

Digitaler elektronischer Entzerrer für die optische Datenübertragung mit bis zu 43 Gbit/s

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

vorgelegt von

Thomas Veigel

aus Stuttgart

Hauptberichter: Prof. Dr.-Ing. Manfred Berroth
Mitberichter: Prof. Dr.-Ing. Joachim Speidel

Tag der mündlichen Prüfung: 23. März 2012

Institut für Elektrische und Optische Nachrichtentechnik
der Universität Stuttgart

2012

Inhaltsverzeichnis

Abkürzungen	vii
Formelzeichen.....	xi
Abstract.....	xv
1 Einleitung.....	1
1.1 Optische Kommunikation	2
1.2 Technologie und Architektur	3
1.3 Stand der Technik.....	5
1.3.1 Optische Übertragungssysteme	5
1.3.2 Hardware-Entwürfe verschiedener Viterbi-Dekoder	5
1.4 Aufbau dieser Arbeit	6
2 Entzerrung optischer Übertragungskanäle	7
2.1 Dispersionseffekte	7
2.1.1 Polarisationsmodendispersion	8
2.1.2 Chromatische Dispersion	9
2.2 Entzerrerkonzepte.....	10
2.2.1 Vorwärtsentzerrer.....	10
2.2.2 Vorwärtsentzerrer und entscheidungsrückgekoppelter Entzerrer	10
2.2.3 Wahrscheinlichkeitsbasierter Entzerrer.....	11
2.2.4 Bewertung der Entzerrerkonzepte.....	12
2.3 Der Übertragungskanal	12
2.4 Schätzung der wahrscheinlichsten Empfangsfolge	16
2.4.1 Viterbi-Algorithmus	17
2.4.1.1 Blockdekodierverfahren	19
2.4.1.2 Parallelisierung.....	20
2.4.1.3 Schiebefensterverfahren	22
2.4.1.4 Der systolische Blockdekoder.....	25
3 Logikentwurf mittels komplementärer Metalloxid-Halbleiter-Technologie	29
3.1 Statische CMOS-Logik	30
3.2 Transfertransistor- und Transfergatterlogik	37
3.3 Speicher- und Kippschaltungen	41
3.3.1 Statische CMOS-Speicherzelle	42
3.3.2 Dynamische CMOS-Speicherzelle.....	48
3.3.3 D-Flipflop.....	49
3.3.4 Schmitt-Trigger	53
3.4 Stromschalterlogik	55
3.4.1 CML-Verstärker	57

3.4.2	Schaltung zur Arbeitspunkteinstellung	61
3.4.3	CML-Latch und CML-Flipflop	62
3.4.4	CML-Multiplexer	64
3.5	Schnittstellen zwischen CML- und CMOS-Logik	65
3.5.1	Wandlung von CML- zu CMOS-Pegeln	65
3.5.2	Wandlung von CMOS- zu CML-Pegeln	67
4	Hardware-Architektur des Viterbi-Entzerrers	69
4.1	Zweigmetriekrechner	69
4.1.1	Anforderungen an den Speicher	70
4.1.2	Vergleich der Speichertechnologien	70
4.1.3	Architektur des Zweigmetriekrechners	71
4.1.4	Der zentrale Speicher	72
4.1.5	Adressdekoder	76
4.1.6	Synchronisation der Metriken beim Einschreiben	78
4.1.7	Verteilung des Speicherinhaltes und Übersprechen	81
4.1.8	Auslesen des Speichers	85
4.1.9	Einschreiben der Metriken	86
4.1.10	Übersicht über die realisierte Zweigmetriekrecheneinheit	88
4.2	Addier-Vergleichs-Auswahleinheit	90
4.2.1	Der Volladdierer	91
4.2.2	Addierer mit sukzessiver Übertragskorrektur	94
4.2.3	Addierer mit teilsukzessiver und teilparalleler Übertragskorrektur	96
4.2.4	Addierer ohne Übertragskorrektur	97
4.2.5	Paralleladdierer mit Übertragsvorausberechnung	98
4.2.5.1	Parallele Übertragskorrektur	98
4.2.5.2	Baumstruktur	99
4.2.6	Komparator	100
4.2.6.1	Linearer Komparator	101
4.2.6.2	Baumförmiger Komparator	103
4.2.7	Bewertung der Addierverfahren	103
4.2.8	Hardwaretechnische Umsetzung der Addier-Vergleichs-Auswahleinheit	104
4.3	Minimumauswahlschaltung	110
4.4	Pfadrückverfolgungseinheit	115
4.5	Eingangsdemultiplexer	118
4.5.1	Hintergrundüberlegungen zum Demultiplexer	118
4.5.2	Aufbau eines CML-Demultiplexers	119
4.5.3	Taktteiler	121
4.5.4	Vom 1:2- zum 1:8-Demultiplexer	122
4.5.5	Taktteilung und Taktverteilung	127
4.5.6	Gesamtüberblick Demultiplexer	130
4.6	Ausgangsmultiplexer	131
4.6.1	Aufbau eines CML-Multiplexers	131
4.6.2	8:1 Multiplexer	132
4.6.3	Gesamtüberblick Multiplexer	135
4.7	Gesamtarchitektur des Viterbi-Entzerrers	136
4.7.1	Globale Taktverteilung	139
4.7.2	Globale Verdrahtung	140

4.7.3	Spannungsversorgung und Stromaufnahme	141
4.7.4	Ein- und Ausgangsschnittstellen für Hochgeschwindigkeitsdaten	143
4.7.5	Digitale Konfigurations- und Statistikschnittstelle	143
5	Chipentwurf und Chipfertigung	145
5.1	Design Kit	145
5.2	Chipentwurf	146
5.2.1	Vorgehensweise	146
5.2.2	Zeitlicher Rahmen der Chipentwicklung	148
5.2.3	Eingesetzte Software-Werkzeuge	149
5.3	Chipfertigung	150
6	Messaufbau und Messergebnisse	155
6.1	Aufbautechnik	155
6.2	Funktionstest	156
6.2.1	Messaufbau	157
6.2.2	Messergebnisse	158
6.3	Hochgeschwindigkeitsmessungen	159
6.3.1	Vorbereitende Maßnahmen und Messaufbau	160
6.3.2	Messergebnisse	162
6.3.3	Grenzen des Messaufbaus	165
6.4	Einordnung der Messergebnisse dieser Arbeit	166
7	Automatische Anpassung an den Kanal	169
7.1	Randbedingungen für eine Kanalschätzung	169
7.2	Grundlagen der Kanalschätzung	170
7.3	Bestimmung der Metriken mit Hilfe der Histogrammmethode	173
7.4	Implementierung und Simulation	175
8	Zusammenfassung und Ausblick	177
8.1	Zusammenfassung	177
8.2	Ausblick	180
8.2.1	Verbesserung der Messumgebung	180
8.2.2	Weiterentwicklung des Viterbi-Entzerrers	180
8.2.3	Erhöhung der Rechengeschwindigkeit	181
8.2.4	Konzepte für noch höhere Datenraten	181
	Liste eigener Veröffentlichungen	183
	Literaturliste	184
	Danksagung	190
	Lebenslauf	191

Abkürzungen

Abkürzung	Bedeutung
ACS	Add-Compare-Select
ACSU	Add-Compare-Select-Unit (Addier-Vergleichs-Auswahleinheit)
AD	Analog-Digital-(Wandler)
ADS	Advanced Design System (Software-Werkzeug von Agilent zum Schaltungs-entwurf von Hochfrequenzanwendungen)
ASE	Amplified Spontaneous Emission
BMU	Branch-Metric-Unit (Zweigmetrikrecheneinheit)
BSIMx	Berkeley Short-channel IGFET Model, x ... Versionsnummer (MOSFET-Modell, das von der „University of California“ in Berkeley stammt)
CD	Chromatische Dispersion
CLA	Carry-Lookahead-Adder (Paralleladdierer mit Übertragsvorausberechnung)
clk	Clock (Taktsignal in Blockschaltbildern)
CML	Current-Mode-Logic (Stromschalterlogik)
CMOS	Complementary Metal Oxide Semiconductor
CMP	Circuits Multi-Projets
CS	Carry-Save
CSA	Carry-Save-Adder (Paralleladdierer ohne Übertragskorrektur)
CSelA	Carry-Select-Adder (Paralleladdierer mit teilsukzessiver und teilparalleler Übertragskorrektur)
CSU	Compare-Select-Unit
DA	Digital-Analog-(Wandler)
DEMUX	Demultiplexer
DFE	Decision Feedback Equalizer (Entscheidungsrückgekoppelter Entzerrer)
DGD	Differential Group Delay (Differenzielle Gruppenlaufzeit)
DMD	Differential Mode Dispersion (Differenzielle Modendispersion)
DRAM	Dynamic Random Access Memory
DRC	Design Rule Check (Überprüfung der geometrischen Entwurfsregeln)
DVB	Digital Video Broadcast
FEC	Forward Error Correction (vorwärtsgerichtete Fehlerkorrektor im Empfänger)
FF	Fast/Fast (schnelle n-Kanal- und p-Kanaltransistormodelle)
FFE	Feedforward Equalizer (Vorwärtsentzerrer, Transversalentzerrer)
FIR	Finite Impulse Response (endliche Impulsantwort)
FOM	Figure of Merit
FPGA	Field Programmable Gate Array

Abkürzung	Bedeutung
GDSII	Graphical Design Station II oder Graphic Data System II (Standarddatenformat für den Austausch von Layoutdaten integrierter Schaltungen)
GND	Ground (Massepotenzial)
GP	General Purpose (Transistoren)
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HA	Halbaddierer
HVT	High threshold voltage (Transistoren mit hoher Schwellenspannung)
IC	Integrated Circuit (Integrierte Schaltung)
ICI	Inter Channel Interference (Nachbarkanalnebensprechen)
IDFT	Inverse Discrete Fourier-Transform (Inverse Diskrete Fourier-Transformation)
IFFT	Inverse Fast Fourier-Transform (Inverse Schnelle Fourier-Transformation)
IGFET	Insulated Gate Field Effect Transistor (Feldeffekttransistor mit einer isolierten Steuerelektrode)
IPTV	Internet Protocol Television (Fernsehen über eine Internetverbindung)
ISI	Intersymbol Interference (Intersymbolinterferenz, Symbolnebensprechen)
L_x	Latch mit Indexnummer x
LP	Low Power (Transistoren)
LTE	Long Term Evolution
LVS	Layout versus Schematic Check (Überprüfung der Übereinstimmung zwischen Layout und Schaltplan)
LVT	Low threshold voltage (Transistoren mit niedriger Schwellenspannung)
MAN	Metropolitan Area Network
MGT	Multi-Gigabit-Transceiver
ML	Maximum Likelihood (größte Wahrscheinlichkeit)
MLSE	Maximum Likelihood Sequence Estimation
MM	Minimized Method (Minimalmethode)
MMF	Multimode Fiber
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor (Metall Oxid Halbleiter Feldeffektransistor)
MUX	Multiplexer
NAND	Verneinte Und-Funktion/Gatter
NOR	Verneinte Oder-Funktion/Gatter
NRZ	Non Return to Zero (Binärkodierungsform, bei der kein Signalpegelwechsel innerhalb eines Bits auftritt)
OFDM	Orthogonal Frequency Division Multiplex (Orthogonaler Frequenzmultiplex)
PC	Personal Computer

Abkürzung	Bedeutung
PCB	Printed Circuit Board (Leiterplatte)
PCI	Peripheral Component Interconnect (Standard-Bus-Schnittstelle um Peripheriegeräte mit dem Chipsatz eines Prozessors zu verbinden)
PEX	Parasitic Extraction (Extraktion parasitärer Verdrahtungswiderstände und Verdrahtungskapazitäten)
PMD	Polarisationsmodendispersion
PROM	Programmable Read Only Memory
QAM	Quadraturamplitudenmodulation
RCA	Ripple-Carry-Adder (Paralleladdierer mit sukzessiver Übertragskorrektur)
RCMAX	Extraktion von parasitären Elementen mit maximalen Leitungswiderständen und Kapazitäten (3-Sigma-Abweichung vom typischen Wert)
RCTYP	Extraktion von parasitären Elementen mit typischen Leitungswiderständen und Kapazitäten
SATA	Serial Advanced Technology Attachment
SMA	Sub-Miniature-A (HF-Steckverbinder für 1 bis 18 GHz)
SMD	Surface Mountable Device
SMF	Singlemode Fiber
SMP	Sub-Miniature-P (kleiner HF-Steckverbinder für bis zu 40 GHz)
SMU	Select-Minimum-Unit (Minimumauswahlschaltung)
SPICE	Simulation Program with Integrated Circuits Emphasis (Simulationsprogramm mit Schwerpunkt auf integrierte Schaltungen)
SRAM	Static Random Access Memory
SS	Slow/Slow (langsame n-Kanal- und p-Kanaltransistormodelle)
SVT	Standard threshold voltage (Transistoren mit mittlerer Schwellenspannung)
TB	Trace-Back (Pfadrückverfolgung)
TBU	Trace-Back-Unit (Pfadrückverfolgungseinheit)
TG	Transferrgatter
TT	Typical/Typical (typische n-Kanal- und p-Kanaltransistormodelle)
UI	Unit Intervall (Einheitsintervall)
UMTS	Universal Mobile Telecommunications System
UV	Ultraviolett
VA	Volladdierer
VoD	Video-on-Demand (Filmabrufdienst)
VLSI	Very Large Scale Integration
VMA	Vector Merging Adder
WAN	Wide Area Network
WE	Write-Enable (Schreibfreigabe)
WIMAX	Worldwide Interoperability for Microwave Access

Abkürzung	Bedeutung
xDSL	Diverse „Digital Subscriber Line“-Übertragungsverfahren, wie z.B. Asymmetric (A) DSL, Symmetric (S) DSL, Very High Speed (V) DSL
XOR	Exklusiv-Oder-Funktion/Gatter

Formelzeichen

Formelzeichen: Lateinische Buchstaben

Variable	Einheit	Bedeutung
A		(erster) Summand / allg. Zahl
A		Eingangssignal/-knoten einer CMOS-Schaltung
A _i		Bit der Wertigkeit i einer Zahl (eines Summanden)
a _k		gesendetes Symbol
B		Dekodierblocklänge
B		Eingangssignal/-knoten einer CMOS-Schaltung
B		(zweiter) Summand / allg. Zahl
B _i		Bit der Wertigkeit i einer Zahl (eines Summanden)
C		Übertrag
C _i		Übertragsbit der Wertigkeit i
C' _{OX}	F/m ²	Gatekapazitätsbelag
C _L	F	Lastkapazität
C _{db}	F	Sperrschichtkapazität zwischen Drain- und Bulk-Anschluss
C _j	F	Allgemeine Sperrschichtkapazität (engl. Junction)
C _{sb}	F	Sperrschichtkapazität zwischen Source- und Bulk-Anschluss
C _v	F	Kapazität einer Verbindungsleitung
c	m/s	Lichtgeschwindigkeit
D _{CD}	ps/(nm·km)	Dispersionskoeffizient der chromatischen Dispersion
D _{PMD}	ps/km ^{1/2}	Dispersionskoeffizient der Polarisationsmodendispersion
d _k		Gewichtungsfaktoren, Einfluss des Symbolnebensprechens
E	V/m	Elektrische Feldstärke
E _{sat}	V/m	Elektrische Feldstärke, bei der die Ladungsträger im MOSFET ihre Sättigungsgeschwindigkeit erreicht haben
f	Hz (=1/s)	Frequenz
f _{clk}	Hz (=1/s)	Taktfrequenz
G		Übertraggenerierendes Signal (engl. Generate)
G _i		Bit der Wertigkeit i des übertraggenerierenden Signals
g _m	A/V	Steilheit eines Transistors
H		Logischer High-Pegel eines Signals oder Taktes
I ₀	A	Strom durch den Stromquellentransistor einer CML-Schaltung
I _D	A	Drain-Strom eines MOSFETs
I _{Ref}	A	Referenzstrom
k		diskrete Zeit

Variable	Einheit	Bedeutung
L		Logischer Low-Pegel eines Signals oder Taktes
L		Länge des überlebenden Pfades
L	km	Länge einer Glasfaser
L	m	Kanallänge eines MOSFETs
L_{eff}	m	Effektive Kanallänge eines MOSFETs
$L_{\mu v}$		Maximum-Likelihood-Funktional
l		Kanalgrad, Kanalgedächtnis
l_b	m	Bitlänge
M		Stufigkeit des Eingangssignals
N_S		Anzahl der Zustände (S... Zustände, engl. States)
N_T		Anzahl der Zustandsübergänge (T... Übergänge, engl. Transitions)
n_n		Anzahl in Serie geschalteter n-Kanal-MOSFETs
n_p		Anzahl in Serie geschalteter p-Kanal-MOSFETs
P		Übertragweitergebendes Signal (engl. Propagate)
P_i		Bit der Wertigkeit i des übertragweitergebenden Signals
P_{dynC}	F	Kapazitive dynamische Verlustleistung
$P_{\mu, k}$		Pfadmetriken zum Zeitpunkt k
p		Schaltwahrscheinlichkeit
$p_{\mu v, k}$		Pfadkosten/Zweigmetriken zum Zeitpunkt k
q_k		Empfangssymbol
R	Ω	Elektrischer Widerstand
r_k		Rauschen
S		Steuersignal/-knoten einer CMOS-Schaltung
S		Summe / Summenpfadmetrik
S_i		Summenbit der Wertigkeit i / Summenpfadmetrik mit Index i
S_i		Zustand mit Index i in einem Trellis (i... dezimal oder binär)
t_b	s	Bitdauer
t_{C2O}	s	Clock-to-Output-Zeit eines Flipflops (Verzögerungszeit eines Flipflops)
t_{clk}	s	Taktperiode
t_{HL}	s	Anstiegszeit am Ausgang eines CMOS-Gatters
t_{Hold}	s	Hold-Zeit eines Flipflops
t_{LH}	s	Abfallzeit am Ausgang eines CMOS-Gatters
t_p	s	Verzögerung eines CML-Inverters
t_{pHL}	s	Verzögerungszeit zwischen Ein- und Ausgang eines CMOS-Gatters mit fallender Flanke am Ausgang
t_{pLH}	s	Verzögerungszeit zwischen Ein- und Ausgang eines CMOS-Gatters mit steigender Flanke am Ausgang

Variable	Einheit	Bedeutung
t_{Setup}	s	Rüstzeit eines Flipflops bzw. Rüstzeit eines Addierers
U_a	V	Aggressorspannung
U_a	V	Ausgangsspannung eines Schmitt-Triggers
U_{Bias}	V	Bias-Spannung einer CML-Schaltung
U_{CM}	V	Common-Mode-Spannung einer CML-Schaltung
U_{Deg}	V	Degradierte Spannung
U_{DD}	V	Positive Versorgungsspannung einer CMOS-Schaltung
U_{DS}	V	Drain-Source-Spannung eines MOSFET
U_{DSsat}	V	Drain-Source-Spannung, bei der der MOSFET in Geschwindigkeitssättigung arbeitet (Sättigungsspannung)
U_{GS}	V	Gate-Source-Spannung eines MOSFET
U_{glitch}	V	Spannung eines Störimpulses
U_e	V	Eingangsspannung eines Schmitt-Triggers
U_{e+}	V	Schaltsschwelle eines Schmitt-Triggers bei ansteigender Eingangsspannung
U_{e-}	V	Schaltsschwelle eines Schmitt-Triggers bei fallender Eingangsspannung
U_H	V	High-Pegel/maximaler Spannungspegel eines Logiksignals
U_{Hub}	V	Spannungsdifferenz zwischen zwei Spannungspegeln
U_{in}	V	Eingangsspannung einer CML-Schaltung
U_L	V	Low-Pegel/minimaler Spannungspegel eines Logiksignals
U_M	V	Schaltsschwelle eines CMOS-Inverters (entspricht i.d.R. der mittleren Spannung zwischen statischem High- und Low-Pegel)
U_{max}	V	Maximale Spannung
U_{min}	V	Minimale Spannung
U_{nom}	V	Nominale Spannung
U_{out}	V	Ausgangsspannung einer CML-Schaltung
U_{Th}	V	Schwellen-/Einsatzspannung eines MOSFET
U_{Tn}	V	Schwellen-/Einsatzspannung eines n-Kanal-MOSFET
U_{Tp}	V	Schwellen-/Einsatzspannung eines p-Kanal-MOSFET
U_S	V	Steuerspannung eines Transfertransistors bzw. -gatters
U_{SS}	V	Negative Versorgungsspannung einer CMOS-Schaltung
V_U		Spannungsverstärkung
W	m	Kanalweite eines MOSFET
W_{min}	m	Minimale Kanalweite eines MOSFET
W_n	m	Kanalweite eines n-Kanal-MOSFET
W_p	m	Kanalweite eines p-Kanal-MOSFET
Z		Ausgangssignal/-knoten einer CMOS-Schaltung

Variable	Einheit	Bedeutung
$z_{\mu,v,k}$		Mögliche Signalniveaus mit Symbolnebensprechen ohne Rauschen zur Zeit k
z_k		Erwartetes Symbol mit Symbolnebensprechen ohne Rauschen

Formelzeichen: Griechische Buchstaben

β	A/V^2	Stromverstärkungsfaktor des MOSFET
β_n	A/V^2	Stromverstärkungsfaktor des n-Kanal-MOSFET
β_p	A/V^2	Stromverstärkungsfaktor des p-Kanal-MOSFET
β_R		Verhältnis zwischen Elektronen- zur Löcherbeweglichkeit
ϵ_r		relative Dielektrizitätszahl
η_D		Dekodiereffizienz
λ	m	Lichtwellenlänge
μ		Zustandsindex
μ	A/V^2	Ladungsträgerbeweglichkeit in einem MOSFET
μ_n	A/V^2	Elektronenbeweglichkeit in einem n-Kanal-MOSFET
μ_p	A/V^2	Löcherbeweglichkeit in einem p-Kanal-MOSFET
ν		Zustandsübergangsindex
σ_0		Standardabweichung
Δt	s	zeitliche Differenz allgemein
Δt_p	s	Differenz von Verzögerungszeiten
τ	s	Zeit
ΔU_{in}	V	Eingangsdifferenzspannung eines CML-Bausteins
ΔU_{out}	V	Ausgangsdifferenzspannung eines CML-Bausteins
φ		Taktsignal in Schaltbildern auf Transistorebene
ω	1/s	Kreisfrequenz

Abstract

In the recent years the demand on communication rose rapidly. Up to now there exists separate communication networks for telephony, internet and for transmission of television channels. In course of digitalisation all of these applications merge into one communication network. New applications like video on demand services and television over the internet and the increased use of the internet in general need an upgrade of the existing broadband networks. The more and more economical use of mobile communication requires more bandwidth in the mobile channel itself and in the backbone of the operator's wire-bound network. Wired communication is subject of the electrical network like telephony and digital subscriber line. Optical networks are for the most part core networks of the providers for telecommunication or cable television. Other parts like communication units in servers and personal computers need also more bandwidth because the data of previously mentioned services are processed inside there. Mass-produced articles like personal computers are under pricing pressure. Thus, components for backplanes, cables (SATA) and plugs between memory and mother board must be very cheap. Due to that, the signal rates increase over lossy transmission channel the requirements increase, too.

The bit rate in recent metro- and long-haul optical networks is 10 Gbit/s per wavelength. Currently an upgrade from 10 Gbit/s links to 40 Gbit/s links is in progress. At these data rates the signal distortions resulting from chromatic dispersion (CD) and time variant polarisation-mode dispersion (PMD) are severe. Commercial equalizers are available for 10 Gbit/s operation. First 40 Gbit/s systems using feed-forward, decision feedback equalizers and analogue optical filters have been reported. Theoretical it is shown that the Maximum likelihood sequence estimation (MLSE) is the best equalisation method for all possible impairments. A possibility for the realisation of MLSE with less computing effort in hardware is the trellis-based Viterbi algorithm. However, the implementation of the Viterbi algorithm is a challenge at these high bit rates. This work presents a Viterbi equalizer for a 40 Gbit/s optical receiver system in a CMOS technology with 90 nm gate length.

The optical transmission channel is assumed to have a memory of two bits which is sufficient to model CD and PMD. Thus, the channel can be described by a four-state finite state process with eight possible transitions. The time-indexed version of a state transition diagram is called trellis. Each node represents a state at a certain time index and the branches correspond to the state transitions.

There are several approaches to implement the Viterbi algorithm in hardware. The most obvious approach is the serial approach, but this one has the highest speed requirement because the recursive nature of the Add-Compare-Select (ACS) structure prevents pipelining. So in one clock

cycle of 43 GHz a complete ACS operation must be done. This approach cannot be implemented with any current technology. The Viterbi algorithm has to be parallelised by block decoding concepts. Two different block decoding concepts like “Sliding Block Viterbi Decoder” (SBVD) and “Minimized Method” (MM) are promising. The major advantage of a block decoding scheme is that pipelining can be applied. The SBVD architecture has lower coding loss and smaller hardware complexity than the MM concept for a given survivor length. In block decoding methods the typical block length, which corresponds to the trellis length, should be five times the channel memory. For well-chosen metrics the survivor path length can be reduced to 2.5 times the channel memory, without any significant performance degradation. Thus, the trace back length in this Viterbi equalizer is chosen to eight based on a channel memory of two bits. The SBVD is realised with simultaneous forward and backward processing that halves the throughput delay at a certain block length.

The Viterbi equalizer is partitioned into four identical systolic block decoders (SBD). The advantage of using four SBDs instead of one large SBD is the four times shorter throughput delay time. Each SBD contains two trellises (one forward and one backward trellis) that are formed by the interconnections of Add-Compare-Select-Units. The main building blocks of each SBD are:

- Branch-Metric-Unit (BMU),
- Add-Compare-Select-Unit (ACSU),
- Trace-Back-Unit (TBU),
- Select-Minimum-Unit (SMU).

Demultiplexers on the input side and multiplexers on the output side are used for parallelisation and serialisation to keep slim the input and the output interface.

The BMU calculates the Euclidian distances (branch metrics) of all possible paths of a received sequence to an expected value. Since the incoming 3 bit symbols and the expectation values are quantised, the BMU can be realised as a look up table. Due to the channel model, this look up table consists of 64 elements that contain the inverse logarithmic probability of a transmitted sequence under the occurrence of a received sequence. These probabilities are interpreted as costs. The BMU is implemented as an array of 384 SRAM cells. The cells can be initialised prior the data transmission. Updating during equalisation is also possible to cope with time variant transmission channels.

Each state of the trellis is implemented by an ACSU that consists of two adders, a digital comparator and a multiplexer which selects the lower sum of the two adders. Calculating with costs with smaller values being more likely than bigger ones instead of probabilities has the advantage that no normalization of the sum path metrics is necessary because overflow is avoided. By the nature of the Viterbi algorithm the paths with the higher weights will be

dropped. Thus, a theoretical overflow will not occur, due to signal characteristics and the larger chosen word length of the path metrics. The probability of this event can be neglected because of the logarithmic distribution of the costs. The decisions of the ACSUs are stored and delayed in buffer flip-flops that compensate for the skew in different paths. The most complex operation of the Viterbi algorithm is the addition and comparison of the branch and path metrics. The ACSU is implemented by Ripple-Carry-Adders (RCA). The quantised branch metrics of the BMU have a word length of 6 bit. To avoid overflow, the path metrics have a word length of 8 bit. The RCA has a major advantage in the ACSU because the comparison of the sums can be done simultaneously with the add operation. The overall delay of the add-compare operation is only one full adder delay larger than the delay of the adder itself. Thus, the RCA architecture is very efficient with respect to chip area and current consumption. It is better suited for the ACS operation than Carry-Lookahead-Adders (CLA). Although the CLA architecture is the fastest known adder – due to its logarithmic increasing delay over the word length – the comparison in the ACSU cannot start prior the result of the addition is calculated. A CLA needs about twice the number of transistors, chip area and current consumption than a RCA. The full adder cell is implemented by delay optimised gates. CMOS gates like OR- and AND-gates are slower than the inverted ones. Especially OR- and also NOR-gates should be avoided because of the larger and respectively slower PMOS-paths. The calculation of the sum contains two XOR functions. These XOR functions are implemented using fast transmission gates. The critical path is formed by the carry-path. Instead of applying slow OR- and AND-gates, the carry-path can be translated into fast NAND-gates. Thus, only a single XOR- and three fast NAND-gates are in the critical path. Output flip-flops buffer the sum and the decision bit of the path with the lower costs.

At the end of a forward and backward trellis of each SBD the four sums of each trellis will be concatenated and the absolute minimum will be chosen by the SMU. Thus, the processing block length is doubled. The surviving path is that one that has the concatenated minimum of path costs. The SMU provides the address of the state of the surviving path. The SMU is implemented by two parallel ACSU and followed by a digital comparator with two pipeline stages.

Twelve demultiplexers parallelise twelve 10.7 Gbit/s binary input streams resulting from a 42.8 GS/s 3 bit ADC with internal 1:4 DEMUX functionality that feeds the Viterbi equalizer. The twelve 1:8 demultiplexers are distributed over a distance of 4 mm at the input side of the chip. On the output side of the Viterbi equalizer four 8:1 multiplexers serialise the 32-fold parallel decoded data stream of the DSP. All twelve input data streams and the four output streams are applied differentially.

The digital building blocks of the Viterbi equalizer are implemented in static CMOS logic whereas the high-speed input- and output-interfaces are implemented in current mode logic (CML). All the gates are full custom implementations. Simulations of the standard-logic-cells of

the 90 nm technology have shown that the delays are almost twice as high as the full custom gates. Due to the large number of flip-flops, a high dynamic current up to 5.5 A occurs during clock transitions. Thus, block capacitors of 2 nF in total are distributed all over the chip.

The mask layout of the Viterbi equalizer is sent via internet to the chip broker CMP. The circuitry is fabricated by ST Mircoelectronics. After seven months 25 dies are returned by CMP. The size of a die is 4.08 mm times 1.40 mm = 5.7 mm².

The Viterbi equalizer is bonded on a PCB. The RocketIO interfaces (RIO) of a Virtex-4 FPGA are used as binary data sources. With the RIO line rates up to 10 Gbit/s are possible. The Virtex-4 FPGA provides 20 RIOs. Twelve of them are used to transmit data, one of them is used as the clock signal for the Viterbi equalizer and another one as the trigger signal for the oscilloscope. The FPGA is programmed in VHDL with 12 memories. Each of them contains a 48 bit sequence that corresponds to a 192 symbol sequence that suffers from typical intersymbol interference (ISI) on an optical fiber link. The memories are read-out repetitively. A problem of the RIO is that the internal phase-locked-loop doesn't start in a defined state and the lines on the FPGA-board do not have the same length. Thus, the twelve RIOs must be synchronised using a VHDL-implemented barrel-shifter. The branch metrics matching to the ISI are fed to the VEC by the low-speed branch-metric memory interface of the same FPGA. The equalised output data of the Viterbi equalizer are displayed on a sampling oscilloscope.

The logic of the Viterbi equalizer is tested first in low frequency mode at 12.5 MHz to avoid the problems concerning the synchronisation of the RIO. In low frequency mode there is no measurable difference in the power consumption between a switched-on and a switched-off clock signal. Thus, almost the whole power is consumed by the CML demultiplexers and the CML multiplexers. The simulated power consumption of the CML part of the circuit is 1.55 W at 1 V. Hence the measurement and simulation results match very well concerning power consumption. The 90 nm CMOS process defines three corner voltages: A minimum, a nominal and an overdrive voltage. Between the lower and upper voltage corner the Viterbi equalizer works properly until an input frequency up to 4 GHz. That means that the equalised bit sequence matches with the simulated one. A clock frequency of 4 GHz corresponds to a data rate of 32 Gbit/s that is more than three times as fast as of a commercial available Viterbi equalizer for optical communication links. The power dissipation at 4 GHz is 2.39 W. This corresponds to a normalised energy effort of 75 pJ/bit and a needed chip area of 0.178 mm²/Gbit/s. Concerning the normalised energy effort this design needs up to 4.6 times less energy than a comparable design in a 90 nm CMOS technology to process 1 bit. Further, this design needs up to 5.6 times less active silicon area to process 1 Gbit/s.

At higher frequencies first impairments, caused by phase differences between the input RIO channels, appear. As mentioned the synchronisation of the RIO is a big challenge because of the different line length on the FPGA-PCB. Since the FPGA allows only bitwise shifting of RIO output data, it is not possible to do a proper alignment for higher data rates, although the single RIO channel line rate may reach 10 Gbit/s. Phase differences of about 50 ps between the channels cannot be compensated digitally on the FPGA. This problem will be addressed in future work by switchable external phase shifters or a wirelength compensating PCB that will be placed in front of each input.

Due to the limitations of the measurement mentioned above, a combination of an already fabricated SiGe 43 GS/s ADC with this Viterbi equalizer is proposed. Since the 24 wire high-speed input interface of the Viterbi equalizer can be bonded directly to this ADC, there is no difference in the delay between the interconnects. The Viterbi equalizer is designed for data rates up to 43 Gbit/s. Simulations have shown that the longest combinatorial path of the ACSU including the setup- and clock-to-output delays of the flip-flops does not exceed 614 ps by applying typical transistor and parasitic parameters. This is lower than the internal clock period of the DSP of 747 ps.

On-Wafer measurements, where only the clock signal and the power supply are connected, show that the internal clocking tree and the flip-flops will work up to an external applied clock frequency of 6.1 GHz. This corresponds to a data rate of 48.8 Gbit/s. The ADC-combined measurement setup is in preparation.

At the targeted data rate of 43 Gbit/s corresponding to a clock frequency of 5.35 GHz the power dissipation is expected to 2.6 W leading to the normalised energy effort of 59 pJ/bit. The normalised chip area will reach 0.133 mm²/Gbit/s.

The Viterbi equalizer provides an interface to read out part of the incoming symbols and the corresponding decoded bits. With this data a channel approximation can be done outside the chip to calculate new branch metrics. Thus, the time variant PMD can be compensated.

In this work a Viterbi equalizer with a four times parallelised SBD architecture is presented. To the best of the author's knowledge this Viterbi equalizer is the world first that is measured successfully at a data rate of 32 Gbit/s due to test limitations. This is three times as fast as a commercial available product. Concerning chip area and power dissipation this work dominates publications in comparable CMOS technologies. Further measurement setups that will show the functionality at the targeted data rate of 43 Gbit/s are in progress.

1 Einleitung

In den vergangenen Jahren stieg der Bedarf an Kommunikation rasant an. Während bisher getrennte Kommunikationsnetze für Telefonie, Internet und auch zur Übertragung von Fernsehsignalen existierten, konvergieren im Zuge der Digitalisierung alle diese Anwendungen in ein Datenkommunikationsnetz. Neue Anwendungen wie beispielsweise Filmabrufdienste (engl. Video-on-Demand, VoD), Fernsehen über das Internet (IPTV) als auch die generell gesteigerte Internetnutzung erfordern eine Aufrüstung vorhandener Breitbandnetze. Bedingt durch die immer weiter verbreitete und wirtschaftlichere Mobilkommunikation (UMTS, LTE, WIMAX) kommt es ebenfalls zu einem weiteren Anstieg an Bandbreitenbedarf sowohl im Mobilfunkkanal selbst als auch in den daran angeschlossenen drahtgebundenen oder optischen Netzen. Beispiele für drahtgebundene Netze sind das Telefon-, xDSL- und das Kabelfernsehtz im Teilnehmerzugangsbereich. Optische Netze sind meist Teil der Kernnetze von Telekommunikationsunternehmen und Kabelfernsehtzbetreiber.

In weiteren Bereichen, wie den Kommunikationseinheiten innerhalb von Servern und PCs, steigt ebenfalls der Bandbreitenbedarf, da dort die Daten oben genannter Dienste verarbeitet werden. Da Massenartikel wie PCs einem hohen Preisdruck unterliegen, kommen hier besonders günstige Komponenten bei Rückwandplatinen, Verbindungskabeln (SATA) und Steckverbindern (Stecker zwischen Speicher oder PCI-Einsteckkarten und Rückwandplatinen) zum Einsatz.

Aufgrund dieses Trends werden an die signalverarbeitenden Komponenten immer höhere Anforderungen gestellt, um die schlechten Kanäle für die Übertragung hochbitratiger Signale tauglich zu machen.

In der Hochgeschwindigkeitskommunikation gewinnt die optische Datenübertragung immer mehr an Bedeutung, da die Eigenschaften des elektrischen Kanals bei Geschwindigkeiten von über 1 Gbit/s hinsichtlich Dämpfung und Nebensprechen zu schlecht sind.

Bei der optischen Datenübertragung erfolgt derzeit die Aufrüstung von 10 Gbit/s auf 40 Gbit/s pro Wellenlängenkanal in den Kernnetzen.

Allerdings kommen bei Geschwindigkeiten von über 10 Gbit/s Dispersionseffekte bei Glasfaserleitungen zum Tragen, die teilweise sogar zeitveränderlich sind [1], [2]. Deshalb sind geeignete Maßnahmen zur Kompensation dieser Signalverzerrungen zu finden.

In der Übertragungstechnik wie z.B. dem digitalen Mobilfunkstandard GSM oder dem digitalen Fernsehen DVB [3], aber auch bei der Speicherung von Daten auf Festplatten werden durch eine spezielle Vorkodierung dem Signal Redundanzen hinzugefügt, indem eine Beziehung des aktuellen Symbols zum Vorgänger und Nachfolger hergestellt wird. Die Abhängigkeit der einzelnen Symbole untereinander nennt man Korrelation. Die Korrelation muss dann im

Empfänger wieder dekodiert werden. Die Entflechtung des Signals erfolgt in diesen Systemen mittels Viterbi-Dekodern.

Bei einem System mit Entzerrer hingegen wird die Kodierung, aus der eine Korrelation der Symbole hervorgeht, durch den gestörten Kanal selbst vorgenommen. Bei dieser Art der Kodierung spricht man von Symbolnebensprechen. Der Unterschied besteht darin, dass die Korrelation bei den erstgenannten Verfahren beabsichtigt im Sender vorgenommen wird, während sie bei einem Entzerrersystem ungewollt auf Grund von Dispersionseffekten des Übertragungsmediums entsteht. Hierbei spricht man dann von einem Viterbi-Entzerrer anstelle eines Viterbi-Dekoders.

1.1 Optische Kommunikation

Abbildung 1.1 zeigt Komponenten eines digitalen optischen Übertragungssystems. Als Sender dient eine Laserdiode, die bei der für optische Datenkommunikation typischen Wellenlänge von 1550 nm betrieben wird. Die Datenübertragung erfolgt bei 40 Gbit/s-Systemen noch als binäre Intensitätsmodulation, das heißt, das Licht wird im Rahmen der zu übertragenen Bitfolge ein- und ausgeschaltet. Als Übertragungsmedium kommen Einmodenfasern zum Einsatz, die je nach Länge bei 40 Gbit/s, aber auch schon bei 10 Gbit/s Dispersionseffekte aufweisen. Dies hat zur Folge, dass sich die Lichtpulse aufgrund unterschiedlicher Ausbreitungsgeschwindigkeiten überlagern. Die Entfernungen, welche durch die Glasfaser überbrückt werden sollen, liegen im Bereich von 80 km bis 100 km. Es handelt sich hierbei um sogenannte „Metropolitan Area Networks“ (MAN). Es sind aber auch Weitverkehrsnetze (WAN), die Distanzen bis zu 1000 km abdecken, möglich. Dort kommen zusätzlich optische Verstärker zum Einsatz, die das Signal weiter verzerrern.

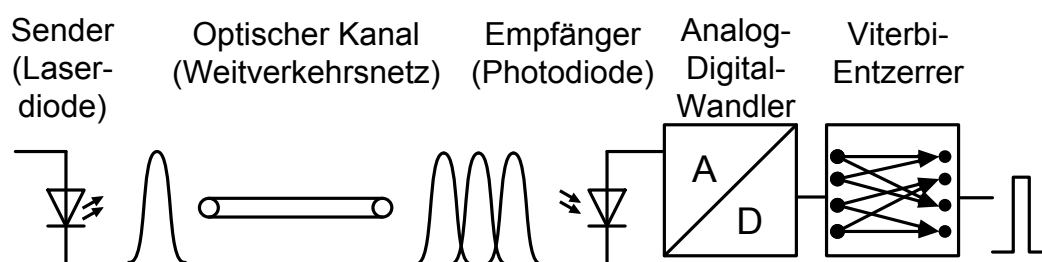


Abbildung 1.1: Vereinfacht dargestelltes optisches Übertragungssystem

Auf Empfangsseite wird das verzerrte Lichtsignal von einer Photodiode in ein elektrisches Signal umgewandelt. Da das empfangene Signal aus einer Überlagerung von Pulsen besteht, muss es für die digitale Weiterverarbeitung mittels eines Analog-Digital-(AD)-Wandlers zunächst abgetastet und quantisiert werden. Ein nachfolgender Entzerrer versucht auf Basis mathematischer Methoden das ursprüngliche Sendesignal wiederherzustellen. In einem realen

Übertragungssystem werden zusätzlich zu den in Abbildung 1.1 gezeichneten Blöcken zwischen der Photodiode und dem AD-Wandler noch ein Transimpedanzverstärker und eine Schaltung zur Taktrückgewinnung benötigt. Für eine Taktrückgewinnung ist es notwendig, dass das Signal vorkodiert wird, damit eine bestimmte Anzahl an Signalwechseln pro Zeitperiode garantiert wird. Bekannte Leitungskodierungen, die dies sicherstellen, sind 8B/10B bei optischem Gigabit-Ethernet [4] oder 64B/66B bei elektrischem Gigabit-Ethernet [5].

In optischen Kommunikationssystemen treten auf der Glasfaser folgende Dispersionseffekte (Signalverzerrungen) auf:

- Chromatische Dispersion (CD)
- Polarisationsmodendispersion (PMD)

Es muss also ein Mittel gefunden werden, um die Dispersionseffekte zu kompensieren. Eine einfache Möglichkeit wäre die Verwendung dispersionskompensierender Fasern. Allerdings lassen sich dadurch nur statische Dispersionseffekte wie die chromatische Dispersion kompensieren. Außerdem müsste man für jede optische Übertragungsstrecke eine individuelle Anpassung vornehmen. Bei der Polarisationsmodendispersion handelt es sich hingegen um eine zeitvariante Verzerrung, die nur durch eine adaptive Entzerrerschaltung kompensiert werden kann. Mit einem adaptiven Entzerrer können sowohl statische als auch dynamische Dispersionseffekte ausgeglichen werden, wodurch sich eine einfache Installation nach dem „Plug-and-play“-Prinzip bei beliebigen Übertragungskanälen ergibt. Somit kristallisiert sich eine elektronische Entzerrerlösung als vorteilhaft heraus. Die Dispersionseffekte lassen sich als Symbolnebensprechen (engl. Intersymbol interference, ISI) beschreiben. Es gibt Methoden auf mathematischer Basis, mit denen sich ISI korrigieren lässt. Ein bekanntes Beispiel hierfür ist der trellisbasierte Viterbi-Algorithmus.

1.2 Technologie und Architektur

Der Entwurf einer Entzerrerschaltung auf Basis des Viterbi-Algorithmus ist in einer Halbleitertechnologie bei der gewünschten Datenrate von 43 Gbit/s eine große Herausforderung. Für eine serielle Architektur würde eine maximale Rechendauer von rund 23 ps zur Verfügung stehen. Dies ist mit heutigen industriell erhältlichen Technologien nicht realisierbar. Simulationen in einer kommerziellen komplementären Metalloxid-Halbleiter-(CMOS)-Technologie mit 90 nm Kanallänge weisen eine Inverterlaufzeit von 9,0 ps auf. Der Viterbi-Algorithmus besteht jedoch aus komplexen Rechenoperationen, die eine deutlich höhere kombinatorische Laufzeit aufweisen. Bipolartechnologien sind aufgrund der höheren Steilheit g_m des Bipolartransistors deutlich schneller als der MOS-Transistor, jedoch ist die Verlustleistung von Bipolartransistoren erheblich höher als die von MOS-Transistoren [6], was einer Höchstintegration (VLSI) widerspricht. Da CMOS-Technologien im Vergleich relativ langsam sind, muss man sich Architekturen überlegen, mit denen es möglich ist, Datenraten von 43 Gbit/s

zu verarbeiten. Eine Möglichkeit besteht darin, die Rechenoperationen in viele Teiloperationen zu zerhacken – in der Fachsprache „Pipelining“ genannt –, so dass nur wenig kombinatorische Logik in einer Taktperiode verarbeitet werden muss. Dennoch ist das Pipelining auch mit Nachteilen verbunden: Es werden viel mehr Flipflops benötigt, die einen hohen dynamischen Stromverbrauch aufweisen. Außerdem haben Flipflops auch kombinatorische Laufzeiten (Rüst-, Halte- und Verzögerungszeiten), die eingehalten werden müssen und somit von der zur Verfügung stehenden Taktperiode abgezogen werden müssen. Durch den immensen Einsatz von Pipelining verschlechtert sich also das Aufwand-Nutzenverhältnis. Außerdem erhöht sich durch die vielen in Serie geschalteten Flipflops die Durchlaufzeit, was sich aber durch die höhere Taktfrequenz relativiert.

Ein anderer Ansatz ist die Möglichkeit zur Parallelisierung des Viterbi-Algorithmus [7]. Dadurch ist es möglich, bei langsamerer Taktfrequenz, als die mit der die seriellen Daten verarbeitet werden müssen, zu rechnen. Bei der Parallelisierung muss ebenfalls ein Aufwand-Nutzenverhältnis abgeschätzt werden, da zu starke Parallelisierung zu einer großen Chipfläche führen würde. Einerseits hätte dies hohe Prozessierungskosten zur Folge, andererseits müssten an die Verdrahtung höhere Anforderungen in Bezug auf Signallaufzeiten und Übersprechen gestellt werden.

Somit muss bei der Entwicklung der Architektur ein gesundes Verhältnis zwischen Pipelining und Parallelisierung gefunden werden, um eine möglichst hohe Effizienz in Bezug auf Leistung, Chipfläche, Kosten zu erhalten.

Ein weiterer Aspekt, der immer mehr an Bedeutung gewinnt, ist der Energieverbrauch. Eine Grundregel besagt, dass ein System, das viermal schneller ist, nur zwei bis zweieinhalbmal so viel Energie verbrauchen darf. Daher wird der zu entwerfende digitale Signalprozessor (DSP) ausschließlich in statischer CMOS-Logik entworfen. Pseudo-NMOS oder gar differenzielle Stromschaltertechnik (engl. Current-Mode-Logic, CML) scheiden aus. Ein weiterer Vorteil aufgrund des niedrigen Energiebedarfs von statischem CMOS ist die Höchstintegration. Es lassen sich auf kleinster Fläche enorm viele Transistoren platzieren. Bei Pseudo-NMOS-Logik und CML hingegen fließen Querströme, die den Chip stark aufheizen. Damit nicht zu viel Wärme zum Hitzetod der Schaltung führt, können die Transistoren nicht so dicht angeordnet werden. Außerdem wird mehr Platz für die größer zu dimensionierenden Leitungen benötigt. Das Kostenargument spricht ebenfalls für den Einsatz der VLSI-fähigen statischen CMOS-Logik, da die Maskenkosten in Höhe von etwa 5000 EUR/mm² [8] immens hoch sind.

Eine Alternative zu statischem CMOS ist dynamisches CMOS. Hierbei kann ähnlich wie bei Pseudo-NMOS Fläche eingespart werden, da ein Teilzweig (p-Kanal- oder n-Kanalzweig) eingespart werden kann. Querströme wie bei Pseudo-NMOS gibt es bei dynamischem CMOS

nicht. Daher ist auch dieses Schaltungskonzept VLSI-fähig. Den Flächenvorteil erkauft man sich mit einer geringeren Zuverlässigkeit der Schaltung, da Zwischenknoten des Schaltungsnetzwerkes hochohmige Zustände einnehmen können. Aufgrund von Leckströmen können sich diese Knoten entladen, was zu einer Fehlinterpretation des Logikpegels führen kann. Um diesen Nachteil zu kompensieren, betreibt man dynamisches CMOS mit einer Mindesttaktfrequenz, so dass die Logikpegel einmal pro Taktperiode auf den statischen High- bzw. Low-Pegel initialisiert werden. Daraus entsteht insbesondere beim Test der Schaltung ein weiterer Nachteil, dass die Taktfrequenz nicht beliebig verringert werden kann.

Alle diese Gründe verfestigen weiter die Entscheidung, den DSP in statischer CMOS-Logik zu entwerfen.

1.3 Stand der Technik

1.3.1 Optische Übertragungssysteme

Es existieren bereits optische Übertragungssysteme mit einer Datenrate von 43 Gbit/s, bei denen Dispersionskompensatoren auf Basis analoger optischer Filter Anwendung finden. Die Nachteile dieser Verfahren sind die geringere Flexibilität und die höheren Kosten. Die Anpassung an den zeitvarianten Kanal erfolgt im Wesentlichen durch eine mechanische Verstellung der Filter. Dies erschwert, Änderungen nachfolgen zu können, die sich im Millisekundenbereich ereignen. Elektronische Implementierungen – teilweise mit Adaption der Entzerrerkoeffizienten – in schnellen Indium-Phosphid- und Silizium-Germanium-Technologien, die auf Vorwärtsentzerrern mit Rückkopplung basieren, werden in [9] und [10] beschrieben. Untersuchungen zeigen aber, dass Entzerrer auf Basis von Wahrscheinlichkeitsberechnungen die Dispersion besser kompensieren und somit optische Kompensationsfilter überflüssig werden [11]. Digital-schaltungen in CMOS-Technologien sind aufgrund des Massenmarktes kostengünstiger als in Nischentechnologien gefertigte Schaltungen.

1.3.2 Hardware-Entwürfe verschiedener Viterbi-Dekoder

Tabelle 1.1 fasst in einer Übersicht den aktuellen Stand der Technik bisher veröffentlichter Hardware-Viterbi-Dekoder bzw. -Entzerrer zusammen. Der Vergleich stellt die Chipentwürfe im Wesentlichen hinsichtlich der Technologie, Chipgröße, Datendurchsatz und Leistungsaufnahme gegenüber. Es wird ersichtlich, dass der bisher schnellste kommerziell erhältliche Viterbi-Entzerrer einen Datendurchsatz von 10 Gbit/s aufweist. Ziel dieser Arbeit ist ein Viterbi-Entzerrer-Chip, der 43 Gbit/s verarbeitet und gleichzeitig bezüglich des Energiebedarfs pro Bit und Chipfläche pro Gigabit Durchsatz im Vergleich zu anderen Implementierungen deutlich höhere Werte erreicht.

Tabelle 1.1: Stand der Technik verschiedener Hardware-Viterbi-Dekoder

Veröffentlichung	Dawid, Fettweis, Meyr 1996 [12]	Black, Meng 1997 [13]	Core-Optics 2000 [14]	Intel 2008 [15]
Technologie	1,0 μm CMOS ES2 (Atmel)	1,2 μm CMOS HP	0,12 μm CMOS IBM	90 nm CMOS Intel
Chipgröße	95 mm ² (4 ICs)	81 mm ²	keine Angabe	1,9 mm ²
Anzahl Transistoren	260000	150000	1.8 Mio. Gatter	698000
Takt (intern)	50 MHz	80 MHz	keine Angabe	3,8 / 4,7 GHz
Durchsatz (nom., ggf. min., max.)	1,2 Gbit/s	0,144 Gbit/s / 1 Gbit/s	10 Gbit/s	1,9 Gbit/s / 2,35 Gbit/s
Leistung (nom., ggf. min., max.)	3,6 W	0,024 W / 3,0 W	keine Angabe	0,358 W / 0,813 W
Versorgungsspannung	keine Angabe	1,5 V / 5 V	keine Angabe	1,3 V / 1,7 V
FOM 1: Energie pro Datenmenge	3 nJ/bit	1,67 nJ/bit / 3 nJ/bit	keine Angabe	0,188 nJ/bit / 0,346 nJ/bit
FOM 2: Chipfläche pro Durchsatz	79,2 mm ² /Gbit/s	81 mm ² /Gbit/s	keine Angabe	1,0 mm ² /Gbit/s / 0,809 mm ² /Gbit/s
Architektur	Minimized Method (parallel)	Sliding Block (parallel)	keine Angabe	seriell

1.4 Aufbau dieser Arbeit

Im zweiten Kapitel erfolgt zunächst eine Erläuterung zu den Dispersionseffekten optischer Übertragungskanäle. Nach einem Überblick über verschiedene Entzerrerkonzepte wird der Übertragungskanal eingehender betrachtet. Das Verfahren der größten Wahrscheinlichkeit (engl. Maximum Likelihood) gilt als das aussichtsreichste und wird daher näher beschrieben, bevor dann auf den Spezialfall des Viterbi-Algorithmus und dessen Implementierung auf Systemebene eingegangen wird. Kapitel 3 befasst sich mit Grundsaltungen und deren Dimensionierung in komplementärer Metalloxid-Halbleitertechnologie. Darauf aufbauend werden in Kapitel 4 komplexere Schaltungsblöcke auf Register-Transferebene wie z.B. ganze Speicherfelder und Addierer entworfen, bevor diese Teilblöcke zu der Gesamtschaltung zusammengefügt werden. In Kapitel 5 werden der Chipentwurf und die Chipfertigung erläutert. Der Messaufbau und die Messergebnisse werden in Kapitel 6 präsentiert. Kapitel 7 befasst sich mit dem Entwurf einer Adaptionsschaltung für die Entzerrerkoeffizienten in einer Hardwarebeschreibungssprache. Kapitel 8 schließt diese Arbeit mit einer Zusammenfassung ab und gibt einen Ausblick auf optische Übertragungsverfahren, die Datenraten von 100 Gbit/s und mehr adressieren.

2 Entzerrung optischer Übertragungskanäle

In diesem Kapitel werden zunächst Dispersionseffekte auf dem optischen Übertragungskanal behandelt, bevor dann verschiedene Entzerrerverfahren diskutiert werden. Danach erfolgt eine Beschreibung des Übertragungskanals im Hinblick auf das aussichtsreichste Entzerrerverfahren, das auf der Schätzung der größten Wahrscheinlichkeit einer Bitsequenz (engl. Maximum Likelihood Sequence Estimation, MLSE) basiert. Daraufhin wird der Viterbi-Algorithmus, der einen Spezialfall des MLSE-Verfahrens darstellt, erläutert. Abschließend wird eine Möglichkeit der Implementierung des Viterbi-Algorithmus in Hardware auf Architekturebene aufgezeigt.

2.1 Dispersionseffekte

In der Kommunikationstechnik bestehen optische Übertragungsmedien für lange Strecken aus Glasfasern, die sich in Mehrmodenfasern (engl. Multimode-Fiber, MMF) und Einmodenfasern (engl. Singlemode-Fiber, SMF) unterteilen lassen. Bei MMF breitet sich der eingekoppelte Lichtstrahl aufgrund des großen Durchmessers von typischerweise $62,5\ \mu\text{m}$ auf unterschiedlichen Wegen durch das Medium aus. Dies hat zur Folge, dass das Licht am Ende der Faser aufgesplittet ist und zu unterschiedlichen Zeitpunkten ankommt. Je nach Übertragungsrate und der Faserlänge sind die zeitlichen Differenzen so groß, dass sich mehrere übertragene Symbole miteinander vermischen. Dieser Effekt wird differenzielle Modendispersion (DMD) genannt. Die DMD im Optischen ist bei der drahtlosen Übertragung mit der Mehrwegeausbreitung, die z.B. aufgrund von Signalreflexionen an Gebäuden entsteht, vergleichbar. Auch dort führen unterschiedliche Signallaufzeiten zu unerwünschten Überlagerungseffekten. Allgemein handelt es dabei um Symbolnebensprechen. Daher werden MMF in der Regel nur für kurze Übertragungstrecken wie LAN verwendet. Bei SMF hingegen gibt es nur einen Lichtausbreitungsweg, bedingt durch den geringen Faserkerndurchmesser von nur $9\ \mu\text{m}$. Dadurch wird DMD vermieden. Trotzdem wird der Lichtstrahl aufgeteilt. Einerseits gibt es zwei Polarisations Ebenen, anhand derer sich das Licht unterschiedlich schnell ausbreitet, woraus Polarisationsmodendispersion entsteht. Andererseits ist der Brechungsindex abhängig von der Lichtwellenlänge. Da eine Laserdiode kein rein monochromatisches Licht aussendet und das Licht mit der doppelten Signalbandbreite moduliert ist, entsteht die chromatische Dispersion. Diese Effekte machen sich bei SMF erst bei Datenraten ab $10\ \text{Gbit/s}$ und bei Entfernungen von mehr als $10\ \text{km}$ bemerkbar. Daher werden für Strecken zur Verbindung zwischen größeren Städten (engl. Metropolitan Area Network, MAN), worunter man in Deutschland Entfernungen von 80 bis $100\ \text{km}$ versteht, SMF eingesetzt. Die beiden Dispersionseffekte werden nun genauer betrachtet.

2.1.1 Polarisationsmodendispersion

In einer Glasfaser gibt es zwei zueinander orthogonale Polarisations Ebenen. In jeder der beiden Ebenen breitet sich das Licht unterschiedlich schnell aus. Die Glasfaser wirkt doppelbrechend. Der Dispersionseffekt wird Polarisationsmodendispersion (PMD) genannt. Den Laufzeitunterschied $\Delta t = D_{\text{PMD}} \sqrt{L}$, der von der Quadratwurzel der Faserlänge abhängt, nennt man differenzielle Gruppenlaufzeit (DGD) und wird in Pikosekunden angegeben. Ein typischer Wert für einen PMD-Dispersionskoeffizienten einer Standard-Einmodenfaser [16] ist $D_{\text{PMD}} \leq 0,2 \frac{\text{ps}}{\sqrt{\text{km}}}$. Bei der Detektion eines PMD-behafteten Signals mittels einer Photodiode

werden die Lichtimpulse beider Polarisations Ebenen aufgrund der unterschiedlichen Laufzeit zu einem verbreiterten elektrischen Impuls zusammengefasst, was sich negativ auf die Augenöffnung des empfangenen Signals auswirkt. Je größer der Laufzeitunterschied ist, desto kleiner ist die Augenöffnung und desto größer ist das daraus hervorgehende Symbolnebensprechen. Da die Geschwindigkeit der Änderung von der Ursache der PMD abhängig ist [17], [18], müssen geeignete Kompensationsverfahren angewandt werden. Jede Faser hat eine Grund-DGD, welche rein von Faserparametern abhängt. Unter den Faserparametern versteht man unterschiedliche Brechungsindizes der Polarisations Ebenen und die Faserlänge. Die Dispersion, die durch die Grund-DGD hervorgerufen wird, ist allerdings statisch und braucht nicht im Betrieb nachgeführt werden. Die Grund-DGD entsteht durch Unregelmäßigkeiten bei der Herstellung, weshalb sie zufällig über die gesamte Glasfaser verteilt ist. Jede einzelne Glasfaser hat daher eine andere Grund-DGD. Unsymmetrische mechanische Beanspruchung wie z.B. Krümmungen bei der Verlegung führen ebenfalls zu einer unterschiedlichen DGD der beiden Polarisations Ebenen auf den entsprechenden Abschnitten.

Es gibt aber auch Effekte, die die DGD im Betrieb verändern. Temperaturschwankungen, die auf Tages- oder Jahreszeiten beruhen, führen zu einer Änderung der DGD. Dabei handelt es sich um langsame Änderungen im Bereich von Stunden oder noch größeren Zeitabschnitten. Die Kompensation der DGD kann, wie in [19] beschrieben, durch Zwischenschaltung verschiedener dispersionskompensierender Fasern durchgeführt werden, was aber auf Grund des mechanischen Aufbaus nur für langsame Änderungen der DGD in Frage kommt.

Druckschwankungen, die z.B. durch den Straßenverkehr hervorgerufen werden, bewirken ebenfalls eine Änderung der DGD. Allerdings ist das Zeitintervall, das hier im Sekundenbereich oder darunter liegt, wesentlich kürzer als bei Temperaturänderungen. Aus diesem Grund scheiden mechanische Verfahren, die zur Kompensation eingesetzt werden können, aus. Daher wird in dieser Arbeit ein elektronisches Dispersionskompensationsverfahren vorgeschlagen.

Die Einflüsse der Verlegung, Druck- und Temperaturschwankungen auf die DGD sind um Größenordnungen stärker als die Grund-DGD der Faser. Messungen verlegter Glasfasern zeigen, dass die zu kompensierende DGD bis zu 16,6 ps betragen kann. Eine fehlerfreie Datenübertragung bei 40 Gbit/s ist dabei nicht möglich [20].

Es gibt auch noch PMD höherer Ordnungen. Da deren Einfluss aber als gering eingestuft werden kann, werden diese Effekte vernachlässigt. In [11] werden sowohl optische als auch elektronische PMD-Kompensatoren untersucht. Prinzipiell liefern optische Kompensatoren bessere Ergebnisse, da die Kompensation vor der opto-elektronischen Wandlung durchgeführt wird. Die Photodiode und auch der folgende Analog-Digital-Wandler fügen dem Signal nämlich zusätzliches Rauschen hinzu. Im kommerziellen Bereich werden hingegen Kostengrößen wie Leistungsaufnahme und Flächenverbrauch herangezogen. Diesbezüglich sind elektronische Entzerrerverfahren im Vorteil.

2.1.2 Chromatische Dispersion

Außer der Polarisationsmodendispersion gibt es auch noch die chromatische Dispersion (CD). Die CD ist von der Faserlänge, von der spektralen Breite des Sendelasers und von der Signalbandbreite abhängig. Somit ergibt sich eine Bitratenabhängigkeit des Symbolnebensprechens auf Grund der CD. Bei der CD handelt es sich um eine zeitinvariante Verzerrung, die durch wellenlängenabhängige Brechungsindizes hervorgerufen wird. Die Folge sind Laufzeitunterschiede einzelner Wellenlängen. Diese Art der Dispersion ließe sich mithilfe dispersionskompensierender Fasern beheben, die man auf neu zu verlegenden Strecken, die für Datenraten mit 40 Gbit/s konzipiert sind, einsetzen kann. Sollen jedoch ältere, verlegte, auf 2,5 bis 10 Gbit/s entworfene Strecken für 40 Gbit/s tauglich gemacht werden, hilft nur das zusätzliche Hinzufügen einer entsprechend langen Faser, die die Dispersion kompensiert. Gründe, die gegen den Einsatz dieser zusätzlichen dispersionskompensierenden Fasern sprechen, sind Leistungsverluste, Signallaufzeitverzögerungen und auch die rein physische Spulengröße. Um die Dämpfungsverluste auf der Leitung auszugleichen, müssten noch weitere Verstärker in die Gesamtstrecke integriert werden. Elektronische Entzerrerverfahren haben dagegen den Vorteil der einfacheren Konfiguration und Wartbarkeit. Besondere Flexibilität wird durch die automatische Adaption des elektronischen Entzerrers erreicht.

Ein typischer Wert für den Dispersionskoeffizienten einer Standard-Einmodenfaser [16] liegt bei $D_{CD} \leq 18 \frac{\text{ps}}{\text{nm} \cdot \text{km}}$ bei der im Telekommunikationsbereich üblichen Wellenlänge von $\lambda = 1550 \text{ nm}$. Bei einer spektralen Breite des Lichtimpulses von 1 nm verbreitert sich der Lichtimpuls bei einer Faserlänge von 200 km um einen Laufzeitunterschied von 3600 ps [21]. Daraus folgt, dass man mit einer solchen Faser Daten bei einer Datenrate von 277 Mbit/s übertragen könnte, ohne dass man eine Dispersionskompensation auf Grund von Symbolnebensprechen vornehmen muss. Ziel dieser Arbeit ist es allerdings, eine Datenrate von 43 Gbit/s zu erreichen.

2.2 Entzerrerkonzepte

Im vorangegangenen Kapitel sind die Dispersionseffekte auf dem optischen Übertragungskanal erläutert worden, die sich als Symbolnebensprechen bzw. Intersymbolinterferenz (ISI) beschreiben lassen. Dieses Kapitel zeigt Verfahren auf, mit denen die ISI mit elektronischen Schaltungen rückgängig gemacht werden kann. Im Wesentlichen unterscheidet man zwischen analogen und digitalen Entzerrerverfahren.

2.2.1 Vorwärtsentzerrer

Ein weit verbreitetes analoges Konzept ist das des Vorwärtsentzerrers (engl. Feedforward Equalizer, FFE) oder auch Transversalentzerrers. Hierbei handelt es sich um ein FIR-Filter. Der zu entzerrende Eingangsimpuls wird über eine Flipflopkette verzögert, deren Ausgänge unterschiedlich gewichtet und aufsummiert werden. Allgemein besteht ein ISI-behafteter Impuls aus einem Impulsvorschwinger und einem Impulsnachschwinger. Mittels FFE lässt sich hauptsächlich der Impulsvorschwinger und bei entsprechender Filterlänge auch der Impulsnachschwinger korrigieren. Abbildung 2.1 zeigt einen Vorwärtsentzerrer mit einem Verzögerungsglied z^{-1} . Ein eingehender Impuls wird über zwei analoge Gewichtungsfaktoren d_0 und d_1 zeitlich nacheinander aufaddiert. Die Gewichtungsfaktoren müssen der Übertragungscharakteristik des Kanals angepasst werden. Trotz der zeitdiskreten Signalverarbeitung spricht man hier noch von einer analogen Schaltung, da das Eingangssignal wertekontinuierlich verarbeitet wird.

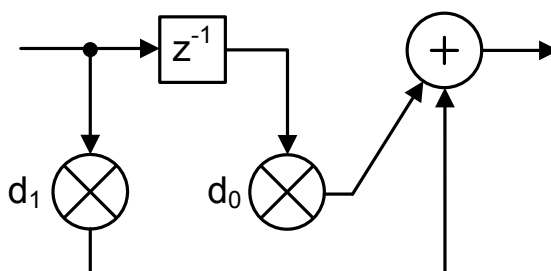


Abbildung 2.1: Blockschaltbild eines Vorwärtsentzerrers

2.2.2 Vorwärtsentzerrer und entscheidungsrückgekoppelter Entzerrer

Eine Verbesserung der Entzerrereistung kann mit Hilfe einer Kombination aus FFE und entscheidungsrückgekoppeltem Entzerrer (engl. Decision Feedback Equalizer, DFE) erreicht werden. Mit Hilfe des DFE kann bei relativ kurzer Filterlänge nicht nur die Wirkung des Impulsvorschwingers, sondern auch die des Impulsnachschwingers mit geringem Aufwand verringert werden. Abbildung 2.2 zeigt einen schematischen Aufbau eines kombinierten FFE und

DFE. Der FFE-Teil ist genau gleich aufgebaut wie im vorhergehenden Kapitel. Im entscheidungsrückgekoppelten Filter (DFE) befinden sich wie beim FFE ebenfalls Verzögerungsglieder und analoge Multiplizierer. Zusätzlich befindet sich darin aber noch eine Quantisierstufe, die ein wertekontinuierliches Signal in ein wertediskretes Signal wandelt. Daher wird der DFE-Teil als eine Digitalschaltung bezeichnet. Um die teilweise analoge und teilweise digitale Signalverarbeitung des Gesamtentzerrers hervorzuheben, wird eine solche Architektur als analoge Mischsignalschaltung bezeichnet. Ein Nachteil eines rückgekoppelten Filters ist die Fehlerfortpflanzung.

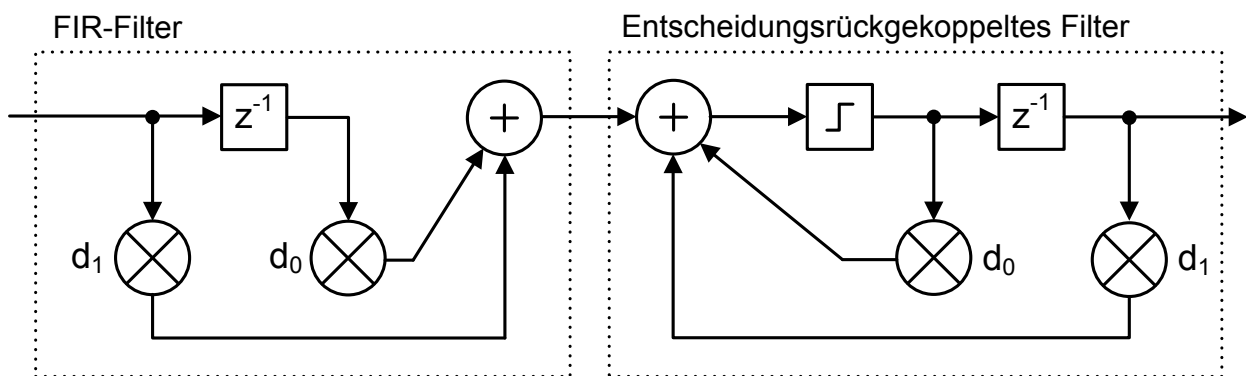


Abbildung 2.2: Blockschaltbild eines kombinierten Vorwärtsentzerrers mit entscheidungsrückgekoppeltem Entzerrerr

2.2.3 Wahrscheinlichkeitsbasierter Entzerrerr

In den beiden vorhergehenden Kapiteln sind Entzerrerr beschrieben, die im Wesentlichen einen wertekontinuierlichen Impuls eines zeitlich abgetasteten Signals mit einem oder mehreren zuvor gesendeten Impulsen unterschiedlich gewichten. Ein anderer Ansatz ist ein Entzerrungsverfahren, das auf der Wahrscheinlichkeitsberechnung basiert. Ein bekannter Vertreter eines solchen Verfahrens ist der sogenannte Maximum Likelihood (ML-) Entzerrerr. Hierbei handelt es sich um ein rein digitales Entzerrerverfahren, da das Eingangssignal zunächst mit Hilfe eines Analog-Digital-Wandlers (ADC) in ein zeit- und wertediskretes Signal gewandelt wird. Als Auflösung werden 3 bis 4 bit [22] als ausreichend betrachtet. Das digitale Signal wird einem digitalen Signalprozessor (DSP) zugeführt, der die wahrscheinlichste gesendete Bitfolge unter Betrachtung mehrerer möglicher Empfangsfolgen nach dem Schätzverfahren der höchsten Wahrscheinlichkeit (engl. Maximum Likelihood Sequence Estimation, MLSE) ermittelt. Die Bestimmung erfolgt über komplexe Rechenoperationen und nichtlineare Entscheidungen. Eine mögliche Implementierung des MLSE-Verfahrens ist der Viterbi-Algorithmus [23], [24], bei dem mehrere Pfade (mögliche Folgen empfangener Bits) entlang eines Trellis mit einer Wahrscheinlichkeit gewichtet werden und unwahrscheinliche Pfade vorab aussortiert werden. Abbildung 2.3 zeigt ein vereinfachtes Blockschaltbild eines ML-Entzerrerrers.

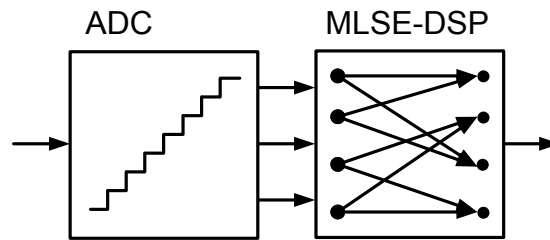


Abbildung 2.3: Blockschaltbild eines ML-Entzerrers

2.2.4 Bewertung der Entzerrerkonzepte

Eine Realisierung eines FFE mit kombiniertem DFE zur Entzerrung der PMD in optischen 43 Gbit/s Kanälen ist in [10] beschrieben. Es wird gezeigt, dass bei einer zulässigen Bitfehlerrate von 10^{-4} der alleinige Einsatz von FFE die DGD-Toleranz um 40 % gegenüber einem System ohne Entzerrung bei gleicher optischer Signalleistung erhöht. Wird zusätzlich DFE hinzugeschaltet, beträgt die DGD-Toleranz sogar 50 %. Untersuchungen in [11], [14], [25] und [26] lassen erwarten, dass bei gleicher optischer Signalleistung das MLSE-Verfahren eine deutlich höhere DGD-Toleranz als FFE und DFE aufweist. Da sich das MLSE-Verfahren als das aussichtsreichste herausstellt, wird dieses Konzept weiterverfolgt.

2.3 Der Übertragungskanal

Einem digitalen Übertragungssystem liegen zeitdiskrete Eingangssignale und zeitdiskrete Ausgangssignale zu Grunde. Dazwischen befindet sich der Kanal, der durch einen äquivalenten Tiefpass beschrieben werden kann. Darin werden sowohl der Impulsformer am Sender, der eigentliche Kanal selbst und auch das Empfangsfilter zu der zeitdiskreten Größe d_k zusammengefasst, welche als Kanalimpulsantwort bezeichnet wird. Die zu übertragenden Symbole werden als a_k , die empfangenen Symbole werden als q_k bezeichnet. Auf die Symbole z_k wird Rauschen r_k addiert. Abbildung 2.4 zeigt ein allgemeines Blockschaltbild eines digitalen Übertragungskanals.

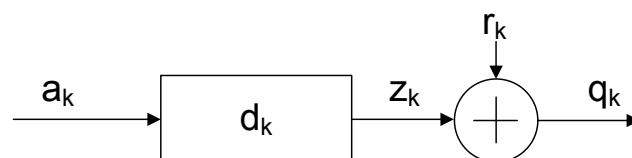


Abbildung 2.4: Allgemeines Blockschaltbild eines digitalen Übertragungskanals

Überträgt man über den Kanal aus Abbildung 2.4 eine Symbolfolge a_k erhält man das Empfangssignal [27]

$$q(t) = \sum_{i=-\infty}^{\infty} a_i d(t - iT) + r(t) \quad (2.1)$$

Das Empfangssignal $q(t)$ wird nun mit der Periode T abgetastet. Die Signaldurchlaufzeit durch den Kanal sei t_0 und wird als konstant betrachtet.

Mit Hilfe von (2.1) ergibt sich

$$q(t_0 + kT) = \sum_{i=-\infty}^{\infty} a_i d(t + kT - iT) + r(t + kT) \quad \text{bzw.} \quad (2.2)$$

$$q_k = \sum_{i=-\infty}^{\infty} a_k d(t + kT - iT) + r_k$$

Durch weitere Umformungen ergibt sich

$$q_k = a_k d(t_0) + \underbrace{\sum_{i=-\infty, i \neq k}^{\infty} a_k d(t + (k - i)T)}_{\text{Symbolnebensprechen}} + \underbrace{r_k}_{\text{additives Rauschen}} \quad (2.3)$$

Nach (2.3) besteht das Empfangssignal aus der Überlagerung aller gesendeten Symbole a_k . Dieser Effekt wird Nachbarimpulsnebensprechen bzw. Intersymbolinterferenz (ISI) genannt. Durch das Verschmieren der Sendesymbole wird auf der Empfangsseite die Augenöffnung verkleinert, was zu Fehlentscheidungen bei der Detektion führt. Eine Möglichkeit, ISI zu vermeiden, besteht darin, die Symbole in einem größeren zeitlichen Abstand zu übertragen. Dadurch würde aber die Bitrate sinken, was jedoch nicht gewünscht ist.

Der Übertragungskanal wird aus Gleichung (2.2) hergeleitet. Die Modellierung der ISI erfolgt in Form einer Kette von Speichergliedern, deren Ausgänge mit der zeitdiskreten Impulsantwort des Kanals gewichtet werden. In realen Systemen wirkt sich die ISI auf die direkt benachbarten Bits stärker aus als auf zeitlich weiter entfernte Bits. Daher kann sie nach einer bestimmten Zeit vernachlässigt werden. Es reicht aus, den Übertragungskanal als zustandsbehafteten Kanal mit zwei Speichern und Rauschen zu modellieren (Abbildung 2.5) [26], [28]. In diesem Fall spricht man von einem Finite-Impulse-Response-Kanal (FIR). Dabei wird der Hauptimpuls sowohl durch das vorherige als auch durch das nachfolgende Symbol beeinflusst. Die Beeinflussung wird über die Gewichtungsfaktoren d_{-1} , d_0 und d_{+1} beschrieben. Dabei entspricht der Index 0 dem Hauptimpuls, während die Indizes -1 dem Vorgängerimpuls bzw. $+1$ dem nachfolgenden Impuls entsprechen. Die beiden Speicher z^{-1} enthalten die nacheinander eingeschriebenen Symbole, welche dann mit den Gewichtungsfaktoren multipliziert und danach zu z_k aufsummiert werden. Dadurch wird die Impulsverbreiterung modelliert.

Das Zustandsübergangsdiagramm (Markov-Modell) eines Kanals aus Abbildung 2.5 kann man allgemein wie folgt ableiten. Die Anzahl der Zustände hängt von der Anzahl der Speicher-

glieder l und der Stufigkeit M des Eingangssignals a_k ab. Die Anzahl der Zustände N_S lässt sich allgemein mit

$$N_S = M^l \quad (2.4)$$

bestimmen.

Die Anzahl der Zustandsübergänge N_T ergibt sich zu

$$N_T = M^{l+1} \quad (2.5)$$

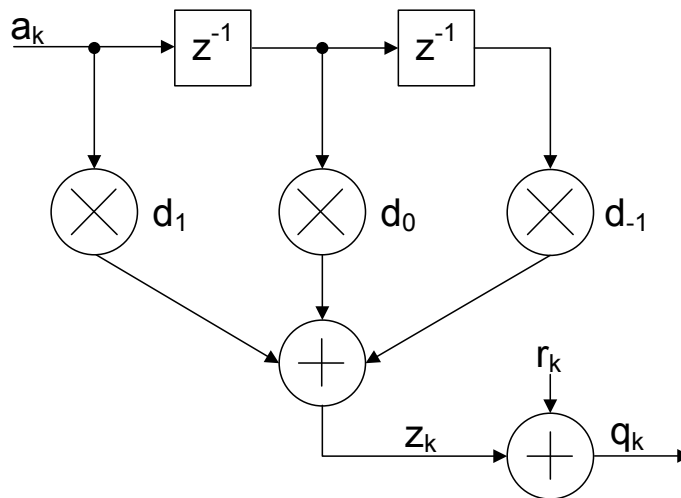


Abbildung 2.5: Kanalmodell eines FIR-Kanals zweiter Ordnung

Bei einem binären ($M = 2$) Eingangssignals entspricht ein Symbol genau einem Bit. Der Kanalgrad $l = 2$ entspricht der Anzahl der Speicherglieder. Daraus ergibt sich das Zustandsübergangsdiagramm in Abbildung 2.6 mit den vier Zuständen S_0 bis S_3 und acht Zustandsübergängen μv . Es ist zu erkennen, dass nicht alle Zustandsübergänge möglich sind, was darauf zurückzuführen ist, dass immer nur ein Bit a_k in den Speicher hineingeschoben wird. Das Bit, das im rechten Speicher steht, wird durch das Bit des linken Speichers ersetzt. Die entstehenden Symbole z_k sind also korreliert. Daraus folgt, dass bei diesem Kanalmodell genau acht Symbole z_k generiert werden können.

Zu den mit ISI versehenen Symbolen z_k wird Rauschen r_k addiert, woraus sich auf Empfangsseite die Empfangsfolge q_k ergibt. Es gilt:

$$q_k = z_k + r_k \quad (2.6)$$

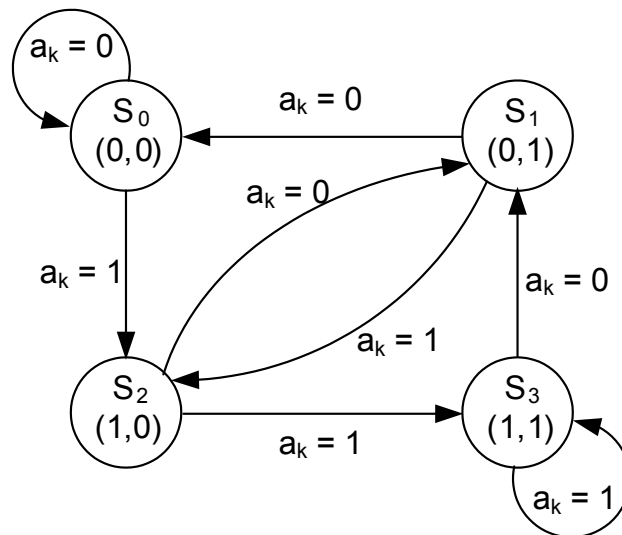


Abbildung 2.6: Zustandsübergangsdiagramm eines Kanals zweiter Ordnung

Trägt man nun alle möglichen Zustandsübergänge eines Zustandsübergangsdiagramms über der Zeit auf, erhält man ein Trellisdiagramm. Unter einem Trellis versteht man einen gerichteten Graphen mit Gewichtungen. Abbildung 2.7 zeigt ein Beispiel eines Trellisdiagramms mit drei Zeitpunkten, an dem die Kanalkodierung verdeutlicht werden soll. Betrachtet man einen Kanal ohne Rauschen, ergeben sich aufgrund der Faltung mit der Impulsantwort die acht Symbole

$$z_{\mu,v,k} = a_k * d_k \quad \forall \mu, v \text{ mit } \mu \in \{0,1,2,3\}, v \in \{0,1\} \quad (2.7)$$

nach Tabelle 2.1. Dabei ist μ die Zustandsnummer und v beschreibt den Zustandsübergang.

Tabelle 2.1: Kanalimpulsantwort in Abhängigkeit von gesendeten Bits

Zustand zur Zeit k	gesendetes Bit	Kanalimpulsantwort
S_0	$a_k = 0$	$z_{00,k} = 0$
	$a_k = 1$	$z_{01,k} = d_1$
S_1	$a_k = 0$	$z_{10,k} = d_{-1}$
	$a_k = 1$	$z_{11,k} = d_1 + d_{-1}$
S_2	$a_k = 0$	$z_{20,k} = d_0$
	$a_k = 1$	$z_{21,k} = d_1 + d_0$
S_3	$a_k = 0$	$z_{30,k} = d_0 + d_{-1}$
	$a_k = 1$	$z_{31,k} = d_1 + d_0 + d_{-1}$

Je nach Startzustand, der dem Speicherinhalt des Kanals zu Beginn entspricht, und einer gegebenen Folge a_k , ergibt sich ein bestimmter Pfad durch das Trellisdiagramm.

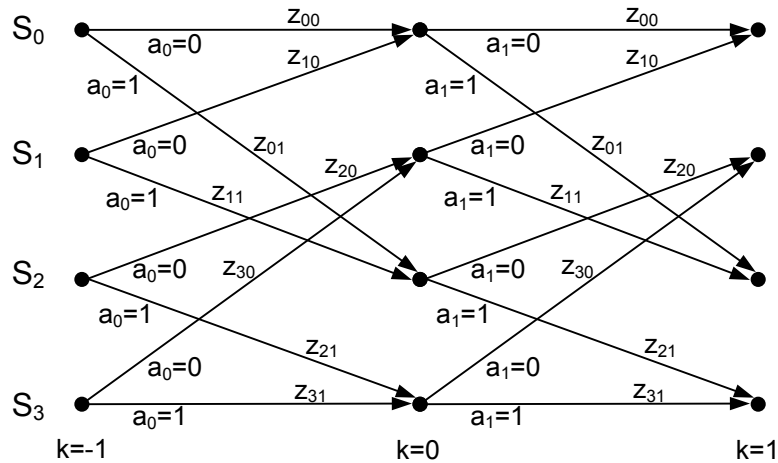


Abbildung 2.7: Trellisdiagramm eines Kanals zweiter Ordnung

2.4 Schätzung der wahrscheinlichsten Empfangsfolge

Am Empfänger möchte man die ursprünglich gesendete Bitfolge a_k wieder erkennen. Dazu bedient man sich einem Schätzverfahren, das die möglichen Pfade durch ein Trellisdiagramm hindurch anhand von Wahrscheinlichkeiten auswertet. Nach (2.3) erfahren die Bits einerseits ISI und andererseits Rauschen. Die Erkennung erfolgt aufgrund des Kanalgedächtnisses am besten, wenn man Entscheidungen nicht nur anhand von einzelnen Symbolen vornimmt, sondern eine Symbolfolge heranzieht. Die Schätzung der wahrscheinlichsten Empfangsfolge (engl. Maximum Likelihood Sequence Estimation, MLSE) erfolgt auf Basis des Kanalgedächtnisses, indem die Wahrscheinlichkeit aufeinander folgender Symbole bestimmt wird.

Nimmt man mittelwertfreies weißes Gaußsches Rauschen mit der Varianz σ_0^2 an, errechnet sich die Wahrscheinlichkeit, dass das Symbol q_k empfangen wird unter der Bedingung, dass z_k mit ISI gesendet wurde, mit Hilfe der Wahrscheinlichkeitsdichtefunktion (2.8) zu [29]

$$p(q_k | z_k) = \frac{1}{\sqrt{2\pi\sigma_0}} e^{-\frac{1}{2\sigma_0^2}(q_k - z_k)^2} \quad (2.8)$$

Legt man eine Symbolfolge der Länge K zu Grunde, ergibt sich die Gesamtwahrscheinlichkeit, dass eine bestimmte Folge a_k gesendet wurde, aus dem Produkt der Einzelwahrscheinlichkeiten mit Hilfe der Wahrscheinlichkeitsdichtefunktion (2.9).

$$p(q_0, q_1, \dots, q_{K-1} | z_0, z_1, \dots, z_{K-1}) = \prod_{k=0}^{K-1} p(q_k | z_k) = \left(\frac{1}{\sqrt{2\pi\sigma_0}} \right)^K e^{-\frac{1}{2\sigma_0^2} \sum_{k=0}^{K-1} (q_k - z_k)^2} \quad (2.9)$$

Somit ist es möglich, mit Hilfe dieses Schätzverfahrens die wahrscheinlichste Bitfolge (engl. Maximum Likelihood Sequence) eines gesendeten Signals zu bestimmen.

Der Teilausdruck im Exponenten aus (2.9)

$$(q_k - z_k)^2 \quad (2.10)$$

beschreibt den quadratischen Abstand des Empfangssignals q_k zu dem erwarteten Symbol z_k (Erwartungswert). Alle anderen Ausdrücke in (2.9) sind konstant. Daher beschränkt sich die Berechnung auf die Methode des kleinsten Fehlerquadrates. Daraus lässt sich das ML-Funktional ableiten:

$$L_{\mu v, k_0} = \sum_{k=0}^{k_0} |q_k - z_{\mu v}|^2 \quad (2.11)$$

$$= \sum_{k=0}^{k_0} |q_k - d_k * a_{\mu, k}|^2 \quad (2.12)$$

$$= \sum_{k=0}^{k_0} p_{\mu v, k} \quad (2.13)$$

Dabei geht die Bildung des kleinsten Fehlerquadrats aus (2.10) in die so genannten Pfadkosten $p_{\mu v, k}$ über. $L_{\mu v, k_0}$ entspricht den Summenpfadkosten bis zum Zeitpunkt k_0 , μ ist der Zustand und v der Zustandsübergang, der durch das gesendete Symbol $a_{\mu, k}$ hervorgerufen wird.

Um nun den kürzesten Pfad durch das Trellisdiagramm zu ermitteln, müssen die Summenpfadkosten nach Gleichung (2.12) über alle möglichen Bitfolgen a_k bzw. q_k bestimmt werden.

$$\begin{aligned} L_{\mu v, k} &= L_{\mu v, k-1} + |q_k - z_{\mu v, k}|^2 \\ &= L_{\mu v, k-1} + p_{\mu v, k} \end{aligned} \quad (2.14)$$

Die Gleichung (2.14) führt zu einer Rekursion, da die neuen Summenpfadkosten immer von den Pfadkosten aus dem vorhergehenden Zeitschritt $k - 1$ abhängen.

2.4.1 Viterbi-Algorithmus

Der Nachteil des MLSE-Verfahrens besteht darin, dass alle möglichen Pfade durch das Trellis hindurch verfolgt werden müssen, was zu einem entsprechend hohen Schaltungsaufwand führt. Ein Beispiel für einen Algorithmus, der alle Pfade durch einen Graphen hindurch berücksichtigt,

ist der Dijkstra-Algorithmus [30]. Dieser Algorithmus basiert auf der Lösung des Problems des Handelsreisenden nach Kruskal [31]. Diese beiden Algorithmen finden vorzugsweise ihre Anwendung in GPS-basierten Navigationsgeräten.

A. Viterbi stellte 1967 dagegen einen Algorithmus zur Dekodierung von Faltungscodes vor [23], bei dem nicht mehr alle Pfade verfolgt werden müssen. Fünf Jahre später erkannte G.D. Forney [24], dass sich dieser Algorithmus zur Dekodierung von ML-Sequenzen, die aus Intersymbolinterferenzen resultieren, eignet. Somit handelt es sich beim Viterbi-Algorithmus um einen Spezialfall der MLSE [29], [32], bei dem immer nur die wahrscheinlichsten Pfade, die über die kumulierten Metriken ermittelt werden, zu jedem diskreten Zeitschritt k gespeichert werden. Mathematisch besteht der Viterbi-Algorithmus aus der Berechnung der euklidischen Distanzen (Zweigmetriken) aller möglichen Pfade zu einer beobachteten Empfangsfolge.

Daher lässt sich Gleichung (2.14) wie folgt vereinfachen:

$$P_{\mu,k} = \min_v \{ P_{\mu,k-1} + p_{\mu v,k-1} \} \quad (2.15)$$

Anstelle der Summenpfadkosten $L_{\mu v,k}$ stehen jetzt bereits die vorentschiedenen Summenpfadkosten $P_{\mu,k}$. Die rekursive Berechnung und vorzeitige Aussortierung unwahrscheinlicher Pfade nach Gleichung (2.15) sind die Kerngedanken des Viterbi-Algorithmus. Das heißt, wenn man wie in diesem Beispiel von zwei möglichen Zustandsübergängen ausgeht, reduziert sich der Auswerteaufwand pro Stufe um die Hälfte.

Es handelt sich dabei um ein Verfahren, bei dem nicht mehr alle möglichen Wege durch das Trellisdiagramm hindurch gespeichert werden müssen. Trotzdem ist es ein mathematisch korrektes Verfahren, da die Vergangenheit mitberücksichtigt wird. Der Viterbi-Algorithmus sortiert frühzeitig Pfade aus, die am Ende einer Aufsummierung voraussichtlich nicht mehr minimal werden können. Zur Schätzung der am wahrscheinlichsten gesendeten Symbolfolge muss der Pfad mit der geringsten Distanz ausgewählt werden.

Die Gleichung (2.15) beschreibt eine Addier-Vergleichs-Auswahloperation (engl. Add-Compare-Select-Operation, ACS-Operation), bei der die neue Summenpfadmetrik $P_{\mu,k}$ nach Addition von Zweigmetriken $p_{\mu v,k-1}$ und der Pfadmetrik des vorhergehenden Zustands $P_{\mu,k-1}$ mit der minimalen Distanz ausgewählt wird. Ein Blockschaltbild einer Addier-Vergleichs-Auswahleinheit (engl. Add-Compare-Select-Unit, ACSU) ist in Abbildung 4.21 dargestellt.

Ein wichtiger Aspekt, der noch zu diskutieren ist, ist die Länge des Trellis, die zur Entzerrung mit ausreichender Bitfehlerrate benötigt wird. Mathematisch gesehen, handelt es sich bei den Beziehungen (2.14) und (2.15) um rekursive Berechnungsvorschriften, woraus ein unendlich langer Trellis resultiert. Der Viterbi-Algorithmus dient zur Dekodierung von Intersymbol-

interferenzen. Die ISI wirkt sich in einem realen System aber nicht unendlich lang aus, weshalb die rekursive Berechnung nach einer bestimmten Schrittzahl abgebrochen werden kann.

2.4.1.1 Blockdekodierverfahren

Der Viterbi-Algorithmus betrachtet eine unendlich lange Symbolfolge. Eine hardwaretechnische Umsetzung für unendlich lange Symbolfolgen ist aber nicht realisierbar, da die Kapazität jeder Hardware begrenzt ist. Da sich aber die ISI zeitlich betrachtet nicht unendlich lang auswirkt, reicht es aus, nur einen Abschnitt der zu entzerrenden Symbolfolge zu betrachten [33], [34]. Daher ist es möglich, so genannte Blockdekodierverfahren [35] anzuwenden. Abbildung 2.8 zeigt einen Ausschnitt eines Trellisdiagramms mit der Blocklänge $2L$ aus dem die Vereinigung mehrerer Pfade während der Pfadrückverfolgung verdeutlicht wird. Die Symbolfolge wird zunächst als unendlich lang betrachtet. Die überlebenden Pfade sind in schwarz und die verworfenen Pfade sind in grau dargestellt.

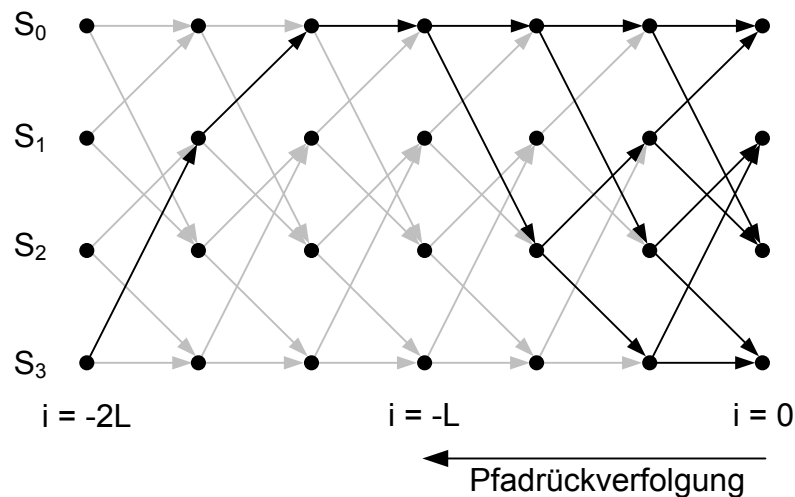


Abbildung 2.8: Ausschnitt eines Datenblocks in einem Trellisdiagramm

Geht man vom Zeitschritt $i = 0$ von einem beliebigen Zustand aus L Schritte zurück, haben sich mit hoher Wahrscheinlichkeit alle Pfade zu einem überlebenden Pfad vereinigt. Daraus folgt, dass sich der Zustand zum Zeitschritt $i = -L$ dekodieren lässt.

Da aufgrund von Hardwarebeschränkungen keine unendlich lange Symbolfolge vorgehalten werden kann, muss man einen Beginn der Symbolfolge festlegen. Das heißt, dass man mit der Trellisverarbeitung mit beliebigen, aber festen Startmetriken beginnt. Durchläuft man den Trellis mit vorzugsweise zu Null initialisierten Startmetriken, sind die Metriken nach L Trellisstufen unabhängig von den Startmetriken. Anders ausgedrückt, nach L Schritten kann mit hoher Wahrscheinlichkeit der überlebende Pfad bestimmt werden. In diesem Fall bezeichnet man den Parameter L als Synchronisationslänge [36]. Abbildung 2.9 zeigt einen Block, dessen Verarbeitung mit zu Null initialisierten Metriken beginnt.

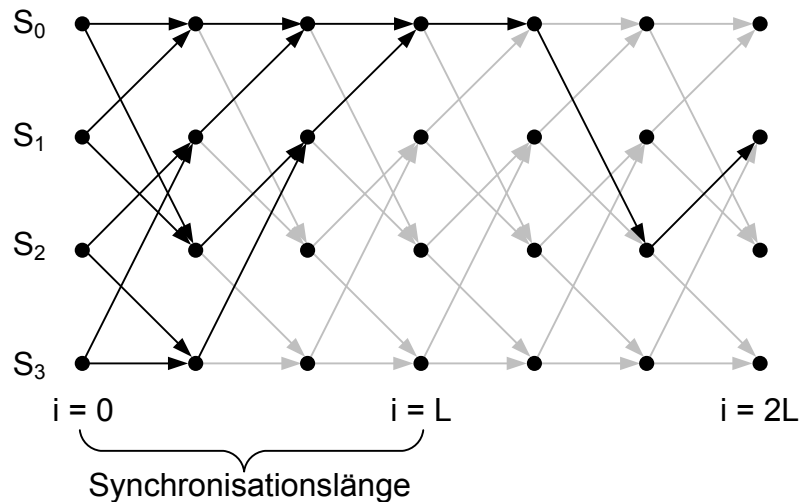


Abbildung 2.9: Pfadvereinigung mit Initialmetriken

Daraus folgt, dass zur Dekodierung eines einzigen Zustandes bzw. Eingangssymbols $2L$ Symbole (L für Synchronisation und L für Pfadrückverfolgung bei begrenzten Blocklängen) durch einen Trellis verarbeitet werden müssen. Will man mehr als ein Symbol gleichzeitig dekodieren, muss die Blocklänge weiter erhöht werden.

Nun stellt sich die Frage, welche Größe der Parameter L annehmen muss, damit das Eingangssignal fehlerfrei bzw. mit einer tolerierten Bitfehlerrate rekonstruiert werden kann. In der Literatur [32] wird als maximale Pfadvereinigungslänge L das Fünffache des Kanalgedächtnisses l

$$L \leq 5l \quad (2.16)$$

angesetzt. Es handelt sich dabei um einen Erfahrungswert. Eine exakte Angabe ist nicht möglich, da die tatsächliche Pfadvereinigungslänge von der Wahl der Kanalkoeffizienten und von Startbedingungen abhängig ist. Die Trellislänge orientiert sich an der Pfadvereinigungslänge. Höhere Trellislängen haben zwar keine nachteiligen Auswirkungen auf die Bitfehlerrate, erhöhen aber nur unnötig die Durchlaufzeit und den Chipflächenbedarf.

Anhand dieser Erkenntnis ist es möglich, Blockdekodierverfahren anzuwenden, bei denen immer nur bestimmte Abschnitte einer zu entzerrenden Symbolfolge verarbeitet werden. Im Wesentlichen gibt es die Minimalmethode [7] und die Schiebefenstermethode [13]. Die Schiebefenstermethode weist die besseren Eigenschaften hinsichtlich Schaltungsaufwand und Bitfehlerrate auf [13], [37]. Daher wird im Folgenden die Schiebefenstermethode weiterverfolgt.

2.4.1.2 Parallelisierung

Die Möglichkeit den Datenstrom in Blöcke zu unterteilen, ist der Schlüssel zur Parallelisierung des Viterbi-Algorithmus. Um eine Verarbeitungsgeschwindigkeit von mehreren Gbit/s zu erreichen, ist eine Parallelisierung unerlässlich. Daher ist es notwendig, den Datenstrom in

mehrere quasi-unabhängige Symbolblöcke zu unterteilen und diese Blöcke gleichzeitig zu verarbeiten. Es muss nun eine sinnvolle Blocklänge gefunden werden, die zur ISI passt, damit MLSE funktioniert. Anschließend ist anhand von Simulationen in der eingesetzten CMOS-Technologie zu entscheiden, wie viele dieser Blöcke parallel geschaltet werden müssen, um die Zieldatenrate von 43 Gbit/s zu erreichen.

Die Verarbeitung der Symbole kann sogar gleichzeitig in Vorwärtsrichtung und in Rückwärtsrichtung erfolgen. Dass dies möglich ist, kann dadurch begründet werden, dass die Weglänge von einem Ort A zu einem Ort B und umgekehrt gleich sein muss. Daher gibt es nur einen kürzesten Weg zwischen den Orten A und B, der von der Richtung her unabhängig ist. Aus diesem Grund reicht es aus, im Trellisdiagramm, welches einem gerichteten und gewichteten Graphen entspricht, die Pfeilrichtungen zu vertauschen. Die Kosten bzw. Metriken der Zustandsübergänge bleiben gleich, da die Weglänge unabhängig von der Richtung ist. Zur Verdeutlichung zeigt Abbildung 2.10 jeweils ein Trelliselement für die Vorwärtsverarbeitung (links) mit der Wegstrecke von A nach B und für die Rückwärtsverarbeitung (rechts) mit der Wegstrecke von B nach A. Die Kodierung der Zustandsübergänge $p_{\mu\nu}$ ist folgendermaßen gewählt: Bei der Vorwärtsverarbeitung entspricht μ dem Zielzustand und ν der Nummer des eingehenden Zustandsübergangs bezogen auf das Ziel.

Es gilt:

$$\begin{aligned} v = 0 & \text{ wenn } \mu_A \text{ gerade} \\ v = 1 & \text{ wenn } \mu_B \text{ ungerade} \end{aligned} \quad (2.17)$$

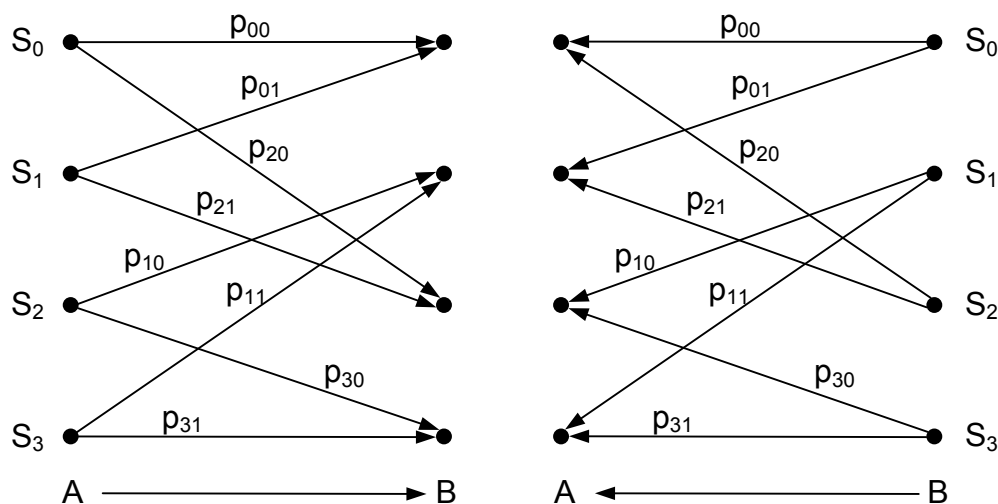


Abbildung 2.10: Ausschnitt aus dem Trellisdiagramm für Vorwärts- (links) und Rückwärtsverarbeitung (rechts)

Bei der Rückwärtskodierung hingegen entspricht μ dem Ursprungszustand und ν der Nummer des ausgehenden Zustandsübergangs bezogen auf den Ursprung. Die Beziehung (2.17) gilt daher genauso, nur mit der vertauschten Bedeutung von A und B.

Alternativ kann anstelle der Unterteilung in mehrere Blöcke auch ein großer Block, der die Symbole in Vorwärts- und Rückwärtsrichtung verarbeitet, verwendet werden. Da die Blocklänge gleich der Blocktiefe ist, ergibt sich bei einer Verdopplung der Blocklänge automatisch eine Verdopplung der Blocktiefe. Dadurch verdoppelt sich die Durchlaufzeit, während sich der Aufwand an Flipflops sogar vervierfacht. Daher ist es günstiger, die Blocklänge so kurz zu halten, wie es das Kanalgedächtnis erfordert. Eine Verlängerung der Blocklänge, was einer Verlängerung der Trellislänge entspricht, hätte nur eine geringe bis gar keine positive Auswirkung auf die Bitfehlerrate. Die gleichzeitige Vorwärts- und Rückwärtsverarbeitung verringert nochmals die Durchlaufzeit und die Anzahl an Flipflops.

2.4.1.3 Schiebefensterverfahren

Die Erkenntnisse der beiden vorherigen Kapitel lassen sich kombinieren. Es gilt, dass alle Pfade durch einen Trellis bei geeignet gewählten Pfadkosten mit hoher Wahrscheinlichkeit nach $L = 5l$ Schritten in einen überlebenden Pfad münden [33]. Andererseits wird eine ebenfalls $5l$ lange Symbolfolge zur Synchronisation benötigt [36]. Damit lässt sich nun innerhalb des Abschnitts $i - L$ bis $i + L$ das Symbol zum Zeitschritt i dekodieren. Bezieht man das Dekodierresultat auf den Aufwand, kann die Dekodiereffizienz η_D nach (2.18) angegeben werden:

$$\eta_D = \frac{1}{2L} \quad (2.18)$$

Das Ziel ist aber mehr als nur ein Symbol gleichzeitig zu dekodieren. Um die Effizienz der Dekodierung zu steigern, muss die Blocklänge L , welche im Wesentlichen mit der Trellislänge zusammenhängt, erhöht werden. Durch die Anwendung gleichzeitiger Vorwärts- und Rückwärtsdekodierung kann die Blocklänge verdoppelt werden, wenn am jeweiligen Ende der beiden Teiltrellis die Summe der Metriken zueinander gehörender Zustände gebildet wird. Der Zustand, an dem die kleinste gemeinsame Summe anliegt, liegt auf dem überlebenden Pfad.

Der Übertragungskanal wird mit $l = 2$ Verzögerungsgliedern modelliert. Daraus ergibt sich für die Länge des überlebenden Pfades ein Wert von $L = 10$. Die Länge des überlebenden Pfades lässt sich auf $2,5l$ ohne signifikanten negativen Einfluss auf die Bitfehlerrate reduzieren [34]. Durch eine geschickte Wahl der Zweigmetriken und einer tolerierten Bitfehlerrate von 10^{-3} wird als Synchronisationslänge wie auch als Pfadvereinigungslänge $L = 4$ als ausreichend betrachtet [37]. Entwirft man eine Architektur mit einer Trellislänge von $B = 2L = 8$ in jeder Richtung, ergibt sich die Dekodiereffizienz zu

$$\eta_D = \frac{B}{B + 2L} = \frac{1}{2} \quad (2.19)$$

Nach (2.19) bedeutet dies, dass insgesamt 16 Symbole zur Dekodierung von 8 Symbolen benötigt werden. Beim Schiebefensterverfahren wird die Symbolfolge in Blöcke unterteilt, deren enthaltene Symbole dann blockweise durch den Trellis verarbeitet werden. Im vorliegenden Fall wird die Trellislänge einer Richtung zu $2L = 8$ gewählt. Das bedeutet, dass eine Gesamtblock- bzw. Trellislänge von 16 zur Verfügung steht, aus der dann $B = 8$ Symbole gleichzeitig dekodiert werden können. Damit im dekodierten Datenstrom keine Lücke entsteht, wird das zu dekodierende Symbolfenster immer nur um 8 Symbole pro Zeitschritt weitergeschoben. Die Durchlaufzeit der Symbole durch einen Blockdekoder beträgt $4L/f_{\text{clk}}$. Diese Zeit resultiert aus dem quadratischen Aufbau des Blockdekoders, auf dessen Diagonale der Trellis liegt. In Abbildung 2.11 wird dieser Zusammenhang verdeutlicht.

Zur weiteren Steigerung des Durchsatzes können mehrere sogenannte systolische Blockdekoder (in Abbildung 2.11 grau hinterlegt) der Länge 8 parallel betrieben werden, deren Enden miteinander verbunden sind. Am jeweiligen Ende eines Blockes erfolgt eine Schätzung des gemeinsam kürzesten, überlebenden Pfades. Der detaillierte Aufbau eines systolischen Blockdekoders (SBD) ist in Abbildung 2.12 dargestellt. Mögliche Architekturen mit einer seriellen Eingangsdatenrate von 43 Gbit/s sind in Tabelle 2.2 wiedergegeben.

Tabelle 2.2: Mögliche Architekturentwürfe für den Viterbi-Entzerrer

Dekodierblocklänge B	Parallelbetrieb SBD	Taktfrequenz f_{clk} eines SBDs	Anzahl ACSUs	Anzahl Register pro SBD
8	1	5375 MHz	64	256
8	2	2688 MHz	128	512
8	4	1344 MHz	256	1024
8	8	672 MHz	512	2048
16	1	2688 MHz	128	1024
16	2	1344 MHz	256	2048
16	4	672 MHz	512	4096
16	8	336 MHz	1024	8192

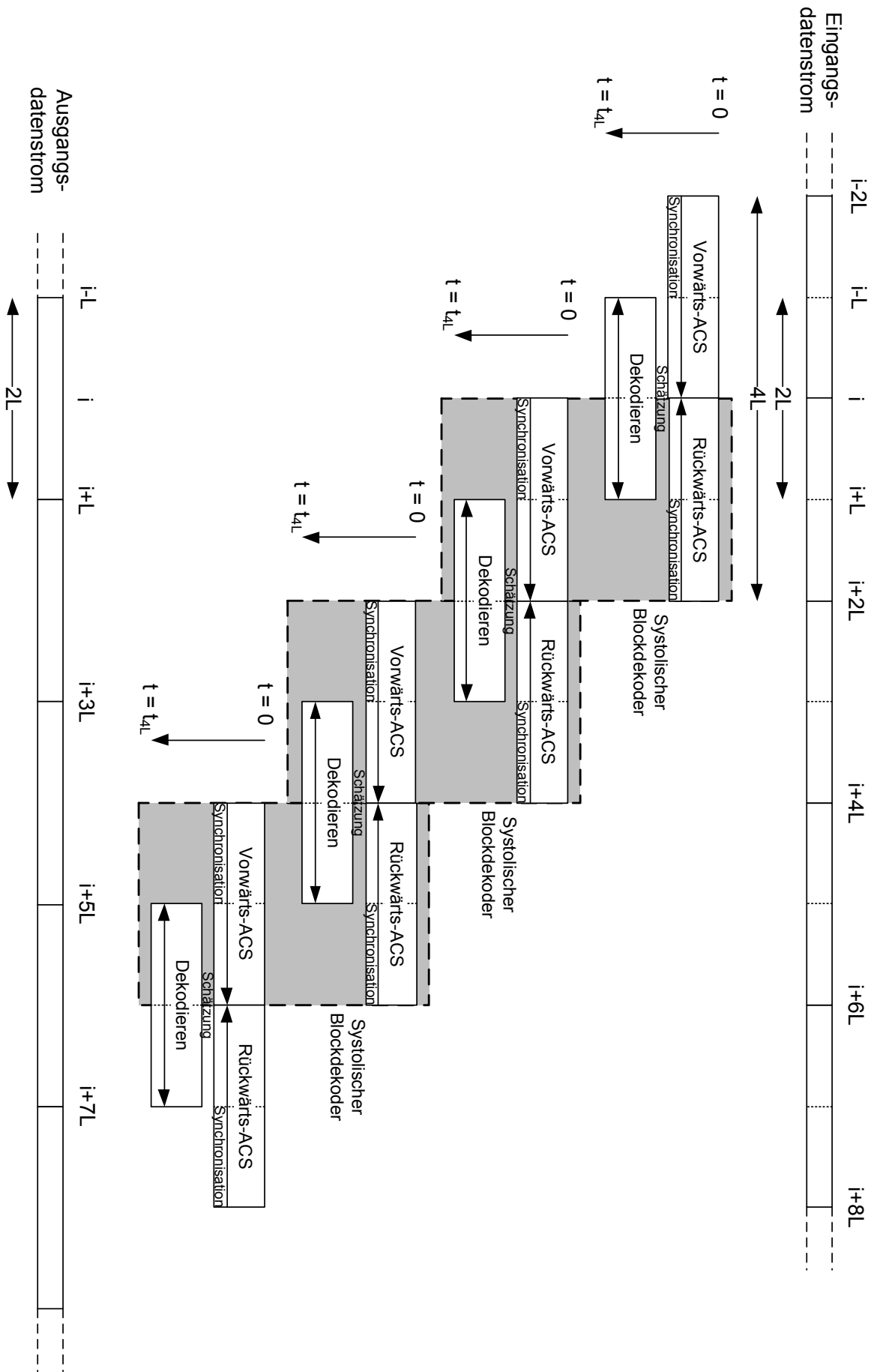


Abbildung 2.11: Schiebefensterverfahren

Möchte man mit dem Viterbi-Entzerrer beispielsweise 32 Symbole gleichzeitig verarbeiten, bieten sich die beiden in Tabelle 2.2 hervorgehobenen Architekturen an. Beide Varianten arbeiten mit einer Taktfrequenz von 1,34 GHz und benötigen 256 ACSUs. Der Aufwand an Registern ist aber bei einer Dekodierblocklänge von $B = 16$ doppelt so hoch wie bei $B = 8$. Die Durchlaufverzögerung ist bei $B = 16$ ebenfalls doppelt so groß. Der Vorteil der Architektur mit einer Dekodierblocklänge von $B = 16$ ist, dass die Pfadvereinigungslänge bzw. Synchronisationslänge L ebenfalls verdoppelt wird, was zu einer geringeren Fehlerrate führt. Eine Verlängerung von L ist aber nach Simulationen nicht nötig [37]. Die Anzahl der Register pro SBD ist ein Schätzwert, der sich aus dem Quadrat der vierfachen Pfadrückverfolgungslänge ergibt. Die tatsächliche Registeranzahl wird sich aber noch erhöhen, da für die Schätzung des kürzesten Pfades an den beiden Trellisenden weitere Register notwendig sind.

Wie viele SBDs letztendlich parallel betrieben werden müssen, muss anhand von Simulationen in der Zieltechnologie entschieden werden. Erst dann kann die Taktfrequenz der kombinatorischen Logik festgelegt werden.

2.4.1.4 Der systolische Blockdekoder

Es bietet sich an, die Verarbeitung der Symbole entlang eines Trellis als systolisches Feld aufzubauen. Die Bezeichnung leitet sich daraus ab, dass die Daten parallel durch ein Zellenfeld über Pipeline-Register mit der Taktfrequenz f_{clk} weitergereicht werden. Die Vorteile in der Verwendung systolischer Felder liegen im regulären Aufbau lokal miteinander verbundener Zellen, die einfache Rechenoperationen durchführen. Somit lassen sich systolische Felder relativ leicht und kosteneffizient mehrfach innerhalb komplexer Strukturen wiederverwenden. Die lokale Verdrahtung und der Einsatz von Pipeline-Registern ermöglichen einen hohen Datendurchsatz [38]. Ein systolischer Blockdekoder (SBD), der einen vorwärts- und rückwärtsgerichteten Trellis enthält, ist in Abbildung 2.12 dargestellt.

Die Daten des seriellen Datenstroms müssen zunächst um die Breite eines oder mehrerer SBD parallelisiert werden und in Eingangsflipflops zwischengespeichert werden. In einer sogenannten Zweigmetrikrecheneinheit (engl. Branch-Metric-Unit, BMU) wird aus den Eingangssymbolen nach der Methode des kleinsten Fehlerquadrates eine Metrik berechnet, die eine Übergangswahrscheinlichkeit zwischen zwei Zuständen eines Zustandsübergangsdiagramms beschreibt. Im vorliegenden Fall eines Kanals, der mit zwei Verzögerungsgliedern modelliert ist, gibt es insgesamt acht Zustandsübergänge. Daher werden in einer BMU acht Wahrscheinlichkeiten, die vom quantisierten Eingangssymbol und von der erwarteten Kanalimpulsantwort abhängen, bestimmt (vgl. Gleichung (2.10) und Tabelle 2.1). Entlang des Trellis, der einem zeitlich aufgefalteten Zustandsübergangsdiagramm entspricht, werden zu jedem Zeitschritt i neue Wahrscheinlichkeiten bestimmt und mit den Wahrscheinlichkeiten vorherliegender Zeitpunkte aufaddiert und verglichen. Dies entspricht allgemein der Bestimmung der Verbundwahrscheinlichkeit aus Gleichung (2.9).

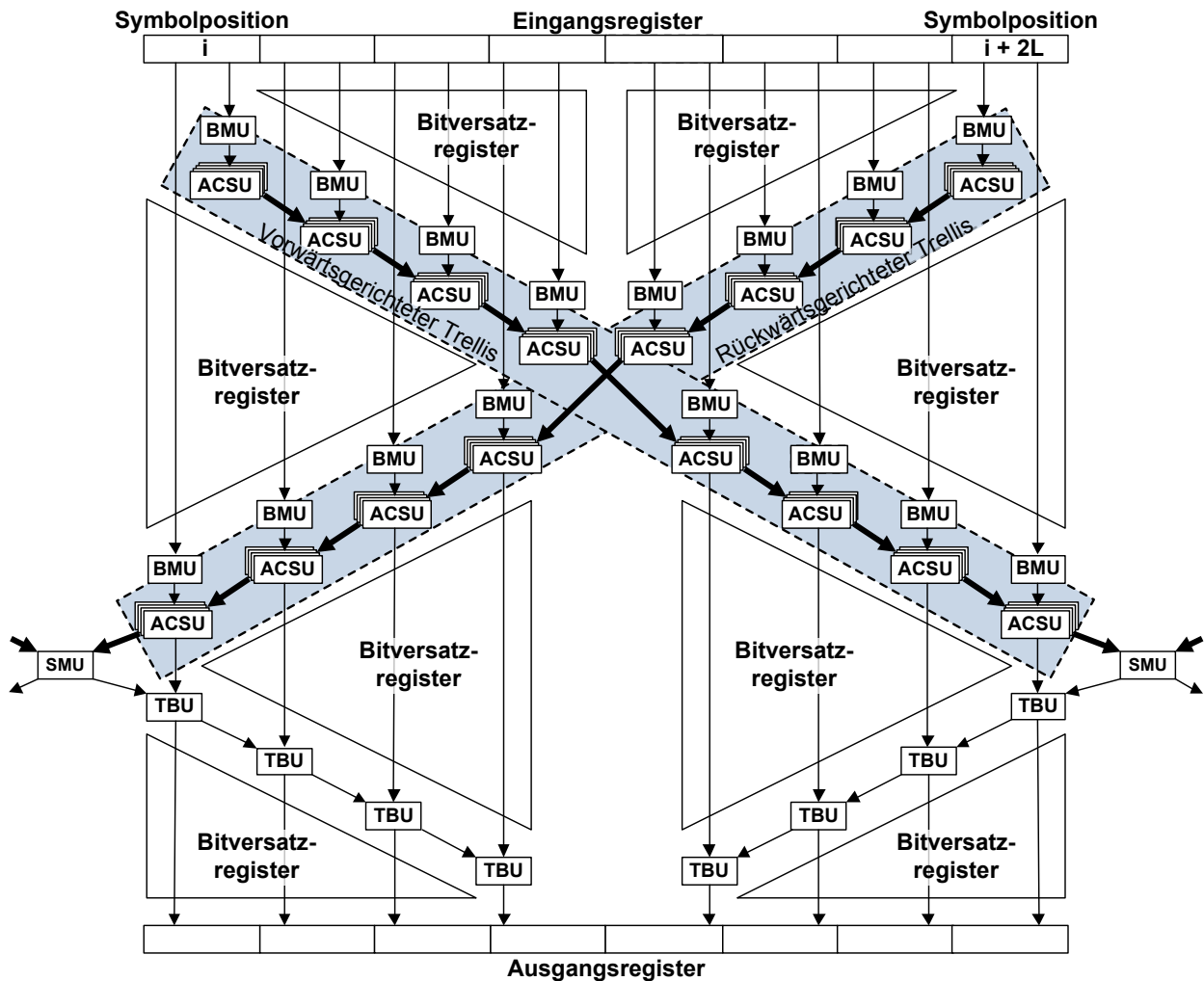


Abbildung 2.12: Systolischer Blockdekoeder

Im vorliegenden Fall wird die Multiplikation durch Logarithmieren der Metriken in eine Summe überführt, wodurch die Berechnung nach dem ML-Funktional (vgl. Gleichungen (2.11) bis (2.13)) erfolgt. Die Berechnung der Verbundwahrscheinlichkeiten geschieht in den Addier-Vergleichs-Auswahleinheiten (ACSU), wovon sich in jedem Zustand bzw. Knoten in einem Trellis eine davon befindet.

Der Kanal wird im gegebenen Fall mit zwei Verzögerungsgliedern modelliert, weshalb bei einem binären Signal das Zustandsübergangsdiagramm vier Zustände enthält. Jeder dieser vier Zustände wird durch eine ACSU repräsentiert. Beim Viterbi-Algorithmus gehen nur die Zustandsübergänge mit der höchsten Wahrscheinlichkeit bzw. den geringsten Kosten in die weitere Berechnung ein. Die lokale Vorauswahl des Pfades mit den geringsten Kosten erfolgt ebenfalls in den ACSUs.

In den ACSUs werden die eingehenden Metriken der Vorgängerzustände mit jenen des aktuellen Symbols addiert und verglichen. Dann wird entschieden, über welchen von zwei Vorgänger-

zuständen die höchste Wahrscheinlichkeit zustande gekommen ist. Die Adresse des Vorgängerzustandes wird gespeichert und die neue Metrik mit der höheren Wahrscheinlichkeit an die ACSU im nächsten Zeitschritt weitergeleitet. Die Verarbeitung der Symbole erfolgt zeitlich nacheinander. Bitversatzflipflops verzögern die Eingangssymbole so, dass sie zum richtigen Zeitpunkt an der jeweiligen BMU bzw. ACSU anliegen. Der Trellis verläuft diagonal innerhalb eines SBDs. Nach acht Verarbeitungsschritten treffen die Metriken am Ende des Trellis auf die Metriken eines benachbarten SBDs. In der Minimumauswahlschaltung (engl. Select-Minimum-Unit, SMU) werden die Metriken beider SBDs addiert. Der Zustand, auf dem die kleinste der vier Summen liegt, ist der überlebende Pfad, sofern man die Wahrscheinlichkeiten als Kosten interpretiert. Die SMU gibt die Adresse des Zustandes bzw. der ACSU zurück auf dem der global kürzeste Pfad, bezogen auf die Blocklänge $2L$, liegt. Dieser Pfad wird nun jeweils zur Hälfte der Blocklänge mittels Pfadrückverfolgungseinheiten (engl. Trace-Back-Units, TBU) zurückverfolgt und die ACSU-Entscheidungen ausgelesen. Aus den ACSU-Entscheidungen wird das dekodierte Bit gewonnen, das am wahrscheinlichsten dem gesendeten Bit entspricht. Weitere Bitversatzflipflops zwischen den ACSUs und TBUs sowie zwischen den TBUs und den Ausgangsflipflops stellen sicher, dass die Daten innerhalb des SBDs synchron verzögert werden.

Im nächsten Kapitel wird auf den CMOS-Schaltungsentwurf eingegangen, welcher sich mit der Entwicklung und der Simulation einiger Grundzellen auf Transistorebene befasst. Darauf aufbauend erfolgt dann der Entwurf der eigentlichen Funktionsblöcke. Auf Basis der Simulationen kann dann die endgültige Architektur mit der erforderlichen Parallelisierung festgelegt werden.

3 Logikentwurf mittels komplementärer Metalloxid-Halbleiter-Technologie

Der Viterbi-Entzerrer wird größten Teils aus digitalen Logikbausteinen in einer komplementären Metalloxid-Halbleiter-(CMOS)-Technologie entworfen. Bipolartechnologien scheiden aus, da sich damit aufgrund der deutlich höheren Verlustleistung keine hochintegrierten Schaltungen mit mehreren hunderttausend Transistoren auf wenigen Quadratmillimetern Chipfläche realisieren lassen.

Auf der untersten Ebene stehen die Transistoren, aus denen dann Logikgatter wie z.B. UND- und ODER-Funktionen aufgebaut werden können. Diese Logikgatter werden in der nächsthöheren Hierarchieebene zu komplexeren Logikbausteinen wie Volladdierer, Speicherzellen und Flipflops zusammgebaut. Aus diesen Logikbausteinen werden dann noch komplexere Blöcke wie ganze Rechenwerke und Speicherfelder zusammengesetzt. Darauf aufbauend kann dann ein komplettes System wie im vorliegenden Fall der Viterbi-Entzerrer aufgebaut werden.

Für digitale Schaltungen gibt es in CMOS Design Kits verschiedener Halbleiterhersteller meist Standardzellen für Logikgatter, Addierer oder Speicherzellen. Die folgende Simulation zeigt die Verzögerungszeit eines 8 bit Paralleladdierers mit sukzessiver Übertragskorrektur (engl. Ripple-Carry-Adder, RCA), der auf Volladdiererzellen aus einer Standardzellenbibliothek des Design Kit Herstellers beruht.

Erläuterung zur Simulation:

$A = 0$ bzw. $B = 0$: Alle Eingangsbits $A_7...A_0$ bzw. $B_7...B_0$ sind konstant 0

$A = 1$ bzw. $B = 1$: Alle Eingangsbits $A_7...A_0$ bzw. $B_7...B_0$ sind konstant 1

$C_{in}: 0 \rightarrow 1$ Der niederwertigste Übertragseingang schaltet von 0 nach 1

$C_{in}: 1 \rightarrow 0$ Der niederwertigste Übertragseingang schaltet von 1 nach 0

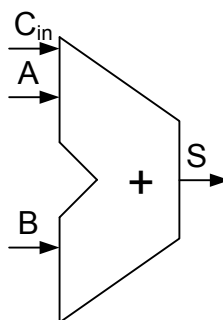


Abbildung 3.1: Addierer

Die Kombination der Eingangsbelegung stellt sicher, dass das Übertragsbit jeder einzelnen Volladdiererzelle an die nächste weitergeleitet wird und somit Schaltvorgänge mit möglichst

hohen Verzögerungszeiten erreicht werden. Tabelle 3.1 stellt die simulierten Verzögerungszeiten zu den Eingangskombinationen in Relation.

Tabelle 3.1: Simulation eines 8 bit RCAs mit Standardzellen

Schaltvorgang am Eingang	Verzögerungszeit
A = 0, B = 1, C _{in} : 0 → 1	392 ps
A = 0, B = 1, C _{in} : 1 → 0	494 ps
A = 1, B = 0, C _{in} : 0 → 1	429 ps
A = 1, B = 0, C _{in} : 1 → 0	488 ps

Diese Standardzellen sind in der Regel nicht für besonders schnelle Logikoperationen optimiert, so dass der Entwurf eigener Standardzellen verfolgt wird. Die starke Asymmetrie der Verzögerungszeiten ist ein deutlicher Hinweis darauf, dass Optimierungen möglich sind.

Daher wird die eingesetzte 90 nm CMOS-Technologie zunächst evaluiert [39]. Um gute Ergebnisse zu bekommen, werden CMOS-Logikgatter mit möglichst symmetrischen Schaltflanken $t_{LH} = t_{HL}$ entworfen, woraus dann symmetrische Verzögerungszeiten $t_{pLH} = t_{pHL}$ folgen. Um das zu erreichen, müssen die Transistoren so dimensioniert werden, dass der dynamische Aufladestrom der intrinsischen Kapazitäten gleich groß ist wie der Entladestrom.

Bei komplementärer MOS-Logik stehen die beiden Transistortypen NMOS und PMOS zur Verfügung. Die Bezeichnung NMOS-Transistor steht dafür, dass für den Ladungstransport die negativen Ladungsträger verantwortlich sind. Beim PMOS-Transistor sind die positiven Ladungsträger für den Ladungstransport verantwortlich. Unter den negativen Ladungsträgern versteht man die Elektronen, während die positiven Ladungsträger von den Löchern (Elektronenfehlstellen) gebildet werden. Da sich die Elektronen- und Löcherbeweglichkeiten voneinander unterscheiden [40], müssen die Transistortypen verschieden dimensioniert werden.

3.1 Statische CMOS-Logik

Am Beispiel eines CMOS-Inverters soll erklärt werden, wie man CMOS-Logikgatter auf symmetrische Schaltflanken dimensioniert. Der CMOS-Inverter ist das trivialste Logikgatter, das aus einem PMOS- und einem NMOS-Transistor besteht (Abbildung 3.2). Da in dieser Arbeit eine digitale Schaltung entwickelt werden soll, werden Effekte, die durch Kanallängenmodulation, Abschnürbereich und Geschwindigkeitssättigung hervorgerufen werden, in den folgenden Betrachtungen vernachlässigt.

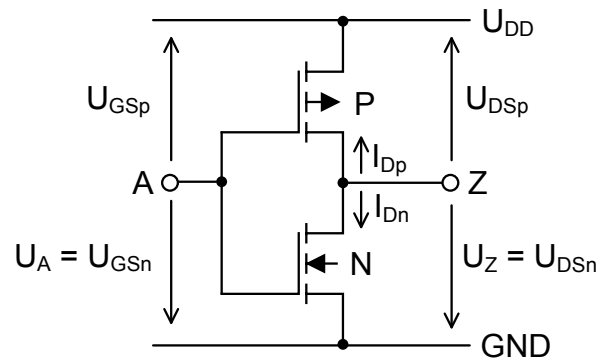


Abbildung 3.2: CMOS-Inverter

Für symmetrische Schaltflanken muss für die Drain-Ströme folgende Bedingung gelten:

$$I_{Dn} = -I_{Dp} \quad (3.1)$$

Dadurch wird sichergestellt, dass sowohl Auflade- als auch Entladestrom der Lastkapazität gleich groß sind. Im Folgenden wird der Entladevorgang, der am Ausgang Z zusammengefassten äquivalenten Lastkapazität, näher betrachtet. Die Entladung erfolgt über den NMOS-Transistor, da dieser für positive Eingangsspannungen $U_{GSn} > 0$ leitet. Das Aufladen des Knotens Z erfolgt über den PMOS-Transistor. Aus Symmetriegründen gelten für den Aufladevorgang dieselben Gleichungen mit den entsprechenden Indizes.

Der MOSFET weist die drei Arbeitsbereiche Sperrbereich, Widerstandsbereich und Sättigungsbereich auf, die im einfachsten Fall durch folgende Stromgleichungen [6] beschrieben werden können.

Sperrbereich

$$I_{Dn} = 0 \quad \text{für} \quad U_{GSn} - U_{Tn} \leq 0 \quad (3.2)$$

Widerstandsbereich

$$I_{Dn} = \beta_n \left[(U_{GSn} - U_{Tn}) U_{DSn} - \frac{U_{DSn}^2}{2} \right] \quad \text{für} \quad 0 < U_{DSn} < U_{GSn} - U_{Tn} \quad (3.3)$$

Sättigungsbereich

$$I_{Dn} = \frac{1}{2} \beta_n (U_{GSn} - U_{Tn})^2 \quad \text{für} \quad U_{DSn} \geq U_{GSn} - U_{Tn} > 0 \quad (3.4)$$

Der Stromverstärkungsfaktor errechnet sich mit

$$\beta_{n,p} = \mu_{n,p} C'_{ox} \frac{W}{L_{eff}} \quad (3.5)$$

Zur Bestimmung der Entladezeit wird das Ersatzschaltbild aus Abbildung 3.3 zugrunde gelegt. Vereinfachend wird angenommen, dass nur der NMOS-Transistor leitet.

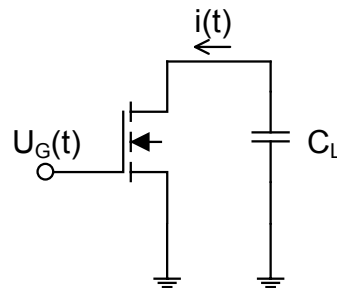


Abbildung 3.3: Ersatzschaltbild zur Herleitung der Entladezeit

Folgende Differenzialgleichung beschreibt den Zusammenhang zwischen dem Entladestrom einer Kapazität in Abhängigkeit ihrer zeitlich veränderlichen anliegenden Spannung.

$$i(t) = -C_L \frac{du_C}{dt} \quad (3.6)$$

Der Entladevorgang beginnt zunächst im Sättigungsbereich und wird beschrieben durch

$$I_{Dn} = \frac{1}{2} \beta_n (U_{GSn} - U_{Tn})^2 = -C_L \frac{du_C}{dt} \quad \text{für } U_{DD} > u_C > U_{DSsat} \quad (3.7)$$

Nach Übergang in den Widerstandsbereich gilt

$$I_{Dn} = \beta_n \left[(U_{DD} - U_{Tn}) u_C(t) - \frac{u_C^2}{2} \right] = -C_L \frac{du_C}{dt} \quad \text{für } U_{DSsat} > u_C > 0 \quad (3.8)$$

Unter den Randbedingungen

$$U_{DSsat} = U_{GS} - U_{Tn} \quad U_{GS} = U_{DD} \quad U_{Tn} \approx 0,2U_{DD} \quad u_{C,end} = 0,1U_{DD}$$

kommt man nach Integration und weiteren Umformungen der Gleichungen (3.7) und (3.8) zu dem Ergebnis

$$t_{HL} \approx 4 \frac{C_L}{\beta_n U_{DD}} \quad (3.9)$$

Aus Symmetriegründen reicht es aus, für den Aufladevorgang den Index n durch p zu ersetzen, wodurch sich Formel (3.10) ergibt.

$$t_{LH} \approx 4 \frac{C_L}{\beta_p U_{DD}} \quad (3.10)$$

Die Verzögerungszeiten eines Inverters sind zwischen dem Eingangs- und dem zugehörigen Ausgangssignal bei der halben Versorgungsspannung definiert. Daher ergibt sich als Definition für die Verzögerungszeit

$$t_{pHL} \approx 2 \frac{C_L}{\beta_n U_{DD}} \quad \text{bzw.} \quad t_{pLH} \approx 2 \frac{C_L}{\beta_p U_{DD}} \quad (3.11)$$

Aus (3.9) und (3.10) bzw. (3.11) wird ersichtlich, dass für symmetrische Auf- und Entladeflanken bzw. für symmetrische Verzögerungszeiten

$$\beta_n = \beta_p \quad (3.12)$$

gelten muss. Wie aus Formel (3.5) ersichtlich ist, hängt die Größe β einerseits von den Technologieparametern Ladungsträgerbeweglichkeit μ und dem Gatekapazitätsbelag C'_{OX} ab und andererseits von den geometrischen Transistorabmessungen wie Kanallänge L und Kanalweite W . Die Technologieparameter sind vom Halbleiterhersteller vorgegeben und können nicht verändert werden. Bei den Transistorabmessungen würde eine Erhöhung der Kanallänge die Verzögerungszeit in quadratischem Maße ansteigen lassen [41]. Folglich steht nur die Transistorweite W als Designparameter, um symmetrische Schaltflanken zu erreichen, zur Verfügung.

Da die Elektronenbeweglichkeit μ_n ungefähr doppelt so hoch ist wie die Löcherbeweglichkeit μ_p [40], muss der p-Kanal-MOSFET eines Inverters doppelt so weit dimensioniert werden, als der n-Kanal-MOSFET.

Simulationen zeigen [39], dass in der verwendeten 90 nm CMOS-Technologie das Verhältnis für symmetrische Schaltflanken bei $W_p:W_n$ bei 2,3:1 liegt. Daraus ergibt sich die folgende Standarddimensionierung der Weiten eines Inverters: p-Kanal-MOSFET mit $W_p = 2,3 \mu\text{m}$ und n-Kanal-MOSFET mit $W_n = 1,0 \mu\text{m}$. Die Verzögerungszeiten liegen bei dieser Dimensionierung bei $t_{pLH} = 9,0 \text{ ps}$ und $t_{pHL} = 8,9 \text{ ps}$.

Beim Übergang zu Logikgattern müssen mehrere p- oder n-Kanal-MOSFETs parallel oder in Serie geschaltet werden. Zur Realisierung einer UND-Verknüpfung wird im n-Kanalzweig eine Serienschaltung benötigt. Der p-Kanalzweig hingegen muss als Parallelschaltung ausgeführt werden, da beide Zweige zueinander komplementär sein müssen. Für eine ODER-Verknüpfung gilt genau das Gegenteil. Hierbei muss der n-Kanalzweig als Parallelschaltung und der p-Kanalzweig als Serienschaltung ausgeführt werden. Komplexere Logikfunktionen lassen sich

immer auf UND- bzw. ODER-Verknüpfungen zurückführen, welche sich durch entsprechende Kombination von Serien- und Parallelschaltung der einzelnen Transistoren realisieren lassen. Für eine allgemeine Herleitung wird auf Lehrveranstaltungen [41] bzw. Lehrbücher [6] verwiesen. Am Beispiel eines verneinten UND-Gatters – dem NAND-Gatter – soll kurz auf die Dimensionierung der Einzeltransistoren eingegangen werden.

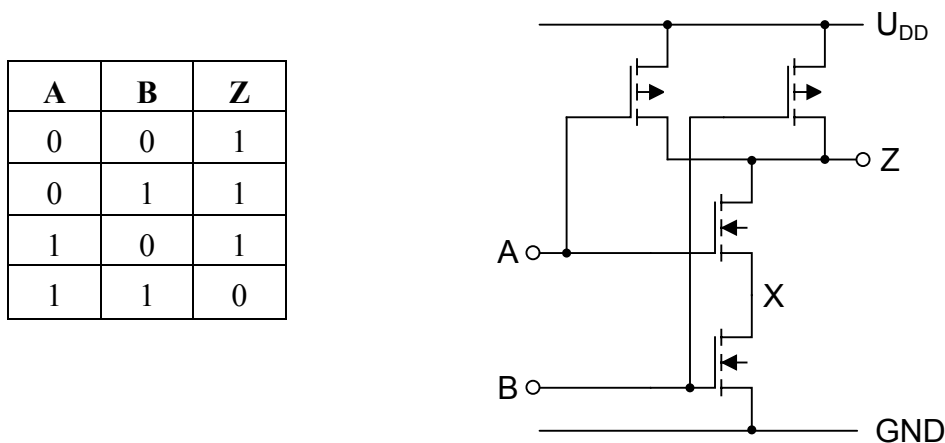


Abbildung 3.4: Wahrheitstabelle und Schaltplan eines NAND-Gatters

Wie aus Abbildung 3.4 ersichtlich wird, besteht eine logische NAND-Verknüpfung in statischer CMOS-Logik aus zwei in Serie geschalteten n-Kanal-MOSFETs und zwei parallel geschalteten p-Kanal-MOSFETs. Soll am Ausgang Z ein statischer Low-Pegel erzeugt werden, hat dies zur Folge, dass die Ladung am Knoten Z über die beiden in Serie geschalteten n-Kanaltransistoren abfließen muss. Zwei in Serie geschaltete Transistoren leiten nur halb so gut wie ein einzelner Transistor mit der gleichen Kanalweite [41]. Um symmetrische Schaltflanken zu bekommen, muss also die Weite der n-Kanaltransistoren verdoppelt werden. Für in Serie geschaltete p-Kanaltransistoren gilt dasselbe, nur muss zusätzlich noch die schlechtere Löcherbeweglichkeit gegenüber der Elektronenbeweglichkeit berücksichtigt werden, so dass die Weite der p-Kanaltransistoren noch größer ist als die der n-Kanaltransistoren. Allgemein kann die Weite der Transistoren eines beliebigen Gatters für symmetrische Schaltflanken wie in (3.13) ermittelt werden.

$$W_n = n_n W_{\min} \quad W_p = \beta_R n_p W_{\min} \quad \text{mit } \beta_R = \frac{\mu_n}{\mu_p} \quad (3.13)$$

Tabelle 3.2 zeigt die Verzögerungszeiten eines NAND-Gatters mit zwei Eingängen und verschiedenen Transistorweiten auf Schaltplanebene. Die Simulationen erfolgen unter folgenden Randbedingungen: Den Transistoren liegen typische Prozessparameter zu Grunde. Die Kanal-längen sind minimal mit 100 nm gewählt. Bei einem NAND-Gatter sind vier Schaltvorgänge möglich. Damit der Ausgang schaltet, muss immer einer der beiden Eingänge A oder B einen

logischen High-Pegel aufweisen. Der Eingang, an dem der High-Pegel anliegt, ist in der Tabelle mit einer 1 gekennzeichnet. Der schaltende Eingang ist durch einen Pfeil gekennzeichnet, der die Richtung der Schaltflanke wiedergibt. Der Ausgang trägt die Bezeichnung Z und der Pfeil gibt ebenfalls die Richtung der Flanke an. Die ungünstigsten Verzögerungszeiten sind hervorgehoben.

Tabelle 3.2: Schaltplansimulation eines NAND-Gatters mit zwei Eingängen

Dimensionierung		$W_p = 2,3 \mu\text{m}$ $W_n = 2,0 \mu\text{m}$	$W_p = 2,3 \mu\text{m}$ $W_n = 1,5 \mu\text{m}$	$W_p = 2,3 \mu\text{m}$ $W_n = 1,4 \mu\text{m}$
Verzögerungszeit	A:↑, B=1, Z:↓	9,7 ps	11,3 ps	11,8 ps
	A:↓, B=1, Z:↑	12,8 ps	12,1 ps	11,9 ps
	A=1, B:↑, Z:↓	11,4 ps	13,0 ps	13,4 ps
	A=1, B:↓, Z:↑	16,0 ps	14,3 ps	14,0 ps

Die Simulation zeigt, dass die nach der Formel (3.13) errechnete Weite für $W_n = 2,0 \mu\text{m}$ für den n-Kanalzweig nicht optimal ist, was an der deutlichen Asymmetrie von $\Delta t_p = 6,3 \text{ ps}$ der Verzögerungszeiten zu erkennen ist. Die beiden Schaltvorgänge, die eine fallende Flanke am Ausgang Z erzeugen, sind deutlich schneller als die, die eine ansteigende Flanke hervorrufen. Daraus kann geschlossen werden, dass der n-Kanalzweig zu weit dimensioniert ist. Eine Reduzierung der Weiten auf $W_n = 1,5 \mu\text{m}$ zeigt eine deutliche Verbesserung hinsichtlich der ungünstigsten Verzögerungszeit als auch der Symmetrie der Verzögerungszeiten von $\Delta t_p = 3,0 \text{ ps}$ bzw. der Schaltflanken. Die Verzögerungszeit für das Entladen des Ausgangsknotens steigt zwar an, was mit der geringeren Treiberfähigkeit des n-Kanalzweiges begründet werden kann. Andererseits sinkt aber die Zeit für das Aufladen stärker ab, da durch den kleineren n-Kanalzweig die am Ausgang durch den p-Kanalzweig aufzuladende Lastkapazität verringert wird. Bei einer n-Kanalweite von $W_n = 1,4 \mu\text{m}$ ist die Verzögerungszeit zwar noch kürzer, jedoch lassen sich diese Transistoren bei drei Gate-Fingern im Layout mit nur noch einem Drain-Source-Kontakt realisieren, wodurch die Zuverlässigkeit bei der Prozessierung deutlich verringert wird. Deshalb wird diese Dimensionierung nicht gewählt. Bei der Erstellung des Layouts können die Gatter einerseits aus Einzeltransistoren zusammengesetzt werden, andererseits ist es aber auch möglich, dass sich nebeneinander liegende Drain- und Source-Kontakte zweier Transistoren einen Kontakt teilen. Dadurch wird Verdrahtungskapazität und Fläche eingespart, woraus höhere Schaltgeschwindigkeiten und niedrigere Kosten resultieren. Abbildung 3.5 zeigt einen schematischen Querschnitt der Verdrahtung von zwei in Serie geschalteten Transistoren. Das linke Bild stellt die Realisierung aus Einzeltransistoren dar. Man erkennt, dass an jedem Zwischenknoten eine Sperrschichtkapazität (engl. Junction) zwischen Source-Anschluss (C_{sb}) bzw. Drain-Anschluss (C_{db}) und Substrat (engl. Bulk, B) wirkt. Die Verbindungsleitung wirkt ebenfalls als Kapazität (C_v) gegenüber dem Substrat. Außerdem weist die Verbindungsleitung einen Widerstand auf. Im rechten Bild ist der Netzknoten zwischen dem

Source-Anschluss des oberen n-Kanaltransistors und dem Drain-Anschluss des unteren n-Kanal-MOSFETs zusammengefasst, wodurch sich die Kapazität gemäß Parameterextraktion an diesem Knoten um beinahe 30 % verringert [41], [42]. Abbildung 3.6 zeigt das Layout des NAND-Gatters mit zwei Eingängen.

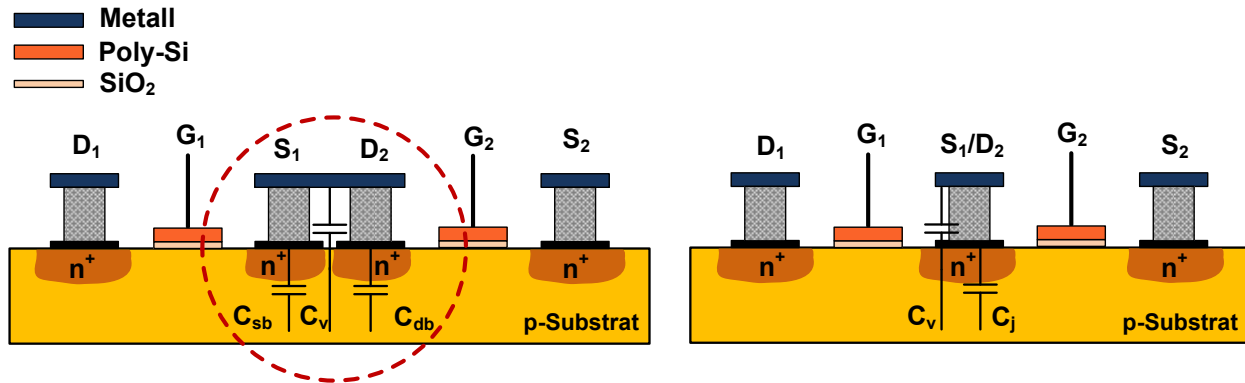


Abbildung 3.5: Standard-Layout-Querschnitt (links) und optimierter Layout-Querschnitt bei Serien- oder Parallelschaltung mit Einsparung von Kapazitäten am gemeinsamen Knoten (rechts)

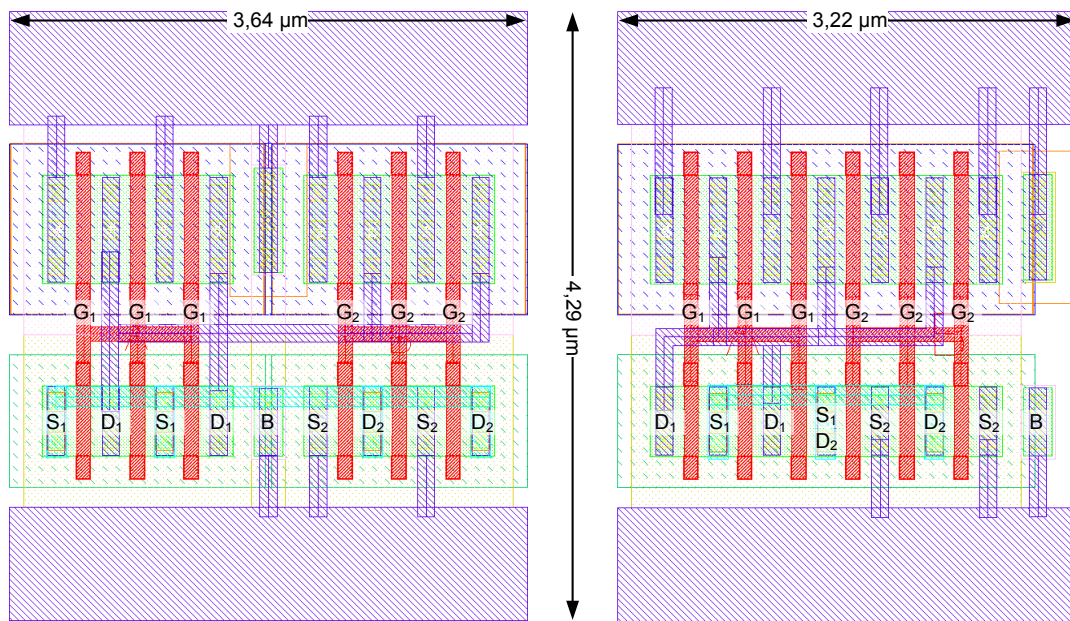


Abbildung 3.6: Standard-Layout (links) und optimiertes Layout bei Serien- oder Parallelschaltung mit Einsparung von Kapazitäten am gemeinsamen Knoten (rechts)

Tabelle 3.3 stellt die Verzögerungszeiten der beiden NAND-Gatters der Layoutvarianten aus Abbildung 3.6 gegenüber. Die Randbedingungen der Simulationen sind genauso gewählt, wie bei der vorhergehenden Schaltplansimulation. Den Verbindungsleitungen liegen ebenfalls typische Prozessparameter zu Grunde. Das Übersprechen ist vernachlässigt. Für die Transistorweiten gilt $W_p = 2,3 \mu\text{m}$ und $W_n = 1,5 \mu\text{m}$. Um ein optimiertes Layout erstellen zu können,

weisen die Einzeltransistoren drei Gate-Finger auf. Der Zwischenknoten X im Schaltplan (Abbildung 3.4) bzw. S₁/D₂ im Layout (Abbildung 3.6) gibt die Kapazität des gemeinsamen Drain-Source-Kontaktes der beiden in Serie geschalteten NMOS-Transistoren an.

Tabelle 3.3: Layoutsimulation eines NAND-Gatters mit zwei Eingängen

		Standard- Layout	optimiertes Layout
Verzögerungs- zeit	A:↑, B=1, Z:↓	13,2 ps	12,3 ps
	A:↓, B=1, Z:↑	14,0 ps	13,7 ps
	A=1, B:↑, Z:↓	15,2 ps	14,3 ps
	A=1, B:↓, Z:↑	16,9 ps	16,5 ps
parasitäre Kapazitäten an den Knoten	A	1,274 fF	1,218 fF
	B	1,278 fF	1,240 fF
	Z	1,441 fF	1,230 fF
	X	892 aF	696 aF
Fläche		15,62 μm ²	13,81 μm ²

Die Simulation zeigt, dass das optimierte Layout durchweg bessere Eigenschaften hinsichtlich der Verzögerungszeit und des Flächenverbrauchs aufweist. Die geringere Verzögerungszeit des optimierten Layouts resultiert aus den insgesamt kleineren parasitären Kapazitäten.

Die Dimensionierung weiterer Gatter wie NOR und XOR erfolgt nach eben vorgestellter Vorgehensweise. Für Details wird auf [39], [42], [43] verwiesen.

3.2 Transfertransistor- und Transfergatterlogik

Eine weitere Möglichkeit, um Signale mit Hilfe von MOS-Logik zu schalten, ist die Transfertransistorlogik. Der Vorteil dieser Logik besteht darin, dass man deutlich weniger Transistoren benötigt als in statisch komplementär aufgebauter Logik. Bei der Transfertransistorlogik werden, wie in Abbildung 3.7 dargestellt, der Eingang A mit dem Ausgang Z direkt über einen sogenannten Transfertransistor, dessen Gate über das Steuersignal S angesteuert wird, verknüpft.

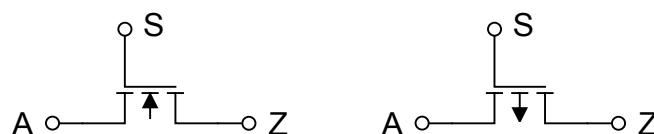


Abbildung 3.7: n- und p-Kanaltransfertransistor

Bei der statischen CMOS-Logik wird der Ausgangspegel immer durch den leitenden p-Kanal- oder n-Kanal-MOSFET auf die positive oder negative Versorgungsspannung regeneriert. Für die Spannungspegel an den Transfertransistoren gilt nach Tabelle 3.4:

Tabelle 3.4: Spannungspegel an Transfertransistoren

Z an NMOS	$U_S = 0 \text{ V}$	$U_S = U_{DD}$	Z an PMOS	$U_S = 0 \text{ V}$	$U_S = U_{DD}$
$A = U_{DD}$	X	$U_{DD} - U_{Tn}$	$A = U_{DD}$	U_{DD}	X
$A = 0 \text{ V}$	X	0 V	$A = 0 \text{ V}$	$-U_{Tp}$	X

Liegt am Gate eines n-Kanaltransfertransistors die Steuerspannung $U_S = 0 \text{ V}$ an, sperrt der Transistor. Somit kann für den Ausgang Z kein Spannungspegel vorhergesagt werden, ohne zu wissen, welcher Pegel zuvor an A angelegen hat. Dies ist in Tabelle 3.4 mit einem X in der betreffenden Spalte gekennzeichnet. Liegt am Gate ein logischer High-Pegel an, entsteht für $A = 0 \text{ V}$ ein leitender Kanal, so dass der Ausgang Z ebenfalls 0 V annimmt. Liegt der Eingang hingegen auf $A = U_{DD}$ geht der Transistor in einen Bereich zwischen Sperren und Leiten über, da die Bedingung $U_S - U_Z = U_{GS} = U_{Tn}$ erfüllt ist. Daher wird der High-Pegel am Ausgang eines n-Kanaltransfertransistors degeneriert.

Gleiches gilt für den p-Kanaltransfertransistor mit dem Unterschied, dass für einen sperrenden Transistor $U_S = U_{DD}$ der Ausgangspegel Z nicht definiert ist und im leitenden Fall der statische Low-Pegel für $A = 0 \text{ V}$ um $|U_{Tp}|$ erhöht wird.

Ein weiterer Nachteil von Transfertransistoren besteht im dynamischen Verhalten. Die Flanke, die den degenerierten Logikpegel erzeugt, ist deutlich langsamer, als die Flanke, die den nicht-degenerierten Pegel erzeugt. Für die Erläuterung hierfür wird auf [41] verwiesen.

Eine Verbesserung des Verhaltens kann durch eine Parallelschaltung aus n- und p-Kanaltransfertransistoren erzielt werden. Hierbei spricht man von einem Transfergatter (Abbildung 3.8). Die statischen High- und Low-Pegel werden in beiden Fällen erreicht. Wird die Weite des p-Kanaltransistors um den Faktor β_R größer als die des n-Kanaltransistors dimensioniert, erhält man auch symmetrische Schaltflanken. Diese Vorteile werden jedoch dadurch erkauft, dass das Transfergatter über zwei Steuerleitungen S und \bar{S} angesteuert werden muss und dass die Lastkapazität steigt. Die Erhöhung der Lastkapazität führt allgemein zu langsameren Schaltvorgängen, allerdings orientiert sich die maximale Taktfrequenz an der am langsamsten ansteigenden bzw. fallenden Flanke im System. Dennoch sind bei einem Transfergatter die Flanken steiler und damit schneller als die flachere und damit langsamere Flanke eines Transfertransistors.

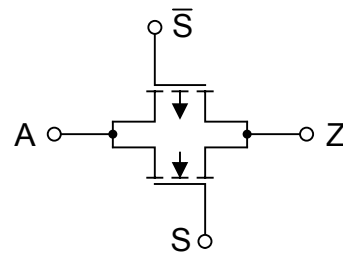


Abbildung 3.8: Transfergatter

Abbildung 3.9 zeigt den Spannungsverlauf am Knoten Z in Abhängigkeit des Eingangssignals A und des Steuersignals S.

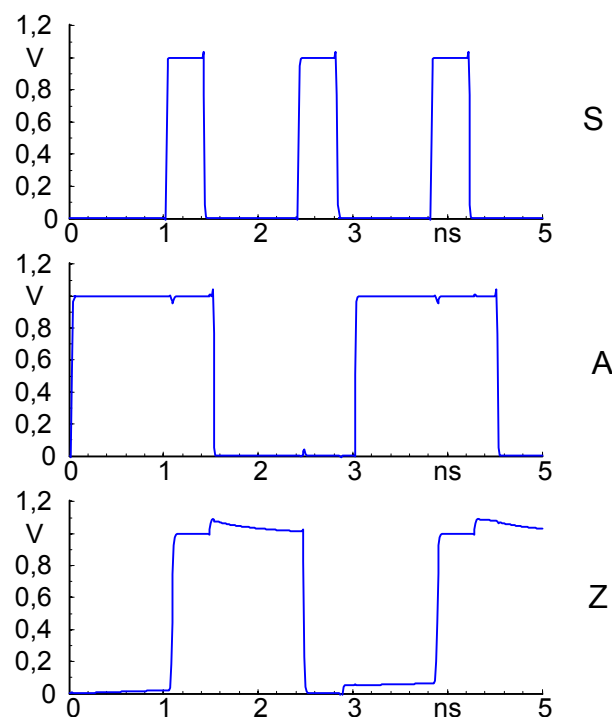


Abbildung 3.9: Spannungsverläufe am Transfergatter

Es fällt auf, dass der Ausgangsknoten Z immer dann eine Spannungserhöhung erfährt, wenn S eine fallende Flanke aufweist. Eine fallende Flanke von S bedeutet gleichzeitig eine steigende Flanke von \bar{S} . Da der p-Kanaltransistor aufgrund der Dimensionierung auf symmetrische Schaltflanken eine doppelt so große Weite wie der n-Kanaltransistor aufweist, gilt für die Koppelkapazitäten entsprechend $C_{GDp} = 2 C_{GDn}$ und $C_{GSp} = 2 C_{GSn}$. Schaltet nun der n-Kanaltransistor aus, müsste die Spannung am Knoten Z leicht abfallen, da aber gleichzeitig der p-Kanaltransistor ebenfalls ausschaltet, was einem Übergang von $\bar{S}:L \rightarrow H$ entspricht und dessen Kapazität doppelt so groß ist, überwiegt der Effekt der Spannungserhöhung am Knoten Z, so dass der Spannungsverlauf aus Abbildung 3.9 entsteht. Die Spannungssprünge am Knoten Z beruhen auf dem Energieerhaltungssatz. Die energietragende Größe einer Kapazität C (hiermit

ist die Koppelkapazität C_{GD} gemeint) ist die Spannung. Die Spannung kann sich in unendlich kurzer Zeit nicht ändern. Sonst müsste zum Zeitpunkt des Sprunges ein unendlicher Strom die Kapazität umladen. Allgemein gilt, dass der Strom durch den Kondensator $i_c(t) = \frac{du_c(t)}{dt}$ zu jedem Zeitpunkt einen stetigen Verlauf haben muss. Springt nun die Spannung $\bar{S}:L \rightarrow H$ folgt daraus, dass auch am Knoten Z die Spannung um den gleichen Betrag in die gleiche Richtung springen müsste. Da an S aber der umgekehrte Schaltvorgang stattfindet und über Verdrahtungswiderstände Spannung abfällt, wird dieser Effekt teilweise kompensiert, so dass die am Knoten Z simulierte Spannung nicht um den Betrag des Spannungssprungs an \bar{S} erhöht ist.

Eine weit verbreitete Anwendung von Transferrgatterschaltungen sind Multiplexer und Exklusiv-Oder-Verknüpfungen, die damit mit deutlich geringerem Transistoraufwand gegenüber einer Umsetzung in statischer komplementärer MOS-Logik aufgebaut werden können. In Abbildung 3.10 ist ein Multiplexer in Transferrgatterlogik dargestellt. Charakteristisch ist der Inverter, der die für die Transfertransistoren notwendigen inversen Steuersignale generiert. Exklusiv-Oder-Verknüpfungen finden vor allem bei der Berechnung des Summensignals von Volladdierern Anwendung.

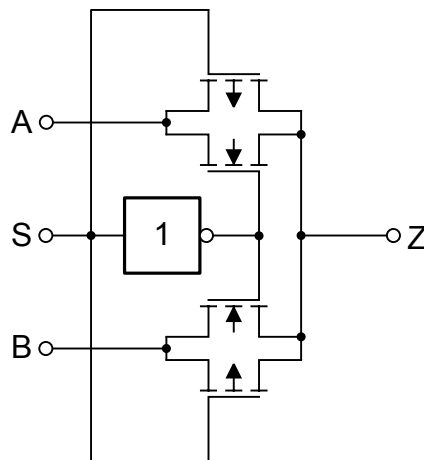


Abbildung 3.10: Multiplexer

Transferrgatter weisen keine hohe Treiberfähigkeit auf, da sich der leitende Kanal wie ein Widerstand verhält, was dazu führt, dass Schaltvorgänge eine höhere Verzögerungszeit aufweisen. Um diesen Nachteil auszugleichen, empfiehlt es sich, sowohl an den Eingängen A und B als auch am Ausgang einen Regenerationsinverter anzufügen. Gleichzeitig wird das Problem der Spannungspegelerhöhung am Knoten Z, wie in Abbildung 3.9 dargestellt, bei fallenden Flanken des Steuersignals S behoben. Eine solche Schaltung ist in Abbildung 3.11 gezeigt. Multiplexer nach Abbildung 3.11 weisen auf Simulationsbasis eine Verzögerungszeit von $t_{MUX} = 35$ ps auf.

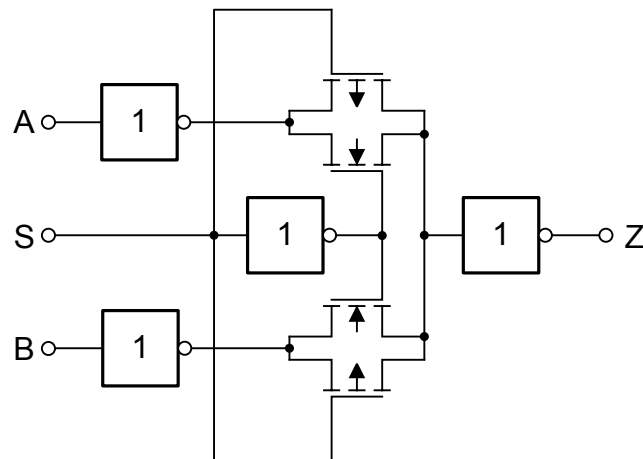


Abbildung 3.11: Multiplexer mit Pegelregeneration

3.3 Speicher- und Kippschaltungen

Eine Kippschaltung weist zwei Zustände auf. Diese können stabil oder labil sein. Daher unterteilt man Speicher- und Kippschaltungen in drei Gruppen. Die Einteilung erfolgt anhand der stabilen und labilen Zustände. Es gibt:

- Monostabile Kippschaltungen
- Astabile Kippschaltungen
- Bistabile Kippschaltungen

Eine monostabile Kippschaltung hat einen stabilen und einen labilen Zustand. Ausgehend von einem stabilen Zustand erhält das Monoflop eine Anregung von außen. Daraufhin wechselt es in den labilen Zustand. Nach einer bestimmten Zeit springt es wieder in seinen ursprünglichen stabilen Zustand zurück. Die Zeit hängt von einer internen RC-Zeitkonstante des Monoflops ab. Die RC-Elemente können aus diskreten Bauteilen aufgebaut sein oder aus parasitären Elementen bestehen. Dieser Schaltungstyp bietet sich z.B. für Zeitverzögerungen in analogen, ungetakteten Schaltungen an.

Astabile Kippschaltungen weisen zwei labile Zustände auf. Das heißt, wird die Schaltung eingeschaltet, wechselt sie ständig ihre beiden Zustände. Ein Beispiel hierfür wäre ein Taktgenerator, der aus einer ungeradzahligen Anzahl rückgekoppelter Inverter aufgebaut ist. Über RC-Elemente kann die Oszillationsfrequenz angepasst werden.

Bistabile Kippschaltungen haben zwei stabile Zustände. Regt man eine bistabile Kippschaltung von außen an, wechselt sie ihren Zustand und verbleibt in diesem solange, bis sie wieder einen Impuls bekommt. Hierunter fallen alle Arten an Speicherschaltungen. Dieser Schaltungstyp ist derjenige, der im Viterbi-Entzerrer hauptsächlich vorkommt. Daher werden in den folgenden Teilkapiteln verschiedene Speicherschaltungen ausführlicher behandelt.

3.3.1 Statische CMOS-Speicherzelle

Eine Speicherzelle für 1 bit kann man aus zwei rückgekoppelten Invertern, bestehend aus den Transistoren P_1 und N_1 bzw. P_2 und N_2 , aufbauen, was einer Serienschaltung zweier Inverter entspricht, bei der der Ausgang des zweiten Inverters auf den Eingang des ersten Inverters rückgekoppelt ist (Abbildung 3.12).

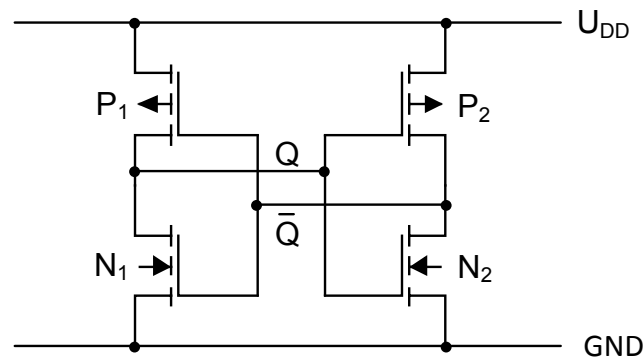


Abbildung 3.12: Speicherzelle für 1 bit

Dadurch stellt sich ein stabiler Logikpegel an den internen Knoten Q und \bar{Q} der Speicherzelle ein, der sich von sich aus nicht ändert. Legt man über eine äußere Beschaltung eine Störspannung an einem der beiden internen Knoten der rückgekoppelten Inverter an, können bei ausreichender Treiberfähigkeit und einer Amplitude, die den Schaltpunkt übersteigt, die internen Logikpegel kippen. Der neue Logikpegel ist auf Grund der Mitkopplung wieder stabil. Diese Art einer Schaltung nennt man bistabile Kippstufe oder Latch.

Um von außen Information in die Speicherzelle einschreiben zu können, erweitert man die Speicherzelle um die beiden Transfertransistoren N_3 und N_4 , wie in Abbildung 3.13 dargestellt.

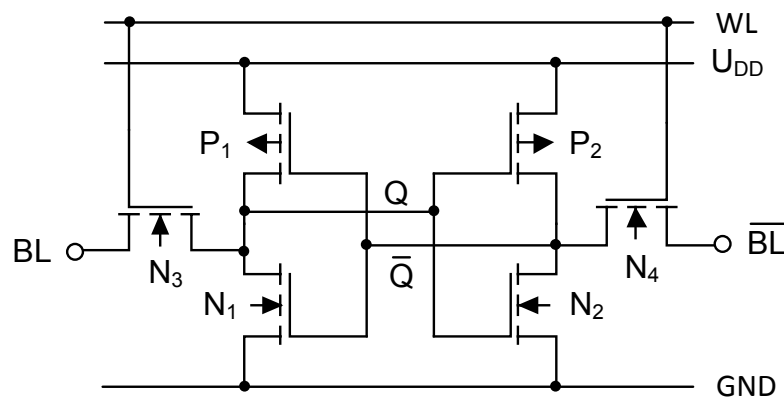


Abbildung 3.13: Sechs-Transistorspeicherzelle für 1 bit

Diese beiden Transistoren werden über die Wortleitung WL angesteuert. Liegt an der Wortleitung ein logischer High-Pegel an, wird bei ausreichender Transistorweite der Transistoren N_3 und N_4 der Logikpegel der Bitleitungen BL und \overline{BL} in die Speicherzelle eingeschrieben. Für das Auslesen der Speicherzelle ist keine besondere Schaltung notwendig. Der Speicherinhalt kann jederzeit direkt über die Anschlüsse Q bzw. \overline{Q} abgegriffen und weiterverarbeitet werden. Da in der zu entwickelten Zweigmetrikrecheneinheit nur eines der beiden Signale benötigt wird, reicht es aus, dass nur eines der beiden Signale Q oder \overline{Q} herausgeführt wird. Dadurch reduziert sich der Verdrahtungsaufwand des Speichers.

Bei einem Speicherfeld sind die Transistoren sehr dicht gepackt. Da Transistoren mit höherer Schwellenspannung einen geringeren Leckstrom aufweisen [44], [45] ist es von Vorteil, diese für die kreuzgekoppelten Inverter zu verwenden. Der Vorteil wird jedoch mit leichten Geschwindigkeitseinbußen erkauft, da die Stromtreiberfähigkeit eines MOSFETs von der Schwellenspannung abhängt; siehe Formeln (3.3) und (3.4). Da das Auslesen jedoch über einen Direktabgriff eines der beiden inneren Signale erfolgt, haben die inneren Transistoren keine Auswirkung auf die Lesegeschwindigkeit. Damit das Einspeichern neuer Werte mit hoher Geschwindigkeit abläuft, sind dagegen für die beiden Transfertransistoren N_3 und N_4 MOSFETs mit niedriger Schwellenspannung besser geeignet.

Die Weite der beiden Transfertransistoren kann wie folgt bestimmt werden. Für die rechnerische Betrachtung wird vereinfachend angenommen, dass sich beim Umschalten eines Knotens z.B. Q die Spannung am anderen Knoten \overline{Q} nicht ändert. Diese Vereinfachung ist erlaubt, da das Ergebnis eine größere Mindestweite der Transfertransistoren liefert, als tatsächlich notwendig ist. Aufgrund der Mitkopplung trägt der jeweils andere Inverter positiv zum Kippvorgang bei.

Annahmen zu Beginn des Schreibvorganges: Zunächst sei $Q = H$, $\overline{Q} = L$, $BL = L$, $\overline{BL} = H$. Sobald die Wortleitung WL einen High-Pegel einnimmt, beginnt der Einschreibvorgang. Um über die beiden Transfertransistoren die kreuzgekoppelten Inverter zum Kippen zu bringen, muss im Falle von Q das Potenzial von dem statischen High-Pegel V_{DD} auf den Schaltpunkt der Inverter $U_M = \frac{U_{DD}}{2}$ abfallen bzw. im Falle von \overline{Q} das Potenzial vom statischen Low-Pegel auf den Schaltpunkt U_M ansteigen.

Die Berechnung teilt sich in zwei Bereiche. Zunächst soll der Fall betrachtet werden, dass auf den Knoten \overline{Q} über den Transistor N_4 ein High-Pegel eingeschrieben werden soll. Der Pegel auf Q wird als konstant high angenommen. Daher sperrt der Transistor P_2 . Der Transistor N_1 wird ebenfalls als sperrend angenommen, da zu Beginn $\overline{Q} = L$ gilt. Durch den Anstieg der Spannung an \overline{Q} wird der Kippvorgang durch das Leitendwerden von N_1 zwar beschleunigt, aber für die Abschätzung des ungünstigsten Falls kann er vernachlässigt werden.

Es gilt:

$$U_{\text{GSN}2} = U_{\text{DD}}, U_{\text{DSN}2} = U_{\text{M}} = \frac{U_{\text{DD}}}{2} \quad (3.14)$$

Da $U_{\text{DSN}2} < U_{\text{GSN}2} - U_{\text{Th}}$ erfüllt ist, befindet sich N_2 im linearen Bereich

$$U_{\text{GSN}4} = U_{\text{DD}} - \frac{U_{\text{DD}}}{2}, U_{\text{DSN}4} = U_{\text{DD}} - \frac{U_{\text{DD}}}{2} \quad (3.15)$$

Da $U_{\text{GSN}4} - U_{\text{Th}} < U_{\text{DSN}4}$ erfüllt ist, befindet sich N_4 im Sättigungsbereich

Nach dem Kirchhoffschen Gesetz muss

$$I_{\text{DN}2} = I_{\text{DN}4} \quad (3.16)$$

gelten. Aus (3.14) und (3.15) folgt mit (3.3) und (3.4) der Ansatz

$$\beta_{\text{N}2} \left[(U_{\text{GSN}2} - U_{\text{Th}}) U_{\text{DSN}2} - \frac{1}{2} U_{\text{DSN}2}^2 \right] = \frac{\beta_{\text{N}4}}{2} (U_{\text{GSN}4} - U_{\text{Th}})^2 \quad (3.17)$$

Da beide am Kippvorgang beteiligten Transistoren n-Kanaltransistoren sind, gilt $\beta \sim W$. Daher kann man für β direkt die Kanalweite W einsetzen. Um Fläche zu sparen und um die kapazitive Belastung der Transfertransistoren möglichst gering zu halten, wird für die Weite des Latches statt $2,3 \mu\text{m} / 1,0 \mu\text{m}$ (PMOS-/NMOS-Weite) wie beim Standardinverter nur die halbe Weite $1,15 \mu\text{m} / 0,5 \mu\text{m}$ verwendet. Daher wird $W_{\text{N}2} = 0,5 \mu\text{m}$ gesetzt. Aus (3.17) folgt

$$W_{\text{N}4} \geq \frac{2W_{\text{N}2} \left[(U_{\text{GSN}2} - U_{\text{Th}2}) U_{\text{DSN}2} - \frac{1}{2} U_{\text{DSN}2}^2 \right]}{(U_{\text{GSN}4} - U_{\text{Th}4})^2} \quad (3.18)$$

Für die Berechnung von $W_{\text{N}4}$ liegen folgende Werte zugrunde: $W_{\text{N}2} = 0,5 \mu\text{m}$, $U_{\text{GSN}2} = 1 \text{ V}$, $U_{\text{DSN}2} = 0,5 \text{ V}$, $U_{\text{GSN}4} = 0,5 \text{ V}$. Die Schwellenspannung für n-Kanaltransistoren mit niedriger Schwellenspannung beträgt laut Simulationen $U_{\text{Th}4} = U_{\text{Th}2} = 0,170 \text{ V}$. Die Schwellenspannung für n-Kanaltransistoren mit hoher Schwellenspannung liegt bei $U_{\text{Th}4} = U_{\text{Th}2} = 0,315 \text{ V}$. Damit ergibt sich für den Transfertransistor N_4 eine Mindestweite von $W_{\text{N}4} = 2,0 \mu\text{m}$.

Der zweite Fall betrachtet den Schaltvorgang, dass über den Transfertransistor N_3 der statische Low-Pegel eingeschrieben werden soll. Dabei soll der Transistor N_3 so dimensioniert werden, damit das Potenzial des Knotens $Q = H$ auf den Schaltpunkt $U_{\text{M}} = 0,5 \text{ V}$ abfallen kann. Der Knoten \bar{Q} wird als konstant logisch low angenommen. Daher können die Transistoren N_2 und auch P_1 für die Ermittlung der Mindestweite vernachlässigt werden.

Es gilt:

$$U_{GSP1} = -U_{DD}, \quad U_{DSP1} = -U_M = -\frac{U_{DD}}{2} \quad (3.19)$$

Da $U_{DSP1} > U_{GSP1} - U_{Tp}$ erfüllt ist, befindet sich P_1 im linearen Bereich

$$U_{GSN3} = U_{DD}, \quad U_{DSN3} = \frac{U_{DD}}{2} \quad (3.20)$$

Da $U_{GSN3} - U_{Tn} > U_{DSN3}$ erfüllt ist, befindet sich N_3 im linearen Bereich

Nach dem Kirchhoffschen Gesetz muss

$$I_{DN3} = -I_{DP1} \quad (3.21)$$

gelten.

Aus (3.19) und (3.20) folgt mit (3.3) und (3.4) der Ansatz

$$\beta_{N3} \left[(U_{GSN3} - U_{Tn}) U_{DSN3} - \frac{1}{2} U_{DSN3}^2 \right] = \beta_{P1} \left[(U_{GSP1} - U_{Tp}) U_{DSP1} - \frac{1}{2} U_{DSP1}^2 \right] \quad (3.22)$$

Die beiden am Kippvorgang beteiligten Transistoren P_1 und N_3 haben eine unterschiedliche Ladungsträgerbeweglichkeit. Nach [39] beträgt das Verhältnis $\mu_n:\mu_p = 2,3:1$, was dazu führt, dass $\beta_n = 2,3 \beta_p$ gilt. Aus dieser Randbedingung ergibt sich mit (3.22):

$$W_{N3} \geq W_{P1} \frac{\left[(U_{GSP1} - U_{Tp1}) U_{DSP1} - \frac{1}{2} U_{DSP1}^2 \right]}{2,3 \left[(U_{GSN3} - U_{Tn3}) U_{DSN3} - \frac{1}{2} U_{DSN3}^2 \right]} \quad (3.23)$$

Für die Berechnung von W_{N3} liegen folgende Werte zugrunde: $W_{P1} = 1,15 \mu\text{m}$, $U_{GSP1} = -1 \text{ V}$, $U_{DSP1} = -0,5 \text{ V}$, $U_{GSN3} = 1 \text{ V}$ und $U_{DSN3} = 0,5 \text{ V}$. Die Schwellenspannung für n-Kanaltransistoren mit niedriger Schwellenspannung beträgt nach Simulation $U_{Tnvt} = U_{Tn3} = 0,170 \text{ V}$. Die Schwellenspannung für p-Kanaltransistoren mit hoher Schwellenspannung liegt bei $U_{Tphvt} = U_{Tp1} = -0,34 \text{ V}$. Damit ergibt sich für den Transfertransistor N_3 eine Mindestweite von $W_{N3} = 0,35 \mu\text{m}$.

Die Rechnung zeigt, dass es möglich ist, mit einem Transfertransistor der Weite $W_{N3} = 0,35 \mu\text{m}$ einen Kippvorgang auszulösen. Eine Dimensionierung des Transfertransistors mit der Weite

$W_{N4} = 2,0 \mu\text{m}$ ist nicht notwendig, da das Erreichen der Schaltschwelle eines Inverters ausreicht, um den anderen zum Kippen zu bringen. Betrachtet man nur den zweiten Fall im zuvor berechneten Beispiel, hat ein Absinken der Spannung am Knoten Q auf die Schaltschwelle $U_M = 0,5 \text{ V}$ zur Folge, dass die Transistoren des Inverters, bestehend aus P_2 und N_2 , in den Sättigungsbereich übergehen, wodurch sich am Knoten \bar{Q} ebenfalls die Schaltschwelle $U_M = 0,5 \text{ V}$ einstellt, was zur Mitkopplung führt. Durch die gegenphasige Ansteuerung der Speicherzelle mit BL und \bar{BL} treten also immer beide Schaltvorgänge gleichzeitig auf, wodurch die Mitkopplung unterstützt wird.

Die Dimensionierung des Transfertransistors N_3 mit $W_{N3} = 0,35 \mu\text{m}$ gibt also die Mindestweite für beide Transfertransistoren an, damit der Kippvorgang überhaupt erfolgen kann.

Anhand von Simulationen zeigt sich, dass die Speicherzelle schon bei einer Transfertransistorweite von $W_{N3} = 0,21 \mu\text{m}$ umgeschrieben werden kann, was über die Mitkopplung begründet wird. Da die Mitkopplung in der Rechnung unberücksichtigt bleibt, ist die berechnete Weite größer als die simulierte Weite. So gesehen ist die Rechnung eine gute Abschätzung, bei der die Schaltung auf jeden Fall noch funktioniert.

Dynamische Betrachtung der SRAM-Zelle

Dimensioniert man auf die ermittelte Mindestweite, dauert das Umladen relativ lange. Tabelle 3.5 zeigt die Einschreibdauer einer logischen Eins in eine SRAM-Zelle auf Schaltplanebene mit typischen Transistoren. Unter der Einschreibdauer versteht man die Verzögerungszeit zwischen dem Anlegen eines High-Pegels an die Wortleitung WL und dem Kippen des inneren Knotens Q. Die Einschreibdauer nimmt zunächst mit zunehmender Weite der beiden Wortleitungstransistoren ab. Bei einer Weite von $1,5 \mu\text{m}$ ist jedoch eine Sättigung erreicht. Bei noch größeren Weiten nimmt die Einschreibdauer wieder zu.

Tabelle 3.5: Einschreibdauer in eine SRAM-Zelle in Abhängigkeit der Wortleitungstransistorweite

Weite des Wortleitungstransistors	Einschreibdauer t_{SRAM}
$0,21 \mu\text{m}$	37,3 ps
$0,5 \mu\text{m}$	36,3 ps
$1,0 \mu\text{m}$	32,2 ps
$1,2 \mu\text{m}$	28,2 ps
$1,5 \mu\text{m}$	26,4 ps
$2,0 \mu\text{m}$	26,5 ps

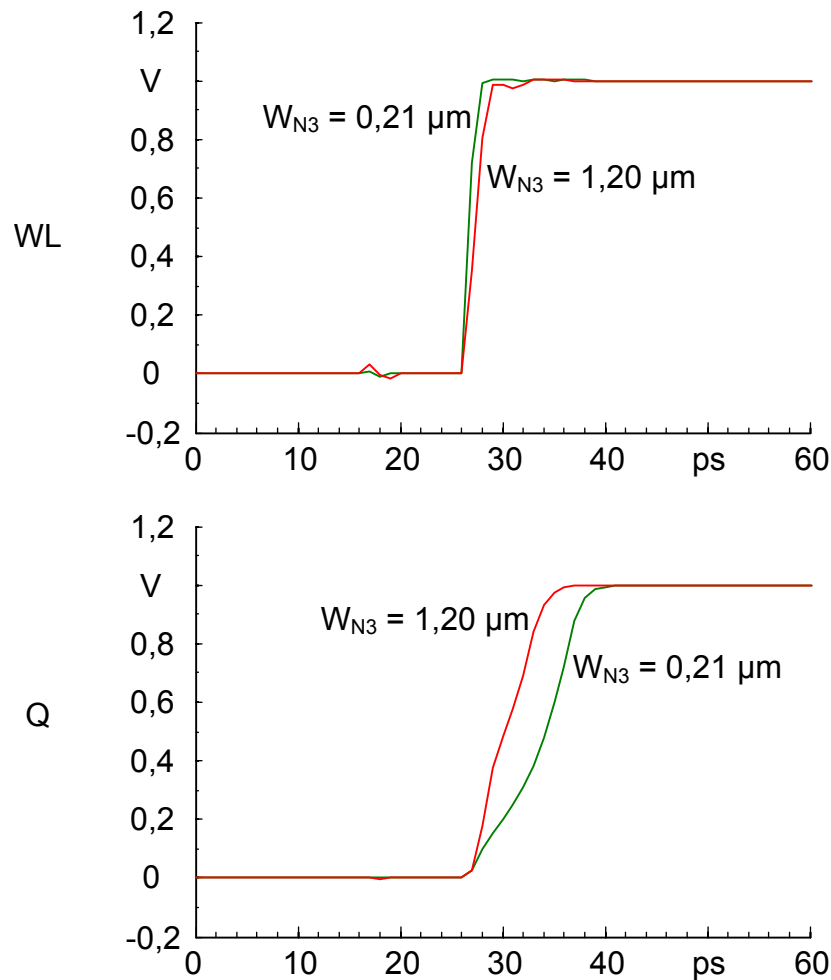


Abbildung 3.14: Einschreibvorgang einer SRAM-Zelle

In Abbildung 3.14 steigt die Flanke der Wortleitung des Transistors N_3 mit der Weite $W_{N3} = 0,21 \mu\text{m}$ aufgrund der kleineren Gate-Kapazität früher an, als mit der Weite von $W_{N3} = 1,20 \mu\text{m}$.

Da eine Verkürzung der Verzögerungszeit um 1,8 ps bei gleichzeitiger Erhöhung der Transistorweite um $0,3 \mu\text{m}$ nicht als notwendig angesehen wird, hat man sich zugunsten eines niedrigeren Stromverbrauches beim Wortleitungstransistor für eine Weite von $1,2 \mu\text{m}$ entschieden. Die Transistoren der kreuzgekoppelten Inverter weisen die folgenden Weiten auf: $W_{P1} = W_{P2} = 1,15 \mu\text{m}$ und $W_{N1} = W_{N2} = 0,5 \mu\text{m}$. Die Abmessung der Speicherzelle beträgt $3,2 \mu\text{m} \times 4,9 \mu\text{m}$ einschließlich Zuführung der Versorgungsspannung. In Abbildung 3.15 ist das Layout der Speicherzelle abgebildet.

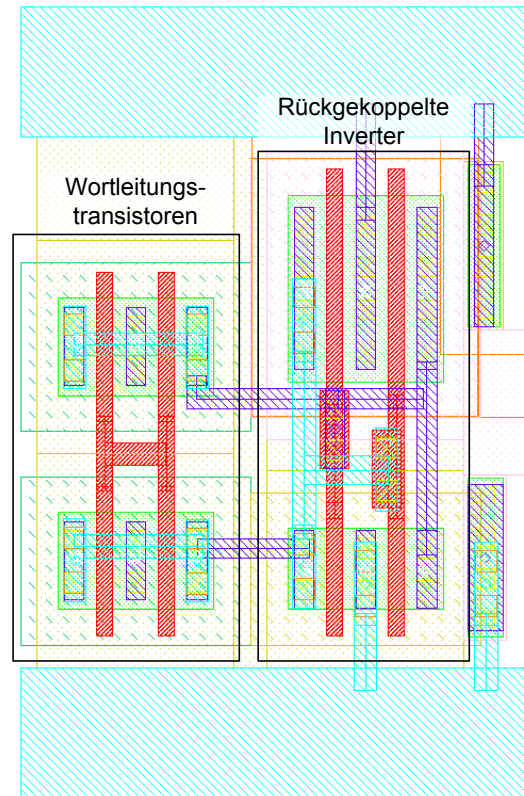


Abbildung 3.15: Layout einer SRAM-Zelle

3.3.2 Dynamische CMOS-Speicherzelle

Eine Alternative zur statischen CMOS-Speicherzelle stellt die dynamische CMOS-Speicherzelle, kurz DRAM genannt, dar.

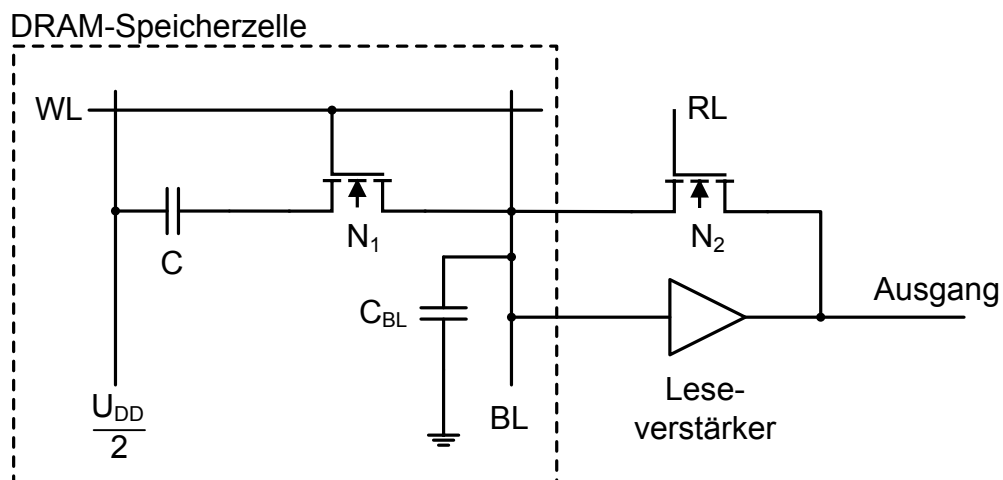


Abbildung 3.16: DRAM-Speicherzelle mit Refresh-Schaltung

In Abbildung 3.16 ist eine DRAM-Speicherzelle mit Refresh-Schaltung dargestellt. Die Information wird nicht, wie bei der SRAM-Zelle in einem Latch gespeichert, sondern in der Speicherkapazität C . Die eine Platte der Kapazität C wird immer auf dem konstanten Potenzial $U_{DD}/2$ gehalten, während die andere Platte über den Wortleitungstransistor N_1 auf das Potenzial der Bitleitung auf- bzw. entladen wird. Beim Lesen der Speicherzelle geht die darin eingespeicherte Information verloren, da die Bitleitungskapazität C_{BL} deutlich größer als die Speicherkapazität C ist. Der nutzbare Spannungsunterschied an der Bitleitung beträgt beim Auslesen

$$U_{BL} \approx \frac{U_{DD}}{2} \frac{C}{C + C_{BL}} \quad (3.24)$$

Deshalb müssen sehr empfindliche Leseverstärker eingesetzt werden. Wenn der Transistor N_1 ausgeschaltet ist, also kein Lesevorgang stattfindet, geht die eingespeicherte Information mit der Zeit aufgrund von Leckströmen ebenfalls verloren. Daher erfordern beide Gegebenheiten ein Wiedereinschreiben der gespeicherten Information. Dieser Refresh-Vorgang erfolgt über den rückgekoppelten Transfertransistor N_2 , dessen Auslösung über die Leitung RL erfolgt. Hierfür ist eine separate Refresh-Schaltung notwendig, was zusätzlichen Aufwand bedeutet. Die Refresh-Zyklen finden üblicherweise im Millisekundenbereich statt.

3.3.3 D-Flipflop

Jedes sequenziell arbeitende Netzwerk besteht aus einem kombinatorischen Anteil und aus Speicherelementen, in denen das Ergebnis des kombinatorischen Netzwerkes taktsynchron zwischengespeichert wird. Hierfür eignen sich sogenannte D-Flipflops. Ein D-Flipflop hat einen Dateneingang D , einen Datenausgang Q und einen Takteingang ϕ (Abbildung 3.17). Die Funktion des D-Flipflops besteht darin, beim Anlegen einer aktiven Taktflanke, die je nach interner Verdrahtung eine positive oder eine negative Flanke sein kann, den Eingangswert D auf den Ausgang Q durchzuschalten. Solange das Taktsignal konstant ist bzw. die nicht-aktive Taktflanke auftritt, wird der Ausgang Q unabhängig vom Eingang D gehalten.

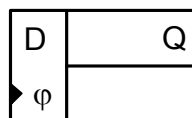


Abbildung 3.17: Symbolschaltbild eines D-Flipflop

Realisiert wird dieses Verhalten durch zwei in Serie geschaltete Latches. Abbildung 3.18 zeigt einen detaillierten Schaltplan eines D-Flipflops auf Transistorebene. Die beiden Latches sind

markiert. Betrachtet man das Latch, erkennt man, dass solange das Taktsignal ϕ einen Low-Pegel aufweist, das obere Transferrgatter geöffnet ist. Dadurch wird der invertierte Wert D in das Latch eingeschrieben bzw. an seinen Ausgang invertiert weitergeleitet. Der Ausgang des Latches folgt dem Eingangssignal, weshalb man in diesem Fall von der Folgephase spricht, oder anders ausgedrückt, dass das Latch in diesem Zustand transparent ist. Kommt es nun zu einem Wechsel des Taktsignals, schließt das obere Transferrgatter und das untere Transferrgatter wird geöffnet. Folglich entsteht über die beiden Inverter eine Rückkopplung des eingeschriebenen Signals. Die beiden Inverter verstärken das Signal, um die Flankensteilheit wiederherzustellen. Das Eingangssignal wird gehalten, das heißt, das Latch befindet sich in der Haltephase. Das zweite Latch wird mit dem gegenphasigen Takt angesteuert. Das bedeutet, dass solange sich das erste Latch in der Folgephase befindet, wird der Ausgang Q des zweiten Latches gehalten. Umgekehrt folgt daraus, dass das zweite Latch den Wert des ersten Latches an den Ausgang Q weitergibt, wenn das erste Latch sperrt. Somit ist das gesamte Flipflop nie transparent, da sich immer ein Latch in der Haltephase befindet.

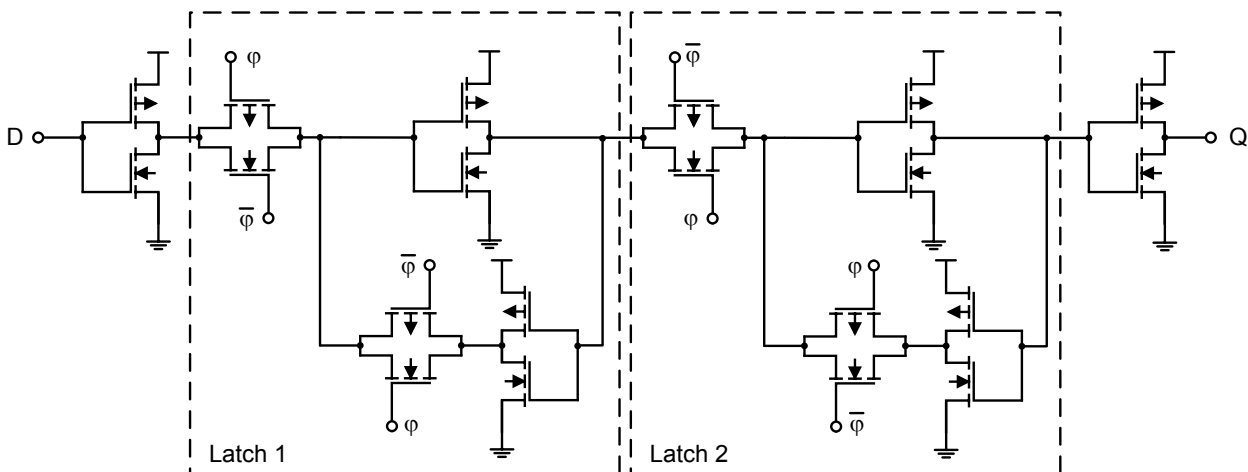


Abbildung 3.18: Schaltplan eines D-Flipflops

Charakteristische Kenngrößen für ein Flipflop sind die Rüstzeit (engl. Setup-time) t_{Setup} , die Haltezeit (engl. Hold-time) t_{Hold} und die interne Verzögerung des Signals bis zum Ausgang, welche auch „Clock-To-Output“-Zeit t_{C2O} genannt wird. Unter der Rüstzeit versteht man die Zeit, zu der das Eingangssignal D vor der aktiven Taktflanke ϕ am Flipflop anliegen muss. Die Verzögerungszeit des Flipflops gibt an, wie lange es dauert, bis das Eingangssignal D an den Ausgang Q nach dem Anliegen der aktiven Taktflanke weitergegeben wird. Die Haltezeit beschreibt die Zeit, wie lange das Signal D nach der aktiven Taktflanke noch mindestens stabil anliegen muss, damit es am Ausgang Q übernommen wird. Da das Eingangssignal aufgrund der vorgeschalteten kombinatorischen Logik immer einer gewissen Verarbeitungszeit unterliegt und sich nicht so schnell ändern kann, wird die Betrachtung der Haltezeit nicht weiterverfolgt. Deshalb wird für den zu entwerfenden Viterbi-Entzerrer ein Flipflop so entworfen, dass sich eine in der Summe aus Rüst- und Verzögerungszeit möglichst kurze Zeit ergibt. Am besten lässt sich

die Verzögerungszeit t_{C20} verkürzen, indem man den oberen Inverter im zweiten Latch einspart und ihn in den Rückkoppelpfad einbaut [43]. Um am Ausgang Q ein nicht-invertiertes Ausgangssignal zu erhalten, muss auch der Ausgangsinverter wegfallen. Dadurch spart man insgesamt zwei Inverterlaufzeiten ein. Abbildung 3.19 zeigt einen Schaltplan des optimierten Flipflops. Die Treiberfähigkeit des oberen Transfegatters des zweiten Latches muss jedoch erhöht werden, um die externe Last des Ausgangsknotens Q umzuladen.

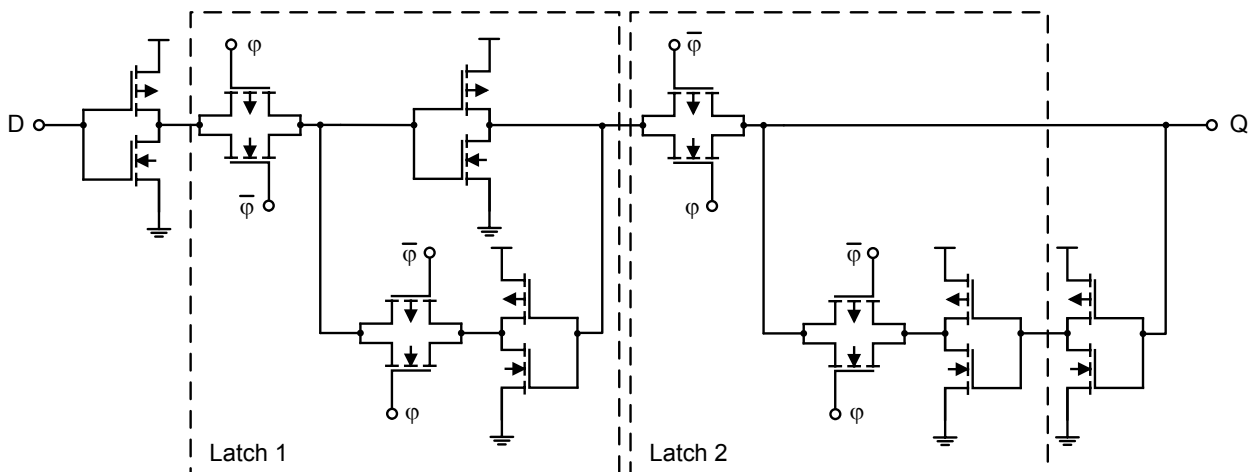


Abbildung 3.19: Schaltplan des optimierten D-Flipflops

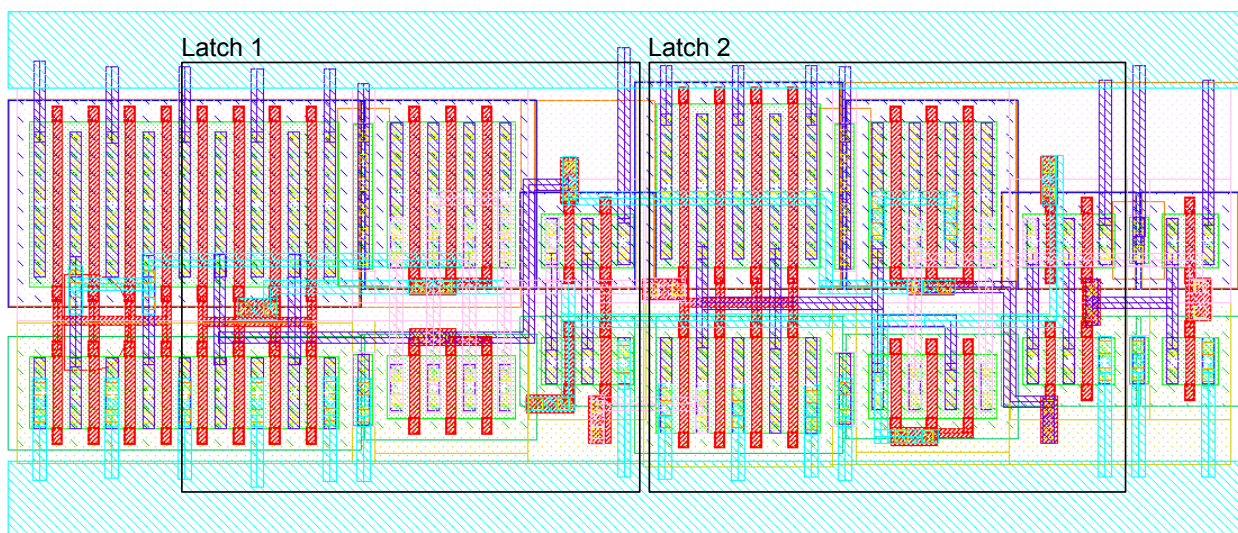


Abbildung 3.20: Layout des optimierten D-Flipflops

Zur Ermittlung der Rüst- und der Verzögerungszeit des Flipflops wird eine Scharparameter-simulation durchgeführt (siehe Abbildung 3.21). Die Eingangsflanke D wird im 2 ps Abstand immer weiter an die aktive Taktflanke ϕ herangeschoben. Solange sich daraufhin das Ausgangssignal Q noch ändert, ist die Rüstzeit des Flipflops noch eingehalten. Je kürzer die Rüstzeit ist, desto länger ist die Verzögerungszeit. Das lässt sich dadurch erklären, dass sich das

Flipflop bei kurzer Rüstzeit in der Nähe seines labilen Arbeitspunktes befindet, da die Abtastung des Eingangssignals D schon während seiner steigenden bzw. fallenden Flanke erfolgt. Dies erkennt man in Abbildung 3.21 nicht direkt, da die Abtastung um eine Inverterlaufzeit verzögert (vgl. Schaltplan in Abbildung 3.19) gegenüber dem Eingangssignal D stattfindet.

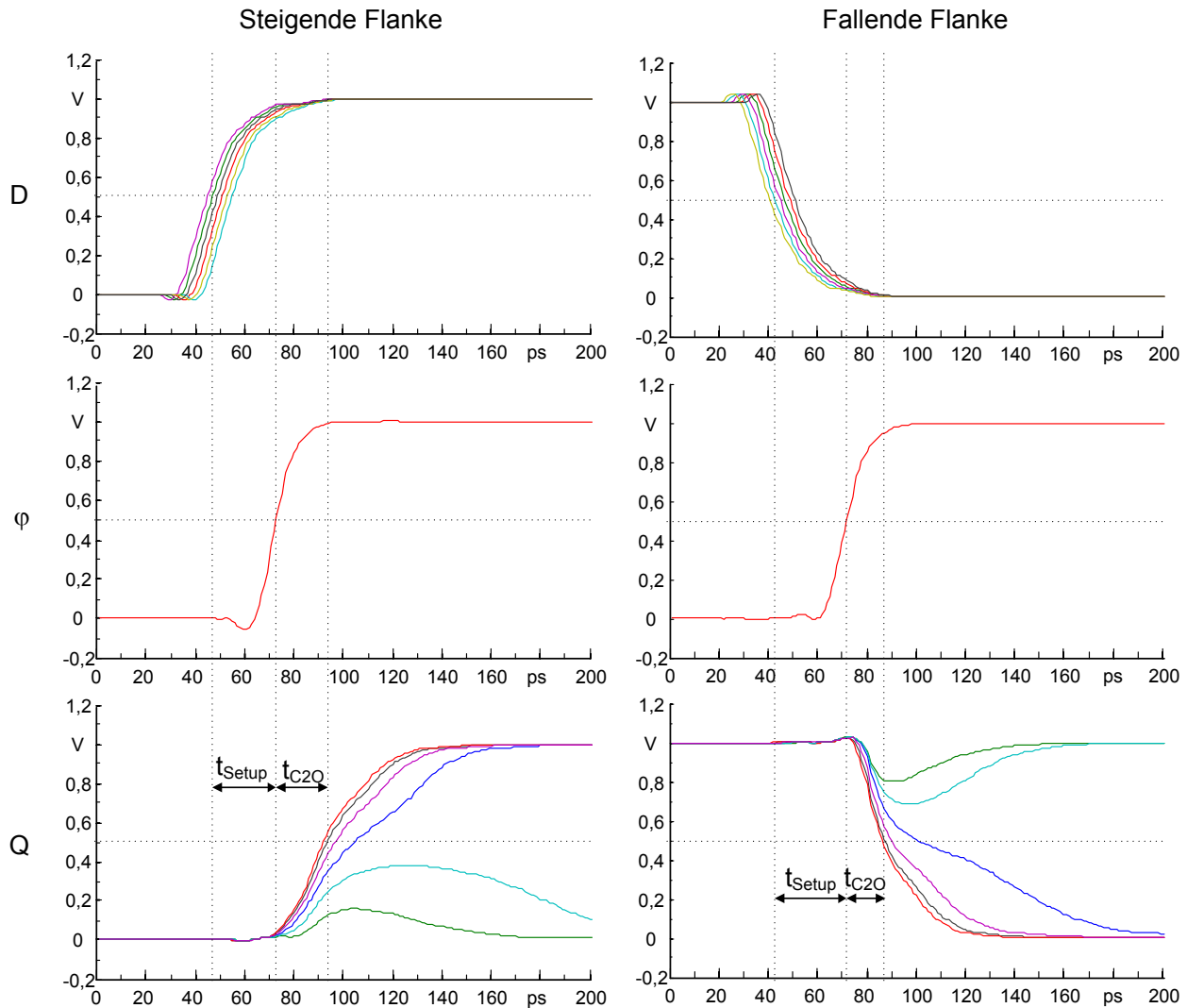


Abbildung 3.21: Simulation der Rüst- und Verzögerungszeit des optimierten D-Flipflops

Tabelle 3.6 zeigt eine Zusammenfassung der Scharparametersimulation aus Abbildung 3.21 des optimierten D-Flipflops. Es werden für jede der drei Simulationsarten diejenigen Rüst- und Verzögerungszeiten dargestellt, die in der Summe die geringsten Zeiten aufweisen. Legt man der Simulation typische Transistoren (TT) und typische parasitäre Elemente der Verdrahtung (RCTYP) zugrunde, beträgt die langsamste Summenzeit aus Rüst- und Verzögerung 54,1 ps. Dies hat zur Folge, dass der vorgeschalteten kombinatorischen Logik 54,1 ps weniger Zeit als die Periodendauer der Taktfrequenz des sequenziellen Netzwerkes zur Verfügung steht.

Tabelle 3.6: Simulationsergebnisse der Rüst- und Verzögerungszeit des optimierten D-Flipflops

	steigende Flanke an D			fallende Flanke an D		
	t_{Setup}	t_{C2O}	$t_{\text{Setup}} + t_{\text{C2O}}$	t_{Setup}	t_{C2O}	$t_{\text{Setup}} + t_{\text{C2O}}$
Schaltplan, TT	26,3 ps	20,0 ps	46,3 ps	28,7 ps	15,7 ps	44,4 ps
RCTYP, TT	28,6 ps	25,5 ps	54,1 ps	35,0 ps	18,6 ps	53,6 ps
RCMAX, SS	36,4 ps	28,2 ps	64,6 ps	45,8 ps	28,6 ps	74,4 ps

3.3.4 Schmitt-Trigger

Beim Schmitt-Trigger handelt es sich um eine Schaltung mit Hysterese. Das heißt, die Schaltschwellen für ansteigende bzw. abfallende Eingangsspannungen sind verschieden. Dieses Schaltverhalten benötigt man an Eingängen einer Schaltung, deren Eingangssignale keine große Flankensteilheit aufweisen bzw. mit Störsignalen beaufschlagt sind. Hätte die Eingangsschaltung nur eine Schaltschwelle, würden kleine Änderungen in der Nähe des Schaltpunktes zu einem unerwünschten Kippen des Ausgangslogikpegels führen. Eine Schaltung mit Hysterese hingegen verhindert diesen unerwünschten Schaltvorgang.

Abbildung 3.22 zeigt einen Schaltplan einer Hystereseschaltung mit nicht-invertierendem Schaltverhalten. Die Hysterese lässt sich durch Rückkopplung des Ausgangssignals auf den Eingang bzw. im Fall eines nicht-invertierenden Schmitt-Triggers auf den Zwischenknoten X realisieren.

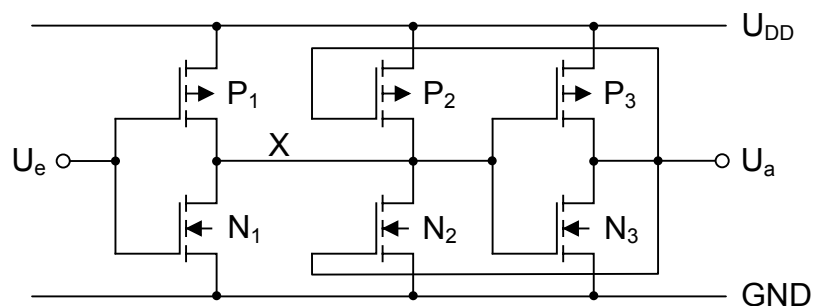


Abbildung 3.22: Schmitt-Triggerschaltung

Der Schmitt-Trigger besteht im Wesentlichen aus zwei in Serie geschalteten Invertern (P_1, N_1 sowie P_3, N_3). Der Ausgang des Inverters, bestehend aus P_3 und N_3 , wird über einen weiteren Inverter, bestehend aus P_2 und N_2 , auf den Zwischenknoten X rückgekoppelt. Zwei in Serie geschaltete und rückgekoppelte Inverter wirken wie eine Speicherzelle, die ihren eingespeicherten Zustand selbst erhält. Ähnlich verhält es sich bei diesem Schmitt-Trigger, bei dem der innere Zustand, der auf dem Knoten X gespeichert ist, von außen über den Inverter, bestehend aus P_1 und N_1 , umgeschrieben werden muss. Die beiden Inverter P_1, N_1 sowie P_3, N_3

werden wie Standard-Inverter mit den Weiten $W_{P1,3} = 2,3 \mu\text{m}$ und $W_{N1,3} = 1,0 \mu\text{m}$ dimensioniert. Möchte man die Schaltschwellen so legen, dass bei einer ansteigenden Eingangsspannung U_e der Ausgangspegel U_a sich bei etwa $U_e = U_{e+} = 0,65 \text{ V}$ von einem Low-Pegel in einen High-Pegel ändert bzw. bei einer fallenden Eingangsspannung der Ausgangspegel U_a sich bei etwa $U_e = U_{e-} = 0,35 \text{ V}$ vom High-Pegel in den Low-Pegel ändert, können die Weiten der Transistoren P_2 und N_2 wie folgt berechnet werden. Im Kippunkt gilt: $U_x = \frac{U_{DD}}{2} = 0,5 \text{ V}$.

Fall 1:

U_e soll von 0 V bis $U_{DD} = 1 \text{ V}$ ansteigen, wobei die Schaltschwelle bei $U_{e+} = 0,65 \text{ V}$ liegen soll. Kurz vor dem Kippen sei $U_a = 0 \text{ V}$. Unter den genannten Spannungspegeln an U_e und U_x gilt nach den Bedingungen der MOSFET-Gleichungen (3.2) bis (3.4), dass sich die Transistoren P_1 und N_1 im Sättigungsbereich befinden, sich P_2 im linearen Bereich befindet und N_2 sperrt. Für die Bestimmung der Weite von N_2 , wird die Knotenregel am Knoten X angesetzt.

$$-I_{DP1} - I_{DP2} = I_{DN1} \quad (3.25)$$

Setzt man die Beziehungen für die entsprechenden Arbeitsbereiche der Transistoren ein, ergibt sich

$$\frac{1}{2} \beta_p (U_{GSP1} - U_{Tp})^2 + \beta_p \left[(U_{GSP2} - U_{Tp}) U_{DSP2} - \frac{U_{DSP2}^2}{2} \right] = \frac{1}{2} \beta_n (U_{GSN1} - U_{Tn})^2 \quad (3.26)$$

Für $\beta_n = 2,3\beta_p$, $W_{N1} = 1 \mu\text{m}$, $W_{P1} = 2,3 \mu\text{m}$ und $L = 90 \text{ nm}$ ergibt sich mit Hilfe von (3.5)

$$W_{P2} = 2,3 \mu\text{m} \frac{(U_{GSN1} - U_{Tn})^2 - (U_{GSP1} - U_{Tp})^2}{2 \left[(U_{GSP2} - U_{Tp}) U_{DSP2} - \frac{U_{DSP2}^2}{2} \right]} = 0,97 \mu\text{m} \quad (3.27)$$

Fall 2:

U_e soll von 1 V auf 0 V abfallen, wobei die Schaltschwelle bei $U_{e-} = 0,35 \text{ V}$ liegen soll. Kurz vor dem Kippen, sei $U_a = 1 \text{ V}$. Unter den genannten Spannungspegeln an U_e und U_x gilt nach den Bedingungen der MOSFET-Gleichungen (3.2) bis (3.4), dass sich die Transistoren P_1 und N_1 im Sättigungsbereich befinden, sich N_2 im linearen Bereich befindet und P_2 sperrt. Die Knotenregel bezüglich X sieht wie folgt aus

$$-I_{DP1} = I_{DN1} + I_{DN2} \quad (3.28)$$

Analog zu (3.26) und (3.27) ergibt sich

$$\frac{1}{2} \beta_p (U_{GSP1} - U_{Tp})^2 = \frac{1}{2} \beta_n (U_{GSN1} - U_{Tn})^2 + \beta_n \left[(U_{GSN2} - U_{Tn}) U_{DSN2} - \frac{U_{DSN2}^2}{2} \right] \quad (3.29)$$

$$W_{N2} = 1 \mu\text{m} \frac{(U_{GSP1} - U_{Tp})^2 - (U_{GSN1} - U_{Tn})^2}{2 \left[(U_{GSN2} - U_{Tn}) U_{DSN2} - \frac{U_{DSN2}^2}{2} \right]} = 0,24 \mu\text{m} \quad (3.30)$$

Die bestimmten Weiten W_{P2} und W_{N2} dienen nun als Richtlinie zur Dimensionierung. Unter Berücksichtigung der Entwurfsregeln können die berechneten Werte nicht anpassungslos im Layout umgesetzt werden. Im Layout werden als Transistorweiten $W_{P2} = 1,2 \mu\text{m}$ und $W_{N2} = 1,0 \mu\text{m}$ gewählt.

Die Simulation zeigt unter Berücksichtigung von Prozessschwankungen folgende Ergebnisse bezüglich der Ausgangsspannung U_a .

Tabelle 3.7: Simulationsergebnisse des Schmitt-Triggers

	U_a ($U_e: 0 \text{ V} \rightarrow 1 \text{ V}$)	U_a ($U_e: 1 \text{ V} \rightarrow 0 \text{ V}$)
Schaltplan, SS	670 mV	330 mV
Schaltplan, TT	650 mV	350 mV
Schaltplan, FF	630 mV	370 mV

Die Ergebnisse weichen von den berechneten Daten ab. Dies lässt sich aber über das bei der Berechnung verwendeten vereinfachten Transistormodells begründen. Die Berechnung dient jedoch der groben Abschätzung der Transistordimensionen.

3.4 Stromschalterlogik

Bei der Stromschalterlogik (engl. Current-Mode-Logic, CML) handelt es sich im Gegensatz zur statischen CMOS-Logik um eine differenzielle Logikfamilie. Das bedeutet, dass sowohl die Eingangs- als auch die Ausgangssignale gleichzeitig in inverser und nicht-inverser Form vorliegen. Die Information ist in der Differenz gespeichert. Durch einfaches Vertauschen der Ausgangssignalleitungen lassen sich somit invertierte und nicht-invertierte Logikfunktionen erzeugen. Abbildung 3.23 zeigt einen Schaltplan der einfachsten CML-Schaltung. Betrachtet man die Ein- und Ausgangssignale analog, spricht man von einem CML-Verstärker. Bei digitaler Betrachtung versteht man unter der Schaltung bei der eingezeichneten Benennung der Ein- und

Ausgangssignale eine Identitätsfunktion. Vertauscht man die Ausgangssignale, entsteht daraus ein Inverter.

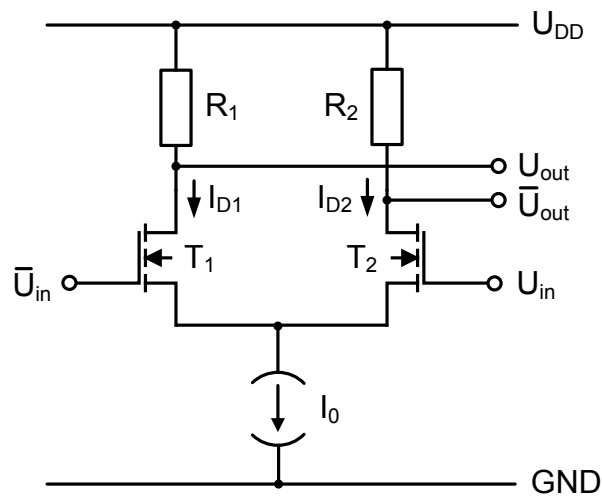


Abbildung 3.23: CML-Verstärker

Typisch für CML-Schaltungen ist, dass immer ein konstanter Strom I_0 zwischen der Versorgungsspannung und dem Massepotenzial fließt, der sich auf die beiden Zweige aufteilt. Er wird über eine Referenzspannungsquelle, die noch näher erläutert wird, eingestellt. Liegt am Eingang U_{in} ein logischer High-Pegel und an \bar{U}_{in} ein logischer Low-Pegel an, gilt für die Spannungspegel am Ausgang $U_{out} = U_{DD} - R_1 I_0 = U_{min}$ und $\bar{U}_{out} = U_{DD} = U_{max}$. Der maximale Signalhub ist definiert zu $\Delta U_{out} = R I_0$ für $R = R_1 = R_2$.

Die CML-Technik hat folgende Vorteile [46], [47]:

- Höhere Schaltgeschwindigkeit gegenüber komplementärer CMOS-Logik, da nur schnelle n-Kanaltransistoren Verwendung finden.
- Robust gegenüber Gleichtaktstörungen, da Störungen (z.B. Einkopplungen/Übersprechen benachbarter Taktleitungen oder Versorgungsspannungsrauschen bzw. -schwankungen) in der Regel auf beide Leitungen erfolgen, jedoch durch Differenzbildung kompensiert werden.
- Konstanter Stromverbrauch, so dass es keine Versorgungsspannungseinbrüche aufgrund von Schaltvorgängen gibt, die zu Fehlinterpretationen von Logikpegeln führen können.

Als Nachteile sind allgemein ein höherer Strom- und Flächenverbrauch gegenüber statischer CMOS-Logik zu nennen, weshalb CML für Höchstintegration ungeeignet ist.

3.4.1 CML-Verstärker

Charakteristische Eigenschaften des CML-Verstärkers sind die Spannungsverstärkung und seine Bandbreite. Je höher die Bandbreite, desto größer ist die Schaltgeschwindigkeit, welche die maximale Taktfrequenz limitiert. Die Spannungsverstärkung ist eine Größe, welche die Regeneration des Signals beschreibt. Ist die Spannungsverstärkung größer als 1 kann der Verstärker ein schwaches Eingangssignal verstärken. Eine Spannungsverstärkung kleiner als 1 ist für einen Verstärkerbetrieb irrelevant, da dann das Signal gedämpft werden würde. Die Verstärkung kann unter der Voraussetzung, dass die Transistoren T_1 und T_2 im Sättigungsbereich arbeiten, gleich dimensioniert sind und die Kanallängenmodulation vernachlässigt wird, wie in (3.31) berechnet werden. Für den differentiellen Ausgangsstrom gilt [46]:

$$\Delta I = I_{D1} - I_{D2} = \frac{1}{2} \mu_n C'_{ox} \frac{W}{L} (\bar{U}_{in} - U_{in}) \sqrt{\frac{4I_0}{\mu_n C'_{ox} \frac{W}{L}} - (\bar{U}_{in} - U_{in})^2} \quad (3.31)$$

Ersetzt man den differentiellen Ausgangsstrom ΔI durch die differentielle Ausgangsspannung $\Delta U_{out} = -R(I_{D1} - I_{D2})$, ergibt sich für $\Delta U_{in} = U_{in} - \bar{U}_{in}$

$$\Delta U_{out} = R \frac{1}{2} \mu_n C'_{ox} \frac{W}{L} \Delta U_{in} \sqrt{\frac{4I_0}{\mu_n C'_{ox} \frac{W}{L}} - (\Delta U_{in})^2} \quad (3.32)$$

Die maximale Spannungsverstärkung V_U (Gleichtaktverstärkung) am Schaltpunkt $\Delta U_{in} = 0V$ erhält man durch Bildung der Ableitung

$$V_U = \frac{\delta \Delta U_{out}}{\delta \Delta U_{in}} = \sqrt{\mu_n C'_{ox} \frac{W}{L}} I_0 R \quad (3.33)$$

Die Bandbreite des CML-Verstärkers und die daraus resultierende maximale Taktfrequenz der Schaltung kann man abschätzen, indem man an seinen Ausgängen sowohl die interne Lastkapazität also auch die externe Last aus Folgestufen und Verdrahtung C_L berücksichtigt. Die 3-dB-Bandbreite errechnet sich aus dem Kehrwert der Zeit τ , die der Ausgangsknoten zum Umladen benötigt.

$$\omega_{3dB} = \frac{1}{\tau} = \frac{1}{RC_L} \quad (3.34)$$

Das Bandbreite-Verstärkerprodukt lässt sich nun mit (3.33) und (3.34) zu

$$V_U \omega_{3dB} = \sqrt{\mu_n C'_{ox} \frac{W}{L} I_0 R \frac{1}{RC_L}} = \sqrt{\mu_n C'_{ox} \frac{W}{L} I_0 \frac{1}{C_L}} \quad (3.35)$$

bestimmen. Da $C_L \sim W$ ist, führt das zu

$$V_U \omega_{3dB} \sim \sqrt{\frac{I_0}{W}} \quad (3.36)$$

Für ein maximales Bandbreite-Verstärkerprodukt müsste man einerseits I_0 maximieren und W minimieren. Eine Minimierung der Transistorkanalweite W führt zwar zu einer geringeren internen Lastkapazität, aber nach Gleichung (3.33) auch zu einer geringeren Verstärkung. Eine Erhöhung von I_0 erfordert aber eine Verringerung des Widerstandes R , damit der maximale Signalhub nicht überschritten wird. Bei zu hohem Signalhub steht für den Stromquellentransistor nicht mehr genügend Drain-Source-Spannung zur Verfügung, um ihn im Sättigungsbereich zu betreiben. Der Stromquellentransistor muss im Sättigungsbetrieb arbeiten, damit sein Drain-Strom unabhängig von seiner Drain-Source-Spannung ist. Ein weiterer Nachteil eines hohen Stromes I_0 ist der allgemein höhere Stromverbrauch der Gesamtschaltung. Für ein maximales Bandbreite-Verstärkerprodukt und daraus resultierender maximaler Geschwindigkeit sollte der CML-Verstärker auf maximal möglichen Hub an R dimensioniert werden.

Abbildung 3.24 zeigt einen CML-Verstärker mit Stromquellentransistor. Die Versorgungsspannung der CMOS-Technologie ist mit 1 V angegeben. Dimensioniert man auf 600 mV Hub, stehen für die beiden Schalttransistoren T_1 und T_2 sowie den Stromquellentransistor T_3 noch jeweils 200 mV Drain-Source-Spannung zur Verfügung. Die Schwellenspannung der n-Kanaltransistoren beträgt $U_{Th} = 170$ mV. Unter Berücksichtigung des Langkanal-MOSFET-Modells bedeutet dies, dass die Gate-Source-Spannung des Stromquellentransistors T_3 im Bereich $170 \text{ mV} < U_{GS3} < 370 \text{ mV}$ liegen muss. Die Langkanalnäherung zur Bestimmung der Sättigungsspannung U_{DSsat} gilt für moderne CMOS-Technologien, wie für die hier eingesetzte 90 nm-CMOS-Technologie, nicht mehr. Die Elektronengeschwindigkeit steigt nämlich ab einer elektrischen Feldstärke von etwa $E = 10^6 \text{ V/m}$ nicht mehr linear mit der Feldstärke an, sondern erreicht etwa bei $E_{sat} = 4 \cdot 10^6 \text{ V/m}$ ihre Sättigungsgeschwindigkeit [48]. Die Sättigungsspannung errechnet sich für kurzkanalige MOSFETs daher wie folgt:

$$U_{DSsat} = \frac{(U_{GS} - U_{Th})(LE_{sat})}{(U_{GS} - U_{Th}) + (LE_{sat})} \quad (3.37)$$

Wählt man eine Gate-Source-Spannung von $U_{GS} = 400 \text{ mV}$, erhält man als Sättigungsspannung $U_{DSsat} = 140 \text{ mV}$. Das hat zur Folge, dass nach dem Kurzkanalmodell der Übergang vom Sättigungs- zum Widerstandsbereich erst unterhalb von 140 mV erfolgt. Beim Langkanalmodell würde der Übergang bei gleicher Gate-Source-Spannung schon unterhalb von 230 mV erfolgen. Nach Gleichungen (3.4) und (3.5) verlangen kleine Gate-Source-Spannungen im Vergleich zur Versorgungsspannung große Transistorweiten gegenüber höheren Gate-Source-Spannungen für einen gewünschten Strom. Bei höheren Gate-Source-Spannungen kann die Transistorweite zwar geringer ausfallen, jedoch muss für den Sättigungsbetrieb die Drain-Source-Spannung ebenfalls erhöht werden, was zwangsläufig zu einem niedrigeren Hub führt. Legt man für die Bias-Spannung $U_{Bias} = U_{GS} = 400 \text{ mV}$ den Strom auf $I_0 = 1 \text{ mA}$ fest, ergibt sich für T_3 nach Gleichung (3.4) eine Weite von $W = 19 \text{ }\mu\text{m}$.

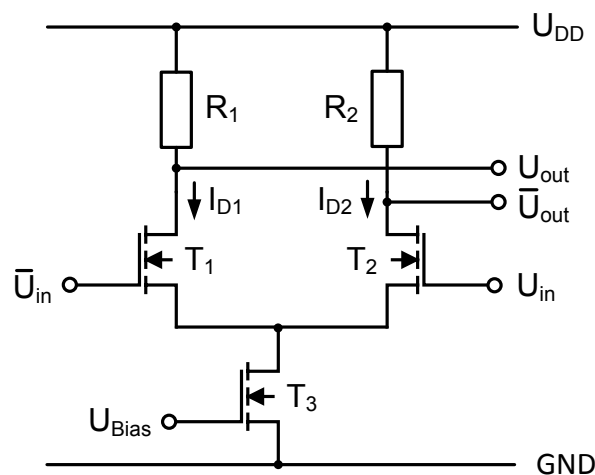


Abbildung 3.24: CML-Verstärker mit Stromquellentransistor

Aufgrund des relativ hohen Ausgangsspannungshubes erfolgt am Eingang eines folgenden CML-Verstärkers Kompression, da die Eingangsspannung so groß ist, dass der Strom vollständig zwischen den beiden Zweigen umgeschaltet wird. Das heißt, der nicht leitende Transistor des Differenzpaares aus T_1 und T_2 sperrt. Am Ausgang erfolgt ebenfalls Kompression, da der leitende Transistor für $U_{GS} = 0,8 \text{ V}$ vom Sättigungsbereich in den Widerstandsbereich übergeht. Nach der Kurzkanalnäherung (3.37) beträgt seine minimale Sättigungsspannung $U_{DSsat,min} = 0,229 \text{ V}$. Um $U_{DSsat,min}$ zu erreichen, müsste der Hub auf $U_{DD} - U_{SS} - U_{DSsat,min} - U_{DS,T3} = 0,571 \text{ V}$ begrenzt werden. Im Gegensatz zu Bipolartransistoren (dort entspricht der Sättigungsbereich dem Widerstandsbereich der MOSFETs) steigt die Verzögerungszeit bei CMOS-Transistoren im Widerstandsbereich nicht an [47], so dass sich diese Dimensionierung nicht nachteilig auf das Schaltverhalten auswirkt.

Die Schalttransistoren T_1 und T_2 sollen einerseits die maximale Schaltgeschwindigkeit aufweisen und andererseits eine ordentliche Pegelregeneration durchführen. Bei einer Gate-Source-Spannung von $U_{GS} = 0,8 \text{ V}$ und einer zu erwartenden Drain-Source-Spannung von $U_{DS} = 0,2 \text{ V}$

beträgt der Drain-Strom etwa $I_D \approx 250 \mu\text{A}/\mu\text{m}$. Um einen Strom von 1 mA zu erreichen, müsste man die Schalttransistoren mindestens $4 \mu\text{m}$ weit machen. Für eine ordentliche Pegelregeneration wird eine Verstärkung von $V_U = 1,8$ angestrebt, damit auch nach dem Layout aufgrund der parasitären Elemente noch genügend Reserve da ist. Diese Verstärkung wird für $W_1 = W_2 = 5 \mu\text{m}$ erreicht. Die Verzögerungszeit liegt bei $t_p = 5,3 \text{ ps}$. Ein so dimensionierter CML-Inverter ist damit beinahe doppelt so schnell wie ein statischer CMOS-Inverter ($t_{pLH} = 9,0 \text{ ps}$).

Je nach Anwendungsfall werden verschiedene CML-Verstärker entworfen. Für schnelle Schaltungsteile, wie sie an den Eingangsschnittstellen benötigt werden und wo zugleich Pegelregeneration erforderlich ist, sind CML-Verstärker mit 1 mA notwendig. Sobald die Datenrate intern geteilt ist und somit die Geschwindigkeitsanforderungen geringer sind, werden stromsparende CML-Verstärker mit 0,5 mA eingesetzt. Zur Verzögerung interner Signale können noch stromsparendere CML-Verstärker mit nur 0,25 mA verwendet werden. Um die Verzögerungszeit flexibler anzupassen, existiert eine Variante mit $t_p = 6,5 \text{ ps}$ und eine weitere mit $t_p = 9,8 \text{ ps}$. Zum Aufbau einer Treiberkette werden CML-Verstärker mit 2 mA und 4 mA eingesetzt. Die Endstufe, die dann eine Last von 50Ω treiben soll, wird aus drei parallel geschalteten 4 mA-CML-Verstärker aufgebaut. Zur Verstärkung des Taktes dient der CML-Verstärker mit 3 mA. Tabelle 3.8 zeigt eine Übersicht über die Dimensionierung der verschiedenen CML-Verstärker sowie über den Zusammenhang zwischen Strom, Verstärkung und Verzögerungszeit. Alle CML-Verstärker mit Ausnahme dessen, welcher den Takt verstärkt, sind auf einen Ausgangsspannungshub von $U_{\text{Hub}} = 600 \text{ mV}$ ausgelegt. Der CML-Verstärker für den Takttreiber könnte theoretisch, um große Lasten zu treiben, 900 mV Hub erzeugen. Dieser ist aber aufgrund der Versorgungsspannungswahl von $U_{DD} = 1 \text{ V}$ und dem obligatorischen Spannungsabfall im Stromquellentransistor von $U_{DS} \approx 200 \text{ mV}$ auf geringere Werte beschränkt.

Tabelle 3.8: Parameter verschiedener CML-Verstärker

I_0	V_U	$W_{1,2}$	W_3	$R_{1,2}$	t_p
1 mA	1,81	$5 \mu\text{m}$	$19 \mu\text{m}$	600Ω	$5,3 \text{ ps}$
0,5 mA	2,46	$5 \mu\text{m}$	$9,5 \mu\text{m}$	1200Ω	$9,2 \text{ ps}$
0,25 mA	1,74	$1,26 \mu\text{m}$	$4,75 \mu\text{m}$	2400Ω	$6,5 \text{ ps}$
0,25 mA	2,36	$2,5 \mu\text{m}$	$4,75 \mu\text{m}$	2400Ω	$9,8 \text{ ps}$
2 mA	1,84	$10 \mu\text{m}$	$38 \mu\text{m}$	300Ω	$5,2 \text{ ps}$
4 mA	1,86	$20 \mu\text{m}$	$76 \mu\text{m}$	150Ω	$5,1 \text{ ps}$
3 mA	1,78	$10 \mu\text{m}$	$76 \mu\text{m}$	300Ω	$5,4 \text{ ps}$

3.4.2 Schaltung zur Arbeitspunkteinstellung

Eine wichtige Schaltung, die in CML-Schaltungen den Arbeitspunkt des Stromquellentransistors einstellt, ist in Abbildung 3.25 dargestellt. Der Zweck dieser Schaltung besteht darin, den Referenzstrom I_{Ref} über die Bias-Spannung U_{Bias} in den Stromquellentransistor einer CML-Schaltung zu spiegeln. Aufgrund des kurzgeschlossenen Drain- und Gain-Anchlusses arbeitet der Transistor T_1 im Sättigungsbereich. Für die Bias-Spannung lässt sich folgende Beziehung aufstellen:

$$\frac{R_1 \beta_n}{2} (U_{\text{Bias}} - U_{\text{Th}})^2 + U_{\text{Bias}} - U_{\text{DD}} = 0 \quad (3.38)$$

Man erkennt, dass die Bias-Spannung U_{Bias} nur vom Widerstand R_1 und von den Transistorparametern von T_1 abhängt.

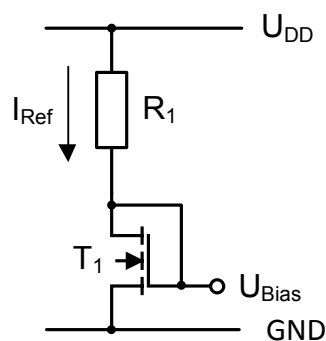


Abbildung 3.25: Schaltung zur Arbeitspunkteinstellung für CML-Bausteine

Im Wesentlichen handelt es sich bei dieser Schaltung um eine Regelschaltung. Steigt der Strom durch den Transistor T_1 infolge von Parameterschwankungen der Bauteile, sinkt die Bias-Spannung ab, was einem weiteren Stromanstieg entgegenwirkt.

Breibt man die Schaltung zur Arbeitspunkteinstellung an einem CML-Verstärker, besteht eine Abweichung von 10 % zwischen Stromquellenstrom und Referenzstrom. Außerdem unterscheiden sich die Drain-Source-Spannung des Stromspiegeltransistors T_1 und des Stromquellentransistors des CML-Verstärkers um 51 mV.

Eine Verbesserung kann erreicht werden, indem man, wie in Abbildung 3.26 dargestellt, einen zweiten Transistor in den Referenzpfad schaltet. Die Dimensionierung des Widerstands mit $R_1 = 600 \Omega$ und der Transistoren T_1 mit $W_1 = 5 \mu\text{m}$ und T_2 mit $W_2 = 19 \mu\text{m}$ entspricht derselben Dimensionierung wie der eines CML-Verstärkers für den Schalt- bzw. Stromquellentransistor. Der Unterschied zwischen Referenzstrom und Stromquellenstrom beträgt damit nur noch wenige Promille. Der Stromspiegel ist daher deutlich besser abgeglichen. Außerdem wird die Drain-Source-Spannung des Stromquellenstromes auf rund 256 mV abgesenkt, so dass ein höherer

Ausgangsspannungshub am Ausgang des CML-Verstärkers zur Verfügung steht. Der Unterschied der Drain-Source-Spannungen zwischen dem Stromspiegeltransistor T_2 und dem Stromquellentransistor des CML-Verstärkers beträgt nur noch 32 mV.

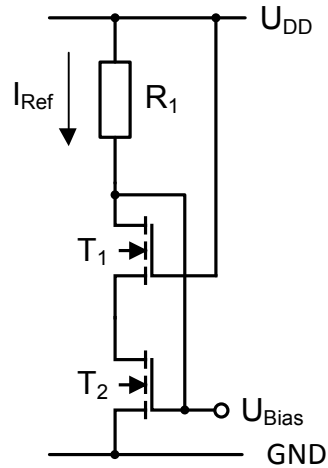


Abbildung 3.26: Optimierte Schaltung zur Arbeitspunkteinstellung für CML-Bausteine

3.4.3 CML-Latch und CML-Flipflop

Das CML-Latch aus Abbildung 3.27 ist eine grundlegende Schaltung, die der Realisierung von Demultiplexern und Multiplexern dient.

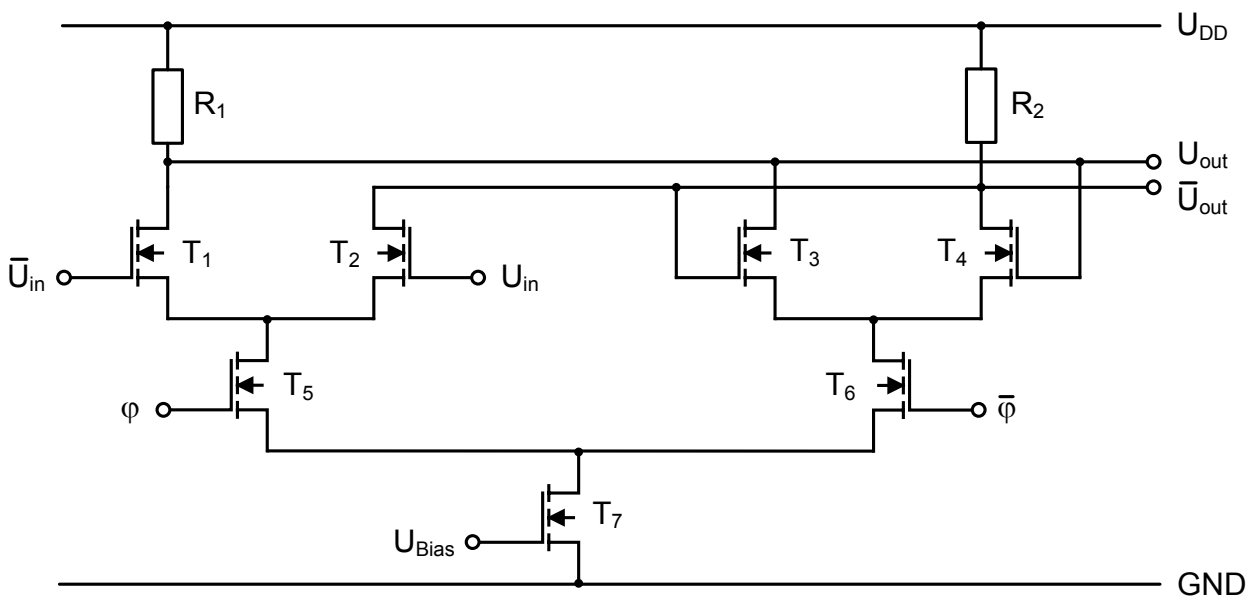


Abbildung 3.27: CML-Latch

Transistor T_7 stellt den Stromquellentransistor dar, dessen Strom über die Bias-Spannung U_{Bias} eingestellt wird. Der eingeprägte Strom fließt abhängig vom Taktsignal φ entweder durch das linke oder das rechte Differenzpaar. Liegt das Taktsignal $\varphi = H$ an, fließt der Strom über die Transistoren T_5 und T_7 ab. Liegt an U_{in} ein logischer High-Pegel an, führt dies am Ausgang U_{out} ebenfalls zu einem High-Pegel. Für die gestrichenen Eingänge gilt gegenteiliges. Daher wirkt die linke CML-Stufe als logische Identitätsfunktion. Das heißt, dass der Ausgang dem Eingang folgt. Das Latch befindet sich daher in der Folgephase und ist transparent. Solange $\varphi = H$ gilt, ist $\bar{\varphi} = L$. Das rechte Differenzpaar ist also ausgeschaltet. Ändert sich der Pegel des Taktsignals von $\varphi = H$ nach $\varphi = L$ bzw. von $\bar{\varphi} = L$ nach $\bar{\varphi} = H$ ist das linke Differenzpaar ausgeschaltet und das rechte Differenzpaar eingeschaltet. Dies bewirkt einerseits, dass nun das Ausgangssignal vom Eingangssignal unabhängig ist und andererseits, da die Gate-Anschlüsse der Transistoren T_3 und T_4 invers mit den Ausgangssignalen U_{out} und \bar{U}_{out} rückgekoppelt sind, eine Verstärkung des Ausgangssignals. Das heißt, dass das zuvor in der Folgephase angelegte Eingangssignal in der zweiten Takthälfte gehalten wird. Daher heißt die zweite Takthälfte Haltephase.

Für den Viterbi-Entzerrer werden zwei CML-Latch-Varianten entworfen. Die Eingangsstufe, die die höchste Datenrate verarbeiten muss, arbeitet mit einem Strom von $I_0 = 1 \text{ mA}$. Folgestufen, die geringere Datenraten verarbeiten, sind, um Strom zu sparen, für $I_0 = 0,5 \text{ mA}$ entworfen. Die Dimensionierung beider Latch-Varianten ist in Tabelle 3.9 angegeben.

Tabelle 3.9: Dimensionierung der CML-Latches

I_0	W_{1-6}	W_7	$R_{1,2}$
1 mA	5 μm	19 μm	600 Ω
0,5 mA	5 μm	9,5 μm	1200 Ω

Zwei in Serie geschaltete CML-Latches bilden ein CML-Flipflop. Dadurch wird gewährleistet, dass das Flipflop nie transparent ist. Dies erreicht man, indem man die beiden Latches invers zueinander taktet. Wenn sich eines der beiden Latches in der Folgephase befindet, befindet sich das andere in der Haltephase und umgekehrt. Die Rüst- und Verzögerungszeiten werden ähnlich wie beim CMOS-Flipflop ermittelt. Tabelle 3.10 enthält die Simulationsergebnisse auf Schaltungsplanebene für typische Transistoren für die kleinste Summe aus Rüst- t_{Setup} und Verzögerungszeit t_{C2O} . Die Stromstärkenangaben beziehen sich jeweils auf eines von beiden enthaltenen Latches.

Tabelle 3.10: Rüst- und Verzögerungszeiten der CML-Flipflops

I_0	t_{Setup}	t_{C2O}
1 mA	15,9 ps	15,0 ps
0,5 mA	18,7 ps	28,0 ps

3.4.4 CML-Multiplexer

Der CML-Multiplexer ist ähnlich wie das CML-Latch aufgebaut. Der Unterschied besteht darin, dass der Eingang des rechten Transistorpaares T_3 und T_4 nicht auf den Ausgang rückgekoppelt ist, sondern wie in Abbildung 3.28 dargestellt, herausgeführt ist.

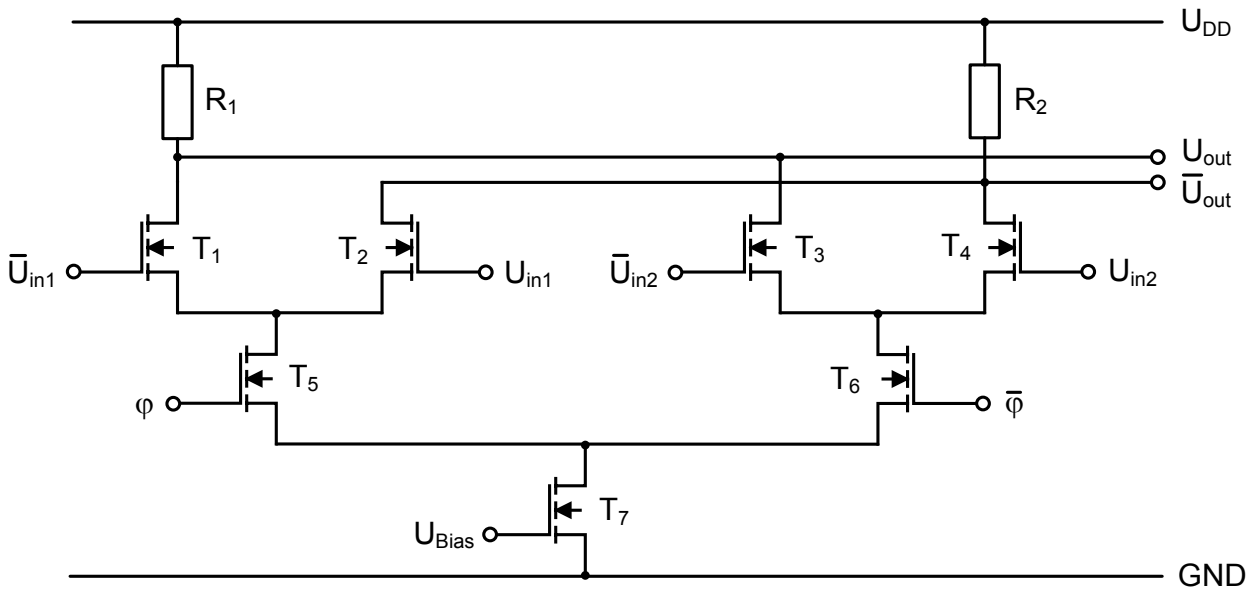


Abbildung 3.28: CML-Multiplexer

Der Multiplexer hat zwei differenzielle Eingänge U_{in1} , \bar{U}_{in1} und U_{in2} , \bar{U}_{in2} . Der Auswähleingang wird von dem Taktsignal ϕ angesteuert. Befindet sich das Taktsignal am Eingang ϕ in der High-Phase, wird das Eingangssignal U_{in1} , \bar{U}_{in1} auf den Ausgang U_{out} , \bar{U}_{out} durchgeschaltet. Weist das Taktsignal an ϕ einen Low-Pegel auf, wird hingegen das Eingangspaar U_{in2} , \bar{U}_{in2} auf den Ausgang geschaltet. Daraus folgt, dass es sich beim Takt bezüglich des Eingangssignals um einen Vollratentakt und bezüglich des Ausgangssignals um einen Halbratentakt handelt. Auch beim Multiplexer müssen wie beim Latch die Rüst- und Verzögerungszeiten eingehalten werden. Die Rüst- und Verzögerungszeiten entsprechen denen des CML-Latches. Die Dimensionierung der Bauteile des Multiplexers sowie die Rüst- und Verzögerungszeiten entsprechen jenen des CML-Latches aus Tabelle 3.9 und Tabelle 3.10.

Durch geringe Änderungen in der Verschaltung des CML-Multiplexers lassen sich noch weitere Funktionen realisieren. Schließt man die Eingänge $U_{in1} = \bar{U}_{in2} = U_{in}$ und $\bar{U}_{in1} = U_{in2} = \bar{U}_{in}$ kurz und interpretiert man nun die Eingänge U_{in} und ϕ als digitale Logiksignale, ergibt sich am Ausgang U_{out} eine logische XOR- bzw. XNOR-Verknüpfung. Betrachtet man die

Eingangssignale als analoge Größen, entsteht eine analoge Multiplizierschaltung, die unter der Bezeichnung Gilbert-Zelle bekannt ist.

3.5 Schnittstellen zwischen CML- und CMOS-Logik

Die CML-Logik ist für schnelle Schaltungsteile wie Ein- und Ausgangsschnittstellen erforderlich, während sich für die digitale Signalverarbeitung statische CMOS-Logik aufgrund der Höchstintegrierbarkeit anbietet. Daher müssen schnelle Eingangssignale zuerst mit Hilfe der CML-Logik parallelisiert werden, um in statischer CMOS-Logik weiterverarbeitet werden zu können. Um den Schnittstellenaufwand nach außen gering zu halten, müssen nach der Datenprozessierung in CMOS die Daten wieder mit Hilfe der CML-Logik serialisiert werden.

Das Problem besteht darin, dass CML- und CMOS-Logik mit unterschiedlichen Spannungspegeln arbeiten, die eine logische 1 oder eine logische 0 repräsentieren. Der Logikhub bei 1 V Versorgungsspannung beträgt bei CML-Logik 600 mV. Bei CMOS hingegen entspricht er der vollen Versorgungsspannung von 1 V. Außerdem sind die Schaltschwellen verschieden. Bei CML liegt der Common-Mode bei 700 mV; bei CMOS liegt die Schaltschwelle hingegen bei $U_{DD}/2 = 500$ mV. Daher sind Schaltungen vonnöten, die die Schaltschwellen angleichen.

3.5.1 Wandlung von CML- zu CMOS-Pegeln

Die Wandlung von CML-Pegeln in CMOS-Pegeln kann mit Hilfe zweier Teilschaltungen erfolgen. Die erste Teilschaltung sorgt für eine Absenkung des Common-Modes, um sich der Schaltschwelle der CMOS-Logik anzunähern. Die zweite Teilschaltung verschiebt die Schaltschwelle noch weiter nach unten und konvertiert den geringen CML-Logikhub in den vollen CMOS-Logikhub. Zunächst wird die erste Teilschaltung (Abbildung 3.29) besprochen, die einem CML-Verstärker stark ähnelt.

Zusätzlich zum normalen CML-Verstärker, enthält die Schaltung den Widerstand $R_1 = 200 \Omega$. Dieser Widerstand erzeugt eine virtuelle Versorgungsspannung $U'_{DD} = U_{DD} - I_0 R_1 = 800$ mV, da aufgrund des Stromquellentransistors unabhängig der Eingangsdifferenzspannung ΔU_{in} immer ein Strom von $I_0 = 1$ mA fließt. Bei Vollaussteuerung der Eingangssignale werden am Ausgang Pegel zwischen $U_L = 400$ mV und $U_H = 800$ mV erreicht. Das heißt, dass der Common-Mode bei $U_{CM} = \frac{U_L + U_H}{2} = 600$ mV liegt. Eine Absenkung des unteren Pegels wird nicht in Betracht gezogen, um eine Mindest-Drain-Source-Spannung von 200 mV der einzelnen Transistoren sicherzustellen.

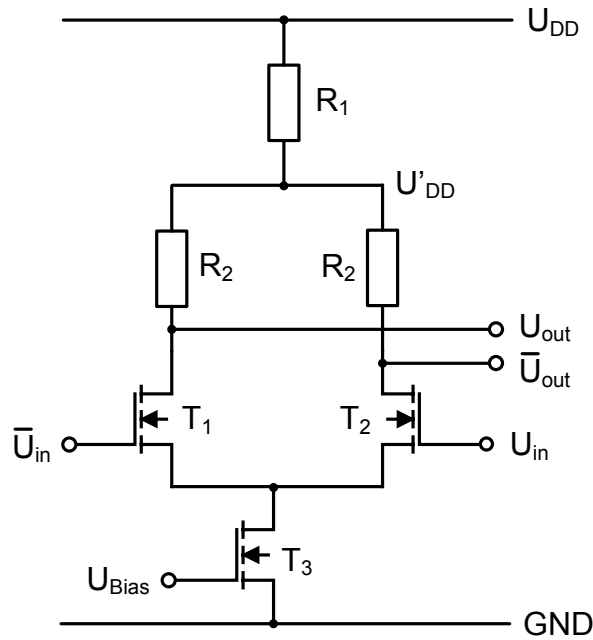


Abbildung 3.29: Teilschaltung des CML- zu CMOS-Pegelkonverters

Nun gilt es, im zweiten Schritt einen CMOS-Inverter zu entwerfen, dessen Eingangsschwelle bei $U_M = U_{CM} = 600 \text{ mV}$ liegt. Die Schaltschwelle eines CMOS-Inverters berechnet sich allgemein zu [41]

$$U_M = \frac{U_{DD} + \sqrt{\frac{\beta_n}{\beta_p}} U_{Tn} + U_{Tp}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \quad (3.39)$$

Für symmetrische Schaltflanken wird $\beta_n = \beta_p$ gewählt. Daraus folgt

$$U_M = \frac{U_{DD} + U_{Tn} + U_{Tp}}{2} \quad (3.40)$$

Die Schaltschwelle lässt sich also nur durch Verwendung von Transistoren mit unterschiedlicher Schwellenspannung für p- und n-Kanal verschieben. Um den Schaltpunkt nach oben zu verschieben, muss beim n-Kanal-MOSFET eine höhere Einsatzspannung gewählt werden als der Betrag der Einsatzspannung des p-Kanal-MOSFETs. Der Halbleiterhersteller stellt einen n-Kanal-MOSFET mit hoher Schwellenspannung von $U_{Tnhvt} = 0,315 \text{ V}$ und einen p-Kanal-MOSFET mit niedriger Schwellenspannung von $U_{Tplt} = -0,15 \text{ V}$ Verfügung. Setzt man diese aus Simulationen ermittelten Schwellenspannungen in Gleichung (3.40) ein, ergibt sich bei 1 V Versorgungsspannung ein Schaltpunkt des CMOS-Inverters bei $U_M = 0,58 \text{ V}$. Simulativ wird ein

Schaltpunkt von $U_M = 0,595$ V ermittelt. Das bedeutet, dass das simulierte Ergebnis noch näher am gewünschten Schaltpunkt von $U_M = 0,6$ V liegt als dem errechneten.

Dieser CMOS-Inverter mit asymmetrischem Schaltpunkt wird an einem der beiden CML-Ausgänge der Schaltung aus Abbildung 3.29, je nachdem ob das Ausgangssignal invertiert oder nicht-invertiert vorliegen soll, angeschlossen. Der andere Ausgang wird offen gelassen.

Weiterhin zeigt die Simulation, dass bei voller Aussteuerung der CML-Ausgangssignale auf $U_L = 0,4$ V bzw. $U_H = 0,8$ V die statischen CMOS High- bzw. Low-Pegel erreicht werden.

Mit Hilfe dieser beiden Schaltungen ist es also möglich, CML-Signale in CMOS-Signale zu wandeln.

3.5.2 Wandlung von CMOS- zu CML-Pegeln

CMOS-Signale sind unipolar, CML-Signale hingegen sind differenziell. Die Wandlung eines unipolaren Signals in ein differenzielles Signal kann mit Hilfe eines CMOS-Inverters erfolgen, der aus einem nicht-invertierten Signal zusätzlich das invertierte zur Ansteuerung der CML-Schaltungselemente bereitstellt. Aufgrund der endlichen Verzögerungszeit eines Inverters ist das invertierte Signal gegenüber dem nicht-invertierten um eine Inverterlaufzeit phasenverschoben. Solange die Phasendifferenz im Verhältnis zur Taktfrequenz klein ist, stellt dies kein Problem dar.

Die Eingangsspiegel der CML-Schaltung entsprechen den vollen statischen CMOS-Pegeln. Das heißt, die Gate-Source-Spannung der Schalttransistoren kann im Sperrbetrieb um den Betrag der Drain-Source-Spannung des Stromquellentransistors negativ werden. Auch dies stellt kein Problem dar.

4 Hardware-Architektur des Viterbi-Entzerrers

Der Viterbi-Entzerrer besteht im Wesentlichen aus den folgenden fünf Komponenten, die in folgenden Teilkapiteln eingehender behandelt werden:

- Zweigmetriekrechner
- Addier-Vergleichs-Auswahleinheit
- Minimumauswahlschaltung
- Pfadrückverfolgungseinheit
- Demultiplexer und Multiplexer

4.1 Zweigmetriekrechner

Die Zweigmetriken werden für jeden Zustand im Trellis über die Beziehung

$$p_{\mu,v,k} = |q_k - z_{\mu,v,k}|^2 \quad (4.1)$$

berechnet. Da es sich bei dem zu entwerfenden Viterbi-Entzerrer um eine digitale Schaltung handelt, liegen alle Werte quantisiert vor. Die Eingangswerte q_k werden von einem vorgeschalteten Analog-Digital-Umsetzer quantisiert. Aufgrund der Kanalmodellierung von Kapitel 2.3 mit acht möglichen Zustandsübergängen, ergeben sich unter Vernachlässigung des Rauschens genau acht verschiedene Empfangswerte. Im Empfänger werden die Empfangswerte, die einem analogen Eingangssignal entsprechen, mit Hilfe eines AD-Wandlers mit 3 bit Auflösung gewonnen.

Daher kann der Zweigmetriekrechner (engl. Branch-Metric-Unit, BMU) als Speicher mit $8 \times 8 = 64$ Tabelleneinträgen realisiert werden. Die Zeilen enthalten die Wahrscheinlichkeiten für einen Abtastwert. Die Spalten enthalten die Wahrscheinlichkeiten für eine bestimmte gesendete Bitfolge. Ein Tabellenelement gibt somit die Wahrscheinlichkeit bei einem bestimmten Eingangswert einer bestimmten Sendefolge wieder.

Um nun die wahrscheinlichste Sendefolge bestimmen zu können, müssen die Einzelwahrscheinlichkeiten über eine gewisse Symbolfolgenlänge [32] nach Formel (2.9) multipliziert werden. Da der eigentliche Wahrscheinlichkeitswert nicht von Interesse ist, und die Multiplikation eine komplexe Rechenoperation ist, kann man durch vorheriges Logarithmieren, die Multiplikation durch eine Addition ersetzen. Dadurch bleibt unter Weglassung konstanter Faktoren nur mehr die Beziehung (2.10) übrig. Diese Werte lassen sich im Vorfeld alle bestimmen und in inverse Wahrscheinlichkeiten, den Kosten bzw. Zweigmetriken umrechnen. Die Quantisierung der Tabelleneinträge richtet sich danach, dass sich aufgrund des Verlaufs der Logarithmusfunktion noch sinnvolle Werte zwischen $-\infty$ (hohe Kosten) und 1 (niedrige Kosten)

ergeben. Eine Quantisierung mit 6 bit zeigt sich als sinnvoll. Die Gesamtspeicherkapazität für die Zweigmetriken beträgt somit 384 bit. Die Bestimmung der Metriken wird in Kapitel 7 noch ausführlich behandelt; sie ist jedoch zum Verständnis dieses Kapitels nicht erforderlich.

Die Umsetzung des Zweigmetrikrechners als Speicher hat auch den Vorteil, dass die Eingangsdaten in einer anderen Kodierung als im Binärzahlensystem wie z.B. im Gray-Code vorliegen können. Bei einer fehlerhaften Übertragung vom AD-Umsetzer zum Viterbi-Entzerrer ändert sich beim Kippen eines Bits der Wert immer nur um 1, während er sich bei Binärcodierung im ungünstigsten Fall um den Betrag der höchstwertigen Stelle einer Zahl ändern kann. Die eigentliche Rechnung entlang des Trellis mit den Metriken kann dann wieder im binären Zahlensystem erfolgen. Der Speicher dient hierbei gleichzeitig als Übersetzer zwischen Gray-Code am Eingang und Binärcode am Ausgang.

4.1.1 Anforderungen an den Speicher

Die Kernkomponente des Zweigmetrikrechners ist ein Speicherfeld für 384 bit. Da der Kanal zeitvariant ist, muss das Speicherfeld generell und auch im Betrieb beschreibbar sein. Daher eignen sich vor allem statische oder dynamische Speicher mit wahlfreiem Zugriff (SRAM bzw. DRAM). Einmal programmierbare Speicher (PROM) scheiden aus. Das Speicherfeld ist in Zeilen und Spalten organisiert. Um für einen bestimmten Empfangswert die möglichen Zweigmetriken auszugeben, muss ein Zeilendekoder die zu lesende Tabellenzeile aktivieren. Über die Schnittstelle zum Beschreiben des Speichers soll eine Zweigmetrik über ein asynchrones Setzsignal eingeschrieben werden können. Daraus folgt, dass der Speicher sowohl zeilen- als auch spaltenweise adressierbar sein muss. Außerdem muss eine Synchronisationsschaltung dafür sorgen, dass neu einzuschreibende Metriken den Auslesevorgang nicht stören. Der Auslesevorgang soll stets Vorrang gegenüber dem Schreibvorgang haben.

4.1.2 Vergleich der Speichertechnologien

Prinzipiell kommen die beiden Speichertechnologien DRAM und SRAM in Frage. Bei DRAM kann eine größere Packungsdichte erreicht werden, da die Anzahl der Transistoren pro Bit geringer ist als bei SRAM. Die Speicherung der Information erfolgt bei DRAM auf einer Kapazität. Hierfür steht die Planar-, Stapel- oder Grabentechnologie zur Verfügung. Die beiden letztgenannten Technologien ermöglichen eine höhere Packungsdichte, da für die Kapazität die dritte räumliche Dimension genutzt wird. Allerdings wird dazu eine spezielle CMOS-Speichertechnologie benötigt. Mit CMOS-Logiktechnologien sind nur planare Anordnungen möglich, was dazu führt, dass für die Speicherkapazität eine größere Fläche benötigt wird. Hierdurch relativiert sich der Flächenvorteil der DRAM- gegenüber der SRAM-Technologie.

DRAM-Speicher benötigen Leseverstärker, da die Bitleitungskapazität wesentlich größer als die eigentliche Speicherkapazität ist, um eine Spannungsänderung der Bitleitung beim Öffnen des Wortleitungstransistors aufgrund des Ladungsverteilungseffektes zu detektieren. Bei einem Speicherfeld von 384 bit werden insgesamt 48 Leseverstärker benötigt, da in jedem Takt 8x6 bit gelesen werden müssen. Zusätzlich ist eine Auffrischungsschaltung (engl. Refresh) notwendig, die im Millisekundenbereich die eingespeicherte Information wieder erneut in die Speicherzelle einschreibt. Die Auffrischungsschaltung benötigt eine Steuerlogik, die Lese-, Schreib- und Wiedereinschreibvorgänge koordiniert. DRAM ist allgemein langsamer als SRAM. Kommerziell erhältliche DDR2-RAM-Module werden mit einer Taktfrequenz von bis zu 266 MHz betrieben, während für den Viterbi-Entzerrer eine Taktfrequenz im GHz-Bereich angestrebt wird.

Daraus wird klar, dass DRAM sich vor allem für große Speicherfelder eignet, wie sie z.B. in PCs und Servern vorkommen. Für die speziellen Anforderungen des Viterbi-Entzerrers hingegen ist ein SRAM-Speicherfeld besser geeignet.

4.1.3 Architektur des Zweigmetrikrechners

Wie in Abbildung 2.12 zu erkennen ist, werden in einem systolischen Blockdekoder (SBD) insgesamt 16 Zweigmetrikrechner (BMUs) benötigt, deren Tabellenelemente alle die identische Information enthalten. Daher bietet es sich an, für einen systolischen Blockdekoder ein zentrales Speicherfeld zu implementieren, den gesamten Speicherinhalt an alle Addier-Vergleichs-Auswahleinheiten (engl. Add-Compare-Select-Unit, ACSU) eines SBDs zu verteilen und die gewünschten Metriken über einen Multiplexer direkt an der jeweiligen ACSU auszulesen.

Abbildung 4.1 zeigt die Architektur des Zweigmetrikrechners auf der obersten Hierarchiestufe. Der Block „BMU Speicher (384 bit)“ speichert alle 64 möglichen Metriken in einem zentralen Speicher. Über die jeweils 6 bit breiten Signale a für die Speicheradressen und d für die eigentlichen Metrikdaten wird das Speicherfeld konfiguriert. Das zum Takt clk asynchrone Schreibfreigabesignal WE initiiert den eigentlichen Schreibvorgang an den einzelnen Speicherzellen. Bei Inbetriebnahme des Viterbi-Entzerrers muss das Speicherfeld konfiguriert werden. Sonst würden die an die BMU angeschlossenen ACSUs zufällige Daten erhalten. Da die Metriken auch im Betrieb nachgeführt werden können, muss das Einschreiben der Metriken in den zentralen Speicher mit dem Auslesevorgang synchronisiert werden, damit zu jedem Zeitpunkt immer definierte Daten zur Verfügung stehen. Die Schaltung zum synchronen Einschreiben der Metriken wird in den folgenden Kapiteln behandelt.

Aus dem zentralen Speicher werden alle 384 bit parallel herausgeführt, so dass sie gleichzeitig an allen ACSUs anliegen. Da es sich hierbei um Leitungen mit mehreren hundert Mikrometern Länge handelt, sind Treiberschaltungen in Form von CMOS-Invertern notwendig, um einerseits Leitungsverluste auszugleichen und Übersprecheffekte zu minimieren. Jeder Treiberblock versorgt vier Multiplexer mit den Metriken. Die einzelnen Multiplexer geben in Abhängigkeit des mit 3 bit quantisierten Eingangswertes q jeweils acht Metriken p_{00} bis p_{31} an eine Stufe im

Trellisdiagramm aus. Zunächst wird jedoch auf die Bausteine des zentralen Speichers eingegangen, wovon das wichtigste Element die einzelne Speicherzelle ist.

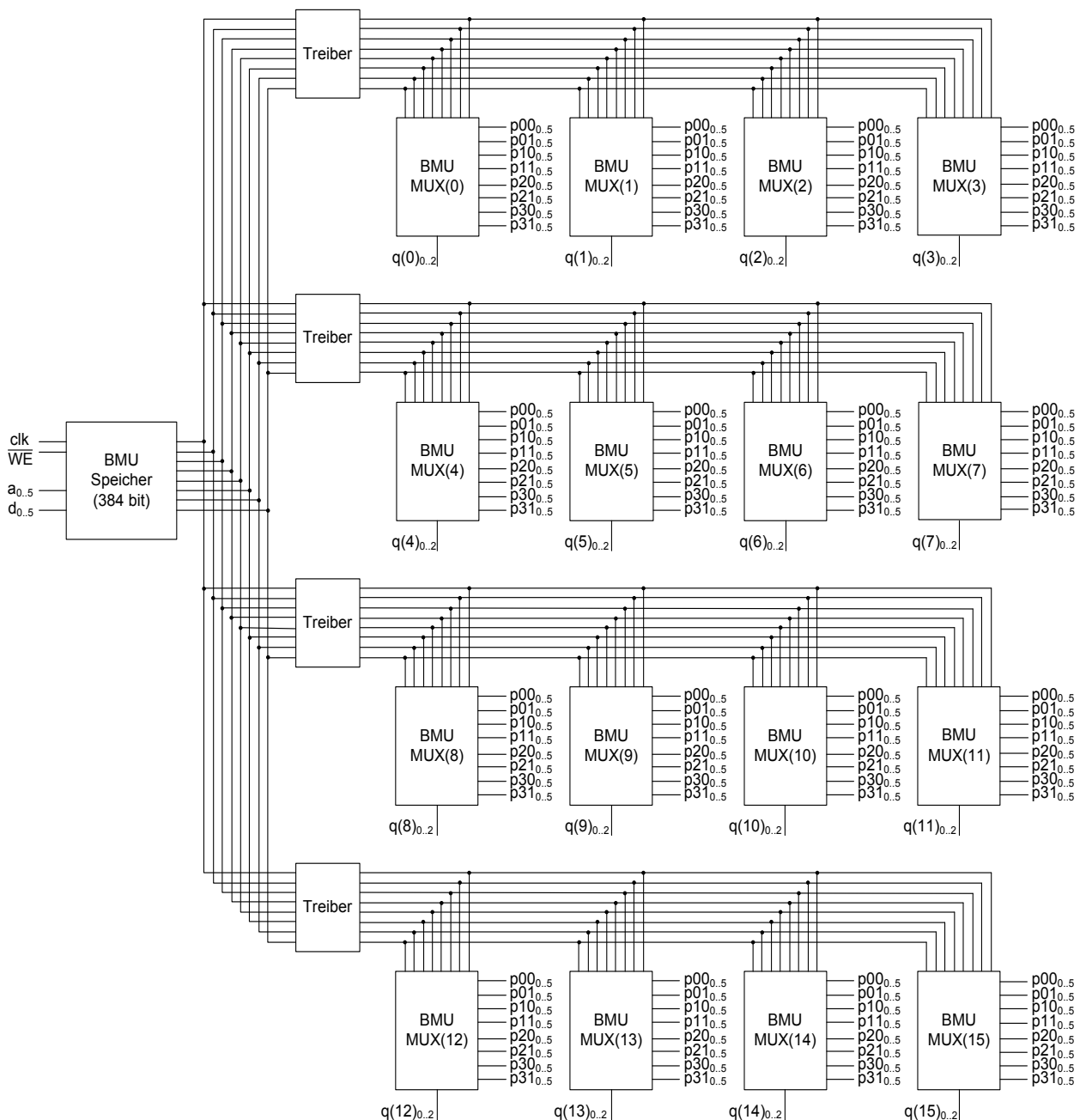


Abbildung 4.1: Architektur des Zweigmetrikrechners

4.1.4 Der zentrale Speicher

Abbildung 4.2 zeigt den Aufbau des zentralen Speichers für die Metriken. In der Mitte befinden sich acht SRAM-Blöcke, die von 00 bis 31 durchnummeriert sind. Jeder SRAM-Block speichert acht Metriken mit jeweils 6 bit Auflösung. Jede Metrik entspricht den Kosten eines bestimmten Zustandsübergang im Trellisdiagramm. Der entsprechende Übergang ergibt sich aus einer von

acht möglichen Quantisierungsstufen des Eingangssignals. Die erste Ziffer im Block SRAM gibt die Zustandsnummer des Zielzustands an, die zweite Ziffer adressiert den Zustandsübergang. Tabelle 4.1 enthält eine Übersicht, in welchem SRAM-Block, welche Metriken gespeichert sind.

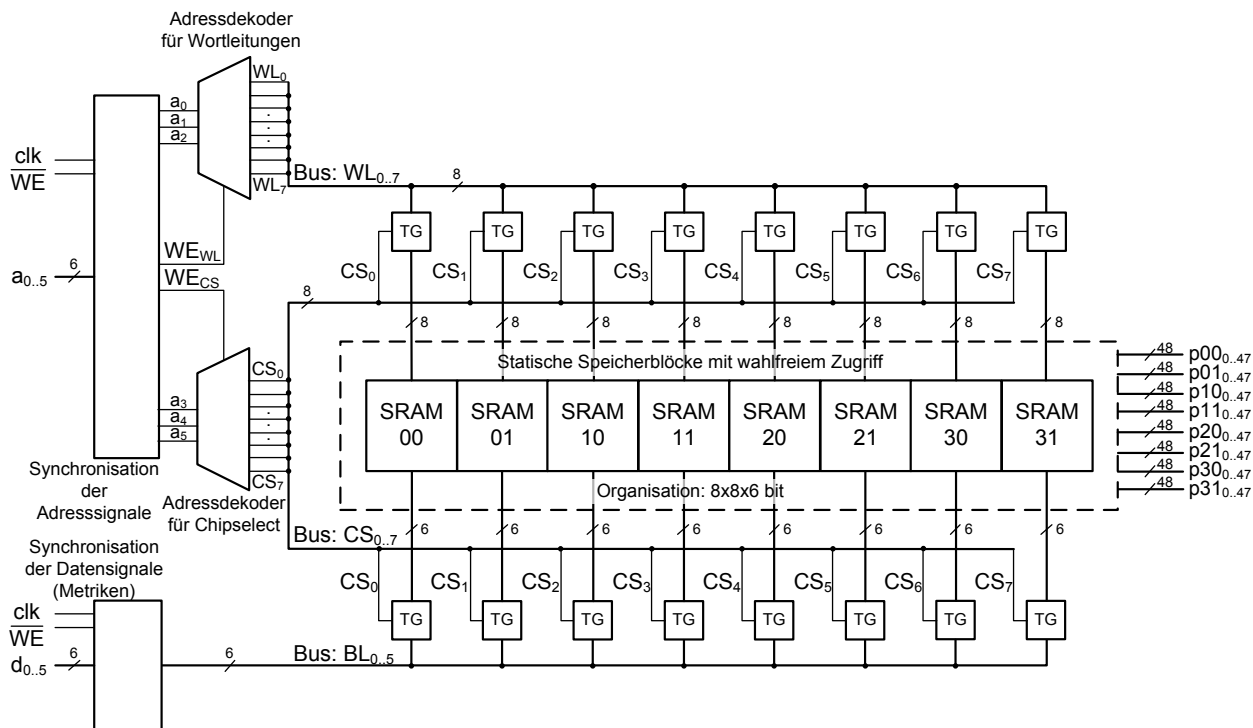


Abbildung 4.2: Gesamtübersicht über den zentralen Speicher

Tabelle 4.1: In welchem SRAM-Block stehen die Metriken für welchen Zustandsübergang

Zielzustand	Übergang 0	Übergang 1
0	00	01
1	10	11
2	20	21
3	30	31

Die Ansteuerung der SRAM-Blöcke erfolgt über sogenannte Chipselect-Leitungen CS_X mit $X \in \{0...7\}$. Eine aktive Chipselect-Leitung schaltet über die Transferrgatter TG der oberen Reihe die Wortleitungen WL_X und über die Transferrgatter der unteren Reihe die Bitleitungen BL_Y $Y \in \{0...5\}$ auf den gewünschten SRAM-Block durch. Dadurch wird sichergestellt, dass die parallel anliegenden Wort- und Bitleitungen nur an einem SRAM-Block einen Schreibvorgang auslösen.

Abbildung 4.3 zeigt die interne Organisation eines SRAM-Blocks. Ein SRAM-Block enthält sechs 8 bit SRAM-Zellblöcke. Daraus folgt, dass eine 6 bit Metrik über sechs 8 bit SRAM-Zellblöcke verteilt gespeichert ist. Durch diese Organisation kann eine einzelne Wortleitung WL_X mit $X \in \{0...7\}$ in jedem der sechs 8 bit SRAM-Zellblöcke eine Speicherzelle zum Beschreiben aktivieren, wodurch eine 6 bit Metrik gespeichert werden kann.

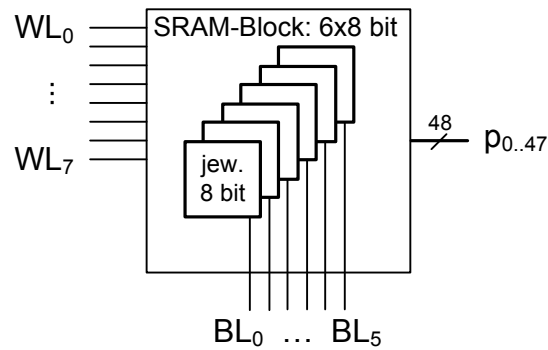


Abbildung 4.3: Architektur eines SRAM-Blocks

Der genaue Aufbau der 1 bit SRAM-Zelle innerhalb eines 8 bit SRAM-Zellblocks ist in Abbildung 4.4 dargestellt.

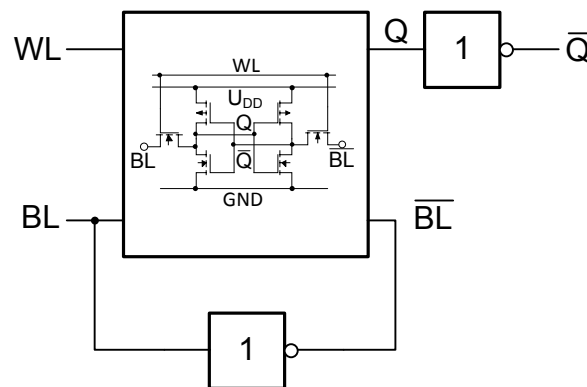


Abbildung 4.4: Äußere Beschaltung einer 1 bit SRAM-Zelle

Eine typische SRAM-Zelle enthält immer eine Bitleitung BL und eine inverse Bitleitung \overline{BL} (vgl. Abbildung 3.13). Da die inverse Bitleitung in der äußeren Beschaltung des zentralen Speichers nicht benötigt wird, sondern nur den Verdrahtungsaufwand verdoppeln würde, wird sie lokal über einen statischen CMOS-Inverter aus der nicht-invertierten Bitleitung erzeugt. Das Auslesen der SRAM-Zelle erfolgt nicht über die Transfertransistoren, die über die Wortleitungen aktiviert werden, sondern es wird der Inhalt der Speicherzelle direkt an den kreuzgekoppelten Invertern abgegriffen. Der interne Zustand Q wird über einen Treiberinverter nach außen geführt. Ein weiterer Treiberinverter, der zur Verteilung der Metriken notwendig ist, erzeugt dann aus dem inversen Signal \overline{Q} wieder das ursprüngliche Signal Q .

Die obere Reihe der Transferrgatter (TG) in Abbildung 4.2 dient zum Setzen der Wortleitungen, was gleichzeitig einem Öffnen der Speicherzelle zum Beschreiben dient. Wenn nach Abschluss des Schreibvorgangs das Transferrgatter wieder geschlossen wird, bleibt die Ladung am Knoten Z zunächst erhalten, da der Knoten Z dann isoliert ist. Der Wortleitungstransistor bliebe geöffnet. Ein weiteres Problem können auch die undefinierten Pegel darstellen, wenn der Knoten S eine fallende Flanke aufweist. Da die Wortleitungen aber nur solange aktiv sein müssen, wie der Schreibvorgang dauert, wird das Transferrgatter aus Abbildung 3.8 um einen Transistor erweitert [49], der den Knoten Z bei ausgeschaltetem Transferrgatter auf Massepotenzial zieht. Dieses erweiterte Transferrgatter ist in Abbildung 4.5 dargestellt.

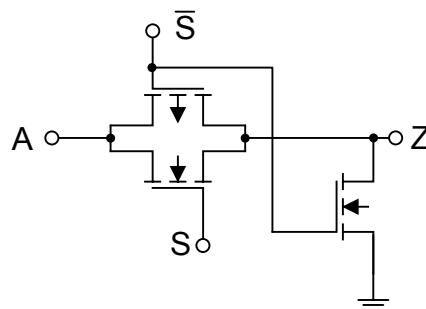


Abbildung 4.5: Transferrgatter für die Wortleitungen

Abbildung 4.6 zeigt den Spannungsverlauf am Knoten Z in Abhängigkeit der Eingangsspannung am Knoten A und der Steuerspannung am Knoten S.

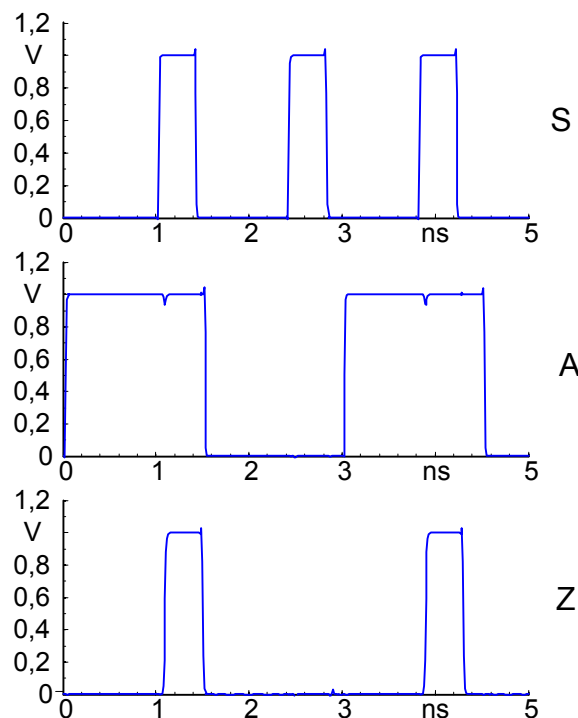


Abbildung 4.6: Simulation des Transferrgatters für die Wortleitungen

Es ist deutlich zu erkennen, dass der Knoten Z immer nur dann das Potenzial des Knotens A annimmt, wenn S einen High-Pegel aufweist. Sobald S auf low schaltet, wird der Knoten Z ebenfalls low, unabhängig davon ob der Knoten A zu dem Zeitpunkt auf high war.

Die untere Transfergatterreihe (TG) in Abbildung 4.2 dient der Ansteuerung der Bitleitungen. Diese Transfergatter können, wie in Abbildung 3.8 gezeigt, herkömmlich aufgebaut werden, da die Einschreibelogik sicherstellt, dass die Bitleitungen einen definierten Pegel aufweisen, bevor durch die Wortleitungen ein Schreibvorgang ausgelöst wird. Weiterhin werden die Wortleitungen wieder deaktiviert, bevor die Transfergatter der Bitleitungen schließen.

4.1.5 Adressdekoder

Wie bereits erwähnt, sind die Eingangsdaten mit 3 bit quantisiert, woraus folgt, dass es für jeden Zustandsübergang acht Wahrscheinlichkeitswerte gibt. Daher müssen acht Metriken pro Zustandsübergang in der BMU gespeichert werden. Da es in dem Trellisdiagramm acht Zustandsübergänge gibt, sind also insgesamt 64 Metriken zu speichern. Daher werden im zentralen Speicherfeld zwei 3 bit Adressdekoder eingesetzt. Der „Adressdekoder für Wortleitungen“ (siehe Abbildung 4.2) adressiert die Zeilen der SRAM-Blöcke zur Speicherung der acht Metriken pro Zustandsübergang. Der „Adressdekoder für Chipselect“ (siehe Abbildung 4.2) adressiert einen der acht SRAM-Blöcke. So kann man mit sechs Adressleitungen den kompletten Speicher adressieren. In Abbildung 4.7 ist ein Adressdekoder auf NAND-Gatterbasis dargestellt, welchen man auch als Invers-Adressdekoder bezeichnen kann. Als Grundbaustein dient ein NAND-Gatter mit drei Eingängen, die in Abhängigkeit der Schaltfunktion mit Invertern beschaltet sind. Die Inverter sind so angebracht, dass immer nur einer der acht Ausgänge eine logische 0 ausgibt, während alle anderen Ausgänge eine logische 1 ausgeben.

Um den Adressdekoder als Zeilen- bzw. Spaltendekoder einsetzen zu können, werden die Ausgänge um Multiplexer, wie in Abbildung 4.8 dargestellt, erweitert. Mit Hilfe der Multiplexer lassen sich die Ausgänge des Adressdekoders auch explizit abschalten, was insbesondere dazu dient, die Chipselect-Leitungen, welche die Transfergatter ansteuern, auszuschalten, wenn gerade kein Schreibvorgang erfolgen soll. Die Ausgangsmultiplexer sind mit Invertern beschaltet, die als Treiber benötigt werden, um die Wort- und Chipselect-Leitungen über das ganze Speicherfeld zu verteilen. Dies ist ein Grund für den Einsatz des Invers-Adressdekoders auf NAND-Gatterbasis. Mit NOR-Gattern könnte man die Adressdekoderlogik zwar direkt implementieren, dies hätte aber nur Nachteile:

1. NOR-Gatter sind langsamer als NAND-Gatter
2. NOR-Gatter benötigen mehr Fläche als NAND-Gatter
3. Da man am Ausgang sowieso Treiberinverter benötigt, müsste man das Ausgangssignal zweimal invertieren, was zu einem noch größeren Zeitverlust, höheren Flächen- und Strombedarf führt.

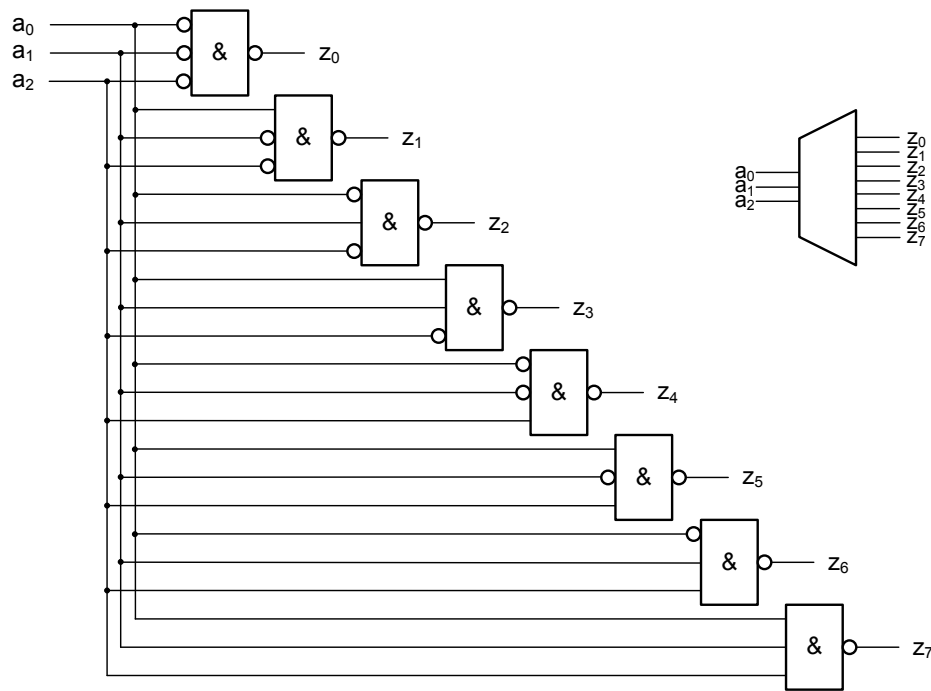


Abbildung 4.7: Adresdecoder auf NAND-Gatterbasis

Abbildung 4.8 zeigt den Zeilen- und Spaltendekoder für die Wort- und Chipselect-Leitungen, der den Adresdecoder und zwei 4 bit Multiplexer enthält. Über den Eingang S können die Ausgänge $out_{0..7}$ ein- oder ausgeschaltet werden.

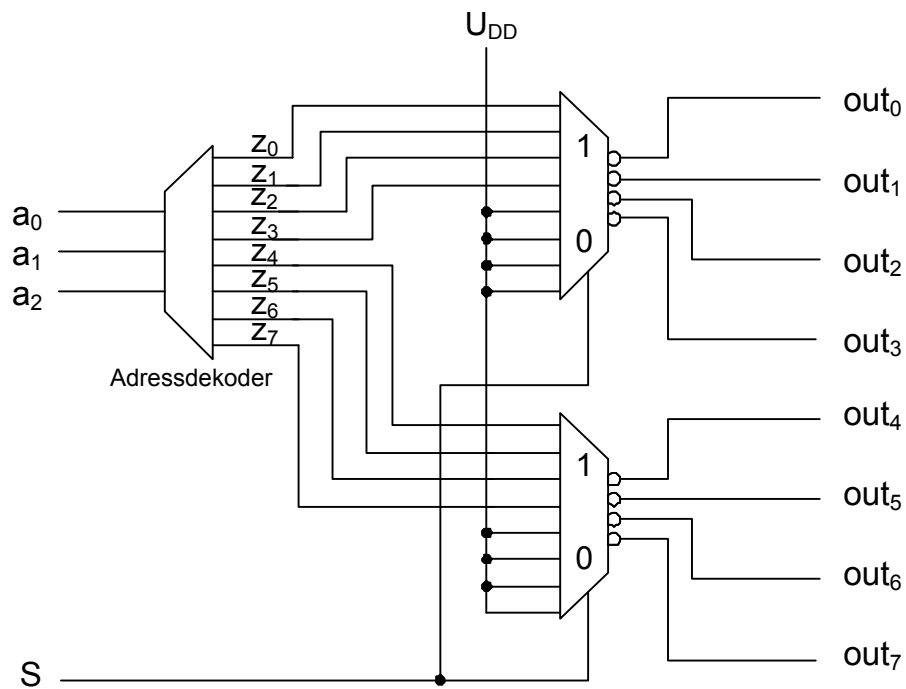


Abbildung 4.8: Zeilen- und Spaltendekoder für Wort- und Chipselect-Leitungen

4.1.6 Synchronisation der Metriken beim Einschreiben

Der Systemtakt des Viterbi-Entzerrers ist von den Eingangsdaten abhängig. Die Metriken werden unabhängig davon extern auf einem FPGA berechnet. Da diese beiden Systeme völlig unabhängig voneinander arbeiten, müssen die Metriken asynchron zum internen Takt des Viterbi-Rechenwerks eingeschrieben werden können. Außerdem gibt es Stand heute noch keinen FPGA, der im Gigahertzbereich Daten verarbeiten kann. Daher muss eine Synchronisationsschaltung entworfen werden, die die externen Metrikdaten auf den internen Systemtakt aufsynchronisiert und dann in das Speicherfeld innerhalb eines Taktzyklus des Viterbi-Rechenwerks einschreibt. Die Freigabe der externen Metrikdaten erfolgt über ein Schreibfreigabesignal, das der FPGA bereitstellt, sobald ein Metrikdatum (6 bit) und ein Adressdatum (ebenfalls 6 bit) vom FPGA berechnet wurden. In Abbildung 4.9 ist gezeigt, wie mit Hilfe von zwei Flipflops und einem NAND-Gatter das externe Schreibfreigabesignal \overline{WE} auf den internen Viterbi-Systemtakt clk aufsynchronisiert werden kann [49].

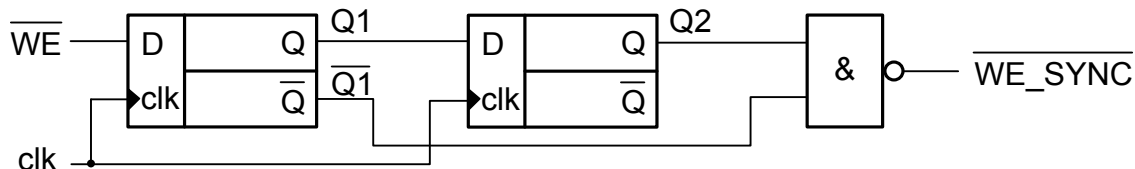


Abbildung 4.9: Synchronisationslogik für die Schreibfreigabe

Abbildung 4.10 zeigt an Hand von drei Eingangssignalkonstellationen, wie sich die Synchronisationslogik verhält.

Der erste \overline{WE} Impuls ist länger als eine Taktperiode. Sobald die steigende Taktflanke kommt, wird das Signal \overline{WE} vom ersten Flipflop an Q1 übernommen. Um eine Taktperiode verzögert liegt es dann an Q2 an. Durch die inverse UND-Verknüpfung aus $\overline{Q1}$ und Q2 wird sichergestellt, dass das Signal \overline{WE} auf eine Länge einer Taktperiode begrenzt wird. Das Resultat zeigt sich im Verlauf von $\overline{WE_SYNC}$. Der zweite \overline{WE} Impuls ist kürzer als eine Taktperiode. Da er von einer positiven Taktflanke erfasst wird, wird er auf eine Taktperiode verlängert. Der dritte \overline{WE} Impuls ist ebenfalls kürzer als eine Taktperiode, wird aber von keiner positiven Taktflanke erfasst. Deshalb wird er nicht nach $\overline{WE_SYNC}$ weitergegeben. Da ein kürzerer \overline{WE} Impuls als eine Taktperiode in der Realität nicht vorkommt, bedeutet jenes Schaltverhalten keine Fehlfunktion. Aus dem auf eine Taktperiode verkürzten Schreibfreigabesignal $\overline{WE_SYNC}$ können nun die Steuersignale WE_CS und WE_WL , die den eigentlichen Schreibvorgang einleiten, generiert werden.

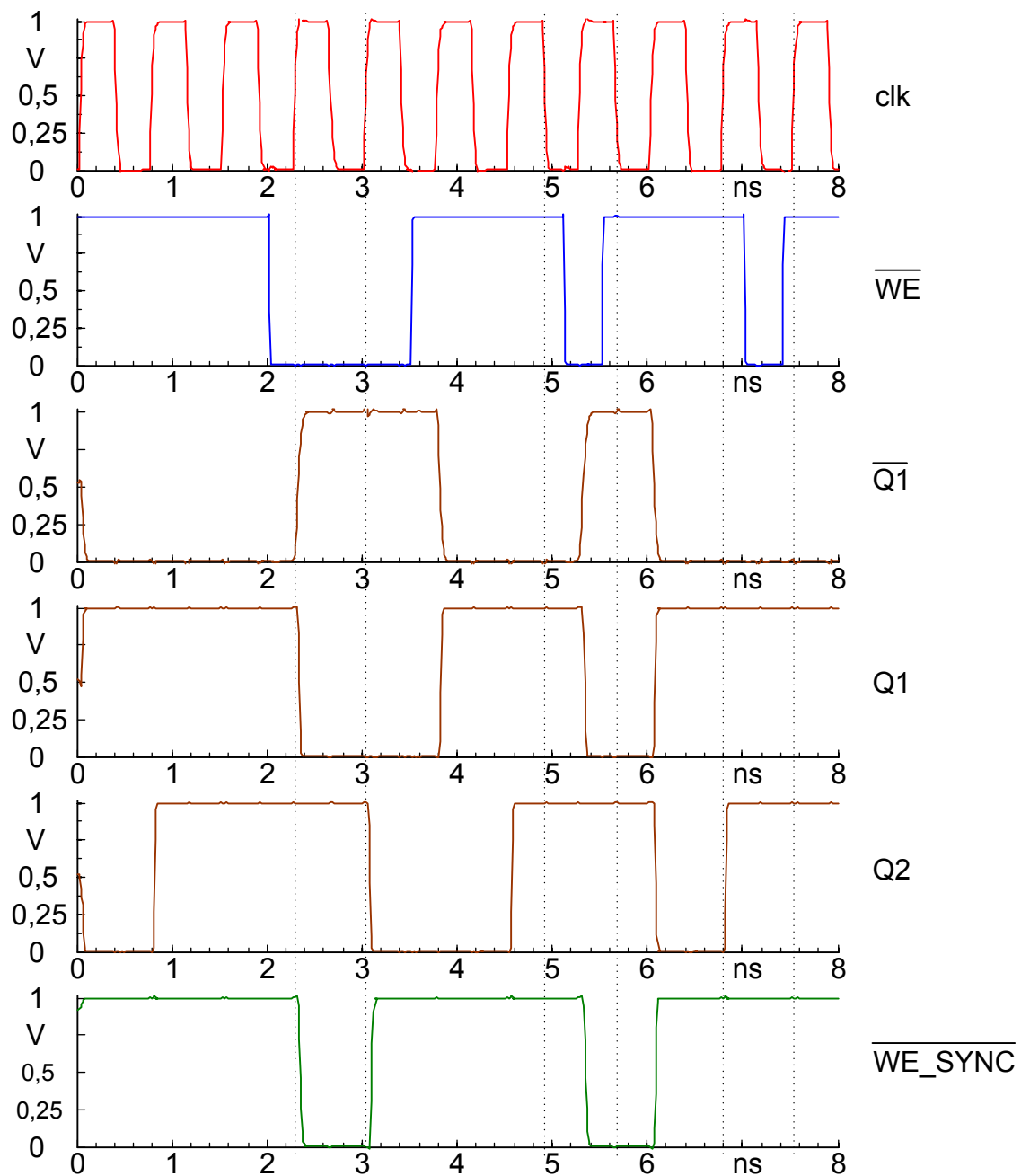


Abbildung 4.10: Simulation der Synchronisationslogik für die Schreibfreigabe

Abbildung 4.11 zeigt die entsprechende Schaltung, die die Schaltung aus Abbildung 4.9 enthält. Über ein verneintes ODER-Gatter wird das Taktsignal mit dem auf eine Taktperiode begrenzten Schreibfreigabesignal $\overline{\text{WE_SYNC}}$ verknüpft [49]. Daraus entsteht das Schreibfreigabesignal WE_WL , welches eine Länge einer halben Taktperiode aufweist. Parallel dazu wird das $\overline{\text{WE_SYNC}}$ Signal mit Hilfe von neun Invertern so verzögert, damit das resultierende Signal WE_CS in der Mitte von WE_WL liegt. Diese Verschiebung entspricht ungefähr der Dauer einer viertel Taktperiode. Abbildung 4.12 verdeutlicht dieses Schaltverhalten.

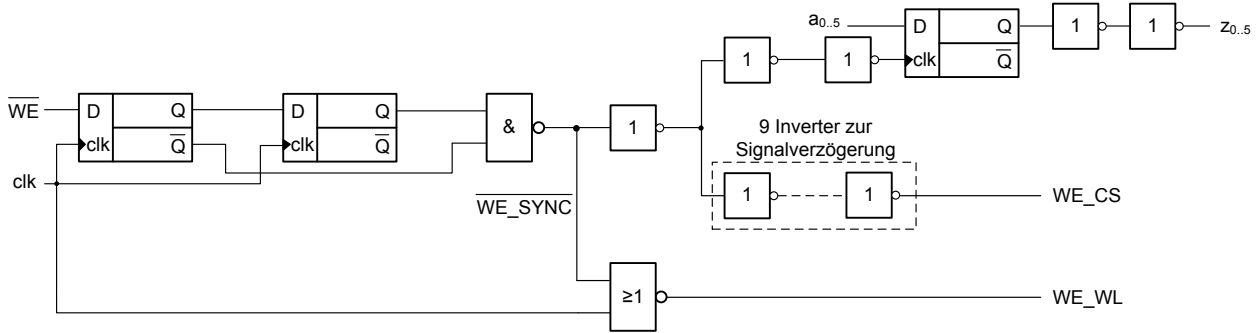


Abbildung 4.11: Synchronisation der Adresssignale

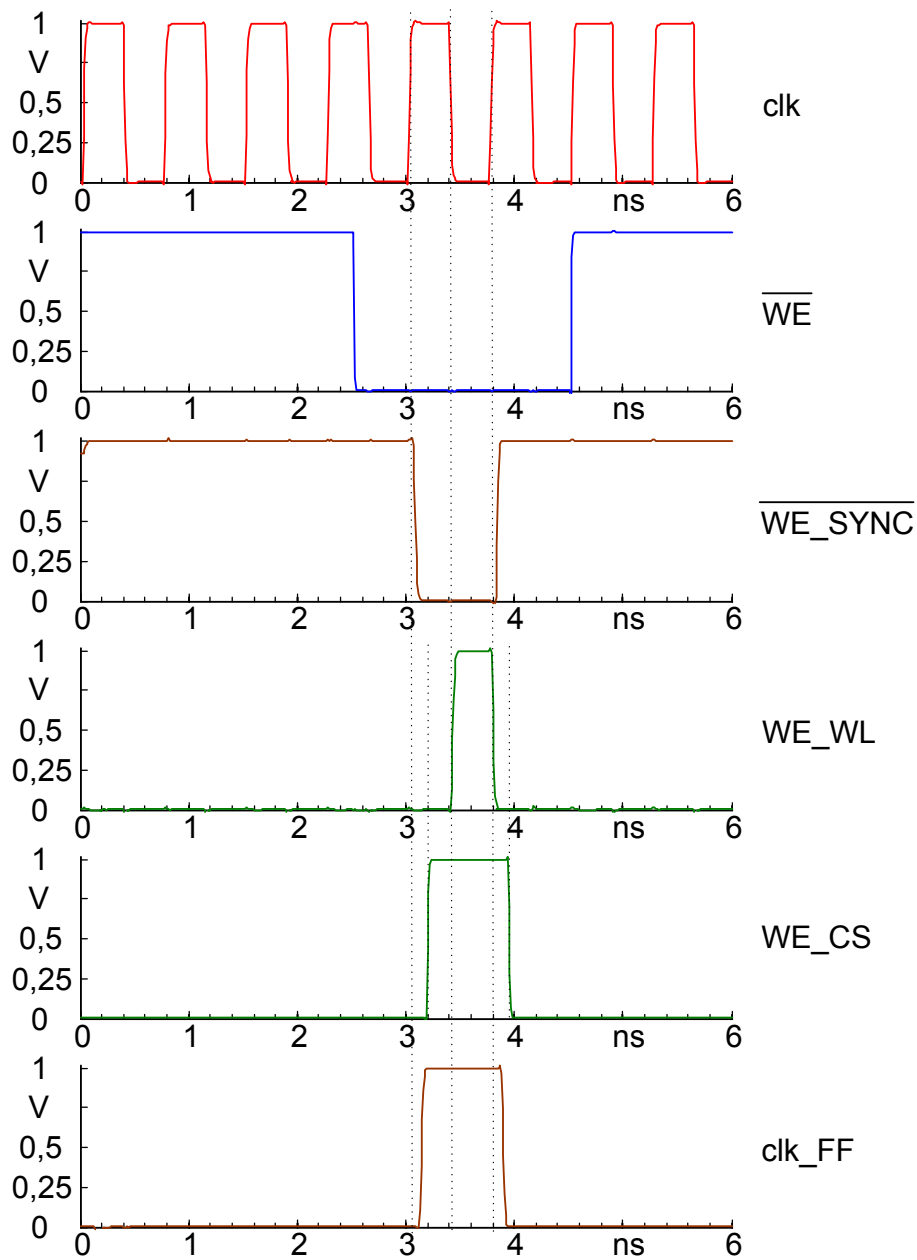


Abbildung 4.12: Simulation der Synchronisationslogik für Wort- und Chipselect-Leitungen

WE_CS aktiviert den „Adressdeko­der für Chipselect“ (siehe Abbildung 4.2) in Abhängigkeit der Teiladressbits $a_{3..5}$. Der Adressdeko­der schaltet dann die Transf­ergatter für Bit- und Wortleitungen eines bestimmten SRAM-Blocks frei. Mit rund einer viertel Taktperiode Verzögerung gibt WE_WL den „Adressdeko­der für Wortleitungen“ frei. Dieser Adressdeko­der aktiviert eine Wortleitung einer Zeile in einem SRAM-Block in Abhängigkeit der Teiladressbits $a_{0..2}$, wodurch das Einschreiben einer Metrik ausgelöst wird. Bevor nun die Bitleitungen wieder einen hochohmigen Zustand einnehmen, wird das Wortleitungssignal zurückgesetzt. Dadurch wird ein sauberer Einschreibvorgang gewährleistet. Die Adresssignale werden über Flipflops zwischengepuffert und synchron zu $\overline{\text{WE_SYNC}}$ auf die beiden Adressdeko­der gegeben.

Der Schaltungsteil „Synchronisation der Datensignale (Metriken)“ aus Abbildung 4.2 ist im Wesentlichen gleich aufgebaut, wie der Schaltungsteil „Synchronisation der Adresssignale“. Der Unterschied besteht darin, dass für die Bitleitungen keine weiteren Steuersignale benötigt werden. Somit erübrigt sich die Erzeugung weiterer abgeleiteter Schreibfreigabesignale. Es findet nur eine Pufferung der Datensignale (Metriken) in Flipflops statt sowie die taktsynchrone Ansteuerung dieser Flipflops in Bezug auf das externe Schreibfreigabesignal $\overline{\text{WE}}$.

4.1.7 Verteilung des Speicherinhaltes und Übersprechen

Der komplette Speicherinhalt wird parallel über einzelne Leitungen den Stellen der Schaltung zugeführt, wo die Speicherinhalte benötigt werden. Somit entsteht keine Konkurrenzsituation zwischen Schreib- und Lesezugriff auf den Speicher, da die Bitleitungen der einzelnen Speicherzellen nur zum Schreiben und nicht zum Lesen der eingespeicherten Information benötigt werden. Ein weiterer Vorteil dieser Architektur besteht in der Notwendigkeit, dass ein gleichzeitiger Lesezugriff auf alle Speicherzellen erfolgen kann. Die fettgedruckten Leitungen in Abbildung 4.13 stellen den 384 bit breiten Bus zur Verteilung der Metriken dar. Die einzelnen Leitungen greifen das Signal direkt intern an den kreuzgekoppelten Invertern der Speicherzelle ab.

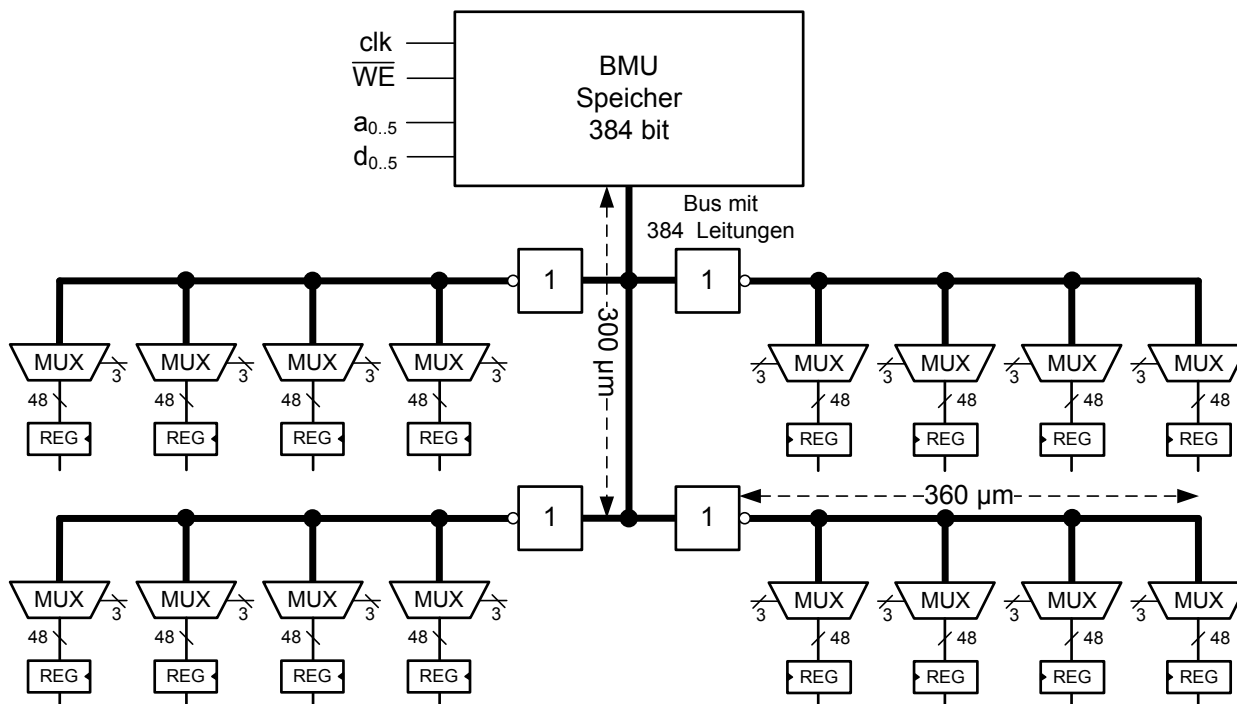


Abbildung 4.13: Anordnung/Verteilung der Metriken

Um Platz zu sparen, werden die Leitungen mit minimaler Breite in minimalem Abstand angeordnet. Außerdem werden für jede Richtung zwei Metalllagen verwendet, um eine Flächenhalbierung zu erreichen. Pro Metalllage sind dann 192 Leitungen zu verlegen. Sowohl die Leitungsbreite als auch der Leitungsabstand betragen 140 nm. Die Gesamtleitungslänge beträgt 660 µm. Aufgrund dieser relativ zum Leitungsabstand langen parallelen Leitungsführung muss das Übersprechen zwischen Leitungen betrachtet werden.

Beim Übersprechen unterscheidet man zwei Arten von Störungen:

1. Funktionelles Übersprechen (engl. Functional Noise)
2. Verzögerungsübersprechen (engl. Delay Noise)

Beim funktionellen Übersprechen wird eine passive Leitung (engl. Victim) durch ihre umliegenden Aggressorleitungen so stark beeinflusst, dass auf der passiven Leitung eine Störung auftritt, die zu einer Fehlinterpretation des Signalpegels führt. Es entsteht ein Störspannungsimpuls (engl. Glitch), welcher von den Koppelkapazitäten C_C zu den benachbarten Leitungen und gegenüber Masseleitungen C_G abhängt. Die Höhe des Störimpulses U_{glitch} in Abhängigkeit der Spannungsänderung U_a auf der Aggressorleitung berechnet sich zu [50]:

$$U_{\text{glitch}} = U_a \frac{C_C}{C_C + C_G} \quad (4.2)$$

Unter Verzögerungsübersprechen versteht man eine Verschiebung des Schaltzeitpunkts des Signals, das durch gleichzeitiges, entgegengesetztes Umschalten von Aggressorleitungen auftritt. Das bedeutet, dass die verursachenden Leitungen gleichzeitig auch eine Wirkung erfahren.

Daraus folgt, dass auch das Einfügen von Masselagen zu einer Verlangsamung der Signalausbreitung auf den Aggressorleitungen führt. Die Auswirkung des Verzögerungsübersprechens wird im Rahmen der Analyse des Übersprechens nicht untersucht, da das Zeitverhalten des Einschreibvorgangs später noch allgemein behandelt wird.

Abbildung 4.14 zeigt einen Layout-Querschnitt von Leitungen, mit Hilfe dessen das funktionelle Übersprechen näher betrachtet werden soll. Alle Leitungen bis auf die passive Leitung in der Mitte der oberen Reihe sind Aggressoren und vollziehen einen Signalwechsel sowohl von high nach low als auch von low nach high. Das Potenzial der passiven Leitung wird konstant entweder auf Versorgungsspannungs- oder auf Massepotenzial gehalten.

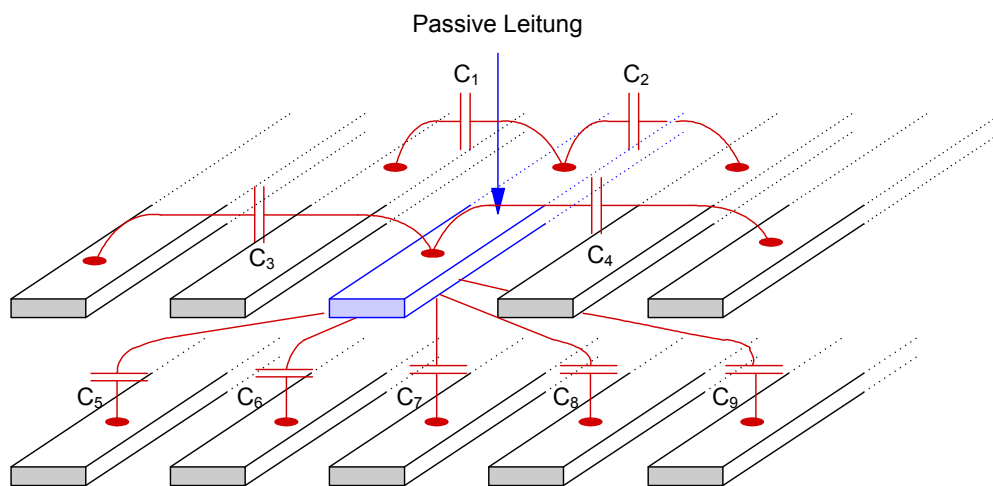


Abbildung 4.14: Aufbau der Simulationsumgebung für das Übersprechen

Der Anordnung aus Abbildung 4.13 und Abbildung 4.14 liegt eine Leitungslänge von $660\ \mu\text{m}$ zugrunde. Die Leitungsbreite sowie der horizontale Leitungsabstand betragen $140\ \text{nm}$. Dies sind die minimalen Strukturgrößen für Leitungen der verwendeten Metallisierungslagen des $90\ \text{nm}$ CMOS-Halbleiterprozesses [51]. Der vertikale Leitungsabstand ist durch die Oxiddicke des Halbleiterprozesses festgelegt.

Abbildung 4.15 zeigt das Simulationsergebnis des Übersprechens des Aufbaus nach Abbildung 4.14. Im linken Bild wird die passive Leitung auf Massepotenzial gehalten. Im rechten Bild wird der Fall betrachtet, dass die passive Leitung auf die positive Versorgungsspannung von $1\ \text{V}$ vorgespannt wird. Die Auswirkungen des Aggressors sind erheblich. Insbesondere im rechten Bild kommt man dem Schwellwert der Logik ($0,5\ \text{V}$) relativ nahe. Nach einem Regenerationsinverter am Ende der passiven Leitung sind jedoch kaum mehr Auswirkungen des Übersprechens zu erkennen.

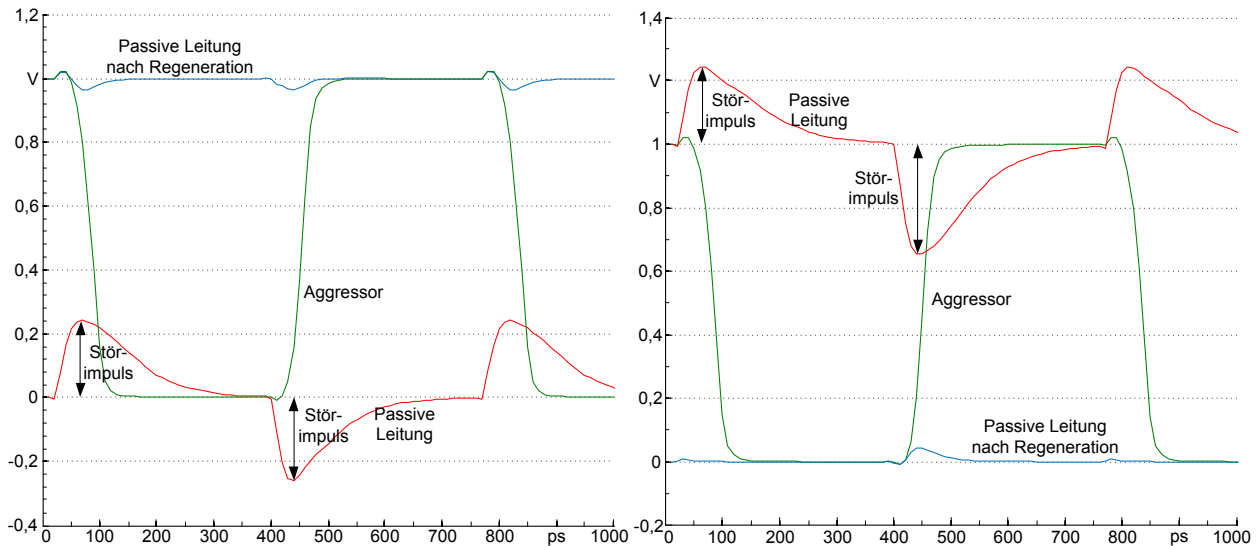


Abbildung 4.15: Simulation des Übersprechens auf den Leitungen zur Verteilung der Metriken

Es werden zwei Extraktionsmethoden angewandt. Einerseits wird mit typisch zu erwartenden Widerständen und Kapazitäten extrahiert (RCTYP), andererseits mit maximalen parasitären Elementen (RCMAX). Die genaue Auswertung der Simulationsdaten ist in Tabelle 4.2 dargestellt. Die Unterschiede sind jedoch gering.

Tabelle 4.2: Simulationsdaten zum Übersprechen

Glitch / Extraktion	Auswirkung auf GND	Auswirkung auf U_{DD}
$\Delta U_{\text{glitch}} @ \text{RCTYP}$	237 mV ... 255 mV	246 mV ... 344 mV
$\Delta U_{\text{glitch}} @ \text{RCMAX}$	242 mV ... 262 mV	248 mV ... 348 mV

Trotz des hohen Störspannungsimpulses von bis zu 35 % der Versorgungsspannung, kommt es zu keiner Fehlinterpretation der Signale auf Grund von funktionellem Übersprechen. Nach einem Regenerationsinverter am Ende der passiven Leitung beträgt die maximale Abweichung vom Sollpegel weniger als 50 mV. Dennoch werden Vorkehrungen getroffen, die Auswirkung des funktionellen Übersprechens so gering wie möglich zu halten. Aufgrund der langen Leitungen und der daraus resultierenden Dämpfung werden auf etwa der Hälfte der Gesamtleitungslänge Treiberinverter eingefügt, um das Signal, insbesondere die Flankensteilheit, wiederherzustellen.

Im ungünstigsten Fall könnte sich in jedem Takt eine der 64 Metriken ändern. Die anderen Metriken bleiben konstant. Um nun die Wirkung des Übersprechens zu reduzieren, werden die Leitungen, die zu einer Metrik gehören, nicht nebeneinander verlegt, sondern im Abstand von acht Leitungen, welcher dann 2,1 μm beträgt. Somit wirken die sieben Leitungen zwischen zwei zusammengehörenden Metrikleitungen als Abschirmung, da sie sich während eines Schaltvorganges auf konstantem Potenzial befinden. Durch diese intelligente Verdrahtung kann

der Effekt des funktionellen Übersprechens deutlich reduziert werden. Um Fläche zu sparen, werden die Leitungen auf zwei unterschiedlichen Metallisierungsebenen verlegt. Der Bus mit den 384 Leitungen hat eine Breite von 56 μm .

Am Ende des langen Busses befinden sich Multiplexer, um aus den 64 Metriken, die acht Metriken auszuwählen, die zu dem eingehenden, zu entzerrenden Symbol korrespondieren. Register schließen den langen Pfad ab.

4.1.8 Auslesen des Speichers

Um den zentralen Speicher auszulesen, werden Multiplexer verwendet. Am Eingang der Ausleseeinheit liegen alle 64 Metriken mit 6 bit quantisiert an. Die Eingangssymbole, die zur Auswahl der Metriken herangezogen werden, sind mit 3 bit quantisiert, woraus sich acht adressierbare Amplitudenstufen ergeben. Zu jeder Amplitudenstufe gibt es aufgrund des Kanalmodells genau acht Zustandsübergänge. Die Dekodierung der entsprechenden Amplitudenstufe erfolgt über einen Adressdekoder. Die Leitungen der Amplitudenstufen werden dann auf acht Multiplexerfelder verteilt, um die Metriken für alle acht Zustandsübergänge zu adressieren. Ein Multiplexerfeld enthält sechs 8:1 Multiplexer, woraus dann genau die Metrik einer bestimmten Amplitudenstufe eines bestimmten Zustandsübergangs extrahiert wird. Abbildung 4.16 zeigt die acht Multiplexerfelder und den Adressdekoder.

Die Eingänge $p_{\mu\nu,0..47}$ repräsentieren die 64 Metriken und sind gleichbedeutend mit den Ausgängen des zentralen Speichers aus Abbildung 4.2. Die Kodierung für $\mu\nu$ steht für einen von acht Zustandsübergängen und kann Tabelle 4.1 entnommen werden. Die acht Amplitudenstufen sind in dem Vektor 0..47 zusammengefasst. Tabelle 4.3 gibt eine Übersicht, welcher 6 bit breite Teilvektor aus den 48 Leitungen einer bestimmten Quantisierungsstufe (Q.-Stufe) entspricht.

Tabelle 4.3: Zuordnung der Teilvektoren $p_{\mu\nu}$ zu den Quantisierungsstufen

Teilvektor	0..5	6..11	12..17	18..23	24..29	30..35	36..41	42..47
Q.-Stufe	1	2	3	4	5	6	7	8

Am Ausgang $p_{\mu\nu,0..5}$ wird dann für jeden Zustandsübergang $\mu\nu$ eine Metrik, die sich aus der kodierten Quantisierungsstufe des Eingangs $q_{0..2}$ ergibt, ausgegeben.

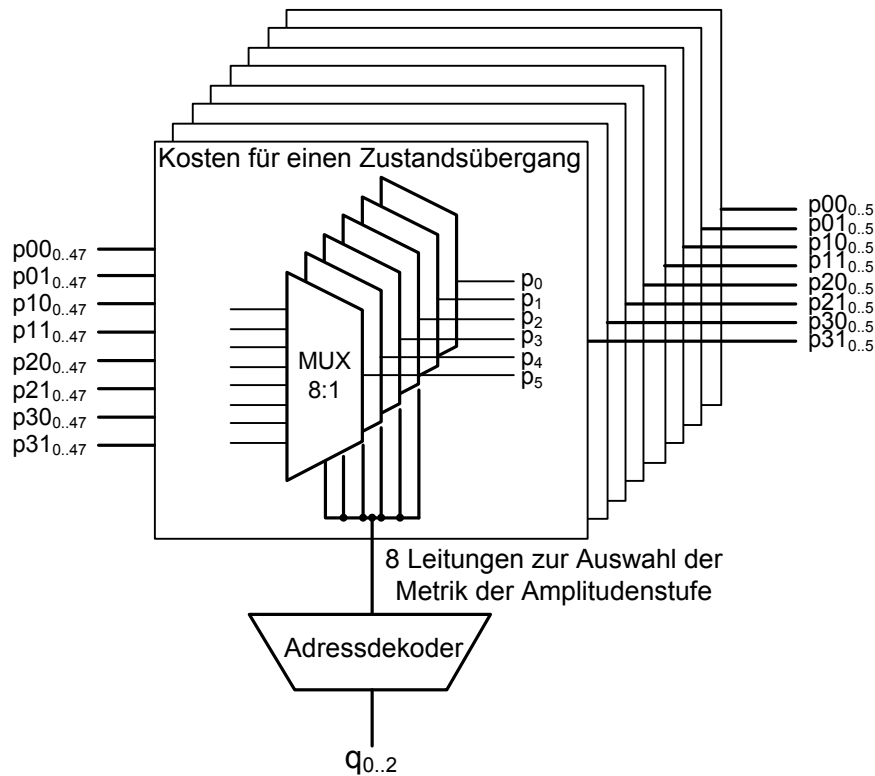


Abbildung 4.16: Ausgangsmultiplexerarchitektur zum Auslesen des zentralen Speichers

4.1.9 Einschreiben der Metriken

Abschließend wird überprüft, ob es möglich ist, innerhalb einer Taktperiode des Viterbi-Rechenwerks eine Metrik einzuschreiben, bevor sie an der Addier-Vergleichs-Auswahleinheit ausgelesen wird. Es muss sichergestellt sein, dass für eine korrekte Entzerrung immer definierte Signale anliegen. Eine komplette Simulation des Layouts der BMU ist auf Transistorebene nicht durchführbar. Deshalb wird ein Teillayout erstellt, das acht Speicherzellen, die längste Leitungslänge und den Ausgangsmultiplexer enthält. Details zum Aufbau sind in [42] und [49] zu finden. Abbildung 4.17 gibt einen Überblick über den Signallaufpfad vom Einschreiben bis zum Auslesen der Metrik in der gesamten BMU.

Abbildung 4.18 zeigt den Ablauf eines Einschreibvorganges. Zu Beginn wird das Schreibfreigabesignal WE für ungefähr eine Taktperiode (750 ps) auf high gesetzt, welches dann auf den internen Takt aufsynchronisiert wird. Daraufhin wird ab der fallenden Taktflanke die Bitleitung BL verzögert gesetzt. Nach Anliegen des Datums an der Bitleitung der Speicherzelle, wird die Wortleitung WL aktiviert, wodurch das Einspeichern ausgelöst wird. Der Signalwechsel auf BM_out zeigt an, dass das Datum der Bitleitung an dem Ausgangsflipflop angekommen ist. Die Datenübernahme an den Addier-Vergleichs-Auswahleinheiten erfolgt zur nächsten fallenden Flanke des Taktes $clk_FF_BM_out$. Der Takt $clk_FF_BM_out$ ist gegenüber dem Eingangstakt clk_BMU verschoben, da die Taktverteilung signalflussorientiert erfolgt.

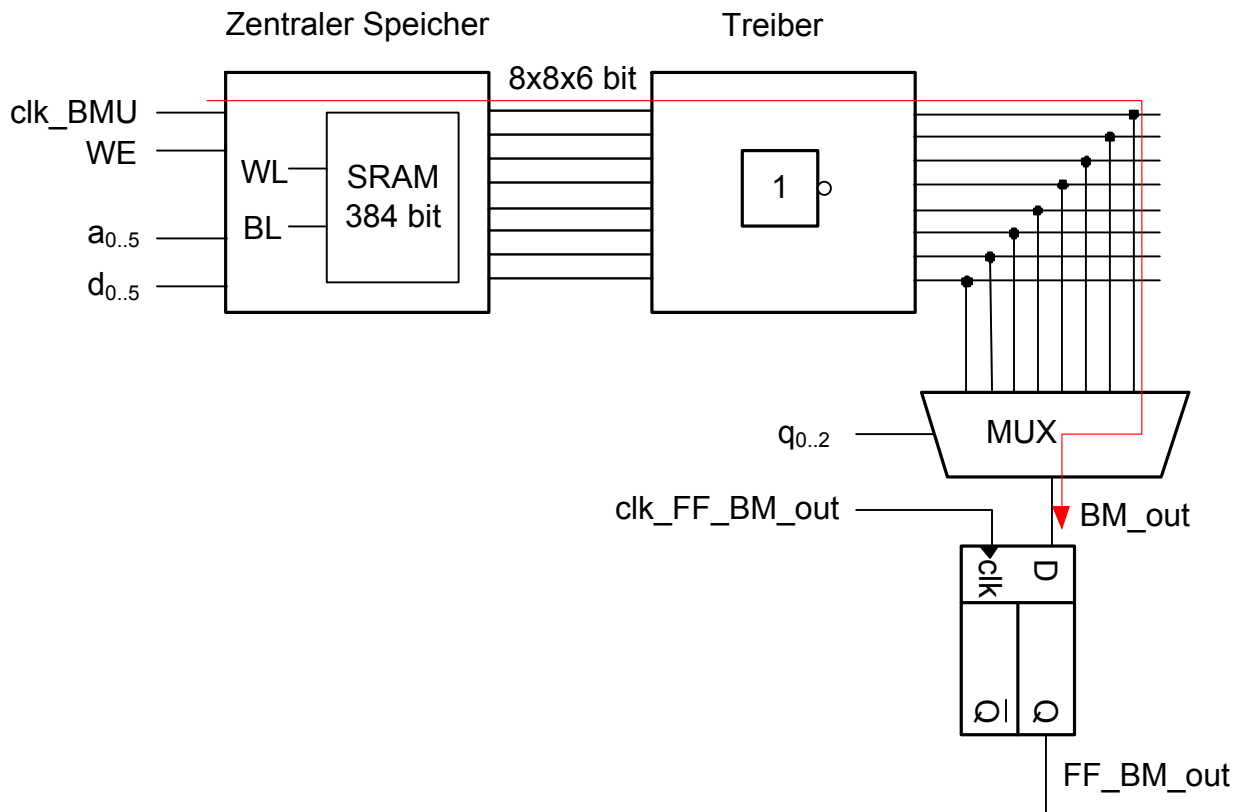


Abbildung 4.17: Schaltplan der zur Verdeutlichung des Einschreibvorgangs der Metriken

Zusammenfassend ist festzustellen, dass der gesamte Einschreibvorgang mit typischen parasitären Elementen nach Tabelle 4.4 im ungünstigsten Fall 480 ps dauert. Als Rüstzeit zur nächsten Taktflanke stehen noch 428 ps zur Verfügung. Damit ist gezeigt, dass der Einschreibvorgang innerhalb einer Taktperiode abgeschlossen wird.

Tabelle 4.4: Verzögerungszeiten zur Simulation aus Abbildung 4.18

	BL-Wechsel: H → L	BL-Wechsel: L → H
$t_{\text{clk}_{\text{BMU}}, \text{WL}}$	225 ps	223 ps
$t_{\text{BL}, \text{WL}}$	102 ps	113 ps
$t_{\text{WL}, \text{BM}_{\text{out}}}$	228 ps	257 ps
$t_{\text{BM}_{\text{out}}, \text{clk}_{\text{FF}_{\text{BM}_{\text{out}}}}}$	463 ps	428 ps
$t_{\text{Einschreibvorgang}}$	453 ps	480 ps

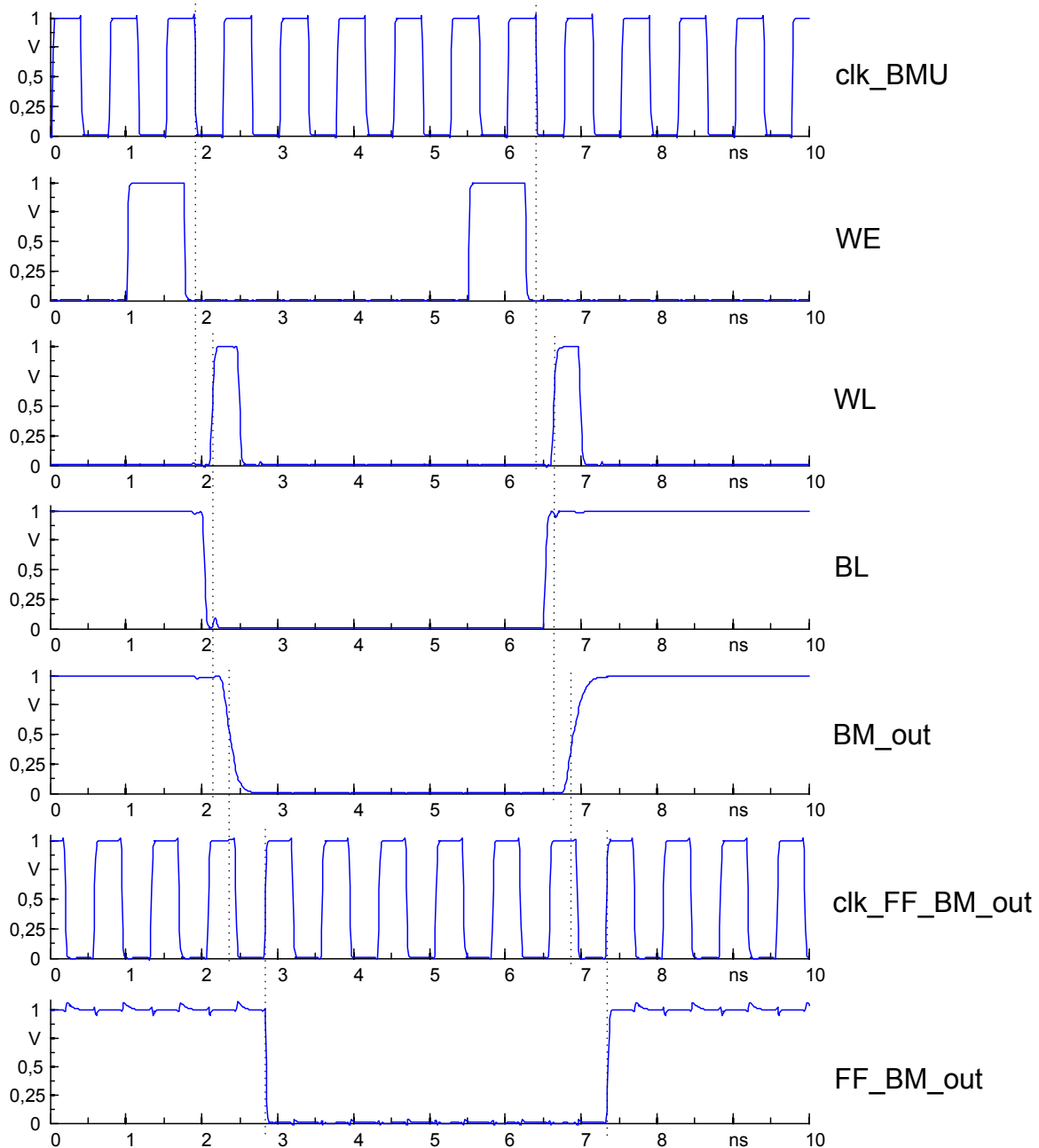


Abbildung 4.18: Simulation des Einschreibvorgangs der Metriken

4.1.10 Übersicht über die realisierte Zweigmetrikrecheneinheit

In Abbildung 4.19 ist das Layout der gesamten Zweigmetrikrecheneinheit, die aus einem Speicherfeld für 384 bit, 16 Auslesemultiplexern, Leitungstreibern und Busleitungen zur Verteilung der Zweigmetriken besteht, dargestellt. Das Layout orientiert sich an dem Blockschaltbild aus Abbildung 4.13.

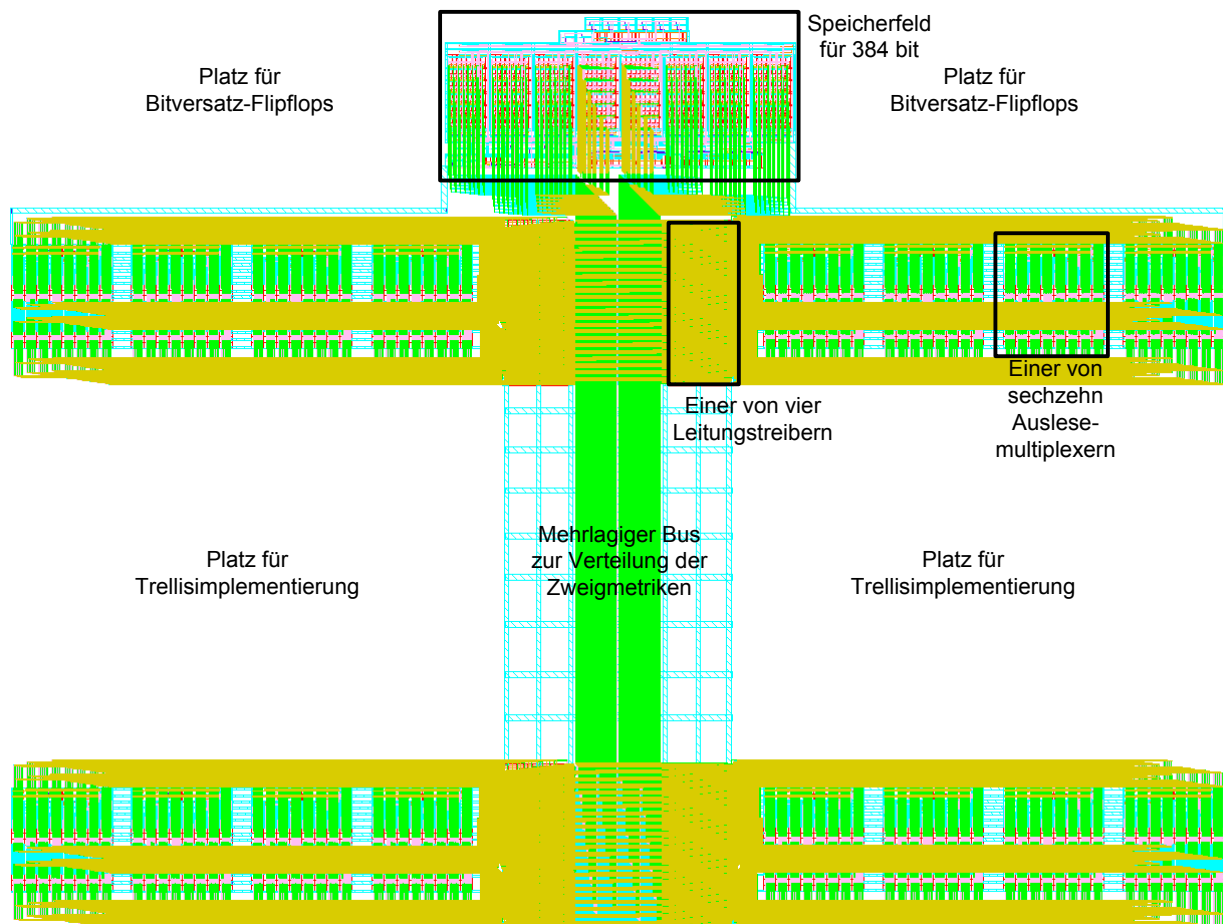


Abbildung 4.19: Layout der gesamten Zweigmetrikrecheneinheit

Die geometrischen Abmessungen sowie Angaben zum Schaltungsaufwand der Zweigmetrikrecheneinheit (BMU) sind in Tabelle 4.5 zusammengefasst.

Tabelle 4.5: Eckdaten der gesamten Zweigmetrikrecheneinheit

	Speicherfeld einschl. Einschreibelogik	Auslesemultiplexer	Gesamlayout der BMU
Breite	231 μm	66 μm	800 μm
Höhe	99 μm	69 μm	600 μm
Fläche	22.869 μm^2	4.554 μm^2	480.000 μm^2
Anzahl Transistoren	4.568	1.164	26.264

4.2 Addier-Vergleichs-Auswahleinheit

Die Berechnung des kürzesten Pfades erfolgt entlang des Trellis, wie er in Abbildung 4.20 dargestellt ist.

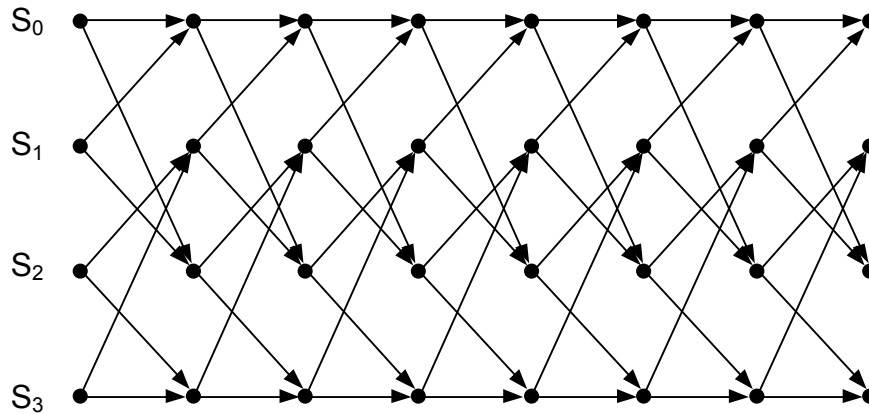


Abbildung 4.20: Trellis mit vier Zuständen

In diesem Fall besteht er auf Grund der Kanalmodellierung (vgl. Kapitel 2.3) aus vier Zuständen. Zu allen Zeitpunkten erfolgt in jedem Zustand eine Addier-Vergleichs-Auswahl-Operation (engl. Add-Compare-Select, ACS) wie sie Abbildung 4.21 dargestellt ist.

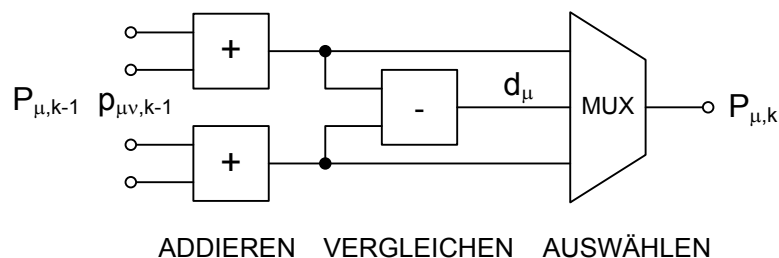


Abbildung 4.21: Addier-Vergleichs-Auswahleinheit

In jedem Zustand im Trellis enden zwei Pfade vorheriger Zustände μ , die als Pfadmetriken $P_{\mu,k-1}$ bezeichnet werden. Zu jeder Pfadmetrik werden in einem Zustandsübergang die Zweigmetriken $p_{\mu\nu,k-1}$ addiert. Die Zweigmetriken repräsentieren die inversen Wahrscheinlichkeiten für den Zustandsübergang. Beide Summen werden verglichen und die kleinere der beiden Summen $P_{\mu,k}$ sowie die Information d_{μ} , über welchen Pfad sich die kleinere Summe ergeben hat, in jeweils einem Ausgangsregister gespeichert. Dieses Entscheidungsbit ist eine Art Adressbit, das auf einen von zwei möglichen Vorgängerzuständen zeigt.

Abbildung 4.21 zeigt ein Beispiel für eine ACS-Operation an einem Zustand im Trellis. Die Addier-Vergleichs-Auswahleinheit (engl. Add-Compare-Select-Unit, ACSU) besteht aus zwei Addierern, einem Komparator und einem Ausgangsmultiplexer. Beim Komparator handelt es

sich ebenfalls um einen Addierer, bei dem die Ausgangsbits eines der beiden Addierer invertiert werden und das Eingangsübertragsbit des Komparators auf 1 gesetzt wird. Dadurch ergibt sich im mathematischen Sinne eine Subtraktion, deren eigentliches Ergebnis für die weitere Verarbeitung ohne Belang ist. Es ist nur das Übertragsbit des Subtrahierers von Interesse, welches angibt, welche der beiden Summen der vorgeschalteten Addierer die kleinere ist. Dieses Übertragsbit des Vergleichers dient also als Entscheidungsbit für den Multiplexer am Ausgang, der die kleinere Summe ausgibt. Mathematisch realisiert die ACSU die Gleichung (2.15).

Die Addition ist ein rechentechnisch aufwändiger Vorgang und bildet das Nadelöhr im Hinblick auf die maximale Geschwindigkeit des Viterbi-Entzerrers. Deshalb erfolgt in den nächsten Unterkapiteln eine Diskussion verschiedener Addierverfahren. Da die meisten Addierverfahren auf einem Volladdierer basieren, wird dieser zunächst behandelt.

4.2.1 Der Volladdierer

Der Volladdierer realisiert mathematisch zwei Funktionen. Er hat zwei Eingänge für die zu addierenden Bits A_i und B_i und einen weiteren Eingang C_i für einen eventuellen Übertrag. Die Ausgänge geben das Summenbit S_i und das Übertragsbit für die Folgestelle C_{i+1} an. Tabelle 4.6 zeigt die Wahrheitstabelle eines Volladdierers.

Tabelle 4.6: Wahrheitstabelle eines Volladdierers

A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Daraus lassen sich die Booleschen Gleichungen für die Summe (4.3) und für den Übertrag (4.4) herleiten:

$$S_i = A_i \cdot B_i \cdot C_i + \bar{A}_i \cdot \bar{B}_i \cdot C_i + \bar{A}_i \cdot B_i \cdot \bar{C}_i + A_i \cdot \bar{B}_i \cdot \bar{C}_i = A_i \oplus B_i \oplus C_i \quad (4.3)$$

$$C_{i+1} = A_i \cdot B_i + A_i \cdot C_i + B_i \cdot C_i = A_i \cdot B_i + C_i \cdot (A_i + B_i) \quad (4.4)$$

Aus den zusammengefassten Logikgleichungen erkennt man, dass sich das Summensignal mit Hilfe von zwei XOR-Gattern (Exklusiv-Oder-Verknüpfung) und das Übertragungssignal aus zwei UND-Gattern sowie zwei ODER-Gattern realisieren lassen. Bei eingehender Betrachtung der Wahrheitstabelle (Tabelle 4.6) stellt man fest, dass der Übertrag C_i an die Folgestelle weitergegeben wird, sobald eines der beiden Eingangssignale A_i oder B_i eine logische 1 aufweist. Deshalb nennt man die Teilfunktion

$$P_i = A_i \cdot \bar{B}_i + \bar{A}_i \cdot B_i = A_i \oplus B_i \quad (4.5)$$

auch Übertragsweitergabefunktion bzw. Übertragsweitergabesignal (engl. propagate). Sind hingegen beide Eingangssignale A_i und B_i logisch 1, wird auf jeden Fall ein Übertrag generiert. Deshalb nennt man die Teilfunktion

$$G_i = A_i \cdot B_i \quad (4.6)$$

auch Generierungsfunktion bzw. Generierungssignal (engl. generate). Aufgrund dieser Eigenschaften kann man den Übertrag auch als

$$C_{i+1} = G_i + P_i \cdot C_i \quad (4.7)$$

schreiben. Wird (4.5) und (4.6) in (4.7) eingesetzt, ergibt sich für den Übertrag C_{i+1} :

$$C_{i+1} = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i \quad (4.8)$$

Bei der Realisierung des Übertragungspfades in Hardware kann auf das Zwischenergebnis P_i zurückgegriffen werden, wodurch ein ODER-Gatter wegfällt.

Die vollständige Umsetzung des Volladdierers, bestehend aus zwei XOR-Gattern, zwei UND-Gattern und einem ODER-Gatter, in einem Schaltplan ist in Abbildung 4.22 dargestellt.

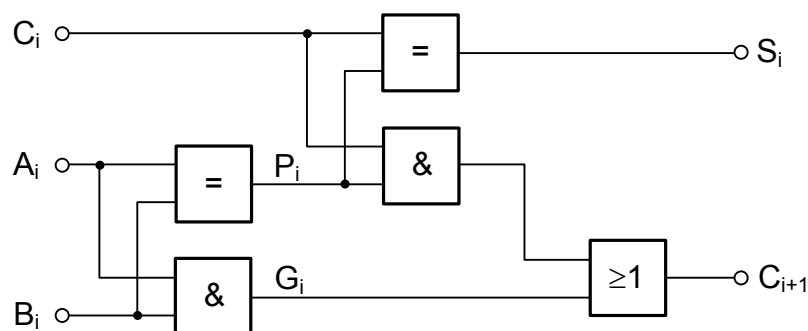


Abbildung 4.22: Umsetzung des Volladdierers aus den Booleschen Gleichungen

Die Struktur der statischen CMOS-Logik bedingt einen komplementären Aufbau des p- und n-Kanalzweiges zueinander. Das bedeutet, dass immer nur einer der beiden Zweige elektrisch leiten kann. Der p-Kanalzweig sorgt dafür, dass das CMOS-Logikgatter am Ausgang einen logischen High-Pegel annimmt, wenn an seinen Gate-Anschlüssen ein Low-Pegel anliegt. Daher wird der p-Kanalzweig auch „Pull-up-Pfad“ genannt. Der n-Kanalzweig sorgt hingegen für einen logischen Low-Pegel am Gatterausgang, wenn an seinen Gate-Anschlüssen ein High-Pegel anliegt. Deshalb der n-Kanalzweig auch „Pull-down-Pfad“ genannt wird [6]. Aus diesen Eigenschaften geht hervor, dass CMOS-Logikgatter in Ihrem Grundaufbau immer invertierende Logikfunktionen darstellen. Die Grundgatter hierfür sind der Inverter (Trivialgatter), das verneinte UND- (NAND) und das verneinte ODER- (NOR) Gatter. Wie in Abbildung 4.22 dargestellt ist, besteht der Volladdierer jedoch aus zwei nicht-negierten UND- und einem nicht-negierten ODER-Gatter. Daher wird diese Art der Realisierung nicht zu dem kürzesten möglichen kombinatorischen Pfad führen, denn nicht-invertierende Gatter müssen z.B. durch Nachschaltung eines Inverters aus einem invertierenden Grundgatter gebildet werden. Dadurch erhöht sich die kombinatorische Laufzeit mindestens um eine Inverterlaufzeit. Weitere Nachteile sind der erhöhte Transistoraufwand und der daraus folgende erhöhte Flächen- und Strombedarf. Tabelle 4.7 fasst einige Logikgatter, die in einer 90 nm-CMOS-Technologie simuliert sind, zusammen. Daraus lässt sich erkennen, dass die Verzögerungszeiten der negierten Grundgatter deutlich niedriger ausfallen, als die der nicht-negierten.

Tabelle 4.7: Gatterverzögerungszeiten in einer 90 nm CMOS Technologie

Gatter	Verzögerungszeit
Inverter	9 ps
UND	26 ps
ODER	24 ps
Negiertes UND	14 ps
Negiertes ODER	16 ps
XOR	33 ps
Negiertes XOR	32 ps

Deshalb ist zu überlegen, ob bzw. wie die UND- und ODER-Glieder in negierte Logikfunktionen übergeführt werden können. Durch zweimalige Inversion der Logikgleichung (4.8) des Übertragungspfad es mit Hilfe des De Morganschen Theorems erhält man

$$\overline{\overline{C_{i+1}}} = \overline{\overline{A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i}}$$

$$C_{i+1} = \overline{\overline{A_i \cdot B_i} \cdot \overline{\overline{(A_i \oplus B_i) \cdot C_i}}} \quad (4.9)$$

Die Summenfunktion, bestehend aus zwei Exklusiv-Oder-Verknüpfungen, kann nicht

vereinfacht werden und es bleibt deshalb bei

$$S_i = A_i \oplus B_i \oplus C_i \quad (4.3)$$

Mit (4.9) ist gezeigt, dass sich der Volladdierer mit drei NAND-Gattern und zwei XOR-Gattern realisieren lässt. Der Schaltplan für den optimierten Volladdierer ist in Abbildung 4.23 dargestellt. Legt man dem Übertragungspfad die Verzögerungszeiten aus Tabelle 4.7 zugrunde, erhält man für den Übertragungspfad nach (4.9) eine um rund 27 % kürzere kombinatorische Laufzeit als bei Realisierung nach (4.8).

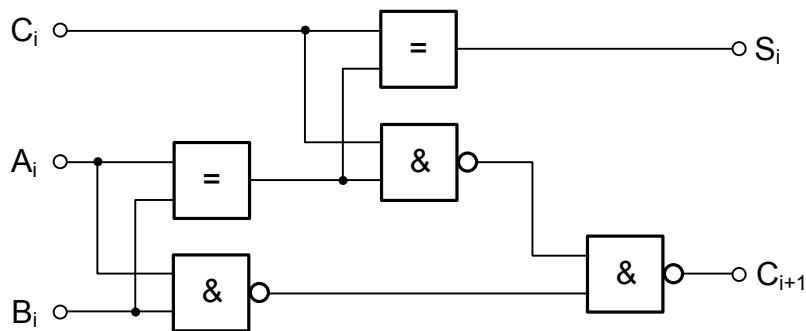


Abbildung 4.23: Volladdierer mit optimiertem Übertragungspfad

4.2.2 Addierer mit sukzessiver Übertragskorrektur

Sollen mehrstellige Binärzahlen addiert werden, kann dies durch eine Serienschaltung mehrerer Volladdierer (VA) bewerkstelligt werden. Daraus entsteht ein Paralleladdierwerk mit sukzessiver Übertragskorrektur. Die Addition zweier n bit breiter Worte erfolgt parallel. Der Übertrag, der sich aus den einzelnen 1 bit Additionen ergibt, wird sukzessive an die nächsthöhere Stelle weitergereicht. Ein Addierer mit sukzessiver Übertragskorrektur (engl. Ripple-Carry-Adder, RCA), der zwei Binärzahlen mit einer Wortbreite von n = 8 bit addiert, ist in Abbildung 4.24 dargestellt.

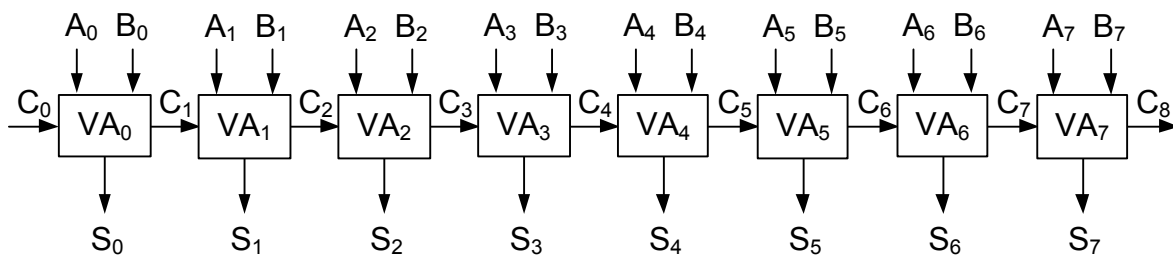


Abbildung 4.24: Paralleladdierwerk mit sukzessiver Übertragskorrektur (RCA) mit einer Wortbreite von 8 bit

Geht man davon aus, dass alle Eingangssignale A_i und B_i gleichzeitig anliegen, lässt sich die Berechnungsdauer für die Summe wie folgt angeben:

$$t_{\text{RCA}} = t_{\text{Setup}} + (n-2)t_c + t_s \quad (4.10)$$

Aus (4.10) lässt sich ableiten, dass die kombinatorische Laufzeit des RCAs linear mit der Wortbreite n ansteigt und im Wesentlichen von der Laufzeit des Übertragungssignals t_c vom Eingang C_i zum Ausgang C_{i+1} abhängt. t_{Setup} beschreibt die Verzögerungszeit des ersten VAs vom Eingang A_0 zum Ausgang C_1 . t_s gibt die Laufzeit des letzten Volladdierers hier vom Eingang C_7 zum Ausgang S_7 an. Berechnet man mit (4.10) die Verzögerungszeit eines 8 bit RCAs mit Volladdierern nach Abbildung 4.22 und legt man die Gatterlaufzeiten nach Tabelle 4.7 zugrunde, erhält man eine kombinatorische Laufzeit von 416 ps. Verwendet man dagegen Volladdierer mit optimiertem Übertragungspfad nach Abbildung 4.23, erhält man für den 8 bit RCA eine Verzögerungszeit von 262 ps. Dies entspricht einer Verkürzung der Berechnungsdauer um 37 %. Nebenbei kommt die schnellere Variante mit weniger Transistoren aus, was sich zusätzlich günstig auf den Flächenverbrauch und die Stromaufnahme auswirkt.

Tabelle 4.8: Simulation des 8 bit RCAs mit optimiertem Übertragungspfad

Schaltvorgang am Eingang	Verzögerungszeit
A = 0, B = 1, C_{in} : 0 → 1	243 ps
A = 0, B = 1, C_{in} : 1 → 0	241 ps
A = 1, B = 0, C_{in} : 0 → 1	245 ps
A = 1, B = 0, C_{in} : 1 → 0	241 ps

In Tabelle 4.8 zeigt sich ein deutlich verbessertes Schaltverhalten gegenüber den Standardzellenaddierern des Halbleiterherstellers (vgl. Tabelle 3.1). Einerseits ist die Verzögerungszeit nahezu unabhängig vom Schaltvorgang, andererseits erfolgt die Addition mit fast 50 % höherer Geschwindigkeit. Daher ist eine Optimierung des Übertragungspfad es unerlässlich, wie sie beispielsweise in Abbildung 4.23 dargestellt ist. Eine Verwendung von Standardzellen scheidet aus. Die maximale Verzögerungszeit, die in Tabelle 4.8 angegeben ist, tritt am Ausgang S_7 und nicht am Ausgang C_8 auf, da die Laufzeit des Summensignals aufgrund des XOR-Gatters größer ist als die Laufzeit von zwei in Serie geschalteten NAND-Gattern im Übertragungssignal.

Für die Schaltvorgänge aus Tabelle 4.8 ergeben sich für den einzelnen Volladdierer die Verzögerungszeiten nach Tabelle 4.9.

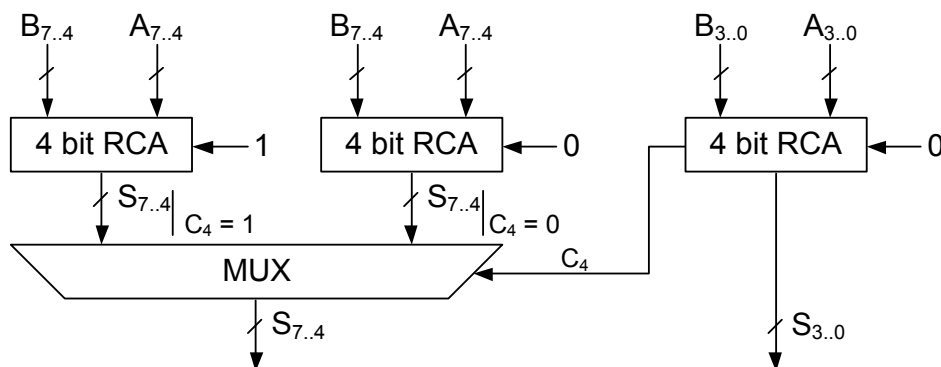
Tabelle 4.9: Simulation des Volladdierers mit optimiertem Übertragungspfad

Ausgang	Verzögerungszeit
Summe S	36,4 ps
Übertrag C	29,2 ps

Weitere Volladdiererarchitekturen wie unsymmetrischer und symmetrischer Volladdierer, direkte Umsetzung der Volladdierlogikgleichungen sowie Volladdierer mit Transferelementen zur Optimierung des Übertragungspfad werden in [6], [41], [52], [53] diskutiert. Die Standardzellen des Halbleiterherstellers wurden ebenfalls untersucht. Durch Simulationen [43], [54] und Zugrundelegung der Gatterlaufzeiten aus Tabelle 4.7 hat sich herausgestellt, dass die Volladdiererarchitektur nach Abbildung 4.23 die schnellste ist.

4.2.3 Addierer mit teilsukzessiver und teilparalleler Übertragskorrektur

Bei einem Addierer mit sukzessiver Übertragskorrektur bestimmt die Übertragslaufzeit im Wesentlichen die Gesamtrechendauer. Eine Alternative zum RCA ist ein Addierer, der eine teilparallele Berechnung des Übertrags vornimmt. Dazu fasst man mehrere Volladdierer zu kurzen RCA-Ketten zusammen. Abbildung 4.25 verdeutlicht dieses Vorgehen am Beispiel eines 8 bit Addierers. Der Addierer in Abbildung 4.25 besteht aus insgesamt drei 4 bit RCAs. Dabei sind die RCAs, die das höherwertigere 4 bit Teilwort berechnen, doppelt ausgeführt. Einer der beiden RCAs bestimmt die Teilsumme für den Fall, dass das Übertragsbit aus dem niederwertigeren 4 bit Teilwort eine 0 ergibt, während der andere RCA die Teilsumme für den Fall berechnet, dass das Übertragsbit des niederwertigeren Teilworts eine 1 ergibt. Ein Multiplexer entscheidet auf Basis des Übertragsausgangs C_4 des RCAs des niederwertigeren Teilworts, ob für das Summenergebnis des höherwertigeren Teilworts der Übertrag $C_4 = 0$ oder $C_4 = 1$ berücksichtigt werden muss. Aufgrund dieses Auswahlmultiplexers wird dieser Addierer in der Fachsprache „Carry-Select-Addierer“ (CSelA) genannt.

**Abbildung 4.25:** Carry-Select-Addierer mit einer Wortbreite von 8 bit

$$t_{\text{CSelA}} = t_{\text{Setup}} + \left(\frac{n}{m+1} - 2 \right) t_{\text{C}} + m t_{\text{MUX}} + t_{\text{S}} \quad \text{für } \frac{n}{m+1} > 2 \quad (4.11)$$

Mit Gleichung (4.11) kann allgemein Verzögerungszeit bestimmt werden, dabei entspricht die Variable n der Wortbreite und die Variable m der Anzahl der Multiplexer. Legt man dem CSelA die gleichen Gatterlaufzeiten wie dem RCA mit optimierten Logikgattern zugrunde, ergibt sich für einen 8 bit CSelA aus Abbildung 4.25 eine Verzögerungszeit von 183 ps. Die Verzögerungszeit des Multiplexers t_{MUX} wird durch ein XOR-Gatter genähert.

Dadurch ist erkennbar, dass bei geeigneter Wahl von m sich die Verzögerung gegenüber dem RCA verringern lässt. Der Vorteil der geringeren Verzögerungszeit wird allerdings durch einen höheren Schaltungsaufwand erkauft. Im Beispiel eines 8 bit Addierers aus Abbildung 4.25 beträgt der zusätzliche Mehraufwand mehr als 50 %. Bei einer höheren Multiplexeranzahl kann sich der Aufwand auf über 100 % belaufen. Eine Reduzierung des Aufwandes der Architektur nach Abbildung 4.25 kann erreicht werden, indem die Summenbildung an den Ausgang des Multiplexers verlagert wird. Dadurch wird nur der Übertragsberechnungsanteil der einzelnen Volladdierer verdoppelt. Auf die Verzögerungszeit hat diese Optimierung der Architektur keine Auswirkung.

4.2.4 Addierer ohne Übertragskorrektur

Beim Addierer ohne Übertragskorrektur ist die Laufzeit von der Wortbreite n der zu addierenden Zahlen unabhängig. Die Verzögerungszeit entspricht daher nur der Laufzeit eines Volladdierers, jedoch wird das Ergebnis als Tupel, bestehend aus Summenbit und Übertragsbit, dargestellt. Dabei entsteht eine redundante Zahlendarstellung, da sich jede einzelne Zahl aus verschiedenen Tupeln bilden lässt [54], [55]. Die Ergebnisbits der einzelnen Stufen werden parallel, also völlig unabhängig voneinander berechnet. Da das Übertragsbit nicht weiterverrechnet wird, sondern gespeichert wird, nennt man diesen Addierer in der Fachsprache „Carry-Save-Addierer“ (CSA). Um zu einer Binärzahlendarstellung zu kommen, müssen diese beiden Tupel mit Hilfe eines Addierers zusammengeführt werden (engl. Vektor-Merging-Adder, VMA). Dieser VMA kann aus einem einfachen RCA bestehen oder auch baumförmig aufgebaut werden. Daraus wird ersichtlich, dass sich CSAs hauptsächlich bei mehrstufigen Additionen eignen, um Zwischenergebnisse darzustellen. Eine bedeutende Anwendung des CSAs ist der Feldmultiplizierer [6], [54]. Hierbei kann bei der Addition der Partialprodukte eine deutliche Beschleunigung des Rechenvorgangs im Gegensatz zum RCA erreicht werden. Nach Addition der Partialprodukte ist jedoch eine Umwandlung der redundanten Zahlendarstellung in die Binärzahlendarstellung notwendig.

4.2.5 Paralleladdierer mit Übertragsvorausberechnung

Paralleladdierer mit Übertragsvorausberechnung (engl. Carry-Lookahead-Adder, CLA) sind die schnellsten bekanntesten Mehrbit-Addierer. Man unterscheidet im Wesentlichen zwischen Architekturen mit paralleler Übertragskorrektur und Architekturen mit Baumstruktur. Beide Architekturen werden in den folgenden Teilkapiteln kurz behandelt.

4.2.5.1 Parallele Übertragskorrektur

Der CLA mit paralleler Übertragskorrektur (Abbildung 4.26) basiert auf dem Prinzip, dass alle Überträge parallel aus den gleichzeitig anliegenden Übertragsweitergabe- ($P_{0..7}$) und übertrags-generierenden ($G_{0..7}$) Signalen berechnet werden. Das Summensignal lässt sich aus einer einfachen XOR-Verknüpfung aus dem Übertrags- und dem Übertragsweitergabesignal gewinnen.

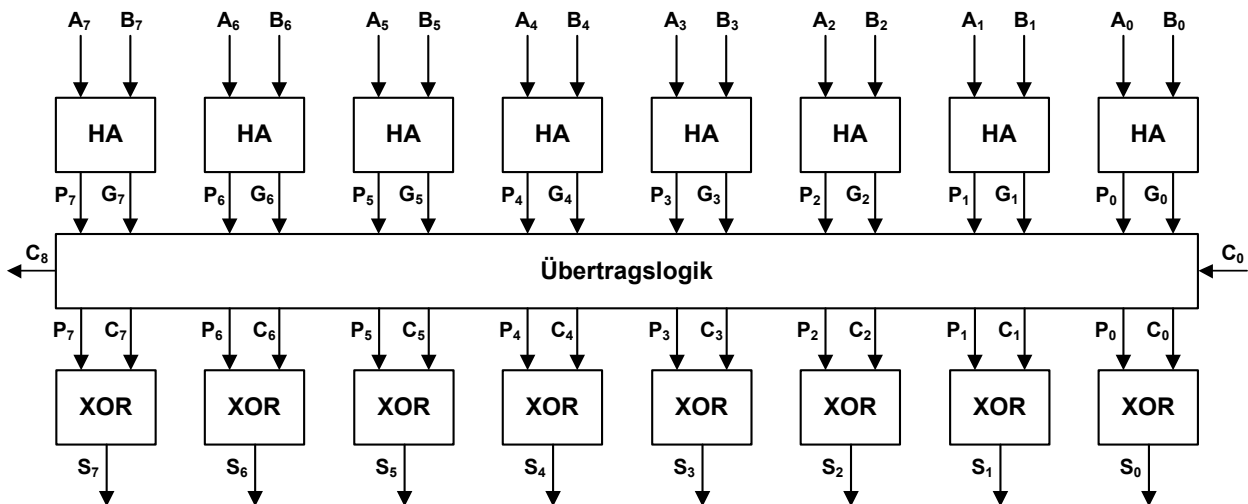


Abbildung 4.26: Addierer mit paralleler Übertragskorrektur (CLA) mit 8 bit Wortbreite

Der Übertrag jeder Bitstelle lässt sich rekursiv durch

$$C_{i+1} = G_i + P_i C_i \quad (4.12)$$

bestimmen. Um den Übertrag parallel in Abhängigkeit der Eingangsbits zu berechnen, muss die Rekursion entflochten werden. Durch Ineinandereinssetzen der Gleichung (4.12) erhält man für einen i bit breiten Addierer

$$C_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i \cdot P_{i-1} \dots P_0 C_0 \quad (4.13)$$

Das hat zur Folge, dass für jedes einzelne Übertragsbit die Gleichung (4.13) in Hardware realisiert werden muss. Für einen 8 bit CLA ergibt sich das Gleichungssystem für die Übertragslogik nach (4.14).

$$\begin{aligned}
 C_1 &= G_0 + P_0 C_0 \\
 \dots & \\
 C_8 &= G_7 + P_7 G_6 + P_7 P_6 G_5 + P_7 P_6 P_5 G_4 + P_7 P_6 P_5 P_4 G_3 + P_7 P_6 P_5 P_4 P_3 G_2 + \\
 &\quad P_7 P_6 P_5 P_4 P_3 P_2 G_1 + P_7 P_6 P_5 P_4 P_3 P_2 P_1 G_0 + P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 C_0
 \end{aligned}
 \tag{4.14}$$

Durch die parallele Übertragsberechnung ist so zu sagen eine gleichzeitige Berechnung des Gesamtergebnisses in nur drei Schritten (Abbildung 4.26) möglich. Allerdings ist zu beachten, dass für die höherwertigen Überträge UND-Verknüpfungen mit einem Fan-In von bis zu 9 Eingängen notwendig sind. Dadurch erhöht sich die kombinatorische Laufzeit im ungünstigsten Fall quadratisch [41] in Abhängigkeit der Anzahl der Eingänge eines Gatters. Simulationen hingegen zeigen immerhin eine lineare Abhängigkeit [39]. Daraus folgt, dass es trotzdem eine lineare Abhängigkeit zwischen Wortbreite und Verzögerungszeit gibt. Eine Möglichkeit, die die Rechendauer der Addition beschleunigt, wird im folgenden Kapitel diskutiert.

4.2.5.2 Baumstruktur

Um die lineare Abhängigkeit der Verzögerungszeit von der Wortbreite zu vermeiden, wird eine Baumstruktur des CLAs nach Abbildung 4.27 [56] vorgeschlagen.

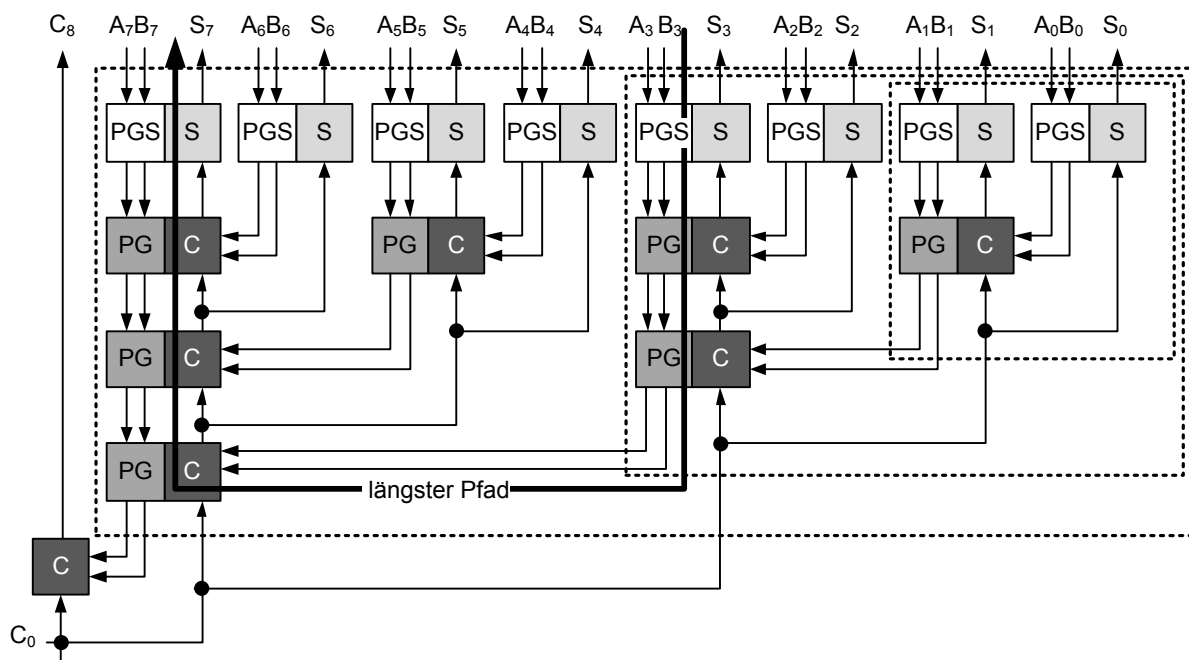


Abbildung 4.27: Addierer mit baumförmiger Übertragskorrektur (CLA) mit 8 bit Wortbreite

Die ersten Übertragsweitergabesignale und übertragsgenerierenden Signale werden mittels einer ODER- bzw. UND-Verknüpfung erzeugt (Propagate-Generate-Setup-Element, PGS). Das Element, das die weiteren Übertragsweitergabesignale und übertragsgenerierenden Signale erzeugt

(Propagate-Generate-Element, PG), erfüllt die Assoziationsoperation $(g, p) \bullet (g', p') = (g + (p \cdot g'), p \cdot p')$ [57] und ist Teil der Vorwärtsbaumstruktur nach Brent und Kung [58], während das Übertragsberechnungselement (Carry-Element, C) die Rückwärtsbaumstruktur nachbildet.

Wie anhand der Abbildung 4.27 durch die gestrichelten Rahmenlinien zu erkennen ist, erhöht sich die Laufzeit bei Verdopplung der Wortbreite jeweils nur um ein PG- und ein C-Element. Daraus ergibt sich ein logarithmischer Zusammenhang zwischen Eingangswortbreite und Verzögerungszeit. Formel (4.15) beschreibt für die Berechnung der Verzögerungszeit des CLAs mit binärem Baum:

$$t_{CLA} = (t_{PGS,n} + t_{PG,n} \cdot (\lg(n) - 1) + t_{C,n} \cdot \lg(n) + t_{S,n}) t_n \quad (4.15)$$

t_n gibt eine normierte Verzögerungszeit an, die der niedrigsten absoluten Gatterlaufzeit der in den Blöcken HA, PG, C und S enthaltenen Logikgattern nach Tabelle 4.7 entspricht. Im hier vorliegenden Fall handelt es sich um das UND- bzw. ODER-Gatter.

$$t_{CLA} = (1 + 2 \cdot (\lg(n) - 1) + 2 \cdot \lg(n) + 2,6) t_n = (4 \cdot \lg(n) + 1,6) t_n \quad (4.16)$$

Nach Einsetzen der normierten Verzögerungszeit von $t_n = 25$ ps erhält man für einen 8 bit CLA eine Verzögerungszeit von $t_{CLA} = 340$ ps.

Es gibt Möglichkeiten die Gatterlaufzeit des baumförmigen CLAs zu optimieren [53], [57], [58], [59], [60], indem z.B. invertierte Logik verwendet wird. Dies wirkt sich zwar positiv auf die normierte Verzögerungszeit t_n in (4.16) aus, jedoch wird die Konstante $t_{S,n}$ jedoch größer, da das Verhältnis der normierten Verzögerungszeit zwischen XOR- und NAND-/NOR-Gatter größer ist als zwischen XOR- und UND-/ODER-Gatter. Nach aktuellem Kenntnisstand sind keine schnelleren Addierer bekannt, als welche, die einen logarithmischen Zusammenhang zwischen Wortbreite und Verzögerungszeit aufweisen.

4.2.6 Komparator

Wie aus der Bezeichnung Addier-Vergleichs-Auswahl-Operation hervorgeht, müssen die aufsummierten Metriken verglichen werden. Dazu wird ein digitaler Komparator benötigt. Mit Hilfe der Zweierkomplementdarstellung binärer Zahlen, lässt sich die Subtraktion leicht auf die Addition zurückführen [61]. Um zwei positive Binärzahlen voneinander zu subtrahieren, bildet man vom Subtrahenden das Zweierkomplement. Unter dem Zweierkomplement versteht man die

entsprechende negative Zahl einer positiven Binärzahl. Durch Addition des Zweierkomplements des Subtrahenden ergibt sich die Differenz.

Folgendes Beispiel verdeutlicht die Vorgehensweise:

Es gilt: $A > 0$ und $B > 0$

$$A - B = A + (-B)$$

dabei entspricht $-B$ dem Zweierkomplement der Binärzahl B .

Für die Addier-Vergleichs-Auswahl-Operation ist die eigentliche Differenz irrelevant. Es ist nur von Interesse, welche der beiden Zahlen A oder B die größere bzw. die kleinere ist. Daraus folgt, dass im Komparator keine Summenbits berechnet werden müssen sondern nur die Übertragsbits. Bei einer Zweierkomplementdarstellung kann am höchstwertigen Bit einer Binärzahl abgelesen werden, ob es sich bei der Differenz um eine positive oder negative Zahl handelt. Dabei gilt: Wenn $A > B$, das heißt, die Differenz aus $A - B > 0$ ist, dann ist das höchstwertige Übertragsbit der Subtraktion 0. Bei $A < B$, das heißt, die Differenz aus $A - B < 0$ ist, dann ist das höchstwertige Übertragsbit 1. An Hand dieser Information kann ein nachgeschalteter Multiplexer entscheiden, welche der beiden aufsummierten Metriken weitergeleitet wird, bzw. welcher Pfad im Trellis zum ausgewählten Ergebnis geführt hat.

Im Wesentlichen gibt es zwei Architekturansätze für digitale Komparatoren, die im Folgenden kurz diskutiert werden.

4.2.6.1 Linearer Komparator

Der lineare Komparator basiert im Wesentlichen auf dem Addierer mit sukzessiver Übertragskorrektur. Betrachtet man zwei Binärzahlen A und B , ist diejenige die größere Zahl, bei der bei bitweisem Vergleich das Bit mit der höchsten Wertigkeit eine 1 aufweist.

Beispiel:

$$\begin{array}{r} A_7 \dots\dots\dots A_0 \\ S_0 \quad 0010.1011 \\ S_1 \quad 0001.0111 \end{array}$$

Vergleicht man vom höchstwertigen Bit her beginnend die einzelnen Bits zum niederwertigsten Bit, erkennt man, dass das Bit A_5 der Zahl S_0 eine 1 aufweist, während das Bit gleicher Wertigkeit der Zahl S_1 0 ist. Daraus folgt: $S_0 > S_1$.

Da bei fast allen in den vorhergehenden Kapiteln vorgestellten Addierern die niederwertigeren Bits früher als die höherwertigeren Bits stabil ausgerechnet sind, kann der Vergleich vom niederwertigsten Bit her beginnen. Sind zwei Bits gleicher Wertigkeit i beider Zahlen gleich

($S_{0,i} = S_{1,i}$), entscheidet das nächst niederwertigere Bit $S_{j,i-1}$. Für den Fall, das $S_0 = S_1$ gilt, soll per Definition S_1 als die kleinere Zahl betrachtet werden.

Die Wahrheitstabelle eines Einzelkomparators ist in Tabelle 4.10 dargestellt.

Tabelle 4.10: Wahrheitstabelle eines einzelnen digitalen Komparators

$S_{0,i}$	$S_{1,i}$	$C_{in,i}$	$C_{out,i}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Daraus lässt sich folgende Logikgleichung ableiten:

$$C_{out} = C_{in,i} \cdot S_{0,i} + S_{0,i} \cdot \bar{S}_{1,i} + C_{in,i} \cdot \bar{S}_{0,i} \cdot \bar{S}_{1,i} = S_{0,i} \cdot S_{1,i} + C_{in,i} (S_{0,i} + \bar{S}_{0,i} \cdot \bar{S}_{1,i}) \quad (4.17)$$

Da invertierende Logikgatter schneller sind als nicht invertierende, kann Gleichung (4.17) umgeformt werden zu

$$C_{out} = \overline{S_{0,i} \cdot \bar{S}_{1,i} \cdot (S_{0,i} \oplus \bar{S}_{1,i})} \cdot C_{in,i} \quad (4.18)$$

Mit Hilfe dieser Umformung erhält man fast den gleichen kritischen Pfad wie auf dem Übertragungspfad eines Volladdierers. Die Verzögerungszeit erhöht sich nur um die Laufzeit eines Inverters, da das Signal \bar{S}_1 erzeugt werden muss.

Da, wie bereits erwähnt, bei Gleichheit der Zahlen S_0 und S_1 per Definition gelten soll, dass S_1 als die kleinere Zahl gewertet werden soll, vereinfacht sich der Komparator für das niederwertigste Bit zu

$$C_{out} = \overline{S_{0,i} \cdot \bar{S}_{1,i}} \quad (4.19)$$

Die Gleichung (4.19) ergibt sich, indem man in (4.18) $C_{in,i} = 1$ setzt.

Die Realisierung in Hardware erfolgt durch ein NAND-Gatter mit einem invertierten S_1 -Eingang.

4.2.6.2 Baumförmiger Komparator

Ähnlich wie bei der Addition lässt sich die Vergleichsoperation mit Hilfe eines vereinfachten Übertragvorausrechnungsverfahrens durchführen, bei dem nur die Berechnung des Übertrags relevant ist und auf die Berechnung des Ergebnisses der Differenz verzichtet werden kann. Betrachtet man die Gleichung (4.14), erkennt man, dass für den Übertrag die Übertragsweitergabe- und übertragsgenerierenden Signale benötigt werden. Überträgt man das auf Abbildung 4.27, folgt daraus, dass das Summenbildungselement (S) und die Rückwärtsbaumstruktur weggelassen werden kann. Die Halbaddierer sowie die wesentlich aufwändigeren Elemente zur Berechnung der Übertragsweitergabe- und übertragsgenerierenden Signale (PG) werden dennoch benötigt. Daraus folgt, dass die Verzögerungszeit der Vergleichsoperation ebenfalls eine logarithmische Abhängigkeit zur Wortbreite aufweist. Die Verzögerungszeit $t_{S,n}$ des Summenelements aus Gleichung (4.15) fällt weg, sodass die Gesamtlaufzeit des Komparators geringer ist als die der Addition.

4.2.7 Bewertung der Addierverfahren

In den vorhergehenden Kapiteln sind die vier möglichen Addierverfahren

- Paralleladdierer mit sukzessiver Übertragskorrektur (RCA)
- Paralleladdierer mit teilsukzessiver und teilparalleler Übertragskorrektur (CSelA)
- Paralleladdierer ohne Übertragskorrektur (CSA)
- Paralleladdierer mit Übertragsvorausberechnung (CLA)

diskutiert worden. Es folgt eine Eignungsabschätzung der Addierverfahren im Hinblick auf die Verwendung in einer ACSU. Dabei wird eine 8 bit-Addition zugrunde gelegt. Die Verzögerungszeiten der Addierertypen werden anhand ihrer Formeln und den Verzögerungszeiten der darin enthaltenen Gattern aus Tabelle 4.7 ermittelt. Tabelle 4.11 vergleicht die Addierertypen hinsichtlich Verzögerungszeit und Transistoraufwand.

Tabelle 4.11: Vergleich der Addierertypen für eine 8 bit-Addition

Addierertyp	Verzögerungszeit	Transistoraufwand
RCA	262 ps	240
CSelA	183 ps	370-382
CSA	61 ps	240
CLA	340 ps	510

Architekturbedingt ist der Paralleladdierer mit Übertragsvorausberechnung (CLA) der schnellste Addierer für große Wortbreiten, aber auch der flächenmäßig aufwändigste. Im hier vorliegenden Fall ist der CLA sogar langsamer als der RCA. Der Geschwindigkeitsvorteil des CLAs gegenüber dem RCA kommt erst bei größeren Wortbreiten ab ca. 13 bis 14 bit zum Tragen.

Bei dem Paralleladdierer ohne Übertragskorrektur (CSA) kann, anhand dessen Ergebnisses ohne weitergehende Betrachtung keine Aussage getroffen werden kann, ob eine Zahl größer oder kleiner als eine andere ist. Für einen Vergleich müsste ein aufwändiger CS-Vergleicher wie z.B. in [7] entworfen werden. Eine komplette Berechnung der Metriken in CS-Arithmetik hätte aufgrund der separaten Behandlung von Summe und Übertrag eine doppelt so aufwändige Verdrahtung zwischen den ACSUs zur Folge. Außerdem verdoppelt sich die Anzahl der Flipflops, wodurch der Stromverbrauch erheblich ansteigt.

Der Paralleladdierer mit teilsukzessiver und teilparalleler Übertragskorrektur (CSelA) ist hinsichtlich der Verzögerungszeit schneller als der RCA. Kombiniert man jedoch einen CSelA und einen CSel-Vergleicher, wie es in der ACSU notwendig ist, verdoppelt sich die Verzögerungszeit ungefähr.

Der Paralleladdierer mit sukzessiver Übertragskorrektur (RCA) ist architekturbedingt der langsamste, aber auch der sparsamste hinsichtlich des Flächen- und Strombedarfs. Betrachtet man die Kombination aus RCA und Vergleicher, zeigt sich die volle Leistungsfähigkeit dieser Architektur in einer ACSU. Da die Teilsummen bitweise verzögert berechnet werden, kann dazu parallel mit dem Vergleich gestartet werden. Daraus folgt, dass das Ergebnis der ACS-Operation nur eine Volladdiererlaufzeit später anliegt als das der Summenbildung. Daher eignet sich für die Implementierung der ACSU die Kombination aus RCA und Vergleicher am besten.

4.2.8 Hardwaretechnische Umsetzung der Addier-Vergleichs-Auswahleinheit

Die endgültige ACSU besteht aus zwei parallel arbeitenden Addierern mit sukzessiver Übertragskorrektur (RCA) zur Berechnung der Summenpfadmetriken. Parallel zur Addition erfolgt ein bitsukzessiver Vergleich der beiden Summenpfadmetriken. Hierzu kommt ein linearer Komparator zum Einsatz (RCA-Komparator). Die beiden Summenpfadmetriken liegen an einem Multiplexer an, der die kleinere Summenpfadmetrik auf Basis der Komparatorentscheidung weitergibt. Abbildung 4.28 zeigt den detaillierten Aufbau der ACSU. Der dicke Pfeil zeigt den längsten kombinatorischen Pfad an. Daraus folgt, dass die Dauer der Addition und des Vergleichs nur um eine Laufzeit eines Komparators verlängert wird. Aufgrund der unterschiedlichen Wortbreite der Zweig- und Pfadmetriken können die erste als auch die letzten beiden 1-Bitadditionen als Halbaddierer HA0 ausgeführt werden. Gleiches gilt für den ersten Komparator K0. Dadurch wird zu Beginn eine halbe Volladdiererlaufzeit gespart. Die beiden Halbaddierer HA6 und HA7 führen zwar zu keiner Verkürzung des kombinatorischen Pfades, jedoch kann durch die geringere Zahl an Transistoren ein wenig Fläche und Strom gespart werden.

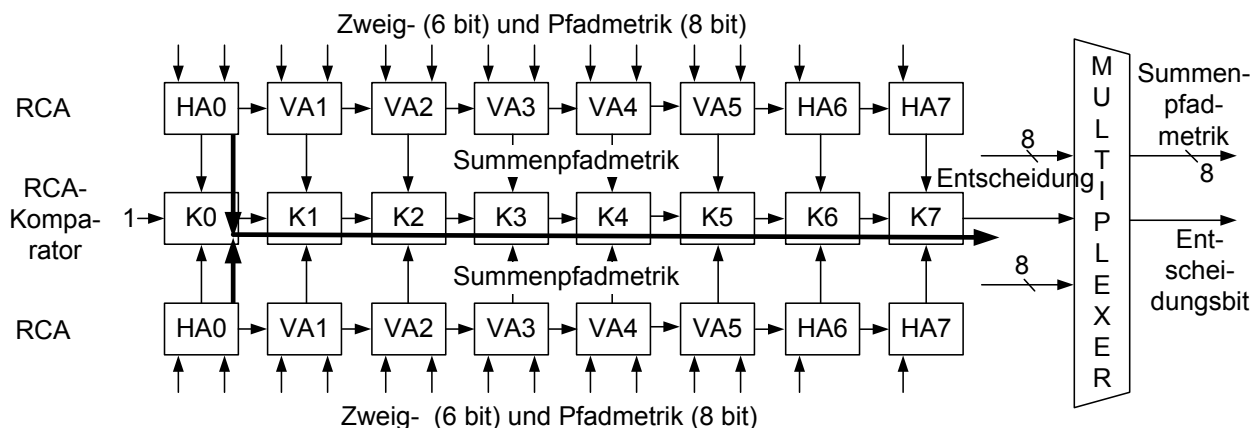


Abbildung 4.28: Blockschaltbild der realisierten Addier-Vergleichs-Auswahleinheit

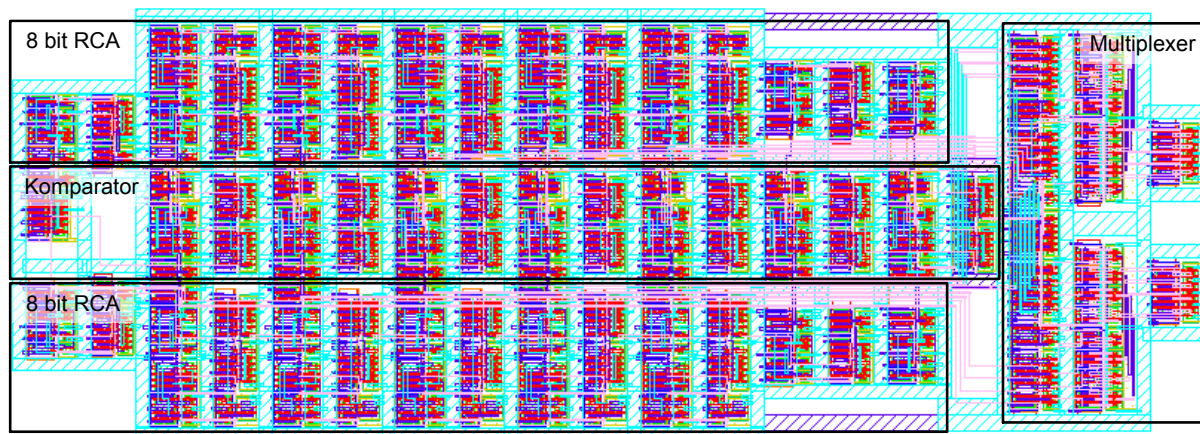


Abbildung 4.29: Layout der realisierten Addier-Vergleichs-Auswahleinheit

Abbildung 4.29 zeigt das Layout der ACSU. Die Eckdaten des Layouts sind in Tabelle 4.12 zusammengefasst.

Tabelle 4.12: Eckdaten des Layouts der ACSU

Höhe	26 μm
Breite	74 μm
Fläche	1924 μm^2
Anzahl Transistoren	632

Die interne Verdrahtung der einzelnen Logikgatter wie der Volladdierer und Komparatoren erfolgt auf den untersten beiden Metalllagen. Erst für die Verbindung der Volladdierer, Komparatoren und Multiplexer wird die dritte Metalllage verwendet. Die regelmäßige Anordnung der Addierer-, Komparatorzellen und des Multiplexers ist zu erkennen bzw. dem Blockschaltbild aus Abbildung 4.28 nachempfunden. Ebenso erkennt man aufgrund des geringeren Flächenbedarfs die Halbaddierer und den Startkomparator.

Mathematisch kann die Verzögerungszeit der ACSU wie folgt beschrieben werden:

$$t_{\text{ACSU}} = t_{\text{Setup}} + 7t_{\text{C}} + t_{\text{S}} + t_{\text{MUX}} \quad (4.20)$$

Dem längsten kombinatorischen Pfad liegt der Testfall in Tabelle 4.13 zugrunde:

Tabelle 4.13: Längster ACSU-Schaltvorgang (a) und (b)

	a)		b)	
	Zeitpunkt t_0	Zeitpunkt t_1	Zeitpunkt t_0	Zeitpunkt t_1
A _{0,7..0}	0000.0000	0000.0010	0111.1111	0111.1111
B _{0,5..0}	01.1111	01.1111	00.0000	00.0001
A _{1,7..0}	0000.0000	0000.0011	0111.1111	0111.1111
B _{1,5..0}	01.1111	01.1111	00.0000	00.0011
ACS-Entscheidung	0001.1111	0010.0001	0111.1111	1000.0000

In der Simulation ergeben sich Verzögerungszeiten nach Tabelle 4.14.

Tabelle 4.14: Verzögerungszeiten der ACSU (a) und (b)

Parameter	Verzögerungszeit a)	Verzögerungszeit b)
Schaltplan, TT	352 ps	335 ps
RCTYP, TT	505 ps	480 ps
RCMAX, SS	692 ps	656 ps

Der Testfall a) aus Tabelle 4.13 weist eine größere Verzögerungszeit auf als der Testfall b). Der wesentliche Unterschied besteht darin, dass in Testfall a) der Eingangsvektor A variabel ist und der Eingangsvektor B konstant gehalten wird, während in Testfall b) der Vektor B variabel ist und A konstant bleibt. Prinzipiell sind beide Schaltvorgänge möglich, da sich beide Eingangsvektoren ändern können. Allerdings scheint der Testfall a) der langsamere und systemrelevantere zu sein, da die durchschnittliche Leitungslänge zwischen den ACSUs (Eingang a) länger ist als die Leitungen vom Ausgangsflipflop der BMU zur ACSU (Eingang b). Daher kann man argumentieren, dass das Signal an Eingang B schon anliegt, bevor auf Eingang A ein Wechsel erfolgt.

Gemäß Formel (4.20) müsste sich auf Schaltplanebene für den Testfall a) aus Tabelle 4.13 eine Verzögerungszeit von $t_{\text{ACSU}} = 318$ ps ergeben. Auf Schaltplanebene ergibt sich eine um 11 % höhere Verzögerungszeit als berechnet. Das lässt sich über ein zweifaches Fan-Out der Summenausgänge begründen (Komparator- und Multiplexereingänge), so dass es etwas länger

dauert, bis das Rechenergebnis endgültig stabil am Ausgang anliegt. Bezieht man die parasitären Elemente des Layouts in die Simulation mit ein, erhöht sich die kombinatorische Laufzeit auf 505 ps. In beiden Testfällen erhöht sich die Laufzeit gegenüber der Schaltplansimulation mit typischen Transistoren um 43 %. Legt man dem Layout maximale Widerstände und Kapazitäten sowie besonders langsame Transistoren zugrunde, ergibt sich eine Verzögerungszeit von 692 ps. Daraus erfolgt eine Laufzeiterhöhung um 97 % gegenüber der Schaltplansimulation mit typischen Transistoren.

Vor und hinter jeder ACSU befinden sich Flipflops, die die Summenpfadmetriken des Zeitschritts t_1 bzw. t_0 puffern. Daher muss auch noch deren Verzögerungszeit sowie deren Rüstzeit einberechnet werden. Da die ACSUs örtlich nicht aneinander gereiht sind, sondern die Verdrahtung den Zustandsübergängen im Trellisdiagramm entspricht, muss die Signallaufzeit auf den Verbindungsleitungen ebenfalls berücksichtigt werden. Hierzu wird ein Teillayout von zwei ACSUs mit Flipflops am Ein- und Ausgang sowie der längstmöglichen Verbindungsleitungen erstellt und simuliert. Es wird ein anderer Testfall als Tabelle 4.13 zugrunde gelegt, da dieser die parasitären RC-Elemente der Leitungen zwischen den beiden ACSUs nicht umlädt. Der neue Testfall in Tabelle 4.15 ist zwar kombinatorisch bedingt ein wenig kürzer, berücksichtigt jedoch den weitaus stärkeren Effekt der langen Verdrahtung.

Tabelle 4.15: Längster ACSU-Schaltvorgang im Trellis

	Zeitpunkt t_0	Zeitpunkt t_1
A _{0,7..0}	0111.1111	0111.1111
B _{0,5..0}	00.0000	00.0001
A _{1,7..0}	0000.0000	0111.1111
B _{1,7..0}	00.0000	00.0011
ACS-Entscheidung	0000.0000	1000.0000

Abbildung 4.30 zeigt ein vereinfachtes Testlayout zur Simulation der parasitären Effekte der Verdrahtung zwischen zwei ACSUs [43]. In Wirklichkeit besteht dieser Block aus 16 ACSUs. Eine Simulation einer so großen Schaltung ist auf Transistorebene mit endlichen Ressourcen nicht durchführbar. Die simulierte Schaltung umfasst 2254 Transistoren. Die maximale Leitungslänge zwischen dem Ausgang der ersten ACSU und dem Eingang der zweiten ACSU beträgt 160 μm .

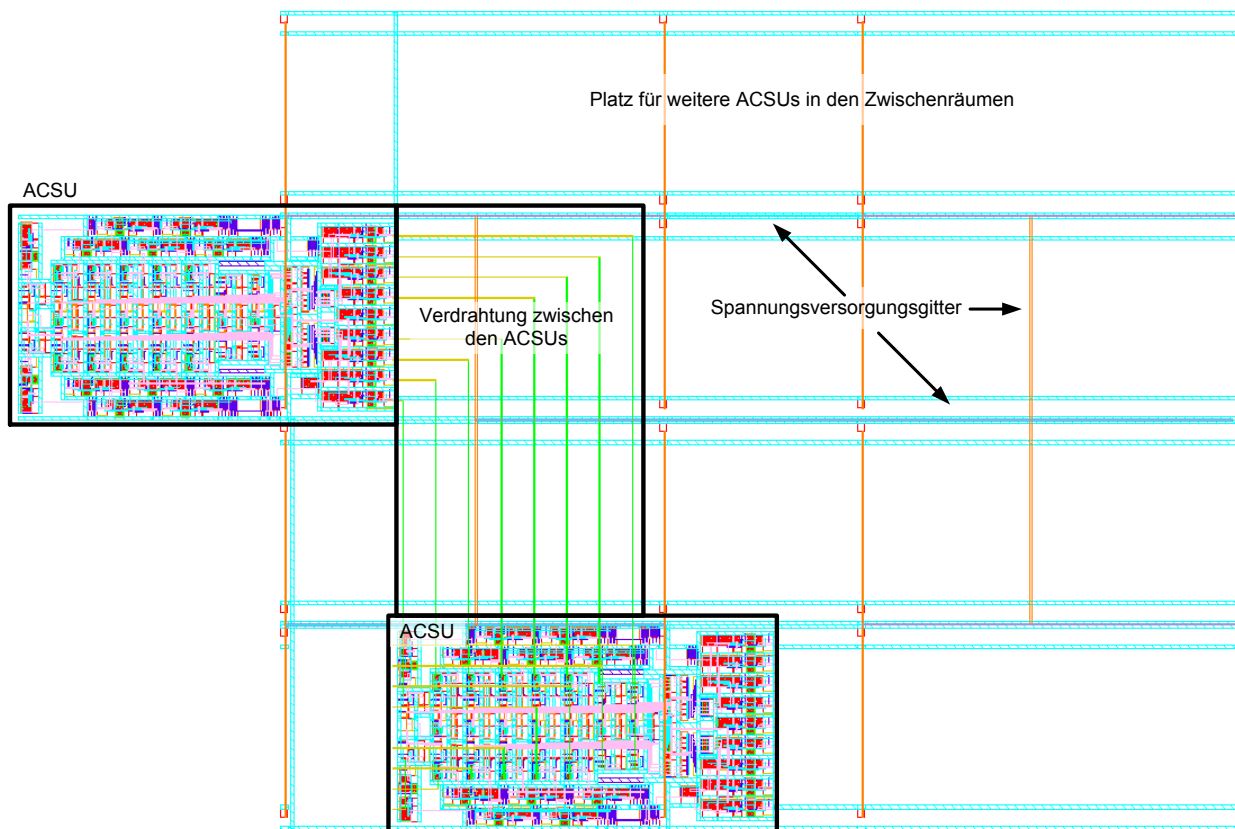


Abbildung 4.30: vereinfachtes Testlayout zur Simulation der parasitären Effekte der Verdrahtung zwischen zwei ACSUs

In der Simulation sind die Verzögerungszeiten der ACSU einschließlich Rüst- und Verzögerungszeiten der Flipflops sowie der parasitären Elemente der Verdrahtung nach Tabelle 4.16 enthalten.

Tabelle 4.16: Gesamtverzögerungszeiten der ACSU, Flipflops einschließlich Verdrahtung

Parameter	Verzögerungszeit
RCTYP, TT	614 ps
RCMAX, SS	864 ps

Diskussion der Simulationsergebnisse

Dieses Simulationsergebnis ist bedeutend für weiteren Entwurf der gesamten Architektur. Die ACSU ist das Element mit der komplexesten kombinatorischen Logik und somit das Bauteil, aus dem sich die maximale Taktfrequenz und die Parallelisierung ableitet.

Bei einer Eingangsdatenrate von 43 Gbit/s und einer kombinatorischen Laufzeit von 614 ps einschließlich parasitärer Elemente bietet sich ein Parallelisierungsgrad von 32 an. Dies bedeutet, dass die maximale Laufzeit 744 ps betragen darf. So gesehen ist also noch ein Sicherheitspuffer von 130 ps, was 17 % der Taktperiode entspricht, vorhanden, sofern sich die Technologie wie

erwartet verhält (RCTYP, TT). Legt man der Technologie besonders langsame Transistoren (SS) und besonders hohe Leitungswiderstände und Kapazitäten (RCMAX) zugrunde, würde eine 32-fache Parallelisierung nicht ausreichen. Aufgrund der Wahrscheinlichkeitsrechnung ist dieser Fall nahezu unmöglich. Für die Transistoren lässt sich sagen, dass sie zu 99,7 % (3-fache Standardabweichung / 3 Sigma) schneller sind als der SS-Simulationsfall. Ähnliches gilt für die parasitären Elemente. Da sich Wahrscheinlichkeiten multiplizieren, ist der Fall, dass RCMAX und SS gleichzeitig auftreten, nahezu ausgeschlossen.

Weitere Randbedingungen, welche o.g. Simulation zugrunde liegen, sind eine Versorgungsspannung von 1,0 V und eine Temperatur von 27 °C. Durch eine Spannungserhöhung bei gleichbleibender Temperatur kann die Geschwindigkeit zwar gesteigert werden, allerdings ist eine Temperatur von 27 °C im Betrieb unrealistisch. Ein Temperaturanstieg verlangsamt die Schaltzeiten der Transistoren [43]. Daher ist zu erwarten, dass der Viterbi-Entzerrter bei seiner maximalen Taktfrequenz nur mittels externer Kühlung funktioniert. Eventuell muss sogar auf eine aktive Kühlung mittels Peltier-Element zurückgegriffen werden.

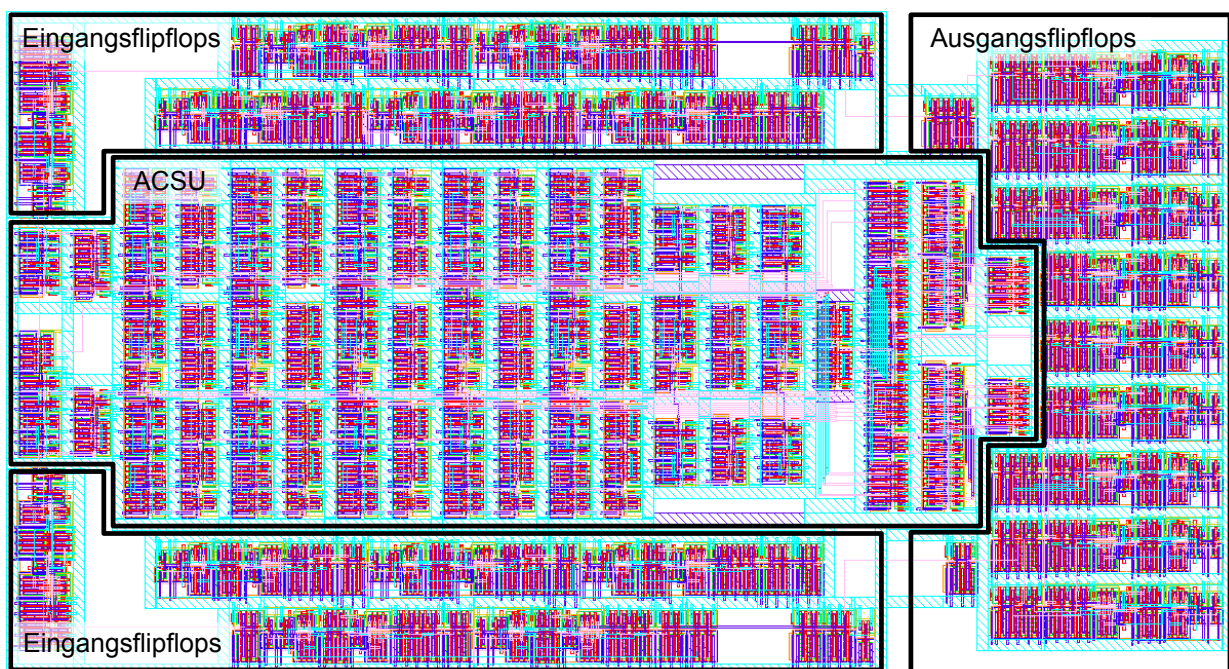


Abbildung 4.31: ACSU mit Eingangs- und Ausgangsflipflops

Aufgrund eines flächenoptimierteren Layouts werden sowohl die Ausgangsflipflops der Summenpfadmetriken und auch die Ausgangsflipflops des Zweigmetrikrechners in Form von Eingangsflipflops an der ACSU angeordnet. Abbildung 4.31 zeigt das Layout. Es umfasst 1126 Transistoren und belegt 4106 μm^2 Fläche bei einer Kantenlänge von 87 μm x 47,2 μm .

4.3 Minimumauswahlschaltung

Wie bereits in Kapitel 2.4.1.1 beschrieben, kann nicht einfach am Ende des Trellis der Pfad mit den kleinsten Metriken als überlebender Pfad gewählt werden, da man einerseits die Synchronisationslänge und andererseits die Pfadvereinigungslänge L zurückgehen muss, bis der kürzeste Pfad ermittelt ist. Daher werden zwei zeitlich aufeinander folgende Datenblöcke der jeweiligen Länge $2L$ parallel in Vorwärts- als auch in Rückwärtsrichtung verarbeitet und an den jeweiligen Enden über eine Minimumauswahlschaltung (engl. Select-Minimum-Unit, SMU) verknüpft, sodass die Entscheidung in der Mitte eines doppelt so langen Blocks der Gesamtlänge $4L$ (vgl. Abbildung 2.11) getroffen wird. Somit befindet man sich sicher auf dem kürzesten Pfad und nicht mehr im Ein- bzw. Ausschwingbereich der Teiltrellis der Länge L . Das Ergebnis dieser Verknüpfung sind nun gemeinsame Summenpfadmetriken, aus denen der kürzeste gemeinsame Pfad ermittelt werden kann. Dadurch wird eine sofortige Dekodierung der wahrscheinlichsten Datenfolge ermöglicht. Abbildung 4.32 zeigt die Einordnung der SMU. Aus zwei Richtungen wird der Trellis mit der Blocklänge $2L$ durchlaufen. Nach jeweils $2L$ Schritten werden die Summenpfadmetriken der jeweilig zueinander gehörenden Zustände addiert. Danach werden alle vier gemeinsamen Summenpfadmetriken miteinander verglichen. Der Zustand, an dem die kleinste der vier Metriken anliegt, ist der Startzustand, an dem die Pfadrückverfolgung und Dekodierung beginnt.

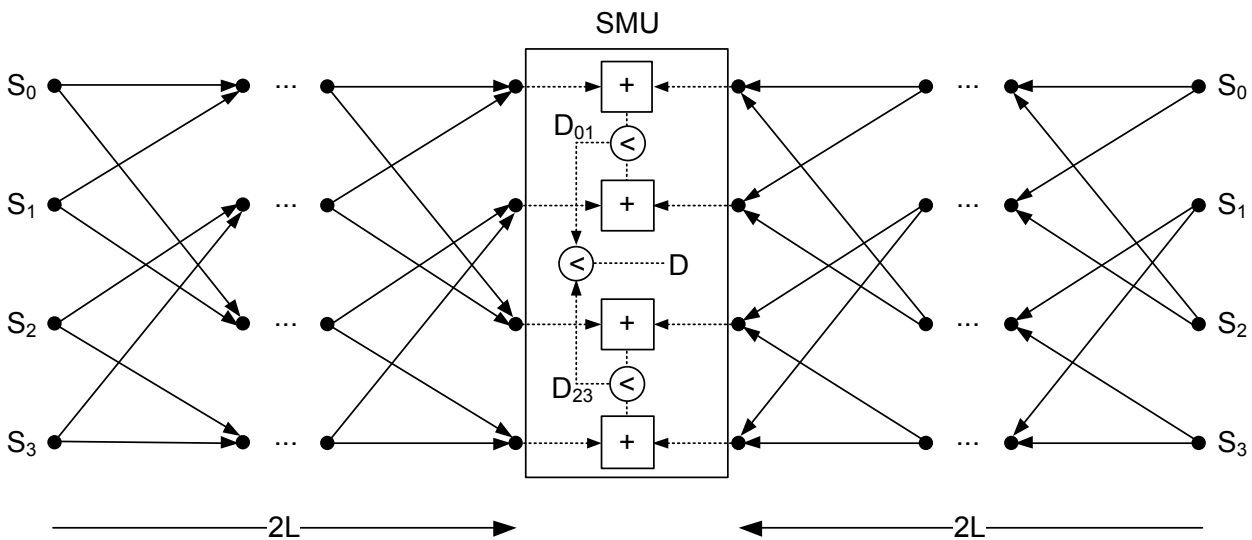


Abbildung 4.32: Minimumauswahlschaltung im Trellisdiagramm

Um aus vier Zahlen die kleinste zu finden, sind insgesamt $4! = 24$ Vergleiche notwendig, woraus ein immenser Hardwareaufwand resultiert. Daher wird in [62] ein baumförmiger Vergleich vorgeschlagen, wodurch sich die Anzahl der Vergleiche auf drei reduziert, wovon zwei parallel durchgeführt werden (siehe Abbildung 4.32).

Die Vorgehensweise ist wie folgt: Zunächst werden die gemeinsamen Summenpfadmetriken aus den zusammengehörenden Zuständen des vorwärts- und rückwärtsgerichteten Trellis gebildet. Danach wird ein paralleler Vergleich der Summenpfadmetriken der Zustandspaare S_0 und S_1 sowie S_2 und S_3 durchgeführt. Dabei erhält man aus vier möglichen zwei relative Minima D_{01} und D_{23} . Im folgenden Taktschritt vergleicht man die beiden relativen Minima D_{01} und D_{23} und erhält dadurch das absolute Minimum D , auf dem der kürzeste Pfad liegt.

Für die ersten beiden Schritte können Addier-Vergleichs-Auswahleinheiten (ACSUs) aus Kapitel 4.2 mit der Modifikation, dass alle Eingänge eine Wortbreite von 8 bit aufweisen müssen, wiederverwendet werden. Dadurch lassen sich die gemeinsamen Summenpfadmetriken und die relativen Minima bilden. Die kleinere Summe und die Entscheidung, welcher der Zustände zu der kleineren Summe führt, werden in einem Register zwischengespeichert. Im nächsten Taktschritt werden die beiden relativen Minima mit einer sogenannten Vergleichs-Auswahleinheit (engl. Compare-Select-Unit, CSU) miteinander verglichen und hieraus das absolute Minimum bestimmt. Da die absolute Größe der kleinsten Summenpfadmetrik nicht von Bedeutung ist, muss sie auch nicht berechnet werden. Daher wird im zweiten Schritt keine vollständige ACSU, sondern eine abgespeckte CSU eingesetzt. Die CSU enthält nur einen linearen 8 bit Vergleich, der nur aus dem Übertragszweig eines Addierers mit sukzessiver Übertragskorrektur besteht. Abbildung 4.33 zeigt ein Blockschaltbild der realisierten SMU.

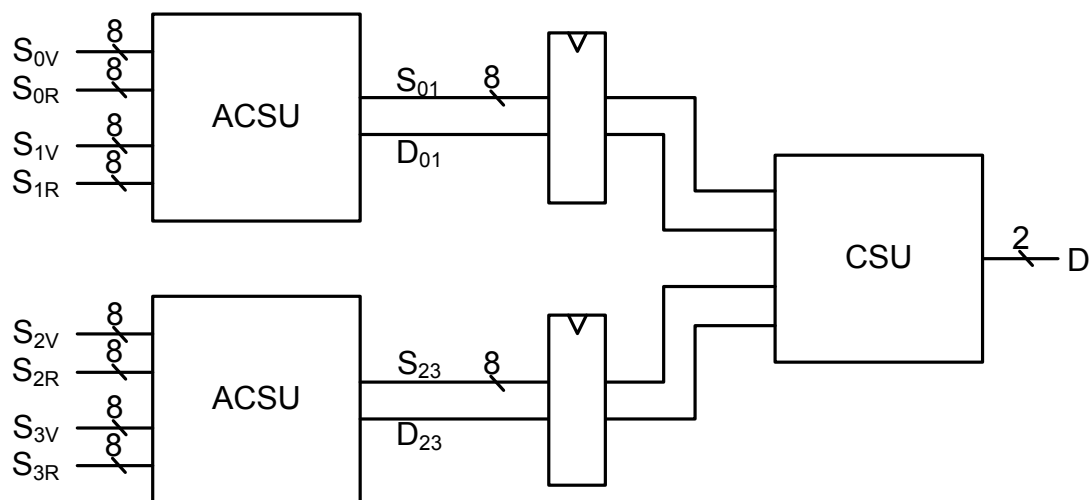


Abbildung 4.33: SMU, bestehend aus zwei ACSUs und einer CSU

Am Ausgang der CSU liegt die Zustandsadresse D des absoluten Minimums als 2 bit kodierter Wert an. Die Kodierung ist wie folgt:

- niederwertigeres Bit: 0: Kürzester Pfad entweder auf Zustand 0 oder 1
- 1: Kürzester Pfad entweder auf Zustand 2 oder 3
- höherwertigeres Bit: 0: Kürzester Pfad entweder auf Zustand 0 oder 2
- 1: Kürzester Pfad entweder auf Zustand 1 oder 3

Dadurch ist der Zustand, auf dem der kürzeste Pfad liegt eindeutig festgelegt. Die Kodierung ist Tabelle 4.17 zu entnehmen und entspricht der Binärdarstellung.

Tabelle 4.17: Kodierung des Entscheidungssignals D

D(1)	D(0)	Minimum auf Zustand
0	0	0
0	1	1
1	0	2
1	1	3

Bei Gleichheit der Summenpfadmetriken entscheiden sich die Komparatoren wie bei der im Kapitel 4.2 entworfenen ACSU für den Zustand mit der höheren Nummer. Das heißt, sind alle Metriken gleich, wird per Definition der Zustand 3 als der Zustand festgelegt, auf dem der kürzeste Pfad liegt.

Da die SMU zwei Taktzyklen für die Entscheidung benötigt, können sowohl die ACSU und die CSU unabhängig voneinander simuliert werden. Die Simulation enthält die kombinatorische Laufzeit der ACSU bzw. der CSU sowie die Verzögerungszeit des der ACSU und der CSU vorgeschalteten Registers. Die Rüstzeit des Registers bleibt unberücksichtigt. Jedoch kann basierend auf Tabelle 3.6 und aus der logischen Korrektheit der Simulationsergebnisse heraus argumentiert werden, dass die verbleibende Taktperiodendauer als Rüstzeit ausreicht. Die längsten Schaltvorgänge sind in Tabelle 4.18 dargestellt, woraus sich dann die Verzögerungszeiten nach Tabelle 4.19 ergeben.

Tabelle 4.18: Längster ACSU- und CSU-Schaltvorgang der SMU

	ACSU		CSU	
	Zeitpunkt t_0	Zeitpunkt t_1	Zeitpunkt t_0	Zeitpunkt t_1
A _{0,7..0}	0000.0000	0000.0010	0111.1111	0111.1111
B _{0,7..0}	0111.1111	0111.1111	0000.0000	0000.0001
A _{1,7..0}	0000.0000	0000.0011	0111.1111	0111.1111
B _{1,7..0}	0111.1111	0111.1111	0000.0000	0000.0011
Entscheidung	0111.1111	1000.0001	0111.1111	1000.0000

Der ACSU-Schaltvorgang der SMU ist im Wesentlichen derselbe, wie schon von der ACSU her bekannt (vgl. Tabelle 4.13 a), nur mit dem Unterschied, dass die Bits mit der Wertigkeit 7 und 6 mittels eines Volladdierers statt Halbaddierers verarbeitet werden.

Tabelle 4.19: Verzögerungszeiten der ACSU und CSU einschließlich Flipflops

Parameter	Verzögerungszeit ACSU	Verzögerungszeit CSU
Schaltplan, TT	381 ps	261 ps
RCTYP, TT	547 ps	347 ps
RCMAX, SS	748 ps	512 ps

Rein rechnerisch müsste sich als Verzögerungszeit die Summe aus ACSU (Tabelle 4.13 a) und Flipflop-Verzögerungszeit (Tabelle 3.6) ergeben.

$$t_{\text{SMU+REG}} = t_{\text{ACSU}} + t_{\text{C2O}} \quad (4.21)$$

Auf Schaltplanebene und unter Berücksichtigung von typischen Transistoren müsste sich für eine Verzögerungszeit von $t_{\text{SMU+REG}} = 352 \text{ ps} + 20,0 \text{ ps} = 372 \text{ ps}$ ergeben. Die durch Simulation ermittelte Verzögerungszeit ist um 9 ps größer, was über die komplett ausgebauten Volladdierer der Bitwertigkeiten 6 und 7 begründet wird. Gleiches gilt für die höheren Verzögerungszeiten der Layoutsimulationen.

Tabelle 4.20: Eckdaten des Layouts der SMU

Höhe der ACSU	110 μm
Höhe der CSU	16,5 μm
Breite der ACSU	26,5 μm
Breite der CSU	47,5 μm
Fläche	6614 μm^2
Anzahl Transistoren ACSU	892
Anzahl Transistoren CSU	216
Gesamtanzahl an Transistoren	2000

Die geometrischen Abmessungen der SMU sind so gewählt, dass sie genau zwischen zwei systolische Blockdekoder (vgl. Abbildung 2.12) passt und somit die Leitungslänge minimiert wird.

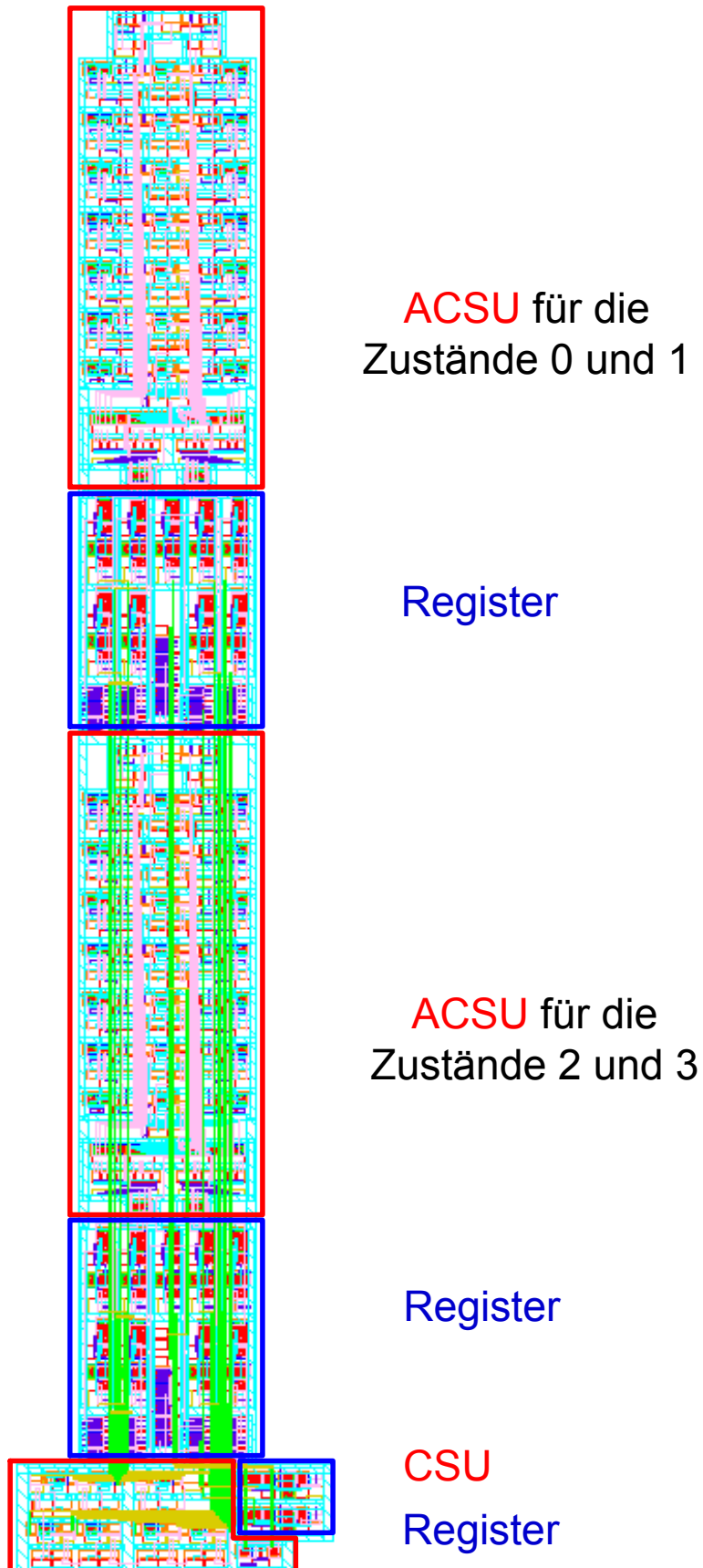


Abbildung 4.34: Layout der SMU

4.4 Pfadrückverfolgungseinheit

Mit Hilfe der Minimumauswahlschaltung (SMU), die in Kapitel 4.3 beschrieben wird, lässt sich ein Zustand ermitteln, der auf dem kürzesten Pfad liegt. Bei einem Blockdekodierverfahren möchte man mehrere Bits gleichzeitig dekodieren. Daher muss nun, ausgehend von der Entscheidung der SMU, der kürzeste Pfad rückverfolgt werden. Wie bereits in Kapitel 4.2.8 beschrieben, wird in jeder ACSU in einem Flipflop die Entscheidung, welcher von beiden möglichen Pfaden der wahrscheinlichere ist, gespeichert. Die Definition ist wie folgt: Weist der obere Pfad die kleineren Kosten auf, wird eine 0 gespeichert. Weist dagegen der untere Pfad die geringeren Kosten auf oder sind die Kosten auf dem oberen und unteren Pfad gleich groß, wird eine 1 gespeichert. Um nun zur Pfadrückverfolgung einen der vier Zustände im Trellis adressieren zu können, werden zwei Bits benötigt. Es reicht aber aus, in jedem Zustand nur ein Bit zu speichern, da aufgrund der Trellisstruktur von einem bestimmten Zustand aus nur zwei mögliche Zustände adressiert werden können. Abbildung 4.35 zeigt die Trellisstruktur für Vorwärts- und Rückwärtsdekodierung. Die jeweils fettgedruckte Zahl an den Zuständen entspricht der ACSU-Entscheidung. Zur Pfadrückverfolgung muss der Trellis entgegen der Pfeilrichtung durchlaufen werden. Dabei wird anhand Abbildung 4.35 schnell ersichtlich, dass das zweite Bit, das zur vollständigen Zustandsadressierung benötigt wird, immer fest ist. Außerdem ist zu erkennen, dass bei der Vorwärtsstruktur das Entscheidungsbit der ACSU das LSB ist und das feste Bit das MSB ist, während bei der Rückwärtsstruktur das MSB das Entscheidungsbit der ACSU ist und das LSB das feste Bit ist.

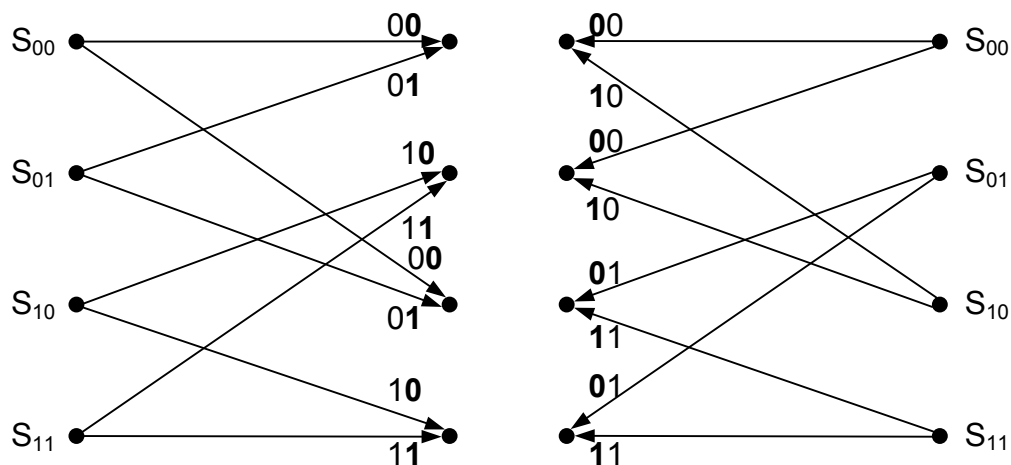


Abbildung 4.35: Trellisstruktur für Vorwärts- und Rückwärtsprozessierung zur Pfadrückverfolgung

Abbildung 4.36 zeigt einen Ausschnitt der Viterbi-Dekodierung mit realen Daten mit jeweils der halben Blocklänge, da nur die Hälfte der prozessierten Daten zur Dekodierung herangezogen werden kann (vgl. Kapitel 2.4.1.3). Im ersten Schritt werden die Zustände von außen zur Mitte hin durchlaufen. Die Pfadmetriken, die am Ausgang der ACSUs anliegen, sind über den

Zuständen eingetragen. Sind beide Blöcke durchlaufen, erfolgt die Ermittlung des kürzesten Pfades mit Hilfe der SMU. In diesem Beispiel entscheidet die SMU, dass der kürzeste Pfad (fettgedruckt) auf Zustand 2 (binär 10) liegt, da die Zahl 17 die kleinste der vier Pfadmetriken ist. Ausgehend von diesem Zustand, wird der Pfad, der zu genau diesem Zustand geführt hat, in umgekehrter Richtung durchlaufen. Diese Pfadrückverfolgung wird fachsprachlich „Trace-Back“ (TB) genannt.

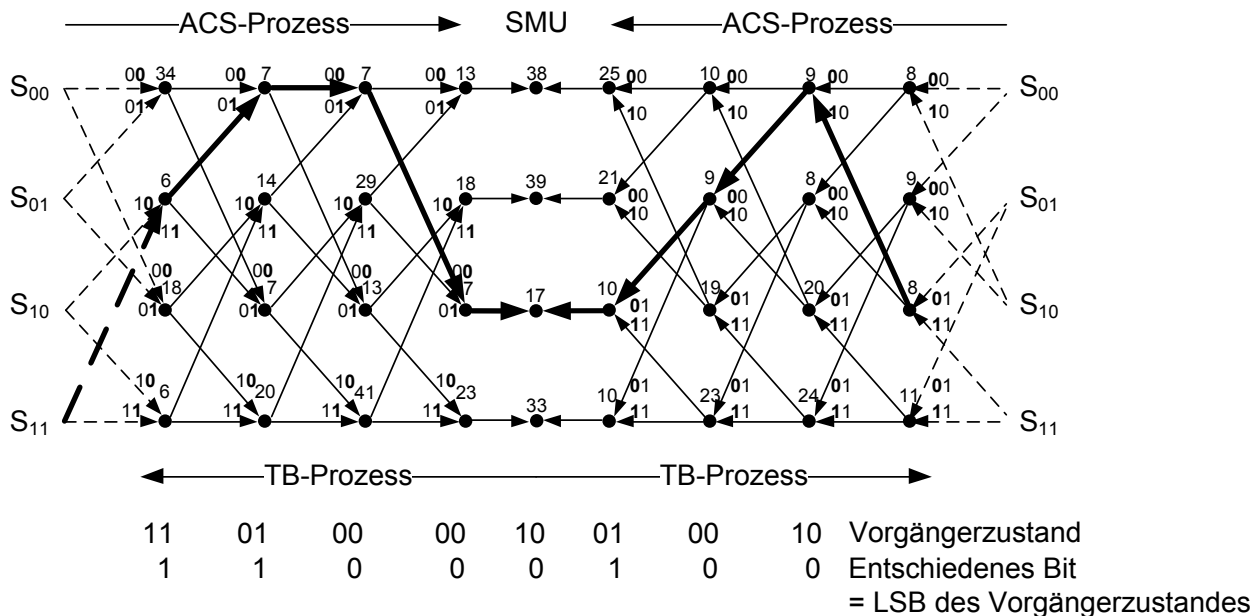


Abbildung 4.36: Beispiel eines ACS- und TB-Dekodierdurchlaufs

Die Adresse des Vorgängerzustands ist aufgrund der ACS-Entscheidungen eindeutig festgelegt. Die ACS-Entscheidung ist die fettgedruckte Ziffer der Adresse. Das entschiedene bzw. entzerrte Bit ist das LSB des Vorgängerzustandes. Der Zustand der SMU ist kein vollwertiger Zustand, da in ihn keine neuen Zweigmetriken einfließen. Er dient nur zur Verknüpfung der in Vorwärts- und Rückwärtsrichtung verlaufenden Trellisblöcke.

Die TBU besteht aus einem 4:1-Multiplexer mit jeweils zwei Eingängen [63] und einem 2 bit-Register zum Zwischenspeichern des Ausgangswertes. Der 4:1-Multiplexer ist zweistufig aufgebaut (siehe Abbildung 4.37). Das heißt, dass das höherwertigere Entscheidungsbit der vorhergehenden TBU-Stufe eine Vorauswahl trifft, ob das zu dekodierende Bit bzw. die Adresse der Folge-ACSU von der aktuellen Trellisstufe vom Zustand 0 oder 1 oder vom Zustand 2 oder 3 kommt. Das niederwertigere Entscheidungsbit der vorhergehenden TBU-Stufe entscheidet dann, welches Bit bzw. Adresse aus der Vorauswahl genommen wird. Die beiden Ergebnisbits werden in zwei Flipflops zwischengespeichert und mit dem Entscheidungseingang der folgenden TBU verknüpft. Das LSB der entschiedenen Adresse wird zusätzlich abgezweigt und durch weitere Bitversatzregister (vgl. Abbildung 2.12) verzögert und dann am Ausgang des DSPs als dekodiertes Ergebnisbit ausgegeben.

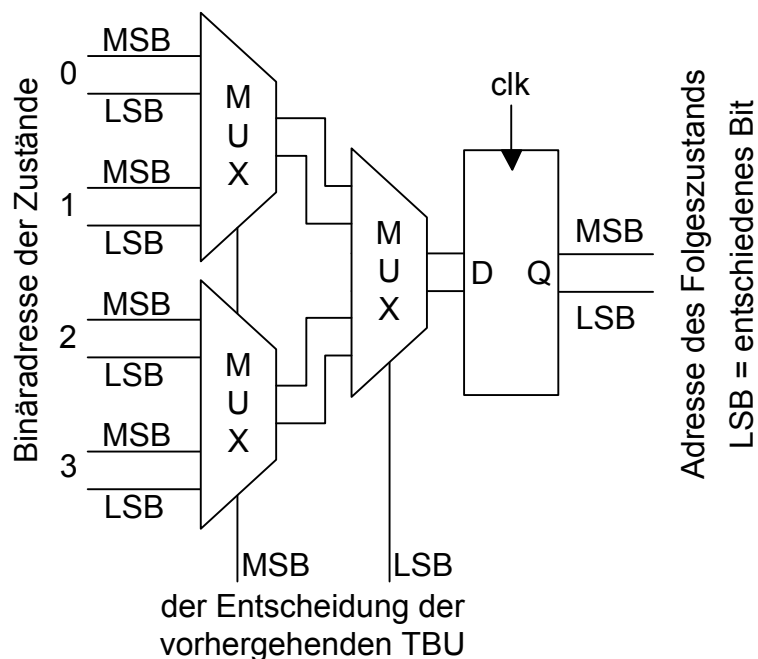


Abbildung 4.37: Einzelne allgemeine TBU

Für die Adressierung des vorhergehenden Zustandes müssen für den Vorwärts- bzw. Rückwärts-Trellisdurchlauf die Eingänge der TBU entsprechend Abbildung 4.38 vorbelegt werden. Da jeweils vier der Eingänge eine feste Größe aufweisen, hätte man spezielle vereinfachte Multiplexer für den Vorwärts- und Rückwärtsdurchlauf erstellen können. Aus Gründen der Wiederverwendbarkeit und der nur geringen Flächeneinsparnis hat man sich für einen allgemeinen Multiplexer entschieden.

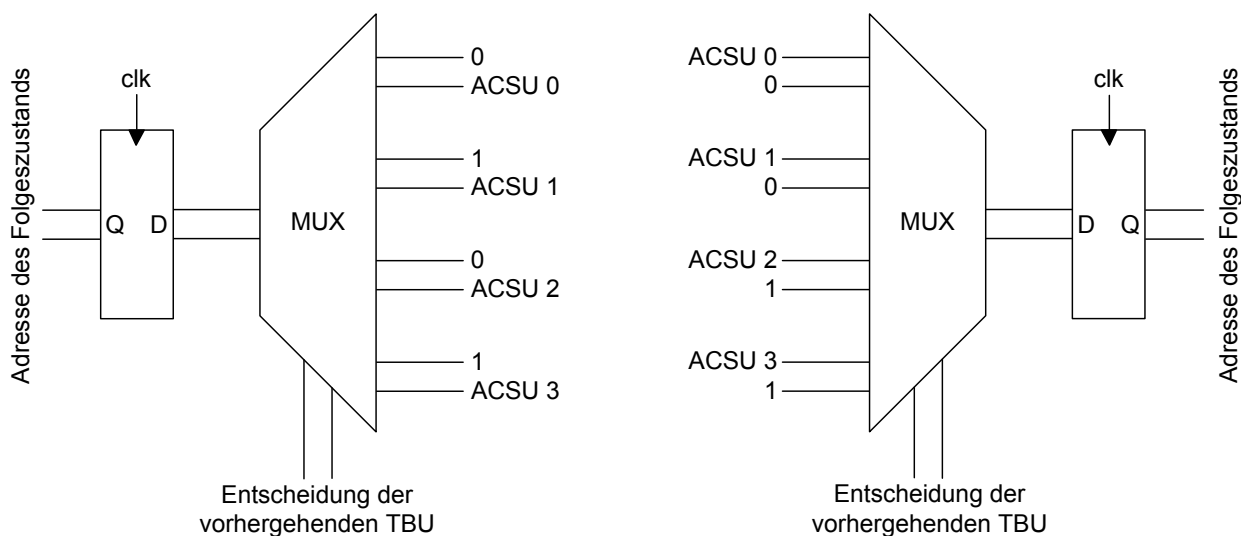


Abbildung 4.38: Für Vorwärts- (links) und Rückwärts- (rechts) Trellisdurchlauf konfigurierte TBUs

Das Layout der TBU ist in Abbildung 4.39 dargestellt, und die Eckdaten des Layouts sind in Tabelle 4.21 zusammengefasst.

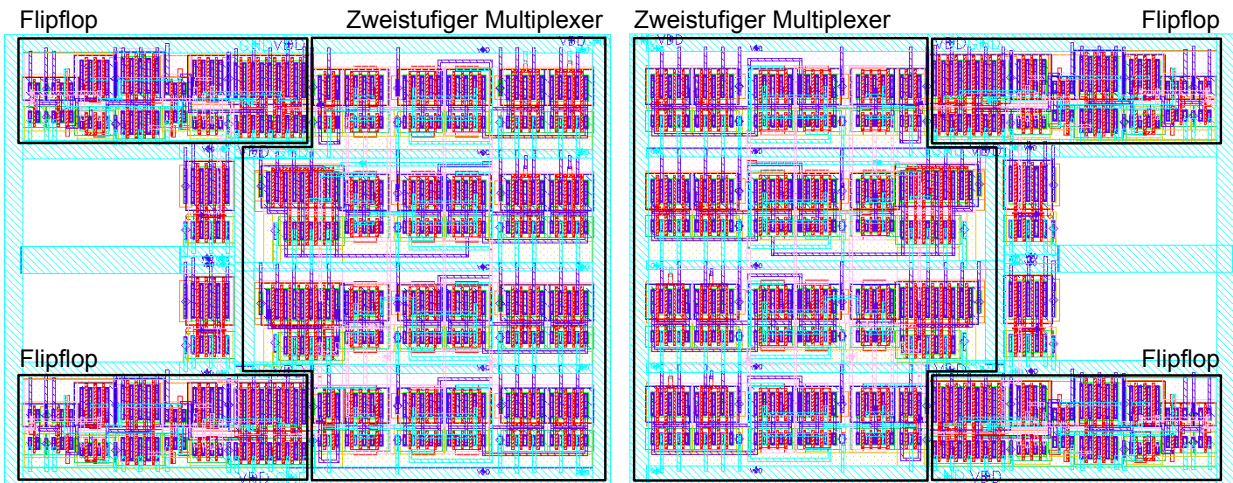


Abbildung 4.39: Layout der TBUs für den Vorwärts- (links) und Rückwärts- (rechts) Trellisdurchlauf

Tabelle 4.21: Eckdaten des Layouts der TBU

Höhe	20 μm
Breite	27 μm
Fläche	540 μm^2
Anzahl Transistoren	92

4.5 Eingangsdemultiplexer

4.5.1 Hintergrundüberlegungen zum Demultiplexer

Die in den vorhergehenden Teilkapiteln vorgestellten Schaltungsteile des Viterbi-Entzerrers zeigen, dass die maximal mögliche Taktfrequenz auf ca. 1,33 GHz begrenzt ist. Daraus resultiert, dass bei einer Eingangsdatenrate von rund 43 Gbit/s eine 32-fache Parallelverarbeitung erforderlich ist. Da die Eingangsdaten 3 bit quantisiert vorliegen, würde die Eingangsschnittstelle 96 Anschlüsse aufweisen. Im Layout werden die Anschlüsse in Form von Pads ausgeführt. Bei einer Padbreite von 75 μm und einem Padabstand von 25 μm hätte dies bei einreihiger Padanordnung eine Kantenlänge des Chips von 9,6 mm zur Folge. Dadurch würde sehr viel teure Chipfläche ungenutzt bleiben. Außerdem wären sehr lange Signalleitungen vonnöten, womit man sich Synchronisierungsprobleme aufgrund von Laufzeitunterschieden und notwendiger Regenerationsinverter einhandelt. Es wird eine einreihige Padanordnung angestrebt, damit die

Bonddrahte alle gleich lang werden. Um die Schnittstelle schlanker zu gestalten und dadurch Chipflache einzusparen, werden die Eingangsdaten mit hoherer Geschwindigkeit ubertragen und dann auf dem Viterbi-Entzerrer-Chip parallelisiert.

Zur Realisierung bieten sich einerseits eine Schieberegisterstruktur und andererseits eine Baumstruktur an. Die Schieberegisterstruktur hatte zwar einen geringeren Transistoraufwand und folglich einen geringeren Strom- und Flachenbedarf; dennoch fallt die Wahl auf die Baumstruktur, da diese eine hohere Geschwindigkeit [64] gegenuber der Schieberegisterstruktur besitzt. Prinzipbedingt sind in binaren Baumen nur Teilerfaktoren mit Zweierpotenzen moglich. Wendet man einen dreistufigen binaren Baum an, erhalt man einen Teilerfaktor von acht. Daraus folgt, dass die Eingangsdaten die achtfache Datenrate aufweisen mussen als das interne Viterbi-Rechenwerk. Folglich reduziert sich der Schnittstellenaufwand auf ein Achtel. Anstelle von 96 Pads waren bei einem unipolaren Demultiplexer nur noch 12 Pads notwendig. Bei Datenraten von 10,7 Gbit/s sind unipolare Ubertragungen aufgrund von Storeinkopplungen nicht zu empfehlen. Daher wird auf eine differenzielle Eingangsschnittstelle zuruckgegriffen. Die Padanzahl erhohet sich auf 24 zuzuglich zwei Takteingange. Sieht man zwischen jedem differenziellen Eingangspaar ein Massepad vor, bedeutet dies eine Gesamtanzahl von 40 Pads, was einer Kantenlange von rund 4 mm entspricht. Wie spater in Kapitel 4.7 ersichtlich wird, entspricht dies in etwa der Breite des gesamten Viterbi-Entzerrers, so dass keine Chipflache verschenkt wird.

4.5.2 Aufbau eines CML-Demultiplexers

Ein Demultiplexer auf Basis eines binaren Baumes parallelisiert ein Eingangssignal mit Hilfe eines Auswahleinganges auf zwei Ausgangskanale. Dabei halbiert sich die Datenanderungsfrequenz pro Ausgangskanal gegenuber dem Eingang. Abbildung 4.40 zeigt eine Funf-Latch-Realisierung eines Demultiplexers [64]. Das Auswahlsignal ist das Taktsignal clk . Alle Leitungen sind in der Schaltung differenziell ausgefuhrt. Die Schaltungsbeschreibung erfolgt des besseren Verstandnisses halber mit einfacher Leitungsfuhung.

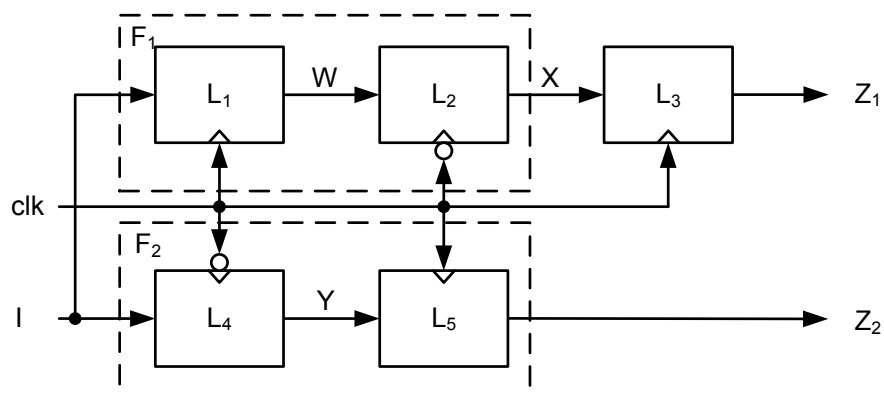


Abbildung 4.40: Funf-Latch-Realisierung eines Demultiplexers

Das Eingangssignal I wird auf zwei Latch-Ketten aufgeteilt, wobei die obere Serienschaltung aus drei und die untere Serienschaltung aus zwei Latches besteht. Das Auswahlsignal entspricht einem Taktsignal, das die gleiche Frequenz hat wie die maximale Datenänderungsfrequenz des Eingangssignals. Das heißt, dass es sich auf das Eingangssignal bezogen, um einen Halbratentakt handelt, während es sich auf das Ausgangssignal bezogen, um einen Vollratentakt handelt. Dies ist Voraussetzung für die Funktion des Demultiplexers. Das Eingangssignal wird zunächst von den beiden Latches L_1 und L_4 abgetastet, welche von einem zueinander gegenphasigen Taktsignal angesteuert werden. Da sich bei einem Latch Folge- und Haltephase zyklisch abwechseln, wird aufgrund des Halbratentaktes beim Latch L_1 immer nur der Logikpegel des Eingangssignals berücksichtigt, solange die Taktperiode einen High-Pegel aufweist. Das Latch L_4 , welches mit dem invertierten Takt gegenüber dem Latch L_1 angesteuert wird, berücksichtigt den Logikpegel des Eingangssignals, solange das Taktsignal einen Low-Pegel aufweist. Dadurch ist das Demultiplexerverhalten gegeben. In Abbildung 4.41 (links) wird dieses Verhalten ersichtlich.

Auf das Latch L_1 folgt das Latch L_2 und auf L_4 folgt L_5 . Diese Serienschaltung der beiden Latches ergeben die Flipflops F_1 und F_2 , sodass zu keinem Zeitpunkt einer der beiden Pfade transparent ist. Die parallelisierten Bits am Zwischenknoten X und am Ausgang Z_2 weisen einen zeitlichen Versatz um eine halbe Taktperiode auf. Das Latch L_3 verzögert das Signal am Knoten X nochmals um eine halbe Taktperiode, damit beide Ausgänge Z_1 und Z_2 ein phasenrichtiges Signal bezüglich des Taktes clk liefern.

Beim Vergleich der Abbildung 4.41 wird ersichtlich, dass die Reihenfolge der parallelisierten Bits am Ausgang unabhängig von der Phasenlage des Taktsignals ist. Am Ausgang Z_1 liegt, bezogen auf eine Taktperiode, also immer das zeitlich zuerst gesendete Bit an.

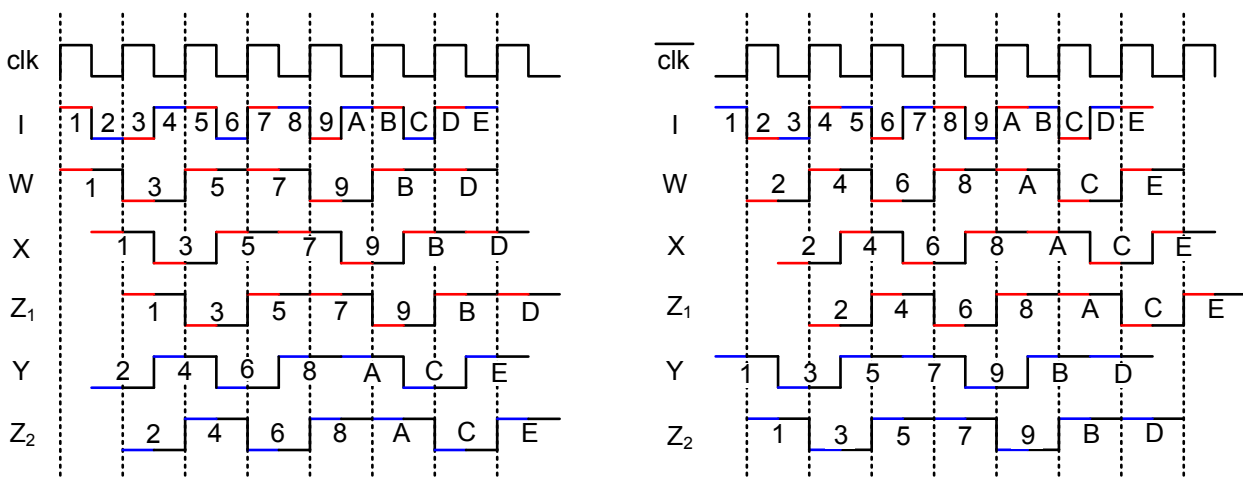


Abbildung 4.41: Signalverläufe am Demultiplexer bei
(links) nicht-invertiertem und (rechts) invertiertem Taktsignal

4.5.3 Takteiler

Der Demultiplexer soll den Datenstrom im Verhältnis 1:8 parallelisieren, damit das Viterbi-Rechenwerk mit einer Taktfrequenz von 1,3375 GHz betrieben werden kann. Der Eingangsdatenstrom, der vom AD-Wandler-Ausgang geliefert wird, hat eine Datenrate von $4 \times 10,7$ Gbit/s. Parallel dazu liefert der AD-Wandler ein zu den Daten synchrones Taktsignal mit einer Frequenz von 5,35 GHz. Dies entspricht einem Halbratentakt. Die maximale Datenänderungsfrequenz (01-Folge) ist gleich der Taktfrequenz. Der Vorteil eines Halbratentaktes besteht darin, dass die Anforderungen an die Taktleitungen und Bonddrähte des externen Aufbaus geringer sind. Nachteile gibt es in einer CML-Schaltung nicht, da aufgrund der differenziellen Leitungsführung beide Taktflanken gleichzeitig zur Verfügung stehen.

Für den 1:8-Demultiplexer muss der Takt also nur zweimal in Folge halbiert werden, um den Vollratentakt des in statischer CMOS-Logik realisierten Viterbi-Rechenwerks zu erzeugen. Der Takteiler erzeugt für die zweite Demultiplexer-Stufe eine Taktfrequenz von 2,675 GHz und für die dritte Stufe eine Frequenz von 1,3375 GHz. Die Frequenz der dritten Stufe ist der Halbratentakt der Demultiplexer-Eingangsdaten. Da die Datenrate in der dritten Stufe geteilt wird, kann eine Phase des Eingangstaktes der dritten Stufe nach CML-CMOS-Pegel-Konversion direkt als Vollratentakt für die nachfolgende CMOS-Schaltung verwendet werden.

Abbildung 4.42 zeigt einen Takteiler, der die eingehende Taktfrequenz halbiert. Der Takteiler besteht aus zwei in Serie geschalteten und über Kreuz rückgekoppelten CML-Latches.

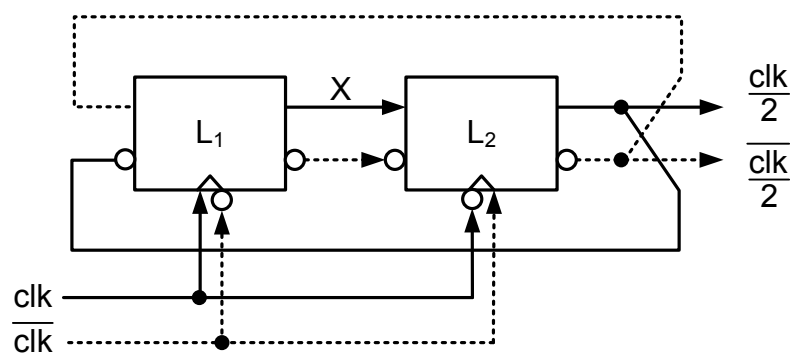


Abbildung 4.42: Takteiler

Zwei in Serie geschaltete Latches bilden ein Flipflop. Der Einfachheit halber erfolgt die Funktionsbeschreibung der Takteilerschaltung nur für den nicht-invertierten Signalfluss. Für den invertierten Signalfluss (gestrichelt gezeichnet) gelten entsprechend die invertierten Logikpegel. Negierte Ein- und Ausgänge sind durch einen Kreis gekennzeichnet.

Liegt am Takteingang $\text{clk} = L$ an, befindet sich das Latch L_1 in der Haltephase, während sich das Latch L_2 in der Folgephase befindet. Nimmt man an, dass am Zwischenknoten X ein Low-Pegel anliegt, wird dieser an den Ausgang $\text{clk}/2$ weitergeleitet. Wechselt der Takt nach $\text{clk} = H$, wird der Low-Pegel am Ausgang gehalten, da sich das Latch L_2 jetzt in der Haltephase befindet. Das

Latch L_1 befindet sich dagegen in der Folgephase. Am Knoten X liegt aufgrund der kreuzgekoppelten Latches ein High-Pegel an. Geht nun das Latch L_2 aufgrund eines Taktwechsels von clk von H nach L wieder in die Folgephase über, wird $X = H$ an den Ausgang $clk/2$ durchgeschaltet. Der Ausgang $clk/2$ ändert sich nur bei einer fallenden Flanke von clk . Die Signalfrequenz des Ausgangs $clk/2$ ist also nur halb so groß wie die Signalfrequenz des Eingangs clk . Die Taktteilerfunktion der Schaltung ist damit erfüllt.

Da der Zustand des Knotens X nicht durch eine äußere Beschaltung bestimmt werden kann, kann man nicht sagen, ob am Ausgang zuerst ein Low- oder High-Pegel anliegt. Im ungünstigsten Fall befindet sich das Taktteiler-Flipflop in einem metastabilen Zustand. Bei Demultiplexern, welche aus Latches bestehen (vgl. Abbildung 4.40), kann also nicht vorhergesagt werden, an welchem seiner beiden Ausgänge aufeinander folgende Symbole ausgegeben werden. Da die Symbole am Viterbi-Entzerrer-Eingang aufgrund des dispersiven Kanals zeitlich korreliert sind, ist die Reihenfolge der zu dekodierenden Symbole von entscheidender Bedeutung. In Kapitel 4.5.4 wird dieses Problem näher betrachtet.

4.5.4 Vom 1:2- zum 1:8-Demultiplexer

Die Rückkopplung des Taktteilers ist eine in sich geschlossene Schleife. Da die beiden Latches völlig symmetrisch aufgebaut sind, kann nicht vorhergesagt werden, an welchem der beiden Ausgänge zuerst eine steigende oder fallende Flanke anliegt. Wie in Kapitel 4.5.2 ersichtlich wird, hängt aber die Reihenfolge der parallelisierten Ausgangsbits von der Phasenlage des Taktes ab. Das Problem soll an Hand eines 1:4-Demultiplexers, wie er in Abbildung 4.43 dargestellt ist, verdeutlicht werden.

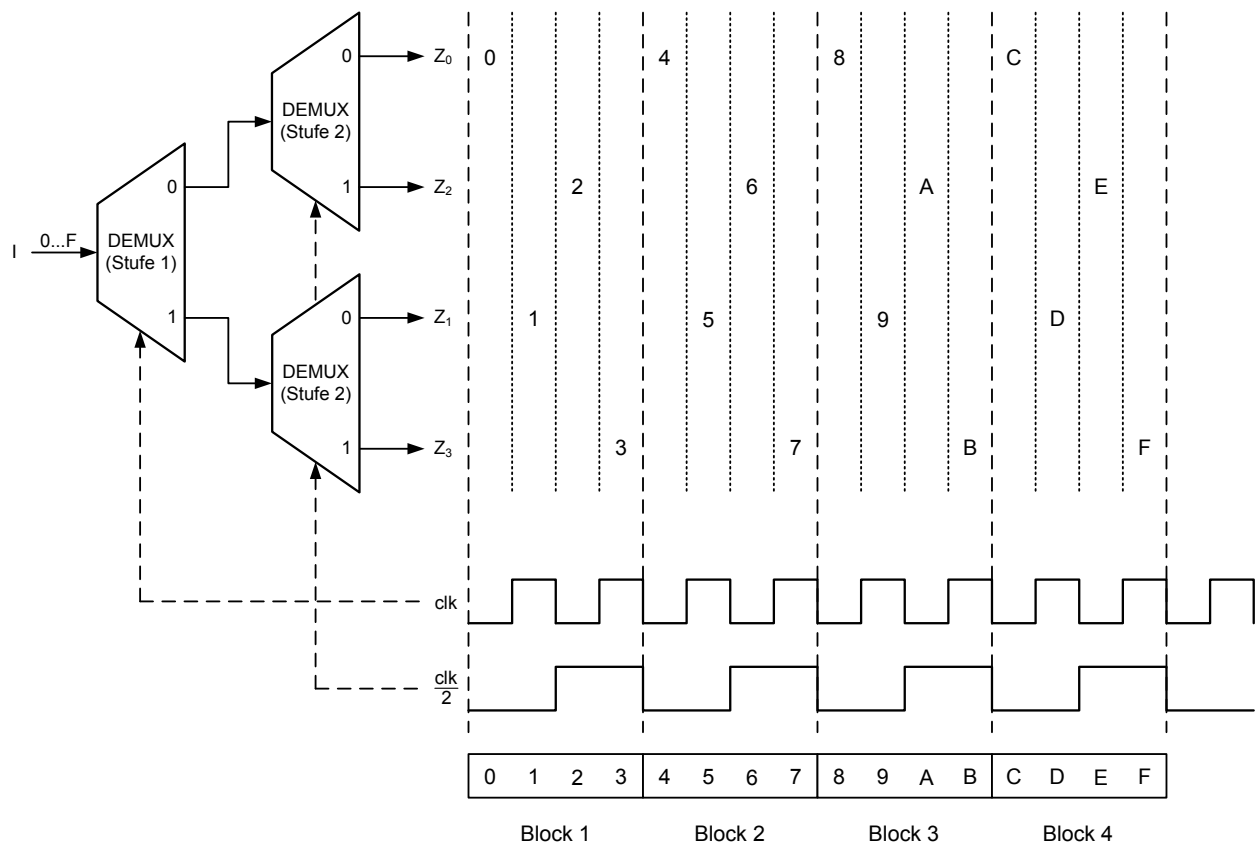


Abbildung 4.43: Reihenfolge Fall 1

In diesem Beispiel wird ein serieller Bitstrom parallelisiert, der aus den Symbolen 0..F besteht und am Eingang I anliegt. Der Demultiplexer der Stufe 1 wird mit dem Takt der höchsten Taktfrequenz f_{clk} angesteuert; die beiden Demultiplexer der Stufe 2 entsprechend mit dem durch einen Takteiler halbierten Takt mit der Frequenz $f_{clk}/2$. Weist das Taktsignal einen Low-Pegel auf, wird das Eingangssymbol an den Demultiplexerausgang 0 weitergeleitet, bei einem High-Pegel liegt es am Ausgang 1 an. Nach zwei Demultiplexerstufen ergibt sich in Abhängigkeit der eingezeichneten Taktphasen an den Ausgängen Z_0 bis Z_3 die zeitliche Reihenfolge der parallelisierten Symbole nach Abbildung 4.43. Am Ende einer Taktperiode, welche durch eine fallende Flanke des Taktes $clk/2$ definiert ist, liegt am Ausgang ein Viererblock mit den parallelisierten Symbolen an. Die Verdrahtung ist so gewählt, dass am Ausgang Z_0 das zeitlich zuerst gesendete und am Ausgang Z_3 das zuletzt gesendete Symbol eines Viererblockes parallel anliegt. Nach vier parallelisierten Symbolen wiederholt sich das Spiel aufgrund der Periodizität und Synchronizität der beiden Takte.

Wie sieht es aber in dem zweiten Fall aus, bei dem der geteilte Takt $clk/2$ mit einer steigenden Taktflanke beginnt? In jenem Fall gibt der Demultiplexer das erste Symbol am Ausgang 1 aus. Beim Vierfachdemultiplexer entspricht die Reihenfolge der Ausgangssymbole, wie aus Abbildung 4.44 hervorgeht, dann den Ausgängen Z_2, Z_3, Z_0, Z_1 . Die Folge wäre, dass die für den Reihenfolgefall 1 gültige Verdrahtung falsch wäre.

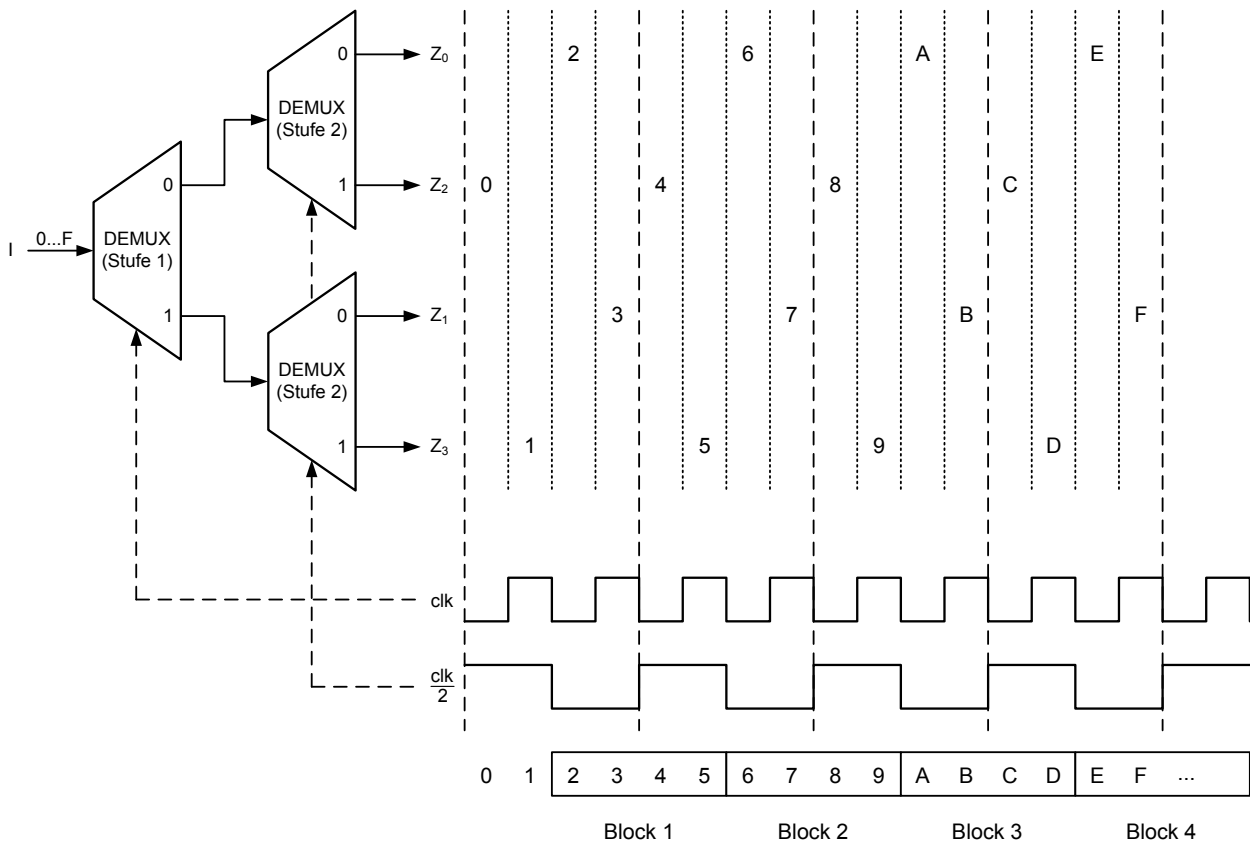


Abbildung 4.44: Reihenfolge Fall 2

Wie bereits in Kapitel 4.5.3 erwähnt, wird das Viterbi-Rechenwerk mit dem Takt $clk/2$ getaktet, dessen Eingangsflipflops auf die fallende Taktflanke sensitiv sind. Das heißt, ein Viererblock beginnt und endet auch hier wieder mit einer fallenden Taktflanke von $clk/2$. Zeichnet man den Viererblock immer mit seinen Grenzen zur fallenden Taktflanke von $clk/2$ in den parallelisierten Datenstrom ein, erkennt man, dass die Reihenfolge der parallel anliegenden Symbole doch wieder korrekt ist, mit dem Unterschied, dass die ersten beiden Symbole 0 und 1 unberücksichtigt bleiben. Das bedeutet, dass die parallel ausgegebenen Daten erst mit Symbol 2 beginnen. Ab hier ist die Verdrahtungsfolge Z_0, Z_1, Z_2, Z_3 wieder gegeben. Da man den Datenstrom am Eingang als unendlich lang ansehen kann, entsteht kein Nachteil, wenn der Dekodiervorgang des Viterbi-Rechenwerks erst ein paar Takte später beginnt. Bei der Inbetriebnahme einer optischen Übertragungsstrecke werden zu Beginn sowieso erst Trainingssequenzen und noch keine Nutzdaten übermittelt, so dass es zu keinem Datenverlust kommt.

Der Vollständigkeit halber müsste man die Untersuchung auch für die beiden Fälle des schnellen Taktes clk durchführen. Darauf kann aber mit dem Vorwissen aus dem 1:2-Demultiplexer verzichtet werden.

Gleiches gilt dann auch für den 1:8-Demultiplexer. Daher wird auf eine weitere anschauliche Herleitung verzichtet.

Damit ist gezeigt, dass man für die Demultiplexerschaltung keinen separaten Eingang vorsehen muss, um eine bestimmte Reihenfolge der parallelisierten Symbole sicherzustellen.

Der 1:8-Demultiplexer, der als binärer Baum realisiert wird, ist in Abbildung 4.45 dargestellt.

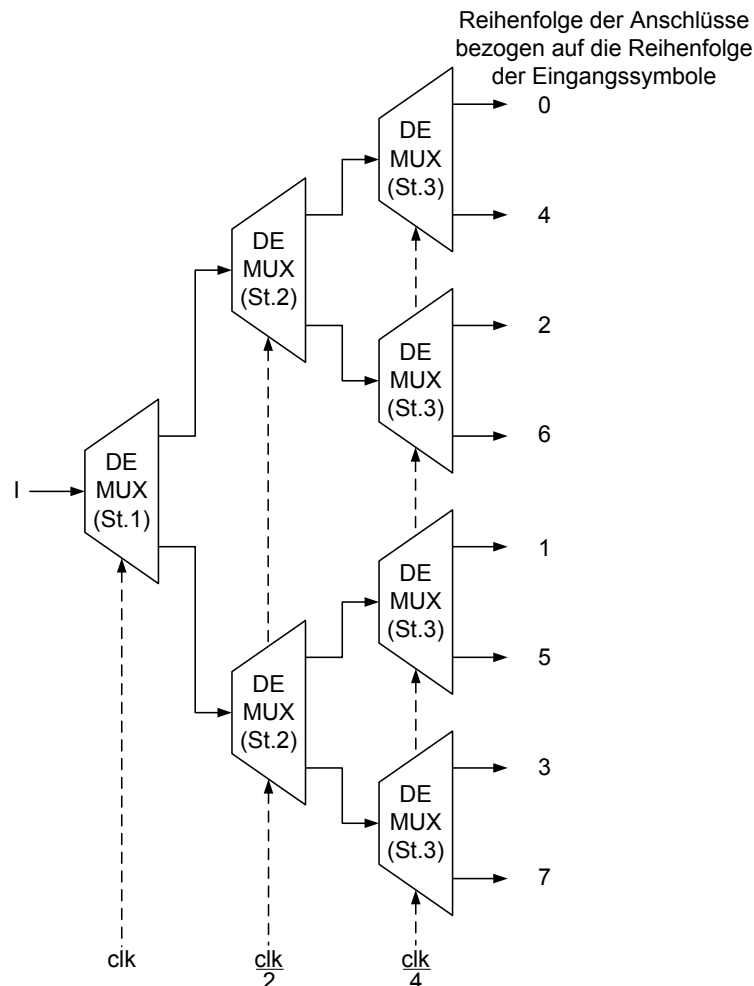


Abbildung 4.45: 1:8-Demultiplexer als binärer Baum

Die Bezeichnung der ausgangsseitigen Anschlüsse ergibt sich unter der Randbedingung, dass bei einem Low-Pegel des Taktsignals das Eingangssymbol I am oberen Ausgang und bei einem High-Pegel des Taktsignals das Eingangssymbol am unteren Ausgang eines 1:2-Demultiplexers ausgegeben wird. Die Taktfrequenzen und Datenraten an den Ein- und Ausgängen sind in Tabelle 4.22 angegeben.

Tabelle 4.22: Taktfrequenzen und Datenraten an den einzelnen Demultiplexern

Demultiplexerstufe	Taktfrequenz	Eingangsdatenrate	Ausgangsdatenrate
1	5,35 GHz	10,7 Gbit/s	5,35 Gbit/s
2	2,675 GHz	5,35 Gbit/s	2,675 Gbit/s
3	1,3375 GHz	2,675 Gbit/s	1,3375 Gbit/s

Für den Betrieb des 1:8-Demultiplexers sind außer den 1:2-Demultiplexern noch weitere Elemente nötig. Abbildung 4.46 zeigt den Schaltplan des vollständigen 1:8-Demultiplexers. Zur Verteilung der Taktsignale sind Takttreiberschaltungen erforderlich. Die Ausgänge der parallelisierten Signale werden von differentiellen CML-Pegeln in CMOS-Pegel gewandelt. Weiterhin sind individuell an die Demultiplexerposition auf dem Chip angepasste Verzögerungsglieder notwendig, damit an allen zwölf 1:8-Demultiplexern die Eingangssignale gleichzeitig anliegen. Dadurch ist der Betrieb von der Taktfrequenz unabhängig, da ein zeitlich definierter Signal-Takt-Abstand besteht. Für Details wird auf [64] verwiesen. Außerdem wird für jeden 1:8-Demultiplexerblock eine lokale Referenzspannungsquelle zum Einstellen des Bias-Stroms vorgesehen, damit es über den ganzen Chip verteilt nicht zu Spannungsabfällen und daraus resultierenden unterschiedlichen Strömen der CML-Bausteine kommt.

Der 1:8-Demultiplexer besteht in seiner ersten Stufe aus einem 1:2-Demultiplexer, welcher aus 1 mA-Latches aufgebaut ist. Da die zeitlichen Anforderungen in der zweiten und dritten Stufe geringer sind, können, um Strom zu sparen, Demultiplexer mit 0,5 mA-Latches eingesetzt werden. Die Verzögerungsglieder sind mit 0,25 mA-CML-Invertern aufgebaut. Diese weisen einerseits die höchste Verzögerungszeit und den geringsten Stromverbrauch auf.

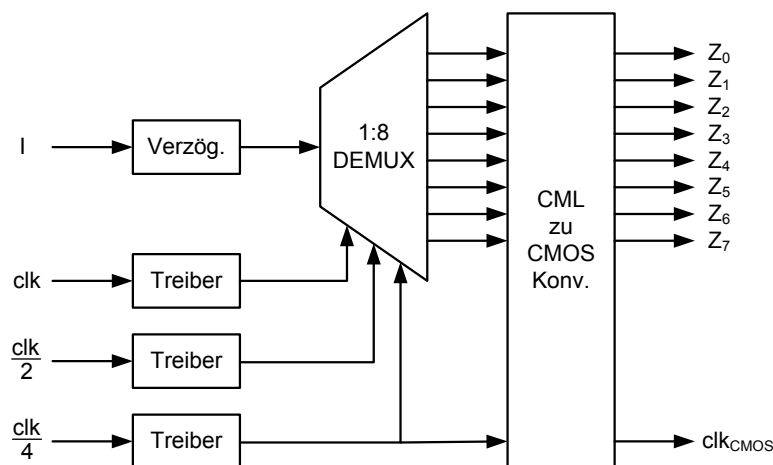


Abbildung 4.46: Schaltplan des 1:8 Demultiplexerblocks

Das Layout eines 1:8-Demultiplexers einschließlich CML-CMOS-Konverter, Treiber-, Verzögerungsschaltungen und einer Schaltung zur Arbeitspunkteinstellung ist in Abbildung 4.47 dargestellt.

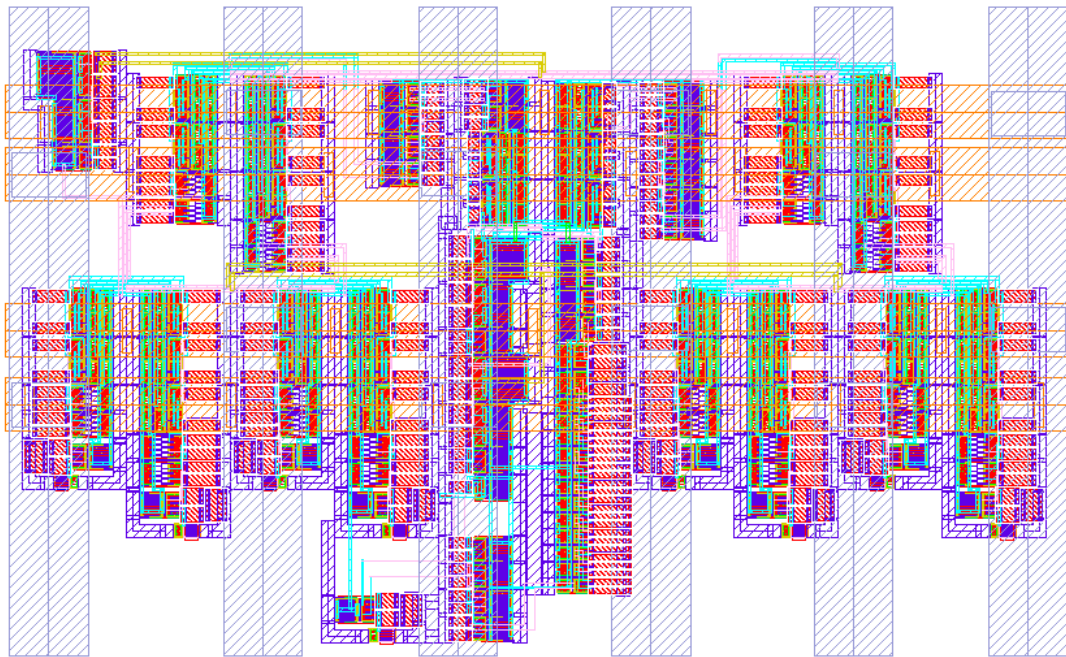


Abbildung 4.47: Layout eines 1:8-Demultiplexerblocks

Tabelle 4.23 fasst die Eckdaten des 1:8-Demultiplexerblocks zusammen.

Tabelle 4.23: Eckdaten des 1:8-Demultiplexerblocks

Höhe	100 μm
Breite	163 μm
Fläche	16.300 μm^2
Anzahl Transistoren	493
Stromaufnahme	51 mA

4.5.5 Takteilung und Taktverteilung

Das Taktsignal des Viterbi-Chips wird an einem zentralen Punkt differenziell vom vorgeschalteten AD-Wandler übergeben. Da die CML-Demultiplexer und das CMOS-Viterbi-Rechenwerk auf drei verschiedenen Taktebenen arbeiten, müssen die drei Taktfrequenzen (vgl. Tabelle 4.22) aus dem Eingangstakt abgeleitet und an die entsprechenden Schaltungselemente weitergeleitet werden.

Damit die Reihenfolge der parallelisierten Signale bei allen zwölf Demultiplexern stimmt, müssen alle Demultiplexer mit der gleichen Taktphase auf allen Taktebenen versorgt werden. Daher wird ein zentraler Takteiler, der alle drei Taktfrequenzen zur Verfügung stellt, entworfen. Das Layout des zentralen Takteilers ist in Abbildung 4.48 dargestellt.

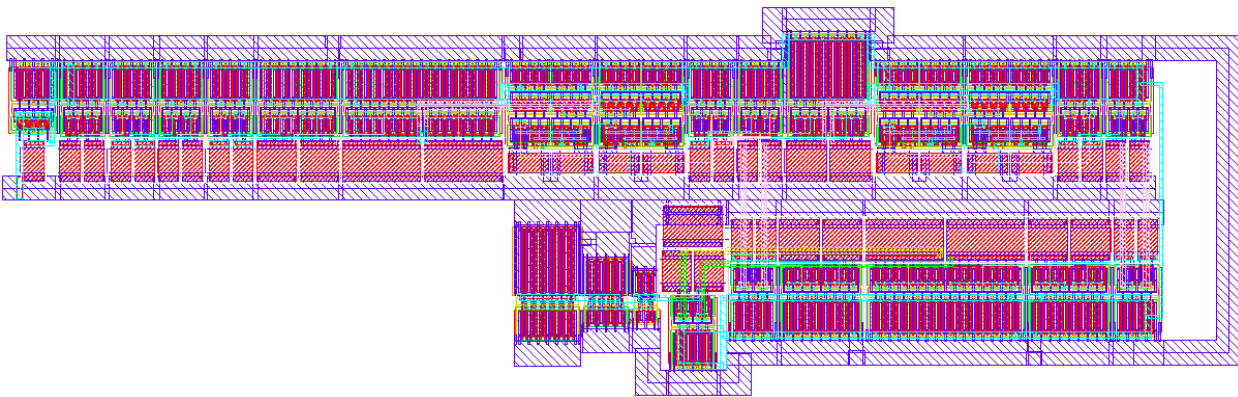


Abbildung 4.48: Layout des zentralen Taktteilers

Die Eckdaten des zentralen Taktteilers sind in Tabelle 4.24 wiedergegeben.

Tabelle 4.24: Eckdaten des zentralen Taktteilers

Höhe	101 μm
Breite	32 μm
Fläche	3.232 μm^2
Anzahl Transistoren	87
Stromaufnahme	35 mA

Es muss sichergestellt sein, dass an allen Demultiplexern das Taktsignal ohne Phasenversatz ankommt. Um dies zu erreichen, wird eine baumförmige Taktverteilstruktur angestrebt, so dass die Taktleitungen einer Taktebene zu allen Demultiplexern die gleiche Länge aufweisen. Aufgrund der langen zu erwartenden Leitungslänge von ca. 2 mm von der Chipmitte bis an den Rand müssen Takttreiberstufen in das Verteilnetzwerk eingefügt werden, um Dämpfungsverluste zu kompensieren. Weiterhin muss die Spannungsversorgung entsprechend ausgelegt werden, damit es aufgrund der hohen Stromaufnahme des gesamten Demultiplexers zu keinem Spannungsabfall auf dem Spannungsversorgungsnetz kommt. Die Auswirkung des Spannungsabfalls auf die Pegel der Taktsignale kann in Abbildung 4.49 betrachtet werden. Der zentrale Taktteiler stellt die drei Taktebenen auch den Multiplexern, die am Ausgang des Viterbi-Entzerrers benötigt werden, zur Verfügung. Daher werden die differenziellen Taktleitungen auch quer über den Chip verlegt. Ein ähnlich aufgebautes Taktverteilnetzwerk auf der Ausgangsseite des Viterbi-Entzerrers schließt die einzelnen Taktebenen dann an den Multiplexern an.

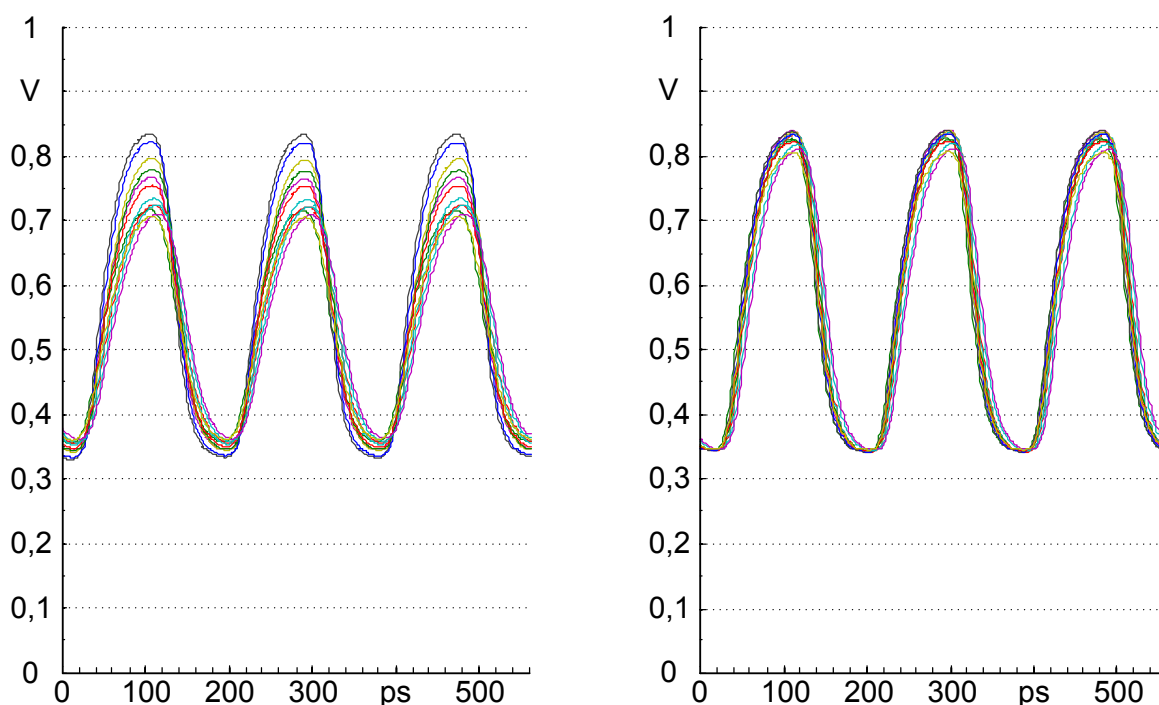


Abbildung 4.49: Pegel der Taktsignale

(links) mit zentraler Spannungsversorgung und (rechts) mit verteilter Spannungsversorgung

Die linke Abbildung 4.49 zeigt die Spannungspegel auf den Taktleitungen bei zentralem Versorgungsspannungsanschluss. Man erkennt, dass auf den Leitungen, die nahe am Versorgungsspannungsanschluss liegen, beinahe die vollen Logikpegel erreicht werden, während die Taktsignale der äußeren ca. 2 mm entfernten Demultiplexer mit deutlich geringeren Pegeln versorgt werden. Details sind in Tabelle 4.25 angegeben.

Tabelle 4.25: Spannungspegel der Taktsignale im Taktnetzwerk

	a) zentrale Spannungsversorgung			b) verteilte Spannungsversorgung		
	U_H	U_L	U_{Hub}	U_H	U_L	U_{Hub}
innen	836 mV	334 mV	502 mV	838 mV	341 mV	497 mV
außen	709 mV	361 mV	348 mV	804 mV	347 mV	457 mV
U_{Deg}	127 mV	-27 mV	154 mV	34 mV	-6 mV	40 mV

Zusammenfassend lässt sich sagen, dass bei einer zentralen Spannungsversorgung sich der Signalhub der Taktsignale um 154 mV degradiert, während bei einer verteilter Spannungsversorgung (Abbildung 4.49 rechts) sich der Signalhub nur um 40 mV verringert. Eine noch kleinere Degradation zwischen den innen- und ausliegenden Demultiplexern kann nicht erreicht werden, da die Spannungszuführung aus layouttechnischen Gründen nicht vollständig homogen

ausgeführt werden kann. Die CML-Verstärker für die Takttreiber müssten nach dem Entwurf eigentlich einen Hub von 900 mV erreichen. Dieser Hub kann aber nicht erreicht werden, da die Stromquellentransistoren einen Spannungsabfall von ca. 200 mV über U_{DS} erfordern. Die absoluten Spannungspegel sind ebenfalls gegenüber den theoretisch zu erwartenden Werten verringert, da es auf der zuführenden Verdrahtung der Versorgungsspannung schon einen Spannungsabfall gibt.

4.5.6 Gesamtüberblick Demultiplexer

Abbildung 4.50 zeigt das Blockschaltbild und darunter das Layout des gesamten Eingangsdemultiplexers, der aus 12 einzelnen 1:8-Demultiplexern besteht. Ein zentraler Takteiler stellt die drei Taktebenen aus Tabelle 4.22 für die einzelnen Demultiplexer zur Verfügung. Die Eingänge der Demultiplexer sind mit D_{xy} gekennzeichnet. Der Platzhalter x adressiert das Bit 0, 1 oder 2 des vorgeschalteten AD-Wandlers. Der Platzhalter y steht für den Demultiplexerkanal 0 bis 3 des AD-Wandlers, da dieser das 43 Gbit/s Signal vierfach parallel mit jeweils 10,7 Gbit/s zur Verfügung stellt. Der Platzhalter z am Ausgang jedes einzelnen Ausgangs des 1:8-Demultiplexers gibt die Nummer des parallelisierten Bits 0 bis 7 an. Da es sich bei der Demultiplexerstruktur um einen binären Baum handelt, entsprechen die nebeneinander liegenden Ausgänge nicht der Reihenfolge der parallelisierten Bits. Takttrieberrschaltungen befinden sich entlang des Taktverteilnetzwerkes verteilt auf der ganzen Chiplänge von etwa 4 mm.

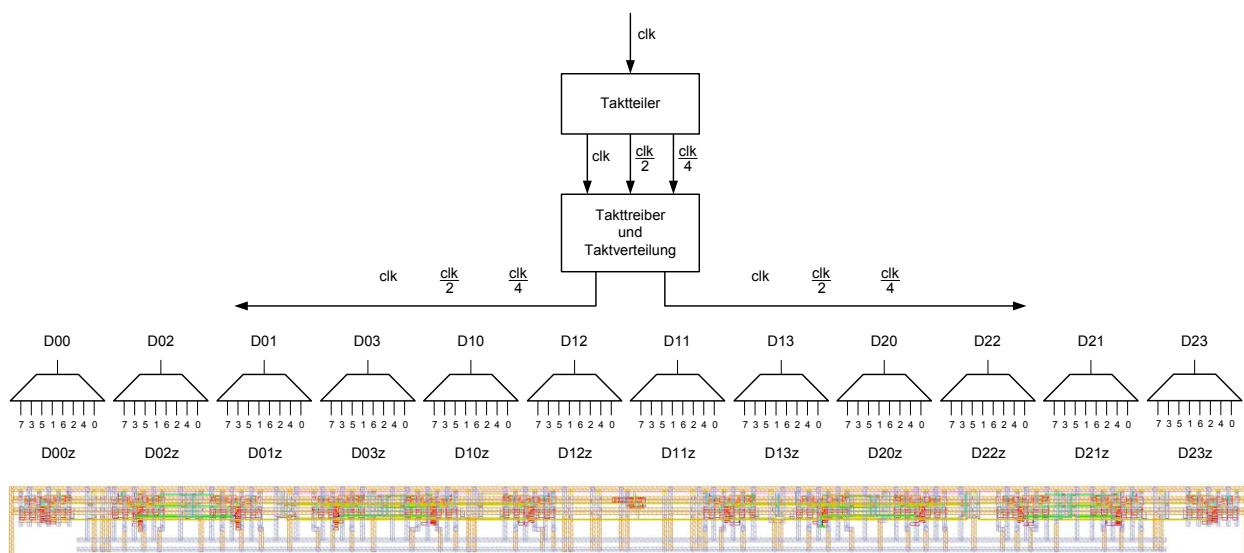


Abbildung 4.50: Blockdiagramm und Layout des 12-fach 1:8-Demultiplexers

Tabelle 4.26 gibt die geometrischen Abmessungen des Layouts, die Anzahl der Transistoren sowie die Stromaufnahme wieder.

Tabelle 4.26: Eckdaten des gesamten Demultiplexers

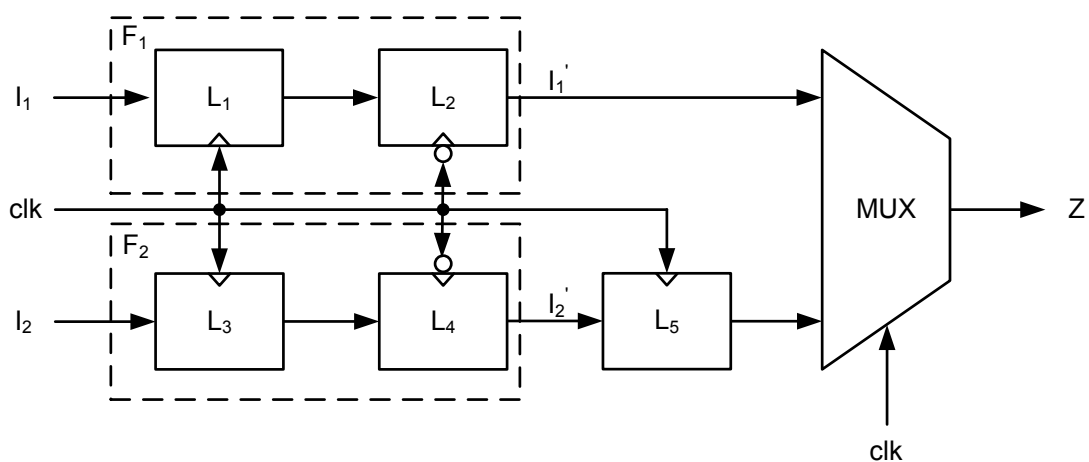
Breite	3.808 μm
Höhe	127 μm
Fläche	483.616 μm^2
Anzahl Transistoren	6455
Stromaufnahme Schaltplan	1,2 A
Stromaufnahme Layout	0,79 A

Der Stromverbrauch auf Layoutebene ist um ca. 33 % geringer als auf Transistorebene, da bei größeren CML-Schaltungen der Spannungsabfall auf den Zuleitungswiderständen der internen Verdrahtung eine nicht unerhebliche Rolle spielt.

4.6 Ausgangsmultiplexer

4.6.1 Aufbau eines CML-Multiplexers

Ein Multiplexer serialisiert im Allgemeinen n parallel anliegende Datenpfade zu einem Datenpfad der n -fachen Datenrate bezogen auf die Eingangsdatenrate. Ein einzelner CML-Multiplexer, der zwei parallele Datenströme zu einem serialisiert, ist in Kapitel 3.4.4 vorgestellt. Probleme ergeben sich bei dem Multiplexer nach Abbildung 3.28, wenn sich die Eingangssignale bzw. das Auswählsignal gleichzeitig ändern. Es kommt dabei zu Störimpulsen (engl. Glitches) auf dem Ausgangssignal [65]. Zur Vermeidung von Störimpulsen wird die Schaltung nach Abbildung 4.51 vorgeschlagen. Als Auswählsignal dient das Taktsignal clk . Die Taktfrequenz bezüglich des Eingangssignals entspricht einem Vollratentakt und bezüglich des Ausgangssignals einem Halbratentakt.

**Abbildung 4.51:** Fünf-Latch-Realisierung eines Multiplexers

Wie bereits beim Demultiplexer wird auch beim Multiplexer eine ungerade Anzahl an Latches verwendet. Die ersten beiden aufeinander folgenden Latches L_1 und L_2 bzw. L_3 und L_4 bilden jeweils ein Flipflop F_1 bzw. F_2 , wodurch verhindert wird, dass ein transparenter Pfad entsteht. Mithilfe der Flipflops werden die beiden Eingangssignale I_1 und I_2 synchronisiert, woraus die Zwischensignale I'_1 und I'_2 entstehen. Angenommen die Zwischensignale I'_1 und I'_2 liegen direkt am Multiplexer MUX an und wechseln bei Anlegen einer Taktflanke gleichzeitig ihren Pegel in entgegengesetzter Richtung, kommt es am Ausgang Z zu einem Störimpuls. Das heißt, dass sich der Spannungspegel am Ausgang kurzzeitig ändert, bevor er wieder seinen ursprünglichen Pegel annimmt. Die Ursache dieses Störimpulses bzw. kombinatorischen Spikes liegt darin, dass in oben genanntem Fall an allen Eingängen des CML-Multiplexers (vgl. Abbildung 3.28) die Common-Mode-Spannung anliegt, weshalb alle Stromzweige leiten. Da sich die Ströme gleichmäßig aufteilen, fällt an beiden Widerständen die halbe Spannung ab, was zu der kurzzeitigen Spannungsänderung am Ausgang führt. Sobald das Taktsignal seinen Endpegel erreicht hat, nimmt auch der Ausgang des Multiplexers den Pegel des entsprechenden Eingangspegels an.

Eine Möglichkeit diese Impulsspitze zu vermeiden, besteht darin, eines der beiden Eingangssignale nach der Synchronisierung durch das Flipflop zu verzögern, damit sich am Multiplexereingang niemals beide Signale gleichzeitig ändern. Optimal ist eine Verzögerung der Daten um eine halbe Taktperiode. Dies lässt sich mit Hilfe des zwischen Flipflop und Multiplexer geschalteten Latches L_5 erreichen. Der Grund, dass sich das Verzögerungs-Latch im Gegensatz zum Demultiplexer im unteren Zweig befindet, besteht in der Problematik, dass die zeitliche Reihenfolge der Bits am Ausgang - ähnlich wie beim Demultiplexer - korrekt sein muss. Für Details wird auf [64] verwiesen.

4.6.2 8:1 Multiplexer

Der 8:1-Multiplexer ist wie auch der 1:8-Demultiplexer aus drei Stufen aufgebaut. Von Stufe zu Stufe verdoppelt sich die Datenrate. Somit werden acht Ausgänge des Viterbi-Rechenwerkes zu einem Ausgang mit achtfacher Datenrate zusammengefasst. Abbildung 4.52 zeigt das Blockschaltbild des als binären Baum realisierten 8:1-Multiplexers. Die Zahlen an den Eingängen geben die Reihenfolge an, in der die Eingangssymbole an den Ausgang durchgeschaltet werden.

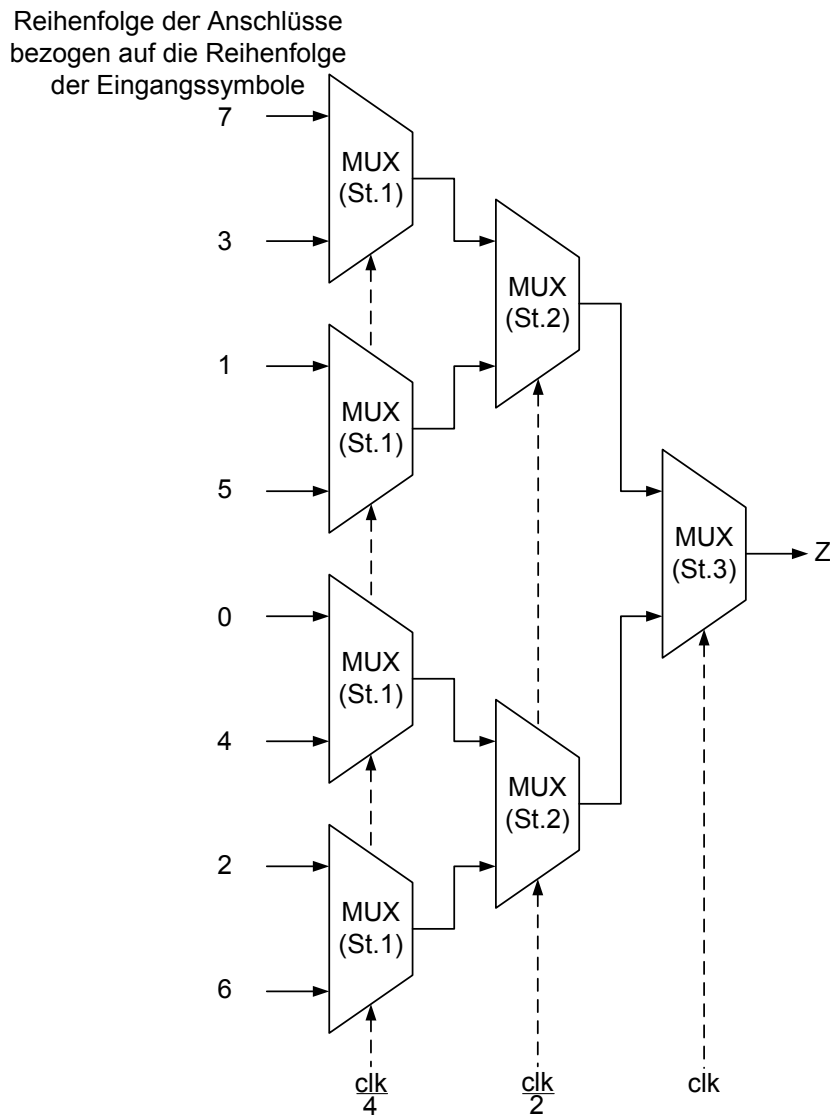


Abbildung 4.52: 8:1-Multiplexer als binärer Baum

Die Taktfrequenzen und Datenraten an den Ein- und Ausgängen sowie an den Zwischenstufen sind in Tabelle 4.27 angegeben.

Tabelle 4.27: Taktfrequenzen und Datenraten an den einzelnen Multiplexern

Multiplexerstufe	Taktfrequenz	Eingangsdatenrate	Ausgangsdatenrate
1	1,3375 GHz	1,3375 Gbit/s	2,675 Gbit/s
2	2,675 GHz	2,675 Gbit/s	5,35 Gbit/s
3	5,35 GHz	5,35 Gbit/s	10,7 Gbit/s

Ähnlich wie beim Demultiplexer sind beim Multiplexer die langsameren, ersten beiden Stufen mit 0,5 mA-Latches aufgebaut. Erst die dritte, schnelle Multiplexerstufe enthält die 1 mA-

Latches. CMOS zu CML-Konverter und Treiberschaltungen für die Taktleitungen vervollständigen den 8:1-Multiplexerblock, welcher in Abbildung 4.53 dargestellt ist.

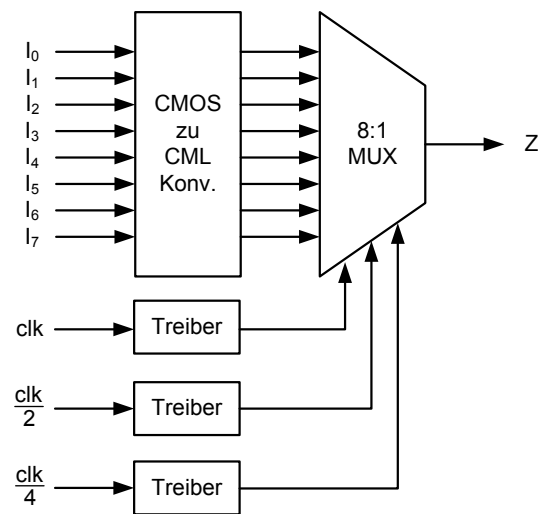


Abbildung 4.53: Schaltplan des 8:1-Multiplexerblocks

Das Layout eines 8:1-Multiplexers einschließlich CMOS-CML-Konverter, Treiber-, Verzögerungsschaltungen und einer Schaltung zur Arbeitspunkteinstellung ist in Abbildung 4.54 dargestellt.

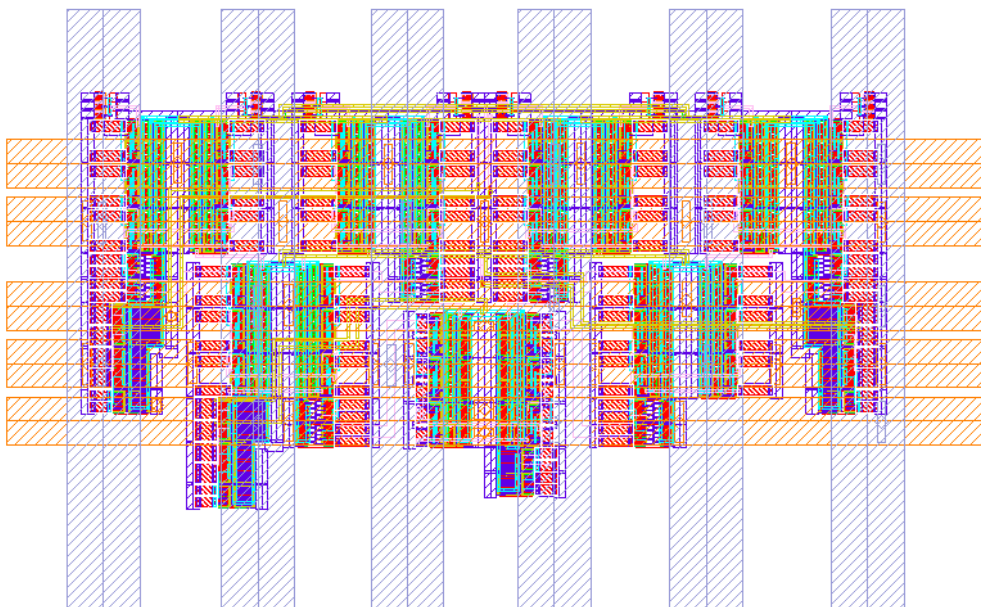


Abbildung 4.54: Layout eines 8:1-Multiplexerblocks

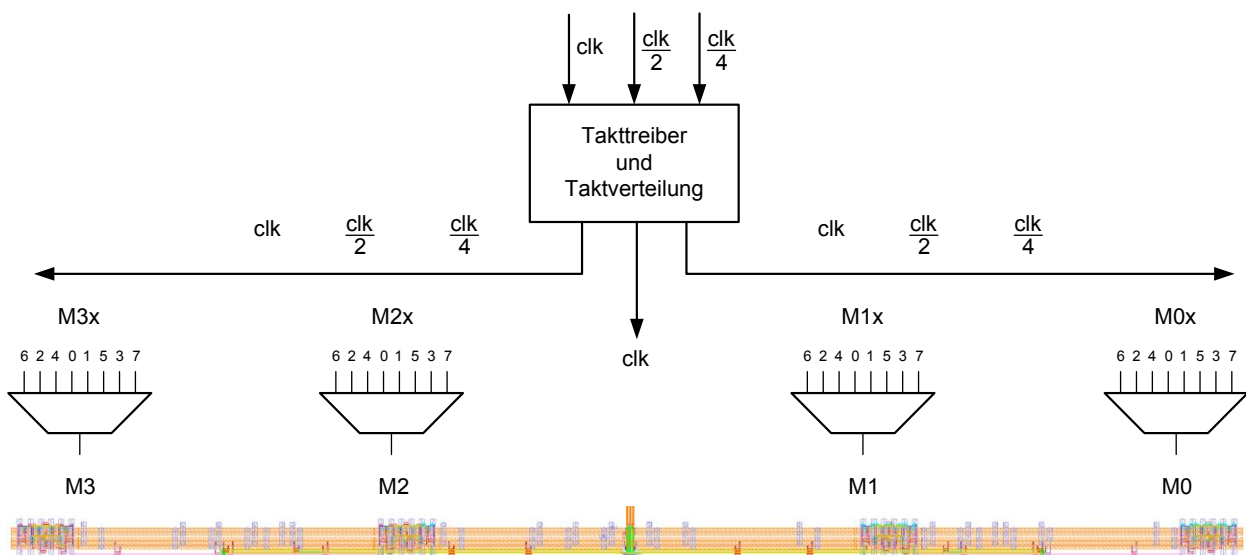
Tabelle 4.28 fasst die Eckdaten des 8:1-Multiplexerblocks zusammen.

Tabelle 4.28: Eckdaten des 8:1-Multiplexerblocks

Höhe	99 μm
Breite	162 μm
Fläche	16.038 μm^2
Anzahl Transistoren	411
Stromaufnahme	86 mA

4.6.3 Gesamtüberblick Multiplexer

Abbildung 4.55 zeigt das Blockschaltbild und das Layout des Vierfach-8:1-Multiplexers, der am Ausgang des Viterbi-Rechenwerks angeschlossen ist. Die drei Taktebenen werden vom zentralen Taktteiler aus Kapitel 4.5.5 zur Verfügung gestellt. Auf dem Taktverteilnetzwerk, das entlang des ganzen Chips geführt ist, befinden sich Takttreiberschaltungen, damit an den vier Multiplexern regenerierte Taktsignale anliegen. Die Struktur des Taktverteilnetzwerkes ist baumförmig, damit die Taktsignale an allen Stellen entlang des Chips synchron sind. Für die längeren Leitungen wird Metalllage 5 verwendet, da hier die kapazitiven Einflüsse auf die Verzögerungszeit (RC-Zeitkonstante) aufgrund des größeren Abstands zum Substrat hin geringer sind als auf einem weiter unten liegenden Metall. Für die kürzeren Leitungen kommt Metalllage 3 zum Einsatz. Aufgrund der kürzeren Entfernungen spielen die kapazitiven Einflüsse hier eine geringere Rolle.

**Abbildung 4.55:** Blockdiagramm und Layout des vierfachen 8:1-Multiplexers

Die Einzelmultiplexer sind von M0 bis M3 durchnummeriert und geben jeweils ein 10,7 Gbit/s Signal aus. Die Nummerierung der Multiplexer entspricht der Reihenfolge der Bits aus der Sichtweise eines seriellen 43 Gbit/s Signals. Der Index x an den Eingängen der einzelnen Multiplexer gibt die zeitliche Reihenfolge an, in welcher die Bits am Ausgang anliegen.

Tabelle 4.29 gibt die geometrischen Abmessungen, den Transistoraufwand und die Stromaufnahme wieder. Wie auch beim Demultiplexer ist die Stromaufnahme beim Multiplexer in der Layoutsimulation geringer als bei der Schaltplansimulation. Aufgrund der geringeren Stromaufnahme des Multiplexers ist der Spannungsabfall auf dem Versorgungsspannungsnetzwerk geringer als beim Demultiplexer. Daher beträgt der Unterschied der Stromaufnahme nur 20 %.

Tabelle 4.29: Eckdaten des gesamten Multiplexers

Breite	2.950 μm
Höhe	90 μm
Fläche	265.500 μm^2
Anzahl Transistoren	1869
Stromaufnahme Schaltplan	0,3 A
Stromaufnahme Layout	0,24 A

4.7 Gesamtarchitektur des Viterbi-Entzerrers

Der Entzerrer soll für optische Übertragungssysteme mit Datenraten von bis zu 43 Gbit/s eingesetzt werden. Der vorgeschaltete AD-Wandler parallelisiert den seriellen Datenstrom und stellt vier einzelne, jedoch zeitverschachtelte 10,7 GS/s Datenströme zu Verfügung. Dies ist ein Kompromiss im Schnittstellenaufwand. Einerseits benötigt die Schnittstelle dann nicht zu viel Chipfläche und andererseits lassen sich Signale mit Datenraten von 10,7 Gbit/s pro Anschlusspin noch mit vertretbarem Aufwand über Bonddrähte zuführen. Daher bietet sich eine vierfach parallele Verarbeitung an. Das eigentliche Viterbi-Rechenwerk wird demzufolge aus vier identischen, parallel geschalteten systolischen Blockdekodern (SBD) (vgl. Abbildung 2.12) aufgebaut. Ein SBD besteht aus einem Zweigmetrikspeicher (BMU) für 384 bit einschließlich Verteilung der eingespeicherten Metriken, 64 Addier-Vergleichs-Auswahleinheiten (ACSU) und 28 Pfadrückverfolgungseinheiten (TBU). Zwischenergebnisse werden in Flipflops gespeichert. Abbildung 4.56 zeigt das Layout eines SBDs.

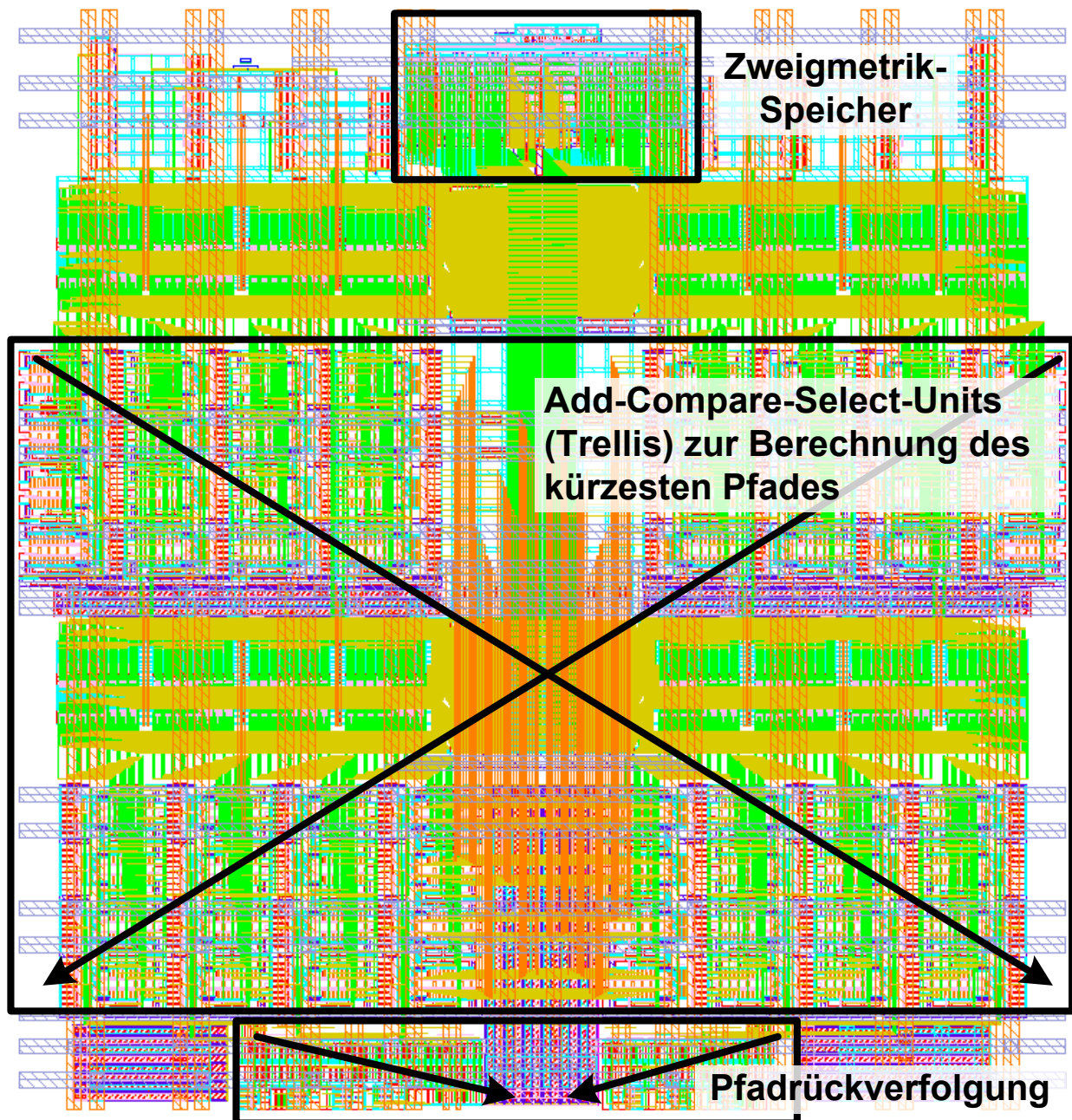


Abbildung 4.56: Layout eines SBDs

Die Verknüpfung der vier SBDs wird über Minimumauswahlschaltungen (SMU) vorgenommen. Die einzelnen Bausteine sind in den vorhergehenden Kapiteln erläutert. Ein Register verbindet die beiden äußeren SBDs. Diese Verbindung ist notwendig, da es sich um einen unendlich langen Datenstrom ohne Anfang und Ende, also ohne Unterteilung in Rahmen auf der Bitübertragungsschicht, handelt. Demultiplexer am Eingang und Multiplexer am Ausgang des Viterbi-Entzerrers erlauben es, dass das Viterbi-Rechenwerk mit nur 1,34 GHz betrieben werden kann. Die Eingangssymbole, die vierfach parallel mit 10,7 GS/s von einem vorgeschalteten AD-Wandler geliefert werden, bestehen aus drei Bits. Die entzerrten Ausgangsbits werden ebenfalls vierfach parallel mit 10,7 Gbit/s ausgegeben. Abbildung 4.57 zeigt die interne Struktur des

Viterbi-Entzerrers mit den Systemdaten aus Tabelle 4.30 einschließlich der Schnittstellen der zu prozessierenden Eingangssymbole und der entzerrten Ausgangsbits.

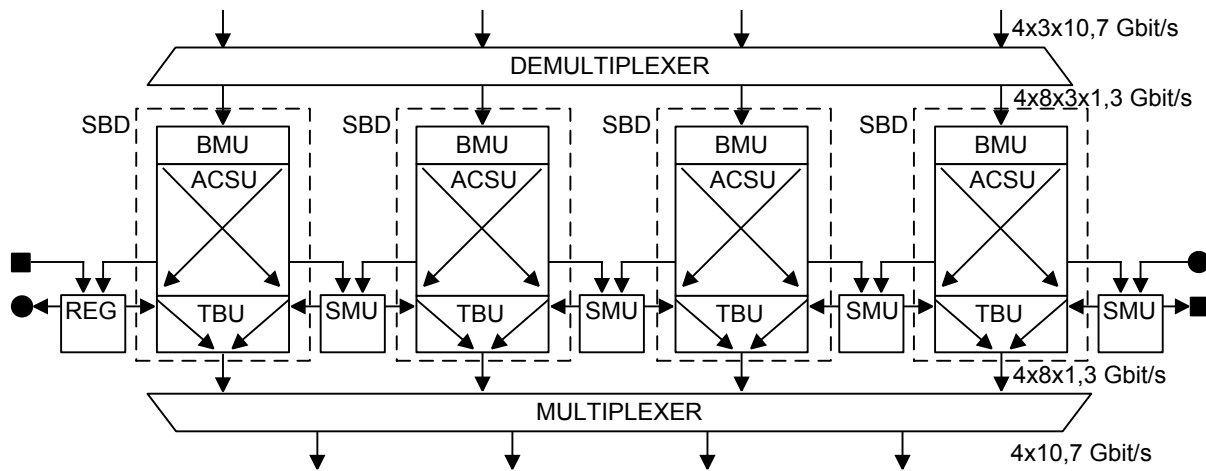


Abbildung 4.57: Architektur des Viterbi-Entzerrers

Tabelle 4.30: Systemdaten des Viterbi-Entzerrers

Anzahl Zustände	4
Blocklänge	16
Pfadrückverfolgungslänge	8
Anzahl ACSUs	256

Das Layout des Viterbi-Entzerrers ist in Abbildung 4.58 dargestellt. Die einzelnen Teilblöcke aus Abbildung 4.57 sowie die Anschluss pads ringsherum sind deutlich zu erkennen. Die Eckdaten des Layouts sowie die Beschreibung der Pads sind in Tabelle 4.31 zusammengefasst.

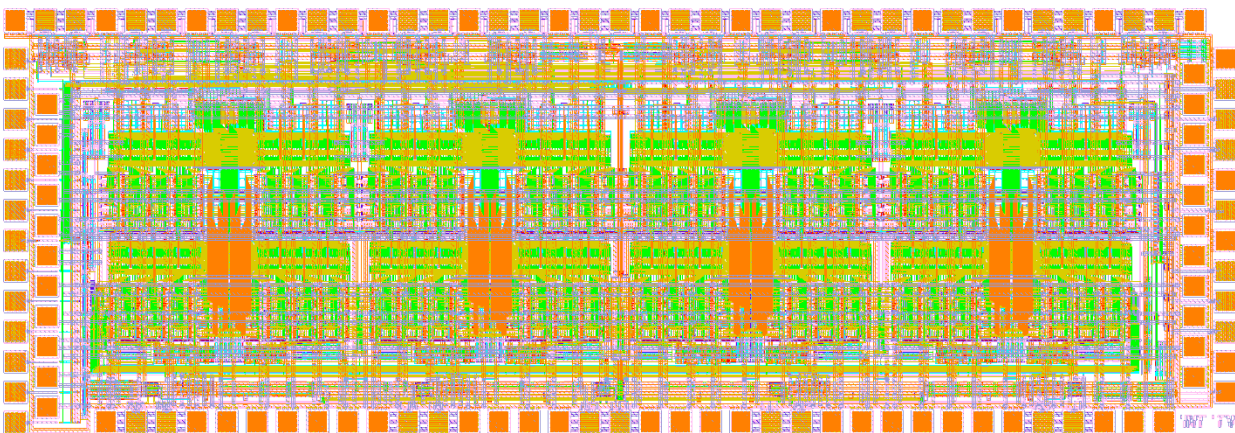


Abbildung 4.58: Maskenlayout des Viterbi-Entzerrers für 43 Gbit/s in 90 nm CMOS

Das digitale Rechenwerk des Viterbi-Entzerrer-Chips ist in statischem CMOS aufgebaut, während die schnellen Ein- und Ausgangsschnittstellen in CML aufgebaut sind. Beide Logikfamilien arbeiten mit unterschiedlichen Logikpegeln. Daher sind Konversionsschaltungen sowohl von CML- zu CMOS-Pegeln als auch von CMOS- zu CML-Pegeln notwendig. Die dazu erforderlichen Schaltungen sind in Kapitel 3.5 vorgestellt.

Tabelle 4.31: Layoutdaten des Viterbi-Entzerrers

Fläche (Länge x Breite)	4,08 mm x 1,40 mm = 5,7 mm ²
Transistoranzahl	451504
Padanzahl	123
davon	
Eingangssymbole (10,7 GS/s)	24
Ausgangsbits (10,7 Gbit/s)	8
Takt	4 (Ein- und Ausgang jeweils 2)
Zweigmetriken + Schreibfreigabe	12 + 1
Statistikdaten + Gültigkeitssignal	6 + 1
Impedanzumschaltung Eingangssymbole	1
Positive Versorgungsspannung	37
Negative Versorgungsspannung	29

4.7.1 Globale Taktverteilung

Der Viterbi-Entzerrer-Chip hat einen zentralen differenziellen Takteingang. Aus diesem 5,35 GHz Takt werden alle internen Takte sowohl für die CML-Logik als auch für die statische CMOS-Logik abgeleitet und entsprechend verteilt. Die Schwierigkeit besteht darin, den Takt so zu verteilen, dass er an allen Flipflops synchron anliegt. Innerhalb der CML-Schaltung wird der Takt daher baumförmig verteilt. An fünf Stellen am linken und rechten Rand sowie zwischen den einzelnen SBDs und erfolgt ein Abgriff des langsamsten CML-Taktes. Dieser wird von CML-Pegeln in CMOS-Pegel umgewandelt und dann in einer 850 µm langen Leitung quer zum Chip nach unten geleitet. Auf dieser Leitung befinden sich vier Doppelinverter, um das Taktsignal zu regenerieren. Dies führt zu einer signalflossorientierten Taktverzögerung. Am Ausgang dieser Regenerationsinverter befinden sich Abzweigungen, die den Teilschaltungen innerhalb eines SBDs den Takt zuführen. Innerhalb der Teilschaltungen wie BMU, ACSU und TBU erfolgt die Taktverteilung mittels einer H-Struktur, wodurch sich eine synchrone Taktzuführung zu den einzelnen Flipflops ergibt. Auf den Verästelungen der H-Struktur befinden sich weitere Inverter, die das Taktsignal regenerieren.

4.7.2 Globale Verdrahtung

Wie in Abbildung 4.57 zu erkennen ist, sind das linke und rechte Ende des Chips miteinander verbunden. Die Konnektoren sind durch schwarze Quadrate und Kreise dargestellt. Dabei handelt es sich um rund 4 mm lange Busleitungen. Der Bus mit den runden Konnektoren besteht aus 32 Leitungen, der vier Pfadmetriken der ACSUs ans andere Chipende weiterleitet. Der Bus mit den quadratischen Konnektoren besteht aus zwei Leitungen, worüber die Entscheidung der SMU an die TBUs des linken SBDs übertragen wird.

Eine über 4 mm lange parallele Leitungsführung auf engem Raum führt zwangsläufig zu Übersprechen. Die Simulation eines 4,1 mm langen und 5 bit breiten Busses, bei einer Leitungsbreite von 140 nm und einem Leitungsabstand von 300 nm ist in Abbildung 4.59 links dargestellt. Die beiden äußeren und die mittlere Leitungen dienen als Aggressoren, die jeweils dazwischen liegende Leitung ist die passive Leitung. Am Leitungsende erkennt man eine deutliche Pegeldegeneration auf den passiven Leitungen. Die Schaltschwelle von 0,5 V wird zwar nicht erreicht, aber es besteht nur ein geringer Störabstand von 100 mV. Die Aggressor-Leitungen sind ebenfalls deutlich degradiert. Hier wird die Schaltschwelle teilweise nur knapp unterschritten. Es kann zu Fehlinterpretationen kommen. Die Leitungsverzögerung liegt bei rund 530 ps.

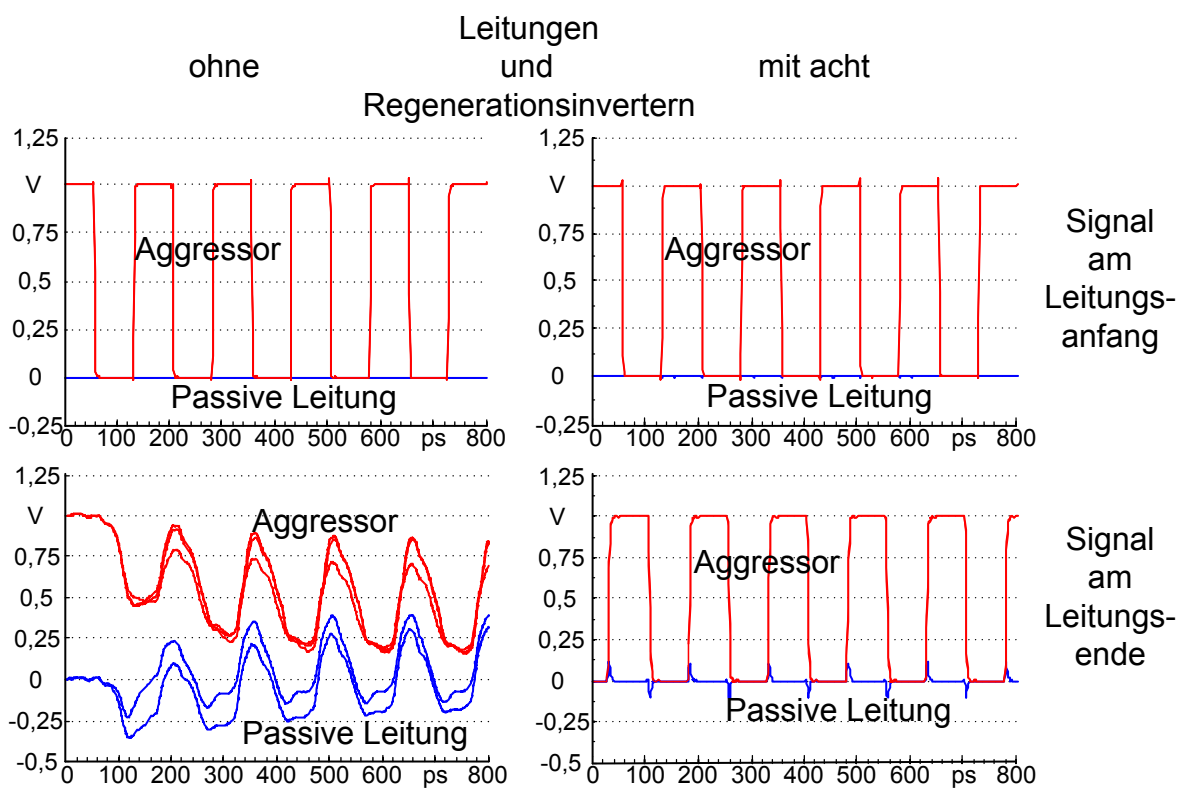


Abbildung 4.59: Simulation des Übersprechens eines 4,1 mm langen Busses links ohne und rechts mit Regenerationsinverter

Zur Vermeidung von Übersprechen wird der Leitungsabstand auf 860 nm deutlich erhöht, und es werden insgesamt acht Regenerationsinverter im Abstand von 500 μm eingebaut. Das Simulationsergebnis dieser Leitungsanordnung ist im rechten Bild von Abbildung 4.59 dargestellt. Weder auf den Aggressoren noch auf den passiven Leitungen werden die Logikpegel degeneriert. Der maximale Störimpuls auf der passiven Leitung beträgt 111 mV. Ein ausreichender Störabstand zur Schaltschwelle ist somit sichergestellt. Mit den Invertern erfährt das Signal eine Verzögerung von insgesamt 510 ps und ist daher schneller als ohne Regenerationsinverter. Da eine volle Taktperiode von 747 ps zur Verfügung steht, hat diese Verzögerung keine Auswirkung auf die Schaltung.

Um Platz zu sparen, wird der 32 bit breite Bus auf die beiden Metalllagen 3 und 5 aufgeteilt. Der horizontale Abstand zweier benachbarter Leitungen auf gleicher Metalllage beträgt 820 nm. Der vertikale Abstand zwischen den Metalllagen beträgt 670 nm. Da der vertikale Abstand in etwa dem horizontalen Abstand entspricht, wird eine Interferenz in Form von funktionellem Übersprechen zwischen beiden Metalllagen ausgeschlossen. Auf eine schirmende Metalllage 4 wird zugunsten von zusätzlichem Verzögerungsübersprechen verzichtet.

4.7.3 Spannungsversorgung und Stromaufnahme

Die Gesamtschaltung beinhaltet CML-Schaltungen für die schnellen Ein- und Ausgangssignale und statische CMOS-Schaltungen für digitale Signalverarbeitung. CML-Schaltungen weisen eine konstante Stromaufnahme auf. Hierbei spricht man von einem statischen Strom. Bei der statischen CMOS-Logik fließt nur während Schaltvorgängen Strom. Es handelt sich hierbei um einen dynamischen Stromfluss, der einerseits aus dem Auf- und Entladen der Transistorkapazitäten resultiert und andererseits aus dem Querstrom zwischen positiver und negativer Versorgungsspannung, der zum Schaltzeitpunkt der Logikgatter sein Maximum erreicht, besteht.

Der statische Strom, der von den CML-Demultiplexern und -Multiplexern benötigt wird, beträgt konstant 1,55 A. Die Stromaufnahme der digitalen Rechenschaltung ist in Abbildung 4.60 über der Zeit dargestellt.

Hierbei erkennt man, dass zweimal pro Taktperiode ein Strom von über 5 A fließt, während der über eine Taktperiode gemittelte Strom nur ca. 1 A beträgt. Das bedeutet, dass der meiste Strom für das Taktverteilnetzwerk und für die Flipflops benötigt wird. Die Flipflops schalten zweimal pro Taktperiode, da ihre beiden Latches pro Halbtakt zwischen Folge- und Haltephase wechseln. Weiterhin erkennt man, dass jede zweite Stromspitze breiter ist als die dazwischenliegende. Die breitere Stromspitze zeigt an, dass die eigentliche Logikschaltung rechnet. Fällt der dynamische Strom vor jeder Spitze, die eine Logikgatteraktivität anzeigt, auf Null ab, deutet dies darauf hin, dass die Rechenvorgänge der CMOS-Logik innerhalb einer Taktperiode komplett abgeschlossen sind.

Aus diesem hohen dynamischen Stromanteil ergeben sich besondere Anforderungen an den Schaltungsentwurf. Aufgrund von Zuleitungs- und Verdrahtungswiderständen kommt es zu einem Spannungsabfall auf dem Chip, wodurch sowohl die CML- als auch die CMOS-Logik in ihrer Funktion beeinträchtigt wird.

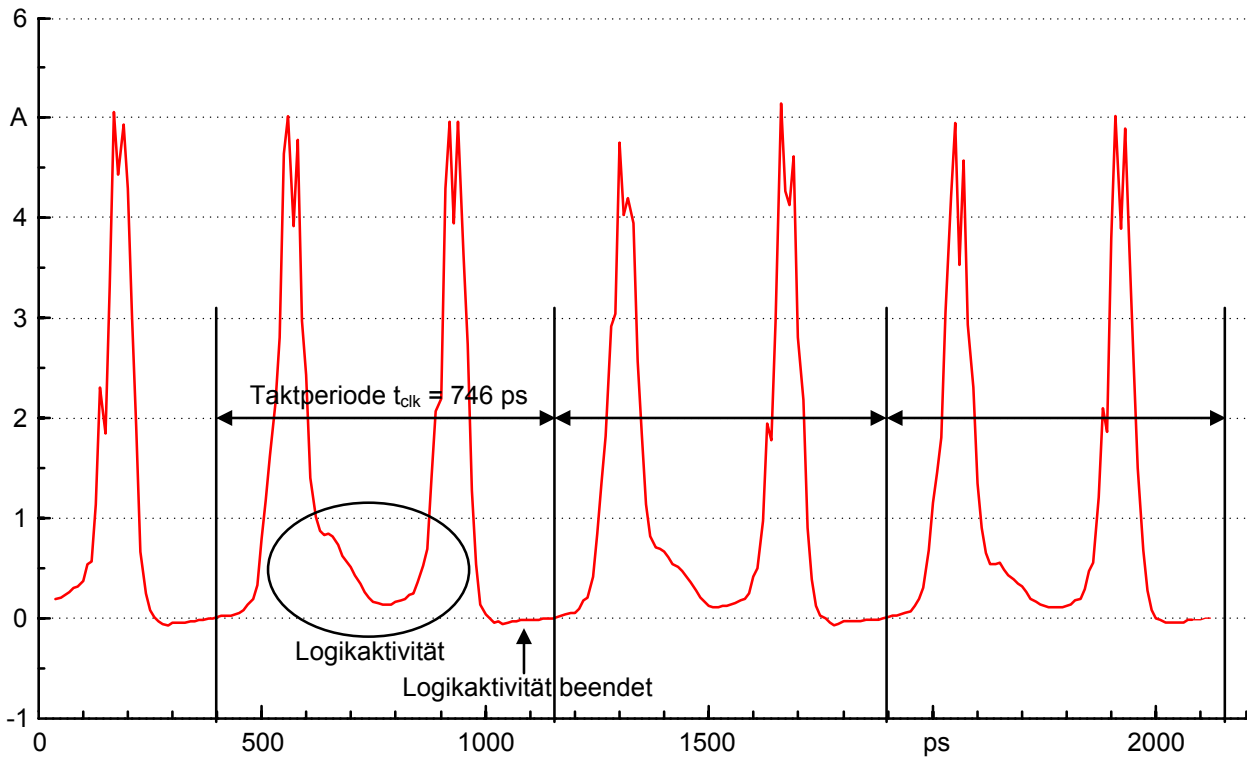


Abbildung 4.60: Stromaufnahme der statischen CMOS-Logik

Um dieser Problematik entgegenzuwirken, werden folgende Maßnahmen ergriffen:

Auf dem ganzen Chip werden auf freien Flächen Kapazitäten aus Polysilizium verteilt, die eine Gesamtkapazität von 2 nF erreichen. Bei einer Poly-Kapazität handelt es sich im Wesentlichen um einen MOSFET, dessen parasitäre Gate-Bulk-Kapazität ladungsspeichernd wirkt. Die Gate-Bulk-Kapazität wirkt, wenn sich im MOSFET an der Grenzschicht zwischen Gate-Oxid-Isolator und dem p-Substrat (Bulk) Löcher anhäufen. Diese Löcheranhäufung nennt man Akkumulation [40]. Akkumulation erreicht man, wenn der MOSFET im Sperrbereich betrieben wird und seine Gate-Source-Spannung kleiner als die Flachbandspannung ist. In diesem Arbeitsbereich weist der MOSFET die höchste Kapazität auf. Da der Halbleiterprozess p-Substrat vorsieht, muss ein p-Kanal-MOSFET mit n-Wanne verwendet werden. Schließt man das Gate an der positiven Versorgungsspannung und die n-Wanne an der negativen Versorgungsspannung an, arbeitet der p-Kanal-MOSFET ständig in Akkumulation, da seine Gate-Source-Spannung ständig positiv ist. Durch diese verteilten Kapazitäten sollen die Stromspitzen gedämpft werden, um einem Spannungsabfall des Versorgungsspannungsnetzes entgegenzuwirken.

Eine weitere Maßnahme, die Spannung auf dem Chip möglichst konstant zu halten, ist ein engmaschig aufgebautes Versorgungsspannungsnetz. Für die Leitungen werden die beiden obersten Metalllagen mit einem besonders geringen Widerstand von $22 \text{ m}\Omega/\square$ verwendet.

Weiterhin dienen über 50 % der Gesamtanzahl an Pads der Stromzuführung. Auf eine gleichmäßige Padverteilung um den Chip herum wird ebenso Wert gelegt.

4.7.4 Ein- und Ausgangsschnittstellen für Hochgeschwindigkeitsdaten

Die Eingangsschnittstelle für die Hochgeschwindigkeitsdaten ist differenziell ausgelegt. Um die Zieldatenrate von 43 Gbit/s zu erreichen, muss jedes Eingangsschnittstellenpaar mit 10,7 Gbit/s betrieben werden. Dies entspricht einer maximalen Datenfrequenz von 5,35 GHz. Da ein Halbratentakt verwendet wird, beträgt die Taktfrequenz ebenfalls 5,35 GHz. Aufgrund der Störanfälligkeit und der Geschwindigkeitslimitierung des 90 nm CMOS-Prozesses ist eine differenzielle Logik notwendig. Die Eingänge sind mit einem 50Ω Widerstand abgeschlossen, um Signalreflexionen zu vermeiden. Diese Eingangsimpedanz kann über einen Transistor entweder gegen die positive oder die negative Versorgungsspannung geschaltet werden. Dadurch ist es möglich, das Massebezugspotenzial der CML-Schaltungen zu wählen. Dies ist je nach Anwendungsfall erforderlich. Verwendet man zum Testen eine FPGA-Platine, die mit positiver Versorgungsspannung arbeitet, beträgt das Massepotenzial 0 V, was dem negativen Pol der Versorgungsspannung entspricht. Verbindet man den Viterbi-Chip mit einem AD-Wandler, der mit negativer Versorgungsspannung betrieben wird, entspricht das Massepotenzial der positiven Versorgungsspannung von 0 V. Die Eingänge sind so dimensioniert, dass eine Eingangsdifferenzspannung von 300 mV als Logikpegel erkannt wird.

Die Ausgangsschnittstelle der entzerrten Hochgeschwindigkeitsdaten besteht aus einer Kette von fünf CML-Verstärkern, deren Ausgangsstufe eine externe ohmsche Last von 50Ω bei einem Logikhub von ebenfalls 300 mV treiben kann.

4.7.5 Digitale Konfigurations- und Statistikschnittstelle

Der Zweigmetrikspeicher muss zu Beginn der Datenprozessierung mit entsprechenden Metrikdaten beschrieben werden. Es ist ebenfalls vorgesehen, die Metriken im Betrieb zu aktualisieren. Die Aktualisierungsgeschwindigkeit soll maximal $1/512$ der Eingangsdatenrate bezogen auf 43 Gbit/s betragen. Dies entspricht einer Dateneinschreibefrequenz von rund 84 MHz. Daher kann die Schnittstelle mit unipolaren CMOS-Pegeln betrieben werden. Außerdem wird bei diesen niedrigen Frequenzen auf eine Anpassung auf 50Ω verzichtet. Jedoch werden an den Eingangsschnittstellen Schmitt-Trigger (siehe Kapitel 3.3.4) eingesetzt, um Schwingungen, die über die Schaltschwelle gehen, hervorgerufen durch Fehlanpassungen und Induktivitäten in den Zuleitungen, unschädlich zu machen.

An der Statistikschnittstelle werden ein analog-digital gewandeltes 3 bit-Symbol und drei nebeneinanderliegende entzerrte Bits ausgegeben. Außerdem wird noch ein Gültigkeitssignal ausgegeben, das dem durch 64 geteilten Takt des Eingangstaktsignals entspricht. Mit Hilfe dieser Signale ist es möglich, eine Kanalschätzung im Betrieb durchzuführen, um die Metriken im Betrieb nachführen zu können (siehe Kapitel 7). Die Ausgangstreiber bestehen aus einer Kette von CMOS-Invertern. Sie sind so dimensioniert, dass sie eine ohmsche Last von 50Ω bei 1 V treiben können.

5 Chipentwurf und Chipfertigung

5.1 Design Kit

Für den Chipentwurf steht ein Design Kit eines Drei-Wannen-CMOS-Halbleiterprozesses des 90 nm Technologieknotens der Firma ST Microelectronics in der Version 4.01 zur Verfügung. Zu einem späteren Zeitpunkt wird auf Version 6.1 gewechselt. Der Halbleiterprozess stellt Transistoren für den allgemeinen Anwendungsfall (engl. General-Purpose, GP) mit hoher (HVT), mittlerer (SVT) und niedriger (LVT) Schwellenspannung zur Verfügung. GP-Transistoren weisen mit 1,6 nm eine dünnere Gate-Oxidschicht auf während Transistoren für niedrige Verlustleistung (engl. Low-Power, LP) eine Gate-Oxiddicke von 2,1 nm haben [51]. Die physikalische Kanallänge der GP-Transistoren beträgt 70 nm, während die Kanallänge der LP-Transistoren 90 nm beträgt [51]. Die gezeichnete Kanallänge beträgt hingegen 100 nm. Die GP-Transistoren zeichnen sich gegenüber den LP-Transistoren durch eine höhere Schaltgeschwindigkeit aus. Aufgrund des dünneren Gate-Oxids und der kürzeren effektiven Kanallänge sind die Leckströme höher und die Versorgungsspannung der GP-Transistoren niedriger als die der LP-Transistoren. Da der Viterbi-Entzerrer nicht batteriebetrieben ist, sind die Leckströme zweitrangig. Daher werden schnelle GP-Transistoren bevorzugt. Weiterhin werden für die geschwindigkeitslimitierenden Schaltungsteile LVT-Transistoren eingesetzt, da diese aufgrund der niedrigeren Kanaldotierung schneller sind als Transistoren mit höherer Schwellenspannung. Der Nachteil der LVT-Transistoren ist ebenfalls der höhere Leckstrom. In Bereichen mit hoher Transistordichte (z.B. Speicherfelder), welche nicht im kritischen Geschwindigkeitspfad liegen, werden HVT-Transistoren eingesetzt, da sich dort der höhere Stromverbrauch nachteiliger auswirkt als in anderen Schaltungsteilen. Die nominale Versorgungsspannung der GP-Transistoren beträgt 1,0 V, wobei der Halbleiterhersteller eine Spannung von 0,9 V bis 1,2 V als zuverlässigen Betriebsbereich definiert [51]. Ein Betrieb mit erhöhter Versorgungsspannung ist vorübergehend zulässig, jedoch sinkt die Lebensdauer der Schaltung aufgrund der größeren Wärmeentwicklung.

Die LVT-Transistoren werden vom Design Kit Hersteller im BSIM3-Modell zur Verfügung gestellt, während die HVT-Transistoren im neueren BSIM4-Modell verfügbar sind. Die Modelle liegen sowohl für besonders langsame als auch für besonders schnelle Transistoren vor, die aufgrund von Prozessschwankungen auftreten können. Bei den Prozessschwankungen geht man von einer Normalverteilung aus, wobei die Standardabweichung zwischen langsamen und schnellen Transistoren eine Schwankungsbreite von $\pm 3\sigma$ umfasst. Das bedeutet, dass sich die Abweichung des Schaltverhaltens vom Mittelwert mit einer Wahrscheinlichkeit von 99,7 % innerhalb der beiden Extrema befindet. Standardmäßig wird mit dem Mittelwert der beiden Extrema simuliert. Welcher der beiden CMOS-Transistoren (NMOS oder PMOS) als besonders langsam oder schnell angenommen werden soll, kann separat eingestellt werden.

Weiterhin bietet der Prozess sieben Metallisierungslagen aus Kupfer [51] zur Verdrahtung an. Für die Simulation der Metalle stehen, ähnlich wie für die Transistoren auch, unterschiedliche Modelle, die aufgrund von Prozessschwankungen auftreten können, zur Verfügung. Die Modelle umfassen typische, hohe und niedrige Widerstands- bzw. Kapazitätsbeläge. Die obersten beiden Lagen sind dicker als die anderen und haben den geringsten elektrischen Widerstand. Diese werden für die Zuführung der Versorgungsspannung verwendet. Die unterste Metalllage weist dagegen den höchsten Widerstand auf. Diese wird nur für kurze Verbindungen zur Verschaltung der einzelnen Transistoren innerhalb der Grundgatter verwendet. Der Widerstandswert der dazwischen liegenden Metalle hat einen mittleren Wert. Die Metalllage 2 wird zur Verdrahtung der Grundgatter zu komplexeren kombinatorischen Schaltungen wie z.B. Volladdierer verwendet, während die Metalllage 3 in ganzen Funktionsgruppen wie z.B. Addier-Vergleichs-Auswahleinheiten Anwendung findet. Die Metalllagen 4 und 5 dienen der großflächigeren Verdrahtung auf der Ebene des systolischen Blockdekoders (vgl. Abbildung 2.12 und Abbildung 4.56). Hierbei wird vor allem die Verdrahtung der Trellisstruktur und die Verbindung der großen Blöcke wie BMU, ACSU, TBU und SMU adressiert. Ab den Metalllagen 4 und 5 wird darauf geachtet, dass jede einzelne Metalllage nur in einer Richtung (senkrecht bzw. waagrecht) verlegt wird. Der Vorteil dieser Metalllagen ist der relativ große Abstand zum Substrat, sodass auch bei längeren Leitungen keine zu großen Verzögerungen aufgrund der kapazitiven Wirkung zum Substrat hin auftreten.

Widerstände können in dem Halbleiterprozess aus Metallen oder Polysilizium realisiert werden. Aufgrund der deutlich geringeren Leitfähigkeit von Polysilizium gegenüber Metallen werden Widerstände, wie sie in CML-Schaltungsteilen benötigt werden, aus Polysilizium aufgebaut. Damit lassen sich auf kleiner Fläche einfacher größere Widerstandswerte realisieren.

Kapazitäten zum Blocken der Versorgungsspannung gegen Spannungsschwankungen werden aus parasitären Transistorkapazitäten gewonnen, indem die Kanalkapazität ausgenutzt wird. Aufgrund des dünnen Gate-Oxids von 1,6 nm lassen sich große Kapazitäten auf kleiner Fläche erreichen. Betreibt man die Transistoren im Akkumulationsbetrieb, ist die Kapazität am größten. Dies erreicht man, indem man p-Kanaltransistoren, die in einer n-Wanne liegen, einsetzt. Im Kondensatorbetrieb ist die Gate-Elektrode der positive Pol der Versorgungsspannung und der n-Wannenkontakt der negative Pol.

5.2 Chipentwurf

5.2.1 Vorgehensweise

Bevor mit dem eigentlichen Chipentwurf begonnen wird, erfolgt eine Analyse auf algorithmischer Ebene, welche Rechenoperationen für die Umsetzung des Viterbi-Algorithmus notwendig sind. Die Rechenoperationen werden modularisiert und in spezielle Rechenwerke und

Speicher unterteilt. Hiermit sind vor allem die funktionelle Beschreibung der Zweigmetrik-recheneinheit und der Addier-Vergleichs-Auswahleinheit gemeint.

Nachdem nun die Funktionsweise dieser Module feststeht, beginnt der Chip-Entwurf nach der Bottom-up-Methode.

Zunächst werden auf Schaltplanebene kombinatorische Logikzellen (Grundgatter) aus den schnellsten Transistoren entworfen, die in der Halbleitertechnologie zur Verfügung stehen. Diese werden simuliert und solange optimiert, bis sie die geringste Verzögerungszeit unter Berücksichtigung symmetrischer Schaltflanken aufweisen. Bei den Grundgattern handelt es sich im Wesentlichen um Inverter, UND-, ODER-, Exklusiv-ODER-Funktionen sowie deren verneinten Versionen. Flipflops werden ebenfalls auf Transistorebene entworfen. Nach erfolgreicher Simulation wird der Schaltplan in ein Layout umgesetzt. Unter einem Layout versteht man den Maskenentwurf, aus dem die geometrische Anordnung der Transistoren und der Verdrahtung auf der zu prozessierenden Siliziumscheibe hervorgeht. Mit Hilfe spezieller Softwarewerkzeuge wird das Layout auf die Einhaltung der Entwurfsregeln wie z.B. Leitungsbreiten und -abstände geprüft (engl. Design Rule Check, DRC). Verlaufen diese Überprüfungen erfolgreich, wird in einem weiteren Test die Übereinstimmung zwischen Layout und Schaltplan überprüft (engl. Layout versus Schematic Check, LVS). Mit Hilfe des LVS können Verdrahtungsfehler, wie z.B. unbeabsichtigte Kurzschlüsse zweier Leitungen, gefunden werden. Sofern hierbei keine Korrekturen erforderlich sind, können mit einem weiteren Softwarewerkzeug, die durch die Verdrahtung hinzugekommenen parasitären Elemente wie Widerstände und Kapazitäten extrahiert werden (engl. Parasitic Extraction, PEX). Bei den Kapazitäten wird zwischen Signal-Substrat-Kapazität und Koppelkapazitäten benachbarter Leitungen unterschieden. Die Koppelkapazitäten beschreiben das Übersprechen. Das PEX-Werkzeug erstellt eine neue Netzliste, die diese parasitären Elemente enthält. Im Anschluss kann diese Netzliste wieder mit dem Simulator eingelesen werden und die Schaltung somit resimuliert werden. Durch die hinzugekommenen parasitären Elemente wird die Schaltung im Wesentlichen langsamer. Weiterhin werden die Auswirkungen des Übersprechens ersichtlich. Ist man mit dem Resultat der Resimulation nicht zufrieden, muss das Layout entsprechend abgeändert werden. Reicht das nicht aus, muss der Schaltplan angepasst werden, indem z.B. größere Transistoren zur Erhöhung der Stromtreiberfähigkeit eingesetzt werden.

Wenn das Simulationsergebnis die geforderten Rahmenbedingungen erfüllt, können die Grundgatter als Blöcke zusammengefasst werden. Auf der nächsthöheren Hierarchieebene, der Logikebene, werden diese Grundgatter beispielsweise zu 1 bit-Volladdierern zusammengesetzt. Die einzelnen elementaren arithmetischen Einheiten werden dann wiederum auf der nächsthöheren Hierarchieebene, der Register-Transfer-Ebene, zu komplexeren Recheneinheiten einschließlich Flipflops erweitert. Unter den Schaltungsteilen, die auf Register-Transfer-Ebene aufgebaut sind, versteht man z.B. die Addier-Vergleichs-Auswahleinheiten (ACSUs) oder auch Speicher. Darüber folgt die algorithmische Ebene, die die Verschaltung der ACSUs zu einem Trellisdiagramm beschreibt. Die Kombination aller Teilalgorithmen ergibt dann die

Beschreibung der Gesamtschaltung auf Systemebene. Hierunter versteht man in diesem Zusammenhang die Funktion des gesamten digitalen Viterbi-Entzerrers als in sich abgeschlossene Einheit.

Die Umsetzung und die Simulation auf den höheren Hierarchieebenen erfolgt sowohl auf Schaltplan- als auch auf Layoutebene in den gleichen Schritten, wie schon auf Transistorebene beschrieben. Das heißt, dass die prinzipielle Vorgehensweise auf allen Ebenen grundsätzlich identisch ist. Abbildung 5.1 fasst noch mal den Entwurf integrierter Schaltungen zusammen.

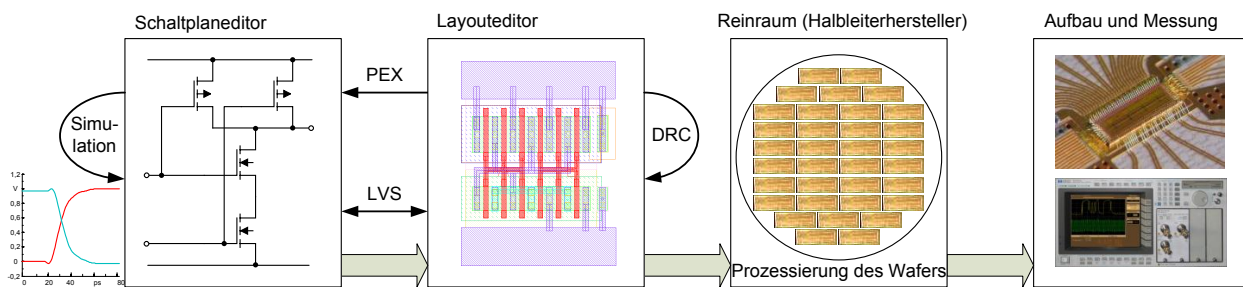


Abbildung 5.1: Vorgehensweise beim Entwurf integrierter Schaltung

Parallel zum Entwurf der integrierten Schaltung, wird der Viterbi-Entzerrer in einer VHDL-Beschreibung aufgebaut. Die eigentliche VHDL-Verhaltensbeschreibung erfolgt auf Grundgatterebene. Diese Grundgatter werden dann auf Blockschaltbildebene in Form einer Strukturbeschreibung zu den Schaltungsteilen der höheren Ebenen zusammenschaltet. Die Vorgehensweise entspricht ebenfalls der Bottom-up-Methode. Mit VHDL lässt sich die Funktion komplexer Systeme schneller verifizieren als durch Simulation auf Transistorebene. Aufgrund der hohen Abstraktion lassen sich dagegen mit VHDL die Verzögerungszeiten und die Stromaufnahme nur grob bzw. gar nicht simulieren.

5.2.2 Zeitlicher Rahmen der Chipentwicklung

In einer Vorstudie wird die 90 nm CMOS-Technologie evaluiert. Hierzu werden CML-Grundsaltungen und eine Addier-Vergleichs-Auswahleinheit in statischem CMOS entworfen, vom Halbleiterhersteller prozessiert und vermessen. Diese Vorstudie ohne Vermessung der Chips beansprucht einen zeitlichen Umfang von neun Monaten. Der darauffolgende eigentliche Entwurf des Schaltplans, des Layouts und die dazugehörigen Simulationen bis zum Tapeout nehmen einen Zeitraum von zwei Jahren und neun Monaten in Anspruch. Die Prozessierung beim Halbleiterhersteller einschließlich Sägen des Wafers dauert mit sieben Monaten länger als erwartet. Vom Beginn des Messaufbaus bis zu den abschließenden Messergebnissen vergehen weitere zwei Jahre und einen Monat, da für die komplexe Aufbautechnik mehrere Versuche notwendig werden. Ohne Vorstudie umfasst der Zeitraum vom Schaltungsentwurf bis zu den Messergebnissen insgesamt fünf Jahre und fünf Monate.

5.2.3 Eingesetzte Software-Werkzeuge

Der Entwurf des Viterbi-Entzerrer-Chips erfolgt mit dem IC-Entwurfswerkzeug für integrierte Schaltungen „Cadence-IC“ zunächst in der Version 5.0.33. Später erfolgt ein Umstieg auf Version 5.1.41. Cadence-IC stellt einen Schaltplaneditor zur Eingabe der Schaltungen auf Transistorebene zur Verfügung. Für die Layouterstellung stellt Cadence den Editor „Virtuoso“ zur Verfügung. Mithilfe des Werkzeugs „Spectre“ von Cadence werden die Netzlisten sowohl der Transistorschaltungen mit idealen Verbindungsleitungen als auch der extrahierten, mit parasitären Elementen behafteten Netzliste des Layouts simuliert. Bei Spectre handelt es sich um einen analogen Schaltungssimulator, der SPICE-Parameter der Halbleiterbauelemente versteht und das Schaltverhalten mittels Differenzialgleichungen berechnet.

Die Überprüfung der Entwurfsregeln (engl. Design Rule Check, DRC) als auch der Vergleich des Layouts mit dem Schaltplan (engl. Layout versus Schematic Check, LVS) erfolgt mit dem in Cadence eingebundenen Werkzeug „Calibre“ von Mentor Graphics.

Die Extraktion der parasitären Widerstände und Kapazitäten (engl. Parasitic Extraction, PEX) erfolgt mit „Star-RCXT“ von Synopsys. Auch dieses Werkzeug ist aus dem Layouteditor Virtuoso von Cadence heraus aufrufbar. Mit Star-RCXT können Netzlisten erstellt werden, die entweder nur die Verdrahtungswiderstände oder Kapazitäten zum Substrat oder Koppelkapazitäten enthalten. Die gleichzeitige Berücksichtigung aller drei Effekte ist ebenfalls möglich und wird in der Regel angewandt. Die extrahierte Netzliste lässt sich mit dem Cadence-Schaltplaneditor öffnen und darstellen. Allerdings sind die einzelnen Bauelemente nur über ihre Knotennamen miteinander verbunden. Eine grafische Darstellung der parasitären Elemente ist über den Layouteditor Virtuoso möglich. Die Simulation extrahierter Netzlisten ist mit Spectre bis hinauf zur Register-Transfer-Ebene sinnvoll. Beispielsweise umfasst die Netzliste zweier miteinander verbundener ACSUs einen Umfang von 26404 Knoten, 5560 Transistoren, 77142 parasitären Kapazitäten und 56457 parasitären Widerständen. Die Dauer einer Zeitbereichssimulation über 6 ns beträgt rund 35 Minuten auf einer CPU mit 2,4 GHz. Für Simulationen auf höherer Ebene sind Simulationen auf Basis detaillierter Transistormodelle nicht mehr notwendig, da das zeitliche Verhalten der kombinatorischen Logik nur zwischen zwei Flipflops interessant ist. Für Simulationen einer Transistorschaltung auf algorithmischer oder gar Systemebene bietet Cadence daher den „Fast-SPICE“ Simulator „UltraSim“ an, der Abstrahierungen an den Transistormodellen vornimmt. Durch Partitionierung der Schaltung wird die Simulationsdauer weiter beschleunigt. Für die Implementierung und Simulation in VHDL werden die Werkzeuge „HDL-Designer“ und „ModelSim“ von Mentor Graphics eingesetzt.

5.3 Chipfertigung

Nachdem die komplette Viterbi-Entzerrer-Schaltung auf Schaltplan- und Layoutebene erstellt ist, korrekte Simulationsergebnisse liefert, DRC und LVS in Ordnung sind, werden die Layoutdaten mit Hilfe von Cadence-IC in das GDSII-Format übersetzt, welches eine Beschreibung aller geometrischen Formen für den Maskenentwurf jedes Prozessschritts enthält. Anschließend werden die Fertigungsdaten über eine Internetverbindung an den Halbleiterhersteller übertragen. In diesem speziellen Fall werden die Prozessdaten nicht direkt an den Halbleiterhersteller ST Microelectronics übergeben, sondern an den Zwischenhändler (engl. Chipbroker) CMP, der Schaltungsentwürfe von diversen Institutionen sammelt. Dadurch ist es möglich, kleine Flächen ab 1 mm² einer dünnen Scheibe aus Halbleitermaterial, auf die integrierte Schaltungen aufgebracht werden (engl. Wafer), zu belegen.

CMP führt ebenfalls einen DRC durch und prüft Antennenregeln. Lange Leitungen, die an einer Gate-Elektrode eines MOSFETs enden, wirken bei der Prozessierung als Antenne. Das heißt, dass sich bei der Prozessierung in der Halbleiterfabrik auf der Leitung Ladungen ansammeln, die zum Durchbruch des Gate-Oxids führen können, was zur elektrostatischen Zerstörung der Schaltung führt. Davon sind insbesondere die Eingangsschnittstellen und große Metallflächen betroffen. Werden solche Antennen festgestellt, müssen am Layout Nachbesserungen durchgeführt werden. Hierzu muss in die betreffende Leitung eine Substratdiode eingefügt werden, die ein Aufladen der Gate-Elektrode gegenüber dem Substrat verhindert.

Weiterhin werden von CMP Füllstrukturen auf den Metallebenen eingefügt, um eine Mindestmetалldichte auf jeder Metalllage zu erreichen. Dies ist notwendig, damit die Ebenen planar bleiben. In gewissen Grenzen können diese Füllstrukturen unterdrückt werden, um die Hochfrequenzeigenschaften z.B. von Spulen auf dem Chip zu verbessern. Ein positiver Nebeneffekt ist, dass die Schaltungsstruktur sichtbar bleibt und mit einem Mikroskop betrachtet werden kann (siehe Abbildung 5.2 und Abbildung 5.3). Bei digitalen Schaltungen, wie bei diesem Viterbi-Entzerrer, spielt der Einfluss der Füllstrukturen auf das Schaltverhalten nur eine untergeordnete Rolle.

Ferner fügt CMP einen Versiegelungsring (engl. Seal ring, crack stop) um die Schaltung herum hinzu, damit der Chip am Rand eine besondere Festigkeit erhält und beim späteren Vereinzeln des Wafers keinen Schaden nimmt.

Die eigentliche Herstellung des Wafers in der Halbleiterfabrik dauert in der Regel drei Monate und umfasst über 20 Prozessschritte, die jedoch aus mehreren Teilschritten bestehen. Im Wesentlichen handelt es sich dabei um folgende Arbeitsabläufe [66].

Nach dem Reinigen des Silizium-Wafers, der aus p-Substrat besteht, wird durch thermische Oxidation Feldoxid aufgebracht. Nach Aufbringen von Fotolack und einer Maske, die die n-Wannen ausspart, wird mit UV-Licht bei einer Wellenlänge von 193 nm belichtet. Nach der Belichtung wird die Maske entfernt und der Fotolack entwickelt, damit er an den belichteten Stellen entfernt werden kann. Anschließend wird das Feldoxid mittels Fluorwasserstoffsäure

weggeätzt. Der restliche Fotolack wird mit Aceton entfernt. Nun liegt das p-Substrat frei und wird über Ionenimplantation oder Diffusion mit Arsenatomen [51], einem Element der fünften Hauptgruppe des Periodensystems, dotiert. Dotieren bezeichnet den Vorgang des Einbringens von Fremdatomen, um die elektrischen Eigenschaften eines Halbleitermaterials lokal zu ändern. Arsen hat ein freies Elektron in seiner äußersten Schale. Daher bildet das Arsen die n-Wanne. Das Feldoxid wird über der n-Wanne wieder lokal aufoxidiert. Die nächsten Schritte laufen nach dem gleichen Prinzip ab. Die Teilschritte für das Aufbringen des Fotolacks, der Masken, des Belichtens, des Entfernens des Fotolacks, des Ätzens usw. wiederholen sich für alle folgenden Prozessschritte und werden daher im Folgenden nicht mehr detailliert erläutert.

Um die Schwellenspannung der diversen NMOS- und PMOS-Transistoren auf die gewünschten Werte einzustellen, wird Borimplantation durchgeführt [40]. Dabei gilt, je höher die Borkonzentration im Kanal ist, desto höher ist die Schwellenspannung.

Der nächste größere Schritt dient der Erstellung der aktiven Transistorgebiete. Hierzu wird mit den beschriebenen Verfahren das Feldoxid abgeätzt und eine definierte Dicke von 1,6 nm für die GP-Transistoren aufgewachsen. Darüber wird Polysilizium abgeschieden, welches später die Gate-Elektrode bildet. Durch Trockenätzen wird das Polysilizium an den Stellen, an welchen es nicht benötigt wird, wieder entfernt. Durch Ionenimplantation mit Arsenatomen werden die Source- und Drain-Gebiete der n-Kanaltransistoren erstellt. Die Source- und Drain-Gebiete der p-Kanaltransistoren werden danach ebenfalls durch Ionenimplantation jedoch mit Bor- oder Indiumatomen [51] erstellt. Die chemischen Elemente Bor und Indium befinden sich beide in der dritten Hauptgruppe des Periodensystems und weisen eine Elektronenfehlstelle in ihrer äußersten Schale auf. Im nächsten Schritt werden die Kontakte der Source- und Drain-Gebiete sowohl der NMOS- als auch der PMOS-Transistoren durch Abscheiden von Kupfer gebildet. Nun sind alle Transistoren erstellt. Die nächsten Prozessschritte erstellen die Metallisierung aus Kupfer einschließlich der Füllstrukturen der insgesamt sieben Verdrahtungsebenen. Als oberste Lage auf dem Wafer wird eine Passivierungsschicht aus Nitrid aufgetragen, um die Schaltung nach außen hin elektrisch zu isolieren. Die Pads, die zum Kontaktieren der Schaltung dienen, werden freigeätzt. Zwischen den einzelnen Schritten des Abscheidens der Metalle, erfolgt ein chemisch mechanisches Polieren der Oberfläche, damit die Oberfläche aller Schichten eben wird. Hierfür ist eine Mindestmetalldicke notwendig, die durch die Füllstrukturen sichergestellt wird.

Nach der Prozessierung des Wafers durch ST Microelectronics erhält CMP den fertigen Wafer zurück und sägt die einzelnen Chips heraus. Außerdem wird der Wafer von CMP noch auf eine Dicke von 185 µm gedünnt. Die Standarddicke eines Wafers beträgt 600 µm. Das Dünnen ist notwendig, damit der Chip dieselbe Dicke aufweist als der vorzuschaltende AD-Wandler, um möglichst kurze Bonddrähte zu erhalten. Außerdem wird die Wärmeableitung nach unten hin verbessert.

Anschließend werden die Chips an die verschiedenen Kunden versandt. Jeder Kunde erhält standardmäßig 25 Chips. Abbildung 5.2 zeigt ein Foto des fertig prozessierten Chips. Der ausgesägte Chip ist einschließlich Versiegelungsring und Sägestraßenzuschläge 4,27 mm breit,

1,61 mm hoch und 0,185 mm dick. Das gezeichnete Layout hat hingegen nur eine Breite von 4,08 mm und eine Höhe von 1,40 mm.

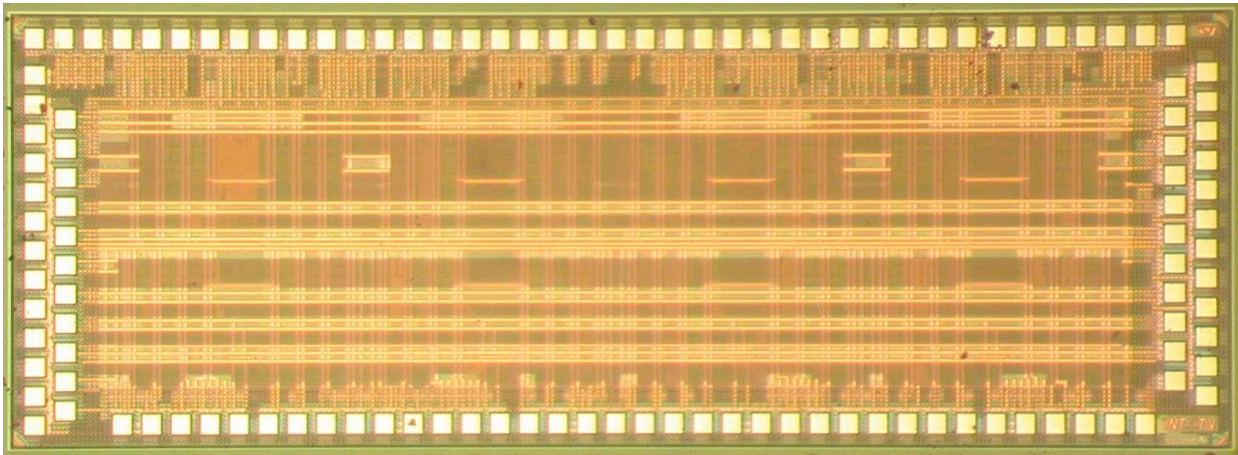


Abbildung 5.2: Foto des prozessierten Viterbi-Entzerrer-Chips

Die folgenden Abbildungen zeigen mikroskopierte Ausschnitte aus dem prozessierten Viterbi-Entzerrer-Chip.

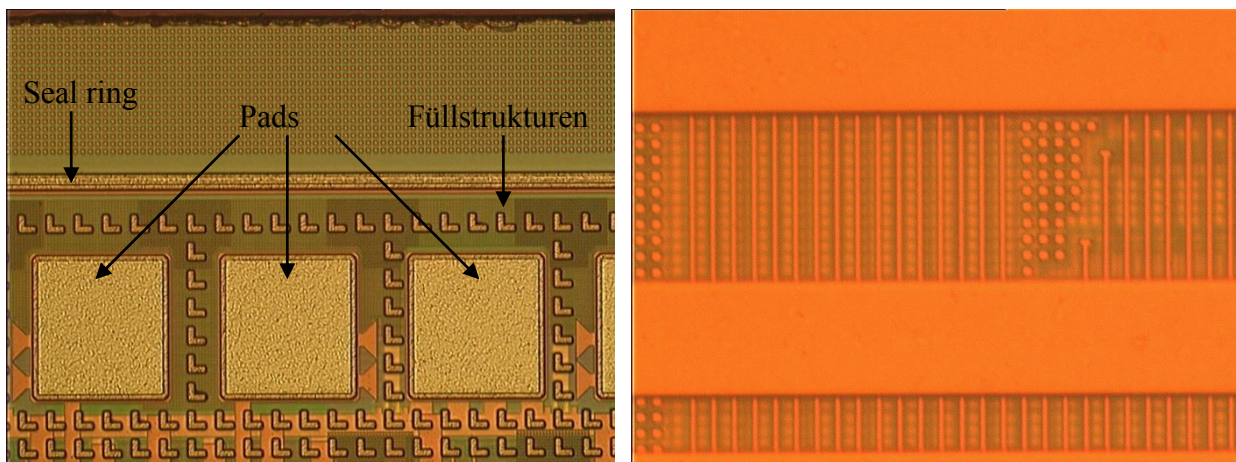


Abbildung 5.3: Ausschnitte aus dem prozessierten Viterbi-Entzerrer-Chip

Das linke Foto in Abbildung 5.3 zeigt die obere Chipkante, die vom Sägen ausgefrant ist. Darunter folgen Füllstrukturen zur Erfüllung der Metaldichte. Die waagrecht verlaufende Linie ist der Versiegelungsring (engl. Seal ring), der dazu dient, dass die eigentliche Schaltung beim Sägen nicht beschädigt wird. Der Versiegelungsring hat eine Breite von $13\ \mu\text{m}$. Auffällig sind die L-förmigen Strukturen. Hierbei handelt es sich um Füllstrukturen der obersten Ebene, die aus Aluminium besteht. Weiterhin sind drei Pads zu sehen. Die Pads sind $75\ \mu\text{m}$ breit und $80\ \mu\text{m}$ hoch. Das linke und mittlere Pad dienen der Zuführung der $10,7\ \text{Gbit/s}$ Signale. Man erkennt an beiden Pads jeweils zwei sich verbreiternde Metallstrukturen, wobei es sich um die Anschlüsse der $50\ \Omega$ Abschlusswiderstände handelt. Die Verbreiterung ist notwendig, damit der daran

angeschlossene Widerstand aus Polysilizium durch den Stromfluss thermisch nicht zerstört wird. Das rechte Pad dient der Spannungszuführung. Am unteren Padrand sind vier 12 µm breite Leitungen auf Metallebene 7 zu erkennen, die auf einen Versorgungsspannungsring innerhalb des Padrings führen.

Das rechte Foto in Abbildung 5.3 zeigt einen Ausschnitt aus der Mitte des Chips. Die dicken, waagrecht verlaufenden Leitungen sind Teil des auf Metalllage 7 verlaufenden Versorgungsspannungsgitters und haben eine Breite von 12 µm. Die dünneren senkrecht verlaufenden Leitungen sind Teil der Verdrahtung der Trellisstruktur auf Metalllage 6. Die Breite dieser Leitungen beträgt 0,42 µm. Am Ende der siebten und achten Leitung von rechts befindet sich eine Durchkontaktierung von Metallebene 6 auf Metallebene 5. Die daran folgende Leitung verläuft in waagrechtlicher Richtung nach rechts. Diese ist auf der Originalabbildung noch als leichter Schatten erkennbar, da diese nur noch eine Breite von 0,14 µm aufweist und somit eine Größe unterhalb der sichtbaren Lichtwellenlänge aufweist. Weiterhin erkennt man runde Metallstrukturen auf verschiedenen Ebenen. Hierbei handelt es sich wieder um Füllstrukturen.

6 Messaufbau und Messergebnisse

6.1 Aufbautechnik

Um den Chip vermessen zu können, muss er auf eine Platine aufgebaut werden. Aufgrund der großen Padanzahl ist eine Vermessung auf einem Wafer-Prober mit am Institut vorhandenen Messspitzen nicht möglich. Als Substratmaterial wird das Hochfrequenzmaterial RF-60A [67] von Taconic mit einer hohen Dielektrizitätszahl von $\epsilon_r = 6,15$ verwendet. Das Substrat mit einer Dicke von $250 \mu\text{m}$ ist mit einer zweiseitigen Kupfereauflage von jeweils $18 \mu\text{m}$ beschichtet.

Beim Layout der Platine wird darauf geachtet, dass alle Hochgeschwindigkeitsleitungen eine maximale Toleranz von weniger als 1 mm aufweisen.

Die Länge eines Bits auf einer Leitung kann wie folgt bestimmt werden:

Die minimale Bitdauer beträgt $t_b = \frac{1}{10,7 \text{ Gbit/s}} = 93,5 \text{ ps}$. Legt man eine

Ausbreitungsgeschwindigkeit der elektromagnetischen Welle von ungefähr der Hälfte der Lichtgeschwindigkeit c im Medium zugrunde, ergibt sich als Bitlänge

$$l_b = \frac{1}{2} c t_b \approx 1,4 \text{ cm} \quad (6.1)$$

Damit ist gezeigt, dass eine Abweichung der Leitungslänge auf der Platine von weniger als 1 mm mehr als ausreichend ist.

Die Gesamtlänge der Hochgeschwindigkeitsleitungen beträgt durchschnittlich $74,5 \text{ mm}$. Diese Leitungslänge ist notwendig, damit genügend Platz für die SMP-Steckverbinder vorhanden ist. Bei SMP-Steckern handelt es sich um hochfrequenztaugliche SMD-Koaxialverbinder, die einen geringeren Platzbedarf haben als SMA-Konnektoren. Die Leitungen sind in einem Halbkreis angeordnet, um nahezu identische Leitungslängen zu erreichen. Die Leitungen haben eine Breite von $75 \mu\text{m}$. Der Abstand zwischen den Leitungen beträgt ebenfalls $75 \mu\text{m}$. Vom Mittelpunkt der Platine ausgehend werden die Leitungen zunächst 2 bis 4 cm differenziell geführt. Danach erfolgt ein Übergang hin zu breiteren unipolaren Mikrostreifenleitungen. Die Führung als Mikrostreifenleitungen bleibt bis zu den SMP-Steckern bestehen. Die Dimensionierung der Mikrostreifenleitungen auf einen Wellenwiderstand von 50Ω erfolgt mit Hilfe des in ADS integrierten Werkzeugs „LineCalc“. Die differenziell geführten Leitungsabschnitte haben einen differenziellen Wellenwiderstand von 100Ω . Unter den gegebenen Substratparametern ergibt sich eine Leitungsbreite von $320 \mu\text{m}$.

An der Position auf der Platine, an der der Chip angebracht werden soll, ist eine Metallfläche mit Durchkontaktierungen auf die Platinenunterseite vorgesehen. Diese dient einerseits der Wärmeableitung und andererseits lassen sich die Massepads des Chips leicht kontaktieren. Um die Wärmeabfuhr zu verbessern, ist der Chip mit einem metallischen Kleber auf der Platine

befestigt. Weiterhin ist eine aktive Kühlung mittels Peltier-Element und ventilatorgekühlten Kühlkörpern vorgesehen, um die erwartete thermische Verlustleistung abzuführen. Die Leistungsdichte beträgt $0,37 \text{ W/mm}^2$. Da dies der Leistungsdichte aktueller Computerprozessoren (Intel Pentium Core™2 Duo bei maximaler Taktfrequenz: $0,373 \text{ W/mm}^2$ [68], [69]) entspricht, wird eine aufwändige Kühlung in Betracht gezogen.

Der Chip ist mit insgesamt 123 Bonddrähten elektrisch mit der Platine verbunden. Als Bonddraht wird ein Golddraht mit $25 \mu\text{m}$ Durchmesser verwendet. Die Anordnung der Bondpads auf der Platine ist so gewählt, dass die Hochfrequenz-Bondleitungen eine maximale Länge von 2 mm nicht überschreiten. Anhand von Simulationen wird sichergestellt, dass bei einer Datenfrequenz von 5 GHz die Dämpfung vernachlässigbar ist.

Die interne CMOS-Logik auf dem Chip weist einen hohen dynamischen Stromanteil (vgl. Abbildung 4.60) auf. Daher werden auf der Platine im Abstand von ca. 2 mm vom Chip zwölf Blockkondensatoren angebracht, die eine Gesamtkapazität von 628 nF haben.

Der Aufbau des Viterbi-Entzerrer-Chips auf einer Platine ist in Abbildung 6.1 dargestellt.

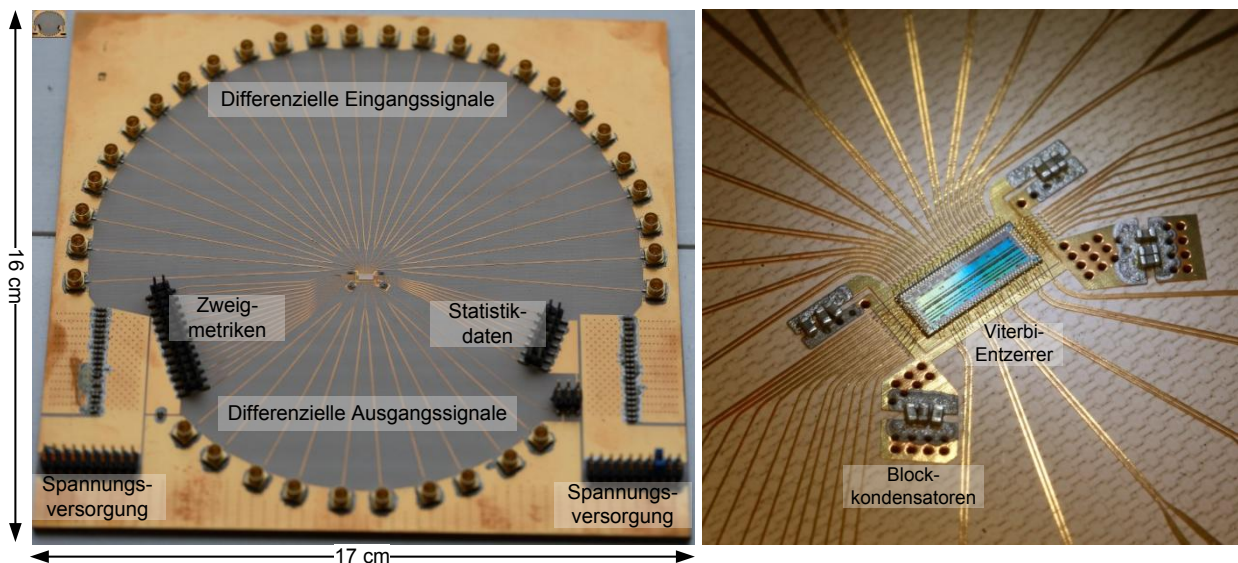


Abbildung 6.1: Aufbau des Viterbi-Entzerrer-Chips auf einer Platine

6.2 Funktionstest

Zunächst soll mit Hilfe eines einfachen Aufbaus die korrekte Funktion des Viterbi-Entzerrers als digitale Logikschaltung überprüft werden. Erst danach werden deutlich aufwändigere Geschwindigkeitstests durchgeführt.

6.2.1 Messaufbau

Für den Funktionstest kommt als Datenquelle der Virtex-4 FPGA XC4VFX100 von Xilinx zum Einsatz. Der FPGA ist auf einer Leiterplatte (hier: ML423-Brett der „Virtex-4 FX FPGA RocketIO Characterization Platform“ [70]) aufgebaut, welche über diverse Schnittstellen verfügt, worüber mit dem FPGA kommuniziert werden kann. Charakteristisch für diesen FPGA sind SelectIO-Schnittstellen für langsame Datenübertragung von mehreren hundert Megahertz und schnelle RocketIO-Schnittstellen für Datenraten von bis zu 10 Gbit/s. Da es bei einem Funktionstest nicht auf die Geschwindigkeit ankommt, können die langsamen SelectIO-Schnittstellen verwendet werden. Der Vorteil der SelectIO-Schnittstellen gegenüber den RocketIO-Schnittstellen besteht darin, dass parallel anliegende Daten synchron zueinander sind. Die physikalischen Spannungspegel der SelectIO-Schnittstellen betragen für den logischen High-Pegel 2,5 V und für den Low-Pegel 0 V. Da der Viterbi-Chip für eine Nominalspannung von 1,0 V ausgelegt ist, muss der Pegel der SelectIO-Schnittstellen über eine Spannungsteilerplatine an die Pegel des Viterbi-Chips angepasst werden. Eine Adapterplatine wird sowieso benötigt, um die unterschiedlichen Steckverbindungen aufeinander anzupassen.

Der FPGA wird in VHDL mit zwölf Speichern programmiert. Jeder Speicher enthält eine Bitfolge von 48 bit. Der Speicherinhalt von drei Speichern entspricht einer Symbolfolge, da die Viterbi-Eingangsdaten mit 3 bit quantisiert sind. Diese Symbolfolge ist mit Symbolnebensprechen versehen, wie es für optische Kanäle charakteristisch ist. Da die Viterbi-Eingangsschnittstelle vierfach parallel ausgelegt ist, werden insgesamt zwölf Speicher benötigt. Der gesamte Speicherinhalt entspricht daher 192 Symbolen. Für den Funktionstest wird diese Symbolfolge in einer unendlichen Schleife aus den Speichern ausgelesen und dem Viterbi-Chip zugeführt. Für den Test ist eine in sich wiederholende Datenfolge vorgesehen, da einerseits die Speicherkapazität der Datenquelle begrenzt ist und andererseits die entzerrten Daten auf einem Sampling-Oszilloskop dargestellt und analysiert werden sollen.

In einem weiteren Speicher auf dem FPGA werden die Zweigmetriken abgelegt, die zu Beginn des Funktionstests in den Zweigmetrikspeicher des Viterbi-Entzerrer-Chips eingeschrieben werden.

Die Speicher auf dem FPGA lassen sich über eine RS232-Schnittstelle auf der FPGA-Platine über einen Computer beschreiben. Daher können relativ einfache Tests mit verschiedenen Symbolfolgen und Zweigmetriken vorgenommen werden, ohne dass der VHDL-Code angepasst und der FPGA neu programmiert werden muss. Details zum VHDL-Entwurf sind in [71] dokumentiert.

6.2.2 Messergebnisse

Mit den langsamen FPGA-Schnittstellen kann der Viterbi-Chip mit einer maximalen Taktfrequenz von 12,5 MHz betrieben werden. Die Datenrate ergibt sich aus der achtfachen Taktfrequenz zu 100 Mbit/s. Die Begrenzung der Taktfrequenz resultiert aus dem speziellen VHDL-Design, das es ermöglicht, den Takt gegenüber den Daten zeitlich zu verschieben.

Tabelle 6.1: Messergebnisse zum Leistungsverbrauch des Viterbi-Entzerrers bei 12,5 MHz

Spannung am Chip	Strom	Leistung Gesamtchip
0,90 V	1,34 A	1,21 W
1,00 V	1,58 A	1,58 W
1,20 V	2,12 A	2,54 W

Messungen bei ein- und ausgeschalteter Taktfrequenz zeigen im Rahmen der Messgenauigkeit keine unterschiedliche Stromaufnahme. Daher lässt sich schlussfolgern, dass der Leistungsverbrauch des eigentlichen Viterbi-Rechenwerkes im Milliwattbereich liegt. Der meiste Strom wird also nur durch die vorgeschalteten CML-Demultiplexer und nachgeschalteten CML-Multiplexer hervorgerufen, die nichts zur eigentlichen Funktion des Viterbi-Algorithmus beitragen. Die CML-Bausteine dienen nur einer schlanken Schnittstelle, um Chipfläche zu sparen.

Abbildung 6.2 zeigt die vier Ausgänge (Signale M0 bis M3) des Viterbi-Entzerrer-Chips bei Anliegen der zwölf Eingangssignale mit typischer ISI optischer Übertragungskanäle. Die Darstellung erfolgt auf dem TLS 216 Logic Scope von Tektronix. Bei dieser Messung liegt an jedem Eingang zweimal hintereinander die gleiche 48 bit lange Folge an, um zu untersuchen, ob es aufgrund der Speicherwirkung des Viterbi-Algorithmus Auswirkungen auf das Ergebnis am Ende der ersten Datenfolge und am Anfang der nächsten Datenfolge gibt. Die dazu passenden Zweigmetriken sind ebenfalls eingeschrieben. Kanal 5 zeigt den Halbratentakt am Ausgang des Viterbi-Entzerrer-Chips. Kanal 6 zeigt den um den Faktor 64 herunter geteilten Takt an, der für die Kanalschätzung benötigt wird. Für den Funktionstest ist die Taktfrequenz unerheblich. Sie ist hier beispielhaft zu 390,4 kHz gewählt. Der durch 64 geteilte Takt entspricht dann 6,1 kHz.

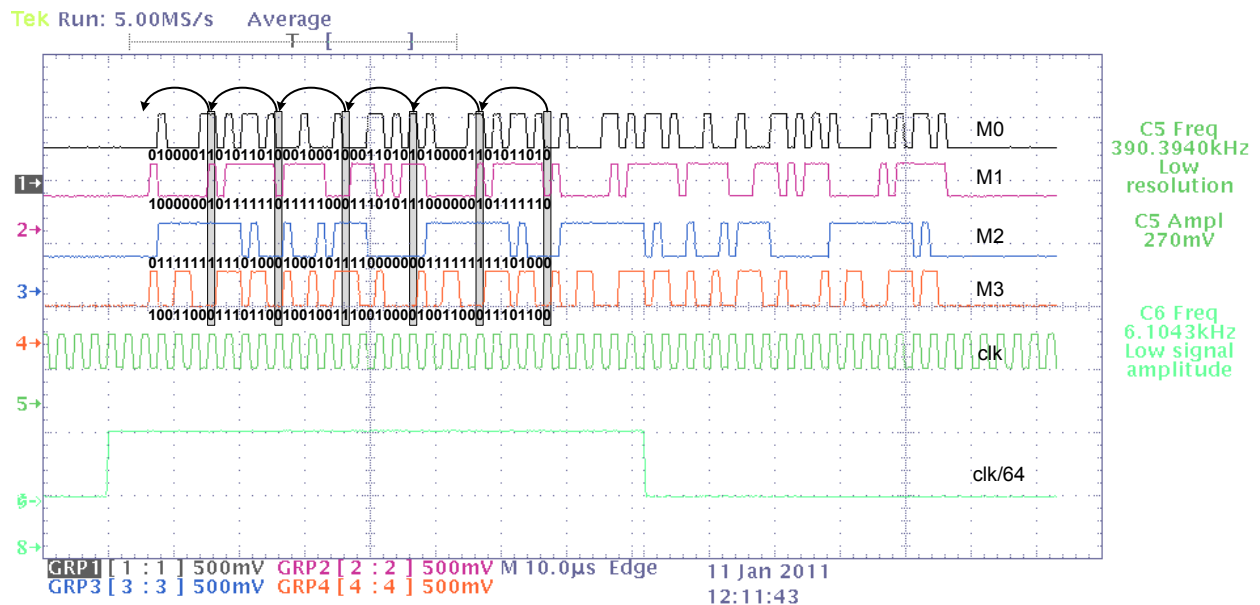


Abbildung 6.2: Ausgänge des Viterbi-Entzerrers

Setzt man die vier parallel an den Ausgängen anliegenden Daten zu einem seriellen Datenstrom zusammen und vergleicht ihn mit dem Originaldatenstrom von vor der Verzerrung, fällt auf, dass immer eine Folge von 28 bit korrekt entzerrt wird. Die weiteren vier aufeinanderfolgenden Bits (in Abbildung 6.2 grau hinterlegt) sind zwar korrekt, aber um ein Zeitfenster von 32 bit verschoben (in Abbildung 6.2 durch Pfeile gekennzeichnet). Die Ursache hierfür liegt an einem 4 bit Register, das benötigt wird, damit die Simulationsergebnisse korrekt sind. Eine algorithmische Erklärung gibt es für dieses Register nicht. In der Messung hat sich gezeigt, dass dieses Register in Wirklichkeit auch nicht notwendig ist. Daher wird dieser Fehler auf ein Problem in der Simulationsumgebung zurückgeführt. Eine tiefer gehende Behandlung des 4 bit Registers erfolgt nicht, da es weder eine Erklärung auf algorithmischer Ebene gibt noch aus messtechnischer Sicht notwendig ist.

Wird dieser Viterbi-Entzerrer-Chip einmal in einem kommerziellen System eingebaut, kann z.B. über die Firmware eines folgenden DSPs die Reihenfolge der Bits korrigiert werden. Somit wird ein teures Redesign bzw. die kostenintensive Herstellung eines neuen Maskensatzes vermieden.

6.3 Hochgeschwindigkeitsmessungen

Der Virtex-4-FPGA besitzt sogenannte RocketIO-Schnittstellen, über die Datenraten bis zu 10 Gbit/s gesendet als auch empfangen (engl. Transceiver) werden können. In dieser Arbeit werden die RocketIO-Schnittstellen nur als Datenquelle eingesetzt, da die Darstellung der entzerrten Daten auf dem Sampling-Oszilloskop HP 54750A erfolgt.

Im Gegensatz zur langsamen SelectIO-Schnittstelle sind die Ausgangsdaten der RocketIO-Schnittstelle nicht zueinander synchron. Die Ursache liegt darin, dass jedes RocketIO-Paar eine eigene PLL enthält, die die Frequenz zum Serialisieren des Datenstroms aus den Speichern des FPGAs erzeugt. Da diese Frequenz unabhängig von der Frequenz ist, mit welcher das im FPGA befindliche VHDL-Design läuft und die PLLs zusätzlich unabhängig voneinander los laufen, wird mit dem Auslesen der Speicher an verschiedenen Adressen und mit der Serialisierung an verschiedenen Positionen innerhalb eines 32 bit Wortes begonnen. Das hat zu Folge, dass im VHDL-Entwurf eine Möglichkeit zum wortweisen und bitweisen Schieben der Bitfolgen jedes einzelnen Speichers vorgesehen werden muss. Erschwerend kommt noch hinzu, dass die Leitungslängen der RocketIO-Kanäle auf der FPGA-Platine sich teilweise um mehr als 8 cm unterscheiden. Bei einer Bitlänge von ca. 1,5 cm bei 5 GHz ist es daher nicht möglich, die Daten bitsynchron an zwölf RocketIO-Kanälen parallel zur Verfügung zu stellen. Deshalb kann jetzt schon vorhergesagt werden, dass eine Messung bei einer Datenrate von 43 Gbit/s mit dem FPGA als Datenquelle schwer realisierbar ist.

Das VHDL-Design, das die Speicher für die Daten und Zweigmetriken sowie die serielle Schnittstelle implementiert, ist identisch zu dem Design, mit dem der Funktionstest durchgeführt wird. Einzig an den Leseschnittstellen der Datenspeicher sind jetzt die RocketIO-Schnittstellen anstelle der SelectIO-Schnittstellen angeschlossen. Details zum VHDL-Entwurf können [71] entnommen werden. Weiterhin werden ein Speicher, der als Taktquelle des Viterbi-Entzerrer-Chips dient, und ein weiterer Speicher, der ein Triggersignal für das Sampling-Oszilloskop zur Verfügung stellt, vorgesehen.

6.3.1 Vorbereitende Maßnahmen und Messaufbau

Die Spannungspegel der RocketIO-Ausgangsschnittstelle des FPGAs liegen bei Belastung mit 50Ω bei 1,5 V für den High- und 0,9 V für den Low-Pegel. Die Ausgänge sind gegen die positive Versorgungsspannung von 1,5 V terminiert. Die Abschlusswiderstände der Eingänge des Viterbi-Entzerrers sind schaltbar und können sowohl gegen die positive als auch gegen die negative Versorgungsspannung terminiert werden. Die Eingangsdifferenzstufe ist auf einen Spannungshub von 300 mV ausgelegt. Allerdings beträgt die positive Versorgungsspannung des Viterbi-Entzerrers 1 V. Die Hubspannung von 600 mV wird aber nur bis zu einer Signalfrequenz von rund 2 GHz erreicht. Bei Frequenzen um 5 GHz werden nur noch 200 bis 300 mV erreicht. Da die Information aber in der Differenz der beiden Spannungen steckt, werden keine Schwierigkeiten aufgrund der Pegelunterschiede zwischen FPGA- und Viterbi-Versorgungsspannung sowie des geringen Differenzspannungshubes erwartet. Die Ausgänge des FPGAs werden daher direkt mit den Eingängen ohne Pegelanpassung oder Pegelregeneration miteinander verbunden.

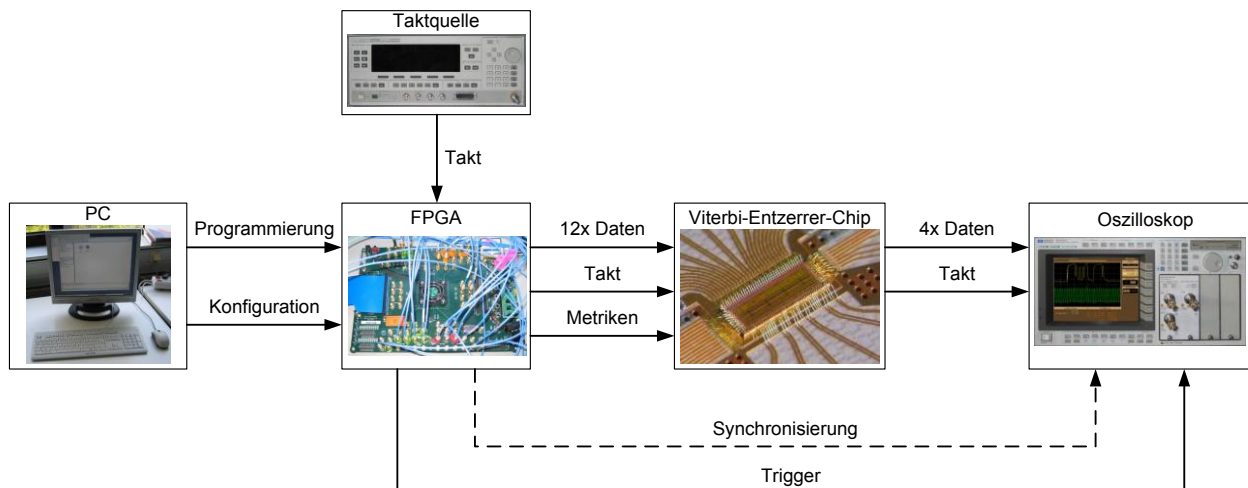


Abbildung 6.3: Schematischer Aufbau der Messumgebung des Viterbi-Entzerrers

In Abbildung 6.3 ist der schematische Aufbau der Messumgebung des Viterbi-Entzerrers dargestellt. Über einen Frequenzgenerator werden die Multi-Gigabit-Transceiver (MGT) des FPGAs mit einem Takt versorgt, welcher in einem bestimmten Verhältnis zur RocketIO-Datenrate steht. Dieses Verhältnis lässt sich über den VHDL-Code konfigurieren. Nach der Programmierung des FPGAs und Inbetriebnahme des MGT-Taktgenerators, wird in jeden der zwölf Datenspeicher eine Bitsequenz mit einer Länge von 256 bit eingespeichert, welche zur Synchronisierung dient. Die Bitfolgen zweier RocketIO-Kanäle werden nun wiederholend und gleichzeitig auf dem Sampling-Oszilloskop ausgegeben (gestrichelte Verbindung), wovon ein Kanal als Referenz benutzt wird. Über einen Taster auf dem FPGA-Brett kann die Speicheradresse, ab der ein 32 bit-Wort gelesen werden soll, angepasst werden, bis die beiden Bitmuster um weniger als ein Wort zeitlich verschoben sind. Mithilfe eines in VHDL implementierten Barrel-Shifters kann die Startposition der Serialisierung über einen weiteren Taster verändert werden. Damit kann die Bitfolge solange bitweise verschoben werden, bis der Phasenunterschied beider Folgen weniger als ein Bit beträgt. Dieser Vorgang wird nun mit allen weiteren RocketIO-Kanälen durchgeführt. Danach werden die RocketIO-Ausgänge am Viterbi-Entzerrer angeschlossen.

Bei dieser Art der Synchronisierung beträgt die Phasendifferenz im ungünstigsten Fall aufgrund der unterschiedlich langen Leitungen auf dem FPGA-Brett immer noch 0,5 Einheitsintervalle (engl. Unit-Interval, UI). Daher ist die maximale Datenrate, mit der der Viterbi-Entzerrer-Chip betrieben werden kann, begrenzt. Eine noch genauere Übereinanderlagerung der Bitsequenzen ist nur durch analoge Verzögerungsleitungen oder Phasenschieber realisierbar.

Um auf digitalem Wege Phasendifferenzen von maximal 0,25 UI oder 0,125 UI zu erreichen, kann man unter Beibehaltung der MGT-Taktfrequenz jedes Bit, das über die RocketIO-Schnittstelle übertragen wird, zwei- oder gar viermal senden [72]. Dadurch dass die Bitperiode verdoppelt bzw. vervierfacht wird, die Phasendifferenz aber konstant bleibt, wird die

Augenöffnung größer. Der Nachteil dieses Verfahrens ist aber, dass sich die effektive Bitrate halbiert oder sogar viertelt. Der vollständige Messaufbau ist in Abbildung 6.4 dargestellt.

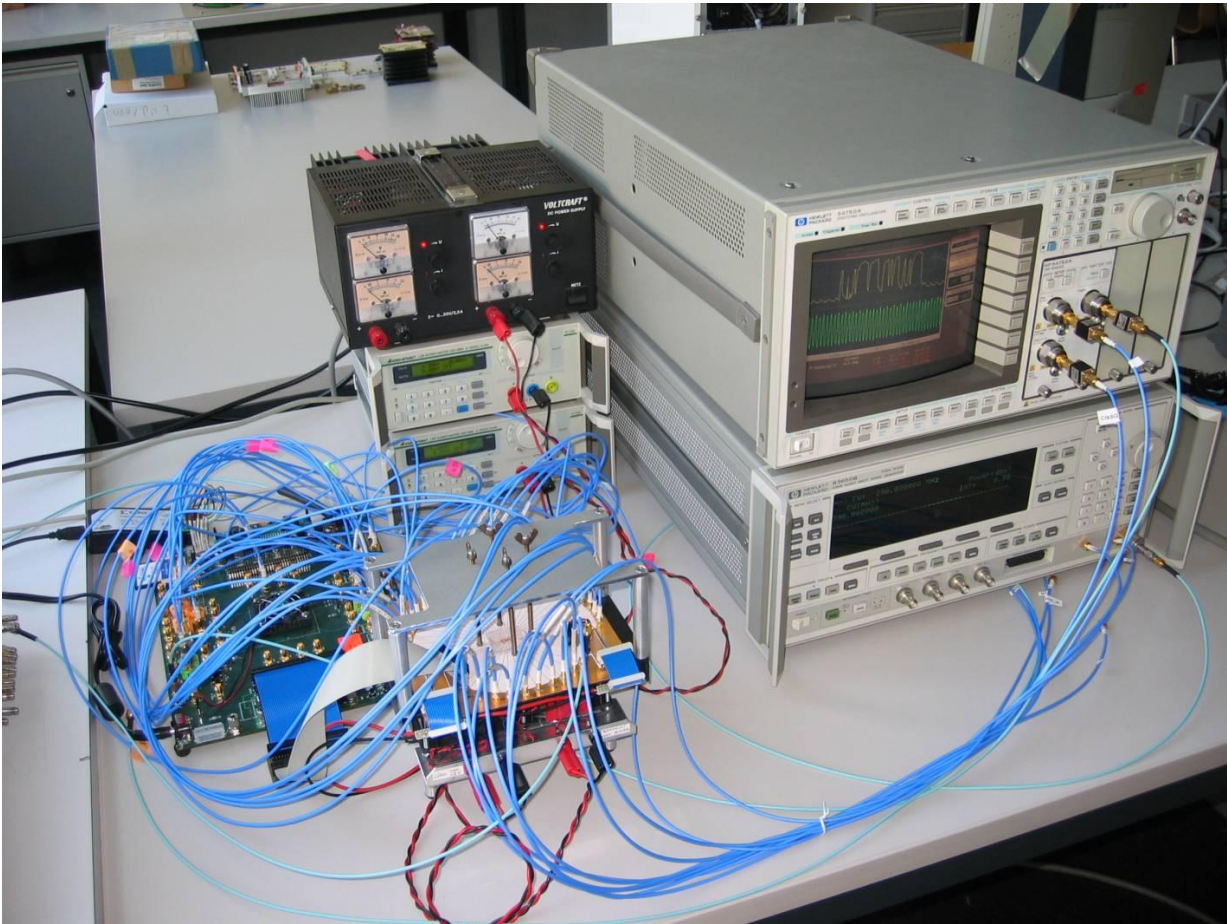


Abbildung 6.4: Vollständiger Messaufbau des Viterbi-Entzerrers

6.3.2 Messergebnisse

Tabelle 6.2 zeigt die Strom- und Leistungsaufnahme des Viterbi-Entzerrer-Chips bei unterschiedlichen Taktfrequenzen und Versorgungsspannungen. Der 90 nm CMOS-Prozess definiert drei Eckspannungen: U_{\min} ist die minimale mögliche Betriebsspannung, U_{nom} ist die nominale Versorgungsspannung und U_{\max} ist die maximal erlaubte Betriebsspannung. Innerhalb des Bereiches zwischen minimaler und maximaler Versorgungsspannung soll nach Angaben des Halbleiterherstellers der ordnungsgemäße Betrieb gewährleistet sein. Die Messungen bestätigen diese Aussage.

Tabelle 6.2: Messergebnisse zum Leistungsverbrauch des Viterbi-Entzerrers bei verschiedenen Taktfrequenzen

Taktfrequenz		12,5 MHz		0,6 GHz		1,2 GHz		2,4 GHz		4,0 GHz	
Datenrate		100 Mbit/s		4,8 Gbit/s		9,6 Gbit/s		19,2 Gbit/s		32 Gbit/s	
Spannung		Strom	Leistung	Strom	Leistung	Strom	Leistung	Strom	Leistung	Strom	Leistung
U_{\min}	0,9 V	1,34 A	1,21 W	1,51 A	1,36 W	1,59 A	1,43 W	1,81 A	1,63 W	2,04 A	1,84 W
U_{nom}	1,0 V	1,58 A	1,58 W	1,79 A	1,79 W	1,89 A	1,89 W	2,12 A	2,12 W	2,39 A	2,39 W
U_{\max}	1,2 V	2,12 A	2,54 W	2,41 A	2,89 W	2,54 A	3,05 W	2,84 A	3,41 W	3,20 A	3,84 W

Bei langsamer Taktfrequenz (12,5 MHz) besteht zwischen ein- und ausgeschaltetem Taktsignal kein Unterschied hinsichtlich der Stromaufnahme im Rahmen der Messgenauigkeit. Daraus wird deutlich, dass fast die gesamte Leistung von den CML-Demultiplexern am Eingang und den CML-Multiplexern am Ausgang verbraucht wird. In der Simulation beträgt die Stromaufnahme des CML-Teils der Schaltung 1,55 A bei 1 V Versorgungsspannung. Von daher gesehen, stimmen Simulation und Messung ziemlich genau überein.

Der dominante Teil des dynamischen Leistungsverbrauchs von statischem CMOS resultiert aus dem Umladen der internen Lastkapazitäten. Mathematisch lässt sich die kapazitive dynamische Verlustleistung nach Gleichung (6.2) beschreiben.

$$P_{\text{dynC}} = pf C_L U_{\text{DD}}^2 \quad (6.2)$$

Daraus folgt ein linearer Zusammenhang zwischen dynamischer Verlustleistung P_{dynC} und Taktfrequenz f . Die Spannung U_{DD} und die Lastkapazität C_L sowie die Schaltwahrscheinlichkeit p , die im Mittel konstant sind, werden in der Konstanten κ zusammengefasst. Für die Messreihe aus Tabelle 6.2 ergibt sich $\kappa = 0,18 \text{ W/GHz}$. Daraus lässt sich bei der Nominalspannung von $U_{\text{DD}} = 1,0 \text{ V}$ und bei der Zieltaktfrequenz von $f = 5,35 \text{ GHz}$ eine dynamische Verlustleistung von $P_{\text{dynC}} = 0,963 \text{ W}$ berechnen. Addiert man die statische Verlustleistung des CML-Schaltungsteils von 1,58 W hinzu, kommt man auf 2,543 W. Dies entspricht sehr genau der simulierten Verlustleistung von 2,55 W.

Die vier Ausgangssignale M0 bis M3 des Viterbi-Entzerrer-Chips sind in Abbildung 6.5 bei einer Taktfrequenz von 4,0 GHz dargestellt. Die Ausgangssignale sind mit der berechneten bzw. gemessenen Bitfolge aus dem Funktionstest hinterlegt.

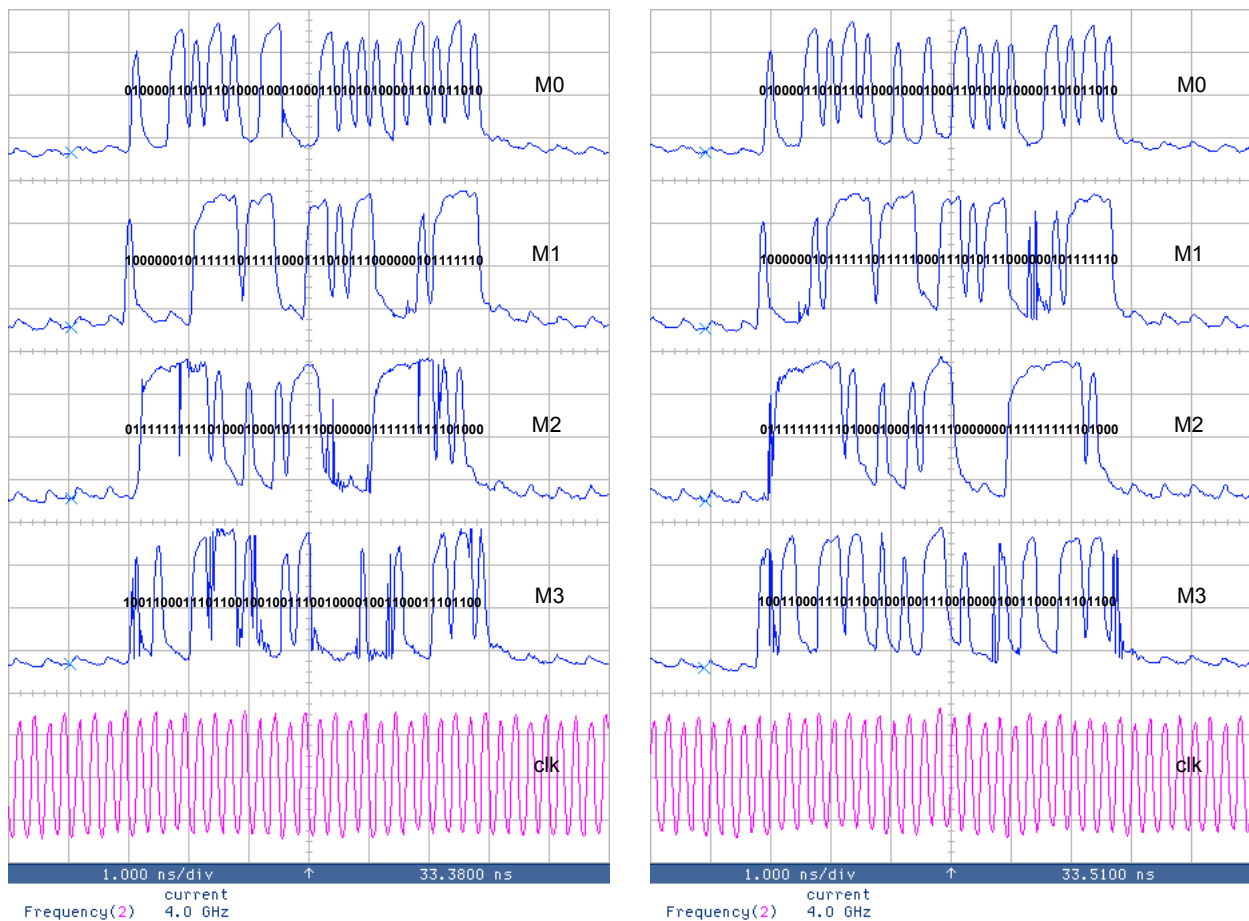


Abbildung 6.5: Ausgänge des Viterbi-Entzerrers mit (links) ausgeschalteter und (rechts) eingeschalteter Vorverzerrung der RocketIO-Ausgänge

Im linken Bild fällt auf, dass sich die Signalverläufe häufig von der erwarteten Bitsequenz unterscheiden. Dieses Problem ist der Phasenverschiebung von bis zu 0,5 UI der RocketIO-Ausgangssignale zuzurechnen. Die Ursache liegt darin, dass im ungünstigsten Fall die Flanken der Datensignale mit der Taktflanke zusammenfallen. Das erkennt man daran, dass bei einigen Signalen kurze Spannungsspitzen bzw. -einbrüche auftreten, welche auf zeitlichen Jitter bei in etwa übereinanderliegenden Takt- und Datenflanken und der daraus resultierenden Verletzung der Rüstzeiten der Flipflops zurückzuführen sind.

Das Signal der RocketIO-Kanäle lässt sich vorverzerrern. Dadurch erreicht man einerseits steilere Signalfanken und andererseits auch eine geringe zeitliche Verschiebung der Signale. Im rechten Bild sind die Ausgangssignale des Viterbi-Entzerrer-Chips mit eingeschalteter Vorverzerrung dargestellt. Vergleicht man in diesem Bild die Signalverläufe mit der erwarteten Bitfolge, erkennt man, dass bis auf ganz wenige Ausnahmen die Signalverläufe mit der hinterlegten Bitfolge übereinstimmen. Im Vergleich zwischen linkem und rechtem Bild erkennt man, dass es im rechten Bild keine echten falschen Bits mehr gibt, sondern an manchen Stellen nur Störimpulse aufgrund von Jitter im Eingangssignal auftreten.

Daher wird davon ausgegangen, dass der Viterbi-Entzerrer bei einer Datenrate von 32 Gbit/s unter der Voraussetzung, dass die Eingangsdaten ohne Phasenversatz anliegen, korrekt arbeitet. Der vollständige Beweis kann nur durch einen verbesserten Messaufbau erbracht werden.

Tabelle 6.3: Zusammenfassung der Leistungsdaten des Viterbi-Entzerrers auf Basis der Messergebnisse

Gesamtdatendurchsatz	32 Gbit/s
Interne DSP-Taktfrequenz	1,0 GHz
Leistung	2,39 W bei 1 V Versorgungsspannung
FOM 1: Energie pro Datenmenge	0,075 nJ/bit (Kehrwert: 13,4 bit/nJ)
FOM 2: Fläche pro Datendurchsatz	0,178 mm ² /Gbit/s (Kehrwert: 5,6 Gbit/smm ²)

Auf Basis der Messergebnisse (Tabelle 6.3) ist gezeigt, dass der entworfene Viterbi-Entzerrer mindestens dreimal so schnell ist, wie ein kommerziell verfügbares Produkt [14]. Hinsichtlich des Energieaufwandes, der für die Verarbeitung von 1 Gbit/s nötig ist, benötigt der in dieser Arbeit vorgestellte Viterbi-Entzerrer 2,5 mal weniger Energie als ein anderer in vergleichbarer Halbleitertechnologie veröffentlichter Chip [15].

Simulationen zeigen, dass der längste kombinatorische Pfad der Addier-Vergleichs-Auswahleinheit einschließlich Rüst- und Verzögerungszeiten der Flipflops 614 ps beträgt, wenn man typische Transistorparameter und typische parasitäre Widerstände und Kapazitäten des Verdrahtungsnetzwerks zugrunde legt. Bei 43 Gbit/s beträgt die Taktfrequenz des Viterbi-Rechenwerks 1,34 GHz, was einer Taktperiode von 746 ps entspricht. Demnach besteht ein Puffer von 132 ps. Daher lassen Simulationen weitaus höhere Datenraten erwarten.

On-Wafer-Messungen, bei denen nur der Takt und die Versorgungsspannung über Messspitzen am Viterbi-Entzerrer angeschlossen sind, zeigen, dass der interne Taktteiler noch bei einer Eingangstaktfrequenz von 6,1 GHz funktioniert, was einer Datenrate von 48,8 Gbit/s entspricht.

Anhand dieser beiden Indizien wird davon ausgegangen, dass der entworfene Viterbi-Entzerrer-Chip auch bei seiner Zieldatenrate von 43 Gbit/s korrekt funktioniert, wenn auch zum aktuellen Zeitpunkt der Messaufbau noch keine so schnellen Messungen zulässt.

6.3.3 Grenzen des Messaufbaus

Es zeigt sich, dass das Datensignal auf allen RocketIO-Kanälen, welches aus den Speichern des FPGAs stammt, nur bei einer maximalen MGT-Taktfrequenz von 400 MHz korrekt ist. Für den Betrieb des Viterbi-Entzerrers bei der vollen Datenrate sind 535 MHz notwendig. Bei mehr als

400 MHz sind auf einzelnen RocketIO-Kanälen falsche Bits zu finden, da die Schnittstelle zwischen Speicher und RocketIO-Kanäle in ihrer Datenrate begrenzt ist.

Ein weiteres Problem besteht in der Phasenverschiebung der RocketIO-Ausgangssignale aufgrund der unterschiedlichen Leitungslängen auf der FPGA-Platine, die sich nicht digital durch Bitschieben korrigieren lässt. Der Phasenversatz kann bis zu 50 ps betragen, was bei einer Bitdauer von 93,5 ps (10,7 Gbit/s) ungefähr der Hälfte eines Bits entspricht. Es liegen zwar einige Datensignale in der Mitte des Taktauges, bei anderen Signalen fällt der Signalwechsel mit der Taktflanke zusammen. Dadurch entsteht eine Verletzung der Rüstzeit der internen Flipflops auf dem Viterbi-Entzerrer-Chip. Somit ist eine korrekte Funktionsweise nicht mehr gegeben.

6.4 Einordnung der Messergebnisse dieser Arbeit

Tabelle 6.4 stellt die Ergebnisse dieser Arbeit in Relation zu anderen Veröffentlichungen.

Tabelle 6.4: Einordnung dieser Arbeit im internationalen Vergleich

Veröffent- lichung	Dawid, Fettweis, Meyr 1996 [12]	Black, Meng 1997 [13]	Core-Optics 2000 [14]	Intel 2008 [15]	Diese Arbeit: 2009 [73], [74]
Technologie	1,0 μm CMOS ES2 (Atmel)	1,2 μm CMOS HP	0,12 μm CMOS IBM	90 nm CMOS Intel	90 nm CMOS ST Micro.
Chipgröße	95 mm ² (4 ICs)	81 mm ²	keine Angabe	1,9 mm ²	5,7 mm ²
Anzahl Transistoren	260000	150000	1.8 Mio. Gatter	698000	451504
DSP Takt	50 MHz	80 MHz	keine Angabe	3,8 / 4,7 GHz	1,0 GHz
Durchsatz (nom., ggf. min., max.)	1,2 Gbit/s	0,144 Gbit/s / 1 Gbit/s	10 Gbit/s	1,9 Gbit/s / 2,35 Gbit/s	gemessen: 32 Gbit/s
Leistung (nom., ggf. min., max.)	3,6 W	0,024 W / 3,0 W	keine Angabe	0,358 W / 0,813 W	2,39 W
Vers.-Spg.	keine Angabe	1,5 V / 5 V	keine Angabe	1,3 V / 1,7 V	1,0 V
FOM 1: Energie pro Datenmenge	3 nJ/bit	1,67 nJ/bit / 3 nJ/bit	keine Angabe	0,188 nJ/bit / 0,346 nJ/bit	0,075 nJ/bit @ 32 Gbit/s
FOM 2: Chipfl.- pro Durchsatz	79,2 mm ² /Gbit/s	81 mm ² /Gbit/s	keine Angabe	1,0 mm ² /Gbit/s / 0,809 mm ² /Gbit/s	0,178 mm ² /Gbit/s
Architektur	Minimized Method (parallel)	Sliding Block (parallel)	keine Angabe	seriell	Schiebefenster- verfahren (parallel)

Diskussion der Messergebnisse

Im Rahmen dieser Arbeit wird ein digitaler Entzerrer auf Basis des MLSE-Verfahrens in einer 90 nm CMOS-Technologie entworfen und hergestellt. Im Vergleich mit anderen Veröffentlichungen (Tabelle 6.4) ist gezeigt, dass dieser Viterbi-Entzerrer mit 32 Gbit/s einen dreifach höheren Datendurchsatz aufweist als ein kommerziell erhältliches Produkt [14]. Stellt man den Energieverbrauch, der für die Verarbeitung von einem Bit notwendig ist, dem der Veröffentlichung in einer anderen 90 nm CMOS-Technologie [15] gegenüber, benötigt der Entwurf dieser Arbeit je nach Wahl der Versorgungsspannung 2,5 bis 4,6-mal weniger Energie. Ein niedriger Energieverbrauch ist hinsichtlich einer passiven Kühlung von Vorteil. Bezieht man die benötigte Chipfläche auf den Durchsatz, wird in dieser Arbeit 4,5 bis 5,6-mal weniger Chipfläche benötigt, um 1 Gbit/s zu verarbeiten. In Anbetracht der hohen Maskenkosten ist das Erzielen einer hohen Packungsdichte auf dem Chip ein weiterer Wettbewerbsvorteil.

Der Messaufbau lässt aufgrund der Limitierungen im FPGA keine schnelleren Messungen zu. Simulationen lassen eine höhere Datenrate vorhersagen. Der kritische Pfad liegt laut Layoutsimulationen bei 614 ps. Legt man eine Eingangsdatenrate von 43 Gbit/s zugrunde, entspricht das einer Taktfrequenz des DSPs von 1,34 GHz bzw. einer Taktperiode von 746 ps. Messungen an einem On-Wafer-Messplatz mit Zuführung der Taktsignale und der Versorgungsspannung über Messspitzen zeigen, dass der interne Taktteiler sogar bis zu einer Eingangstaktfrequenz von 6,1 GHz funktioniert, was einer Datenrate von 48,8 Gbit/s entspricht. Da die Versorgungsspannung über Messspitzen nur teilweise angeschlossen werden kann und somit der Chip nur schlecht versorgt wird, könnte der Taktteiler sogar bei noch höheren Frequenzen funktionieren.

Rechnet man nun die Messergebnisse auf die realistische Zieldatenrate von 43 Gbit/s um, wäre das mehr als viermal so schnell wie das kommerziell vertriebene Produkt von Cisco (früher: Core Optics) [14]. Die Leistungsaufnahme würde dann 2,543 W betragen. Dies entspräche einem Energieaufwand für ein zu verarbeitendes Bit von nur noch 59 pJ. Dies wäre eine 3,2 bis 5,9-fache Verbesserung gegenüber einer Implementierung in einer ähnlichen Technologie [15]. Ähnliches gilt für die benötigte Chipfläche pro auf 1 Gbit/s normierten Datendurchsatz. Dieser würde nur noch bei 0,133 mm²/Gbit/s liegen. Hier würde die Verbesserung bei Faktor 6,1 bis 7,5 liegen.

Selbst bei einer Datenrate von 32 Gbit/s ist dieser Viterbi-Entzerrer nach bestem Wissen der weltweit erste in CMOS hergestellte Chip, der erfolgreich vermessen ist und hinsichtlich Chipflächenbedarf und Leistungsaufnahme bisheriger Veröffentlichungen deutlich überlegen ist.

7 Automatische Anpassung an den Kanal

Wie bereits in den einleitenden Kapiteln erwähnt ist, handelt es sich bei dem optischen Kanal hinsichtlich der Polarisationsmodendispersion (PMD) um einen zeitvarianten Kanal. Daher muss der Viterbi-Entzerrer Schnittstellen besitzen, um während des Betriebs neue Entzerrerkoeffizienten (Zweigmetriken) einzuschreiben und gleichzeitig Daten für die Kanalschätzung bereitzustellen. Hieraus können dann neue, an den Kanal angepasste Metriken berechnet werden.

7.1 Randbedingungen für eine Kanalschätzung

Es muss ein Verfahren gefunden werden, mit dem es möglich ist, anhand des eingehenden und dekodierten Datenstroms Daten zu gewinnen, die eine externe Berechnung der Metriken zulassen. Darüber hinaus müssen Randbedingungen wie das Zeitintervall, in dem sich die Kanaleigenschaften ändern und die maximal mögliche Taktfrequenz, mit der eine externe Berechnung der Metriken durchgeführt werden soll, betrachtet werden.

Die Berechnung der Metriken soll auf einem FPGA realisiert werden. Kommerziell erhältliche FPGAs können, wie durch eigene Simulationen ermittelt, mit einer Taktfrequenz von 100 MHz einigermaßen komplexe Rechenoperationen durchführen.

Es bietet sich an, eine Datenrate für die Schnittstelle zu definieren, die sich aus der internen Taktfrequenz des Viterbi-Entzerrers ableiten lässt. Die interne Taktfrequenz beträgt bei einer Eingangsdatenrate von 43 Gbit/s 1,34 GHz. Am einfachsten ist es, wenn man diesen Takt so oft mit Hilfe von Takteilerflipflops herunterteilt, dass man in den Bereich von 100 MHz kommt. Aufgrund der Takteilerflipflops kommen als Takteilerverhältnis nur Zweierpotenzen in Frage. Ein Takteiler von 16 würde zu einer Taktfrequenz von 83,6 MHz führen, was 1/512 der Eingangsdatenrate entspricht. Daraus ergibt sich, dass über diese Schnittstelle nur jedes 512. Eingangssymbol und die dazugehörigen dekodierten Daten ausgegeben werden.

Für eine aussagekräftige Kanalschätzung werden rund 100000 bis 200000 Symbole [26] benötigt. Die dazwischenliegende Zweierpotenz beträgt $2^{17} = 131072$. Für die Datensammlung würde bei 83,6 MHz eine Zeit von rund 1,6 ms benötigt. Für das Einschreiben der neuen Zweigmetriken werden 64 Takte benötigt, was einer Zeit von rund 76,6 μ s entspricht. Das heißt, die Zeit für die Adaption wird im Wesentlichen durch das Sammeln der Daten zur Metrikberechnung benötigt. Da die Kanaländerung in etwa im Sekundentakt erfolgt, also die Zeit, die für die Adaption zur Verfügung steht, ungefähr drei Größenordnungen höher ist, kann ein Systementwurf für die Adaptionsschaltung auf obiger Berechnungsbasis erfolgen.

7.2 Grundlagen der Kanalschätzung

Ein bekanntes Verfahren zur Schätzung verrauschter Kanäle ist die Gaussische Approximation. Bei optischer Datenübertragung eignet sich die Chi-Quadrat-Methode besonders, da hierbei die nichtlinearen Effekte von Laser- und Photodioden sowie das bei optischen Verstärkern charakteristische ASE-Rauschen mitberücksichtigt werden. Die Implementierung dieser Methoden in Hardware ist aufwändig. Daher wird eine generische Methode angestrebt, die nach [21] eine vergleichbare Performance aufweist als die Chi-Quadrat-Methode.

Für die Metrikberechnung stehen die verzerrten Eingangssymbole als auch die entzerrten bzw. dekodierten Bits zur Verfügung. Das zugrundeliegende Kanalmodell in Abbildung 2.5 zeigt einen Kanal mit zwei Speichern. Das hat zur Folge, dass auf Grund von Symbolnebensprechen ein gesendetes Bit sowohl von seinem Vorgänger als auch von seinem Nachfolger abhängt. Diese drei Bits können unterschiedlich gewichtet sein. Daraus folgt, dass es unter Vernachlässigung von Rauschen maximal $2^3 = 8$ Amplitudenstufen für ein Symbol am Eingang des Viterbi-Entzerrers gibt.

Nun kann man eine Tabelle erstellen, die die Häufigkeiten für jede mögliche Bitfolge einer bestimmten Amplitudenstufe enthält. Die Amplitudenstufe entspricht dem quantisierten Abtastwert des vorgeschalteten AD-Wandlers.

Um aus einem unbekanntem Kanal eine solche Amplitudenstatistik erstellen zu können, muss der Viterbi-Entzerrer mit gewissen Startmetriken vorkonfiguriert werden. Wie die Startmetriken zu wählen sind, wird im Folgenden an Hand eines Augendiagramms ohne Berücksichtigung von Rauschen für die acht möglichen Bitfolgen erläutert. Die Gewichtung des ersten und dritten Bits der Bitfolge wird gleich gewählt. Das mittlere Bit wird am stärksten gewichtet. Daraus ergibt sich ein symmetrisches Augendiagramm. Durch eine Verschiebung des Abtastzeitpunktes aus dem Augenmittelpunkt heraus kann jedoch eine Asymmetrie erzeugt werden.

Betrachtet man die gesendeten Bitfolgen zu dem in Abbildung 7.1 eingezeichneten Abtastzeitpunkt, erkennt man, dass wenn der 3 bit AD-Wandler die Amplitudenstufe 0 ausgibt, die gesendete Bitfolge entweder 1_0_0 oder 0_0_0 entsprochen haben muss. Gibt der AD-Wandler die Amplitudenstufe 1 aus, kann man rückwärts daraus schließen, dass ursprünglich die Bitfolge 0_0_1 übertragen wurde. So kann man jedem Abtastwert des AD-Wandlers eine Spur im Augendiagramm zuordnen, aus der die gesendete Bitfolge hervorgeht. Tabelle 7.1 fasst diesen Sachverhalt noch einmal zusammen.

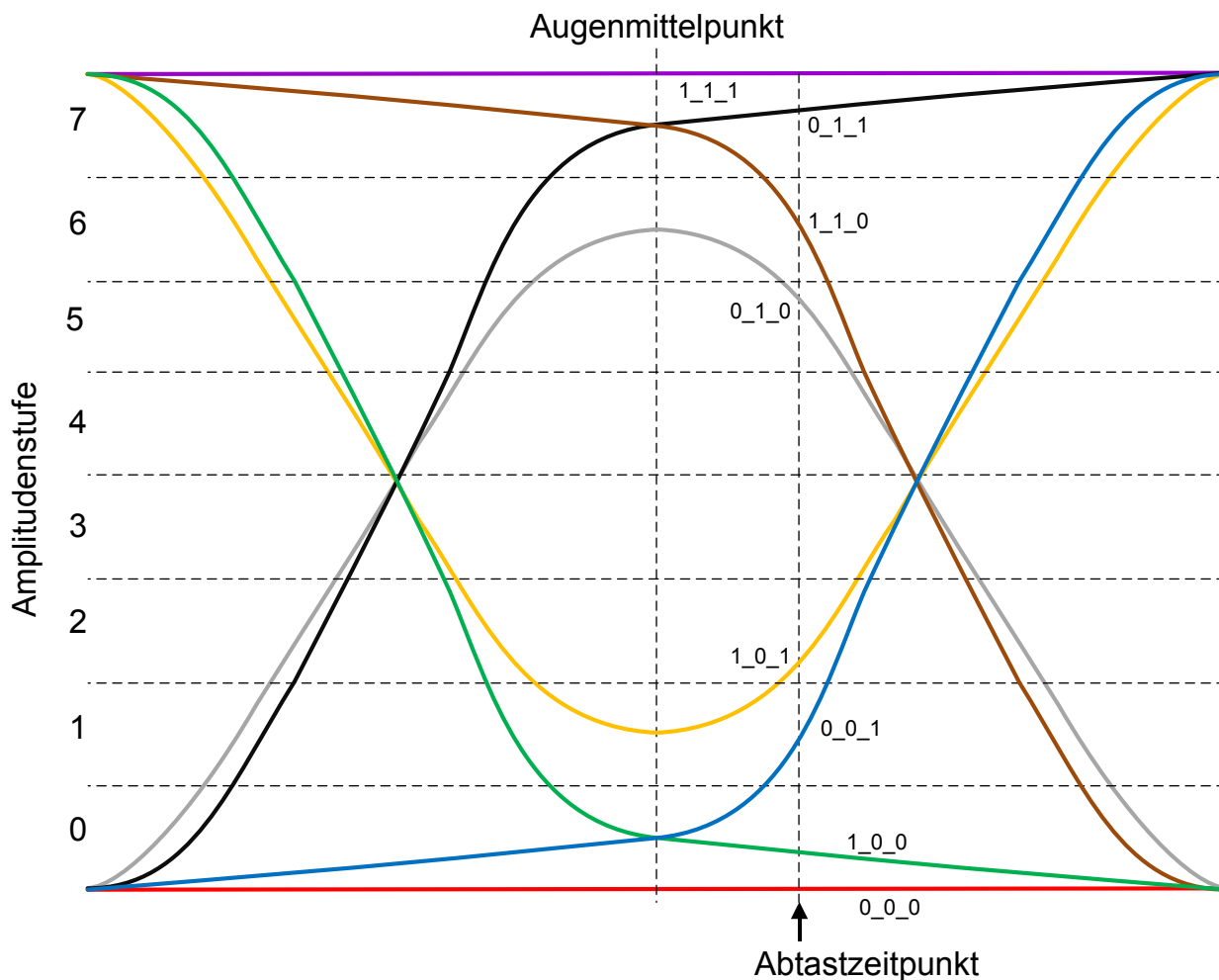


Abbildung 7.1: Beispiel für ein Augendiagramm ohne Rauschen für eine 3 bit lange Folge

Tabelle 7.1: Amplitudenstufe in Abhängigkeit der Bitfolge

	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	X				X			
1		X						
2						X		
3								
4								
5			X					
6							X	
7				X				X

Es ist zu erkennen, dass nicht alle Amplitudenstufen einer Bitfolge zugeordnet sind, dafür sind andere doppelt besetzt. Die Belegung der Tabellenelemente ist einerseits vom Abtastzeitpunkt, wie man in Abbildung 7.1 leicht erkennen kann, abhängig, andererseits hängt sie auch von der Gewichtung der Einzelbits ab.

Erstellt man nun ein Histogramm, indem man eine Zuordnung zwischen abgetastetem Eingangswert (Amplitudenstufe) und der dekodierten Bitfolge vornimmt, unter der Annahme, dass die dekodierten Bits richtig sind, erhält man in einem realen Kanal eine Häufigkeitsverteilung der Amplitudenstufen für jede Bitfolge. Unter Berücksichtigung von Rauschen, würden sich nämlich die Spuren im Augendiagramm verwischen. Auf Grund der Varianz würden für jede Bitfolge mehrere Amplitudenstufen detektiert werden. Legt man additives weißes gaußsches Rauschen zu Grunde, ergibt sich für jede Bitfolge eine Verteilung nach (2.9).

Abbildung 7.2 zeigt eine normierte Häufigkeitsverteilung der einzelnen dekodierten Bitfolgen in Abhängigkeit des abgetasteten Eingangswertes des Viterbi-Entzerrers.

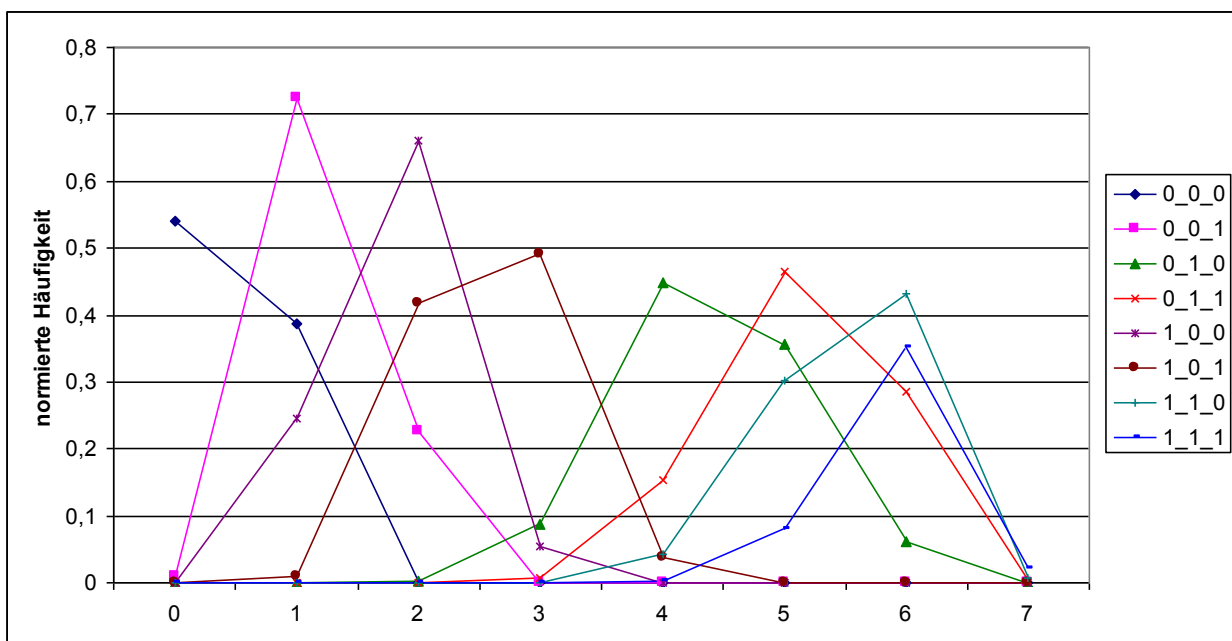


Abbildung 7.2: Beispiel einer Häufigkeitsverteilung für jede Bitfolge in Abhängigkeit der Amplitudenstufen

Diese Häufigkeitsverteilung, der ein realer Kanal zugrunde liegt, wurde im Vorfeld durch den Projektpartner zur Verfügung gestellt. Man erkennt anhand der Spuren, dass die Abtastung außerhalb der Augenmitte erfolgt sein muss, da sich der Abtastwert der unsymmetrischen Bitfolgen („0_0_1“, „0_1_1“, „1_0_0“ und „1_1_0“) eher einer Flanke in Abbildung 7.1 zuordnen lässt als einem Wert im Augenmittelpunkt. Eine weitere Möglichkeit wäre, dass der Abtastzeitpunkt zwar in der Mitte liegt, aber die Gewichtung der Bits nicht symmetrisch ist. Eine genaue Aussage kann aufgrund des beaufschlagten Rauschens nicht getroffen werden.

7.3 Bestimmung der Metriken mit Hilfe der Histogrammmethode

Abbildung 7.3 erläutert schematisch die Vorgehensweise bei der Histogrammmethode.

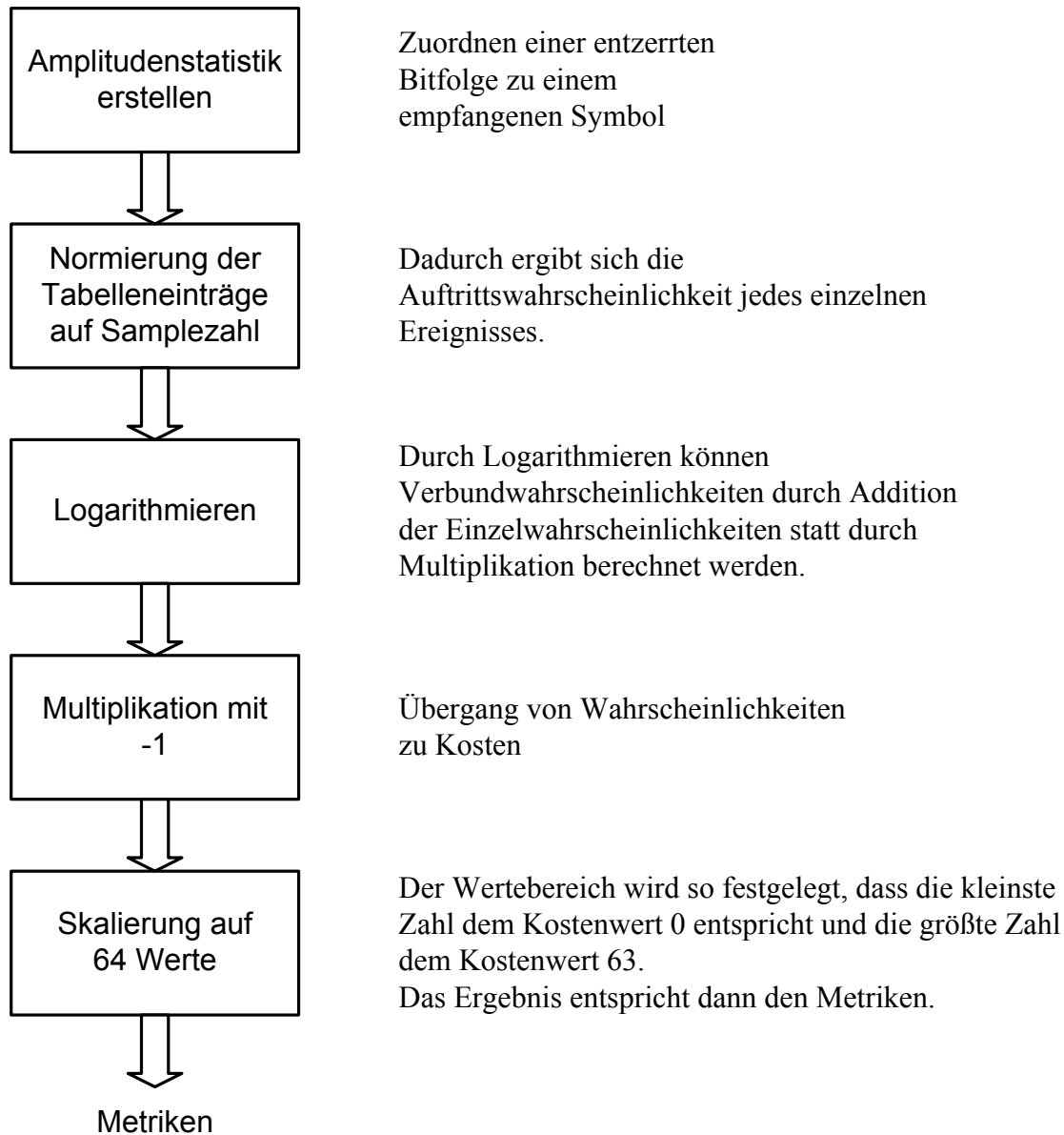


Abbildung 7.3: Algorithmus zur Bestimmung der Metriken mit der Histogrammmethode

Der Algorithmus aus Abbildung 7.3 wird in [75] mathematisch hergeleitet. Das Ergebnis ist in Gleichung (7.1) zusammengefasst.

$$M = \text{Round} \left[63 \frac{\log \frac{W}{W_g}}{\log \frac{W_k}{W_g}} \right] \quad (7.1)$$

Sortiert nach Zeilen und Spalten ergibt sich

$$M_{j,k} = \text{Round} \left[63 \frac{\log \frac{W_{j,k}}{W_g}}{\log \frac{W_k}{W_g}} \right] \quad \text{mit } j, k \in [0..7] \quad (7.2)$$

Man erkennt, dass zur Berechnung der Metriken $M_{j,k}$ nur der größte Wert W_g , der kleinste Wert W_k und die eigentlichen Tabelleneinträge $W_{j,k}$ benötigt werden.

Die Implementierung der Formel (7.1) in Hardware wäre eine nicht triviale Angelegenheit, da in ihr drei Divisionen, zwei Logarithmusfunktionen und eine Multiplikation vorkommen.

Erstellt man die Amplitudenstatistik mit realen Daten (vgl. Tabelle 7.2), erkennt man, dass die übertragenen und verrauschten Daten zufälligen Charakters sind. Daraus folgt, dass die Summe jeder Spalte der Amplitudenstatistik ungefähr einem Achtel der Gesamtanzahl betrachteter Samples entspricht. Weiterhin zeigt sich, dass es immer Tabelleneinträge gibt, in denen keine Ereignisse gezählt werden, das heißt, als kleinsten Wert in der Amplitudenstatistik kann die 0 angenommen werden. Außerdem gibt es über alle Spalten hinweg in einer Zeile immer einen Wert, der fast alle Ereignisse (ein Achtel der Gesamtzahl-sampleanzahl) enthält. Somit sind der größte Wert W_g und kleinste Wert W_k festgelegt. Der Logarithmus im Nenner von Formel (7.1) kann also als eine Konstante dargestellt werden. Da der Logarithmus von 0 nicht definiert ist, wählt man $W_k = 1$.

Tabelle 7.2: Beispielsimulation einer Amplitudenstatistik über 131072 Samples

	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	16161	0	0	3	15560	1	1	0
1	228	0	1	0	1832	1	14	0
2	0	1300	508	0	0	1109	3795	0
3	3	10546	8288	1	0	9944	10983	2
4	0	5260	7041	2	0	4171	1570	1
5	0	283	533	1	0	144	23	0
6	0	3	4	2	0	1	1	86
7	1	0	2	16378	0	1	0	15284

Trägt man Formel (7.1) in Form eines Schaubildes (siehe Abbildung 7.4) auf, kann man direkt die Metrik zu einer aufgetretenen Anzahl an Ereignissen ablesen und zuordnen. Da es nur eine endliche Anzahl von 64 Metriken gibt, lassen sich diskrete Bereiche auf der Kurve definieren, die einer Metrik entsprechen. Diese Bereiche und deren zugeordnete Metrik lassen sich in Tabellenform in einem ROM-Speicher ablegen. Daraus folgt, dass die eigentliche Berechnung

der Formel (7.1) offline durchgeführt werden kann und demzufolge nicht in Hardware umgesetzt werden muss.

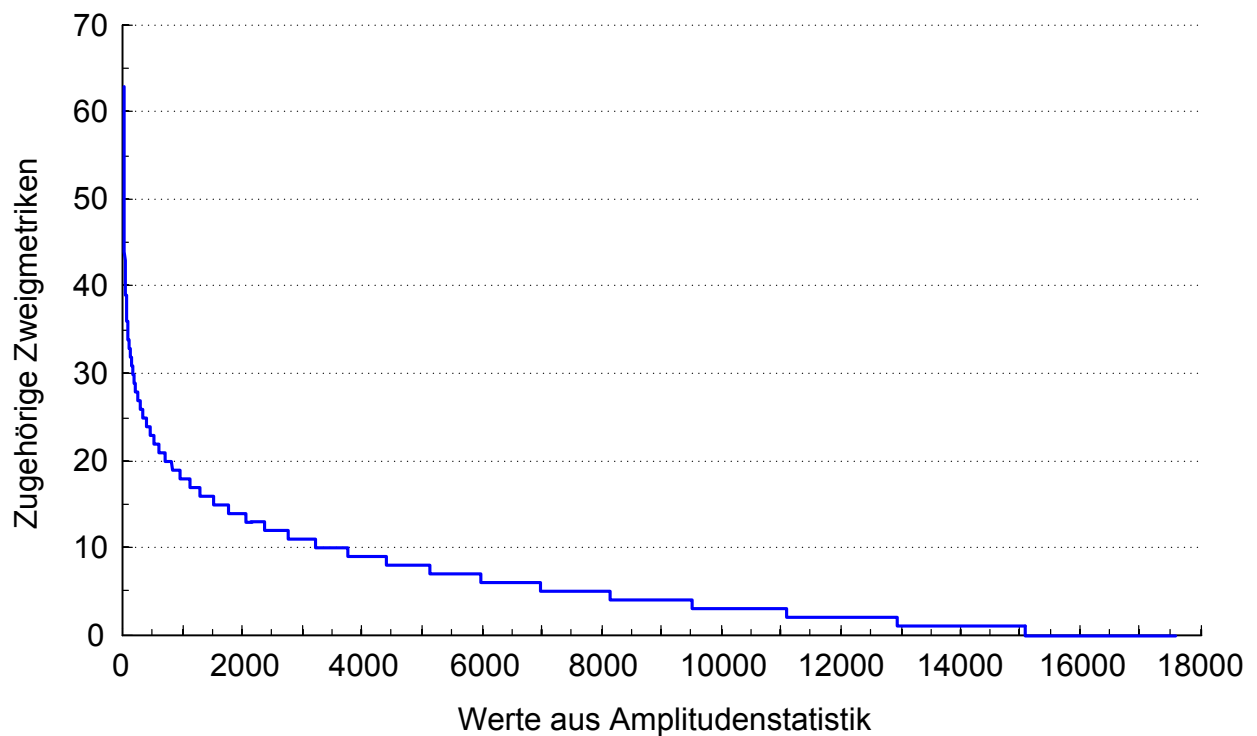


Abbildung 7.4: Abbildung der Werte aus der Amplitudenstatistik auf die Metriken

Die Schritte aus Abbildung 7.3 werden ständig mit aktuellen Daten wiederholt, um die zeitvariante Dispersion ausgleichen zu können. Da sowohl die unprozessierten Daten wie auch die dekodierten Daten in die neuen Metriken eingehen, auf deren Basis dann wieder neue Daten dekodiert werden, entsteht ein rückgekoppeltes System, welches zu schwingen beginnt. Um das Aufschwingen abzufedern, wird die zeitlich vorhergehende Amplitudenstatistik nicht komplett verworfen, sondern mit halber Gewichtung zur aktuellen Amplitudenstatistik hinzuaddiert.

7.4 Implementierung und Simulation

Wie in Kapitel 7.3 dargelegt ist, kann die Berechnung der Metriken stark vereinfacht werden. Die Vorgehensweise besteht im Wesentlichen aus folgenden Schritten:

1. Einmaliges Abbilden der Formel (7.1) in eine Tabelle, die eine Zuordnung der aufgetretenen Ereignisse zu den Metriken erhält. Dieser Schritt erfolgt offline außerhalb der Adaptionsschaltung.
2. Einschreiben der in Schritt 1 erstellten Tabelle in die Adaptionsschaltung.

3. Erstellung der Amplitudenstatistik durch Zählen der Ereignisse über eine bestimmte Sampleanzahl.
4. Zuordnung der Amplitudenstatistiktable in eine Metriktabelle mit Hilfe von Abbildung 7.4.
5. Einschreiben der Metriken in den Viterbi-Entzerrer.
6. Wiederholung der Schritte 3 bis 5 bei Halbierung der Tabelleneinträge der bereits erstellten Amplitudenstatistik.

Die Schritte 2 bis 6 können mittels der Hardwarebeschreibungssprache VHDL auf einem FPGA implementiert werden. Eine Möglichkeit der Implementierung ist in [75] gegeben. Für die praktische Umsetzung sind allerdings noch Verbesserungen in Bezug auf Ressourcenverbrauch auf dem FPGA notwendig. Die Simulationsergebnisse des VHDL-Entwurfs nach [75] sind in Abbildung 7.5 dargestellt.

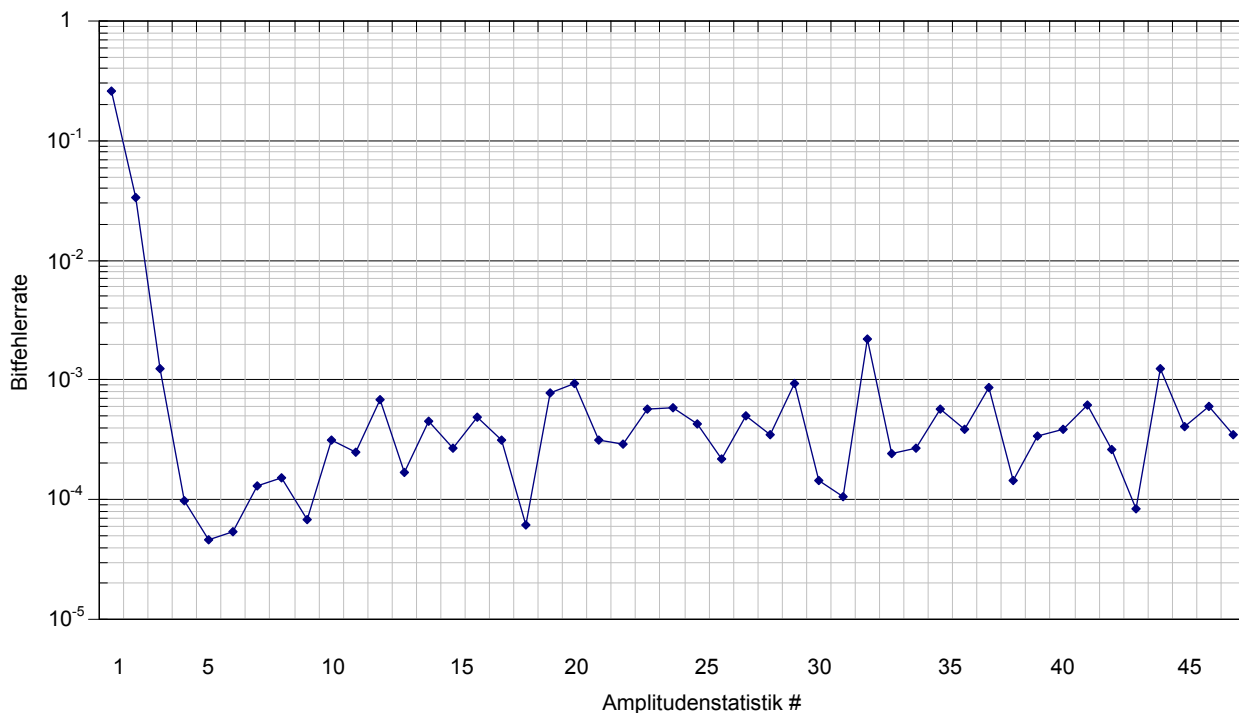


Abbildung 7.5: Bitfehlerrate über der Anzahl der Adaptionvorgänge

Man erkennt, dass die Bitfehlerrate nach drei Adaptionvorgängen schon den angestrebten Wert von 10^{-3} erreicht. Im weiteren Verlauf zeigt sich, dass bis auf wenige Ausnahmen die Bitfehlerraten immer zwischen 10^{-3} und 10^{-4} liegen. Der schlechteste Wert liegt bei $2,2 \cdot 10^{-3}$. Dieser Wert kann toleriert werden, da die FEC-Schwelle für nachgeschaltete Fehlerkorrektursysteme bei $2,5 \cdot 10^{-3}$ liegt, um Fehlerraten von 10^{-12} zu erreichen.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Der Bandbreitenbedarf heutiger Kommunikationssysteme steigt immer weiter an, da Anwendungen wie z.B. Fernsehen über das Internet (IPTV) und Filmabrufdienste (VoD) vom Endverbraucher immer stärker nachgefragt werden. Dabei ist es unbedeutend, ob diese Breitbanddienste über drahtgebundene Leitungen wie z.B. xDSL-Anschlüsse oder schnelle Mobilfunknetze (UMTS, LTE) zum Kunden gelangen. Beide Verbreitungswege erfordern ein den Anforderungen gerecht werdendes und gut ausgebautes Kernnetz (engl. Backbone) der Telekommunikationsanbieter. Insbesondere die immer beliebtere hochauflösende Bildqualität und dreidimensionale Ausstrahlung von Fernsehprogrammen benötigt eine noch höhere Datenrate als die herkömmliche Bildauflösung.

Um diesen Anforderungen gerecht zu werden, müssen Konzepte entwickelt werden, die es ermöglichen, unter Weiterverwendung der vorhandenen Kabelnetzinfrastruktur die Datenrate signifikant zu erhöhen. Ein Austausch der bereits verlegten Leitungen hätte immense Kosten zur Folge, die den Erfolg der neuen Breitbanddienste schmälern würden.

Die Telekommunikationsanbieter denken daher an eine Vervierfachung der aktuellen Datenraten im Kernnetz von 10 Gbit/s auf 40 Gbit/s. Vorhandene Glasfaserleitungen zeigen bei diesen Datenraten Dispersionseffekte, die insbesondere im Falle der Polarisationsmodendispersion zeitvarianten Verhalten aufweist. Die zeitinvariante chromatische Dispersion ließe sich zwar mit dispersionskompensierenden Fasern korrigieren, ist aber trotzdem aufwändig, da die Kompensationsfasern an die entsprechenden verlegten Fasern angepasst werden müssen.

Eine alternative Möglichkeit, um den Dispersionseffekten entgegen zu wirken, ist ein elektronischer Entzerrer, der sich nach dem Plug-and-play-Prinzip adaptiv an den gegebenen Kanal anpasst.

Bevor mit dem Entwurf einer Entzerrerschaltung begonnen werden kann, ist zu klären, was für eine Halbleitertechnologie und welche Logikfamilien am besten dafür geeignet sind, die wirtschaftlichen Ziele wie die Herstellungskosten und die laufenden Kosten so niedrig wie möglich zu halten. Dabei stellt sich heraus, dass CMOS-Technologien gegenüber Bipolar-technologien bei Massenprodukten kostengünstiger in der Herstellung sind. Im laufenden Betrieb fallen vor allem Energiekosten an. Auch da sind CMOS-Technologien den Bipolar-technologien überlegen. Besonders interessant ist die statische CMOS-Logik, da diese sich im Gegensatz zu Pseudo-NMOS und CML durch einen besonders niedrigen Energieverbrauch auszeichnet.

Als Nächstes erfolgt eine Abwägung, welches Entzerrerkonzept sich am besten eignet, die auftretenden Dispersionseffekte zu kompensieren. Es stellt sich heraus, dass das trellisbasierte

MLSE-Verfahren mit dem Spezialfall des Viterbi-Algorithmus vorteilhaft ist. Die Implementierung erfolgt nach einem Blockdekodierverfahren, dem Schiebefensterverfahren. Bei diesem Verfahren werden sowohl Pipelining und Parallelisierung eingesetzt. Serielle Verfahren sind ungeeignet, da die aufwändige digitale Signalverarbeitung nicht bei der dafür notwendigen Taktfrequenz von 43 GHz arbeiten kann. Auf Basis von Simulationen wird ermittelt, dass die zu erwartende kombinatorische Logik bei etwas über einem 1 GHz getaktet werden kann. Daher ist eine 32-fache Parallelisierung notwendig. Anstelle eines großen systolischen Blockdekoders werden vier parallel arbeitende SBDs mit entsprechend kürzerer Trellislänge vorgesehen, wodurch sich die Datendurchlaufzeit verkürzt.

Es erfolgen Überlegungen auf Strukturebene, welche Recheneinheiten für die Umsetzung des Viterbi-Algorithmus benötigt werden. Das charakteristische Trellisdiagramm lässt sich mit Hilfe sogenannter Addier-Vergleichs-Auswahleinheiten (engl. Add-Compare-Select-Units, ACSUs) darstellen, die wiederum aus Addierern, Subtrahierern und Multiplexern aufgebaut sind. Die Bestimmung der Zweigmetriken erfolgt über eine Zweigmetriekrecheneinheit (engl. Branch-Metric-Unit), die vorzugsweise aus einer im Betrieb aktualisierbaren Wertetabelle realisiert wird. Die Rückverfolgung des kürzesten Pfades geschieht mittels Pfadrückverfolgungseinheiten (engl. Trace-Back-Units), die aus Multiplexern und Flipflops bestehen. Die Verknüpfung der vier systolischen Blockdecker erfolgt mit Hilfe von Minimumauswahlschaltungen (engl. Select-Minimum-Units), wobei es sich im Wesentlichen um erweiterte ACSUs handelt. Aufgrund der 32-fachen Parallelisierung der Ein- und Ausgangsdaten würden die Schnittstellen sehr viele Pads beanspruchen. Daher werden die Eingangsdaten vierfach parallel statt 32-fach parallel eingeschrieben. Für die Ausgangsdaten gilt ähnliches; diese werden statt 32-fach nur vierfach parallel ausgegeben.

Der Viterbi-Entzerrer wird anhand eines Bottom-up-Entwurfs aufgebaut. Es werden zunächst CMOS-Grundgatter entworfen und ein auf Verzögerungszeit optimiertes Layout erstellt. Es zeigt sich, dass mit einem optimalen Layout der Geschwindigkeitsfaktor 2 gegenüber den Standardzellen, die vom Halbleiterhersteller zur Verfügung gestellt werden, erreicht werden kann. Aufgrund dieses Geschwindigkeitsvorteils lässt sich die Chipfläche reduzieren, was sich in niedrigeren Maskenkosten niederschlägt. Der Zeitaufwand für einen Full-Custom-Entwurf ist damit begründet.

Die schnellen Ein- und Ausgangsdaten lassen sich in der vorliegenden Halbleitertechnologie nicht mehr mit statischer CMOS-Logik verarbeiten. Deshalb werden CML-Multiplexer und Demultiplexer sowie Pegelkonversionsschaltungen zwischen CML und CMOS entworfen.

Nach dem Grundgatterentwurf erfolgt der Entwurf komplexerer Logikbausteine wie z.B. Volladdierer, Speicherzellen und Flipflops aus den Grundgattern. Diese Komponenten werden auf Transistorebene simuliert, um die Verzögerungszeit zu ermitteln. Ausgehend davon erfolgen

Untersuchungen, welche Addiererarchitekturen sich für Additionen mit Wortbreiten von 8 bit in der zur Verfügung stehenden CMOS-Technologie besonders eignen. Es stellt sich heraus, dass ein Paralleladdierer mit sukzessiver Übertragskorrektur in diesem speziellen Anwendungsfall besonders aussichtsreich ist, was Verzögerungszeit, Flächenaufwand und Stromverbrauch betrifft. Für die Umsetzung der Wertetabelle als Teil der Zweigmetrikrecheneinheit werden statische SRAM-Speicherzellen gegenüber dynamischen Speicherkonzepten bevorzugt. Die Pfadrückverfolgungseinheiten, bestehend aus Multiplexern und Flipflops, sind vom zeitlichen Aspekt her unkritisch.

Der Viterbi-Entzerrer soll zeitvariante Dispersionseffekte korrigieren können. Daher wird eine Statistikschnittstelle vorgesehen, an der jedes 512. verzerrte 3 bit-Eingangssymbol und die drei nebeneinanderliegenden dazugehörigen entzerrten Bits ausgegeben werden. Mittels einer Histogrammmethode lassen sich über eine Kostenfunktion neue Zweigmetriken bestimmen. Die Berechnung der Zweigmetriken wird nicht auf dem Chip vorgenommen, sondern kann extern auf einem FPGA oder Mikrocontroller durchgeführt werden. Die neuen Zweigmetriken können dann im laufenden Betrieb aktualisiert werden. Simulationen zeigen, dass sich mit dem entworfenen Algorithmus Fehlerraten von 10^{-3} erreichen lassen. Die Änderung der Kanaleigenschaften geschieht im ungünstigsten Fall im Sekundenbereich. Daher reicht es aus, für eine Kanalschätzung nur jedes 512. Symbol heranzuziehen. Eine Betrachtung von Symbolen in kürzerem zeitlichem Abstand würde erhöhte Anforderungen an die externe Schaltung stellen, die die Berechnung der neuen Metriken vornimmt.

In der vorgestellten Arbeit wird ein digitaler Viterbi-Entzerrer für 40 Gbit/s-Systeme in einer 90 nm CMOS-Technologie entworfen und erfolgreich für Datenraten bis 32 Gbit/s vermessen. Dies ist ungefähr dreimal so schnell als kommerziell verfügbare Produkte [14]. Eine Messung höherer Datenraten ist mit der eingesetzten Messumgebung noch nicht möglich, da der FPGA ab 8 Gbit/s pro RIO-Kanal fehlerhafte Daten ausgibt. Weiterhin spielt der Phasenversatz der Eingangssymbole aufgrund unterschiedlicher Leitungslängen auf der FPGA-Platine eine bedeutende Rolle, dessen Ausgleich sich bei diesen hohen Frequenzen zunehmend schwierig gestaltet.

Die Leistungsaufnahme beträgt bei einer Datenrate von 32 Gbit/s 2,39 W bei 1 V Versorgungsspannung. Bezieht man die aufgewendete Energie auf die verarbeitete Datenmenge von einem Bit, ergibt sich ein Wert von 75 pJ/bit. Dies ist eine Verbesserung um den Faktor 2,5 bis 4,6 bisheriger Veröffentlichungen. Setzt man die Chipfläche in Relation zum Durchsatz, erhält man einen Wert von 0,178 mm²/Gbit/s, was einer Verbesserung um etwa Faktor 5 entspricht.

Hochrechnungen auf die Zieldatenrate von 43 Gbit/s versprechen eine Leistungsaufnahme von 2,543 W bei 1 V, eine normierte aufgewendete Energie von 59 pJ/bit bzw. 0,133 mm²/Gbit/s.

Simulationen der kombinatorischen Logik und Messungen des internen Taktteilers lassen erwarten, dass der Viterbi-Entzerrer-Chip sogar bei einer Datenrate von 48,8 Gbit/s betrieben werden kann.

8.2 Ausblick

8.2.1 Verbesserung der Messumgebung

In der vorgestellten Arbeit wird gezeigt, dass der Viterbi-Entzerrer-Chip bei einer Datenrate von bis zu 32 Gbit/s korrekt funktioniert. Dies entspricht ungefähr $3/4$ der Zieldatenrate.

Um noch höhere Datenraten mit dem FPGA als Datenquelle zu erreichen, müsste der MGT-Takt von aktuell 400 MHz auf 535 MHz (bei 43 Gbit/s) erhöht werden. Es zeigt sich, dass bei höheren Taktfrequenzen als 400 MHz (entspricht 8 Gbit/s pro RocketIO-Kanal) auf einzelnen RocketIO-Ausgängen fehlerhafte Bits ausgegeben werden. Da die interne Logik des FPGAs die maximale Taktfrequenz begrenzt, könnte eine Erweiterung des internen Datenbusses zwischen Speicher und RocketIO-Schnittstelle von 32 bit auf 64 bit Abhilfe schaffen. Mit Hilfe dieser Maßnahme lässt sich die interne Taktfrequenz der Schnittstelle zwischen Speicher und RocketIO-Kanal auf dem FPGA halbieren. Dadurch könnte erreicht werden, dass die Daten an allen RocketIO-Kanälen logisch korrekt anliegen. Ein weiterer Grund für falsche Signale könnte ein Spannungsabfall auf dem FPGA-Chip selbst sein. Dies muss aber erst noch überprüft werden. Eine Behebung dieses Problems ist möglich, indem man eine eigene Spannungsregelungsschaltung entwirft und an der FPGA-Platine anschließt.

Das zweite Problem ist die Phasenverschiebung der RocketIO-Ausgangssignale aufgrund der unterschiedlichen Leitungslänge auf der FPGA-Platine. Über Verzögerungsglieder, die in die Leitungen zwischen FPGA-Platine und Viterbi-Entzerrer geschaltet werden, kann die Phasenverschiebung korrekt eingestellt werden. Eine Alternative wäre ein Entwurf einer Adapterplatine, die die unterschiedlichen Leitungslängen auf der FPGA-Platine ausgleicht.

8.2.2 Weiterentwicklung des Viterbi-Entzerrers

In dieser Arbeit wird ein digitaler Viterbi-Entzerrer zur Kompensation von chromatischer und Polarisationsmodendispersion entwickelt. Für einen kompletten Empfänger werden zusätzlich eine Photodiode, ein AD-Wandler und eine Taktrückgewinnung benötigt. Der Demonstrator soll zunächst mit einem Bipolar-AD-Wandler, der von einem externen Projektpartner entworfen wurde, betrieben werden. Der Bipolar-AD-Wandler besitzt an seinem Ausgang genau die gleiche Schnittstellenkonfiguration wie der Viterbi-Entzerrer an seinem Eingang. Daher können beide Chips direkt aneinandergesetzt werden und die elektrische Verbindung über kurze Bonddrähte

hergestellt werden. Sollten die Versuche ein positives Ergebnis hervorbringen, wird im Hinblick auf eine spätere Produktentwicklung eine Gesamtintegration eines CMOS-AD-Wandlers und des Viterbi-Entzerrers als Ein-Chip-Lösung angestrebt.

Mithilfe dieses Aufbaus lassen sich dann Bitfehlermessungen mit langen zufälligen Bitfolgen unter realen Bedingungen durchführen. Dadurch eröffnet sich die Möglichkeit, auch die Schnittstelle für die automatische Anpassung der Metriken zu testen.

8.2.3 Erhöhung der Rechengeschwindigkeit

Das Bauelement, das die Geschwindigkeit begrenzt, ist die Addier-Vergleichs-Auswahleinheit (ACSU). Die ACSU besteht im Wesentlichen aus einem Paralleladdierwerk mit sukzessiver Übertragskorrektur (engl. Ripple-Carry-Adder, RCA), welches wiederum aus Volladdierern aufgebaut ist. Somit liegt es nahe, die Volladdiererlaufzeit durch Verbesserung der Schaltungsarchitektur zu verkürzen. Wie aus Abbildung 4.23 zu erkennen ist, liegen zwei NAND-Gatter auf dem Übertragspfad. Bei einem 8 bit-RCA sind dies nach Formel (4.10) 14 NAND-Gatterlaufzeiten. In [76] wird eine Volladdiererarchitektur vorgeschlagen, bei der nur ein NAND-Gatter im längsten kombinatorischen Pfad liegt. Der Vorteil der geringeren Rechendauer muss jedoch mit einem höheren Transistoraufwand erkauft werden. Der Volladdierer nach [76] enthält sechs Gatter, welche man mit 38 Transistoren aufbauen kann. Der Entwurf aus Abbildung 4.23 kommt dagegen mit 30 Transistoren aus.

8.2.4 Konzepte für noch höhere Datenraten

Aktuell wird die Einführung von Datenraten von 100 Gbit/s und höher pro optischen Wellenlängenkanal diskutiert. Eine Übertragung unter Anwendung der NRZ-Kodierung mit anschließender Entzerrung auf Empfangsseite ist auf Grund der chromatischen Dispersion nicht mehr möglich. Die ISI würde aufgrund der kurzen Bitdauer von 10 ps zu groß werden. Eine Maßnahme, um die ISI zu verringern, ist die Aufteilung eines einzelnen breitbandigen Übertragungskanals auf viele Unterkanäle mit jeweils geringer Bitrate. Hierbei entsteht jedoch Nachbarkanalnebensprechen (ICI). Um den Einfluss des ICI zu verringern, müsste man den Trägerfrequenzabstand der Unterkanäle erhöhen, woraus eine ineffiziente Bandbreitennutzung resultiert. Eine Möglichkeit, um ICI zu vermeiden, ist die Benutzung von zueinander orthogonalen Trägern. Die Erzeugung eines Signals mit orthogonalem Frequenzmultiplex (OFDM), erfolgt mit Hilfe der inversen diskreten Fourier-Transformation (IDFT). Eine deutliche Vereinfachung des Aufwandes der IDFT bringt der IFFT-Algorithmus von Cooley und Tukey [77] mit. Der Kern der schnellen inversen Fourier-Transformation sind schnelle Multiplikationen und Additionen, die Teil der Schmetterlingsstruktur der IFFT sind. Als Multipliziererarchitekturen eignen sich insbesondere Wallace-Tree- [78] und Booth-Multiplizierer [79],

während für die Addition Carry-Lookahead-Architekturen [80] in Frage kommen. Erste Abschätzungen zeigen [54], dass sich die Multiplikationen bei Umsetzung als Full-Custom-Entwurf in einer 65 nm CMOS-Technologie mit einer Taktfrequenz von 1 GHz realisieren lassen. Mit Hilfe der Signed-Digit-Zahlendarstellung [81] besteht Aussicht auf noch höhere Taktfrequenzen des digitalen Signalprozessors (DSP). Eingehendere Untersuchungen [55] bestätigen, dass sich bei intelligenter Wahl des Addier- und Multiplizierverfahrens sowie der Zahlendarstellung ein derartiger DSP sogar mit Standardzellen aus einer 65 nm CMOS-Technologie bei mehr als 1 GHz Taktfrequenz realisieren lässt. Die spätere Zieltaktfrequenz muss anhand von Systemparametern und weiteren Rahmenbedingungen wie Kanallänge, DA-Wandlerrate, Dispersionskoeffizient der Glasfasern und daraus resultierende Schutzintervalllänge sowie IFFT-Blockgröße festgelegt werden. Zur weiteren Erhöhung der Datenrate unter Beibehaltung der Symbolrate wird als Modulation der Unterträger eine Quadraturamplitudenmodulation (QAM) vorgeschlagen. Durch Polarisationsmultiplex lässt sich die Datenrate noch einmal verdoppeln. Die Verarbeitung komplexwertiger Signale stellt auch die optische Seite vor neue Herausforderungen, da hier die Anwendung von Mach-Zehnder-Modulatoren als auch von Polarisationsplittern erforderlich wird.

Durch die Anwendung des in dieser Arbeit vorgestellten Viterbi-Entzerrers ergeben sich ganz neue Möglichkeiten in der optischen Datenkommunikation. Mit geringem Aufwand lassen sich vorhandene langsame Übertragungsstrecken kostengünstig auf Datenraten von 43 Gbit/s aufrüsten. Beim Kostenargument ist die hier angewandte Direktmodulation gegenüber komplexen Modulationsverfahren wie z.B. einer Kombination aus QAM und OFDM, wie sie für 100 G-Systemen geplant ist, deutlich im Vorteil.

Der in dieser Arbeit vorgestellte Viterbi-Entzerrer ist der weltweit erste in CMOS hergestellte Chip, der bei einer Datenrate von 32 Gbit/s erfolgreich vermessen ist und hinsichtlich Chipflächenbedarf und Leistungsaufnahme bisheriger Veröffentlichungen deutlich überlegen ist. Simulationen und rudimentäre Messungen lassen vorhersagen, dass eine Funktion bei der Zieldatenrate von 43 Gbit/s gegeben ist. Stellt sich heraus, dass der Chip auch bei der vollen Datenrate funktioniert, vergrößert sich der Abstand zur Konkurrenz bezüglich Leistungsaufnahme und Flächenbedarf noch weiter.

Liste eigener Veröffentlichungen

1. T. Veigel, F. Buchali, A. Bindel, F. Lang, T. Alpert, P. Mata, A. Botey, M. Grözing, S. Seemann, M. Berroth, „Entwurf eines elektronischen Entzerrers mit Maximum-Likelihood-Sequence-Estimation für Kanäle mit 43 Gbit/s serieller Eingangsdatenrate“, U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V. Kleinheubacher Tagung 2007, 24. bis 27. September 2007, Miltenberg, Deutschland.
2. T. Alpert, M. Schmidt, I. Dettmann, T. Veigel, M. Grözing, M. Berroth, „Concept for a 12-bit Digital Bandpass Delta-Sigma Modulator for Power Amplifier Applications“, ESSCIRC 2008 Fringe (European Solid-State Circuits Conference Fringe Event), 15. bis 19. September 2008, Edinburgh, Schottland, UK
3. T. Veigel, M. Grözing, M. Berroth, F. Buchali, „Design of a Viterbi Equalizer Circuit for Data Rates up to 43 Gb/s“, ESSCIRC 2009 Fringe (European Solid-State Circuits Conference Fringe Event), 14. bis 18. September 2009, Athen, Griechenland.
4. M. Schmidt, T. Veigel, S. Haug, M. Grözing, M. Berroth, „Low Latency Architectures of a Comparator for Binary Signed Digits in a 28-nm CMOS Technology“, U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V. Kleinheubacher Tagung 2010, 4. bis 6. Oktober 2010, Miltenberg, Deutschland
5. T. Veigel, T. Hipp, M. Grözing, M. Berroth, „Schnelle komplexe Multiplizierer zur Umsetzung der schnellen diskreten Fourier-Transformation für OFDM-Sender mit Datenraten über 100 Gbit/s“, U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V. Kleinheubacher Tagung 2010, 4. bis 6. Oktober 2010, Miltenberg, Deutschland

Literaturliste

- [1] S. Kieckbusch, S. Ferber, H. Rosenfeldt, R. Ludwig, C. Boerner, A. Ehrhardt, E. Brinkmeyer, H.G. Weber, „Adaptive PMD compensator in 160 Gb/s DPSK transmission over installed fiber“, *Optic. Fiber Comm. Conf., OFC, PDP-31*, Feb. 2004.
- [2] H. Bülow, F. Buchali, W. Banmert, R. Ballentin, T. Wehren, „PMD Mitigation at 10 Gbit/s using linear and nonlinear integrated electronic equalizer circuits“, *Electronic Lett.* vol. 36, no. 2, pp. 163-164, 2000.
- [3] U. Reimers, „Digitale Fernsehtechnik“, Berlin: Springer Verlag, 1995.
- [4] Standard IEEE 802.3z
- [5] Standard IEEE 802.3ab
- [6] H. Klar, „Integrierte Digitale Schaltungen MOS / BICMOS“, Berlin: Springer Verlag, 1996.
- [7] G. Fettweis, „Parallelisierung des Viterbi-Decoders: Algorithmus und VLSI-Architektur“, Dissertation, Düsseldorf: VDI Verlag, 1990.
- [8] „Circuits Multi-Projets“[®], CMP, 90 nm CMOS-Technologie von ST Microelectronics, Stand Juli 2008.
- [9] M. Nakamura, H. Nosaka, M. Ida, K. Kurishima, M. Tokumitsu, „Electrical PMD equalizer ICs for a 40-Gbit/s transmission“, *OFC 2004*, Los Angeles, USA, California, Februar 2004
- [10] B. Franz, D. Rösener, F. Buchali, H. Bülow, G. Veith, „Adaptive Electronic Feed-Forward Equaliser and Decision Feedback Equaliser for the Mitigation of Chromatic Dispersion and PMD in 43 Gbit/s Optical Transmission Systems“, *ECOC 2006*, Cannes, France, September 2006
- [11] F. Buchali, H. Bülow, „Adaptive PMD Compensation by Electrical and Optical Techniques“, *Journal of Lightwave Technology*, Vol. 22, No. 4, April 2004, pp. 1116-1126.
- [12] H. Dawid, G. Fettweis and H. Meyr, „A CMOS IC for Gb/s Viterbi Decoding: System Design and VLSI Implementation“, *IEEE. Transactions on Very Large Scale Integration Systems*, vol. 4, March 1996, pp. 17-31.
- [13] Peter J. Black and Teresa H.-Y. Meng, „A 1-Gb/s, Four-State, Sliding Block Viterbi Decoder“, *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 797–805, June 1997.
- [14] C. Fludger, S. Langenbach, N. Stojanovic, A. Färbert, C. Dorschky, T. Kupfer, C. Schulien, "Enabling Open Tolerant Networks," *ECOC Workshop*, 2005.
- [15] M. Anders, S. Mathew, S. Hsu, R. Krishnamurthy, S. Borkar, „A 1.9 Gb/s 358 mW 16-256 State Reconfigurable Viterbi Accelerator in 90 nm CMOS“, *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 214-222, Jan. 2008.

- [16] Draka Communications, Datenblatt "Single-Mode Optical Fiber (SMF)" für Telekommunikationsanwendungen, Produkt Typ: G.652.B.
- [17] H. Bülow, W. Baumert, H. Schmuck, F. Mohr, T. Schulz, F. Kuppers, W. Weiershausen, „Measurement of the maximum speed of PMD fluctuation in installed field fiber”, Optical Fiber Communication Conference, and the International Conference on Integrated Optics and Optical Fiber Communication. OFC/IOOC 1999, Vol 2, pp. 83-85.
- [18] H. Bülow, G. Veith, „Temporal dynamics of error-rate degradation induced by polarization mode dispersion fluctuation of a field fiber link,” in Proc. ECOC 97, vol. 1, Edinburgh, Scotland, pp. 115-118.
- [19] H. Rosenfeldt, R. Ulrich, U. Feiste, R. Ludwig, H.G.Weber, A. Ehrhardt, „First Order PMD-Compensation in a 10 Gbit/s NRZ Field Experiment Using a Polarimetric Feedback-Signal”, Proc. Europ. Conf. on Opt. Comm. (ECOC), Nice, France Vol. II (WeC3.2), pp. 134-135, 1999.
- [20] D. Fritzsche, M. Paul, L. Schuerer, A. Ehrhardt, D. Breuer, W. Weiershausen, N. Cyr, H. Chen, G.W. Schinn, „Measuring the Link Distribution of PMD: Field Trial Using an RS-POTDR“, OFC/NFOEC 2008, San Diego, California, USA, 24-28 Februar 2008.
- [21] F. Buchali, G. Thielecke, H. Bülow, „Anwendung von Viterbi-Entzerrern in optischen Systemen mit dominierendem ASE Rauschen“, ITG München, 2004.
- [22] F. Buchali, B. Franz, H. Bülow, „Electronic mitigation for 10, 40 and 100G“, Digest of the IEEE/LEOS Summer Topical Meetings, Acapulco, Mexico, 21-23 Juli 2008
- [23] A. J. Viterbi, „Error bounds for convolutional codes and asymptotically optimum decoding algorithm“, IEEE Transactions on Information Theory, vol. IT-13, pp. 260-269, Apr. 1967.
- [24] G. D. Forney, „The Viterbi Algorithm“, Proceedings of the IEEE, vol. 61, pp. 268-278, Mar. 1973.
- [25] H. Bülow, „Electronic Distortion Equalization for 10 and 40 Gb/s Transmission Systems“, VIOLA-Workshop-2, 22-23 März 2006, Stiftung Caesar, Bonn, Deutschland
- [26] F. Buchali, G. Thielecke, H. Bülow, „Viterbi equalizer for mitigation of distortions from chromatic dispersion and PMD at 10 Gb/s”, Proceedings of the OFC, WM5, 2004.
- [27] J. Speidel, Skriptum zur Vorlesung „Übertragungstechnik II“, Institut für Nachrichtenübertragung, Universität Stuttgart, 2001.
- [28] F. Buchali, H. Bülow, „Correlation sensitive Viterbi equalization of 10 Gb/s signals in bandwidth limited receivers”, OFC, vol. 5, Anaheim, März 2005.
- [29] J. Speidel, Skriptum zur Vorlesung „Communications III“, Institut für Nachrichtenübertragung, Universität Stuttgart, 2000.
- [30] E. W. Dijkstra, „A note on two problems in connexion with graphs”, Numerische Mathematik 1, 1959, S. 269–271.
- [31] J. B. Kruskal, „On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem“, Proceedings of the American Mathematical Society 7, pp. 48-50; 1956.

- [32] K.D. Kammeyer, „Nachrichtenübertragung“, 2. Auflage, Stuttgart: Teubner Verlag, 1996.
- [33] G. C. Clark and J. B. Cain, “Error-Correction Coding for Digital Communications,” New York: Plenum, 1981, pp. 227–264.
- [34] G. Feygin and P. G. Gulak, “Survivor sequence memory management in Viterbi decoders”, Tech. Rep. CSRI-252, University of Toronto, Jan. 1991.
- [35] K.-H Tzou and J. G. Dunham, “Sliding block decoding of convolutional codes”, IEEE Trans. Commun., vol. COM-29, pp. 1401–1403, Sept. 1981.
- [36] A. J. Viterbi and J. K. Omura, “Principles of Digital Communication and Coding”, New York: McGraw-Hill, 1979, pp. 258-261.
- [37] F. Buchali, “Viterbi equalizer algorithm”, Technisches Meeting Universität Stuttgart, September 2005.
- [38] H. T. Kung, „Why Systolic Architectures?“, IEEE Computer Magazine, Januar 1982, pp. 37-46.
- [39] S. Seemann, „Vergleich verschiedener Logikschaltungskonzepte in CMOS-Technologie“, Diplomarbeit am Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2005.
- [40] K. Hoffmann, „VLSI-Entwurf, Modelle und Schaltungen“, 4. Auflage, Oldenbourg Verlag, 1998.
- [41] M. Berroth, „Impuls- und Digitaltechnik I“, Skript zur Vorlesung, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2009.
- [42] A. Bindel, „Entwicklung einer Branch-Metric-Unit für einen parallelen Hardware-Viterbi-Decoder in 90 nm-CMOS-Technologie“, Diplomarbeit am Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2006.
- [43] F. Lang, „Design einer Add-Compare-Select-Unit für einen parallelen Hardware-Viterbi-Decoder in 90 nm CMOS-Technologie“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2007.
- [44] K. Kumagai, „A new SRAM Cell design using 0.35 μm CMOS/ SIMOX technology“, IEEE International SOI Conference, Okt. 1997.
- [45] C. Wang, „An SRAM Design Using Dual Threshold Voltage Transistors and Low-Power Quenchers“, IEEE Journal of Solid-State Circuits, vol. 38, no. 10, Oct 2003.
- [46] B. Razavi, „Design of Analog CMOS Integrated Circuits“, McGraw-Hill Higher Edition, 2001.
- [47] M. Grözing, D. Moser, B. Philipp, M. Kabbab, M. Berroth, „High Speed CMOS Circuits – Analog und Digitalschaltungen in Stromschaltertechnik für hochbitratige serielle Empfänger“, EEefCOM Ulm, 2004.
- [48] T. H. Lee, The design of CMOS radio-frequency integrated circuits, Cambridge University Press, Cambridge, 1998.

- [49] T. Alpert, „Redesign einer Branch-Metric-Unit für einen parallelen Hardware-Viterbi-Decoder in 90 nm-CMOS-Technologie“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2007.
- [50] C. Ruppert, „Crosstalk Effects on IC Interconnect Structures“, Diplomarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Uni Stuttgart, Telecom Paris, Motorola, 2003.
- [51] T. Devoivre¹, M. Lunenborg², C. Julien¹, J-P. Carrere¹, P. Ferreira¹, W.J. Toren², A. VandeGoor², P. Gayet¹, T. Berger¹, O. Hinsinger¹, P. Vannier¹, Y. Trouiller³, Y. Rody², P.-J. Goirand¹, R. Palla¹, I. Thomas¹, F. Guyader¹, D. Roy¹, B. Borot¹, N. Planes¹, S. Naudet¹, F. Pico¹, D. Duca¹, F. Lalanne¹, D. Heslinga², M. Haond¹, „Validated 90nm CMOS Technology Platform with Low-k Copper Interconnects for Advanced System-on-Chip (SoC)“, ¹STMicroelectronics, Crolles, France, ²PHILIPS Semiconductors, Crolles, France, ³LETI, Grenoble, France, Proceedings of the 2002 IEEE International Workshop on Memory Technology, Design and Testing (MTDT 2002), pp. 157-162.
- [52] H. Kapp, „CMOS-Schaltungstechnik für Integrierte Viterbi-Decoder“, Dissertation, Institut für Mikroelektronik, Technische Universität Berlin, 1992.
- [53] N. Weste, K. Eshraghian, „Principles of CMOS VLSI Design“, Second Edition, Addison-Wesley, 1993.
- [54] T. Hipp, „Entwurf einer Butterflyeinheit für eine inverse schnelle Fourier-Transformation“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2010.
- [55] B. Fischbach, „VHDL-Entwurf schneller Butterflies für die Inverse Schnelle Fourier-Transformation (IFFT) und Umsetzung der IFFT mit Standardzellen“, Bachelorarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2011
- [56] H. W. Lang, „Algorithmen in Java“, 2. Auflage, Oldenbourg Wissenschaftsverlag, 2006.
- [57] R. Brent, H. Kung, „A Regular Layout for Parallel Adders“, IEEE Transactions on Computers, vol. C-31, no. 3, pp. 260-264, März 1982.
- [58] J. Rabaey: „Digital Integrated Circuits“, Prentice Hall Inc., 1996.
- [59] Y. Pai, Y. Chen „The Fastest Carry Lookahead Adder“, Proceedings of The Second IEEE International Workshop on Electronic Design, Test and Applications, DELTA 2004.
- [60] S. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, “A 4 GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90 nm CMOS,” IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 44-51, Jan. 2005.
- [61] P. Kühn, „Technische Informatik I“, Skript zur Vorlesung, Institut für Kommunikationsnetze und Rechnerarchitekturen, Universität Stuttgart, 2000.
- [62] F. Maier, „Design einer Select-Minimum-Unit für einen parallelen Hardware-Viterbi-Entzerrer in einer 90 nm CMOS-Technologie“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2008.

- [63] P. Mata, „Entwicklung einer Trace-Back-Unit für einen parallelen Hardware-Viterbi-Dekoder in 90 nm-CMOS-Technologie“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2007.
- [64] F. Fertig, „Entwicklung eines Demultiplexers und Multiplexers in CML-Technik für hohe Datenraten in 90 nm CMOS-Technologie“, Wahlstudienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2009.
- [65] B. Razavi, „Design of Integrated Circuits for Optical Communications“, McGraw-Hill Higher Edition, 2003.
- [66] M. Berroth, Vorlesung „Entwurf integrierter Schaltungen“, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart.
- [67] Datenblatt zum Substrat RF-60A, Taconic.
- [68] Datenblatt: „Intel® Core™2 Duo Processors and Intel® Core™2 Extreme Processors for Platforms Based on Mobile Intel® 965 Express Chipset Family“, Document Number: 316745-005, Intel Corporation.
- [69] <http://ark.intel.com/Product.aspx?id=29760>, abgerufen am 02.06.2011
- [70] Xilinx: ML42x User Guide, „Virtex-4 FX FPGA RocketIO Characterization Platform“ – UG087.
- [71] S. Jakobs: „Entwicklung einer FPGA Messumgebung mit Multi-Gigabit Datenraten“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2009.
- [72] D. Ferenci, M. Berroth „A 100 Gigabit Measurement System with State of the Art FPGA Technology for Characterization of High Speed ADCs and DACs“, 6th Conference on Ph.D. Research in Microelectronics & Electronics, 18 – 21 Juli 2010, Berlin, Germany.
- [73] T. Veigel, F. Buchali, A. Bindel, F. Lang, T. Alpert, P. Mata, A. Botey, M. Grözing, S. Seemann, M. Berroth, „Entwurf eines elektronischen Entzerrers mit Maximum-Likelihood-Sequence-Estimation für Kanäle mit 43 Gbit/s serieller Eingangsdatenrate“, Kleinheubacher Tagung des U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V., Miltenberg, Deutschland, Sept. 2007.
- [74] T. Veigel, M. Grözing, M. Berroth, F. Buchali, „Design of a Viterbi Equalizer Circuit for Data Rates up to 43 Gb/s“, ESSCIRC Fringe (European Solid-State Circuits Conference Fringe Event), Athen, Griechenland, Sept. 2009.
- [75] B. Ehni, „Bestimmung von Koeffizienten für einen digitalen Entzerrer“, Studienarbeit, Institut für Elektrische und Optische Nachrichtentechnik, Universität Stuttgart, 2009.
- [76] Smith, Michael John Sebastian: „Application-Specific Integrated Circuits“, Addison-Wesley, 1997
- [77] J. Cooley and J. Tukey, „An algorithm for the machine calculation of complex Fourier series, Math Comput.“, pp. 297-301, 1965.

-
- [78] C. S. Wallace, „A Suggestion for a Fast Multiplier”, IEEE Transactions On Electronic Computers, 1964.
- [79] A. D. Booth, „A Signed Binary Multiplication Technique”, Quart. Journ. Mech. and Applied Math., Vol. IV Pt. 2, 1951.
- [80] Y.-T. Pai, Y.-K. Chen, „The Fastest Carry Lookahead Adder”, Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04), 2004.
- [81] N. Besli, R. G. Deshmukh, „A Novel Redundant Binary Signed-Digit (RBSD) Booth's Encoding”, Proceedings IEEE Southeastcon, 2002.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Angestellter am Institut für Elektrische und Optische Nachrichtentechnik (INT) an der Universität Stuttgart.

Ich danke meinem Doktorvater Prof. Dr.-Ing. Manfred Berroth für die wissenschaftliche Betreuung und Unterstützung meiner Arbeit sowie für das mir entgegengebrachte Vertrauen.

Herrn Prof. Dr.-Ing. Joachim Speidel danke ich für die Übernahme des Mitberichts.

Bei allen meinen Kollegen bedanke ich mich für die wertvollen Diskussionen, Anregungen und die Unterstützung, die zu einem erfolgreichen Gelingen dieser Arbeit beigetragen haben. Weiterhin danke ich meinen Kollegen für die Unterstützung bei der täglichen Institutsarbeit.

Mein Dank gilt den Studenten, die in ihren zahlreichen und teilweise sehr umfangreichen Studien-, Diplom- und Hiwi-Arbeiten wertvolle Beiträge zu dieser Arbeit geliefert haben.

Für Anregungen und das Begleiten der Arbeit bedanke ich mich bei Herrn Buchali und seinen Kollegen von Alcatel-Lucent.

Schließlich danke ich meinen Eltern für das Korrekturlesen und die persönliche Unterstützung.

Lebenslauf

Name	Thomas Veigel
Geburtsdatum	1. April 1977
Geburtsort	Stuttgart
Sept. 1984 – Juni 1988	Hermann-Butzer-Schule Schwieberdingen Grundschule
Aug. 1988 – Juni 1997	Hans-Grüniger-Gymnasium Markgröningen Abschluss: Abitur
Juli 1997 – Juli 1998	Klinik Markgröningen Zivildienst
Okt. 1998 – Okt. 2004	Universität Stuttgart Studium der Elektrotechnik, Fachrichtung: Technische Informatik, Informationsverarbeitung Abschluss: Diplomingenieur
Seit Nov. 2004	Universität Stuttgart Institut für Elektrische und Optische Nachrichtentechnik Wissenschaftlicher Angestellter