung unterschiedliche Triggerraten zwischen einem und 100 kHz (Tabelle 3.1). Unter Berücksichtigung der Zahl der Kanäle und der mittleren Trefferzahl der auszulesenden Detektoren sowie der zu erwartenden Ereignisgrößen können die Datenraten vorhergesagt werden (siehe Tabelle 3.1).

Abbildung 3.1 gibt einen Überblick über die Auslese-Elektronik. Die wich-

Tabelle 3.1: Die Trigger- und Datenraten der verschiedenen Physik-Programme bei COMPASS.

Programm	Triggerrate	Daten pro Spill
Primakoff-Programm	1-3 kHz	30-90 MByte
Myonen-Programm	10-20 kHz	600-1200 MByte
Glueball-Programm	50-100 kHz	3-6 MByte
Charm-Programm	10-100 kHz	600-6000 MByte

tigsten Komponenten werden im Folgenden beschrieben:

3.1 Das Trigger-Kontroll-System

Die Entscheidung, wann ein physikalisches Ereignis stattgefunden hat, und der Auslese-Mechanismus ausgelöst werden soll, wird von der Trigger-Logik getroffen. Die Verwaltung und Verteilung der Trigger-Signale wird vom Trigger-Kontroll-System (TCS [13]) übernommen. Das TCS ordnet jedem Trigger eine eindeutige Ereignisnummer zu, zeichnet den Ereignistyp auf und übermittelt diese Daten zusammen mit der aktuellen Spill-Nummer den CATCH-Modulen.

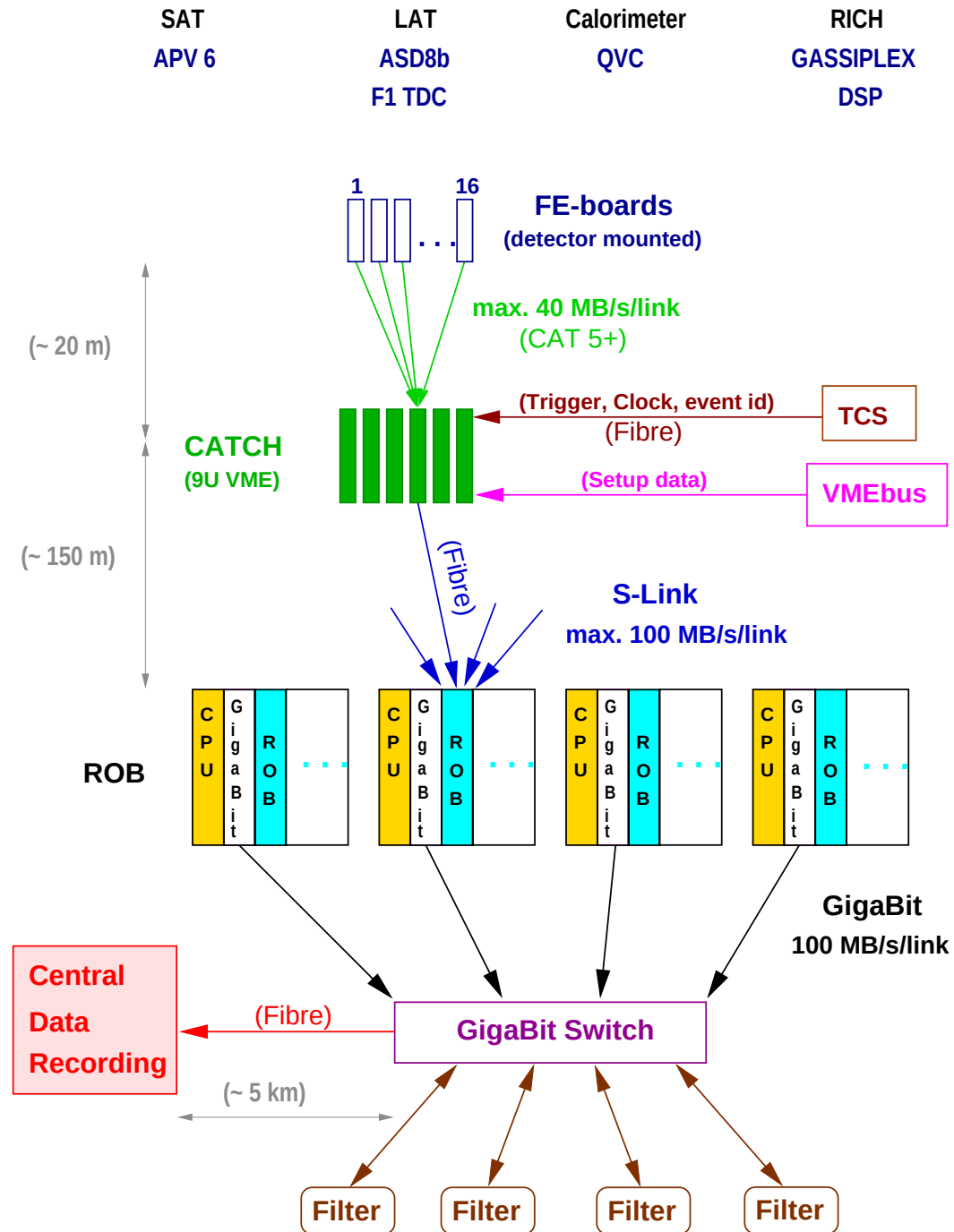


Abbildung 3.1: Das Konzept der Auslese-Elektronik des COMPASS-Experiments.

Außerdem verteilt das TCS einen stabilen 38.88-MHz-Clock-Zyklus, das globale Reset-Signal zum Zurücksetzen der Auslese-Elektronik, sowie Signale, die den Anfang und das Ende des Spills markieren. Alle Informationen werden über ein optisches Netzwerk verteilt (Abbildung 3.2).

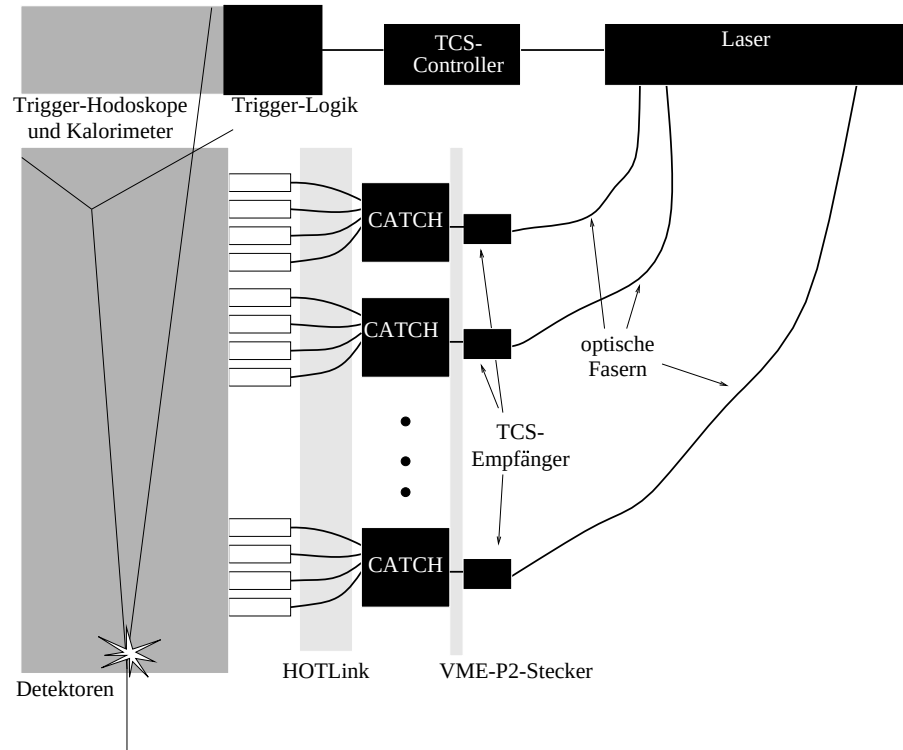


Abbildung 3.2: Die Verteilung der Trigger-Signale.

3.2 Der Weg der Daten zur Datennahme

Um die hohen Datenmengen zu verarbeiten, muss von einem gut konzipierten Auslese-Prinzip Gebrauch gemacht werden. Das COMPASS-Auslesesystem ist modular und skalierbar. Modular bedeutet, dass einzelne Komponenten jederzeit gegen modernere ausgetauscht werden können, ohne dass eine Anpassung der übrigen Elektronik notwendig ist. Skalierbar bedeutet, dass die Auslese-Kette jederzeit durch weitere Komponenten parallel erweiterbar ist. So kann das Auslesesystem angepasst werden, falls das Experiment um zusätzliche Detektoren erweitert werden sollte.

Um diese beiden Anforderungen zu erreichen, werden so viele Schnittstellen wie möglich standardisiert. Das CATCH z.B. dient zur Vereinheitlichung der Schnittstelle zwischen Datennahme und Detektor. Wenn der Datenfluss das

CATCH erreicht hat, werden alle Daten gleich behandelt, unabhängig von welchem Detektor sie stammen.

Die ersten Bausteine in der Auslekette sind die Front-End-Karten. Diese Karten digitalisieren die Daten direkt am Detektor. Die Digitalisierung erfolgt beispielsweise durch Diskriminatoren und Vorverstärker. Sie wandeln die Signale in definierte LVDS-Pulse um. LVDS ist ein differenzieller Signalstandard, dessen Definition Abbildung 3.3 entnommen werden kann. Auf Twisted-Pair-

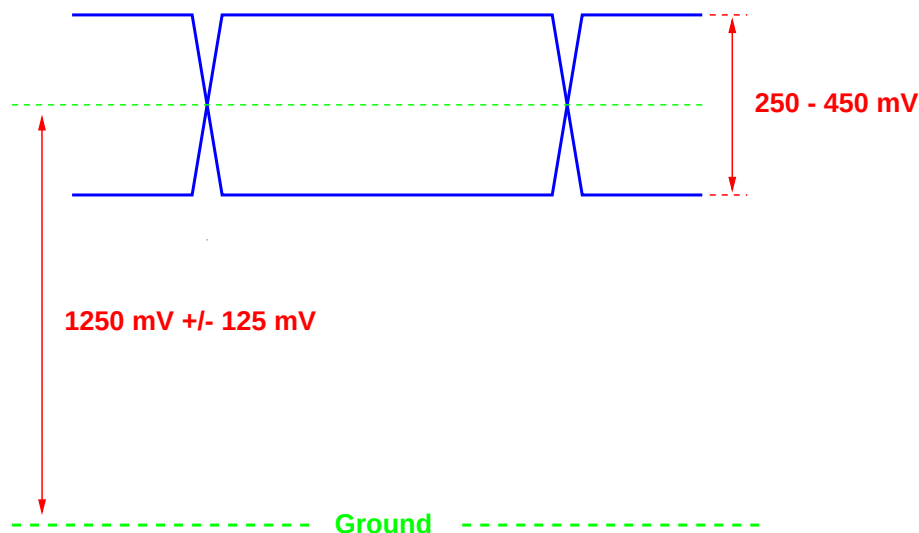


Abbildung 3.3: Der differenzielle Signalstandard LVDS.

Kabeln werden die Signale an Common Mezzanine Cards (CMC, Abschnitt 4.1) auf dem CATCH gesendet (Abbildung 3.4). Es gibt zwei verschiedene CMCs, die LVDS-Signale verarbeiten können: die Scaler¹-CMC, das Objekt dieser Arbeit, und die Time-to-Digital-Converter-CMC (TDC-CMC). Auf der TDC-CMC befinden sich vier an der Universität Freiburg entwickelte F1-Time-to-Digital-Converter (F1-TDC), die die Zeitdifferenzen zwischen zwei Signalen messen können [14].

Es gibt Front-End-Karten, auf denen nicht nur Diskriminatoren sind, sondern auch TDCs oder Analog-Digital-Converter (ADC). Solche Front-End-Karten geben keine LVDS-Signale, sondern ganze Datenworte aus. Die Daten werden über das Kommunikationsprotokoll HOTLink [15, 16] zum CATCH übermittelt. Dazu befinden sich auf den Front-End-Karten HOTLink-Sender-Chips und auf den CATCH-Modulen werden CMCs mit HOTLink-Empfänger-Chips aufgesteckt. Die HOTLink-Chips können pro Sekunde 40 MB übertragen. Wahlweise können die Daten über ein Twisted-Pair-Kabel (HOTLink-CMC)

¹engl. Synonym für Zähler

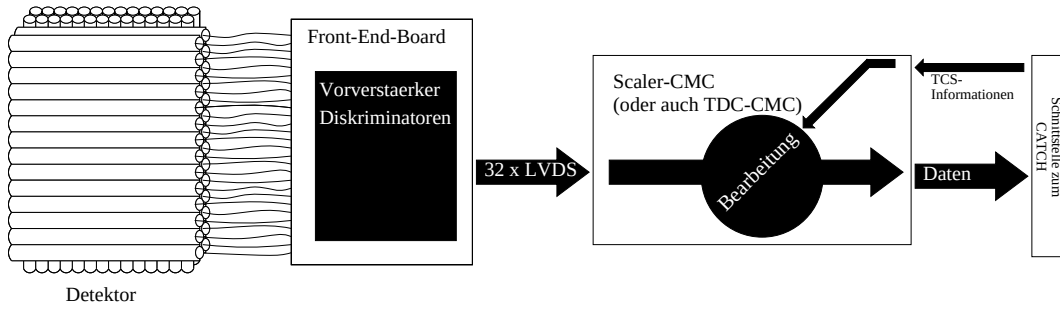


Abbildung 3.4: Der Scaler, in der Auslekette eingebunden zwischen Detektor und CATCH.

oder eine optische Faser (HOTFibre-CMC) übertragen werden (Abbildung 3.5).

Für alle Arten von Front-End-Karten gilt, dass die Daten auf den CMCs zwi-

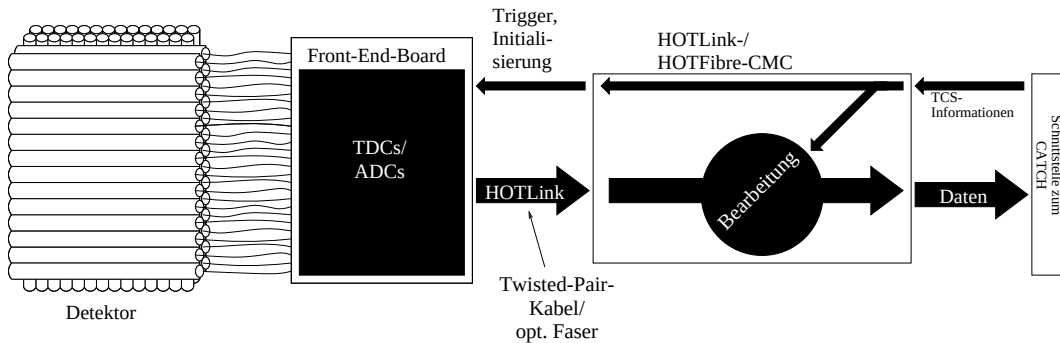


Abbildung 3.5: Datenübertragung per HOTLink.

schengespeichert werden. Hier greift das CATCH auf sie zu. Das CATCH ist ein 9U-Modul² für einen VME³-Überahmen [17]. Auf ihm werden die Daten gebündelt und nach Ereignis-Nummer sortiert. Außerdem erhält das CATCH Informationen vom Trigger-Kontroll-System (TCS) (Abschnitt 3.1).

Es fügt diese Informationen den Daten hinzu, so dass diese später richtig zusammengesetzt werden können.

Dann sendet es die Daten über die S-LINK-Schnittstelle [18] an die Zwischenspeicher weiter. S-LINK ist ein am CERN entwickelter Standard zur Datenübertragung mit bis zu 160 MB/s. Beim S-LINK des COMPASS-Experiments werden die Daten über optische Fasern übertragen. Der S-LINK-Sender befindet sich nicht direkt auf dem CATCH-Einschub, sondern auf einer weiteren Steckkarte. Dadurch wird auch an dieser Stelle die modulare Bauweise real-

²367 x 400 mm Eurocard

³Versabus Modified for Eurocard

isiert. Wenn es im Verlauf des Experiments nötig sein sollte, Daten mit höheren Raten zu übertragen, könnte auf eine andere Übertragungsart umgestiegen werden.

In den Zwischenspeichern werden die im Spill anfallenden Daten gespeichert und können während der Spill-Pause weiterverarbeitet werden. In einem Readout-Buffer-Computer (ROB) befinden sich je ein 256-MB-Zwischenspeicher auf vier PCI-Steckkarten. Die Readout-Buffer-Computer verfügen über zwei separate PCI-Busse und zwei Prozessoren und sind für das Weiterleiten der Daten verantwortlich. Dies geschieht über Gigabit-Ethernet-Interface-Karten. Die Daten gehen über dieses Netzwerk weiter an die Computer, die die Ereignisse zusammensetzen und werden vom zentralen Datenspeichersystem des CERNs (CDR) aufgenommen.

Kapitel 4

Physikalische und technische Vorgaben

4.1 Vorgaben des Detektors

Maximale Frequenz des Zählers

Beim Durchgehen der Liste der Spezifizierungen für den Zähler sollte mit der maximalen Zählrate begonnen werden. Für das COMPASS-Experiment ist eine Rate von 200 MHz notwendig. Diese Zahl richtet sich nach der minimalen Totzeit der Detektoren. Bei COMPASS haben die szintillierenden Fasern und die Trigger-Hodoskope mit 5 ns die kleinste Totzeit. Wenn diese Detektoren ein Signal gegeben haben, können sie erst nach 5 ns das nächste Mal ein Teilchen nachweisen. Im Extremfall könnten die Detektoren also ein Signal mit einer Periodendauer von 5 ns erzeugen, dies entspricht einem 200-MHz-Signal (Abbildung 4.1). Der Zähler muss somit ein regelmäßiges 200-MHz-Signal fehlerfrei zählen können.

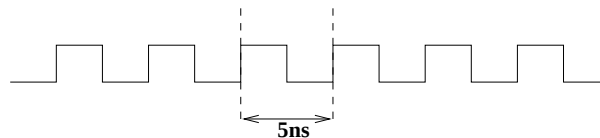


Abbildung 4.1: Ein 200 MHz-Signal.

Die Datenbreite des Zählerregisters soll 31 bit betragen. Das sollte den Anforderungen bei COMPASS genügen. Mit einem 31-bit-Register ist es möglich, bis 2.15 Milliarden zu zählen. Das reicht auch bei der maximalen Rate 200 MHz für mehr als 10 s, was etwa zwei kompletten Spills entspricht.

Die Auslese und die Einbindung in das Trigger-Kontroll-System

Einfache Zähler zählen die eingehenden Signale, werden dann angehalten und anschließend ausgelesen. Sie können nach der Auslese nicht mehr weiterzählen oder haben zumindest eine beträchtliche Totzeit. Diese Totzeit darf beim Zähler für COMPASS nicht vorhanden sein. Der Zähler wird im Experiment mit jedem Trigger ausgelesen werden. Die Triggerfrequenz bei COMPASS liegt zwischen 10 und 100 kHz. Der Zähler muss somit alle $100\text{ }\mu\text{s}$ totzeitfrei ausgelesen werden können. Gerade die totzeitfreie Auslese (Abbildung 4.2) ist eine Eigenschaft, die bisher bei kommerziellen Zählern nicht zu finden ist. Die Trigger werden im Experiment vom Trigger-Kontroll-System (TCS, Seite 17) verschickt. Das TCS schickt neben diesem Signal auch das Begin-Of-Burst- und das End-Of-Burst-Signal (BOB, EOB,). Diese Signale markieren den Anfang und das Ende des Spills. Bei BOB soll der Zähler im Experiment zurückgesetzt werden.

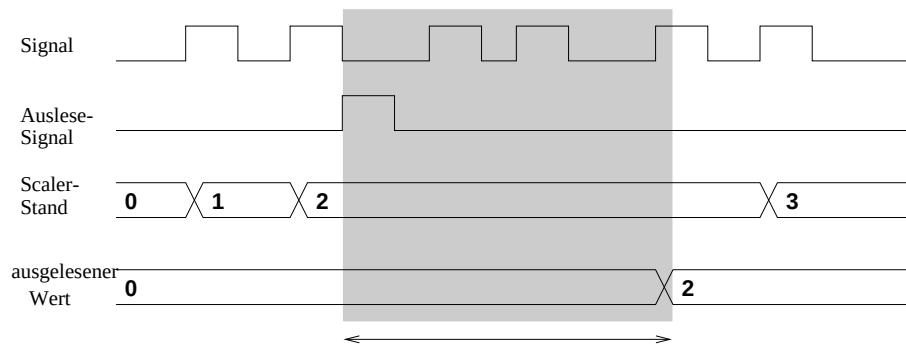


Abbildung 4.2: Totzeit nach der Auslese. Diese Totzeit darf beim Zähler für COMPASS *nicht* vorhanden sein.

Gate-Funktion

Im COMPASS-Experiment gibt es einige Gas-Detektoren, bei denen es durch Fehlfunktionen (vollständige Entladung) zu relativ langen Totzeiten von einigen $100\text{ }\mu\text{s}$ kommen kann. Während dieser Zeit sollen keine weiteren Daten von den Detektoren genommen werden. Der Zähler müsste währenddessen angehalten werden. Zu diesem Zweck soll es das *Gate*-Signal (Abbildung 4.3) geben. Dieses Signal zeigt eine solche Pause an. Im Zählerdesign muss dies berücksichtigt werden. Auch während der Spillpause wird der Zähler mit diesem Signal angehalten werden.

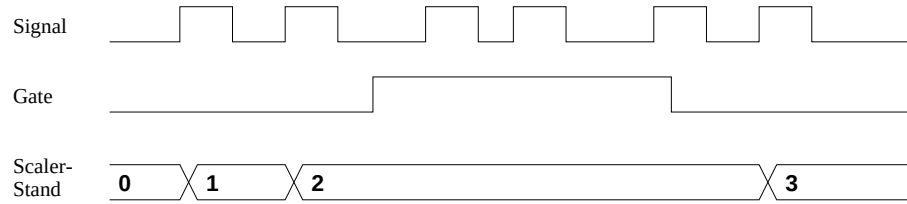


Abbildung 4.3: Die Gate-Funktion des Zählers. Während das Gate-Signal 1 ist, wird nicht weitergezählt.

Ausgleich der Trigger-Verzögerung

Die Erzeugung des Triggers wird im Experiment ca. $1\text{--}2\ \mu\text{s}$ dauern (Abbildung 4.4). Das heißt, dass alle Detektoren den Trigger $1\text{--}2\ \mu\text{s}$ zu spät bekommen. Im Falle des Zählers bedeutet das, dass ein falscher, zu hoher Zählerstand übermittelt wird. Die klassische Methode einen solchen Fehler zu verhindern sind lange Kabel, auf denen die Signale entsprechend verzögert werden. Diese Methode ist aber kostenintensiv. Außerdem kann sich die Verzögerung im Laufe der Messungen ändern. Häufig wird dieser Effekt auch vernachlässigt. Doch bei COMPASS ist die Strahlrate mit bis zu 200 Millionen Teilchen pro Spill und die Triggerfrequenz mit bis zu $100\ \text{kHz}$ so hoch, dass sich der Fehler zu stark bemerkbar machen würde. Aus diesen Gründen soll der Zähler in der Lage sein, seine Signale intern um eine variable Zeit zwischen 0 und $3\ \mu\text{s}$ zu verzögern.

Bauweise

Der Zähler soll kein autarkes Gerät sein, sondern seinen Platz in der Architektur der COMPASS-Auslese-Elektronik haben. Er soll als Common Mezzanine Card (CMC) [19] für das CATCH-Board entwickelt werden. Damit ist auch die Art der Schnittstelle (CMC-Schnittstelle) und der Ausgänge festgelegt. Auch die Abmessungen ($74\ \text{mm} \times 149\ \text{mm}$) richten sich nach der CMC-Norm (Abbildung 4.5).

Sämtliche TCS-Informationen bezieht der Zähler vom CATCH via CMC-Schnittstelle. Allein die Eingangssignale kommen direkt vom Detektor zum Eingangsstecker. Der Signalstandard für die Eingangssignale ist LVDS (Seite 20) oder LVPECL [20]. Jede Scaler-CMC soll 32 dieser differenziellen Kanäle zählen können. Es soll außerdem die Möglichkeit bestehen, den Zähler so umzurüsten, dass er auch in der Lage ist, 64 LVTTTL-Kanäle¹ [21] zu zählen. Insgesamt sind momentan vier Scaler-CMCs gebaut, die insgesamt 128 differenzielle Signale zählen können.

¹nicht-differenzieller Signalstandard

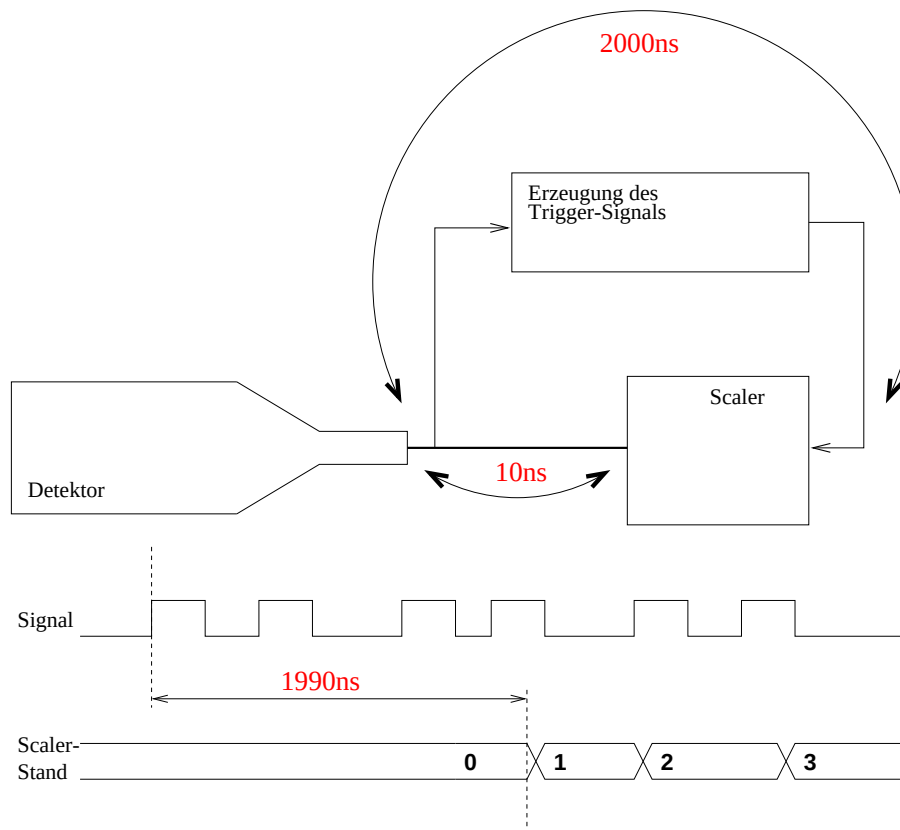


Abbildung 4.4: Verzögerung der Ausgabe des Zählerstands zum Ausgleich der Trigger-Verzögerung.

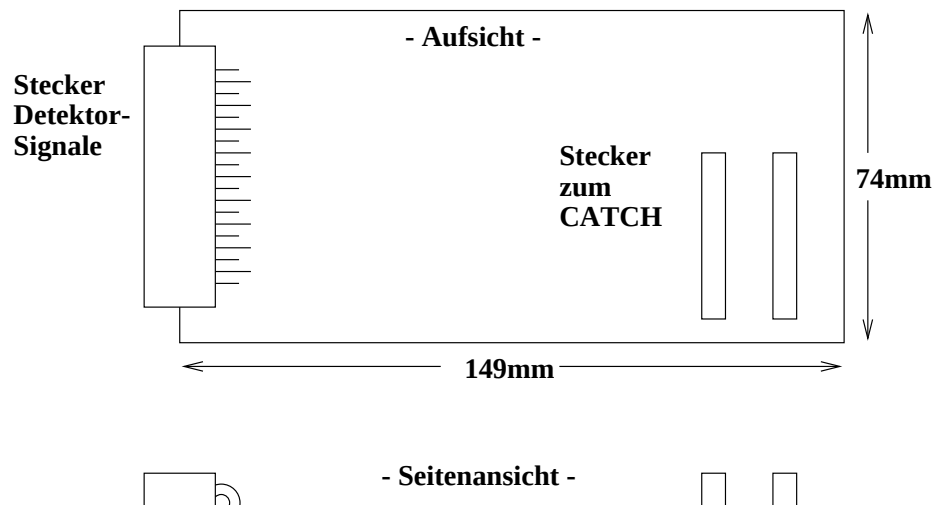


Abbildung 4.5: Der CMC-Standard für CATCH-Aufsteckkarten bei COMPASS.

Die gesamte Logik des Zählers soll in einem FPGA² untergebracht werden. Näheres wird sehr ausführlich in Abschnitt A.2 des Anhangs erklärt.

Einfache Bedienbarkeit und Kompatibilität

Ein weiteres wichtiges Kriterium für eine Elektronikkomponente in einem modernen Hochenergie-Experiment ist die einfache Bedienbarkeit. Während der Datennahme wird es häufig vorkommen, dass auch Mitglieder anderer Arbeitsgruppen mit dem Zähler arbeiten. Deshalb ist eine Art Plug-and-Play-Komponente erstrebenswert, die so wenig Konfigurationsarbeit wie möglich macht. Da beispielsweise FPGAs bei jedem Ausschalten ihr Programm verlieren, muss ein Weg gefunden werden, den Vorgang des Programmierens komfortabel zu gestalten. Allgemein sollte die Konfiguration, dort wo sie unerlässlich ist, so einfach wie möglich sein.

Um die Einbindung in die COMPASS-Auslese-Elektronik zu erleichtern, werden beim Design der Scaler-CMC die anderen bereits fertigen CMCs, wie TDC-CMC und HOTLink-CMC, als Leitbild dienen. Ein ähnliches Datenformat und ähnliche Bauteile auf der CMC erleichtern später die Inbetriebnahme und Fehlersuche, weil auf viele Erfahrungen zurückgegriffen werden kann.

Mehrere Anwendungsmöglichkeiten

Bei der Entwicklung des Zählers wurde sehr bald deutlich, dass eine auf FPGAs basierende Schaltung eine gute Grundlage dafür bietet, noch andere Auslese-Elektronik-Elemente zu realisieren. Bei der Wahl eines genügend großen FPGAs gibt es eine Vielzahl anderer Funktionen, die diese Einheit übernehmen kann. Die Scaler-CMC hat 32 Eingänge, die mit jeder Signalart und auch als In- oder Output betrieben werden können.

Ein Beispiel wäre ein Input-Register. Es wäre aber auch möglich einen TDC mit geringer Zeitauflösung oder sogar eine komplette Triggerlogik zu programmieren.

4.2 Ein improvisierter Test

Nachdem im letzten Abschnitt die Anforderungen an den Zähler aufgelistet wurden, folgt jetzt eine Passage, in der von den ersten Tests des Zähler-Designs berichtet wird. Bei diesen Tests wurde ein vorläufiges Design benutzt, das die wesentlichen Elemente des endgültigen Designs bereits enthielt. In Kapitel 5 und 6 werden die elektronischen Grundlagen und die logischen Konzepte vorgestellt, die auch diesem ersten Testdesign zu Grunde lagen.

²Field Programmable Gate Array

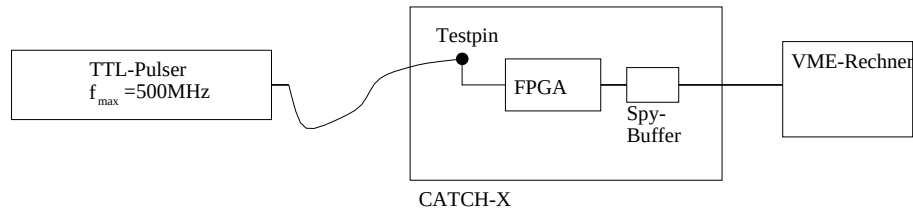


Abbildung 4.6: Der Aufbau des ersten Tests mit Pulser, Catch-X und VME-Rechner.

Ganz zu Beginn der Entwicklung des Zählers musste zunächst getestet werden, ob das erdachte Zählerdesign überhaupt funktionsfähig ist. Um dies zu testen, wurde ein Test mit dem CATCH-Vorgänger Catch-X [22] konzipiert (Abbildung 4.6). Der schnellste FPGA auf dem Catch-X ist der XC4000XLA [23]. Diesem kann ein Zählsignal zugeführt werden. Um den Zähler auszulesen, wird ein anderes Element des Catch-X-Moduls benutzt: die VME-Schnittstelle. Diese beinhaltet ein Auslese-Puffer (Spy-Buffer), das es erlaubt, über den VME-Rechner ausgelesen zu werden. Die Zähler-Daten werden hinein geschrieben und direkt ausgelesen, ohne die komplette Ausleseketten (S-LINK, ...) durchlaufen zu müssen.

Zum Generieren des Signals wurde ein Pulser verwendet. So konnten die Grenzen der Fähigkeiten des Zählers genau vermessen werden.

Die Tests verliefen erfolgreich. Das Design-Prinzip erwies sich als geeignet. Es konnten Raten über 235 MHz gezählt werden bei einer minimalen Pulsbreite von 0.8 ns (Abbildung 4.7). Auch die totzeitfreie Auslese funktioniert bei diesem Zähler-Design. Die Ergebnisse des Tests waren sehr ermutigend, und machten Hoffnung, dass auch das endgültige Design funktionieren würde.

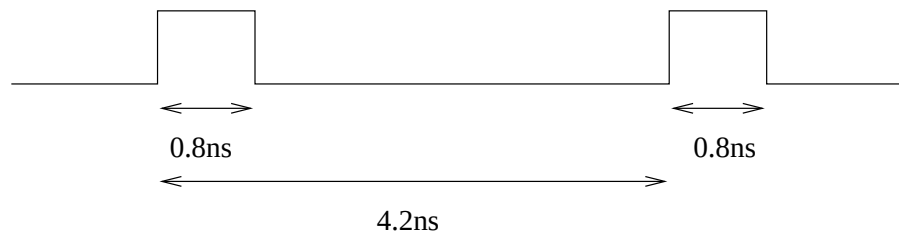


Abbildung 4.7: Beim ersten Test konnten Pulse mit einer minimalen Breite von 0.8 ns und einer minimalen Periode von 4.2 ns ($\Leftrightarrow \nu = 234.7 \text{ MHz}$) fehlerfrei gezählt werden.

Kapitel 5

Grundlagen eines Zählers

In diesem Kapitel werden die elektronischen Grundlagen eines Zählers beschrieben, die bei der Entwicklung eines schnellen Zählers verwendet werden.

5.1 Elektronische Grundlagen

Das Prinzip eines Zählers basiert auf einem Register, in dem der aktuelle Wert gespeichert ist, und einer Logik, die aus dem aktuellen Zählerstand den nächsten berechnet ($i \rightarrow i + 1$).

5.1.1 Das Flip-Flop

Ein Register setzt sich aus Flip-Flops (Abbildung 5.1) zusammen. Flip-Flops gehören zu den wichtigsten Bestandteilen einer modernen elektronischen Schaltung. Sie dienen als Speicherelemente [24]. Es handelt sich um Bausteine „mit Gedächtnis“. Ein Flip-Flop kann zwei stabile Zustände annehmen (1 und 0). Die Grundform eines Flip-Flops hat einen Dateneingang (D) und einen Datenausgang (Q). Der Wert am Dateneingang wird genau in dem Moment gespeichert und auf den Ausgang gelegt, in dem am Clock-Eingang (CLK) eine aufsteigende Signalflanke auftritt.

Die Grundform eines Flip-Flops verfügt nur über diese drei Eingänge. Es gibt auch andere Flip-Flop-Formen mit weiteren Steuersignalen. Eines davon ist das Clear-Signal (CLR), mit welchem das Flip-Flop zu jedem Zeitpunkt in den Grundzustand (0) gesetzt werden kann. Weiter gibt es den Clock-Enable-Eingang (CE). Wenn ein solcher Eingang vorhanden ist, schaltet ein Flip-Flop nur bei einer 1 am CE-Eingang.

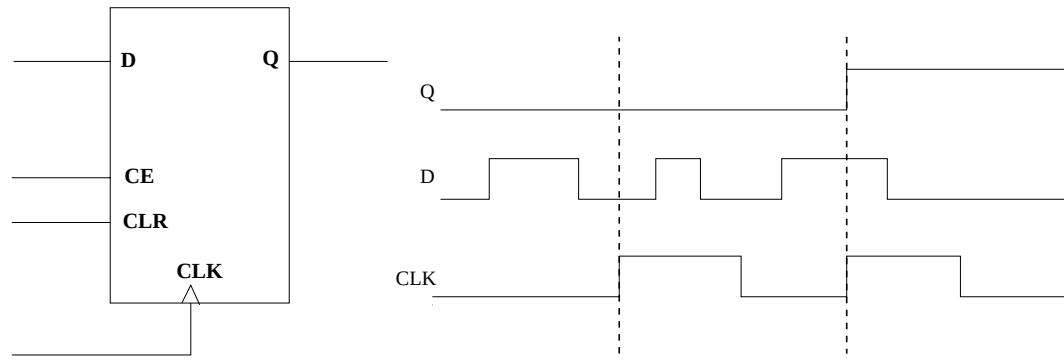


Abbildung 5.1: Das Schaltverhalten eines Flip-Flops.

5.1.2 Der Standardzähler

Ein Standardzähler (Abbildung 5.2) besteht aus einem Register und einer Logik, die aus dem aktuellen Zählerstand i den darauf folgenden Zählerstand $i + 1$ berechnet [25]. Ein Zähler kann also auch als ein Addierer aufgefasst werden,

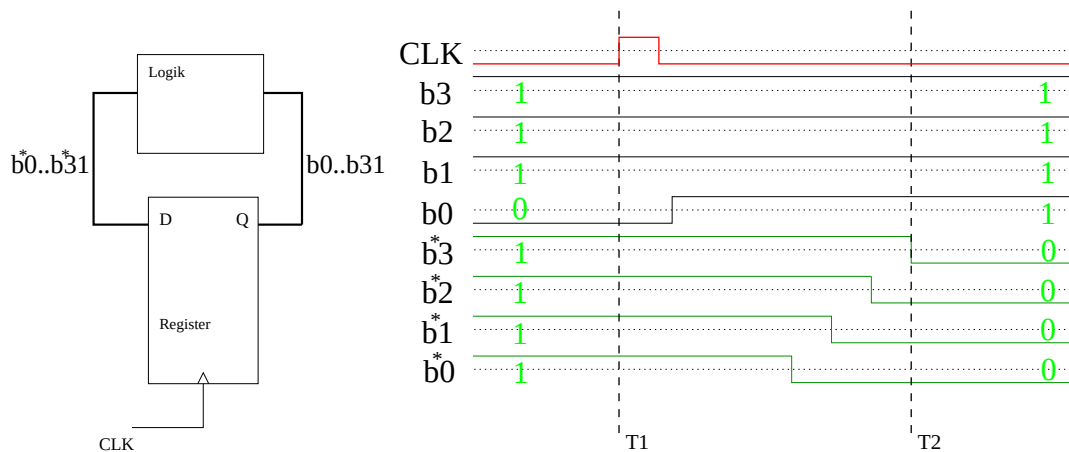


Abbildung 5.2: Nach einer Taktflanke braucht die Logik die Zeit $\Delta T_{min} = T2 - T1$, um den neuen Zählerstand zu berechnen.

der zu seinem Speicherinhalt mit jeder Taktflanke eine 1 addiert. Bei der Berechnung der Bits des Registers tritt, wie bei jedem Addierer, das Problem der Überträge auf. So ist zum Beispiel die Funktion, die zur Berechnung des Least Significant Bit (LSB, b^*0^1) dient, eine Funktion, die nur vom LSB selbst abhängt². Die Funktion für das zweite Bit ($b1$), hängt vom zweiten und, wegen

¹Im Folgenden wird zwischen den im Register gespeicherten Werten $b0..b31$ und den von der Logik berechneten Werten $b^*0..b^*31$ unterschieden.

² $\bar{b}0$ ist das Inverse von $b0$.

des möglichen Übertrags, ersten Bit des Registers ab. Dies setzt sich fort bis zum Bit 32, welches eine Funktion aller 32 Bit des Registers ist.

$$\begin{aligned}
 b^*0 &= f(b0) &= \overline{b0} \\
 b^*1 &= f(b0, b1) &= (\overline{b0} \wedge b1) \vee (b0 \wedge \overline{b1}) \\
 b^*2 &= f(b0, b1, b2) &= (b0 \wedge b1 \wedge \overline{b2}) \vee (\overline{b0} \wedge \overline{b1} \wedge b2) \vee \dots \\
 &\vdots &\vdots \\
 b^*31 &= f(b0, b1, b2, \dots, b31) &= \dots
 \end{aligned}$$

Die Funktion für das 32. Bit ist sehr komplex. Die Zeit, die notwendig ist, um aus dem alten Zählerstand den neuen zu berechnen, ist auf Grund der vielen Überträge nicht zu vernachlässigen. Nach einer steigenden Taktflanke darf für eine Zeit ΔT keine weitere kommen. Wenn trotzdem eine zweite Taktflanke am Register ankommt, führt dies den Zähler in einen undefinierten Zustand. ΔT ist im wesentlichen identisch mit der Zeit, die benötigt wird, um das Most Significant Bit (MSB, hier das 32. Bit) zu berechnen. Es steigt also mit der Größe des Zählers an. Je kleiner der Zähler, desto schneller kann er arbeiten. Mit ΔT wird auch die maximale Frequenz festgelegt. Und es muss bei einem Standardzähler sichergestellt sein, dass diese Zeit zwischen zwei Pulsen auf der Clockleitung nicht unterschritten werden kann, sonst kann er nicht funktionieren:

$$\Rightarrow \nu < \frac{1}{\Delta T}. \text{ bzw. } \Delta T < \frac{1}{\nu}$$

Dieser Fall würde dazu führen, dass im Moment des Schaltens des Registers ein undefinierter Zustand am Register anliegen könnte. Zum Beispiel könnte ein Zähler von 0111 nach 0000 statt nach 1000 schalten, weil die Zeit zwar für die drei letzten Bits, nicht aber für das MSB gereicht hat.

Die Auslese

Eine weitere Schwierigkeit ist die Auslese. Ein üblicher Auslesemechanismus besteht aus einem zweiten Register, dessen Eingang mit dem Ausgang des Zählerregisters verbunden ist (Abbildung 5.3). Am CLK-Eingang liegt das Auslesesignal an. Wenn das Auslesesignal von 0 auf 1 wechselt, wird der Wert aus dem Zählerregister in das Ausleseregister übernommen und dort weiterverarbeitet.

Es kann der Fall eintreten, dass das Zählerregister genau im Moment der Auslese inkrementiert wird. Da nicht alle Flip-Flops exakt zum gleichen Zeitpunkt

schalten, entstehen hier Probleme. Die Bits können teilweise mit dem neuen Wert und teilweise mit dem alten Wert ausgelesen werden. Dies führt zu völlig falschen Werten. Bei den meisten Anwendungen können diese Probleme nicht vernachlässigt werden, und es muss auf andere, fortgeschrittene Zählerkonzepte zurückgegriffen werden.

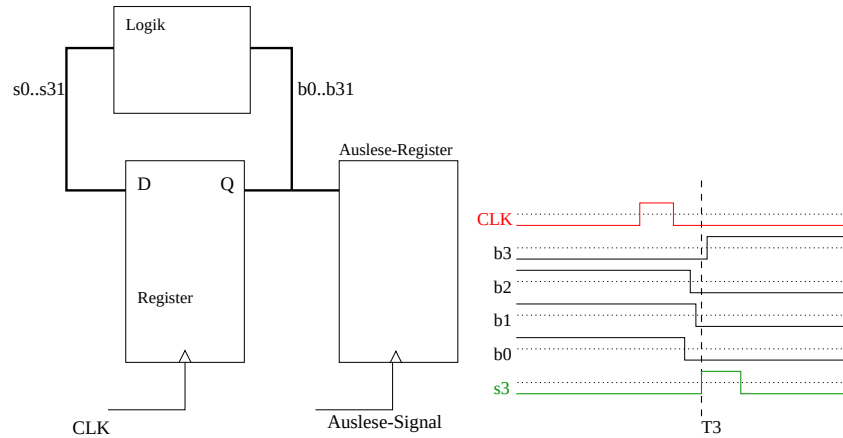


Abbildung 5.3: Bei der Auslese zum Zeitpunkt T3, wird statt dem Wert 1000 der Wert 0000 ausgelesen.

5.1.3 Konzept zweier Zähler in einer Kaskade

Die einfachste Möglichkeit ist das Konzept von zwei kaskadierten Zählern. Wobei der erste Zähler ein kleiner, schneller Vorzähler ist und der zweite ein Standardzähler. Der Vorzähler sollte so beschaffen sein, dass sowohl die Ausleseprobleme, als auch die Probleme, die auftreten, wenn ΔT unterschritten wird, nicht in Erscheinung treten.

Man verbindet zwei kaskadierte Zähler, indem der Vorzähler mit einem Terminal-Count-Signal (TC) ausgestattet wird, das den Wert 1 annimmt, wenn der Vorzähler einmal durchlaufen wurde, und sonst 0 ist. Dieses TC wird mit dem CE des Registers des Standardzählers verbunden. Nachdem der Vorzähler einmal durchgelaufen ist, schaltet der Standardzähler mit der nächsten steigenden Taktflanke.

Für einen Vorzähler mit der maximalen Rechenzeit Δt , der in einem Durchlauf bis n zählen kann, und einen Standardzähler mit ΔT , gilt

$$\Delta T < n \cdot \Delta t$$

und

$$\Delta t < \frac{1}{\nu}.$$

Bei einer vorgegebenen Rate von $\nu = 200 \text{ MHz}$ reicht ein 3-bit-Vorzähler mit $\Delta t < 5 \text{ ns}$ und ein Standardzähler mit $\Delta T < 8 \cdot 5 \text{ ns} = 40 \text{ ns}$.

Es stellt sich immer noch die Aufgabe, einen Vorzähler zu entwerfen, der nicht völlig falsche Werte annimmt, wenn eine Taktflanke zu früh kommt, und mit dem eine totzeitfreie Auslese möglich ist. Zu diesem Zweck bietet sich z.B. der Gray-Code [25, 26, 27] an.

5.1.4 Gray-Code

Bei einer Analyse der Probleme des Standardzählers stellt man fest, dass ihr Ursprung darin liegt, dass im normalen binären Stellenwertsystem zwischen zwei aufeinanderfolgenden Werten in der Hälfte der Fälle mehr als ein Bit seinen Wert ändert. Ein Extremfall ist der Übergang $01111111 \rightarrow 10000000$. Wenn sich zwischen zwei Werten immer nur ein Bit ändern würde, würde der Zähler eine zu früh kommende Taktflanke einfach ignorieren. Auch die Auslese wäre immer ± 1 richtig. Eine Umordnung der Binärzahlen, die dies leistet, ist der Gray-Code (Tabelle 5.1), benannt nach Frank Gray (1953).

Ein Zähler, der mit diesem Code statt mit Binärzahlen zählt, kann mit klei-

Tabelle 5.1: Die Sequenz eines 4-bit-Gray-Counters.

Dezimal	Binär	Gray-Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000
0	0000	0000

neren Fehlern ausgelesen werden und kann auch nicht in einen undefinierten Zustand kommen. Leider hat ein solcher Zähler immer noch Nachteile. Zum

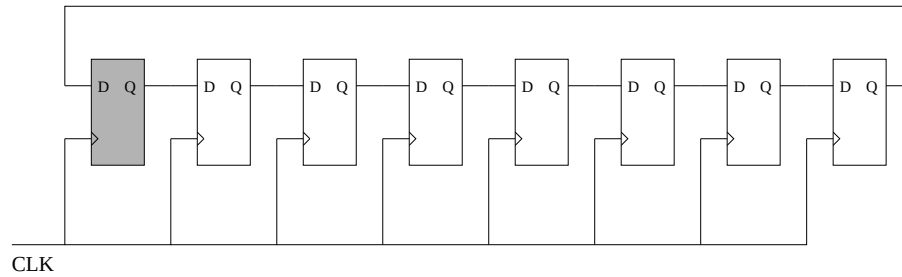


Abbildung 5.4: 8-bit-Schieberegister-Vorzähler: Das graue Flip-Flop nimmt nach dem Zurücksetzen den Wert 1 an, alle anderen den Wert 0.

einen muss jeder Gray-Code-Wert in einen Binärwert übersetzt werden. Dies lässt sich jedoch einfach realisieren. Zum anderen ist die Logik, die zum Inkrementieren benötigt wird, ähnlich kompliziert wie bei einem Binärzähler. ΔT ist recht groß. Die komplizierte, langsame Logik begrenzt in diesem Fall die maximale Frequenz. Ein Zählerprinzip mit einer einfacheren, schnelleren Logik wäre deshalb wünschenswert.

5.1.5 Das Schieberegister

Ein sehr schneller Zähler lässt sich aus einem Schieberegister (Abbildung 5.4) entwickeln. Ein Schieberegister besteht aus n Flip-Flops, deren Ausgänge mit den Eingängen der nächsten Flip-Flops verbunden sind [25]. Die CLK-Eingänge der Flip-Flops werden alle an dieselbe Clock angeschlossen. Jedes Flip-Flop gibt bei einer ansteigenden Clock-Flanke seinen Wert an das nächste Flip-Flop weiter. Daher der Name *Schieberegister*. Die Werte werden durch die Register „geschoben“. Diese Schaltung kann z.B. zur Verzögerung von digitalen Signalen dienen. Die Verzögerung T_V lässt sich mit $T_V = \frac{n}{\nu_{CLK}}$ berechnen. Aus diesem Element kann auch ein Zähler gebaut werden. Dazu muss der Ausgang des letzten Flip-Flops mit dem Eingang des ersten Flip-Flops verbunden werden. Dieser geschlossene Zyklus kann als schneller Vorzähler benutzt werden, wenn dem ersten Flip-Flop als Startwert eine 1 zugeordnet wird und allen anderen eine 0. Bei jeder Taktflanke wird die 1 ein Flip-Flop weiter geschoben (Tabelle 5.2). Zum Auslesen muss an jedes Flip-Flop ein Auslese-Flip-Flop angeschlossen werden. Dieser Vorzähler durchläuft n verschiedene Zustände, zählt also von 0 bis $n - 1$. Natürlich müssen auch hier, ähnlich wie beim Gray-Code, die Werte aus dem Schieberegister-Code in Binärcode umgerechnet werden. Doch auch das ist kein Problem.

Der Vorteil eines solchen Schieberegisterzählers liegt darin, dass sich zwischen den Flip-Flops keine Logik befindet. ΔT ist sehr klein und es können mit diesem Zähler sehr hohe Frequenzen erreicht werden. Ein Nachteil ist dagegen, dass sehr viele Flip-Flops gebraucht werden. Mit n Flip-Flops wird nur bis $n - 1$

Tabelle 5.2: Die Sequenz eines 8-bit-Schieberegisters.

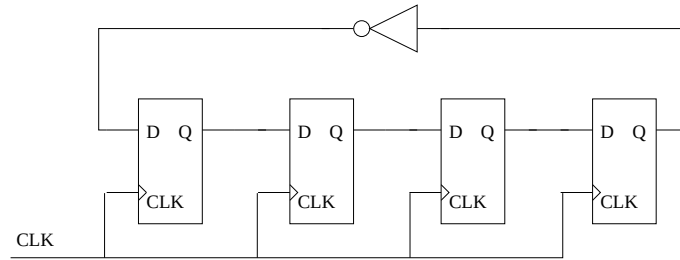
Dezimal	Schieberegister
0	10000000
1	01000000
2	00100000
3	00010000
4	00001000
5	00000100
6	00000010
7	00000001
0	10000000

gezählt. Ein Binärzähler könnte dagegen bis $2^n - 1$ zählen. Da ein FPGA in der Regel über eine ausreichende Anzahl an Flip-Flops verfügt, wiegt dieses Argument nicht sehr schwer. Wichtiger ist dagegen, dass auch beim Schieberegister auf jeder ansteigenden Taktflanke *zwei* Flip-Flops schalten. Das i -te Flip-Flop schaltet von 0 auf 1, und der $(i - 1)$. von 1 auf 0. Wenn nun eine Taktflanke zu früh kommt oder eine schlechte Signal-Form hat, kann es sein, dass nur einer der Flip-Flops schaltet. Die Konsequenz ist, dass der Vorzähler in einen undefinierten Zustand gelangt, also entweder zwei Flip-Flops mit Wert 1 hat, oder alle Flip-Flops haben den Wert 0 angenommen. Beides würde den Zähler bzw. die Auslese zum Stillstand bringen. Ein Schieberegister ist also nur bedingt geeignet.

5.1.6 Der Johnson-Counter

Der ideale Kompromiss aus Geschwindigkeit, Effizienz und guter Auslesbarkeit ist der Johnson-Counter (Abbildung 5.5). Der Johnson-Counter ist im Aufbau dem Schieberegister sehr ähnlich. Der Unterschied liegt allein darin, dass das letzte mit dem ersten Flip-Flop *über einen Inverter* verbunden ist. Auch sind *alle* Flip-Flops bei diesem Zähler im Grundzustand 0. Diese kleinen Unterschiede bewirken in den wesentlichen Punkten ein völlig anderes Verhalten. Schon die durchzählte Sequenz eines 4-bit-Johnson-Counters (Tabelle 5.3) sieht ganz anders aus.

Wie man sieht, schaltet hier wirklich jeweils nur ein Flip-Flop. Undefinierte Zustände sind nicht möglich. Die Auslese ist immer ± 1 richtig, und die Logik zwischen den Registern ist sehr einfach. Nur zwischen zwei der vier Flip-Flops befindet sich ein Inverter. Aber auch dies führt zu keiner Vergrößerung von Δt , da der verwendete FPGA über invertierbare Flip-Flop-Eingänge verfügt. Diese erzeugen keine bemerkbare Verzögerung. Zur Auslese müssen an die

**Abbildung 5.5:** 4-bit-Johnson-Counter.**Tabelle 5.3:** Die Sequenz eines 4-bit-Johnson-Counters.

Dezimal	Johnson-Code
0	0000
1	1000
2	1100
3	1110
4	1111
5	0111
6	0011
7	0001
0	0000

Ausgänge der Flip-Flops Auslese-Flip-Flops angeschlossen werden. Die Effizienz ist auch größer als beim Schieberegister, da mit n Flip-Flops von 0 bis $2n - 1$ gezählt werden kann.

5.1.7 Das FIFO

Im Rahmen dieses einführenden Kapitels über die Grundlagen der Elektronik soll nun das FIFO [24] erklärt werden. FIFO steht für „First In — First

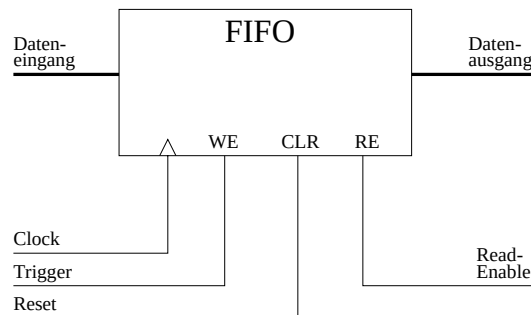


Abbildung 5.6: Ein synchrones FIFO.

Out“-Speicher (Abbildung 5.6). Es handelt sich hierbei um ein Element, das die Daten, die am Dateneingang zuerst eingespeist werden, am Ausgang auch wieder zuerst ausgibt. Auf den ersten Blick scheint es so, als würde ein Schieberegister genau dieselbe Funktion haben. Doch während ein Schieberegister mit jeder ansteigenden Taktflanke Daten speichert und Daten am Ausgang ausschreibt, speichert das FIFO nur Daten, wenn eine 1 am Write-Enable-Eingang (WE) anliegt. Ebenso gibt es seinen Speicherinhalt nur preis, wenn am Read-Enable-Eingang (RE) eine 1 anliegt. Das Schreiben und Lesen lässt sich also genau steuern.

An Stellen, an denen zeitweise mehr Daten ankommen, als verarbeitet werden können, bietet sich ein FIFO als Lösung an. Es gibt auch asynchrone FIFOs, die auf der Schreib- und Leseseite mit unterschiedlichen Clockfrequenzen betrieben werden. Mit ihnen können zwei Elektronik-Elemente verbunden werden, die auch mit anderen Frequenzen arbeiten.

5.2 Programmierbare Logik

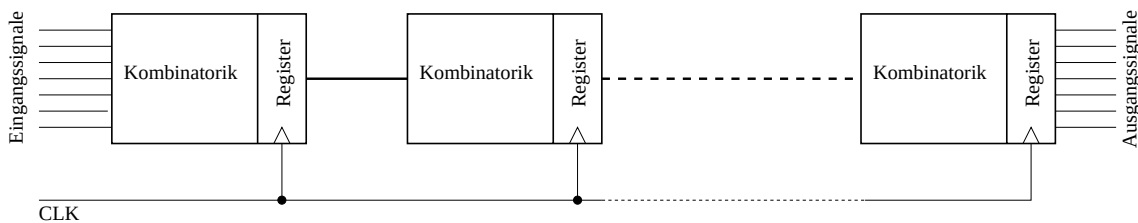
Der Zähler soll in einem programmierbaren Logikbaustein realisiert werden. Solche Bausteine haben gegenüber den gewöhnlichen ASICs (Application-Specific Integrated Circuits) einen großen Vorteil. Während man sich beim ASIC schon von Anfang an auf alle Spezifizierungen festlegen muss, hat man bei programmierbarer Logik die Möglichkeit die Funktion im Nachhinein zu ändern. Dadurch wird eine große Flexibilität erhalten. Außerdem gibt es die hohen Fixkosten nicht, die beim Entwurf eines ASICs anfallen.

5.2.1 Der FPGA — Xilinx Virtex-E XCV600E-HQ240

Wenn dargestellt werden soll, wie ein FPGA funktioniert, muss zuerst noch einmal daran erinnert werden, dass moderne Logik immer sequenziell ist (Abbildung 5.7). Sie besteht also aus Kombinatorik (zum Rechnen) und Registern (zum Speichern). Als Speicherelemente werden beim FPGA herkömmliche Flip-Flops benutzt. Die Kombinatorik wird durch andere Elemente realisiert: die Look-Up-Table (LUT).

Wie schon der Name andeutet, sind LUTs eigentlich nichts anderes als pro-

Sequenzielle Logik im Allgemeinen:



Der FPGA:

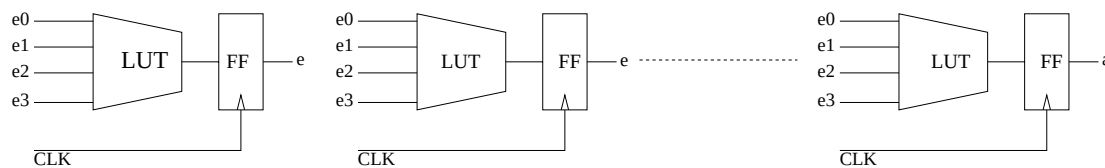


Abbildung 5.7: Realisierung einer sequenziellen Schaltung in programmierbarer Logik. Die Kombinatorik wird aus LUTs gebildet.

grammierbare Speicherelemente. Eine Look-Up-Table hat vier Eingänge und einen Ausgang. Die Eingänge können als Zeiger für einen 16×1 -bit-RAM interpretiert werden. Wenn über diesen Zeiger eine Adresse angesprochen wird, wird deren Inhalt an den Ausgang gelegt. Dieses Element kann auch als beliebige binärwertige Funktion von vier Variablen interpretiert werden. Mit einer Look-Up-Table kann somit jede Funktion von vier Signalen (beispielsweise ein

Tabelle 5.4: Wahrheitstabelle einer als UND4 konfigurierten Look-Up-Table.

e0	e1	e2	e3	a
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
⋮	⋮	⋮	⋮	⋮
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

UND4 oder ODER4) realisiert werden (Tabelle 5.4).

Die LUTs befinden sich in den Complex Logic Blocks (CLB) den Herzstücken der FPGAs. Der XCV600E verfügt über 3 456 CLBs. Sie sind in einem Feld von 48×72 Zeilen und Spalten angeordnet. Die CLBs bestehen jeweils aus zwei „Scheiben“ (*slices*). Diese Scheiben bestehen wiederum aus zwei Logic Cells (LC), dem Grundbaustein des CLBs. Jede Logic Cell hat einen Flip-Flop und eine LUT. Somit gibt es in jedem CLB vier Flip-Flops und LUTs.

Ein weiterer wichtiger Bestandteil des FPGA sind die Input-Output-Blocks (IOB). Dies sind die Bauteile im FPGA, die entscheiden, welche Signalstandards an die entsprechenden Pins angelegt werden können. Dazu wird jedem Pin ein IOB zugeordnet. Virtex-E FPGAs unterstützen verschiedene Signalstandards. Von denen insbesondere die differenziellen Standards LVPECL und LVDS hervorgehoben werden sollten. Der Virtex-E-FPGA ist der erste FPGA von Xilinx der solche Standards unterstützt. Bei diesem Projekt werden Eingangssignale für die Zähler und einige Steuersignale in LVDS übertragen. Differenzielle Signale bieten den Vorteil, dass mit ihnen auf Grund des größeren Störabstands viel höhere Frequenzen erreicht werden können. Insbesondere bei den zu zählenden Signalen spielt dies eine große Rolle.

Wesentliches Merkmal der IOB-Architektur ist die Auslegung darauf, dass die Puffer sowohl als Input, als auch als Output fungieren können.

Ausführlicher werden der FPGA und alle seine Elemente noch einmal im Anhang A erklärt. Dabei wird auch auf die Routing-Struktur und die Speichermöglichkeiten des Virtex-E näher eingegangen. Auch die Vorgänge, die zum Erstellen eines Designs nötig sind, werden dort beschrieben.

Kapitel 6

Das Zähler-Design

In diesem Kapitel wird das Hard- und Software-Design für den Zähler vorgestellt und erläutert. Dabei wird auch auf die Elektronik-Umgebung eingegangen. Der erste Abschnitt beschäftigt sich mit der Gestaltung der Zähler-Platine. Dabei wird nur bedingt auf technische Details eingegangen. Explizite Werte für Widerstände und Kondensatoren sowie andere Einzelheiten können dem Schaltplan im Anhang B entnommen werden. Im zweiten Abschnitt wird das Design für den FPGA erklärt. Dies ist der Kern der Arbeit. Schließlich beschäftigt sich der dritte Abschnitt mit dem Datenformat, in dem die Zählerstände ausgelesen werden.

6.1 Die Zähler-Platine

Alle relevanten logischen Vorgänge des Zählers werden im FPGA durchgeführt. Die Zähler-Platine (Abbildung 6.1, [31]) muss die Arbeitsumgebung für den FPGA bereitstellen. Dazu gehören in erster Line die Spannungsversorgung, die Programmiermöglichkeit, Zähl- und Steuersignale und der Datenbus zum CATCH. Es gibt auch zahlreiche Testsignale, Testpins, Lötpads und Service-Leuchtdioden. Alles in allem ist das Design aber relativ einfach. Die Platine wurde sechslagig ausgeführt. Auch optisch wird die Platine vom FPGA geprägt. Der Chip ist sehr groß (32 mm×32 mm). Er hat 240 Beinchen.

Spannungsversorgung

Am wichtigsten für den FPGA ist die Spannungsversorgung des Chips. Der Virtex-E wird auf Grund seiner 0.18- μ m-Technologie mit 1.8 V interner Versorgungsspannung V_{int} betrieben. Da das CATCH nur 3.3 und 5 V zur Verfügung stellt, muss man sich mit einem Linearregler oder DC/DC-Wandler behelfen. Auf der Zähler-Platine sind beide Alternativen vorgesehen. Zum Einen kann

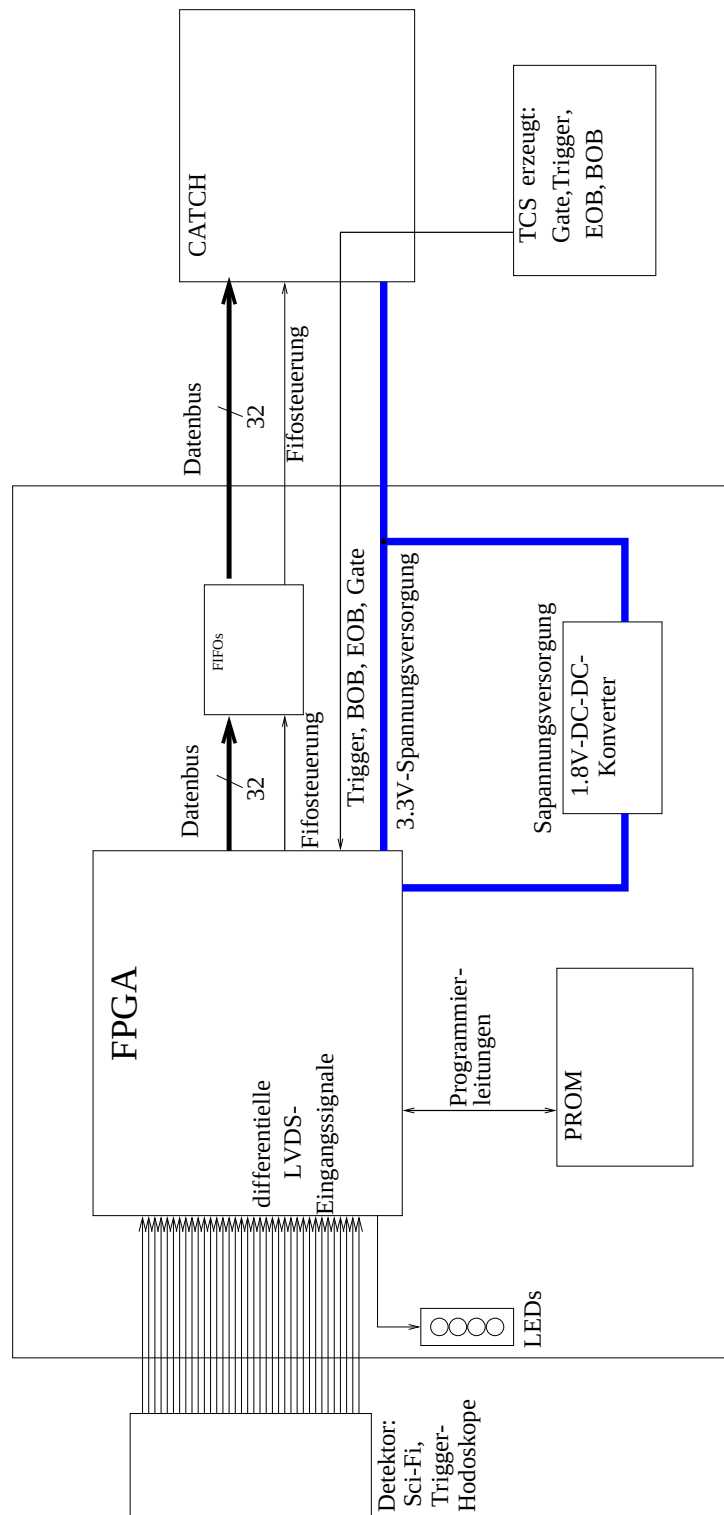


Abbildung 6.1: Das Schema für die Zähler-Platine mit allen Bauteilen.

sie mit einem DC/DC-Wandler [32] bestückt werden. Dieser Konverter ist in der Lage, eine Eingangsspannung von 5 V mit einem Wirkungsgrad größer 0.92 in eine 1.8V-Spannung umzuwandeln. Sein Nachteil liegt darin, dass er so groß ist, dass der Zähler nicht auf dem Catch-X eingesetzt werden kann, weil sich an der Stelle beim Catch-X andere Bauteile befinden. Für die Anwendung der Scaler-CMC auf dem Catch-X muss die Platine deshalb mit einem Linearregler [33] bestückt werden, der wesentlich kleiner ist. Dieser erzeugt die 1.8 V aus 3.3 V. Sein Nachteil liegt im konzeptbedingten niedrigem Wirkungsgrad von weniger als $\frac{1.8V}{3.3V} = 0.55$. Dies könnte unter Umständen einen Stromfluss erfordern, den das CATCH nicht bereitstellen kann, da bis zu vier Scaler-CMCs auf einem CATCH Platz finden. Fürs erste werden alle Zähler mit dem Linear-Regler ausgerüstet, damit sie sowohl auf dem CATCH, als auch auf dem Catch-X genutzt werden können.

Die 1.8 V reichen nicht aus, um LVTTL- und LVDS-Outputs zu betreiben. Hierzu sind 3.3 V nötig. Die 3.3 V werden vom CATCH direkt bereitgestellt. Theoretisch bestünde die Möglichkeit, auch andere Spannungen an die FPGA-IOs zu legen. Die Outputpads sind zu diesem Zweck in 8 Bänke unterteilt, die mit unterschiedlichen V_{cc} betrieben werden können. Somit können andere Signalarten unterstützt werden. Beim Zähler-Design ist eine Unterteilung jedoch nicht notwendig.

Für die Spannungsverteilung werden zwei der sechs Ebenen der Platine reserviert. Es gibt eine Masse-Ebene und eine gemeinsame Ebene für 3.3 und 1.8 V. Weitere wichtige Bauteile im Spannungsnetzwerk sind die Abblockkondensatoren. Diese werden gebraucht, da der Stromfluss am FPGA nicht kontinuierlich ist, sondern im Moment einer ansteigenden Clock-Flanke Spitzen hat. Dies liegt daran, dass in dem Moment viele Flip-Flops im Chip schalten. Die Abblockkondensatoren (Abbildung 6.2) stellen kurzzeitig die nötige Ladung zur Verfügung.

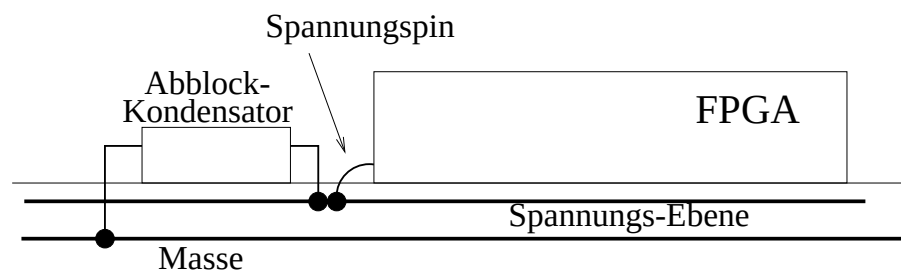


Abbildung 6.2: Ein Abblockkondensator.

Die Zählsignale

Im Folgenden wird das Platinen-Design entlang des Datenstroms erläutert: Die LVDS-Sigale von den Detektoren gelangen auf Twisted-Pair-Kabeln zur Scaler-CMC. Die Kabel können über Robinson-Nugent-Stecker mit der Platine verbunden werden. Auf der Platine werden die Leitungen durch ein Widerstandsnetzwerk (Terminierungsnetzwerk, Abbildung 6.3) zum FPGA geführt. Der 100- Ω -Widerstand zwischen den Signalleitungen gehört zum LVDS-Standard. Die Widerstände verhindern, dass die Signale reflektiert bzw. zu stark gedämpft werden, nebenbei sorgen sie für eine definiertes Potenzial zwischen den Eingangsstufen für den Fall, dass kein Kabel aufgesteckt ist.

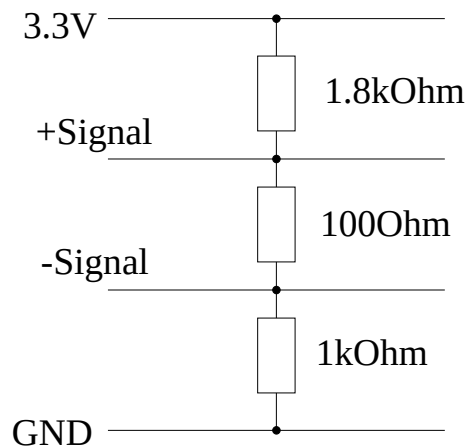


Abbildung 6.3: LVDS-Terminierung.

Um Störungen zu vermeiden, ist es wichtig, dass die +Leitungen und die –Leitungen jeweils möglichst parallel an die Pins herangeführt werden. Am FPGA kann nicht jedes beliebige Pin-Paar als Eingang benutzt werden, sondern es müssen spezielle Paare benutzt werden, die aus dem Datenblatt [44] entnommen werden können. Auch die Polarität muss hier beachtet werden.

Steuer- und Clock-Signale

Die Zählsignale sind nicht die einzigen Eingangssignale am FPGA. Es gibt eine Vielzahl von Steuersignalen. Diese kommen vom CATCH über den CMC-Stecker zum Zähler. Die wichtigsten Steuersignale sind die TCS-Signale (Trigger, BOB, EOB, Gate) die entsprechenden Signale auf der Platine heißen USR0 - USR3 und START/STARTN. Zum Initialisieren der Zähler reichen serielle Leitungen mit geringer Datentransferrate. Das Signal SND0 ist zu diesem Zweck vorgesehen.

Ganz besonders große Aufmerksamkeit muss bei sequenzieller Logik den Clock-Signalen gewidmet werden. Wenn es keine stabile Clock gibt, arbeitet das Bauteil nicht richtig. Vom CATCH kommen zwei verschiedenen Clock-Signale auf die Scaler-CMC: DTCLK und TCLK. Die DTCLK wird vom TCS verteilt. Sie ist ein differenzielles Clock-Signal mit 38.88 MHz. Im Experiment spielt dieses Signal eine große Rolle, denn das Trigger-Signal ist synchron zu dem von der DTCLK erzeugten Zeitraster.

Das TCLK-Signal wird auf dem CATCH erzeugt.

FPGA-Programmierung und JTAG-Netzwerk

Xilinx FPGAs unterstützen Boundary-Scan [34]. Dies ist ein Testverfahren zum automatischen Prüfen der Verbindungen zwischen zwei Bauteilen. Dieser Test kann auf dem Zähler-Board nicht durchgeführt werden, da nur Verbindungen zwischen zwei Boundary-Scan-fähigen Bauteilen getestet werden können. Und der FPGA ist das einzige auf der Scaler-CMC. Dennoch kann das zugehörige JTAG-Netzwerk (Abbildung 6.4) genutzt werden, denn ein Programmieren des FPGAs und des PROMs ist über diese Leitungen möglich. Die Leitungen hierzu kommen vom CATCH, aber auch über die Testpins kann auf den FPGA zugegriffen werden.

Im Experiment wird es später so sein, dass das PROM vorab über die JTAG-Leitungen programmiert wird. Beim Anlegen von Spannung kommunizieren FPGA und PROM, und das PROM sendet dem FPGA dann seinen Speicherinhalt, das bit-File [44].

Das JTAG-Netzwerk kann mit Jumpers konfiguriert werden. Die genauen Konfigurationsmöglichkeiten können dem Schaltplan im Anhang B entnommen werden.

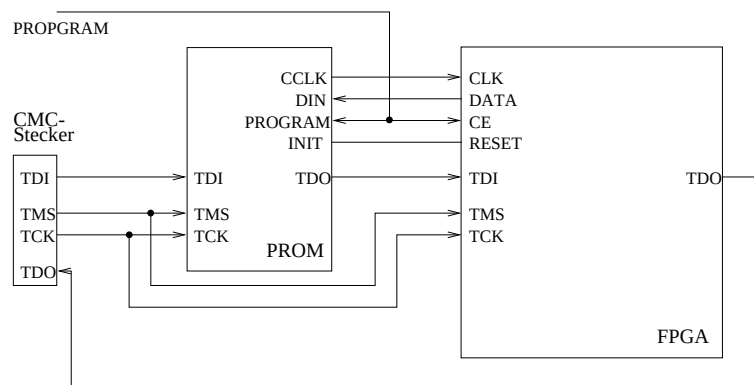


Abbildung 6.4: Die JTAG-Umgebung (TDO, TDI, TMS, TCK) und die Programmierleitungen zwischen PROM und FPGA.

Die Testbuchsen

Die Testbuchsen sind in zwei Felder mit jeweils 18 Buchsen unterteilt. Die Felder bestehen wiederum aus zwei Reihen. Die Pins haben im wesentlichen zwei Aufgaben. Zum Einen sollen sie eine schnelle und unkomplizierte Programmiermethode für Testzwecke bereitstellen, und zum Anderen soll die Möglichkeit bestehen, an den Buchsen einige Signale vom FPGA abzugreifen. Die Testbuchsen bieten für diese Aufgaben maximale Flexibilität. So werden alle möglichen Programmiermodi unterstützt, und man kann sich nach dem Programmieren 18 Testsignale des FPGAs ansehen.

Die Frontplatte und die LEDs

Die Frontplatte ist, bis auf die Aufschrift „Scaler-CMC“, identisch mit den TDC-CMC-Frontplatten.

Oberhalb der Stecker befinden sich die LEDs. Die LEDs sind mit Buchstaben bezeichnet. Die einzelnen LEDs haben dabei folgende Bedeutung (Tabelle 6.1):

Tabelle 6.1: Die Zähler-LEDs und ihre Funktion.

LED	Zustand	Funktion
F	ein	das Gate ist 0
	aus	das Gate ist 1
T	ein	es kommen gerade Trigger
	aus	es kommen gerade keine Trigger
L	ein	initialisiert
	aus	nicht initialisiert
	blinkend	Initialisierung fehlgeschlagen
I	ein	Input-Register
	aus	Zähler

Identifikationsdioden

Die Dioden sind, abgesehen von Widerständen und Kondensatoren, die einzigen Bauteile auf der Rückseite (Lötseite). Die Dioden dienen zur automatischen Identifikation der CMC. Mittels des Device-ID-Signals (DEVID, Anhang B) kann vom CATCH abgefragt werden, um was für eine CMC es sich handelt. Über die Datenleitungen wird ein 16-bit-Wort übertragen. Bit 0–9

enthält die Seriennummer, Bit 10–12 die Reversionsnummer¹ und Bit 13–15 eine vier (100), die Nummer für die Scaler-CMC [19].

6.2 Das Design für den FPGA

Das Programm lässt sich in zwei funktionelle Teile einteilen. Zum einen gibt es den Zählmechanismus, bestehend aus Vorzähler, Gesamt-Zähler und Verzögerungseinheit. Diese Einheit gibt es 32 mal, je einmal pro Kanal. Zum anderen gibt es den Teil der Schaltung, der für das Auslesen der 32 Zählerstände und das serielle Herausschreiben in die externen FIFOs zuständig ist. Dieser zweite Teil besteht aus einer Steuerung und einem Multiplexer.

Die beiden Teile werden durch ein FIFO verbunden.

6.2.1 Der Johnson-Counter und dessen Auslese

Der geschwindigkeitsbestimmende Teil des Zählers ist der Johnson-Counter. Die Vorteile dieses Designansatzes wurden schon im Grundlagenkapitel dargelegt. Mit einem Johnson-Counter können maximale Geschwindigkeiten erreicht werden, er läuft am stabilsten und die Auslese ist am unproblematischsten. Die Auslese funktioniert folgendermaßen: Zu jedem der vier Flip-Flops des Johnson-Counters gibt es einen Auslese-Flip-Flop. Diese Flip-Flops sind dazu da, die Werte der Flip-Flops des Johnson-Counters im Moment der Auslese zu speichern. Wenn aber hier von Auslese die Rede ist, ist damit nicht die Auslese des kompletten Zählers gemeint. Denn die soll ja nur mit jedem Trigger erfolgen. Die Trigger kommen in Abständen der Größenordnung Millisekunde. In dieser Zeit wäre der Johnson-Counter schon einige hundert mal durchlaufen, und aus den Werten könnten keine Rückschlüsse mehr auf den Zählerstand gezogen werden. Stattdessen wird zur Auslese des Johnson-Counters ein künstliches Signal verwendet, die 38.88-MHz-Clock vom TCS (Abbildung 6.6). Das ist äquivalent zu einer Auslese alle 25.73 ns. Der 4-bit-Johnson-Counter braucht 8 Signale für einen Umlauf. Die maximale Frequenz, die vom 4-bit-Johnson-Counter gezählt werden kann, ist also $\nu = \frac{1}{T} < \frac{8}{25.73 \text{ ns}} < 310.9 \text{ MHz}$. Das reicht in jedem Falle, zumal ja immer statistisch verteilte Signale gezählt werden sollen. D.h., bei einem Detektor mit einer Totzeit von 5 ns und einer maximalen Frequenz von 200 MHz ist nicht zu erwarten, dass innerhalb von 50 ns wirklich zehn Ereignisse auftreten. Es sei denn, das zu zählende physikalische Signal hat eine viel höhere Frequenz. Doch dann ist der ganze Aufbau auch nicht mehr sinnvoll, da man ja nicht an der Zahl der vom Detektor erzeugten Pulse, sondern an der Zahl physikalischer Ereignisse interessiert ist.

¹Die Revisionsnummer gibt über mögliche Modifikationen gegenüber der ursprünglichen Version Auskunft.

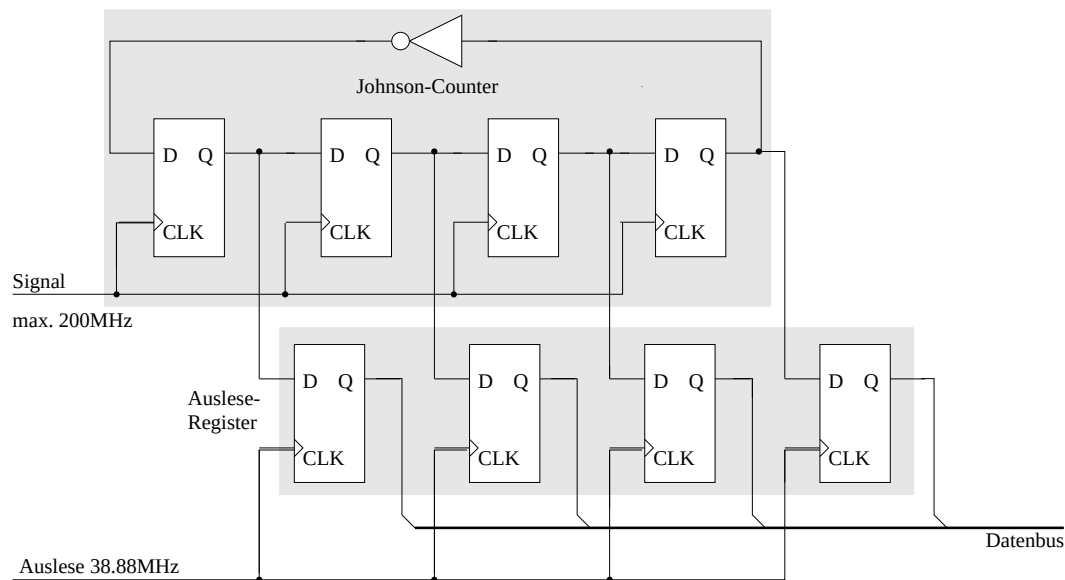


Abbildung 6.5: Die Flip-Flops des Johnson-Counters und deren Auslese.

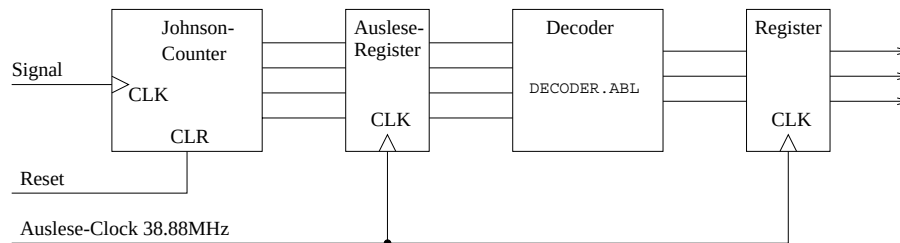


Abbildung 6.6: Der Johnson-Counter und dessen Auslese (schematisch).

6.2.2 Der Dekoder

Damit richtig gezählt werden kann, müssen die Werte des Johnson-Counters dekodiert werden. Der dafür notwendige Dekoder ist in der Abbildung 6.5 schon eingezeichnet. Ein Dekoder wird am besten durch eine Wahrheitstabelle einer HDL (in diesem Fall ABEL) beschrieben. In einer Wahrheitstabelle wird jedem möglichen Wert am Eingang der entsprechende Wert am Ausgang zugeordnet. Für den Dekoder des Johnson-Counters ist sie in Tabelle 6.2 dargestellt.

Der ABEL-Code der dieser Wahrheitstabelle entspricht findet sich im Beispiel-Listing in Abbildung A.8 auf Seite 86.

Tabelle 6.2: Die Sequenz eines 4-bit-Johnson-Counters und seine Dekodierung.

Johnson-Code	dekodierter Wert
0000	000
1000	001
1100	010
1110	011
1111	100
0111	101
0011	110
0001	111
0000	000

6.2.3 Das programmierbare Schieberegister

In Abschnitt 4.1 wurde beschrieben, dass der Zähler in der Lage sein soll, die Trigger-Verzögerung auszugleichen. Im FPGA-Design wird diese Funktion durch eine Verzögerung der Vorzähler-Werte verwirklicht. Verzögerungen können mit einem Schieberegister herbeigeführt werden. Das Schieberegister wurde schon im Grundlagen-Kapitel in Abschnitt 5.1.5 beschrieben. Um synchron zu den übrigen Zähler-Elementen zu sein, wird das Schieberegister [35] mit der Auslese-Clock (38.88 MHz) betrieben (Abbildung 6.7). Das bedeutet, dass die Trigger-Verzögerung auf einem Zeitraster mit einer Schrittweite von 25.7 ns ausgeglichen werden kann. Dies genügt unseren Anforderungen. Natürlich könnte auch der Gesamtzählerstand am Ende durch ein Schieberegister verzögert werden, doch dies hätte einen großen Nachteil: Es würde viel mehr Speicherplatz benötigen, wenn 31 bit durch jede Registerstufe geschoben werden müsste. Bei einer Schrittweite von ca. 25.7 ns werden für eine Mikrosekunde etwa 40 Schieberegisterschritte benötigt. Bei 31 bit Registerbreite wären das 1 240 Speicherplätze bzw. Flip-Flops. Bei einer Verzögerung der Vorzähler-Auslese-Werte werden für dieselbe Verzögerung 120 Speicherplätze gebraucht. Aus dem gleichen Grund ist auch der Dekoder vor dem Schieberegister und nicht dahinter.

Die Trigger-Verzögerung ist keine feste Größe, sondern sie kann sich im Laufe des Experiments verändern. Man könnte bei jeder Änderung dieser Größe eine Änderung des FPGA-Programmes vornehmen, doch das ist keine komfortable Lösung. Besser ist ein programmierbares Schieberegister, bei dem die Verzögerungszeit immer wieder unkompliziert eingestellt werden kann. Bei der Nutzung eines solchen Registers muss eine Möglichkeit gefunden werden, dieses zu initialisieren. Das Register soll Verzögerungen von bis zu 3 μ s erreichen. Dazu werden 120 Stufen gebraucht. Wenn die Verzögerung auf 25.7 ns genau eingestellt werden soll, wird ein 7 bit Steuersignal benötigt. Dieses Signal wird

dem Chip nicht parallel übertragen, da nicht genügend Signalleitungen zur Verfügung stehen. Notwendig ist daher ein Wandler, der ein seriellles Signal in ein paralleles umwandelt. Dieser wird in Abschnitt 6.2.8 beschrieben.

Ein großes Problem des Schieberegisters ist, dass es nicht zurücksetzbar ist. Der Grund hierfür liegt darin, dass es auf dem FPGA mit LUTs realisiert ist. Das bringt sowohl beim Reset des Zählers, als auch beim Umprogrammieren Probleme mit sich.

Beim Zurücksetzen des Zählers werden alle Flip-Flops in ihren Grundzustand gesetzt. Dies geschieht innerhalb von Nanosekunden. Bei den Schieberegistern jedoch wurde die RAM-Funktion der LUTs [44] ausgenutzt, um Flip-Flops zu sparen. Bei einer LUT kann aber pro Taktflanke nur einen Wert hineingeschrieben werden. Hierzu wäre auch eine komplizierte Logik notwendig. Aus diesen Gründen wurde beim Schieberegister auf eine Reset-Möglichkeit verzichtet. Dies bedeutet, dass beim Reset die Werte im Schieberegister nicht gelöscht werden können. Nach dem Zurücksetzen des Zählers würde für eine Zeit ΔT_{SR} , die der Dauer der Trigger-Verzögerung entspricht, ein undefinierter Datenstrom aus den Schieberegistern kommen. Falsche Zählerstände wären das Resultat. Dieses Problem wird umgangen, indem in einem Zeitraum ΔT_{SR} Nullen in das Schieberegister geschrieben werden. Dazu muss das Reset-Signal so lange gehalten werden. Da die maximale Triggervverzögerung $3\mu\text{s}$ beträgt, werden alle Reset-Signale auf $3\mu\text{s}$ verlängert. Die Zählerstände bleiben hiervon unbeeinträchtigt, da kein Reset-Signal weniger als 1 ms vor dem Beginn des Spills kommen darf.

Dasselbe Problem tritt beim Umprogrammieren des Schieberegisters auf, es wird auf die gleiche Art gelöst.

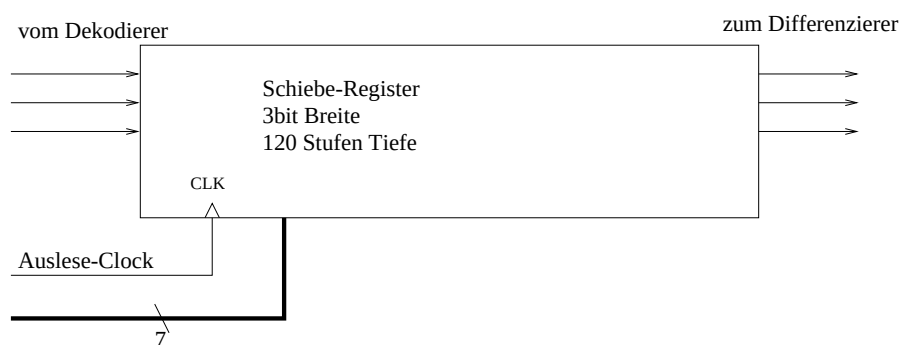


Abbildung 6.7: Das 3 bit-Schieberegister zum Ausgleich der Trigger-Verzögerung bis maximal $3\mu\text{s}$.

6.2.4 Der Subtrahierer und der 31-bit-Addierer

Bisher wurde von der Auslese des Vorzählers und dem Ausgleich der Trigger-Verzögerung berichtet. Was fehlt, ist der eigentliche 31-bit-Zählmechanismus. An das Schieberegister schließen sich noch einmal zwei Registerstufen an. Es

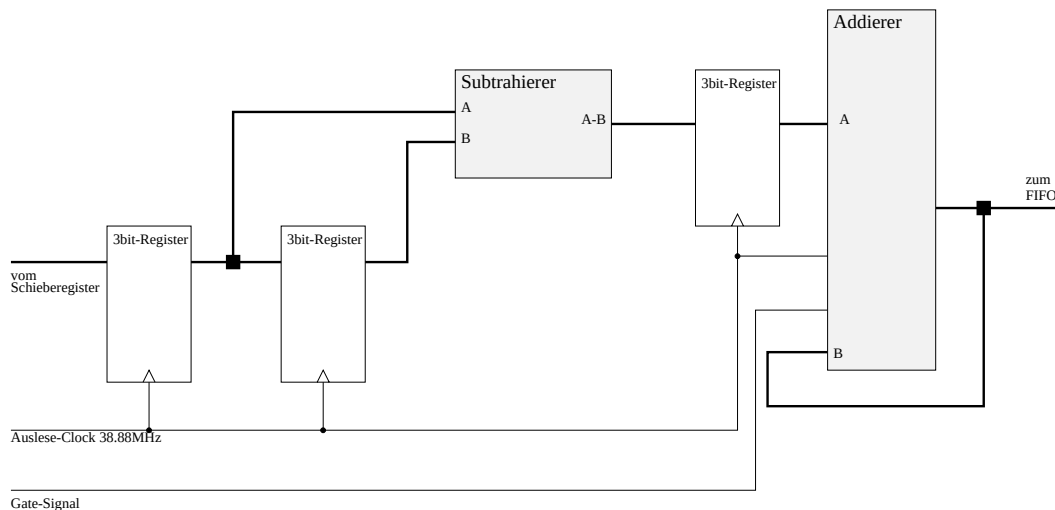


Abbildung 6.8: Aus einem Subtrahierer und einem Addierer wird ein Zähler gebaut.

muss beachtet werden, dass im zweiten Register immer ein Wert steht, der 25.7ns älter ist als der Wert im ersten Register. Nun soll die Differenz aus den beiden Werten gebildet werden. Dazu werden die Ausgänge der beiden Register an die Eingänge eines Subtrahierers gelegt. Das erste Register geht an Eingang A, das zweite Register geht an Eingang B. Am Ausgang des Subtrahierers liegt die Differenz $A - B$ an (Abbildung 6.8). Das ist genau die Anzahl der Ereignisse, die zwischen beiden Auslesen des Johnson-Counters lag. Und zwar unabhängig davon, ob der Johnson-Counter zwischen den beiden ausgelesenen Werten wieder auf 0 gesprungen ist oder nicht. Das ist vielleicht im ersten Moment überraschend. Wenn beim ersten mal 7 ausgelesen wurde und beim zweiten mal 3, würde man bei der Subtraktion $3 - 7$ das Ergebnis -4 erwarten. Subtrahierer ohne Vorzeichenbit bekommen aber automatisch das in diesem Fall richtige Ergebnis 4 heraus. Dies liegt daran, dass ein elektronischer Subtrahierer nicht wirklich subtrahiert, sondern das Komplement+1 addiert. Hinter dem Subtrahierer befindet sich wieder ein Register, das die Differenz abspeichert. In diesem Register steht immer ein Wert, der die Anzahl der gezählten Ereignisse in den letzten 25.7ns angibt. Diese Werte müssen aufaddiert werden, und man hat einen Zähler.

Der eigentliche Zähler besteht also aus einem 31-bit-Addierer mit zwei Eingängen A, B. An Eingang A liegt das Ergebnis des Subtrahierers an, und an

Eingang B liegt der Ausgang des Addierers an. So wird bei jeder Clock-Flanke dem aktuellen Zählerwert die aktuelle Johnson-Counter-Differenz hinzugefügt.

Die Gate-Funktion

Auf den ersten Blick scheint der Mechanismus unnötig kompliziert zu sein. Schließlich wird erst subtrahiert, nur damit später wieder addiert werden muss. Der Grund für dieses Vorgehen ist die Gate-Funktion. Die lässt sich jetzt sehr leicht implementieren. Dem Addierer muss ein CE-Eingang hinzugefügt werden. An diesen Eingang wird das Gate-Signal gelegt. Nur wenn das Gate-Signal oben ist, wird die Clock-Flanke der Auslese-Clock am Addierer gesehen, und dann wird die aktuelle Differenz addiert.

Bei der Gate-Funktion muss wiederum das Zeitverhalten beachtet werden. Denn die Zählsignale sind um etwa eine Mikrosekunde verzögert. Das Gate-Signal muss also auch verzögert werden. Die Verzögerungszeit für das Gate-Signal kann aber von der für die Zählerstände abweichen, da die Erzeugung dieses Signals auch eine endliche Zeit braucht.

Man könnte auf den Gedanken kommen, dass die Gate-Funktion durch ein CE-Signal an den Flip-Flops des Johnson-Counters implementiert werden könnte. Doch dies würde bedeuten, dass pro Flip-Flop eine weitere Leitung gebraucht wird. Dies würde das Routing der Johnson-Counter erschweren, und gerade dieser Teil des Zählers ist der, der im Endeffekt die maximale Geschwindigkeit bestimmt.

6.2.5 Die Auslese

Der Addierer erzeugt also ca. alle 25.7ns einen neuen Zählerstand. Dieser Zählerstand soll nicht jedes Mal ausgelesen und an das CATCH übermittelt werden. Dies soll nur mit jedem Trigger geschehen.

Die Auslese bei jedem Trigger wird so realisiert, dass mit jedem Trigger der aktuelle Wert des Addierers in ein FIFO (Abbildung 6.9, [36]) geschrieben wird. Das FIFO hinter jedem Zähler ist zum Zwischenspeichern notwendig, da alle 32 Zähler gleichzeitig ausgelesen werden und nacheinander in die externen FIFOs² [37] zum CATCH geschrieben werden müssen.

Als Schreib- und Lese-Clock wird beim FIFO wieder die übliche 38.88-MHz-Clock verwendet. Das Trigger-Signal wird auf das Write-Enable-Signal des FIFOs gegeben. Damit es zu keinen Fehlern kommen kann, muss gewährleistet sein, dass das Timing zwischen der Clock und dem Trigger-Signal stimmt. Das Trigger-Signal muss zeitlich so liegen, dass es immer gleichzeitig mit einer ansteigenden Clock-Flanke kommt. Dies ist aber kein Problem, denn das

²Cypress CY7C4225V-15ASC

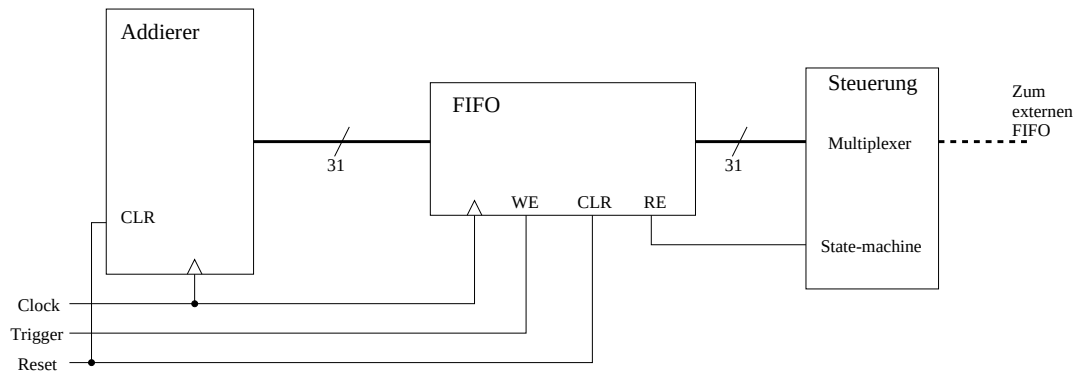


Abbildung 6.9: Das Auslese-FIFO.

Trigger-Signal wird vom TCS so erzeugt, dass es genau 25.7 ns lang und genau synchron zur 38.88-MHz-Clock ist (Abbildung 6.10).

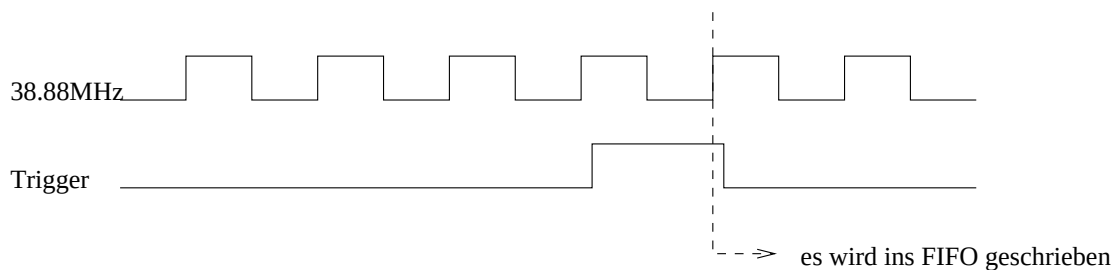


Abbildung 6.10: Das Trigger-Signal ist synchron zur TCS-Clock (38.88 MHz).

6.2.6 Zusammenfassung des Zählmechanismus

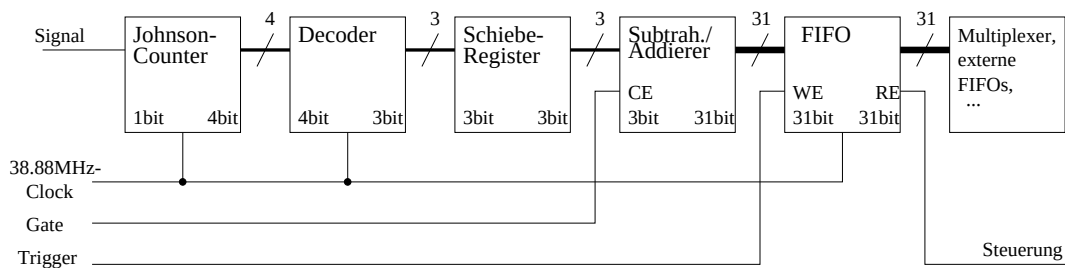


Abbildung 6.11: Der schematische Schaltplan eines Zählers auf einen Blick.

Wenn man den ganzen Zählmechanismus (Abbildung 6.11) in wenigen Sätzen zusammenfassen möchte, muss gesagt werden, dass der Zähler aus einem Addierer besteht, der alle 25.7 ns die Differenzen der Zwischenstände eines

schnellen Vorzählers addiert. Die Gatefunktion wird durch ein Clock-Enable am Addierer realisiert. Und die Auslese wird durch das Trigger-Signal am FIFO-Write-Enable möglich gemacht.

6.2.7 Die Auslese der Zähler-FIFOs

Wie die Daten in die 32 FIFOs hinter den Zählerkanälen kommen, wurde ausführlich beschrieben. In den folgenden Abschnitten wird geschildert, was danach mit den Daten passiert. Der Zweck der FIFOs ist es, einen Puffer zu schaffen, in dem alle Daten abgelegt werden, bis sie vom Auslese-Mechanismus in das externe FIFO geschrieben werden können. Zu diesem Zweck wird ein Multiplexer mit einer Steuerung (State-Machine) für diesen Multiplexer, sowie die 32 internen und das externe FIFO gebraucht.

Der Multiplexer und seine Steuerung

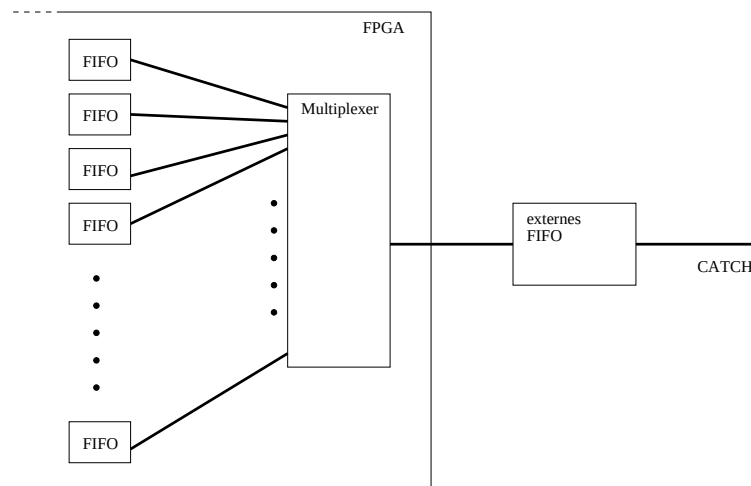


Abbildung 6.12: Durch den Multiplexer können die Daten nacheinander von den Zähler-FIFOs in das externe FIFO geschrieben werden.

Diese Steuerung muss folgendermaßen aussehen: Zuerst muss vom FPGA ein Start-Wort (Header) (siehe Abschnitt 7.2) erzeugt und ins externe FIFO geschrieben werden. Dann müssen die 32 Datenworte ins FIFO. Schließlich muss ein Abschluss-Wort (Trailer) erzeugt werden (Abbildung 6.13). Die Header und Trailer können mit Hilfe eines Trigger-Zählers generiert werden.

Für das Herausschreiben der Worte aus den internen FIFOs reicht es, am Anfang des Auslese-Prozesses einmal ein Read-Enable an die FIFOs zu schicken. Das Schreiben in das externe FIFO ist genauso einfach. Es muss für genau

34 Clock-Zyklen (Trailer + 32 Datenworte + Header) ein Read-Enable an das externe FIFO gelegt werden.

Komplizierter ist es, den Multiplexer so anzusteuern, dass die Worte in der richtigen Reihenfolge und zum richtigen Zeitpunkt das externe FIFO erreichen. Aber auch dies ist ein lösbares Standardproblem der Elektronik.

Die State-Machine, die diese Aufgaben erfüllt, wird übrigens nicht vom Trigger-Signal gestartet, sondern sie orientiert sich daran, ob Daten in den internen FIFOs sind. Solange diese nicht leer sind, schreibt sie ein Ereignis nach dem anderen in das externe FIFO.

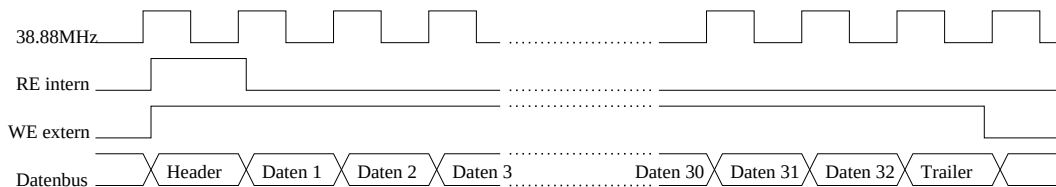


Abbildung 6.13: Die Daten werden seriell von den internen in die externen FIFOs geschrieben.

6.2.8 Die serielle Programmierung

Der Ausgleich der Trigger- und Gate-Verzögerung ist eine wichtige Funktion des Zählers. Sie wird durch programmierbare Schieberegister realisiert (Abschnitt 6.2.3). Damit das Register programmiert werden kann, muss es einen Initialisierungsvorgang geben, bei dem Daten vom Benutzer dem Zähler mitgeteilt werden können. Dies geschieht über die serielle Datenleitung SND0. Die Zähler-CMC ist nicht die einzige Karte im COMPASS-Experiment, die initialisiert werden muss. Das Verfahren ist schon für die Front-End-Karten erprobt [38]. Dabei sendet das CATCH mit der Frequenz $\nu_{\text{SND0}} = \frac{38.88\text{MHz}}{4} = 9.72\text{ MHz}$ ein Datenwort à 24 bit, das mit zwei Startbits (10) eingeleitet wird und von derselben Folge abgeschlossen wird (Abbildung 6.14). In diesem bewährten Datenformat wird auch beim Zähler die Information übertragen.

Damit die Information zur Programmierung des Schieberegisters benutzt werden kann, müssen aus den seriellen Daten durch einen entsprechenden Wandler parallele Daten gemacht werden. Diese werden in ein Register geschrieben und an die entsprechenden Elemente geschickt.

Interpretiert werden die Daten folgendermaßen (Abbildung 6.15): Das erste Bit enthält die Information, ob die Scaler-CMC als Zähler (0) oder als Input-Register (1, Abschnitt 6.4) arbeiten soll. Die nächsten 7 bit enthalten die Länge des Schieberegisters für die Datenverzögerung. Die 6 bit danach geben die Länge der Gateverzögerung an. Dabei muss man den Zahlenwert mit 25.7 ns multiplizieren, um die Trigger-Verzögerung zu erhalten. Mehr Daten

sind. Wenn nicht genauso lange resettet wird, wie die Schieberegister die Daten verzögern, kommen nach dem Reset Daten aus dem Register und werden aufsummiert. Daher empfiehlt es sich, die Resetdauer länger als die maximale Länge des Schieberegisters zu wählen, also $>3\ \mu\text{s}$. Dies ist jedoch kein Problem, da das BOB-Signal auf jeden Fall mehr als 1 ms vor dem ersten Trigger-Signal des Spills kommt.

6.3 Die Datenstruktur

Um den Anforderungen der COMPASS-Auslese-Elektronik zu genügen, reicht es nicht aus, dass der Zähler mit jedem Trigger die 32 Zählerstände ausgibt. Damit am Ende die Ereignisse wieder zusammengesetzt werden können, müssen die Daten in einer festen Struktur übermittelt werden. Diese Struktur (Abbildung 6.16) ist für den Zähler unkompliziert. Pro Trigger ist ein Header und ein Trailer erforderlich.

In dem Moment, in dem Header und Trailer neben den Datenworten eingeführt werden, wird es erforderlich, dass das CATCH diese auch unterscheiden kann. Deshalb wird das führende Bit für die Identifikation von Header/Trailer bzw. Daten reserviert. Wenn das MSB gleich 0 ist, handelt es sich um einen Header bzw. Trailer, und wenn es gleich 1 ist um Daten. Bei den Datenworten stehen somit die restlichen 31 bit für die Zählerstände zur Verfügung. Bei Worten mit einer 0 im MSB ist es notwendig, zwischen Header und Trailer zu differenzieren. Um Header handelt es sich, wenn das zweite Bit gleich 0 ist. Den Trailern wird dort eine 1 zugewiesen. Damit das Zusammenfügen von Ereignissen möglich ist, müssen weitere Informationen in die Header und Trailer: die Ereignis-Nummer und die geographische ID.

Aus diesem Grund werden im Zähler die Trigger-Signale mitgezählt. Der Zählerstand gibt die aktuelle Ereignis-Nummer an. Diese Ereignisnummer (20 bit) wird mit in die Header und Trailer geschrieben.

Die geographische ID gibt an, von welcher Karte die Daten kommen. Da die Scaler-CMC die eigene ID nicht auslesen kann, muss die Information auf dem CATCH eingetragen werden. Das CATCH ist in der Lage die Identifikationsdioden auf der Scaler-CMC auszulesen.

6.4 Weiter Anwendungsmöglichkeiten für die Zähler-CMC

Die Scaler-CMC kann nicht nur zum Zählen von Detektorsignalen eingesetzt werden. Mit ihren 32 Eingangs-Signalen bietet sie eine breite Basis für eine Vielzahl von Anwendungen. Eine Möglichkeit wäre der Einsatz als Input-

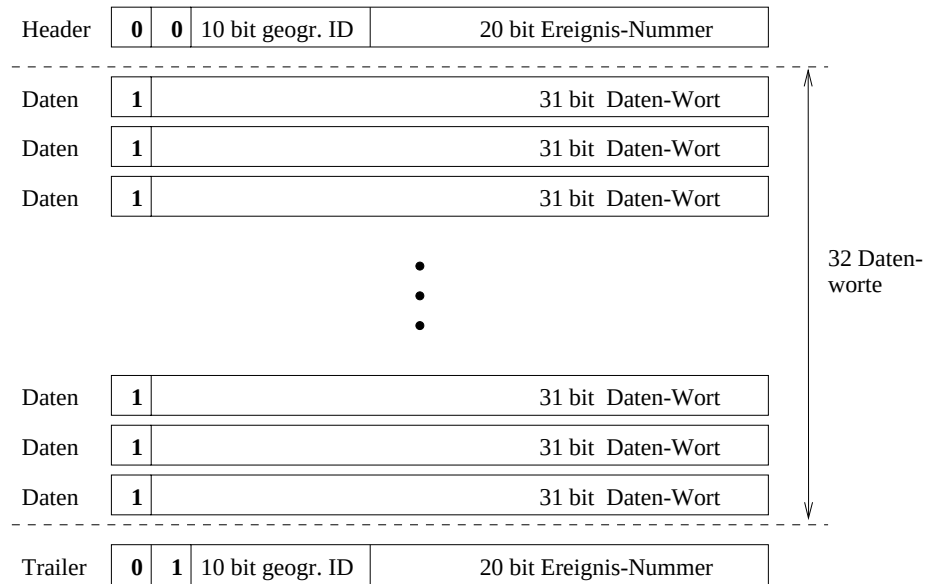


Abbildung 6.16: Die Datenstruktur der Zähler-Daten.

Register.

Input-Register

Ein Input-Register (Abbildung 6.17) speichert mit jedem Trigger eine Anzahl von Signalen. Es besteht somit lediglich aus einem Register, an dem die Eingangssignale anliegen. Ein solches Modul kann z.B. dazu genutzt wer-

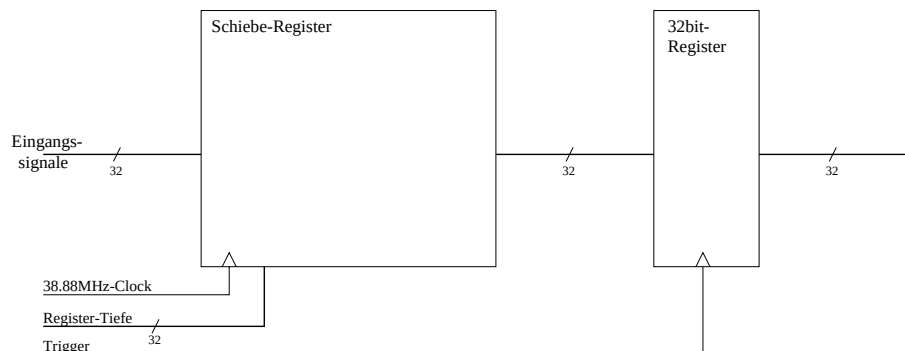


Abbildung 6.17: Das Input-Register.

den, Trigger-Masken abzuspeichern. Mit Trigger-Masken können Trigger klassifiziert werden. Bei einem Hochenergie-Experiment wird grundsätzlich auf mehrere Klassen von Ereignissen getriggert. Mit elektronischen Schaltungen

können diese Klassen aus den Detektorsignalen extrahiert werden. Die Scaler-CMC ist in der Lage, diese Signale mit jedem Trigger abzuspeichern und der Auslese-Elektronik (CATCH) zuzuführen.

Auch beim Input-Register muss die Triggervverzögerung berücksichtigt werden. Hierzu wird wie beim Zähler ein Schieberegister eingesetzt, das die Daten verzögert.

Die Länge des Schieberegisters wird auch wieder über die serielle Schnittstelle initialisiert. Das Input-Register für die Scaler-CMC beinhaltet eine weitere Funktion. Die aufsteigenden Clock-Flanken der 38.88-MHz-Clock können gezählt werden und werden ebenfalls mit jedem Trigger ausgelesen. Diese Funktion ist z.B. bei der Erstellung eines zeitlichen Strahlprofils von Nutzen. Da das Design für ein Input-Register sehr einfach ist und sehr wenig Platz beansprucht, wird es gleichzeitig mit dem Zählerdesign in den FPGA programmiert. Es wird dann ebenfalls mit den seriellen Initialisierungsdaten entschieden, ob die Scaler-CMC als Zähler oder als Input-Register arbeiten soll.

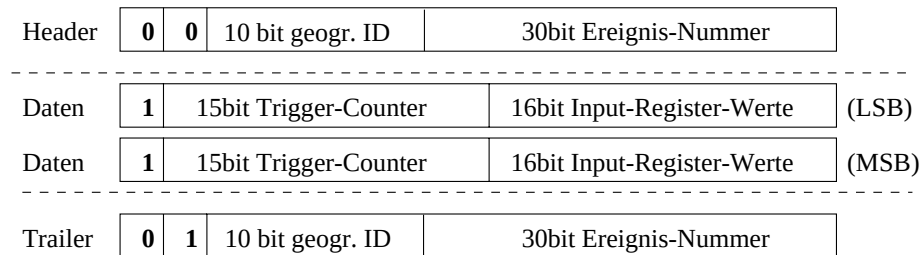


Abbildung 6.18: Die Datenstruktur für das Input-Register. Im ersten Datenwort stecken die 15 LSBs vom Clock-Zähler und die 16 LSBs vom Input-Register.

Bei Verwendung des Input-Registers werden sowohl die 32 bit Information der Eingangssignale, als auch die 30 bit Information über die Anzahl der Triggerflanken auf zwei Datenworte aufgeteilt. Die Header und Trailer sind mit denen im Zählermodus identisch (Abbildung 6.18).

Weitere Anwendungen

Die Scaler-CMC ist auf Grund ihrer variablen Auslegung auch für weitere Anwendungen geeignet. Es stehen 32 Eingabesignale und ein Datenbus zum CATCH zur Verfügung. Der FPGA kann beliebig den jeweiligen Anforderungen angepasst werden. Es könnte z.B. auch eine komplette Triggerlogik in eine Scaler-CMC implementiert werden. Somit stellt die Scaler-CMC ein wichtiges zusätzliches Element für das CATCH-Ausleseprinzip dar, durch das das gesamte System sehr an Flexibilität gewinnt.

Kapitel 7

Die Testphase

7.1 Der Verbindungstest

Wenn eine Scaler-CMC bestückt und gelötet aus der E-Werkstatt kommt, wird als erstes getestet, ob alle Leitungen richtig verbunden sind. Leider kann für die Scaler-CMC das Boundary-Scan-Testsystem (Abschnitt 6.1) nicht benutzt werden, weil es mit dem Virtex-E nur ein Bauteil auf der Platine gibt, das dies unterstützt. Deshalb wird mit einer anderen Methode geprüft, ob alle Datenleitungen funktionieren, da im Betrieb kaum auffällt, wenn eine einzige Leitung nicht funktioniert. Zu diesem Zweck gibt es ein spezielles Testprogramm, das auf den 32 Datenleitungen eine vordefinierte Datenfolge (Tabelle 7.1) sendet. Die Folge muss speziell darauf abgestimmt sein, alle möglichen Arten von Fehlern zu erkennen. Die einfachsten Fehler liegen darin, dass ein Bit permanent eins oder null ist. Aber auch Fehler, die auftreten, wenn es Wechselwirkungen zwischen zwei Leitungen gibt, müssen erkannt werden. Wenn die ausgelesene Bitfolge von der geschickten abweicht, wird anhand der Abweichung rekonstruiert, welche Leitungen fehlerhaft sind. Das Programm ist ursprünglich zum Testen von HOTLink-CMCs geschrieben worden und wurde geringfügig für die Scaler-CMC modifiziert.

Tabelle 7.1: Die FIFO-Testsequenz zum Test der Datenleitungen auf einer CMC.

1. Datenwort	00000000000000000000000000000000
2. Datenwort	11111111111111111111111111111111
3. Datenwort	01010101010101010101010101010101
4. Datenwort	10101010101010101010101010101010

7.2 Das Datenformat

Wenn sichergestellt ist, dass die Datenleitungen funktionieren, sollte überprüft werden, ob die State-Machines zuverlässig arbeiten. Hierzu werden keine Zähl-signale gebraucht, sondern nur Trigger. Es wird überprüft, ob die richtige Anzahl an Header- und Datenworten geschrieben wird und ob die Trigger-Nummern stimmen.

7.3 Der Zähltest

Wenn auch dies sichergestellt ist, kann mit den Zähl-Tests begonnen werden. Als erstes können manuell Einzelpulse eines Pulsers auf die Zähler-Kanäle gegeben werden. Wenn es Probleme gibt, die durch schlechtes Routing hervorgerufen werden, würden diese schon jetzt auftreten. Um einwandfreies Zählen bei höheren Frequenzen zu verifizieren, muss zu fortgeschrittenen Testmethoden (Abbildung 7.1) gegriffen werden.

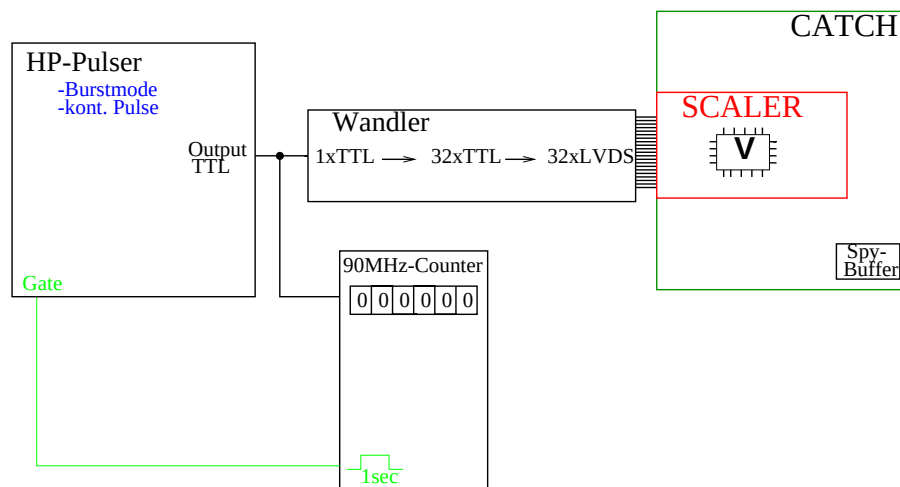


Abbildung 7.1: Der Testaufbau mit Pulser, Wandler und Referenzzähler.

Zum Testen gibt es drei konkurrierende Verfahren. Bei allen dreien wird ein Pulser¹ genutzt, der beliebige LVTTTL-Signale erzeugen kann. Die Signale werden durch eine Elektronik vervielfacht und nach LVPECL konvertiert. Diese Signale werden vom Terminierungsnetzwerk der Scaler-CMC auf LVDS-Niveau gezogen, so dass sie vom FPGA registriert werden können. Die Qualität dieser Schaltung entscheidet über die Qualität der Signale und ist daher mitbestimmend, wie gut gezählt werden kann.

Die Schwachstelle bei diesem Aufbau liegt bei den LVTTTL-Eingangssignalen.

¹Hewlett Packard 8131 A 500MHz Pulse Generator

Schon bei Frequenzen von ca. 100 MHz wird nicht mehr jedes Signal am Eingang des Wandlers erkannt. Es muss bei diesem Aufbau damit gerechnet werden, dass nicht jedes LVTTTL-Eingangssignal nach LVPECL übersetzt werden kann. Wenn aber ein Signal am Eingang erkannt wird, wird es auf alle 32 Kanäle verteilt.

- Das erste Verfahren benutzt den Burstmodus des Pulsers. Hierbei wird von dem Gerät ein Burst von Pulsen generiert, dessen Frequenz und Intensität einstellbar ist. Nach dem Auslösen eines solchen Bursts wird die Anzahl der registrierten Pulse mit der der generierten verglichen. Leider ist dieser Modus nur bei relativ niedrigen Frequenzen und hohen Pulsbreiten verfügbar.
- Eine weitere Methode besteht darin, einen Referenz-Zähler parallel mitzählen zu lassen. Die Zählerstände werden dann verglichen. Diese Methode wird nur bei niedrigen Frequenzen angewandt, da nur ein 90-MHz-Zähler zur Verfügung steht.
- Die dritte Methode für Raten über 90 MHz nutzt den Gate-Eingang des Pulsers. An diesem wird für die Dauer einer Sekunde ein Signal angelegt, worauf für diese Zeit kontinuierliche Pulse geschickt werden. Wenn beispielsweise Signale mit 192 MHz generiert wurden, erwartet man den Zählerstand 192 Millionen vorzufinden. Diese Methode ist abhängig von der Genauigkeit, mit der das Zeitfenster zum Zählen eingestellt werden kann.

Die Ergebnisse

Die ersten beiden Testverfahren lieferten Ergebnisse, die als eindeutige Erfolge zu werten sind. Beim Burst-Verfahren gab es im gesamten Anwendungsbe- reich nie einen Unterschied zwischen der Anzahl der gesendeten Pulse und den ausgelesenen Zählerständen. Ebenso ermutigend waren die Ergebnisse beim zweiten Verfahren. Die Ergebnisse des dritten Verfahrens machen jedoch auf Grund der oben genannten Probleme eine genauere Betrachtung notwendig. Die Periode des Testsignals wurde im Bereich von 3.0 ns bis 10.0 ns variiert. Die Länge der Signale (also die Zeit, in der der Signalpegel eins war) wurde zwischen 0.4 ns und 5 ns eingestellt. In Tabelle 7.2 wird eine Auswahl der Messergebnisse dargestellt.

Beim Betrachten der Ergebnisse fällt auf, dass die gemessenen Zählraten erheblich von den vorhergesagten abweichen. Die Ursache dieser Abweichungen liegt in der mangelnden Qualität der LVTTTL-Signale (siehe oben). Ein einfacher Vergleich $\text{Zählerwert} - \text{Signalrate} \times 1 \text{ s}$ reicht nicht aus. Da es aber

Tabelle 7.2: Ausgewählte Ergebnisse der Zählertests vom 04.11.2000. (Abbildung 7.1).

Messung	Periode	Frequenz	Breite	Zählerstand	Abweichung
1	10 ns	100 MHz	5 ns	99.01 M	0.99 %
2	10 ns	100 MHz	1.5 ns	98.95 M	1.05 %
3	10 ns	100 MHz	1.4 ns	98.96 M*	1.04 %
4	5 ns	200 MHz	1.5 ns	199.41 M	0.30 %
5	4 ns	250 MHz	1.5 ns	243.14 M	0.90 %
6	3.8 ns	263 MHz	1.5 ns	254.13 M	3.43 %
7	3.5 ns	285 MHz	1.5 ns	*	*
8	3 ns	333 MHz	1.5 ns	*	*

32 Zähler gibt, ist es möglich die Ergebnisse der Zähler untereinander zu vergleichen. Wenn es zwischen den Zählern Abweichungen gibt, wird das Signal nicht mehr richtig gezählt. Abweichungen zwischen den Zählern sind in Tabelle 7.2 durch ein „*“ gekennzeichnet.

Es ist notwendig abzuwägen, wie wahrscheinlich es ist, dass die Abweichungen vielleicht durch ein kollektives Versagen aller Zähler im selben Moment ausgelöst wird. Solch ein Fehler ist bei niedrigeren Raten nie aufgetreten. Dadurch wird ausgeschlossen, dass es sich um einen Designfehler handelt. Es kann also nur daran liegen, dass bei hohen Geschwindigkeiten die Vorzähler aussetzen. Dies ist unwahrscheinlich, da die Leitungen bei allen Vorzählern anders verlaufen, und sie sich deshalb auch im Zeitverhalten unterscheiden. Wenn ein solcher Fehler doch auftreten sollte, dann würde das bedeuten, dass das Ergebnis um etwa 12.5 % falsch sein muss, da der Vorzähler nur bis acht zählt und anschließend wieder bei null startet. Bei Abweichungen unter 12.5 % kann daher davon ausgegangen werden, dass sie allein auf die mangelnde Umwandlung von LVTTTL auf LVDS zurückzuführen sind.

Die Messungen 1–4 wurden durchgeführt, um eine geeignete Pulslänge für die weiteren Messungen zu finden. Es stellte sich heraus, dass LVTTTL-Signale mit einer Pulslänge von ≥ 1.5 ns korrekt gezählt werden können.

Bei Messung 4–8 wurde dann die Frequenz variiert, wobei Messung 6 die Messung mit der höchsten erfolgreich gezählten Frequenz ist. Bemerkenswert an dieser Messung ist, dass es bei einer höheren Frequenz gelungen ist, mehr als 250 Millionen Pulse in einer Sekunde zu zählen.

Durch diese Tests konnte bestätigt werden, dass der Zähler in der Lage ist, 250-MHz-Signale zu zählen. Damit wird die Anforderung von 200 MHz (vgl. Abschnitt 4.1) deutlich übertroffen. Vermutlich sind seine Leistungen besser, aber die zur Verfügung stehenden Testmethoden reichen nicht aus, um höhere Geschwindigkeiten zweifelsfrei zu bestätigen.

7.4 Der Bit-Test

Nachdem während der Strahlzeit im August 2000 (siehe Kapitel 8) mit dem Zähler auch physikalische Detektorsignale gezählt worden sind, erschloss sich noch eine weitere Testmöglichkeit. Eine Scaler-CMC liefert mit jeder Auslese 32 Zählerstände. Während eines Spills wurden die Zähler unter den Testbedingungen bis zu 12 000 mal ausgelesen. Somit werden pro Spill 384 000 Datenworte ausgelesen. Zwischen zwei Auslesen lag eine Zeit von mindestens $1\,\mu\text{s}$, d.h., der Zählerstand wird bei den Testbedingungen (ca. 4 MHz pro Kanal) mindestens um ca. 4 000 erhöht. Daraus folgt, dass die ersten 12 Bit nicht korreliert sein dürfen, sondern sie sollten bei jeder Auslese jeweils mit einer Wahrscheinlichkeit von 0.5 eins oder null sein. Bei den höheren Bits sollte sich die Wahrscheinlichkeit zugunsten der Null verschieben, weil bei den ersten Auslesen z.B. Bit 20 nicht eins sein kann, weil dieser hohe Zählerstand in der kurzen Zeit nicht erreicht werden konnte. Bit 23 und höher können bei den gemessenen Raten und der Spilldauer niemals eins werden. Bei Abweichungen von dieser Vorhersage kann davon ausgegangen werden, dass irgendwo ein Bitfehler aufgetreten ist.

Dank der großen Zahl ausgelesener Datenworte kann die Vorhersage kontrolliert werden, indem gezählt wird, wie oft ein Bit eins (N_1) und null (N_0) ist. In Abbildung 7.2 wurde der Wert der normierten Differenz $\frac{N_0 - N_1}{N_0 + N_1}$ für jedes Bit aufgetragen. Die Ergebnisse zeigen, dass keine Abweichungen von den Erwartungen zu beobachten sind.

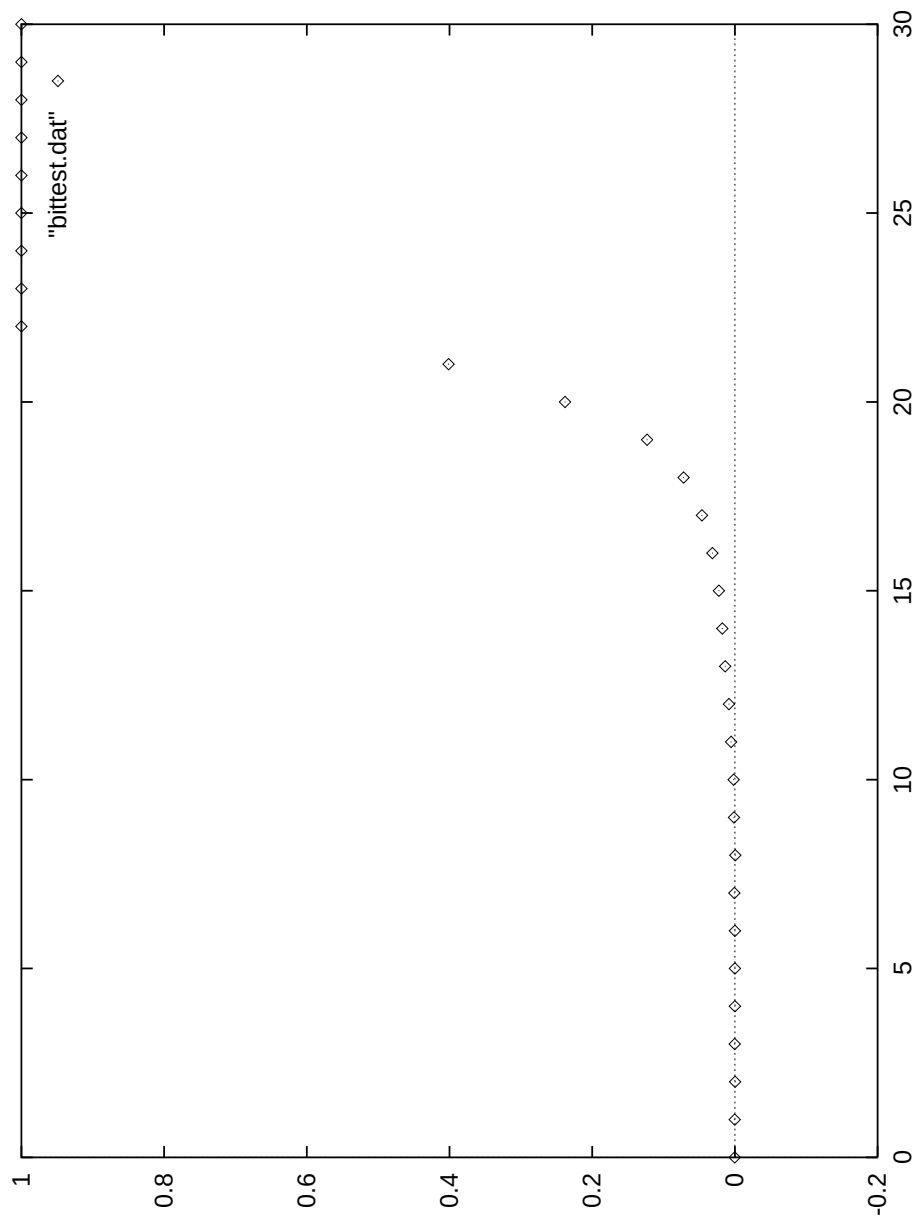


Abbildung 7.2: Der Wert der Asymmetrie $\frac{N_0 - N_1}{N_0 + N_1}$ (Ordinate) gegen Bit-Nummer (Abszisse).

Kapitel 8

Der Einsatz im COMPASS-Experiment

Im August 2000 wurde der Zähler im COMPASS-Experiment eingesetzt. Die erste Aufgabe, an der der Zähler mitwirken konnte, war die Messung des räumlichen und zeitlichen Strahlprofils. Hierzu wurden szintillierende Fasern (SciFi, Abbildung 8.1) an den Zähler angeschlossen. Sie stehen etwa an der Position des polarisierten COMPASS-Targets. Die SciFis bestehen aus jeweils 96 Fasern in x- und y-Richtung. Jede Faser hat einen Durchmesser von 1 mm, wobei sich zwei benachbarte Fasern jeweils zur Hälfte überlappen. Mit drei Scaler-CMCs auf einem CATCH kann die Projektion des Strahlprofils auf die x- bzw. y-Achse gemessen werden. Bei dieser Messung wurde es nötig, die Zählerstände zu visualisieren. Zu diesem Zweck wurde das Monitoring-Programm für die TDCs `monTDC` so modifiziert, dass es möglich wurde, damit die Zählerstände anzuzeigen.

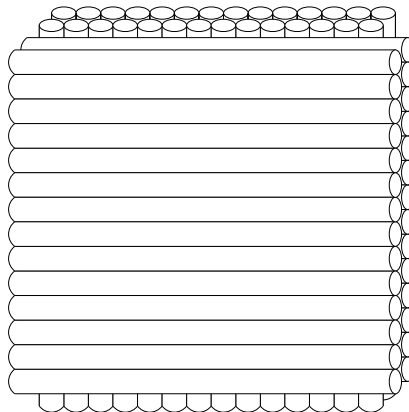


Abbildung 8.1: Schematische Darstellung der SciFis.

Das neu entstandene Programm `monSCALER` wird auf die von DATE¹ [41] aufgezeichneten Daten angewandt. Es bearbeitet sie dann und trägt sie in Histogramme ein.

8.1 Das räumliche Strahlprofil

Die einfachste Messung für den Zähler ist die des räumlichen Strahlprofils integriert über einen Spill (Abbildung 8.2 und 8.3). Hierzu muss nur das letzte Ereignis im Spill ausgewertet werden.

Zu den Abbildungen muss noch angemerkt werden, dass nur die statistischen Fehler eingezeichnet sind. Die Verstärkungsfaktoren und die Diskriminatorschwellen des SciFi-Detektors sind zum Zeitpunkt der Messung noch nicht optimal eingestellt gewesen. Insbesondere bewirken unterschiedlich eingestellte Vorverstärker Ungleichheiten bei den Sensitivitäten der einzelnen Kanäle. Dies führt zu einer deutlichen Substruktur, die das Strahlprofil überlagert. Diese Struktur taucht bei allen räumlichen Profilmessungen auf.

8.2 Das räumliche Strahlprofil im zeitlichen Verlauf

Ein Zähler mit totzeitfreier Auslese kann nicht nur das über einen Spill integrierte Strahlprofil messen, sondern auch Änderungen des Profils im Verlauf eines Spills.

Exemplarisch wurden die Projektionen auf die x- und y-Achse für zwei Spills [42] aufgenommen. Bei diesen Spills gab es verhältnismäßig geringe Strahlintensitäten von ca. $3 \cdot 10^7$ Myonen pro Spill bei einer Spilldauer von 3.2s. Das entspricht einer Rate von ca. 300 kHz pro SciFi-Kanal. Parallel laufende Tests an anderen Detektoren erforderten sehr enge Triggerbedingungen. Daher wurden in einem Spill nur etwa 350 physikalische Trigger generiert.

Um die zeitliche Veränderung des Strahlprofils darzustellen (Abbildungen 8.4 und 8.5), wurde zu acht verschiedenen Zeitpunkten eine Momentaufnahme des Profils gemacht. Abbildung 8.4.1 bzw. 8.5.1 entstand durch das Subtrahieren des Zählerstand beim vierten Trigger vom Zählerstand beim sechzehnten Trigger. Es wird das Profil zwischen dem vierten und sechzehnten Trigger dargestellt. Dies wurde sieben mal zu anderen Zeitpunkten wiederholt (siehe Tabellen 8.1 und 8.2). Zur genauen Bestimmung des Maximums des Profils

¹Data Acquisition Test Environment, von der ALICE-Kollaboration [40] entwickeltes Programm-Paket für die Datennahme.

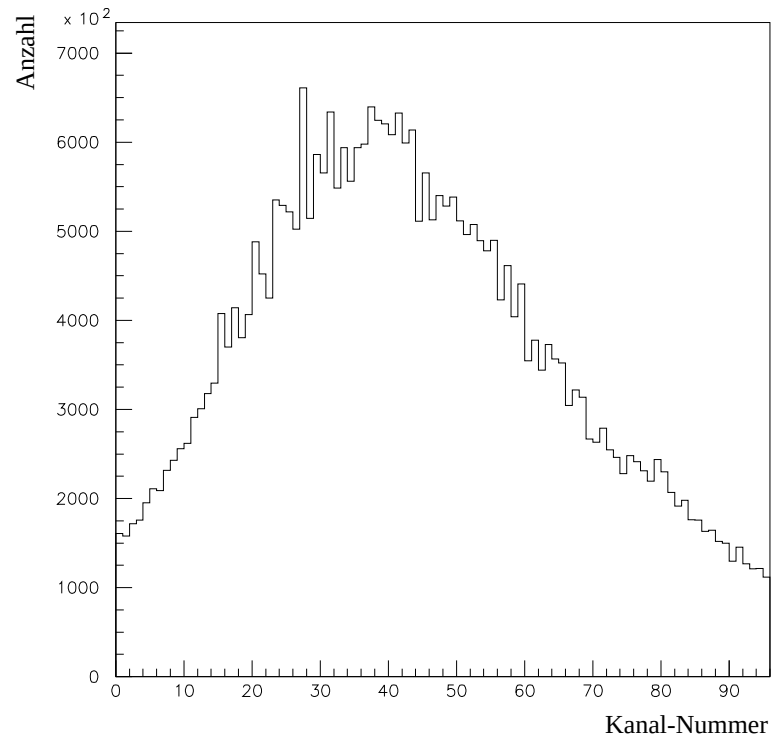


Abbildung 8.2: Projektion des Strahlprofils auf die x-Achse, über einen Spill integriert.

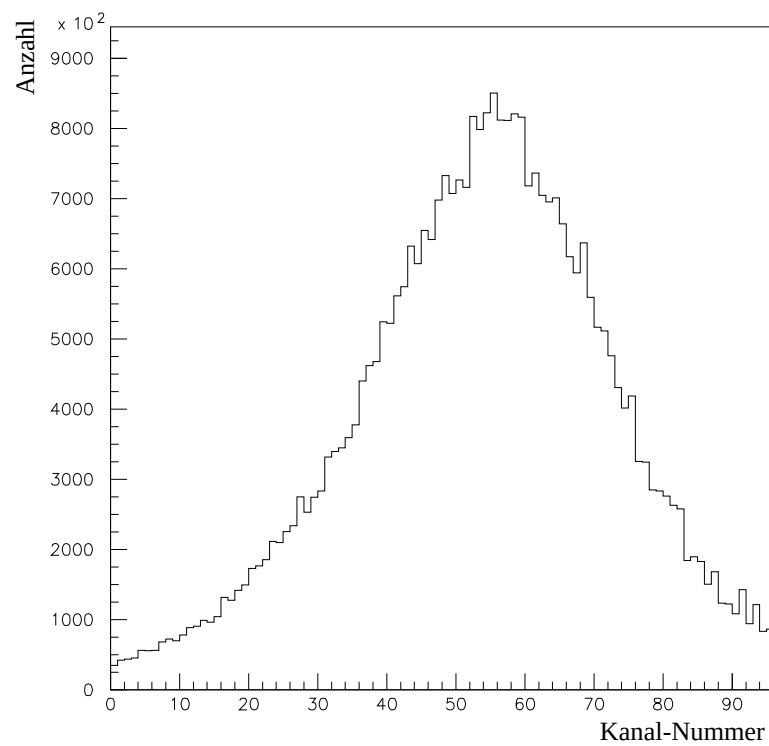


Abbildung 8.3: Projektion des Strahlprofils auf die y-Achse, über einen Spill integriert.

wurde eine Gaußkurve an die zentralen Bereiche der Spektren angepasst.

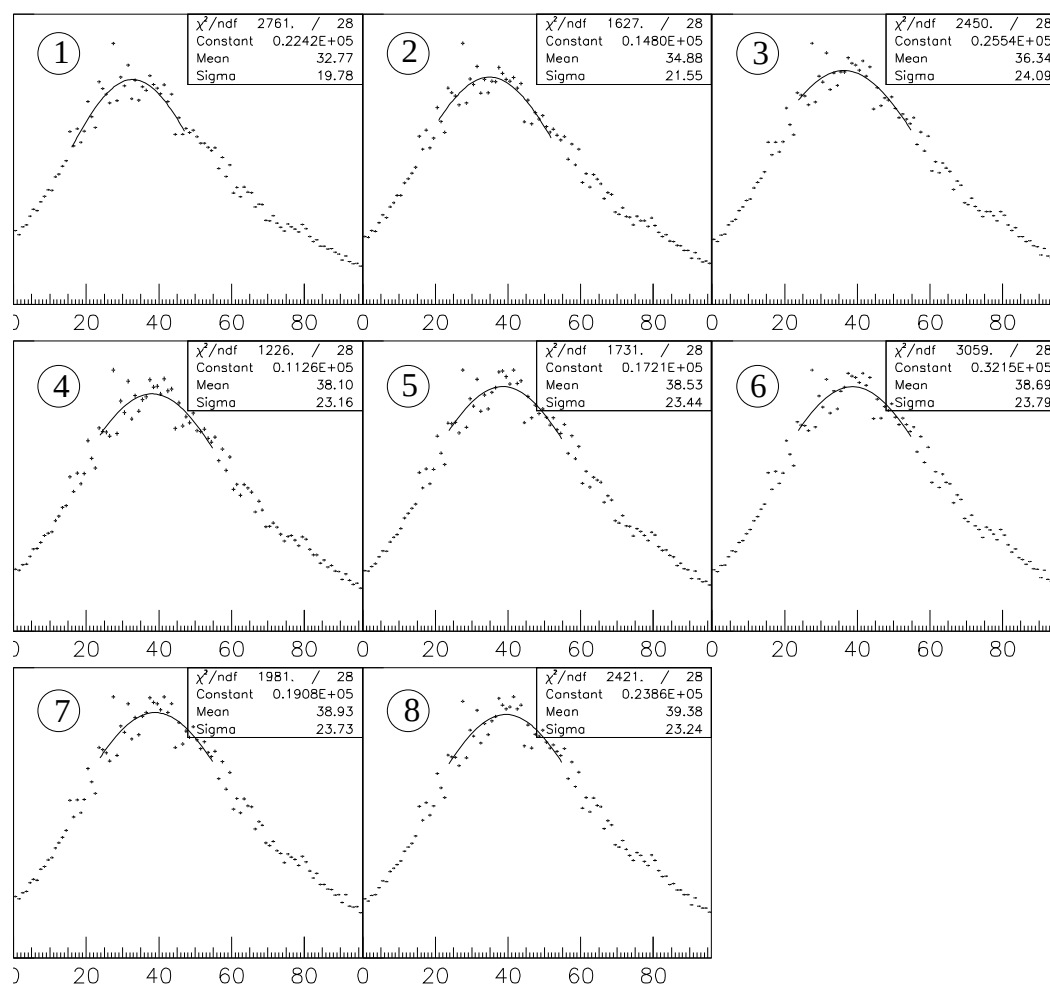


Abbildung 8.4: Veränderung des mit dem SciFi-Detektor gemessenen Strahlprofil innerhalb eines Spills, Projektion auf die x-Achse. Auf der x-Achse sind die Kanalnummern aufgetragen. Rückschlüsse auf die Intensitäten lassen sich aus dieser Abbildung nicht ziehen (siehe Fit-Parameter und Tabelle 8.1).

Bei den Projektionen des Strahlprofils auf die x-Achse (Abbildung 8.4, Tabelle 8.1) ist deutlich zu erkennen, dass der Strahl im Verlauf eines Spills ca. 6–7 Kanäle, also ca. 3–3.5 mm über den Detektor wandert. Anders ist dies bei den y-Profilen (Abbildung 8.5, Tabelle 8.2). Hier bewegt sich der Mittelpunkt des Strahls nur innerhalb von 0.5 mm.

Da die Daten mit physikalischen Triggern genommen wurden, lassen sich keine Rückschlüsse auf die Intensitätsentwicklung ziehen. Aus diesem Grund sind

Tabelle 8.1: Daten zu Abbildung 8.4, von links oben nach rechts unten.

Bild- Nummer	Beginn Messung (Trigger-#)	Ende Messung (Trigger-#)	Anzahl der Myonen	Kanäle f. Gauß-Fit	Mittelpunkt (Kanal)
1	4	16	$1.3 \cdot 10^6$	17–47	32.8 ± 0.1
2	52	64	$0.9 \cdot 10^6$	22–52	34.9 ± 0.1
3	103	115	$1.5 \cdot 10^6$	25–55	36.3 ± 0.1
4	151	163	$0.7 \cdot 10^6$	25–55	38.1 ± 0.1
5	202	214	$1.0 \cdot 10^6$	25–55	38.5 ± 0.1
6	250	262	$1.9 \cdot 10^6$	25–55	38.7 ± 0.1
7	301	313	$1.1 \cdot 10^6$	25–55	38.9 ± 0.1
8	346	358	$1.4 \cdot 10^6$	25–55	39.4 ± 0.1

Tabelle 8.2: Daten zu Abbildung 8.5, von links oben nach rechts unten.

Bild- Nummer	Beginn Messung (Trigger-#)	Ende Messung (Trigger-#)	Anzahl der Myonen	Kanäle f. Gauß-Fit	Mittelpunkt (Kanal)
1	4	16	$0.6 \cdot 10^6$	45–65	55.6 ± 0.1
2	52	64	$1.0 \cdot 10^6$	45–65	55.8 ± 0.1
3	103	115	$1.2 \cdot 10^6$	45–65	56.0 ± 0.1
4	151	163	$1.1 \cdot 10^6$	45–65	55.8 ± 0.1
5	202	214	$0.6 \cdot 10^6$	45–65	55.8 ± 0.1
6	250	262	$1.2 \cdot 10^6$	45–65	55.7 ± 0.1
7	301	313	$1.6 \cdot 10^6$	45–65	55.7 ± 0.1
8	346	358	$1.0 \cdot 10^6$	45–65	55.6 ± 0.1

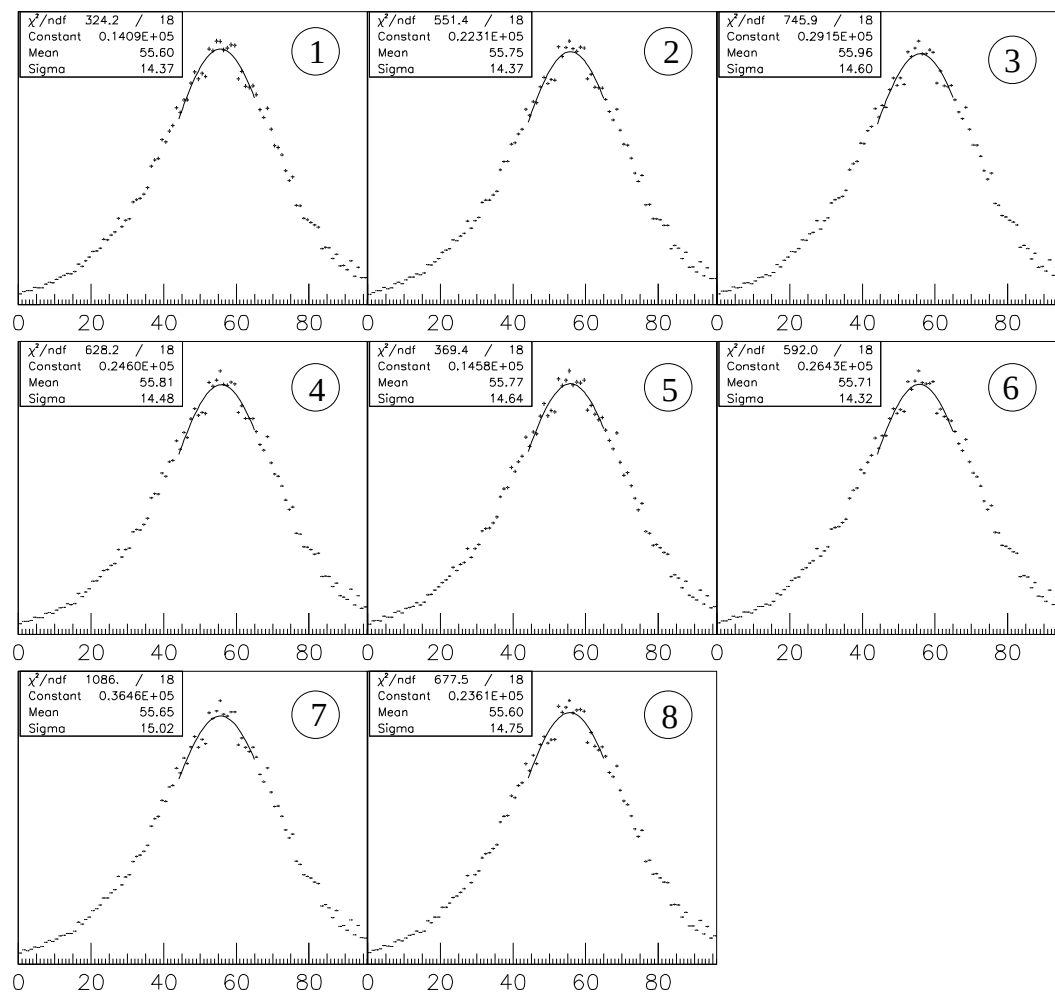


Abbildung 8.5: Veränderung des mit dem SciFi-Detektor gemessenen Strahlprofils innerhalb eines Spills, Projektion auf die y-Achse. Auf der y-Achse sind die Kanalnummern aufgetragen. Rückschlüsse auf die Intensitäten lassen sich aus dieser Abbildung nicht ziehen (siehe Fit-Parameter und Tabelle 8.2).

auf den Ordinaten der Abbildungen nur relative Einheiten aufgetragen.

8.3 Das zeitliche Strahlprofil

Bei den bisher vorgestellten Messungen konnte sich auf Grund der niedrigen Triggerraten die volle Leistungsfähigkeit des Zählers noch nicht entfalten. Der Zähler kann innerhalb einer Sekunde mehr als 100 000 mal ausgelesen werden. Dadurch ist es möglich, ein hochauflösendes zeitliches Strahlprofil (Abbildung 8.6) zu erstellen. Bei den Messungen in Abschnitt 8.2 konnte ein solches Profil nicht angefertigt werden, da es physikalische Triggerbedingungen gab.

Ein periodisches Trigger-Signal wäre ideal zur Aufnahme eines zeitlichen Strahlprofils. Solche Daten liegen bisher nicht vor. Es gibt aber eine Messung mit hoher Strahlintensität [43], bei der es ein zufällig generiertes Trigger-Signal gab. Diese Daten wurden mit einem Zähler aufgenommen. Es konnten bei

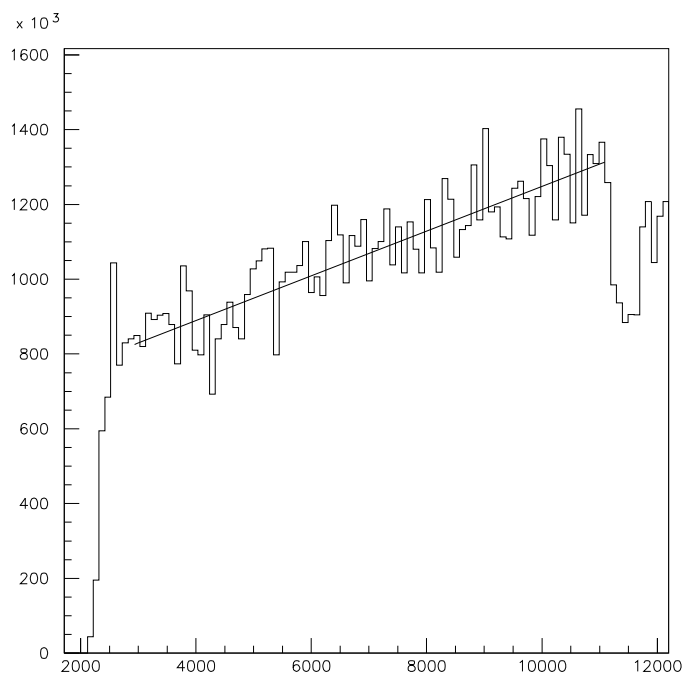


Abbildung 8.6: Das mit den SciFis gemessene zeitliche Strahlprofil.

den Messungen des zeitlichen Strahlprofils somit 32 SciFi-Kanäle aufgenommen werden. Demzufolge können solche zeitlichen Betrachtungen auf knapp einem Drittel der räumlichen Ausdehnung des Strahls gemacht werden. Dies

reicht aus, um einen Eindruck der zeitlichen Entwicklung zu erhalten. Um das Histogramm in Abbildung 8.6 zu erstellen, wurden jeweils die Ereignisse in allen 32 Kanälen zwischen jedem hundertsten Trigger summiert und gegen die Trigger-Nummer aufgetragen.

Es ist deutlich zu erkennen, dass die Schwankungen des Profils über den statistischen Zählratenfluktuationen liegen. Dies ist darin begründet, dass die Zeiten zwischen den einzelnen zufälligen Trigger unterschiedlich sind. Die ansteigende Tendenz der Intensität innerhalb des Spills ist deutlich zu erkennen. Von Ereignis 300 bis 11 000 steigt die Intensität etwa um den Faktor 1.6 an. Das Abfallen der Intensität ist nicht mehr ganz zu erkennen. Aus anderen Messungen ist jedoch bekannt, dass bei diesen Daten (Abbildung 8.6) mehr als 90 % des zeitlichen Strahlverlaufs gemessen wurden.

Kapitel 9

Fazit

Es wurde ein 250-MHz-Zähler mit totzeitfreier Auslese entwickelt, der den Anforderungen des COMPASS-Experiments entspricht (vgl. Kapitel 4.1). Dieser Zähler basiert auf einer Platine, deren wichtigstes Bauteil ein Field Programmable Gate Array (FPGA) ist, auf welchem alle logischen Schaltungen realisiert werden. Die Auswahl des FPGAs und das Einbinden desselben in die Elektronikumgebung erwies sich als anspruchsvolle Aufgabe. Bei der Gestaltung der Platine wurde viel Wert auf flexible Anwendungsmöglichkeiten gelegt. So kann die Scaler-CMC nicht nur als Zähler, sondern auch als Input-Register benutzt werden.

Der größte Teil der Arbeit wurde in das Zähler-Design des FPGAs investiert. Ergebnis der Bemühungen war ein innovatives Design, das alle Anforderungen des Experiments erfüllt, sie sogar in einigen Bereichen deutlich übertrifft.

Das Testen der Scaler-CMC und die Erprobung des Zählerdesigns war ein weiterer umfangreicher Teil der Arbeit. Auch hierbei zahlte sich die weitsichtige Konstruktion mit vielen Testmöglichkeiten aus.

Der Zähler konnte im Sommer 2000 während der ersten COMPASS-Strahlzeit erfolgreich zur Bestimmung von Strahlprofilen eingesetzt werden. Neben der Messung integraler räumlicher Strahlprofile (x-, y-Projektion), konnte nachgewiesen werden, dass sich der Strahlfleck auf dem Target im Laufe eines Spills ca. 3–3.5 mm in x-Richtung bewegt. Die Ergebnisse dieser Messung werden mit den Experten des SPS besprochen. Auch die Intensitätsschwankung (Faktor 1.6) im Laufe eines Spills konnte gemessen werden. Wegen der Auslese-Architektur des Experiments kann diese Messung nur mit einem Zähler mit totzeitfreier Auslese durchgeführt werden. Um die Aussagekraft dieser Messungen zu erhöhen, sollten sie noch einmal mit einer Clock als Triggersignal durchgeführt werden.

Es wurden alle vorgesehenen vier Scaler-CMCs hergestellt. Auf Grund des gestiegenen Bedarfs an Zählern innerhalb der Kollaboration und der positiven Erfahrungen mit ihnen wird überlegt, ob nicht noch weitere Exemplare gefer-

tigt werden sollen. Dazu wird die Möglichkeit untersucht, einen preiswerteren FPGA einzusetzen, um die Materialkosten zu senken.

Die Zähler-Aufsteckkarte ist so konzipiert, dass aus ihren 32 Eingangssignalen mit Hilfe des FPGAs beliebige logische Funktionen berechnet werden können. Auch eine komplette Trigger-Logik ließe sich implementieren. Damit ist die Scaler-CMC eine wichtige Erweiterung des an der Universität Freiburg entwickelten CATCH-Auslesesystems. Zusammen mit den TDC- und HOTLink-Aufsteckkarten verfügt man über alle Elektronik-Komponenten, die zum Betreiben eines Experiments notwendig sind.

Insgesamt ist die Entwicklung der Scaler-CMC ein weiteres Beispiel dafür, dass mit Hilfe von programmierbarer Logik innerhalb kürzester Zeit einsatzfähige Elektronikkomponenten entwickelt werden können. Der Zähler beweist, dass man mit Hilfe dieser Technik Geräte fertigen kann, die mit ihren Leistungen kommerziellen Produkten in vielerlei Hinsicht überlegen sind und sich viel besser an bestehende Umgebungen anpassen lassen.

Anhang A

Programmierbare Logik

Der Zähler soll in einem programmierbaren Logikbaustein realisiert werden. Solche Bausteine haben gegenüber den gewöhnlichen ASICs (Application-Specific Integrated Circuits) einen großen Vorteil. Während man sich beim ASIC schon von Anfang an auf alle Spezifizierungen festlegen muss, hat man bei programmierbarer Logik die Möglichkeit die Funktion im Nachhinein zu ändern. Dadurch wird eine große Flexibilität erhalten. Außerdem gibt es die hohen Fixkosten nicht, die beim Entwurf eines ASICs anfallen.

A.1 Entwicklung der programmierbaren Logik

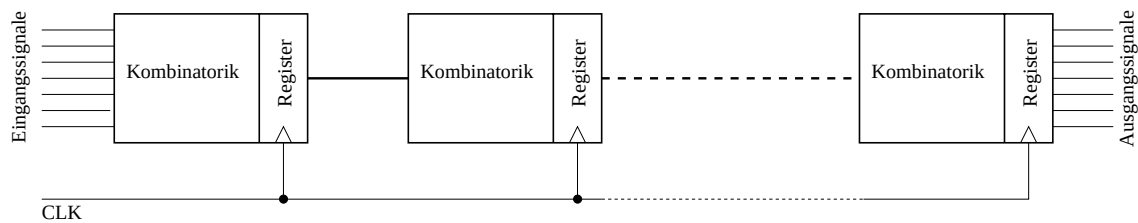
Die Entwicklung der programmierbaren Logik begann in den frühen 70er Jahren mit dem PLA (Programmable Logic Array). Dies war eine Anordnung von UND-Gattern, ODER-Gattern und Invertiern, die flexibel hintereinander geschaltet werden konnten und so eine Vielzahl von kombinatorischen Schaltungen ermöglichten. Dieses Prinzip wurde im PAL (Programmable Array Logic) noch weiter verfeinert. Nach wie vor gab es keine Möglichkeit, auf diese Weise sequenzielle Logik zu designen. Dies wurde durch den SPLD (Simple Programmable Logic Device) möglich. In diesem wurden die bewährten PALs mit Flip-Flops zusammengeschaltet. Die Ausgänge der Flip-Flops gingen wieder auf Eingänge von PALs.

Als nächstes folgte der Schritt zum CPLD (Complex Programmable Logic Device), einer noch heute häufig angewandten Form der programmierbaren Logik. CPLDs sind eine Matrix von SPLDs, die über eine große Schaltermatrix verbunden sind. Alle bisher behandelten Formen wiesen ein gemeinsames Merkmal auf. Die kombinatorische Logik wird immer durch Arrays von UND- und ODER-Gattern realisiert. Ein anders funktionierender Entwurf eines programmierbaren Logikbausteins ist der Field-Programmable Gate Array (FPGA). Seine Technik wird im folgenden Abschnitt ausführlicher beschrieben.

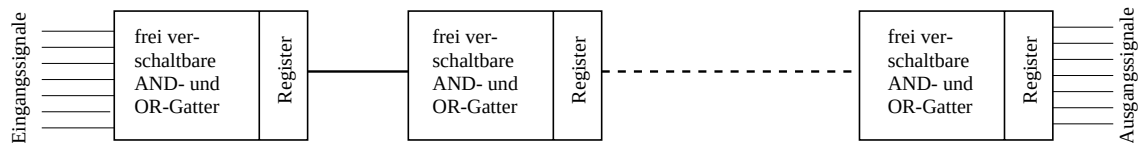
A.2 Der FPGA — Xilinx Virtex-E XCV600E-HQ240

Wenn dargestellt werden soll, wie ein FPGA funktioniert, muss zuerst noch einmal daran erinnert werden, dass moderne Logik immer sequenziell ist (Abbildung A.1). Sie besteht also immer aus Kombinatorik (zum Rechnen) und Registern (zum Speichern). Beim CPLD wird die Kombinatorik durch frei zusammenschaltbare UND- und ODER-Gatter realisiert. Zum Speichern werden herkömmliche Flip-Flops benutzt. Diese Flip-Flops sind zum selben Zweck auch in FPGAs vorhanden. Die Kombinatorik wird durch andere Elemente realisiert: die Look-Up-Table (LUT). Diese Elemente sind nichts anderes als 16×1 -bit-Speicher. In den folgenden Abschnitten wird beschrieben, wie sich der FPGA aus diesen und anderen Elementen zusammensetzt [44], und welche Vor- und Nachteile damit verbunden sind.

Sequenzielle Logik im Allgemeinen:



Der CPLD:



Der FPGA:

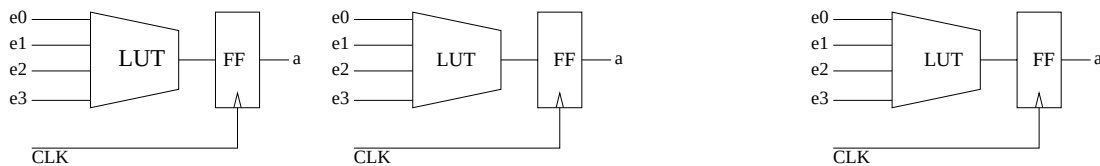


Abbildung A.1: Realisierung einer sequenziellen Schaltung in programmierbarer Logik.

Die Input-Output-Blocks

Die Input-Output-Blocks (IOB, Abbildung A.2) sind die Bauteile im FPGA, die entscheiden, welche Signalstandards an die entsprechenden Pins angelegt werden

können. Dazu wird jedem Pin ein IOB zugeordnet. Virtex-E FPGAs unterstützen eine lange Liste von Signal-Standards, wie LVTTTL, LVCMOS2, LVCMOS18, SSTL3 I & II, GTL, GTL+, BLVDS & LVDS, LVPECL, usw..

Von diesen sollten insbesondere die differenziellen Standards LVPECL und LVDS hervorgehoben werden. Der Virtex-E-FPGA ist der erste FPGA von Xilinx der solche Standards unterstützt. Bei diesem Projekt werden Eingangssignale für die Zähler und einige Steuersignale in LVDS übertragen. Differenzielle Signale bieten den Vorteil, dass mit ihnen auf Grund des größeren Störabstands viel höhere Frequenzen erreicht werden können. Insbesondere bei den zu zählenden Signalen spielt dies eine große Rolle. Mit 200 MHz LVTTTL-Signalen kann es schon Probleme geben. Wesentliches Merkmal der IOB-Architektur ist die Auslegung darauf, dass die Puffer sowohl als Input, als auch als Output fungieren können. Im wesentlichen sind die IOBs aus folgenden Bauteilen aufgebaut:

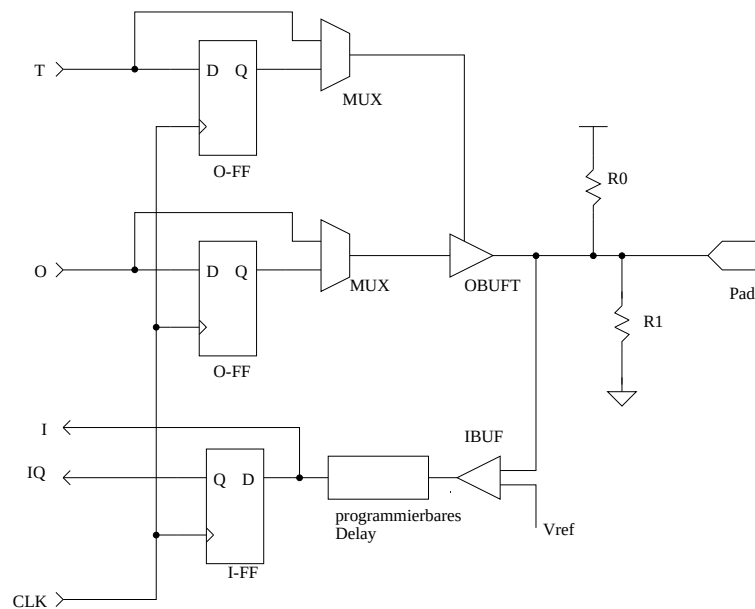


Abbildung A.2: Vereinfachte Darstellung eines IOBs.

- ein Pad, die Verbindung zum Pin.
- drei Flip-Flops
- Terminierungswiderstände für die unterstützten Signalstandards
- ein programmierbares Delay
- ein Tristate Ausgabe-Puffer

Die drei Flip-Flops dienen als Output- bzw. Input-Flip-Flops. Diese machen es möglich, synchrone Ausgangssignale zu generieren bzw. die eingehenden Signale schon in den IOBs zu synchronisieren. Da es auch Anwendungen gibt, bei denen diese Register nicht erwünscht sind, muss es eine Bypass-Möglichkeit geben. Diese wird durch programmierbare Multiplexer (MUX) geschaffen.

Darüber hinaus enthalten die IOBs ein Tristate-Buffer, welches es ermöglicht, dass an den Ausgängen nicht nur 1 und 0, sondern auch ein dritter, hochohmiger Zustand anliegen kann. In diesem Zustand verhalten sich die Pins neutral. Sie haben also weder eine Verbindung zu Masse noch zu V_{cc} ¹. Sie machen es möglich einen Multiplexer zu bauen.

Die Werte der Terminierungswiderstände können nicht pauschal angegeben werden, sie werden je nach Signalart gewählt. Das programmierbare Delay-Element am Eingang vereinfacht die Kommunikation zwischen dem FPGA und anderen elektronischen Bauteilen. Häufig treten bei der Inter-Chip-Kommunikation Timing-Probleme auf. Bestimmte Signale kommen früher als andere, und dies führt zu Schwierigkeiten. Indem diese Signale mit den Delays verzögert werden, kann das Problem gelöst werden.

Die Complex-Logic-Blocks

Die Complex Logic Blocks (CLB, Abbildung A.3) sind das Herzstück der FPGAs. In sie kann die sequenzielle Logik einprogrammiert werden. Der XCV600E verfügt über 3 456 CLBs. Sie sind in einem Feld von 48×72 Zeilen und Spalten angeordnet.

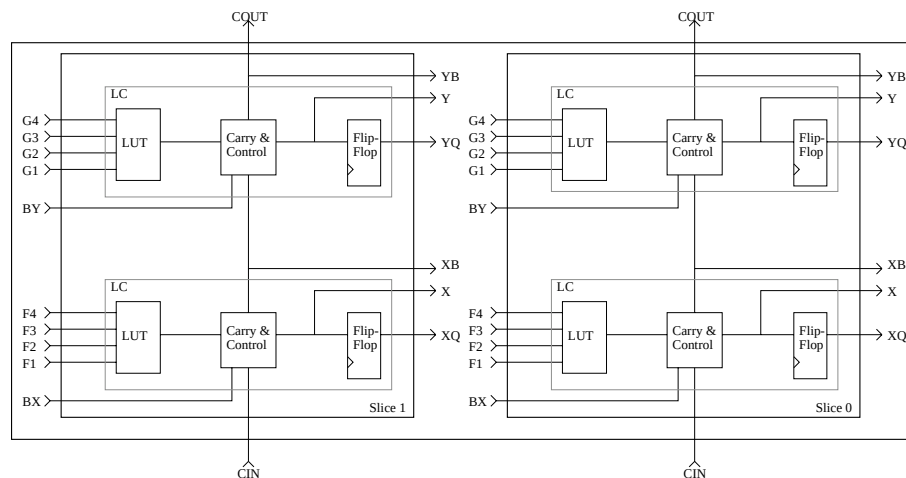


Abbildung A.3: Vereinfachter 2-Slice-CLB des Virtex-E.

Die CLBs bestehen jeweils aus zwei „Scheiben“ (Slices). Diese Scheiben bestehen wiederum aus zwei Logic Cells (LC), dem Grundbaustein des CLBs. Will man den

¹Betriebsspannung

Aufbau des CLBs und somit die Funktionsweise eines Virtex-E-FPGAs verstehen, so muss man zuerst verstehen, wie die Logic Cell aufgebaut ist. Als Speicherelement für sequenzielle Logik hat jede Logic Cell einen Flip-Flop. Somit gibt es in jedem CLB vier Flip-Flops. Kombinatorische Logik wird in LUTs (Abbildung A.4) programmiert. Wie schon der Name andeutet, sind LUTs eigentlich nichts anderes als programmierbare Speicherelemente.

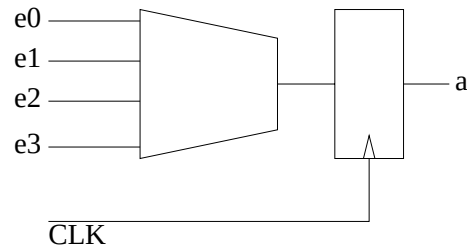


Abbildung A.4: Look-Up-Table mit Flip-Flop.

Eine Look-Up-Table hat vier Eingänge und einen Ausgang. Die Eingänge können als Zeiger für einen 16×1 -bit-RAM interpretiert werden. Wenn über diesen Zeiger eine Adresse angesprochen wird, wird deren Inhalt an den Ausgang gelegt. Dieses Element kann auch als beliebige binärwertige Funktion von vier Variablen interpretiert werden (Abbildung A.1). Mit einer Look-Up-Table kann somit jede Funktion von vier Signalen (beispielsweise ein UND4 oder ODER4) realisiert werden. Die

Tabelle A.1: Wahrheitstabelle einer als UND4 konfigurierten Look-Up-Table.

e0	e1	e2	e3	a
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
⋮	⋮	⋮	⋮	⋮
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

Struktur der CLBs bietet noch andere Möglichkeiten. So besteht ein CLB nicht nur aus vier Logic Cells. Es gibt auch eine Struktur, die die Logic Cells miteinander verbindet. Teil dieser Struktur ist eine weitere Look-Up-Table (F5), die es ermöglicht, jede logische Funktion von fünf Variablen oder sogar geeignete Funktionen von bis zu neun Variablen zu implementieren. Dazu würden sich beispielsweise die Funktionen UND9 und ODER9 eignen.

Die LUTs können nicht nur zum Generieren von Kombinatorik dienen, sondern sie

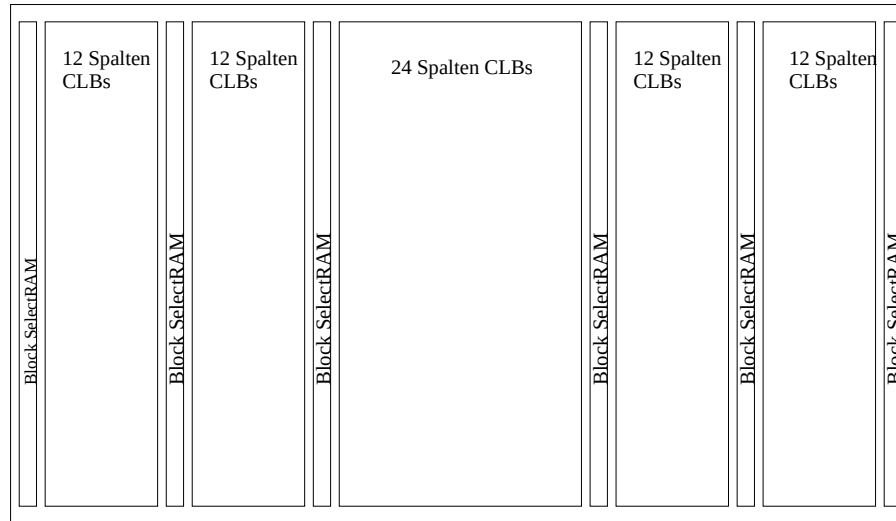


Abbildung A.5: Der Block SelectRAM im Virtex-E XVC600E.

stehen dem Programmierer auch in ihrer eigentlichen Funktion als RAM zur Verfügung. Jede LUT bietet 16 bit Speicherkapazität, insgesamt kommt man so auf 221 184 bit *Distributed RAM*. Mit den LUTs können z.B. Schieberegister gebaut werden, ohne auf Flip-Flops angewiesen zu sein. Die Addition zweier Zahlen ist eine triviale, in Mathematik und Informatik sehr häufig vorkommende Operation, die relativ viel Zeit in Anspruch nimmt, da die beiden Operanden bitweise addiert werden und sämtliche Überträge berücksichtigt werden müssen. D.h., eigentlich müsste eine Addition sequenziell durchgeführt werden und nicht parallel. Um diesen Vorgang zu beschleunigen, gibt es bei schnellen Hardware-Addierern eine spezielle Vorrichtung, die Carry-Chains. Bei solchen Bauteilen gibt es für die Überträge spezielle, schnelle Leitungen. Über solche Leitungen (*Carry & Control*) verfügen auch die CLBs des Virtex-E. Wie in den IOBs befinden sich auch zwischen den CLBs Tristate-Buffer. Hier dienen sie zum Betreiben von Bussen innerhalb des FPGAs.

Block SelectRAM

Neben den von den LUTs zur Verfügung gestellten *Distributed RAM* gibt es auch den *Block SelectRAM*. Dies sind Speicherkapazitäten, die sich außerhalb der CLBs befinden. Auf dem XCV600E sind 72 dieser Speicherzellen spaltenweise an bestimmten Stellen zwischen den CLBs angeordnet (Abbildung A.5). Die 6 Spalten bieten zusammen eine Kapazität von 294 912 bit. Insgesamt stehen auf dem FPGA also rund 64 kByte Speicher zur Verfügung. Der Block SelectRAM wird beispielsweise genutzt, wenn ein FIFO implementiert wird.

Routing-Möglichkeiten und DLL

Virtex-E FPGAs bieten eine Vielzahl von Routing-Möglichkeiten. Unter Routing-Möglichkeiten versteht man alle Arten von Leitungen zwischen den IOBs, CLBs und RAM-Blöcken. Hier gibt es verschiedene Kategorien von Leitungen und Schaltern. Es gibt beispielsweise lange, schnelle Leitungen mit wenigen Schaltern für große Distanzen, und es gibt kurze Leitungen mit vielen Schaltern für das Nah-Routing zwischen benachbarten CLBs und IOBs.

Dies ist notwendig, da das Routing das Zeitverhalten wesentlich beeinträchtigt. Ein großes Problem sind Clock-Netze: Bei großen Designs ist es wichtig, dass das Clock-Signal an allen Flip-Flops gleichzeitig ankommt. Besonders wichtig sind deshalb die globalen Clock-Netze. Dies sind Netze, die direkt an die Clock-Eingänge der Flip-Flops gehen. Auf diesen Netzen kann sich das Clock-Signal schnell ausbreiten, und dadurch wird ein sauberes Zeitverhalten gewährleistet. Aber auch diese schnellen Netze haben Verzögerungen auf Grund der Laufzeiten der Signale im FPGA. Diese können von einer DLL (Delay Locked Loop) beseitigt werden. Die DLL schickt in verschiedene Bereiche des FPGAs Clock-Signale mit unterschiedlicher Phasenlage, so dass diese laufzeitbedingten Verzögerungen eliminiert werden.

A.3 Der Designprozess

In diesem Abschnitt wird erläutert, wie ein Design für einen FPGA erstellt und implementiert wird [28, 29]. Zur Programmierung des FPGAs sind folgende Schritte notwendig (Abbildung A.6):

- Erstellung des Designs
- Eingabe der Pin- und Timing-Vorgaben
- Erstellung einer Netzliste
- Anpassung der Netzliste auf den FPGA
- Erstellung einer Programmierdatei für den FPGA
- Eigentliche Programmierung

Erstellung des Designs

Es gibt mehrere Möglichkeiten das Design in ein Format zu bringen, das für den Computer zugänglich ist. Die klassische Möglichkeit ist es, einen schematischen Schaltplan mit UND- und ODER-Gattern einzugeben (Abbildung A.7). Diese Art der Eingabe kann als klassisch bezeichnet werden, weil bei ihr im Prinzip jedes einzelne Gatter angegeben wird. Die elegantere und heute im allgemeinen bevorzugte

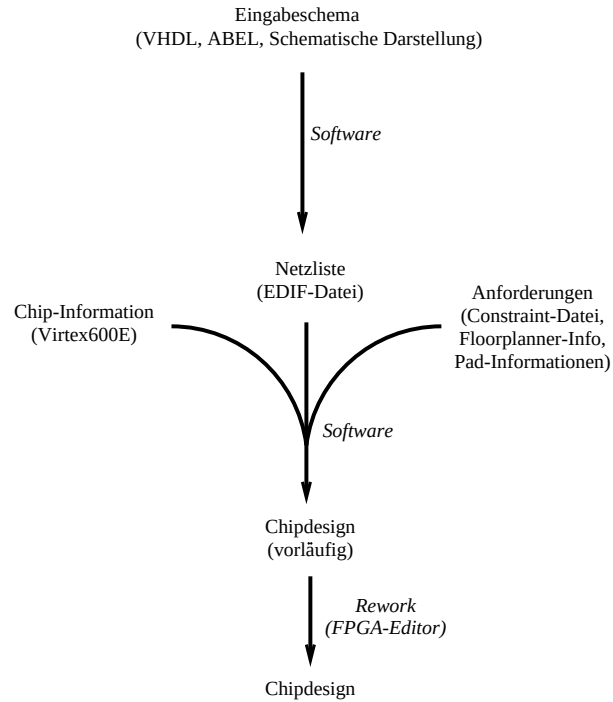


Abbildung A.6: Der Implementierungsprozess.

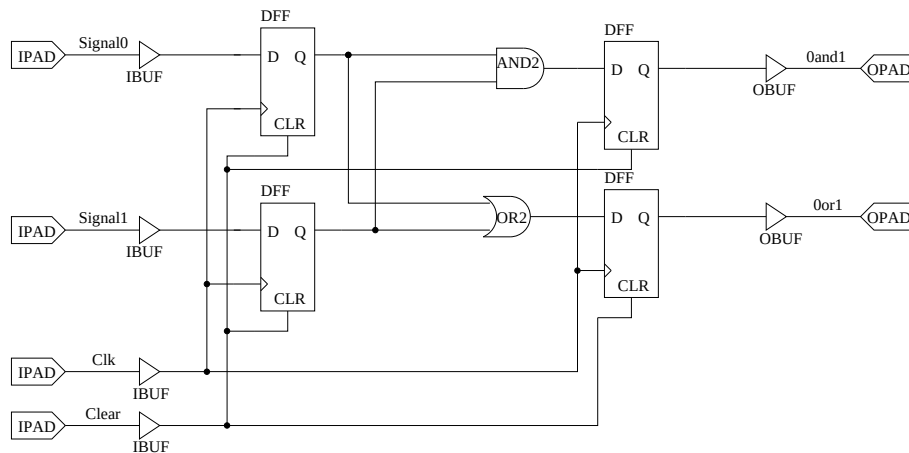


Abbildung A.7: Schaltplan eines einfachen Designs.

Methode ist eine funktionelle Beschreibung mittels einer Hardware Description Language (HDL). Gängige HDLs sind z.B. VHDL² und ABEL³ (Abbildung A.8). HDLs bieten den Vorteil, dass komplizierte Designs, die zahlreiche CLBs füllen, mit wenig Aufwand erstellen werden können. Das Design eines 32 bit-Addierers auf LUT-Ebene zu formulieren, würde sicher Stunden in Anspruch nehmen. In einer HDL sind es wenige Zeilen. Die direkte Eingabe des Schaltplans bietet dagegen den Vorteil, dass man, insbesondere durch den Einsatz von Registern, anschließend eine genaue Vorstellung davon hat, wie das Design im FPGA umgesetzt wird. Das Zeitverhalten kann dann z.B. sehr viel besser eingeschätzt werden.

Eingabe der Pin- und Timing-Vorgaben

Mit der Eingabe des Schaltplans bzw. der funktionellen Umschreibung ist es noch nicht getan. Die Implementierungssoftware braucht weitere Informationen. Wenn z.B. ein Clock-Signal benutzt wird, ist es wichtig, mit welcher Frequenz dieses später betrieben werden soll. Denn bei einem 200 MHz Design wird eine viel schnellere Logik benötigt als bei 10 MHz. Im *Place-and-Route*-Schritt muss die Software viel präziser arbeiten. Aus diesem Grund ist es sinnvoll, die Frequenz anzugeben.

Auch die physikalischen Informationen müssen der Software mitgeteilt werden. Der Designer muss der Software beispielsweise vorgeben, welche Signale an welche Pins gelegt werden sollen. Auch muss die Art der IO-Standards vorgegeben werden. Für diese Zwecke gibt es das UCF (User Constraint File, `filename.ucf`, Abbildung A.9).

Erstellung einer Netzliste

Es gibt eine Vielzahl weiterer Möglichkeiten, das Design zu konkretisieren. Auf alle kann aber hier nicht eingegangen werden. Wenn diese Eingaben getätigt sind, erstellt die Software aus dem Design eine Netzliste. D.h., die funktionelle Beschreibung des Designs wird optimiert und den Grundlagen des FPGAs angepasst. Die Netzliste beinhaltet schon die genaue Zahl der Leitungen, Flip-Flops und LUTs. Mit ihr kann jedoch nicht das Zeitverhalten vorausgesagt werden, weil nicht festgelegt wurde, wo sich die einzelnen Komponenten im FPGA befinden werden. Dies wird im nächsten Schritt getan.

Der „Place & Route“-Schritt

Dieser Schritt ist bei Designs, die den Platz auf dem FPGA voll ausnutzen, der zeitintensivste. Allen Komponenten wird hierbei ein Platz zugewiesen, und alle Ein- und Ausgänge werden mit Leitungen verbunden. Dieser Schritt ist alles andere als trivial, da jetzt auch die Anforderungen aus dem UCF mit einfließen. Diese Aufgabe würde auch einen erfahrenen Designer sicherlich einige Wochen kosten, wenn er sie

²Very High Speed Integrated Circuit (VHSIC) Hardware Description Language

³Advanced Boolean Expression Language

```

module decoder
Title 'decoder'
Declarations
jc3..jc0 PIN;                " Eingangssignale
jc = [jc0..jc3];
bin3..bin0 PIN istype 'com';
bin = [bin3..bin0];         " Ausgangssignale
err PIN istype 'com';

Equations
when (jc==0) then err=0;      " Fehlermeldung bei
else when (jc==^h0) then err=0; " undefinierten Zustaenden
else when (jc==^h8) then err=0;
else when (jc==^hc) then err=0;
else when (jc==^he) then err=0;
else when (jc==^hf) then err=0;
else when (jc==^h7) then err=0;
else when (jc==^h3) then err=0;
else when (jc==^h1) then err=0;
else err=1;

TRUTH_TABLE
([jc0,jc1,jc2,jc3] -> bin)    " Wahrheitstabelle
[0,0,0,0] -> 0;
[1,0,0,0] -> 1;
[1,1,0,0] -> 2;
[1,1,1,0] -> 3;
[1,1,1,1] -> 4;
[0,1,1,1] -> 5;
[0,0,1,1] -> 6;
[0,0,0,1] -> 7;

end decoder

```

Abbildung A.8: Die ABEL-Datei `decoder.abl` für einen 4-bit-Johnson-Counter-Dekoder.

```

INST AA IOSTANDARD=LVDS;  # Differenzielle LVDS-Signale
INST AB IOSTANDARD=LVDS;
INST AC IOSTANDARD=LVDS;
INST AD IOSTANDARD=LVDS;
INST AE IOSTANDARD=LVDS;

...

# Die gewuenschte Clock-Frequenz wird festgelegt.

NET "ACLK2" TNM_NET = "ACLK2";
TIMESPEC "TS_ACLK2" = PERIOD "ACLK2" 22.5 ns HIGH 50 %;
NET "TRIGGER2" TNM_NET = "TRIGGER2";
TIMESPEC "TS_TRIGGER2" = PERIOD "TRIGGER2" 16.5 ns HIGH 50 %;
NET "CLKAA" TNM_NET = "CLKAA";
TIMESPEC "TS_CLKAA" = PERIOD "CLKAA" 2.5 ns HIGH 50 %;
NET "CLKAB" TNM_NET = "CLKAB";
TIMESPEC "TS_CLKAB" = PERIOD "CLKAB" 2.5 ns HIGH 50 %;

...

NET "O<0>" LOC = "P108";      # die Signale werden an bestimmte
NET "O<1>" LOC = "P109";      # Pins gelegt.
NET "O<10>" LOC = "P127";
NET "O<11>" LOC = "P128";
NET "O<12>" LOC = "P130";
NET "O<13>" LOC = "P131";
NET "O<14>" LOC = "P132";
NET "O<15>" LOC = "P133";
NET "O<16>" LOC = "P139";

...

```

Abbildung A.9: Auszüge aus einem UCF (sc-jc4.ucf).

von Hand erfüllen müsste. Auch ein moderner PC kann dafür bei einem im Zeitverhalten anspruchsvollen Design auf einem XCV600E mehrere Tage benötigen. Wenn dieser Arbeitsgang beendet ist, verfügt die Software über alle Informationen, um zu entscheiden, ob die Timing-Vorgaben erfüllt werden können. Diese Informationen werden auch in einer Report-Datei ausgegeben. Das eigentliche Design ist hiermit fertig.

Erstellung einer Programmierdatei für den FPGA

Es gibt noch ein Problem: Die Design-Informationen müssen dem FPGA mitgeteilt werden. Zu diesem Zweck wird eine spezielle Datei erzeugt: ein bit-File. In dieser Datei steckt bitweise die Einstellung jedes Schalters und jeder LUT auf dem FPGA. Die Datei wird von der Software generiert und muss vom FPGA geladen werden. Hierzu gibt es mehrere Programmiermöglichkeiten. Die für die Testphase nützlichste ist die über eine Schnittstelle des PCs. Es werden zu diesem Zweck spezielle Kabel verwendet. Mit Hilfe dieser Kabel kann z.B. über die JTAG-Schnittstelle⁴ (Seite 45) das Bit-File an den FPGA geschickt werden. Später im Experiment kann auf diese Möglichkeit nicht mehr zurückgegriffen werden, weil ein FPGA immer, wenn die Spannungsversorgung abgeschaltet wird, sein Programm verliert. Daher bietet es sich später an, das Bit-File über ein PROM⁵ [30] an den FPGA zu schicken. Das PROM ist auch ein Bauteil auf der Scaler-CMC und kann im Gegensatz zum FPGA auch ohne Spannung die Datei speichern und so konfiguriert werden, dass es nach dem Einschalten seinen Speicherinhalt, das bit-File, an den FPGA schickt.

⁴Joint Test Action Group-Schnittstelle, standardisierte Test-Schnittstelle

⁵Programmable Read Only Memory

Anhang B

Der Schaltplan





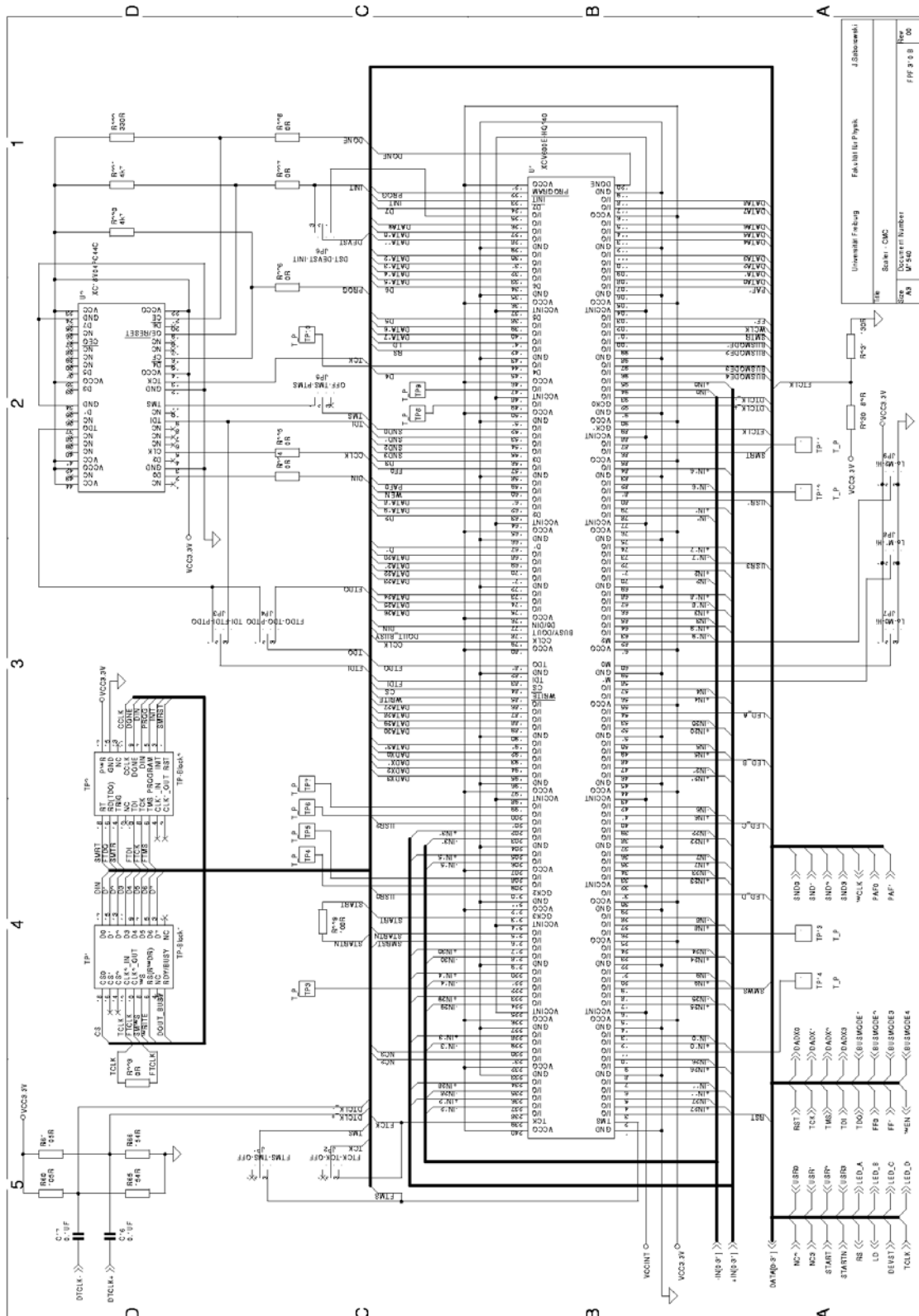


Abbildung B.3: Seite 3—FPGA, Testbuchsen und PROM.

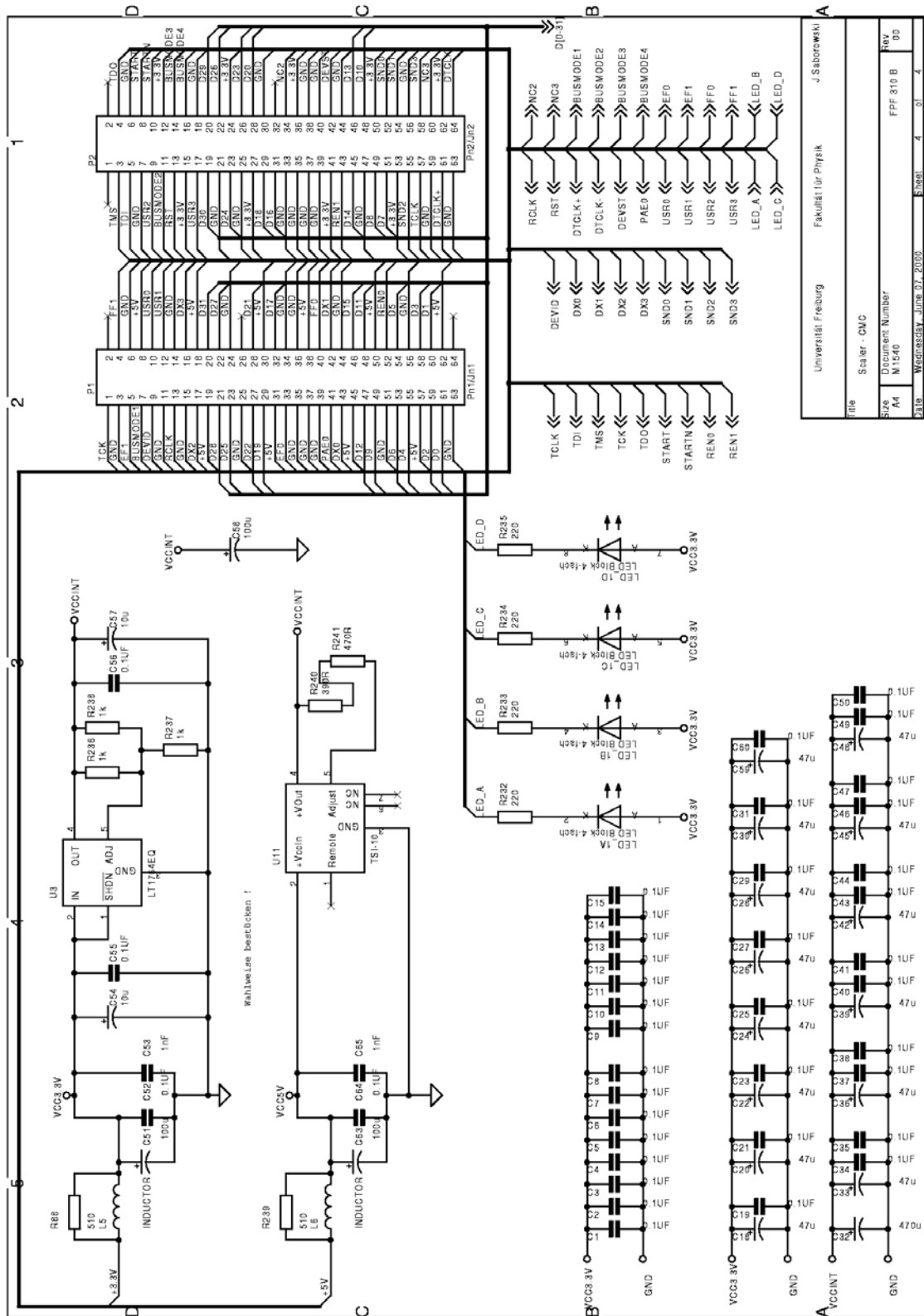


Abbildung B.4: Seite 4—CMC-Stecker, Spannungsversorgung und Kondensatoren.

Anhang C

Das Platinen-Design

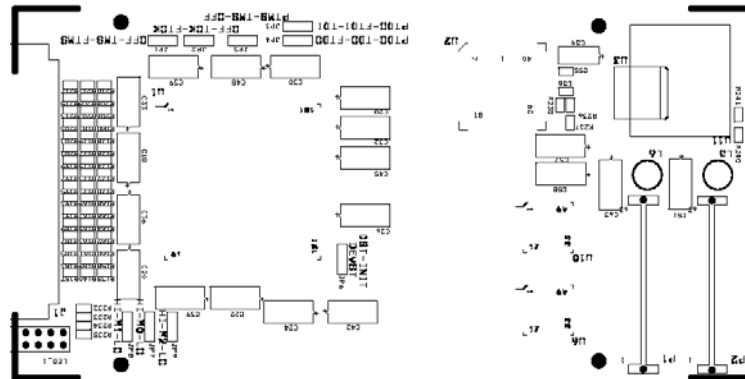


Abbildung C.1: Aufdruck der Bestückungsseite.

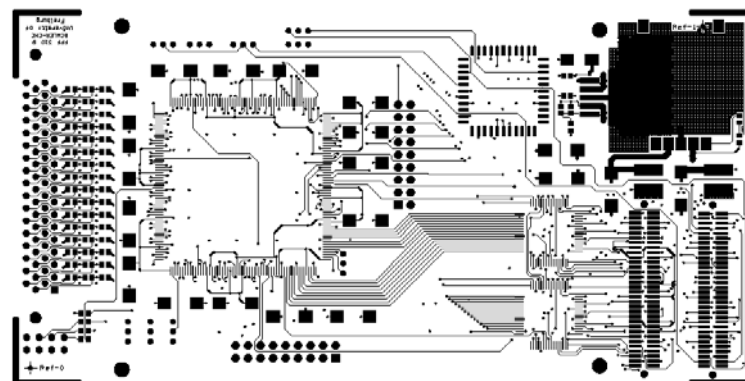


Abbildung C.2: Signalleitungen auf der Bestückungsseite.

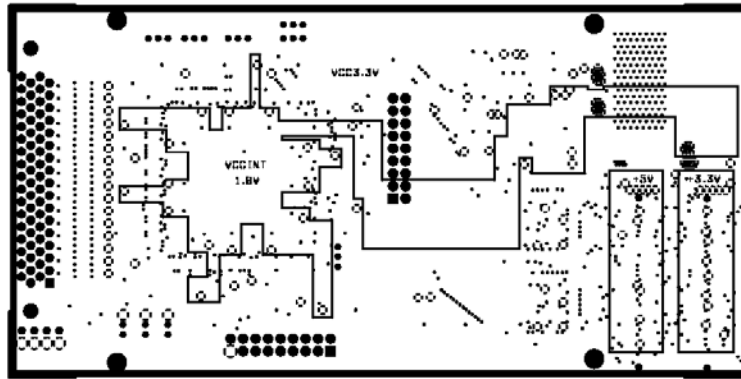


Abbildung C.3: Die Spannungsebene (3.3 V und 1.8 V), invertierte Darstellung.

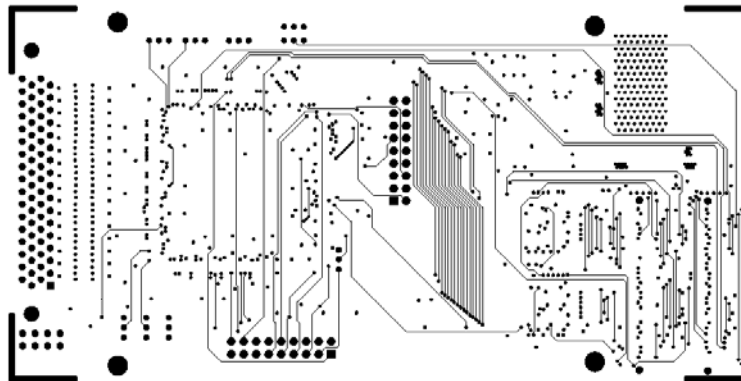


Abbildung C.4: Erste Signal-Innenlage.

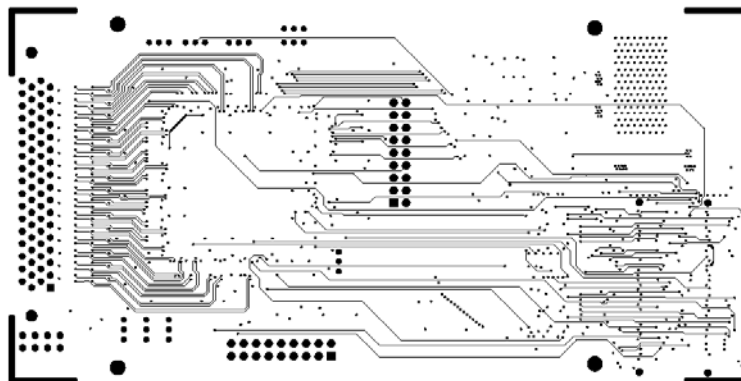


Abbildung C.5: Zweite Signal-Innenlage.

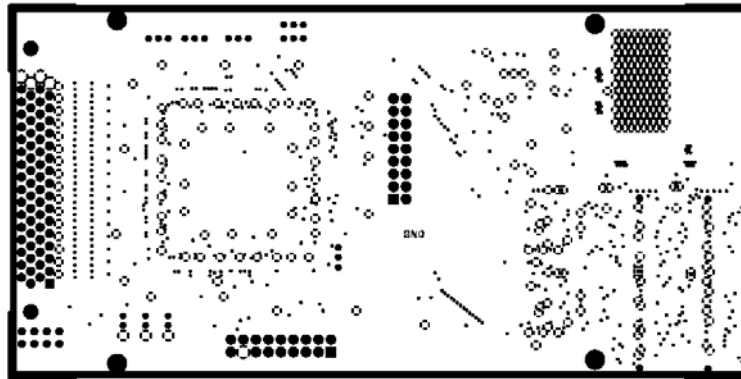


Abbildung C.6: Masseebene, invertierte Darstellung.

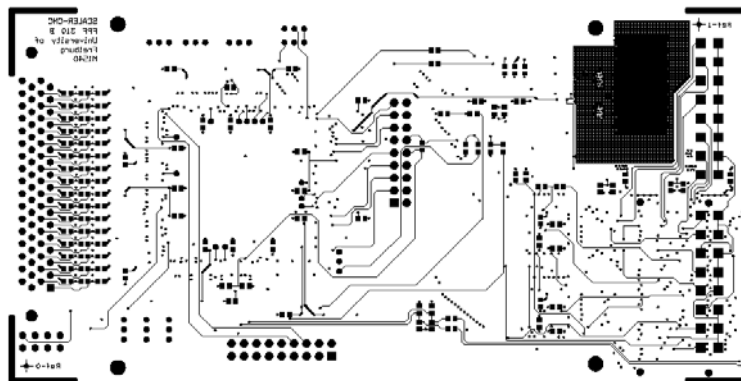


Abbildung C.7: Signalleitungen auf der Lötseite.



Abbildung C.8: Aufdruck der Lötseite.

Anhang D

Bilder



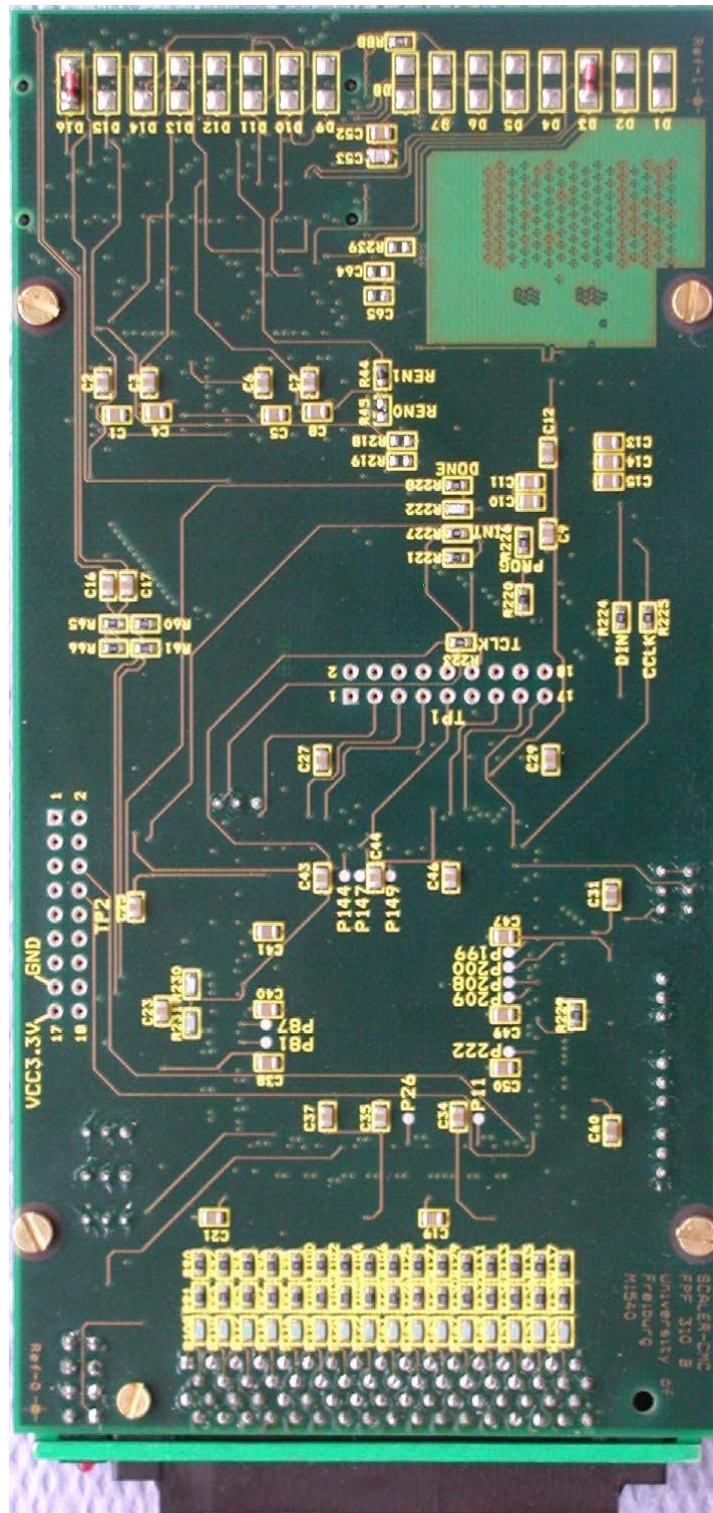


Abbildung D.2: Lötseite mit Testbuchsen.

Anhang E

Glossar

9U Größenangabe für VME-Module, 367 x 400 mm.

ABEL *Advanced Boolean Expression Language*, Programmiersprache zur Beschreibung von Hardware.

ADC *Analog-to-Digital-Converter*, wandelt Pulsformen in digitale Signale um.

ALICE *A Large Ion Collider Experiment*, zukünftiges CERN-Experiment am LHC.

AND-Gatter fundamentales, logisches Gatter, verknüpft seine Eingangssignale durch die Operation „UND“.

Apertur Kleine Öffnung am Spektrometermagneten.

ASIC *Application-Specific Integrated Circuit*, ein statischer integrierter Schaltkreis.

Bestückungsseite Die Seite der Platine, auf der sich die Bauteile befinden.

Bit Kleinste Einheit der Information.

Bit-File Das File zum Programmieren eines FPGAs.

Block SelectRAM Zusätzlicher Speicher auf dem Virtex-E-FPGA.

BMS *Beam Momentum Station*, Detektor der durch Ortsmessungen an den Spektrometermagneten die Impulse der Strahlteilchen bestimmt.

BOB *Begin Of Burst*, Signal, das den Beginn eines Spills ankündigt.

Boundary Scan Testverfahren für die Verbindungen zwischen Chips.

Burst Synonym für Spill.

CEDAR Ein auf bestimmte Teilchengeschwindigkeiten sensibilisierter Čerenkov-Counter.

CATCH *COMPASS Accumulate, Transfer & Control Hardware*, Herzstück der COMPASS-Auslese-Elektronik

Catch-o-matic Grafische Benutzeroberfläche für den VME-Zugriff auf das CATCH.

Catch-X Testversion des CATCH

CDR *Central Data Recording*, zentrale Datennahme des CERNs.

CE *Clock-Enable*, aktiviert den Clock-Eingang eines Flip-Flops.

CERN *Conseil Européen pour la Recherche Nucléaire*, Europäisches Forschungszentrum für Teilchenphysik in Genf.

Cheops *CHarm Experiment with Omni-Purpose Setup*,

CLB *Complex Logic Blocks*, Herzstück des FPGAs.

CLK Abkürzung für Clock.

Clock Periodisches, rechteckiges Signal für sequenzielle Logik, erzeugt von einem Oszillator.

CLR Abkürzung für *Clear*

CMC *Common Mezzanine Card*, Aufsteckkarte für das CATCH-Modul

COMPASS *COmmon Muon and Proton Apparatus for Structure and Spectroscopy*, Experiment am CERN, für das der Zähler entwickelt wurde.

CPLD *Complex Programmable Logic Device*, Programmierbarer Logikbaustein auf Basis von UND- und ODER-Gattern.

CPU *Central Processing Unit*, Prozessor.

Crystall-Barrel Experiment am CERN.

DATE *Data Acquisition Test Environment*, Datennahme-Programm des ALICE-Experiments.

DC/DC-Wandler Gleichstrom-Gleichstrom-Spannungswandler.

DLL *Delay Locked Loop* Bauteil eines Virtex-E zum Ausgleich der Clock-Laufzeiten.

DTCLK , vom TCS verteilte 38.88-MHz-Clock, differenzielles Signal.

EDIF Format für die Netzliste.

EMC Experiment am CERN.

EOB *End Of Burst*, Signal, das das Ende eines Spills ankündigt.

F1-TDC *F1-Time-To-Digital-Converter*, wandelt Zeitdifferenzen in digitale Daten um.

F1-TDC-CMC Aufsteckkarte für das CATCH mit vier F1-TDCs.

FIFO *First In—First Out*-Speicher.

Flip-Flop Logische Speicherzelle (1 bit).

FPGA *Field Programmable Gate Array*, programmierbarer Logik-Baustein.

Front-End-Karte Elektronische Schnittstelle zu den Detektoren. Dort werden die Daten digitalisiert.

Gate Signal, das den Zähler anhalten kann.

Gigabit-Ethernet Netzwerkstandard.

GND *Ground*, Masse, Erde.

Gray-Code Umordnung der Binärzahlen.

Hardware Elektronikkomponenten aller Art.

Header Leitet eine Datensequenz ein.

HDL *Hardware Description Language*, Oberbegriff für Programmiersprachen zur Beschreibung von Hardware.

HMC Eines der Ursprungs-Experimente von COMPASS.

HOTFibre-CMC Aufsteckkarte für das CATCH zur Datenübertragung via Glasfaser mit HOTLink-Protokoll.

HOTLink Protokoll zur Datenübertragung via Twisted-Pair-Kabel oder Glasfaser.

- HOTLink-CMC** Aufsteckkarte für das CATCH zur Datenübertragung via Twisted-Pair-Kabel mit HOTLink-Protokoll.
- IBUF** *Input-Buffer* Element zur schematischen Eingabe einer Hardware-Beschreibung.
- ID** Identifikationsnummer.
- Initial Set-up** Aufbau des COMPASS-Experiments zur Bestimmung der Gluonpolarisation.
- Input-Register** Speichert 32 Werte mit jedem Trigger.
- Inverter** Elementares Logik-Gatter.
- IO** Oberbegriff für Ein- und Ausgänge.
- IOB** *Input-/Output-Buffer* Schnittstelle des FPGAs zwischen interner Logik und Pins.
- JTAG** *Joint Test Action Group*, Standard-Schnittstelle für Boundary-Scan.
- Johnson-Counter** Invertierter Ringzähler.
- LC** *Logic Cell*, Grundbaustein des FPGAs.
- LHC** *Large Hadron Collider* Zukünftiges CERN-Experiment.
- Linearregler** wandelt Gleichspannungspegel ineinander um.
- Lötseite** Die Seite der Platine, auf der sich die Lötstellen der Bauteile befinden.
- LSB** *Least Significant Bit*, Bit mit kleinstem Stellenwert.
- LUT** *Look-Up Table*, kombinatorisches Element im FPGA.
- LVDS** *Low Voltage Differential Signal*, Differenzieller Signalstandard.
- LVTTL** *Low Voltage Transistor Transistor Logic*, Signalstandard.
- LVPECL** Differenzieller Signalstandard.
- M2-Beamline** Strahlführung zwischen SPS und COMPASS-Experiment.
- monTDC** Programm zur Auswertung der TDC-Daten.
- monSCALER** Programm zur Auswertung der Zähler-Daten.
- MSB** *Most Significant Bit*, Bit mit höchstem Stellenwert.
- MUX** Multiplexer.
- OBUF** *Output-Buffer* Element zur schematischen Eingabe einer Hardware-Beschreibung.
- ODER-Gatter** fundamentales, logisches Gatter, verknüpft seine Eingangssignale durch die Operation „ODER“.
- OR-Gatter** fundamentales, logisches Gatter, verknüpft seine Eingangssignale durch die Operation „ODER“.
- PAL** Vorgänger des CPLDs
- PCI** *Peripheral Component Interconnect*, Schnittstelle zum Verbinden der Peripheriegeräte eines PCs.
- PLA** Vorgänger des CPLDs.
- PROM** *Programmable Read-Only Memory*, Speicher zum Programmieren des FPGA. Behält seinen Speicherinhalt auch ohne Spannung.
- Proposal Set-up** Geplanter Aufbau des COMPASS-Experiments in der Endausbaustufe.
- Pulser** Pulsgenerator.

- Quark-Parton-Modell** Grundlegendes Modell der Physik der Nukleonen.
- RAM** *Random Access Memory*, Arbeitsspeicher.
- Reset** Signal zum Zurücksetzen.
- ROB** *Read-Out Buffer*, Zwischenspeicher der COMPASS-Auslese-Elektronik.
- Routing** Die Gesamtheit der Leitungen im FPGA.
- Scaler** Englisches Synonym für Zähler.
- Scaler-CMC** Zähler-Aufsteckkarte.
- SciFi** Abkürzung für die Szintillierenden Fasern.
- Slice** Die Hälfte eines CLBs. Besteht aus zwei Logic Cells (LC).
- S-LINK** CERN-Protokoll zur optischen Datenübertragung.
- SMC** *Spin Muon Collaboration*, Experiment am CERN.
- Software** Programme für PCs.
- Spill** Zeitraum in dem der Teilchenstrahl auf das COMPASS-Target trifft,
- SPLD** Vorgänger des CPLDs
- Spot** Strahlfleck, Durchmesser des Strahls.
- SPS** *Super Proton Synchrotron accelerator*, Beschleuniger am CERN.
- Spy-Buffer** FIFO auf dem Catch-X, das über die VME-Schnittstelle ausgelesen werden kann.
- Target** Anordnung auf die der Teilchenstrahl trifft.
- TC** *terminal count*, Signal eines Vorzählers nach jedem Durchlauf.
- TCLK** Clocksignal vom CATCH.
- TCS** *Trigger-Control-System*, Trigger-Kontroll-System.
- TDC** siehe F1-TDC.
- Trailer** Beendet eine Datensequenz.
- Tristate-Buffer** Puffer mit einem dritten Zustand („floating“).
- Twisted-Pair-Kabel** Verdrilltes zweiadriges Kabel.
- UCF** *User Constraint File*, Datei mit Routing-Anweisungen für das Synthese-Programm.
- UND-Gatter** fundamentales, logisches Gatter, verknüpft seine Eingangssignale durch die Operation „UND“.
- Vint** interne Spannung des FPGAs.
- Vcco** Spannung zum Betrieb der IOs des FPGAs.
- VHDL** *Very High Speed Integrated Circuit (VHSIC) Hardware Description Language*, eine moderne HDL.
- Virtex-E** Moderner FPGA der Firma Xilinx.
- VME-Crate** VME-Überrahmen, in diesem werden die Module und der VME-Rechner eingeschoben.
- VME-Rechner** CPU nach dem VME-Standard.
- VME-Modul** Platine für einen VME-Rechner.
- WA89** Experiment am CERN.
- WE** *Write-Enable*, Signal, das das Schreiben in ein FIFO ermöglicht.
- XC4000XLA** FPGA-Typ von Xilinx.
- XCV600E** Virtex-E-FPGA-Typ von Xilinx.

Literaturverzeichnis

- [1] *Semi-inclusive Muon Scattering from a polarised Target* E. Nappi *et al.*, in *Letter of Intent CERN/SPSLC/95-27, SPSLC/I204 (1995)*.
- [2] *CHEOPS: CHarm Experiment with Omni-Purpose Setup*, Yu. A. Alexandrov, *et al.*, in *Letter of Intent CERN/SPSLC/95-22, SPSLC/I202 (1995)*.
- [3] *The Spin Structure of the Nucleon* G. Mallot, *Habilitationsschrift, Johannes Gutenberg-Universität Mainz, August 1996*.
- [4] *Hyperon physics with a tagged hyperon beam at the CERN omega spectrometer: Letter of intent*, C. Berat *et al.*, CERN-SPSLC-94-10.
- [5] COMPASS: A Proposal for a Common Muon and Proton Apparatus for Structure and Spectroscopy, G. Baum *et al.* [COMPASS Collaboration], *CERN-SPSLC-96-14*.
- [6] F. Halzen, *et al.*, Quarks and Leptons, John Wiley and Sons (WIE), 1984.
- [7] High statistics study of $f_0(1500)$ decay into $\pi^0\pi^0$, C. Amsler *et al.*, *Phys. Lett.* **B342** (1995) 433.
- [8] *The Modifications to the M2 Beam for COMPASS*, L. Gatignon, <http://sl.web.cern.ch/SL/eagroup/NewM2/main.html>, preliminary, 11 August 2000 .
- [9] *Teilchen und Kerne*, B. Povh, *et al.*, Springer 1999.
- [10] *The SMC polarized target: Systems and operations*, D. Kramer [Spin Muon Collaboration], *Prepared for 11th International Symposium on High-energy Spin Physics and the 8th International Symposium on Polarization Phenomena in Nuclear Physics (SPIN 94), Bloomington, Indiana, 15-22 Sep 1994*.

- [11] *The gluon contribution to the nucleon spin and the COMPASS experiment at CERN*, F. Bradamante, Prog. Part. Nucl. Phys. **44** (2000) 339.
- [12] *COMPASS DAQ Pages*, COMPASS Collaboration, <http://wwwcompass.cern.ch/compass/detector/daq/>, 09/02/99.
- [13] *Statusbericht zum Datennahmesystem des COMPASS Experiments am CERN*, I. Konorov, *et al.*, Statusbericht für das BMBF, S.4, 1999/2000.
- [14] *Implementation of the dead-time free F1 TDC in the COMPASS detector readout*, H. Fischer *et al.*, hep-ex/0010065.
- [15] CYPRESS Semiconductor Corporation, *CY7B923/33 HOTLink Transmitter/Receiver Data Sheet*.
- [16] *CATCH and CMC-HOTLink Readout Driver Specification*, G. Braun, *et al.*, COMPASS note 1999-15.
- [17] *IEEE 1014-1987*, IEEE Standard for A Versatile Backplane Bus: VME-bus 1981.
- [18] *S-LINK: A prototype of the ATLAS read-out link*, E. van der Bij, R. McLaren and Z. Meggyesi, *Given at 4th Workshop on Electronics for LHC Experiments (LEB 98), Rome, Italy, 21-25 Sep 1998*.
- [19] *Draft Standard for CATCH Mezzanine Cards: CATCH/OPEN*, G. Braun, *et al.*, 28.4.1999 (preliminary).
- [20] *Using the Virtex SelectI/O Resource*, Xilinx Application Note: Virtex Series, XAPP133 (v2.5) September 7, 2000.
- [21] *Standard for Description of Low Voltage TTL-Compatible CMOS Logic Devices*, EIA/JESD53, November 1995.
- [22] *CATCH-X Users Manual*, H.Fischer *et al.*, COMPASS note 2000-12.
- [23] *XC4000XLA/XV Field Programmable Gate Arrays*, Xilinx Data Sheet, May 14, 1999 (Version 1.2).
- [24] *The practical Xilinx Designer Lab Book*, D. Van den Bout, Prentice Hall, 1997.
- [25] *Contemporary Logic Design*, Randy H. Katz, The Benjamin/Cummings Publishing Company 1994.
- [26] *Pulse Code Communication*, F. Gray, U. S. Patent 2 632 058, March 17, 1953.

- [27] *Mathematical Games*, Martin Gardner, Scientific American v.227,n.2 August, 1972 p.106.
- [28] *Development System Reference Guide 2.1i*, Xilinx, Inc., 1999, http://support.xilinx.com/support/sw_manuals/2_1i/download/dev_ref.pdf.
- [29] *Foundation Series 2.1i User Guide*, Xilinx, Inc., 1999, http://support.xilinx.com/support/sw_manuals/2_1i/download/fsuguide.pdf.
- [30] *XC18V00 Series of In-System Programmable Configuration PROMs*, DS026 (v2.3) July 5, 2000, Product Specification.
- [31] FPF¹ 1540, Projektnummer der Elektro-Werkstatt.
- [32] *DC/DC Converter TSI-10 Series 10 Watt*, Traco Power, www.tracopower.com, Rev. 10/00.
- [33] *LT1764 Series 3A, Fast Transient Response, Low Noise, LDO Regulators*, Final Electrical Specifications, February 2000.
- [34] *Integration einer Kontroll- und Ausleseseinheit in das COMPASS-Experiment*, L. Hennig, August 2000.
- [35] *FD-based Shift Register V1.0.3*, Xilinx, Inc., December 17, 1999 .
- [36] *First In First Out memory queue with independent read and write clocks (ASYNCRONOUS FIFO)*, (BETA Xilinx FAE Release), October 15, 1999.
- [37] *CY7C4225*, Cypress Data Sheet, 1997.
- [38] *Designing Front-End Boards for use with the CATCH-HOTLink Interface*, G.Braun,et al., COMPASS note 1999-7.
- [39] *CatchX-o-Matic User Manual*, L. Hennig, 1999.
- [40] *Alice in Wonderland*, L. Carroll, W.W.Norton 1992.
- [41] *ALICE DATE User's Guide*, CERN ALICE DAQ group, ALICE 99/46 Internal Note/DAQ, December 1999.
- [42] Projektion auf die x-Achse: Run 2843, Spill 4,
Projektion auf die y-Achse: Run 2852, Spill 1,
COMPASS-Strahlzeit, September 2000

¹Fakultät für Physik Freiburg

- [43] Run 1003, Spill 5, *COMPASS-Strahlzeit*, August 2000
- [44] *Virtex-E 1.8 V Field Programmable Gate Arrays*, Xilinx DS022 (v1.7) September 20, 2000.

Danksagung

Ich möchte mich bei allen, die mich bei dieser Arbeit unterstützt haben, herzlich bedanken. Ich habe in den letzten $1\frac{1}{2}$ Jahren sehr viel gelernt und hatte dabei eine schöne Zeit.

Ich danke

- Herrn Prof. Kay Königsmann für die Vergabe und Betreuung der Arbeit.
- Herrn Dr. Horst Fischer und Herrn Dr. Fritz-Herbert Heinsius, die sich sehr viel Zeit nahmen, um mir zu helfen, und auf jede Frage (mindestens) eine Antwort hatten.
- Lars Hennig, Thomas Schmidt und besonders Andreas Grünemaier, die mich mit der Welt der FPGAs vertraut machten.
- allen Abteilungsmitgliedern für das gute Klima und die große Hilfsbereitschaft.
- Herrn Jürgen Saborowski aus der Elektro-Werkstatt für die Umsetzung der Zählerplatine.
- dem Post-Script-König Stephan Brauksiepe für die T_EXnische Hilfe.
- allen, die für mich Korrektur gelesen haben, insbesondere bei Yu-Kai The, Matthias Hoffmann und meinem Bruder Dirk, der in einer Nacht-und-Nebel-Aktion alle restlichen Fehler entfernt hat.
- meiner Freundin Christine Funke für ihre Unterstützung außerhalb der Welten der Myonen und Nukleonen sowie Bits und Bytes.

Ganz besonders danken möchte ich meinen Eltern, ohne deren Unterstützung diese Arbeit sicher nicht existieren würde.

Erklärung

Diese Arbeit ist von mir selbstständig verfasst worden, und ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt.